

Метаданные SQL Server

[В основном справочный материал]

Метаданные, в общем случае, это данные о данных, информация об информации, описание контента. Каждая СУБД сохраняет метаданные обо всех сущностях базы данных. Так в SQL Server с помощью инструкции CREATE можно создать 52 сущности:

- CREATE AGGREGATE
- CREATE FULLTEXT INDEX
- CREATE SEARCH PROPERTY LIST (Transact-SQL)
- CREATE APPLICATION ROLE
- CREATE FULLTEXT STOPLIST
- CREATE SEQUENCE (Transact-SQL)
- CREATE ASSEMBLY
- CREATE FUNCTION
- CREATE SERVER AUDIT
- CREATE ASYMMETRIC KEY
- CREATE INDEX
- CREATE SERVER AUDIT SPECIFICATION
- CREATE BROKER PRIORITY
- CREATE LOGIN
- CREATE SERVICE
- CREATE CERTIFICATE
- CREATE MASTER KEY
- CREATE SPATIAL INDEX
- CREATE COLUMNSTORE INDEX
- CREATE MESSAGE TYPE
- CREATE STATISTICS
- CREATE CONTRACT
- CREATE PARTITION FUNCTION
- CREATE SYMMETRIC KEY
- CREATE CREDENTIAL
- CREATE PARTITION SCHEME
- CREATE SYNONYM
- CREATE CRYPTOGRAPHIC PROVIDER
- CREATE PROCEDURE
- CREATE TABLE
- CREATE DATABASE
- CREATE QUEUE
- CREATE TRIGGER
- CREATE DATABASE AUDIT SPECIFICATION
- CREATE REMOTE SERVICE BINDING
- CREATE TYPE
- CREATE DATABASE ENCRYPTION KEY
- CREATE RESOURCE POOL
- CREATE USER
- CREATE DEFAULT
- CREATE ROLE
- CREATE VIEW
- CREATE ENDPOINT
- CREATE ROUTE
- CREATE WORKLOAD GROUP
- CREATE EVENT NOTIFICATION
- CREATE RULE
- CREATE XML INDEX
- CREATE EVENT SESSION
- CREATE SCHEMA
- CREATE XML SCHEMA COLLECTION
- CREATE FULLTEXT CATALOG

В разных СУБД применяются разные названия для метаданных - системный каталог, словарь данных и др. Однако общим свойством всех современных реляционных СУБД является то, что каталог/словарь сам состоит из таблиц, а точнее - системных таблиц. В результате пользователь может обращаться к метаданным так же, как и к прикладным данным, используя инструкцию SELECT. Изменения же в каталоге/словаре производятся автоматически при выполнении пользователем инструкций, изменяющих состояние объектов базы данных. Системные таблицы не должны изменяться непосредственно ни одним пользователем. Например, не стоит изменять системные таблицы с помощью инструкций DELETE, UPDATE или INSERT либо с помощью пользовательских триггеров. Обращение к документированным столбцам системных таблиц разрешено. Однако многие столбцы системных таблиц не документированы. В приложениях непосредственные запросы к недокументированным столбцам применять не следует. Чтобы исключить прямой доступ к системным таблицам, пользователь «видит» не сами таблицы, а созданные на их базе представления, которые он, конечно же, не может изменять. Состав и структура каталога/словаря очень различны для различных СУБД. Ограничимся рассмотрением метаданных и способов доступа к ним на примере СУБД MS SQL Server.

Microsoft SQL Server предоставляет следующие коллекции системных представлений, содержащие метаданные:

- Представления информационной схемы
- Представления каталога
- Представления совместимости
- Представления репликации
- Динамические административные представления и функции
- Представления приложения уровня данных (DAC)

Представления информационной схемы определяются в особой схеме с именем INFORMATION_SCHEMA. Эта схема содержится в любой базе данных и состоит из 20 представлений:

- CHECK_CONSTRAINTS
- REFERENTIAL_CONSTRAINTS
- COLUMN_DOMAIN_USAGE
- ROUTINES
- COLUMN_PRIVILEGES
- ROUTINE_COLUMNS
- COLUMNS
- SCHEMATA
- CONSTRAINT_COLUMN_USAGE
- TABLE_CONSTRAINTS
- CONSTRAINT_TABLE_USAGE
- TABLE_PRIVILEGES
- DOMAIN_CONSTRAINTS
- TABLES
- DOMAINS
- VIEW_COLUMN_USAGE
- KEY_COLUMN_USAGE
- VIEW_TABLE_USAGE
- PARAMETERS
- VIEWS

Каждое представление информационной схемы содержит метаданные для объектов, хранящихся в этой конкретной базе данных. Представления информационной схемы, включенные в SQL Server, соответствуют стандартному определению ISO для INFORMATION_SCHEMA. При ссылке на представления информационной схемы необходимо использовать полное имя, включающее имя схемы INFORMATION_SCHEMA. Например:

```
SELECT TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, COLUMN_NAME, COLUMN_DEFAULT
FROM Northwind.INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = N'Products';
GO
```

Представления каталога имеют наиболее универсальный интерфейс к метаданным каталога и предоставляют наиболее эффективный способ для получения этих данных. Некоторые представления каталога наследуют строки других представлений каталога. Например, представление каталога [sys.tables](#) наследует строки из представления каталога [sys.objects](#). Представление каталога sys.objects называется базовым представлением, а представление sys.tables называется производным представлением. Представление каталога sys.tables возвращает столбцы, определенные для

таблиц, а также все столбцы, которые возвращает представление каталога sys.objects. Представление каталога sys.objects возвращает строки для объектов, отличных от таблиц, например для хранимых процедур или представлений.

Представления каталога в SQL Server организованы в следующие 25 категории:

- Представления каталога групп доступности AlwaysOn
- Представления каталога связанных серверов
- Представления каталога отслеживания изменений
- Представления каталога сообщений (для ошибок)
- Представления каталога сборки среды CLR
- Представления каталога объектов
- Представления каталога баз данных и файлов
- Представления каталога функции секционирования
- Представления компонента Database Mail
- Представления управления на основе политик
- Представления каталога зеркального отображения базы данных
- Представления каталога регулятора ресурсов
- Представления сборщика данных
- Представления каталога скалярных типов
- Пространства данных
- Представления каталога схем
- Представления каталога конечных точек
- Представления каталога безопасности
- Представления каталога расширенных событий
- Представления каталога компонента Service Broker
- Представления каталога расширенных свойств
- Представления каталога конфигурации уровня сервера
- Представления каталога FileTable
- Представления каталога схем XML (система типов XML)
- Представления каталога полнотекстового и семантического поиска

Представления каталогов баз данных и файлов содержит следующие 6 представления:

- sys.backup_devices
- sys.database_files
- sys.database_mirroring
- sys.database_recovery_status
- sys.databases
- sys.master_files

sys.databases - содержит одну строку для каждой базы данных в экземпляре Microsoft SQL Server. Чтобы проверить существование базы данных можно воспользоваться предикатом EXISTS и запросом на выборку из представления sys.databases. Следующий пример проверяет существование указанной базы данных. Если база данных существует, то она удаляется. Если базы данных не существует, то инструкция DROP DATABASE не выполняется.

```
USE [master]
GO
IF EXISTS (SELECT name FROM sys.databases WHERE name = N'FolktaleDB')
    DROP DATABASE [FolktaleDB]
GO
```

Представления каталога объектов содержит следующие 36 представления:

- sys.allocation_units
- sys.parameters
- sys.assembly_modules
- sys.partitions
- sys.check_constraints
- sys.procedures
- sys.columns
- sys.sequences
- sys.computed_columns
- sys.service_queues
- sys.default_constraints

- sys.sql_dependencies
- sys.events
- sys.sql_expression_dependencies
- sys.event_notifications
- sys.sql_modules
- sys.extended_procedures
- sys.stats
- sys.foreign_key_columns
- sys.stats_columns
- sys.foreign_keys
- sys.synonyms
- sys.function_order_columns
- sys.table_types
- sys.identity_columns
- sys.tables
- sys.index_columns
- sys.trigger_event_types
- sys.indexes
- sys.trigger_events
- sys.key_constraints
- sys.triggers
- sys.numbered_procedure_parameters
- sys.views
- sys.numbered_procedures
- sys.objects

sys.objects содержит одну строку для каждого определенного пользователем объекта в области схемы, который создан в базе данных. Представление sys.objects не показывает триггеры DDL, так как они не принадлежат области схемы. Все триггеры (как DML, так и DDL) размещены в представлении sys.triggers. Чтобы проверить существование объекта можно воспользоваться предикатом EXISTS и запросом на выборку из представления sys.objects. Аналогичный результат можно получить, если воспользоваться системной функцией OBJECT_ID. Возвращает идентификационный номер объекта базы данных для объекта области схемы. Следующий пример проверяет существование указанной таблицы, проверяя наличие у таблицы идентификатора объекта. Если таблица существует, то она удаляется. Если таблица не существует, то инструкция DROP TABLE не выполняется.

```
USE Northwind;
GO
IF OBJECT_ID (N'dbo.Products', N'U') IS NOT NULL
    DROP TABLE dbo.Products;
GO
```

Замечание. В справочнике приводятся сведения о скалярных функциях, которые возвращают информацию о базе данных и объектах баз данных. Пример:

```
-- Как найти все ограничения определенной таблицы?
SELECT OBJECT_NAME(object_id) AS [constraint_name]
      , SCHEMA_NAME(schema_id) AS [schema_name]
      , OBJECT_NAME(parent_object_id) AS [table_name]
      , type_desc
FROM sys.objects
WHERE type_desc LIKE '%CONSTRAINT' AND parent_object_id = OBJECT_ID('dbo.Products');
GO
```

Замечание. В электронной документации по SQL Server приводится список часто задаваемых вопросов (33 шт.) и ответы на эти вопросы — запросы, основанные на представлениях каталога.

В SQL Server для получения информации о состоянии сервера применяются **динамические административные представления и функции**, расположенные в схеме sys. Их имена следуют такому соглашению по именованию: dm_* (от англ. dynamic_management). Есть два типа динамических административных представлений и функций:

- области сервера** - для них необходимо разрешение VIEW SERVER STATE на сервере;
- области базы данных** - для них необходимо разрешение VIEW DATABASE STATE на базе данных.

Динамические административные представления и функции организованы в виде 20 категорий:

- Динамические административные представления управления и функции для групп доступности AlwaysOn
- Динамические административные представления и функции, связанные с индексами
- Динамические административные представления, относящиеся к системе отслеживания измененных данных
- Динамические административные представления и функции, связанные с вводом-выводом
- Динамические административные представления, относящиеся к отслеживанию изменений
- Динамические административные представления и соответствующие функции, связанные с объектами
- Динамические административные представления, связанные со средой CLR
- Динамические административные представления, связанные с уведомлениями запроса
- Динамические административные представления, связанные с зеркальным отображением базы данных
- Динамические административные представления, относящиеся к репликации
- Динамические административные представления базы данных
- Динамические административные представления для регулятора ресурсов
- Динамические административные представления и соответствующие функции, связанные с выполнением
- Динамические административные представления, связанные с безопасностью
- Динамические административные представления расширенных событий
- Динамические административные представления компонента Service Broker
- Динамические административные представления Filestream and FileTable (Transact-SQL)
- Динамические административные представления, относящиеся к операционной системе SQL Server
- Динамические административные представления полнотекстового и семантического поиска
- Динамические административные представления и функции, связанные с транзакциями

Например, категория «Динамические административные представления, относящиеся к операционной системе SQL Server» содержит 31 представление и функции. Работа с этими представлениями и функциями предполагает соответствующий уровень профессиональной квалификации. Только два примера.

`sys.dm_os_wait_stats` - возвращает данные обо всех случаях ожидания, обнаруженных выполняющимися потоками. Это представление можно использовать для диагностики проблем производительности как всего SQL Server, так и конкретных запросов и пакетов. Ожидание ресурсов имеет место, когда исполнитель запрашивает доступ к ресурсу, который недоступен по причине его использования другим исполнителем или пока он еще недоступен. Несмотря на небольшое количество атрибутов представления (5), интерпретация полученных результатов может вызвать определенные затруднения, т. к. атрибут `wait_type` (имя типа ожидания) может принимать ориентировочно 370 различных значений. Если тип ожидания `WAITFOR` имеет очевидный смысл, т. к. является результатом выполнения инструкции `WAITFOR T-SQL`, в которой длительность ожидания определяется параметрами инструкции. То тип ожидания `PREEMPTIVE_CLOSEBACKUPVDIDEVICE` возникает, когда планировщик `SQLOS` переключается в режим с вытеснением, чтобы закрыть виртуальное устройство резервного копирования. (Реплика. Звучит красиво, но не совсем понятно).

`sys.dm_os_threads` - возвращает список всех потоков SQL Server в операционной системе, запущенных процессом SQL Server. Содержит 26 атрибутов.

Для получения данных, хранящихся в системных таблицах, в приложениях можно использовать один из следующих компонентов:

- системные хранимые процедуры;
- инструкции и функции языка Transact-SQL;
- Управляющие объекты SQL Server (SMO)
- объекты RMO;
- функции каталога Database API

Системные хранимые процедуры

Системные хранимые процедуры объединяются в категории (20 категорий)

ЛР: `sp_bindefault`, `sp_bindrule`, [sp_configure](#).

Для Ш.: [sp_lock](#), [sp_who](#), [sp_who2](#)

Функции

Функции метаданных (41 штука)

Возвращают информацию о базе данных и объектах баз данных, например, `DB_ID` и `OBJECT_ID`.

Для Ш.: [APPLOCK_MODE](#) и [APPLOCK_TEST](#).

Функции конфигурации (15 штук)

Возвращают сведения о текущих значениях параметров конфигурации, например, [@@NESTLEVEL](#).

Для Ш.: [@@LOCK_TIMEOUT](#).

Функции безопасности (26 штук)

Возвращают сведения, полезные для управления безопасностью, например, PERMISSIONS. Но есть еще Хранимые процедуры безопасности (50 штук).

Системные функции (29 штук)

Возвращают сведения об объектах и настройках SQL Server, например, [@@ERROR](#) и [@@IDENTITY](#).

Для Ш.: [@@TRANCOUNT](#) и [XACT_STATE](#).

DBCC команда (34 штуки)

Для Ш.: DBCC ROWLOCK

AF = агрегатная функция (среда CLR)

C = ограничение CHECK

D = значение по умолчанию (DEFAULT), в ограничении или независимо заданное

F = ограничение FOREIGN KEY

FN = скалярная функция SQL

FS = скалярная функция сборки (среда CLR)

FT = возвращающая табличное значение функция сборки (среда CLR)

IF = встроенная возвращающая табличное значение функция SQL

IT = внутренняя таблица

P = хранимая процедура SQL

PC = хранимая процедура сборки (среда CLR)

PG = структура плана

PK = ограничение PRIMARY KEY

R = правило (старый стиль, изолированный)

RF = процедура фильтра репликации

S = системная базовая таблица

SN = синоним

SQ = очередь обслуживания

TA = триггер DML сборки (среда CLR)

TF = возвращающая табличное значение функция SQL

TR = триггер DML SQL

TT = табличный тип

U = таблица (пользовательская)

UQ = ограничение UNIQUE

V = представление

X = расширенная хранимая процедура

Следующие скалярные функции возвращают информацию о базе данных и объектах баз данных:

@@PROCID	fn_listextendedproperty
ASSEMBLYPROPERTY	FULLTEXTCATALOGPROPERTY
COL_LENGTH	FULLTEXTSERVICEPROPERTY
COL_NAME	INDEX_COL
COLUMNPROPERTY	INDEXKEY_PROPERTY
DATABASEPROPERTY	INDEXPROPERTY
DATABASEPROPERTYEX	OBJECT_ID
DB_ID	OBJECT_NAME
DB_NAME	OBJECTPROPERTY
FILE_ID	OBJECTPROPERTYEX
FILE_IDEX	SCHEMA_ID
FILE_NAME	SCHEMA_NAME
FILEGROUP_ID	SQL_VARIANT_PROPERTY
FILEGROUP_NAME	TYPE_ID
FILEGROUPPROPERTY	TYPE_NAME
FILEPROPERTY	TYPEPROPERTY

Хранимые процедуры каталога

SQL Server поддерживает следующие системные хранимые процедуры, реализующие функции словаря данных ODBC и изоляцию приложений ODBC от изменений базовых системных таблиц.

- sp_column_privileges
- sp_special_columns
- sp_columns
- sp_sproc_columns
- sp_databases
- sp_statistics
- sp_fkeys
- sp_stored_procedures
- sp_pkeys
- sp_table_privileges
- sp_server_info
- sp_tables