

Wonderware® FactorySuite™ IndustrialSQL™ Server

Справочное руководство

Редакция F
Июль 1999

Wonderware Corporation

Все права защищены. Никакая часть настоящей документации не может воспроизводиться, храниться в информационной системе или передаваться любым способом, электронным или механическим, или путем фотокопирования, записи или как-то иначе без предварительного письменного разрешения Wonderware Corporation. Использование содержащейся здесь информации не влечет за собой никакой ответственности, связанной с авторскими или патентными правами. Хотя при подготовке этой документации авторами и издателями были приложены все усилия, они не несут ответственность за возможные ошибки или неточности, равно как за возможный ущерб, причиненный в результате использования содержащейся здесь информации.

Содержащаяся здесь информация может быть изменена без предупреждения и не носит характера обязательств со стороны корпорации Wonderware. Описанное в документации программное обеспечение предоставляется по соглашению о лицензии или неразглашении. Это программное обеспечение может использоваться только на условиях данных соглашений.

© 1999 Wonderware Corporation. Все права сохранены.

100 Technology Drive

Irvine, CA 92618

U.S.A.

(949) 727-3200

<http://www.wonderware.com>

Товарные знаки

Все упоминаемые в этой книге названия, известные как товарные или сервисные знаки, помечены должным образом. Корпорация Wonderware не может гарантировать точность этой информации. Использование любых названий в этой книге не следует считать преднамеренным нарушением каких-либо товарных или сервисных знаков.

Wonderware, InTouch и FactorySuite Web Server являются зарегистрированными товарными знаками Wonderware Corporation.

FactorySuite, Wonderware FactorySuite, WindowMaker, WindowViewer, SQL Access Manager, Recipe Manager, SPCPro, DBDump, DBLoad, HDMerge, HistData, Wonderware Logger, Alarm Logger, InControl, InTrack, InBatch, IndustrialSQL, FactoryOffice, FactoryFocus, License Viewer, Scout, SuiteLink и NetDDE являются товарными знаками Wonderware Corporation.

Оглавление

Глава 1 - Введение	1-1
Структуры данных IndustrialSQL Server	1-2
Базы данных IndustrialSQL Server	1-2
База данных Holding	1-2
База данных Runtime	1-3
Привилегии доступа к базам данных	1-4
Таблицы IndustrialSQL Server	1-6
Таблицы конфигурации	1-6
Таблицы расширения (удаленные таблицы)	1-7
Расширения временного домена Industrial-SQL	1-10
Глава 2 - Тематические области и общие	
таблицы	2-1
Тематические области	2-2
Общие таблицы	2-3
Таблица AnalogTag	2-4
Таблица ComplexTag	2-5
Таблица DcreteTag	2-6
Таблица EventHistory	2-6
Таблица EventTag	2-7
Таблица StringTag	2-9
Таблица Tag	2-10
Таблица TagRef	2-12
Таблица Topic	2-13
Таблица UserDetail	2-14
Глава 3 - Таблицы хранения данных	3-1
Таблицы хранения данных	3-2
Таблица AnalogHistory	3-4
Таблица AnalogLive	3-6
Таблица AnalogTag	3-6
Таблица AnalogWideHistory	3-7
Таблица Annotation	3-9
Таблица ComplexHistory	3-10
Таблица ComplexTag	3-10
Таблица DiscreteHistory	3-11
Таблица DiscreteLive	3-13
Таблица DiscreteTag	3-14
Таблица DiscreteWideHistory	3-14
Таблица EventHistory	3-15
Таблица EventTag	3-15
Таблица HistoryBlock	3-16
Таблица ManualAnalogHistory	3-17
Таблица ManualDiscreteHistory	3-18
Таблица StringHistory	3-18
Таблица StringLive	3-19
Таблица StringTag	3-19
Таблица Tag	3-20
Таблица UserDetail	3-20
Глава 4 - Таблицы хранения событий	4-1
Таблицы хранения событий	4-2
Таблица ActionType	4-4
Таблица AnalogSnapshot	4-5

Таблица DetectorType.....	4-6
Таблица DiscreteSnapshot.....	4-7
Таблица EventHistory.....	4-7
Таблица EventTag.....	4-7
Таблица EventTagPendingDelete.....	4-7
Таблица Frequency.....	4-8
Таблица SnapshotTag.....	4-8
Таблица SQLTemplate.....	4-9
Таблица StringSnapshot.....	4-10
Таблица Tag.....	4-10
Таблица TimeDetectorDetail.....	4-11
Таблица TimeDetectorDetailPendingDelete.....	4-12
Глава 5 - Таблицы интеграции InTouch	5-1
Таблицы интеграции InTouch	5-2
Таблица InTouchNode	5-3
Таблица InTouchSpecific.....	5-5
Таблица Tag.....	5-5
Таблица TopicImportInfo.....	5-6
Глава 6 - Таблицы отслеживания изменений	6-1
Таблицы отслеживания изменений	6-2
Таблица InSQLSysObjects	6-3
Таблица ModBinary.....	6-3
Таблица ModBit.....	6-4
Таблица ModChar.....	6-4
Таблица ModDateTime	6-5
Таблица ModDecimal	6-5
Таблица ModInteger	6-6
Таблица ModLog.....	6-6
Таблица ModReal	6-7
Таблица ModText.....	6-7
Глава 7 - Таблицы именных пространств и групп	7-1
Таблицы именных пространств и групп	7-2
Таблица GroupTagList	7-4
Таблица NameSpaceIcons	7-4
Таблица PrivateGroupTag	7-4
Таблица PrivateNameSpace	7-5
Таблица PublicGroupTag.....	7-5
Таблица PublicNameSpace.....	7-6
Таблица ServerList	7-6
Таблица SystemNameSpace	7-7
Таблица Tag.....	7-7
Таблица TagGroup	7-7
Таблица TagRef	7-8
Таблица UserDetail.....	7-8
Глава 8 - Таблицы сводок.....	8-1
Таблицы сводок.....	8-2
Таблица CalcType.....	8-4
Таблица SummaryData.....	8-4
Таблица SummaryHistory	8-5
Таблица SummaryOperation	8-7
Таблица SummaryTagList	8-8
Таблица Tag.....	8-8

Глава 9 - Таблицы системной

конфигурации 9-1

Таблицы системной конфигурации	9-2
Таблица IOServer.....	9-4
Таблица IOServerType.....	9-5
Таблица NamedSystemParameter	9-6
Таблица QualityMap.....	9-6
Таблица StorageLocation.....	9-7
Таблица StorageNode	9-8
Таблица Tag.....	9-10
Таблица Topic.....	9-10
Таблица UserDetail	9-10
Таблица WideTableDictionary.....	9-10

Глава 10 - Таблицы определения тэгов..... 10-1

Таблицы определения тэгов	10-2
Таблица AnalogTag	10-4
Таблица ComplexTag.....	10-4
Таблица Context.....	10-4
Таблица Deviation	10-5
Таблица DiscreteTag	10-6
Таблица EngineeringUnit	10-6
Таблица EventTag	10-6
Таблица Limit.....	10-7
Таблица LimitName	10-7
Таблица Message	10-8
Таблица RateOfChange	10-9
Таблица StringTag	10-9
Таблица Tag.....	10-9
Таблица TagRef	10-9
Таблица Topic.....	10-9

Глава 11 - Удаленные таблицы OLE DB 11-1

Удаленные таблицы OLE DB.....	11-2
Таблица AnalogHistory (OLE DB).....	11-3
Таблица AnalogLive (OLE DB).....	11-4
Таблица AnalogWideHistory (OLE DB).....	11-5
Таблица DiscreteHistory (OLE DB)	11-7
Таблица DiscreteLive (OLE DB)	11-8
Таблица DiscreteWideHistory (OLE DB)	11-9
Таблица HistoryBlock (OLE DB)	11-11
Таблица StringLive (OLE DB).....	11-13
Столбцы для запросов с параметрами	11-14

Глава 12 - Хранимые процедуры

IndustrialSQL Server 12-1

Хранимые процедуры IndustrialSQL Server	12-2
Системные хранимые процедуры	12-2
ww_ActionStringSelect	12-2
ww_AnalogConfig.....	12-2
ww_AnalogCount.....	12-4
ww_AnalogTagDelete	12-4
ww_AnalogTagInsert	12-5
ww_AnalogTagSelect	12-5
ww_AnalogTagUpdate.....	12-7
ww_AnnotationDelete	12-8
ww_AnnotationInsert.....	12-9
ww_AnnotationSelect.....	12-9

ww_AnnotationUpdate.....	12-10
ww_CheckClientVersion.....	12-10
ww_CheckWhichDb.....	12-12
ww_dbCheck.....	12-12
ww_DeleteOlderEvents.....	12-12
ww_DeleteOlderStrings.....	12-13
ww_DeleteOlderSummaries.....	12-13
ww_DetectorStringSelect.....	12-13
ww_DiscreteConfig.....	12-14
ww_DiscreteCount.....	12-14
ww_DiscreteTagDelete.....	12-14
ww_DiscreteTagInsert.....	12-16
ww_DiscreteTagSelect.....	12-16
ww_DiscreteTagUpdate.....	12-18
ww_EngineeringUnitDelete.....	12-18
ww_EngineeringUnitInsert.....	12-20
ww_EngineeringUnitSelect.....	12-20
ww_EngineeringUnitUpdate.....	12-21
ww_EventHistory.....	12-21
ww_EventHistoryInsert.....	12-23
ww_EventSnapshot.....	12-24
ww_EventTagDelete.....	12-24
ww_EventTagDetail.....	12-25
ww_EventTagInsert.....	12-26
ww_EventTagSelect.....	12-27
ww_EventTagSelectAll.....	12-28
ww_EventTagSelectDeleted.....	12-28
ww_EventTagSelectDisabled.....	12-29
ww_EventTagSelectInserted.....	12-30
ww_EventTagSelectUpdated.....	12-30
ww_EventTagUpdate.....	12-31
ww_GetDbRevision.....	12-31
ww_GetInSQLProcedureNames.....	12-32
ww_GetLastTagKey.....	12-33
ww_InsertStringSnapshot.....	12-34
ww_InSQLNSExpand.....	12-34
ww_InSQLStatusSelect.....	12-35
ww_InSQLStatusSet.....	12-35
ww_IOServerDelete.....	12-37
ww_IOServerInsert.....	12-38
ww_IOServerSelect.....	12-39
ww_IOServerUpdate.....	12-40
ww_LoadInSQLProcedureBody.....	12-41
ww_MessageDelete.....	12-41
ww_MessageInsert.....	12-41
ww_MessageSelect.....	12-42
ww_MessageUpdate.....	12-42
ww_ModLogStatus.....	12-42
ww_PrivateNSAddGroup.....	12-44
ww_PrivateNSAddLeaf.....	12-44
ww_PrivateNSDeleteGroup.....	12-44
ww_PrivateNSDeleteLeaf.....	12-46
ww_PrivateNSExpand.....	12-47
ww_PrivateNSSelect.....	12-47
ww_PrivateNSUpdateGroup.....	12-48
ww_PublicNSAddGroup.....	12-49
ww_PublicNSAddLeaf.....	12-50
ww_PublicNSDeleteGroup.....	12-50

ww_PublicNSDeleteLeaf	12-50
ww_PublicNSExpand	12-52
ww_PublicNSSelect	12-52
ww_PublicNSUpdateGroup	12-53
ww_RedirectToInTouch	12-54
ww_SetAISamples	12-55
ww_SnapToSummary	12-56
ww_SpaceManager	12-57
ww_StorageLocationSelect	12-58
ww_StorageLocationUpdate	12-58
ww_StringConfig	12-58
ww_StringCount	12-60
ww_StringTagDelete	12-60
ww_StringTagInsert	12-61
ww_StringTagSelect	12-61
ww_StringTagUpdate	12-63
ww_SummaryDetail	12-64
ww_SummaryOperationDelete	12-64
ww_SummaryOperationInsert	12-66
ww_SummaryOperationSelect	12-67
ww_SummaryOperationUpdate	12-68
ww_SummaryTagListDelete	12-69
ww_SummaryTagListInsert	12-70
ww_SummaryTagListSelect	12-70
ww_SummaryTagListUpdate	12-71
ww_SystemNSExpand	12-71
ww_SystemNSExpand2	12-73
ww_TagInfo	12-73
ww_TagType	12-73
ww_TimeDetectorDetailInsert	12-75
ww_TimeDetectorDetailSelect	12-75
ww_TimeDetectorDetailUpdate	12-76
ww_TopicDelete	12-76
ww_TopicInsert	12-77
ww_TopicSelect	12-77
ww_TopicUpdate	12-78
ww_UserDetailUpdate	12-78
Пользовательские хранимые операции	12-80
ww_AddTag	12-80
ww_AnalogDetail	12-80
ww_Annotation	12-82
ww_DeleteTag	12-82
ww_DiscreteDetail	12-82
ww_ResetDefaults	12-83
ww_SetAnalogCyclic	12-83
ww_SetContinuous	12-83
ww_SetDiscreteCyclic	12-85
ww_SetDuration	12-85
ww_SetEdgeDetection	12-87
ww_SetEndDate	12-88
ww_SetResolution	12-88
ww_SetRowCount	12-89
ww_SetTimeDeadband	12-89
ww_SetValueDeadband	12-91
ww_StringDetail	12-91
ww_UserAccessLevelSelect	12-92
Расширенные хранимые процедуры	12-93
xp_proclist	12-95

xp_AnalogHistory	12-95
xp_AnalogHistoryDelta	12-96
xp_AnalogWideHistory	12-98
xp_AnalogWideHistoryDelta	12-99
xp_DiscreteHistory	12-100
xp_DiscreteHistoryDelta	12-101
xp_DiscreteWideHistory	12-102
xp_DiscreteWideHistoryDelta	12-103
xp_DiskCopy	12-104
xp_NewHistoryBlock	12-104
xp_SetStorageTimeDeadband	12-105
xp_SetStorageValueDeadband	12-105
Создание хранимых процедур	12-106
Глава 13 - Представления данных	13-1
Представления данных	13-2
Представления данных системы управления событиями	13-2
v_EventSnapshot	13-2
v_SummaryData	13-5
Представления для удаленных таблиц поставщика данных	
OLE DB	13-6
Синтаксис представлений	13-6
Глава 14 - Выборка данных	14-1
Синтаксис запроса к серверу базы данных IndustrialSQL ODS	14-2
Ограничения для запросов через сервер БД IndustrialSQL ODS	14-2
Синтаксис запросов для поставщика данных INSQL OLE DB	14-3
Использование имени связанного сервера	14-3
Использование представления поставщика данных	
InSQL OLE DB	14-4
Использование функции OPENQUERY	14-4
Использование функции OPENROWSET	14-4
Использование расширений реального времени IndustrialSQL ...	14-5
Ограничения для запросов через OLE DB	14-6
Типичные запросы IndustrialSQL Server	14-7
Выбор значений столбцов для одиночной таблицы	14-7
Использование выражения LIKE	14-8
Использование сложного выражения WHERE	14-9
Использование псевдонимов таблиц	14-10
Запрос к "архивной" таблице расширения	14-11
Запрос к "живой" архивной таблице	14-12
Запрос к "широкой" архивной таблице	14-13
Запросы к широким таблицам дельта-методом	14-15
Использование арифметической функции	14-16
Использование функций даты	14-17
Использование функции агрегации	14-18
Связывание таблиц сервера SQL выражением JOIN	14-19
Связывание архивных таблиц выражением JOIN	14-19
Создание и запрос аннотаций	14-20
Определение границ	14-21
Использование расширений временного домена Industrial-SQL	14-22
Указание количества строк	14-22
Указание разрешения	14-25
Глобальная установка стартовой и конечной дат	14-27
SELECT INTO из архивной таблицы	14-29
Продолжительное получение значений "живых" данных	14-31
Циклическая выборка	14-32
Дельта-выборка	14-33
Установка временной мертвой зоны для дельта-выборки	14-35

Установка значения мертвой зоны для дельта-выборки	14-38
Установка временной мертвой зоны и значения мертвой зоны для выборки	14-40
Указание разрешения, счетчика строк и метода выборки в одном запросе.....	14-41
Использование операторов сравнения при дельта-выборке	14-42
Использование операторов сравнения с циклическим методом и счетчиком строк.....	14-48
Использование операторов сравнения с циклическим методом и разрешением	14-52
Распознавание событий	14-56
Определение края для аналоговых тэгов.....	14-56
Определение края для дискретных тэгов.....	14-61
Связи между таблицами Manual History и Dynamic History.....	14-68
Использование курсоров сервера	14-69
Неподдерживаемый синтаксис SQL для поставщика данных INSQL OLE DB.....	14-71
Синтаксис, неподдерживаемый запросами из четырех частей .	14-71
Синтаксис, не поддерживаемый в OPENQUERY.....	14-73
Приложение А - Диаграммы таблиц	A-1
Приложение В - Толковый словарь	B-1
Приложение I - Предметный указатель.....	I-1

Предисловие

Это *Справочное руководство IndustrialSQL Server* описывает модель базы данных системы IndustrialSQL™ Server. Здесь описаны все ячейки базы данных и все отношения между ними. Эти отношения и структуры данных очень важно усвоить для успешной работы с IndustrialSQL Server и создания эффективных приложений-клиентов.

Данное руководство также предоставляет информацию по выборке данных из базы данных IndustrialSQL Server с помощью диалогов SQL.

Что нового в IndustrialSQL Server 7.1

IndustrialSQL Server выпуска FactorySuite 2000 7.1 обеспечивает новые, расширенные возможности сервера. Главные усовершенствования сервера IndustrialSQL включают в себя поддержку Microsoft SQL Server 7.0 и улучшения в системе сводок и системе управления событиями.

Поддержка Microsoft SQL Server 7.0

IndustrialSQL Server полностью использует все новые возможности Microsoft SQL Server 7.0. Эти возможности включают улучшения в производительности и в процессе управления хранением информации, а также усовершенствование приложений для администрирования и обращения с базой данных. Вместе с тем, главным преимуществом Microsoft SQL Server 7.0 (для IndustrialSQL Server) является усовершенствованный обработчик запросов.

В более ранних версиях IndustrialSQL Server грамматически проверял все поступающие запросы клиента, передавал запросы на некоторые типы данных в Microsoft SQL Server, а запросы на данные архивных блоков выполнял с помощью собственного механизма выборки данных. Теперь Microsoft SQL Server 7.0 может успешно проверять и обрабатывать все запросы клиента, вне зависимости от того, какой тип данных запрашивается, будь то база данных SQL Server, файлы с данными, базы данных Oracle и т.п. Технология доступа к данным из различных, структурированных или неструктурированных, источников, таких как указанные выше, называется Object Linking and Embedding for Databases или OLE DB (Связывание и встраивание объектов для баз данных).

Использование улучшенного обработчика запросов Microsoft SQL Server делает объединение данных из базы данных Microsoft SQL Server и из архивных блоков гораздо легче и эффективнее. Несколько примеров улучшений:

- Повышенная возможность связывания дискретных и аналоговых таблиц.
- Поддержка курсоров базы данных. Это позволяет проводить достаточно сложные связи, такие как определение средних или накопительных аналоговых значений (например, поток) когда включен дискретный тэг (например, клапан).

Чтобы приложение-клиент получило доступ к данным через OLE DB, запрос должен быть составлен с использованием правильного синтаксиса OLE DB. К инструментам клиента IndustrialSQL, использующим OLE DB для доступа к данным, относятся: InSQL Report, InSQL Configure, а также элементы управления ActiveX IndustrialSQL —ActiveDataGrid, ActiveGraph, ActiveTagBrowser и ActiveTimeSelector.

Подробную информацию см. в главе 14, "Выборка данных".

Интеграция сводок в систему управления событиями

Система сводок теперь интегрирована в систему управления событиями и улучшена. Сводки все так же хранятся в таблицах сводок, но их выполнение осуществляется системой управления событиями. Система управления событиями теперь включает в себя временной детектор событий, который называется планировщиком. Планировщик определяет, если событие происходит в определенное время или позже по системным часам. Этот новый детектор наиболее полезен для определения времени сводок. Сводки теперь используются в системе управления событиями как особый тип операций и более гибко настраиваются пользователем; можно выбрать любое число тэгов для любого числа расчетов.

Подробную информацию см. в главе 4, "Таблицы хранения событий" и в главе 8, "Таблицы сводок".

Условные обозначения

В этом руководстве используются следующие условные обозначения:

Обозначение	Трактовка
ВЕРХНИЙ РЕГИСТР	Пути, названия файлов и синтаксис Industrial-SQL™.
СМЕШАННЫЙ регистр	Сокращенный синтаксис. Сегмент в верхнем регистре обязателен. Сегмент в нижнем регистре необязателен.
МАЛЫЕ ЗАГЛАВНЫЕ	Названия клавиш, например ESC,
полужирный	Системные процедуры, утилиты, команды и текст, вводимый пользователем.
<i>курсив</i>	Названия баз данных, таблиц, столбцов, индексов и тэгов.
Пропорциональный	Текст на экране, сообщения об ошибках и примеры.
{фигурные скобки}	Обязательные элементы синтаксиса. Сами скобки вводить в синтаксисе не нужно.
[квадратные скобки]	Необязательные элементы синтаксиса. Сами скобки вводить в синтаксисе не нужно.
(вертикальная черта)	Равнозначна "или". В синтаксисе можно ввести только один из двух элементов, разделенных вертикальной чертой.
... (многоточие)	Предшествующий элемент синтаксиса можно повторять.

Дополнительные источники информации

В дополнение к этому руководству информацию о системе IndustrialSQL Server можно почерпнуть из следующих источников.

Электронная справка по IndustrialSQL Server

Электронная справка вызывается из меню справки в любом приложении-клиенте IndustrialSQL Server.

Электронная документация по IndustrialSQL Server

Электронная документация по IndustrialSQL Server доступна на установочном компакт-диске FactorySuite. Во время установки IndustrialSQL Server вы можете выбрать установку электронной документации на жесткий диск.

Набор электронной документации IndustrialSQL Server включает электронные версии следующих книг по IndustrialSQL Server:

Руководство администратора системы FactorySuite

Стандартная книга по FactorySuite, объясняющая установку, настройку, сетевые возможности, интеграцию и стандартные компоненты.

Руководство по подготовке работы с IndustrialSQL Server

Содержит сведения об использовании базовых и расширенных возможностей IndustrialSQL Server и клиентских программ.

Руководство администратора IndustrialSQL Server

Описывает процедуры сопровождения и администрирования IndustrialSQL Server.

Руководство по клиентским инструментам IndustrialSQL Server

Описывает все имеющиеся клиентские инструменты Wonderware, используемые для получения информации из IndustrialSQL Server.

Справочное руководство IndustrialSQL Server

Предоставляет подробное описание всех таблиц баз данных IndustrialSQL Server и процедур доступа к информации.

Руководство по элементам ActiveX IndustrialSQL Server

Описывает элементы управления ActiveX IndustrialSQL и процедуры их настройки во время разработки приложения.

Техническая поддержка

Служба технической поддержки корпорации Wonderware располагает обширными возможностями, позволяющими решить множество вопросов по любым аспектам использования продуктов Wonderware. Подробные сведения о службе технической поддержки приводятся в *Руководстве администратора системы FactorySuite*.

Г Л А В А 1

Введение

Эта глава содержит краткое описание баз данных и таблиц IndustrialSQL Server, а также введение в расширения временного домена IndustrialSQL Server.

Подробную информацию по работе с расширениями временного домена см. в главе 14, "Выборка данных".

Содержание

- Структуры данных IndustrialSQL Server
- Базы данных IndustrialSQL Server
- Таблицы IndustrialSQL Server
- Расширения временного домена Industrial-SQL

Структуры данных IndustrialSQL Server

IndustrialSQL Server предстает для приложений-клиентов как реляционная база данных SQL в сети. Реляционные базы данных характеризуются табличными структурами данных, в которых определены содержащиеся в них объекты. Система IndustrialSQL Server отражает объекты типовой структуры архивных данных производства.

Определение модели типичной системы реляционной базы данных формируется системной моделью данных или моделью взаимосвязи объектов. Модель взаимосвязи объектов описывает структуры данных и отношения между ними.

В этом руководстве описывается модель взаимосвязи объектов системы IndustrialSQL Server. Определены и описаны все объекты и отношения между ними. Понимание этих структур данных и отношений между ними очень важно для формирования правильных запросов к IndustrialSQL Server.

Базы данных IndustrialSQL Server

Система управления реляционной базой данных (СУРБД), такая как IndustrialSQL Server, может охватывать множество баз данных. База данных представляет собой различные объекты, такие как:

- Таблицы
- Процедуры
- Представления
- Пользовательские представления данных
- Пользователи и группы

IndustrialSQL Server поставляется с двумя уже готовыми базами данных: база данных *Holding* и база данных *Runtime*,

Внимание! Если базы данных *Runtime* и *Holding* установлены в системе Microsoft SQL Server, в которой учитывается регистр, эти базы работают с учетом регистра. Следите за использованием правильного регистра при формировании запроса.

База данных Holding

База данных Holding служит для временного хранения данных тем и конфигураций, импортированных из узла InTouch. При импортировании этих данных из InTouch они сначала попадают в табличную структуру базы данных *Holding*. Затем они перемещаются в базу данных *Runtime*. Подробную информацию об импортировании данных из InTouch см. в *Руководстве администратора системы IndustrialSQL Server*,

База данных Runtime

База данных *Runtime* — это база данных реального времени, с которой система IndustrialSQL Server начинает сбор данных. База данных *Runtime* хранит все данные конфигурации, а также данные событий, сводок и текстовые данные для той системы IndustrialSQL Server, которая работает в текущий момент. Имя этой базы данных не может быть изменено.

База данных *Runtime* содержит данные для следующих восьми категорий:

- **Конфигурация системы.** Определяет настройки системы IndustrialSQL Server. Это включает в себя параметры настройки физических узлов системы IndustrialSQL Server, параметры, свойственные данному предприятию, и параметры настройки подключенных к системе физических устройств ввода-вывода.
- **Определения тэгов.** Определяет, какие данные должны накапливаться системой. Это включает идентификационную информацию, такую как тэги и описания, а также собираемую информацию, такую как адресные коды, факторы преобразования и единицы измерения.
- **Хранение архивных данных.** Это фактически накопленные значения, сначала значение "реального времени" или текущее значение каждого тэга ввода-вывода, а затем архивирование всех полученных данных.
- **Именные пространства и группы.** Определяет способы группировки тэгов ввода-вывода для таких объектов как алармы, окна, для управления событиями и пакетами. Здесь также определяются иерархии элементов системы, общие или личные именные пространства
- **Отслеживание изменений.** Сохраняет изменения, совершенные в столбцах базы данных.
- **Хранение событий.** Определяет операции с тэгами событий, детекторами событий и последующими действиями.
- **Сводки.** Определяет, как системой управления событиями будут делаться сводки и как будут сохраняться результаты.
- **Интеграция InTouch.** Определяет, как IndustrialSQL Server будет взаимодействовать с InTouch на предприятии. В этих таблицах хранится информация об импортированных узлах InTouch и о подробностях импортирования.

Привилегии доступа к базам данных

IndustrialSQL Server позволяет использовать средства безопасности Microsoft SQL Server. По правилам безопасности SQL Server все пользователи должны получить учетную запись, чтобы иметь доступ к базе данных. Учетная запись пользователя состоит из имени пользователя и его идентификатора Действия, разрешенные пользователю по отношению к определенной базе данных, основываются на информации в его учетной записи. Эти ограничения называются привилегиями доступа.

Чтобы упростить управление большим количеством пользователей базы данных, каждому из них можно назначить определенную роль, что равнозначно отнесению его к группе пользователей. Все члены группы получают привилегии, разрешенные для этой группы. Например, если пользователь "МэриБ" включен в группу "ОпытныеПользователи", то этот пользователь автоматически получает соответствующие этой группе привилегии доступа.

Добавляя новую группу, нужно сначала дать самой системе IndustrialSQL Server права доступа к этой группе, которые расположены в столбце AccessLevel таблицы UserDetail. Новые пользователи и группы автоматически добавляются в таблицу *UserDetail* каждый час (при условии, что запущен агент SQL). Их можно добавить и вручную с помощью процедуры **ww_UserDetailUpdate**.

IndustrialSQL Server поддерживает смешанный режим безопасности, См. подробности в *Руководстве администратора IndustrialSQL Server*,

Стандартные пользователи и группы IndustrialSQL Server

IndustrialSQL Server поставляется с несколькими уже готовыми ролями и группами пользователей, а также со стандартной моделью безопасности. Стандартная модель безопасности является первоначальным вариантом, который подходит к многим типам установок. Эта модель может быть позже изменена системным администратором.

Стандартные пользователи IndustrialSQL Server уже имеют пароль и роль, Первоначальный пароль аналогичен имени пользователя. Дополнительные пользователи и группы могут быть добавлены при помощи обычных средств Microsoft SQL Server. Стандартное описание каждой группы:

Группа	Описание
wwUser	Пользователь с правом "только чтение", имеет доступ к любым данным, но не может изменять их или как-либо влиять на ресурсы базы данных.
WwPower	Пользователю разрешается просмотр материалов, разрешение на создание объектов и внесение изменений в неосновные таблицы.
WwAdmin	Пользователь с правом на доступ и изменение любых данных и создание объектов. Не имеет права усекать таблицы и сбрасывать базу данных.
wwdbo	Владелец базы данных. Абсолютные полномочия.

Следующая таблица описывает учетные записи и группы пользователей, в которых они располагаются.

Имя пользо- вателя	Иденти- фикатор	Группа поль- зователей	Допуск
wwUser	wwUser	wwUsers	SELECT во всех таблицах. INSERT, UPDATE, DELETE в таблицах <i>GroupTagList</i> , <i>TagGroup</i> , <i>Annotation</i>
wwPower	wwPower	wwPowerUsers	CREATE таблицу CREATE представление CREATE хранимую процедуру CREATE стандарт CREATE правило SELECT во всех таблицах INSERT, UPDATE, DELETE в групповых таблицах
wwAdmin	wwAdmin	wwAdministrators	CREATE таблицу CREATE представление CREATE хранимую процедуру CREATE стандарт CREATE правило DUMP базу данных DUMP транзакцию SELECT, INSERT, UPDATE, DELETE во всех таблицах
dbo	wwdbo	Полный доступ	Полные возможности владельца базы данных.

См. подробности в *Руководстве администратора IndustrialSQL Server*.

Таблицы IndustrialSQL Server

Таблицы базы данных в IndustrialSQL Server делятся на две категории: таблицы конфигурации и таблицы расширения. IndustrialSQL Server направляет запросы в таблицы конфигурации и таблицы расширения по-разному. Этот раздел описывает основные различия между таблицами конфигурации и таблицами расширения и типы хранящейся в них информации.

Таблицы расширения для OLE DB называются удаленными таблицами. [Подробнее см. в главе 11, "Удаленные таблицы OLE DB".](#)

Система IndustrialSQL Server содержит полностью функциональный Microsoft SQL Server, благодаря чему поддерживаются все таблицы, совместимые с SQL Server. Подробную информацию по таблицам Microsoft SQL Server см. в документации Microsoft.

Таблицы конфигурации

В таблицах конфигурации содержатся данные, которые редко изменяются. В таблицах конфигурации содержится такая информация, как:

- Определения системной машины.
- Конфигурации узла хранения.
- Определения сервера ввода-вывода
- Определения тэгов
- Определения пользователей и пользовательских групп.
- Конфигурации именных пространств
- Сводки данных
- Определение детекторов событий и действий
- Хранение "снимков" данных

Эти таблицы обычно используются как ссылочные или консультационные таблицы системы IndustrialSQL Server и приложений-клиентов. Все изменения в системе IndustrialSQL Server отражаются в таблицах конфигурации.

Таблицы конфигурации являются обычными таблицами SQL Server, и содержащиеся в них данные могут обрабатываться при помощи любого кода Microsoft Transact-SQL. Таблицы конфигурации хранятся на устройстве данных базы данных. См. подробности в документации Microsoft.

Таблицы расширения (удаленные таблицы)

Поскольку в обычные функции SQL Server не входит управление хранением и получением больших объемов быстро меняющихся данных, эту функцию берет на себя система хранения данных IndustrialSQL Server, таблицы расширения и расширения временного домена Industrial-SQL. Для OLE DB эквивалентом таблиц расширения являются удаленные таблицы.

Подробнее см. главу 11, "Удаленные таблицы OLE DB".

Таблицы расширения (удаленные таблицы) являются средством, позволяющим обращаться с собранными производственными данными. Эти таблицы не являются частью обычных функций SQL Server. Стандартные таблицы SQL Server хранят данные, которые непосредственно и физически расположены в файле устройства данных. Таблица расширения (удаленная таблица), тем не менее, представляет данные как реально существующую таблицу, но таблица эта не существует на устройстве данных. Таблица расширения (удаленная таблица) есть логическая таблица, заполненная данными из файлов, созданных системой хранения.

Данные в таблице расширения (удаленной таблице) управляются при использовании обычного Transact-SQL кода, так же, как и расширения специализированного временного домена SQL, которые входят в состав IndustrialSQL Server и называются Industrial-SQL. Так как таблицы расширения (удаленные таблицы) обеспечивают дополнительную функциональность, которой нет в обычном SQL Server, то этими таблицами поддерживаются естественные ограничения, применимые к Transact-SQL. Запросы по удаленным таблицам обрабатываются системой выборки данных IndustrialSQL Server. Такие операции, как ВСТАВКА, ОБНОВЛЕНИЕ и УДАЛЕНИЕ не работают в таблицах расширения.

Подробнее см. гл. 14, "Выборка данных".

В системе IndustrialSQL Server используются следующие таблицы расширения:

- *AnalogHistory*
- *AnalogWideHistory*
- *AnalogLive*
- *DiscreteHistory*
- *DiscreteWideHistory*
- *DiscreteLive*
- *HistoryBlock*
- *StringLive*

Полученные данные в удаленной таблице могут быть представлены в трех различных форматах: архивный формат, "широкий" (wide) формат и "живой" (live) формат.

Архивные удаленные таблицы

Архивные таблицы представляют полученные производственные данные в архивном формате. Архивные таблицы включают *AnalogHistory* и *DiscreteHistory*. Формат архивной таблицы выглядит следующим образом:

DateTime	TagName	Value	Quality	QualityDetail	wwTagKey
02:17:97:03	Temp1	78	0	192	42
02:17:97:03	Temp2	79	0	192	43
02:17:97:03	Temp3	77	0	192	44
02:17:97:03	Temp4	80	0	192	45
02:17:97:04	Temp1	77	0	192	42
02:17:97:04	Temp2	78	0	192	43
02:17:97:04	Temp3	76	0	192	44
02:17:97:04	Temp4	79	0	192	45
02:17:97:05	Temp1	77	0	192	42
02:17:97:05	Temp2	78	0	192	43
02:17:97:05	Temp3	76	0	192	44
02:17:97:05	Temp4	79	0	192	45

Для подачи запроса на данные из одной из архивных таблиц в синтаксисе запроса следует указать тэг. Архивную таблицу можно связать с любой другой стандартной таблицей SQL Server.

Подробнее см. главу 14, "Выборка данных".

"Широкие" расширенные (удаленные) таблицы

"Широкие" расширенные (удаленные) таблицы являются преобразованными таблицами *AnalogHistory* и *DiscreteHistory*. "Широкие" таблицы включают *AnalogWideHistory* и *DiscreteWideHistory*. "Широкие" таблицы используются, когда требуется найти значение одного или нескольких тэгов за какой-то период. Использовать с той же целью таблицы *AnalogHistory* и *DiscreteHistory* гораздо сложнее. Формат "широкой" таблицы следующий:

DateTime	Temp1	Temp2	Temp3	Temp4
02:17:97:03	78	79	77	80
02:17:97:04	77	78	76	79
02:17:97:05	77	78	76	79

Для подачи запроса на данные из "широкой" таблицы в синтаксисе запроса следует указать стартовую отметку даты и времени и один или несколько тэгов. Можно конкретизировать критерий поиска для нужных вам значений (например, Temp1 > 75). "Широкие" таблицы могут быть привязаны к другим таблицам только на основе даты и времени.

Подробнее см. главу 14, "Выборка данных".

Примечание. Названия столбцов в SQL ограничены по длине и должны состоять лишь из допустимых SQL символов. Если тэги импортированы из InTouch и не соответствуют правильному формату столбцов SQL, но требуется их обработка в формате "широких" таблиц, следует переименовать их в соответствии со стандартами столбцов SQL. Эти дополнительные "имена широких столбцов" хранятся в таблице *Tag*. При указании тэгов в запросе к "широкой" таблице следует использовать имя "широкого" столбца вместо оригинального имени.

"Живые" расширенные (удаленные) таблицы

"Живые" расширенные (удаленные) таблицы выдают самые последние данные для каждого тэга. "Живые" таблицы включают *AnalogLive*, *DiscreteLive* и *StringLive*. Эти таблицы используются для получения текущего значения тэга или тэгов в IndustrialSQL Server. Формат "живых" таблиц показан ниже. Столбец *DateTime* указывает время сохранения значения.

DateTime	TagName	Value	Quality	QualityDetail	wwTagKey
02:17:97:05	Temp1	77	0	192	42
02:17:97:05	Temp2	78	0	192	43
02:17:97:05	Temp3	76	0	192	44
02:17:97:05	Temp4	79	0	192	45

Для подачи запроса на данные "живой" таблицы в синтаксисе запроса следует указать тэг. "Живая" таблица может быть привязана к любой другой таблице SQL Server.

Подробнее см. главу 14, "Выборка данных".

Доступ к информации о таблице базы данных

Для получения базовой информации о базе данных, такой как столбцы, основные и зарубежные символы или индексы, используется хранимая в памяти процедура **sp_help** из базы данных *Runtime*. Например:

```
sp_help AnalogTag
```

Информацию о ссылочной целостности таблиц см. в файле InSQLDatabaseSchema.PDF.

Расширения временного домена Industrial-SQL

IndustrialSQL Server использует Структурированный Язык Запросов (SQL). IndustrialSQL Server поддерживает два вида SQL:

- SQL для Microsoft SQL Server (Transact-SQL)
- Расширенные версии SQL для запросов по архивным таблицам (Industrial SQL)

IndustrialSQL Server поддерживает все функции Transact-SQL для таблиц конфигурации и их большую часть, с некоторыми исключениями, для расширенных (удаленных) таблиц. Расширения Industrial-SQL являются удобным средством для запроса данных на основе времени из архивных таблиц. Они обеспечивают дополнительную функциональность, которая не поддерживается в языке Transact-SQL.

Если запрос подается не через OLE DB, то следует использовать уже готовые процедуры, хранящиеся в расширениях Industrial-SQL, и соответствующие операторы SET. Это следующие расширения:

- ww_SetAnalogCyclic
- ww_SetContinuous
- ww_SetDiscreteCyclic
- ww_SetDuration
- ww_SetEndDate
- ww_SetResolution
- ww_SetRowCount
- ww_SetTimeDeadband
- ww_SetValueDeadband
- ww_SetEdgeDetection

В OLE DB эти расширения являются столбцами удаленных таблиц. Кроме того, OLE DB не поддерживает длительность, дату окончания и повторяющиеся расширения.

[Подробнее см. главу 14, "Выборка данных".](#)

Подробнее о выборке данных в IndustrialSQL Server см. в *Руководстве администратора IndustrialSQL Server*.

Г Л А В А 2

Тематические области и общие таблицы

Тематическая область есть группа таблиц, которые совместно облегчают выполнение конкретных функций системы IndustrialSQL Server. Некоторые таблицы, такие как таблица *Tag*, входят сразу в несколько тематических областей. Данная глава представляет собой алфавитный справочник таблиц, которые входят в несколько тематических областей. В этой главе описаны атрибуты таблиц и типы их данных.

Содержание

- Тематические области
- Общие таблицы
- Таблица AnalogTag
- Таблица ComplexTag
- Таблица DcreteTag
- Таблица EventHistory
- Таблица EventTag
- Таблица StringTag
- Таблица Tag
- Таблица TagRef
- Таблица Topic
- Таблица UserDetail

Тематические области

В базе данных IndustrialSQL Server существует восемь тематических областей. Тематическая область — это группа таблиц, которые совместно облегчают выполнение конкретных функций системы IndustrialSQL Server.

Тематическая область	Описание
Хранение информации	Таблицы хранения информации содержат данные, полученные с производственного участка.
Хранение событий	Таблицы хранения событий содержат описания событий, включая связанные с этими событиями тэги, детекторы событий и действия для событий. Таблицы хранения событий также могут хранить "снимки" значений тэгов во время возникновения события, а также детали самого события.
Сводка	Таблицы сводок содержат информацию, связанную с настройкой, созданием и хранением данных автоматических сводок.
Интеграция InTouch	Таблицы интеграции InTouch содержат информацию по темам, которые импортируются в базу данных IndustrialSQL Server из InTouch.
Отслеживание изменений	Таблицы отслеживания изменений содержат информацию об изменениях в столбцах базы данных.
Именные пространства и группы	Таблицы именных пространств и групп содержат информацию, определяющую элементы иерархии именных пространств и групп тэгов.
Конфигурация системы	Таблицы конфигурации системы содержат все параметры системы IndustrialSQL Server. Системная конфигурация содержит определения устройств ввода-вывода, узлы хранения и определения тэгов.
Определения тэгов	Таблицы определений тэгов содержат информацию о аналоговых, дискретных, текстовых и сложных тэгах. Эта информация включает максимальные и минимальные величины, ограничения, единицы измерения, количество хранимых данных и т.п.

Подробнее об этих таблицах см. в соответствующих главах данного справочного руководства.

Общие таблицы

Некоторые таблицы принадлежат к нескольким тематическим областям. Эти таблицы включают в себя:

- *AnalogTag*
- *ComplexTag*
- *DiscreteTag*
- *EventTag*
- *EventHistory*
- *StringTag*
- *Tag*
- *TagRef*
- *Topic*
- *UserDetail*

IndustrialSQL Server определяет пять типов тэгов: аналоговый, сложный, дискретный, тэг события и текстовый. Все пять типов обладают некоторыми общими признаками, такими как имя тэга, описание тэга, объем хранимой информации и т.п. Эти признаки хранятся в таблице *Tag*, тогда как уникальные признаки хранятся в таблицах конкретного типа тэгов. Например, полное определение аналогового тэга частично хранится в таблице *Tag* и частично в таблице *AnalogTag*.

Все эти таблицы описаны далее в этой главе.

Таблица AnalogTag

Для каждого аналогового тэга в этой таблице отведена отдельная строка.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Уникальное имя аналогового тэга или переменной в системе.
EUKey	int	Уникальный идентификатор для единицы измерения. Внешний ключ из таблицы <i>EngineeringUnit</i> .
MinEU	float(25)	Минимальное значение аналогового тэга в единицах измерения.
MaxEU	float(25)	Максимальное значение аналогового тэга в единицах измерения.
MinRaw	float(25)	Минимальное значение полученного необработанного значения.
MaxRaw	float(25)	Максимальное значение полученного необработанного значения.
Scaling	int	Алгоритм преобразования необработанных значений в единицы измерения. 0 = Нет; 1 = Линейное; 2 = Квадратичное. Квадратичное масштабирование пока не поддерживается.
RawType	int	Числовое значение необработанного значения. 1 = Euro Float (4 байта); 2 = MS Float (4 байта); 3 = Целое число (2 или 4 байта); 4 = MS Double (зарезервировано на будущее) (8 байтов).
ValueDeadband	float(25)	Процент величины единицы измерения, до которого тэг должен измениться для того, чтобы значение было сохранено. Применяется только при дельта-хранении. Если для этого атрибута введены 0 или NULL, то мертвой зоны не будет. Мертвые зоны не активизируются, когда данные получаются из активного объекта.
InitialValue	float(25)	Исходное значение, импортированное из словаря данных InTouch. Не используется в IndustrialSQL Server.
IntegerSize	tinyint	Битовая величина аналогового тэга. 12 = 12-bit; 15 = 15-bit; 16 = 16-bit; 32 = 32-bit; 64 = 64-bit (зарезервировано на будущее).

(продолжение)

Атрибут	Тип данных	Описание
SignedInteger	bit	Уточняет, имеет ли целое число знак (положительный или отрицательный). 0 = Без знака (положительно); 1 = Со знаком (отрицательно).

Таблица ComplexTag

Содержит по одной строке для каждого определенного сложного тэга. Эта таблица является вспомогательной (вспомогательная таблица 4) таблицей тэгов и содержит данные, относящиеся к сложным тэгам. Сложным называется любой тэг, кроме аналогового, дискретного или текстового. Яркими образцами являются массивы и видео.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Уникальное имя сложного тэга или переменной в системе.
InitialValue	image	Значение сложного тэга при первом импортировании в систему.
Provider	varchar(128)	Это функция, выполняющая конкретную операцию с данными (содержимым) до того, как данные поступают в базу данных для хранения. Например, алгоритм сжатия. Функции поставщика данных находятся в IndustrialSQL Server .DLL.
Retriever	varchar(128)	Это функция, выполняющая конкретную операцию с данными (содержимым) до того, как эти данные могут быть получены из базы данных. Например, алгоритм распаковки данных. Функции получателя описаны в IndustrialSQL Server .DLL.
Type	int	Тип сложного тэга: 0 = Объект COM; 1 = Данные управляются функциями; 2 = Данные, используемые программой-исполнителем как файл.

Таблица DcreteTag

Содержит одну строку для каждого определенного дискретного тэга. Эта таблица является вспомогательной (вспомогательная таблица 2) таблицей тэгов и содержит данные, относящиеся к дискретным тэгам.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Уникальное имя дискретного тэга или переменной в системе.
MessageKey	int	Уникальный числовой идентификатор для сообщения. Внешний ключ из таблицы <i>Message</i> .
InitialValue	tinyint	Исходное значение, импортированное из словаря данных InTouch. В системе IndustrialSQL Server не используется.

Таблица EventHistory

Содержит одну строку для каждого хранимого события на основе имени тэга. События должны быть сконфигурированы для того, чтобы при их возникновении они записывались в эту таблицу.

Атрибут	Тип данных	Описание
EventLogKey	int	Уникальный идентификатор для каждого возникновения события.. Это значение автоматически вычисляется системой. <i>EventLogKey</i> формирует столбец IDENTITY.
TagName	varchar(33)	Имя тэга, для которого сохраняется событие в архиве.
DateTime	Дата и время	Метка времени показывает, когда были получены архивные данные события. Это точное время возникновения события.
DetectDateTime	Дата и время	Метка времени показывает, когда событие было зафиксировано системой управления событиями.
Edge	tinyint	Сохраняет "край" фиксации события. 0 = Ведомый; 1 = Ведущий (по умолчанию); 2 = Оба; 3 = Нет; 4 = Фиксатор времени; 5 = Внешний детектор.

Таблица EventTag

Содержит одну строку для каждого составного события. Эта таблица является вспомогательной (вспомогательная таблица 5) таблицей тэгов и содержит данные, относящиеся к событиям. Таблица тэгов событий используется вспомогательными системами как таблица конфигурации для событий, содержащихся в системе IndustrialSQL Server.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Уникальное концептуальное имя тэга для определенного события.
DetectorTypeKey	int	Уникальное обозначение типа детектора, который связан с тэгом события. Внешний ключ из таблицы <i>DetectorType</i> .
ActionTypeKey	int	Уникальный идентификатор типа действия, связанного с тэгом события. Внешний ключ из таблицы <i>ActionType</i> .
ScanRate	int	Скорость (в миллисекундах), с которой возникают условия события. Если условия события оцениваются как ИСТИНА, то событие сохраняется и все относящиеся к этому событию действия активизируются.
TimeDeadband	int	Минимальный промежуток между событиями (в миллисекундах). Если за промежуток мертвой зоны происходит более одного события, то сохранится только самое последнее из них. Система не сохранит другое событие, пока не кончится заданный промежуток времени. Если <i>TimeDeadband</i> получает значение 0, это значит, что система сохранит все события.
Logged	bit	Определяет, когда нужно регистрировать событие для данного тэга в таблице <i>EventHistory</i> . 0 = Не регистрировать; 1 = Регистрировать.

(продолжение)

Атрибут	Тип данных	Описание
Status	tinyint	Специальный флаг, используемый системой управления событиями при запуске системы и во время выполнения для того, чтобы определить, изменялся ли тэг. 0 = Зафиксировано. Система зафиксировала и ввела в действие все изменения. 1 = Новый. Тэг события вставлен, но еще не выполняется. 2 = Изменен. Тэг события обновлен, но пока выполняется старый. 99 = Отключен. Тэг события не выполняется, даже если его определение еще присутствует в схеме. Следует заметить, что прежде, чем работающая система определит задержку в тэге события, может пройти до 30 секунд.
PostDetectorDelay	int	Количество времени (мс), которое должно пройти до фиксации события перед тем, как могут быть предприняты действия для данного события.
UseThreadPool	bit	Определяет, как системные потоки используются для обработки событий. 1 = Все события идут одним потоком и одинаково регистрируются системой SQL Server; 0 = Каждое событие использует отдельный системный поток и регистрируется отдельно. Это позволяет системе управления событиями параллельно управлять скоростью сканирования каждого детектора. (Пока не реализовано).
DetectorString	текст	Сценарий, содержащий критерии фиксации событий. Сценарии фиксации выполняются системой IndustrialSQL Server локально.
ActionString	текст	Сценарий, определяющий действия над событиями. Сценарии действий выполняются локальным сервером IndustrialSQL.
Priority	tinyint	Уровень приоритета для действия. 0 = Обычный; 1 = Критический. Уровень приоритета определяет место, которое действие займет в списке.

(продолжение)

Атрибут	Тип данных	Описание
Edge	tinyint	Определяет, на каком "краю" фиксации событие будет зарегистрировано. 0 = Отслеживаемый; 1 = Ведущий (по умолчанию); 2 = Оба; 3 = Нет; 4 = Фиксатор времени; 5 = Внешний детектор.

Таблица StringTag

Содержит одну строку для каждого определенного тестового тэга. Эта таблица является вспомогательной (вспомогательная таблица 3) таблицей тэгов и содержит данные, относящиеся конкретно к текстовым тэгам.

Атрибут	Тип данных	Описание
Тэг	varchar(33)	Уникальное имя тэга или переменной в системе.
Длина	tinyint	Максимальная длина строки в байтах.
Исходное Значение	varchar(130)	Исходное значение, используемое InTouch при запуске системы (не используются в IndustrialSQL Server).

Таблица Tag

Содержит одну строку для каждого тэга в системе и включает в себя основное определение для конкретного тэга. Тэг есть элементарная переменная аналогового, дискретного, текстового, сложного типа или события, которая может быть рассчитана и сохранена.

Атрибуты, содержащиеся в данной таблице используются для отображения тэгов, настройки системы сбора данных и идентификации типов тэгов. Определяющая информация для тэга "статична" в том смысле, что она меняется только в промежуток между просмотрами базы данных.

Все тэги получают значение через RT Server. Триггеры системных правил наполняют таблицу *TagRef* и системные именные пространства из этой таблицы.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Уникальное наименование тэга или переменной в системе.
IOServerKey	int	Уникальный числовой идентификатор для сервера ввода-вывода.
StorageNodeKey	int	Уникальный числовой идентификатор для узла хранения.
WwTagKey	int	Уникальный числовой идентификатор для тэга в одиночном IndustrialSQL Server. Это значение автоматически вычисляется системой. <i>wwTagKey</i> есть столбец IDENTITY.
TopicKey	int	Уникальный числовой идентификатор темы.
Description	varchar(50)	Описание тэга
AcquisitionType	tinyint	Метод, при помощи которого получается значение тэга. 0 = Не получено; 1 = Получено через сервер ввода-вывода 2 = Подсчитано (будет использоваться в дальнейшем).
StorageType	smallint	Тип хранения для определенного тэга. 0 = Не сохранен; 1 = Циклически; 2 = Хранение дельта-данных (исключение).
AcquisitionRate	int	Для опрошенных тэгов типа <i>AcquisitionRate1</i> это скорость опроса в миллисекундах. Сейчас не используется.
StorageRate	int	Скорость в миллисекундах, с которой тэг сохраняется при циклическом способе сохранения.

(продолжение)

Атрибут	Тип данных	Описание
ItemName	varchar(80)	Строка адреса тэга. Обычно то же самое, что и имя DDE-элемента.
TagType	int	Определяет тип тэга. 1 = Аналоговый; 2 = Дискретный; 3 = Текстовый; 4 = Сложный; 5 = Событие.
WideColumnName	varchar(30)	Вариант TagName, приемлемый для SQL Server. Обычно <i>WideColumnName</i> соответствует <i>TagName</i> , если длина имени тэга не превышает 30 символов и не содержит недопустимых символов.
TimeDeadband	int	Минимальный промежуток времени (в миллисекундах) между сохраненными значениями одного тэга. Тэг, значение которого изменялось за этот период более одного раза, игнорируется алгоритмом сохранения дельта-данных. При значении <i>TimeDeadband</i> 0 (по умолчанию) значение тэга будет сохраняться системой при каждом изменении. <i>TimeDeadband</i> не влияет на значения активного объекта.
DateCreated	datetime	Дата создания данного тэга. Без дополнительного уточнения дата будет занесена автоматически.
CreatedBy	varchar(30)	Имя пользователя или приложения, создавшего тэг. Без указания вносится автоматически.
CurrentEditor	int	Определяет редактор, в котором создан тэг - IndustrialSQL Server или InTouch. 0 = IndustrialSQL Server; 1 = InTouch.
SamplesInActiveImage	int	Количество выборок для каждого сконфигурированного в системе тэга, содержащееся в активном объекте. NULL показывает, что активный объект использует стандарт - 65 значений.

Таблица TagRef

Содержит одну строку для каждого тэга в системе. Используется как справочная таблица для *таблицы Tag*. При использовании TagRef система избегает распространения *Имени тэга* как первичного ключа вложенных таблиц.

Атрибут	Тип данных	Описание
wwDomainTagKey	int	Уникальный числовой идентификатор для тэга указанного домена. Это значение автоматически вычисляется системой. <i>wwDomainTagKey</i> есть столбец IDENTITY.
WwTagKey	int	Уникальный числовой идентификатор для тэга в одиночном IndustrialSQL Server. <i>WwTagKey</i> является внешним ключом таблицы <i>Tag</i> .
ServerKey	int	Уникальный числовой идентификатор для IndustrialSQL Server.
TagName	varchar(33)	Уникальное имя тэга или переменной в системе.
TagType	int	Определяет тип тэга 1 = Аналоговый; 2 = Дискретный; 3 = Текстовый; 4 = Сложный; 5 = Событие.

Таблица Topic

Содержит одну строку для каждой темы, считываемой из сервера ввода-вывода. Тема есть логический блок данных, полученных от сервера ввода-вывода. Темы обрабатываются сервером ввода-вывода соответственно локальной настройке сервера ввода-вывода. Обычно тема — это комбинация физического устройства и скорости получения данных.

Атрибут	Тип данных	Описание
TopicKey	int	Уникальный числовой идентификатор темы. Это значение автоматически вычисляется системой. <i>TopicKey</i> является столбцом IDENTITY.
IOServerKey	int	Уникальный числовой идентификатор сервера ввода-вывода.
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения.
Name	varchar(80)	Название темы. Тема есть логический блок данных сервера ввода-вывода.
TimeOut	int	Промежуток времени (в миллисекундах), за который данные по теме должны быть получены. Если за этот промежуток времени данные не получены, тема считается "мертвой" и все тэги этой темы получают метку "плохое качество", которая останется с ними до тех пор, пока не будет получено новое значение тэга.

Таблица UserDetail

Содержит одну строку для каждого. Используется для хранения дополнительной информации, не хранящейся в таблице *Runtime.sysusers* Microsoft SQL Server. Относится как к пользователям, так и к группам пользователей.

При установке IndustrialSQL Server в системе Microsoft SQL Server создается задание SQL , автоматически обновляющее эту таблицу каждый час. Для того, чтобы это задание выполнялось, должен быть запущен агент SQL Server. Более подробную информацию см. в электронной документации Microsoft.

Атрибут	Тип данных	Описание
UserKey	int	Уникальный числовой идентификатор для пользователя базы данных,
UserName	varchar(30)	Имя пользователя базы данных.
AccessLevel	int	Уровень безопасности для пользователя. Наименьшее значение 1, а наибольшее 9999. Используется для ограничения доступа определенных пользователей.
uid	int	Идентификатор пользователя. Берется из таблицы <i>sysusers</i> системы Microsoft SQL Server.
gid	int	Идентификатор группы, к которой принадлежит пользователь. Берется из таблицы <i>sysusers</i> системы Microsoft SQL Server.

Г Л А В А 3

Таблицы хранения данных

В этой главе содержится алфавитный справочник таблиц хранения данных в базе данных IndustrialSQL Server. Таблицы хранения данных — это таблицы, в которых хранятся данные точного времени и архивных значений тэгов. Здесь объясняются все атрибуты таблиц хранения данных и типы их данных.

Содержание

- Таблицы хранения данных
- Таблица AnalogHistory
- Таблица AnalogLive
- Таблица AnalogTag
- Таблица AnalogWideHistory
- Таблица Annotation
- Таблица ComplexHistory
- Таблица ComplexTag
- Таблица DiscreteHistory
- Таблица DiscreteLive
- Таблица DiscreteTag
- Таблица DiscreteWideHistory
- Таблица EventHistory
- Таблица EventTag
- Таблица HistoryBlock
- Таблица ManualAnalogHistory
- Таблица ManualDiscreteHistory
- Таблица StringHistory
- Таблица StringLive
- Таблица StringTag
- Таблица Tag
- Таблица UserDetail

Таблицы хранения данных

Таблицы хранения данных содержат данные, полученные с производственных участков. Система IndustrialSQL Server может хранить как текущие значения аналоговых, дискретных, текстовых и сложных тэгов, так и архивные данные, базирующиеся на штампе даты и времени. Для аналоговых и дискретных тэгов архивные данные помещаются во множество табличных форматов для более высокой эффективности обработки разнообразных запросов.

Список таблиц хранения данных:

- *AnalogHistory*
- *AnalogLive*
- *AnalogTag*
- *AnalogWideHistory*
- *Annotation*
- *ComplexHistory*
- *ComplexTag*
- *DiscreteHistory*
- *DiscreteLive*
- *DiscreteTag*
- *DiscreteWideHistory*
- *EventHistory*
- *EventTag*
- *HistoryBlock*
- *ManualAnalogHistory*
- *ManualDiscreteHistory*
- *StringHistory*
- *StringLive*
- *StringTag*
- *Tag*
- *UserDetail*

Схема хранения данных выглядит следующим образом:

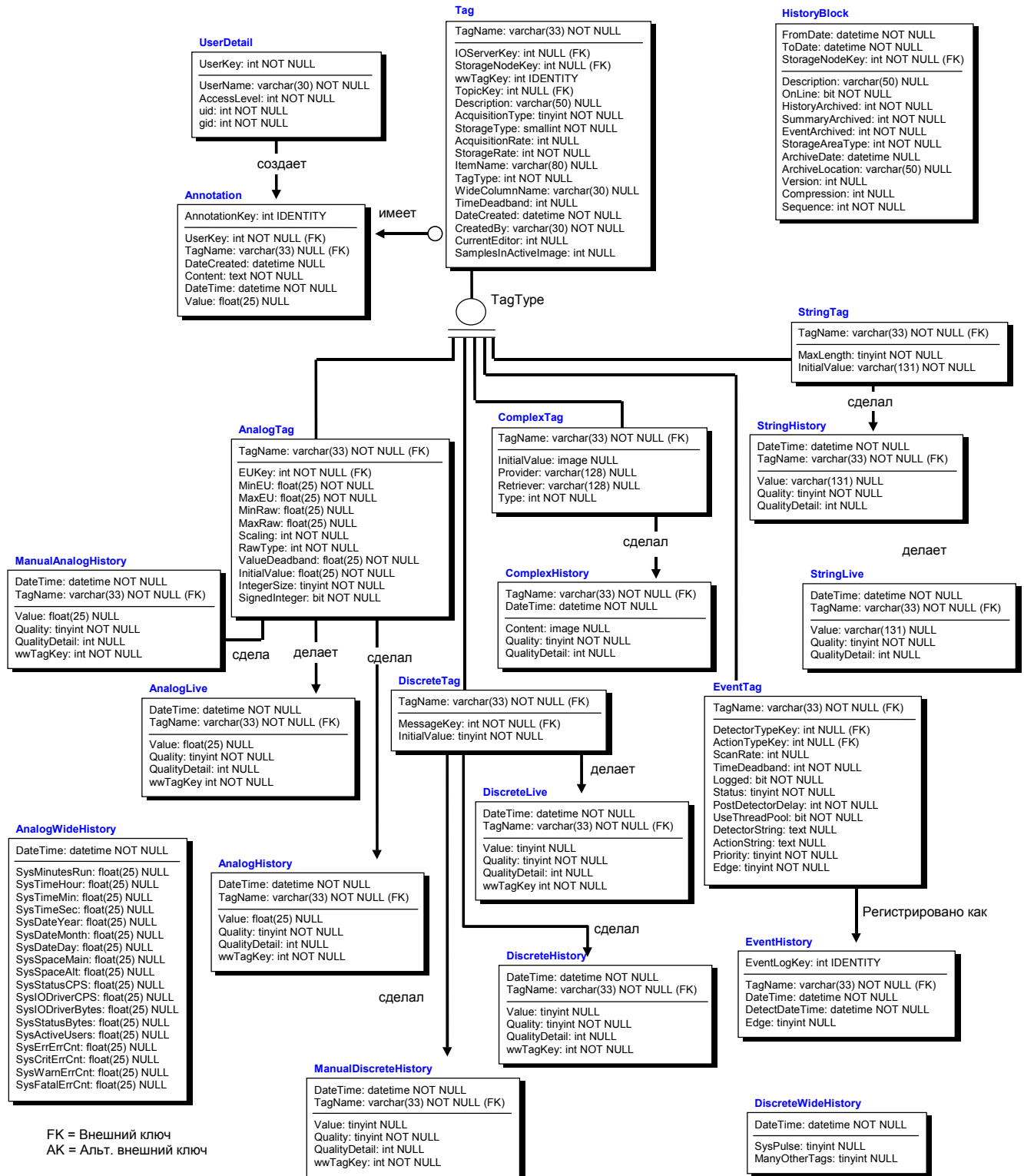


Таблица AnalogHistory

Содержит одну строку для каждого хранимого аналогового тэга на период (циклическое сохранение) или изменение значения (хранение дельта-данных). Таблица AnalogHistory хранит каждое аналоговое значение, записанное системой IndustrialSQL Server. Таблица AnalogHistory есть таблица расширения, хранимая вне Microsoft SQL Server и контролируемая расширениями языка Industrial-SQL.

Это логическая таблица, выдающая количество строк от стартовой до конечной даты в соответствии с запросом и независимо от частоты получения данных. Для большинства приложений это освобождает пользователя от обязанности знать подробности получения данных и скорости сканирования. Запрашивая любую *аналогово-архивную таблицу*, следует указать имя тэга.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда были получены или сохранены аналоговые архивные данные. Если данные получены через SuiteLink, метка времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя аналогового тэга, для которого сохраняется архив.
Value	float(25)	Значение аналогового тэга в момент даты/времени. Указывается в единицах измерения. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества, прикрепленный к тэгу аналогового архива. Подробности в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробности в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой

идентификатор тэга в одиночной
системе IndustrialSQL Server.
WwTagKey есть внешний ключ из
таблицы *Tag*.

Таблица AnalogLive

Содержит одну строку для каждого аналогового тэга. Значение этого тэга обновляется при каждом получении нового значения с сервера ввода-вывода. Эта таблица показывает самые последние данные. Таблица AnalogLive является таблицей расширения, которая хранится вне Microsoft SQL Server и управляется расширениями языка Industrial-SQL.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда обновленные аналоговые данные были получены или сохранены. Если данные получены через SuiteLink, то метка времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя аналогового тэга.
Value	float(25)	Значение аналогового тэга на момент даты/времени. Указывается в единицах измерения. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества, связанный со значением аналогового "живого" тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> есть внешний ключ из таблицы <i>Tag</i> .

Таблица AnalogTag

См. в главе 2 Справочного руководства разделы "Тематические области и общие таблицы".

Таблица AnalogWideHistory

Содержит одну строку значений для множества аналоговых тэгов. Эта удаленная таблица ужесточает обычные правила обработки данных, но для некоторых запросов она очень полезна. Эта таблица используется для получения значений множества аналоговых тэгов в заданный момент времени.

Таблица AnalogWideHistory содержит те же данные, что и таблица AnalogTag, но в ином формате. В "широком" формате существует столбец метки даты/времени и столбцы для значений каждого выбранного тэга в этот момент времени. Имена запрашиваемых тэгов используются как названия столбцов. В этой таблице присутствуют один столбец для метки даты и времени и столбцы для каждого имени тэга, указанного в запросе.

Таблица AnalogWideHistory является таблицей расширения, хранимой вне Microsoft SQL Server и контролируемой расширениями языка Industrial-SQL. Подробную информацию о системных тэгах, являющихся стандартом для этой таблицы, см. в *Руководстве администратора системы IndustrialSQL Server*, в главе 8, "Мониторинг системы".

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда данные в строке были получены или сохранены. Если данные получены через SuiteLink, то этот метка ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных. В этой удаленной таблице точность метки времени обычно определяется разрешением запроса.
SysMinutesRun	real	Стандартный тэг, который появляется в таблице. Данный тэг присутствует для демонстрации функций "широкого" табличного формата и может быть заменен любым аналоговым тэгом.
SysTimeHour	real	См. атрибут <i>SysMinutesRun</i> .
SysTimeMin	real	См. атрибут <i>SysMinutesRun</i> .
SysSpaceMain	real	См. атрибут <i>SysMinutesRun</i> .
SysSpaceAlt	real	См. атрибут <i>SysMinutesRun</i> .
SysStatusCPS	real	См. атрибут <i>SysMinutesRun</i> .
SysIODriverCPS	real	См. атрибут <i>SysMinutesRun</i> .
SysStatusBytes	real	См. атрибут <i>SysMinutesRun</i> .
SysIODriverBytes	real	См. атрибут <i>SysMinutesRun</i> .

Атрибут	Тип данных	Описание
SysActiveUsers	real	См. атрибут <i>SysMinutesRun</i> .
ManyOtherTags	real	Столбец "заглушки" для одного или более аналоговых тэгов в широком табличном формате. В этом формате имена тэгов используются как названия столбцов. Столбец ManyOtherTags дублируется для всех тэгов, указанных в запросе базы данных.

Примечание. Названия столбцов являются именами тэгов, а значения этих тэгов являются содержимым столбцов.

Столбец *Quality* не виден в этом представлении из-за большого количества переменных. Вместо этого значение изменяется на NULL, если полученная переменная недоступна, заполнена или не открывается. Это представление данных очень удобно для разработчиков, но оно не может быть таблицей Microsoft SQL Server из-за слишком большого количества столбцов.

При подаче запроса в таблицу *AnalogWideHistory* действуют следующие ограничения:

- Должно быть указано имя столбца.
- Запросы к словарям данных (таким как **sp_tables**, **sp_columns**, **sp_help**) определяют, что таблица имеет следующий вид:

Атрибут	Тип данных	Описание
DateTime	datetime	Дата и время измерения
Val_1_n	real	Таблица общих значений.

Внимание! Многие графические инструменты, использующие словарь данных для операций типа "выбрать и нажать", не будут работать с этой таблицей.

Поскольку имена тэгов используются в качестве названий столбцов, идентификаторы Microsoft SQL Server будут работать. Была сделана специальная доработка — теперь поддерживается "\$" как первый символ в имени тэга (например, для системных тэгов InTouch и для имен тэгов, начинающихся с одиночной цифры).

При перенесении данных из этой таблицы в новую таблицу Microsoft SQL Server система Microsoft SQL Server может не распознать названия столбцов как правильные. IndustrialSQL Server автоматически переводит названия столбцов, используя следующие правила:

1. Дефисы заменяются на подчеркивания.
2. В начало имени тэга, начинающегося с цифры, ставится знак подчеркивания.

Кроме этих двух исключений, любое имя тэга, содержащее недопустимые для названия столбца символы (идентификатор), не будет содержать данных.

Таблица Annotation

Содержит одну строку для каждого пользовательского описания тэга. Пользователь может вносить личные (или общедоступные) аннотации о работе любого тэга в любой момент времени. Каждая аннотация в этой таблице относится к отдельному пользователю базы данных.

Атрибут	Тип данных	Описание
AnnotationKey	int	Уникальный числовой идентификатор аннотации. Данное значение автоматически генерируется системой. AnnotationKey есть столбец IDENTITY.
UserKey	int	Уникальный числовой идентификатор пользователя базы данных. Вводится в базу данных администратором вручную.
TagName	varchar(33)	Имя тэга, ассоциируемое с этой аннотацией.
DateCreated	datetime	Дата создания аннотации.
Content	text	Текст аннотации.
DateTime	datetime	Метка времени создания пользователем аннотации к имени тэга.
Value	float(25)	Значение тэга на момент создания аннотации.

Таблица ComplexHistory

Содержит одну строку для каждого периодически сохраняемого сложного тэга. Сложные тэги включают в себя BLOВЫ, такие как массивы, файлы с расширением AVI, рисунки и т.д. Таблица ComplexHistory заполняется не системой, а приложением-клиентом.

Таблица ComplexHistory является обычной таблицей SQL Server.

Примечание. В настоящем выпуске системы сложные тэги не поддерживаются.

Атрибут	Тип данных	Описание
TagName	varchar(33)	Имя сложного тэга, для которого сохраняется архив.
DateTime	datetime	Метка времени, показывающая, когда были получены данные сложного архива.
Content	image	Содержимое хранится как сложное значение. Содержимым могут быть массивы, файлы AVI, рисунки и т.п.
Quality	tinyint	Основной показатель качества для этого значения тэга сложного архива. Подробнее в Приложении Б, "Качество данных" .
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных" .

Таблица ComplexTag

[Подробнее информацию об этой таблице см. в глава 2 Справочного руководства, "Тематические области и общие таблицы"](#).

Таблица DiscreteHistory

Содержит одну строку для каждого хранимого дискретного тэга на период. Таблица DiscreteHistory является таблицей расширения, которая хранится вне Microsoft SQL Server и управляется расширениями языка Industrial-SQL. Содержит каждое дискретное событие, записанное системой IndustrialSQL Server. В каждом запросе к таблице DiscreteHistory следует указать имя тэга.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, показывающий, когда были получены или сохранены данные дискретного архива. Если данные получены через SuiteLink, то метка времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя дискретного тэга, для которого сохраняется архив.
Value	tinyint	Значение дискретного тэга в момент даты/времени. 0 = ЛОЖЬ; 1 = ИСТИНА; NULL = Данных нет. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества, прикрепленный к тэгу дискретного архива. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> есть внешний ключ из таблицы <i>Tag</i> .

Каждый раз при изменении состояния тэга его состояние сохраняется с идентификатором тэга и индикатором качества.

Примечание. Если IndustrialSQL Server остановлен и перезапущен, или же источник данных отсоединен и затем вновь подсоединен, за этот период данные не сохраняются. Дельта-метод выборки данных показывает последнее сохраненное значение, после которого идет значение NULL, а затем идет первое положительное значение после перезапуска системы.

Когда запрос подается дельта-методом, сервер выдает строку для каждого сохраненного события в промежутке времени подачи запроса, независимо от величины промежутка времени между событиями. Максимальное количество выдаваемых строк определяется расширением области временного счетчика строк. Если дельта-выборка производится с использованием расширенного временного домена разрешения IndustrialSQL Server, то IndustrialSQL Server разделит длительность на разрешение, и результат будет являться количеством выдаваемых строк. Данные будут выдаваться с интервалом, определенным разрешением.

Таблица DiscreteLive

Содержит одну строку для каждого дискретного тэга. Значение каждого тэга в этой таблице обновляется при каждом получении нового значения с сервера ввода-вывода. Эта таблица предоставляет самые последние данные. Таблица DiscreteLive есть таблица расширения, хранимая вне Microsoft SQL Server и управляемая расширениями языка Industrial-SQL.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда обновленные "живые" данные были получены или сохранены. Если данные получены через SuiteLink, то метка времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя дискретного тэга.
Value	tinyint	Значение дискретного тэга в момент даты/времени. 0 = ЛОЖЬ; 1 = ИСТИНА; NULL = Данных нет. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества прикрепленный к дискретному "живому" тэгу. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> есть внешний ключ из таблицы <i>Tag</i> .

Таблица DiscreteTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица DiscreteWideHistory

Содержит одну строку значений для множества дискретных тэгов. Эта удаленная таблица ужесточает обычные правила обработки данных, но для некоторых запросов она очень полезна. Она позволяет получать значения множества дискретных тэгов в заданный момент времени.

Таблица DiscreteWideHistory содержит те же данные, что и таблица DiscreteHistory, но в ином формате. В "широком" формате существуют столбцы метки даты/времени и столбцы для значений каждого выбранного тэга в этот момент времени. Имена запрашиваемых тэгов используются как названия столбцов. В этой таблице присутствует один столбец для метки даты и времени и столбцы для каждого имени тэга, указанного в запросе.

Таблица DiscreteWideHistory есть таблица расширения, хранимая вне Microsoft SQL Server и управляемая расширениями языка Industrial-SQL.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда данные в строке были получены или сохранены. Если данные получены через SuiteLink, то метка ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных. Для дискретных значений это метка момента изменения элемента данных. Если более чем один элемент данных изменился за время между расчетами двух строк, то это самая ранняя метка времени.
SysPulse	tinyint	Стандартный тэг, который появляется в таблице. Данный тэг присутствует для демонстрации функций "широкого" табличного формата и может быть заменен любым дискретным тэгом.
ManyOtherTags	tinyint	Столбец "заглушки" для одного или более дискретных тэгов в "широком" табличном формате. В этом формате имена тэгов используются как названия столбцов. ManyOtherTags дублируется для всех тэгов, указанных в запросе базы данных.

Примечание. Названия столбцов являются именами тэгов, а значения этих тэгов являются содержимым столбцов.

Столбец *Quality* не виден в этом представлении из-за большого количества переменных. Вместо этого значение изменяется на NULL, если полученная переменная недоступна, заполнена или не открывается. Это представление данных очень удобно для разработки и не может быть таблицей Microsoft SQL Server из-за слишком большого количества столбцов.

В результате появляются определенные ограничения:

- Должно быть указано имя столбца.
- Запросы к словарям данных (таким как **sp_tables**, **sp_columns** и **sp_help**) определяют, что таблица имеет следующий вид:

Атрибут	Тип данных	Описание
DateTime	datetime	Дата и время измерения.
Val_1_n	real	Столбец общих значений.

Внимание! Многие графические инструменты, использующие словарь данных для операций типа "выбрать и нажать", не будут работать с этой таблицей.

Поскольку имена тэгов используются в качестве названий столбцов, то идентификаторы Microsoft SQL Server будут работать. Была сделана специальная доработка — теперь поддерживается "\$" как первый символ в имени тэга (например, для системных тэгов InTouch и для имен тэгов, начинающихся с одиночной цифры).

При перенесении данных из этой таблицы в новую таблицу Microsoft SQL Server система Microsoft SQL Server может не распознать названия столбцов как правильные. IndustrialSQL Server автоматически переводит названия столбцов, используя следующие правила:

1. Дефисы заменяются на подчеркивания.
2. В начало имен тэгов, начинающихся с цифры, ставится знак подчеркивания.

Кроме этих двух исключений, любое название тэга, содержащее недопустимые для названия столбца символы (идентификатор), не будет содержать данных.

Таблица EventHistory

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица EventTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица HistoryBlock

Содержит одну строку для каждого блока архивных данных, хранимых в узле хранения.

В IndustrialSQL Server архивные данные делятся на отдельные части, называемые блоками. Деление архивных данных на блоки облегчает их резервирование и восстановление. Новый блок создается (или требуется) в следующих случаях:

- Обновляется версия базы данных.
- Перезапускается сервер хранения IndustrialSQL Server.
- По истечении определенного времени (по умолчанию, 24 часа) при непрерывном сохранении данных. За этот специально запланированный период появляется новый блок данных, удобный для резервирования и восстановления.
- Для создания нового блока вручную следует воспользоваться программой InSQL Control или расширенной процедурой сохранения **xp_NewHistoryBlock**.

Таблице HistoryBlock можно подать запрос на поиск известных системе блоков данных. Строка таблицы HistoryBlock относится к физическому подкаталогу файлов, содержащих архивные данные.

Таблица HistoryBlock есть таблица расширения, хранимая вне Microsoft SQL Server и управляемая расширениями языка Industrial-SQL.

Атрибут	Тип данных	Описание
FromDate	datetime	Стартовая дата и время создания архивного блока.
ToDate	datetime	Время окончания создания архивного блока.
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения.
Description	varchar(50)	Описание архивного блока.
OnLine	bit	Проверяет доступность данных на диске. 1 = Активно; 2 = Включено; 3 = Отключено. Зарезервировано для использования в будущем.
HistoryArchived	int	Показывает, был ли сжат (зарезервирован) архивный блок. 1 = Нет; 2 = Сжат; 3 = Распакован; 4 = Удален. Зарезервировано для использования в будущем.
SummaryArchived	int	Показывает, была ли сжата (зарезервирована) сводка тэгов. 1 = Нет; 2 = Сжата; 3 = Распакована; 4 = Удалена. Зарезервировано для использования в будущем.

(продолжение)

Атрибут	Тип данных	Описание
EventArchived	int	Показывает, было ли сжато (зарезервировано) событие. 1 = Нет; 2 = Сжато; 3 = Распаковано; 4 = Удалено. Зарезервировано для использования в будущем.
StorageAreaType	int	Парадигма, используемая для хранения данных. 1 = Циклическая; 2 = Альтернативная; 3 = Буферная; 4 = Постоянная. Зарезервировано для использования в будущем.
ArchiveDate	datetime	Дата, когда был сжат архивный блок. Зарезервировано для использования в будущем.
ArchiveLocation	varchar(50)	Место, где располагается сжатый архивный блок. Зарезервировано для использования в будущем.
Version	int	Номер версии архивного блока. 1 = Формат блока ниже версии 3.0; 2 = Формат блока версии 3.0 и выше. Зарезервировано для использования в будущем.
Compression	int	Номер версии циклического сжатия. 1 = Сжатия нет; 2 = Кодирование Хаффмана. Зарезервировано для использования в будущем.
Sequence	int	Частота потока данных. (1...n) Зарезервировано для использования в будущем.

Таблица ManualAnalogHistory

Эта таблица может использоваться различными приложениями-клиентами для хранения значений аналоговых тэгов. По умолчанию эта таблица пуста. При заполнении приложением-клиентом эта таблица будет получать одну строку для каждого определенного за период аналогового тэга. Таблица ManualAnalogHistory является стандартной таблицей SQL Server и не поддерживает ни одного расширения управления данными Industrial-SQL.

Столбцы этой таблицы те же, что и в таблице *аналоговых архивов*.

Таблица ManualDiscreteHistory

Эта таблица может использоваться различными приложениями-клиентами для хранения значений аналоговых тэгов. По умолчанию эта таблица пуста. При заполнении приложением-клиентом она будет получать одну строку для каждого определенного за период аналогового тэга. Таблица ManualDiscreteHistory является стандартной таблицей SQL Server и не поддерживает ни одного расширения управления данными Industrial-SQL.

Столбцы этой таблицы те же, что и в таблице *DiscreteHistory*.

Таблица StringHistory

Таблица StringHistory содержит архивные значения текстовых тэгов за разные промежутки времени. Для каждого нового значения автоматически создается новая строка. Таблица StringHistory есть стандартная таблица SQL Server, а не таблица расширения IndustrialSQL Server.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка времени, когда были получены или сохранены текстовые данные. Если данные получены через SuiteLink, то метка времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя текстового тэга, для которого сохраняется архив.
Value	varchar(131)	Значение текстового тэга в момент даты/времени. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества, прикрепленный к тэгу текстового архива. Подробнее в Приложении Б, "Качество данных" .
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных" .

Таблица StringLive

Содержит одну строку для каждого текстового тэга. Значение каждого тэга в этой таблице обновляется при каждом получении нового значения с сервера ввода-вывода. Эта таблица предоставляет самые последние данные.

Атрибут	Тип данных	Описание
DateTime	datetime	Метка стартового времени, когда были получены или сохранены текстовые данные. Если данные получены через SuiteLink, то метка стартового времени ставится драйвером ввода-вывода; если данные получены через DDE, то она ставится системой IndustrialSQL Server при сохранении данных.
TagName	varchar(33)	Имя текстового тэга
Value	varchar(131)	Значение текстового тэга (как текст) в момент даты/времени. Это не фактическое значение тэга на дату и время, а последнее значение, известное непосредственно перед моментом (дата/время + скорость сохранения).
Quality	tinyint	Основной показатель качества, связанный со значением текстового "живого" тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> есть внешний ключ из таблицы <i>Tag</i> .

Таблица StringTag

[Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".](#)

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица UserDetails

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Г Л А В А 4

Таблицы хранения событий

Эта глава содержит алфавитный справочник таблиц хранения данных системы управления событиями. Таблицы системы хранения событий есть таблицы, используемые для конфигурации детекторов событий и последовательных действий системы управления событиями и хранения "снимков" аналоговых, дискретных и тестовых тэгов. В этой главе описываются все атрибуты хранения и типы их данных.

Описания таблиц системы управления событиями, используемых для подведения итогов, см. главу 8, "Таблицы сводок".

Содержание

- Таблицы хранения событий
- Таблица ActionType
- Таблица AnalogSnapshot
- Таблица DetectorType
- Таблица DiscreteSnapshot
- Таблица EventHistory
- Таблица EventTag
- Таблица EventTagPendingDelete
- Таблица Frequency
- Таблица SnapshotTag
- Таблица SQLTemplate
- Таблица StringSnapshot
- Таблица Tag
- Таблица TimeDetectorDetail
- Таблица TimeDetectorDetailPendingDelete

Таблицы хранения событий

Таблицы хранения событий содержат определения событий, включая тэги событий, детекторы событий и действия над событиями. Таблицы хранения событий также могут хранить "снимки" значений тэгов в моменты происхождения событий, а также детали самих событий.

Таблицы хранения информации:

- *ActionType*
- *AnalogSnapshot*
- *DetectorType*
- *DiscreteSnapshot*
- *EventHistory*
- *EventTag*
- *EventTagPendingDelete*
- *Frequency*
- *SnapshotTag*
- *SQLTemplate*
- *StringSnapShot*
- *Tag*
- *TimeDetectorDetail*
- *TimeDetectorDetailPendingDelete*

```

classDiagram
    class Tag {
        tagName: varchar(33) NOT NULL
        ioserverkey: int NULL (FK)
        storagenodekey: int NULL (FK)
        wwtagkey: int IDENTITY
        topickey: int NULL (FK)
        description: varchar(50) NULL
        acquisitiontype: tinyint NOT NULL
        storagetype: smallint NOT NULL
        acquisitionrate: int NULL
        storagerate: int NOT NULL
        itemname: varchar(80) NULL
        tagtype: int NOT NULL
        widecolumnname: varchar(30) NULL
        timedeadband: int NULL
        datecreated: datetime NOT NULL
        createdby: varchar(30) NOT NULL
        currenteditor: int NULL
        samplesinactivelimage: int NULL
    }
    class SummaryOperation {
        operationkey: int IDENTITY
        tagname: varchar(33) NOT NULL (FK)
        calcType: (CalcTypes) char(3) NOT NULL (FK)
        description: varchar(50) NULL
        duration: real NOT NULL
        resolution: int NOT NULL
        timestamp: tinyint NOT NULL
    }
    class EventTag {
        tagName: varchar(33) NOT NULL (FK)
        detectorTypeKey: int NULL (FK)
        actionTypeKey: int NULL (FK)
        scanRate: int NULL
        timeDeadband: int NOT NULL
        logged: bit NOT NULL
        status: tinyint NOT NULL
        postDetectorDelay: int NOT NULL
        useThreadPool: bit NOT NULL
        detectorString: text NULL
        actionString: text NULL
        priority: tinyint NOT NULL
        edge: tinyint NOT NULL
    }
    class DetectorType {
        detectorTypeKey: int IDENTITY
        name: varchar(33) NOT NULL
        description: varchar(50) NULL
        editorClassName: varchar(80) NULL
        detectorClassName: varchar(80) NULL
        executionMode: tinyint NOT NULL
    }
    class EventHistory {
        eventLogKey: int IDENTITY
        tagName: varchar(33) NULL (FK)
        dateTime: datetime NOT NULL
        detectDateTime: datetime NOT NULL
        edge: tinyint NULL
    }
    class StringSnapshot {
        snapshotTagKey: int NOT NULL (FK)
        eventLogKey: int NOT NULL (FK)
        value: varchar(131) NULL
        quality: tinyint NOT NULL
        qualityDetail: int NULL
    }
    class AnalogSnapshot {
        snapshotTagKey: int NOT NULL (FK)
        eventLogKey: int NOT NULL (FK)
        value: float(25) NULL
        quality: tinyint NOT NULL
        qualityDetail: int NULL
    }
    class DiscreteSnapshot {
        snapshotTagKey: int NOT NULL (FK)
        eventLogKey: int NOT NULL (FK)
        value: tinyint NULL
        quality: tinyint NOT NULL
        qualityDetail: int NULL
    }
    class ActionTag {
        operationKey: int IDENTITY
        tagName: varchar(33) NOT NULL (FK)
        calcType: (CalcTypes) char(3) NOT NULL (FK)
        description: varchar(50) NULL
        duration: real NOT NULL
        resolution: int NOT NULL
        timestamp: tinyint NOT NULL
    }
    class Frequency {
        frequencyID: int IDENTITY
        frequency: varchar(12) NOT NULL
    }
    class TimeDetectorDetail {
        timeDetectorDetailKey: int IDENTITY
        frequencyID: int NOT NULL (FK)
        tagName: varchar(33) NOT NULL (FK)
        periodicity: int NULL
        startDateTime: datetime NULL
        runTimeDay: tinyint NULL
        runTimeHour: tinyint NULL
        runTimeMin: tinyint NULL
    }
    class ActionType {
        actionTypeKey: int IDENTITY
        name: varchar(33) NOT NULL
        description: varchar(50) NULL
        editorClassName: varchar(80) NULL
        actionClassName: varchar(80) NULL
    }
    class EventTagPendingDelete {
        tagName: varchar(33) NOT NULL
        detectorTypeKey: int NULL
        actionTypeKey: int NULL
        scanRate: int NULL
        timeDeadband: int NOT NULL
        logged: bit NOT NULL
        status: tinyint NOT NULL
        postDetectorDelay: int NOT NULL
        useThreadPool: bit NOT NULL
        detectorString: text NULL
        actionString: text NULL
        priority: tinyint NOT NULL
        edge: int NULL
    }
    class TimeDetectorDetailPendingDelete {
        tagName: varchar(33) NOT NULL (FK)
        frequencyID: int NULL
        periodicity: int NULL
        startDateTime: datetime NULL
        runTimeDay: int NULL
        runTimeHour: bit NULL
        runTimeMin: tinyint NULL
    }

    Tag --> SummaryOperation : Чем активизируется
    Tag --> EventTag : Тип
    EventTag --> DetectorType : Тип
    EventTag --> ActionTag : Тип
    EventTag --> EventHistory : Регистрируется как
    EventTag --> StringSnapshot : Имеет
    EventTag --> AnalogSnapshot : Имеет
    EventTag --> DiscreteSnapshot : Имеет
    EventTag --> EventTagPendingDelete : Имеет
    EventTag --> TimeDetectorDetailPendingDelete : Имеет
    EventHistory --> StringSnapshot : Привязывается к
    EventHistory --> AnalogSnapshot : Привязывается к
    EventHistory --> DiscreteSnapshot : Привязывается к
    EventTagPendingDelete --> TimeDetectorDetailPendingDelete : Имеет
    
```

The diagram illustrates the relationships between various components of a detection system. Key entities and their attributes include:

- Tag**: tagName, ioserverkey, storagenodekey, wwtagkey, topickey, description, acquisitiontype, storagetype, acquisitionrate, storagerate, itemname, tagtype, widecolumnname, timedeadband, datecreated, createdby, currenteditor, samplesinactivelimage.
- SummaryOperation**: operationkey, tagName, calcType, description, duration, resolution, timestamp.
- EventTag**: tagName, detectorTypeKey, actionTypeKey, scanRate, timeDeadband, logged, status, postDetectorDelay, useThreadPool, detectorString, actionString, priority, edge.
- DetectorType**: detectorTypeKey, name, description, editorClassName, detectorClassName, executionMode.
- EventHistory**: eventLogKey, tagName, dateTime, detectDateTime, edge.
- StringSnapshot**: snapshotTagKey, eventLogKey, value, quality, qualityDetail.
- AnalogSnapshot**: snapshotTagKey, eventLogKey, value, quality, qualityDetail.
- DiscreteSnapshot**: snapshotTagKey, eventLogKey, value, quality, qualityDetail.
- ActionTag**: operationKey, tagName, calcType, description, duration, resolution, timestamp.
- Frequency**: frequencyID, frequency.
- TimeDetectorDetail**: timeDetectorDetailKey, frequencyID, tagName, periodicity, startDateTime, runTimeDay, runTimeHour, runTimeMin.
- ActionType**: actionTypeKey, name, description, editorClassName, actionClassName.
- EventTagPendingDelete**: tagName, detectorTypeKey, actionTypeKey, scanRate, timeDeadband, logged, status, postDetectorDelay, useThreadPool, detectorString, actionString, priority, edge.
- TimeDetectorDetailPendingDelete**: tagName, frequencyID, periodicity, startDateTime, runTimeDay, runTimeHour, runTimeMin.

Relationships are defined as follows:

- Tag** is associated with **SummaryOperation** (Чем активизируется).
- Tag** is associated with **EventTag** (Тип).
- EventTag** is associated with **DetectorType** (Тип).
- EventTag** is associated with **ActionTag** (Тип).
- EventTag** is associated with **EventHistory** (Регистрируется как).
- EventTag** has **StringSnapshot** (Имеет).
- EventTag** has **AnalogSnapshot** (Имеет).
- EventTag** has **DiscreteSnapshot** (Имеет).
- EventTag** has **EventTagPendingDelete** (Имеет).
- EventTag** has **TimeDetectorDetailPendingDelete** (Имеет).
- EventHistory** is associated with **StringSnapshot** (Привязывается к).
- EventHistory** is associated with **AnalogSnapshot** (Привязывается к).
- EventHistory** is associated with **DiscreteSnapshot** (Привязывается к).
- EventTagPendingDelete** is associated with **TimeDetectorDetailPendingDelete** (Имеет).

FK = Внешний ключ
AK = Альт. внешний ключ

Таблица ActionType

Содержит одну строку для каждого типа действий над событиями. Для каждого действия существует свой компонент, который запускается сервером как объект COM и выполняет все процедуры по выполнению выбранного действия. Настраивать действия для событий необязательно. Тэги действий и событий соединяются через атрибут ActionTypeKey.

Атрибут	Тип данных	Объяснение
ActionTypeKey	int	Уникальный идентификатор типа действия. ActionTypeKey принадлежит к столбцу типа IDENTITY.
Name	varchar(33)	Название типа действия.
Description	varchar(50)	Описание действия,
EditorClassName	varchar(80)	Название, под которым компонент действия известен приложению-клиенту, такому как программа InSQL Configure, служащая для визуального представления действия.
ActionClassName	varchar(80)	Название, под которым компонент действия (объект COM) известен системе, служит для выполнения этого действия.

Примечание. Система управления событиями использует следующие значения ActionTypeKey, которые получены во время установки: 1 = Действия нет. 2 = Общий SQL; 3 = Снимок; 4 = E-mail; 5 = Мертвая зона; 6 = Сводка.

Таблица AnalogSnapshot

Содержит одну строку для каждого значения аналогового тэга, которая сохраняется при происхождении заданного события. Обычно в этой таблице содержатся значения заранее созданного списка аналоговых тэгов на момент времени фиксации события.

Атрибут	Тип данных	Объяснение
SnapshotTagKey	int	Уникальный числовой идентификатор "отснятого" тэга, соответствующий таблице <i>SnapshotTag</i> .
EventLogKey	int	Уникальный числовой идентификатор происхождения события, включившего "снятие", полученный из таблицы <i>EventHistory</i> .
Value	float(25)	Значение тэга в момент даты/времени происхождения события. Указывается в единицах измерения.
Quality	tinyint	Основной показатель качества соответствующий значению аналогового "снимка". Подробнее в Приложении Б, "Качество данных" .
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных" .

Таблица **DetectorType**

Содержит одну строку для каждого типа детектора событий. Для каждого детектора существует свой компонент, который запускается сервером как объект COM и выполняет все процедуры по определению возникновения события в системе. Один детектор может быть связан с несколькими тэгами. Тэги событий связываются через *DetectorTypeKey*.

Атрибут	Тип данных	Объяснение
DetectorTypeKey	int	Уникальный идентификатор типа детектора. Столбец DetectorTypeKey относится к типу IDENTITY.
Name	varchar(33)	Имя, присваиваемое типу детектора.
Description	varchar(50)	Описание детектора
EditorClassName	varchar(80)	Название, под которым компонент детектора известен приложению-клиенту, такому как InSQL Configure, используется для наглядного отображения детектора.
DetectorClassName	varchar(80)	Название, под которым компонент детектора (объект COM) известен системе, служит для выполнения фиксации.
ExecutionMode	tinyint	Определяет режим работы детектора. 0 = Циклически выполняется системой управления событиями при любой скорости сканирования. 1 = Асинхронно. Не выполняется системой управления событиями. Обычно, эти события включаются какой-либо внешней программой (например, InTouch).

Примечание. Система управления событиями опирается на следующие полученные при установке значения DetectorTypeKey: 1 = Система; 2 = Внешнее событие; 2 = Общий SQL; 4 = Особое аналоговое значение; 5 = Особое дискретное значение; 6 = Основанное на времени (планировщик).

Таблица DiscreteSnapshot

Содержит одну строку для каждого дискретного тэга, значение которого будет сохранено при происхождении определенного события. Обычно в этой таблице содержатся значения заранее созданного списка дискретных тэгов на момент времени фиксации события.

Атрибут	Тип данных	Объяснение
SnapshotTagKey	int	Уникальный числовой идентификатор "отснятого" тэга, полученный из таблицы <i>SnapshotTag</i> .
EventLogKey	int	Уникальный числовой идентификатор происхождения события, зафиксированный на "снимке", полученном из таблицы <i>EventHistory</i> .
Value	tinyint	Состояние дискретного тэга на момент даты/времени. 0 = ЛОЖЬ; 1 = ИСТИНА.
Quality	tinyint	Основной показатель качества соответствующий значению дискретного "снимка" Подробнее в Приложении Б, "Качество данных" .
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных" .

Таблица EventHistory

[Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы"](#).

Таблица EventTag

[Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы"](#).

Таблица EventTagPendingDelete

Содержит одну строку для каждого тэга события, который ожидает удаления. Эта таблица постоянно используется системой при процессе удаления. Столбцы в этой таблице такие же, как в таблице *EventTag*.

[Подробнее о таблице *EventTag* см. в главе 2 "Тематические области и общие таблицы" этого руководства.](#)

Таблица Frequency

Содержит одну строку для каждой доступной частоты операций создания сводок. Частотой называется конкретная частота, с которой должны создаваться сводки.

Атрибут	Тип данных	Объяснение
FrequencyID	int	Идентификатор типа частоты фиксации времени. 1= Ежечасно; 2 = Ежедневно; 3 = Еженедельно; 4 = Ежемесячно; 5 = Периодически; 6 = Иная (для дальнейшего использования). FrequencyID относится к типу IDENTITY.
Frequency	varchar(12)	Название частоты.

Таблица SnapshotTag

Содержит одну строку для каждого тэга снимка, связанного с заданным тэгом события.

Атрибут	Тип данных	Объяснение
SnapshotTagKey	int	Уникальный числовой идентификатор "отснятого" тэга определенного события. Данное значение автоматически получается системой. Столбец SnapshotTagKey относится к типу IDENTITY.
TagName	varchar(33)	Имя тэга снимка, связанного со снимком события. Этот тэг будет "отснят".
EventTagName	varchar(33)	Имя тэга, к которому относится снимок.
TagType	int	Определяет тип "отснятого" тэга. 1 = Аналоговый; 2 = Дискретный; 3 = Текстовый.

Таблица SQLTemplate

Содержит одну строку для каждого готового сценария SQL, который может быть скопирован и использован в качестве сценария фиксации события. Таблица SQLTemplate располагает готовыми функциями установки базовых событий.

Атрибут	Тип данных	Объяснение
TemplateKey	int	Уникальный числовой идентификатор шаблона SQL. Данное значение автоматически получается системой. Столбец TemplateKey относится к типу IDENTITY. TemplateKey есть первичный ключ.
Description	varchar(50)	Описание шаблона SQL.
Type	int	Тип шаблона SQL. 0 = Детектор; 1 = Действие.
Script	text	Готовый сценарий SQL. Сценарий может быть скопирован и использован как детектор события или сценарий действия.

Таблица StringSnapshot

Содержит одну строку для каждого текстового тэга, значение которого сохраняется при происхождении определенного события. Обычно в этой таблице содержатся значения из готового списка текстовых тэгов на момент происхождения события.

Атрибут	Тип данных	Объяснение
SnapshotTagKey	int	Уникальный числовой идентификатор "отснятого" тэга, полученный из таблицы <i>SnapshotTag</i> .
EventLogKey	int	Уникальный числовой идентификатор происхождения события, зафиксированного на "снимке", полученном из таблицы <i>EventHistory</i> .
Value	varchar(131)	Значение текстового тэга на момент даты/времени происхождения события.
Quality	tinyint	Основной показатель качества соответствующий значению текстового "снимка". Подробнее в Приложении Б, "Качество данных" .
QualityDetail	int	Информация о значении показателя качества. Подробнее в Приложении Б, "Качество данных" .

Таблица Tag

[Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы"](#).

Таблица TimeDetectorDetail

Содержит, по крайней мере, одну строку для каждого тэга события, соответствующего детектору времени.

Атрибут	Тип данных	Объяснение
TimeDetectorDetailKey	int	Уникальный числовой идентификатор каждого тэга события времени. Данное значение получается системой автоматически. Столбец TimeDetectorDetailKey относится к типу IDENTITY.
FrequencyID	int	Уникальный числовой идентификатор для частоты детектора/ Используется для связи частоты с детектором времени. 1= Ежечасно; 2 = Ежедневно; 3 = Еженедельно; 4 = Ежемесячно; 5 = Периодическая; 6 = Иная (для дальнейшего использования).
TagName	Varchar(33)	Уникальное имя тэга, связанного с детектором времени
Periodicity	int	Интервал в минутах между событиями детектора. Используется только для периодического определения.
StartDateTime	datetime	Метка даты и времени запуска детектора времени. Используется только для периодического определения.
RunTimeDay	tinyint	В контексте еженедельного детектора RunTimeDay задает номер дня недели (0 = воскресенье - 6 - суббота). В контексте ежемесячного детектора RunTimeDay задает день месяца. Не используется для периодического определения.
RunTimeHour	tinyint	Час дня, в котором запустится детектор времени. Не используется для периодического определения.
RunTimeMin	tinyint	Минута в течении часа, когда запускается детектор времени. Не используется для периодического определения.

Таблица TimeDetectorDetailPendingDelete

Содержит одну строку для каждого детектора времени, ожидающего удаления. Столбцы в этой таблице такие же, как в таблице *TimeDetectorDetail*.

Подробности см. в разделе *"InTouch"* в начале этой главы.

Таблицы интеграции InTouch

Эта глава содержит алфавитный справочник таблиц интеграции InTouch в базе данных IndustrialSQL Server. Таблицы интеграции InTouch определяют совместную работу InTouch и IndustrialSQL Server. Глава рассказывает о таблицах интеграции InTouch, типах их данных и атрибутах этих таблиц.

Содержание

- Таблицы интеграции InTouch
- Таблица InTouchNode
- Таблица InTouchSpecific
- Таблица Tag
- Таблица TopicImportInfo

Таблицы интеграции InTouch

Таблицы интеграции InTouch содержат информацию о данных, импортированных в IndustrialSQL Server из InTouch.

К таблицам интеграции InTouch относятся:

- *InTouchNode*
- *InTouchSpecific*
- *Tag*
- *TopicImportInfo*

Схема базы данных для таблиц интеграции InTouch выглядит следующим образом:

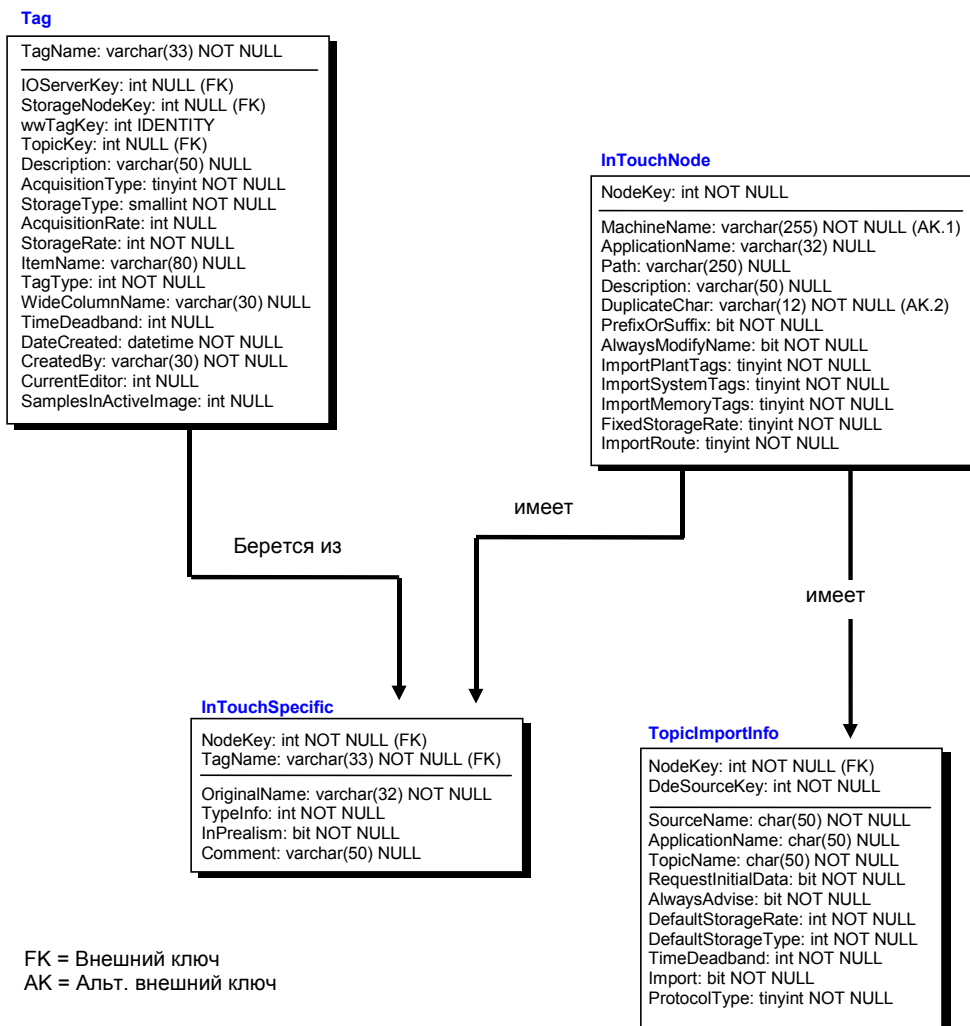


Таблица InTouchNode

Содержит одну строку для каждого узла InTouch, из которого данные импортируются в IndustrialSQL Server.

Атрибут	Тип данных	Объяснение
NodeKey	int	Уникальный числовой идентификатор для имеющего имя узла. Данное значение получается системой автоматически. Столбец NodeKey относится к типу IDENTITY.
MachineName	varchar(255)	Имя компьютера, на котором запущено приложение InTouch.
ApplicationName	varchar(255)	Название приложения InTouch.
Path	varchar(250)	Путь UNC к файлу InTouch TAGNAME.X .
Description	varchar(50)	Описание узла InTouch.
DuplicateChar	varchar(12)	Символ, добавленный в начало или конец имени тэга для его унификации. Добавление символа определяется функцией "Префикса или Суффикса".
PrefixOrSuffix	bit	Проверяет, был ли уникальный тэг создан путем добавления символа в начало его имени или же в конец (1). 1 = В конец; 0 = В начало.
AlwaysModifyName	bit	Проверяет, был ли унифицирующий символ добавлен во все тэги узла. 1 = Добавлен; 0 = Не добавлен.
ImportPlantTags	tinyint	Проверяет, были ли импортированы производственные тэги. В InTouch производственные тэги называются DDE-тэгами). 0 = Не импортированы; 1 = Импортированы все; 2 = Импортированы зарегистрированные тэги.
ImportSystemTags	tinyint	Проверяет, были ли импортированы системные тэги. 0 = Не импортированы; 1 = Импортированы все; 2 = Импортированы зарегистрированные тэги.
ImportMemoryTags	tinyint	Проверяет, были ли импортированы внутренние тэги. 0 = Не импортированы; 1 = Импортированы все; 2 = Импортированы зарегистрированные тэги.

(продолжение)

Атрибут	Тип данных	Объяснение
FixedStorageRate	tinyint	Скорость циклического сохранения импортированных тэгов в секундах. 0 = Задействована скорость сохранения с настройками для тем; 1- <i>n</i> = Циклические тэги получили новый стандарт скорости сохранения. Эта скорость превышает первоначально настроенную для тем скорость сохранения.
ImportRoute	tinyint	Определяет тип импортирования, последний раз предпринятого на этом узле. 1 = Импортирование произведено с помощью стандартных настроек, поддерживаемых системой; 2 = Импортирование произведено с помощью стандартных настроек предыдущего импортирования; 3 = Администратору позволено редактировать опции импорта, основанные на поддерживаемых системой стандартах; 4 = Администратору позволено редактировать опции импорта, основанные на опциях предыдущего импортирования.

Таблица InTouchSpecific

Содержит одну строку относящейся к импорту информации для каждого импортированного из InTouch тэга.

Атрибут	Тип данных	Объяснение
TagName	varchar(33)	Имя тэга, импортированного в систему (это InSQL-имя тэга).
NodeKey	int	Уникальный числовой идентификатор имеющего имя узла InTouch. Этот идентификатор берется из таблицы <i>InTouchNode</i> .
OriginalName	varchar(32)	Оригинальное имя тэга InTouch. Это имя может отличаться от имени в IndustrialSQL Server, если для унификации было создано новое имя.
TypeInfo	int	Уточняет тип тэга. 1 = DDE-целый; 2 = DDE-действительный; 3 = Внутренний целый; 4 = Внутренний Действительный; 5 = DDE-дискретный; 6 = Внутренний Дискретный; 7 = DDE-текстовый; 8 = Внутренний текстовый.
InPrealism	bit	Уточняет, была ли информация импортирована из InTouch в структуры данных IndustrialSQL Server.
Comment	varchar(50)	Первоначальное описание тэга, импортированного из InTouch.

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица TopicImportInfo

Содержит одну строку для каждой темы, импортированной из таблицы узла InTouch.

Атрибут	Тип данных	Объяснение
NodeKey	int	Уникальный идентификатор имеющего имя узла. Берется из таблицы <i>InTouchNode</i> .
DdeSourceKey	int	Уникальный идентификатор источника DDE. Выдается системой IndustrialSQL Server при импортировании данных.
SourceName	char(50)	Имя доступа к DDE, взятое из InTouch.
ApplicationName	char(50)	Название приложения InTouch, из которого импортируется тема.
TopicName	char(50)	Название импортированной темы.
RequestInitialData	bit	Определяет, настроена ли тема для запроса исходных данных. Подробнее см. в <i>Руководстве пользователя InTouch</i> .
AlwaysAdvise	bit	Определяет, настроена ли тема для опроса всех элементов данных. Подробнее см. в <i>Руководстве пользователя InTouch</i> .
DefaultStorageRate	int	Скорость циклического сохранения темы. Запоминается тэг-импортирующим элементом IndustrialSQL Server, основанным на настройках импортируемых данных.
DefaultStorageType	int	Стандартный тип сохранения темы. Запоминается тэг-импортирующим элементом IndustrialSQL Server, основанным на настройках импортируемых данных. Сейчас поддерживается только циклическое (1) сохранение.
TimeDeadband	int	При дельта-сохранении это временной промежуток сохранения тэга. Сейчас не поддерживается.
Import	bit	Проверяет, была ли тема ранее импортирована из InTouch в IndustrialSQL Server.
ProtocolType	tinyint	Протокол общения с сервером ввода-вывода. 1 = DDE; 2 = SuiteLink; 3 = Драйвер IndustrialSQL Server (для совместимости с IndustrialSQL Server версии 3.0 и более ранних).

Г Л А В А 6

Таблицы отслеживания изменений

Эта глава содержит алфавитный справочник таблиц отслеживания изменений в базе данных IndustrialSQL Server. Таблицы отслеживания изменений — это таблицы, которые отслеживают и сохраняют изменения в столбцах базы данных. Здесь рассказывается о таблицах отслеживания изменений, типах их данных и атрибутах.

См. более подробно в Руководстве администратора системы IndustrialSQL Server.

Содержание

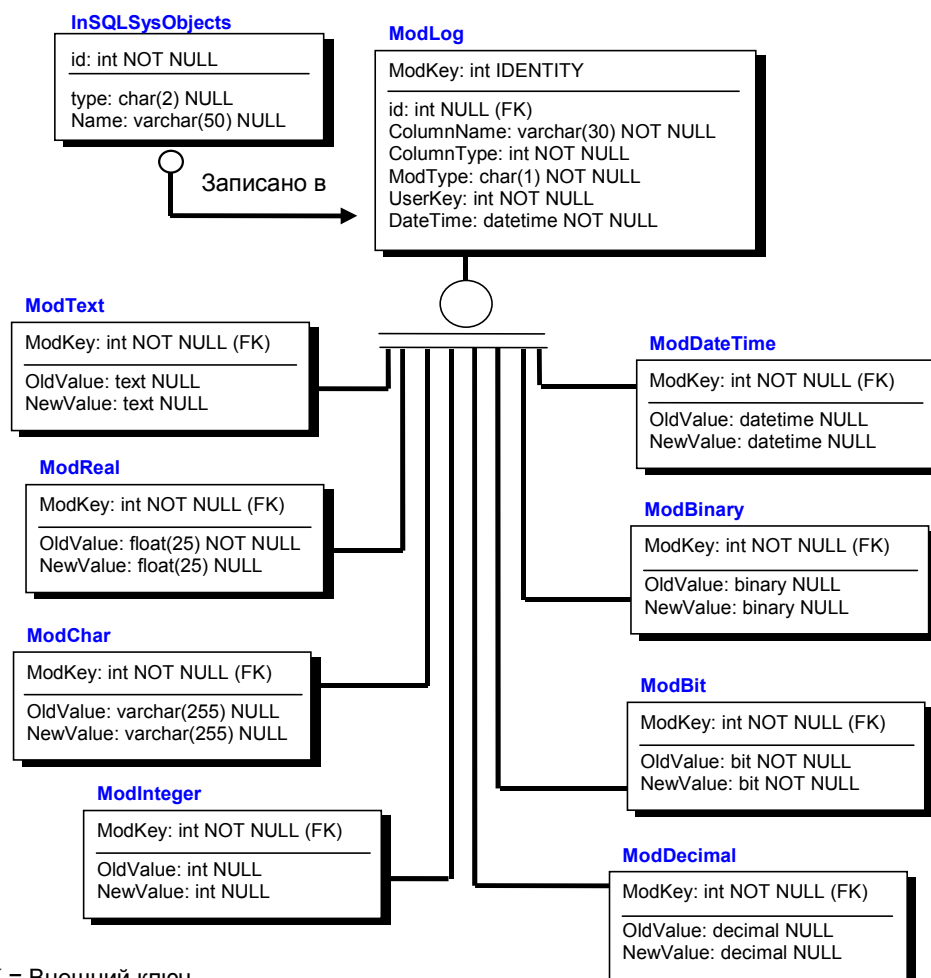
- Таблицы отслеживания изменений
- Таблица InSQLSysObjects
- Таблица ModBinary
- Таблица ModBit
- Таблица ModChar
- Таблица ModDateTime
- Таблица ModDecimal
- Таблица ModInteger
- Таблица ModLog
- Таблица ModReal
- Таблица ModText

Таблицы отслеживания изменений

Таблицы отслеживания изменений содержат информацию о данных, импортированных в IndustrialSQL Server из InTouch.

- *InSQLSysObjects*
- *ModBinary*
- *ModBit*
- *ModChar*
- *ModDateTime*
- *ModDecimal*
- *ModInteger*
- *ModLog*
- *ModReal*
- *ModText*

Схема базы данных для таблиц отслеживания изменений выглядит следующим образом:



FK = Внешний ключ

AK = Альт. внешний ключ

Таблица InSQLSysObjects

Содержит одну строку для объекта базы данных, для которого отслеживаются изменения.

Атрибут	Тип данных	Объяснение
id	int	Уникальный идентификатор объекта.
Type	char(2)	Тип объекта C = ПРОВЕРКА; D = СТАНДАРТ; F = ВНЕШНИЙ КЛЮЧ; K = ПЕРВИЧНЫЙ КЛЮЧ или УНИКАЛЬНЫЙ; L = Журнал; P = Хранимая процедура; R = Правило; RF = Хранимая для тиражированная процедура; S = Системная таблица; TR = Включатель; U = Пользовательская таблица; V = Обзор; X = Расширенная процедура хранения.
Name	varchar(50)	Имя объекта.

Таблица ModBinary

Содержит одну строку для каждого столбца базы данных двоичного типа, в котором проводились операции INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	binary	Значение, записанное в столбец до произведения изменения.
NewValue	binary	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModBit

Содержит одну строку для всех столбцов базы данных битового типа, над которыми были проведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	bit	Значение, записанное в столбец до произведения изменения.
NewValue	bit	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModChar

Содержит одну строку для каждого столбца базы данных (любой длины) символьного типа или переменного-символьного типа, в котором проводились операции INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	varchar(255)	Значение, записанное в столбец до произведения изменения.
NewValue	varchar(255)	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModDateTime

Содержит одну строку для всех столбцов базы данных типа *datetime* (*datetime*, *smalldatetime*), в которых была проведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	datetime	Значение, записанное в столбец до произведения изменения.
NewValue	datetime	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModDecimal

Содержит одну строку для всех столбцов базы данных десятичного типа, над которыми были проведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	decimal	Значение, записанное в столбец до произведения изменения.
NewValue	decimal	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModInteger

Содержит одну строку для каждого столбца базы данных целого типа (*tinyint*, *smallint* или *int*) в котором была произведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	int	Значение, записанное в столбец до произведения изменения.
NewValue	int	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModLog

Содержит одну строку для каждого столбца, в котором была проведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
Id	int	Идентификатор измененного столбца.
ColumnName	varchar(30)	Имя измененного столбца.
ColumnType	int	Тип измененного столбца.
ModType	char(1)	Тип изменения U = ОБНОВЛЕНИЕ; I = ВСТАВКА; D = УДАЛЕНИЕ.
UserKey	int	Уникальный числовой идентификатор пользователя базы данных. Вводится вручную администратором базы данных.
DateTime	DateTime	Метка даты и времени произведения изменения.

Таблица ModReal

Содержит одну строку для всех столбцов базы данных действительного типа или *типа значений с плавающей точкой*, в которых была проведена операция INSERT, UPDATE или DELETE.

Атрибут	Тип данных	Объяснение
ModKey	int	Уникальный числовой идентификатор изменения. Данное значение получается системой автоматически. Столбец <i>ModKey</i> относится к типу IDENTITY.
OldValue	float(25)	Значение, записанное в столбец до произведения изменения.
NewValue	float(25)	Новое значение, сохраняемое в особом отдельном столбце.

Таблица ModText

Примечание. Так как включатели в столбцах типа text не поддерживаются системой Microsoft SQL Server, изменения текста не отслеживаются в IndustrialSQL Server.

Г Л А В А 7

Таблицы именных пространств и групп

Эта глава содержит алфавитный справочник таблиц именных пространств и групп в базе данных IndustrialSQL Server. Таблицами именных пространств и групп называются таблицы, определяющие и сохраняющие информацию об иерархии системы и тэгах. Здесь рассказывается обо всех типах данных и атрибутах этих таблиц.

Содержание

- Таблицы именных пространств и групп
- Таблица GroupTagList
- Таблица NameSpaceIcons
- Таблица PrivateGroupTag
- Таблица PrivateNameSpace
- Таблица PublicGroupTag
- Таблица PublicNameSpace
- Таблица ServerList
- Таблица SystemNameSpace
- Таблица Tag
- Таблица TagGroup
- Таблица TagRef
- Таблица UserDetail

Таблицы именных пространств и групп

Таблицы именных пространств и групп содержат информацию, определяющую место элементов в иерархии именных пространств и в определениях групп тэгов. К таблицам именных пространств и групп относятся следующие:

- *GroupTagList*
- *NameSpaceIcons*
- *PrivateGroupTag*
- *PrivateNameSpace*
- *PublicGroupTag*
- *PublicNameSpace*
- *ServerList*
- *SystemNameSpace*
- *Tag*
- *TagGroup*
- *TagRef*
- *UserDetail*

Схема баз данных для таблиц групп и именных пространств выглядят следующим образом:

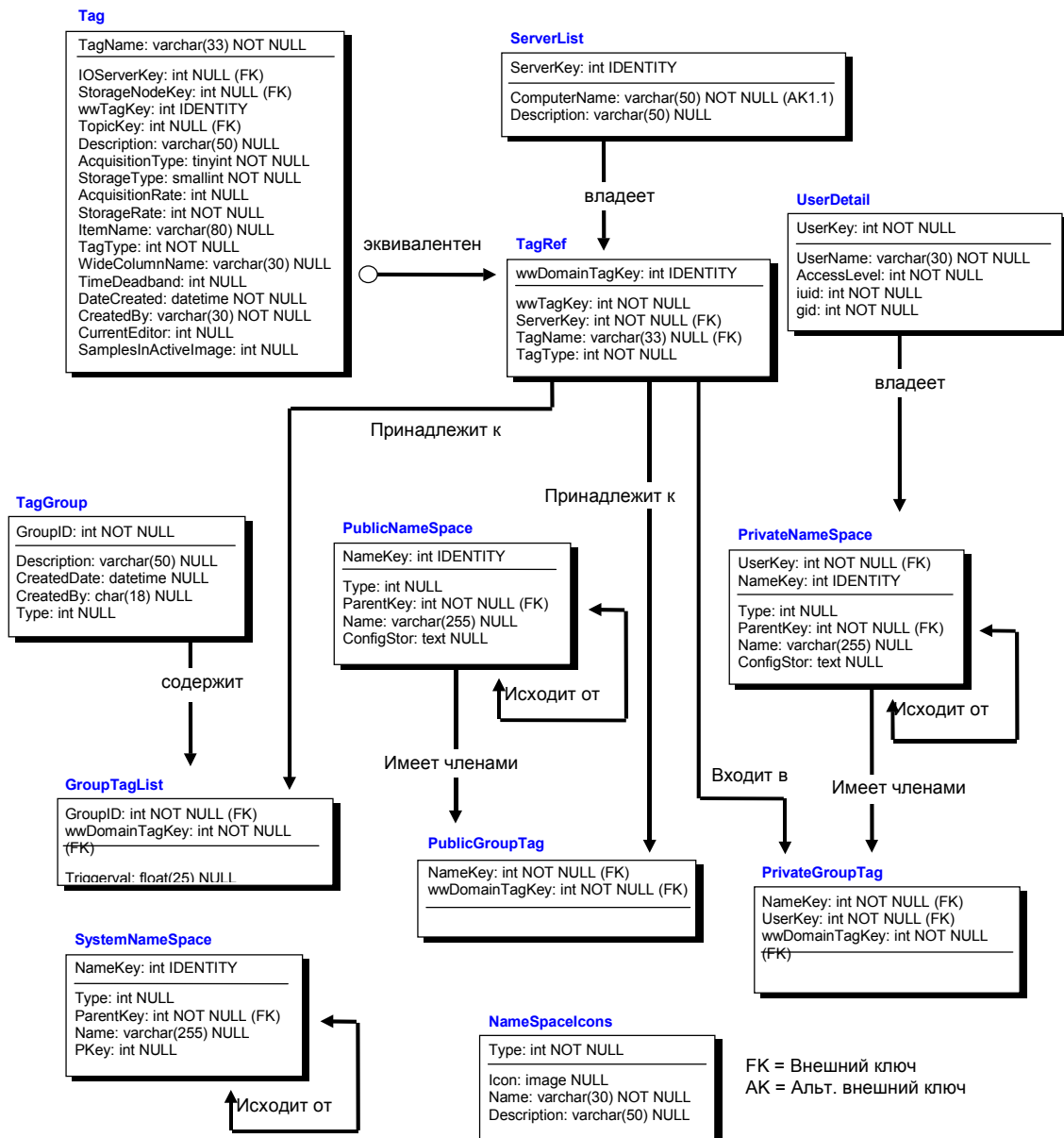


Таблица GroupTagList

Содержит одну строку для каждой определенной группы тэгов.

Атрибут	Тип данных	Объяснение
GroupID	int	Идентификатор группы. тэгов. Берется из таблицы <i>TagGroup</i> .
WwDomainTagKey	int	Уникальный идентификатор тэга определенного домена. Берется из таблицы <i>TagGroup</i> .
Triggerval	float(25)	Распознается как значение триггера.

Таблица NameSpacelcons

Содержит одну строку для каждой определенной пиктограммы именных пространств. Пиктограммы именных пространств могут быть показаны в приложении броузера для каждого уровня имен (системное, общее и личное).

Атрибут	Тип данных	Объяснение
Type	int	Значение, определяющее тип имени. От 1 до 6 = Тэг; От 1 до 2 миллионов = Системный; 2 и более миллионов = Группы.
Icon	image	Пиктограмма именного пространства.
Name	varchar(30)	Имя пиктограммы.
Description	varchar(50)	Описание пиктограммы.

Таблица PrivateGroupTag

Содержит одну строку для каждой группы тэгов в личном именном пространстве.

Атрибут	Тип данных	Объяснение
NameKey	int	Уникальный идентификатор элемента личного именного пространства.
UserKey	int	Уникальный идентификатор пользователя базы данных, создавшего личную группу тэгов.
WwDomainTagKey	int	Уникальный идентификатор тэга определенного домена. Берется из таблицы <i>TagRef</i> .

Таблица PrivateNameSpace

Содержит одну строку для каждого элемента личного именного пространства. Элементы именного пространства, имеющие свою иерархию, включают в себя производственные компьютеры, области, тэги и т.п. Позволяет подключить к одному тэгу более одного имени.

Атрибут	Тип данных	Объяснение
UserKey	int	Уникальный числовой идентификатор пользователя базы данных. Берется из таблицы <i>UserDetail</i> .
NameKey	int	Уникальный идентификатор элемента личного именного пространства.
Type	int	Тип имени в иерархии. От 1 до 6 = Тэг; От 1 до 2 миллионов = Системный; 2 и более миллионов = Группы.
ParentKey	int	Уникальный идентификатор элемента, имеющего название.
Name	varchar(255)	Имя элемента в иерархии.
ConfigStor	текст	Имя файла, содержащего информацию установки, если элемент именного пространства связан с какими-то установками (например, установки для трендов).

Таблица PublicGroupTag

Содержит одну строку для каждой группы тэгов в общем именном пространстве.

Атрибут	Тип данных	Объяснение
NameKey	int	Уникальный идентификатор элемента, имеющего имя. Берется из таблицы <i>PublicNameSpace</i> .
WwDomainTagKey	int	Уникальный числовой идентификатор тэга определенного домена. Берется из таблицы <i>TagRef</i> .

Таблица PublicNameSpace

Содержит одну строку для каждого элемента общего именного пространства. Элементы общего именного пространства, имеющие свою иерархию, включают в себя производственные компьютеры, области, тэги и т.п. Позволяет подключить к одному тэгу более одного имени.

Атрибут	Тип данных	Объяснение
NameKey	int	Уникальный идентификатор элемента, имеющего имя. Данное значение получается системой автоматически. Столбец NameKey относится к типу IDENTITY.
Type	int	Тип имени в иерархии. От 1 до 6 = Тэг; От 1 до 2 миллионов = Системный; 2 и более миллионов = Группы.
ParentKey	int	Уникальный идентификатор элемента, имеющего имя.
Name	varchar(255)	Имя данного элемента в иерархии.
ConfigStor	текст	Имя файла, содержащего информацию установки, если элемент именного пространства связан с какими-то установками (например, установки для трендов).

Таблица ServerList

Содержит одну строку для каждого сервера, используемого в системе предприятия. Позволяет создать системное именное пространство, содержащее список серверов и имен тэгов для каждого сервера.

Атрибут	Тип данных	Объяснение
ServerKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. Данное значение получается системой автоматически. Столбец ServerKey относится к типу IDENTITY.
ComputerName	varchar(50)	Имя компьютера-сервера с таблицами базы данных, к которым требуется доступ.
Description	varchar(50)	Описание сервера.

Таблица SystemNameSpace

Содержит одну строку для каждого элемента системного именного пространства. Элементы системного именного пространства, имеющие свою иерархию, включают в себя серверы, темы, пользователей и т.п.

Атрибут	Тип данных	Объяснение
NameKey	int	Уникальный идентификатор элемента, имеющего имя. Данное значение получается системой автоматически. Столбец NameKey относится к типу IDENTITY.
Type	int	Тип имени в иерархии. От 1 до 6 = Тэг; От 1 до 2 миллионов = Системный; 2 и более миллионов = Группы.
ParentKey	int	Уникальный идентификатор элемента, имеющего имя.
Name	varchar(255)	Имя этого элемента в иерархии.
PKey	int	Справка по первичным ключам для других таблиц.

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица TagGroup

Содержит одну строку для каждой определенной группы тэгов. Группа тэгов есть простое, не имеющее иерархии объединение тэгов, которые могут быть использованы системой. Группы тэгов могут использоваться для детекторов событий, действий и т.п.

Атрибут	Тип данных	Объяснение
GroupID	int	Идентификатор группы тэгов.
Description	varchar(50)	Описание группы тэгов.
CreatedDate	datetime	Дата создания группы тэгов.
CreatedBy	char(18)	Имя пользователя или приложения, создавшего группу тэгов.
Type	int	Тип группы тэгов. От 1 до 100 = Для системного использования. 100 и выше = Последние пользователи и сторонние приложения-клиенты.

Таблица TagRef

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица UserDetail

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Г Л А В А 8

Таблицы сводок

Эта глава содержит алфавитный справочник таблиц сводок, управляемых системой управления событиями. Эти таблицы содержат информацию, связанную с настройкой, созданием и хранением данных автоматических сводок. Здесь рассказывается все о типах данных и атрибутах таблиц сводок.

См. подробности в Руководстве администратора системы IndustrialSQL Server,

Содержание

- Таблицы сводок
- Таблица CalcType
- Таблица SummaryData
- Таблица SummaryHistory
- Таблица SummaryOperation
- Таблица SummaryTagList
- Таблица Tag

Таблицы сводок

Таблицы сводок управляются таблицами системы управления событиями и обеспечивают автоматическое создание сводок по данным аналоговых, дискретных и текстовых тэгов.

Информация сводок используется для многих задач, в том числе:

- Длительное хранение данных. Даже средних размеров система может сохранять сводки дня на протяжении многих лет. Это позволяет в любой момент получить доступ к данным с малым разрешением на период времени больший, чем в случае нормального полного разрешения.
- Создание отчетов. Для большинства отчетов агрегированные данные более важны, чем необработанные. Например, данные о полном объеме работы, сделанной за день, обычно важнее, чем фактическая производительность за то же время.
- Интеграция с бизнес-системами. Доступ к архивам IndustrialSQL с полным разрешением и высокой производительностью и таблицам реального времени лучше обеспечивают инструменты, использующие расширения реального времени языка SQL и входящие в состав Industrial-SQL. Тем не менее, эти расширения SQL поддерживаются не всеми инструментами клиентов. Таблицы сводок позволяют уменьшить количество томов информации до удобного для управления объема, что делает их пригодными для использования в обычных приложениях-клиентах SQL.

Основными компонентами для создания сводок являются:

- Готовые наборы объектов базы данных (таблицы, сохраненные процедуры, стандарты и представления), определяющие и выполняющие операции со сводками.
- Программа InSQL Configure, в которой можно выбрать тэги для формирования сводок, агрегацию и частоту создания. Сводки создаются как отдельные типы действий над событиями.

Таблицы сводок:

- *CalcType*
- *SummaryData*
- *SummaryHistory*
- *SummaryOperation*
- *SummaryTagList*
- *Tag*
- *Event tables as defined in the event system*

Схема базы данных для таблиц сводок выглядит следующим образом:

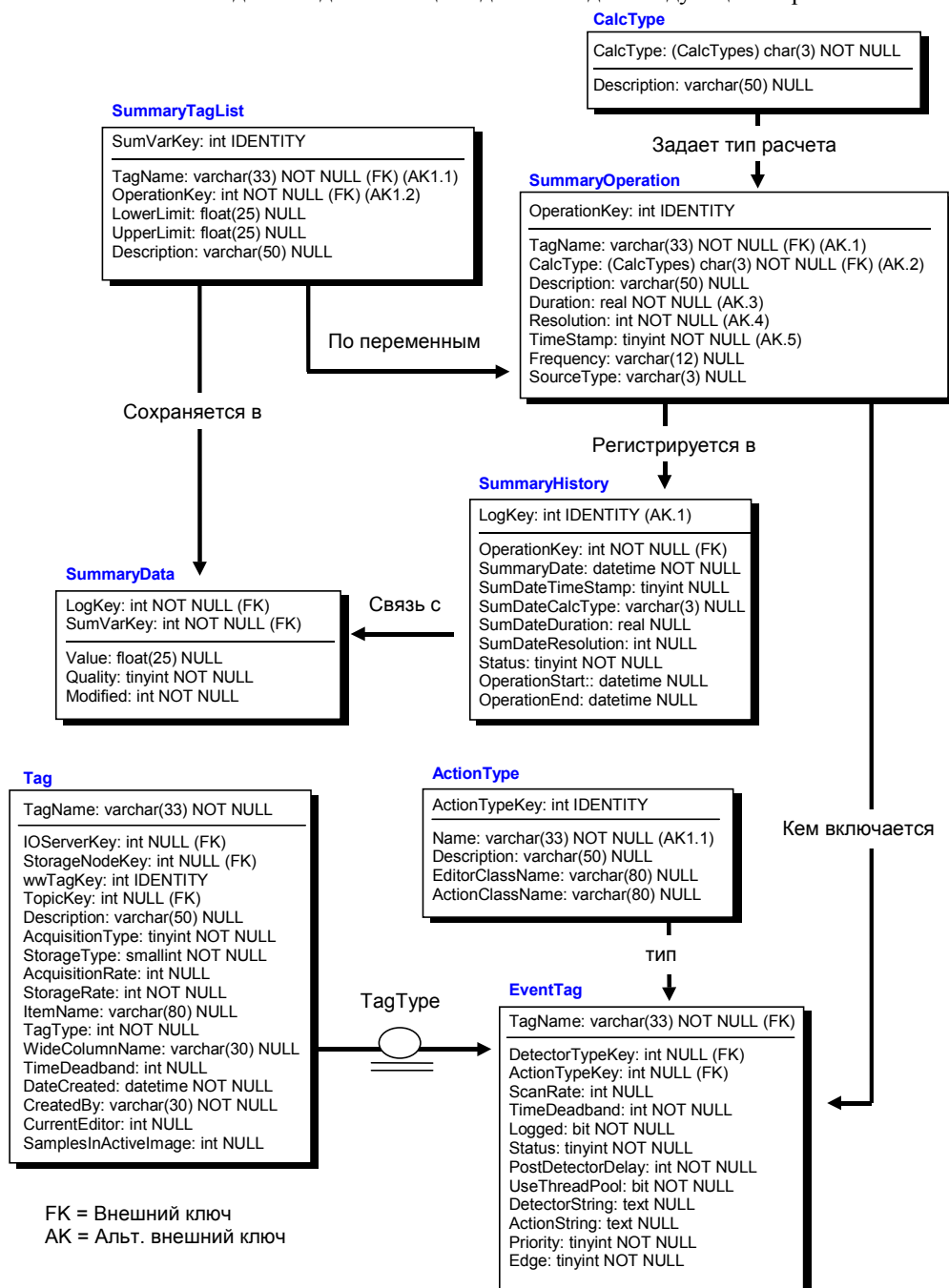


Таблица CalcType

Содержит одну строку для каждого типа подсчета сводок, выполняемого системой управления событиями. Подсчеты могут быть четырех типов: SUM, AVG, MIN, или MAX.

Атрибут	Тип данных	Объяснение
CalcType	Типы подсчетов: char(3)	Выполняемые типы подсчетов: SUM, MAX, MIN, или AVG.
Description	varchar(50)	Описание подсчета.

Таблица SummaryData

Содержит одну строку для каждого подсчитанного значения или результата тэга. Таблица SummaryData может содержать очень большое количество строк.

Атрибут	Тип данных	Объяснение
LogKey	int	Уникальный числовой идентификатор из архивного журнала (таблица <i>SummaryHistory</i>).
SumVarKey	int	Уникальный числовой идентификатор тэга сводки. Берется из таблицы <i>SummaryTagList</i> .
Value	float(25)	Значение сводки.
Quality	tinyint	Показатель качества результата. Высший показатель качества необработанных данных, из которых будет получен результат.
Modified	int	Уточняет, изменялись ли данные. 1 = Изменялись; 0 = Не изменялись.

Таблица SummaryHistory

Содержит одну строку для каждого случая составления сводки. Таблица SummaryHistory ведет журнал выполненных алгоритмов составления сводок. Строка вносится в таблицу при каждом выполнении алгоритма подсчета для определенной операции. Даже если в результате выполнения алгоритма данные не были получены, строки все равно заносятся в таблицу.

Атрибут	Тип данных	Объяснение
LogKey	int	Уникальный числовой идентификатор для архивного журнала сводки. Данное значение получается системой автоматически. Столбец LogKey относится к типу IDENTITY.
OperationKey	int	Уникальный числовой идентификатор операции составления сводки. OperationKey есть иностранный символ из таблицы <i>SummaryOperation</i> .
SummaryDate	datetime	Дата получения результатов подсчета. Это дата начала или окончания периода подсчета.
SumDateTimeStamp	tinyint	Дубликация столбца <i>TimeStamp</i> таблицы <i>SummaryOperation</i> в <i>SummaryDate</i> . Этот столбец позволяет выполнять оригинальные установки временной отметки подсчета в случае перспективы изменения параметров сводки.
SumDateCalcType	Типы подсчетов: varchar(3)	Дубликация столбца <i>CalcType</i> таблицы <i>SummaryOperation</i> в <i>SummaryDate</i> . Этот столбец позволяет выполнять оригинальный способ подсчета в случае перспективы изменения параметров сводки.
SumDateDuration	real	Дубликация столбца <i>Duration</i> таблицы <i>SummaryOperation</i> в <i>SummaryDate</i> . Этот столбец позволяет сохранять продолжительность подсчета в случае перспективы изменения параметров сводки.

(продолжение)

Атрибут	Тип данных	Объяснение
SumDateResolution	int	Дубликация столбца <i>Resolution</i> таблицы <i>SummaryOperation</i> в <i>SummaryDate</i> . Таблица позволяет сохранять оригинальное разрешение выполняемого подсчета в случае перспективы изменения параметров сводки.
Status	tinyint	Флаг, отображающий состояние последней операции. 0 = Операция завершена успешно; Не 0 = Операция все еще выполняется или неудачна. Сейчас не используется.
OperationStart	datetime	Метка даты/времени, отмечающая время начала подсчета.
OperationEnd	datetime	Метка даты/времени, отмечающая время завершения подсчета.

Таблица SummaryOperation

Содержит одну строку для каждого случая составления сводки. Таблица SummaryOperation определяет типы составления сводок, которые будут предприняты системой. Пользовательская редакция не предусмотрена.

Атрибут	Тип данных	Объяснение
OperationKey	int	Уникальный числовой идентификатор создания сводки. Данное значение получается системой автоматически. Столбец OperationKey относится к типу IDENTITY.
TagName	varchar(33)	Имя тэга события, связанного с подсчетом сводки.
CalcType	Типы подсчетов: char(3)	Тип подсчета может быть SUM, MAX, MIN, или AVG. Это значение берется из таблицы CalcType.
Description	varchar(50)	Описание сводки.
Duration	действительный	Период (в секундах), за который должны быть совершены подсчеты.
Resolution	int	Разрешение выборки данных (в миллисекундах), при котором должны проводиться подсчеты.
TimeStamp	tinyint	Определяет, какую временную метку поставить для результата подсчетов. 0 = Начало периода подсчетов; 1 = Завершение периода подсчетов (по умолчанию).
Frequency	varchar(12)	Дубликация столбца Frequency из таблицы Frequency через соединение TimeDetectorDetail.TagName, где TagName — это имя EventTag. Используется для обратной совместимости с Industrial Workbook.
SourceType	varchar(3)	Тип сводки, установленной на 'DYN' (для "динамических" данных). Используется для обратной совместимости с Industrial Workbook.

Таблица SummaryTagList

Содержит одну строку для каждой комбинации тэгов сводок и специфических операций сводок. SummaryTagList есть связующая таблица, позволяющая ассоциировать тэги с типами действий. Эта таблица используется программой InSQL Configure в таких операциях, как удаление тэга, для которого в данный момент создается сводка, из заданной операции сводки (подсчета), связанной со специальным тэгом сводки события.

Атрибут	Тип данных	Объяснение
SumVarKey	int	Уникальный числовой идентификатор тэга готовой сводки. Данное значение получается системой автоматически. Столбец SumVarKey относится к типу IDENTITY.
TagName	varchar(33)	Имя тэга для создания сводки.
OperationKey	int	Уникальный числовой идентификатор создания сводки. Берется из таблицы <i>SummaryOperation</i> .
LowerLimit	float(25)	Нижняя граница допустимых значений тэга. Значения ниже этой границы не обрабатываются при подсчете.
UpperLimit	float(25)	Верхняя граница допустимых значений тэга. Значения выше этой границы не обрабатываются при подсчете.
Description	varchar(50)	Описание тэга готовой сводки. Обычно используется для описания результата операции, причем это описание может соответствовать описанию переменной, над которой проводится операция.

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Г Л А В А 9

Таблицы системной конфигурации

Эта глава является алфавитным справочником таблиц системной конфигурации в базе данных IndustrialSQL Server. Таблицы системной конфигурации определяют настройки IndustrialSQL Server. Таблицы системной конфигурации используются программами InSQL Control, InSQL Configure, а также классами состояний и настроек. В этой главе описаны все атрибуты таблиц системной конфигурации и типы их данных.

Содержание

- Таблицы системной конфигурации
- Таблица IOServer
- Таблица IOServerType
- Таблица NamedSystemParameter
- Таблица QualityMap
- Таблица StorageLocation
- Таблица StorageNode
- Таблица Tag
- Таблица Topic
- Таблица UserDetail
- Таблица WideTableDictionary

Таблицы системной конфигурации

Система IndustrialSQL Server изначально настроена на базе параметров, выбранных при установке. Системные настройки включают в себя определения серверов ввода-вывода, узлы хранения и определения тэгов. Все параметры системы IndustrialSQL Server хранятся в таблицах системной конфигурации.

К таблицам системной конфигурации относятся:

- *IOServer*
- *IOServerType*
- *NamedSystemParameter*
- *QualityMap*
- *StorageLocation*
- *StorageNode*
- *Tag*
- *Topic*
- *UserDetail*
- *WideTableDictionary*

Схема базы данных для таблиц конфигурации системы выглядит следующим образом:

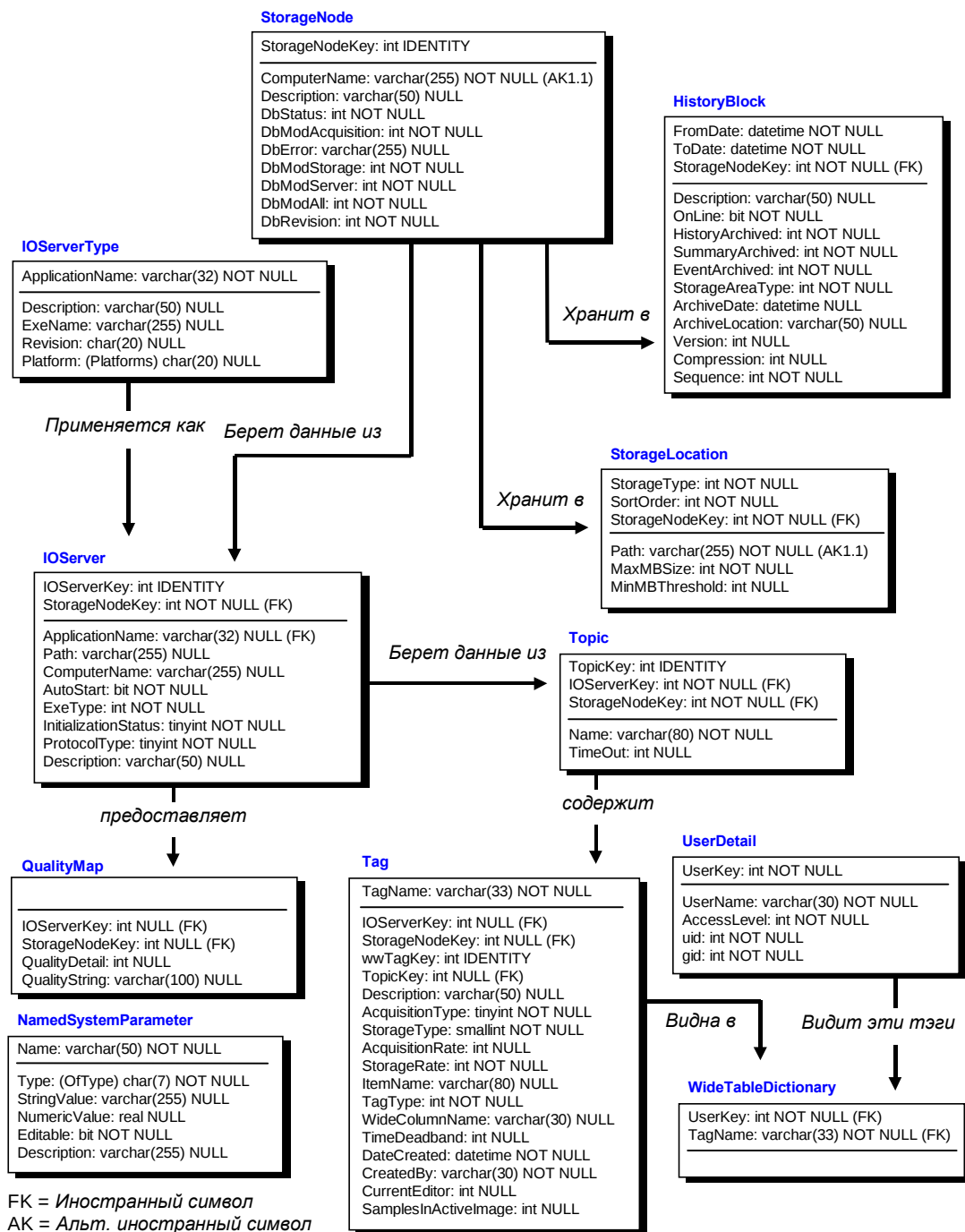


Таблица IOServer

Содержит одну строку для каждого сервера ввода-вывода, передающего данные в систему IndustrialSQL Server. Атрибуты в этой таблице позволяют системе обнаружить функции сообщения и запуска сервера ввода-вывода. Таблица IOServer используется системой сбора данных для определения источника данных.

Атрибут	Тип данных	Объяснение
IOServerKey	int	Уникальный числовой идентификатор для сервера <i>ввода-вывода</i> . Данное значение получается системой автоматически. Столбец IOServerKey относится к типу IDENTITY.
StorageNodeKey	int	Уникальный числовой идентификатор для узла хранения. Берется из таблицы <i>StorageNode</i> .
ApplicationName	varchar(32)	Имя приложения для сервера <i>ввода-вывода</i> . Это имя обычно такое же, как и имя исполнимого файла. Этот идентификатор берется из таблицы <i>IOServerType</i> .
Path	varchar(255)	Если символ типа сервера <i>ввода-вывода</i> не указан, то это полный путь UNC (включая имя файла) для нахождения исполнимого файла сервера <i>ввода-вывода</i> . Если символ типа сервера <i>ввода-вывода</i> указан, имя файла может быть пропущено.
ComputerName	varchar(255)	Имя компьютера, на котором запущен сервер <i>ввода-вывода</i> . Если компьютер не уточнен, сервер <i>ввода-вывода</i> запускается на узле хранения.
AutoStart	bit	Управляет запуском сервера <i>ввода-вывода</i> . 0 = Автоматический запуск системы программой управления. 1 = Требуется установка вручную.
ExeType	int	Тип исполнимой программы для сервера <i>ввода-вывода</i> . Используется в InSQL Control для определения способа запуска сервера <i>ввода-вывода</i> . 0 = Служба; 1 = Приложение-консоль; 2 = Приложение Windows.

(продолжение)

Атрибут	Тип данных	Объяснение
InitializationStatus	tinyint	Управляющий флаг, служащий для того, чтобы каждый сервер <i>ввода-вывода</i> опрашивался на получение типа данных (целый или действительный) для каждого тэга, который отсылается сервером. Нужен лишь в случае изменения базы данных.
ProtocolType	tinyint	Протокол, используемый узлом для общения с сервером ввода-вывода. 1 = DDE; 2 = SuiteLink; 3 = Системный (драйвер совместимости).
Description	varchar(50)	Описание сервера ввода-вывода.

Таблица IOServerType

Содержит одну строку для каждого известного типа сервера ввода-вывода. Информация о новом сервере ввода-вывода добавляется в эту таблицу после его установки. Эта таблица заполняется последней информацией о серверах Wonderware во время загрузки. Таблица IOServerType является справочной таблицей для всех типов серверов ввода-вывода.

Атрибут	Тип данных	Объяснение
ApplicationName	varchar(32)	Имя приложения для сервера ввода-вывода. Это имя обычно такое же, как и имя исполнимого файла.
Description	varchar(50)	Описание типа сервера ввода-вывода.
ExeName	varchar(255)	Имя исполнимого файла сервера ввода-вывода.
Revision	cher(20)	Номер редакции сервера ввода-вывода.
Platform	Платформы: cher(20)	Операционная система, необходимая для работы с сервером ввода-вывода. Допустимые ОС: WINDOWS NT, WINDOWS 95, WINDOWS 98.

Таблица NamedSystemParameter

Содержит одну строку для каждого системного параметра, имеющего имя. Именной системный параметр управляет определенными аспектами поведения системы.

Инструкции по созданию каталога для альтернативной области хранения даны в *Руководстве администратора системы IndustrialSQL Server*.

Атрибут	Тип данных	Объяснение
Name	varchar(50)	Уникальное имя системного параметра.
Type	char(7)	Тип именованного системного параметра. Допустимые типы: NUMERIC (ЧИСЛОВОЙ), STRING (ТЕКСТОВЫЙ).
StringValue	varchar(255)	Значение именованного системного параметра, если он текстового типа.
NumericValue	действительный	Значение именованного системного параметра, если он числового типа.
Editable	bit	Определяет, может ли именной системный параметр редактироваться в программе InSQL Configure. 1 = Редакция возможна; 0 = Редакция невозможна.
Description	varchar(255)	Описание системного параметра.

Таблица QualityMap

Содержит одну строку для каждого изменения деталей качества для значения тэга.

Атрибут	Тип данных	Объяснение
IOServerKey	int	Уникальный числовой идентификатор сервера ввода-вывода. Берется из таблицы <i>IOServer</i> .
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения. Берется из таблицы <i>IOServer</i> .
QualityDetail	int	Значение строки качества в том виде, в каком оно получено из базы данных. См. ПриложениеБ, "Качество данных" .
QualityString	varchar(100)	Текстовая строка, описывающая смысл значения детали качества.

Таблица StorageLocation

Содержит одну строку для каждого определенного узла хранения. Таблица StorageLocation содержит информацию о типах хранения данных и путях сохранения данных.

Атрибут	Тип данных	Объяснение
StorageType	int	Тип сохранения используемый для определенного места сохранения. 1 = Циклическое; 2 = Альтернативное; 3 = Буферное; 4 = Постоянное. При циклическом типе может быть только одно место сохранения.
SortOrder	int	Используется только в альтернативной области хранения. Если выбрано более одного места хранения, порядок сортировки определяет порядок использования альтернативных областей хранения. Зарезервировано на будущее.
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения. Берется из таблицы <i>StorageNode</i> .
Path	varchar(255)	Полный путь UNC для места хранения. Если пуст, то используется стандарт: <SystemDataPath>\InSQL\Data\Circular
MaxMBSize	int	Ограничение для количества данных, сохраняемых в определенном месте, измеряется в мегабайтах. Если MaxMBSize равняется 0, то используется все доступное дисковое пространство на месте хранения.
MinMBThreshold	int	Порог, по достижении которого система начнет предпринимать попытки освобождения пространства. Применяется только при циклическом и альтернативном методе хранения. Следует заметить, что высшее значение и динамически подсчитанное значение используются всегда.

Таблица StorageNode

Содержит одну строку для каждого пути к узлу хранения. Таблица StorageNode содержит информацию обо всех узлах хранения в системе. Система IndustrialSQL Server может иметь более одного узла хранения, в следствии чего IndustrialSQL Server должен знать подробные сведения о различных компьютерах, входящих в систему.

Примечание. В этой версии поддерживается только один узел хранения.

В общем случае в этой таблице содержится одна строка для каждого узла хранения в системе. В конфигурации для одиночного компьютера имя компьютера необязательно. В этой таблице отсутствует строка для компьютеров, работающих только как серверы.

Каждому узлу хранения присваивается уникальный номер, который хранится в столбце *StorageNodeKey* таблицы. Одна из процедур проверки целостности базы данных, которую выполняет IndustrialSQL Server при запуске, заключается в проверке того, чтобы низшее значение *StorageNodeKey* было равно 1 и чтобы оно возрастало последовательно.

При инициализации IndustrialSQL Server загружается информация об узлах, подтверждающая число узлов в системе. Сервер проверяет наличие вероятных значений атрибута *StorageNodeKey* и потом сравнивает значение символа со значением иностранного символа в таблице *Tag*. Если один из этих тестов не проходит, IndustrialSQL Server не запустится и выдаст сообщение об ошибке.

Для разрешения интеграционных несоответствий между таблицей *StorageNode* и таблицей *Tag*, в таблице *StorageNode* должна существовать, по крайней мере, одна строка, а в таблице *Tag* должны присутствовать ссылки на столбец *StorageNodeKey* в качестве строк этой таблицы.

Атрибут	Тип данных	Объяснение
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения. Данное значение получается системой автоматически. Столбец StorageNodeKey относится к типу IDENTITY.
ComputerName	varchar(255)	Сетевое имя компьютера Microsoft, где располагается узел хранения.
Description	varchar(50)	Описание, показывающее роль компьютера, сохраняющего данные.

(продолжение)

Атрибут	Тип данных	Объяснение
DbStatus	int	Управляет повторной инициализацией серверной службы со стороны клиента, такого как InSQL Configure. Принимает значение 2 включателями обновления, вставки и удаления в таблицах <i>Tag</i>, <i>AnalogTag</i>, <i>DiscreteTag</i>, <i>StringTag</i>, <i>Topic</i> и <i>IOServer</i>, в зависимости от того, какой столбец был изменен. Это указывает на необходимость реинициализации и позволяет программе InSQL Configure предложить пользователю сделать это. Серверной службой не используется. Принимает значение 3 от InSQL Configure для команды серверной службе начать реинициализацию. Принимает значение 4 от серверной службы во время выполнения фазы обновления базы данных. Это показывает, что идет инициализация серверной службы. Сбрасывается сервером на 0 при окончании реинициализации.
DbModAcquisition	int	Отрицательное значение говорит об ошибке при реинициализации. Вместе с <i>DbStatus</i> указывает серверной службе о необходимости перезапуска системы сбора данных. 0 = Перезапуск не требуется; 1 = Требуется перезапуск.
DbError	varchar(255)	Описание ошибки при создании базы данных.
DbModStorage	int	Вместе с <i>DbStatus</i> указывает серверной службе о необходимости перезапуска системы сохранения. 0 = Перезапуск не требуется; 1 = Требуется перезапуск.
DbModServer	int	Вместе с <i>DbStatus</i> указывает серверной службе о необходимости перезапуска сервера базы данных. 0 = Перезапуск не требуется; 1 = Требуется перезапуск.
DbModAll	int	Вместе с <i>DbStatus</i> указывает серверной службе о необходимости перезапуска всех систем базы данных. 0 = Перезапуск не требуется; 1 = Требуется перезапуск. Превалирует над значениями всех других подсистем.

(продолжение)

Атрибут	Тип данных	Объяснение
DbRevision	int	Текущий номер редакции базы данных. Значение автоматически обновляется, если <i>DbStatus</i> = 2. <i>DbRevision</i> используется поставщиками данных InSQL OLE DB для выявления необходимости повторной инициализации.

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица Topic

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица UserDetail

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица WideTableDictionary

Содержит одну строку значений до 249 тэгов. Эти 249 тэгов выглядят как столбец, видимый в словаре данных каждому пользователю. В основном используется инструментами специализированных запросов. Не влияет на возможность доступа к сохраняемым для тэга данным.

Атрибут	Тип данных	Объяснение
UserKey	int	Уникальный числовой идентификатор пользователя базы данных. Берется из таблицы UserDetail.
TagName	varchar(33)	Уникальное имя тэга или переменной в системе. Берется из таблицы Tag.

Г Л А В А 10

Таблицы определения тэгов

Эта глава является алфавитным справочником таблиц определения тэгов в базе данных IndustrialSQL Server. Таблицы определения тэгов — это таблицы, определяющие и хранящие информацию о тэгах. В этой главе описаны все атрибуты таблиц определения тэгов и типы их данных.

Содержание

- Таблицы определения тэгов
- Таблица AnalogTag
- Таблица ComplexTag
- Таблица Context
- Таблица Deviation
- Таблица DiscreteTag
- Таблица EngineeringUnit
- Таблица EventTag
- Таблица Limit
- Таблица LimitName
- Таблица Message
- Таблица RateOfChange
- Таблица StringTag
- Таблица Tag
- Таблица TagRef
- Таблица Topic

Таблицы определения тэгов

В таблицах определения тэгов хранится информация об аналоговых, дискретных, текстовых и сложных тэгах. Эта информация включает в себя максимальные и минимальные значения, границы, единицы измерения, типы и скорости сохранения, и т.п.

К таблицам определения тэгов относятся следующие:

- AnalogTag
- ComplexTag
- Context
- Deviation
- DiscreteTag
- EngineeringUnit
- EventTag
- Limit
- LimitName
- Message
- RateOfChange
- StringTag
- Tag
- TagRef
- Topic

Допустимые имена тэгов в IndustrialSQL Server

У имен тэгов, хранящихся в IndustrialSQL Server есть следующие ограничения:

- Количество символов в имени тэга может быть от 1 до 33.
- Может быть использован любой символ, кроме пробела; вместе с тем, для лучшей интеграции с Microsoft SQL Server настоятельно рекомендуется также соблюдать в именах тэгов, правила для идентификаторов Microsoft SQL Server.

См. также документацию по Microsoft SQL Server.

Если используется сервер базы данных IndustrialSQL или OPENQUERY, то символ "@" не поддерживается. Символы "\$", "#" и "_" поддерживаются сервером базы данных IndustrialSQL или OPENQUERY, если не являются первым символом в имени тэга, а стоят внутри него.

Если имя тэга не соответствует правилам идентификации Microsoft SQL Server, следует изменить его при помощи InSQL Configure для того, чтобы это имя тэга можно было использовать в запросах для таблиц *AnalogWideHistory*, *DiscreteWideHistory* и *WideTableDictionary*. Некоторые инструменты клиента FactoryOffice также не смогут получать данные, если в именах тэгов присутствуют недопустимые символы.

Введен специальный элемент управления для поддержки "\$" в качестве первого символа (например, для системных тэгов InTouch) и для имен тэгов, начинающихся с одной цифры. Вместе с тем, имена тэгов, начинающиеся с двух цифр не поддерживаются.

Примечание. В системе наименования тэгов InTouch буква "А" эквивалентна "а". Вместе с тем, регистро-чувствительная база данных *Runtime* IndustrialSQL Server, различает регистры.

Схема базы данных для таблиц определения тэгов выглядит следующим образом:

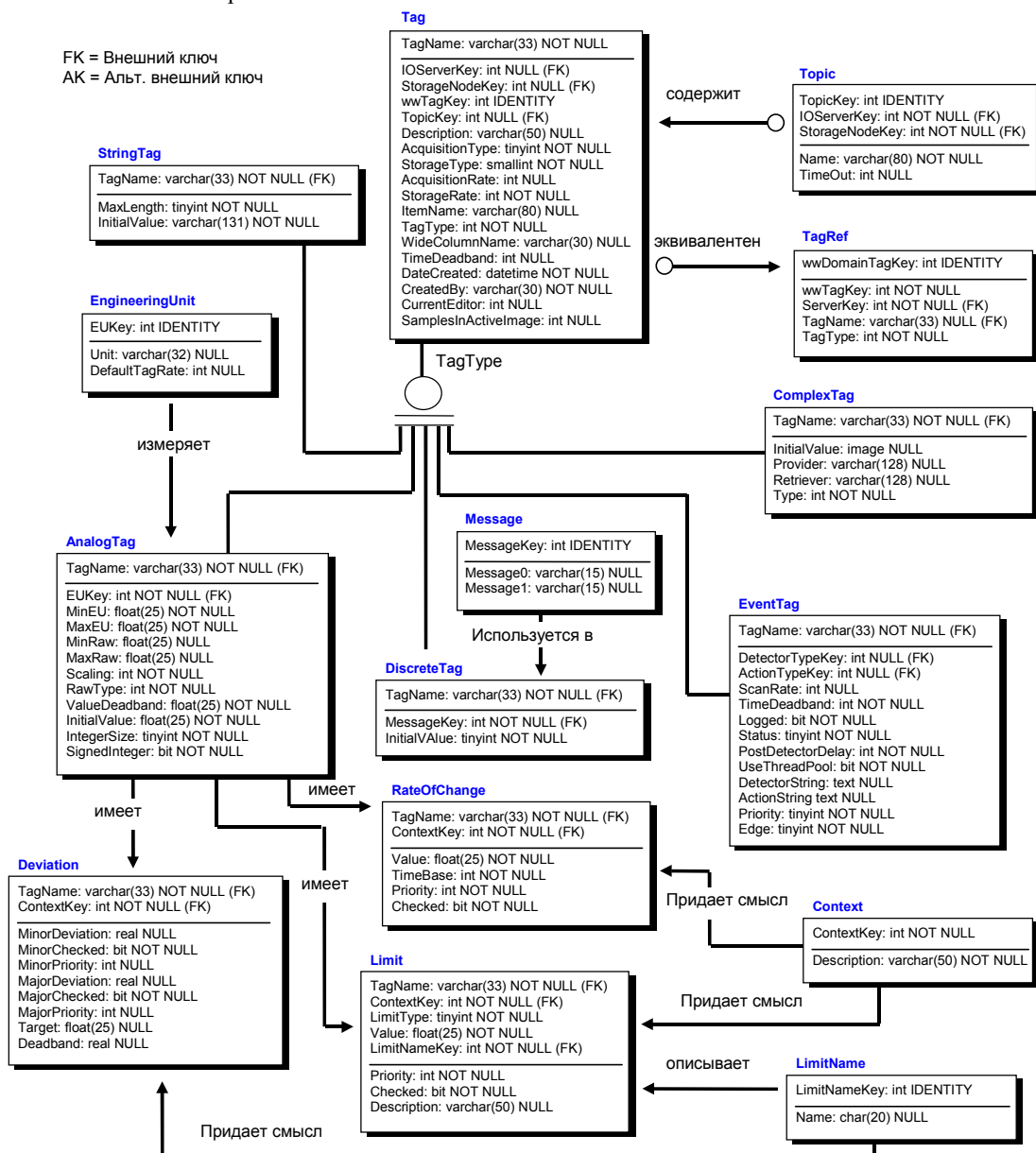


Таблица AnalogTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица ComplexTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица Context

Содержит одну строку для каждого контекста, к которому может принадлежать группа границ, скоростей изменения или отклонений. Примерами могут быть "Нормальная работа", "Холодное выключение", "Особое внимание".

Атрибут	Тип данных	Объяснение
ContextKey	int	Уникальный числовой идентификатор контекста, в котором используется группа ограничений, скорости изменения или отклонений. Данное значение получается системой автоматически. Столбец ContextKey относится к типу IDENTITY.
Description	varchar(50)	Описание контекста, в котором используется группа ограничений, скорости изменения или отклонений.

Таблица Deviation

Содержит одну строку для каждого определенного отклонения аналогового тэга. Отклонением называется процент изменения значения тэга по отношению к зафиксированному значению, называемому целевым. У каждого аналогового тэга могут быть два возможных отклонения: значительное и незначительное.

Атрибут	Тип данных	Объяснение
TagName	varchar(33)	Имя тэга, в котором присутствует отклонение.
ContextKey	int	Уникальный числовой идентификатор контекста группы ограничений, скорости изменения или отклонений. Иностраный символ из таблицы <i>Context</i> .
MinorDeviation	real	Процент возможного изменения тэга от целевого значения до того, как будут выполнены условия аларма незначительного отклонения.
MinorChecked	bit	Определяет состояние аларма тэга, основанного на незначительном отклонении. 0 = Не в состоянии аларма; 1 = В состоянии аларма.
MinorPriority	int	Уровень приоритета для незначительного отклонения. Допустимы значения от 1 до 999, где 1 - высший уровень приоритета, а 999 - низший.
MajorDeviation	real	Процент возможного изменения тэга от целевого значения до того, как исполнятся условия аларма значительного отклонения.
MajorChecked	bit	Определяет состояние аларма тэга, основанное на значительном отклонении. 0 = Не в состоянии аларма; 1 = В состоянии аларма.
MajorPriority	int	Уровень приоритета для значительного отклонения. Допустимы значения от 1 до 999, где 1 - высший уровень приоритета, а 999 - низший.
Target	float(25)	Справочное значение тэга, на котором основан процент высшего и/или низшего отклонения.
Deadband	real	Процент отклонения, на который значение тэга должно упасть ниже целевого значения, прежде чем тэг будет выведен из состояния аларма.

Таблица DiscreteTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица EngineeringUnit

Содержит одну строку для каждой определенной единицы измерения (инженерной единицы). Каждый аналоговый тэг имеет соответствующую единицу измерения.

Атрибут	Тип данных	Объяснение
EUKey	int	Уникальный числовой идентификатор единицы измерения. Данное значение получается системой автоматически. Столбец EUKey относится к типу IDENTITY.
Unit	varchar(32)	Единица измерения. Примерами служат м/с, граммы, фунты.
DefaultTagRate	int	Стандартная скорость, в миллисекундах, с которой тэги циклически сохраняются, основанная на единице измерения. Например, этот столбец может использоваться системным администратором для установки стандартной скорости циклического сохранения.

Таблица EventTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица Limit

Содержит одну строку для каждого отслеживаемого ограничения. Для любого тэга может быть установлено любое число границ.

Атрибут	Тип данных	Объяснение
TagName	varchar(33)	Имя тэга, к которому относится ограничение.
ContextKey	int	Уникальный числовой идентификатор (из таблицы <i>Context</i>) для контекста, в котором используется группа ограничений, скорости изменений или отклонений.
LimitType	tinyint	Тип ограничения. 0 = Возрастание; 1 = Понижение.
Value	float(25)	Значение, используемое как определенное ограничение тэга. Теоретически, тэг может иметь неопределенное число ограничений.
LimitNameKey	int	Уникальный числовой идентификатор (из таблицы <i>LimitName</i>), связанный с именем ограничения.
Priority	int	Приоритет ограничения. Приоритет может варьироваться от 1 (единица) до 2 миллиардов, где 1 - высший приоритет.
Checked	bit	Уточняет, был ли импортированный из InTouch тэг настроен для автоматической проверки ограничений. 1 = Проверка включена; 0 = Проверка выключена.
Description	varchar(50)	Описание ограничения.

Таблица LimitName

Содержит одну строку для каждого наименования, связанного с конкретным ограничением. Примерами могут быть "Верхнее", "Нижнее" и "Обслуживание".

Атрибут	Тип данных	Объяснение
LimitNameKey	int	Уникальный идентификатор имени ограничения. Данное значение получается системой автоматически. Столбец LimitNameKey относится к типу IDENTITY.
Name	char(20)	Имя ограничения.

Таблица Message

Содержит одну строку для каждой пары сообщений типа on/off. Пары сообщений присутствуют в таблице лишь один раз. Например, пара сообщений может быть "Открыто" и "Закрыто" и может быть связана с положением клапана и или переключателя.

Атрибут	Тип данных	Объяснение
MessageKey	int	Уникальный числовой идентификатор сообщения. Данное значение получается системой автоматически. Стобец MessageKey относится к типу IDENTITY.
Message0	varchar(15)	Сообщение, связанное с состоянием ЛОЖЬ дискретного тэга. Дискретный тэг получает значение 0 в случае, если он находится в состоянии ЛОЖЬ.
Message1	varchar(15)	Сообщение, связанное с состоянием ПРАВДА дискретного тэга. Дискретный тэг получает значение 1 в случае, если он находится в состоянии ПРАВДА.

Таблица RateOfChange

Содержит одну строку для каждой отслеживаемой скорости изменения тэга.

Атрибут	Тип данных	Объяснение
TagName	varchar(33)	Имя тэга, связанного с данной скоростью изменения.
ContextKey	int	Уникальный числовой идентификатор (из таблицы <i>Context</i>) контекста, в котором используется группа ограничений, скорости изменений или отклонений.
Value	float(25)	Процент изменения тэга за период времени, определенный временной базой.
TimeBase	int	Единица времени, которой измеряется скорость изменения.
Priority	int	Приоритет ограничения. Приоритет может варьироваться от 1 (единица) до 2 миллиардов, где 1 - высший приоритет.
Checked	bit	Уточняет, был ли импортированный из InTouch тэг настроен для автоматической проверки ограничений. 1 = Проверка включена; 0 = Проверка выключена.

Таблица StringTag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица Tag

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица TagRef

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Таблица Topic

Подробнее об этой таблице см. в главе 2 Справочного руководства, "Тематические области и общие таблицы".

Удаленные таблицы OLE DB

В этой главе описаны удаленные таблицы OLE DB, дающие доступ к архивным данным и данным реального времени IndustrialSQL Server. В этой главе описаны все атрибуты хранения таблиц и типы их данных.

Примеры получения данных из этих таблиц см. в главе 14, "Выборка данных".

Содержание

- Удаленные таблицы OLE DB
- Таблица AnalogHistory (OLE DB)
- Таблица AnalogLive (OLE DB)
- Таблица AnalogWideHistory (OLE DB)
- Таблица DiscreteHistory (OLE DB)
- Таблица DiscreteLive (OLE DB)
- Таблица DiscreteWideHistory (OLE DB)
- Таблица HistoryBlock (OLE DB)
- Таблица StringLive (OLE DB)

Удаленные таблицы OLE DB

Доступ к удаленной таблице OLE DB такой же, как и к набору необработанных данных. Поставщик данных OLE DB для IndustrialSQL Server содержит один каталог (Runtime) и одну схему (dbo) и имеет доступ к следующим удаленным таблицам:

- *AnalogHistory*
- *AnalogLive*
- *AnalogWideHistory*
- *DiscreteHistory*
- *DiscreteLive*
- *DiscreteWideHistory*
- *HistoryBlock*
- *StringLive*

Хотя имена этих таблиц совпадают с именами таблиц расширения в схеме базы данных *Runtime*, удаленные таблицы — не то же самое, что таблицы расширения. Удаленные таблицы не создаются в базе данных *Runtime* Microsoft SQL Server. Вместо этого они создаются внутри поставщика OLE DB IndustrialSQL.

Для следующих таблиц имеются представления. Имена этих представлений могут использоваться в запросах данных из таблиц OLE DB.

- *v_AnalogHistory*
- *v_AnalogLive*
- *v_DiscreteHistory*
- *v_DiscreteLive*
- *v_HistoryBlock*
- *v_StringLive*

См. также главу 13, "Представления данных".

Таблица AnalogHistory (OLE DB)

Содержит одну строку для каждого сохраненного аналогового тэга за взятый период (циклическое сохранение) или изменения значения (дельта-сохранение). Таблица AnalogHistory содержит каждое аналоговое значение, сохраненное в IndustrialSQL Server. Таблица AnalogHistory — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком OLE DB IndustrialSQL.

Если не использовать представление для этой таблицы, то в запросе следует полностью указать ее имя (INSQL.Runtime.dbo.AnalogHistory).

Это логическая таблица, выдающая запрошенное количество строк за промежуток времени между начальной и конечной датами запроса, независимо от частоты получения данных. В большинстве приложений это избавляет пользователя от необходимости понимания скоростей сканирования и других особенностей механизма сбора данных. В любом запросе к таблице AnalogHistory следует указать имя тэга.

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени получения аналоговых данных.
TagName	varchar(33)	Имя аналогового тэга, для которого сохраняется архив.
Value	float(15)	Значение аналогового тэга в момент даты/времени. Указывается в единицах измерения.
Quality	tinyint	Основной показатель качества, связанный со значением аналогового архивного тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении индикатора качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> — иностранный символ из таблицы <i>Tag</i> .
WwRowCount	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwResolution	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwEdgeDetection	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.

(продолжение)

Атрибут	Тип данных	Объяснение
wwValueDeadband	real	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

Таблица AnalogLive (OLE DB)

Содержит одну строку для каждого аналогового тэга. Значение каждого тэга этой таблицы обновляется каждый раз при получении нового значения от сервера ввода-вывода. Эта таблица выдает последние данные. Таблица AnalogLive — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком данных OLE DB IndustrialSQL.

Если не использовать представление для этой таблицы, то в запросе следует полностью указать ее имя (INSQL.Runtime.dbo.AnalogLive.).

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени, когда были получены обновленные аналоговые данные.
TagName	varchar(33)	Имя аналогового тэга,
Value	float(15)	Значение аналогового тэга в момент даты/времени. Указывается в единицах измерения.
Quality	tinyint	Основной показатель качества , связанный со значением аналогового "живого" тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении показателя качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> — иностранный символ из таблицы <i>Tag</i> .
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwValueDeadband	real	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

Таблица AnalogWideHistory (OLE DB)

Содержит строки со значениями множественных аналоговых тэгов. Эта удаленная таблица нарушает стандартные правила обработки данных, но очень полезна для некоторых запросов. Используется для получения значения множественных аналоговых тэгов в заданный момент времени.

Таблица поддерживается, только если используется OPENQUERY.

Подробнее см. в главе 14, "Выборка данных".

Таблица AnalogWideHistory содержит те же данные, что и таблица AnalogHistory, но в другом формате. В широком формате существуют столбцы для метки даты/времени и столбцы для значений каждого выбранного тэга на этот момент времени. Имена тэгов запроса используются как заголовки столбцов. В таблице существует один столбец для метки даты/времени и столбцы для каждого имени тэга, указанного в запросе к базе данных.

Таблица AnalogWideHistory — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком данных IndustrialSQL OLE DB.

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени получения данных в строке. В этой удаленной таблице точность этой метки обычно определяется разрешением запроса.
ManyOtherTags	float(15)	Столбец-заглушка для одного и более аналоговых тэгов в широком табличном формате. В этом формате имена тэгов используются как заголовки столбцов. Столбец ManyOtherTags дублируется столько раз, сколько тэгов указано в запросе базы данных.
WwRowCount	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwResolution	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwEdgeDetection	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwValueDeadband	real	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

Примечание. Заголовки столбцов идентичны именам тэгов, а содержание столбцов — значениям тэгов.

Из-за большого количества переменных столбец *Quality* не виден в этом представлении. Вместо этого, если полученная переменная недоступна, заполнена или неверна, то значение становится NULL. Это представление данных очень удобно для разработки и не может существовать в качестве таблицы Microsoft SQL Server из-за слишком большого количества столбцов.

При формировании запроса к таблице *AnalogWideHistory* действуют следующие ограничения:

- Должно быть указано имя столбца.
- Доступ только через OPENQUERY

Так как в качестве заголовков столбцов выступают имена тэгов, действуют также правила для идентификаторов Microsoft SQL Server. Добавлен специальный элемент управления для поддержки символа "\$" в качестве первого символа (например, для системных тэгов InTouch и для имен тэгов, начинающихся с одиночной цифры).

При переносе столбцов из этой таблицы в таблицу Microsoft SQL Server имена столбцов могут не распознаться системой Microsoft SQL Server. IndustrialSQL Server автоматически преобразует заголовки столбцов в правильный формат используя следующие критерии:

1. "-" заменяется на "_".
2. В начало имен тэгов, начинающихся с одиночной цифры, добавляется знак "_".

Отдельно от этих двух случаев, любое имя тэга, содержащее недопустимые символы (идентификатор), данных не выдаст.

Таблица DiscreteHistory (OLE DB)

Содержит одну строку для каждого сохраненного за некоторый период дискретного тэга. Таблица DiscreteHistory — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком данных IndustrialSQL OLE DB. Содержит каждое дискретное событие, записанное IndustrialSQL Server. В любом запросе к таблице DiscreteHistory следует указать имя тэга.

Если не используется представление этой таблицы, то в запросе следует указать полное имя (INSQLD.Runtime.dbo.DiscreteHistory).

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени получения дискретных архивных данных.
TagName	varchar(33)	Имя дискретного тэга, для которого сохраняется архив.
Value	tinyint	Значение дискретного тэга в момент даты/времени. 0 = ЛОЖЬ; 1 = ИСТИНА; NULL = Данных нет.
Quality	tinyint	Основной показатель качества, связанный со значением дискретного архивного тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении показателя качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> — иностранный символ из таблицы <i>Tag</i> .
WwRowCount	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwResolution	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwEdgeDetection	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

При каждом изменении состояния тэга это состояние сохраняется с идентификатором тэга, состоянием тэга и показателем качества.

Примечание. Если происходит сбой в IndustrialSQL Server или источнике данных и IndustrialSQL Server или этот источник перезапущены, то записывается пара событий как изменение от правильного тэга к неправильному и обратно.

Когда в таблицу подается запрос, сервер выдает одну строку для каждого события, произошедшего за время подачи запроса, не взирая на промежуток времени между событиями. Это называется методом "дельта-получения". Максимальное количество строк, выдаваемых при дельта-получении, определяется минимальным значением `wwRowcount` и установкой `ROWCOUNT SQL Server`.

Таблица DiscreteLive (OLE DB)

Содержит одну строку для каждого дискретного тэга. Значение каждого тэга этой таблицы обновляется каждый раз при получении нового значения от сервера ввода-вывода. Эта таблица выдает последние данные. Таблица `DiscreteLive` — это удаленная таблица, хранимая вне Microsoft SQL Server и управляемая поставщиком данных IndustrialSQL OLE DB.

Если не используется представление этой таблицы, то в запросе следует указать полное имя (`INSQL.Runtime.dbo.DiscreteLive`).

Атрибут	Тип данных	Объяснение
<code>DateTime</code>	<code>datetime</code>	Метка времени получения обновления.
<code>TagName</code>	<code>varchar(33)</code>	Имя дискретного тэга.
<code>Value</code>	<code>tinyint</code>	Значение дискретного тэга в момент даты/времени. 0 = ЛОЖЬ; 1 = ИСТИНА; NULL = Данных нет.
<code>Quality</code>	<code>tinyint</code>	Основной показатель качества, связанный с дискретным тэгом. Подробнее в Приложении Б, "Качество данных" .
<code>QualityDetail</code>	<code>int</code>	Информация о значении показателя качества. Подробнее в Приложении Б, "Качество данных" .
<code>WwTagKey</code>	<code>int</code>	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> — иностранный символ из таблицы <i>Tag</i> .
<code>WwRetrievalMode</code>	<code>varchar(10)</code>	См. "Столбцы для запросов с параметрами" далее в этой главе.
<code>WwTimeDeadband</code>	<code>int</code>	См. "Столбцы для запросов с параметрами" далее в этой главе.
<code>WwParameters</code>	<code>varchar(128)</code>	См. "Столбцы для запросов с параметрами" далее в этой главе.

Таблица DiscreteWideHistory (OLE DB)

Содержит одну строку значений для множества дискретных тэгов. Эта удаленная таблица нарушает стандартные правила обработки данных, но очень полезна для некоторых запросов. Используется для получения значения множественных дискретных тэгов в заданный момент времени.

Таблица поддерживается, только если используется OPENQUERY.

Подробнее см. в главе 14, "Получение данных".

Таблица DiscreteWideHistory содержит те же данные, что и таблица DiscreteHistory, но в ином формате. В широком формате существуют столбцы для метки даты/времени и столбцы для значений каждого выбранного тэга на этот момент времени. Имена тэгов запроса используются как заголовки столбцов. В этой таблице присутствует один столбец для метки даты и времени и столбцы для каждого имени тэга, указанного в запросе.

Таблица DiscreteWideHistory — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком данных IndustrialSQL OLE DB.

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени получения данных в строке. Для дискретных значений это время изменения элемента данных. Если за промежуток времени между появлением строк изменилось более одного элемента данных, то это время самого раннего из них.
SysPulse	tinyint	Стандартный тэг, который появляется в таблице. Данный тэг присутствует для демонстрации функций "широкого" табличного формата и может быть заменен любым дискретным тэгом.
ManyOtherTags	tinyint	Столбец "заглушки" для одного или более дискретных тэгов в широком табличном формате. В этом формате имена тэгов используются как заголовки столбцов. Столбец ManyOtherTags дублируется столько раз, сколько тэгов указано в запросе к базе данных.
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwEdgeDetection	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

Примечание. Заголовки столбцов идентичны именам тэгов, а содержание столбцов — значениям тэгов.

Из-за большого количества переменных столбец *Quality* не виден в этом представлении. Вместо этого, если полученная переменная недоступна, заполнена или неверна, то значение становится NULL. Это представление данных очень удобно для разработки и не может существовать в качестве таблицы Microsoft SQL Server из-за слишком большого количества столбцов.

При формировании запроса к таблице *AnalogWideHistory* действуют следующие ограничения:

- Должно быть указано имя столбца.
- Доступ только через OPENQUERY

Так как в качестве заголовков столбцов выступают имена тэгов, действуют также правила для идентификаторов Microsoft SQL Server. Добавлен специальный элемент управления для поддержки символа "\$" в качестве первого символа (например, для системных тэгов InTouch и для имен тэгов, начинающихся с одиночной цифры).

При переносе столбцов из этой таблицы в таблицу Microsoft SQL Server имена столбцов могут не распознаться системой Microsoft SQL Server. IndustrialSQL Server автоматически преобразует заголовки столбцов в правильный формат используя следующие критерии:

1. "-" заменяется на "_".
2. В начало имен тэгов, начинающихся с одиночной цифры, добавляется знак "_".

Отдельно от этих двух случаев, любое имя тэга, содержащее недопустимые символы (идентификатор), данных не выдаст.

Таблица HistoryBlock (OLE DB)

Содержит одну строку для каждого блока архивных данных, хранимых в узле хранения.

Если не используется представление этой таблицы, то в запросе следует указать полное имя (INSQL.Runtime.dbo.DiscreteLive).

В IndustrialSQL Server архивные данные делятся на отдельные части, называемые блоками. Деление архивных данных на блоки облегчает их резервирование и восстановление. Новый блок создается (или требуется) в следующих случаях:

- Обновляется база данных.
- Перезапущен сервер хранения IndustrialSQL Server.
- По истечении определенного времени (по умолчанию, 24 часа) длительного хранения данных. За этот специальный запланированный период появляется блок данных, удобный для резервирования и восстановления.
- Для самостоятельного создания блока следует воспользоваться программой InSQL Control или расширенной процедурой сохранения **xp_NewHistoryBlock**.

Таблице HistoryBlock можно подать запрос на поиск известных системе блоков данных. Строка таблицы HistoryBlock относится к физическому подкаталогу файлов, содержащих архивные данные.

Таблица HistoryBlock — это удаленная таблица, хранящаяся вне Microsoft SQL Server и управляемая поставщиком данных IndustrialSQL OLE DB.

Атрибут	Тип данных	Объяснение
FromDate	datetime	Стартовые дата и время создания архивного блока.
ToDate	datetime	Время окончания создания архивного блока.
StorageNodeKey	int	Уникальный числовой идентификатор узла хранения.
Description	varchar(50)	Описание архивного блока.
OnLine	bit	Проверяет доступность данных на диске. 1 = Активно; 2 = Включено; 3 = Выключено (для использования в будущем).
HistoryArchived	int	Показывает, был ли сжат (зарезервирован) архивный блок. 1 = Нет; 2 = Сжат; 3 = Распакован; 4 = Удален. Зарезервировано на будущее.
SummaryArchived	int	Показывает, была ли сжата (зарезервирована) сводка тэгов. 1 = Нет; 2 = Сжата; 3 = Распакована; 4 = Удалена. Зарезервировано на будущее.

(продолжение)

Атрибута	Тип данных	Объяснение
EventArchived	int	Показывает, было ли сжато (зарезервировано) событие. 1 = Нет; 2 = Сжат; 3 = Распакован; 4 = Удален. Зарезервировано на будущее.
StorageAreaType	int	Метод, используемый для хранения данных. 1 = Циклический; 2 = Альтернативный; 3 = Буферный; 4 = Постоянный. Зарезервировано на будущее.
ArchiveDate	datetime	Дата, когда был сжат архивный блок. Зарезервировано на будущее.
ArchiveLocation	varchar(50)	Место, где располагается сжатый архивный блок. Зарезервировано на будущее.
Version	int	Номер версии архивного блока. 1 = Формат блока ниже версии 3.0; 2 = Формат блока версии 3.0 и выше. Зарезервировано на будущее.
Compression	int	Номер версии циклического сжатия. 1 = Сжатия нет; 2 = Кодирование Хаффмана. Зарезервировано на будущее.
Sequence	int	Последовательность потока данных. (1...n) Зарезервировано на будущее.

Таблица StringLive (OLE DB)

Содержит одну строку для каждого текстового тэга. Значение каждого тэга этой таблицы обновляется каждый раз при получении нового значения от сервера ввода-вывода. Эта таблица выдает последние данные.

Если не используется представление этой таблицы, то в запросе следует указать полное имя (INSQL.Runtime.dbo.StringLive).

Атрибут	Тип данных	Объяснение
DateTime	datetime	Метка времени последнего изменения текстовых данных.
TagName	varchar(33)	Имя текстового тэга
Value	varchar(131)	Значение текстового тэга (как текст) в момент даты/времени.
Quality	tinyint	Основной показатель качества, связанный со значением текстового "живого" тэга. Подробнее в Приложении Б, "Качество данных".
QualityDetail	int	Информация о значении показателя качества. Подробнее в Приложении Б, "Качество данных".
WwTagKey	int	Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>WwTagKey</i> — иностранный символ из таблицы <i>Tag</i> .
WwRetrievalMode	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwEdgeDetection	varchar(10)	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwTimeDeadband	int	См. "Столбцы для запросов с параметрами" далее в этой главе.
WwParameters	varchar(128)	См. "Столбцы для запросов с параметрами" далее в этой главе.

Столбцы для запросов с параметрами

В удаленных таблицах OLE DB находятся следующие столбцы хранения данных:

Атрибут	Тип	Объяснение
wwRowCount	int	Количество строк, которые будут выданы за указанный период времени, равно распределенные. По умолчанию это 100 строк. Подсчитывается для каждого тэга в запросе.
WwResolution	int	Временной интервал (в мс) между выдачей строк за указанный период времени.
wwRetrievalMode	varchar(10)	Указывает, запрашивать ли аналоговые данные циклично или методом исключения (дельта). Правильными значениями являются CYCLIC и DELTA. CYCLIC = Выдаются все сохраненные данные для тэгов за указанный промежуток времени. DELTA = Выдаются только изменившиеся за указанный промежуток времени данные. По умолчанию параметр CYCLIC стоит для получения аналоговых данных, а DELTA — для дискретных данных.
wwEdgeDetection	varchar(10)	Тип результата определения края, который будет выдан в ответ на запрос. Правильными значениями являются NONE, LEADING, TRAILING и BOTH. По умолчанию: NONE. См. "Определение края" далее.
WwTimeDeadband	int	Минимальное время (в мс) между выдачей результатов для одного тэга. Действует только при получении данных методом исключения (дельта-методом). По умолчанию действует значение 0.
WwValueDeadband	real	Процент полного масштаба (диапазона) в единицах измерения. Все изменения значений, которые меньше этого процента, выдаваться не будут. Действует только при получении данных методом исключения (дельта-методом). По умолчанию действует значение 0.
WwParameters	varchar(128)	Используется для дополнительных параметров, которые могут быть указаны в запросе. Правильными значениями являются DEBUG или пустая строка. По умолчанию это пустая строка.

Определение края

При использовании параметра *wwEdgeDetection* выдается набор значений, связанных с событием, происходящим в системе. Набор результатов, выданный при использовании параметра *wwEdgeDetection*, основан на том, соответствуют ли полученные значения критерию WHERE. По умолчанию этот параметр установлен так, что определения края не происходит.

Правильными значениями являются *None*, *Leading*, *Trailing* и *Both*. Если указано значение *None*, то будут выданы все строки, соответствующие критериям WHERE; определения края не происходит. Если указан параметр *Leading*, то в набор результатов входят только те строки, которые первыми соответствовали критериям WHERE (выдана истина) после тех, которые этим критериям не соответствовали (выдана ложь). Если указан параметр *Trailing*, то в набор результатов входят только те, которые первыми не соответствовали критериям WHERE (выдана ложь) после тех, которые этим критериям соответствовали (выдана истина). Если указан параметр *Both*, то выдаются все строки.

Подробнее см. в главе 14, "Выборка данных".

Г Л А В А 12

Хранимые процедуры

Хранимая процедура — это скомпилированная группа выражений SQL. Используя хранимую процедуру, можно выполнить группу последовательно идущих в запросе заданий. В дополнение к хранимым процедурам, поставляемым вместе с SQL Server, существуют специальные хранимые процедуры для использования системой IndustrialSQL Server.

В этой главе детально описаны все хранимые процедуры, включая синтаксис сценариев, параметры и примеры. Также в этой главе приводится информация о расширенных хранимых процедурах IndustrialSQL Server.

Подробнее см. в документации Microsoft.

Содержание

- [Хранимые процедуры IndustrialSQL Server](#)
- [Системные хранимые процедуры](#)
- [Пользовательские хранимые процедуры](#)
- [Расширенные хранимые процедуры](#)
- [Создание хранимых процедур](#)

Хранимые процедуры IndustrialSQL Server

Хранимая процедура есть скомпилированная группа выражений SQL. Используя хранимую процедуру, можно выполнить группу последовательно идущих в запросе заданий. В дополнение к хранимым процедурам, поставляемым вместе с SQL Server, существуют два типа хранимых процедур для использования системой IndustrialSQL Server: системные хранимые процедуры и пользовательские хранимые процедуры.

Системные хранимые процедуры

Системные хранимые процедуры используются для настройки системы IndustrialSQL Server. Системные хранимые процедуры обычно выполняются во время запуска системы и изменения системных настроек. Эти хранимые процедуры в основном используются программой настройки IndustrialSQL Server, системой управления событиями, программой InSQL Configure и приложениями-клиентами.

ww_ActionStringSelect

Выбирает строку действия для указанного тэга события.

Синтаксис	<code>ww_ActionStringSelect</code>
Параметры	<i>ИмяТэга</i> Имя тэга события.
Допуск	Выполняет правила допуска для группы <i>wwAdministrators</i> .
Таблицы	<i>EventTag</i>
См. также	Хранимая процедура <code>ww_DetectorStringSelect</code>

ww_AnalogConfig

Выдает детали настройки для всех аналоговых тэгов в указанном узле хранения. Хранимая процедура создает адрес аналогового тэга, состоящий из имени аналогового тэга и имени элемента.

Синтаксис	<code>ww_AnalogConfig</code>
Параметры	<i>НомерУзла</i> Числовой идентификатор узла хранения.
Примечание	Числовой идентификатор конкретного узла хранения хранится в столбце <i>StorageNodeKey</i> таблицы <i>StorageNode</i> .
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i> , <i>Topic</i> , <i>AnalogTag</i> , <i>IOServer</i>
См. также	<code>ww_DiscreteConfig</code> <code>ww_StringConfig</code>

ww_AnalogCount

Выдает итоговый подсчет всех аналоговых тэгов для указанного узла хранения.

Синтаксис `ww_AnalogCount    .....    .....`

Параметры *номер узла*
Числовой идентификатор узла хранения.

Примечание Числовой идентификатор конкретного узла хранения хранится в столбце *StorageNodeKey* таблицы *StorageNode*.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *Tag, TagRe*

См. также `ww_DiscreteConfig`

ww_AnalogTagDelete

Удаляет аналоговый тэг.

Синтаксис `ww_AnalogTagDelete   wwTagKey`

Параметры *wwTagKey*
Уникальный идентификатора тэга.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *Tag*

См. также `ww_AnalogTagInsert`
`ww_AnalogTagSelect`
`ww_AnalogTagUpdate`

ww_AnalogTagInsert

Вставляет аналоговый тэг.

Синтаксис	<code>ww_AnalogTagInsert TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, EUKey, MinEU, MaxEU, MinRaw, MaxRaw, Scaling, RawType, ValueDeadband, InitialValue, IntegerSize, SignedInteger, TopicKey, IOServerKey, StorageNodeKey</code>
Параметры	<i>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, TopicKey, IOServerKey, StorageNodeKey</i> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> см. в главе 2, "Предметные области и общие таблицы". <i>EUKey, MinEU, MaxEU, MinRaw, MaxRaw, Scaling, RawType, ValueDeadband, InitialValue, IntegerSize, SignedInteger</i> Эти параметры идентичны столбцам в таблице <i>AnalogTag</i>. Подробнее о таблице <i>AnalogTag</i> см. в главе 2, "Предметные области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>AnalogTag, Tag</i>
См. также	<code>ww_AnalogTagSelect</code> <code>ww_AnalogTagDelete</code> <code>ww_AnalogTagUpdate</code>

ww_AnalogTagSelect

Выбирает аналоговый тэг.

Синтаксис	<code>ww_AnalogTagSelect wwTagKey</code>
Параметры	<i>wwTagKey</i> Уникальный идентификатора тэга.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>AnalogTag, Tag</i>
См. также	<code>ww_AnalogTagInsert</code> <code>ww_AnalogTagDelete</code> <code>ww_AnalogTagUpdate</code>

ww_AnalogTagUpdate

Обновляет аналоговый тэг.

Синтаксис	<code>ww_AnalogTagUpdate wwTagKey, TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, EUKey, MinEU, MaxEU, MinRaw, MaxRaw, Scaling, RawType, ValueDeadband, InitialValue, IntegerSize, SignedInteger, TopicKey, IOServerKey, StorageNodeKey</code>
Параметры	<code>wwTagKey</code> Уникальный идентификатор тэга. <code>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, TopicKey, IOServerKey, StorageNodeKey</code> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> см. в главе 2, "Тематические области и общие таблицы". <code>EUKey, MinEU, MaxEU, MinRaw, MaxRaw, Scaling, RawType, ValueDeadband, InitialValue, IntegerSize, SignedInteger</code> Эти параметры идентичны столбцам в таблице <i>AnalogTag</i>. Подробнее о таблице <i>AnalogTag</i> см. в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>AnalogTag, Tag</i>
См. также	<code>ww_AnalogTagSelect</code> <code>ww_AnalogTagInsert</code> <code>ww_AnalogTagDelete</code>

ww_AnnotationDelete

Удаляет аннотацию из таблицы *Annotation*.

Синтаксис	<code>ww_AnnotationDelete AnnotationKey</code>
Параметры	<code>AnnotationKey</code> Уникальный числовой идентификатор аннотации.
Допуск	Выполняет правила допуска для групп <i>wwUsers</i> , <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Annotation</i>
См. также	<code>ww_AnnotationSelect</code> <code>ww_AnnotationUpdate</code> <code>ww_AnnotationInsert</code>

ww_AnnotationInsert

Вставляет аннотацию в таблицу *Annotation*.

Синтаксис `ww_AnnotationInsert TagName, UserKey, DateTime, DateCreated, Content, Value`

Параметры *TagName, UserKey, DateTime, DateCreated, Content, Value*

Эти параметры идентичны столбцам в таблице *Annotation*.

Подробнее о таблице *Annotation* см. в главе 3, "Таблицы хранения данных".

Допуск Выполняет правила допуска для групп *wwUsers*, *wwPowerUsers* и *wwAdministrators*.

Таблицы *Annotation*

См. также `ww_AnnotationSelect`
 `ww_AnnotationUpdate`
 `ww_AnnotationDelete`

ww_AnnotationSelect

Выбирает аннотацию из таблицы *Annotation*.

Синтаксис `ww_AnnotationSelect AnnotationKey`

Параметры *AnnotationKey*

Уникальный числовой идентификатор аннотации.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *Annotation*

См. также `ww_AnnotationInsert`
 `ww_AnnotationUpdate`
 `ww_AnnotationDelete`

ww_AnnotationUpdate

Вставляет аннотацию в таблицу *Annotation*.

Синтаксис	<code>ww_AnnotationUpdate AnnotationKey, TagName, UserKey, DateTime, DateCreated, Content, Value</code>
Параметры	<i>AnnotationKey</i> Уникальный числовой идентификатор аннотации. <i>TagName, UserKey, DateTime, DateCreated, Content, Value</i> Эти параметры идентичны столбцам в таблице <i>Annotation</i> . Подробнее о таблице <i>Annotation</i> см. в главе 3, "Таблицы хранения данных" .
Допуск	Выполняет правила допуска для групп <i>wwUsers</i> , <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Annotation</i>
См. также	<code>ww_AnnotationSelect</code> <code>ww_AnnotationDelete</code> <code>ww_AnnotationInsert</code>

ww_CheckClientVersion

Проверяет версию запущенного приложения-клиента.

Синтаксис	<code>ww_CheckClientVersion AppName, AppVersion</code>
Параметры	<i>AppName</i> Название приложения. <i>AppVersion</i> Номер версии приложения.
Примечания	Эта хранимая процедура используется в системе IndustrialSQL Server для проверки, подходит ли версия приложения-клиента Wonderware для работы с базой данных. Приложение-клиент не сможет работать с базой данных, которая не поддерживает данного клиента. Внимание! Эта хранимая процедура предназначена только для использования в системе Wonderware. Не стоит пытаться использовать эту хранимую процедуру для каких-либо сторонних приложений-клиентов.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
См. также	<code>ww_CheckWhichDb</code>

ww_CheckWhichDb

Используется для определения, подается ли запрос в нужную базу данных.

Синтаксис `ww_CheckWhichDb dbType`

Параметры `dbType`

Идентификатор базы данных 1 = База данных Runtime; 2 = База данных Holding; 3 = Только для разработки.

Допуск Выполняет правила допуска для *общей* группы.

См. также `ww_CheckClientVersion`

ww_dbCheck

Используется для проверки клиентов FactorySuite 1000.

Синтаксис `ww_dbCheck dbType`

Параметры `dbType`

Идентификатор базы данных 1 = База данных Runtime; 2 = База данных Holding; 3 = Только для разработки.

Примечания Эта хранимая процедура используется только приложениями-клиентами, выпущенными до FS2000.

Допуск Выполняет правила допуска для *общей* группы.

ww_DeleteOlderEvents

Удаляет старые события из области хранения событий.

Синтаксис `ww_DeleteOlderEvents`

Примечания Эта хранимая процедура используется хранимой процедурой **ww_SpaceManager** каждые десять минут. Длительность хранения событий указывается в таблице *NamedSystemParameter*. События удаляются из таблицы *EventHistory*.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *NamedSystemParameter*, *EventHistory*

См. также `ww_DeleteOlderStrings`

`ww_DeleteOlderSummaries`

`ww_SpaceManager`

ww_DeleteOlderStrings

Удаляет устаревшие строки из области хранения текстовых данных.

Синтаксис `ww_DeleteOlderStrings`

Примечания Эта хранимая процедура используется хранимой процедурой **ww_SpaceManager** каждые десять минут. Длительность хранения строк указывается в таблице *NamedSystemParameter*. Текстовые строки удаляются из таблицы *StringHistory*.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *NamedSystemParameter*, *StringHistory*

См. также `ww_DeleteOlderEvents`
`ww_DeleteOlderSummaries`
`ww_SpaceManager`

ww_DeleteOlderSummaries

Удаляет устаревшие сводки из области хранения сводок.

Синтаксис `ww_DeleteOlderSummaries`

Примечания Эта хранимая процедура используется хранимой процедурой **ww_SpaceManager** каждые десять минут. Длительность хранения сводок указывается в таблице *NamedSystemParameter*. Текстовые строки будут удалены из таблицы *SummaryHistory*.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *NamedSystemParameter*, *SummaryHistory*

См. также `ww_DeleteOlderStrings`
`ww_DeleteOlderEvents`
`ww_SpaceManager`

ww_DetectorStringSelect

Выбирает строку детектора для указанного тэга события.

Синтаксис `ww_DetectorStringSelect` *TagName*

Параметры *TagName*
Имя тэга события.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *EventTag*

См. также `ww_ActionStringSelect`

ww_DiscreteConfig

Выдает детали настройки для всех дискретных тэгов в указанном узле хранения. Эта хранимая процедура также создает адрес дискретного тэга, состоящий из имени дискретного тэга и имени элемента.

Синтаксис `ww_DiscreteConfig NodeNo`

Параметры `NodeNo`

Числовой идентификатор узла хранения.

Примечание Числовой идентификатор конкретного узла хранения хранится в столбце *StorageNodeKey* таблицы *StorageNode*.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *Tag, Topic, DiscreteTag, IOServer*

См. также `ww_AnalogConfig`

`ww_StringConfig`

ww_DiscreteCount

Выдает суммарный подсчет всех дискретных тэгов для указанного узла хранения.

Синтаксис `ww_DiscreteCount NodeNo`

Параметры `NodeNo`

Числовой идентификатор узла хранения.

Примечания Числовой идентификатор конкретного узла хранения содержится в столбце *StorageNodeKey* таблицы *StorageNode*.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *Tag, TagRef*

См. также `ww_AnalogCount`

ww_DiscreteTagDelete

Удаляет дискретный тэг.

Синтаксис `ww_DiscreteTagDelete wwTagKey`

Параметры `wwTagKey`

Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. Это значение получается системой автоматически при добавлении тэга .

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *Tag*

См. также `ww_DiscreteTagInsert`

`ww_DiscreteTagUpdate`

`ww_DiscreteTagSelect`

ww_DiscreteTagInsert

Вставляет дискретный тэг.

Синтаксис	<code>ww_DiscreteTagInsert TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, MessageKey, InitialValue, TopicKey, IOServerKey</code>
Параметры	<i>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CurrentEditor, TopicKey, IOServerKey</i> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> в главе 2, "Тематические области и общие таблицы". <i>MessageKey, InitialValue</i> Эти параметры идентичны столбцам в таблице <i>DiscreteTag</i>. Подробнее о таблице <i>DiscreteTag</i> в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag, DiscreteTag</i>
См. также	<code>ww_DiscreteTagSelect</code> <code>ww_DiscreteTagUpdate</code> <code>ww_DiscreteTagDelete</code>

ww_DiscreteTagSelect

Выбирает дискретный тэг.

Синтаксис	<code>ww_DiscreteTagSelect wwTagKey</code>
Параметры	<i>wwTagKey</i> Уникальный идентификатора тэга.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag, DiscreteTag</i>
См. также	<code>ww_DiscreteTagInsert</code> <code>ww_DiscreteTagUpdate</code> <code>ww_DiscreteTagDelete</code>

ww_DiscreteTagUpdate

Обновляет дискретный тэг.

Синтаксис	<code>ww_DiscreteTagUpdate wwTagKey, TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, MessageKey, InitialValue, TopicKey, IOServerKey</code>
Параметры	<code>wwTagKey</code> Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. Это значение получается системой автоматически при добавлении тэга. <code>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CurrentEditor, TopicKey, IOServerKey</code> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> главе 2, "Тематические области и общие таблицы". <code>MessageKey, InitialValue</code> Эти параметры идентичны столбцам в таблице <i>DiscreteTag</i>. Подробнее о таблице <i>DiscreteTag</i> главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag, DiscreteTag</i>
См. также	<code>ww_DiscreteTagSelect</code> <code>ww_DiscreteTagInsert</code> <code>ww_DiscreteTagDelete</code>

ww_EngineeringUnitDelete

Удаляет единицу измерения из таблицы *EngineeringUnit*.

Синтаксис	<code>ww_EngineeringUnitDelete EUKey</code>
Параметры	<code>EUKey</code> Уникальный числовой идентификатор единицы измерения. Это значение получается системой автоматически при добавлении единицы измерения.
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>EngineeringUnit</i>
См. также	<code>ww_EngineeringUnitSelect</code> <code>ww_EngineeringUnitInsert</code> <code>ww_EngineeringUnitUpdate</code>

ww_EngineeringUnitInsert

Вставляет единицу измерения в таблицу *EngineeringUnit*.

Синтаксис `ww_EngineeringUnitInsert Unit, DefaultTagRate`

Параметры `Unit, DefaultTagRate`

Эти параметры идентичны столбцам в таблице *EngineeringUnit*.

Подробнее о таблице *EngineeringUnit* в главе 10, "Таблицы определения тэгов".

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *EngineeringUnit*

См. также `ww_EngineeringUnitDelete`
`ww_EngineeringUnitSelect`
`ww_EngineeringUnitUpdate`

ww_EngineeringUnitSelect

Выбирает единицу измерения из таблицы *EngineeringUnit*.

Синтаксис `ww_EngineeringUnitSelect EUKey`

Параметры `EUKey`

Уникальный числовой идентификатор единицы измерения. Это значение получается системой автоматически при добавлении единицы измерения.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *EngineeringUnit*

См. также `ww_EngineeringUnitDelete`
`ww_EngineeringUnitInsert`
`ww_EngineeringUnitUpdate`

ww_EngineeringUnitUpdate

Обновляет единицу измерения в таблице *EngineeringUnit*.

Синтаксис `ww_EngineeringUnitUpdate EUKey, Unit, DefaultTagRate`

Параметры `EUKey`

Уникальный числовой идентификатор единицы измерения. Это значение получается системой автоматически при добавлении новой единицы измерения.

`Unit, DefaultTagRate`

Эти параметры идентичны столбцам в таблице *EngineeringUnit*.

Подробнее о таблице *EngineeringUnit* в главе 10, "Таблицы определения тэгов".

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *EngineeringUnit*

См. также `ww_EngineeringUnitDelete`
 `ww_EngineeringUnitSelect`
 `ww_EngineeringUnitUpdate`

ww_EventHistory

Выдает информацию, хранящуюся в таблице *EventHistory* для каждого указанного тэга события.

Синтаксис `ww_EventHistory TagList, StartTime, EndTime`

Параметры *TagList*

Список тэгов событий, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.

StartTime

Метка даты/времени начала обработки запроса.

EndTime

Метка даты/времени окончания обработки запроса.

Примечания Эта хранимая процедура выдает информацию обо всех событиях, произошедших в промежутке между *StartTime* и *EndTime*.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *EventHistory*

ww_EventHistoryInsert

Вставляет строку в таблицу *EventHistory* для каждого произошедшего события.

Синтаксис `ww_EventHistoryInsert TagName, DateTime, DetectDateTime, Edge`

Параметры *TagName, DateTime, DetectDateTime, Edge*

Эти параметры идентичны столбцам в таблице *EventHistory*.

[Подробнее о таблице *EventHistory* в главе 2, "Тематические области и общие таблицы"](#).

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *EventHistory*

ww_EventSnapshot

Выдает значения тэга снимка для каждого тэга события.

Синтаксис `ww_EventSnapshot TagList, StartTime, EndTime`

Параметры `TagList`

Список тэгов событий, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.

`StartTime`

Метка даты/времени начала обработки запроса.

`EndTime`

Метка даты/времени окончания обработки запроса.

Допуск Выполняет правила допуска для *общей* группы.

Представление `v_EventSnapshot`

ww_EventTagDelete

Удаляет тэг события.

Синтаксис `ww_EventTagDelete wwTagKey`

Параметры `wwTagKey`

Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. Это значение получается системой автоматически при добавлении тэга события.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы `Tag`

См. также `ww_EventTagInsert`
`ww_EventTagUpdate`
`ww_EventTagSelect`

ww_EventTagDetail

Выдает детали каждого указанного тэга события, включая информацию детектора времени, если применимо.

Синтаксис `ww_EventTagDetail TagName`

Параметры `TagName`

Уникальный числовой идентификатор тэга события, для которого требуется выдать детали.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `Tag, EventTag, TimeDetectorDetail, DetectorType, ActionType`

ww_EventTagInsert

Вставляет тэг события.

Синтаксис	<code>ww_EventTagInsert TagName, Description, WideColumnName, CreatedBy, DateCreated, CurrentEditor, ScanRate, TimeDeadband, Logged, Status, PostDetectorDelay, UseThreadPool, DetectorTypeKey, DetectorString, ActionTypeKey, ActionString, Priority, Edge</code>
Параметры	<i>TagName, Description, WideColumnName, CreatedBy, DateCreated, CurrentEditor, ScanRate</i> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> в главе 2, "Тематические области и общие таблицы". <i>TimeDeadband, Logged, Status, PostDetectorDelay, UseThreadPool, DetectorTypeKey, DetectorString, ActionTypeKey, ActionString, Priority, Edge</i> Эти параметры идентичны столбцам в таблице <i>EventTag</i>. Подробнее о таблице <i>EventTag</i> в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag, EventTag</i>
См. также	<code>ww_EventTagDelete</code> <code>ww_EventTagUpdate</code> <code>ww_EventTagSelect</code>

ww_EventTagSelect

Выбирает тэг события.

Синтаксис	<code>ww_EventTagSelect wwTagKey,</code>
Параметры	<i>wwTagKey</i> Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. Это значение получается системой автоматически при добавлении тэга события.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag, EventTag</i>
См. также	<code>ww_EventTagDelete</code> <code>ww_EventTagUpdate</code> <code>ww_EventTagInsert</code>

ww_EventTagSelectAll

Используется системой управления событиями для определения изменений при динамической повторной инициализации.

Синтаксис	<code>ww_EventTagSelectAll</code>
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>EventTag, TimeDetectorDetail</i>
См. также	<code>ww_EventTagSelectDeleted</code> <code>ww_EventTagSelectUpdated</code> <code>ww_EventTagSelectInserted</code>

ww_EventTagSelectDeleted

Используется системой управления событиями для определения изменений при динамической повторной инициализации.

Синтаксис	<code>ww_EventTagSelectDeleted</code>
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>EventTagPendingDelete, TimeDetectorDetailPendingDelete</i>
См. также	<code>ww_EventTagSelectInserted</code> <code>ww_EventTagSelectUpdated</code> <code>ww_EventTagSelectDisabled</code>

ww_EventTagSelectDisabled

Используется системой управления событиями для определения изменений при динамической повторной инициализации.

Синтаксис	<code>ww_EventTagSelectDisabled</code>
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>EventTag, TimeDetectorDetail</i>
См. также	<code>ww_EventTagSelectInserted</code> <code>ww_EventTagSelectUpdated</code> <code>ww_EventTagSelectDeleted</code>

ww_EventTagSelectInserted

Используется системой управления событиями для определения изменений при динамической повторной инициализации.

Синтаксис	<code>ww_EventTagSelectInserted</code>
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>EventTag, TimeDetectorDetail</i>
См. также	<code>ww_EventTagSelectDeleted</code> <code>ww_EventTagSelectUpdated</code> <code>ww_EventTagSelectDisabled</code>

ww_EventTagSelectUpdated

Используется системой управления событиями для определения изменений при динамической повторной инициализации.

Синтаксис	<code>ww_EventTagSelectUpdated</code>
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>EventTag, TimeDetectorDetail</i>
См. также	<code>ww_EventTagSelectInserted</code> <code>ww_EventTagSelectUpdated</code>

ww_EventTagUpdate

Обновляет тэг события.

Синтаксис	<code>ww_EventTagUpdate wwTagKey, TagName, Description, WideColumnName, CreatedBy, DateCreated, CurrentEditor, ScanRate, TimeDeadband, Logged, Status, PostDetectorDelay, UseThreadPool, DetectorTypeKey, DetectorString, ActionTypeKey, ActionString, Priority, Edge</code>
Параметры	<code>wwTagKey</code> Уникальный числовой идентификатор тэга в одиночной системе IndustrialSQL Server. <i>TagName, Description, WideColumnName, CreatedBy, DateCreated, CurrentEditor, ScanRate</i> Эти параметры идентичны столбцам в таблице <i>Tag</i>. О таблице Tag см. в главе 2, "Тематические области и общие таблицы". <i>TimeDeadband, Logged, Status, PostDetectorDelay, UseThreadPool, DetectorTypeKey, DetectorString, ActionTypeKey, ActionString, Priority, Edge</i> Эти параметры идентичны столбцам в таблице <i>EventTag</i>. О таблице EventTag см. в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .

Таблицы	<i>Tag, EventTag</i>
См. также	ww_EventTagDelete ww_EventTagSelect ww_EventTagInsert

ww_GetDbRevision

Используется приложениями-клиентами IndustrialSQL для проверки редакции базы данных *Runtime*.

Синтаксис	ww_GetDbRevision
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>StorageNode</i>

ww_GetInSQLProcedureNames

Используется системой для отслеживания хранимой процедуры, относящейся к таблицам расширения.

Синтаксис	ww_GetInSQLProcedureNames
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>sysobjects, syscomments</i>
См. также	ww_LoadInSQLProcedureBody

ww_GetLastTagKey

Выдает детали последнего вставленного тэга.

Синтаксис	ww_GetLastTagKey <i>TagType</i>
Параметры	<i>TagType</i> Определяет тип тэга. 1 = Аналоговый; 2 = Дискретный; 3 = Текстовый; 4 = Сложный; 5 = События.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i>

ww_InsertStringSnapshot

Вставляет текстовую строку в таблицу *StringSnapshot* как результат действия над событием снимка текста.

Синтаксис `ww_InsertStringSnapshot EventLogKey, EventTagName, EventTime`

Параметры *EventLogKey*

Уникальный числовой идентификатор происхождения события, включившего операцию "снимка".

EventTagName

Имя тэга события, к которому относится тэг снимка.

EventTime

Метка времени происхождения события.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SnapshotTag, StringSnapshot, StringHistory*

ww_InSQLNSExpand

Расширяет отображение имен в IndustrialSQL Server до представления в виде дерева.

Синтаксис `ww_InSQLNSExpand PKey`

Параметры *PKey*

Локальная переменная для идентификации имени в IndustrialSQL Server.

Примечания Любой объект может иметь один или несколько объектов в иерархии именных пространств.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *PublicNameSpace, PrivateNameSpace*

См. также `ww_PrivateNSExpand`

`ww_PublicNSExpand`

`ww_SystemNSExpand`

ww_InSQLStatusSelect

Используется для выбора значения флага состояния базы данных, *DbStatus*, из таблицы *StorageNode*.

Синтаксис `ww_InSQLStatusSelect`

Примечания Эта хранимая процедура используется программами InSQL Configure и InSQL Control для определения изменений в базе данных.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *StorageNode*

См. также `ww_InSQLStatusSet`

ww_InSQLStatusSet

Устанавливает значение флага состояния, *DbStatus*, на значение больше 0, когда требуется провести изменение базы данных сервером. Устанавливает значение *DbStatus* обратно на 0, когда изменение базы данных завершено.

Синтаксис `ww_InSQLStatusSet DbStatus, Acquisition, Storage, DBServer`

Параметры `DbStatus, Acquisition, Storage, DBServer`

Эти параметры идентичны столбцам *DbStatus*, *DBModAcquisition*, *DBModStorage* и *DBModServer* в таблице *StorageNode*.

Подробнее о таблице *StorageNode* см. в главе 9, "Таблицы настройки системы".

Примечание. Обязателен только первый параметр; остальные служат для определения, нуждается ли указанная система в инициализации.

Примечания Когда в настройку базы данных *Runtime* вносятся изменения при помощи программы InSQL Configure, атрибут *DbStatus* в таблице *StorageNode* устанавливается на значение больше 0, что означает, что изменения необычные и необходима повторная инициализация. После определения необычности изменения, основанного на значении атрибута *DbStatus*, программа InSQL Control производит необходимые изменения, повторно инициализирует IndustrialSQL Server, если это необходимо, после чего устанавливает значение *DbStatus* обратно на 0, что означает, что повторная инициализация выполнена.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *StorageNode*

См. также `ww_InSQLStatusSelect`

ww_IOServerDelete

Удаляет сервер ввода-вывода из системных настроек.

Синтаксис `ww_IOServerDelete IOServerKey`

Параметры *IOServerKey*

Уникальный числовой идентификатор сервера ввода-вывода. Это значение получается системой автоматически при добавлении в системные настройки сервера ввода-вывода.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *IOServer*

См. также `ww_IOServerInsert`

`ww_IOServerSelect`

`ww_IOServerUpdate`

ww_IOServerInsert

Вставляет сервер ввода-вывода в системные настройки.

Синтаксис	<code>ww_IOServerInsert StorageNodeKey, ApplicationName, Description, Path, ComputerName, AutoStart, ExeType, InitializationStatus, ProtocolType</code>
Параметры	<i>StorageNodeKey, ApplicationName, Description, Path, ComputerName, AutoStart, ExeType, InitializationStatus, ProtocolType</i> Эти параметры идентичны столбцам в таблице <i>IOServer</i> . Подробнее о таблице <i>IOServer</i> см. в главе 9, "Таблицы настройки системы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>IOServer</i>
См. также	<code>ww_IOServerUpdate</code> <code>ww_IOServerSelect</code> <code>ww_IOServerDelete</code>

ww_IOServerSelect

Выбирает сервер ввода-вывода из системных настроек.

Синтаксис	<code>ww_IOServerSelect IOServerKey</code>
Параметры	<i>IOServerKey</i> Уникальный числовой идентификатор сервера ввода-вывода. Это значение получается системой автоматически при добавлении в системные настройки сервера ввода-вывода.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>IOServer</i>
См. Также	<code>ww_IOServerInsert</code> <code>ww_IOServerUpdate</code> <code>ww_IOServerDelete</code>

ww_IOServerUpdate

Обновляет сервер ввода-вывода в системных настройках.

Синтаксис	<code>ww_IOServerUpdate IOServerKey, StorageNodeKey, ApplicationName, Description, Path, ComputerName, AutoStart, ExeType, InitializationStatus, ProtocolType</code>
Параметры	<i>IOServerKey</i> Уникальный числовой идентификатор сервера ввода-вывода. Это значение получается системой автоматически при добавлении в системные настройки сервера ввода-вывода. <i>StorageNodeKey, ApplicationName, Description, Path, ComputerName, AutoStart, ExeType, InitializationStatus, ProtocolType</i> Эти параметры идентичны столбцам в таблице <i>IOServer</i>. Подробнее о таблице <i>IOServer</i> см. в главе 9, "Таблицы настройки системы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>IOServer</i>
См. также	<code>ww_IOServerUpdate</code> <code>ww_IOServerSelect</code> <code>ww_IOServerDelete</code>

ww_LoadInSQLProcedureBody

Используется системой для отслеживания хранимой процедуры, относящейся к таблицам расширения.

Синтаксис	<code>ww_LoadSQLProcedureBody objname</code>
Параметры	<i>objname</i> Имя хранимой процедуры для загрузки.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>syscomments</i>

ww_MessageDelete

Удаляет сообщение для дискретного тэга.

Синтаксис	<code>ww_MessageDelete MessageKey</code>
Параметры	<i>MessageKey</i> Уникальный числовой идентификатор сообщения.
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Message</i>
См. также	<code>ww_MessageSelect</code> <code>ww_MessageUpdate</code> <code>ww_MessageDelete</code>

ww_MessageInsert

Вставляет сообщение для дискретного тэга.

Синтаксис `ww_MessageInsert Message0, Message1`

Параметры `Message0, Message1`

Эти параметры идентичны столбцам в таблице *Message*.

Подробнее о таблице *Message* см. в главе 10, "Таблицы определения тэгов".

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *Message*

См. также `ww_MessageSelect`

`ww_MessageUpdate`

`ww_MessageDelete`

ww_MessageSelect

Выбирает сообщение для дискретного тэга.

Синтаксис `ww_MessageSelect MessageKey`

Параметры `MessageKey`

Уникальный числовой идентификатор сообщения.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *Message*

См. также `ww_MessageInsert`

`ww_MessageUpdate`

`ww_MessageDelete`

ww_MessageUpdate

Обновляет сообщение для дискретного тэга.

Синтаксис `ww_MessageUpdate MessageKey, Message0, Message1`

Параметры `MessageKey`

Уникальный числовой идентификатор сообщения. Это значение получается системой автоматически при создании нового сообщения.

`Message0, Message1`

Эти параметры идентичны столбцам в таблице *Message*.

Подробнее о таблице *Message* см. в главе 10, "Таблицы определения тэгов".

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *Message*

См. также `ww_MessageSelect`

`ww_MessageInsert`

`ww_MessageDelete`

ww_ModLogStatus

Используется для получения статистики отслеживания изменений.

Синтаксис	<code>ww_ModLogStatus</code>
Примечания	Эта хранимая процедура используется программой InSQL Configure. Отслеживание изменений управляется значением системного параметра с именем <i>ModLogStatus</i> , который хранится в столбце <i>NumericValue</i> таблицы <i>NamedSystemParameter</i> . Если значение этого столбца равно 1, то отслеживание изменений включено (0 = выключено).
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>NamedSystemParameter</i>

ww_PrivateNSAddGroup

Добавляет группу объектов, стоящих ниже выбранного объекта в иерархии именных пространств, в личные именные пространства.

Синтаксис	<code>ww_PrivateNSAddGroup Name, ParentKey, Type</code>
Параметры	<i>Name, ParentKey, Type</i> Эти параметры идентичны столбцам в таблице <i>PrivateNameSpace</i> . Подробнее о таблице <i>PrivateNameSpace</i> см. в главе 7, "Таблицы именных пространств и групп" .
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>PrivateNameSpace</i>
См. также	<code>ww_PrivateNSDeleteGroup</code> <code>ww_PrivateNSUpdateGroup</code>

ww_PrivateNSAddLeaf

Добавляет одиночный объект, стоящий ниже выбранного объекта в иерархии именных пространств, в личные именные пространства.

Синтаксис	<code>ww_PrivateNSAddLeaf wwTagKey, NameKey, ServerKey</code>
Параметры	<i>wwTagKey</i> Уникальный идентификатор для тэга в IndustrialSQL Server. <i>NameKey</i> Уникальный идентификатор имени объекта в личных именных пространствах. <i>ServerKey</i> Уникальный идентификатор для IndustrialSQL Server.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>PrivateGroupTag, TagRef</i>
См. также	<code>ww_PrivateNSDeleteLeaf</code>

ww_PrivateNSDeleteGroup

Удаляет группу объектов, стоящих ниже выбранного объекта в иерархии именных пространств, из личных именных пространств.

Синтаксис `ww_PrivateNSDeleteGroup NameKey`

Параметры `NameKey`

Уникальный идентификатор имени группы объектов в личных именных пространствах.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `PrivateNameSpace`

См. также `ww_PrivateNSAddGroup`

`ww_PrivateNSUpdateGroup`

ww_PrivateNSDeleteLeaf

Удаляет одиночный объект, стоящий ниже выбранного объекта в иерархии именных пространств, из личных именных пространств.

Синтаксис `ww_PrivateNSDeleteLeaf NameKey, wwTagKey`

Параметры `NameKey`

Уникальный идентификатор имени объекта в личных именных пространствах.

`wwTagKey`

Уникальный идентификатор для тэга в IndustrialSQL Server.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `PrivateGroupTag, TagRef`

См. также `ww_PrivateNSAddLeaf`

ww_PrivateNSExpand

Расширяет уровень дерева ниже одиночного объекта верхнего уровня в личных именных пространствах.

Синтаксис `ww_PrivateNSExpand Pkey, FilterStr`

Параметры `PKey`

Локальная переменная, служащая для идентификации объекта в личных именных пространствах.

`FilterStr`

Используется для упорядочивания имен тэгов в личных именных пространствах.

Примечания В иерархии именных пространств перед объектом верхнего уровня может стоять один или несколько объектов.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `PrivateNameSpace, PrivateGroupTag, TagRef`

См. также `ww_PrivateNSSelect`

ww_PrivateNSSelect

Выбирает одиночный объект в личных именных пространствах.

Синтаксис `ww_PrivateNSSelect`

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `PrivateNameSpace`

См. также `ww_PublicNSExpand`

ww_PrivateNSUpdateGroup

Обновляет объект, стоящий ниже выбранного объекта в иерархии именных пространств, в личных именных пространствах.

Синтаксис `ww_PrivateNSUpdateGroup NameKey, Name, Type`

Параметры `NameKey, Name, Type`

Эти параметры идентичны столбцам в таблице `PrivateNameSpace`.

Подробнее о таблице `PrivateNameSpace` см. в главе 8, "Таблицы именных пространств и групп".

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `PrivateNameSpace`

См. также `ww_PrivateNSAddGroup`

`ww_PrivateNSDeleteGroup`

ww_PublicNSAddGroup

Добавляет группу объектов в общие именные пространства ниже выбранного объекта в иерархии именных пространств.

Синтаксис `ww_PublicNSAddGroup Name, ParentKey, Type`

Параметры `Name, ParentKey, Type`

Эти параметры идентичны столбцам в таблице *PublicNameSpace*.
 Подробнее о таблице *PublicNameSpace* см. в главе 7, "Таблицы именных пространств и групп".

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *PublicNameSpace*

См. также `ww_PublicNSDeleteGroup`
`ww_PublicNSUpdateGroup`

ww_PublicNSAddLeaf

Добавляет одиночный объект в общие именные пространства, ниже выбранного объекта в иерархии именных пространств.

Синтаксис `ww_PublicNSAddLeaf wwTagKey, NameKey, ServerKey`

Параметры `wwTagKey`

Уникальный идентификатор для тэга в IndustrialSQL Server.

NameKey

Уникальный идентификатор имени объекта в общих именных пространствах.

ServerKey

Уникальный идентификатор для IndustrialSQL Server.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *PublicGroupTag, TagRef*

См. также `ww_PublicNSDeleteLeaf`

ww_PublicNSDeleteGroup

Удаляет группу объектов в общих именных пространствах, стоящую ниже выбранного объекта в иерархии именных пространств.

Синтаксис `ww_PublicNSDeleteGroup NameKey`

Параметры *NameKey*

Уникальный идентификатор имени группы объектов в общих именных пространствах.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *PublicNameSpace*

См. также `ww_PublicNSAddGroup`
`ww_PublicNSUpdateGroup`

ww_PublicNSDeleteLeaf

Удаляет одиночный объект в общих именных пространствах, стоящий ниже выбранного объекта в иерархии именных пространств.

Синтаксис `ww_PublicNSDeleteLeaf NameKey, wwTagKey`

Параметры *NameKey*
Уникальный идентификатор имени объекта в общих именных пространствах.
wwTagKey
Уникальный идентификатор для тэга в IndustrialSQL Server.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы *PublicGroupTag, TagRef*

См. также `ww_PublicNSAddLeaf`

ww_PublicNSExpand

Расширяет уровень дерева ниже одиночного объекта верхнего уровня в общих именных пространствах.

Синтаксис `ww_PublicNSExpand Pkey, FilterStr`

Параметры *PKey*
Локальная переменная для идентификации объекта в общих именных пространствах.
FilterStr
Используется для упорядочивания имен тэгов в общих именных пространствах

Примечания В иерархии именных пространств перед объектом может стоять один или несколько объектов.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *PublicNameSpace, PublicGroupTag, TagRef*

См. также `ww_PublicNSSelect`

ww_PublicNSSelect

Выбирает одиночный объект общих именных пространств.

Синтаксис `ww_PublicNSSelect`

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *PublicNameSpace*

См. также `ww_PublicNSExpand`

ww_PublicNSUpdateGroup

Обновляет группу объектов в общих именованных пространствах ниже выбранного объекта в иерархии именованных пространств.

Синтаксис

`ww_PublicNSAddGroup NameKey, Name, Type`

Параметры

NameKey, Name, Type

Эти параметры идентичны столбцам в таблице *PublicNameSpace*.

Подробнее о таблице *PublicNameSpace* см. в главе 7, "Таблицы именованных пространств и групп".

Допуск

Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы

PublicNameSpace

См. также

ww_PublicNSAddGroup

ww_PublicNSDeleteGroup

ww_RedirectToInTouch

Перенаправляет адрес тэга (имя элемента) на узел InTouch, но не на исходный сервер ввода-вывода.

Синтаксис

`ww_RedirectToInTouch IOServerKey, InTouchNodeKey`

Параметры

IOServerKey

Уникальный числовой идентификатор сервера ввода-вывода.

InTouchNodeKey

Уникальный числовой идентификатор узла.

Примечания

При перенаправлении на InTouch все значения тэгов будут подаваться с InTouch, а не прямо с сервера ввода-вывода. Если перенаправляется сервер ввода-вывода, задействуются все темы и тэги определенного сервера ввода-вывода.

Допуск

Выполняет правила допуска для группы *wwAdministrators*.

Таблицы

IOServer, Tag, Topic, InTouchSpecific

ww_SetAISamples

Устанавливает количество выборок, которые активный объект может содержать для указанного тэга.

Синтаксис `ww_SetAISamples TagName, Samples`

Параметры `TagName`

Имя тэга, для которого устанавливается количество выборок.

`Samples`

Количество выборок в активном объекте указанного тэга.

Примечания Система ставит начальное количество выборок для каждого тэга = 65. Количество выборок для тэга хранится в столбце *SampleInActiveImage* таблицы *Tag*. Система неоднократно пересчитывает оптимальное количество выборок для каждого тэга, основанное на полученной скорости сбора данных. Если подсчитанное значение превышает текущее значение в базе данных, система обновляет значение в таблице *Tag*.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `Tag`

ww_SnapToSummary

Используется системой управления событиями при настройке операций создания сводок.

Синтаксис `ww_SnapToSummary opkey, start, end, datestamp`

Параметры `start`

Дата/время начала подсчета.

`end`

Дата/время окончания подсчета.

`datestamp`

Время, когда была создана сводка.

`opkey`

Внутренняя переменная, идентифицирующая операцию создания сводки, которая должна быть выполнена.

Допуск Выполняет правила допуска для групп *wwPowerUsers* и *wwAdministrators*.

Таблицы `SummaryHistory, SummaryData, SummaryTagList, AnalogSnapshot, EventHistory, SnapshotTag`

ww_SpaceManager

Используется системой управления дисковым пространством для хранения архивных данных для строк, сводок и событий.

Синтаксис `ww_SpaceManager`

Примечания Эта хранимая процедура автоматически выполняется системой каждые 10 минут. Она выполняет хранимые процедуры **ww_DeleteOlderStrings**, **ww_DeleteOlderEvents**, и **ww_DelectOlderSummaries** для удаления устаревших архивных данных. Продолжительность хранения текстовых архивов, архивов сводок и архивов событий основана на системных параметрах с именем, хранящихся в таблице *NamedSystemParameter*.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

См. также `ww_DeleteOlderStrings`
`ww_DeleteOlderEvents`
`ww_DeleteOlderSummaries`

ww_StorageLocationSelect

Выбирает место хранения.

Синтаксис `ww_StorageLocationSelect StorageType, StorageNodeKey`

Параметры *StorageType*

Тип хранения, используемый для указанной области. 1 = Циклическое; 2 = Альтернативное; 3 = Буферное; 4 = Постоянное. При циклическом методе место хранения может быть только одно.

StorageNodeKey

Уникальный числовой идентификатор узла хранения.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *StorageLocation*

См. также `ww_StorageLocationUpdate`

ww_StorageLocationUpdate

Обновляет место хранения.

Синтаксис `ww_StorageLocationUpdate StorageType, StorageNodeKey, SortOrder, Path, MaxMBSIZE, MinMBThreshold`

Параметры *StorageType, StorageNodeKey, SortOrder, Path, MaxMBSIZE, MinMBThreshold*

Эти параметры идентичны столбцам в таблице *StorageLocation*.

Подробнее о таблице *StorageLocation* см. в главе 9, "Таблицы настройки системы".

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *StorageLocation*

См. также `ww_StorageLocationSelect`

ww_StringConfig

Выдает детали настройки для всех текстовых тэгов в указанном узле хранения. Эта хранимая процедура также создает адрес текстового тэга, состоящий из имени текстового тэга и имени элемента.

Синтаксис	<code>ww_StringConfig NodeNo</code>
Параметры	<i>NodeNo</i> Числовой идентификатор узла хранения.
Примечания	Числовой идентификатор конкретного узла хранения содержится в столбце <i>StorageNodeKey</i> таблицы <i>StorageNode</i> .
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i> , <i>Topic</i> , <i>StringTag</i> , <i>IOServer</i>
См. также	<code>ww_AnalogConfig</code> <code>ww_DiscreteConfig</code>

ww_StringCount

Выдает итоговый подсчет всех текстовых тэгов для указанного узла хранения.

Синтаксис	<code>ww_StringCount nodeno</code>
Параметры	<i>nodeno</i> Числовой идентификатор узла хранения.
Примечания	Числовой идентификатор конкретного узла хранения содержится в столбце <i>StorageNodeKey</i> таблицы <i>StorageNode</i> .
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i> , <i>TagRef</i> , <i>Topic</i>

ww_StringTagDelete

Удаляет текстовый тэг.

Синтаксис	<code>ww_StringTagDelete wwTagKey</code>
Параметры	<i>wwTagKey</i> Уникальный идентификатора тэга.
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag</i>
See Also	<code>ww_StringTagInsert</code> <code>ww_StringTagSelect</code> <code>ww_StringTagUpdate</code>

ww_StringTagInsert

Вставляет текстовый тэг.

Синтаксис	<code>ww_StringTagInsert TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, MaxLength, InitialValue, TopicKey, IOServerKey, CurrentEditor</code>
Параметры	<i>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, TopicKey, IOServerKey</i> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> см. в главе 2, "Тематические области и общие таблицы". <i>MaxLength, InitialValue</i> Эти параметры идентичны столбцам в таблице <i>StringTag</i>. Подробнее о таблице <i>StringTag</i> см. в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag, StringTag</i>
См. Также	<code>ww_StringTagDelete</code> <code>ww_StringTagSelect</code> <code>ww_StringTagUpdate</code>

ww_StringTagSelect

Выбирает текстовый тэг.

Синтаксис	<code>ww_StringTagSelect wwTagKey</code>
Параметры	<i>wwTagKey</i> Уникальный идентификатора тэга.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag, StringTag</i>
См. также	<code>ww_StringTagDelete</code> <code>ww_StringTagInsert</code> <code>ww_StringTagUpdate</code>

ww_StringTagUpdate

Обновляет текстовый тэг.

Синтаксис	<code>ww_StringTagUpdate wwTagKey, TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, MaxLength, InitialValue, TopicKey, IOServerKey, CurrentEditor</code>
Параметры	<code>wwTagKey</code> Уникальный идентификатора тэга. <code>TagName, Description, AcquisitionType, StorageType, StorageRate, ItemName, WideColumnName, TimeDeadband, CreatedBy, DateCreated, CurrentEditor, TopicKey, IOServerKey</code> Эти параметры идентичны столбцам в таблице <i>Tag</i>. Подробнее о таблице <i>Tag</i> см. в главе 2, "Тематические области и общие таблицы". <code>MaxLength, InitialValue</code> Эти параметры идентичны столбцам в таблице <i>StringTag</i>. Подробнее о таблице <i>StringTag</i> см. в главе 2 "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>StringTag, Tag</i>
См. также	<code>ww_StringTagDelete</code> <code>ww_StringTagInsert</code> <code>ww_StringTagUpdate</code>

ww_SummaryDetail

Выдает детали сводки для одного или нескольких тэгов, связанных с конкретной операцией создания сводки.

Синтаксис	<code>ww_SummaryDetail TagList, StartTime, EndTime, OrderBy</code>
Параметры	<code>TagList</code> Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8 знаков. Тип агрегации тэга может быть выборочно внесен в список для каждого тэга в этом списке. <code>StartTime</code> Метка даты/времени начала подсчета. <code>EndTime</code> Метка даты/времени окончания подсчета. <code>OrderBy</code> Столбец, согласно которому будут упорядочены результаты. По умолчанию используется столбец <i>Tagname</i>.
Допуск	Выполняет правила допуска для <i>общей</i> группы.

Представление *v_SummaryData*

Примеры Этот пример выдает средние и минимальные значения 'ReactTemp' и максимальное значение 'ReactLevel' между 12:12 p.m. и 2:14 p.m. 12 мая 1999 г. Выданные строки сортируются по дате создания сводки.

```
ww_SummaryDetail "ReactTemp('AVG','MIN'), ReactLevel('MAX')",
"May 12, 1999 12:12:00", "May 12, 1999 14:14:00",
"SummaryDate"
```

Этот пример выдает все агрегированные значения для 'ReactTemp' между 12:12 p.m. и 2:14 p.m. 12 мая 1999.

```
ww_SummaryDetail "ReactTemp, ReactLevel", "May 12, 1999
12:12:00", "May 12, 1999 14:14:00"
```

ww_SummaryOperationDelete

Удаляет операцию создания сводки из таблицы *SummaryOperation*.

Синтаксис *ww_SummaryOperationDelete OperationKey*

Параметры *OperationKey*
Уникальный числовой идентификатор создания сводки. Это значение получается системой автоматически при определении процедуры.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SummaryOperation*

См. также *ww_SummaryOperationInsert*
ww_SummaryOperationSelect
ww_SummaryOperationUpdate

ww_SummaryOperationInsert

Вставляет операцию создания сводки из таблицы *SummaryOperation*.

Синтаксис *ww_SummaryOperationInsert TagName, CalcType, Duration, Resolution, TimeStamp, Description*

Параметры *TagName, CalcType, Duration, Resolution, TimeStamp, Description*
Эти параметры идентичны столбцам в таблице *SummaryOperation*.
[Подробнее о таблице *SummaryOperation* см. в главе 8, "Таблицы сводок".](#)

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SummaryOperation*

См. также *ww_SummaryOperationDelete*
ww_SummaryOperationSelect
ww_SummaryOperationUpdate

ww_SummaryOperationSelect

Выбирает операцию создания сводки из таблицы *SummaryOperation*.

Синтаксис `ww_SummaryOperationSelect EventTagName, CalcType, Duration, Resolution, TimeStamp`

Параметры *EventTagName*
Имя тэга события, с которым связана операция создания сводки.
CalcType, Duration, Resolution, TimeStamp, Description

Эти параметры идентичны столбцам в таблице *SummaryOperation*.
[Подробнее о таблице *SummaryOperation* см. в главе 8, "Таблицы сводок".](#)

Примечания Параметры этой хранимой процедуры используются в трех случаях: (1) если параметры не указаны, выдаются все данные сводок; (2) если указан параметр *EventTagName*, выдаются все данные созданных сводок для тэгов событий; (3) если указаны все параметры, выдается только отвечающая им всем сводка.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *SummaryOperation*

См. также `ww_SummaryOperationInsert`
`ww_SummaryOperationDelete`
`ww_SummaryOperationUpdate`

ww_SummaryOperationUpdate

Обновляет операцию создания сводки в таблице *SummaryOperation*.

Синтаксис `ww_SummaryOperationUpdate OperationKey, TagName, CalcType, Duration, Resolution, TimeStamp, Description`

Параметры *OperationKey*
Уникальный числовой идентификатор создания сводки. Это значение получается системой автоматически при определении процедуры.
TagName, CalcType, Duration, Resolution, TimeStamp, Description

Эти параметры идентичны столбцам в таблице *SummaryOperation*.
[Подробнее о таблице *SummaryOperation* см. в главе 8, "Таблицы сводок".](#)

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SummaryOperation*

См. также `ww_SummaryOperationInsert`
`ww_SummaryOperationSelect`
`ww_SummaryOperationDelete`

ww_SummaryTagListDelete

Удаляет информацию сводки для тэга из таблицы *SummaryTagList*.

Синтаксис `ww_SummaryTagListDelete SumVarKey`

Параметры *SumVarKey*
Уникальный числовой идентификатор тэга, для которого создается сводка.
Это значение получается системой автоматически при создании тэга сводки.

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SummaryTagList*

См. также `ww_SummaryTagListInsert`
`ww_SummaryTagListSelect`
`ww_SummaryTagListUpdate`

ww_SummaryTagListInsert

Вставляет информацию сводки для тэга в таблицу *SummaryTagList*.

Синтаксис `ww_SummaryTagListInsert TagName, OperationKey, LowerLimit, UpperLimit, Description`

Параметры *TagName, OperationKey, LowerLimit, UpperLimit, Description*
Эти параметры идентичны столбцам в таблице *SummaryTagList*.
[Подробнее о таблице *SummaryTagList* см. в главе 8, "Таблицы сводок"](#)

Допуск Выполняет правила допуска для группы *wwAdministrators*.

Таблицы *SummaryTagList*

См. также `ww_SummaryTagListDelete`
`ww_SummaryTagListSelect`
`ww_SummaryTagListUpdate`

ww_SummaryTagListSelect

Выбирает информацию о сводке для тэга из таблицы *SummaryTagList*.

Синтаксис `ww_SummaryTagListSelect OperationKey, TagName`

Параметры *OperationKey*
Уникальный числовой идентификатор создания сводки. Это значение получается системой автоматически при создании сводки.
TagName
Имя тэга, для которого создается сводка.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *SummaryTagList, SummaryOperation*

См. также `ww_SummaryTagListDelete`
`ww_SummaryTagListInsert`
`ww_SummaryTagListUpdate`

ww_SummaryTagListUpdate

Обновляет информацию сводки в таблице *SummaryTagList*.

Синтаксис	<code>ww_SummaryTagListUpdate SumVarKey, TagName, OperationKey, LowerLimit, UpperLimit, Description</code>
Параметры	<i>SumVarKey</i> Уникальный числовой идентификатор тэга, для которого создается сводка. Это значение получается системой автоматически при создании тэга сводки. <i>TagName, OperationKey, LowerLimit, UpperLimit, Description</i> Эти параметры идентичны столбцам в таблице <i>SummaryTagList</i>. Подробнее о таблице <i>SummaryTagList</i> см. в главе 8, "Таблицы сводок".
Допуск	Выполняет правила допуска для группы <i>wwAdministrators</i> .
Таблицы	<i>SummaryTagList</i>
См. также	<code>ww_SummaryTagListDelete</code> <code>ww_SummaryTagListSelect</code> <code>ww_SummaryTagListInsert</code>

ww_SystemNSEExpand

Расширяет уровень дерева ниже одиночного объекта верхнего уровня в системных именованных пространствах.

Синтаксис	<code>ww_SystemNSEExpand PKey, FKey1, FKey2, FKey3, TokenType, FilterStr</code>
Параметры	<i>PKey</i> Локальная переменная, служащая для идентификации объекта в "дереве". <i>FKey1 - FKey2</i> Локальная переменная, служащая для определения положения объекта в "дереве". <i>TokenType</i> Определяет тип группы системных именованных пространств. <i>FilterStr</i> Используется для упорядочивания имен тэгов в системных именованных пространствах.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>UserDetail, ServerList, StorageLocation, StorageNode, IOServerType, IOServer, Message, EngineeringUnit, InTouchNode, Topic, Tag</i>
См. также	<code>ww_PrivateNSEExpand</code> <code>ww_PublicNSEExpand</code>

ww_SystemNSEExpand2

Расширяет уровень дерева ниже одиночного объекта верхнего уровня в системных именованных пространствах.

Примечание. Это упрощенный вариант хранимой процедуры `ww_SystemNSEExpand`.

Синтаксис	<code>ww_SystemNSEExpand PKey, FilterStr</code>
Параметры	<i>PKey</i> Локальная переменная, служащая для идентификации объекта в "дереве". <i>FilterStr</i> Используется для упорядочивания имен тэгов в системных именованных пространствах.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>SystemNameSpace, UserDetail, StorageLocation, InTouchNode, Tag, Message, EngineeringUnit, IOServerType, Topic</i>
См. также	<code>ww_PrivateNSEExpand</code> <code>ww_PublicNSEExpand</code>

ww_TagInfo

Выдает информацию тэга из таблицы *Tag* для каждого указанного тэга.

Синтаксис	<code>ww_TagInfo TagList</code>
Параметры	<i>TagList</i> Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i>

ww_TagType

Выдает тип тэга из таблицы *Tag* для каждого указанного тэга.

Синтаксис	<code>ww_TagType TagList</code>
Параметры	<i>TagList</i> Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag</i>

ww_TimeDetectorDetailInsert

Вставляет детали детектора времени, связанные с конкретным тэгом события.

Синтаксис	<code>ww_TimeDetectorDetailInsert FrequencyID, TagName, Periodicity, StartDateTime, RunTimeDay, RunTimeHour, RunTimeMin</code>
Параметры	<i>FrequencyID, TagName, Periodicity, StartDateTime, RunTimeDay, RunTimeHour, RunTimeMin</i> Эти параметры идентичны столбцам в таблице <i>TimeDetectorDetail</i> . Подробнее о таблице <i>TimeDetectorDetail</i> см. в главе 4, "Таблицы хранения событий" .
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>TimeDetectorDetail</i>
См. также	<code>ww_TimeDetectorDetailSelect</code> <code>ww_TimeDetectorDetailUpdate</code>

ww_TimeDetectorDetailSelect

Выбирает детектор времени из таблицы *TimeDetectorDetail*.

Синтаксис	<code>ww_TimeDetectorDetailSelect TagName</code>
Параметры	<i>TagName</i> Уникальное имя тэга события, связанного с детектором времени.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>TimeDetectorDetail</i>
См. также	<code>ww_TimeDetectorDetailUpdate</code> <code>ww_TimeDetectorDetailInsert</code>

ww_TimeDetectorDetailUpdate

Обновляет детектор времени в таблице *TimeDetectorDetail*.

Синтаксис	<code>ww_TimeDetectorDetailUpdate FrequencyID, TagName, Periodicity, StartDateTime, RunTimeDay, RunTimeHour, RunTimeMin</code>
Параметры	<i>FrequencyID, TagName, Periodicity, StartDateTime, RunTimeDay, RunTimeHour, RunTimeMin</i> Эти параметры идентичны столбцам в таблице <i>TimeDetectorDetail</i> . Подробнее о таблице <i>TimeDetectorDetail</i> см. в главе 4, "Таблицы хранения событий" .
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>TimeDetectorDetail</i>
См. также	<code>ww_TimeDetectorDetailSelect</code> <code>ww_TimeDetectorDetailInsert</code>

ww_TopicDelete

Удаляет тему ввода-вывода.

Синтаксис	<code>ww_TopicDelete TopicKey</code>
Параметры	<i>TopicKey</i> Уникальный числовой идентификатор темы.
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Topic</i>
См. также	<code>ww_TopicInsert</code> <code>ww_TopicSelect</code> <code>ww_TopicUpdate</code>

ww_TopicInsert

Вставляет тему ввода-вывода.

Синтаксис	<code>ww_TopicInsert StorageNodeKey, IOServerKey, Name, TimeOut</code>
Параметры	<i>StorageNodeKey, IOServerKey, Name, TimeOut</i> Эти параметры идентичны столбцам в таблице <i>Topic</i> . Подробнее о таблице <i>Topic</i> см. в главе 2, "Тематические области и общие таблицы" .
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Topic</i>
См. также	<code>ww_TopicDelete</code> <code>ww_TopicSelect</code> <code>ww_TopicUpdate</code>

ww_TopicSelect

Выбирает тему ввода-вывода.

Синтаксис	<code>ww_TopicSelect TopicKey</code>
Параметры	<i>TopicKey</i> Уникальный числовой идентификатор темы.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Topic</i>
См. также	<code>ww_TopicDelete</code> <code>ww_TopicInsert</code> <code>ww_TopicUpdate</code>

ww_TopicUpdate

Обновляет тему ввода-вывода.

Синтаксис	<code>ww_TopicUpdate TopicKey, StorageNodeKey, IOServerKey, Name, TimeOut</code>
Параметры	<i>TopicKey</i> Уникальный числовой идентификатор темы. <i>StorageNodeKey, IOServerKey, Name, TimeOut</i> Эти параметры идентичны столбцам в таблице <i>Topic</i>. Подробнее о таблице <i>Topic</i> см. в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Topic</i>
См. также	<code>ww_TopicDelete</code> <code>ww_TopicInsert</code> <code>ww_TopicSelect</code>

ww_UserDetailUpdate

Позволяет наполнять таблицу *UserDetail* информацией из таблицы *sysusers*.

Синтаксис	<code>ww_UserDetailUpdate</code>
Допуск	Выполняет правила допуска для группы <i>wwAdministrators</i> .
Таблицы	<i>UserDetail, sysusers</i>

Пользовательские хранимые операции

Пользовательские хранимые процедуры используются при подаче запроса в базу данных на выдачу информации об указанных тэгах. Эти хранимые процедуры позволяют получать информацию об определениях тэгов или сужать область запроса до определенной таблицы. Эти хранимые процедуры можно использовать для подаче запроса с использованием специальных инструментов, таких как SQL Server Query Analyzer.

ww_AddTag

Вставляет тэг.

Синтаксис	<code>ww_AddTag wTagName, wTopicKey, wIOServerKey, wStorageNodeKey, wDescription, wAcquisitionType, wStorageType, wAcquisitionRate, wStorageRate, wItemName, wWideColumnName, wTimeDeadband, wCurrentEditor</code>
Параметры	<code>wTagName, wTopicKey, wIOServerKey, wStorageNodeKey, wDescription, wAcquisitionType, wStorageType, wAcquisitionRate, wStorageRate, wItemName, wWideColumnName, wTimeDeadband, wCurrentEditor</code> Эти параметры идентичны столбцам в таблице <i>Tag</i> (без префикса "w"). Подробнее о таблице Tag см. в главе 2, "Тематические области и общие таблицы".
Допуск	Выполняет правила допуска для групп <i>wwPowerUsers</i> и <i>wwAdministrators</i> .
Таблицы	<i>Tag</i>
См. также	ww_DeleteTag

ww_AnalogDetail

Выдает информацию об одном или нескольких указанных аналоговых тэгах, включая имя тэга, описание, скорость получения, единицу измерения и минимальные и максимальные значения в единицах измерения.

Синтаксис	<code>ww_AnalogDetail TagList</code>
Параметры	<i>TagList</i> Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>Tag, EngineeringUnit, AnalogTag</i>
См. также	ww_DiscreteDetail ww_StringDetail

ww_Annotation

Выдает любые аннотации, созданные для одного или нескольких указанных тэгов за указанный период времени.

Синтаксис `ww_Annotation TagList, StartTime, EndTime`

Параметры `TagList`

Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.

`StartTime`

Стартовое время обработки запроса.

`EndTime`

Время окончания обработки запроса.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `UserDetail, Annotation`

ww_DeleteTag

Удаляет тэг.

Синтаксис `ww_DeleteTag TagName`

Параметры `TagName`

Имя удаляемого тэга.

Допуск Выполняет правила допуска для групп `wwPowerUsers` и `wwAdministrators`.

Таблицы `Tag`

См. также `ww_AddTag`

ww_DiscreteDetail

Выдает информацию об одном или нескольких указанных дискретных тэгах, включая имя тэга, описание, сообщение для состояния ИСТИНА (1) и сообщение для состояния ЛОЖЬ (0).

Синтаксис `ww_DescreteDetail TagList`

Параметры `TagList`

Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `Tag, Message, DiscreteTag`

См. также `ww_AnalogDetail`

`ww_StringDetail`

ww_ResetDefaults

Сбывает все хранимые процедуры, относящиеся к таблицам расширения (например, `ww_SetResolution`, `ww_SetRowCount`), на оригинальные значения.

Синтаксис `ww_ResetDefaults`

Допуск Выполняет правила допуска для *общей* группы.

ww_SetAnalogCyclic

Уточняет способ обработки запроса — циклично или методом исключения (дельта). По умолчанию эта хранимая процедура находится в положении ВКЛЮЧЕНО. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока значение не будет сброшено.

Синтаксис `ww_SetAnalogCyclic Mode`

Параметры *Mode*

Определяет объем выданных по запросу данных. Включено = Циклическое. Выдаются все сохраненные данные для аналоговых тэгов за указанный промежуток времени. Выключено = Дельта. Выдаются только те аналоговые значения, которые изменились за указанный промежуток времени. [Подробную информацию и примеры см. в разделе "Циклический метод сбора данных" в главе 14, "Доступ к данным".](#)

Допуск Выполняет правила допуска для *общей* группы.

См. также `ww_SetDiscreteCyclic`

ww_SetContinuous

Уточняет, следует ли обновлять клиент в соответствии с текущим значением запрошенного тэга (тэгов). По умолчанию эта хранимая процедура находится в положении ВЫКЛЮЧЕНО. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока значение не будет сброшено.

Синтаксис `ww_SetContinuous On|Off, [RefreshTime]`

Параметры *On|Off*

Если стоит параметр *On*, сервер будет поставлен в режим непрерывного запроса, благодаря чему следующий запрос для "живой" таблицы расширения будет периодически подаваться в ядро выборки данных и результаты, полученные методом исключения, будут выдаваться клиенту, пока клиент не отменит запрос. Если стоит параметр *Off*, то клиент автоматически ничего не будет получать.

RefreshTime

Интервал (в мс), после которого, если клиент не получил никаких изменений данных, результаты будут все равно высланы ему для поддержания соединения с клиентом. Некоторые клиенты обрывают связь с сервером, если никакие значения данных не получаются с этого сервера.

Примечания	Эта хранимая процедура работает только с "живыми" таблицами расширения (<i>AnalogLive</i> , <i>DiscreteLive</i>). Подробную информацию и примеры см. в разделе "Непрерывное получение "живых" данных" в главе 14, "Доступ к данным".
Допуск	Выполняет правила допуска для <i>общей</i> группы.

ww_SetDiscreteCyclic

Уточняет, получать ли дискретные данные циклично или методом исключения (дельта). По умолчанию эта хранимая процедура находится в положении ВЫКЛЮЧЕНО. При установке этого значения вручную, все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока значение не будет сброшено.

Синтаксис	<code>ww_SetDiscreteCyclic *****</code>
Параметры	<i>Режим</i> Определяет объем выдаваемых по запросу данных. On = Выдаются все сохраненные данные за указанный промежуток времени. Off = Выдаются только дискретные значения, изменившиеся за указанный промежуток времени. Подробную информацию с примерами см. в разделе "Дельта-метод" в главе 14, "Выборка данных".
Допуск	Выполняет правила допуска для <i>общей</i> группы.
См. также	<code>ww_SetAnalogCyclic</code> <code>ww_SetTimeDeadband</code> <code>ww_SetValueDeadband</code>

ww_SetDuration

Определяет смещение от даты окончания, которое используется в запросе стартовая дата. По умолчанию временной промежуток этой хранимой процедуры установлен на 1 час. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлено новое время.

Синтаксис	<code>ww_SetDuration (datepart, value)</code>
Параметры	<i>datepart</i> Параметр, указывающий единицу времени. Допустимые показатели даты те же, что и показатели даты в Microsoft SQL Server (например, hh или час; mi или минута; ss или секунда; ms или миллисекунда). <i>value</i> Значение смещения продолжительности.
Примечания	Стандартная дата окончания — это последняя метка даты/времени данных. Стартовая дата подсчитывается при вычитании значения смещения из значения стандартной даты окончания. По мере накопления данных результаты запроса, использующего эту хранимую процедуру, будут изменяться. (Дата окончания изменяется по мере получения данных.) Можно указать конкретную дату окончания, используя хранимую процедуру <code>ww_SetEndDate</code>.

Используя эту хранимую процедуру в союзе с такими, как **ww_SetEndDate**, **ww_SetRowCount** и **ww_SetResolution**, можно повысить эффект запросов и облегчить обработку полученных данных.

Подробную информацию с примерами см. в разделе "Глобальная установка стартовой и конечной дат" в главе 14, "Доступ к данным".

Допуск

Выполняет правила допуска для *общей* группы.

См. также

ww_SetEndDate

ww_SetEdgeDetection

Выдает значения, используемые для включения событий. Результаты, выдаваемые хранимой процедурой **ww_SetEdgeDetection**, основаны на том, соответствовали ли выданные при предыдущем запросе значения критериям WHERE. По умолчанию эта хранимая процедура не будет определять край. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет указан тип определения края.

Синтаксис

ww_SetEdgeDetection •••••

Параметры

Режим

Результаты определения края, выдаваемые хранимой процедурой. Правильными значениями являются *None*, *Leading*, *Trailing* и *Both*. Если указано значение *None*, то будут выданы все строки, соответствующие критериям WHERE; определения края не происходит. Если указан параметр *Leading*, то в набор результатов входят только те строки, которые первыми соответствовали критериям WHERE (выдана истина) после тех, которые этим критериям не соответствовали (выдана ложь). Если указан параметр *Trailing*, то в набор результатов входят только те, которые первыми не соответствовали критериям WHERE (выдана ложь) после тех, которые этим критериям соответствовали (выдана истина). Если указан параметр *Both*, то выдаются все строки.

Подробную информацию с примерами см. в разделе "Определение событий" в главе 14, "Доступ к данным".

Примечания

ww_SetEdgeDetection используется для сужения набора результатов, где критерии WHERE изменялись с ИСТИНЫ на ЛОЖЬ или наоборот. Абстрактная "линия", на которой происходит изменение называется *краем*. Чтобы использовать эту хранимую процедуру, должен быть выполнен запрос к базе данных. Определение края затем проводится по результатам запроса.

Допуск

Выполняет правила допуска для *общей* группы.

ww_SetEndDate

Устанавливает значение даты окончания обработки запроса любой таблицы хранения данных. По умолчанию эта хранимая процедура установлена на текущее время. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлено новое время.

Синтаксис `ww_SetEndDate thedate`

Параметры *thedata*

Запрос выдаст дату и время последней записи.

Примечания Использование этой хранимой процедуры в союзе с хранимыми процедурами **ww_SetDuration**, **ww_SetRowCount** и **ww_SetResolution** повышает эффективность запросов базы данных и облегчает обращение с полученными данными.

Подробную информацию с примерами см. в разделе "Глобальная установка стартовой и конечной дат" в главе 14, "Доступ к данным".

Допуск Выполняет правила допуска для *общей* группы.

См. также **ww_SetDuration**
ww_SetResolution
ww_SetRowCount

ww_SetResolution

Устанавливает разрешение, или интервал выборки, в миллисекундах для получения данных в любых аналоговых и дискретных таблицах IndustrialSQL Server.

Синтаксис `ww_SetResolution count`

Параметры *count*

Интервал в миллисекундах между выдачей строк по запросу базы данных при использовании этой хранимой процедуры.

Примечания Разрешение определяет скорость обработки запроса базы данных в данный промежуток времени. При использовании этой хранимой процедуры следует указать промежуток времени. По умолчанию время этой хранимой процедуры установлено на 1000 мс. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлено новое значение разрешения. Разрешение используется только при циклическом методе получения данных.

Подробную информацию с примерами см. в разделе "Установка разрешения" в главе 14, "Выборка данных".

Допуск Выполняет правила допуска для *общей* группы.

См. также **ww_SetDuration**
ww_SetEndDate
ww_SetRowCount

ww_SetRowCount

Подсчет строк определяет число строк, которые должны быть получены из любой архивной таблицы IndustrialSQL Server.

Синтаксис `ww_SetRowCount count`

Параметры `count`

Количество строк, которые должны быть выданы за определенный период времени.

Примечания Подсчет строк определяет количество строк, которые будут выданы по запросу базы данных. Система IndustrialSQL Server сохраняет данные в равные промежутки времени за определенный период времени. Интервал между сохранением данных автоматически подсчитывается системой IndustrialSQL Server. Применение подсчета строк зависит от того, используется ли циклическое или дельта-метод, и запрашивается ли широкая таблица. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлен новый счет строк.
 Подробную информацию с примерами см. в разделе "Подсчет строк" в главе 14, "Выборка данных".

Допуск Выполняет правила допуска для *общей* группы.

См. также `ww_SetDuration`
`ww_SetEndDate`
`ww_SetResolution`

ww_SetTimeDeadband

Управляет разрешением времени данных, полученных дельта-методом (исключения). Любые изменения, произошедшие за время мертвой зоны (в мс), выдаваться не будут. По умолчанию мертвая зона этой хранимой процедуры установлена на 0. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлена новая мертвая зона.

Синтаксис `ww_SetTimeDeadband TimeDeadband`

Параметры `TimeDeadband`

Минимальное время в мс между получаемыми значениями одиночного тэга.

Примечания Хранимая процедура `ww_SetTimeDeadband` может использоваться только во время выборки данных методом исключения. Любые значения, изменяющиеся более одного раза за время указанной мертвой зоны, будут игнорированы.
 Подробную информацию с примерами см. в разделе "Установка временной мертвой зоны для дельта-выборки" в главе 14, "Выборка данных".

Допуск Выполняет правила допуска для *общей* группы.

См. также `ww_SetDiscreteCyclic`
`ww_SetValueDeadband`

ww_SetValueDeadband

Управляет разрешением значения данных, полученных дельта-методом. Любые значения данных, которые изменялись меньше, чем указанная мертвая зона, выдаваться не будут. Это значение есть процент от полного масштаба тэга в единицах измерения. По умолчанию эта хранимая процедура устанавливается на мертвую зону 0. При установке этого значения вручную все запросы базы данных будут обрабатываться в соответствии с ним до тех пор, пока не будет установлена новая мертвая зона.

Синтаксис `ww_SetValueDeadband Deadband`

Параметры `Deadband`

Процент от полного масштаба тэга в единицах измерения. Все изменения значений, которые меньше этого процента, выдаваться не будут.

Примечания Хранимая процедура **ww_SetValueDeadband** используется только во время выборки данных дельта-методом.

Подробную информацию с примерами см. в разделе "Установка значения мертвой зоны для дельта-метода" в гл. 14, "Выборка данных".

Допуск Выполняет правила допуска для *общей* группы.

См. также `ww_SetDiscreteCyclic`
`ww_SetTimeDeadband`

ww_StringDetail

Выдает описание одного или нескольких указанных тэгов.

Синтаксис `ww_StingDetail TagList`

Параметры `TagList`

Список тэгов, разделенных запятой (.). Значение этого параметра может содержать до 8000 знаков.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы `Tag`

См. также `ww_DiscreteDetail`
`ww_AnalogDetail`

ww_UserAccessLevelSelect

Выдает уровень доступа для текущего пользователя. Уровни доступа: 1 = Неопределенный пользователь IndustrialSQL (например, *общий*); 2 = Пользователь (допуск wwUser); 3 = Опытный пользователь (допуск wwPowerUser); 3 = Администратор (допуск wwAdmin); 9999 = Владелец базы данных.

Синтаксис `ww_UserAccessLevelSelect`

Примечания Значение уровня доступа соответствуют таблице *UserDetail*. Таблица *UserDetail* получает правильные значения для столбца *AccessLevel* во время установки. Эта хранимая процедура используется клиентом Wonderware при соединении с IndustrialSQL Server. Некоторые приложения, такие как InSQL Configure, проверяют уровень доступа пользователя. Если пользователь не имеет допустимого уровня доступа, программа не запустится.

Допуск Выполняет правила допуска для *общей* группы.

Расширенные хранимые процедуры

Расширенная хранимая процедура — это функция SQL Server для динамической загрузки и выполнения функций библиотек динамической связи (DLL). Расширенная хранимая процедура может выполнять определенное действие вне SQL Server, результаты которого будут сохранены в SQL Server. К расширенным хранимым процедурам, поставляемым с IndustrialSQL Server, относятся:

- xp_proclist
- xp_AnalogHistory
- xp_AnalogHistoryDelta
- xp_AnalogWideHistory
- xp_AnalogWideHistoryDelta
- xp_DiscreteHistory
- xp_DiscreteHistoryDelta
- xp_DiscreteWideHistory
- xp_DiscreteWideHistoryDelta
- xp_SetStorageTimeDeadband
- xp_SetStorageValueDeadband
- xp_NewHistoryBlock
- xp_DiskCopy

Примечание. Расширенные хранимые процедуры должны выполняться из основной базы данных.

Каждая из этих процедур обращается к одной и той же библиотеке динамической связи, XPPDS.DLL. Выполняет правила допуска для всех расширенных хранимых процедур *общей* группы.

Внимание! Эти расширенные хранимые процедуры различают регистр!

Архивные расширенные хранимые процедуры позволяют запрашивать архивные данные из хранимых процедур, созданных в MS SQL Server. Любая функция SQL Server, способная обращаться к хранимой процедуре, может обращаться к любой из этих хранимых процедур.

Например, чтобы запросить данные из таблицы *AnalogHistory*, расширенная хранимая процедура **xp_AnalogHistory** обратится к XPPDS.DLL. Этот DLL подаст запрос в таблицу *AnalogHistory* (что поддерживается только в IndustrialSQL Server) и доставит результаты в MS SQL Server.

xp_proclist

Выдает детали синтаксиса для всех расширенных хранимых процедур IndustrialSQL Server, которые присутствуют на Microsoft SQL Server.

Синтаксис xp_proclist

Допуск Выполняет правила допуска для *общей* группы.

xp_AnalogHistory

Выдает данные из таблицы *AnalogHistory* в IndustrialSQL Server. Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы Microsoft SQL Server.

Синтаксис xp_AnalogHistory StartTime, EndTime, Resolution, var1[, var2, ... varN]

Параметры *StartTime*

Метка стартовых даты/времени запрашиваемых данных.

EndTime

Метка конечных даты/времени запрашиваемых данных.

Resolution

Разрешение или скорость получения данных, в миллисекундах. Система выдаст данные, сохраненные за указанный период времени с интервалом, определяемым разрешением. Например, если указать разрешение в 5000 мс, система запросит все данные за период времени, но выдаст только те значения, которые были получены с интервалом в 5000 миллисекунд, начиная от стартовой даты и заканчивая датой окончания.

Примечание Разрешение указывается в миллисекундах (мс). Дата/время должны указываться в следующем формате: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.

Допуск Выполняет правила допуска для *общей* группы.

Таблицы *AnalogHistory*

Пример Этот пример выдает значения данных для тэга 'MeterStatus', сохраненного в таблице *AnalogHistory* между 8:30 и 9:00 a.m. 1 января 1999 г. Эти данные имеют разрешение в 1000 мс.

```
xp_AnalogHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, MeterStatus
```

См. также xp_AnalogWideHistory
xp_DiscreteHistory
xp_DiscreteWideHistory

xp_AnalogHistoryDelta

Выдает строку из таблицы *AnalogHistory* для каждого момента изменения аналогового значения (дельта-метод). Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы Microsoft SQL Server.

Синтаксис	<code>xp_AnalogHistoryDelta StartTime, EndTime, MaxRowCount, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>MaxRowCount</i> Максимальное количество выдаваемых строк.
Замечания:	Дата/время должны указываться в следующем формате: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>AnalogHistory</i>
Пример	Этот пример выдает дельта-значения данных для тэга 'MeterStatus', сохраненного в таблице <i>AnalogHistory</i> между 8:30 и 9:00 а.м. 1 января 1999 г. Максимальное число строк равняется 1000. <pre>xp_AnalogHistoryDelta '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, MeterStatus</pre>
См. также	<code>xp_AnalogWideHistoryDelta</code> <code>xp_DiscreteHistoryDelta</code> <code>xp_DiscreteWideHistoryDelta</code>

xp_AnalogWideHistory

Выдает данные из таблицы *AnalogWideHistory* в IndustrialSQL Server. Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы Microsoft SQL Server.

Синтаксис	<code>xp_AnalogWideHistory StartTime, EndTime, Resolution, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>Resolution</i> Разрешение или скорость получения данных в миллисекундах. Система выдаст данные, сохраненные за указанный период времени с интервалом, определяемым разрешением. Например, если вы указать разрешение в 5000 мс, система запросит все данные за период времени, но выдаст только те значения, которые были получены с интервалом в 5000 миллисекунд, начиная от стартовой даты и заканчивая датой окончания.
Примечание	Разрешение указывается в миллисекундах (мс). Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>AnalogWideHistory</i>
Пример	Этот пример выдает значения данных для тэгов 'MeterStatus', 'TankFlow' и 'Temp1', сохраненных в таблице <i>AnalogWideHistory</i> между 8:30 и 9:00 a.m. 1 января 1999 г. Эти данные имеют разрешение в 1000 мс. <pre>xp_AnalogWideHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, MeterStatus, TankFlow, Temp1</pre>
См. также	<code>xp_AnalogHistory</code> <code>xp_DiscreteHistory</code> <code>xp_DiscreteWideHistory</code>

xp_AnalogWideHistoryDelta

Выдает строку из таблицы *AnalogWideHistory* для каждого момента изменения аналогового значения (дельта-метод). Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы Microsoft SQL Server.

Синтаксис	<code>xp_AnalogWideHistoryDelta StartTime, EndTime, MaxRowCount, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>MaxRowCount</i> Максимальное количество выдаваемых строк.
Замечания:	Дата/время должны указываться так: MM/DD/YY or MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>AnalogWideHistory</i>
Пример	Этот пример выдает дельта-значения данных для тэгов 'MeterStatus', 'TankFlow' и 'Temp1', сохраненных в таблице <i>AnalogWideHistory</i> между 8:30 и 9:00 a.m. 1 января 1999 г. Максимальное число строк равняется 1000. <pre>xp_AnalogWideHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, MeterStatus, TankFlow, Temp1</pre>
См. также	<code>xp_AnalogHistoryDelta</code> <code>xp_DiscreteHistoryDelta</code> <code>xp_DiscreteWideHistoryDelta</code>

xp_DiscreteHistory

Выдает данные из таблицы *DiscreteHistory* IndustrialSQL Server. Позволяет получать данные, хранимые в IndustrialSQL Server, при помощи запросов Microsoft SQL Server.

Синтаксис	<code>xp_DiscreteHistory StartTime, EndTime, Resolution, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>Resolution</i> Разрешение или скорость получения данных в миллисекундах. Система выдаст данные, сохраненные за указанный период времени с интервалом, определяемым разрешением. Например, если указать разрешение в 5000 мс, система запросит все данные за период времени, но выдаст только те значения, которые были получены с интервалом в 5000 миллисекунд, начиная от стартовой даты и заканчивая датой окончания.
Примечания	Разрешение указывается в миллисекундах (мс). Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>DiscreteHistory</i>
Пример	Этот пример выдает значения данных для тэга 'PumpOn', сохраненного в таблице <i>DiscreteHistory</i> между 8:30 и 9:00 a.m. 1 января 1999 г. Эти данные имеют разрешение 1000 мс. <pre>xp_DiscreteHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, PumpOn</pre>
См. также	<code>xp_AnalogHistory</code> <code>xp_AnalogWideHistory</code> <code>xp_DiscreteWideHistory</code>

xp_DiscreteHistoryDelta

Выдает строку из таблицы *DiscreteHistory* для каждого момента изменения аналогового значения (дельта-метод). Позволяет получать данные, хранящиеся в IndustrialSQL Server используя запросы Microsoft SQL Server.

Синтаксис	<code>xp_DiscreteHistory StartTime, EndTime, MaxRowCount, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>MaxRowCount</i> Максимальное количество выдаваемых строк.
Замечания:	Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>DiscreteHistory</i>
Пример	Этот пример выдает значения данных для тэга 'PumpOn', сохраненного в таблице <i>DiscreteHistory</i> между 8:30 и 9:00 а.м. 1 января 1999 г. Максимальное число строк равняется 1000. <pre>xp_DiscreteHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, PumpOn</pre>
См. также	<code>xp_AnalogHistoryDelta</code> <code>xp_AnalogWideHistoryDelta</code> <code>xp_DiscreteWideHistoryDelta</code>

xp_DiscreteWideHistory

Выдает данные из таблицы *DiscreteWideHistory* IndustrialSQL Server. Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы MS SQL Server.

Синтаксис	<code>xp_DiscreteWideHistory StartTime, EndTime, Resolution, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>Resolution</i> Разрешение или скорость получения данных в миллисекундах. Система выдаст данные, сохраненные за указанный период времени с интервалом, определяемым разрешением. Например, если указать разрешение в 5000 мс, система запросит все данные за период времени, но выдаст только те значения, которые были получены с интервалом в 5000 миллисекунд, начиная от стартовой даты и заканчивая датой окончания.
Примечание	Разрешение указывается в миллисекундах (мс). Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>DiscreteWideHistory</i>
Пример	Этот пример выдает значения данных для тэгов 'PumpOn', 'Valve1' и 'Valve2', сохраненных в таблице <i>DiscreteWideHistory</i> между 8:30 и 9:00 a.m. 1 января 1999 г. Эти данные имеют разрешение в 1000 мс. <pre>xp_DiscreteWideHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, PumpOn, Valve1, Valve2</pre>
См. также	<code>xp_AnalogHistory</code> <code>xp_AnalogWideHistory</code> <code>xp_DiscreteHistory</code>

xp_DiscreteWideHistoryDelta

Выдает строку из таблицы *DiscreteWideHistory* для каждого момента изменения аналогового значения (дельта-метод). Позволяет получать данные, хранящиеся в IndustrialSQL Server, используя запросы Microsoft SQL Server.

Синтаксис	<code>xp_DiscreteWideHistory StartTime, EndTime, MaxRowCount, var1[, var2, ... varN]</code>
Параметры	<i>StartTime</i> Метка стартовых даты/времени для запрашиваемых данных. <i>EndTime</i> Метка конечных даты/времени запрашиваемых данных. <i>MaxRowCount</i> Максимальное количество выдаваемых строк.
Замечания:	Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Таблицы	<i>DiscreteWideHistory</i>
Пример	Этот пример выдает значения данных для тэгов 'PumpOn', 'Valve1' и 'Valve2', сохраненных в таблице <i>DiscreteWideHistory</i> между 8:30 и 9:00 a.m. 1 января 1999 г. Максимальное число строк равняется 1000. <pre>xp_DiscreteWideHistory '01/01/1999 08:30:00', '01/01/1999 09:00:00', 1000, PumpOn, Valve1, Valve2</pre>
См. также	<code>xp_AnalogHistoryDelta</code> <code>xp_AnalogWideHistoryDelta</code> <code>xp_DiscreteHistoryDelta</code>

xp_DiskCopy

Копирует архивные блоки или подмножество архивных блоков в архивный путь "сжатия" (постоянный путь).

Синтаксис

xp_DiskCopy StartTime, EndTime, Description

Параметры

StartTime

Метка стартовых даты/времени для запрашиваемых данных.

EndTime

Метка конечных даты/времени запрашиваемых данных.

Description

Описание процедуры копирования данных.

Замечания:

Дата/время должны указываться так: MM/DD/YY или MM/DD/YYYY HH:MM:SS.MS.

Допуск

Выполняет правила допуска для группы *wwAdministrators*.

Пример

Например:

```
xp_DiskCopy <02/20/1999 13:10:00>, <02/20/1999  
14:00:00>, <Данные по цеху за 02/20/1999>
```

xp_NewHistoryBlock

Предотвращает запись данных в текущий архивный блок и создает новый. Этот процесс занимает около 10 минут, но потери данных не происходит. Система действует так же, как в случае с запланированной сменой блока.

Синтаксис

xp_NewHistoryBlock

Примечания

IndustrialSQL Server сохраняет полученные данные на диск в подкаталоги, называемые архивными блоками. Архивные блоки создаются при запуске системы, или по истечении заранее определенного интервала времени. Эта расширенная хранимая процедура обеспечивает принудительное создание новых архивных блоков.

После выполнения этой процедуры в файл .ERR записывается сообщение, как для обычной смены блока. Вместе с тем, при выполнении этой процедуры вручную с помощью SQL Server Query Analyzer сообщение появится в разделе Results, если процедура выполнена успешно.

Допуск

Выполняет правила допуска для *системных администраторов*.

Таблицы

HistoryBlock

xp_SetStorageTimeDeadband

Устанавливает временную зону хранения для одного или нескольких тэгов для сохранения связи с клиентом.

Синтаксис	<code>xp_SetStorageTimeDeadband Time, var1[,var2,... varN]</code>
Параметры	<i>Time</i> Минимальное время (в мс) между сохранением значений для одиночного тэга. Тэг, значение которого изменялось более одного раза за этот период, будет игнорироваться алгоритмом дельта-сохранения. Время равное 0 (по умолчанию) показывает, что система будет сохранять значение тэга при каждом его изменении.
Примечания	Расширенная хранимая процедура xp_SetStorageTimeDeadband является функцией сервера, которая будет изменять правила дельта-сохранения для указанного тэга. Это не приводит к обновлению базы данных и действительно до перезапуска сервера.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Пример	Этот пример устанавливает значение временной мертвой зоны хранения на 2000 миллисекунд для аналогового тэга "BoilerTag": <pre>xp_SetStorageTimeDeadband 2000, BoilerTag</pre>

xp_SetStorageValueDeadband

Устанавливает мертвую зону значения хранения для одного или нескольких тэгов для сохранения связи с клиентом.

Синтаксис	<code>xp_SetStorageValueDeadband Deadband, var1[,var2,... varN]</code>
Параметры	<i>Deadband</i> Процент от масштаба единицы измерения, на который должен измениться тэг для сохранения его значения. Применяется только при дельта-сохранении. Если этот атрибут получает значение 0 или NULL, мертвая зона не используется.
Примечания	Расширенная хранимая процедура xp_SetStorageTimeDeadband является функцией сервера, которая изменяет правила дельта-сохранения для указанного тэга. Это не приводит к обновлению базу данных и действительно до перезапуска сервера.
Допуск	Выполняет правила допуска для <i>общей</i> группы.
Пример	Этот пример устанавливает мертвую зону значения хранения на 5 процентов от диапазона единицы измерения для аналогового тэга "BoilerTag": <pre>xp_SetStorageValueDeadband 5, BoilerTag</pre>

Создание хранимых процедур

Вы можете создать собственные хранимые процедуры для работы с IndustrialSQL Server. Все имена этих процедур будут сохранены в базе данных *Runtime*. Текст хранимой процедуры будет сохранен в Microsoft SQL Server и получен при запуске (от процедур, созданных при предыдущем сеансе работы), как и время создания (от процедур, созданных во время текущего сеанса). Временные процедуры не поддерживаются. Никакие параметры не допускаются.

Как и в случае поддержки Microsoft SQL Server, динамически хранимые процедуры поддерживаются в IndustrialSQL Server таким образом, что при определении хранимой процедуры есть возможность создавать ее только в текущей базе данных, при этом выражение CREATE PROCEDURE нельзя совмещать с другими выражениями SQL в одном пакетном файле.

Создание собственных хранимых процедур весьма полезно, если требуется выполнить некоторые типы запросов через соединение ODBC. Системе IndustrialSQL Server требуется особая настройка ODBC, если вы не создаете хранимую процедуру для обработки запроса.

Более подробную информацию о настройке ODBC для IndustrialSQL Server можно найти в *Руководстве администратора системы FactorySuite*.

Например, следующий запрос создает хранимую процедуру, выдающую метку даты/времени и значение тэга 'ReactLevel' за последние 15 минут:

```
CREATE PROCEDURE MyProc
AS
SELECT DateTime, TagName, Value
FROM AnalogHistory
WHERE TagName = 'ReactLevel'
AND DateTime >= DATEADD(mi, -15, GETDATE())
AND DateTime <= GETDATE()
```

Представления данных

В этой главе описаны представления данных, входящие в состав IndustrialSQL Server. Эти представления данных поставляются для облегчения подачи запросов в таблицы архивов событий и сводок. Также в этой главе рассматриваются представления данных, используемые для доступа к данным через поставщика данных OLE DB.

Содержание

- Представления данных
- Представления данных системы управления событиями
- Представления для удаленных таблиц поставщика данных OLE DB

Представления данных

Представление данных — это логический способ просмотра данных в одной или нескольких таблицах базы. Представление данных является "виртуальной" таблицей; в действительности она не присутствует в базе данных. Представление содержит указатели на существующие в базе данных таблицы. Представления позволяют составлять подмножество информации, хранимой в одной или нескольких таблицах, не затрагивая остальную информацию. Представления входят в состав стандартных функций сервера SQL.

Представления данных системы управления событиями

Для облегчения подачи запросов к данным таблиц архивов событий и таблиц сводок предусмотрен ряд представлений данных. Запросы для этих представлений системы управления событиями выглядят как обычные физические таблицы.

v_EventSnapshot

Выдает одну строку для каждого события, хранящегося в архиве.

Атрибут	Тип данных	Объяснение
Event	varchar(33)	Имя тэга события, для которого сохраняется архив.
EventTime	datetime	Метка времени получения архивных данных. Это истинное время, когда событие в действительности произошло.
DetectionTime	datetime	Метка времени, когда событие было зафиксировано системой управления событиями.
Edge	tinyint	Сохраняет "край", на котором было зафиксировано событие. 0 = Trailing; 1 = Leading (по умолчанию); 2 = Both; 3 = Нет; 4 = Детектор времени; 5 = Внешний детектор. Представление данных выдает текстовое значение.
TagName	varchar(33)	Имя тэга, связанного со "отснятием" события. Тэг будет включен в снимок.
Value	float(25)	Значение тэга в момент даты/времени происхождения события. Указывается в единицах измерения.

(продолжение)

Атрибут	Тип данных	Объяснение
Quality	tinyint	Основной показатель качества, связанный со значением снимка. Подробнее см. в Приложении Б, "Качество данных".
QualityDetail	int	Детали основного показателя качества. Подробнее см. в Приложении Б, «Качество данных».

Синтаксис представления

Примечание. Синтаксис представления приводится в качестве справки. IndustrialSQL Server поставляется с уже настроенным данным представлением.

```
CREATE VIEW v_EventSnapshot
AS
SELECT eh.TagName as Event, eh.DateTime as EventTime,
eh.DetectDateTime as DetectionTime,
    Edge =
        CASE eh.Edge
            WHEN 1 THEN 'Leading'
            WHEN 0 THEN 'Trailing'
            WHEN 2 THEN 'Both'
            WHEN 3 THEN 'None'
            WHEN 4 THEN 'Time'
            WHEN 5 THEN 'External'
            ELSE '-'
        END,
    sst.TagName as TagName, ass.Value as Value, ass.Quality as
    Quality,
    ass.QualityDetail as QualityDetail
FROM EventHistory eh LEFT OUTER JOIN AnalogSnapshot ass on
eh.EventLogKey = ass.EventLogKey
INNER JOIN SnapshotTag sst on sst.SnapshotTagKey =
ass.SnapshotTagKey
UNION
SELECT eh.TagName as Event, eh.DateTime as EventTime,
eh.DetectDateTime as DetectionTime,
    Edge =
        CASE eh.Edge
            WHEN 1 THEN 'Leading'
            WHEN 0 THEN 'Trailing'
            WHEN 2 THEN 'Both'
            WHEN 3 THEN 'None'
            WHEN 4 THEN 'Time'
            WHEN 5 THEN 'External'
            ELSE '-'
        END,
    sst.TagName as TagName, ass.Value as Value, ass.Quality as
    Quality,
    ass.QualityDetail as QualityDetail
```

```
FROM EventHistory eh LEFT OUTER JOIN DiscreteSnapshot ass on
eh.EventLogKey = ass.EventLogKey

INNER JOIN SnapshotTag sst on sst.SnapshotTagKey =
ass.SnapshotTagKey

UNION

SELECT eh.TagName as Event, eh.DateTime as EventTime,
eh.DetectDateTime as DetectionTime,
    Edge =
        CASE eh.Edge
            WHEN 1 THEN 'Leading'
            WHEN 0 THEN 'Trailing'
            WHEN 2 THEN 'Both'
            WHEN 3 THEN 'None'
            WHEN 4 THEN 'Time'
            WHEN 5 THEN 'External'
        ELSE '-'
    END,
    '-' as TagName, NULL as Value, NULL as Quality,
    NULL as QualityDetail
FROM EventHistory eh, EventTag et
WHERE eh.TagName = et.TagName
AND et.ActionTypeKey <> 3
```

Примечание. Когда событие не связано со снимком, столбец *TagName* установлен на значение '-', а столбцы *Value*, *Quality* и *QualityDetail* установлены на NULL.

v_SummaryData

Выдает одну строку для каждой операции создания сводки тэга.

Атрибут	Тип данных	Объяснение
TagName	varchar(33)	Имя тэга сводки.
CalcType	Типы Подсчетов: char(3)	Тип проведенного подсчета: SUM, MAX, MIN, или AVG.
SummaryDate	datetime	Дата, связанная с результатами подсчета. Это дата начала или окончания подсчета.
Value	float(25)	Значение сводки
Quality	tinyint	Показатель качества результата. Это высшее значение качества необработанных данных, из которых получен результат.
Duration	действительный	Период (в секундах) за который проведен подсчет.
Resolution	int	Разрешение выборки данных (в секундах), с которым проведен подсчет.
TimeStamp	tinyint	Определяет, какая метка времени использована для результата подсчета. 0 = Начало; 1 = Окончание
EventTag	varchar(33)	Имя тэга события, связанного с подсчетом сводки.

Синтаксис представления

Примечание. Синтаксис представления приведен в качестве справки. IndustrialSQL Server поставляется с уже настроенным данным представлением.

```
CREATE VIEW v_SummaryData
AS
SELECT stl.TagName, sh.SumDateCalcType as CalcType,
sh.SummaryDate, sd.Value, sd.Quality, sh.SumDateDuration as
Duration, sh.SumDateResolution as Resolution,
sh.SumDateTimeStamp as TimeStamp, so.TagName as EventTag
FROM SummaryTagList stl
INNER JOIN SummaryData sd on stl.SumVarKey = sd.SumVarKey
INNER JOIN SummaryHistory sh on sd.LogKey = sh.LogKey
INNER JOIN SummaryOperation so on sh.OperationKey =
so.OperationKey
INNER JOIN EventTag et on so.TagName = et.TagName
```

Представления для удаленных таблиц поставщика данных OLE DB

Созданы представления данных для удаленных таблиц поставщика OLE DB для упрощения подачи запросов к этим таблицам. Обычно при запросе к удаленным таблицам в каждом запросе требуется указать полное описание запрашиваемой таблицы, что выглядит следующим образом:

linked_server.catalog.schema.objectname. При использовании представления удаленной таблицы достаточно лишь указать имя представления без его длинного описания.

Подробнее о поставщике данных IndustrialSQL OLE DB см. в главе 14, "Выборка данных".

Все следующие представления имеют табличную структуру, аналогичную удаленным таблицам, по которым они созданы.

- *v_AnalogHistory*
- *v_AnalogLive*
- *v_DiscreteHistory*
- *v_DiscreteLive*
- *v_HistoryBlock*
- *v_StringLive*

Подробнее см. в главе 11, "Удаленные таблицы OLE DB".

Примечание. Удаленные таблицы поставщика данных OLE DB — это не то же, что таблицы расширения в базе данных *Runtime*.

Синтаксис представлений

Синтаксис представления приведен в качестве справки. IndustrialSQL Server поставляется с уже настроенными этими представлениями.

Для связи между аналоговыми и дискретными таблицами аналоговые представления ссылаются на связанный с OLE DB сервер "INSQL", тогда как дискретные представления ссылаются на связанный с OLE DB сервер "INSQLD".

Подробнее см. в главе 14, "Выборка данных".

Примечание. Чтобы изменить какое-либо из этих представлений, следует войти в систему IndustrialSQL Server как пользователь *wwdbo*. Допуск для всех представлений OLE DB желательно установить на *общую* группу.

v_AnalogHistory

```
CREATE VIEW v_AnalogHistory
AS
SELECT * FROM INSQL.Runtime.dbo.AnalogHistory
```

v_AnalogLive

```
CREATE VIEW v_AnalogLive
AS
SELECT * FROM INSQL.Runtime.dbo.AnalogLive
```

v_DiscreteHistory

```
CREATE VIEW v_DiscreteHistory
AS
SELECT * FROM INSQLD.Runtime.dbo.DiscreteHistory
```

v_DiscreteLive

```
CREATE VIEW v_DiscreteLive
AS
SELECT * FROM INSQLD.Runtime.dbo.DiscreteLive
```

v_StringLive

```
CREATE VIEW v_StringLive
AS
SELECT * FROM INSQL.Runtime.dbo.StringLive
```

v_HistoryBlock

```
CREATE VIEW v_HistoryBlock
AS
SELECT * FROM INSQL.Runtime.dbo.HistoryBlock
```

Выборка данных

В этой главе описаны способы выборки данных с использованием языка SQL. IndustrialSQL Server поддерживает как расширения Transact-SQL, поддерживаемые Microsoft SQL Server, так и расширения Industrial-SQL.

Описан синтаксис запроса для выборки данных как с сервера базы данных IndustrialSQL ODS, так и от поставщика INSQL OLE DB. В этой главе приводятся примеры запросов стандартных типов и запросов, использующих расширение временного домена Industrial-SQL. Для каждого типа запроса описан синтаксис ODS и OLE DB и, где возможно, предоставлены типичные результаты данных.

Содержание

- Синтаксис запроса к серверу базы данных IndustrialSQL ODS
- Синтаксис запросов для поставщика данных INSQL OLE DB
- Типичные запросы IndustrialSQL Server
- Использование расширений временного домена Industrial-SQL
- Неподдерживаемый синтаксис SQL для поставщика данных INSQL OLE DB

Синтаксис запроса к серверу базы данных IndustrialSQL ODS

Таблицы расширения есть логические таблицы, показывающие данные из архивных блоков при использовании служб Open Data Services. Данными в таблицах расширения можно манипулировать при помощи кода Transact-SQL, а также специальными расширениями SQL, которые поставляются с IndustrialSQL Server и называются Industrial-SQL. Запросы к таблицам расширения управляются сервером базы данных IndustrialSQL Server ODS (PDSSRV.EXE). Сервис NT для этой базы данных называется InSQL DbServer.

Расширения временного домена поддерживают как выражения SET, так и хранимые процедуры. После установки расширения временного домена оно становится доступным для всех запросов, пока установки не изменятся.

Следующий пример использует расширение разрешения:

```
SET wwResolution 2000

SELECT DateTime, Value FROM AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND Value >= 50
```

Ограничения для запросов через сервер БД IndustrialSQL ODS

Поскольку сервер базы данных IndustrialSQL ODS полностью отвечает за разбор, декодирование и выполнение запросов к таблицам расширения, весь набор выражений Transact-SQL не поддерживается. Запросы к таблицам расширения с неподдерживаемым синтаксисом выдадут клиенту сообщение об ошибке.

При подаче запроса через сервер базы данных IndustrialSQL ODS (PDSSRV.EXE) действуют следующие ограничения:

- IndustrialSQL Server не поддерживает математические или текстовые функции для таблиц расширения (*AnalogHistory*, *AnalogLive*, *AnalogWideHistory*, *DiscreteHistory*, *DiscreteLive*, *DiscreteWideHistory* и *HistoryBlock*).
- Поддерживается только синтаксис Transact SQL SELECT. Например, выражения GROUP BY и ORDER BY при работе с таблицами расширения не используются.
- Объявленные переменные не поддерживаются.

Синтаксис запросов для поставщика данных INSQL OLE DB

Получение данных через OLE DB происходит одним из четырех способов: используя имя связанного сервера поставщика данных INSQL OLE DB в выражении SELECT, используя одно из представлений в выражении SELECT, используя OPENQUERY и OPENROWSET.

Использование имени связанного сервера

Имя связанного сервера — это имя, под которым поставщик данных INSQL OLE DB известен системе Microsoft SQL Server. Microsoft SQL Server использует имя связанного сервера для идентификации поставщика данных OLE DB и источника данных. Для того, чтобы запрос поступал к поставщику данных INSQL OLE DB, следует использовать имя связанного сервера в системе наименования, состоящей из четырех частей. Эта система позволяет уточнять, из какого сектора хранилища данных вы собираетесь получать данные.

Например, согласно следующему запросу, данные должны быть получены из удаленной таблицы *AnalogHistory* через поставщика данных INSQL OLE DB:

```
SELECT * FROM INSQL.Runtime.dbo.AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= 'Nov 25 1998 00:59'
AND DateTime <= 'Nov 25 1998 01:59'
```

Система наименований из четырех частей описана в следующей таблице:

Название части	Описание
<i>linked_server</i>	Имя связанного сервера. По умолчанию: "INSQL".
<i>catalog</i>	Каталог источника данных OLE DB, где находится объект, из которого запрашиваются данные. Для баз данных Microsoft SQL Server это имя базы данных. Для использования поставщика данных INSQL OLE DB имя каталога всегда "Runtime".
<i>schema</i>	Схема в каталоге, где хранится объект. Для баз данных Microsoft SQL Server это имя идентификатора регистрации для доступа к данным. Для использования поставщика данных InSQL OLE DB имя каталога всегда должно быть "dbo".
<i>object_name</i>	Объект данных, который поставщик данных OLE DB может представлять как набор строк. Для поставщика данных InSQL OLE DB имя объекта является именем удаленной таблицы, из которой должны быть получены данные. Примером может служить таблица <i>AnalogHistory</i> . Эта часть тоже может быть запросом, который заставит поставщика данных OLE DB отобразить набор строк.

Использование представления поставщика данных InSQL OLE DB

Представления данных Microsoft SQL Server также служат для доступа к каждой удаленной таблице, избавляя от необходимости вводить в запрос наименования из четырех частей. Эти представления называются *v_ProviderTableName*.

Например:

```
SELECT * FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
  AND DateTime >= 'Nov 25 1998 00:59'
  AND DateTime <= 'Nov 25 1998 01:59'
```

Подробнее см. в главе 13, "Представления данных".

Использование функции OPENQUERY

Для доступа к строкам из источника данных OLE DB можно также использовать имя связанного сервера в OPENQUERY. Эти строки можно проверить как таблицу в выражениях Transact-SQL. Например:

```
SELECT * FROM OpenQuery(INSQL, 'SELECT * FROM AnalogHistory
  WHERE TagName = "SysTimeSec"
  AND DateTime >= "Nov 25 1998 00:59"
  AND DateTime <= "Nov 25 1998 01:59"
  ')
```

Часть этих выражений, относящаяся к OPENQUERY, распознается сервером SQL как таблица и может также использоваться при объединении, просмотре и использовании хранимых процедур.

Использование функции OPENROWSET

Имя связанного сервера может служить параметром ввода для функции OPENROWSET. Функция OPENROWSET отправляет поставщику данных OLE DB команду на выполнение. Выданный набор строк может использоваться как ссылка на таблицу или представление для выражений Transact-SQL.

Например:

```
SELECT * FROM OpenRowset('INSQL',' ',
  "SELECT DateTime, Quality, QualityDetail, Value
  FROM AnalogHistory
  WHERE TagName in ('SystimeSec')
  AND DateTime >= 'Nov 25 1998 00:59 '
  AND DateTime <= 'Nov 25 1998 01:59'
  ")
```

Использование расширений реального времени IndustrialSQL

Необходимо указать все параметры для каждого запроса, который будет выполняться через поставщика данных OLE DB. В случае с поставщиком данных InSQL OLE DB это означает, что при каждой подаче запроса следует указывать все параметры расширений реального времени (число строк, разрешение, определение края и т.д.). Следует помнить, что для запросов, выполняемых через ODS (сервер базы данных IndustrialSQL PDSSRV.EXE), каждый указанный параметр действителен, пока не изменен; таким образом расширения переходят от одного запроса к другому. Поставщики данных OLE DB не поддерживают такую непрерывность.

В случае с OLE DB указать параметры расширения реального времени можно с помощью "виртуальных" столбцов в удаленных таблицах. Добавление этих столбцов составляет единственную разницу между таблицами расширения, входящими в состав сервера базы данных IndustrialSQL, и удаленными таблицами поставщика данных InSQL OLE DB. [Подробнее см. в главе 11, "Удаленные таблицы OLE DB".](#)

При запросе удаленных таблиц с использованием синтаксиса OLE DB следует просто указать расширение в выражении WHERE. Например:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND Value >= 50
      AND wwResolution = 2000
```

Использование хранимых процедур в запросах OLE DB

Любая нормальная хранимая процедура Microsoft SQL Server может использовать таблицы, отображаемые поставщиком данных INSQL OLE DB. Хранимые процедуры могут использовать только допустимый синтаксис OLE DB сервера SQL для доступа к архивным данным IndustrialSQL Server.

Другими словами, хранимые процедуры используют запросы из четырех частей, функции OPENQUERY и OPENROWSET, курсоры, запросы с параметрами и представления. Хранимые процедуры используются для хранения сложных группировок и прочих операций, что облегчает их повторное использование приложениями и пользователями.

Вместе с тем, расширения временного домена IndustrialSQL Server, считающиеся хранимыми процедурами, такие как **ww_SetRowCount** или **ww_SetResolution**, не поддерживаются поставщиком данных OLE DB из-за того, что для указания расширений временного домена используются "виртуальные столбцы".

Ограничения для запросов через OLE DB

При запросе, подаваемом через поставщика данных InSQL OLE DB, действуют следующие ограничения. Обычно эти ограничения связаны с ограничениями текущей версии поставщика данных OLE DB Microsoft SQL Server. Большинство из этих ограничений будут устранены в следующих версиях сервера SQL:

- Не поддерживаются функции агрегации даты/времени (такие как *getdate()*, *dateadd()* и т.п.). В настоящее время решение состоит в том, чтобы указывать параметры запроса путем помещения агрегаций даты/времени в локальные переменные. Microsoft SQL Server ставит задачу подачи этих переменных поставщику данных InSQL OLE DB на первое место.
- "Непостоянность" расширений временного домена. В отличие от ODS, поставщик данных OLE DB не имеет понятия о контексте клиента, а, следовательно, параметры от запроса к запросу не сохраняются в течение одного соединения (поставщик данных OLE DB совершенно не запоминает состояния). Это значит, что значения расширения временного домена IndustrialSQL (например, счет строк) следует указывать при каждом выполнении запроса. Расширение указывается как столбец в выражении WHERE.
- Отмена выполнения запросов, возвращающих результаты чрезмерного объема, может сработать не сразу.
- Поддержка группировок за простыми группировками между таблицами сервера SQL и удаленными таблицами OLE DB IndustrialSQL Server. Группировки обычно требуют использования синтаксиса INNER REMOTE JOIN. Группировки между двумя таблицами поставщика данных InSQL OLE DB требуют использования двух различных псевдонимов.

Подробнее см. далее в разделе "Неподдерживаемый синтаксис SQL для поставщика данных InSQL OLE DB".

Типичные запросы IndustrialSQL Server

В этом разделе представлены некоторые часто используемые запросы на получение данных с IndustrialSQL Server. Приведенные примеры таблиц не являются исчерпывающим списком возможных запросов баз данных, но дают некоторое представление о видах запросов.

Примечание. Система Microsoft SQL Server может быть настроена на распознавание регистра. IndustrialSQL Server также может распознавать регистр. При написании запросов используйте правильный регистр.

Выбор значений столбцов для одиночной таблицы

Следующий запрос выдает требуемые детали для аналогового тэга "ReactLevel".

Основной запрос

```
SELECT TagName, MinEU, MaxEU, ValueDeadband
FROM AnalogTag
WHERE TagName = 'ReactLevel'
```

Результаты

Результаты:

TagName	MinEU	MaxEU	ValueDeadband
ReactLevel	0.0	3000.0	0.0

(задействована 1 строка)

Использование выражения LIKE

Следующие запросы выдают все аналоговые тэги системы, которые начинаются с "sys". (Значок % есть символ Transact-SQL).

Основной запрос

```
SELECT TagName, Value FROM AnalogHistory
WHERE TagName LIKE 'Sys%'
AND DateTime > '24 May 99 2:30pm'
AND DateTime < '24 May 99 2:32pm'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запросы поставщика данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT TagName, Value FROM v_AnalogHistory
WHERE TagName LIKE 'Sys%'
AND DateTime > '24 May 99 2:30pm'
AND DateTime < '24 May 99 2:32pm'
```

Результаты

Результаты для всех запросов:

TagName	Value
-----	-----
SYSACTIVEUSERS	10.0
SYSCRITERRCNT	0.0
SYSDATEDAY	24.0
SYSDATEMONTH	5.0
SYSDATEYEAR	1999.0
SYSERRERRCNT	0.0
SYSERRRRATE	3.0
SYSEVENTACTIONTHREADS	5.0
SYSEVENTCRITACTIONQSIZE	0.0
SYSEVENTDETECTORTHREADS	1.0
SYSEVENTNORMACTIONQSIZE	0.0
SYSFATALERRCNT	0.0
SYSIODRIVERBYTES	24.0
.	
.	
.	

Использование сложного выражения WHERE

Следующий запрос выдает все аналоговые тэги системы, которые начинаются с "sys" и имеют MaxEU менее 100.

Основной запрос

```
SELECT TagName, MaxEU FROM AnalogTag
WHERE TagName LIKE 'sys%'
AND MaxEU < 100
```

Результаты

TagName	MaxEU
SysDateDay	31.0
SysDateMonth	12.0
SysTimeHour	23.0
SysTimeMin	59.0
SysTimeSec	59.0

(задействовано 5 строк)

Использование псевдонимов таблиц

Для каждой таблицы в синтаксисе запроса может быть создано укороченное имя, называемое псевдонимом, которое затем можно использовать во всем запросе. Использование псевдонимов сокращает время написания запроса и упрощает синтаксис.

Основной запрос

Заметьте, что для таблицы *AnalogTag* использует псевдоним "A", для таблицы *Tag* используется псевдоним "T", а для таблицы *EngineeringUnit* — псевдоним "E".

Следующий запрос выдает значения всех тэгов, содержащих единицу измерения "M3".

```
SELECT A.TagName, T.Description
FROM AnalogTag A, Tag T, EngineeringUnit E
WHERE A.TagName = T.TagName
      AND A.EUKey = E.EUKey
      AND E.Unit = 'M3'
```

Результаты

Результаты:

TagName	Description
ProdLevel	Product storage level
ReactLevel	Reactor level

(задействовано 2 строки)

Запрос к "архивной" таблице расширения

Архивные таблицы показывают полученные производственные данные в архивном формате. К архивным таблицам относятся *AnalogHistory* и *DiscreteHistory*. Архивные таблицы расширения (удаленные таблицы) можно привязывать к любым другим таблицам IndustrialSQL Server.

Следующие запросы выдают метка даты/времени и значение тэга "ReactLevel".

Основной запрос

```
SELECT DateTime, TagName, Value
FROM AnalogHistory
WHERE TagName = 'ReactLevel'
AND DateTime >= 'Mar 2 1999 6:08'
AND DateTime <= 'Mar 2 1999 6:28'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT Дата и Время, Имя Тэга, Значение
FROM v_AnalogHistory
WHERE Имя Тэга = 'ReactLevel'
AND Дата и Время >= 'Mar 2 1999 6:08'
AND Дата и Время <= 'Mar 2 1999 6:28'
```

Результаты

Результаты всех запросов:

DateTime	TagName	Value
Mar 2 1999 6:08PM	REACTLEVEL	2000.0
Mar 2 1999 6:08PM	REACTLEVEL	65.0
Mar 2 1999 6:09PM	REACTLEVEL	2025.0
Mar 2 1999 6:10PM	REACTLEVEL	0.0
Mar 2 1999 6:10PM	REACTLEVEL	2000.0
Mar 2 1999 6:11PM	REACTLEVEL	-25.0
Mar 2 1999 6:11PM	REACTLEVEL	2025.0
Mar 2 1999 6:12PM	REACTLEVEL	0.0
Mar 2 1999 6:13PM	REACTLEVEL	2000.0
.		
.		
.		

(задействовано 100 строк)

Запрос к "живой" архивной таблице

"Живые" архивные таблицы предоставляют наиболее свежие данные для каждого тэга. Эти таблицы используются для получения текущего значения тэга или тэгов от IndustrialSQL Server. Любые "живые" архивные таблицы можно связывать с другими таблицами базы данных IndustrialSQL Server.

Следующие запросы выдают текущее значение указанного тэга.

Основной запрос

```
SELECT TagName, Value FROM AnalogLive
WHERE TagName = 'ReactLevel'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT TagName, Value FROM v_AnalogLive
WHERE TagName = 'ReactLevel'
```

Результаты

Результаты всех запросов:

TagName	Value
-----	-----
REACTLEVEL	1145.0

(задействована 1 строка)

Получение и хранение данных

В зависимости от того, подается запрос на данные для конкретного тэга в архивную таблицу или в "живую" таблицу, скорость изменения значения тэга может быть различной. Все это зависит от того, откуда поступают данные при подаче запроса. Настройки сервера ввода-вывода определяют скорость "чтения" или проверки PLC на данные. Живые таблицы хранят самые последние данные, полученные от серверов ввода-вывода. Например, если сервер ввода-вывода настроен на считывание данных с интервалом в 1 мс, тогда данные в живой таблице этого сервера ввода-вывода будут меняться каждую 1 мс.

Данные из сервера ввода-вывода могут сохраняться в архив со скоростью отличной от скорости, используемой сервером ввода-вывода, например, каждые 2мс. Скорость сохранения данных определяет, как часто сохраняются эти значения, независимо от остальной системы. То есть, если запрос подается на один и тот же тэг в архивную и в живую таблицы, данные будут изменяться с двумя разными скоростями: 1 мс для живой таблицы и 2 мс для архивной.

Запрос к "широкой" архивной таблице

Широкие таблицы расширения (в OLE DB это удаленные таблицы) являются модификациями таблиц *AnalogHistory* и *DiscreteHistory*. Широкие таблицы — это *AnalogWideHistory* и *DiscreteWideHistory*. Широкие архивные таблицы используются, когда требуется найти значение одного или более тэгов во времени. Использование широких таблиц обычно гораздо удобнее для конкретных типов запросов, чем использование таблиц *AnalogHistory* и *DiscreteHistory*. Широкие таблицы связываются только с другими таблицами, использующими дату/время.

Следующие запросы выдают значение двух тэгов из таблицы *AnalogWideHistory*.

Основной запрос

```
SELECT DateTime, ReactLevel, ReactTemp
FROM AnalogWideHistory
WHERE Reactlevel > 1500
AND ReactTemp > 150
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Доступ к таблицам *AnalogWideHistory* и *DiscreteWideHistory* возможен только через функцию OPENQUERY.

```
SELECT * FROM OpenQuery(INSQL, "
SELECT DateTime, ReactLevel, ReactTemp
FROM Runtime.dbo.AnalogWideHistory
WHERE Reactlevel > 1500
AND ReactTemp > 150
")
```

Результаты

Результаты всех запросов:

DateTime	ReactLevel	ReactTemp
Mar 2 1999 6:20PM	1865.0	191.3
Mar 2 1999 6:21PM	2025.0	195.9
Mar 2 1999 6:22PM	2000.0	195.9
Mar 2 1999 6:23PM	2025.0	180.9
Mar 2 1999 6:27PM	1505.0	177.5

(задействовано 5 строк)

Выражения OR в широких архивных таблицах

В настоящее время выражения OR не поддерживаются поставщиком данных InSQL OLE DB для широких архивных таблиц. Например:

```
SELECT * FROM OpenQuery(INSQL, "SELECT DateTime, SysTimeSec
    FROM Runtime.dbo.AnalogWideHistory
    WHERE SysTimeSec > 50
    OR    SysTimeSec < 10")
```

Для получения тех же результатов используется выражение UNION:

```
SELECT * FROM OPENQUERY(INSQL, "
    SELECT DateTime, SysTimeSec
    FROM Runtime.dbo.AnalogWideHistory
    WHERE SysTimeSec > 50
    ")
UNION ALL
SELECT * FROM OpenQuery(INSQL, "
    SELECT DateTime, SysTimeSec
    FROM Runtime.dbo.AnalogWideHistory
    WHERE SysTimeSec < 10
    ")
ORDER BY DateTime
```

Запросы к широким таблицам дельта-методом

Широкие таблицы нормально работают с дельта-методом, если выдается только один тэг. Тем не менее, для показа нескольких тэгов клиенту выдается целая строка по каждому моменту, когда один или несколько тэгов в запросе выдали разные значения. Строка отражает текущие значения, выдаваемые для запрошенных тэгов, и предыдущие значения для остальных тэгов в наборе результатов.

Примечание. Это значение может быть с "неправильным" или с другим значением качества.

Следующие запросы выдают значения для трех тэгов из таблицы *AnalogWideHistory*. "MyTagName" — тэг, который периодически становится недействительным.

Основной запрос

```
SET wwAnalogCyclic OFF
```

```
SELECT DateTime, SysTimeSec, SysTimeMin, MyTagName
FROM AnalogWideHistory
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Таблицы *AnalogWideHistory* и *DiscreteWideHistory* доступны только через функцию OPENQUERY.

```
SELECT * FROM OpenQuery(INSQL, "
    SELECT DateTime, SysTimeSec, SysTimeMin, MyTagName
    FROM Runtime.dbo.AnalogWideHistory
    WHERE wwRetrievalMode = 'Delta'
")
```

Результаты

DateTime	SysTimeSec	SysTimeMin	MyTagName
:	:	:	:
:	:	:	:
12 May 99 13:00	55	00	1
12 May 99 13:00	56	00	1
12 May 99 13:00	57	00	1
12 May 99 13:00	57	00	null
12 May 99 13:00	58	00	null
12 May 99 13:00	59	00	null
12 May 99 13:01	00	01	null
12 May 99 13:01	00	01	2
12 May 99 13:01	01	01	2
12 May 99 13:01	02	01	2
12 May 99 13:01	03	01	2
:	:	:	:
:	:	:	:
:	:	:	:

Заметьте, что 57 появляется дважды, поскольку первое изменение 1 на NULL для тэга "MyTagName" происходит между 57-ой и 58-ой секундами. То же самое относится к NULL, изменяющемуся на 2. Так же ведут себя и дискретные значения.

Использование арифметической функции

Следующие запросы добавляют значения двух тэгов из таблицы *AnalogWideHistory*.

Основной запрос

```
SELECT DateTime, ReactLevel, ProdLevel, 'Sum' =
    ReactLevel+Prodlevel
FROM AnalogWideHistory
WHERE DateTime > '28 Feb 99 6:56pm'
AND DateTime < '28 Feb 99 7:00pm'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Таблицы *AnalogWideHistory* и *DiscreteWideHistory* доступны только через функцию OPENQUERY.

```
SELECT * FROM OpenQuery(INSQL,"
    SELECT DateTime, ReactLevel, ProdLevel, 'Sum' =
        ReactLevel+Prodlevel
    FROM AnalogWideHistory
    WHERE DateTime > '28 Feb 99 6:56pm'
        AND DateTime < '28 Feb 99 7:00pm'
    ")
```

Результаты

Результаты всех запросов:

DateTime	ReactLevel	Prodlevel	Sum
Feb 28 1999 6:56PM	1525.0	2343.0	3868.0
Feb 28 1999 6:56PM	1525.0	2343.0	3868.0
Feb 28 1999 6:56PM	1525.0	2343.0	3868.0
Feb 28 1999 6:56PM	1525.0	2343.0	3868.0
Feb 28 1999 6:56PM	2025.0	2343.0	4368.0
Feb 28 1999 6:56PM	2025.0	2343.0	4368.0
.			
.			
.			

(задействовано 100 строк)

Использование функций даты

Функции даты проводят операцию со значениями ввода даты и времени и выдают текстовое, числовое значение или значение даты/времени.

Следующие запросы выдают метка даты/времени и значение тэга 'SysTimeSec' за последние 10 минут.

Основной запрос

```
SELECT DateTime, TagName, Value, Quality FROM AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= DateAdd(Minute, -10, GetDate())
      AND DateTime <= GetDate()
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

В настоящее время арифметика даты не пересылается из сервера SQL поставщику данных InSQL OLE DB. Чтобы обойти это ограничение, можно выполнить запрос, использующий заменяемые параметры, передаваемые во время исполнения.

Следующий запрос использует представление удаленной таблицы:

```
DECLARE @StartDate DateTime
DECLARE @EndDate   DateTime

SET @StartDate = dateadd(Minute, -10, GetDate())

SET @EndDate   = GetDate()

SELECT DateTime, TagName, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= @StartDate
      AND DateTime <= @EndDate
```

Результаты

DateTime	TagName	Value	Quality
1998-12-15 13:00:00.000	SYSTIMESEC	0.0	0
1998-12-15 13:00:06.060	SYSTIMESEC	6.0	0
1998-12-15 13:00:12.120	SYSTIMESEC	12.0	0
1998-12-15 13:00:18.180	SYSTIMESEC	18.0	0
1998-12-15 13:00:24.240	SYSTIMESEC	24.0	0
1998-12-15 13:00:30.300	SYSTIMESEC	30.0	0
1998-12-15 13:00:36.360	SYSTIMESEC	36.0	0
1998-12-15 13:00:42.420	SYSTIMESEC	42.0	0
.			
.			
.			

Использование функции агрегации

Следующие запросы выдают минимум, максимум, среднее и сумму для тэга 'AlarmTag1' из таблицы *AnalogWideHistory*.

Основной запрос

```
SELECT 'Minimum' = min(ReactLevel),  
       'Maximum' = max(ReactLevel),  
       'Average' = avg(ReactLevel),  
       'Sum' = sum(ReactLevel)  
FROM AnalogWideHistory  
WHERE DateTime > '28 Feb 99 6:55pm'  
AND DateTime < '28 Feb 99 7:00pm'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

```
SELECT * FROM OpenQuery(INSQL, "  
  SELECT 'Minimum' = min(ReactLevel),  
         'Maximum' = max(ReactLevel),  
         'Average' = avg(ReactLevel),  
         'Sum' = sum(ReactLevel)  
  FROM AnalogWideHistory  
  WHERE DateTime > '28 Feb 99 6:55pm'  
        AND DateTime < '28 Feb 99 7:00pm'  
  ")
```

Результаты

Результаты всех запросов:

Minimum	Maximum	Average	Sum
-25,0	2025,0	1181,2	118120,0

(задействована 1 строка)

Связывание таблиц сервера SQL выражением JOIN

Связка (join) — это класс запроса SQL, который запрашивает данные из одного и нескольких столбцов двух или более таблиц.

Основной запрос

Следующий запрос выдает значения всех тэгов, содержащих единицу измерения "МЗ".

```
SELECT A.TagName, T.Description
FROM AnalogTag A, Tag T, EngineeringUnit E
WHERE A.TagName = T.TagName
AND A.EUKey = E.EUKey
AND E.Unit = 'МЗ'
```

Результаты

Результаты:

TagName	Description
ProdLevel	Product storage level
ReactLevel	Reactor level

(задействовано 2 строки)

Связывание архивных таблиц выражением JOIN

Этот запрос проводит объединение между таблицами *AnalogHistory* и *DiscreteHistory*. Предпочитаемый метод проведения объединений с архивными таблицами есть использование поставщика данных InSQL OLE DB.

Запрос к поставщику данных OLE DB

```
SELECT A.DateTime, A.Value, D.Value from v_AnalogHistory A,
v_DiscreteHistory D
WHERE A.TagName = 'SysTimeSec'
AND D.TagName = 'SysPulse'
AND D.Значение > 0
AND A.DateTime = D.DateTime
AND A.DateTime >='Jan 12 1999 14:10:03'
AND A.DateTime <= 'Jan 12 1999 15:10:03'
AND A.wwResolution = 1000
AND D.DateTime >='Jan 12 1999 14:10:03'
AND D.DateTime <= 'Jan 12 1999 15:10:03'
AND D.wwRetrievalMode = 'Cyclic'
AND D.wwResolution =1000
```

Результаты

Результаты:

DateTime	Value	Value
1999-01-12 14:10:03,000	3,0	1,0
1999-01-12 14:10:04,000	4,0	1,0
1999-01-12 14:10:05,000	5,0	1,0
1999-01-12 14:10:06,000	6,0	1,0
1999-01-12 14:10:07,000	7,0	1,0
1999-01-12 14:10:08,000	8,0	1,0
1999-01-12 14:10:09,000	9,0	1,0
...		
1999-01-12 15:08:58,000	58,0	1,0
1999-01-12 15:08:59,000	59,0	1,0
1999-01-12 15:09:00,000	0,0	1,0
1999-01-12 15:09:01,000	1,0	1,0
1999-01-12 15:09:02,000	2,0	1,0
1999-01-12 15:10:03,000	3,0	1,0

(задействована 1801 строка)

Создание и запрос аннотаций

Следующие запросы создают аннотацию для указанного тэга. Аннотация создается в ответ на выключение насоса. Затем выдаются аннотации для конкретного тэга.

Основной запрос

```

DECLARE @@UserKey      INT

SELECT @@UserKey = UserKey
      FROM UserDetail
      WHERE UserName = 'wwadmin'

INSERT INTO Annotation (TagName, UserKey, DateTime, Content)
VALUES ('ReactLevel', @@UserKey, GetDate(), 'The Pump is
off')

SELECT DateTime, TagName, Content
      FROM Annotation
      WHERE Annotation.TagName = 'ReactLevel'
      AND DateTime > '27 Feb 99'
      AND DateTime <= GetDate()

```

Результаты

Результаты всех запросов:

DateTime	TagName	Content
Feb 28 1999 7:18PM	ReactLevel	The Pump is off

(задействована 1 строка)

Определение границ

Следующие запросы выдают значения границы "Hi" для указанного аналогового тэга.

Основной запрос

```
SELECT TagName, Value, Name FROM Limit, LimitName
WHERE Limit.TagName = 'ReactLevel'
      AND Limit.LimitNameKey = LimitName.LimitNameKey
      AND LimitName.Name = 'Hi'
```

Результаты

Результаты всех запросов:

TagName	Value	Name
ReactLevel	1800.0	Hi

(задействована 1 строка)

Использование расширений временного домена Industrial-SQL

Примеры в этом разделе показывают, как использовать расширения временного домена Industrial-SQL при подаче запроса в любые таблицы хранения данных. Для каждого типа запроса синтаксис требуется как для сервера базы данных IndustrialSQL, так и для поставщика данных InSQL OLE DB.

В примерах с сервером базы данных IndustrialSQL используются выражения SET для расширений временного домена. Тем не менее, можно использовать соответствующие хранимые процедуры вместо выражений SET. Например, вместо выражения **SET wwRowCount** в запросе можно использовать хранимую процедуру **ww_SetRowCount**.

Указание количества строк

Счетчик строк определяет количество строк, которые должны быть получены из любой архивной таблицы IndustrialSQL Server. Расширения временного домена для счетчика строк являются функцией IndustrialSQL Server, которая при циклическом доступе отличается от нормального подсчета строк сервера SQL. Применение подсчета строк расширения временного домена зависит от того, используется ли циклический или дельта-метод и запрашивается ли широкая таблица.

- Если указано ненулевое число строк для циклической выборки, IndustrialSQL Server выдаст количество строк, расположенных в равных интервалах в заданный период времени. Если число строк равно 0 или NULL (нет значения), а разрешение не указана, сервер выдаст столько строк, сколько возможно, максимум 100000. При запросе таблиц *AnalogHistory* и *DiscreteHistory* счетчик строк определяет количество выдаваемых строк на один тэг. Например, запрос, в котором счетчик строк задан 20 для двух тэгов, выдаст 40 строк данных (20 строк для каждого тэга). Если запрос подается в таблицы *AnalogWideHistory* и *DiscreteWideHistory*, счетчик строк указывает полное количество выдаваемых строк, несмотря на то, сколько тэгов запрошено. Например, запрос, в котором счетчик строк задан 20 для двух тэгов, выдаст 20 строк данных (10 строк для каждого тэга).
- Если указан ненулевой счетчик строк (N) для дельта-метода, выдаются первые N значений, изменившиеся за указанный временной интервал. Счетчик строк ограничивает максимальное количество выдаваемых строк, несмотря на то, сколько тэгов запрошено. Например, запрос, в котором счетчик строк задан 20 для четырех тэгов, выдаст максимум 20 строк данных (пять строк для каждого тэга, начиная с первой строки, содержащей первоначальное значение для каждого из четырех тэгов). Если счетчик строк равен 0 или NULL (нет значения), то будут выданы все значения, изменившиеся за указанный период времени. Это аналогично установке счетчика строк для сервера SQL.

При использовании поставщика данных InSQL OLE DB и указании как дискретности, так и счетчика строк для одного и того же запроса, строки не выдаются. Если используется сервер базы данных InSQL, то применяется последнее использованное в запросе расширение.

Подробнее см. в разделе "Указание разрешения, счетчика строк и метода выборки в одном запросе" далее в этой главе.

Следующие запросы демонстрируют работу счетчика строк по отношению к одному или нескольким тэгам в одном и том же запросе.

Основные запросы

Запрос 1: Для одиночного тэга:

```
Set wwRowCount 300

SELECT DateTime, TagName, Value FROM AnalogHistory
  WHERE TagName = 'SysTimeSec'
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
```

Запрос 2: Для множества тэгов:

```
ww_SetRowCount

SELECT DateTime, TagName, Value FROM AnalogHistory
  WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
```

Те же запросы с использованием синтаксиса OLE DB приведены в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленной таблицы:

Запрос 1:

```
SELECT DateTime, TagName, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
        AND wwRowCount = 300
```

Запрос 2:

```
SELECT DateTime, TagName, Value FROM v_AnalogHistory
  WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
        AND wwRowCount = 300
```

Результаты

Результаты всех запросов:

Запрос 1:

DateTime	TagName	Value
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.200	SYSTIMESEC	0.0
1998-12-09 11:35:00.400	SYSTIMESEC	0.0
1998-12-09 11:35:00.600	SYSTIMESEC	0.0
...		
1998-12-09 11:35:59.200	SYSTIMESEC	59.0
1998-12-09 11:35:59.400	SYSTIMESEC	59.0
1998-12-09 11:35:59.600	SYSTIMESEC	59.0
1998-12-09 11:35:59.800	SYSTIMESEC	59.0

Запрос 2:

DateTime	TagName	Value
1998-12-09 11:35:00.000	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.200	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.400	SYSTIMEMIN	35.0
1998-12-09 11:35:00.400	SYSTIMESEC	0.0
1998-12-09 11:35:00.600	SYSTIMEMIN	35.0
1998-12-09 11:35:00.600	SYSTIMESEC	0.0
...		
1998-12-09 11:35:59.200	SYSTIMEMIN	35.0
1998-12-09 11:35:59.200	SYSTIMESEC	59.0
1998-12-09 11:35:59.400	SYSTIMEMIN	35.0
1998-12-09 11:35:59.400	SYSTIMESEC	59.0
1998-12-09 11:35:59.600	SYSTIMEMIN	35.0
1998-12-09 11:35:59.600	SYSTIMESEC	59.0
1998-12-09 11:35:59.800	SYSTIMEMIN	35.0
1998-12-09 11:35:59.800	SYSTIMESEC	59.0

Заметьте, что значения двух тэгов перемешаны в одном и том же столбце.

Указание разрешения

Разрешение — это интервал в миллисекундах для выборки данных из любой аналоговой или дискретной таблицы IndustrialSQL Server. Расширение временного домена разрешения IndustrialSQL Server не поддерживается обычными функциями сервера SQL.

При указании разрешения для выборки данных в запросе должен быть указан конкретный период времени. Система выдаст значения, сохраненные за требуемый период времени с интервалом, определенным дискретностью. Например, если указать разрешение в 5000 мс, система запросит все данные за период времени, но выдаст только те значения, которые были получены с интервалом в 5000 миллисекунд. Количество выдаваемых строк зависит от периода времени и разрешения (количество строк = период времени / разрешение).

Разрешение применяется только при циклическом методе выборки. Если значение разрешения или счетчика строк не указано, то берется 100 строк с равными промежутками. При использовании поставщика данных InSQL OLE DB и указании как разрешения, так и счетчика строк для одного и того же запроса, строки не выдаются. Если используется сервер базы данных InSQL, то применяется последнее использованное в запросе расширение.

Подробнее см. в разделе "Указание разрешения, счетчика строк и метода выборки в одном запросе" далее в этой главе.

Примечание. Для дельта-метода разрешение можно установить при помощи временной мертвой зоны.

Следующие запросы выдают строки с интервалом в 2 секунды (2000 миллисекунд) за требуемый период времени. Данные получаются циклично.

Основной запрос

```
SET wwResolution 2000

SELECT DateTime, TagName, Value FROM AnalogHistory
  WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, TagName, Value FROM v_AnalogHistory
WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
AND wwResolution = 2000
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:36'
```

Результаты

DateTime	TagName	Value
-----	-----	-----
1998-12-09 11:35:00.000	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.000	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:04.000	SYSTIMEMIN	35.0
1998-12-09 11:35:04.000	SYSTIMESEC	4.0
1998-12-09 11:35:06.000	SYSTIMEMIN	35.0
...		
1998-12-09 11:35:54.000	SYSTIMEMIN	35.0
1998-12-09 11:35:54.000	SYSTIMESEC	54.0
1998-12-09 11:35:56.000	SYSTIMEMIN	35.0
1998-12-09 11:35:56.000	SYSTIMESEC	56.0
1998-12-09 11:35:58.000	SYSTIMEMIN	35.0
1998-12-09 11:35:58.000	SYSTIMESEC	58.0
1998-12-09 11:36:00.000	SYSTIMEMIN	36.0
1998-12-09 11:36:00.000	SYSTIMESEC	0.0

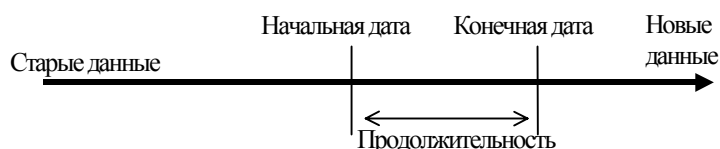
Глобальная установка стартовой и конечной дат

Примечание. Этот раздел относится только к серверу базы данных IndustrialSQL, а не к поставщику данных InSQL OLE DB. При использовании поставщика данных InSQL OLE DB следует указать дату начала и окончания в выражении WHERE для каждого запроса.

При запросе к таблицам хранения данных можно выбрать записи, сохраненные за период между указанной конечной датой и стартовой датой (которая равняется дате окончания минус продолжительность). Так как таблицы хранения данных вообще могут сохранять неограниченное количество данных, следует сузить область запроса во избежание выдачи огромного количества данных. Сужение области запроса также повышает эффективность работы сервера. Выполнение запроса "SELECT ALL" с компьютера, на котором установлен клиент, в большую таблицу, как например *AnalogHistory* приведет к передаче по сети множества гигабайт информации.

В Transact-SQL следует указать требуемую стартовую дату и конечную дату для любого запроса, имеющего критерий в выражении WHERE. Этот критерий следует указать для каждого запроса в наборе. Тем не менее, в Industrial-SQL можно использовать конечную дату и расширение временного домена продолжительности для установки стартовой даты и конечной даты лишь один раз для набора из нескольких запросов.

Конечная дата — последний метка даты/времени для данных, которые требуется получить. Продолжительность — смещение (в секундах) от этой конечной даты. Стартовая дата подсчитывается путем отнимания значения продолжительности от конечной даты. Эта концепция показана в следующей диаграмме:



Если дата не указана, то по умолчанию будут использованы текущие дата и время.

Расчеты проводятся с данными, которые будут выданы по запросу. Неверные данные не включаются в расчет и по сети пересылаются только результаты.

Следующие запросы выдают минимум, максимум, среднее значение и сумму для аналогового тэга 'ReactLevel' за последние 3 минуты, заканчивая в 3:00 p.m. 4 марта 1999 г.

Основной запрос

```
SET wwDuration (minute, 3)
SET wwEndDate '4 Mar 99 3:00pm'
SELECT 'Minimum' = MIN(Value),
       'Maximum' = MAX(Value),
       'Average' = AVG(Value),
       'Sum'     = SUM(Value)
FROM AnalogHistory
WHERE TagName = 'ReactLevel'
```

Результаты

Результаты всех запросов:

Minimum	Maximum	Average	Sum
-25,0	2025,0	1025,45	102545,0

(задействована 1 строка)

SELECT INTO из архивной таблицы

Следующие запросы вставляют данные из таблицы *AnalogWideHistory* в другую таблицу с названием *MyTable*. Затем запрашиваются данные из таблицы *MyTable*.

Основной запрос

```
ww_SetResolution  
  
ww_SetEndDate '13 Mar 1999 2:00pm'  
  
ww_SetDuration (minute, 2)  
  
DROP TABLE MyTable  
  
SELECT DateTime, ReactTemp, ReactLevel  
    INTO MyTable  
    FROM AnalogWideHistory  
  
SELECT * FROM MyTable
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует функцию OPENQUERY.

```
DROP TABLE MyTable  
  
SELECT DateTime,  
    Sec = datepart(ss, DateTime),  
    mS = datepart(ms, DateTime),  
    WHERE wwResolution = 5000  
    AND DateTime < '13 Mar 1999 2:00pm'  
    AND DateTime > '13 Mar 1999 1:58pm'  
    INTO MyTable  
    FROM OPENQUERY(INSQL, "SELECT DateTime, ReactTemp,  
        ReactLevel, FROM AnalogWideHistory WHERE  
        wwResolution = 5000 AND DateTime < '13 Mar 1999  
        2:00pm' AND DateTime > '13 Mar 1999 1:58pm'")  
  
SELECT * FROM MyTable
```

Результаты

Результаты всех запросов:

DateTime	Sec	mS	ReactTemp	ReactLevel
Mar 13 1999 1:58PM		0.0		190.9
Mar 13 1999 1:58PM		5.0		190.9
Mar 13 1999 1:58PM		10.0		168.3
Mar 13 1999 1:58PM		15.0		168.3
Mar 13 1999 1:58PM		20.0		133.8
Mar 13 1999 1:58PM		25.0		133.8
Mar 13 1999 1:58PM		30.0		101.6
Mar 13 1999 1:58PM		35.0		101.6
Mar 13 1999 1:58PM		40.0		32.4
Mar 13 1999 1:58PM		45.0		32.4
Mar 13 1999 1:58PM		50.0		20.9
Mar 13 1999 1:58PM		55.0		20.9
Mar 13 1999 1:59PM		0.0		85.9
Mar 13 1999 1:59PM		5.0		85.9
Mar 13 1999 1:59PM		10.0		185.9
Mar 13 1999 1:59PM		15.0		185.9
Mar 13 1999 1:59PM		20.0		168.3
Mar 13 1999 1:59PM		25.0		168.3
Mar 13 1999 1:59PM		30.0		136.1
Mar 13 1999 1:59PM		35.0		136.1
Mar 13 1999 1:59PM	40	0	103.9	-25.0
Mar 13 1999 1:59PM	45	0	103.9	-25.0
Mar 13 1999 1:59PM		50.0		34.7
Mar 13 1999 1:59PM		55.0		34.7
Mar 13 1999 2:00PM		0.0		20.9

(задействовано 25 строк)

Продолжительное получение значений "живых" данных

При запросе данных из живой таблицы расширения можно выбрать автоматическое обновление для клиента текущих значений запрошенных тэгов при их изменении. Это выполняется при помощи продолжительного расширения временного домена. Это расширение временного домена имеет эффект только на живых таблицах расширения (*AnalogLive*, *DiscreteLive*).

Продолжительное расширение временного домена имеет два параметра. Первый параметр включает и выключает продолжительное получение данных. Когда значение первого параметра - ON, запрос продолжит выполняться, пока не будет отменен. Второй параметр (выборочно) используется для указания мертвой зоны минимального времени обновления для значений данных, независимо от того, изменились они или нет.

Продолжительный запрос остается активным, пока приложение-клиент не разъединится, или пока продолжительное расширение временного домена не будет использовано вновь с параметром OFF.

Примечание. Продолжительное расширение временного домена не поддерживается текущим выпуском поставщика данных InSQL OLE DB.

Основной запрос

Следующий запрос выдает новую строку при каждом изменении аналогового тэга 'ReactLevel'.

```
SET wwContinuous ON

SELECT DateTime, Sec = DATEPART(ss, DateTime), Value, Quality
FROM AnalogLive
WHERE TagName = 'SysTimeSec'
AND DateTime >= '20 Apr 99 1:00:00pm'
AND DateTime <= '20 Apr 99 1:01:00pm'
```

Следующий запрос выдает новую строку при каждом изменении значения аналогового тэга 'ReactLevel', но имеет минимальную мертвую зону в 5 секунд.

```
SET wwContinuous ON, 5000

SELECT DateTime, Sec = DATEPART(ss, DateTime), Value, Quality
FROM AnalogLive
WHERE TagName = 'ReactLevel'
AND DateTime >= '20 Apr 99 1:00:00pm'
AND DateTime <= '20 Apr 99 1:01:00pm'
```

Циклическая выборка

Циклическая выборка — это выборка всех данных, сохраненных за данный интервал времени, независимо от того, изменилось ли значение тэга. Если требуется получать данные циклично, используйте аналоговое циклическое или дискретное циклическое расширение временного домена.

Как аналоговое, так и циклическое расширения временного домена используют один параметр, который включает и выключает циклическую выборку. Если циклическая выборка выключена, используется дельта-выборка (или выборка методом исключения).

При использовании счетчика строк или разрешения в циклической выборке выдаются различные результаты.

Подробнее см. в разделе "Указание разрешения, счетчика строк и метода выборки в одном запросе" далее в этой главе.

Следующие запросы выдают значения данных для аналогового тэга 'ReactLevel'.

Основной запрос

```
SET wwAnalogCyclic ON

SELECT DateTime, Sec = DATEPART(ss, DateTime), Tagname, Value
FROM AnalogHistory
WHERE TagName = 'ReactLevel'
AND DateTime >= '13 Mar 99 1:15:00pm'
AND DateTime <= '13 Mar 99 2:15:00pm'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Sec = DATEPART(ss, DateTime), Tagname, Value
FROM v_AnalogHistory
WHERE TagName = 'ReactLevel'
AND DateTime >= '13 Mar 99 1:15:00pm'
AND DateTime <= '13 Mar 99 2:15:00pm'
AND wwRetrievalMode = 'Cyclic'
```

Результаты

Результаты:

DateTime	Sec	TagName	Value
Mar 13 1999 1:15PM		REACTLEVEL	0.1775
Mar 13 1999 1:15PM		REACTLEVEL	36.1260
Mar 13 1999 1:16PM		REACTLEVEL	12.1650
Mar 13 1999 1:16PM		REACTLEVEL	49.1280
Mar 13 1999 1:17PM		REACTLEVEL	25.1525
Mar 13 1999 1:18PM		REACTLEVEL	1.585
Mar 13 1999 1:18PM		REACTLEVEL	38.1400
Mar 13 1999 1:19PM		REACTLEVEL	14.650
Mar 13 1999 1:19PM		REACTLEVEL	50.2025
Mar 13 1999 1:20PM		REACTLEVEL	27.765
Mar 13 1999 1:21PM		REACTLEVEL	3.2000
Mar 13 1999 1:21PM		REACTLEVEL	39.830
Mar 13 1999 1:22PM		REACTLEVEL	16.1925
.			
.			
.			

(задействована 721 строка)

Дельта-выборка

Дельта-выборка — это выборка только изменившихся значений тэгов за заданный временной интервал. Когда требуется получить данные дельта-методом, используйте аналоговое циклическое или дискретное циклическое расширение временного домена.

Как аналоговое, так и циклическое расширения временного домена используют один параметр, который включает и выключает циклическую выборку. Если циклическая выборка выключена, используется дельта-метод.

При использовании счетчика строк или разрешения в циклической выборке выдаются различные результаты.

Подробнее см. в разделе "Указание разрешения, счетчика строк и метода выборки в одном запросе" далее в этой главе.

Следующие запросы выдают значения данных для аналогового тэга 'SysTimeSec'. Выдается только дата изменения значения.

Примечание. Нельзя использовать расширение временного домена разрешения для дельта-выборки. Тем не менее, для получения тех же результатов можно использовать временную мертвую зону.

Основной запрос

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value FROM AnalogHistory
WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:36'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:36'
AND wwRetrievalMode = 'Delta'
```

Результаты

Результаты:

DateTime	TagName	Value
-----	-----	-----
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.000	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:02.000	SYSTIMESEC	2.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:04.000	SYSTIMESEC	4.0
...		
1998-12-09 11:35:58.000	SYSTIMESEC	58.0
1998-12-09 11:35:59.000	SYSTIMESEC	59.0
1998-12-09 11:36:00.000	SYSTIMESEC	0.0
1998-12-09 11:36:00.000	SYSTIMEMIN	36.0

Установка временной мертвой зоны для дельта-выборки

Временная мертвая зона для выборки управляет разрешением времени данных, полученным дельта-методом. Любые изменения, произошедшие за время мертвой зоны (в мс) выдаваться не будут. Для указания мертвой зоны для дельта-выборки используется расширение временной мертвой зоны. При циклическом методе выборки мертвая зона игнорируется.

Временные мертвые зоны применимы к аналоговым, дискретным и текстовым тэгам. Также временные мертвые зоны применимы к таблицам *AnalogLive*, *DiscreteLive* и *StringLive* при использовании продолжительного расширения запроса временного домена.

"Базовое значение" мертвой зоны сбрасывается при каждой выдаче значения, так что последнее выданное значение служит основой для подсчета мертвой зоны. Каждая выборка сверяется с условиями мертвой зоны для проверки того, следует ли ее выдавать.

Следующие запросы выдают значения данных для аналогового тэга 'SysTimeSec'. Один запрос требует выдавать только данные, изменившиеся в течение 5-секундной мертвой зоны. Второй запрос требует выдавать только данные, изменившиеся в течение мертвой зоны в 4900 миллисекунд.

Основные запросы

Запрос 1: Дельта-выборка с 5-секундной временной мертвой зоной и одним тэгом:

```
SET wwAnalogCyclic OFF

SET wwTimeDeadband 5000

SELECT DateTime, TagName, Value FROM AnalogHistory
  WHERE TagName = 'SysTimeSec'
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:37'
```

Запрос 2: Дельта-выборка с временной мертвой зоной 4900 миллисекунд и одним тэгом:

```
SET wwAnalogCyclic OFF

SET wwTimeDeadband 4900

SELECT DateTime, TagName, Value FROM AnalogHistory
  WHERE TagName = 'SysTimeSec'
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:37'
```

Запрос 3: Дельта-выборка с 2-секундной мертвой зоной и двумя тэгами:

```
SET wwAnalogCyclic OFF

SET wwTimeDeadband 2000

SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName IN ('SysTimeSec', 'SysTimeMin')
        AND DateTime >= 'Dec 9 1998 11:35'
        AND DateTime <= 'Dec 9 1998 11:36'
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:37'
AND wwRetrievalMode = 'Delta'
AND wwTimeDeadband = 5000
```

Запрос 2:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:37'
AND wwRetrievalMode = 'Delta'
AND wwTimeDeadband = 4900
```

Запрос 3:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName IN ('SysTimeSec','SysTimeMin')
AND DateTime >= 'Dec 9 1998 11:35'
AND DateTime <= 'Dec 9 1998 11:36'
AND wwRetrievalMode = 'Delta'
AND wwTimeDeadband = 2000
```

Результаты

Запрос 1: Результаты запроса с 5-секундной мертвой зоной для 'SysTimeSec':

DateTime	TagName	Value
-----	-----	-----
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:06.000	SYSTIMESEC	6.0
1998-12-09 11:35:12.000	SYSTIMESEC	12.0
1998-12-09 11:35:18.000	SYSTIMESEC	18.0
1998-12-09 11:35:24.000	SYSTIMESEC	24.0
1998-12-09 11:35:30.000	SYSTIMESEC	30.0
1998-12-09 11:35:36.000	SYSTIMESEC	36.0
1998-12-09 11:35:42.000	SYSTIMESEC	42.0
1998-12-09 11:35:48.000	SYSTIMESEC	48.0
1998-12-09 11:35:54.000	SYSTIMESEC	54.0
1998-12-09 11:36:00.000	SYSTIMESEC	0.0
1998-12-09 11:36:06.000	SYSTIMESEC	6.0
1998-12-09 11:36:12.000	SYSTIMESEC	12.0
1998-12-09 11:36:18.000	SYSTIMESEC	18.0
1998-12-09 11:36:24.000	SYSTIMESEC	24.0
1998-12-09 11:36:30.000	SYSTIMESEC	30.0
1998-12-09 11:36:36.000	SYSTIMESEC	36.0
1998-12-09 11:36:42.000	SYSTIMESEC	42.0
1998-12-09 11:36:48.000	SYSTIMESEC	48.0
1998-12-09 11:36:54.000	SYSTIMESEC	54.0
1998-12-09 11:37:00.000	SYSTIMESEC	0.0

Запрос 2: Результаты запроса с мертвой зоной в 4900 миллисекунд для тэга 'SysTimeSec':

DateTime	TagName	Value
-----	-----	-----
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:10.800	SYSTIMESEC	10.0
1998-12-09 11:35:15.700	SYSTIMESEC	15.0
1998-12-09 11:35:20.600	SYSTIMESEC	20.0
1998-12-09 11:35:25.500	SYSTIMESEC	25.0
1998-12-09 11:35:30.400	SYSTIMESEC	30.0
1998-12-09 11:35:35.300	SYSTIMESEC	35.0
1998-12-09 11:35:40.200	SYSTIMESEC	40.0
1998-12-09 11:35:45.100	SYSTIMESEC	45.0
1998-12-09 11:35:50.000	SYSTIMESEC	50.0
1998-12-09 11:35:55.900	SYSTIMESEC	55.0
1998-12-09 11:36:00.000	SYSTIMESEC	0.0
1998-12-09 11:36:05.900	SYSTIMESEC	5.0
1998-12-09 11:36:10.800	SYSTIMESEC	10.0
1998-12-09 11:36:15.700	SYSTIMESEC	15.0
1998-12-09 11:36:20.600	SYSTIMESEC	20.0
1998-12-09 11:36:25.500	SYSTIMESEC	25.0
1998-12-09 11:36:30.400	SYSTIMESEC	30.0
1998-12-09 11:36:35.300	SYSTIMESEC	35.0
1998-12-09 11:36:40.200	SYSTIMESEC	40.0
1998-12-09 11:36:45.100	SYSTIMESEC	45.0
1998-12-09 11:36:50.000	SYSTIMESEC	50.0
1998-12-09 11:36:55.900	SYSTIMESEC	55.0
1998-12-09 11:37:00.000	SYSTIMESEC	0.0

Запрос 2: Результаты запроса с мертвой зоной в 2000 миллисекунд для 'SysTimeSec' и 'SysTimeMin':

DateTime	TagName	Value
-----	-----	-----
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:00.000	SYSTIMEMIN	35.0
1998-12-09 11:35:00.000	SYSTIMESEC	0.0
1998-12-09 11:35:06.000	SYSTIMESEC	6.0
1998-12-09 11:35:09.000	SYSTIMESEC	9.0
1998-12-09 11:35:12.000	SYSTIMESEC	12.0
1998-12-09 11:35:15.000	SYSTIMESEC	15.0
1998-12-09 11:35:18.000	SYSTIMESEC	18.0
1998-12-09 11:35:21.000	SYSTIMESEC	21.0
1998-12-09 11:35:24.000	SYSTIMESEC	24.0
1998-12-09 11:35:27.000	SYSTIMESEC	27.0
1998-12-09 11:35:30.000	SYSTIMESEC	30.0
1998-12-09 11:35:33.000	SYSTIMESEC	33.0
1998-12-09 11:35:36.000	SYSTIMESEC	36.0
1998-12-09 11:35:39.000	SYSTIMESEC	39.0
1998-12-09 11:35:42.000	SYSTIMESEC	42.0
1998-12-09 11:35:45.000	SYSTIMESEC	45.0
1998-12-09 11:35:48.000	SYSTIMESEC	48.0
1998-12-09 11:35:51.000	SYSTIMESEC	51.0
1998-12-09 11:35:54.000	SYSTIMESEC	54.0
1998-12-09 11:35:57.000	SYSTIMESEC	57.0

Установка значения мертвой зоны для дельта-выборки

Значение мертвой зоны для выборки данных управляет разрешением значений данных, выдаваемых дельта-методом. Любые значения данных, которые изменились меньше, чем указанная мертвая зона, выдаваться не будут. Значение — это процент от полного масштаба в единицах измерения. Для указания значения мертвой зоны выборки используйте расширение значения мертвой зоны. Мертвая зона игнорируется при циклическом методе выборки данных.

Базовое значение мертвой зоны сбрасывается при каждой выдаче значения, так что последнее выданное значение выступает в качестве основы для подсчета мертвой зоны. Каждая запись сверяется с условиями мертвой зоны для проверки того, следует ли ее выдавать.

Следующие запросы выдают значения данных для аналогового тэга 'SysTimeSec'. Минимальная единица измерения для 'SysTimeSec' — 0, а максимальная — 59.

Первый запрос требует выдавать только данные, изменившиеся более чем на 10 процентов полной протяженности единицы измерения тэга. Использование значения мертвой зоны в 10 процентов приводит к абсолютному изменению на 5,9 для 'SysTimeSec'.

Второй запрос требует выдавать только данные, изменившиеся более чем на 5 процентов полной протяженности единицы измерения тэга. Использование значения мертвой зоны в 10 процентов приводит к абсолютному изменению на 2,95 для 'SysTimeSec'.

Основные запросы

Запрос 1: 10-процентная мертвая зона:

```
SET wwAnalogCyclic OFF
SET wwValueDeadband 10
SELECT DateTime, Value
  FROM AnalogHistory
 WHERE TagName = 'SysTimeSec'
```

Запрос 2: 5-процентная мертвая зона:

```
SET wwAnalogCyclic OFF
SET wwValueDeadband 5
SELECT DateTime, Value
  FROM AnalogHistory
 WHERE TagName = 'SysTimeSec'
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 9 1998 11:35'
      AND DateTime <= 'Dec 9 1998 11:37'
      AND wwRetrievalMode = 'Delta'
      AND wwValueDeadband = 10
```

Запрос 2:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 9 1998 11:35'
      AND DateTime <= 'Dec 9 1998 11:37'
      AND wwRetrievalMode = 'Delta'
      AND wwValueDeadband = 5
```

Результаты

Запрос 1:

Date/Time	Value
1998-12-09 11:35:00.000 0.0	
1998-12-09 11:35:06.000 6.0	
1998-12-09 11:35:12.000 12.0	
1998-12-09 11:35:18.000 18.0	
1998-12-09 11:35:24.000 24.0	
1998-12-09 11:35:30.000 30.0	
1998-12-09 11:35:36.000 36.0	
1998-12-09 11:35:42.000 42.0	
1998-12-09 11:35:48.000 48.0	
1998-12-09 11:35:54.000 54.0	
1998-12-09 11:36:00.000 0.0	
1998-12-09 11:36:06.000 6.0	
1998-12-09 11:36:12.000 12.0	
1998-12-09 11:36:18.000 18.0	
1998-12-09 11:36:24.000 24.0	
1998-12-09 11:36:30.000 30.0	
1998-12-09 11:36:36.000 36.0	
1998-12-09 11:36:42.000 42.0	
1998-12-09 11:36:48.000 48.0	
1998-12-09 11:36:54.000 54.0	
1998-12-09 11:37:00.000 0.0	

Запрос 2:

Date/Time	Value
1998-12-09 11:35:00.000 0.0	
1998-12-09 11:35:03.000 3.0	
1998-12-09 11:35:06.000 6.0	
1998-12-09 11:35:09.000 9.0	
1998-12-09 11:35:12.000 12.0	
1998-12-09 11:35:15.000 15.0	
1998-12-09 11:35:18.000 18.0	
1998-12-09 11:35:21.000 21.0	
1998-12-09 11:35:24.000 24.0	
1998-12-09 11:35:27.000 27.0	
1998-12-09 11:35:30.000 30.0	
1998-12-09 11:35:33.000 33.0	
1998-12-09 11:35:36.000 36.0	
1998-12-09 11:35:39.000 39.0	
1998-12-09 11:35:42.000 42.0	
1998-12-09 11:35:45.000 45.0	
1998-12-09 11:35:48.000 48.0	
1998-12-09 11:35:51.000 51.0	
1998-12-09 11:35:54.000 54.0	
1998-12-09 11:35:57.000 57.0	
1998-12-09 11:36:00.000 0.0	
1998-12-09 11:36:03.000 3.0	
1998-12-09 11:36:06.000 6.0	
1998-12-09 11:36:09.000 9.0	
1998-12-09 11:36:12.000 12.0	
1998-12-09 11:36:15.000 15.0	
1998-12-09 11:36:18.000 18.0	
1998-12-09 11:36:21.000 21.0	
1998-12-09 11:36:24.000 24.0	
1998-12-09 11:36:27.000 27.0	
1998-12-09 11:36:30.000 30.0	
1998-12-09 11:36:33.000 33.0	
1998-12-09 11:36:36.000 36.0	
1998-12-09 11:36:39.000 39.0	
1998-12-09 11:36:42.000 42.0	
1998-12-09 11:36:45.000 45.0	
1998-12-09 11:36:48.000 48.0	
1998-12-09 11:36:51.000 51.0	
1998-12-09 11:36:54.000 54.0	
1998-12-09 11:36:57.000 57.0	
1998-12-09 11:37:00.000 0.0	

Установка временной мертвой зоны и значения мертвой зоны для выборки

Если указаны и временная мертвая зона, и значение мертвой зоны, то каждая запись проверяется на обе мертвые зоны в соответствии с базовым значением (последняя выданная запись).

Если оба теста пройдены, то запись выдается; при этом она служит основой для проверки следующей записи.

Указание разрешения, счетчика строк и метода выборки в одном запросе

Результаты запроса базы данных различаются в зависимости от комбинации разрешения, счетчика строк и метода выборки, используемого в запросе. Эти результаты описываются следующей таблицей:

Разрешение	Счетчик строк	Режим выборки	Результаты
N	0 или NULL	CYCLIC	Опрашиваются все сохраненные данные для тэгов за указанный интервал времени, а затем применяется разрешение в N мс.
0 или NULL	0 или NULL	CYCLIC	Ноль строк выдано.
0 или NULL	N	CYCLIC	Опрашиваются все сохраненные данные для тэга за указанный интервал времени, а затем применяется счетчик строк в N строк с равными промежутками.
N	N	CYCLIC	Ноль строк выдано.
(нет значения)	(нет значения)	CYCLIC	Опрашиваются все сохраненные данные для тэга за указанный интервал времени и сервер выдает столько строк, сколько возможно, до 10000 строк.
(любое значение игнорируется)	0 или NULL	DELTA	Выдаются значения, которые изменились за указанный промежуток времени.
(любое значение игнорируется)	N	DELTA	Запрашиваются значения, изменившиеся за указанный промежуток времени, а затем применяется счетчик строк (первые N строк).
(нет значения)	(нет значения)	DELTA	Выдаются значения, которые изменились за указанный промежуток времени. (По умолчанию счетчик строк равен 0.)

N = Числовое значение. Заметьте, что NULL не является тем же самым что "нет значения" (пустое значение).

Обычно, если в виртуальных столбцах есть ошибка или неразрешаемый конфликт, строки не выдаются.

Использование операторов сравнения при дельта-выборке

В этом разделе описывается работа системы при выполнении типичных дельта-запросов, где стартовая дата и конечная дата заданы при помощи операторов сравнения $\geq$, $>$, $\leq$ и $<$. Операторы сравнения используются с таблицами *AnalogHistory*, *AnalogWideHistory*, *DiscreteHistory* и *DiscreteWideHistory*. Операторы сравнения также применяются несмотря на то тип запроса (наименование из четырех частей, представления поставщика данных OLE DB и т.п.).

Дельта-запросы, использующие операторы сравнения, выдают все правильные изменения для набора тэгов за указанный промежуток времени. Использование мертвых зон и прочих фильтров может изменять набор правильных изменений.

Указание сартовой дты при помощи ">="

Если стартовая дата указана с помощью $\geq$ (больше или равно), то строка всегда выдается в указанную стартовую дату. Если это время в точности совпадает с правильным изменением, то качество нормальное (0). Иначе выдается *исходное значение* и качество это отображает (133). Запрос, в котором $\geq$ используется для указания стартовой даты, всегда выдает минимум одну строку.

Основные запросы

Запрос 1: Для этого запроса стартовая дата не влияет на изменение данных:

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
       AND DateTime >= 'Jan 13 1999 12:00:30'
       AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 2: Для этого запроса стартовая дата влияет на изменение данных:

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
       AND DateTime >= 'Jan 13 1999 12:01:00'
       AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 3: Для этого запроса стартовая дата выдаст минимум одну строку, даже если запрос не обнаруживает изменений данных:

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
       AND DateTime >= 'Jan 13 1999 12:00:30'
       AND DateTime < 'Jan 13 1999 12:01:00'
```

Об этих запросах с синтаксисом OLE DB см. следующий раздел.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime >= 'Jan 13 1999 12:00:30'
      AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 2:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime >= 'Jan 13 1999 12:01:00'
      AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 3:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime >= 'Jan 13 1999 12:00:30'
      AND DateTime < 'Jan 13 1999 12:01:00'
```

Результаты

Результаты 1: Стартовое время (12:00:30) не влияет на изменение значения и поэтому отмечено исходным значением качества -- 133:

Date/Time	Value	Quality
1999-01-13 12:00:30.000	0.0	133
1999-01-13 12:01:00.000	1.0	0
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0

(задействовано 10 строк)

Результаты 2: Стартовое время (12:01:00) прямо влияет на изменение значения и поэтому отмечено нормальным качеством -- 0.

Date/Time	Value	Quality
1999-01-13 12:01:00.000	1.0	0
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0

(задействовано 9 строк)

Результаты 3: Этот запрос не замечает изменения значения и поэтому отмечен исходным значением качества (133) для стартового времени запроса:

Date/Time	Value	Quality
-----	-----	-----
1999-01-13 12:00:30.000	0.0	133

(задействована 1 строка)

Указание стартовой даты с помощью ">"

Если стартовая дата указывается с помощью > (больше), то первая выданная строка есть первое правильное изменение после (не включительно) стартовой даты. Строка *исходного значения* не выдается. Запрос, использующий > для указания стартовой даты, может не выдать строк.

Основные запросы

Запрос 1: Для этого запроса первая выданная строка будет первым правильным изменением после (не включительно) стартового времени (12:00:30):

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND DateTime > 'Jan 13 1999 12:00:30'
     AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 2: Для этого запроса стартовая дата влияет на изменение данных, но будет исключена из результатов, потому что используемым оператором является *больше*, а не *больше или равно*.

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND DateTime > 'Jan 13 1999 12:01:00'
     AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 3: Этот запрос не выдаст строк, потому что изменения данных не определяются:

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND DateTime > 'Jan 13 1999 12:00:30'
     AND DateTime < 'Jan 13 1999 12:01:00'
```

О тех же запросах с использованием синтаксиса OLE DB см. следующий раздел.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND wwRetrievalMode = 'Delta'
     AND DateTime > 'Jan 13 1999 12:00:30'
     AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 2:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime > 'Jan 13 1999 12:01:00'
      AND DateTime < 'Jan 13 1999 12:10:00'
```

Запрос 3:

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime > 'Jan 13 1999 12:00:30'
      AND DateTime < 'Jan 13 1999 12:01:00'
```

Результаты

Результаты 1: Первая выданная строка является первым правильным изменением после (не включительно) стартового времени (12:00:30):

Date/Time	Value	Quality
1999-01-13 12:01:00.000	1.0	0
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0

(задействовано 9 строк)

Результаты 2: Стартовая дата (12:01:00) влияет на изменение данных, но будет исключена из результатов, потому что используемым оператором является *больше*, а не *больше или равно*.

Date/Time	Value	Quality
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0

(задействовано 8 строк)

Результаты 3: Запрос не обнаруживает изменения значения; соответственно, строки не выдаются:

Date/Time	Value	Quality
-----------	-------	---------

(задействовано 0 строк)

Указание конечной даты с помощью "<="

Если конечная дата указана при помощи <= (меньше или равно), то последняя выданная строка есть последнее правильное изменение до конечной даты включительно. Если конечная дата использует "<=", то дата/время последнего изменения может совпадать с конечной датой. Если время получения значения совпадает с конечной датой, значение будет выдано.

Основной запрос

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND DateTime >  'Jan 13 1999  12:00:30'
     AND DateTime <= 'Jan 13 1999  12:10:00'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
  WHERE TagName = 'SysTimeMin'
     AND wwRetrievalMode = 'Delta'
     AND DateTime >  'Jan 13 1999  12:00:30'
     AND DateTime <= 'Jan 13 1999  12:10:00'
```

Результаты

Заметьте, что в момент конечной даты (12:10:00) запроса существует правильное изменение.

Date/Time	Value	Quality
-----	-----	-----
1999-01-13 12:01:00.000	1.0	0
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0
1999-01-13 12:10:00.000	10.0	0

(задействовано 10 строк)

Указание конечной даты с помощью "<"

Если конечная дата указана при помощи < (меньше), то последняя выданная строка есть последнее правильное изменение до конечной даты (не включительно). Если при указании конечной даты используется "<", то дата/время последнего выданного события будет меньше конечной даты. Если в момент конечной даты происходит событие, оно выдано не будет.

Основной запрос

```
SET wwAnalogCyclic OFF

SELECT DateTime, Value, Quality FROM AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND DateTime > 'Jan 13 1999 12:00:30'
      AND DateTime < 'Jan 13 1999 12:10:00'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND DateTime > 'Jan 13 1999 12:00:30'
      AND DateTime < 'Jan 13 1999 12:10:00'
```

Результаты

Заметьте, что в момент конечного времени запроса (12:10:00) происходит правильное изменение, но оно исключено из результатов.

Date/Time	Value	Quality
-----	-----	-----
1999-01-13 12:01:00.000	1.0	0
1999-01-13 12:02:00.000	2.0	0
1999-01-13 12:03:00.000	3.0	0
1999-01-13 12:04:00.000	4.0	0
1999-01-13 12:05:00.000	5.0	0
1999-01-13 12:06:00.000	6.0	0
1999-01-13 12:07:00.000	7.0	0
1999-01-13 12:08:00.000	8.0	0
1999-01-13 12:09:00.000	9.0	0

(задействовано 9 строк)

Использование операторов сравнения с циклическим методом и счетчиком строк

Расширения временного домена циклических запросов со счетчиком строк выдают набор равно распределенных строк за указанный промежуток времени. В результаты всегда будет входить количество строк, определяемое расширением счетчика строк для каждого тэга в запросе. Разрешение для этих строк подсчитывается путем деления промежутка времени на счетчик строк.

Использование двух операторов равенства

Если отрезок времени указан при помощи $\geq$ и $\leq$, то первая строка выдается точно в момент стартового времени, а последняя -- точно в момент конечного времени. В этом случае разрешение подсчитывается как (конечная дата – стартовая дата) / (счет строк – 1).

Основной запрос

Следующий запрос использует счетчик строк 60, что приводит к разрешения данных в 1 секунду.

```
Set wwRowCount 60

SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND DateTime >= 'Jan 13 1999 12:00:00'
     AND DateTime <= 'Jan 13 1999 12:00:59'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND wwRowCount = 60
     AND DateTime >= 'Jan 13 1999 12:00:00'
     AND DateTime <= 'Jan 13 1999 12:00:59'
```

Результаты

Date/Time	Value
1999-01-13 12:00:00.000	0.0
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0

(задействовано 60 строк)

Использование одного оператора равенства

Если один конец отрезка времени исключается (при использовании > вместо >= или < вместо <=), то провал в разрешении остается в начале (или в конце) набора результатов.

Разрешение подсчитывается как (конечная дата – стартовая дата) / (счетчик строк).

Строка, относящаяся ко времени заданному при помощи оператора < (или >), не выдается.

Основные запросы

Запрос 1: Следующий запрос использует счетчик строк 60, что приводит к разрешению данных в 1 секунду. Стартовое время установлено на >=.

```
Set wwRowCount 60
```

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND wwRowCount = 60
AND DateTime >= 'Jan 13 1999 12:00:00'
AND DateTime < 'Jan 13 1999 12:01:00'
```

Запрос 2: Этот запрос также использует счетчик строк 60, что приводит к разрешению данных в 1 секунду. Время окончания установлено на <=.

```
Set wwRowCount 60
```

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND wwRowCount = 60
AND DateTime > 'Jan 13 1999 12:00:00'
AND DateTime <= 'Jan 13 1999 12:01:00'
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND wwRowCount = 60
AND DateTime >= 'Jan 13 1999 12:00:00'
AND DateTime < 'Jan 13 1999 12:01:00'
```

Запрос 2:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND wwRowCount = 60
AND DateTime < 'Jan 13 1999 12:00:00'
AND DateTime >= 'Jan 13 1999 12:01:00'
```

Результаты

Запрос 1:

Date/Time	Value
1999-01-13 12:00:00.000	0.0
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0

(задействовано 60 строк)

Запрос 2:

Date/Time	Value
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0
1999-01-13 12:01:00.000	0.0

(задействовано 60 строк)

Неиспользование операторов равенства

Если исключаются оба конца отрезка времени (при использовании $>$ и $<$), то провал в разрешении остается и в начале, и в конце набора результатов.

Разрешение подсчитывается как (конечная дата – стартовая дата) / (счетчик строк + 1).

Строки, относящиеся к стартовому времени и конечному времени, не выдаются.

Основной запрос

```
Set wwRowCount 60
```

```
SELECT DateTime, Value FROM AnalogHistory
WHERE TagName = 'SysTimeMin'
AND DateTime > 'Jan 13 1999 12:00:00'
AND DateTime < 'Jan 13 1999 12:01:01'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND wwRowCount = 60
      AND DateTime >  'Jan 13 1999  12:00:00'
      AND DateTime <  'Jan 13 1999  12:01:01'
```

Результаты

Date/Time	Value
-----	-----
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0
1999-01-13 12:01:00.000	0.0

(задействовано 60 строк)

Использование операторов сравнения с циклическим методом и разрешением

Циклические запросы, использующие операторы сравнения и расширение временного домена разрешения, выдают набор равно распределенных строк за указанный промежуток времени. Разрешение для этих строк указывается в запросе.

Использование двух операторов равенства

Если отрезок времени указан при помощи $\geq$ и $\leq$, то первая строка выдается точно в момент стартового времени, а последняя -- точно в момент конечного времени. Если при указании конечной даты используется " $\leq$ ", то последняя выданная строка будет иметь дату/время, точно совпадающую с конечной датой.

Основной запрос

Этот запрос устанавливает разрешение в 1 секунду.

```
SET wwResolution 1000

SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND DateTime >= 'Jan 13 1999 12:00:00'
     AND DateTime <= 'Jan 13 1999 12:01:00'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND wwResolution = 1000
     AND DateTime >= 'Jan 13 1999 12:00:00'
     AND DateTime <= 'Jan 13 1999 12:01:00'
```

Результаты

Date/Time	Value
1999-01-13 12:00:00.000	0.0
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0
1999-01-13 12:01:00.000	0.0

(задействована 61 строка)

Использование одного оператора равенства

Если один конец отрезка времени исключается (при использовании > вместо >= или < вместо <=), то провал в разрешении остается или в начале, или в конце набора результатов. Если при указании стартовой даты используется >, то у первой выданной строки будет отметка времени (стартовая дата + разрешение). Если при указании конечной даты используется "<", то последняя выданная строка будет относиться к дате/времени (конечная дата – разрешение).

Строка, относящаяся ко времени заданному при помощи оператора < (или >), не выдается.

Основные запросы

Запрос 1: Следующий запрос использует разрешение в 1000, что приводит к разрешению данных в 1 секунду. Стартовое время установлено на >=.

```
SET wwResolution 1000

SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Jan 13 1999 12:00:00'
      AND DateTime < 'Jan 13 1999 12:01:00'
```

Запрос 2: Этот запрос также использует разрешение строк в 1000, что приводит к разрешению данных в 1 секунду. Стартовое время установлено на <=.

```
SET wwResolution 1000

SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime > 'Jan 13 1999 12:00:00'
      AND DateTime <= 'Jan 13 1999 12:01:00'
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND wwResolution = 1000
      AND DateTime >= 'Jan 13 1999 12:00:00'
      AND DateTime < 'Jan 13 1999 12:01:00'
```

Запрос 2:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND wwResolution = 1000
      AND DateTime > 'Jan 13 1999 12:00:00'
      AND DateTime <= 'Jan 13 1999 12:01:00'
```

Результаты

Запрос 1:

Date/Time	Value
-----	-----
1999-01-13 12:00:00.000	0.0
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:55.000	55.0
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0

(задействовано 60 строк)

Запрос 2:

Date/Time	Value
-----	-----
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0
1999-01-13 12:01:00.000	0.0

(задействовано 60 строк)

Неиспользование операторов равенства

Если исключаются оба конца отрезка времени (при использовании > и <), то провал в *разрешения* остается и в начале, и в конце набора результатов.

Строки, совпадающие со стартовым временем и конечным временем, не выдаются.

Основной запрос

Запрос 1: Следующий запрос использует разрешение в 1000, что приводит к разрешения данных в 1 секунду.

```
SET wwResolution 1000
```

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime > 'Jan 13 1999 12:00:00'
      AND DateTime < 'Jan 13 1999 12:01:01'
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND wwResolution = 1000
      AND DateTime > 'Jan 13 1999 12:00:00'
      AND DateTime < 'Jan 13 1999 12:01:01'
```

Результаты

Date/Time	Value
-----	-----
1999-01-13 12:00:01.000	1.0
1999-01-13 12:00:02.000	2.0
1999-01-13 12:00:03.000	3.0
1999-01-13 12:00:04.000	4.0
...	
1999-01-13 12:00:56.000	56.0
1999-01-13 12:00:57.000	57.0
1999-01-13 12:00:58.000	58.0
1999-01-13 12:00:59.000	59.0
1999-01-13 12:01:00.000	0.0

(задействовано 60 строк)

Распознавание событий

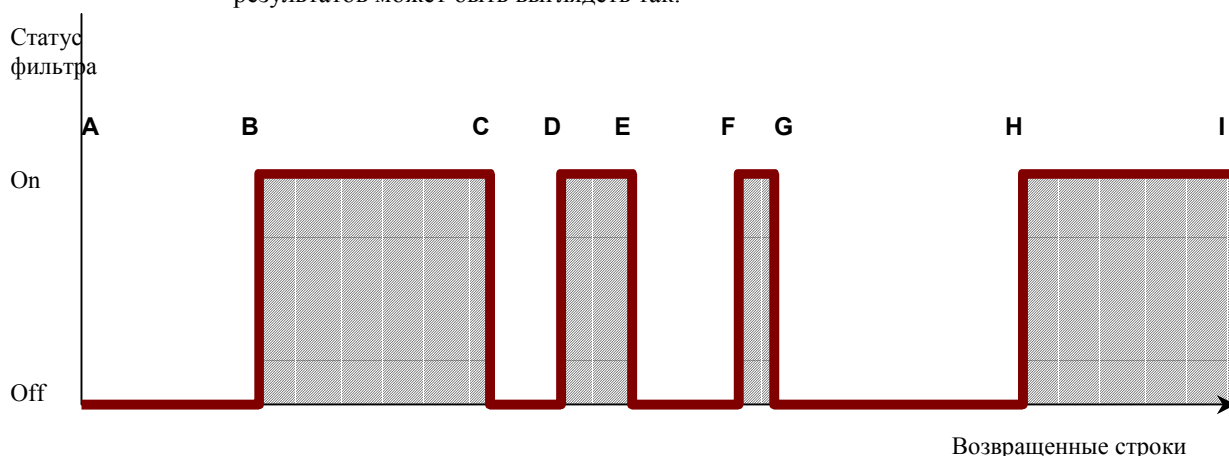
Событие есть набор атрибутов, описывающих момент, когда значения архивных тэгов IndustrialSQL Server удовлетворяют набору критериев. Атрибуты включают в себя дату и время происхождения события и критерии, которым оно удовлетворяет. *Тэгом события* называется определение события в системе. Например, если требуется зарегистрировать событие, когда температура танкера достигла 100 градусов, следует определить тэг события и назвать его "TankAt100".

При распознавании событий полезным является сужение количества строк в наборе результатов, где удовлетворение критериям изменялось с правды на ложь и наоборот. Например, может понадобиться узнать, когда уровень танкера понизится до 5 футов. Выражение WHERE в запросе для этого примера может быть `TANKLEVEL > 5`. Пока уровень танкера не превышает 5 футов, выражение WHERE не выдает истину. Только когда уровень пересекает абстрактную линию между неудовлетворением и удовлетворением критерия в выражении WHERE, событие действительно происходит. Абстрактная "линия", на которой происходит изменение называется *краем*.

За какой-то период времени может неоднократно получиться так, что критерии в выражении WHERE пересекают край от "удовлетворенных" на "неудовлетворенные" и наоборот. Значения с любой стороны этого края могут быть засечены при помощи расширения определения края. Определение края имеет четыре возможных параметра: *нет*, *ведущие*, *ведомые* или *оба*. В зависимости от используемых параметров будут получены разные результаты.

Определение края для аналоговых тэгов

Например, работа выражения WHERE при обработке с его помощью набора результатов может быть выглядеть так:



Здесь набор строк, выданных запросом, идет от A до I, хотя выражение WHERE фильтрует строки от **A-B**, **C-D**, **E-F** и **G-H**, выдавая только строки, совпадающие с теневыми областями графика.

Края фильтра WHERE — это точки, где выражение переключается с Off на On и наоборот. *Ведущие края* есть переключения с Off на On (**B, D, F и H**). *Ведомые края* есть точки переключения с On на Off (**C, E и G**).

Следующая таблица показывает строки, которые выдаются при различных установках определения края:

Определение края	Выдано строк
Ведущие	Только строки B, D, F и H .
Ведомые	Только строки C, E и G .
Оба	Только строки B, C, D, E, F, G и H .
Нет	Все строки от B-C, D-E, F-G и H-I .

Следующие запросы выбирают все значения "SysTimeSec", большие чем 50 из таблицы *AnalogHistory* между 10:00 и 10:02 a.m. 2 декабря 1998.

Определение края не указано. Этот запрос является основой для наблюдения, как можно управлять результатами запроса при использовании расширения определения края.

Основной запрос

```
SET wwSetResolution 2000

SET wwEdgeDetection Нет

SELECT DateTime, Value FROM AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND value >= 50
      AND wwResolution = 2000
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND Value >= 50
      AND wwResolution = 2000
      AND wwEdgeDetection = 'None'
```

Результаты

Результаты всех запросов:

Date/Time	Value
-----	-----
1998-12-02 10:00:50	50
1998-12-02 10:00:52	52
1998-12-02 10:00:54	54
1998-12-02 10:00:56	56
1998-12-02 10:00:58	58
1998-12-02 10:01:50	50
1998-12-02 10:01:52	52
1998-12-02 10:01:54	54
1998-12-02 10:01:56	56
1998-12-02 10:01:58	58

Указание ведущего определения края

Если указан параметр *Ведущее (Leading)* в расширении временного домена определения края, в результаты войдут только те строки, которые первыми соответствуют критериям в выражении WHERE (выдана истина) после строки, этому критерию не соответствовавшей (выдана ложь).

Основной запрос

```
SET wwSetResolution 2000

SET wwEdgeDetection Leading

SELECT DateTime, Value FROM AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND DateTime >= 'Dec 2 1998 10:00:00'
     AND DateTime <= 'Dec 2 1998 10:02:00'
     AND Value >= 50
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы:

```
SELECT DateTime, Value FROM v_AnalogHistory
  WHERE TagName = 'SysTimeSec'
     AND DateTime >= 'Dec 2 1998 10:00:00'
     AND DateTime <= 'Dec 2 1998 10:02:00'
     AND Value >= 50
     AND wwResolution = 2000
     AND wwEdgeDetection = 'Leading'
```

Результаты

Все три запроса выдают только те два значения, которые больше или равны чем 50 для указанного периода времени. Заметьте, что несмотря на то, что первоначальный вопрос выдал 10 строк, определение края выдает только *первую* строку, записанную после выдачи критерием события истины.

Date/Time	Value
-----	-----
1998-12-02 10:00:50	50
1998-12-02 10:01:50	50

Сравните эти результаты с тем же запросом без определения края, как показано в разделе "Определение края для аналоговых тэгов" ранее в этой главе.

Указание ведомого определения края

Если в расширении определения края указан параметр *Ведомый (Trailing)*, то единственными строками в результатах будут те, которые не совпали с критериями в выражении WHERE (выдана ложь) после строки, которая с этими критериями совпала (выдана истина).

Основной запрос

```
SET wwSetResolution 2000
SET wwEdgeDetection Trailing

SELECT DateTime, Value FROM AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND Value >= 50
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
      AND DateTime >= 'Dec 2 1998 10:00:00'
      AND DateTime <= 'Dec 2 1998 10:02:00'
      AND Value >= 50
      AND wwResolution = 2000
      AND wwEdgeDetection = 'Trailing'
```

Результаты

Три запроса выдают только два значения, которые первыми не совпали с критериями в выражении WHERE для указанного периода времени. Заметьте, что несмотря на то, что первоначальный запрос выдал 10 записанных строк для каждого значения, определение края выдает только *первую* строку, записанную после выдачи критерием события лжи.

Date/Time	Value
-----	-----
1998-12-02 10:01:00	00
1998-12-02 10:02:00	00

Сравните эти результаты с тем же запросом не используя определения края, как показано в разделе "Определение края для аналоговых тэгов" ранее в этой главе.

Указание и ведущего, и ведомого определения края

Если указан параметр расширения определения края *Оба (Both)*, то выдаются все строки, удовлетворяющие ведущим и ведомым критериям.

Основной запрос

```
SET wwSetResolution 2000
SET wwEdgeDetection Both

SELECT DateTime, Value FROM AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= 'Dec 2 1998 10:00:00'
AND DateTime <= 'Dec 2 1998 10:02:00'
AND Value >= 50
```

Тот же запрос с синтаксисом OLE DB смотри в следующем разделе.

Запрос к поставщику данных OLE DB

Следующий запрос использует представление удаленной таблицы:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= 'Dec 2 1998 10:00:00'
AND DateTime <= 'Dec 2 1998 10:02:00'
AND Value >= 50
AND wwResolution = 2000
AND wwEdgeDetection = 'Both'
```

Результаты

Заметьте, что значение первой строки исходного запроса выдано среди результатов.

Date/Time	Value
-----	-----
1998-12-02 10:00:50	50
1998-12-02 10:01:00	00
1998-12-02 10:01:50	50
1998-12-02 10:02:00	00

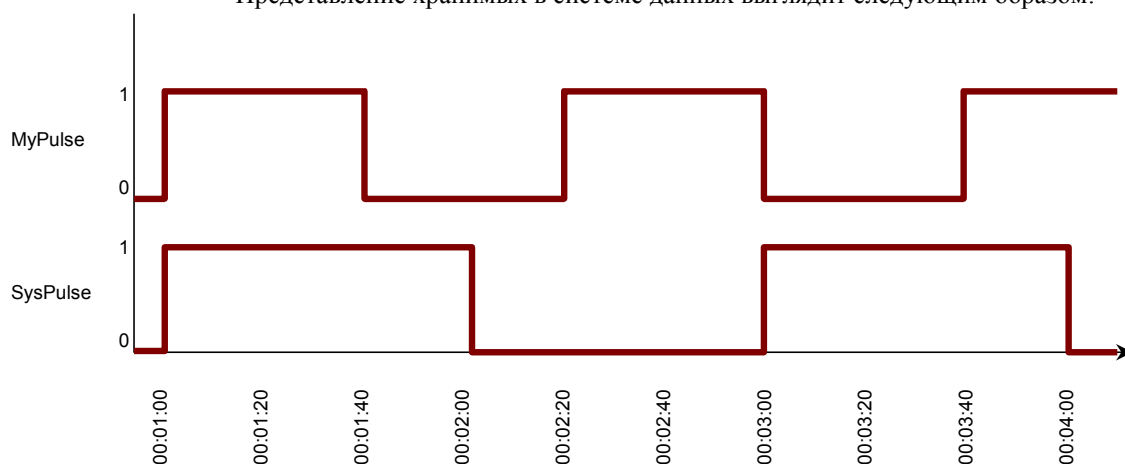
Сравните эти результаты с тем же запросом не используя определения края, как показано в разделе "Определение края для аналоговых тэгов" ранее в этой главе.

Определение края для дискретных тэгов

Определение края для дискретных тэгов отличается от определения края для аналоговых тэгов. Например, предположим, что сохранены следующие дискретные тэги.

Тэг	Описание
SysPulse	Колеблется между 1 и 0 каждую минуту.
MyPulse	Колеблется между 1 и 0 каждые 40 секунд.

Представление хранимых в системе данных выглядит следующим образом:



Следующие запросы выбирают значения "SysPulse" и "MyPulse" со значением 1 (Включено) из таблиц *DiscreteHistory* и *DiscreteWideHistory* между 12:59 и 1:04 a.m. 8 декабря 1998. Определение края не указано. Этот запрос является основой для наблюдения, как можно управлять результатами запроса при использовании расширения определения края.

Основной запрос

Запрос 1: Запрос к таблице *DiscreteHistory*.

```
SET wwSetDiscreteCyclic OFF
SET wwEdgeDetection None
SELECT DateTime, TagName, Value FROM DiscreteHistory
  WHERE TagName IN ('SysPulse','MyPulse')
     AND DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND Value = 1
```

Запрос 2: Запрос к таблице *DiscreteWideHistory*.

```
SET wwSetDiscreteCyclic OFF
SET wwEdgeDetection None
SELECT DateTime, TagName, Value FROM DiscreteWideHistory
  WHERE TagName IN ('SysPulse','MyPulse')
     AND DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND Value = 1
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленной таблицы:

Запрос 1:

```
SELECT DateTime, TagName, Value
FROM v_DiscreteHistory
WHERE TagName IN ('SysPulse','MyPulse')
      AND DateTime > 'Dec 8 1998 00:59:00'
      AND DateTime <= 'Dec 8 1998 01:04:00'
      AND wwRetrievalMode = 'Delta'
      AND Value = 1
      AND wwEdgeDetection = 'None'
```

Запрос 2:

```
SELECT DateTime, SysPulse, MyPulse
FROM v_DiscreteWideHistory
WHERE DateTime > 'Dec 8 1998 00:59:00'
      AND DateTime <= 'Dec 8 1998 01:04:00'
      AND wwRetrievalMode = 'Delta'
      AND SysPulse = 1
      AND MyPulse = 1
      AND wwEdgeDetection = 'None'
```

Результаты

Запрос 1:

DateTime	TagName	Value
1998-12-08 00:01:00	SysPulse	1
1998-12-08 00:01:00	MyPulse	1
1998-12-08 00:02:20	MyPulse	1
1998-12-08 00:03:00	SysPulse	1
1998-12-08 00:03:40	MyPulse	1

Запрос 2:

DateTime	SysPulse	MyPulse
1998-12-08 00:01:00	1	1

Указание ведущего определения края

Если в расширении временного домена определения края указан параметр *Ведущее (Leading)*, в результаты войдут только те строки, которые первыми соответствуют критериям в выражении WHERE (выдана истина) после строки, не соответствовавшей этому критерию (выдана ложь).

Основные запросы

Запрос 1: Для запроса к таблице *DiscreteHistory*, если в выражении WHERE указана выдача только дискретных значений равных 1 (Включено), то применение ведущего определения края не изменяет результаты.

```
SET wwDiscreteCyclic OFF
SET wwEdgeDetection Leading
SELECT DateTime FROM DiscreteHistory
  WHERE TagName IN ('SysPulse','MyPulse')
     AND DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND Value = 1
```

Запрос 2: Для запроса к таблице *DiscreteWideHistory* применение ведущего определения края требует, чтобы условия были оценены как истина, если оба значения равны 1 (Включено).

```
SET wwDiscreteCyclic OFF
SET wwEdgeDetection Leading
SELECT DateTime FROM DiscreteWideHistory
  WHERE TagName IN ('SysPulse','MyPulse')
     AND DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND Value = 1
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют режимы просмотра удаленных таблиц:

Запрос 1:

```
SELECT DateTime FROM v_DiscreteHistory
  WHERE TagName IN ('SysPulse','MyPulse')
     AND DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND wwRetrievalMode = 'Delta'
     AND Value = 1
     AND wwEdgeDetection = 'Leading'
```

Запрос 2:

```
SELECT DateTime, SysPulse, MyPulse
  FROM v_DiscreteWideHistory
  WHERE DateTime > 'Dec 8 1998 00:59:00'
     AND DateTime <= 'Dec 8 1998 01:04:00'
     AND wwRetrievalMode = 'Delta'
     AND SysPulse = 1
     AND MyPulse = 1
     AND wwEdgeDetection = 'Leading'
```

Результаты

Запрос 1: Для запроса к таблице *DiscreteHistory* применение ведущего определения края не изменяет результаты:

```
DateTime
-----
1998-12-08 00:01:00
1998-12-08 00:01:00
1998-12-08 00:02:20
1998-12-08 00:03:00
1998-12-08 00:03:40
```

Запрос 2: Для запроса к таблице *DiscreteWideHistory* применение ведущего определения края требует, чтобы условия оценивались только как истина, если оба значения равны 1 (Включено).

```
DateTime
-----
1998-12-08 00:01:00
1998-12-08 00:03:40
```

Сравните эти результаты с тем же запросом не используя определения края, как показано в разделе "Распознавание событий для дискретных тэгов" ранее в этой главе.

Если взглянуть на диаграмму ранее в этом разделе, можно подумать, что может быть возвращена строка от 00:03:00, но поскольку в этот момент оба тэга меняют свои значения, эти значения не возвращаются. При нормальной работе два тэга обычно не изменяются в один и тот же момент.

Указание ведомого определения края

Если в расширении определения края указан параметр *Следящее (Trailing)*, то единственными строками в результатах будут те, которые не совпали с критериями в выражении WHERE (выдана ложь) после строки, которая с этими критериями совпала (выдана истина).

Основные запросы

Запрос 1: Для запроса к таблице *DiscreteHistory*, если критерий выражения WHERE требует выдавать только дискретные значения равные 1 (Включено), то применение ведущего определения края аналогично реверсированию выражения WHERE (как если бы применялось значение = 0).

```
SET wwDiscreteCyclic OFF

SET wwEdgeDetection Trailing

SELECT DateTime FROM DiscreteHistory
   WHERE TagName IN ('SysPulse','MyPulse')
      AND DateTime >  'Dec 8 1998  00:59:00'
      AND DateTime <= 'Dec 8 1998  01:04:00'
      AND Value = 1
```

Запрос 2: Для запроса к таблице *DiscreteWideHistory* применение ведущего определения края выдаст рамки, в которых условия определяются как истина (одно из значений равняется 0).

```
SET wwDiscreteCyclic OFF

SET wwEdgeDetection Trailing

SELECT DateTime FROM DiscreteWideHistory
WHERE TagName IN ('SysPulse', 'MyPulse')
AND DateTime > 'Dec 8 1998 00:59:00'
AND DateTime <= 'Dec 8 1998 01:04:00'
AND Value = 1
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют режимы просмотра удаленных таблиц.

Запрос 1:

```
SELECT DateTime FROM v_DiscreteHistory
WHERE TagName IN ('SysPulse', 'MyPulse')
AND DateTime > 'Dec 8 1998 00:59:00'
AND DateTime <= 'Dec 8 1998 01:04:00'
AND wwRetrievalMode = 'Delta'
AND Value = 1
AND wwEdgeDetection = 'Trailing'
```

Запрос 2:

```
SELECT DateTime, SysPulse, MyPulse
FROM v_DiscreteWideHistory
WHERE DateTime > 'Dec 8 1998 00:59:00'
AND DateTime <= 'Dec 8 1998 01:04:00'
AND wwRetrievalMode = 'Delta'
AND SysPulse = 1
AND MyPulse = 1
AND wwEdgeDetection = 'Leading'
```

Результаты

Запрос 1: Для запроса к таблице *DiscreteHistory* применение ведущего определения края аналогично реверсированию выражения WHERE (как если бы применялось значение = 0):

```
DateTime
-----
1998-12-08 00:01:40
1998-12-08 00:02:00
1998-12-08 00:03:00
1998-12-08 00:04:00
```

Запрос 2: Для запроса к таблице *DiscreteWideHistory* применение ведущего определения края выдаст рамки, в которых условие определяется как истина:

```
DateTime
-----
1998-12-08 00:01:40
1998-12-08 00:04:00
```

Сравните эти результаты с тем же запросом без определения края, как показано в разделе "Распознавание событий" ранее в этой главе.

Если взглянуть на диаграмму ранее в этом разделе, можно подумать, что строка могла быть выдана в 00:03:00, однако поскольку оба тэга изменились точно в тот момент, их значения не выдаются. При нормальной работе два тэга обычно не изменяются в один и тот же момент.

Указание и ведущего, и ведомого определения края

Если указан параметр расширения определения края *Оба (Both)*, то выдаются все строки, удовлетворяющие ведущим и ведомым критериям.

Основные запросы

Запрос 1: Ведущее и ведомое определения края в таблице *DiscreteHistory*.

```
SET wwDiscreteCyclic OFF
SET wwEdgeDetection Both
SELECT DateTime FROM DiscreteHistory
  WHERE TagName IN ('SysPulse','MyPulse')
         AND DateTime > 'Dec 8 1998 00:59:00'
         AND DateTime <= 'Dec 8 1998 01:04:00'
         AND Value = 1
```

Запрос 2: Ведущее и ведомое определения края в таблице *DiscreteWideHistory*.

```
SET wwDiscreteCyclic OFF
SET wwEdgeDetection Both
SELECT DateTime FROM DiscreteWideHistory
  WHERE TagName IN ('SysPulse','MyPulse')
         AND DateTime > 'Dec 8 1998 00:59:00'
         AND DateTime <= 'Dec 8 1998 01:04:00'
         AND Value = 1
```

Те же запросы с использованием синтаксиса OLE DB см. в следующем разделе.

Запросы к поставщику данных OLE DB

Следующие запросы используют представление удаленных таблиц.

Запрос 1:

```
SELECT DateTime FROM v_DiscreteHistory
  WHERE TagName IN ('SysPulse','MyPulse')
         AND DateTime > 'Dec 8 1998 00:59:00'
         AND DateTime <= 'Dec 8 1998 01:04:00'
         AND wwRetrievalMode = 'Delta'
         AND Value = 1
         AND wwEdgeDetection = 'Both'
```

Запрос 2:

```
SELECT DateTime FROM v_DiscreteWideHistory
  WHERE TagName IN ('SysPulse','MyPulse')
         AND DateTime > 'Dec 8 1998 00:59:00'
         AND DateTime <= 'Dec 8 1998 01:04:00'
         AND wwRetrievalMode = 'Delta'
         AND Value = 1
         AND wwEdgeDetection = 'Both'
```

Результаты

Запрос 1: Результаты запроса таблицы *DiscreteHistory*.

```
DateTime
-----
1998-12-08 00:01:00
1998-12-08 00:01:40
1998-12-08 00:02:00
1998-12-08 00:02:20
1998-12-08 00:03:00
1998-12-08 00:03:40
1998-12-08 00:04:00
```

Запрос 2: Результаты запроса *DiscreteWideHistory*.

```
DateTime
-----
1998-12-08 00:01:00
1998-12-08 00:01:40
1998-12-08 00:03:40
1998-12-08 00:04:00
```

Сравните эти результаты с тем же запросом без определения края, как показано в разделе "Распознавание событий" ранее в этой главе.

Связи между таблицами Manual History и Dynamic History

Использование поставщика данных OLE DB позволяет создавать связи между введенными вручную архивными данными (в Runtime.dbo.ManualAnalogHistory) и динамическими архивными данными (в INSQL.Runtime.dbo.AnalogHistory).

Для создания связи следует включить столбец с названием *wwRetrievalMode* в таблицу *ManualAnalogHistory*.

Следующий сценарий использует представление поставщика данных OLE DB для удаленной таблицы *AnalogHistory*.

```
SELECT DateTime, TagName, Value, wwRetrievalMode
FROM v_AnalogHistory
```

```
UNION
```

```
SELECT DateTime, TagName, Value, wwRetrievalMode='Delta'
FROM ManualAnalogHistory
```

Можно создать собственное представление для SQL Server, которое будет заключать в себе объединенные данные:

```
CREATE VIEW vTestView AS
```

```
SELECT DateTime, TagName, Value, wwRetrievalMode
FROM InSQL.runtime.dbo.AnalogHistory
```

```
UNION
```

```
SELECT DateTime, TagName, Value, wwRetrievalMode='Delta'
FROM ManualAnalogHistory
```

После создания нового представления и создаваемые вручную и динамические архивные данные могут быть запрошены вместе с помощью обычного сценария. Например, следующий сценарий является стандартным сценарием, сортирующим строки по дате/времени:

```
SELECT * FROM vTestView
WHERE Datetime >= '1999-01-07 15:28'
AND Datetime <= '1999-01-07 15:50'
AND wwRetrievalMode = 'Delta'
AND TagName = 'SysTimeSec'
ORDER BY DateTime
```

Следующий сценарий группирует полученные данные по дате/времени:

```
SELECT Дата и Время FROM vTestView
WHERE Дата/Время >= '1999-01-07 15:28'
AND Дата/Время <= '1999-01-07 15:50'
AND wwRetrievalMode = 'Delta'
AND Имя Тэга = 'SysTimeSec'
GROUP BY Дата и Время
```

Следующий сценарий проводит агрегацию сводки над данными и сортирует результаты по дате/времени:

```
SELECT Datepart(minute, DateTime), sum(Value) FROM vTestView
WHERE Datetime >= '1999-01-07 15:28'
AND Datetime <= '1999-01-07 15:50'
AND wwRetrievalMode = 'Delta'
AND TagName = 'SysTimeSec'
GROUP BY Datepart(minute, DateTime)
```

Использование курсоров сервера

Курсоры являются мощной возможностью SQL Server. Они позволяют перемещаться по записям, полученным после выполнения запроса.

Подробнее о курсорах см. в документацию по Microsoft SQL Server.

Поставщик данных InSQL OLE DB поставляется с курсорами сервера.

Курсоры могут использоваться для создания связей, которые иначе невозможны. Они используются для объединения даты/времени из любого источника с датой/временем архивных таблиц.

Запрос к поставщику данных OLE DB

Следующий запрос является образцом использования курсоров сервера. Этот запрос:

- Выбирает все события из таблицы *EventHistory*.
- Показывает "снимок" трех тэгов во время каждого события.
- Показывает тэг события и связанное с ним значение символа.

Этот запрос может быть легко превращен в хранимую процедуру.

Следующий запрос использует наименование из четырех частей для объектов OLE DB.

```
DECLARE @DateValue DateTime
DECLARE @EventTag varchar(33)
DECLARE @EventKey int
DECLARE @Qry1 nvarchar(500)
DECLARE @Qry2 nvarchar(500)
DECLARE @Qry3 nvarchar(500)

SELECT @Qry1 = N'SELECT EventTag = @EventTag,
    EventKey = @EventKey, DateTime, TagName, Value, Quality
FROM INSQL.Runtime.dbo.AnalogHistory
    WHERE TagName IN ('SysTimeSec', 'SysTimeMin',
        'SysTimeHour')
    AND DateTime = ''
SELECT @Qry2 = N''
SELECT @Qry3 = @Qry1 + convert(varchar, @DateValue, 121) +
@Qry2

DECLARE Hist_Cursor CURSOR FOR
SELECT DateTime, TagName, EventLogKey FROM
Runtime.dbo.EventHistory
OPEN Hist_Cursor
FETCH NEXT FROM Hist_Cursor INTO @DateValue, @EventTag,
@EventKey
WHILE @@FETCH_STATUS = 0
BEGIN
    SELECT @Qry3 = @Qry1 + convert(varchar, @DateValue, 121) +
        @Qry2
    --PRINT @Qry3
    EXEC sp_executesql @Qry3, N'@EventTag varchar(33),
        @EventKey int', @EventTag, @EventKey

    FETCH NEXT FROM Hist_Cursor INTO @DateValue, @EventTag,
        @EventKey
END
CLOSE Hist_Cursor
DEALLOCATE Hist_Cursor
```

Результаты

EventTag	EventKey Quality	DateTime	TagName	Value
-----	-----	-----	-----	-----
-				
SysStatusEvent	3	1999-01-12 13:00:27.000	SYSTIMESEC	27.0 0
SysStatusEvent	3	1999-01-12 13:00:27.000	SYSTIMEMIN	0.0 0
SysStatusEvent	3	1999-01-12 13:00:27.000	SYSTIMEHOUR	13.0 0

(задействовано 3 строки)

EventTag	EventKey Quality	DateTime	TagName	Value
-----	-----	-----	-----	-----
-				
SysStatusEvent	4	1999-01-12 14:00:28.000	SYSTIMESEC	28.0 0
SysStatusEvent	4	1999-01-12 14:00:28.000	SYSTIMEMIN	0.0 0
SysStatusEvent	4	1999-01-12 14:00:28.000	SYSTIMEHOUR	14.0 0

(задействовано 3 строки)

Неподдерживаемый синтаксис SQL для поставщика данных INSQLE DB

В синтаксисе SQL существуют ряд ограничений для поставщика данных OLE DB. Все они делятся на две основные категории:

- Синтаксис, не поддерживаемый запросами из четырех частей
- Синтаксис, не поддерживаемый OPENQUERY

Далее приводятся примеры обоих типов неподдерживаемого синтаксиса.

Синтаксис, неподдерживаемый запросами из четырех частей

В случае с запросами из четырех частей, сервер SQL создает выражение, посылаемое поставщику данных INSQLE DB, из выражения, созданного пользователем. Иногда это созданное выражение неправильно, слишком сложно или не имеет некоторых частей выражения WHERE, требуемых поставщиком INSQLE DB для выдачи данных.

Обычное сообщение об ошибке при использовании неподдерживаемого синтаксиса выглядит следующим образом:

Server: Msg 7320, Level 16, State 2, Line 1

Could not execute query against OLE DB provider 'INSQLE'.

[поставщик данных OLE/DB выдал сообщение: InSQL не получил выражения WHERE от сервера SQL. Если получено такое сообщение, обратитесь к документации InSQL OLE DB]

Арифметические операторы даты

Запросы, прямо использующие арифметику даты, не поддерживаются. Например, не поддерживается следующий запрос:

```
SELECT DateTime, TagName, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= dateadd(Minute, -10, GetDate())
AND DateTime <= GetDate()
```

Решение — использовать запрос с параметрами, как показано далее:

```
DECLARE @StartDate DateTime
DECLARE @EndDate DateTime
SET @StartDate = dateadd(Minute, -10, GetDate())
SET @EndDate = GetDate()

SELECT DateTime, TagName, Value, Quality FROM v_AnalogHistory
WHERE TagName = 'SysTimeSec'
AND DateTime >= @StartDate
AND DateTime <= @EndDate
```

Широкие таблицы

Таблицы *AnalogWideHistory* и *DiscreteWideHistory* поддерживаются только при использовании синтаксиса OPENQUERY.

Ограничения для выражения OR

Выражение OR не используется с критерием значения. Например, следующий запрос не выдаст никаких данных:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND wwRetrievalMode = 'Delta'
      AND (Value < 10
           OR Value > 50)
```

Решение — использовать UNION ALL с двумя запросами так, как указано далее:

```
SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND Value < 10
      AND wwRetrievalMode = 'Delta'

UNION ALL

SELECT DateTime, Value FROM v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND Value > 50
      AND wwRetrievalMode = 'Delta'
```

CONVERT в выражении WHERE

Выражение WHERE не может содержать выражение CONVERT в запросе из четырех частей. Например, следующий запрос не сработает:

```
Select * from v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND CONVERT(int, Value) = 0
```

Оператор остатка в выражении WHERE

Оператор остатка (%) не используется для выражения WHERE в запросе из четырех частей. Например, не будет выполнен следующий запрос:

```
Select * from v_AnalogHistory
WHERE TagName = 'SysTimeMin'
      AND CONVERT(int, Value) % 3 = 0
```

Синтаксис, не поддерживаемый в OPENQUERY

В случае с OPENQUERY, сервер SQL посылает неизменную цитату поставщику данных INSQL OLE DB. Следовательно, поддерживается только тот синтаксис, который может распознать поставщик данных INSQL OLE DB.

Связи в OPENQUERY

Связи не поддерживаются в одиночном выражении OPENQUERY. Например, следующий запрос содержит неточную связь между *Tag* и *AnalogLive*, что приведет к неудаче:

```
SELECT * FROM OPENQUERY(INSQL, "
    SELECT v.DateTime, v.TagName, v.Value, t.Description
    FROM Tag t, AnalogLive v
    WHERE t.TagName LIKE '%Date%'
    AND v.TagName = t.TagName
")
```

Решением может быть помещение связи вне OPENQUERY. Например:

```
SELECT v.DateTime, v.TagName, v.Value, t.Description
FROM OPENQUERY(INSQL, "
    SELECT DateTime, TagName, Value FROM AnalogLive
    WHERE TagName LIKE '%Date%'
") v, Tag t
WHERE v.TagName = t.TagName
```

Точные связи в OPENQUERY также не поддерживаются. Например, следующий запрос не сработает:

```
SELECT * FROM OPENQUERY(INSQL, "
    SELECT v.DateTime, v.TagName, v.Value, e.Unit FROM
    AnalogLive v
    JOIN AnalogTag t ON v.TagName = t.TagName
    JOIN EngineeringUnit e ON t.EUKey = e.EUKey
    WHERE v.TagName LIKE '%Date%'
")
```

Решением может быть помещение связи вне OPENQUERY. Например:

```
SELECT v.DateTime, v.TagName, v.Value, e.Unit FROM
OPENQUERY(INSQL, "
    SELECT DateTime, TagName, Value FROM AnalogLive
    WHERE TagName LIKE '%Date%'
") v
JOIN AnalogTag t ON v.TagName = t.TagName
JOIN EngineeringUnit e ON t.EUKey = e.EUKey
ORDER BY t.TagName
```

Другим решением является использование синтаксиса из четырех частей, если есть такая возможность. В случае предыдущего запроса, синтаксис будет следующим:

```
SELECT v.DateTime, v.TagName, v.Value, e.Unit FROM
v_AnalogLive v
JOIN AnalogTag t ON v.TagName = t.TagName
JOIN EngineeringUnit e ON t.EUKey = e.EUKey
WHERE v.TagName LIKE '%Date%'
ORDER BY t.TagName
```

П Р И Л О Ж Е Н И Е А

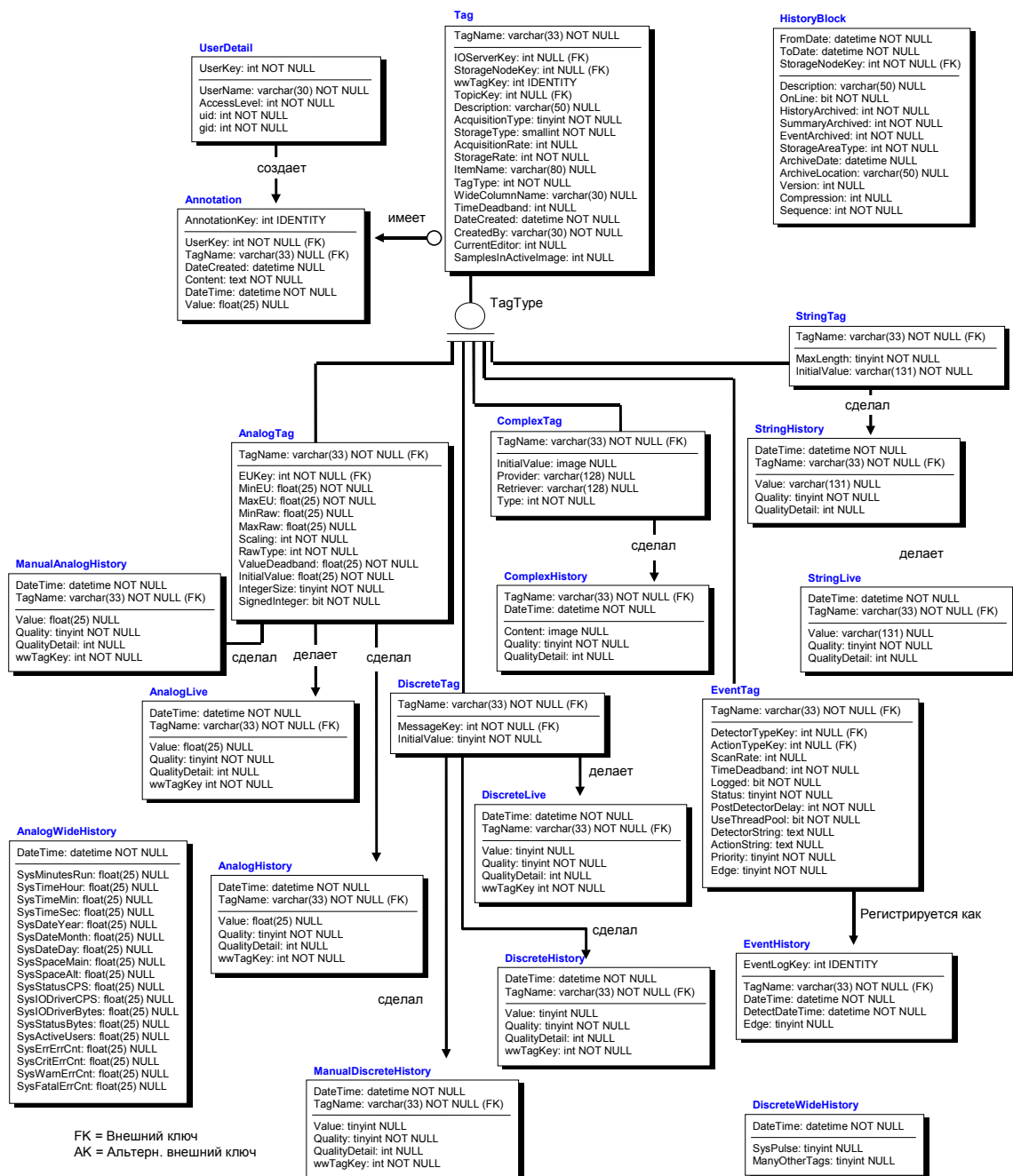
Диаграммы таблиц

В этом приложении содержатся диаграммы всех таблиц схемы IndustrialSQL Server.

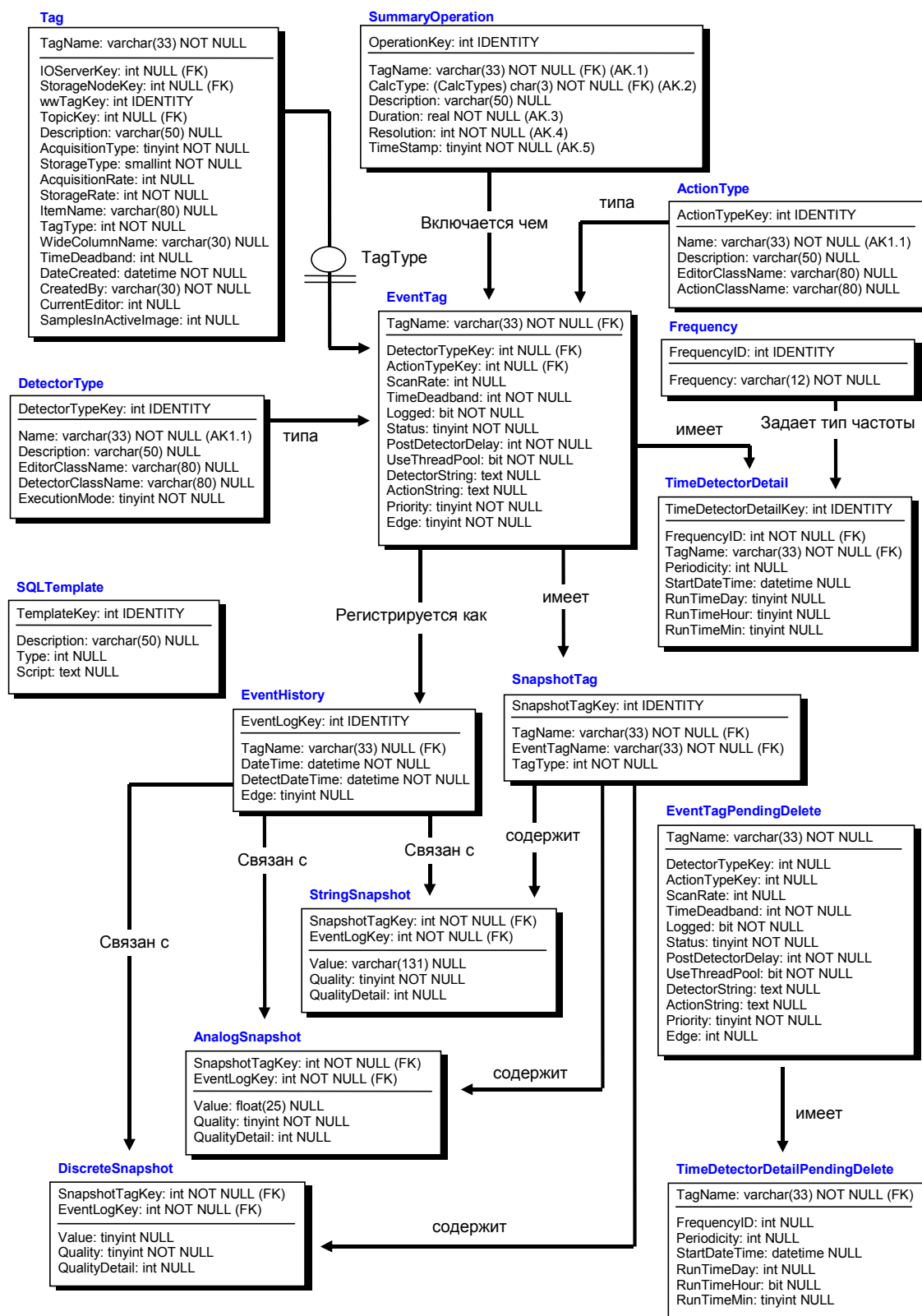
Содержание

- Таблицы хранения данных
- Таблицы хранения событий
- Таблицы интеграции InTouch
- Таблицы отслеживания изменений
- Таблицы именных пространств и групп
- Таблицы сводок
- Таблицы системной конфигурации
- Таблицы определения тэгов

Таблицы хранения данных



Таблицы хранения событий



FK = Внешний ключ

AK = Альтерн. внешний ключ

Таблицы интеграции InTouch

Tag

TagName: varchar(33) NOT NULL
 IOServerKey: int NULL (FK)
 StorageNodeKey: int NULL (FK)
 wwTagKey: int IDENTITY
 TopicKey: int NULL (FK)
 Description: varchar(50) NULL
 AcquisitionType: tinyint NOT NULL
 StorageType: smallint NOT NULL
 AcquisitionRate: int NULL
 StorageRate: int NOT NULL
 ItemName: varchar(80) NULL
 TagType: int NOT NULL
 WideColumnName: varchar(30) NULL
 TimeDeadband: int NULL
 DateCreated: datetime NOT NULL
 CreatedBy: varchar(30) NOT NULL
 CurrentEditor: int NULL
 SamplesInActiveImage: int NULL

InTouchNode

NodeKey: int NOT NULL
 MachineName: varchar(255) NOT NULL (AK.1)
 ApplicationName: varchar(32) NULL
 Path: varchar(250) NULL
 Description: varchar(50) NULL
 DuplicateChar: varchar(12) NOT NULL (AK.2)
 PrefixOrSuffix: bit NOT NULL
 AlwaysModifyName: bit NOT NULL
 ImportPlantTags: tinyint NOT NULL
 ImportSystemTags: tinyint NOT NULL
 ImportMemoryTags: tinyint NOT NULL
 FixedStorageRate: tinyint NOT NULL
 ImportRoute: tinyint NOT NULL

Берется из

имеет

имеет

InTouchSpecific

NodeKey: int NOT NULL (FK)
 TagName: varchar(33) NOT NULL (FK)
 OriginalName: varchar(32) NOT NULL
 TypeInfo: int NOT NULL
 InPrealism: bit NOT NULL
 Comment: varchar(50) NULL

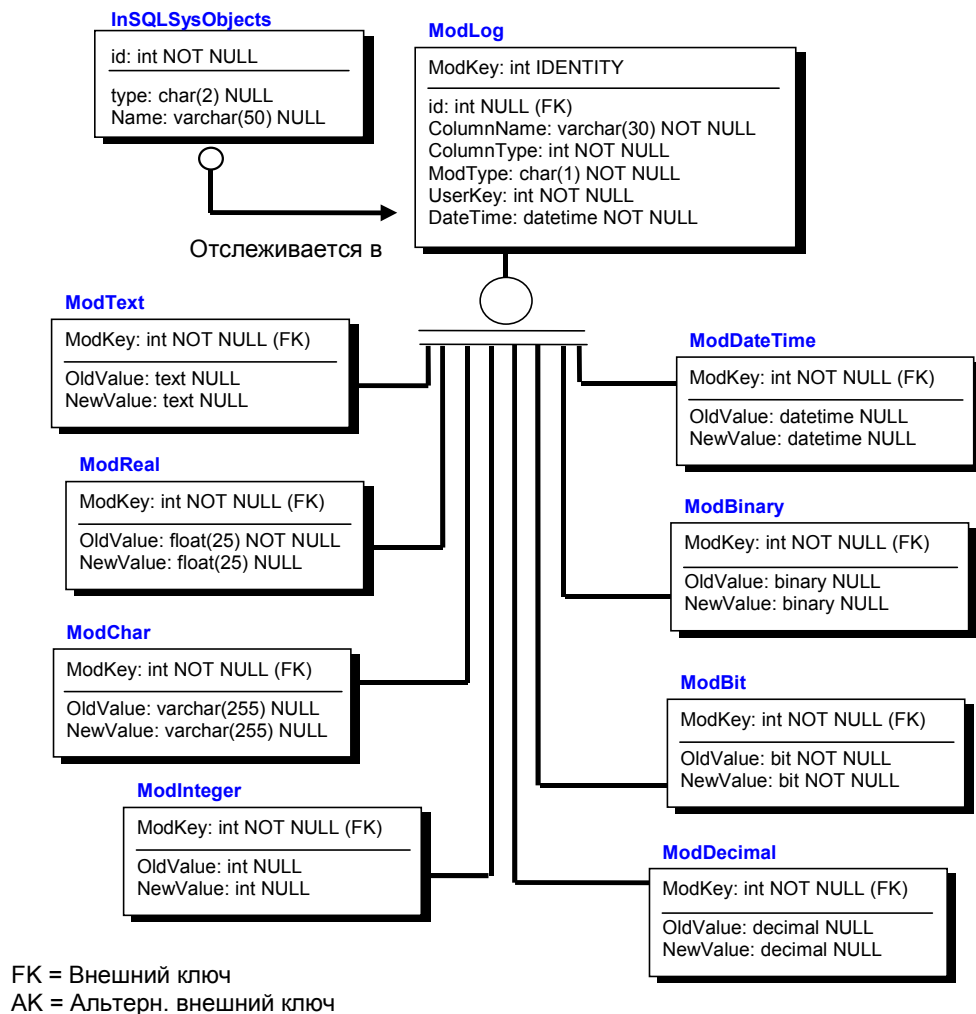
TopicImportInfo

NodeKey: int NOT NULL (FK)
 DdeSourceKey: int NOT NULL
 SourceName: char(50) NOT NULL
 ApplicationName: char(50) NULL
 TopicName: char(50) NOT NULL
 RequestInitialData: bit NOT NULL
 AlwaysAdvise: bit NOT NULL
 DefaultStorageRate: int NOT NULL
 DefaultStorageType: int NOT NULL
 TimeDeadband: int NOT NULL
 Import: bit NOT NULL
 ProtocolType: tinyint NOT NULL

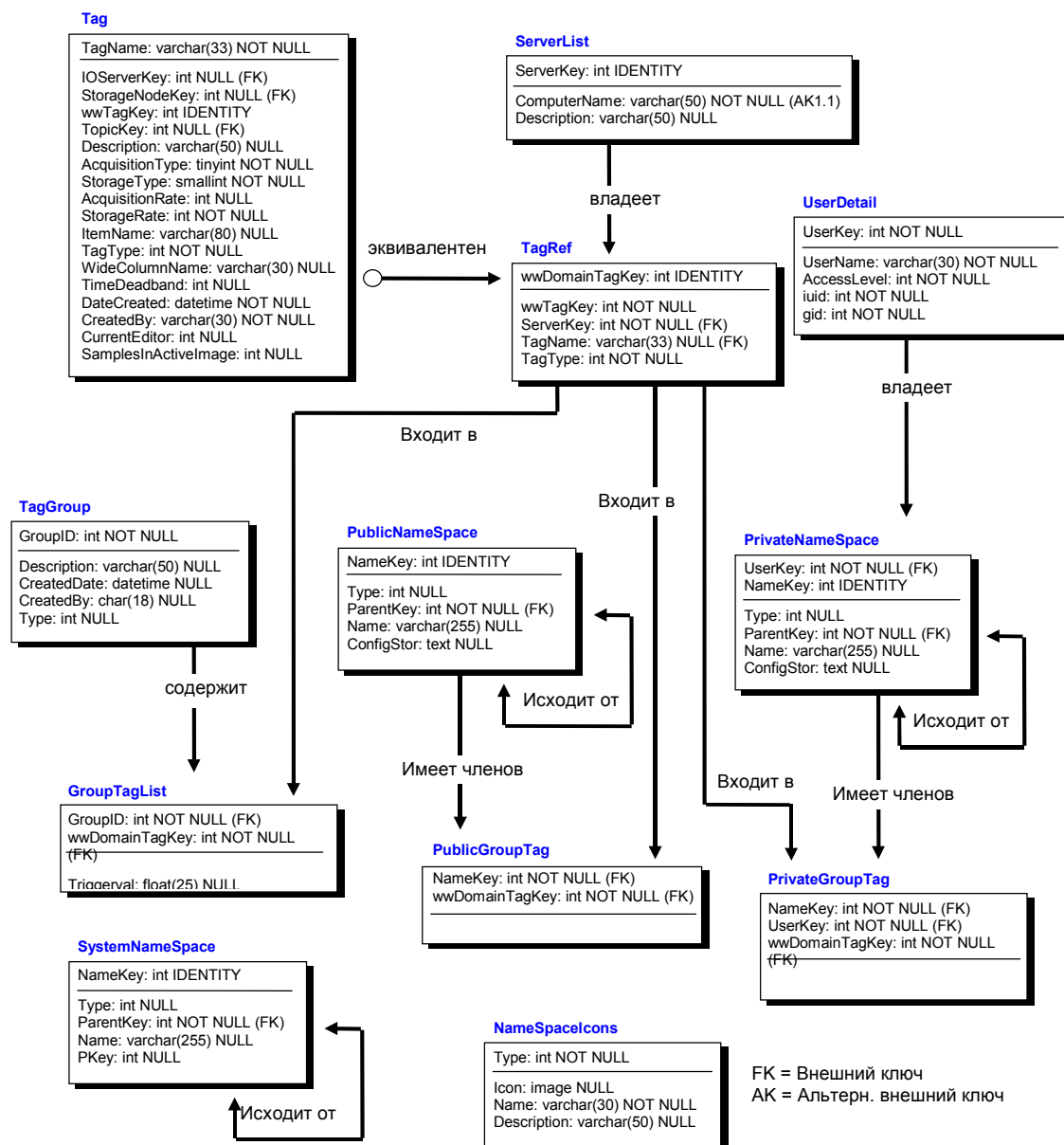
FK = Внешний ключ

AK = Альтерн. внешний ключ

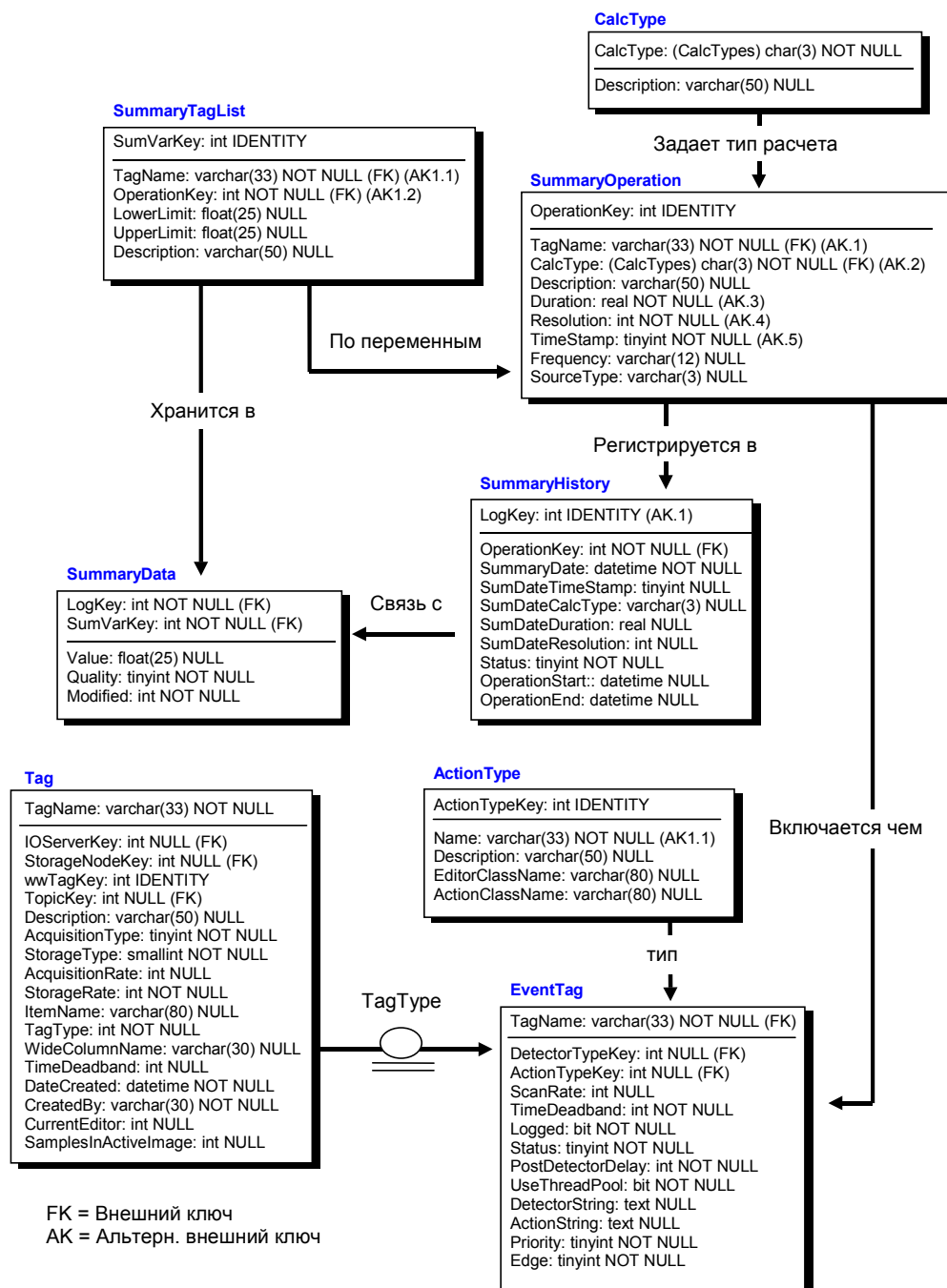
Таблицы отслеживания изменений



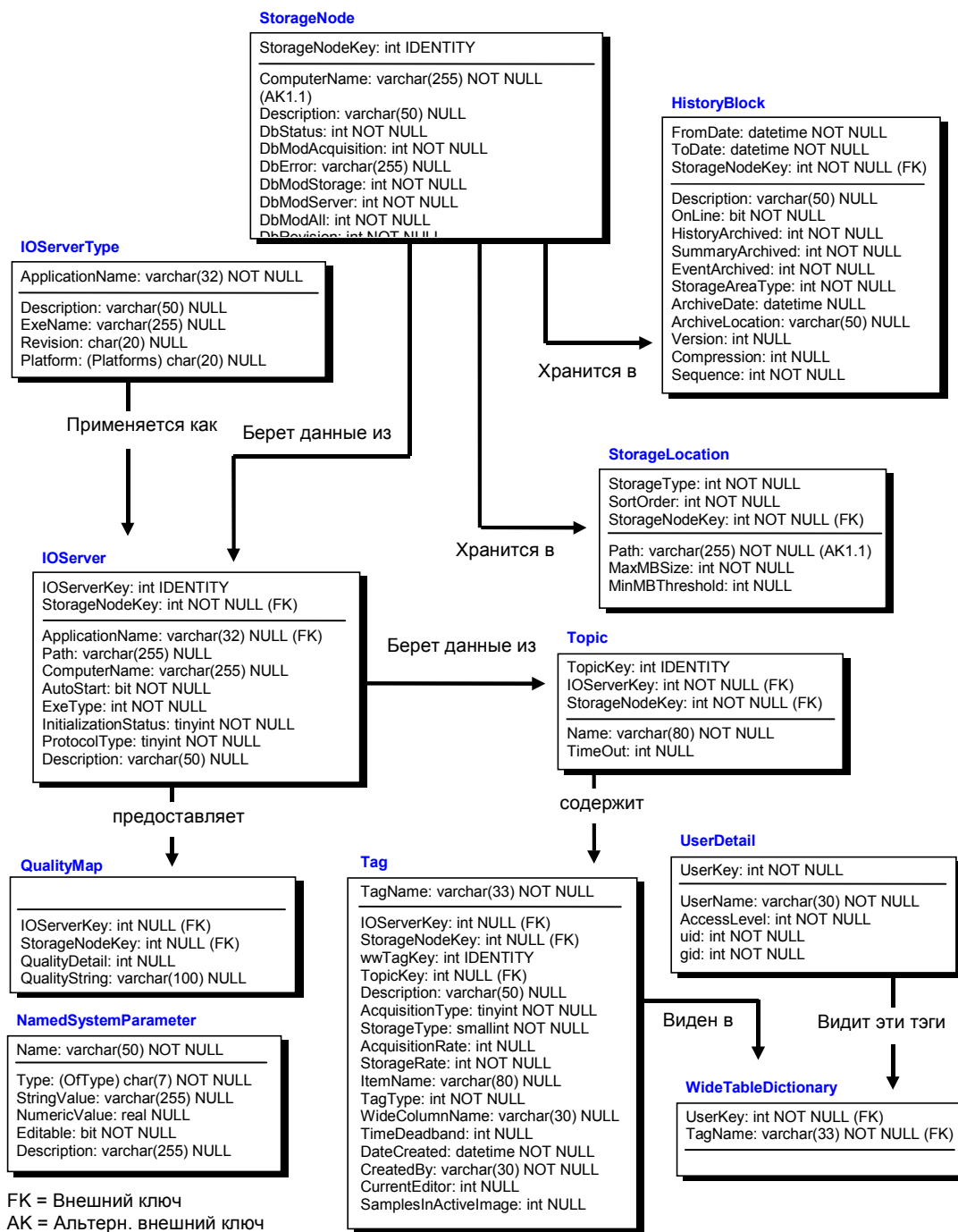
Таблицы именных пространств и групп



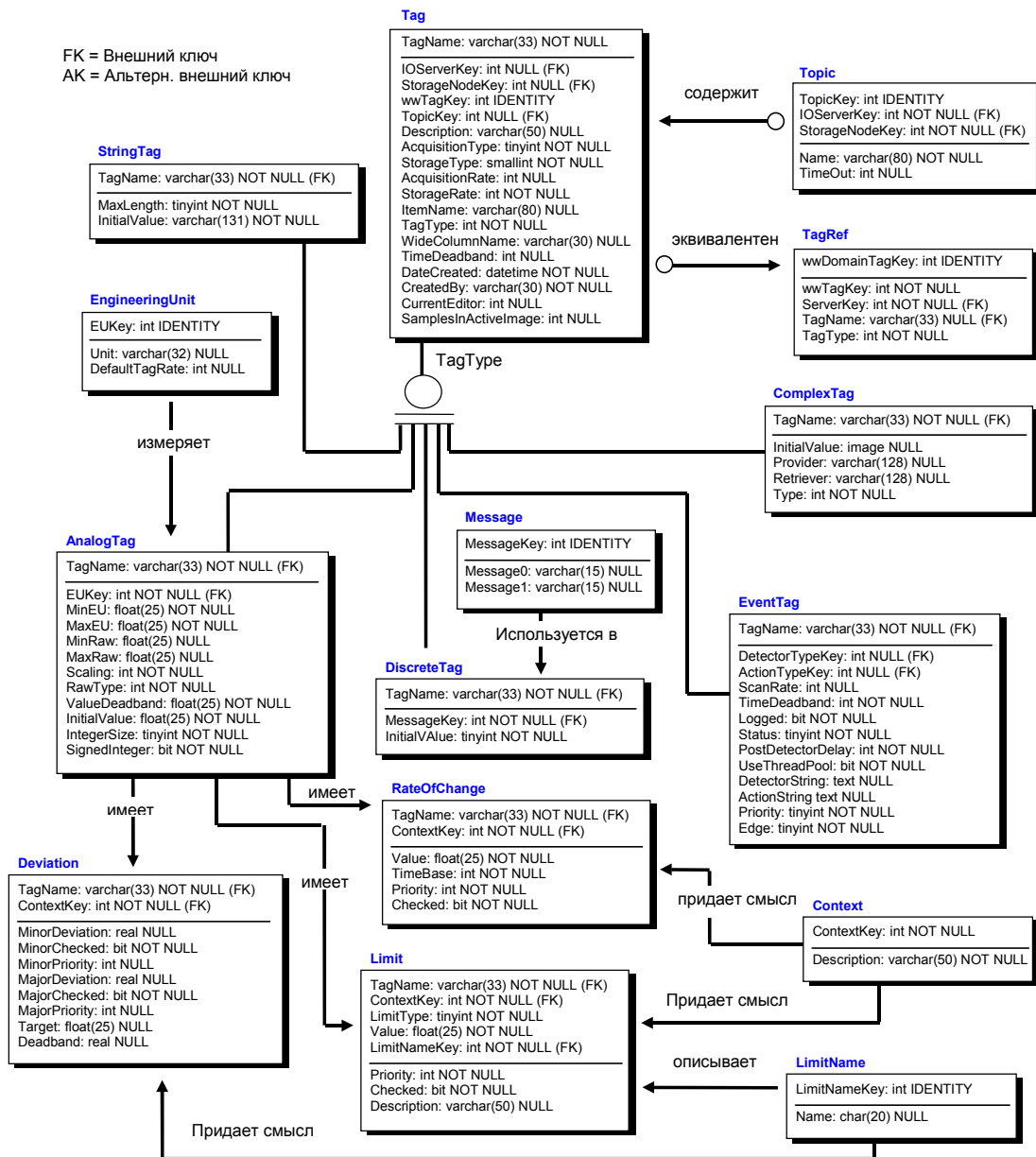
Таблицы сводок



Таблицы системной конфигурации



Таблицы определения тэгов



П Р И Л О Ж Е Н И Е В

Качество данных

В этом приложении описывается обработка информации качества в системе IndustrialSQL Server. Оно охватывает качество от уровня сбора данных через протокол SuiteLink до уровней хранения и архивации. Также здесь описана обработка информации качества со стороны клиента.

Содержание

- [Обзор](#)
- [Качество сбора данных](#)
- [Качество системы хранения](#)
- [Качество клиента](#)
- [Просмотр значений и описаний качества](#)

Обзор

Качество данных — это степень правильности значений данных. Качество данных варьируется от хорошего, когда значение в действительности такое, каким было получено с производственного участка, до плохого, когда значение либо неправильное, либо не может быть проверено. На всем пути обработки данных по мере того, как данные собираются, сохраняются, а затем отображаются, их качество может ухудшаться, поскольку на систему влияют внешние переменные и события. Существует три аспекта обработки качества в системе:

- Значения качества, присваиваемые устройствами сбора данных
- Качество системы хранения IndustrialSQL Server
- Определения качества со стороны клиента

IndustrialSQL Server использует значения качества, или флаги, для проверки состояния качества данных. Если качество нехорошее, то система пытается указать уровень качества во флаге.

Качество сбора данных

Когда значение данных получается сервером ввода-вывода Wonderware через SuiteLink, этому значению присваивается 2-байтовый флаг качества. Этот флаг отражает состояние качества значений элементов данных. Нижние 3 бита флага качества содержат 3 битовых поля: Качество (Q), Подстатус (S) и Граница (L). Верхние 8 битов не определены. Эти три битовых поля расположены в следующем порядке:

QQSSSSL

Присвоение битов в этих двух байтах полностью соответствует OPC-спецификации для качества данных.

Все эти битовые поля описаны в следующей таблице:

Битовое поле	Описание
Качество (Q)	Определяет состояние значения данных. Значение данных может быть плохим (0), неопределенным (1) или хорошим (3). Сервер, не поддерживающий информацию качества, выдает 3 (Хорошее). Для сервера также приемлемо просто выдавать Хорошее или Плохое (0 или 3) и всегда выдавать 0 для подстатуса и границы.
Подстатус (S)	Используется для дальнейшего описания полного качества значения. Например, если качество конкретного значения плохое, то поле Подстатуса содержит число, обозначающее причину плохого качества, как например ошибка устройства или ошибка настройки. Серверы, не поддерживающие подстатус, выдают 0. Заметьте, что 'старое' значение может быть выдано с установкой качества Плохое (0) и подстатусом, установленным на 5. Все это нужно для согласования со спецификацией Fieldbus.
Граница (L)	Выдает информацию о границах значения. Например: удерживается ли оно у верхней границы? Битовое поле Границы всегда правильно, несмотря на Качество и Подстатус. В некоторых случаях, таких как ошибка датчика, оно может содержать полезную диагностическую информацию.

Дополнительная информация содержится в Руководстве администратора системы FactorySuite.

Качество системы хранения

Система хранения IndustrialSQL Server сохраняет качество для каждого значения как в 0-байтовом, так и в 2-байтовом формате. 0-байтовый формат используется только для обратной совместимости; новые установки должны использовать 2-байтовый формат. Размер качества указан в глобальных системных настройках и читается из реестра.

Неопределенное качество

Соответствует 0-байтовому формату качества. Сохраняются только два основных значения качества (Хорошее и Плохое). Эти значения качества вложены в значения хранимых данных, таким образом они не занимают места.

2-байтовое качество

Для хранения деталей качества *всегда* используется парадигма дельта-сохранения (методом исключения). Качество занимает 2 байта, и оно такое же, как указанное для SuiteLink.

Качество клиента

Клиентам предоставляются два флага качества:

- 1-байтовый короткий флаг качества (Quality)
- 4-байтовый длинный флаг качества (QualityDetail)

Quality содержит сводную информацию, делящуюся на категории Хорошее, Плохое, Неопределенное или Исходное значение. Эта информация берется из источника данных, а также из IndustrialSQL Server.

QualityDetail содержит подробную информацию, которая является специфичной для предоставившего ее устройства. Эта информация получается через ряд общих назначений, причем можно добавлять новые назначения.

Качество

Quality есть 1-байтовый флаг, объясняющий смысл соответствующего элемента данных. *Quality* нумеруется следующими значениями:

Hex	Dec	Имя	Описание
0x00	0	Good	Хорошее значение.
0x01	1	Bad	Значение помечено как неправильное.
0x10	16	Doubtful	Значение неопределенное.
0x85	133	Initial Value(Good)	Исходное значение для дельта-запроса.

Initial Value и Doubtful берутся из QualityDetail. Initial Value зависит от типа выполняемого запроса.

Подробную информацию см. в разделах об операторах сравнения в главе 14 "Доступ к данным".

Детальное качество

QualityDetail содержит подробное описание качества как 32-битного значения. Значения качества могут быть получены из различных источников, как например от серверов ввода-вывода или от системы хранения IndustrialSQL Server. Каждый источник имеет байтовую позицию в *QualityDetail*. 32-битное значение:

0xXXSSDDDD

Битовые поля описаны в следующей таблице:

Поле	Диапазон	Источник	Описание
DDDD	Байт 0-1	Серверы ввода-вывода	Серверы ввода-вывода, сообщающиеся через SuiteLink. См. раздел "Качество сбора данных" ранее в этой главе.
SS	Байт 2	Система хранения IndustrialSQL	Система хранения может добавлять значения, связанные с внутренними ошибками, ошибками связи или ошибками времени. См. "Качество, добавленное системой хранения" далее в этой главе.
XX	Байт 3	Система хранения IndustrialSQL.	Используется для дополнительных послекачественных битов.

Качество, добавленное системой хранения

В следующей таблице байт 2 предназначен для качества, добавленного системой хранения IndustrialSQL. "X" используется для назначения байтов качества из других источников (байты 0, 1 и 3).

Hex	Имя	Описание	Связь с
0xXX01XXXX	No Data*	Для заданного времени никогда не сохранялись данные.	Плохое
0xXX02XXXX	Offline	Данные сохранены и зарезервированы, но недоступны.	Плохое
0xXX03XXXX	Lost	Данные сохранены, но не зарезервированы.	Плохое
0xXX04 XXXX	No Value	Запрошенное значение не является сохраненными данными.	Плохое
0xXX05 XXXX	Retrieve Error	Во время получения этого значения произошла ошибка.	Плохое
0xXX06 XXXX	Convert Error*	Произошла ошибка при конвертировании или масштабировании этого значения.	Плохое
0xXX07 XXXX	Event goes Invalid	Дельта-данные стали неверными из-за проблемы в блоке данных IndustrialSQL, а не из-за записанного события.	Плохое
0xXX08 XXXX	Event return to Valid	Дельта-данные вновь стали правильными благодаря проблемам блока данных IndustrialSQL а не из-за записанного события.	Неопределенно
0xXX09 XXXX	Communications error	Получение данных предотвращено ошибкой связи в системе InSQL.	Плохое
0xXX0AXXXX	Non-existent	Переменная в запросе сейчас не существует.	Плохое
0xXX0BXXXX	Deleted	Переменная была удалена.	Плохое
0xXX0CXXXX	Initial	Исходное значение для запроса, основанное на данных событий.	Хорошее
0xXX10XXXX	Overwritten	Оригинальное значение перезаписано после сохранения (например, при обновлении SQL).	Хорошее

(продолжение)

Hex	Имя	Описание	Связь с
0xXX11XXXX	Time Overlap	За положенное время было получено и сохранено более одного значения. Это может случиться, например, во время перевода часов на летнее/зимнее время.	Хорошее

*На данный момент поддерживается только качество системы хранения, в дополнение к "Хорошему".

Initial Value зависит от типа выполняемого запроса.

Подробную информацию см. в разделах об операторах сравнения в главе 14 "Доступ к данным".

Просмотр значений и описаний качества

Поддерживаемые значения качества и их описания хранятся в таблице *QualityMap* базы данных *Runtime*. Для просмотра значений и описаний качества запустите программу SQL Server Query Analyzer и подайте следующий запрос:

```
SELECT QualityDetail, QualityString FROM QualityMap
```

П Р И Л О Ж Е Н И Е С

Толковый словарь

<i>binary</i> тип данных	<i>Бинарные</i> данные есть тип данных, используемый для бинарных строк или битовых выборок. Бинарные строки вводятся шестнадцатиричным кодом. Есть два типа бинарных данных: <i>binary[(n)]</i> и <i>varbinary[(n)]</i> . См. также <i>binary[(n)]</i> и <i>varbinary[(n)]</i> .
<i>binary[(n)]</i> тип данных	Тип данных <i>binary(n)</i> может содержать до 255 байт бинарных данных с фиксированной длиной. Максимальная длина указана через <i>n</i> , где значение <i>n</i> варьируется от 1 до 255. Размер хранения для <i>бинарных</i> данных есть значение <i>n</i> . <i>Бинарные</i> данные — это тип системных данных. См. также <i>varbinary[(n)]</i> .
<i>bit</i> тип данных	<i>Битовые</i> данные могут содержать 1 или 0. Любое целое значение, не равное 0, автоматически понимается как 1. <i>Битовые</i> данные используются для значений да/нет и истина/ложь. Размер хранения для <i>битовых</i> данных 1 байт; вместе с тем, в одном байте могут быть сохранены до восьми атрибутов. См. также дискретное значение.
BLOB	См. крупный бинарный объект
<i>char</i> тип данных	Символьные данные есть тип данных, который может содержать до 255 символов. Символьные данные имеют постоянную длину хранения (<i>n</i>).
COM	См. модель составного объекта.
CPS	См. символов в секунду.
<i>datetime</i> тип данных	Данные <i>datetime</i> содержат дату и время суток. Размер хранения для данных <i>datetime</i> 8 байт, которые разделены на 4-байтовые целые: одно 4-байтовое целое для количества дней до или после 1 января 1900 года, и одно 4-байтовое целое для количества миллисекунд после полуночи. Даты до 1 января 1900 года сохраняются как отрицательные значения. <i>datetime</i> есть системный тип данных.
DBO	См. владелец базы данных.
DCOM	См. Распределенная модель составного объекта
FK	См. внешний ключ.

<i>float</i> тип данных	Данные типа <i>float</i> могут содержать положительные или отрицательные числа с плавающей точкой с точностью до 15 знаков. Значения могут быть от $1.7E - 308$ до $1.7E 308$. Размер хранения — 8 байт. <i>float</i> есть системный тип данных.
HTTP	См. <i>Протокол передачи гипертекста</i> .
IDENTITY столбец	Столбец идентификации содержит созданное системой значение, используемое для уникальной идентификации каждой строки таблицы. Если данные помещаются в таблицу, где определен столбец идентификации, сервер SQL автоматически создает значение. Это значение основано на последнем значении идентификации плюс приращение заранее определенного значения идентификации.
<i>image</i> тип данных	Данные типа <i>image</i> могут содержать от 0 до 2 147 483 647 байт бинарных данных различной величины. Данные <i>image</i> не могут использоваться для переменных или параметров в хранимых процедурах. <i>image</i> — это системный тип данных.
<i>int</i> тип данных	Данные типа <i>int</i> есть целые данные, которые могут содержать целые числа от $2^{31} - 1$ (то есть, 2 147 483 647) до -2^{31} (то есть, -2 147 483 648), включительно. Размер хранения — 4 байта. <i>int</i> есть системный тип данных.
IPX/SPX	IPX/SPX есть протокол передачи данных, используемый в сетях Novell®.
Named pipes	Механизм межпроцессового обмена (IPC), используемый для передачи данных между отдельными процессами, работающими обычно на разных компьютерах. В протоколе named pipes канал передачи (pipe) стабилизируется усилиями обоих процессов. См. также межпроцессовый обмен.
NULL	NULL означает, что запись в столбце не имеет соответствующего ей значения. NULL не означает, что числовое значение равно 0, не означает пустую строку. NULL есть отсутствие значения. Если для столбца не разрешено использование NULL, значение должно быть указано.
ODBC	См. Соединение открытых баз данных Microsoft.
ODS	См. Службы открытых данных.

OLE DB	Object Linking and Embedding for Databases (OLE DB) есть интерфейс программирования приложений (API), позволяющий приложениям-клиентам на COM-основе получать доступ к данным, не хранящимся физически в сервере Microsoft SQL, с которым они соединяются.
OLE DB	См. <i>Object Linking and Embedding for Databases</i> .
PK	См. первичный ключ.
<i>real</i> тип данных	Данные типа <i>real</i> (действительные) могут содержать положительные или отрицательные числа с плавающей точкой, с точностью до 7 знаков. Значения данных <i>float</i> могут быть от 3.4E - 38 до 3.4E 38. Размер хранения для данных типа <i>float</i> — 8 байт. <i>float</i> есть системный тип данных.
RI	См. целостность связей.
<i>smalldatetime</i> тип данных	Данные типа <i>smalldatetime</i> содержат дату и время суток с меньшей точностью, чем <i>datetime</i> . Размер хранения для данных <i>smalldatetime</i> — 4 байта. Данные <i>smalldatetime</i> состоят из одного малого целого числа для количества дней после 1 января 1900 и одного малого целого числа для количества минут после полуночи. Дата варьируется от 1 января 1900 до 6 июня 2079 с точностью до минуты. <i>smalldatetime</i> есть системный тип данных.
<i>smallint</i> тип данных	Данные типа <i>smallint</i> могут содержать целые числа между 215 - 1 (32 767) и -215 (-32 768), включительно. Размер хранения для данных <i>smallint</i> — 2 байта. <i>smallint</i> есть системный тип данных.
SQL	См. Язык структурированных запросов
SuiteLink	Сетевой протокол SuiteLink предназначен для высокопроизводительных промышленных приложений. SuiteLink располагает функцией Value Time Quality (VTQ) и выставляет штамп времени и качества на каждое значение данных, передаваемое совместимым с VTQ клиентам. SuiteLink работает на базе протокола TCP/IP.
TCP/IP	TCP/IP (Transmission Control Protocol/Internet Protocol) есть набор сетевых протоколов, обеспечивающих передачу пакетов данных через разнородные сети. TCP/IP может передавать пакетную информацию между различными видами оборудования и операционных систем.

<i>text</i> тип данных	Данные типа <i>text</i> могут содержать до 231 - (2 147 483 647) символов. Размер хранения для данных типа <i>text</i> может быть до 2 Кб, если они сохраняются не как BLOB. <i>text</i> является системным типом данных. См. также крупный бинарный объект.
<i>tinyint</i> тип данных	Данные типа <i>tinyint</i> могут содержать любое число от 0 до 255, включительно. Размер хранения для данных типа <i>tinyint</i> — 1 байт. <i>tinyint</i> является системным типом данных.
<i>varbinary</i> [(<i>n</i>)] тип данных	Данные типа <i>varbinary</i> (<i>n</i>) могут содержать от 0 до 255 включительно бинарных данных различной длины, где <i>n</i> указывает максимальный размер столбца. Размер хранения для данных <i>varbinary</i> есть размер значений данных для этого столбца, а не значение <i>n</i> . Данные <i>varbinary</i> являются системным типом данных.
<i>varchar</i> [(<i>n</i>)] тип данных	Данные типа <i>varchar</i> (<i>n</i>) могут содержать от 1 до 255 включительно символов данных различной длины. Столбец <i>varchar</i> может содержать 0 символов. Размер хранения для данных типа <i>varchar</i> есть размер значений данных, заполняющих столбец. Данные типа <i>Varchar</i> должны быть заключены в одинарные кавычки. Данные типа <i>varbinary</i> являются системным типом данных.
WORM	WORM есть оптический диск, запись на который возможна только однократно. WORM расшифровывается как "Write Once, Read Many" (записать однократно, считать многократно).
Агрегация	Функции агрегации есть функции SQL, которые проводят числовые подсчеты со столбцами в наборе данных. Функции агрегации: SUM, AVG, MIN, MAX и COUNT.
Активный объект	Активный объект есть область памяти, в которой временно содержится определенное количество значений полученных данных до их сохранения на диск.
Альтернативная область хранения	Альтернативная область хранения есть каталог на компьютере, используемый для хранения файлов, перемещенных их циркулирующей области хранения.
Аналоговое значение	Аналоговое значение есть переменная, измеряющая непрерывную физическую величину. Например, температура котла будет измеряться как аналоговое значение.

Аннотация	Аннотация есть пользовательский комментарий о тэге за определенный момент времени.
Архивировать	Архивировать —перемещать данные на устройства длительного хранения, такие как DAT-кассеты. Архивирование данных предотвращает их потерю в случае удаления или уничтожения исходных данных. <i>См. также резервировать.</i>
Архивные таблицы	В IndustrialSQL Server архивные таблицы представляют полученные производственные данные в архивном формате, где находится одна строка для каждого сохраненного значения тэга. <i>См. также живые таблицы, широкие таблицы.</i>
Архивный блок	Архивным блоком называется группа файлов хранения данных, которые записываются в отдельный каталог, идентифицируемый штампом даты и буквенным суффиксом. IndustrialSQL Server сохраняет полученные данные на диск в блоках. Архивные блоки создаются, когда система запускается или после истечения заранее определенного и указанного в реестре Windows NT периода времени.
Атрибут	В базе данных атрибуты представляют характеристики или свойства ячейки, такие как тэг, и обычно относятся к заголовкам столбцов таблицы.
База данных	База данных — это системное хранилище данных общих типов, организованных с помощью уникальных идентификаторов и таблиц. Базы данных хранятся в файлах, называемых устройствами баз данных.
База данных <i>runtime</i>	В IndustrialSQL Server база данных <i>runtime</i> содержит таблицы, хранящие все данные конфигурации, архивов и текущих процессов. <i>См. также база данных Holding.</i>
База данных <i>Holding</i>	В IndustrialSQL Server база данных <i>Holding</i> содержит таблицы, используемые для временного хранения информации, импортированной с узла InTouch перед помещением ее в базу данных <i>runtime</i> . <i>См. также база данных runtime.</i>
Байт	Байт есть единица информации, состоящая из 8 бит. С точки зрения хранимых данных, байт равен одному символу, как например буква или цифра.
Блок	<i>См. архивный блок.</i>

Броузер	Броузер есть графическое представление иерархий данных. Броузер используется для показа каталогов на диске, папок, файлов и т.п. Клиенты IndustrialSQL Server используют броузер для показа серверов, тэгов и групп тэгов в системных, личных и общих именованных пространствах. <i>См. также</i> именованные пространства.
Булево выражение	Булево выражение выдает значение истины или лжи. Булево выражение содержит логические операторы, такие как И, ИЛИ и НЕ. <i>См. также</i> логические операторы.
Буферная область хранения	Буферная область хранения есть каталог на компьютере, используемый для временного хранения файлов, например, для получения данных из архива.
Ведомый край	Если в расширении определения края указан параметр "Ведомый", это означает, что единственными строками в результатах запроса будут те, которые не совпали с критериями (выдана ложь) после строки, которая с этими критериями совпала (выдана истина). <i>См. также</i> ведущий край, определение края.
Ведущий край	Ведущий край — это когда запрос выдает только те строки, которые первыми удовлетворили критериям (выдана истина) после строки, этим критериям не удовлетворившей (выдана ложь). <i>См. также</i> определение края, ведомый край.
Владелец базы данных (DBO)	Владелец базы данных есть пользователь, создающий базу данных. Владельцы баз данных имеют право полного доступа к созданным ими базам данных, включая право на предоставление доступа к этим базам данных другим пользователям. У каждой базы данных может быть только один владелец. Владение базой данных может передаваться между пользователями и владелец базы данных может иметь несколько идентификационных псевдонимов.
Владелец объекта	<i>См. владелец объекта базы данных.</i>
Владелец объекта базы данных	Владелец объекта базы данных — это пользователь, создающий объект базы данных, такой как таблица, указатель, представление, правило и т.п. Владельцы объектов баз данных имеют полный доступ ко всем созданным ими объектам, включая право на предоставление доступа к этому объекту другим пользователям. Право на владение объектом базы данных не передается.

Внутренний тэг	Внутренние тэги — это тип тэгов, существующих внутри приложения InTouch. Они используются для создания постоянных системных значений и моделей. Также они используются при создании подсчитываемых переменных, к которым обращаются другие программы Windows. Существует 4 типа внутренних тэгов: внутренний дискретный, внутренний целый, внутренний действительный и внутренний текстовый.
Временной интервал	Для системы управления событиями временным интервалом называется скорость, с которой детекторы событий проверяют, произошло ли событие. Этот интервал времени также известен как скорость сканирования.
Граница	Границей называется определяемое пользователем минимальное или максимальное значение для отрезка значений.
Группа	См. группа пользователей.
Группа пользователей	Группой пользователей называется группа пользователей базы данных, имеющих право на определенные действия с базой данных. Группы пользователей являются частью системы безопасности базы данных. Любой пользователь базы данных, добавляемый в группу пользователей, получает права, соответствующие этой группе.
Дамп	См. резервирование.
Дамп на ленту	Дамп на ленту есть копирование структур базы и хранимых данных на ленточное устройство.
Данные	Кодированное представление информации для использования в компьютере. У данных есть атрибуты, такие как тип и длина.
Действие	Действие над событием есть действие, совершаемое в момент распознавания детектором возникновения события. Действия над событиями необязательны; бывают случаи, когда нужно только зафиксировать возникновение события. См. также детектор, тэг события.
Действительное число	Действительное число есть число с плавающей точкой, представленное цифрами с фиксированной основой, как например в десятичной системе. Действительное число может быть составлено из конечного или бесконечного количества цифр.

Дельта-выборка	Дельта-выборка, или выборка методом исключения, есть получение изменения значения тэга в заданный период времени. То есть, дублируемые значения не выдаются. <i>См. также</i> циклическая выборка.
Дельта-хранение	Дельта-хранение есть сохранение данных, основанное на изменении значения. Дельта-хранение позволяет сохранять тэг только при изменении значения. Дельта-хранение также называется "хранение методом исключения". Этот метод используется для сохранения дискретных или аналоговых значений, скорость изменения которых неустойчива (иногда быстра, иногда медленна) и когда требуется дискретность менее 1 секунды. <i>См. также</i> циклическое хранение.
Детектор	Детектор событий есть механизм определения, когда были удовлетворены критерии событий для тэга события. Детекторы событий срабатывают в соответствии с интервалом времени.
Динамический обмен данными (DDE)	DDE есть процесс передачи данных между приложениями без участия пользователя и без просмотра. В среде Windows DDE реализован в виде набора сообщений, рекомендуемых процедур (протоколов) для обработки этих сообщений и некоторых новых типов данных. Следуя этим протоколам, приложения, написанные независимо друг от друга, могут передавать данные друг другу без участия пользователя. Например, InTouch и Excel. <i>См. также</i> тема, элемент, общий ресурс DDE, сервер ввода-вывода, клиент DDE, SuiteLink.
Дискретное	Дискретное значение есть переменная, имеющая только два состояния: '1' (Истина, Включено) или '0' (Ложь, Выключено).
Дискретность	Дискретность есть интервал в миллисекундах для получения данных из любой аналоговой или дискретной архивной таблицы IndustrialSQL Server. Расширение временного домена дискретности IndustrialSQL Server не поддерживается обычными функциями сервера SQL. Количество выдаваемых строк зависит от периода времени запроса и дискретности (количество строк = период времени / дискретность). Дискретность применяется только при циклической выборке.
Допуск к объекту	Допуск к объекту определяет, какие предложения могут быть использованы с объектом базы данных. Допуск к объекту определяется владельцем этого объекта базы данных. <i>См. также</i> объект, владелец объекта базы данных.

Драйвер ODBC	Драйвер ODBC позволяет ODBC-совместимому приложению обращаться к источнику данных. Драйвер ODBC является библиотекой DLL. См. также Соединение открытых баз данных Microsoft, Библиотека динамического соединения.
Единица измерения	Единица измерения для значений тэга. Например, RPM, миллисекунды, градусы.
Живой	"Живой" — термин, описывающий данные, отражающие самое последнее значение тэга.
Живые таблицы	В IndustrialSQL Server "живые" таблицы содержат текущие значения полученных производственных данных для аналоговых, дискретных или текстовых тэгов. См. также <i>архивные таблицы, широкие таблицы</i> .
Журнал	Таблица данных, разделенная на категории по простым типам информации, которая хранится в базе данных и использует штамп времени как первичный ключ.
Журнал транзакций	Журнал транзакций есть запись всех изменений базы данных в базу данных. См. также транзакция.
Задержка	Задержка — это количество времени между действительным происхождением события в системе и его распознаванием детектором событий.
Запись	См. строка.
Запрос	Запрос есть сценарий выражений SQL, подаваемый в базу данных клиентом, который выполняет поиск объектов в таблице базы данных.
Запрос к базе данных	См. запрос.
Идентификатор пользователя	См. учетная запись.
Именное пространство	Именное пространство есть имеющий имя набор объектов. Именное пространство — это логическая "область", содержащая иерархию объектов. Например, серверы, тэги, темы и т.п. IndustrialSQL Server определяет три вида именных пространств: системные, общие и личные. Иерархическое содержание именного пространства отображается в броузере или приложении-клиенте. В IndustrialSQL Server определение, заключенное в именное пространство, управляется внутренними хранимыми процедурами. См. также системное именное пространство, общее именное пространство, именное пространство, браузер, хранимые процедуры.

Имя базы данных	Имя базы данных используется для идентификации базы данных при подключении к ней клиента.
Имя источника данных (DSN)	По стандарту Связи открытых баз данных (ODBC) это имя соединения, при помощи которого клиент обращается к серверу. <i>См. также</i> Связь открытых баз данных.
Имя пользователя	Имя пользователя идентифицирует пользователя базы данных в целях безопасности. Имя пользователя добавляется к учетной записи для доступа конкретного пользователя к базе данных.
Имя сервера	Именем сервера называется имя, присвоенное IndustrialSQL Server. Имя сервера должно соответствовать правилам идентификации SQL.
Имя тэга	Имя тэга есть имя, присвоенное переменной в базе данных.
Индекс	Индексом называется набор указателей, которые дают более быстрый доступ к данным в строках таблицы, чем при сканировании таблицы. Понятие индекса таблицы близко к предметному указателю в конце этой книги; записи в предметном указателе делают поиск данных быстрее, чем при пролистывании книги от начала в поисках нужной информации. Индексы также могут усилить унификацию строк в таблице. Существует два типа индексов: сгруппированные и несгруппированные. <i>См. также</i> сгруппированный индекс, несгруппированный индекс.
Инициализация	Инициализация есть запуск IndustrialSQL Server и систем хранения.
Внешний ключ (FK)	Внешний ключ есть один или несколько столбцов, чьи значения совпадают с первичным ключом (PK) какой-либо другой таблицы. Одиночный первичный ключ может иметь внешние ключи более чем в одной таблице.
Интерфейс человека с машиной (HMI)	Интерфейс человека с машиной есть интерфейс программного обеспечения, позволяющий операторам производственных участков просматривать, манипулировать и сохранять производственные данные. HMI может быть запущен на PC или другом производственном терминале.
Каскадировать	При редактировании ячейки базы данных (например, при операциях UPDATE или DELETE) используется для автоматического проведения того же действия над всеми связанными строками и столбцами базы данных.

Качество	Качеством называется показатель точности, доступности и правильности полученных данных. Значения данных, сохраняемые в IndustrialSQL Server, имеют соответствующее значение качества.
Кластерный индекс	В кластерном индексе логический порядок значений символов аналогичен физическому порядку данных. То есть данные содержатся в порядке, предусмотренном кластерным индексом. В таблице может быть только один кластерный индекс, так как может быть только один физический порядок строк. Индексы используются системой для унификации строк или для улучшения процесса поиска данных в базе.
Клиент DDE	Клиент DDE есть приложение, получающее данные от сервера ввода-вывода через протокол Microsoft Dynamic Data Exchange. <i>См. также</i> Динамический обмен данными.
Контекст	Смысловое описание события или связи, которое может охватывать группу пределов, скорости изменения или отклонения. Примерами являются "Нормальная работа", "Холодное выключение", "Особое внимание".
Крупный бинарный объект (BLOB)	Крупный бинарный объект, или BLOB, есть сложное значение данных, такое как массив, файл .AVI или рисунок, которое не интерпретируется базой данных.
Кэш	Кэш есть специальная подсистема памяти, в которой сохраняются данные, к которым часто происходит обращение, и соответствующие им адреса. Кэш используется как буфер для хранения данных во время перемещения между жестким диском и ОЗУ. Обычно кэш-память быстрее чем ОЗУ.
Личное именное пространство	Личное именное пространство есть набор пользовательских производственных компонентов, таких как производственные участки, имена компьютеров и тэги, связанные с конкретным компьютером или процессом. Личное именное пространство работает также, как и общее именное пространство, за исключением того, что личное именное пространство доступно не всем пользователям системы IndustrialSQL Server. Личное именное пространство создается пользователем IndustrialSQL Server. Иерархия содержимого личного именного пространства может быть просмотрена в дереве каталога, но будет видна только пользователю, создавшему ее. <i>См. также</i> именное пространство, системное именное пространство, общее именное пространство, браузер.

Логические операторы	Логические операторы используются для подсчета или сравнения данных. Образцами логических операторов являются И (AND), ИЛИ (OR), и НЕ (NOT). Логические операторы И, ИЛИ и НЕ могут использоваться в выражениях WHERE для указания условий поиска. Оператор И означает, что оба условия выполнены. ИЛИ означает, что выполнено одно из условий. НЕ означает, что выполнены все условия, кроме тех, на которые указывает этот оператор. <i>См. также</i> булево выражение.
Логические таблицы	<i>См.</i> представление, таблица расширения.
Логическое представление	<i>См.</i> представление.
Межпроцессовая связь (IPC)	Межпроцессовая связь есть система, при которой один процесс или задание может обмениваться данными с другим. <i>См. также</i> named pipe
Мертвая зона	Мертвая зона есть величина увеличения или уменьшения значения, которая допускается до происхождения события в системе.
Место хранения	Место хранения есть каталог, в который сохраняются файлы архивных данных.
Миллисекунда	Одна тысячная доля секунды, сокращенно мс или мсек.
Множественный протокол	Сетевая библиотека множественных протоколов позволяет серверу принимать входящие соединения сети по протоколам named pipes, портам TCP/IP и слотам SPX. Если для получения имен через поддерживаемые протоколы используется локальная база данных RPC, то в строке соединения необязательно указывать порты и слоты. <i>См. также</i> межпроцессовый обмен, вызовы удаленных процедур, слот.
Модель составного объекта (COM)	COM есть набор служб, которые позволяют компонентам программного обеспечения совместно работать в сетевой среде.
Набор строк	Набор строк есть строки, каждая из которых содержит столбцы данных.
Необработанное значение	Необработанным называется значение элемента данных при его получении. Подсчеты и преобразования могут проводиться с необработанными значениями до использования этих значений системой IndustrialSQL Server.

Несгруппированный индекс	Несгруппированным называется индекс, в котором логический порядок номеров не совпадает с физическим порядком строк в таблице. Несгруппированный индекс есть группа логических ссылок на строки. В отличие от сгруппированного индекса, для одной таблицы может быть определено до 249 несгруппированных индексов. Обычно несгруппированные индексы работают медленнее сгруппированных. Индексы используются для быстрого поиска информации в таблицах. <i>См. также</i> индекс, сгруппированный индекс.
Общее именное пространство	Общее именное пространство есть созданный администратором набор производственных компонентов. Так же, как системное именное пространство включает в себя информацию о системе IndustrialSQL Server, общее именное пространство включает в себя информацию об участке, на котором запущен IndustrialSQL Server. Общее именное пространство включает в себя такие объекты, как производственные участки, имена компьютеров и тэги, связанные с конкретным компьютером или процессом. Общее именное пространство создается администратором IndustrialSQL Server. Иерархия содержимого общего именного пространства видна всем пользователям в дереве каталога. <i>См. также</i> именное пространство, системное именное пространство, личное именное пространство, броузер.
Общий ресурс DDE	Общий ресурс DDE есть конфигурация памяти компьютера, доступ к которой имеют приложения с другого компьютера. Общий ресурс DDE используется для взаимодействия между несколькими DDE-совместимыми приложениями. Операции DDE также называются "диалогами" DDE. <i>См. также</i> Динамический обмен данными, тема, элемент.
Объект	Объектом называется любой из компонентов, составляющих базу данных. Примерами объектов базы данных могут служить таблицы, представления, символы, стандарты, триггеры, индексы, хранимые процедуры. Также называется объектом базы данных.
Объект базы данных	<i>См.</i> объект.
Ограничение	Ограничение есть запрет на выдачу данных, который может быть вставлен в столбец или строку.
Определение края	Определение края для конкретных данных. Край есть абстрактная линия, где удовлетворение критериям изменялось с истины на ложь и наоборот. <i>См. также</i> ведущий край, ведомый край.

Первичный ключ (РК)	Первичным ключом называются один или несколько столбцов, которые делают строку таблицы уникальной. Первичный ключ используется при связке с внешними ключами в других таблицах. <i>См. также</i> Внешний ключ.
Отклонение	Отклонение есть процент изменения значения тэга по отношению к зафиксированному значению, называемому целью. У каждого аналогового тэга могут быть два возможных отклонения: высшее и низшее.
Пара сообщений	Состояния, связанные с состояниями ИСТИНА (Включено) или ЛОЖЬ (Выключено) дискретного значения. <i>См. также</i> дискретное значение.
Параметр	Параметр есть информационный элемент, имеющий значение. Параметры определяют, требуется ли записать значение или выдать его из базы данных.
Повторная инициализация	Повторная инициализация есть перезапуск одного или нескольких процессов в системе IndustrialSQL Server. При редактировании базы данных <i>Runtime</i> может понадобиться перезагрузка для того, чтобы изменения возымели эффект.
Поле	<i>См.</i> столбец.
Получатель	Для сложного тэга получателем будет функция или способ получения данных, хранимых в базе данных. Например, правильный поток данных. Функции получателя хранятся в библиотеке DLL в системе IndustrialSQL Server. <i>См. также</i> поставщик данных, сложный тэг, библиотека динамического соединения.
Пользовательские данные	Пользовательскими называются данные, созданные пользователем. Пользовательские данные существуют в дополнение к уже существующим системным типам данных. Данные, которые могут быть сохранены в столбце таблицы базы данных, определяются типом данных для этого столбца. Стандарты и правила применяются только к пользовательским данным, а не к системным типам.
Порядок сортировки	Порядок сортировки есть набор правил, которые указывают, как сервер SQL должен упорядочивать и представлять данные в качестве результата запроса базы данных. Порядок сортировки определяет, как сервер SQL будет обращаться со сравнениями элементов для хранения данных и операций выборки данных при помощи выражений GROUP BY, ORDER BY и DISTINCT. Сервер SQL также использует выделенный порядок сортировки для получения запросов при помощи выражений WHERE и DISTINCT.

Поставщик данных	Для сложного тэга поставщик данных есть функция поставки данных для сохранения в базе данных. Например, правильный поток данных. Функции поставщика данных хранятся в библиотеке DLL в системе IndustrialSQL Server. См. также получатель, сложный тэг, библиотека динамического соединения.
Поставщик данных OLE DB	"Виртуальный" сервер, позволяющий внутренней программе обращаться к хранилищу данных OLE DB.
Постоянная область хранения	Постоянная область хранения есть каталог на компьютере, используемый для хранения критических данных (например, работы реактора), которые нельзя перезаписывать.
Права доступа	Права доступа ограничивают пользователя базы данных в действиях, которые он может проводить с базой данных. Например, пользователь может иметь право на выборку данных из всех таблиц БД, но не иметь прав на запись данных.
Правило	Правило базы данных есть объект, привязанный к столбцу таблицы или пользовательским данным. Правила определяют, какие типы данных могут быть введены в таблицу. Например, правило может указывать, что количество часов должно быть от 0 до 23. Для одной таблицы может быть только одно правило.
Представление	Представление есть логический способ просмотра данных в одной или нескольких таблицах базы данных. Представление является виртуальной таблицей; физически она не присутствует в базе данных. Представление содержит указатели на существующие в базе данных таблицы. Представления могут использоваться для составления подмножества информации, хранимой в одной или нескольких таблицах, если не используется иная информация. Это особенно полезно, если некоторые столбцы таблицы содержат чувствительную информацию. Запросы к представлениям выглядят как запросы к обычной физической таблице. Представления входят в стандартные функции сервера SQL. В IndustrialSQL Server, тем не менее, к данным можно обращаться при помощи таблиц расширения, которые отличаются от обычных представлений. См. также таблица расширения.
Приоритет	Приоритет событий определяет, как будут выполняться события, если система перегружена и не может обработать все события. События, которые имеют приоритет "критические", будут выполняться перед событиями с приоритетом "нормальные".

Протокол передачи гипертекста (HTTP)	HTTP есть протокол передачи информации через Интернет.
Псевдоним	Псевдоним есть имя, под которым сетевой сервер известен сетевым клиентам. Сервер может иметь несколько псевдонимов.
Путь хранения	Путь хранения есть путь к каталогу, в который сохраняются файлы архивных данных.
Разнородный запрос	Разнородным называется запрос, обращающийся к данным из различных непохожих источников данных.
Распределенная модель составного объекта	Распределенная модель составного объекта DCOM есть протокол, позволяющий компонентам программного обеспечения напрямую взаимодействовать через сеть.
Распределенная модель составного объекта (DCOM)	
Реальное время	Операции реального времени совершаются с той же скоростью, что и физические процессы. В среде реального времени компьютер должен реагировать на ситуации по мере их возникновения. Эти ситуации могут включать в себя сдвиг переключателя или включение печи.
Результат	Результатом называются характеристики или атрибуты любого объекта, выявленные запросом к базе данных.
Реляционная база данных	Реляционная база данных есть такое устройство базы данных, которое упорядочивает данные в соответствии с отношениями между ними. В реляционной базе данных отношения между элементами данных выражаются посредством таблиц. Например, могут подаваться запросы на поиск данных в одной таблице плюс в любых связанных с ней таблицах.
Сбор данных	Процесс получения переменных данных из процессов производственного участка.
Сброс	Сброс есть процесс удаления базы данных или устройства базы данных.
Сводка	Операция сводки (MIN, MAX, SUM, AVG) тэга, запланированная пользователем и автоматически выполняемая системой управления событиями в соответствии с расписанием.

Сводные данные	Сводные данные есть данные, полученные в результате внутренних подсчетов, проводимых IndustrialSQL Server (максимум, среднее, сумма и т.п.). Сводные данные обеспечивают верхний уровень просмотра данных и позволяют использовать меньше пространства для данных, которые требуется хранить долгое время. Например, среднее значение пяти тэгов.
Связанный сервер	Связанный сервер есть поставщик данных OLE DB, связанный с сервером SQL. См. также <i>поставщик данных OLE DB</i> .
Связка	Связка есть класс запроса SQL, который запрашивает данные из одного или нескольких столбцов двух или большего числа таблиц.
Сервер DDE	Сервер DDE есть приложение, поставляющее данные клиенту DDE через протокол Microsoft Dynamic Data Exchange. См. также Динамический обмен данными.
Сервер базы данных IndustrialSQL	Сервер базы данных IndustrialSQL (PDSSRV.EXE) перехватывает все запросы клиента на данные до того, как они достигнут Microsoft SQL Server, а если запрос идет на извлечение данных из архивных блоков IndustrialSQL Server, сервер базы данных контактирует с системой получения IndustrialSQL Server, получая данные и представляя их пользователю в виде нормальной таблицы Microsoft SQL Server. Запросы, не касающиеся архивных блоков, неизмененными проходят в Microsoft SQL Server для стандартной обработки.
Сервер PB	Сервер реального времени (PB) есть компонент системы IndustrialSQL Server, принимающий значения данных от устройств сбора производственных данных, проводящий все необходимые операции и отправляющий значения в систему хранения. См. также система хранения
Сервер реального времени	См. Сервер PB.
Символ	(1) символом называется столбец, служащий для идентификации строки. Символ строки должен быть уникален в таблице. (2) Символ есть папка в реестре Windows NT. Символы похожи на заголовки разделов в файле .INI.

Символов в секунду (CPS)	Количество символов в секунду есть общее число символов, посланных в систему хранения всеми источниками данных. Источники данных включают в себя серверы ввода-вывода, IndustrialSQL RT Server и системные драйверы. Количество символов в секунду может быть просмотрено при помощи управляющего приложения IndustrialSQL Server.
Синхронизация времени	Синхронизация времени есть механизм, с помощью которого IndustrialSQL Server посылает сообщение в серверы ввода-вывода для синхронизации штампов даты/времени серверов ввода-вывода с временем IndustrialSQL Server.
Система хранения	Система хранения IndustrialSQL Server отвечает за обработку (если требуется), архивирование и хранение данных на диске.
Системное именное пространство	Системное именное пространство есть созданный системой набор компонентов системы IndustrialSQL Server. Системное именное пространство содержит объекты, составляющие систему IndustrialSQL Server, такие как серверы ввода-вывода, узлы и т.п. IndustrialSQL Server заполняет системное именное пространство на основе текущей конфигурации производственного участка, которая хранится в базе данных в таблицах системной конфигурации. Иерархия содержимого системного именного пространства может быть просмотрена в дереве каталогов. См. также именное пространство, общее именное пространство, личное именное пространство, броузер, хранимые процедуры.
Системные тэги	Системным тэгом называется заранее определенная переменная системы. Системные тэги InTouch имеют префикс \$. Например, \$DateTime. Системные тэги IndustrialSQL имеют префикс SYS. Например, SysTimeSec.
Системный администратор (SA)	Системным администратором называется человек, ответственный за администрирование и сопровождение сервера SQL. Функции администрирования и сопровождения включают в себя изменение базы данных, администрирование безопасности базы данных, резервирование данных и т.п.
Скорость изменения	Скоростью изменения называется скорость изменения значения тэга за определенный период времени, обычно выраженная в процентах.
Скорость опроса	Скорость опроса есть скорость, с которой данные считываются из устройства сбора.

Скорость хранения	Скорость хранения есть интервал времени, с которым периодически сохраняются значения тэгов.
Словарь данных	Словарь данных есть группа таблиц, содержащих информацию обо всех объектах базы данных.
Сложный тэг	Сложный тэг содержит данные, которые не могут быть отнесены ни к одной из четырех основных категорий типов тэгов: аналоговых, дискретных, текстовых или тэгов событий. Сложные тэги включают в себя BLOBы, такие как массивы, файлы с расширением AVI, рисунки и т.п. <i>См. также</i> крупный бинарный объект.
Службы открытых данных (ODS)	Службы открытых данных (ODS) есть интерфейс программирования приложений (API) между сетевыми библиотеками сервера и приложениями сервера. Службы ODS позволяют приложениям-клинтам получать доступ к данным, которые не хранятся физически на сервере Microsoft SQL, к которому они присоединяются.
Снимок	Снимок есть набор значений тэгов данных в конкретный момент времени. Когда в системе происходит событие, значения всех аналоговых или дискретных тэгов в этот момент могут быть зафиксированы и сохранены. Данные снимка используются для определения состояния производственной среды в момент происхождения определенного события в системе.
Событие	Событие есть проявление определенной активности в системе. Определение события сохраняется как тэг события. События засекаются детекторами событий и с ними можно производить действия. <i>См. также</i> детектор, действие, тэг события.
Соединение открытых баз данных (ODBC) Microsoft	ODBC есть API-интерфес соединения, независимый от баз данных .
Составной индекс	Составной индекс есть индекс, использующий от 2 до 16 столбцов как первичный ключ. Например, составной индекс может использовать и <i>дату/время</i> и <i>имя тэга</i> в составном индексе из 2 столбцов.
Составной символ	Составной символ есть символ, который составляется из двух или большего числа столбцов таблицы. Составные столбцы требуют большего количества сложных связей.

Список действий	Список действий системы управления событиями есть пространство в потоке действий, где происходит назначение порядка действий для выполнения. Действие над событием может быть или в критическом, или нормальном списке действий. У каждого тэга есть свой список.
Среда выполнения	Средой выполнения называется период, когда данные переносятся модулем управления и происходит их обработка арифметическо-логическим модулем. Также это период выполнения программы.
Строка	В таблице строкой называется набор стоящих рядом столбцов информации, описывающих определенную ячейку базы данных. Например, для ячейки "личность" строка может содержать столбцы, описывающие рост, вес, цвет волос, возраст человека и т.п.
Сценарий	Сценарием называется набор выражений SQL, используемый для проведения действий над базой данных, таких как изменение данных или добавление новых объектов. Сценарии могут сохраняться как хранимые процедуры или файлы.
Счетчик пакетов	В IndustrialSQL Server количество пакетов данных, полученных системой хранения.
Счетчик строк	Расширения временного домена счетчика строк есть функция IndustrialSQL Server, которая при циклической выборке отличается от обычного счетчика строк сервера SQL. Применение счетчика строк расширения временного домена зависит от того, используется ли циклическая или же дельта-выборка и запрашивается ли широкая таблица.
Таблица	Таблицей называется группа стоящих рядом ячеек базы данных и их характеристик. См. также строка, столбец.

Таблица расширения	В IndustrialSQL Server таблицей расширения называется таблица, позволяющая обрабатывать полученные производственные данные. Эти таблицы не являются частью обычных функций SQL Server. Стандартные таблицы SQL Server хранят данные, которые непосредственно и физически расположены в устройстве данных. Таблица расширения, тем не менее, представляет данные как реально существующую таблицу, но таблица эта не существует в устройстве данных. Таблица расширения есть логическая таблица, заполненная данными из файлов, созданных системой хранения. Данные в таблице расширения управляются с помощью обычного кода Transact-SQL так же, как и специализированные расширения SQL, поставляемые с IndustrialSQL Server и называемые Industrial-SQL. <i>См. также</i> представление, таблица конфигурации.
Таблицы конфигурации	Таблицы конфигурации содержат информацию, определяющую основные аспекты работающей системы. Например, в таблицах конфигурации содержатся определения тэгов, серверов ввода-вывода, пользователей и т.п. <i>См. также</i> таблицы расширения.
Текстовое значение	Текстовым значением называется текстовое выражение, представленное как одиночный элемент данных. Текстовые значения не требуют специального синтаксиса.
Тема	В протоколах DDE или SuiteLink тема есть определяемая приложением подгруппа элементов данных. Эти протоколы используют формат наименования из трех частей для поиска информации в приложениях. Для того, чтобы приложение (например, InTouch) получало данные, ему должны быть известны имя приложения, тема и элемент. Образцом темы может служить имя электронной таблицы Excel. <i>См. также</i> Динамический обмен данными, SuiteLink.
Тип данных	Тип данных указывает, какой тип информации может содержать столбец таблицы и как эта информация сохраняется. Есть два основных типа данных: системные и пользовательские.

Транзакция	Транзакцией называется набор из одного или более командных сценариев, читающих и/или записывающих в базу данных. Транзакция есть запрос в систему IndustrialSQL Server на поиск, ввод, изменение или выдачу информации об объекте реляционной базы данных. Все транзакции выполняются во время запуска как одна рабочая единица. Если в одном из сценариев происходит ошибка, вся транзакция будет возвращена к исходному состоянию базы данных до тех пор, пока начатая транзакция не будет завершена.
Триггер	Триггером называется пакет кодов, привязанный к таблице базы данных. Триггер срабатывает при изменении данных в таблице. Триггеры используются для переноса изменений из одной таблицы базы данных в другую с целью обеспечения целостности. <i>См. также</i> целостность связей.
Тэг	Тэг определяется как элементарная переменная аналогового, дискретного, текстового или сложного типа, которая хранится в базе данных IndustrialSQL Server. Проще говоря, тэг есть средство связи с механизмом или устройством производственного участка. Также он может быть связан с системными переменными, такими как системное время (SYSTIMESEC).
Тэг события	Тэгом события называется определение события в системе. Например, если требуется зарегистрировать событие, когда температура танкера достигает 100 градусов, следует определить тэг события и назвать его "TankAt100". <i>См. также</i> детектор, действие, тэг события.
Тэги DDE	Тэг DDE есть тэг, значения которого читаются или записываются из или в другое приложение через протокол Microsoft Dynamic Data Exchange. <i>См. также</i> Динамический обмен данными.
Удаленная таблица	Удаленной называется таблица, представляющая данные для нелокального хранилища данных, доступ к которому осуществляется посредством OLE DB. <i>См. также</i> <i>хранилище данных</i> , <i>Object Linking and Embedding for Databases</i> , <i>таблица расширения</i> .
Уникальный индекс	Уникальным называется индекс, в котором нет двух строк таблицы с дублируемым значением. <i>См. также</i> индекс.
Устройство	<i>См.</i> устройство базы данных, устройство дампа, устройство журналов.

Устройство базы данных	Устройство базы данных есть файл, в котором хранятся базы данных. Одна база данных может храниться на нескольких устройствах.
Устройство дампа	Устройство дампа есть файл, который сохраняет резервную копию базы данных. <i>См. также</i> резервирование.
Устройство журнала	Устройством журнала называется файл, содержащий журнал операций с базой данных. <i>См. также</i> журнал действий.
Учетная запись	Учетная запись, или идентификатор пользователя, есть уникальное имя, используемое пользователем базы данных для входа в систему. Учетная запись может содержать до 30 буквенных или числовых символов, где первым символом должна быть буква или символы # или _.
Хранение методом исключения	<i>См. дельта-хранение.</i>
Хранилище данных	Хранилище данных есть файл, содержащий данные. Нелокальное хранилище данных есть просто хранилище данных, существующее вне базы данных Microsoft SQL Server.
Хранимая процедура	Хранимая процедура — это скомпилированная группа выражений SQL. Хранимые процедуры позволяют выполнять последовательные действия при помощи выражения SQL. Обычно хранимая процедура вызывается другой программой для выполнения; автоматически, в ответ на событие, она не выполняется. Хранимые процедуры могут использоваться в качестве ярлыков для часто используемых наборов выражений SQL, обеспечивающих дополнительную функциональность. Пользователи IndustrialSQL Server могут использовать любые хранимые процедуры системы Microsoft SQL Server, а также хранимые процедуры системы IndustrialSQL Server. Хранимые процедуры IndustrialSQL Server начинаются с ww_. Также поддерживаются пользовательские хранимые процедуры.
Целое	Целое есть любой член набора данных, состоящий из положительных, отрицательных чисел и нуля. Примеры: -59, -3, 0.
Целостность данных	Целостность данных есть совместимость и точность данных, хранимых в базе данных.

Целостность связей (RI)	Целостность связей есть механизм, удостоверяющий, что все внешние ключи имеют связанный первичный ключ во взаимодействующих таблицах. Целостность связей предотвращает указание внешних ключей на несуществующие первичные ключи и ускоряет связку таблиц. См. также первичный ключ, внешний ключ.
Циклическая выборка	Циклическая выборка есть получение сохраненных данных за заданный период времени, основанное на указанной дискретности, независимо от того, изменилось ли значение тэга (тэгов). См. также дельта-выборка, дискретность.
Циклическое хранение	Циклическое хранение есть процесс хранения данных, основанный на интервале времени. Например, можно сохранять значение аналогового тэга каждую секунду. Циклическое хранение используется для сохранения аналоговых значений, физические свойства которых производят изменения данных быстрее, чем скорость сохранения. См. также дельта-хранение.
Циклическая область хранения	Циклическая область хранения есть каталог на компьютере, используемый для хранения файлов архивных данных.
Шаблонный символ	Шаблонный символ есть клавиша, используемая для обозначения одного или нескольких символов. При поиске в базе данных сервера SQL используйте знак подчеркивания (_), знак процента (%), и скобки ([]) с ключевым словом LIKE, с которым должна совпадать выборка в базе данных. Например, для поиска всех тэгов в системе, начинающихся на "sys", задайте поиск "sys%".
Широкие таблицы	В IndustrialSQL Server широкие таблицы показывают значения одного или нескольких тэгов за период времени. Каждая строка содержит штамп даты/времени для данных и значений одного или нескольких тэгов, указанных в запросе. См. также <i>живые таблицы</i> , <i>архивные таблицы</i> .
Элемент	В протоколах DDE или SuiteLink элементом называется "заглушка" значения данных. Протокол DDE использует формат наименования из трех частей для нахождения информации между приложениями. Для того, чтобы приложение (такое как InTouch) получало данные, оно должно знать имя приложения, тему и элемент. Примером элемента служит имя ячейки в электронной таблице Excel. Другим примером может служить имя тэга в InTouch. См. также Динамический обмен данными, SuiteLink, тема.
Язык структурированных запросов (SQL)	SQL есть язык, используемый в системах реляционных баз данных для определения, поиска и управления данными.

Предметный указатель

"

"Живые" таблицы, 1-9

"Живые" таблицы расширения, 12-49

"Живые" данные, 3-5, 3-11, 3-18, 11-13

"Широкие" таблицы, 1-8

A

AccessLevel, столбец, 2-17

AcquisitionRate, столбец, 2-12

AcquisitionType, столбец, 2-12

ActionClassName, столбец, 4-5

ActionString, столбец, 2-9

ActionTypeKey, столбец, 2-8, 4-5

AlwaysAdvise, столбец, 5-6

AlwaysModifyName, столбец, 5-3

AnnotationKey, столбец, 3-8

ApplicationName, столбец, 5-3, 5-6, 9-4, 9-5

ArchiveDate, столбец, 3-16, 11-12

ArchiveLocation, столбец, 3-16, 11-12

AutoStart, столбец, 9-4

C

CalcType, столбец, 8-4, 8-7, 13-5

Checked, столбец, 10-7, 10-8

ColumnName, столбец, 6-6

ColumnType, столбец, 6-6

Comment, столбец, 5-5

Compression, столбец, 3-16, 11-12

ComputerName, столбец, 7-7, 9-4, 9-8

ConfigStor, столбец, 7-6, 7-7

Content, столбец, 3-8, 3-9

ContextKey, столбец, 10-4, 10-5, 10-7, 10-8

CreatedBy, столбец, 2-14, 7-8

CreatedDate, столбец, 7-8

CurrentEditor, столбец, 2-14

D

DateCreated, столбец, 2-14, 3-8

DateTime, столбец, 2-7, 3-4, 3-5, 3-6, 3-8, 3-9, 3-10, 3-11, 3-12, 3-17, 3-18, 6-6, 11-3, 11-4, 11-5, 11-7, 11-8, 11-9, 11-13

DbError, столбец, 9-9

DbModAcquisition, столбец, 9-9

DbModAll, столбец, 9-9

DbModServer, столбец, 9-9

DbModStorage, столбец, 9-9

DbRevision, столбец, 9-11

DbStatus, столбец, 9-9

DdeSourceKey, столбец, 5-6

DDE-тэгами, 5-3

Deadband, столбец, 10-5

DefaultStorageRate, столбец, 5-6

DefaultStorageType, столбец, 5-6

DefaultTagRate, столбец, 10-6

Description, столбец, 2-12, 3-15, 4-5, 4-7, 4-10, 5-3, 7-7, 7-8, 8-4, 8-7, 8-8, 9-5, 9-6, 9-8, 10-4, 10-7, 11-11

Description, столбец, 7-4

DetectDateTime, столбец, 2-7

DetectionTime, столбец, 13-2

DetectorClassName, столбец, 4-7

DetectorString, столбец, 2-9

DetectorTypeKey, столбец, 2-8, 4-7

DuplicateChar, столбец, 5-3

Duration, столбец, 8-7, 13-5

E

Edge, столбец, 2-9, 13-2

Edge, столбец, 2-7

Editable, столбец, 9-6

EditorClassName, столбец, 4-5, 4-7

EUKey, столбец, 2-4, 10-6

Event, столбец, 13-2

EventArchived, столбец, 3-16, 11-12

EventLogKey, столбец, 2-7, 4-6, 4-8, 4-10

EventTag, столбец, 13-5

EventTagName, столбец, 4-9

EventTime, столбец, 13-2

ExecutionMode, столбец, 4-7

ExeName, столбец, 9-5

ExeType, столбец, 9-4

F

FixedStorageRate, столбец, 5-4

Frequency column, 8-7

Frequency, столбец, 4-9

FrequencyID, столбец, 4-9, 4-12

FromDate, столбец, 3-15, 11-11

G

gid, столбец, 2-17

GroupID, столбец, 7-4, 7-8

H

HistoryArchived, столбец, 3-15, 11-11

I

Icon, столбец, 7-4

Import, столбец, 5-6

ImportMemoryTags, столбец, 5-3

ImportPlantTags, столбец, 5-3

ImportRoute, столбец, 5-4

ImportSystemTags, столбец, 5-3

Industrial-SQL, 1-10

Расширения, 14-22

IndustrialSQL Server

Безопасность, 1-4

Структуры данных, 1-2

Industrial-SQL:, 1-10

InitializationStatus, столбец, 9-5

InitialValue, столбец, 2-4, 2-6, 2-7

InPrealism, столбец, 5-5

int, 6-6

IntegerSize, столбец, 2-4

IOServerKey, столбец, 2-12, 2-16, 9-4, 9-6

ItemName, столбец, 2-12

L

LimitNameKey, столбец, 10-7

LimitType, столбец, 10-7
 Logged, столбец, 2-8
 LogKey, столбец, 8-4, 8-5
 LowerLimit, столбец, 8-8

M

MachineName, столбец, 5-3
 MajorChecked, столбец, 10-5
 MajorDeviation, столбец, 10-5
 MajorPriority, столбец, 10-5
 ManyOtherTags, столбец, 3-7, 3-12, 11-5, 11-9
 MaxEU, столбец, 2-4
 MaxMBSIZE, столбец, 9-7
 MaxRaw, столбец, 2-4
 Message0, столбец, 10-8
 Message1, столбец, 10-8
 MessageKey, столбец, 2-7, 10-8
 MinEU, столбец, 2-4
 MinMBThreshold, столбец, 9-7
 MinorChecked, столбец, 10-5
 MinorDeviation, столбец, 10-5
 MinorPriority, столбец, 10-5
 MinRaw, столбец, 2-4
 Modified, столбец, 8-4
 ModKey, столбец, 6-3, 6-4, 6-5, 6-6, 6-7
 ModType, столбец, 6-6

N

Name, столбец, 2-16, 4-5, 4-7, 6-3, 7-6, 7-7, 7-8, 9-6, 10-7
 Name, столбец, 7-4
 NamedSystemParameter, 9-6
 NameKey, столбец, 7-4, 7-6, 7-7, 7-8
 NewValue, столбец, 6-3, 6-4, 6-5, 6-6, 6-7
 NodeKey, столбец, 5-3, 5-5, 5-6
 NumericValue, столбец, 9-6

O

OldValue, столбец, 6-3, 6-4, 6-5, 6-6, 6-7
 OLE DB, удаленные таблицы, 11-2
 OnLine, столбец, 3-15, 11-11
 Open Data Services, 14-2
 OperationEnd, столбец, 8-6
 OperationKey, столбец, 8-5, 8-7, 8-8
 OperationStart, столбец, 8-6
 OriginalName, столбец, 5-5

P

ParentKey, столбец, 7-6, 7-7, 7-8
 Path, столбец, 5-3, 9-4, 9-7
 Periodicity, столбец, 4-12
 PKey, столбец, 7-8
 Platform, столбец, 9-5
 PostDetectorDelay, столбец, 2-9
 PrefixOrSuffix, столбец, 5-3
 Priority, столбец, 2-9, 10-7, 10-8
 ProtocolType, столбец, 5-6, 9-5
 Provider, столбец, 2-6

Q

Quality, столбец, 3-4, 3-5, 3-9, 3-10, 3-11, 3-17, 3-18, 4-6, 4-8, 4-10, 8-4, 11-3, 11-4, 11-7, 11-8, 11-13, 13-3, 13-5

QualityDetail, столбец, 3-4, 3-5, 3-9, 3-10, 3-11, 3-17, 3-18, 4-6, 4-8, 4-10, 9-6, 11-3, 11-4, 11-7, 11-8, 11-13, 13-3
 QualityString, столбец, 9-6

R

RawType, столбец, 2-4
 RequestInitialData, столбец, 5-6
 Resolution, столбец, 8-7, 13-5
 Retriever, столбец, 2-6
 Revision, столбец, 9-5
 RunTimeDay, столбец, 4-12
 RunTimeHour, столбец, 4-12
 RunTimeMin, столбец, 4-12

S

SamplesInActiveImage, столбец, 2-14
 Scaling, столбец, 2-4
 ScanRate, столбец, 2-8
 Script, столбец, 4-10
 SELECT INTO, 14-29
 Sequence, столбец, 3-16, 11-12
 ServerKey, столбец, 7-7
 ServerKey, строка, 2-15
 SET, операторы, 1-10
 SignedInteger, столбец, 2-5
 smallint, 6-6
 SnapshotTagKey, столбец, 4-6, 4-8, 4-9, 4-10
 SortOrder, столбец, 9-7
 SourceName, столбец, 5-6
 SourceType column, 8-7
 StartDateTime, столбец, 4-12
 Status, столбец, 2-9, 8-6
 StorageAreaType, столбец, 3-16, 11-12
 StorageNodeKey, столбец, 2-12, 2-16, 3-15, 9-4, 9-6, 9-7, 9-8, 11-11
 StorageRate, столбец, 2-12
 StorageType, столбец, 2-12, 9-7
 StringValue, столбец, 9-6
 SumDateCalcType, столбец, 8-5
 SumDateDuration, столбец, 8-5
 SumDateResolution, столбец, 8-6
 SumDateTimeStamp, столбец, 8-5
 SummaryArchived, столбец, 3-15, 11-11
 SummaryDate, столбец, 8-5, 13-5
 SumVarKey, столбец, 8-4, 8-8
 SysPulse, столбец, 3-12

T

TagName, столбец, 2-4, 2-6, 2-7, 2-8, 2-12, 3-4, 3-5, 3-8, 3-9, 3-10, 3-11, 3-17, 3-18, 4-9, 4-12, 5-5, 8-7, 8-8, 9-11, 10-5, 10-7, 10-8, 11-3, 11-4, 11-7, 11-8, 11-13, 13-2, 13-5
 TagName, строка, 2-15
 TagType, столбец, 2-14, 4-9
 TagType, строка, 2-15
 Target, столбец, 10-5
 TemplateKey, столбец, 4-10
 TimeBase, столбец, 10-8
 TimeDeadband, столбец, 2-8, 2-14, 5-6
 TimeDetectorDetailKey, столбец, 4-12
 TimeOut, столбец, 2-16
 TimeStamp, столбец, 8-7, 13-5
 tinyint, 6-6
 ToDate, столбец, 3-15, 11-11

TopicKey, столбец, 2-12, 2-16
 TopicName, столбец, 5-6
 Triggerval, столбец, 7-4
 Type, столбец, 2-6, 4-10, 6-3, 7-4, 7-6, 7-7, 7-8, 9-6
 TypeInfo, столбец, 5-5

U

uid, столбец, 2-17
 Unit, столбец, 10-6
 UpperLimit, столбец, 8-8
 UserKey, столбец, 2-17, 3-8, 6-6, 7-4, 7-6, 9-11
 UserName, столбец, 2-17
 UseThreadPool, столбец, 2-9

V

v_AnalogHistory, представление данных, 13-7
 v_AnalogLive, представление данных, 13-7
 v_DiscreteHistory, представление данных, 13-7
 v_DiscreteLive, представление данных, 13-7
 v_EventSnapshot, представление, 13-2
 v_HistoryBlock, представление данных, 13-7
 v_StringLive, представление данных, 13-7
 v_SummaryData, представление данных, 13-5
 Value, столбец, 3-4, 3-5, 3-8, 3-10, 3-11, 3-17, 3-18, 4-6, 4-8, 4-10, 8-4, 10-7, 10-8, 11-3, 11-4, 11-7, 11-8, 11-13, 13-2, 13-5
 ValueDeadband, столбец, 2-4
 Version, столбец, 3-16, 11-12

W

WideColumnName, столбец, 2-14
 ww_ActionStringSelect, 12-2
 ww_AddTag, 12-47
 ww_AnalogConfig, 12-2
 ww_AnalogCount, 12-3
 ww_AnalogDetail, 12-47
 ww_AnalogTagDelete, 12-3
 ww_AnalogTagInsert, 12-4
 ww_AnalogTagSelect, 12-4
 ww_AnalogTagUpdate, 12-5
 ww_Annotation, 12-48
 ww_AnnotationDelete, 12-5
 ww_AnnotationInsert, 12-6
 ww_AnnotationSelect, 12-6
 ww_AnnotationUpdate, 12-7
 ww_CheckClientVersion, 12-7
 ww_CheckWhichDb, 12-8
 ww_dbCheck, 12-8
 ww_DeleteOlderEvents, 12-8
 ww_DeleteOlderStrings, 12-9
 ww_DeleteOlderSummaries, 12-9
 ww_DeleteTag, 12-48
 ww_DetectorStringSelect, 12-9
 ww_DiscreteConfig, 12-10
 ww_DiscreteCount, 12-10
 ww_DiscreteDetail, 12-48
 ww_DiscreteTagDelete, 12-10
 ww_DiscreteTagInsert, 12-11
 ww_DiscreteTagSelect, 12-11

ww_DiscreteTagUpdate, 12-12
 ww_EngineeringUnit, 12-12
 ww_EngineeringUnitInsert, 12-13
 ww_EngineeringUnitSelect, 12-13
 ww_EngineeringUnitUpdate, 12-14
 ww_EventHistoryInsert, 12-14, 12-15
 ww_EventTagDelete, 12-15, 12-16
 ww_EventTagInsert, 12-16
 ww_EventTagSelect, 12-17
 ww_EventTagSelectAll, 12-17
 ww_EventTagSelectDeleted, 12-17
 ww_EventTagSelectDisabled, 12-18
 ww_EventTagSelectInserted, 12-18
 ww_EventTagSelectUpdated, 12-18
 ww_EventTagUpdate, 12-19
 ww_GetDbRevision, 12-19
 ww_GetInSQLProcedureNames, 12-19
 ww_GetLastTagKey, 12-20
 ww_InsertStringSnapshot, 12-20
 ww_InSQLNSExpand, 12-20
 ww_InSQLStatusSelect, 12-21
 ww_InSQLStatusSet, 12-21
 ww_IOServerDelete, 12-22
 ww_IOServerInsert, 12-22
 ww_IOServerSelect, 12-23
 ww_IOServerUpdate, 12-23
 ww_LoadInSQLProcedureBody, 12-24
 ww_MessageDelete, 12-24
 ww_MessageInsert, 12-24
 ww_MessageSelect, 12-25
 ww_MessageUpdate, 12-25
 ww_ModLogStatus, 12-25
 ww_PrivateNSAddGroup, 12-26
 ww_PrivateNSAddLeaf, 12-26
 ww_PrivateNSDeleteGroup, 12-26
 ww_PrivateNSDeleteLeaf, 12-27
 ww_PrivateNSExpand, 12-27
 ww_PrivateNSSelect, 12-27
 ww_PrivateNSUpdateGroup, 12-28
 ww_PublicNSAddGroup, 12-28
 ww_PublicNSAddLeaf, 12-29
 ww_PublicNSDeleteGroup, 12-29
 ww_PublicNSDeleteLeaf, 12-29
 ww_PublicNSExpand, 12-30, 12-41
 ww_PublicNSSelect, 12-30
 ww_PublicNSUpdateGroup, 12-30
 ww_RedirectToInTouch, 12-31
 ww_ResetDefaults, 12-49
 ww_SetAISamples, 12-31
 ww_SetAnalogCyclic, 1-10, 12-49
 ww_SetContinuous, 1-10, 12-49
 ww_SetDiscreteCyclic, 1-10, 12-50
 ww_SetDuration, 1-10, 12-50
 ww_SetEdgeDetection, 1-10, 12-51
 ww_SetEndDate, 1-10, 12-52
 ww_SetResolution, 1-10, 12-52
 ww_SetRowCount, 1-10, 12-53
 ww_SetTimeDeadband, 1-10, 12-53
 ww_SetValueDeadband, 1-10, 12-54
 ww_SnapToSummary, 12-32
 ww_SpaceManager, 12-32
 ww_StorageLocationSelect, 12-33
 ww_StorageLocationUpdate, 12-33
 ww_StringConfig, 12-33
 ww_StringCount, 12-34
 ww_StringDetail, 12-54

ww_StringTagDelete, 12-34
 ww_StringTagInsert, 12-35
 ww_StringTagSelect, 12-35
 ww_StringTagUpdate, 12-36
 ww_SummaryDetail, 12-37
 ww_SummaryOperationDelete, 12-37
 ww_SummaryOperationInsert, 12-38
 ww_SummaryOperationSelect, 12-38
 ww_SummaryOperationUpdate, 12-39
 ww_SummaryTagListDelete, 12-39
 ww_SummaryTagListInsert, 12-40
 ww_SummaryTagListSelect, 12-40
 ww_SummaryTagListUpdate, 12-41
 ww_SystemNSExpand2, 12-42
 ww_TagInfo, 12-42
 ww_TagType, 12-42
 ww_TimeDetectorDetailInsert, 12-43
 ww_TimeDetectorDetailSelect, 12-43
 ww_TimeDetectorDetailUpdate, 12-44
 ww_TopicDelete, 12-44
 ww_TopicInsert, 12-45
 ww_TopicSelect, 12-45
 ww_TopicUpdate, 12-46
 ww_UserAccessLevelSelect, 12-55
 ww_UserDetailUpdate, 12-46
 wwAdmin, 1-4
 wwdbo, 1-4
 wwDomainTagKey, столбец, 7-4, 7-6
 wwDomainTagKey, строка, 2-15
 wwEdgeDetection, столбец, 11-3, 11-5, 11-7, 11-9, 11-13, 11-14
 wwParameters, столбец, 11-4, 11-5, 11-7, 11-8, 11-9, 11-13, 11-14
 wwPower, 1-4
 wwResolution, столбец, 11-3, 11-5, 11-7, 11-14
 wwRetrievalMode, столбец, 11-3, 11-4, 11-5, 11-7, 11-8, 11-9, 11-13, 11-14
 wwRowCount, столбец, 11-3, 11-5, 11-7, 11-14
 wwTagKey, столбец, 2-12, 3-4, 3-5, 3-10, 3-11, 3-18, 11-3, 11-4, 11-7, 11-8, 11-13
 wwTagKey, строка, 2-15
 wwTimeDeadband, столбец, 11-3, 11-4, 11-5, 11-7, 11-8, 11-9, 11-13, 11-14
 wwUser, 1-4
 wwValueDeadband, столбец, 11-4, 11-5, 11-14

X

xp_AnalogHistory, 12-57, 12-58
 xp_AnalogWideHistory, 12-59
 xp_AnalogWideHistoryDelta, 12-60
 xp_DiscreteHistory, 12-61
 xp_DiscreteHistoryDelta, 12-62
 xp_DiscreteWideHistory, 12-63, 12-64
 xp_DiskCopy, 12-65
 xp_NewHistoryBlock, 12-65
 xp_proclist, 12-57
 xp_SetStorageTimeDeadband, 12-66
 xp_SetStorageValueDeadband, 12-66

A

Активный объект, 12-31
 Аналоговая "живая" таблица, 1-9
 Аналоговые "широкие" архивные таблицы, 1-8
 Аналоговый тэг, 12-4, 12-5, 12-47
 Аннотация, 3-8, 12-5, 12-6, 12-7, 12-48, 14-21

Архив событий, 12-14
 Архивные блоки
 Копирование, 12-65
 Архивные данные, 1-3
 Архивные таблицы, 1-7, 1-8, 1-9
 Активный объект, 2-14
 Аналоговый снимок, 4-6
 Аналоговый тэг, 3-4, 3-5, 3-6, 3-16, 11-3, 11-4, 11-5
 Определение края, 14-58
 Арифметика даты, 14-75
 Арифметическая функция, 14-16
 Архив событий, 13-2
 Архивные таблицы, 1-8, 14-11
 Архивный блок, 3-15, 11-11
 Архив событий, 12-15

B

База данных
 Доступ, 1-4
 Запросы, 14-7
 Изменение, 12-21
 Проверка, 12-8
 База данных выполнения, 1-2, 1-3, 12-8
 База данных накопления, 1-2, 12-8
 Использование, 1-2
 Безопасность, 1-4
 Битовая величина, 2-4
 Битовый тип, 6-4

V

Версия, 12-7
 Временная мертвая зона, 12-53, 12-66
 Взаимосвязь объектов, 1-2
 Виртуальная таблица, 13-2
 Внутренние тэги, 5-3
 Временная мертвая зона, 14-31, 14-36
 Выборка
 Временная мертвая зона, 14-36, 14-39
 Дельта-метод, 14-34
 Циклическая, 14-32
 Выражение CONVERT, 14-76
 Выражение GROUP BY, 14-2
 Выражение JOIN, 14-19, 14-20
 Выражение LIKE, 14-8
 Выражение OR, 14-76
 Выражение ORDER BY, 14-2
 Выражение WHERE, 14-9, 14-27
 Выражение SQL, 12-2
 Выражения OR, 14-14
 Выражения SET, 14-22

G

Граница, 8-8, 14-21
 Группа тэгов, 7-4, 7-8

D

Данные конфигурации, 1-3
 Дата окончания, 12-50, 12-52
 Двоичный тип, 6-3
 Дельта-метод, 12-49, 12-54
 Дельта-метод, 12-53
 Дельта-хранение, 12-66
 Детали настройки, 12-33
 Детектор времени, 12-43, 12-44
 Дискретная "живая" таблица, 1-9

Дискретные "широкие" архивные таблицы, 1-8
 Дискретный тэг, 12-10, 12-11, 12-12, 12-48
 Действие, 2-8, 2-9, 4-5
 Действительный тип, 6-7
 Действия по отношению к событию, 2-9
 Дельта-выборка, 14-34, 14-36, 14-39, 14-42, 14-43
 Дельта-данные, 2-14
 Дельта-метод, 12-50, 14-15
 Дельта-выборка данных
 Таблицы OLE DB, 11-14
 Десятичный тип, 6-5
 Детали, 14-7
 Детали качества, 9-6
 Детектор, 2-8, 4-7, 4-12
 Детектор времени, 4-12, 4-14
 Дискретность, 1-8, 8-6, 12-52, 12-54, 12-57, 12-59,
 12-61, 12-63, 13-5, 14-25, 14-42, 14-50, 14-54
 Таблицы OLE DB, 11-14
 Дискретный снимок, 4-8
 Дискретный тэг, 2-7, 3-10, 3-11, 3-12, 3-16, 10-8,
 11-7, 11-8, 11-9
 Определение края, 14-63
 Длина, 2-11
 Длина текстового тэга, 2-11, 1-4

Е

Единица измерения, 2-4, 10-6, 12-12, 12-13, 12-14

Ж

Живые данные, 11-4
 Живые таблицы, 14-12, 14-31

З

Задание SQL, 2-17
 Запросы, 14-7
 Значение
 Мертвая зона, 12-66
 Значение мертвой зоны, 12-54, 14-39

И

Иерархия именных пространств, 2-2
 Изменение тэга, 2-9
 Идентификатор пользователя, 1-4
 Изменение, 12-21
 Импортирование данных, 1-2
 Имя пользователя, 1-4
 Интеграционные таблицы InTouch, 1-3
 Интервал, 12-53
 Информация тэга, 12-42
 Именные пространства, 1-3, 12-20
 Импорт, 5-5
 Импортирование, 5-4
 Импортированные тэги, 2-11
 Имя пользователя базы данных, 2-17
 Индикатор качества, 4-6
 Инженерная единица. См. Единица измерения
 Интеграционные таблицы InTouch, 5-2
 Интервал выборки, 12-52
 Исходное значение, 2-11

К

Код Transact-SQL, 14-2
 Количество строк
 Таблицы OLE DB, 11-14
 Конечная дата, 14-27

Критерии
 Фиксация событий, 2-9
 Курсоры, 14-73

Л

Личное именное пространство, 7-4, 7-6, 12-26, 12-26, 12-27, 12-28

М

Максимум, 14-18
 Математические функции, 14-2
 Мертвая зона, 2-4, 10-5, 14-31, 14-36, 14-39
 Для событий, 2-8
 Мертвая зона значения, 11-14
 Место хранения, 9-7, 12-33
 Минимум, 14-18
 Мертвая зона, 12-53, 12-54, 12-66

Н

Название столбца, 2-14
 Непрерывный запрос, 12-49

О

Общее именное пространство, 7-6, 7-7
 Общие таблицы, 2-3
 Объект, 6-3
 Ограничение, 10-4, 10-5, 10-7
 Области хранения, 12-33
 Область хранения сводок, 12-9
 Область хранения событий, 12-8
 Область хранения текстовых данных, 12-9
 Общие именные пространства, 12-28, 12-29, 12-30
 Определение края, 12-51
 Отслеживание изменений, 12-25
 Оператор остатка, 14-76
 Операторы сравнения, 14-43, 14-50, 14-54
 Операции создания сводок, 4-9, 8-5
 Определение края
 Таблицы OLE DB, 11-14
 Определения края, 14-58
 Опрошенные тэги, 2-12
 Отклонение, 10-4, 10-5

П

Пара сообщений, 10-8, 12-46
 Пароль, 1-4
 Переменно-символьный тип, 6-4
 Повторная инициализация, 9-11
 Поддержка Transact-SQL, 14-2
 Подсчет, 8-4, 13-5
 Подсчет сводки, 8-7
 Подсчет сводок, 8-4
 Подсчет строк, 12-53
 Показатель качества, 11-4
 Показатель качества, 3-4, 3-5, 3-9, 3-10, 3-11, 3-17,
 3-18, 4-8, 4-10, 8-4, 11-3, 11-7, 11-8, 11-13, 13-3
 Полученное необработанное значение, 2-4
 Пользователи
 Базы данных, 9-11
 Пользователь, 2-17
 Пользователь IndustrialSQL Server, 2-17
 Порог, 9-7
 Поставщик OLE DB
 Представления данных, 13-6
 Поставщик данных OLE DB, 14-3

Неподдерживаемый синтаксис, 14-75
 Ограничения, 14-6
 Потоки, 2-9
 Повторная инициализация, 12-17, 12-18
 Пользователь, 12-55
 Пользовательские хранимые операции, 12-47
 Приложение-клиент, 12-7
 Продолжительность, 12-50
 Промежуток времени, 12-52
 Процент, 12-54
 Предметные области, 2-2
 Представление данных, 13-2
 Представления
 OLE DB, 11-2
 Представления данных
 OLE DB, 14-4
 Поставщик OLE DB, 13-6
 Представления данных системы управления
 событиями, 13-2
 Приложение InTouch, 5-3, 5-6
 приложения-клиенты, 12-19
 Продолжительность, 14-27
 Продолжительный запрос, 14-31
 Производственные данные, 1-7
 Производственные тэги, 5-3
 Процент, 11-14
 Псевдонимы таблиц, 14-10

Р

Расширения, 14-22
 , 1-7, 14-2
 , 1-7, 14-5, 14-22
 Расширения реального времени, 14-5
 Реляционные базы данных, 1-2
 Роль пользователя, 1-4
 Расширения временного домена, 1-10
 Расширенные хранимые операции, 12-56
 Редакция, 12-19

С

Сводка, 8-4
 Связанный сервер, 14-3
 Сервер, 7-7
 Сервер базы данных, 14-2
 Сервер ввода-вывода, 2-12, 2-16, 9-4, 9-5
 Сервер ввода-вывода, 12-22, 12-23, 12-31
 Сервер ввода-вывода, 12-22
 Символьный тип, 6-4
 Системное именованное пространство, 7-7, 7-8
 Системные потоки, 2-9
 Системные тэги, 5-3
 Системный параметр, 9-6
 Скорость изменения, 10-4, 10-5
 Скорость опроса, 2-12
 Скорость сохранения, 5-4, 5-6
 Сложный тэг, 3-9
 Событие, 2-7
 События, 2-8
 Действия, 2-9
 Распознавание, 14-58
 Регистрация, 2-8
 Скорость, 2-8
 Таблицы сводок, 2-2, 8-2
 Таблицы хранения, 2-2
 Таблицы хранения событий, 4-2

Создание сводки, 8-7, 8-8, 12-38, 12-39
 Сообщение, 2-7, 12-24, 12-25
 Состояние аларма, 10-5
 Сохранение дельта-данных, 2-4
 Список действий, 2-9
 Среднее, 14-18
 Сжатие
 Архивные блоки, 12-65
 Система управления событиями, 12-32
 Таблицы сводок, 1-3
 Таблицы системы управления событиями, 1-3
 Система хранения данных, 1-7
 Системные хранимые операции, 12-2
 Системные именованные пространства, 12-41, 12-42
 Скорость, 12-52
 Событие, 12-20
 События, 12-51
 Сохранение данных, 12-53
 Стартовая дата, 12-50
 Строка действия, 12-2
 Строка детектора, 12-9
 Стартовая дата, 14-27
 Строка адреса, 2-12
 Структуры данных
 IndustrialSQL Server, 1-2
 Сумма, 14-18
 СУРБД, 1-2
 Счетчик строк, 14-22, 14-42, 14-50

Т

Тэг, 12-48
 Таблица ActionType, 4-5
 Таблица AnalogHistory, 1-8, 3-4
 Таблица AnalogHistory (OLE DB), 11-3
 Таблица AnalogLive, 3-5
 Таблица AnalogLive (OLE DB), 11-4
 Таблица AnalogSnapshot, 4-6
 Таблица AnalogTag, 2-4
 Таблица AnalogWideHistory, 3-6
 Таблица AnalogWideHistory (OLE DB), 11-5
 Таблица Annotation, 3-8
 Таблица CalcType, 8-4
 Таблица ComplexHistory, 3-9
 Таблица ComplexTag, 2-6
 Таблица Context, 10-4
 Таблица DetectorType, 4-7
 Таблица Deviation, 10-5
 Таблица DiscreteTag, 2-7
 Таблица DiscreteHistory, 1-8, 3-10
 Таблица DiscreteHistory (OLE DB), 11-7
 Таблица DiscreteLive, 3-11
 Таблица DiscreteLive (OLE DB), 11-8
 Таблица DiscreteSnapshot, 4-8
 Таблица DiscreteWideHistory, 3-12
 Таблица DiscreteWideHistory (OLE DB), 11-9
 Таблица EngineeringUnit, 10-6
 Таблица EventHistory, 2-7
 Таблица EventTag, 2-8
 Таблица EventTagPendingDelete, 4-9
 Таблица Frequency, 4-9
 Таблица GroupTagList, 7-4
 Таблица HistoryBlock, 3-15
 Таблица HistoryBlock (OLE DB), 11-11
 Таблица InSQLSysObjects, 6-3
 Таблица InTouchNode, 5-3

Таблица InTouchSpecific, 5-5
 Таблица IOServer, 9-4
 Таблица IOServerType, 9-5
 Таблица Limit, 10-7
 Таблица LimitName, 10-7
 Таблица ManualAnalogHistory, 3-16
 Таблица ManualDiscreteHistory, 3-16
 Таблица Message, 10-8
 Таблица ModBinary, 6-3
 Таблица ModBit, 6-4
 Таблица ModChar, 6-4
 Таблица ModDatetime, 6-5
 Таблица ModDecimal, 6-5
 Таблица ModInteger, 6-6
 Таблица ModLog, 6-6
 Таблица ModReal, 6-7
 Таблица NamedSystemParameter, 9-6
 Таблица NameSpaceIcons, 7-4
 Таблица PrivateGroupTag, 7-4
 Таблица PrivateNameSpace, 7-6
 Таблица PublicGroupTag, 7-6
 Таблица PublicNameSpace, 7-7
 Таблица QualityMap, 9-6
 Таблица RateOfChange, 10-8
 Таблица ServerList, 7-7
 Таблица SnapshotTag, 4-9
 Таблица SQLTemplate, 4-10
 Таблица StorageLocation, 9-7
 Таблица StorageNode, 9-8
 Таблица StringHistory, 3-17
 Таблица StringLive, 3-18
 Таблица StringLive (OLE DB), 11-13
 Таблица StringSnapshot, 4-10
 Таблица SummaryData, 8-4
 Таблица SummaryHistory, 8-5
 Таблица SummaryOperation, 8-7
 Таблица SummaryTagList, 8-8
 Таблица SystemNameSpace, 7-8
 Таблица Tag, 2-12
 Таблица TagGroup, 7-8
 Таблица TimeDetectorDetail, 4-12
 Таблица TimeDetectorDetailPendingDelete, 4-14
 Таблица Topic, 2-16
 Таблица TopicImportInfo, 5-6
 Таблица WideTableDictionary, 9-11
 Таблица архива, 11-5, 11-7, 11-8, 11-9
 Таблица имен тэгов, 2-11
 Таблица *AnalogHistory*, 12-57
 Таблица *AnalogWideHistory*, 12-59
 Таблицы StringTag, 2-11
 Таблицы TagRef, 2-15
 Таблицы базы данных, 1-6
 Таблицы именных пространств и групп, 1-3, 2-2, 7-2
 Таблицы интеграции InTouch, 2-2
 Таблицы конфигурации системы, 2-2
 Таблицы определения тэгов, 10-2
 Таблицы отслеживания изменений, 2-2, 6-2
 Таблицы расширения, 14-2, 14-5
 Ограничения, 14-2
 Таблицы сводок, 8-2
 Таблицы системной конфигурации, 9-2
 Таблицы системы управления событиями, 13-2
 Таблицы хранения данных, 1-3, 2-2, 3-2
 Таблицы хранения событий, 4-2
 Табличные псевдонимы, 14-10

Текстовые функции, 14-2
 Текстовый снимок, 4-10
 Текстовый тэг, 3-17, 3-18, 4-10, 11-13
 Тема, 2-12, 2-16, 5-6
 Тип данных datetime, 6-5
 Тип данных smalldatetime, 6-5
 Тип значений с плавающей точкой, 6-7
 Тип сохранения, 5-6
 Таблица *AnalogHistory*, 12-58
 Таблица *AnalogWideHistory*, 12-60
 Таблица *DiscreteHistory*, 12-61, 12-62
 Таблица *DiscreteWideHistory*, 12-63, 12-64
 Таблицы конфигурации, 1-3, 1-6
 Таблицы определения тэгов, 1-3
 Таблицы отслеживания изменений, 1-3
 Таблицы расширения, 1-6, 12-19, 12-24
 Описание, 1-7
 Текстовый снимок, 12-20
 Текстовый тэг, 12-33, 12-34, 12-35, 12-36, 12-54
 Тема, 12-44, 12-45, 12-46
 Тип тэга, 12-42
 Тэг, 12-47
 Создание сводки, 12-39
 Создание сводки, 12-41
 Тэг снимка, 12-15, 12-20
 Тэг события, 12-9, 12-15, 12-16, 12-17, 12-19, 12-43
 Тэг, 12-20
 Тэг события, 12-16
 Тэги
 Ограничения столбцов SQL, 1-9
 Тэг сводки, 8-8, 13-5
 Тэг снимка, 4-9
 Тэг события, 2-9, 8-7
 Тэги
 Допустимые символы, 10-2
 Таблицы определения, 2-2, 10-2

У

Удаленные таблицы, 11-2, 14-5
 Узел InTouch, 1-2, 5-3, 12-31
 Узел хранения, 2-12, 2-16, 3-15, 9-7, 9-8
 Уровень приоритета, 2-9, 10-5, 10-7, 10-8
 Учет регистра, 1-2, 12-56
 Условия аларма, 10-5
 Устройство базы данных, 1-6
 Устройство данных, 1-7
 Удаленные таблицы, 1-6
 Удаленные таблицы
 Описание, 1-7
 Узел хранения, 12-2, 12-3, 12-10, 12-33, 12-34
 Управление дисковым пространством, 12-32
 Уровень доступа, 12-55
 Учетная запись пользователя, 1-4

Ф

Функции агрегации, 14-18
 Функции даты, 14-17
 Функция OPENQUERY, 14-4, 14-77
 Функция OPENROWSET, 14-4

Х

Хранимые операции, 14-22
 Запросы OLE DB, 14-5
 Создание, 12-67
 Хранение данных, 1-7

Хранимая процедура
Описание, 12-2

Ц

целое число, 2-5
Целый тип, 6-6
Циклическая выборка, 12-50, 14-32, 14-42, 14-50,
14-54
Циклическая выборка,
Таблицы OLE DB, 11-14
Циклическое хранение, 2-12, 9-7
Циклический метод, 12-49

Ч

Частота, 4-9, 4-12

Ш

Шаблон SQL, 4-10
Широкие таблицы, 14-13, 14-76