

Table of Contents

Overview

[Querying the SQL Server System Catalog FAQ](#)

[Schemas - sys.schemas](#)

[Messages \(for errors\) - sys.messages](#)

[Extended Properties - sys.extended_properties](#)

Change tracking

[sys.change_tracking_databases](#)

[sys.change_tracking_tables](#)

Database Mirroring

[Database Mirroring Witness - sys.database_mirroring_witnesses](#)

Data-tier Applications

[Data-tier Application Views - dbo.sysdac_instance](#)

Availability groups

[sys.availability_databases_cluster](#)

[sys.availability_group_listener_ip_addresses](#)

[sys.availability_group_listeners](#)

[sys.availability_groups](#)

[sys.availability_groups_cluster](#)

[sys.availability_read_only_routing_lists](#)

[sys.availability_replicas](#)

Azure SQL Database

[sys.bandwidth_usage](#)

[sys.database_connection_stats](#)

[sys.database_firewall_rules](#)

[sys.database_service_objectives](#)

[sys.database_usage](#)

[sys.elastic_pool_resource_stats](#)

[sys.event_log](#)

[sys.firewall_rules](#)

sys.resource_stats

sys.resource_usage

dbo.server_quotas

dbo.slo_assignment_history

dbo.slo_database_objectives

dbo.slo_service_objectives

CLR Assembly

sys.assemblies

sys.assembly_files

sys.assembly_references

Data Collector

syscollector_collection_items

syscollector_collection_sets

syscollector_collector_types

syscollector_config_store

syscollector_execution_log

syscollector_execution_log_full

syscollector_execution_stats

Data Spaces

sys.data_spaces

sys.destination_data_spaces

sys.filegroups

sys.partition_schemes

Database Mail Views

sysmail_allitems

sysmail_event_log

sysmail_faileditems

sysmail_mailattachments

sysmail_sentitems

sysmail_unsentitems

Databases and Files

sys.backup_devices

[sys.databases](#)

[sys.database_files](#)

[sys.database_mirroring](#)

[sys.database_recovery_status](#)

[sys.database_scoped_configurations](#)

[sys.master_files](#)

Endpoints

[sys.database_mirroring_endpoints](#)

[sys.endpoints](#)

[sys.endpoint_webmethods](#)

[sys.http_endpoints](#)

[sys.service_broker_endpoints](#)

[sys.soap_endpoints](#)

[sys.tcp_endpoints](#)

Extended Events

[sys.database_event_session_targets](#) (Azure SQL Database)

[sys.database_event_session_fields](#) (Azure SQL Database)

[sys.database_event_session_events](#) (Azure SQL Database)

[sys.database_event_session_actions](#) (Azure SQL Database)

[sys.database_event_sessions](#) (Azure SQL Database)

[sys.server_event_sessions](#)

[sys.server_event_session_actions](#)

[sys.server_event_session_events](#)

[sys.server_event_session_fields](#)

[sys.server_event_session_targets](#)

External Operations

[sys.external_tables](#)

[sys.external_data_sources](#)

[sys.external_file_formats](#)

Filestream and FileTable

[sys.database_filestream_options](#)

[sys.filetable_system_defined_objects](#)

sys.filetables

Full-Text Search and Semantic Search

sys.fulltext_catalogs

sys.fulltext_document_types

sys.fulltext_index_catalog_usages

sys.fulltext_index_columns

sys.fulltext_index_fragments

sys.fulltext_indexes

sys.fulltext_languages

sys.fulltext_semantic_language_statistics_database

sys.fulltext_semantic_languages

sys.fulltext_stoplists

sys.fulltext_stopwords

sys.fulltext_system_stopwords

sys.registered_search_properties

sys.registered_search_property_lists

Linked Servers

sys.linked_logins

sys.remote_logins

sys.servers

Object

sys.all_columns

sys.all_objects

sys.all_parameters

sys.all_sql_modules

sys.all_views

sys.allocation_units

sys.assembly_modules

sys.check_constraints

sys.column_store_dictionaries

sys.column_store_row_groups

sys.column_store_segments

sys.columns
sys.computed_columns
sys.default_constraints
sys.events
sys.event_notifications
sys.event_notification_event_types
sys.extended_procedures
sys.external_libraries
sys.external_library_files
sys.foreign_keys
sys.foreign_key_columns
sys.function_order_columns
sys.hash_indexes
sys.identity_columns
sys.indexes
sys.index_columns
sys.index_resumable_operations
sys.internal_partitions
sys.internal_tables
sys.key_constraints
sys.masked_columns
sys.memory_optimized_tables_internal_attributes
sys.module_assembly_usages
sys.numbered_procedures
sys.numbered_procedure_parameters
sys.objects
sys.parameters
sys.partitions
sys.periods
sys.plan_guides
sys.procedures
sys.sequences

sys.server_assembly_modules
sys.server_events
sys.server_event_notifications
sys.server_sql_modules
sys.server_triggers
sys.server_trigger_events
sys.sql_dependencies
sys.sql_expression_dependencies
sys.sql_modules
sys.stats
sys.stats_columns
sys.synonyms
sys.system_columns
sys.system_objects
sys.system_parameters
sys.system_sql_modules
sys.system_views
sys.table_types
sys.tables
sys.trigger_event_types
sys.trigger_events
sys.triggers
sys.views

Partition Function

sys.partition_functions
sys.partition_parameters
sys.partition_range_values

Policy-Based Management Views

syspolicy_conditions
syspolicy_policies
syspolicy_policy_execution_history
syspolicy_policy_execution_history_details

syspolicy_policy_categories
syspolicy_policy_category_subscriptions
syspolicy_system_health_state

Resource Governor

sys.resource_governor_configuration
sys.resource_governor_external_resource_pools
sys.resource_governor_resource_pools
sys.resource_governor_workload_groups

Query Store

sys.database_query_store_options
sys.query_context_settings
sys.query_store_plan
sys.query_store_query
sys.query_store_query_text
sys.query_store_runtime_stats
sys.query_store_wait_stats
sys.query_store_runtime_stats_interval

Scalar Types

sys.assembly_types
sys.types
sys.type_assembly_usages
sys.column_type_usages
sys.parameter_type_usages

Security

sys.asymmetric_keys
sys.certificates
sys.column_encryption_keys
sys.column_encryption_key_values
sys.column_master_keys
sys.credentials
sys.crypt_properties
sys.cryptographic_providers

sys.database_audit_specifications
sys.database_audit_specification_details
sys.database_credentials
sys.database_scoped_credentials
sys.database_permissions
sys.database_principals
sys.database_role_members
sys.key_encryptions
sys.login_token
sys.master_key_passwords
sys.openkeys
sys.securable_classes
sys.security_policies
sys.security_predicates
sys.server_audits
sys.server_audit_specifications
sys.server_audit_specification_details
sys.server_file_audits
sys.server_permissions
sys.server_principals
sys.server_role_members
sys.sql_logins
sys.symmetric_keys
sys.system_components_surface_area_configuration
sys.user_token

Service Broker

sys.conversation_endpoints
sys.conversation_groups
sys.conversation_priorities
sys.message_type_xml_schema_collection_usages
sys.remote_service_bindings
sys.routes

[sys.service_contract_message_usages](#)

[sys.service_contract_usages](#)

[sys.service_contracts](#)

[sys.service_message_types](#)

[sys.service_queue_usages](#)

[sys.service_queues](#)

[sys.services](#)

[sys.transmission_queue](#)

Server-wide Configuration

[sys.configurations](#)

[sys.time_zone_info](#)

[sys.traces](#)

[sys.trace_categories](#)

[sys.trace_columns](#)

[sys.trace_events](#)

[sys.trace_event_bindings](#)

[sys.trace_subclass_values](#)

Spatial Data

[sys.spatial_index_tessellations](#)

[sys.spatial_indexes](#)

[sys.spatial_reference_systems](#)

SQL Data Warehouse

[sys.pdw_column_distribution_properties](#)

[sys.pdw_database_mappings](#)

[sys.pdw_diag_event_properties](#)

[sys.pdw_diag_events](#)

[sys.pdw_diag_sessions](#)

[sys.pdw_distributions](#)

[sys.pdw_health_alerts](#)

[sys.pdw_health_component_groups](#)

[sys.pdw_health_component_properties](#)

[sys.pdw_health_component_status_mappings](#)

sys.pdw_health_components
sys.pdw_index_mappings
sys.pdw_loader_backup_run_details
sys.pdw_loader_backup_runs
sys.pdw_loader_run_stages
sys.pdw_nodes_column_store_dictionaries
sys.pdw_nodes_column_store_row_groups
sys.pdw_nodes_column_store_segments
sys.pdw_nodes_columns
sys.pdw_nodes_indexes
sys.pdw_nodes_partitions
sys.pdw_nodes_pdw_physical_databases
sys.pdw_nodes_tables
sys.pdw_replicated_table_cache_state
sys.pdw_table_distribution_properties
sys.pdw_table_mappings

Stretch Database

sys.remote_data_archive_databases
sys.remote_data_archive_tables

XML Schemas (XML Type System)

sys.column_xml_schema_collection_usages
sys.parameter_xml_schema_collection_usages
sys.selective_xml_index_paths
sys.xml_schema_attributes
sys.xml_schema_component_placements
sys.xml_schema_components
sys.xml_schema_elements
sys.xml_schema_facets
sys.xml_indexes
sys.xml_schema_model_groups
sys.xml_schema_collections
sys.xml_schema_namespaces




`sys.xml_schema_types`

`sys.xml_schema_wildcard_namespaces`

`sys.xml_schema_wildcards`

System Catalog Views (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Catalog views return information that is used by the SQL Server Database Engine. We recommend that you use catalog views because they are the most general interface to the catalog metadata and provide the most efficient way to obtain, transform, and present customized forms of this information. All user-available catalog metadata is exposed through catalog views.

NOTE

Catalog views do not contain information about replication, backup, database maintenance plan, or SQL Server Agent catalog data.

Some catalog views inherit rows from other catalog views. For example, the [sys.tables](#) catalog view inherits from the [sys.objects](#) catalog view. The [sys.objects](#) catalog view is referred to as the base view, and the [sys.tables](#) view is called the derived view. The [sys.tables](#) catalog view returns the columns that are specific to tables and also all the columns that the [sys.objects](#) catalog view returns. The [sys.objects](#) catalog view returns rows for objects other than tables, such as stored procedures and views. After a table is created, the metadata for the table is returned in both views. Although the two catalog views return different levels of information about the table, there is only one entry in metadata for this table with one name and one `object_id`. This can be summarized as follows:

- The base view contains a subset of columns and a superset of rows.
- The derived view contains a superset of columns and a subset of rows.

IMPORTANT

In future releases of SQL Server, Microsoft may augment the definition of any system catalog view by adding columns to the end of the column list. We recommend against using the syntax `SELECT * FROM sys.catalog_view_name` in production code because the number of columns returned might change and break your application.

The catalog views in SQL Server have been organized into the following categories:

Always On Availability Groups Catalog Views (Transact-SQL)	Messages (for Errors) Catalog Views (Transact-SQL)
Azure SQL Database Catalog Views	Object Catalog Views (Transact-SQL)
Change Tracking Catalog Views (Transact-SQL)	Partition Function Catalog Views (Transact-SQL)
CLR Assembly Catalog Views (Transact-SQL)	Policy-Based Management Views (Transact-SQL)

Data Collector Views (Transact-SQL)	Resource Governor Catalog Views (Transact-SQL)
Data Spaces (Transact-SQL)	Query Store Catalog Views (Transact-SQL)
Database Mail Views (Transact-SQL)	Scalar Types Catalog Views (Transact-SQL)
Database Mirroring Witness Catalog Views (Transact-SQL)	Schemas Catalog Views (Transact-SQL)
Databases and Files Catalog Views (Transact-SQL)	Security Catalog Views (Transact-SQL)
Endpoints Catalog Views (Transact-SQL)	Service Broker Catalog Views (Transact-SQL)
Extended Events Catalog Views (Transact-SQL)	Server-wide Configuration Catalog Views (Transact-SQL)
Extended Properties Catalog Views (Transact-SQL)	Spatial Data Catalog Views
External Operations Catalog Views (Transact-SQL)	SQL Data Warehouse and Parallel Data Warehouse Catalog Views
Filestream and FileTable Catalog Views (Transact-SQL)	Stretch Database Catalog Views (Transact-SQL)
Full-Text Search and Semantic Search Catalog Views (Transact-SQL)	XML Schemas (XML Type System) Catalog Views (Transact-SQL)
Linked Servers Catalog Views (Transact-SQL)	

See Also





[Information Schema Views \(Transact-SQL\)](#)

[System Tables \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

Querying the SQL Server System Catalog FAQ

11/21/2017 • 17 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This topic contains a list of frequently asked questions. The answers to these questions are queries that are based on catalog views.

Frequently Asked Questions

The sections below list frequently asked questions by category.

Data Types

- [How do I find the data types of the columns of a specified table?](#)
- [How do I find the LOB data types of a specified table?](#)
- [How do I find the columns that depend on a specified data type?](#)
- [How do I find the computed columns that depend on a specified CLR user-defined type or alias data type?](#)
- [How do I find the parameters that depend on a specified CLR user-defined type or alias type?](#)
- [How do I find the CHECK constraints that depend on a specified CLR user-defined type?](#)
- [How do I find the views, Transact-SQL functions, and Transact-SQL stored procedures that depend on a specified CLR user-defined type or alias type?](#)

Tables, Indexes, Views, and Constraints

- [How do I find all the user-defined tables in a specified database?](#)
- [How do I find all the tables that do not have a clustered index in a specified database?](#)
- [How do I find all the tables that do not have an index?](#)
- [How do I find all the tables that do not have a primary key?](#)
- [How do I find all the tables that have an identity column?](#)
- [How do I find all the tables and indexes that are partitioned?](#)
- [How do I find all views in a database?](#)
- [How do I find the definition of a view?](#)
- [How do I find all the entities that have been modified in the last N days?](#)
- [How do I find the columns of a primary key for a specified table?](#)
- [How do I find the columns of a foreign key for a specified table?](#)
- [How do I determine if a column is used in a computed column expression?](#)
- [How do I find all the columns that are used in a computed column expression?](#)
- [How do I find all the constraints for a specified table?](#)

- [How do I find all the indexes for a specified table?](#)
- [How do I find all the tables that have a specified column name?](#)
- [How do I find all the statistics on a specified object?](#)
- [How do I find all the statistics and statistics columns on a specified object?](#)

Modules (Stored Procedures, User-Defined Functions, and Triggers)

- [How do I find all the stored procedures in a database?](#)
- [How do I find all the user-defined functions in a database?](#)
- [How do I find the parameters for a specified stored procedure or function?](#)
- [How do I find the dependencies on a specified function?](#)
- [How do I view the definition of a module?](#)
- [How do I view the definition of a server-level trigger?](#)

Schemas, Users, Roles, and Permissions

- [How do I find all the owners of entities contained in a specified schema?](#)
- [How do I find the permissions granted or denied to a specified principal?](#)

Answers

How do I find all the tables that do not have a clustered index in a specified database?

Before you run the following queries, replace `<database_name>` with a valid database name.

```
SELECT SCHEMA_NAME(t.schema_id) AS schema_name, t.name AS table_name
FROM sys.tables AS t
WHERE NOT EXISTS
(
    SELECT * FROM sys.indexes AS i
    WHERE i.object_id = t.object_id
    AND i.type = 1 -- or type_desc = 'CLUSTERED'
)
ORDER BY schema_name, table_name;
GO
```

Or, you can use the `OBJECTPROPERTY` function as shown in the following example.

```
USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name, name AS table_name
FROM sys.tables
WHERE OBJECTPROPERTY(object_id, 'TableHasClustIndex') = 0
ORDER BY schema_id, name;
GO
```

[TOP](#)

How do I find all the owners of entities contained in a specified schema?

Before you run the following query, replace `<database_name>` and `<schema_name>` with valid names.

```

USE <database_name>;
GO
SELECT 'OBJECT' AS entity_type
      ,USER_NAME(OBJECTPROPERTY(object_id, 'OwnerId')) AS owner_name
      ,name
FROM sys.objects WHERE SCHEMA_NAME(schema_id) = '<schema_name>'
UNION
SELECT 'TYPE' AS entity_type
      ,USER_NAME(TYPEPROPERTY(SCHEMA_NAME(schema_id) + '.' + name, 'OwnerId')) AS owner_name
      ,name
FROM sys.types WHERE SCHEMA_NAME(schema_id) = '<schema_name>'
UNION
SELECT 'XML SCHEMA COLLECTION' AS entity_type
      ,COALESCE(USER_NAME(xsc.principal_id),USER_NAME(s.principal_id)) AS owner_name
      ,xsc.name
FROM sys.xml_schema_collections AS xsc JOIN sys.schemas AS s
      ON s.schema_id = xsc.schema_id
WHERE s.name = '<schema_name>';
GO

```

[TOP](#)

How do I find all the tables that do not have a primary key?

Before you run the following queries, replace `<database_name>` with a valid database name.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(t.schema_id) AS schema_name
      ,t.name AS table_name
FROM sys.tables t
WHERE object_id NOT IN
      (
        SELECT parent_object_id
        FROM sys.key_constraints
        WHERE type_desc = 'PRIMARY_KEY_CONSTRAINT' -- or type = 'PK'
      );
GO

```

Or, you can run the following query.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
      ,name AS table_name
FROM sys.tables
WHERE OBJECTPROPERTY(object_id, 'TableHasPrimaryKey') = 0
ORDER BY schema_name, table_name;
GO

```

[TOP](#)

How do I find all the tables that do not have an index?

Before you run the following query, replace `<database_name>` with a valid database name.


```
USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
       ,name AS table_name
FROM sys.tables
WHERE OBJECTPROPERTY(object_id,'IsIndexed') = 0
ORDER BY schema_name, table_name;
GO
```

[TOP](#)

How do I find all the tables that have an identity column?

Before you run the following query, replace `<database_name>` with a valid database name.

```
USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
       , t.name AS table_name
       , c.name AS column_name
FROM sys.tables AS t
JOIN sys.identity_columns c ON t.object_id = c.object_id
ORDER BY schema_name, table_name;
GO
```

Or, you can run the following query.

NOTE

This query does not return the name of the columns.

```
USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
       ,name AS table_name
FROM sys.tables
WHERE OBJECTPROPERTY(object_id,'TableHasIdentity') = 1
ORDER BY schema_name, table_name;
GO
```

[TOP](#)

How do I find the data types of the columns of a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```

USE <database_name>;
GO
SELECT c.name AS column_name
      ,c.column_id
      ,SCHEMA_NAME(t.schema_id) AS type_schema
      ,t.name AS type_name
      ,t.is_user_defined
      ,t.is_assembly_type
      ,c.max_length
      ,c.precision
      ,c.scale
FROM sys.columns AS c
JOIN sys.types AS t ON c.user_type_id=t.user_type_id
WHERE c.object_id = OBJECT_ID('<schema_name.table_name>')
ORDER BY c.column_id;
GO

```

[TOP](#)

How do I find the dependencies on a specified function?

Before you run the following query, replace `<database_name>` and `<schema_name.function_name>` with valid names.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS referencing_object_name
      ,COALESCE(COL_NAME(object_id, column_id), '(n/a)') AS referencing_column_name
      ,*
FROM sys.sql_dependencies
WHERE referenced_major_id = OBJECT_ID('<schema_name.function_name>')
ORDER BY OBJECT_NAME(object_id), COL_NAME(object_id, column_id);
GO

```

[TOP](#)

How do I find all the stored procedures in a database?

Before you run the following query, replace `<database_name>` with a valid name.

```

USE <database_name>;
GO
SELECT name AS procedure_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.procedures;
GO

```

[TOP](#)

How do I find the parameters for a specified stored procedure or function?

Before you run the following query, replace `<database_name>` and `<schema_name.object_name>` with valid names.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
      ,o.name AS object_name
      ,o.type_desc
      ,p.parameter_id
      ,p.name AS parameter_name
      ,TYPE_NAME(p.user_type_id) AS parameter_type
      ,p.max_length
      ,p.precision
      ,p.scale
      ,p.is_output
FROM sys.objects AS o
INNER JOIN sys.parameters AS p ON o.object_id = p.object_id
WHERE o.object_id = OBJECT_ID('<schema_name.object_name>')
ORDER BY schema_name, object_name, p.parameter_id;
GO

```

[TOP](#)

How do I find all the user-defined functions in a database?

Before you run the following query, replace `<database_name>` with a valid database name.

```

USE <database_name>;
GO
SELECT name AS function_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.objects
WHERE type_desc LIKE '%FUNCTION%';
GO

```

[TOP](#)

How do I find all views in a database?

Before you run the following query, replace `<database_name>` with a valid database name.

```

USE <database_name>;
GO
SELECT name AS view_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,OBJECTPROPERTYEX(object_id,'IsIndexed') AS IsIndexed
      ,OBJECTPROPERTYEX(object_id,'IsIndexable') AS IsIndexable
      ,create_date
      ,modify_date
FROM sys.views;

```

[TOP](#)

How do I find all the entities that have been modified in the last N days?

Before you run the following query, replace `<database_name>` and `<n_days>` with valid values.

```

USE <database_name>;
GO
SELECT name AS object_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.objects
WHERE modify_date > GETDATE() - <n_days>
ORDER BY modify_date;
GO

```

[TOP](#)

How do I find the LOB data types of a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```

USE <database_name>;
GO
SELECT name AS column_name
      ,column_id
      ,TYPE_NAME(user_type_id) AS type_name
      ,max_length
      ,CASE
          WHEN max_length = -1 AND TYPE_NAME(user_type_id) <> 'xml'
          THEN 1
          ELSE 0
      END AS [(max)]
FROM sys.columns
WHERE object_id=OBJECT_ID('<schema_name.table_name>')
      AND ( TYPE_NAME(user_type_id) IN ('xml','text', 'ntext','image')
          OR (TYPE_NAME(user_type_id) IN ('varchar','nvarchar','varbinary')
              AND max_length = -1)
      );
GO

```

[TOP](#)

How do I view the definition of a module?

Before you run the following query, replace `<database_name>` and `<schema_name.object_name>` with valid names.

```

USE <database_name>;
GO
SELECT definition
FROM sys.sql_modules
WHERE object_id = OBJECT_ID('<schema_name.object_name>');
GO

```

Or, you can use the `OBJECT_DEFINITION` function as shown in the following example.

```

USE <database_name>;
GO
SELECT OBJECT_DEFINITION (OBJECT_ID('<schema_name.object_name>')) AS ObjectDefinition;
GO

```

[TOP](#)

How do I view the definition of a server-level trigger?

```
SELECT definition
FROM sys.server_sql_modules;
GO
```

[TOP](#)

How do I find the columns of a primary key for a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```
USE <database_name>;
GO
SELECT i.name AS index_name
      ,ic.index_column_id
      ,key_ordinal
      ,c.name AS column_name
      ,TYPE_NAME(c.user_type_id)AS column_type
      ,is_identity
FROM sys.indexes AS i
INNER JOIN sys.index_columns AS ic
      ON i.object_id = ic.object_id AND i.index_id = ic.index_id
INNER JOIN sys.columns AS c
      ON ic.object_id = c.object_id AND c.column_id = ic.column_id
WHERE i.is_primary_key = 1
      AND i.object_id = OBJECT_ID('<schema_name.table_name>');
GO
```

Or, you can use the `COL_NAME` function as shown in the following example.

```
USE <database_name>;
GO
SELECT i.name AS index_name
      ,COL_NAME(ic.object_id,ic.column_id) AS column_name
      ,ic.index_column_id
      ,key_ordinal
FROM sys.indexes AS i
INNER JOIN sys.index_columns AS ic
      ON i.object_id = ic.object_id AND i.index_id = ic.index_id
WHERE i.is_primary_key = 1
      AND i.object_id = OBJECT_ID('<schema_name.table_name>');
GO
```

[TOP](#)

How do I find the columns of a foreign key for a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```

USE <database_name>;
GO
SELECT
    f.name AS foreign_key_name
    ,OBJECT_NAME(f.parent_object_id) AS table_name
    ,COL_NAME(fc.parent_object_id, fc.parent_column_id) AS constraint_column_name
    ,OBJECT_NAME (f.referenced_object_id) AS referenced_object
    ,COL_NAME(fc.referenced_object_id, fc.referenced_column_id) AS referenced_column_name
    ,is_disabled
    ,delete_referential_action_desc
    ,update_referential_action_desc
FROM sys.foreign_keys AS f
INNER JOIN sys.foreign_key_columns AS fc
    ON f.object_id = fc.constraint_object_id
WHERE f.parent_object_id = OBJECT_ID('<schema_name.table_name>');

```

[TOP](#)

How do I find the permissions granted or denied to a specified principal?

The following example creates a function to return the name of the entity on which the permissions are checked. The function is invoked in the queries that follow. The function must be created in every database in which you want to check permissions.

```

-- Create a function to return the name of the entity on which the permissions are checked.
IF OBJECT_ID (N'dbo.entity_instance_name', N'FN') IS NOT NULL
    DROP FUNCTION dbo.entity_instance_name;
GO
CREATE FUNCTION dbo.entity_instance_name(@class_desc nvarchar(60), @major_id int)
RETURNS sysname AS
BEGIN
    DECLARE @the_entity_name sysname
    SELECT @the_entity_name = CASE
        WHEN @class_desc = 'DATABASE' THEN DB_NAME()
        WHEN @class_desc = 'SCHEMA' THEN SCHEMA_NAME(@major_id)
        WHEN @class_desc = 'OBJECT_OR_COLUMN' THEN OBJECT_NAME(@major_id)
        WHEN @class_desc = 'DATABASE_PRINCIPAL' THEN USER_NAME(@major_id)
        WHEN @class_desc = 'ASSEMBLY' THEN
            (SELECT name FROM sys.assemblies WHERE assembly_id=@major_id)
        WHEN @class_desc = 'TYPE' THEN TYPE_NAME(@major_id)
        WHEN @class_desc = 'XML_SCHEMA_COLLECTION' THEN
            (SELECT name FROM sys.xml_schema_collections
             WHERE xml_collection_id=@major_id)
        WHEN @class_desc = 'MESSAGE_TYPE' THEN
            (SELECT name FROM sys.service_message_types WHERE message_type_id=@major_id)
        WHEN @class_desc = 'SERVICE_CONTRACT' THEN
            (SELECT name FROM sys.service_contracts
             WHERE service_contract_id=@major_id)
        WHEN @class_desc = 'SERVICE' THEN
            (SELECT name FROM sys.services WHERE service_id=@major_id)
        WHEN @class_desc = 'REMOTE_SERVICE_BINDING' THEN
            (SELECT name FROM sys.remote_service_bindings
             WHERE remote_service_binding_id=@major_id)
        WHEN @class_desc = 'ROUTE' THEN
            (SELECT name FROM sys.routes WHERE route_id=@major_id)
        WHEN @class_desc = 'FULLTEXT_CATALOG' THEN
            (SELECT name FROM sys.fulltext_catalogs WHERE fulltext_catalog_id=@major_id)
        WHEN @class_desc = 'SYMMETRIC_KEY' THEN
            (SELECT name FROM sys.symmetric_keys WHERE symmetric_key_id=@major_id)
        WHEN @class_desc = 'CERTIFICATE' THEN
            (SELECT name FROM sys.certificates WHERE certificate_id=@major_id)
        WHEN @class_desc = 'ASYMMETRIC_KEY' THEN
            (SELECT name FROM sys.asymmetric_keys WHERE asymmetric_key_id=@major_id)
        WHEN @class_desc = 'SERVER' THEN
            (SELECT name FROM sys.servers WHERE server_id=@major_id)
        WHEN @class_desc = 'SERVER PRINCIPAL' THEN SUSER NAME(@major_id)
    END
END

```

```

        WHEN @class_desc = 'ENDPOINT' THEN
            (SELECT name FROM sys.endpoints WHERE endpoint_id=@major_id)
        ELSE '?'
    END
    RETURN @the_entity_name
END;
GO
-- Return server-level permissions for the user.
SELECT class
    ,class_desc
    ,dbo.entity_instance_name(class_desc, major_id) AS entity_name
    ,minor_id
    ,SUSER_NAME(grantee_principal_id) AS grantee
    ,SUSER_NAME(grantor_principal_id) AS grantor
    ,type
    ,permission_name
    ,state_desc
FROM sys.server_permissions
WHERE grantee_principal_id = SUSER_ID('public');
GO
-- Return database-level permissions for the user.
SELECT class
    ,class_desc
    ,dbo.entity_instance_name(class_desc , major_id) AS entity_name
    ,minor_id
    ,USER_NAME(grantee_principal_id) AS grantee
    ,USER_NAME(grantor_principal_id) AS grantor
    ,type
    ,permission_name
    ,state_desc
FROM sys.database_permissions
WHERE grantee_principal_id = DATABASE_PRINCIPAL_ID('public');
GO

```

[TOP](#)

How do I determine if a column is used in a computed column expression?

Before you run the following query, replace `<database_name>`, `<schema_name.table_name>`, and `<column_name>` with valid names.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
    ,COL_NAME(object_id, column_id) AS computed_column
    ,class_desc
    ,is_selected
    ,is_updated
    ,is_select_all
FROM sys.sql_dependencies
WHERE referenced_major_id = OBJECT_ID('<schema_name.table_name>')
    AND referenced_minor_id = COLUMNPROPERTY(referenced_major_id, '<column_name>', 'ColumnId')
    AND class = 1;
GO

```

[TOP](#)

How do I find all the columns that are used in a computed column expression?

Before you run the following query, replace `<database_name>` with a valid name.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(d.referenced_major_id) AS object_name
      ,COL_NAME(d.referenced_major_id, d.referenced_minor_id) AS column_name
      ,OBJECT_NAME(referenced_major_id) AS dependent_object_name
      ,COL_NAME(d.object_id, d.column_id) AS dependent_computed_column
      ,cc.definition AS computed_column_definition
FROM sys.sql_dependencies AS d
JOIN sys.computed_columns AS cc
      ON cc.object_id = d.object_id AND cc.column_id = d.column_id AND d.object_id=d.referenced_major_id
WHERE d.class = 1
ORDER BY object_name, column_name;
GO

```

[TOP](#)

How do I find the columns that depend on a specified CLR user-defined type or alias type?

Before you run the following query, replace `<database_name>` with a valid name and

`<schema_name.data_type_name>` with a valid, schema-qualified CLR user-defined type, or schema-qualified alias type name. The following query requires membership in the **db_owner** role or permissions to see all dependent column and computed column metadata in the database.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
      ,c.name AS column_name
      ,SCHEMA_NAME(t.schema_id) AS schema_name
      ,TYPE_NAME(c.user_type_id) AS user_type_name
      ,c.max_length
      ,c.precision
      ,c.scale
      ,c.is_nullable
      ,c.is_computed
FROM sys.columns AS c
INNER JOIN sys.types AS t ON c.user_type_id = t.user_type_id
WHERE c.user_type_id = TYPE_ID('<schema_name.data_type_name>');
GO

```

The following query returns a restricted and narrow view of columns dependent on a CLR user-defined type or alias, but the result set is visible to the **public** role. You can use this query if you have granted REFERENCE permissions on your user-defined type to others and you do not have permission to view the metadata of the objects others have created that use the type.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
      ,COL_NAME(object_id, column_id) AS column_name
      ,TYPE_NAME(user_type_id) AS user_type
FROM sys.column_type_usages
WHERE user_type_id = TYPE_ID('<schema_name.data_type_name>');
GO

```

[TOP](#)

How do I find the computed columns that depend on a specified CLR user-defined type or alias type?

Before you run the following query, replace `<database_name>` with a valid name and

`<schema_name.data_type_name>` with a valid, schema-qualified CLR user-defined type, alias type name.


```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
      ,COL_NAME(object_id, column_id) AS column_name
FROM sys.sql_dependencies
WHERE referenced_major_id = TYPE_ID('<schema_name.data_type_name>')
      AND class = 2 -- schema-bound references to type
      AND OBJECTPROPERTY(object_id, 'IsTable') = 1; -- exclude non-table dependencies

```

[TOP](#)

How do I find the parameters that depend on a specified CLR user-defined type or alias type?

Before you run the following query, replace `<database_name>` with a valid name and

`<schema_name.data_type_name>` with a valid, schema-qualified CLR user-defined type, alias type name. The following query requires membership in the **db_owner** role or permissions to see all dependent column and computed column metadata in the database.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
      ,NULL AS procedure_number
      ,name AS param_name
      ,parameter_id AS param_num
      ,TYPE_NAME(p.user_TYPE_ID) AS type_name
FROM sys.parameters AS p
WHERE p.user_TYPE_ID = TYPE_ID('<schema_name.data_type_name>')
UNION
SELECT OBJECT_NAME(object_id) AS object_name
      ,procedure_number
      ,name AS param_name
      ,parameter_id AS param_num
      ,TYPE_NAME(p.user_TYPE_ID) AS type_name
FROM sys.numbered_procedure_parameters AS p
WHERE p.user_TYPE_ID = TYPE_ID('<schema_name.data_type_name>')
ORDER BY object_name, procedure_number, param_num;
GO

```

The following query returns a restricted and narrow view of parameters that depend on a CLR user-defined type or alias, but the result set is visible to the **public** role. You can use this query if you have granted REFERENCE permissions on your user-defined type to others and you do not have permission to view the metadata of the objects others have created that use the type.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id) AS object_name
      ,parameter_id
      ,TYPE_NAME(user_type_id) AS type_name
FROM sys.parameter_type_usages
WHERE user_type_id = TYPE_ID('<schema_name.data_type_name>');
GO

```

[TOP](#)

How do I find the CHECK constraints that depend on a specified CLR user-defined type?

Before you run the following query, replace `<database_name>` with a valid name and

`<schema_name.data_type_name>` with a valid, schema-qualified CLR user-defined type name.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(o.schema_id) AS schema_name
      ,OBJECT_NAME(o.parent_object_id) AS table_name
      ,OBJECT_NAME(o.object_id) AS constraint_name
FROM sys.sql_dependencies AS d
JOIN sys.objects AS o ON o.object_id = d.object_id
WHERE referenced_major_id = TYPE_ID('<schema_name.data_type_name>')
      AND class = 2 -- schema-bound references to type
      AND OBJECTPROPERTY(o.object_id, 'IsCheckCnst') = 1; -- exclude non-CHECK dependencies
GO

```

[TOP](#)

How do I find the views, Transact-SQL functions, and Transact-SQL stored procedures that depend on a specified CLR user-defined type or alias type?

Before you run the following query, replace `<database_name>` with a valid name and

`<schema_name.data_type_name>` with a valid, schema-qualified CLR user-defined type, alias type name.

The parameters defined in a function or procedure are implicitly schema bound. Therefore, parameters that depend on a CLR user-defined type or alias type can be viewed by using the [sys.sql_dependencies](#) catalog view. Procedures and triggers are not schema bound. This means that dependencies between any expression defined in the body of the procedure or trigger and a CLR user-defined type or alias type is not maintained. Schema bound views and schema bound user-defined functions that have expressions that depend on a CLR user-defined type or alias type are maintained in the **sys.sql_dependencies** catalog view. Dependencies between types and CLR functions and CLR procedures are not maintained.

The following query returns all schema-bound dependencies in views, Transact-SQL functions, and Transact-SQL stored procedures for a specified CLR user-defined type or alias type.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(o.schema_id) AS dependent_object_schema
      ,OBJECT_NAME(o.object_id) AS dependent_object_name
      ,o.type_desc AS dependent_object_type
      ,d.class_desc AS kind_of_dependency
      ,TYPE_NAME (d.referenced_major_id) AS type_name
FROM sys.sql_dependencies AS d
JOIN sys.objects AS o
      ON d.object_id = o.object_id
      AND o.type IN ('FN','IF','TF','V','P')
WHERE d.class = 2 -- dependencies on types
      AND d.referenced_major_id = TYPE_ID('<schema_name.data_type_name>')
ORDER BY dependent_object_schema, dependent_object_name;
GO

```

[TOP](#)

How do I find all the constraints for a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```

USE <database_name>;
GO
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
      ,create_date
      ,modify_date
      ,is_ms_shipped
      ,is_published
      ,is_schema_published
FROM sys.objects
WHERE type_desc LIKE '%CONSTRAINT'
      AND parent_object_id = OBJECT_ID('<schema_name.table_name>');
GO

```

[TOP](#)

How do I find all the indexes for a specified table?

Before you run the following query, replace `<database_name>` and `<schema_name.table_name>` with valid names.

```

USE <database_name>;
GO
SELECT i.name AS index_name
      ,i.type_desc
      ,is_unique
      ,ds.type_desc AS filegroup_or_partition_scheme
      ,ds.name AS filegroup_or_partition_scheme_name
      ,ignore_dup_key
      ,is_primary_key
      ,is_unique_constraint
      ,fill_factor
      ,is_padded
      ,is_disabled
      ,allow_row_locks
      ,allow_page_locks
FROM sys.indexes AS i
INNER JOIN sys.data_spaces AS ds ON i.data_space_id = ds.data_space_id
WHERE is_hypothetical = 0 AND i.index_id <> 0
AND i.object_id = OBJECT_ID('<schema_name.table_name>');
GO

```

[TOP](#)

How do I find all the objects that have a specified column name?

Before you run the following query, replace `<database_name>` and `<column_name>` with valid names.

```

USE <database_name>;
GO
SELECT OBJECT_NAME(object_id)
FROM sys.columns
WHERE name = '<column_name>';
GO

```

Or

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(o.schema_id) AS schema_name
       ,o.name AS object_name
       ,type_desc
FROM sys.objects AS o
INNER JOIN sys.columns AS c ON o.object_id = c.object_id
WHERE c.name = '<column_name>';
GO

```

[TOP](#)

How do I find all the user-defined tables in a specified database?

Before you run the following query, replace `<database_name>` with a valid name.

```

USE <database_name>;
GO
SELECT *
FROM sys.tables;
GO

```

[TOP](#)

How do I find all the tables and indexes that are partitioned?

Before you run the following query, replace `<database_name>` with a valid name.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(o.schema_id) AS schema_name
       ,OBJECT_NAME(p.object_id) AS table_name
       ,i.name AS index_name
       ,p.partition_number
       ,rows
FROM sys.partitions AS p
INNER JOIN sys.indexes AS i ON p.object_id = i.object_id AND p.index_id = i.index_id
INNER JOIN sys.partition_schemes ps ON i.data_space_id=ps.data_space_id
INNER JOIN sys.objects AS o ON o.object_id = i.object_id
ORDER BY index_name, partition_number;
GO

```

[TOP](#)

How do I find all the statistics on a specified object?

Before you run the following query, replace `<database_name>` with a valid name and `<schema_name.object_name>` with a valid table, indexed view, or table-valued function name.

```

USE <database_name>;
GO
SELECT name AS statistics_name
       ,stats_id
       ,auto_created
       ,user_created
       ,no_recompute
FROM sys.stats
WHERE object_id = OBJECT_ID('<schema_name.object_name>');
GO

```

[TOP](#)

How do I find all the statistics and statistics columns on a specified object?

Before you run the following query, replace `<database_name>` with a valid name and `<schema_name.object_name>` with a valid table, indexed view, or table-valued function name.

```
USE <database_name>;
GO
SELECT s.name AS statistics_name
      ,c.name AS column_name
      ,sc.stats_column_id
FROM sys.stats AS s
INNER JOIN sys.stats_columns AS sc
      ON s.object_id = sc.object_id AND s.stats_id = sc.stats_id
INNER JOIN sys.columns AS c
      ON sc.object_id = c.object_id AND c.column_id = sc.column_id
WHERE s.object_id = OBJECT_ID('<schema_name.object_name>');
GO
```

[TOP](#)

How do I find the definition of a view?

Before you run the following query, replace `<database_name>` and `<schema_name.object_name>` with valid names.

```
USE <database_name>;
GO
SELECT definition
FROM sys.sql_modules
WHERE object_id = OBJECT_ID('<schema_name.object_name>');
GO
```

Or, you can use the `OBJECT_DEFINITION` function as shown in the following example.

```
USE <database_name>;
GO
SELECT OBJECT_DEFINITION (OBJECT_ID('<schema_name.object_name>')) AS ObjectDefinition;
GO
```

[TOP](#)

See Also

[Mapping System Tables to System Views \(Transact-SQL\)](#)

Schemas Catalog Views - sys.schemas

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each database schema.

NOTE

Database schemas are different from XML schemas, which are used to define the content model of XML documents.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the schema. Is unique within the database.
schema_id	int	ID of the schema. Is unique within the database.
principal_id	int	ID of the principal that owns this schema.

Remarks

Database schemas act as namespaces or containers for objects, such as tables, views, procedures, and functions, that can be found in the **sys.objects** catalog view.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also




[Catalog Views \(Transact-SQL\)](#)

[Schemas Catalog Views \(Transact-SQL\)](#)

[sys.objects \(Transact-SQL\)](#)

Messages (for errors) Catalog Views - sys.messages

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each **message_id** or **language_id** of the error messages in the system, for both system-defined and user-defined messages. For more information, see [sp_addmessage \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
message_id	int	ID of the message. Is unique across server. Message IDs less than 50000 are system messages.
language_id	smallint	Language ID for which the text in text is used, as defined in syslanguages . This is unique for a specified message_id .
severity	tinyint	Severity level of the message, between 1 and 25. This is the same for all message languages within a message_id .
is_event_logged	bit	1 = Message is event-logged when an error is raised. This is the same for all message languages within a message_id .
text	nvarchar(2048)	Text of the message used when the corresponding language_id is active.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[THROW \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Messages \(for Errors\) Catalog Views \(Transact-SQL\)](#)



[Exception Message Box Programming](#)

[Error Messages](#)

[Database Engine Events and Errors](#)

Extended Properties Catalog Views - sys.extended_properties

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each extended property in the current database.

COLUMN NAME	DATA TYPE	DESCRIPTION
class	tinyint	Identifies the class of item on which the property exists. Can be one of the following: 0 = Database 1 = Object or column 2 = Parameter 3 = Schema 4 = Database principal 5 = Assembly 6 = Type 7 = Index 10 = XML schema collection 15 = Message type 16 = Service contract 17 = Service 18 = Remote service binding 19 = Route 20 = Dataspace (filegroup or partition scheme) 21 = Partition function 22 = Database file 27 = Plan guide

COLUMN NAME	DATA TYPE	DESCRIPTION
class_desc	nvarchar(60)	<p>Description of the class on which the extended property exists. Can be one of the following:</p> <p>DATABASE</p> <p>OBJECT_OR_COLUMN</p> <p>PARAMETER</p> <p>SCHEMA</p> <p>DATABASE_PRINCIPAL</p> <p>ASSEMBLY</p> <p>TYPE</p> <p>INDEX</p> <p>XML_SCHEMA_COLLECTION</p> <p>MESSAGE_TYPE</p> <p>SERVICE_CONTRACT</p> <p>SERVICE</p> <p>REMOTE_SERVICE_BINDING</p> <p>ROUTE</p> <p>DATASPACE</p> <p>PARTITION_FUNCTION</p> <p>DATABASE_FILE</p> <p>PLAN_GUIDE</p>
major_id	int	<p>ID of the item on which the extended property exists, interpreted according to its class. For most items, this is the ID that applies to what the class represents. Interpretation for nonstandard major IDs is as follows:</p> <p>If class is 0, major_id is always 0.</p> <p>If class is 1, 2, or 7 major_id is object_id.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
minor_id	int	<p>Secondary ID of the item on which the extended property exists, interpreted according to its class. For most items this is 0; otherwise, the ID is as follows:</p> <p>If class = 1, minor_id is the column_id if column, else 0 if object.</p> <p>If class = 2, minor_id is the parameter_id.</p> <p>If class 7 = minor_id is the index_id.</p>
name	sysname	Property name, unique with class, major_id, and minor_id.
value	sql_variant	Value of the extended property.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Extended Properties Catalog Views \(Transact-SQL\)](#)

[sys.fn_listextendedproperty \(Transact-SQL\)](#)





[sp_addextendedproperty \(Transact-SQL\)](#)

[sp_dropextendedproperty \(Transact-SQL\)](#)

[sp_updateextendedproperty \(Transact-SQL\)](#)

Change Tracking Catalog Views - sys.change_tracking_databases

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each database that has change tracking enabled.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	ID of the database. This is unique within the instance of SQL Server.
is_auto_cleanup_on	bit	Indicates whether change tracking data is automatically cleaned up after the configured retention period: 0 = Off 1 = On
retention_period	int	If autocleanup is being used, the retention period specifies how long the change tracking data is kept in the database.
retention_period_units_desc	nvarchar(60)	Specifies the description of the retention period: Minutes Hours Days
retention_period_units	tinyint	Unit of time for the retention period: 1 = Minutes 2 = Hours 3 = Days





Permissions

The same permission checks are made for sys.change_tracking_databases as are made for sys.databases. If the caller of sys.change_tracking_databases is not the owner of the database, the minimum permissions that are required to see the corresponding row are ALTER ANY DATABASE or VIEW ANY DATABASE server-level permission, or CREATE DATABASE permission in the master database or current database.

See Also

Change Tracking Catalog Views - sys.change_tracking_tables

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each table in the current database that has change tracking enabled.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of a table that has a change journal. The table can have a change journal even if change tracking is currently off. The table ID is unique within the database.
is_track_columns_updated_on	bit	Current state of change tracking on the table: 0 = OFF 1 = ON
begin_version	bigint	Version of the database when change tracking began for the table. This version is usually indicates when change tracking was enabled, but this value is reset if the table is truncated.
cleanup_version	bigint	Version up to which cleanup might have removed change tracking information.
min_valid_version	bigint	Minimum valid version of change tracking information that is available for the table. When obtaining changes from the table that is associated with this row, the value of last_sync_version must be greater than or equal to the version reported by this column. For more information, see CHANGE_TRACKING_MIN_VALID_VERSION (Transact-SQL) .

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[CHANGE_TRACKING_MIN_VALID_VERSION \(Transact-SQL\)](#)

[Change Tracking Catalog Views \(Transact-SQL\)](#)

[Track Data Changes \(SQL Server\)](#)

Database Mirroring Witness Catalog Views - sys.database_mirroring_witnesses

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for every witness role that a server plays in a database mirroring partnership.

In a database mirroring session, automatic failover requires a witness server. Ideally, the witness resides on a separate computer from both the principal and mirror servers. The witness does not serve the database. Instead, it monitors the status of the principal and mirror servers. If the principal server fails, the witness may initiate automatic failover to the witness server.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_name	sysname	Name of the two copies of the database in the database mirroring session.
principal_server_name	sysname	Name of partner server whose copy of the database is currently the principal database.
mirror_server_name	sysname	Name of the partner server whose copy of the database is currently the mirror database.
safety_level	tinyint	Transaction safety setting for updates on the mirror database: 0 = Unknown state 1 = Off (asynchronous) 2 = Full (synchronous) Using a witness for automatic failover requires full transaction safety, which is the default.
safety_level_desc	nvarchar(60)	Description of safety guarantee of updates on the mirror database: UNKNOWN OFF FULL
safety_sequence_number	int	Update sequence number for changes to safety_level .

COLUMN NAME	DATA TYPE	DESCRIPTION
role_sequence_number	int	Update sequence number for changes to principal/mirror roles played by the mirroring partners.
mirroring_guid	uniqueidentifier	Identifier of the mirroring partnership.
family_guid	uniqueidentifier	Identifier of the backup family for the database. Used for detecting matching restore states.
is_suspended	bit	Database mirroring is suspended.
is_suspended_sequence_number	int	Sequence number for setting is_suspended .
partner_sync_state	tinyint	<p>Synchronization state of the mirroring session:</p> <p>5 = The partners are synchronized. Failover is potentially possible. For information about the requirements for failover see, Role Switching During a Database Mirroring Session (SQL Server).</p> <p>6 = The partners are not synchronized. Failover is not possible now.</p>
partner_sync_state_desc	nvarchar(60)	<p>Description of the synchronization state of the mirroring session:</p> <p>SYNCHRONIZED</p> <p>UNSYNCHRONIZED</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Database Mirroring Witness](#)





[sys.database_mirroring](#) (Transact-SQL)

[sys.database_mirroring_endpoints](#) (Transact-SQL)

[Querying the SQL Server System Catalog FAQ](#)

Data-tier Application Views - dbo.sysdac_instances

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each data-tier application (DAC) instance deployed to an instance of the Database Engine. sysdac_instances belongs to the dbo schema in the msdb database. The following table describes the columns in the sysdac_instances view.

COLUMN NAME	DATA TYPE	DESCRIPTION
instance_id	uniqueidentifier	Identifier of the DAC instance.
instance_name	sysname	Name of the DAC instance specified when the DAC was deployed.
type_name	sysname	Name of the DAC specified when the DAC package was created.
type_version	nvarchar(64)	Version of the DAC specified when the DAC package was created.
description	nvarchar(4000)	A description of the DAC written when the DAC package was created.
type_stream	varbinary(max)	A bit stream that contains an encoded representation of the logical objects, such as tables and views, contained in the DAC.
date_created	datetime	Date and time the DAC instance was created.
created_by	sysname	Login that created the DAC instance.
database_name	sysname	Name of the database created for the DAC instance.

Remarks

A DAC includes a DAC type, which is a definition of the logical data-tier objects used by an application, such as tables and views. A DAC package is a file used to deploy a DAC. The DAC package contains a representation of all the logical objects contained in the DAC type. The DAC package can be used to deploy one or more copies, or instances, of the DAC to an instance of the Database Engine. Each DAC instance deployed from the same DAC package shares the same type, but is assigned a unique instance name and identifier.

Permissions

Requires membership in the sysadmin fixed server role to view all of the columns. Members of the public role can view the instance_name, description, and type_version columns.





See Also

[Data-tier Applications](#)

[Data-tier Application Views \(Transact-SQL\)](#)

Always On Availability Groups Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains catalog views and functions that are related to Always On availability groups.

In This Section

sys.availability_databases_cluster	sys.availability_groups_cluster
sys.availability_group_listener_ip_addresses	sys.availability_read_only_routing_lists
sys.availability_group_listeners	sys.availability_replicas
sys.availability_groups	

NOTE

For information about joined availability databases, see the **replica_id** and **group_database_id** columns in [sys.databases](#) (Transact-SQL);.

See Also

[sys.databases](#) (Transact-SQL)

[sys.database_mirroring_endpoints](#) (Transact-SQL)





[Always On Availability Groups](#) (SQL Server)

[Monitor Availability Groups](#) (Transact-SQL);

[Always On Availability Groups Dynamic Management Views and Functions](#) (Transact-SQL)

sys.availability_databases_cluster (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each availability database on the instance of SQL Server that is hosting an availability replica for any Always On availability group in the Windows Server Failover Clustering (WSFC) cluster, regardless of whether the local copy database has been joined to the availability group yet.

NOTE

When a database is added to an availability group, the primary database is automatically joined to the group. Secondary databases must be prepared on each secondary replica before they can be joined to the availability group.

COLUMN NAME	DATA TYPE	DESCRIPTION
group_id	uniqueidentifier	Unique identifier of the availability group in which the availability group, if any, in which the database is participating. NULL = database is not part of an availability replica of in availability group.
group_database_id	uniqueidentifier	Unique identifier of the database within the availability group, if any, in which the database is participating. group_database_id is the same for this database on the primary replica and on every secondary replica on which the database has been joined to the availability group. NULL = database is not part of an availability replica in any availability group.
database_name	sysname	Name of the database that was added to the availability group.

Permissions

If the caller of **sys.availability_databases_cluster** is not the owner of the database, the minimum permissions required to see the corresponding row are ALTER ANY DATABASE or VIEW ANY DATABASE server-level permission, or CREATE DATABASE permission in the **master** database.

See Also

[sys.availability_groups \(Transact-SQL\)](#)

[sys.databases \(Transact-SQL\)](#)





[sys.dm_hadr_database_replica_states \(Transact-SQL\)](#)

[sys.dm_hadr_database_replica_cluster_states \(Transact-SQL\)](#)

[Overview of Always On Availability Groups \(SQL Server\)](#)

sys.availability_group_listener_ip_addresses (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for every IP address that is associated with any Always On availability group listener in the Windows Server Failover Clustering (WSFC) cluster.

Primary key: **listener_id** + **ip_address** + **ip_sub_mask**

COLUMN NAME	DATA TYPE	DESCRIPTION
listener_id	nvarchar(36)	Resource GUID from Windows Server Failover Clustering (WSFC) cluster.
ip_address	nvarchar(48)	Configured virtual IP address of the availability group listener. Returns a single IPv4 or IPv6 address.
ip_subnet_mask	nvarchar(15)	Configured IP subnet mask for the IPv4 address, if any, that is configured for the availability group listener. NULL = IPv6 subnet
is_dhcp	bit	Whether the IP address is configured by DHCP, one of: 0 = IP address is not configured by DHCP. 1 = IP address is configured by DHCP
network_subnet_ip	nvarchar(48)	Network subnet IP address that specifies the subnet to which the IP address belongs.
network_subnet_prefix_length	int	Network subnet prefix length of the subnet to which the IP address belongs.
network_subnet_ipv4_mask	nvarchar(45)	Network subnet mask of the subnet to which the IP address belongs. network_subnet_ipv4_mask to specify the DHCP <network_subnet_option> options in a WITH DHCP clause of the CREATE AVAILABILITY GROUP or ALTER AVAILABILITY GROUP Transact-SQL statement. NULL = IPv6 subnet

COLUMN NAME	DATA TYPE	DESCRIPTION
state	tinyint	<p>IP resource ONLINE/OFFLINE state from the WSFC cluster, one of:</p> <p>1 = Online. IP resource is online.</p> <p>0 = Offline. IP resource is offline.</p> <p>2 = Online Pending. IP resource is offline but is being brought online.</p> <p>3 = Failed. IP resource was being brought online but failed.</p>
state_desc	nvarchar(60)	<p>Description of state, one of:</p> <p>ONLINE</p> <p>OFFLINE</p> <p>ONLINE_PENDING</p> <p>FAILED</p>

Security

Permissions





The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Querying the SQL Server System Catalog FAQ](#)
[Catalog Views \(Transact-SQL\)](#)

sys.availability_group_listeners (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

For each Always On availability group, returns either zero rows indicating that no network name is associated with the availability group, or returns a row for each availability-group listener configuration in the Windows Server Failover Clustering (WSFC) cluster. This view displays the real-time configuration gathered from cluster.

NOTE

This catalog view does not describe details of an IP configuration, that was defined in the WSFC cluster.

COLUMN NAME	DATA TYPE	DESCRIPTION
group_id	uniqueidentifier	Availability group ID (group_id) from sys.availability_groups .
listener_id	nvarchar(36)	GUID from the cluster resource ID.
dns_name	nvarchar(63)	Configured network name (hostname) of the availability group listener.
port	int	<p>The TCP port number configured for the availability group listener.</p> <p>NULL = Listener was configured outside SQL Server and its port number has not been added to the availability group. To add the port, please use the MODIFY LISTENER option of the ALTER AVAILABILITY GROUP Transact-SQL statement.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
is_conformant	bit	<p>Whether this IP configuration is conformant, one of:</p> <p>1 = Listener is conformant. Only “OR” relations exist among its Internet Protocol (IP) addresses. <i>Conformant</i> encompasses every an IP configuration that was created by the CREATE AVAILABILITY GROUP Transact-SQL statement. In addition, if an IP configuration that was created outside of SQL Server, for example by using the WSFC Failover Cluster Manager, but can be modified by the ALTER AVAILABILITY GROUP TSQL statement, the IP configuration qualifies as conformant.</p> <p>0 = Listener is nonconformant. Typically, this indicates an IP address that could not be configured by using SQL Server commands and, instead, was defined directly in the WSFC cluster.</p>
ip_configuration_string_from_cluster	nvarchar(max)	<p>Cluster IP configuration strings, if any, for this listener. NULL = Listener has no virtual IP addresses. For example:</p> <p>IPv4 address: <code>65.55.39.10</code>.</p> <p>IPv6 address: <code>2001::4898:23:1002:20f:1fff:feff:b3a3</code></p>

Security

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Always On Availability Groups Dynamic Management Views and Functions \(Transact-SQL\)](#)





[Always On Availability Groups Catalog Views \(Transact-SQL\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

[Always On Availability Groups \(SQL Server\)](#)

sys.availability_groups (Transact-SQL)

11/16/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each availability group for which the local instance of SQL Server hosts an availability replica. Each row contains a cached copy of the availability group metadata.

COLUMN NAME	DATA TYPE	DESCRIPTION
group_id	uniqueidentifier	Unique identifier (GUID) of the availability group.
name	sysname	Name of the availability group. This is a user-specified name that must be unique within the Windows Server Failover Cluster (WSFC).
resource_id	nvarchar(40)	Resource ID for the WSFC cluster resource.
resource_group_id	nvarchar(40)	Resource Group ID for the WSFC cluster resource group of the availability group.
failure_condition_level	int	<p>User-defined failure condition level under which an automatic failover must be triggered, one of the integer values shown in the table immediately below this table.</p> <p>The failure-condition levels (1–5) range from the least restrictive, level 1, to the most restrictive, level 5. A given condition level encompasses all of the less restrictive levels. Thus, the strictest condition level, 5, includes the four less restrictive condition levels (1–4), level 4 includes levels 1–3, and so forth.</p> <p>To change this value, use the FAILURE_CONDITION_LEVEL option of the ALTER AVAILABILITY GROUP Transact-SQL statement.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
health_check_timeout	int	<p>Wait time (in milliseconds) for the sp_server_diagnostics system stored procedure to return server-health information, before the server instance is assumed to be slow or hung. The default value is 30000 milliseconds (30 seconds).</p> <p>To change this value, use the HEALTH_CHECK_TIMEOUT option of the ALTER AVAILABILITY GROUP Transact-SQL statement.</p>
automated_backup_preference	tinyint	<p>Preferred location for performing backups on the availability databases in this availability group. The following are the possible values and their descriptions.</p> <p>0 : Primary. Backups should always occur on the primary replica.</p> <p>1 : Secondary only. Performing backups on a secondary replica is preferable.</p> <p>2 : Prefer Secondary. Performing backups on a secondary replica preferable, but performing backups on the primary replica is acceptable if no secondary replica is available for backup operations. This is the default behavior.</p> <p>3 : Any Replica. No preference about whether backups are performed on the primary replica or on a secondary replica.</p> <p>For more information, see Active Secondaries: Backup on Secondary Replicas (Always On Availability Groups).</p>
automated_backup_preference_desc	nvarchar(60)	<p>Description of automated_backup_preference, one of:</p> <p>PRIMARY</p> <p>SECONDARY_ONLY</p> <p>SECONDARY</p> <p>NONE</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
version	smallint	The version of the availability group metadata stored in the Windows Failover Cluster. This version number is incremented when new features are added.
basic_features	bit	Specifies whether this is a Basic availability group. For more information, see Basic Availability Groups (Always On Availability Groups) .
dtc_support	bit	Specifies whether DTC support has been enabled for this availability group. The DTC_SUPPORT option of CREATE AVAILABILITY GROUP controls this setting.
db_failover	bit	Specifies whether the availability group supports failover for database health conditions. The DB_FAILOVER option of CREATE AVAILABILITY GROUP controls this setting.
is_distributed	bit	Specifies whether this is a distributed availability group. For more information, see Distributed Availability Groups (Always On Availability Groups) .

Failure condition level values

The following table describes the possible failure condition levels for the **failure_condition_level** column.

VALUE	FAILURE CONDITION
1	<p>Specifies that an automatic failover should be initiated when any of the following occurs:</p> <ul style="list-style-type: none"> - The SQL Server service is down. - The lease of the availability group for connecting to the WSFC failover cluster expires because no ACK is received from the server instance. For more information, see How It Works: SQL Server Always On Lease Timeout.
2	<p>Specifies that an automatic failover should be initiated when any of the following occurs:</p> <ul style="list-style-type: none"> - The instance of SQL Server does not connect to cluster, and the user-specified health_check_timeout threshold of the availability group is exceeded. - The availability replica is in failed state.

VALUE	FAILURE CONDITION
3	<p>Specifies that an automatic failover should be initiated on critical SQL Server internal errors, such as orphaned spinlocks, serious write-access violations, or too much dumping.</p> <p>This is the default value.</p>
4	<p>Specifies that an automatic failover should be initiated on moderate SQL Server internal errors, such as a persistent out-of-memory condition in the SQL Server internal resource pool.</p>
5	<p>Specifies that an automatic failover should be initiated on any qualified failure conditions, including:</p> <ul style="list-style-type: none"> - Exhaustion of SQL Engine worker-threads. - Detection of an unsolvable deadlock.

Security

Permissions

Requires VIEW ANY DEFINITION permission on the server instance.

See Also

[sys.availability_replicas \(Transact-SQL\)](#)





[Always On Availability Groups \(SQL Server\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

sys.availability_groups_cluster (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each Always On availability group in the Windows Server Failover Clustering (WSFC) . Each row contains the availability group metadata from the WSFC cluster.

COLUMN NAME	DATA TYPE	DESCRIPTION
group_id	uniqueidentifier	Unique identifier (GUID) of the availability group.
name	sysname	Name of the availability group. This is a user-specified name that must be unique within the Windows Server Failover Cluster (WSFC).
resource_id	nvarchar(40)	Resource ID for the WSFC cluster resource.
resource_group_id	nvarchar(40)	Resource Group ID for the WSFC cluster resource group of the availability group.

COLUMN NAME	DATA TYPE	DESCRIPTION
failure_condition_level	int	<p>User-defined failure condition level under which an automatic failover must be triggered, one of the following integer values:</p> <p>1: Specifies that an automatic failover should be initiated when any of the following occurs:</p> <ul style="list-style-type: none"> - The SQL Server service is down. - The lease of the availability group for connecting to the WSFC failover cluster expires because no ACK is received from the server instance. For more information, see How It Works: SQL Server Always On Lease Timeout. <p>2: Specifies that an automatic failover should be initiated when any of the following occurs:</p> <ul style="list-style-type: none"> - The instance of SQL Server does not connect to cluster, and the user-specified health_check_timeout threshold of the availability group is exceeded. - The availability replica is in failed state. <p>3: Specifies that an automatic failover should be initiated on critical SQL Server internal errors, such as orphaned spinlocks, serious write-access violations, or too much dumping. This is the default value.</p> <p>4: Specifies that an automatic failover should be initiated on moderate SQL Server internal errors, such as a persistent out-of-memory condition in the SQL Server internal resource pool.</p> <p>5: Specifies that an automatic failover should be initiated on any qualified failure conditions, including:</p> <ul style="list-style-type: none"> - Exhaustion of SQL Engine worker-threads. - Detection of an unsolvable deadlock. <p>The failure-condition levels (1–5) range from the least restrictive, level 1, to the most restrictive, level 5. A given condition level encompasses all of the less restrictive levels. Thus, the strictest condition level, 5, includes the four less restrictive condition levels (1-4), level 4 includes levels 1-3, and so forth.</p> <p>To change this value, use the FAILURE_CONDITION_LEVEL option of the ALTER AVAILABILITY GROUP Transact-SQL statement.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
health_check_timeout	int	<p>Wait time (in milliseconds) for the sp_server_diagnostics system stored procedure to return server-health information, before the server instance is assumed to be slow or hung. The default value is 30000 milliseconds (30 seconds).</p> <p>To change this value, use the HEALTH_CHECK_TIMEOUT option of ALTER AVAILABILITY GROUP Transact-SQL statement.</p>
automated_backup_preference	tinyint	<p>Preferred location for performing backups on the availability databases in this availability group. One of the following values:</p> <p>0: Primary. Backups should always occur on the primary replica. 1: Secondary only. Performing backups on a secondary replica is preferable. 2: Prefer Secondary. Performing backups on a secondary replica preferable, but performing backups on the primary replica is acceptable if no secondary replica is available for backup operations. This is the default behavior. 3: Any Replica. No preference about whether backups are performed on the primary replica or on a secondary replica.</p> <p>For more information, see Active Secondaries: Backup on Secondary Replicas (Always On Availability Groups).</p>
automated_backup_preference_desc	nvarchar(60)	<p>Description of automated_backup_preference, one of:</p> <p>PRIMARY</p> <p>SECONDARY_ONLY</p> <p>SECONDARY</p> <p>NONE</p>

Security

Permissions

Requires VIEW ANY DEFINITION permission on the server instance.

See Also

[sys.availability_replicas](#) (Transact-SQL)





[Always On Availability Groups](#) (SQL Server)

Monitor Availability Groups (Transact-SQL)

Monitor Availability Groups (Transact-SQL)

sys.availability_read_only_routing_lists (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for the read only routing list of each availability replica in an Always On availability group in the WSFC failover cluster.

COLUMN NAME	DATA TYPE	DESCRIPTION
replica_id	uniqueidentifier	Unique ID of the availability replica that owns the routing list.
routing_priority	int	Priority order for routing (1 is first, 2 is second, and so forth).
read_only_replica_id	uniqueidentifier	Unique ID of the availability replica to which a read-only workload will be routed.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Always On Availability Groups Dynamic Management Views and Functions \(Transact-SQL\)](#)





[Always On Availability Groups Catalog Views \(Transact-SQL\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

[Always On Availability Groups \(SQL Server\)](#)

sys.availability_replicas (Transact-SQL)

11/16/2017 • 5 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each of the availability replicas that belong to any Always On availability group in the WSFC failover cluster.

If the local server instance is unable to talk to the WSFC failover cluster, for example because the cluster is down or quorum has been lost, only rows for local availability replicas are returned. These rows will contain only the columns of data that are cached locally in metadata.

COLUMN NAME	DATA TYPE	DESCRIPTION
replica_id	uniqueidentifier	Unique ID of the replica.
group_id	uniqueidentifier	Unique ID of the availability group to which the replica belongs.
replica_metadata_id	int	ID for the local metadata object for availability replicas in the Database Engine.
replica_server_name	nvarchar(256)	Server name of the instance of SQL Server that is hosting this replica and, for a non-default instance, its instance name.
owner_sid	varbinary(85)	Security identifier (SID) registered to this server instance for the external owner of this availability replica. NULL for non-local availability replicas.
endpoint_url	nvarchar(128)	String representation of the user-specified database mirroring endpoint that is used by connections between primary and secondary replicas for data synchronization. For information about the syntax of endpoint URLs, see Specify the Endpoint URL When Adding or Modifying an Availability Replica (SQL Server) . NULL = Unable to talk to the WSFC failover cluster. To change this endpoint, use the ENDPOINT_URL option of ALTER AVAILABILITY GROUP Transact-SQL statement.

COLUMN NAME	DATA TYPE	DESCRIPTION
availability_mode	tinyint	<p>The availability mode of the replica, one of:</p> <p>0 Asynchronous commit. The primary replica can commit transactions without waiting for the secondary to write the log to disk.</p> <p>1 Synchronous commit. The primary replica waits to commit a given transaction until the secondary replica has written the transaction to disk.</p> <p>4 Configuration only. The primary replica sends availability group configuration metadata to the replica synchronously. User data is not transmitted to the replica. Available in SQL Server 2017 CU1 and later.</p> <p>For more information, see Availability Modes (Always On Availability Groups).</p>
availability_mode_desc	nvarchar(60)	<p>Description of availability_mode, one of:</p> <p>ASYNCHRONOUS_COMMIT</p> <p>SYNCHRONOUS_COMMIT</p> <p>CONFIGURATION_ONLY</p> <p>To change this the availability mode of an availability replica, use the AVAILABILITY_MODE option of ALTER AVAILABILITY GROUP Transact-SQL statement.</p> <p>You cannot change the availability mode of a replica to CONFIGURATION_ONLY. You cannot change a CONFIGURATION_ONLY replica to a secondary or primary replica.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
failover_mode	tinyint	<p>The failover mode of the availability replica, one of:</p> <p>0 Manual failover. A failover to a secondary replica set to manual failover must be manually initiated by the database administrator. The type of failover that is performed will depend on whether the secondary replica is synchronized, as follows:</p> <p>If the availability replica is not synchronizing or is still synchronizing, only forced failover (with possible data loss) can occur.</p> <p>If the availability mode is set to synchronous commit (availability_mode = 1) and the availability replica is currently synchronized, manual failover without data loss can occur.</p> <p>1 Automatic failover. The replica is a potential target for automatic failovers. Automatic failover is supported only if the availability mode is set to synchronous commit (availability_mode = 1) and the availability replica is currently synchronized.</p> <p>To view a rollup of the database synchronization health of every availability database in an availability replica, use the synchronization_health and synchronization_health_desc columns of the sys.dm_hadr_availability_replica_states dynamic management view. The rollup considers the synchronization state of every availability database and the availability mode of its availability replica.</p> <p>Note: To view the synchronization health of a given availability database, query the synchronization_state and synchronization_health columns of the sys.dm_hadr_database_replica_states dynamic management view.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
failover_mode_desc	nvarchar(60)	<p>Description of failover_mode, one of:</p> <p>MANUAL</p> <p>AUTOMATIC</p> <p>To change the failover mode, use the FAILOVER_MODE option of ALTER AVAILABILITY GROUP Transact-SQL statement.</p>
session_timeout	int	<p>The time-out period, in seconds. The time-out period is the maximum time that the replica waits to receive a message from another replica before considering connection between the primary and secondary replica have failed. Session timeout detects whether secondaries are connected the primary replica.</p> <p>On detecting a failed connection with a secondary replica, the primary replica considers the secondary replica to be NOT_SYNCHRONIZED. On detecting a failed connection with the primary replica, a secondary replica simply attempts to reconnect.</p> <p>Note: Session timeouts do not cause automatic failovers.</p> <p>To change this value, use the SESSION_TIMEOUT option of ALTER AVAILABILITY GROUP Transact-SQL statement.</p>
primary_role_allow_connections	tinyint	<p>Whether the availability allows all connections or only read-write connections, one of:</p> <p>2 = All (default)</p> <p>3 = Read write</p>
primary_role_allow_connections_desc	nvarchar(60)	<p>Description of primary_role_allow_connections, one of:</p> <p>ALL</p> <p>READ_WRITE</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
secondary_role_allow_connections	tinyint	<p>Whether an availability replica that is performing the secondary role (that is, a secondary replica) can accept connections from clients, one of:</p> <p>0 = No. No connections are allowed to the databases in the secondary replica, and the databases are not available for read access. This is the default setting.</p> <p>1 = Read only. Only read-only connections are allowed to the databases in the secondary replica. All database(s) in the replica are available for read access.</p> <p>2 = All. All connections are allowed to the databases in the secondary replica for read-only access.</p> <p>For more information, see Active Secondaries: Readable Secondary Replicas (Always On Availability Groups).</p>
secondary_role_allow_connections_desc	nvarchar(60)	<p>Description of secondary_role_allow_connections, one of:</p> <p>NO</p> <p>READ_ONLY</p> <p>ALL</p>
create_date	datetime	<p>Date that the replica was created.</p> <p>NULL = Replica not on this server instance.</p>
modify_date	datetime	<p>Date that the replica was last modified.</p> <p>NULL = Replica not on this server instance.</p>
backup_priority	int	<p>Represents the user-specified priority for performing backups on this replica relative to the other replicas in the same availability group. The value is an integer in the range of 0..100.</p> <p>For more information, see Active Secondaries: Backup on Secondary Replicas (Always On Availability Groups).</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
read_only_routing_url	nvarchar(256)	Connectivity endpoint (URL) of the read only availability replica. For more information, see Configure Read-Only Routing for an Availability Group (SQL Server) .

Security

Permissions

Requires VIEW ANY DEFINITION permission on the server instance.

See Also

[sys.availability_groups \(Transact-SQL\)](#)

[Overview of Always On Availability Groups \(SQL Server\)](#)





[Always On Availability Groups \(SQL Server\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

[Monitor Availability Groups \(Transact-SQL\)](#)

Azure SQL Database Catalog Views

11/21/2017 • 1 min to read • [Edit Online](#)





THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The following catalog views are available in this section.

sys.bandwidth_usage (Azure SQL Database)	sys.event_log (Azure SQL Database)
sys.database_connection_stats (Azure SQL Database)	sys.firewall_rules (Azure SQL Database)
sys.database_firewall_rules (Azure SQL Database)	sys.resource_stats (Azure SQL Database)
sys.database_usage (Azure SQL Database)	sys.resource_usage (Azure SQL Database)
sys.elastic_pool_resource_stats (Azure SQL Database)	dbo.server_quotas (Azure SQL Database)

sys.bandwidth_usage (Azure SQL Database)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Note: This applies only to Azure SQL Database V11.

Returns information about the network bandwidth used by each database in a **Azure SQL Database V11 logical server**. Each row returned for a given database summarizes a single direction and class of usage over a one-hour period.

This has been deprecated in a Azure SQL Database V12 logical server.

The **sys.bandwidth_usage** view contains the following columns.

COLUMN NAME	DESCRIPTION
time	The hour when the bandwidth was consumed. The rows in this view are on a per-hour basis. For example, 2009-09-19 02:00:00.000 means that the bandwidth was consumed on September 19, 2009 between 2:00 A.M. and 3:00 A.M.
database_name	The name of the database that used bandwidth.
direction	The type of bandwidth that was used, one of: Ingress: Data that is moving into the Azure SQL Database. Egress: Data that is moving out of the Azure SQL Database.
class	The class of bandwidth that was used, one of: Internal: Data that is moving within the Azure platform. External: Data that is moving out of the Azure platform. This class is returned only if the database is engaged in a continuous copy relationship between regions (Active Geo-Replication). If a given database does not participate in any continuous copy relationship, then "Interlink" rows are not returned. For more information, see the "Remarks" section later in this topic.
time_period	The time period when the usage occurred is either Peak or OffPeak. The Peak time is based on the region in which the server was created. For example, if a server was created in the "US_Northwest" region, the Peak time is defined as being between 10:00 A.M. and 6:00 P.M. PST.
quantity	The amount of bandwidth, in kilobytes (KBs), that was used.

Permissions

This view is only available in the **master** database to the server-level principal login.

Remarks

External and Internal Classes

For each database used at a given time, the **sys.bandwidth_usage** view returns rows that show the class and direction of bandwidth usage. The following example illustrates data that might be exposed for a given database. In this example, the time is 2012-04-21 17:00:00, which occurs during the peak time period. The database name is Db1. In this example, **sys.bandwidth_usage** has returned a row for all four combinations of the Ingress and Egress directions and External and Internal classes, as follows:

TIME	DATABASE_NAME	DIRECTION	CLASS	TIME_PERIOD	QUANTITY
2012-04-21 17:00:00	Db1	Ingress	External	Peak	66
2012-04-21 17:00:00	Db1	Egress	External	Peak	741
2012-04-21 17:00:00	Db1	Ingress	Internal	Peak	1052
2012-04-21 17:00:00	Db1	Egress	Internal	Peak	3525

Interpreting Data Direction for Active Geo-Replication





For Active Geo-Replication, bandwidth-usage data is visible in the logical master database on both sides of a continuous copy relationship. So you must interpret the ingress and egress direction indicators from the perspective of the logical server that you are querying. For example, consider a replication stream that transfers 1MB of data from the source server to the target server. In this case, on the source server the 1MB counts toward total data sent, and on the target server, the 1MB is recorded as data received.

NOTE

The bulk of data transferred is from the source server to the target server, in the direction of user data flow. However, some data transfer is required in the other direction.

sys.database_connection_stats (Azure SQL Database)

11/21/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains statistics for SQL Database database **connectivity** events, providing an overview of database connection successes and failures. For more information about connectivity events, see Event Types in [sys.event_log \(Azure SQL Database\)](#).

STATISTIC	TYPE	DESCRIPTION
database_name	sysname	Name of the database.
start_time	datetime2	UTC date and time of the start of the aggregation interval. The time is always a multiple of 5 minutes. For example: '2011-09-28 16:00:00' '2011-09-28 16:05:00' '2011-09-28 16:10:00'
end_time	datetime2	UTC date and time of the end of the aggregation interval. End_time is always exactly 5 minutes later than the corresponding start_time in the same row.
success_count	int	Number of successful connections.
total_failure_count	int	Total number of failed connections. This is the sum of connection_failure_count , terminated_connection_count , and throttled_connection_count , and does not include deadlock events.
connection_failure_count	int	Number of login failures.
terminated_connection_count	int	Only applicable for Azure SQL Database v11. Number of terminated connections.
throttled_connection_count	int	Only applicable for Azure SQL Database v11. Number of throttled connections.

Remarks

Event Aggregation

Event information for this view is collected and aggregated within 5-minute intervals. The count columns represent

the number of times a particular connectivity event occurred for a specific database within a given time interval.

For example, if a user fails to connect to database Database1 seven times between 11:00 and 11:05 on 2/5/2012 (UTC), this information is available in a single row in this view:

DATABASE_NAME	START_TIME	END_TIME	SUCCESS_COUNT	TOTAL_FAILURE_COUNT	CONNECTION_FAILURE_COUNT	TERMINATED_CONNECTION_COUNT	THROTTLED_CONNECTION_COUNT
Database1	2012-02-05 11:00:00	2012-02-05 11:05:00	0	7	7	0	0

Interval start_time and end_time

An event is included in an aggregation interval when the event occurs *on or after* **start_time** and *before* **end_time** for that interval. For example, an event occurring exactly at `2012-10-30 19:25:00.0000000` would be included only in the second interval shown below:

start_time	end_time
2012-10-30 19:20:00.0000000	2012-10-30 19:25:00.0000000
2012-10-30 19:25:00.0000000	2012-10-30 19:30:00.0000000

Data Updates

Data in this view is accumulated over time. Typically, the data is accumulated within an hour of the start of the aggregation interval, but it may take up to a maximum of 24 hours for all the data to appear in the view. During that time, the information within a single row may be updated periodically.

Data Retention

The data in this view is retained for a maximum of 30 days, or possibly less depending on the number of databases in the logical server and the number of unique events each database generates. To retain this information for a longer period, copy the data to a separate database. After you make an initial copy of the view, the rows in the view may be updated as data is accumulated. To keep your copy of the data up-to-date, periodically do a table scan of the rows to look for an increase in the event count of existing rows and to identify new rows (you can identify unique rows by using the start and end times), then update your copy of the data with these changes.

Errors Not Included

This view may not include all connection and error information:

- This view does not include all SQL Database database errors that could occur, only those specified in Event Types in [sys.event_log \(Azure SQL Database\)](#).
- If there is a machine failure within the SQL Database datacenter, a small amount of data for your logical server may be missing from the event table.
- If an IP address has been blocked through DoSGuard, connection attempt events from that IP address cannot be collected and will not appear in this view.

Permissions

Users with permission to access the **master** database have read-only access to this view.

Example

The following example shows a query of **sys.database_connection_stats** to return a summary of the database connections that occurred between noon on 9/25/2011 and noon on 9/28/2011 (UTC). By default, the query

results are sorted by **start_time** (ascending order).





```
SELECT *  
FROM sys.database_connection_stats  
WHERE start_time>='2011-09-25:12:00:00' and end_time<='2011-09-28 12:00:00';
```

See Also

[Troubleshooting Windows Azure SQL Database](#)

sys.database_firewall_rules (Azure SQL Database)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about the database-level firewall settings associated with your Microsoft Azure SQL Database. Database-level firewall settings are particularly useful when using contained database users. For more information, see [Contained Database Users - Making Your Database Portable](#).

The `sys.database_firewall_rules` view contains the following columns:

COLUMN NAME	DATA TYPE	DESCRIPTION
id	INTEGER	The identifier of the database-level firewall setting.
name	NVARCHAR(128)	The name you chose to describe and distinguish the database-level firewall setting.
start_ip_address	VARCHAR(50)	The lowest IP address in the range of the database-level firewall setting. IP addresses equal to or greater than this can attempt to connect to the SQL Database instance. The lowest possible IP address is <code>0.0.0.0</code> .
end_ip_address	VARCHAR(50)	The highest IP address in the range of the firewall setting. IP addresses equal to or less than this can attempt to connect to the SQL Database instance. The highest possible IP address is <code>255.255.255.255</code> . Note: Windows Azure connection attempts are allowed when both this field and the start_ip_address field equals <code>0.0.0.0</code> .
create_date	DATETIME	UTC date and time when the database-level firewall setting was created.
modify_date	DATETIME	UTC date and time when the database-level firewall setting was last modified.

Remarks

To remove a database firewall rule, use [sp_delete_database_firewall_rule \(Azure SQL Database\)](#). To set a firewall rule for all of SQL Database, see [sp_set_firewall_rule \(Azure SQL Database\)](#). To return information about existing database firewall rules, query [sys.database_firewall_rules \(Azure SQL Database\)](#).

Permissions

This view is available in the **master** database and in each user database. Read-only access to this view is available to all users with permission to connect to the database.

See Also

[sp_set_firewall_rule \(Azure SQL Database\)](#)

[sys.database_firewall_rules \(Azure SQL Database\)](#)

[sp_delete_database_firewall_rule \(Azure SQL Database\)](#)





[Configure a Windows Firewall for Database Engine Access](#)

[Configure a Firewall for FILESTREAM Access](#)

[Configure a Firewall for Report Server Access](#)

sys.database_service_objectives (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the edition (service tier), service objective (pricing tier) and elastic pool name, if any, for an Azure SQL database or an Azure SQL Data Warehouse. If logged on to the master database in an Azure SQL Database server, returns information on all databases. For Azure SQL Data Warehouse, you must be connected to the master database.

For information on pricing , see [SQL Database options and performance: SQL Database Pricing](#) and [SQL Data Warehouse Pricing](#).

To change the service settings, see [ALTER DATABASE \(Azure SQL Database\)](#) and [ALTER DATABASE \(Azure SQL Data Warehouse\)](#).

The sys.database_service_objectives view contains the following columns.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	The ID of the database, unique within an instance of Azure SQL Database server. Joinable with sys.databases (Transact-SQL) .
edition	sysname	The service tier for the database or data warehouse: Basic , Standard , Premium or Data Warehouse .
service_objective	sysname	<p>The pricing tier of the database. If the database is in an elastic pool, returns ElasticPool.</p> <p>On the Basic tier, returns Basic.</p> <p>Single database in a standard service tier returns one of the following: S0, S1, S2, or S3.</p> <p>Single database in a premium tier returns of the following: P1, P2, P4, P6/P3, or P11.</p> <p>SQL Data Warehouse returns DW100 through DW2000.</p>
elastic_pool_name	sysname	The name of the elastic pool that the database belongs to. Returns NULL if the database is a single database or a data warehouse.

Permissions

Requires **dbManager** permission on the master database. At the database level, the user must be the creator or owner.





Examples

This example can be run on the master database or on user databases. The query returns the name, service, and performance tier information of the database(s).

```
SELECT  d.name,  
        slo.*  
FROM sys.databases d  
JOIN sys.database_service_objectives slo  
ON d.database_id = slo.database_id;
```

sys.database_usage (Azure SQL Database)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Note: This applies only to Azure SQL Database V11.

Lists the number, type, and duration of databases on the SQL Database server.

The **sys.database_usage** view contains the following columns.

COLUMN NAME	DESCRIPTION
time	The date when the usage events occurred.
sku	The type of service tier for the database: Web, Business, Basic, Standard, Premium
quantity	The maximum number of databases of an SKU type that existed during that day.

Permissions

Read-only access to this view is available to all users with permissions to connect to the **master** database.

Remarks

The **sys.database_usage** view returns one row for each day of your subscription.





See Also

[SQL Database Pricing Details](#)

[Accounts and Billing in Windows Azure SQL Database](#)

sys.elastic_pool_resource_stats (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns resource usage statistics for all the elastic database pools in a logical server. For each elastic database pool, there is one row for each 15 second reporting window (four rows per minute). This includes CPU, IO, Log, storage consumption and concurrent request/session utilization by all databases in the pool. This data is retained for 14 days.

Applies to: SQL Database V12.|

COLUMN NAME	DATA TYPE	DESCRIPTION
start_time	datetime2	UTC time indicating the start of the 15 second reporting interval.
end_time	datetime2	UTC time indicating the end of the 15 second reporting interval.
elastic_pool_name	nvarchar(128)	Name of the elastic database pool.
avg_cpu_percent	decimal(5,2)	Average compute utilization in percentage of the limit of the pool.
avg_data_io_percent	decimal(5,2)	Average I/O utilization in percentage based on the limit of the pool.
avg_log_write_percent	decimal(5,2)	Average write resource utilization in percentage of the limit of the pool.
avg_storage_percent	decimal(5,2)	Average storage utilization in percentage of the storage limit of the pool.
max_worker_percent	decimal(5,2)	Maximum concurrent workers (requests) in percentage based on the limit of the pool.
max_session_percent	decimal(5,2)	Maximum concurrent sessions in percentage based on the limit of the pool.
elastic_pool_dtu_limit	int	Current max elastic pool DTU setting for this elastic pool during this interval.
elastic_pool_storage_limit_mb	bigint	Current max elastic pool storage limit setting for this elastic pool in megabytes during this interval.

Remarks

This view exists in the master database of the logical server. You must be connected to the master database to query **sys.elastic_pool_resource_stats**.

Permissions

Requires membership in the **dbmanager** role.

Examples

The following example returns resource utilization data ordered by the most recent time for all the elastic database pools in the current logical server.

```
SELECT * FROM sys.elastic_pool_resource_stats
ORDER BY end_time DESC;
```

The following example calculates the average DTU percentage consumption for a given pool.

```
SELECT start_time, end_time,
       (SELECT Max(v)
        FROM (VALUES (avg_cpu_percent), (avg_data_io_percent), (avg_log_write_percent)) AS value(v)) AS
[avg_DTU_percent]
FROM sys.elastic_pool_resource_stats
WHERE elastic_pool_name = '<your pool name>'
ORDER BY end_time DESC;
```

See Also

[Tame explosive growth with elastic databases](#)





[Create and manage a SQL Database elastic database pool \(preview\)](#)

[sys.resource_stats \(Azure SQL Database\)](#)

[sys.dm_db_resource_stats \(Azure SQL Database\)](#)

sys.event_log (Azure SQL Database)

11/21/2017 • 8 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns successful Azure SQL Database database connections, connection failures, and deadlocks. You can use this information to track or troubleshoot your database activity with SQL Database.

Caution

For installations having a large number of databases or high numbers of logins, activity in sys.event_log can cause limitations in performance, high CPU usage, and possibly result in login failures. Queries of sys.event_log can contribute to the problem. Microsoft is working to resolve this issue. In the meantime, to reduce the impact of this issue, limit queries of sys.event_log. Users of the NewRelic SQL Server plugin should visit [Microsoft Azure SQL Database plugin tuning & performance tweaks](#) for additional configuration information.

The `sys.event_log` view contains the following columns.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_name	sysname	Name of the database. If the connection fails and the user did not specify a database name, then this column is blank.
start_time	datetime2	UTC date and time of the start of the aggregation interval. For aggregated events, the time is always a multiple of 5 minutes. For example: '2011-09-28 16:00:00' '2011-09-28 16:05:00' '2011-09-28 16:10:00'
end_time	datetime2	UTC date and time of the end of the aggregation interval. For aggregated events, End_time is always exactly 5 minutes later than the corresponding start_time in the same row. For events that are not aggregated, start_time and end_time equal the actual UTC date and time of the event.
event_category	nvarchar(64)	The high-level component that generated this event. See Event Types for a list of possible values.
event_type	nvarchar(64)	The type of event. See Event Types for a list of possible values.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_subtype	int	The subtype of the occurring event. See Event Types for a list of possible values.
event_subtype_desc	nvarchar(64)	The description of the event subtype. See Event Types for a list of possible values.
severity	int	The severity of the error. Possible values are: 0 = Information 1 = Warning 2 = Error
event_count	int	The number of times that this event occurred for the specified database within the time interval specified (start_time and end_time).
description	nvarchar(max)	A detailed description of the event. See Event Types for a list of possible values.
additional_data	XML	<i>Note: This value is always NULL for Azure SQL Database V12. See examples section for how to retrieve deadlock events for V12.</i> For Deadlock events, this column contains the deadlock graph. This column is NULL for other event types.

Event Types

The events recorded by each row in this view are identified by a category (**event_category**), event type (**event_type**), and a subtype (**event_subtype**). The following table lists the types of events that are collected in this view.

For events in the **connectivity** category, summary information is available in the sys.database_connection_stats view.

NOTE

This view does not include all possible SQL Database database events that can occur, only those listed here. Additional categories, event types, and subtypes may be added in future releases of SQL Database.

EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	DESCRIPTION
----------------	------------	---------------	--------------------	----------	-------------

EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	DESCRIPTION
connectivity	connection_successful	0	connection_successful	0	Connected successfully to database.
connectivity	connection_failed	0	invalid_login_name	2	Login name is not valid in this version of SQL Server.
connectivity	connection_failed	1	windows_auth_not_supported	2	Windows logins are not supported in this version of SQL Server.
connectivity	connection_failed	2	attach_db_not_supported	2	User requested to attach a database file which is not supported.
connectivity	connection_failed	3	change_password_not_supported	2	User requested to change the password of the user logging in which is not supported.
connectivity	connection_failed	4	login_failed_for_user	2	Login failed for user.
connectivity	connection_failed	5	login_disabled	2	The login was disabled.
connectivity	connection_failed	6	failed_to_open_db	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Database could not be opened. May be caused because database does not exist or lack of authentication to open the database.</p>
connectivity	connection_failed	7	blocked_by_firewall	2	Client IP address is not allowed to access the server.

EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	DESCRIPTION
connectivity	connection_failed	8	client_close	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Client may have timed out when establishing connection. Try increasing the connection timeout.</p>
connectivity	connection_failed	9	reconfiguration	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Connection failed because the database was going through a reconfiguration at the time.</p>
connectivity	connection_terminated	0	idle_connection_timeout	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Connection has been idle for longer than system defined threshold.</p>
connectivity	connection_terminated	1	reconfiguration	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>The session has been terminated due to a database reconfiguration.</p>
connectivity	throttling	<reason code>	reason_code	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Request is throttled. Throttling reason code: <reason code>. For more information, see Engine Throttling.</p>

EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	DESCRIPTION
connectivity	throttling_long_transaction	40549	long_transaction	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>Session is terminated because you have a long-running transaction. Try shortening your transaction. For more information, see Resource Limits.</p>
connectivity	throttling_long_transaction	40550	excessive_lock_usage	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>The session has been terminated because it has acquired too many locks. Try reading or modifying fewer rows in a single transaction. For more information, see Resource Limits.</p>
connectivity	throttling_long_transaction	40551	excessive_temp_db_usage	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>The session has been terminated because of excessive TEMPDB usage. Try modifying your query to reduce the temporary table space usage. For more information, see Resource Limits.</p>

EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	DESCRIPTION
connectivity	throttling_long_transaction	40552	excessive_log_space_usage	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>The session has been terminated because of excessive transaction log space usage. Try modifying fewer rows in a single transaction. For more information, see Resource Limits.</p>
connectivity	throttling_long_transaction	40553	excessive_memory_usage	2	<p><i>Note: Applies only to Azure SQL Database V11.</i></p> <p>The session has been terminated because of excessive memory usage. Try modifying your query to process fewer rows. For more information, see Resource Limits.</p>
engine	deadlock	0	deadlock	2	Deadlock occurred.

Permissions

Users with permission to access the **master** database have read-only access to this view.

Remarks

Event Aggregation

Event information for this view is collected and aggregated within 5-minute intervals. The **event_count** column represents the number of times a particular **event_type** and **event_subtype** occurred for a specific database within a given time interval.

NOTE

Some events, such as deadlocks, are not aggregated. For these events, **event_count** will be 1 and **start_time** and **end_time** will equal the actual UTC date and time when the event occurred.

For example, if a user fails to connect to database Database1, because of an invalid login name, seven times

between 11:00 and 11:05 on 2/5/2012 (UTC), this information is available in a single row in this view:

DATABASE_NAME	START_TIME	END_TIME	EVENT_CATEGORY	EVENT_TYPE	EVENT_SUBTYPE	EVENT_SUBTYPE_DESC	SEVERITY	EVENT_COUNT	DESCRIPTION	ADDITIONAL_DATA
Database1	2012-02-05 11:00:00	2012-02-05 11:05:00	connectivity	connection	failed	login_fail	2_for_user	7	Login failed for user.	NULL

Interval start_time and end_time

An event is included in an aggregation interval when the event occurs *on* or *after* **start_time** and *before* **end_time** for that interval. For example, an event occurring exactly at `2012-10-30 19:25:00.0000000` would be included only in the second interval shown below:

start_time	end_time
2012-10-30 19:20:00.0000000	2012-10-30 19:25:00.0000000
2012-10-30 19:25:00.0000000	2012-10-30 19:30:00.0000000

Data Updates

Data in this view is accumulated over time. Typically, the data is accumulated within an hour of the start of the aggregation interval, but it may take up to a maximum of 24 hours for all the data to appear in the view. During that time, the information within a single row may be updated periodically.

Data Retention

The data in this view is retained for a maximum of 30 days, or possibly less depending on the number of databases in the logical server and the number of unique events each database generates. To retain this information for a longer period, copy the data to a separate database. After you make an initial copy of the view, the rows in the view may be updated as data is accumulated. To keep your copy of the data up-to-date, periodically do a table scan of the rows to look for an increase in the event count of existing rows and to identify new rows (you can identify unique rows by using the start and end times), then update your copy of the data with these changes.

Errors Not Included

This view may not include all connection and error information:

- This view does not include all SQL Database database errors that could occur, only those specified in [Event Types](#) in this topic.
- If there is a machine failure within the SQL Database datacenter, a small amount of data for your logical server may be missing from the event table.
- If an IP address has been blocked through DoSGuard, connection attempt events from that IP address cannot be collected and will not appear in this view.

Examples

Simple examples

The following query returns all events that occurred between noon on 9/25/2011 and noon on 9/28/2011 (UTC). By default, query results are sorted by **start_time** (ascending order).

```
SELECT * FROM sys.event_log
WHERE start_time >= '2011-09-25:12:00:00'
      AND end_time <= '2011-09-28 12:00:00';
```

The following query returns all deadlock events for database Database1 (applies only to Azure SQL Database V11).

```
SELECT * FROM sys.event_log
WHERE event_type = 'deadlock'
      AND database_name = 'Database1';
```

The following query returns all deadlock events for database Database1 (applies only to Azure SQL Database V12).

```
WITH CTE AS (
    SELECT CAST(event_data AS XML) AS [target_data_XML]
    FROM sys.fn_xe_telemetry_blob_target_read_file('dl', null, null, null)
)
SELECT target_data_XML.value('/event/@timestamp')[1], 'DateTime2') AS Timestamp,
target_data_XML.query('/event/data[@name=' 'xml_report']/value/deadlock') AS deadlock_xml,
target_data_XML.query('/event/data[@name=' 'database_name']/value').value('/value')[1], 'nvarchar(100)') AS
db_name
FROM CTE
```

The following query returns hard throttling on SQL Worker Threads events that occurred between 10:00 and 11:00 on 9/25/2011 (UTC).

```
SELECT * FROM sys.event_log
WHERE event_type = 'throttling'
      AND event_subtype = 4194307
      AND start_time >= '2011-09-25 10:00:00'
      AND end_time <= '2011-09-25 11:00:00';
```

DB-Scoped Extended Event

Use the following sample code to set up the db-scoped Extended Event (XEvent) session:

```
IF EXISTS
    (SELECT * from sys.database_event_sessions
     WHERE name = 'azure_monitor_deadlock_session')
BEGIN
    ALTER EVENT SESSION azure_monitor_deadlock_session
    ON DATABASE
    DROP TARGET package0.ring_buffer;

    DROP EVENT SESSION azure_monitor_deadlock_session
    ON DATABASE;
END

CREATE EVENT SESSION azure_monitor_deadlock_session
ON DATABASE
ADD EVENT sqlserver.database_xml_deadlock_report
ADD TARGET package0.ring_buffer
(
    SET max_memory = 2048, max_events_limit = 10
)
WITH (STARTUP_STATE = ON,
      EVENT_RETENTION_MODE = ALLOW_SINGLE_EVENT_LOSS);

ALTER EVENT SESSION azure_monitor_deadlock_session
ON DATABASE
STATE = START;
```

Check for Deadlock

Use the following query to check if there is a deadlock.

```

WITH CTE AS (
    SELECT CAST(xet.target_data AS XML) AS [target_data_XML]
    FROM      sys.dm_xe_database_session_targets AS xet
    INNER JOIN sys.dm_xe_database_sessions      AS xe
    ON (xe.address = xet.event_session_address)
    WHERE xe.name = 'azure_monitor_deadlock_session'
)
, CTE2 AS (
    SELECT
        T2.EventData.query('.').value(
            '(/event/@timestamp)[1]', 'DateTime2') AS Timestamp,
        T2.EventData.query('.').query(
            '(/event/data/value/deadlock)[1]')      AS deadlock_xml
    FROM CTE
        CROSS Apply [target_data_XML].nodes(
            '/RingBufferTarget/event') AS T2(EventData)
)
SELECT * FROM CTE2;





```

See Also

[Extended events in Azure SQL Database](#)

sys.firewall_rules (Azure SQL Database)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about the server-level firewall settings associated with your Microsoft Azure SQL Database.

The `sys.firewall_rules` view contains the following columns:

COLUMN NAME	DATA TYPE	DESCRIPTION
id	INT	The identifier of the server-level firewall setting.
name	NVARCHAR(128)	The name you chose to describe and distinguish the server-level firewall setting.
start_ip_address	VARCHAR(50)	The lowest IP address in the range of the server-level firewall setting. IP addresses equal to or greater than this can attempt to connect to the SQL Database server. The lowest possible IP address is <code>0.0.0.0</code> .
end_ip_address	VARCHAR(50)	<p>The highest IP address in the range of the server-level firewall setting. IP addresses equal to or less than this can attempt to connect to the SQL Database server. The highest possible IP address is <code>255.255.255.255</code>.</p> <p>Note: Windows Azure connection attempts are allowed when both this field and the start_ip_address field equals <code>0.0.0.0</code>.</p>
create_date	DATETIME	<p>UTC date and time when the server-level firewall setting was created.</p> <p>Note: UTC is an acronym for Coordinated Universal Time.</p>
modify_date	DATETIME	UTC date and time when the server-level firewall setting was last modified.

Remarks

To remove a database firewall rule, use [sp_delete_firewall_rule \(Azure SQL Database\)](#). To set a firewall rule for a single database, see [sys.database_firewall_rules \(Azure SQL Database\)](#). To return information about existing firewall rules, query `sys.firewall_rules` (Azure SQL Database).

Permissions

Read-only access to this view is available to all users with permission to connect to the **master** database.

See Also

[sys.database_firewall_rules \(Azure SQL Database\)](#)

[sp_set_firewall_rule \(Azure SQL Database\)](#)

[Configure a Windows Firewall for Database Engine Access](#)





[Configure a Firewall for FILESTREAM Access](#)

[Configure a Firewall for Report Server Access](#)

[How to: Configure Firewall Settings \(Azure SQL Database\)](#)

sys.resource_stats (Azure SQL Database)

11/16/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns CPU usage and storage data for an Azure SQL Database. The data is collected and aggregated within five-minute intervals. For each user database, there is one row for every five-minute reporting window in which there is change in resource consumption. This includes CPU usage, storage size change or database SKU modification. Idle databases with no changes may not have rows for every five minute interval. Historical data is retained for approximately 14 days.

The **sys.resource_stats** view has different definitions depending on the version of the Azure SQL Database Server that the database is associated with. Consider these differences and any modifications your application requires when upgrading to a new server version.

The following table describes the columns available in a v12 server:

COLUMNS (V12 SERVER)	DATA TYPE	DESCRIPTION
start_time	datetime	UTC time indicating the start of the 5 minute reporting interval.
end_time	datetime	UTC time indicating the end of the 5 minute reporting interval.
database_name	varchar	Name of the user database.
sku	varchar	Service Tier of the database. The following are the possible values: Basic Standard Premium
storage_in_megabytes	float	Maximum storage size in megabytes for the time period, including database data, indexes, stored procedures and metadata.
avg_cpu_percent	numeric	Average compute utilization in percentage of the limit of the service tier.
avg_data_io_percent	numeric	Average I/O utilization in percentage based on the limit of the service tier.
avg_log_write_percent	numeric	Average write resource utilization in percentage of the limit of the service tier.

COLUMNS (V12 SERVER)	DATA TYPE	DESCRIPTION
max_worker_percent	decimal(5,2)	<p>Maximum concurrent workers (requests) in percentage based on the limit of the database's service tier.</p> <p>Maximum is currently calculated for the 5 minute interval based on the 15 second samples of concurrent worker counts.</p>
max_session_percent	decimal(5,2)	<p>Maximum concurrent sessions in percentage based on the limit of the database's service tier.</p> <p>Maximum is currently calculated for the 5 minute interval based on the 15 second samples of concurrent session counts.</p>
dtu_limit	int	Current max database DTU setting for this database during this interval.

TIP

For more context about these limits and service tiers, see the topics [Service Tiers](#) and [Service tier capabilities and limits](#).

The following table describes the columns available in a v11 server:

COLUMNS (V11 SERVER)	DATA TYPE	DESCRIPTION
start_time	datetime	UTC time indicating the start of the 5 minute reporting interval.
end_time	datetime	UTC time indicating the end of the 5 minute reporting interval.
database_name	nvarchar	Name of the database.
sku	nvarchar	<p>Service Tier of the database. The following are the possible values:</p> <p>Web</p> <p>Business</p> <p>Basic</p> <p>Standard</p> <p>Premium</p>

COLUMNS (V11 SERVER)	DATA TYPE	DESCRIPTION
usage_in_seconds	int	<p><i>Note: This column is deprecated for v11 and its value is always set to 0.</i></p> <p>CPU time used since the last measurement was taken.</p> <p>For CPU measurement, we recommend that you use the avg_cpu_cores_used column rather than this column.</p>
storage_in_megabytes	decimal	Maximum storage size in megabytes for the time period, including database data, indexes, stored procedures and metadata.
avg_cpu_cores_used	decimal	<p><i>Note: This column is deprecated for v11 and its value is always set to 0.</i></p> <p>Average CPU cores used in this interval.</p>
avg_physical_read_iops	decimal	<p><i>Note: This column is deprecated for v11 and its value is always set to 0.</i></p> <p>Average read IOPS in this interval.</p>
avg_physical_write_iops	decimal	<p><i>Note: This column is deprecated for v11 and its value is always set to 0.</i></p> <p>Average write IOPS in this interval.</p>
active_memory_used_kb	bigint	<p><i>Note: This column is deprecated for v11 and its value is always set to 0.</i></p> <p>Count of active memory being used at the end of this interval.</p>
active_session_count	int	Count of active sessions at the end of this interval.
active_worker_count	int	Count of active workers at the end of this interval.
avg_cpu_percent	decimal	Average compute utilization in percentage of the limit of the service tier.
avg_physical_data_read_percent	decimal	Average I/O utilization in percentage based on the limit of the service tier.
avg_log_write_percent	decimal	Average write resource utilization in percentage of the limit of the service tier.

Permissions

This view is available to all user roles with permissions to connect to the virtual **master** database.

Remarks

The data returned by **sys.resource_stats** is expressed as a percentage of the maximum allowed DTU limits for the service tier/performance level that you are running for Basic, Standard, and Premium databases. For Web and Business tiers, these numbers indicate the percentages in terms of the Standard S2 performance tier. For example, when executing against a Web database, if avg_cpu_percent returns 70%, that indicates 70% of the S2 tier limit. In addition, for Web and Business tiers, the percentages may reflect a number in excess of 100%, which is also based on the S2 tier limit.

When a database is a member of an elastic pool, resource statistics presented as percent values, are expressed as the percent of the max DTU limit for the databases as set in the elastic pool configuration.

For a more granular view of this data, use **sys.dm_db_resource_stats** dynamic management view in a user database. This view captures data every 15 seconds and maintains historical data for 1 hour. For more information, see [sys.dm_db_resource_stats \(Azure SQL Database\)](#).

Examples

The following example returns all databases that are averaging at least 80% of compute utilization over the last one week.

```
DECLARE @s datetime;
DECLARE @e datetime;
SET @s= DateAdd(d,-7,GetUTCDate());
SET @e= GETUTCDATE();
SELECT database_name, AVG(avg_cpu_percent) AS Average_Compute_Utilization
FROM sys.resource_stats
WHERE start_time BETWEEN @s AND @e
GROUP BY database_name
HAVING AVG(avg_cpu_percent) >= 80
```

The following example calculates the average DTU percentage consumption for a given database. (This query only works when run against a v11 server.)

```
SELECT start_time, end_time,
       (SELECT Max(v)
        FROM (VALUES (avg_cpu_percent), (avg_physical_data_read_percent), (avg_log_write_percent)) AS value(v)) AS
[avg_DTU_percent]
FROM sys.resource_stats
WHERE database_name = '<your db name>'
ORDER BY end_time DESC;
```





See Also

[Service Tiers](#)

[Service tier capabilities and limits](#)

sys.resource_usage (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

IMPORTANT

This feature is in a preview state. Do not take a dependency on the specific implementation of this feature because the feature might be changed or removed in a future release.

While in a preview state, the Azure SQL Database operations team might turn data collection off and on for this DMV:

- When turned on, the DMV returns current data as it is aggregated.
 - When turned off, the DMV returns historical data, which might be stale.

Provides hourly summary of resource usage data for user databases in the current server. Historical data is retained for 90 days.

For each user database, there is one row for every hour in continuous fashion. Even if the database was idle during that hour, there is one row, and the usage_in_seconds value for that database will be 0. Storage usage and SKU information is rolled up for the hour appropriately.





COLUMNS	DATA TYPE	DESCRIPTION
time	datetime	Time (UTC) in hour increments.
database_name	nvarchar	Name of user database.
sku	nvarchar	Name of the SKU. The following are the possible values: Web Business Basic Standard Premium
usage_in_seconds	int	Sum of CPU time used in the hour. Note: This column is deprecated for V11 and does not apply to V12. Value is always set to 0.
storage_in_megabytes	decimal	Maximum storage size for the hour, including database data, indexes, stored procedures and metadata.

Permissions

This view is available to all user roles with permissions to connect to the virtual **master** database.

dbo.server_quotas (Azure SQL Database)

10/31/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  **Azure SQL Database**  Azure SQL Data Warehouse  Parallel Data Warehouse

IMPORTANT!! This applies to **Azure SQL DatabaseV11 only!**

This feature is in a preview state. Do not take a dependency on the specific implementation of this feature because the feature might be changed or removed in a future release.

Returns the database quota types available on the server.

COLUMN NAME	DATA TYPE	DESCRIPTION
quota_name	nvarchar	The type of quota for the server. The type Premium_database is equivalent to databases with a resource reservation.
quota_value	int	The number of quota type allowed in the server.

Permissions





This view is available to all user roles with permissions to connect to the virtual **master** database.

See Also

[Managing Premium Databases](#)

dbo.slo_assignment_history (Azure SQL Database)

10/31/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  **Azure SQL Database**  Azure SQL Data Warehouse  Parallel Data Warehouse

IMPORTANT

This applies only to Azure SQL DatabaseV11.

This feature is in a preview state. Do not take a dependency on the specific implementation of this feature because the feature might be changed or removed in a future release.

Returns the history of database SLO assignments in the server, including the following:

- The history of database SLO assignments in the server.
- Start and end time of each database SLO change request.
- Any SLO assignment errors which are recorded in the error_code and error_desc columns.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_name	sysname	Name of the database.
database_id	int	ID of the database.
create_date	datetimeoffset(7)	Date of database creation.
service_objective_name	sysname	Name of the Service Level Objective (SLO).
service_objective_id	uniqueidentifier	ID of the SLO.
operation_id	uniqueidentifier	ID of the operation.
operation_start_time	datetimeoffset(7)	Start time of the database SLO change request.
operation_end_time	datetimeoffset(7)	End time of the database SLO change request.
error_code	int	Error code of the database SLO change request.
error_desc	nvarchar	Description of the error in the database SLO change request.

Permissions

This view is available to all user roles with permissions to connect to the virtual **master** database.

Examples

The following example returns the history of database SLO assignments for a specified database.





```
SELECT *  
FROM dbo.slo_assignment_history  
WHERE database_name = '<DB NAME>'  
ORDER BY operation_start_time DESC;
```

See Also

[Managing Premium Databases](#)

dbo.slo_database_objectives (Azure SQL Database)

10/31/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  **Azure SQL Database**  Azure SQL Data Warehouse  Parallel Data Warehouse

IMPORTANT

This applies only to Azure SQL DatabaseV11.

For `[!INCLUDEEssSDSfull]` (on master) for the operation `ALTER DATABASE`.

Returns the assignment status of a Service Level Objective (SLO) in a SQL Database.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_name	sysname	Name of the database.
current_slo	sysname	Current SLO of the database.
target_slo	sysname	Target SLO of the database as specified in the SLO change request.
state_desc	nvarchar	Status of SLO change request: completed or pending.

Permissions

This view is available to all user roles with permissions to connect to the virtual **master** database.

Examples

```
SELECT
database_name=database_name.name
, current_slo=current_slo.name
, target_slo=target_slo.name
, state_desc=database_slo.state_desc
FROM slo_database_objectives AS database_slo
INNER JOIN slo_service_objectives AS current_slo ON database_slo.current_objective_id =
current_slo.objective_id
INNER JOIN slo_service_objectives AS target_slo ON database_slo.configured_objective_id =
target_slo.objective_id
INNER JOIN sys.databases AS database_name ON database_slo.database_id = database_name.database_id;
```





See Also

[Managing Premium Databases](#)

[sys.dm_operation_status \(Azure SQL Database\)](#)

dbo.slo_service_objectives (Azure SQL Database)

10/31/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

IMPORTANT

This feature is in a preview state, and has been deprecated in Azure SQL Database V12. Do not take a dependency on the specific implementation of this feature because the feature might be changed or removed in a future release.

Returns Service Level Objective (SLO) information on the current server.

||

| - |

|Applies to: Azure SQL Database V11.|

COLUMN NAME	DATA TYPE	DESCRIPTION
objective_id	uniqueidentifier	ID of the service level objective.
name	sysname	Name of the service level objective.
description	nvarchar	Description of the service level objective.
create_date	datetimeoffset(7)	creation date of the service level object on the server.
is_system	bit	1 = system service level objective
is_default	bit	1 = service level objective is the default SLO.
state	tinyint	1 = service level objective is enable. 2 = service level objective is disabled.
state_desc	nvarchar	Description of the service level objective.
metadata_version	decimal	Version of the service level objective.

Permissions





This view is available to all user roles with permissions to connect to the virtual **master** database.

See Also

[Managing Premium Databases](#)

CLR Assembly Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.assemblies	sys.assembly_references
sys.assembly_files	

See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.assemblies (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each assembly.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the assembly. Is unique within the database.
principal_id	int	ID of the principal that owns this assembly.
assembly_id	int	Assembly identification number. Is unique within a database.
clr_name	nvarchar(4000)	Canonical string that encodes the simple name, version number, culture, public key, and architecture of the assembly. This value uniquely identifies the assembly on the common language runtime (CLR) side.
permission_set	tinyint	Permission-set/security-level for assembly. 1 = Safe Access 2 = External Access 3 = Unsafe Access
permission_set_desc	nvarchar(60)	Description for permission-set/security-level for assembly. SAFE_ACCESS EXTERNAL_ACCESS UNSAFE_ACCESS
is_visible	bit	1 = Assembly is visible to register Transact-SQL entry points. 0 = Assembly is intended only for managed callers. That is, the assembly provides internal implementation for other assemblies in the database.
create_date	datetime	Date the assembly was created or registered.

COLUMN NAME	DATA TYPE	DESCRIPTION
modify_date	datetime	Date the assembly was modified.
is_user_defined	bit	<p>Indicates the source of the assembly.</p> <p>0 = System-defined assemblies (such as Microsoft.SqlServer.Types for the hierarchyid data type)</p> <p>1 = User-defined assemblies</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CLR Assembly Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[ASSEMBLYPROPERTY \(Transact-SQL\)](#)

sys.assembly_files (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each file that makes up an assembly.

COLUMN NAME	DATA TYPE	DESCRIPTION
assembly_id	int	ID of the assembly to which this file belongs.
name	nvarchar(260)	Name of the assembly file.
file_id	int	ID of the file. Is unique within an assembly. The file ID numbered 1 represents the assembly DLL.
content	varbinary(max)	Content of file.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CLR Assembly Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[ASSEMBLYPROPERTY \(Transact-SQL\)](#)

sys.assembly_references (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each pair of assemblies where one is directly referencing another.

COLUMN NAME	DATA TYPE	DESCRIPTION
assembly_id	int	ID of the assembly to which this reference belongs.
referenced_assembly_id	int	ID of the assembly being referenced.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[CLR Assembly Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[ASSEMBLYPROPERTY \(Transact-SQL\)](#)

Data Collector Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The data collector offers the following views for displaying information about the data collector configuration, such as collector type properties, collection sets, and collection set items, as well as execution statistics that are obtained when a collection set runs. These views, which are in the **msdb** database, also provide an abstraction layer for the underlying tables. This abstraction enhances security by preventing direct access to the tables, while permitting changes to the tables without affecting any associated applications.

syscollector_collection_items (Transact-SQL)	syscollector_collection_sets (Transact-SQL)
syscollector_collector_types (Transact-SQL)	syscollector_config_store (Transact-SQL)
syscollector_execution_log (Transact-SQL)	syscollector_execution_log_full (Transact-SQL)
syscollector_execution_stats (Transact-SQL)	

See Also

[Data Collection](#)

[Data Collector Stored Procedures \(Transact-SQL\)](#)

syscollector_collection_items (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about an item in a collection set.

COLUMN NAME	DATA TYPE	DESCRIPTION
collection_set_id	int	Identifies the collection set. Is not nullable.
collection_item_id	int	Identifies an item in the collection set. Is not nullable.
collector_type_uid	uniqueidentifier	The GUID used to identify the collector type. Is not nullable.
name	nvarchar(4000)	The name of the collection set. Is nullable.
frequency	int	The frequency that data is collected by a collection item. Is not nullable.
parameters	xml	Describes the parameterization for the collector type associated with the collection item. The XML schema for this collection item is validated with the XML Schema (XSD) stored in the parameter_schema for a particular collector type. Is nullable. For more information, see syscollector_collector_types (Transact-SQL) .

Permissions

Requires SELECT for **dc_operator**, **dc_proxy**.

See Also


[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

syscollector_collection_sets (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides information about a collection set, including schedule, collection mode, and its state.

COLUMN NAME	DATA TYPE	DESCRIPTION
collection_set_id	int	The local identifier for the collection set. Is not nullable.
collection_set_uid	uniqueidentifier	The globally unique identifier for the collection set. Is not nullable.
name	nvarchar(4000)	The name of the collection set. Is nullable.
target	nvarchar(max)	Identifies the target for the collection set. Is nullable.
is_system	bit	Turned on (1) or off (0) to indicate if the collection set was included with the data collector or if it was added later by the dc_admin. This could be a custom collection set developed in-house or by a third party. Is not nullable.
is_running	bit	Indicates whether or not the collection set is running. Is not nullable.
collection_mode	smallint	Specifies the collection mode for the collection set. Is not nullable. Collection mode is one of the following: 0 - Cached mode. Data collection and upload are on separate schedules. 1 - Non-cached mode. Data collection and upload are on the same schedule.
proxy_id	int	Identifies the proxy that is used to run the collection set job step. Is nullable.
schedule_uid	uniqueidentifier	Provides a pointer to the collection set schedule. Is nullable.
collection_job_id	uniqueidentifier	Identifies the collection job. Is nullable.
upload_job_id	uniqueidentifier	Identifies the collection upload job. Is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
logging_level	smallint	Specifies the logging level (0, 1 or 2). Is not nullable.
days_until_expiration	smallint	The number of days that the collected data is saved in the management data warehouse. Is not nullable.
description	nvarchar(4000)	Describes the collection set. Is nullable.
dump_on_any_error	bit	Turned on (1) or off (0) to indicate whether to create an SSIS dump file on any error. Is not nullable.
dump_on_codes	nvarchar(max)	Contains the list of SSIS error codes that are used to trigger the dump file. Is nullable.

Permissions

Requires SELECT for dc_operator, dc_proxy.

Remarks

The data collector API only allows you to change or delete the collection sets that you create. The collection sets that are provided with the system cannot be modified or deleted. However, you can still enable or disable a system collection set, and change its configuration.

See Also





[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

syscollector_collector_types (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides information about a collector type for a collection item.

COLUMN NAME	DATA TYPE	DESCRIPTION
collector_type_uid	uniqueidentifier	The GUID for a collection type. Is not nullable.
name	sysname	The name of the collection type. Is not nullable.
parameter_schema	xml	The XML schema that describes what the configuration for the specified collector type looks like. This XML schema is used to validate the actual XML configuration associated with a particular collection item instance. Is nullable.
parameter_formatter	xml	Determines the template to use to transform the XML for use in the collection set property page. Is nullable.
collection_package_id	uniqueidentifier	The GUID for a collection package. Is not nullable.
collection_package_path	nvarchar(4000)	Provides the path to the collection package. Is nullable.
collection_package_name	sysname	The name of the collection package. Is not nullable.
upload_package_id	uniqueidentifier	The GUID for the upload package. Is not nullable.
upload_package_path	nvarchar(4000)	Provides the path to the upload package. Is nullable.
upload_package_name	sysname	The name of the upload package. Is not nullable.
is_system	bit	Turned on (1) or off (0) to indicate if the collector type was shipped with the data collector or if it was added later by the dc_admin . This could be a custom type developed in-house or by a third party. Is not nullable.

Permissions

Requires SELECT for **dc_operator**, **dc_proxy**.

Change History

UPDATED CONTENT
Updated collection_type_uid column name to collector_type_uid .
Corrected the description for the parameter_schema column to indicate that the value is nullable.
Added the parameter_formatter column.
Corrected the data type for the collection_package_path column, and updated the description to indicate that the value is nullable.
Corrected the data type for the upload_package_path column, and updated the description to indicate that the value is nullable.

See Also




[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

syscollector_config_store (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns properties that apply to the entire data collector, as opposed to a collection set instance. Each row in this view describes a specific data collector property, such as the name of the management data warehouse, and the instance name where the management data warehouse is located.

COLUMN NAME	DATA TYPE	DESCRIPTION
parameter_name	nvarchar(128)	The name of the property. Is not nullable.
parameter_value	sql_variant	The actual value of the property. Is nullable.

Permissions

Requires SELECT permission on the view or membership in the dc_operator, dc_proxy, or dc_admin fixed database roles.

Remarks

The list of properties available is fixed and their values can only be changed using the appropriate stored procedure. The following table describes the properties that are exposed through this view.

PROPERTY NAME	DESCRIPTION
CacheDirectory	The name of the directory in the file system where the collector type packages store temporary information. NULL = the default temporary SQL Server directory is used.
CacheWindow	Indicates the data retention policy of the cache directory for failed data uploads. -1 = Retain the data from all upload failures. 0 = Do not retain any data from upload failures. n = Retain data from n previous upload failures, where $n \geq 1$. Use the sp_syscollector_set_cache_window stored procedure to change this value.

PROPERTY NAME	DESCRIPTION
CollectorEnabled	<p>Indicates the state of the data collector.</p> <p>0 = disabled</p> <p>1 = enabled</p> <p>Use either the <code>sp_syscollector_enable_collector</code> or <code>sp_syscollector_disable_collector</code> stored procedure to change this value.</p>
MDWDatabase	<p>The name of the management data warehouse. Use the <code>sp_syscollector_set_warehouse_database_name</code> stored procedure to change this value.</p>
MDWInstance	<p>The name of the SQL Server instance for the management data warehouse. Use the <code>sp_syscollector_set_warehouse_instance_name</code> stored procedure to change this value.</p>

Examples

The following example queries the `syscollector_config_store` view.

```
SELECT parameter_name, parameter_value
FROM msdb.dbo.syscollector_config_store;
```

See Also

[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

[sp_syscollector_enable_collector \(Transact-SQL\)](#)

[sp_syscollector_disable_collector \(Transact-SQL\)](#)



[sp_syscollector_set_warehouse_database_name \(Transact-SQL\)](#)

[sp_syscollector_set_warehouse_instance_name \(Transact-SQL\)](#)

[sp_syscollector_set_cache_window \(Transact-SQL\)](#)

syscollector_execution_log (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides information from the execution log for a collection set or package.

COLUMN NAME	DATA TYPE	DESCRIPTION
log_id	bigint	Identifies each collection set execution. Used to join this view with other detailed logs. Is not nullable.
parent_log_id	bigint	Identifies the parent package or collection set. Is not nullable. The IDs are chained in the parent-child relationship, which enables you to determine which package was started by which collection set. This view groups the log entries by their parent-child linkage and indents the names of the packages, so that the call chain is clearly visible.
collection_set_id	int	Identifies the collection set or package that this log entry represents. Is not nullable.
collection_item_id	int	Identifies a collection item. Is nullable.
start_time	datetime	The time that the collection set or package started. Is not nullable.
last_iteration_time	datetime	For continuously running packages, the last time that the package captured a snapshot. Is nullable.
finish_time	datetime	The time the run completed for finished packages and collection sets. Is nullable.
runtime_execution_mode	smallint	Indicates whether the collection set activity was collecting data or uploading data. Is nullable. Values are: 0 = Collection 1 = Upload

COLUMN NAME	DATA TYPE	DESCRIPTION
status	smallint	Indicates the current status of the collection set or package. Is not nullable. Values are: 0 = running 1 = finished 2 = failed
operator	nvarchar(128)	Identifies who started the collection set or package. Is not nullable.
package_id	uniqueidentifier	Identifies the collection set or package that generated this log. Is nullable.
package_name	nvarchar(4000)	The name of the package that generated this log. Is nullable.
package_execution_id	uniqueidentifier	Provides a link to the SSIS log table. Is nullable.
failure_message	nvarchar(2048)	If the collection set or package failed, the most recent error message for that component. Is nullable. To obtain more detailed error information, use the fn_syscollector_get_execution_details (Transact-SQL) function.

Permissions

Requires SELECT for dc_operator.

See Also


[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

syscollector_execution_log_full (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides information about a collection set or package when the execution log is full.

COLUMN NAME	DATA TYPE	DESCRIPTION
log_id	bigint	Identifies each collection set execution. Used to join this view with other detailed logs. Is nullable.
parent_log_id	bigint	Identifies the parent package or collection set. Is not nullable. The IDs are chained in the parent-child relationship, which enables you to determine which package was started by which collection set. This view groups the log entries by their parent-child linkage and indents the names of the packages so that the call chain is clearly visible.
name	nvarchar(4000)	The name of the collection set or package that this log entry represents. Is nullable.
status	smallint	Indicates the current status of the collection set or package. Is nullable. Values are: 0 = running 1 = finished 2 = failed
runtime_execution_mode	smallint	Indicates whether the collection set activity was collecting data or uploading data. Is nullable.
start_time	datetime	The time that the collection set or package started. Is nullable.
last_iteration_time	datetime	For continuously running packages, the last time that the package captured a snapshot. Is nullable.
finish_time	datetime	The time the run completed for finished packages and collection sets. Is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
duration	int	The time, in seconds, that the package or collection set has been running. Is nullable.
failure_message	nvarchar(2048)	If the collection set or package failed, the most recent error message for that component. Is nullable. To obtain more detailed error information, use the fn_syscollector_get_execution_details (Transact-SQL) function.
operator	nvarchar(128)	Identifies who started the collection set or package. Is nullable.
package_execution_id	uniqueidentifier	Provides a link to the SSIS log table. Is nullable.
collection_set_id	int	Provides a link to the data collection configuration table in msdb. Is nullable.

Permissions

Requires SELECT for **dc_operator**.

See Also


[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

syscollector_execution_stats (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides information about task execution for a collection set or package.

COLUMN NAME	DATA TYPE	DESCRIPTION
log_id	bigint	Identifies each collection set execution. Used to join this view with other detailed logs. Is not nullable.
task_name	nvarchar(128)	The name of the collection set or package task that this information is for. Is not nullable.
execution_row_count_in	int	Number of rows processed at the beginning of data flow. Is nullable.
execution_row_count_out	int	Number of rows processed at the end of data flow. Is nullable.
execution_row_count_errors	int	Number of rows that failed during the data flow. Is nullable.
execution_time_ms	int	The time, in milliseconds, required for the task to complete. Is nullable.
log_time	datetime	The time that this information was logged. Is not nullable.

Permissions

Requires SELECT permission for **dc_operator**.

See Also





[Data Collector Stored Procedures \(Transact-SQL\)](#)

[Data Collector Views \(Transact-SQL\)](#)

[Data Collection](#)

Data Spaces (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.data_spaces	sys.filegroups
sys.destination_data_spaces	sys.partition_schemes





See Also

[System Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.data_spaces (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each data space. This can be a filegroup, partition scheme, or FILESTREAM data filegroup.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of data space, unique within the database.
data_space_id	int	Data space ID number, unique within the database.
type	char(2)	Data space type: FG = Filegroup FD = FILESTREAM data filegroup FX = Memory-optimized tables filegroup Applies to: SQL Server 2014 through SQL Server 2017. PS = Partition scheme
type_desc	nvarchar(60)	Description of data space type: FILESTREAM_DATA_FILEGROUP MEMORY_OPTIMIZED_DATA_FILEGROUP Applies to: SQL Server 2014 through SQL Server 2017. PARTITION_SCHEME ROWS_FILEGROUP
is_default	bit	1 = This is the default data space. The default data space is used when a filegroup or partition scheme is not specified in a CREATE TABLE or CREATE INDEX statement. 0 = This is not the default data space.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_system	bit	<p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>1 = Data space is used for full-text index fragments.</p> <p>0 = Data space is not used for full-text index fragments.</p>

Permissions

Requires membership in the public role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Data Spaces \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[sys.databases \(Transact-SQL\)](#)

[sys.destination_data_spaces \(Transact-SQL\)](#)

[sys.filegroups \(Transact-SQL\)](#)





[sys.partition_schemes \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.destination_data_spaces (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each data space destination of a partition scheme.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_scheme_id	int	ID of the partition-scheme that is partitioning to the data space.
destination_id	int	ID (1-based ordinal) of the destination-mapping, unique within the partition scheme.
data_space_id	int	ID of the data space to which data for this scheme's destination is being mapped.

Permissions





Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.filegroups (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each data space that is a filegroup.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	--	For a list of columns that this view inherits, see sys.data_spaces (Transact-SQL) .
filegroup_guid	uniqueidentifier	GUID for the filegroup. NULL = PRIMARY filegroup
log_filegroup_id	int	Identified for informational purposes only. Not supported. Future compatibility is not guaranteed. In SQL Server, the value is NULL.
is_read_only	bit	1 = Filegroup is read-only. 0 = Filegroup is read/write.
is_autogrow_all_files	bit	Applies to: SQL Server (SQL Server 2016 through current version). 1 = When a file in the filegroup meets the autogrow threshold, all files in the filegroup grow. 0 = When a file in the filegroup meets the autogrow threshold, only that file grows. This is the default.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Data Spaces \(Transact-SQL\)](#)

sys.partition_schemes (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each Data Space that is a partition scheme, with **type** = PS.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.data_spaces (Transact-SQL) .
function_id	int	ID of partition function used in the scheme.

For a list of columns that this view inherits, see [sys.data_spaces \(Transact-SQL\)](#)

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

Database Mail Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Database Mail has the following views for displaying Database Mail e-mail content, status of e-mails, and any messages received, such as errors logged by Database Mail. These views are in the **msdb** database.





sysmail_allitems (Transact-SQL)	sysmail_mailattachments (Transact-SQL)
sysmail_event_log (Transact-SQL)	sysmail_sentitems (Transact-SQL)
sysmail_faileditems (Transact-SQL)	sysmail_unsentitems (Transact-SQL)

See Also

- [Database Mail](#)
- [Database Mail Configuration Objects](#)
- [Database Mail Stored Procedures \(Transact-SQL\)](#)

sysmail_allitems (Transact-SQL)

11/27/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each message processed by Database Mail. Use this view when you want to see the status of all messages.

To see only messages with the failed status, use [sysmail_faileditems \(Transact-SQL\)](#). To see only unsent messages, use [sysmail_unsentitems \(Transact-SQL\)](#). To see only messages that were sent, use [sysmail_sentitems \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
mailitem_id	int	Identifier of the mail item in the mail queue.
profile_id	int	The identifier of the profile used to send the message.
recipients	varchar(max)	The e-mail addresses of the message recipients.
copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message.
blind_copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message but whose names do not appear in the message header.
subject	nvarchar(510)	The subject line of the message.
body	varchar(max)	The body of the message.
body_format	varchar(20)	The body format of the message. The possible values are TEXT and HTML.
importance	varchar(6)	The importance parameter of the message.
sensitivity	varchar(12)	The sensitivity parameter of the message.
file_attachments	varchar(max)	A semicolon-delimited list of file names attached to the e-mail message.
attachment_encoding	varchar(20)	The type of mail attachment.
query	varchar(max)	The query executed by the mail program.

COLUMN NAME	DATA TYPE	DESCRIPTION
execute_query_database	sysname	The database context within which the mail program executed the query.
attach_query_result_as_file	bit	When the value is 0, the query results were included in the body of the e-mail message, after the contents of the body. When the value is 1, the results were returned as an attachment.
query_result_header	bit	When the value is 1, query results contained column headers. When the value is 0, query results did not include column headers.
query_result_width	int	The query_result_width parameter of the message.
query_result_separator	char(1)	The character used to separate columns in the query output.
exclude_query_output	bit	The exclude_query_output parameter of the message. For more information, see sp_send_dbmail (Transact-SQL) .
append_query_error	bit	The append_query_error parameter of the message. 0 indicates that Database Mail should not send the e-mail message if there is an error in the query.
send_request_date	datetime	The date and time the message was placed on the mail queue.
send_request_user	sysname	The user who submitted the message. This is the user context of the database mail procedure, not the From: field of the message.
sent_account_id	int	The identifier of the Database Mail account used to send the message.
sent_status	varchar(8)	<p>The status of the mail. Possible values are:</p> <p>sent - The mail was sent.</p> <p>unsent - Database mail is still attempting to send the message.</p> <p>retrying - Database Mail failed to send the message but is attempting to send it again.</p> <p>failed - Database mail was unable to send the message.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
sent_date	datetime	The date and time that the message was sent.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row.

Remarks

Use the **sysmail_allitems** view to see the status of all messages processed by Database Mail. When troubleshooting Database Mail, this view may help you identify the nature of the problem, by showing you the attributes of the messages that were sent compared with the attributes of the messages that were not sent.





The system tables exposed by this view contain all messages and may cause the **msdb** database to grow. Delete old messages from the view periodically to reduce the size of the tables. For more information, see [Create a SQL Server Agent Job to Archive Database Mail Messages and Event Logs](#).

Permissions

Granted to **sysadmin** fixed server role and **DatabaseMailUserRole** database role. When executed by a member of the **sysadmin** fixed server role, this view shows all messages. All other users only see the messages that they submitted.

sysmail_event_log (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each Windows or SQL Server message returned by the Database Mail system. (Message in this context refers to a message such as an error message, not an e-mail message.) Configure the **Logging Level** parameter by using the **Configure System Parameters** dialog box of the Database Mail Configuration Wizard, or the [sysmail_configure_sp](#) stored procedure, to determine which messages are returned.

COLUMN NAME	DATA TYPE	DESCRIPTION
Log_id	int	Identifier of items in the log.
event_type	varchar(11)	The type of notice inserted in the log. Possible values are errors, warnings, informational messages, success messages, and additional internal messages.
log_date	datetime	The date and time the log entry is made.
description	nvarchar(max)	The text of the message being recorded.
process_id	int	The process id of the Database Mail external program. This typically changes each time the Database Mail external program starts.
mailitem_id	int	Identifier of the mail item in the mail queue. NULL if the message is not related to a specific e-mail item.
account_id	int	The account_id of the account related to the event. NULL if the message is not related to a specific account.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row. For e-mails, this is the user who sent the mail. For messages generated by the Database Mail external program, this is the user context of the program.

Remarks

When troubleshooting Database Mail, search the **sysmail_event_log** view for events related to e-mail failures. Some messages, such as the failure of the Database Mail external program, are not associated with specific e-mails. To search for errors related to specific e-mails, look up the **mailitem_id** of the failed e-mail in the

sysmail_faileditems view and then search the **sysmail_event_log** for messages related to that **mailitem_id**. When an error is returned from **sp_send_dbmail**, the e-mail is not submitted to the Database Mail system and the error is not displayed in this view.

When individual account delivery attempts fail, Database Mail holds the error messages during retry attempts until the mail item delivery either succeeds or fails. In case of ultimate success, all of the accumulated errors get logged as separate warnings including the **account_id**. This can cause warnings to appear, even though the e-mail was sent. In case of ultimate delivery failure, all previous warnings get logged as one error message without an **account_id**, since all accounts have failed.

Permissions

You must be a member of the **sysadmin** fixed server role or the **DatabaseMailUserRole** database role to access this view. Members of **DatabaseMailUserRole** who are not members of the **sysadmin** role, can only see the events for e-mails that they submit.





See Also

[sysmail_faileditems \(Transact-SQL\)](#)

[Database Mail External Program](#)

sysmail_faileditems (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each Database Mail message with the **failed** status. Use this view to determine which messages were not successfully sent.

To see all messages processed by Database Mail, use [sysmail_allitems \(Transact-SQL\)](#). To see only unsent messages, use [sysmail_unsentitems \(Transact-SQL\)](#). To see only messages that were sent, use [sysmail_sentitems \(Transact-SQL\)](#). To view e-mail attachments, use [sysmail_mailattachments \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
mailitem_id	int	Identifier of the mail item in the mail queue.
profile_id	int	The identifier of the profile used to submit the message.
recipients	varchar(max)	The e-mail addresses of the message recipients.
copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message.
blind_copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message but whose names do not appear in the message header.
subject	nvarchar(510)	The subject line of the message.
body	varchar(max)	The body of the message.
body_format	varchar(20)	The body format of the message. The possible values are TEXT and HTML.
importance	varchar(6)	The importance parameter of the message.
sensitivity	varchar(12)	The sensitivity parameter of the message.
file_attachments	varchar(max)	A semicolon-delimited list of file names attached to the e-mail message.
Attachment_encoding	varchar(20)	The type of mail attachment.
Query	varchar(max)	The query executed by the mail program.

COLUMN NAME	DATA TYPE	DESCRIPTION
execute_query_database	sysname	The database context within which the mail program executed the query.
attach_query_result_as_file	bit	When the value is 0, the query results were included in the body of the e-mail message, after the contents of the body. When the value is 1, the results were returned as an attachment.
query_result_header	bit	When the value is 1, query results contained column headers. When the value is 0, query results did not include column headers.
query_result_width	int	The query_result_width parameter of the message.
query_result_separator	char(1)	The character used to separate columns in the query output.
exclude_query_output	bit	The exclude_query_output parameter of the message. For more information, see sp_send_dbmail (Transact-SQL) .
append_query_error	bit	The append_query_error parameter of the message. 0 indicates that Database Mail should not send the e-mail message if there is an error in the query.
send_request_date	datetime	The date and time the message was placed on the mail queue.
send_request_user	sysname	The user who submitted the message. This is the user context of the database mail procedure, not the From: field of the message.
sent_account_id	int	The identifier of the Database Mail account used to send the message. Always NULL for this view.
sent_status	varchar(8)	The status of the mail. Always failed for this view.
sent_date	datetime	The date and time that the message was removed from the mail queue.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row.

Remarks





Use the **sysmail_faileditems** view to see which messages were not sent by Database Mail. When troubleshooting Database Mail, this view may help you identify the nature of the problem, by showing you the attributes of the messages that were not sent. To view the reason for the failure, see the entry for the failed message in the [sysmail_event_log \(Transact-SQL\)](#) view.

Permissions

Granted to **sysadmin** fixed server role and **databasemailuserrole** database role. When executed by a member of the **sysadmin** fixed server role, this view shows all failed messages. All other users only see the failed messages that they submitted.

sysmail_mailattachments (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each attachment submitted to Database Mail. Use this view when you want information about Database Mail attachments. To review all e-mails processed by Database Mail use [sysmail_allitems \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
attachment_id	int	Identifier of the attachment.
mailitem_id	int	Identifier of the mail item that contained the attachment.
filename	nvarchar(520)	The file name of the attachment. When attach_query_result is 1 and query_attachment_filename is NULL, Database Mail creates an arbitrary filename.
filesize	int	The size of the attachment in bytes.
attachment	varbinary(max)	The content of the attachment.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row.

Remarks

When troubleshooting Database Mail, use this view to see the properties of the attachments.

Attachments stored in the system tables can cause the **msdb** database to grow. Use **sysmail_delete_mailitems_sp** to delete mail items and their associated attachments. For more information, see [Create a SQL Server Agent Job to Archive Database Mail Messages and Event Logs](#).

Permissions

Granted to the **sysadmin** fixed server role and the **DatabaseMailUserRole** database role. When executed by a member of the **sysadmin** fixed server role, this view shows all attachments. All other users only see the attachments for messages that they submitted.

See Also

[sysmail_allitems \(Transact-SQL\)](#)

[sysmail_faileditems \(Transact-SQL\)](#)





[sysmail_sentitems \(Transact-SQL\)](#)

[sysmail_unsentitems \(Transact-SQL\)](#)

[sysmail_event_log \(Transact-SQL\)](#)

sysmail_sentitems (Transact-SQL)

11/16/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each message sent by Database Mail. Use **sysmail_sentitems** when you want to see which messages were successfully sent.

To see all messages processed by Database Mail, use [sysmail_allitems \(Transact-SQL\)](#). To see only messages with the failed status, use [sysmail_faileditems \(Transact-SQL\)](#). To see only unsent or retrying messages, use [sysmail_unsentitems \(Transact-SQL\)](#). To see e-mail attachments, use [sysmail_mailattachments \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
mailitem_id	int	Identifier of the mail item in the mail queue.
profile_id	int	The identifier of the profile used to send the message.
recipients	varchar(max)	The e-mail addresses of the message recipients.
copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message.
blind_copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message but whose names do not appear in the message header.
subject	nvarchar(510)	The subject line of the message.
body	varchar(max)	The body of the message.
body_format	varchar(20)	The body format of the message. The possible values are TEXT and HTML .
importance	varchar(6)	The importance parameter of the message.
sensitivity	varchar(12)	The sensitivity parameter of the message.
file_attachments	varchar(max)	A semicolon-delimited list of file names attached to the e-mail message.
attachment_encoding	varchar(20)	The type of mail attachment.
query	varchar(max)	The query executed by the mail program.

COLUMN NAME	DATA TYPE	DESCRIPTION
execute_query_database	sysname	The database context within which the mail program executed the query.
attach_query_result_as_file	bit	When the value is 0, the query results were included in the body of the e-mail message, after the contents of the body. When the value is 1, the results were returned as an attachment.
query_result_header	bit	When the value is 1, query results contained column headers. When the value is 0, query results did not include column headers.
query_result_width	int	The query_result_width parameter of the message.
query_result_separator	char(1)	The character used to separate columns in the query output.
exclude_query_output	bit	The exclude_query_output parameter of the message. For more information, see sp_send_dbmail (Transact-SQL) .
append_query_error	bit	The append_query_error parameter of the message. 0 indicates that Database Mail should not send the e-mail message if there is an error in the query.
send_request_date	datetime	The date and time the message was placed on the mail queue.
send_request_user	sysname	The user who sent the message. This is the user context of the database mail procedure, not the From: field of the message.
sent_account_id	int	The identifier of the Database Mail account used to send the message.
sent_status	varchar(8)	The status of the mail. Always sent for this view.
sent_date	datetime	The date and time that the message was sent.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row.

Remarks

When troubleshooting Database Mail, this view may help you identify the nature of the problem, by showing you

the attributes of the messages that were successfully sent. Database Mail marks messages as sent when they are successfully submitted to an SMTP mail server. Normally e-mail is received in a few minutes, but the e-mail can be delayed because of problems with the SMTP server. Database Mail marks the message as sent when it is accepted by the SMTP mail server. E-mail errors that occur on the SMTP mail server, such as an undeliverable recipient e-mail address, are not returned to Database Mail. Those e-mails are recorded as sent, even though they are not delivered. Troubleshoot that type of error on the SMTP server. Also, the SMTP mail server may send an undeliverable message notification to the reply e-mail address for a Database Mail account.

Permissions





Granted to **sysadmin** fixed server role and **databasemailuserrole** database role. When executed by a member of the **sysadmin** fixed server role, this view shows all sent messages. All other users only see the messages that they sent.

See Also

[Database Mail Messaging Objects](#)

sysmail_unsentitems (Transact-SQL)

11/27/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each Database Mail message with the **unsent** or **retrying** status. Messages with unsent or retrying status are still in the mail queue and may be sent at any time. Messages can have the **unsent** status for the following reasons:

- The message is new, and though the message has been placed on the mail queue, Database Mail is working on other messages and has not yet reached this message.
- The Database Mail external program is not running and no mail is being sent.

Messages can have the **retrying** status for the following reasons:

- Database Mail has attempted to send the mail, but the SMTP mail server could not be contacted. Database Mail will continue to attempt to send the message using other Database Mail accounts assigned to the profile that sent the message. If no accounts can send the mail, Database Mail will wait for the length of time configured for the **Account Retry Delay** parameter and then attempt to send the message again. Database Mail uses the **Account Retry Attempts** parameter to determine how many times to attempt to send the message. Messages retain **retrying** status as long as Database Mail is attempting to send the message.

Use this view when you want to see how many messages are waiting to be sent and how long they have been in the mail queue. Normally the number of **unsent** messages will be low. Conduct a benchmark test during normal operations to determine a reasonable number of messages in the message queue for your operations.

To see all messages processed by Database Mail, use [sysmail_allitems \(Transact-SQL\)](#). To see only messages with the failed status, use [sysmail_faileditems \(Transact-SQL\)](#). To see only messages that were sent, use [sysmail_sentitems \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
mailitem_id	int	Identifier of the mail item in the mail queue.
profile_id	int	The identifier of the profile used to submit the message.
recipients	varchar(max)	The e-mail addresses of the message recipients.
copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message.
blind_copy_recipients	varchar(max)	The e-mail addresses of those who receive copies of the message but whose names do not appear in the message header.

COLUMN NAME	DATA TYPE	DESCRIPTION
subject	nvarchar(510)	The subject line of the message.
body	varchar(max)	The body of the message.
body_format	varchar(20)	The body format of the message. The possible values are TEXT and HTML .
importance	varchar(6)	The importance parameter of the message.
sensitivity	varchar(12)	The sensitivity parameter of the message.
file_attachments	varchar(max)	A semicolon-delimited list of file names attached to the e-mail message.
attachment_encoding	varchar(20)	The type of mail attachment.
query	varchar(max)	The query executed by the mail program.
execute_query_database	sysname	The database context within which the mail program executed the query.
attach_query_result_as_file	bit	When the value is 0, the query results were included in the body of the e-mail message, after the contents of the body. When the value is 1, the results were returned as an attachment.
query_result_header	bit	When the value is 1, query results contained column headers. When the value is 0, query results did not include column headers.
query_result_width	int	The query_result_width parameter of the message.
query_result_separator	char(1)	The character used to separate columns in the query output.
exclude_query_output	bit	The exclude_query_output parameter of the message. For more information, see sp_send_dbmail (Transact-SQL) .
append_query_error	bit	The append_query_error parameter of the message. 0 indicates that Database Mail should not send the e-mail message if there is an error in the query.
send_request_date	datetime	The date and time the message was placed on the mail queue.

COLUMN NAME	DATA TYPE	DESCRIPTION
send_request_user	sysname	The user who submitted the message. This is the user context of the database mail procedure, not the From field of the message.
sent_account_id	int	The identifier of the Database Mail account used to send the message. Always NULL for this view.
sent_status	varchar(8)	Will be unsent if Database Mail has not attempted to send the mail. Will be retrying if Database Mail failed to send the message but is trying again.
sent_date	datetime	The date and time the Database Mail last attempted to send the mail. NULL if Database Mail has not attempted to send the message.
last_mod_date	datetime	The date and time of the last modification of the row.
last_mod_user	sysname	The user who last modified the row.

Remarks





When troubleshooting Database Mail, this view may help you identify the nature of the problem, by showing you the number of messages waiting to be sent, and the amount of time the messages have waited. If no messages are being sent, the Database Mail external program may not be running, or there may be a network problem preventing Database Mail from contacting the SMTP servers. If many of the unsent messages have the same **profile_id**, there may be a problem with the SMTP server. Consider adding additional accounts to the profile. If messages are being sent, but messages are spending too much time in the queue, SQL Server may need more resources to process the volume of messages you require.

Permissions

Granted to **sysadmin** fixed server role and **DatabaseMailUserRole** database role. When executed by a member of the **sysadmin** fixed server role, this view shows all **unsent** or **retrying** messages. All other users only see the **unsent** or **retrying** messages that they submitted.

Databases and Files Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.backup_devices (Transact-SQL)	sys.database_recovery_status (Transact-SQL)
sys.database_connection_stats (Azure SQL Database)	sys.database_scoped_configurations (Transact-SQL)
sys.databases (Transact-SQL)	sys.database_usage (Azure SQL Database)
sys.database_files (Transact-SQL)	sys.master_files (Transact-SQL)
sys.database_mirroring (Transact-SQL)	

See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.backup_devices (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each backup-device registered by using **sp_addumpdevice** or created in SQL Server Management Studio.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the backup device. Is unique in the set.
type	tinyint	Type of backup device: 2 = Disk 3 = Diskette (obsolete) 5 = Tape 6 = Pipe (obsolete) 7 = Virtual device (for optional use by third-party backup vendors) Typically, only disk (2) and tape (5) are used.
type_desc	nvarchar(60)	Description of backup device type: DISK DISKETTE (obsolete) TAPE PIPE (obsolete) VIRTUAL_DEVICE (for optional use by third party backup vendors) Typically, only DISK and TAPE are used.
physical_name	nvarchar(260)	Physical file name or path of the backup device.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[BACKUP \(Transact-SQL\)](#)

[Backup Devices \(SQL Server\)](#)





[sp_addumpdevice \(Transact-SQL\)](#)

[Databases and Files Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.databases (Transact-SQL)

11/21/2017 • 15 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per database in the instance of SQL Server.

If a database is not `ONLINE`, or `AUTO_CLOSE` is set to `ON` and the database is closed, the values of some columns may be `NULL`. If a database is `OFFLINE`, the corresponding row is not visible to low-privileged users. To see the corresponding row if the database is `OFFLINE`, a user must have at least the `ALTER ANY DATABASE` server-level permission, or the `CREATE DATABASE` permission in the `master` database.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of database, unique within an instance of SQL Server or within a Azure SQL Database server.
database_id	int	ID of the database, unique within an instance of SQL Server or within a Azure SQL Database server.
source_database_id	int	Non-NULL = ID of the source database of this database snapshot. NULL = Not a database snapshot.
owner_sid	varbinary(85)	SID (Security-Identifier) of the external owner of the database, as registered to the server. For information about who can own a database, see the ALTER AUTHORIZATION for databases section of ALTER AUTHORIZATION .
create_date	datetime	Date the database was created or renamed. For tempdb , this value changes every time the server restarts.

COLUMN NAME	DATA TYPE	DESCRIPTION
compatibility_level	tinyint	<p>Integer corresponding to the version of SQL Server for which behavior is compatible:</p> <p>Value : Applies to</p> <p>70 : SQL Server 2008 through SQL Server 2008 R2</p> <p>80 : SQL Server 2008 through SQL Server 2008 R2</p> <p>90 : SQL Server 2008 through SQL Server 2012</p> <p>100 : SQL Server 2008 through SQL Server 2017 and Azure SQL Database</p> <p>110 : SQL Server 2012 through SQL Server 2017 and Azure SQL Database</p> <p>120 : SQL Server 2014 through SQL Server 2017 and Azure SQL Database</p> <p>130 : SQL Server 2016 through SQL Server 2017</p>
collation_name	sysname	<p>Collation for the database. Acts as the default collation in the database.</p> <p>NULL = Database is not online or AUTO_CLOSE is set to ON and the database is closed.</p>
user_access	tinyint	<p>User-access setting:</p> <p>0 = MULTI_USER specified</p> <p>1 = SINGLE_USER specified</p> <p>2 = RESTRICTED_USER specified</p>
user_access_desc	nvarchar(60)	Description of user-access setting.
is_read_only	bit	<p>1 = Database is READ_ONLY</p> <p>0 = Database is READ_WRITE</p>
is_auto_close_on	bit	<p>1 = AUTO_CLOSE is ON</p> <p>0 = AUTO_CLOSE is OFF</p>
is_auto_shrink_on	bit	<p>1 = AUTO_SHRINK is ON</p> <p>0 = AUTO_SHRINK is OFF</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
state	tinyint	<p>Value Applies to</p> <p>0 = ONLINE</p> <p>1 = RESTORING</p> <p>2 = RECOVERING : SQL Server 2008 through SQL Server 2017</p> <p>3 = RECOVERY_PENDING : SQL Server 2008 through SQL Server 2017</p> <p>4 = SUSPECT</p> <p>5 = EMERGENCY : SQL Server 2008 through SQL Server 2017</p> <p>6 = OFFLINE : SQL Server 2008 through SQL Server 2017</p> <p>7 = COPYING : Azure SQL Database Active Geo-Replication</p> <p>10 = OFFLINE_SECONDARY : Azure SQL Database Active Geo-Replication</p> <p>Note: A database that has just come online is not necessarily ready to accept connections. To identify when a database can accept connections, query the collation_name column of sys.databases or the Collation property of DATABASEPROPERTYEX. The database can accept connections when the database collation returns a non-null value. For Always On databases, query the <code>database_state</code> or <code>database_state_desc</code> columns of sys.dm_hadr_database_replica_states.</p>
state_desc	nvarchar(60)	Description of the database state. See state.
is_in_standby	bit	Database is read-only for restore log.
is_cleanly_shutdown	bit	<p>1 = Database shut down cleanly; no recovery required on startup</p> <p>0 = Database did not shut down cleanly; recovery is required on startup</p>
is_supplemental_logging_enabled	bit	<p>1 = SUPPLEMENTAL_LOGGING is ON</p> <p>0 = SUPPLEMENTAL_LOGGING is OFF</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
snapshot_isolation_state	tinyint	<p>State of snapshot-isolation transactions being allowed, as set by the ALLOW_SNAPSHOT_ISOLATION option:</p> <p>0 = Snapshot isolation state is OFF (default). Snapshot isolation is disallowed.</p> <p>1 = Snapshot isolation state ON. Snapshot isolation is allowed.</p> <p>2 = Snapshot isolation state is in transition to OFF state. All transactions have their modifications versioned. Cannot start new transactions using snapshot isolation. The database remains in the transition to OFF state until all transactions that were active when ALTER DATABASE was run can be completed.</p> <p>3 = Snapshot isolation state is in transition to ON state. New transactions have their modifications versioned. Transactions cannot use snapshot isolation until the snapshot isolation state becomes 1 (ON). The database remains in the transition to ON state until all update transactions that were active when ALTER DATABASE was run can be completed.</p>
snapshot_isolation_state_desc	nvarchar(60)	Description of state of snapshot-isolation transactions being allowed, as set by the ALLOW_SNAPSHOT_ISOLATION option.
is_read_committed_snapshot_on	bit	<p>1 = READ_COMMITTED_SNAPSHOT option is ON. Read operations under the read-committed isolation level are based on snapshot scans and do not acquire locks.</p> <p>0 = READ_COMMITTED_SNAPSHOT option is OFF (default). Read operations under the read-committed isolation level use share locks.</p>
recovery_model	tinyint	<p>Recovery model selected:</p> <p>1 = FULL</p> <p>2 = BULK_LOGGED</p> <p>3 = SIMPLE</p>
recovery_model_desc	nvarchar(60)	Description of recovery model selected.
page_verify_option	tinyint	<p>Setting of PAGE_VERIFY option:</p> <p>0 = NONE</p> <p>1 = TORN_PAGE_DETECTION</p> <p>2 = CHECKSUM</p>
page_verify_option_desc	nvarchar(60)	Description of PAGE_VERIFY option setting.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_auto_create_stats_on	bit	1 = AUTO_CREATE_STATISTICS is ON 0 = AUTO_CREATE_STATISTICS is OFF
is_auto_create_stats_incremental_on	bit	Indicates the default setting for the incremental option of auto stats. 0 = auto create stats are non-incremental 1 = auto create stats are incremental if possible Applies to: SQL Server 2014 through SQL Server 2017.
is_auto_update_stats_on	bit	1 = AUTO_UPDATE_STATISTICS is ON 0 = AUTO_UPDATE_STATISTICS is OFF
is_auto_update_stats_async_on	bit	1 = AUTO_UPDATE_STATISTICS_ASYNC is ON 0 = AUTO_UPDATE_STATISTICS_ASYNC is OFF
is_ansi_null_default_on	bit	1 = ANSI_NULL_DEFAULT is ON 0 = ANSI_NULL_DEFAULT is OFF
is_ansi_nulls_on	bit	1 = ANSI_NULLS is ON 0 = ANSI_NULLS is OFF
is_ansi_padding_on	bit	1 = ANSI_PADDING is ON 0 = ANSI_PADDING is OFF
is_ansi_warnings_on	bit	1 = ANSI_WARNINGS is ON 0 = ANSI_WARNINGS is OFF
is_arithabort_on	bit	1 = ARITHABORT is ON 0 = ARITHABORT is OFF
is_concat_null_yields_null_on	bit	1 = CONCAT_NULL_YIELDS_NULL is ON 0 = CONCAT_NULL_YIELDS_NULL is OFF
is_numeric_roundabort_on	bit	1 = NUMERIC_ROUNDABORT is ON 0 = NUMERIC_ROUNDABORT is OFF
is_quoted_identifier_on	bit	1 = QUOTED_IDENTIFIER is ON 0 = QUOTED_IDENTIFIER is OFF
is_recursive_triggers_on	bit	1 = RECURSIVE_TRIGGERS is ON 0 = RECURSIVE_TRIGGERS is OFF
is_cursor_close_on_commit_on	bit	1 = CURSOR_CLOSE_ON_COMMIT is ON 0 = CURSOR_CLOSE_ON_COMMIT is OFF

COLUMN NAME	DATA TYPE	DESCRIPTION
is_local_cursor_default	bit	1 = CURSOR_DEFAULT is local 0 = CURSOR_DEFAULT is global
is_fulltext_enabled	bit	1 = Full-text is enabled for the database 0 = Full-text is disabled for the database
is_trustworthy_on	bit	1 = Database has been marked trustworthy 0 = Database has not been marked trustworthy
is_db_chaining_on	bit	1 = Cross-database ownership chaining is ON 0 = Cross-database ownership chaining is OFF
is_parameterization_forced	bit	1 = Parameterization is FORCED 0 = Parameterization is SIMPLE
is_master_key_encrypted_by_server	bit	1 = Database has an encrypted master key 0 = Database does not have an encrypted master key
is_query_store_on	bit	1 = The query store is enable for this database. Check sys.database_query_store_options to view the query store status. 0 = The query store is not enabled Applies to: SQL Server (SQL Server 2016 through current version).
is_published	bit	1 = Database is a publication database in a transactional or snapshot replication topology 0 = Is not a publication database
is_subscribed	bit	This column is not used. It will always return 0, regardless of the subscriber status of the database.
is_merge_published	bit	1 = Database is a publication database in a merge replication topology 0 = Is not a publication database in a merge replication topology
is_distributor	bit	1 = Database is the distribution database for a replication topology 0 = Is not the distribution database for a replication topology

COLUMN NAME	DATA TYPE	DESCRIPTION
is_sync_with_backup	bit	1 = Database is marked for replication synchronization with backup 0 = Is not marked for replication synchronization with backup
service_broker_guid	uniqueidentifier	Identifier of the service broker for this database. Used as the broker_instance of the target in the routing table.
is_broker_enabled	bit	1 = The broker in this database is currently sending and receiving messages. 0 = All sent messages will stay on the transmission queue and received messages will not be put on queues in this database. By default, restored or attached databases have the broker disabled. The exception to this is database mirroring where the broker is enabled after failover.

COLUMN NAME	DATA TYPE	DESCRIPTION
log_reuse_wait	tinyint	<p>Reuse of transaction log space is currently waiting on one of the following as of the last checkpoint. (For more detailed explanations of these values, see The Transaction Log.)</p> <p>0 = Nothing</p> <p>1 = Checkpoint (When a database uses a recovery model and has a memory-optimized data filegroup, you should expect to see the log_reuse_wait column indicate checkpoint or xtp_checkpoint.) Applies to SQL Server 2008 through SQL Server 2017</p> <p>2 = Log Backup Applies to SQL Server 2008 through SQL Server 2017</p> <p>3 = Active backup or restore Applies to SQL Server 2008 through SQL Server 2017</p> <p>4 = Active transaction Applies to SQL Server 2008 through SQL Server 2017</p> <p>5 = Database mirroring Applies to SQL Server 2008 through SQL Server 2017</p> <p>6 = Replication Applies to SQL Server 2008 through SQL Server 2017</p> <p>7 = Database snapshot creation Applies to SQL Server 2008 through SQL Server 2017</p> <p>8 = Log scan Applies to</p> <p>9 = An Always On Availability Groups secondary replica is applying transaction log records of this database to a corresponding secondary database. Applies to SQL Server 2012 through SQL Server 2017. In earlier versions of SQL Server, 9 = Other (Transient).</p> <p>10 = For internal use only Applies to SQL Server 2012 through SQL Server 2017</p> <p>11 = For internal use only Applies to SQL Server 2012 through SQL Server 2017</p> <p>12 = For internal use only Applies to SQL Server 2012 through SQL Server 2017</p> <p>13 = Oldest page Applies to SQL Server 2012 through SQL Server 2017</p> <p>14 = Other Applies to SQL Server 2012 through SQL Server 2017</p> <p>16 = XTP_CHECKPOINT (When a database uses a recovery model and has a memory-optimized data filegroup, you should expect to see the log_reuse_wait column indicate checkpoint or xtp_checkpoint.) Applies to SQL Server 2014 through SQL Server 2017</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
log_reuse_wait_desc	nvarchar(60)	Description of reuse of transaction log space is currently waiting on as of the last checkpoint.
is_date_correlation_on	bit	1 = DATE_CORRELATION_OPTIMIZATION is ON 0 = DATE_CORRELATION_OPTIMIZATION is OFF
is_cdc_enabled	bit	1 = Database is enabled for change data capture. For more information, see sys.sp_cdc_enable_db (Transact-SQL) .
is_encrypted	bit	Indicates whether the database is encrypted (reflects the state last set by using the ALTER DATABASE SET ENCRYPTION clause). Can be one of the following values: 1 = Encrypted 0 = Not Encrypted For more information about database encryption, see Transparent Data Encryption (TDE) . If the database is in the process of being decrypted, is_encrypted shows a value of 0. You can see the state of the encryption process by using the sys.dm_database_encryption_keys dynamic management view.
is_honor_broker_priority_on	bit	Indicates whether the database honors conversation priorities (reflects the state last set by using the ALTER DATABASE SET HONOR_BROKER_PRIORITY clause). Can be one of the following values: 1 = HONOR_BROKER_PRIORITY is ON 0 = HONOR_BROKER_PRIORITY is OFF
replica_id	uniqueidentifier	Unique identifier of the local Always On availability groups availability replica of the availability group, if any, in which the database is participating. NULL = database is not part of an availability replica of in availability group. Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database

COLUMN NAME	DATA TYPE	DESCRIPTION
group_database_id	uniqueidentifier	<p>Unique identifier of the database within an Always On availability group, if any, in which the database is participating. group_database_id is the same for this database on the primary replica and on every secondary replica on which the database has been joined to the availability group.</p> <p>NULL = database is not part of an availability replica in any availability group.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
resource_pool_id	int	<p>The id of the resource pool that is mapped to this database. This resource pool controls total memory available to memory-optimized tables in this database.</p> <p>Applies to: SQL Server 2014 through SQL Server 2017</p>
default_language_lcid	smallint	<p>Indicates the local id (lcid) of the default language of a contained database.</p> <p>Note Functions as the Configure the default language Server Configuration Option of sp_configure. This value is null for a non-contained database.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
default_language_name	nvarchar(128)	<p>Indicates the default language of a contained database.</p> <p>This value is null for a non-contained database.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
default_fulltext_language_lcid	int	<p>Indicates the local id (lcid) of the default fulltext language of the contained database.</p> <p>Note Functions as the default Configure the default full-text language Server Configuration Option of sp_configure. This value is null for a non-contained database.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
default_fulltext_language_name	nvarchar(128)	<p>Indicates the default fulltext language of the contained database.</p> <p>This value is null for a non-contained database.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
is_nested_triggers_on	bit	<p>Indicates whether or not nested triggers are allowed in the contained database.</p> <p>0 = nested triggers are not allowed 1 = nested triggers are allowed</p> <p>Note Functions as the Configure the nested triggers Server Configuration Option of sp_configure. This value is null for a non-contained database. See sys.configurations (Transact-SQL) for further information.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
is_transform_noise_words_on	bit	<p>Indicates whether or noise words should be transformed in the contained database.</p> <p>0 = noise words should not be transformed. 1 = noise words should be transformed.</p> <p>Note Functions as the transform noise words Server Configuration Option of sp_configure. This value is null for a non-contained database. See sys.configurations (Transact-SQL) for further information.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017</p>
two_digit_year_cutoff	smallint	<p>Indicates a value of a number between 1753 and 9999 to represent the cutoff year for interpreting two-digit years as four-digit years.</p> <p>Note Functions as the Configure the two digit year cutoff Server Configuration Option of sp_configure. This value is null for a non-contained database. See sys.configurations (Transact-SQL) for further information.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>
containment	tinyint not null	<p>Indicates the containment status of the database.</p> <p>0 = database containment is off.</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database 1 = database is in partial containment</p> <p>Applies to: SQL Server 2012 through SQL Server 2017</p>
containment_desc	nvarchar(60) not null	<p>Indicates the containment status of the database.</p> <p>NONE = legacy database (zero containment) PARTIAL = partially contained database</p> <p>Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
target_recovery_time_in_seconds	int	The estimated time to recover the database, in seconds. Nullable. Applies to: SQL Server 2012 through SQL Server 2017, Azure SQL Database
delayed_durability	int	The delayed durability setting: 0 = DISABLED 1 = ALLOWED 2 = FORCED For more information, see Control Transaction Durability . Applies to: SQL Server 2014 through SQL Server 2017, Azure SQL Database.
delayed_durability_desc	nvarchar(60)	The delayed durability setting: DISABLED ALLOWED FORCED Applies to: SQL Server 2014 through SQL Server 2017, Azure SQL Database. Applies to: SQL Server 2014 through SQL Server 2017.
is_memory_optimized_elevate_to_snapshot_on	bit	Memory-optimized tables are accessed using SNAPSHOT isolation when the session setting TRANSACTION ISOLATION LEVEL is set to a lower isolation level, READ COMMITTED or READ UNCOMMITTED. 1 = Minimum isolation level is SNAPSHOT. 0 = Isolation level is not elevated.
is_federation_member	bit	Indicates if the database is a member of a federation. Applies to: Azure SQL Database
is_remote_data_archive_enabled	bit	Indicates whether the database is stretched. 0 = The database is not Stretch-enabled. 1 = The database is Stretch-enabled. Applies to: SQL Server 2016 through SQL Server 2017 For more information, see Stretch Database .

COLUMN NAME	DATA TYPE	DESCRIPTION
is_mixed_page_allocation_on	bit	Indicates whether tables and indexes in the database can allocate initial pages from mixed extents. 0 = Tables and indexes in the database always allocate initial pages from uniform extents. 1 = Tables and indexes in the database can allocate initial pages from mixed extents. Applies to: SQL Server 2016 through SQL Server 2017 For more information, see the SET MIXED_PAGE_ALLOCATION option of ALTER DATABASE SET Options (Transact-SQL) .
is_temporal_retention_enabled	bit	Indicates whether temporal retention policy cleanup task is enabled. Applies to: Azure SQL Database
catalog_collation_type	int	The catalog collation setting: 0 = DATABASE_DEFAULT 2 = SQL_Latin_1_General_CP1_CI_AS Applies to: Azure SQL Database
catalog_collation_type_desc	nvarchar(60)	The catalog collation setting: DATABASE_DEFAULT SQL_Latin_1_General_CP1_CI_AS Applies to: Azure SQL Database

Permissions

If the caller of `sys.databases` is not the owner of the database and the database is not `master` or `tempdb`, the minimum permissions required to see the corresponding row are `ALTER ANY DATABASE` or the `VIEW ANY DATABASE` server-level permission, or `CREATE DATABASE` permission in the `master` database. The database to which the caller is connected can always be viewed in `sys.databases`.

IMPORTANT

By default, the public role has the `VIEW ANY DATABASE` permission, allowing all logins to see database information. To block a login from the ability to detect a database, `REVOKE` the `VIEW ANY DATABASE` permission from `public`, or `DENY` the `VIEW ANY DATABASE` permission for individual logins.

SQL Database Remarks

In SQL Database, this view is available in the `master` database and in user databases. In the `master` database, this view returns the information on the `master` database and all user databases on the server. In a user database, this view returns information only on the current database and the master database.

Use the `sys.databases` view in the `master` database of the SQL Database server where the new database is being created. After the database copy starts, you can query the `sys.databases` and the `sys.dm_database_copies` views from the `master` database of the destination server to retrieve more information about the copying progress.

Examples

A. Query the sys.databases view

The following example returns a few of the columns available in the `sys.databases` view.

```
SELECT name, user_access_desc, is_read_only, state_desc, recovery_model_desc
FROM sys.databases;
```

B. Check the copying status in SQL Database

The following example queries the `sys.databases` and `sys.dm_database_copies` views to return information about a database copy operation.

Applies to: Azure SQL Database|

```
-- Execute from the master database.
SELECT a.name, a.state_desc, b.start_date, b.modify_date, b.percentage_complete
FROM sys.databases AS a
INNER JOIN sys.dm_database_copies AS b ON a.database_id = b.database_id
WHERE a.state = 7;
```

C. Check the temporal retention policy status in SQL Database

The following example queries the `sys.databases` to return information whether temporal retention cleanup task is enabled. Be aware that after restore operation temporal retention is disabled by default. Use `ALTER DATABASE` to enable it explicitly.

Applies to: Azure SQL Database

```
-- Execute from the master database.
SELECT a.name, a.is_temporal_history_retention_enabled
FROM sys.databases AS a;
```

See Also

[ALTER DATABASE \(Transact-SQL\)](#)

[sys.database_mirroring_witnesses \(Transact-SQL\)](#)





[sys.database_recovery_status \(Transact-SQL\)](#)

[Databases and Files Catalog Views \(Transact-SQL\)](#)

[sys.dm_database_copies \(Azure SQL Database\)](#)

sys.database_files (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per file of a database as stored in the database itself. This is a per-database view.

COLUMN NAME	DATA TYPE	DESCRIPTION
file_id	int	ID of the file within database.
file_guid	uniqueidentifier	GUID for the file. NULL = Database was upgraded from an earlier version of SQL Server.
type	tinyint	File type: 0 = Rows (Includes files of full-text catalogs that are upgraded to or created in SQL Server 2017.) 1 = Log 2 = FILESTREAM 3 = Identified for informational purposes only. Not supported. Future compatibility is not guaranteed. 4 = Full-text (Full-text catalogs earlier than SQL Server 2017; full-text catalogs that are upgraded to or created in SQL Server 2017 will report a file type 0.)
type_desc	nvarchar(60)	Description of the file type: ROWS (Includes files of full-text catalogs that are upgraded to or created in SQL Server 2017.) LOG FILESTREAM FULLTEXT (Full-text catalogs earlier than SQL Server 2017.)
data_space_id	int	Value can be 0 or greater than 0. A value of 0 represents the database log file, and a value greater than 0 represents the ID of the filegroup where this data file is stored.
name	sysname	Logical name of the file in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
physical_name	nvarchar(260)	Operating-system file name. If the database is hosted by an AlwaysOn readable secondary replica , physical_name indicates the file location of the primary replica database. For the correct file location of a readable secondary database, query sys.sysaltfiles .
state	tinyint	File state: 0 = ONLINE 1 = RESTORING 2 = RECOVERING 3 = RECOVERY_PENDING 4 = SUSPECT 5 = Identified for informational purposes only. Not supported. Future compatibility is not guaranteed. 6 = OFFLINE 7 = DEFUNCT
state_desc	nvarchar(60)	Description of the file state: ONLINE RESTORING RECOVERING RECOVERY_PENDING SUSPECT OFFLINE DEFUNCT For more information, see File States .
size	int	Current size of the file, in 8-KB pages. 0 = Not applicable For a database snapshot, size reflects the maximum space that the snapshot can ever use for the file. For FILESTREAM filegroup containers, size reflects the current used size of the container.

COLUMN NAME	DATA TYPE	DESCRIPTION
max_size	int	<p>Maximum file size, in 8-KB pages:</p> <p>0 = No growth is allowed.</p> <p>-1 = File will grow until the disk is full.</p> <p>268435456 = Log file will grow to a maximum size of 2 TB.</p> <p>For FILESTREAM filegroup containers, max_size reflects the maximum size of the container.</p> <p>Note that databases that are upgraded with an unlimited log file size will report -1 for the maximum size of the log file.</p>
growth	int	<p>0 = File is fixed size and will not grow.</p> <p>>0 = File will grow automatically.</p> <p>If is_percent_growth = 0, growth increment is in units of 8-KB pages, rounded to the nearest 64 KB.</p> <p>If is_percent_growth = 1, growth increment is expressed as a whole number percentage.</p>
is_media_read_only	bit	<p>1 = File is on read-only media.</p> <p>0 = File is on read-write media.</p>
is_read_only	bit	<p>1 = File is marked read-only.</p> <p>0 = File is marked read/write.</p>
is_sparse	bit	<p>1 = File is a sparse file.</p> <p>0 = File is not a sparse file.</p> <p>For more information, see View the Size of the Sparse File of a Database Snapshot (Transact-SQL).</p>
is_percent_growth	bit	<p>1 = Growth of the file is a percentage.</p> <p>0 = Absolute growth size in pages.</p>
is_name_reserved	bit	<p>1 = Dropped file name (name or physical_name) is reusable only after the next log backup. When files are dropped from a database, the logical names stay in a reserved state until the next log backup. This column is relevant only under the full recovery model and the bulk-logged recovery model.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
create_lsn	numeric(25,0)	Log sequence number (LSN) at which the file was created.
drop_lsn	numeric(25,0)	LSN at which the file was dropped. 0 = The file name is unavailable for reuse.
read_only_lsn	numeric(25,0)	LSN at which the filegroup that contains the file changed from read/write to read-only (most recent change).
read_write_lsn	numeric(25,0)	LSN at which the filegroup that contains the file changed from read-only to read/write (most recent change).
differential_base_lsn	numeric(25,0)	Base for differential backups. Data extents changed after this LSN will be included in a differential backup.
differential_base_guid	uniqueidentifier	Unique identifier of the base backup on which a differential backup will be based.
differential_base_time	datetime	Time corresponding to differential_base_lsn.
redo_start_lsn	numeric(25,0)	LSN at which the next roll forward must start. Is NULL unless state = RESTORING or state = RECOVERY_PENDING.
redo_start_fork_guid	uniqueidentifier	Unique identifier of the recovery fork. The first_fork_guid of the next log backup restored must match this value. This represents the current state of the file.
redo_target_lsn	numeric(25,0)	LSN at which the online roll forward on this file can stop. Is NULL unless state = RESTORING or state = RECOVERY_PENDING.
redo_target_fork_guid	uniqueidentifier	The recovery fork on which the file can be recovered. Paired with redo_target_lsn.
backup_lsn	numeric(25,0)	The LSN of the most recent data or differential backup of the file.

NOTE

When you drop or rebuild large indexes, or drop or truncate large tables, the Database Engine defers the actual page deallocations, and their associated locks, until after the transaction commits. Deferred drop operations do not release allocated space immediately. Therefore, the values returned by `sys.database_files` immediately after dropping or truncating a large object may not reflect the actual disk space available.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

Examples

The following statement returns the name, file size, and the amount of empty space for each database file.

```
SELECT name, size/128.0 FileSizeInMB,  
size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0  
AS EmptySpaceInMB  
FROM sys.database_files;
```

For more information when using SQL Database, see [Determining Database Size in Azure SQL Database V12](#) on the SQL Customer Advisory Team blog.

See Also

[Databases and Files Catalog Views \(Transact-SQL\)](#)

[File States](#)

[sys.databases \(Transact-SQL\)](#)





[sys.master_files \(Transact-SQL\)](#)

[Database Files and Filegroups](#)

[sys.data_spaces \(Transact-SQL\)](#)

sys.database_mirroring (Transact-SQL)

11/16/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each database in the instance of SQL Server. If the database is not ONLINE or database mirroring is not enabled, the values of all columns except database_id will be NULL.

To see the row for a database other than master or tempdb, you must either be the database owner or have at least ALTER ANY DATABASE or VIEW ANY DATABASE server-level permission or CREATE DATABASE permission in the master database. To see non-NULL values on a mirror database, you must be a member of the **sysadmin** fixed server role.

NOTE

If a database does not participate in mirroring, all columns prefixed with "mirroring_" are NULL.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	ID of the database. Is unique within an instance of SQL Server.
mirroring_guid	uniqueidentifier	ID of the mirroring partnership. NULL= Database is inaccessible or is not mirrored. Note: If the database does not participate in mirroring, all columns prefixed with "mirroring_" are NULL.

COLUMN NAME	DATA TYPE	DESCRIPTION
mirroring_state	tinyint	<p>State of the mirror database and of the database mirroring session.</p> <p>0 = Suspended</p> <p>1 = Disconnected from the other partner</p> <p>2 = Synchronizing</p> <p>3 = Pending Failover</p> <p>4 = Synchronized</p> <p>5 = The partners are not synchronized. Failover is not possible now.</p> <p>6 = The partners are synchronized. Failover is potentially possible. For information about the requirements for failover see, Database Mirroring Operating Modes.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>
mirroring_state_desc	nvarchar(60)	<p>Description of the state of the mirror database and of the database mirroring session, can be one of:</p> <p>DISCONNECTED</p> <p>SYNCHRONIZED</p> <p>SYNCHRONIZING</p> <p>PENDING_FAILOVER</p> <p>SUSPENDED</p> <p>UNSYNCHRONIZED</p> <p>SYNCHRONIZED</p> <p>NULL</p> <p>For more information, see Mirroring States (SQL Server).</p>
mirroring_role	tinyint	<p>Current role of the local database plays in the database mirroring session.</p> <p>1 = Principal</p> <p>2 = Mirror</p> <p>NULL = Database is inaccessible or is not mirrored.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
mirroring_role_desc	nvarchar(60)	<p>Description of the role the local database plays in mirroring, can be one of:</p> <p>PRINCIPAL</p> <p>MIRROR</p>
mirroring_role_sequence	int	<p>The number of times that mirroring partners have switched the principal and mirror roles due to a failover or forced service.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>
mirroring_safety_level	tinyint	<p>Safety setting for updates on the mirror database:</p> <p>0 = Unknown state</p> <p>1 = Off [asynchronous]</p> <p>2 = Full [synchronous]</p> <p>NULL = Database is inaccessible or is not mirrored.</p>
mirroring_safety_level_desc	nvarchar(60)	<p>Transaction safety setting for the updates on the mirror database, can be one of:</p> <p>UNKNOWN</p> <p>OFF</p> <p>FULL</p> <p>NULL</p>
mirroring_safety_sequence	int	<p>Update the sequence number for changes to transaction safety level.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>
mirroring_partner_name	nvarchar(128)	<p>Server name of the database mirroring partner.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
mirroring_partner_instance	nvarchar(128)	<p>The instance name and computer name for the other partner. Clients require this information to connect to the partner if it becomes the principal server.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>
mirroring_witness_name	nvarchar(128)	<p>Server name of the database mirroring witness.</p> <p>NULL = No witness exists.</p>
mirroring_witness_state	tinyint	<p>State of the witness in the database mirroring session of the database, can be one of:</p> <p>0 = Unknown</p> <p>1 = Connected</p> <p>2 = Disconnected</p> <p>NULL = No witness exists, the database is not online, or the database is not mirrored.</p>
mirroring_witness_state_desc	nvarchar(60)	<p>Description of state, can be one of:</p> <p>UNKNOWN</p> <p>CONNECTED</p> <p>DISCONNECTED</p> <p>NULL</p>
mirroring_failover_lsn	numeric(25,0)	<p>Log sequence number (LSN) of the latest transaction log record that is guaranteed to be hardened to disk on both partners. After a failover, the mirroring_failover_lsn is used by the partners as the point of reconciliation at which the new mirror server begins to synchronize the new mirror database with the new principal database.</p>
mirroring_connection_timeout	int	<p>Mirroring connection time out in seconds. This is the number of seconds to wait for a reply from a partner or witness before considering them unavailable. The default time-out value is 10 seconds.</p> <p>NULL = Database is inaccessible or is not mirrored.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
mirroring_redo_queue	int	<p>Maximum amount of log to be redone on the mirror. If mirroring_redo_queue_type is set to UNLIMITED, which is the default setting, this column is NULL. If the database is not online, this column is also NULL.</p> <p>Otherwise, this column contains the maximum amount of log in megabytes. When the maximum is reached, the log is temporarily stalled on the principal as the mirror server catches up. This feature limits failover time.</p> <p>For more information, see Estimate the Interruption of Service During Role Switching (Database Mirroring).</p>
mirroring_redo_queue_type	nvarchar(60)	<p>UNLIMITED indicates that mirroring will not inhibit the redo queue. This is the default setting.</p> <p>MB for maximum size of the redo queue in mega bytes. Note that if the queue size was specified as kilobytes or gigabytes, the Database Engine converts the value into megabytes.</p> <p>If the database is not online, this column is NULL.</p>
mirroring_end_of_log_lsn	numeric(25,0)	<p>The local end-of-log that has been flushed to disk. This is comparable to the hardened LSN from the mirror server (see the mirroring_failover_lsn column).</p>
mirroring_replication_lsn	numeric(25,0)	<p>The maximum LSN that replication can send.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[ALTER DATABASE \(Transact-SQL\)](#)

[sys.database_mirroring_witnesses \(Transact-SQL\)](#)





[sys.database_mirroring_endpoints \(Transact-SQL\)](#)

[Databases and Files Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.database_recovery_status (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per database. If the database is not opened, the SQL Server Database Engine tries to start it.

To see the row for a database other than **master** or **tempdb**, one of the following must apply:

- Be the owner of the database.
- Have ALTER ANY DATABASE or VIEW ANY DATABASE server-level permissions.
- Have CREATE DATABASE permission in the **master** database.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	ID of the database, unique within an instance of SQL Server.
database_guid	uniqueidentifier	Used to relate all the database files of a database together. All files must have this GUID in their header page for the database to start as expected. Only one database should ever have this GUID, but duplicates can be created by copying and attaching databases. RESTORE always generates a new GUID when you restore a database that does not yet exist. NULL= Database is offline, or the database will not start.
family_guid	uniqueidentifier	Identifier of the "backup family" for the database for detecting matching restore states. NULL= Database is offline or the database will not start.
last_log_backup_lsn	numeric(25,0)	The starting log sequence number of the next log backup. If NULL, a transaction log back up cannot be performed because either the database is in SIMPLE recovery or there is no current database backup.
recovery_fork_guid	uniqueidentifier	Identifies the current recovery fork on which the database is currently active. NULL= Database is offline, or the database will not start.

COLUMN NAME	DATA TYPE	DESCRIPTION
first_recovery_fork_guid	uniqueidentifier	Identifier of the starting recovery fork. NULL= Database is offline, or the database will not start.
fork_point_lsn	numeric(25,0)	If first_recovery_fork_guid is not equal (!=) to recovery_fork_guid , fork_point_lsn is the log sequence number of the current fork point. Otherwise, the value is NULL.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)





[Databases and Files Catalog Views \(Transact-SQL\)](#)

[RESTORE HEADERONLY \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.database_scoped_configurations (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per configuration.

COLUMN NAME	DATA TYPE	DESCRIPTION
configuration_id	int	ID of the configuration option.
name	nvarchar(60)	The name of the configuration option. For information about the possible configurations, see ALTER DATABASE SCOPED CONFIGURATION (Transact-SQL) .
value	sqlvariant	The value set for this configuration option for the primary replica.
value_for_secondary	sqlvariant	The value set for this configuration option for the secondary replicas.

Permissions

Requires membership in the **public** role.

Remarks





When NULL is returned as the value for **value_for_secondary**, this means that the secondary is set to PRIMARY.

See Also

[ALTER DATABASE SCOPED CONFIGURATION \(Transact-SQL\)](#)

sys.master_files (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per file of a database as stored in the master database. This is a single, system-wide view.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	ID of the database to which this file applies. The masterdatabase_id is always 1.
file_id	int	ID of the file within database. The primary file_id is always 1.
file_guid	uniqueidentifier	Unique identifier of the file. NULL = Database was upgraded from an earlier version of SQL Server.
type	tinyint	File type: 0 = Rows. 1 = Log 2 = FILESTREAM 3 = Identified for informational purposes only. Not supported. Future compatibility is not guaranteed. 4 = Full-text (Full-text catalogs earlier than SQL Server 2008; full-text catalogs that are upgraded to or created in SQL Server 2008 or higher will report a file type 0.)
type_desc	nvarchar(60)	Description of the file type: ROWS LOG FILESTREAM FULLTEXT (Full-text catalogs earlier than SQL Server 2008.)
data_space_id	int	ID of the data space to which this file belongs. Data space is a filegroup. 0 = Log files

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Logical name of the file in the database.
physical_name	nvarchar(260)	Operating-system file name.
state	tinyint	<p>File state:</p> <p>0 = ONLINE</p> <p>1 = RESTORING</p> <p>2 = RECOVERING</p> <p>3 = RECOVERY_PENDING</p> <p>4 = SUSPECT</p> <p>5 = Identified for informational purposes only. Not supported. Future compatibility is not guaranteed.</p> <p>6 = OFFLINE</p> <p>7 = DEFUNCT</p>
state_desc	nvarchar(60)	<p>Description of the file state:</p> <p>ONLINE</p> <p>RESTORING</p> <p>RECOVERING</p> <p>RECOVERY_PENDING</p> <p>SUSPECT</p> <p>OFFLINE</p> <p>DEFUNCT</p> <p>For more information, see File States.</p>
size	int	<p>Current file size, in 8-KB pages. For a database snapshot, size reflects the maximum space that the snapshot can ever use for the file.</p> <p>Note: This field is populated as zero for FILESTREAM containers. Query the <i>sys.database_files</i> catalog view for the actual size of FILESTREAM containers.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
max_size	int	<p>Maximum file size, in 8-KB pages:</p> <p>0 = No growth is allowed.</p> <p>-1 = File will grow until the disk is full.</p> <p>268435456 = Log file will grow to a maximum size of 2 TB.</p> <p>Note: Databases that are upgraded with an unlimited log file size will report -1 for the maximum size of the log file.</p>
growth	int	<p>0 = File is fixed size and will not grow.</p> <p>>0 = File will grow automatically.</p> <p>If is_percent_growth = 0, growth increment is in units of 8-KB pages, rounded to the nearest 64 KB</p> <p>If is_percent_growth = 1, growth increment is expressed as a whole number percentage.</p>
is_media_read_onlyF	bit	<p>1 = File is on read-only media.</p> <p>0 = File is on read/write media.</p>
is_read_only	bit	<p>1 = File is marked read-only.</p> <p>0 = file is marked read/write.</p>
is_sparse	bit	<p>1 = File is a sparse file.</p> <p>0 = File is not a sparse file.</p> <p>For more information, see View the Size of the Sparse File of a Database Snapshot (Transact-SQL).</p>
is_percent_growth	bit	<p>1 = Growth of the file is a percentage.</p> <p>0 = Absolute growth size in pages.</p>
is_name_reserved	bit	<p>1 = Dropped file name is reusable. A log backup must be taken before the name (name or physical_name) can be reused for a new file name.</p> <p>0 = File name is unavailable for reuse.</p>
create_lsn	numeric(25,0)	Log sequence number (LSN) at which the file was created.
drop_lsn	numeric(25,0)	LSN at which the file was dropped.

COLUMN NAME	DATA TYPE	DESCRIPTION
read_only_lsn	numeric(25,0)	LSN at which the filegroup that contains the file changed from read/write to read-only (most recent change).
read_write_lsn	numeric(25,0)	LSN at which the filegroup that contains the file changed from read-only to read/write (most recent change).
differential_base_lsn	numeric(25,0)	Base for differential backups. Data extents changed after this LSN will be included in a differential backup.
differential_base_guid	uniqueidentifier	Unique identifier of the base backup on which a differential backup will be based.
differential_base_time	datetime	Time corresponding to differential_base_lsn.
redo_start_lsn	numeric(25,0)	LSN at which the next roll forward must start. Is NULL unless state = RESTORING or state = RECOVERY_PENDING.
redo_start_fork_guid	uniqueidentifier	Unique identifier of the recovery fork. The first_fork_guid of the next log backup restored must match this value. This represents the current state of the container.
redo_target_lsn	numeric(25,0)	LSN at which the online roll forward on this file can stop. Is NULL unless state = RESTORING or state = RECOVERY_PENDING.
redo_target_fork_guid	uniqueidentifier	The recovery fork on which the container can be recovered. Paired with redo_target_lsn.
backup_lsn	numeric(25,0)	The LSN of the most recent data or differential backup of the file.
credential_id	int	The <code>credential_id</code> from <code>sys.credentials</code> used for storing the file. For example, when SQL Server is running on an Azure Virtual Machine and the database files are stored in Azure blob storage, a credential is configured with the access credentials to the storage location.

NOTE

When you drop or rebuild large indexes, or drop or truncate large tables, the Database Engine defers the actual page deallocations, and their associated locks, until after the transaction commits. Deferred drop operations do not release allocated space immediately. Therefore, the values returned by `sys.master_files` immediately after dropping or truncating a large object may not reflect the actual disk space available.

Permissions

The minimum permissions that are required to see the corresponding row are `CREATE DATABASE`, `ALTER ANY DATABASE`, or `VIEW ANY DEFINITION`.

See Also

[Databases and Files Catalog Views \(Transact-SQL\)](#)

[File States](#)





[sys.databases \(Transact-SQL\)](#)

[sys.database_files \(Transact-SQL\)](#)

[Database Files and Filegroups](#)

Endpoints Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.database_mirroring_endpoints	sys.service_broker_endpoints
sys.endpoints	sys.soap_endpoints
sys.endpoint_webmethods	sys.tcp_endpoints
sys.http_endpoints	





See Also

[System Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.database_mirroring_endpoints (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for the database mirroring endpoint of an instance of SQL Server.

NOTE

The database mirroring endpoint supports both sessions between database mirroring partners and with witnesses and sessions between the primary replica of a Always On availability group and its secondary replicas.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	—	Inherits columns from sys.endpoints (for more information, see sys.endpoints (Transact-SQL)).
role	tinyint	Mirroring role, one of: 0 = None 1 = Partner 2 = Witness 3 = All Note: This value is relevant only for database mirroring.
role_desc	nvarchar(60)	Description of mirroring role, one of: NONE PARTNER WITNESS ALL Note: This value is relevant only for database mirroring.
is_encryption_enabled	bit	1 means that encryption is enabled. 0 means that encryption is disabled.

COLUMN NAME	DATA TYPE	DESCRIPTION
connection_auth	tinyint	<p>The type of connection authentication required for connections to this endpoint, one of:</p> <p>1 - NTLM</p> <p>2 - KERBEROS</p> <p>3 - NEGOTIATE</p> <p>4 - CERTIFICATE</p> <p>5 - NTLM, CERTIFICATE</p> <p>6 - KERBEROS, CERTIFICATE</p> <p>7 - NEGOTIATE, CERTIFICATE</p> <p>8 - CERTIFICATE, NTLM</p> <p>9 - CERTIFICATE, KERBEROS</p> <p>10 - CERTIFICATE, NEGOTIATE</p>
connection_auth_desc	Nvarchar (60)	<p>Description of the type of authentication required for connections to this endpoint, one of:</p> <p>NTLM</p> <p>KERBEROS</p> <p>NEGOTIATE</p> <p>CERTIFICATE</p> <p>NTLM, CERTIFICATE</p> <p>KERBEROS, CERTIFICATE</p> <p>NEGOTIATE, CERTIFICATE</p> <p>CERTIFICATE, NTLM</p> <p>CERTIFICATE, KERBEROS</p> <p>CERTIFICATE, NEGOTIATE</p>
certificate_id	int	<p>ID of certificate used for authentication, if any.</p> <p>0 = Windows Authentication is being used.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
encryption_algorithm	tinyint	Encryption algorithm, one of: 0 – NONE 1 – RC4 2 – AES 3 – NONE, RC4 4 – NONE, AES 5 – RC4, AES 6 – AES, RC4 7 – NONE, RC4, AES 8 – NONE, AES, RC4
encryption_algorithm_desc	nvarchar(60)	Description of the encryption algorithm, one of: NONE RC4 AES NONE, RC4 NONE, AES RC4, AES AES, RC4 NONE, RC4, AES NONE, AES, RC4

Remarks

NOTE

The RC4 algorithm is only supported for backward compatibility. New material can only be encrypted using RC4 or RC4_128 when the database is in compatibility level 90 or 100. (Not recommended.) Use a newer algorithm such as one of the AES algorithms instead. In SQL Server 2012 and higher, material encrypted using RC4 or RC4_128 can be decrypted in any compatibility level.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Specify the Endpoint URL When Adding or Modifying an Availability Replica \(SQL Server\)](#)

[sys.availability_replicas \(Transact-SQL\)](#)

[sys.database_mirroring \(Transact-SQL\)](#)





[sys.database_mirroring_witnesses \(Transact-SQL\)](#)

[The Database Mirroring Endpoint \(SQL Server\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.endpoints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per endpoint that is created in the system. There is always exactly one SYSTEM endpoint.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the endpoint. Is unique within the server. Is not nullable.
endpoint_id	int	ID of the endpoint. Is unique within the server. An endpoint with an ID less than 65536 is a system endpoint. Is not nullable.
principal_id	int	ID of the server principal that created and owns this endpoint. Is nullable.
protocol	tinyint	Endpoint protocol. 1 = HTTP 2 = TCP 3 = Name pipes 4 = Shared memory 5 = Virtual Interface Adapter (VIA) Is not nullable.
protocol_desc	nvarchar(60)	Description of the endpoint protocol. NULLABLE. One of the following values: HTTP TCP NAMED_PIPES SHARED_MEMORY VIA Note: The VIA protocol is deprecated. This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

COLUMN NAME	DATA TYPE	DESCRIPTION
type	tinyint	<p>Endpoint payload type.</p> <p>1 = SOAP</p> <p>2 = TSQL</p> <p>3 = SERVICE_BROKER</p> <p>4 = DATABASE_MIRRORING</p> <p>Is not nullable.</p>
type_desc	nvarchar(60)	<p>Description of the endpoint payload type. Is nullable. One of the following values:</p> <p>SOAP</p> <p>TSQL</p> <p>SERVICE_BROKER</p> <p>DATABASE_MIRRORING</p>
state	tinyint	<p>The endpoint state.</p> <p>0 = STARTED, listening and processing requests.</p> <p>1 = STOPPED, listening, but not processing requests.</p> <p>2 = DISABLED, not listening.</p> <p>The default state is 1. Is nullable.</p>
state_desc	nvarchar(60)	<p>Description of the endpoint state.</p> <p>STARTED = Listening and processing requests.</p> <p>STOPPED = Listening, but not processing requests.</p> <p>DISABLED = Not listening.</p> <p>The default state is STOPPED.</p> <p>Is nullable.</p>
is_admin_endpoint	bit	<p>Indicates whether the endpoint is for administrative use.</p> <p>0 = Nonadministrative endpoint.</p> <p>1 = Endpoint is an administrative endpoint.</p> <p>Is not nullable.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Endpoints Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.endpoint_webmethods (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Contains a row FOR EACH SOAP method defined on a SOAP-enabled HTTP endpoint. The combination of the endpoint_id and namespace columns is unique.

COLUMN NAME	DATA TYPE	DESCRIPTION
endpoint_id	int	ID of the endpoint that the webmethod is defined on.
namespace	nvarchar(384)	Namespace for the webmethod.
method_alias	nvarchar(64)	<p>Alias for the method.</p> <p>Note: Transact-SQL identifiers allow characters that are not legal in WSDL method names.</p> <p>The alias is used to map the name exposed in the WSDL description of the endpoint to the actual underlying Transact-SQL executable object that is called when the webmethod is invoked.</p>
object_name	nvarchar(776)	<p>The object name that the webmethod is redirected to, as specified in the NAME = option. Name parts are separated by a period (.), and delimited using brackets, <code>[``]</code>.</p> <p>The object name must be a three-part name, as specified in the WSDL option.</p>
result_schema	tinyint	<p>Option that determines which, if any, XSD is sent back with a response.</p> <p>0 = None</p> <p>1 = Standard</p> <p>2 = Default</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
result_schema_desc	nvarchar(60)	<p>Description of option that determines which, if any, XSD is sent back with a response.</p> <p>NONE</p> <p>STANDARD</p> <p>DEFAULT</p>
result_format	tinyint	<p>Option that determines how results are formatted in the response.</p> <p>1 = ALL_RESULTS</p> <p>2 = ROWSETS_ONLY</p> <p>3 = NONE</p>
result_format_desc	nvarchar(60)	<p>Description of the option that determines how results are formatted in the response.</p> <p>ALL_RESULTS</p> <p>ROWSETS_ONLY</p> <p>NONE</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Endpoints Catalog Views \(Transact-SQL\)](#)
[Catalog Views \(Transact-SQL\)](#)

sys.http_endpoints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each endpoint created in the server that uses the HTTP protocol.

COLUMN NAME	DATA TYPE	DESCRIPTION
< inherited columns>		Inherits columns from sys.endpoints (Transact-SQL) .
site	nvarchar(128)	Name of the host computer for the site, as specified in the SITE = option.
url_path	nvarchar(4000)	Path-only portion of the URL for this HTTP endpoint, as specified by the PATH= option.
is_clear_port_enabled	bit	1 = Clear port is enabled using the PORT = CLEAR option.
clear_port	int	Port number specified in the CLEAR PORT = option. NULL = Not specified.
is_ssl_port_enabled	bit	1 = SSL port is enabled using the PORT = SSL option.
ssl_port	int	Port number value specified in the SSL PORT = option. NULL = Not specified.
is_anonymous_enabled	bit	1 = Anonymous access is enabled using the AUTHENTICATION = ANONYMOUS option.
is_basic_auth_enabled	bit	1 = Basic authentication is enabled using the AUTHENTICATION = BASIC option.
is_digest_auth_enabled	bit	1 = Digest authentication is enabled using the AUTHENTICATION = DIGEST option.
is_kerberos_auth_enabled	bit	1 = Integrated authentication enabled using the AUTHENTICATION = KERBEROS option.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_ntlm_auth_enabled	bit	1 = Integrated authentication enabled using the AUTHENTICATION = NTLM option.
is_integrated_auth_enabled	bit	1 = Integrated authentication is enabled using the AUTHENTICATION = INTEGRATED option.
authorization_realm	nvarchar(128)	Hint that is returned to the client as part of the HTTP DIGEST authentication challenge. The value of the AUTH REALM option. Is NULL if not specified or if DIGEST authentication is not enabled.
default_logon_domain	nvarchar(128)	Default login domain if you enable BASIC authentication. The value of the DEFAULT LOGON DOMAIN option. Is NULL if not specified or if BASIC authentication is not enabled.
is_compression_enabled	bit	1 = COMPRESSION = ENABLED option is set.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Endpoints Catalog Views \(Transact-SQL\)](#)

sys.service_broker_endpoints (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains one row for the Service Broker endpoint. For every row in this view, there is a corresponding row with the same **endpoint_id** in the **sys.tcp_endpoints** view that contains the TCP configuration metadata. TCP is the only allowed protocol for Service Broker.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	--	Inherits columns from sys.endpoints (Transact-SQL) .
is_message_forwarding_enabled	bit	Does endpoint support message forwarding. This is initially set to 0 (disabled). Not NULLABLE.
message_forwarding_size	int	The maximum number of megabytes of tempdb space allowed to be used for messages being forwarded. This is initially set to 10 . Not NULLABLE.
connection_auth	tinyint	<p>The type of connection authentication required for connections to this endpoint, one of:</p> <ul style="list-style-type: none">1 - NTLM2 - KERBEROS3 - NEGOTIATE4 - CERTIFICATE5 - NTLM, CERTIFICATE6 - KERBEROS, CERTIFICATE7 - NEGOTIATE, CERTIFICATE8 - CERTIFICATE, NTLM9 - CERTIFICATE, KERBEROS10 - CERTIFICATE, NEGOTIATE <p>Not NULLABLE.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
connection_auth_desc	nvarchar(60)	<p>Description of the type of connection authentication required for connections to this endpoint, one of:</p> <p>NTLM</p> <p>KERBEROS</p> <p>NEGOTIATE</p> <p>CERTIFICATE</p> <p>NTLM, CERTIFICATE</p> <p>KERBEROS, CERTIFICATE</p> <p>NEGOTIATE, CERTIFICATE</p> <p>CERTIFICATE, NTLM</p> <p>CERTIFICATE, KERBEROS</p> <p>CERTIFICATE, NEGOTIATE</p> <p>NULLABLE.</p>
certificate_id	int	<p>ID of certificate used for authentication, if any.</p> <p>0 = Windows Authentication is being used.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
encryption_algorithm	tinyint	<p>Encryption algorithm. The following are the possible values with their descriptions and corresponding DDL options.</p> <p>0 : NONE. Corresponding DDL option: Disabled.</p> <p>1 : RC4. Corresponding DDL option: {Required Required algorithm RC4}.</p> <p>2 : AES. Corresponding DDL option: Required Algorithm AES.</p> <p>3 : NONE, RC4. Corresponding DDL option: {Supported Supported Algorithm RC4}.</p> <p>4 : NONE, AES. Corresponding DDL option: Supported Algorithm AES.</p> <p>5 : RC4, AES. Corresponding DDL option: Required Algorithm RC4 AES.</p> <p>6 : AES, RC4. Corresponding DDL option: Required Algorithm AES RC4.</p> <p>7 : NONE, RC4, AES. Corresponding DDL option: Supported Algorithm RC4 AES.</p> <p>8 : NONE, AES, RC4. Corresponding DDL option: Supported Algorithm AES RC4.</p> <p>Not NULLABLE.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
encryption_algorithm_desc	nvarchar(60)	<p>Encryption algorithm description. The possible values and their corresponding DDL options are listed below:</p> <p>NONE : Disabled</p> <p>RC4 : {Required Required Algorithm RC4}</p> <p>AES : Required Algorithm AES</p> <p>NONE, RC4 : {Supported Supported Algorithm RC4}</p> <p>NONE, AES : Supported Algorithm AES</p> <p>RC4, AES : Required Algorithm RC4 AES</p> <p>AES, RC4 : Required Algorithm AES RC4</p> <p>NONE, RC4, AES : Supported Algorithm RC4 AES</p> <p>NONE, AES, RC4 : Supported Algorithm AES RC4</p> <p>NULLABLE.</p>

Remarks

NOTE

The RC4 algorithm is only supported for backward compatibility. New material can only be encrypted using RC4 or RC4_128 when the database is in compatibility level 90 or 100. (Not recommended.) Use a newer algorithm such as one of the AES algorithms instead. In SQL Server 2012 and later versions, material encrypted using RC4 or RC4_128 can be decrypted in any compatibility level.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

- [ALTER ENDPOINT \(Transact-SQL\)](#)
- [CREATE ENDPOINT \(Transact-SQL\)](#)

sys.soap_endpoints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Contains one row for each endpoint in the server that carries a SOAP-type payload. For every row in this view, there is a corresponding row with the same **endpoint_id** in the **sys.http_endpoints** catalog view that carries the HTTP configuration metadata.

COLUMN NAME	DATA TYPE	DESCRIPTION
< inherited columns >		For a list of columns that this view inherits, see sys.endpoints (Transact-SQL) .
is_sql_language_enabled	bit	1 = BATCHES = ENABLED option was specified, meaning that ad-hoc SQL batches are allowed on the endpoint.
wsdl_generator_procedure	nvarchar(776)	The three-part name of the stored procedure that implements this method. Names of methods require strict three-part syntax. one, two, or four-part names are not allowed.
default_database	sysname	The name of the default database given in the DATABASE = option. NULL = DEFAULT was specified.
default_namespace	nvarchar(384)	The default namespace specified in the NAMESPACE = option, or ' http://tempuri.org ' if DEFAULT was specified instead.
default_result_schema	tinyint	The default value of the SCHEMA = option. 0 = NONE 1 = STANDARD
default_result_schema_desc	nvarchar(60)	Description of the default value of the SCHEMA = option. NONE STANDARD

COLUMN NAME	DATA TYPE	DESCRIPTION
is_xml_charset_enforced	bit	0 = CHARACTER_SET = SQL option was specified. 1 = CHARACTER_SET = XML option was specified.
is_session_enabled	bit	0 = SESSION = DISABLE option was specified. 1 = SESSION = ENABLED option was specified.
session_timeout	int	Value specified in SESSION_TIMEOUT = option.
login_type	nvarchar(60)	Kind of authentication allowed on this endpoint. WINDOWS MIXED
header_limit	int	Maximum allowable size of the SOAP header.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Endpoints Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.tcp_endpoints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each TCP endpoint that is in the system. The endpoints that are described by **sys.tcp_endpoints** provide an object to grant and revoke the connection privilege. The information that is displayed regarding ports and IP addresses is not used to configure the protocols and may not match the actual protocol configuration. To view and configure protocols, use SQL Server Configuration Manager.

COLUMN NAME	DATA TYPE	DESCRIPTION
< inherited columns >		Inherits columns from sys.endpoints .
port	int	The port number that the endpoint is listening on. Is not nullable.
is_dynamic_port	bit	1 = Port number was dynamically assigned. Is not nullable.
ip_address	nvarchar(45)	Listener IP address as specified by the LISTENER_IP clause. Is nullable.

Remarks

Execute the following query to gather information about the endpoints and connections. Endpoints without current connections or without TCP connections will appear with NULL values. Add the **WHERE** clause

`WHERE des.session_id = @@SPID` to return information about the current connection.

```
SELECT des.login_name, des.host_name, program_name, dec.net_transport, des.login_time,
e.name AS endpoint_name, e.protocol_desc, e.state_desc, e.is_admin_endpoint,
t.port, t.is_dynamic_port, dec.local_net_address, dec.local_tcp_port
FROM sys.endpoints AS e
LEFT JOIN sys.tcp_endpoints AS t
ON e.endpoint_id = t.endpoint_id
LEFT JOIN sys.dm_exec_sessions AS des
ON e.endpoint_id = des.endpoint_id
LEFT JOIN sys.dm_exec_connections AS dec
ON des.session_id = dec.session_id;
```

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Endpoints Catalog Views \(Transact-SQL\)](#)

Extended Events Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views for SQL Server Extended Events.





sys.server_event_sessions (Transact-SQL)	sys.server_event_session_fields (Transact-SQL)
sys.server_event_session_actions (Transact-SQL)	sys.server_event_session_targets (Transact-SQL)
sys.server_event_session_events (Transact-SQL)	

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.database_event_session_targets (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event target for an event session.

||

| - |

|Applies to: Azure SQL Database V12 and any later versions.|

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
target_id	int	The ID of the target. ID is unique within the event session object. Is not nullable.
name	sysname	The name of the event target. Is not nullable.
package	sysname	The name of the event package that contains the event target. Is not nullable.
module	sysname	The name of the module that contains the event target. Is not nullable.

Permissions

Requires VIEW DATABASE STATE permission on the server.

Remarks

This view has the following relationship cardinalities.





From	To	Relationship
sys.database_event_session_targets.event_session_id	sys.database_event_sessions.event_session_id	Many to one

See Also

[Extended Events](#)

sys.database_event_session_fields (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each customizable column that was explicitly set on events and targets.

|Applies to: Azure SQL Database V12 and any later versions.|

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
object_id	int	The ID of the object this field is associated with. Is not nullable.
name	sysname	The name of the field. Is not nullable.
value	sql_variant	The value of the field. Is not nullable.

Permissions

Requires VIEW DATABASE STATE permission on the server.

Remarks

This view has the following relationship cardinalities.

From	To	Relationship
sys.database_event_session_actions.event_session_id	sys.database_event_sessions.event_session_id	Many to one
sys.database_event_session_actions.event_id	sys.database_event_session_events.event_session_id	Many to one
sys.database_event_session_actions.object_id	sys.database_event_session_events.event_id	
sys.database_event_session_actions.event_session_id		





sys.database_event_session_actions.event_session_id	sys.database_event_session_targets.event_session_id	Many to one
sys.database_event_session_actions.object_id	sys.database_event_session_targets.target_id	

See Also

[Extended Events](#)

sys.database_event_session_events (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event in an event session.

|Applies to: Azure SQL Database V12 and any later versions.|

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
event_id	int	The ID of the event. This ID is unique within an event session object. Is not nullable.
name	sysname	The name of the event. Is not nullable.
package	sysname	The name of the event package that contains the event. Is not nullable.
module	sysname	The name of the module that contains the event. Is not nullable.
predicate	nvarchar(3000)	The predicate expression that is applied to the event. Is nullable.
predicate_xml	nvarchar(3000)	The XML predicate expression that is applied to the event. Is nullable.

Permissions

Requires VIEW DATABASE STATE permission on the server.

Remarks

This view has the following relationship cardinalities.





From	To	Relationship
sys.database_event_session_events.event_session_id	sys.database_event_sessions.event_session_id	Many to one

See Also

[Extended Events](#)

sys.database_event_session_actions (Azure SQL Database)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each action on each event of an event session.

|Applies to: Azure SQL Database V12 and any later versions.|

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
event_id	int	The ID of the event. This ID is unique within the event session object. Is not nullable.
name	sysname	The name of the action. Is nullable.
package	sysname	The name of the event package that contains the event. Is nullable.
module	sysname	The name of the module that contains the event. Is nullable.

Permissions

Requires VIEW DATABASE STATE permission on the server.





Remarks

This view has the following relationship cardinalities.

From	To	Relationship
sys.database_event_session_actions.event_session_id	sys.sys.database_event_sessions.event_session_id	Many to one
sys.database_event_session_actions.event_id	sys.database_event_session_events.event_session_id	Many to one
sys.database_event_session_actions.event_session_id	sys.database_event_session_events.event_id	

sys.database_event_sessions (Azure SQL Database)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Lists all the event session definitions that exist in the current database, in Azure SQL Database.

NOTE

The similar catalog view named `sys.server_event_sessions` applies only to Microsoft SQL Server.

||

| - |

Applies to: SQL Database, and to any later versions.]

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The unique ID of the event session. Is not nullable.
name	sysname	The user-defined name for identifying the event session. name is unique. Is not nullable.
event_retention_mode	nchar(1)	Determines how event loss is handled. The default is S. Is not nullable. Is one of the following: S. Maps to event_retention_mode_desc = ALLOW_SINGLE_EVENT_LOSS M. Maps to event_retention_mode_desc = ALLOW_MULTIPLE_EVENT_LOSS N. Maps to event_retention_mode_desc = NO_EVENT_LOSS

COLUMN NAME	DATA TYPE	DESCRIPTION
event_retention_mode_desc	sysname	<p>Describes how event loss is handled. The default is ALLOW_SINGLE_EVENT_LOSS. Is not nullable. Is one of the following:</p> <p>ALLOW_SINGLE_EVENT_LOSS. Events can be lost from the session. Single events are dropped only when all event buffers are full. Losing single events when buffers are full allows for acceptable SQL Server performance characteristics, while minimizing the loss in the processed event stream.</p> <p>ALLOW_MULTIPLE_EVENT_LOSS. Full event buffers can be lost from the session. The number of events lost depends on the memory size allocated to the session, the partitioning of the memory, and the size of the events in the buffer. This option minimizes performance impact on the server when event buffers are quickly filled. However, large numbers of events can be lost from the session.</p> <p>NO_EVENT_LOSS. No event loss is allowed. This option ensures that all events raised are retained. Using this option forces all the tasks that fire events to wait until space is available in an event buffer. This may lead to detectable performance degradation while the event session is active.</p>
max_dispatch_latency	int	The amount of time, in milliseconds, that events will be buffered in memory before they are served to session targets. Valid values are from 1 to 2147483648, and -1. A value of -1 indicates that dispatch latency is infinite. Is nullable.
max_memory	int	The amount of memory allocated to the session for event buffering. The default value is 4 MB. Is nullable.
max_event_size	int	The amount of memory set aside for events that do not fit in event session buffers. If max_event_size exceeds the calculated buffer size, two additional buffers of max_event_size are allocated to the event session. Is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
memory_partition_mode	nchar(1)	<p>The location in memory where event buffers are created. The default partition mode is G. Is not nullable. memory_partition_mode is one of:</p> <p>G - NONE</p> <p>C - PER_CPU</p> <p>N - PER_NODE</p>
memory_partition_mode_desc	sysname	<p>The default is NONE. Is not nullable. Is one of the following:</p> <p>NONE. A single set of buffers are created within a SQL Server instance.</p> <p>PER_CPU. A set of buffers is created for each CPU.</p> <p>PER_NODE. A set of buffers is created for each non-uniform memory access (NUMA) node.</p>
track_causality	bit	<p>Enable or disable causality tracking. If set to 1 (ON), tracking is enabled and related events on different server connections can be correlated. The default setting is 0 (OFF). Is not nullable.</p>
startup_state	bit	<p>Value determines whether or not session is started automatically when the server starts. The default is 0. Is not nullable. Is one of:</p> <p>0 (OFF). The session does not start when the server starts.</p> <p>1 (ON). The event session starts when the server starts.</p>

Permissions

Requires VIEW SERVER STATE permission on the server.

sys.server_event_sessions (Transact-SQL)

11/27/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Lists all the event session definitions that exist in SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The unique ID of the event session. Is not nullable.
name	sysname	The user-defined name for identifying the event session. name is unique. Is not nullable.
event_retention_mode	nchar(1)	Determines how event loss is handled. The default is S. Is not nullable. Is one of the following: S. Maps to event_retention_mode_desc = ALLOW_SINGLE_EVENT_LOSS M. Maps to event_retention_mode_desc = ALLOW_MULTIPLE_EVENT_LOSS N. Maps to event_retention_mode_desc = NO_EVENT_LOSS

COLUMN NAME	DATA TYPE	DESCRIPTION
event_retention_mode_desc	sysname	<p>Describes how event loss is handled. The default is ALLOW_SINGLE_EVENT_LOSS. Is not nullable. Is one of the following:</p> <p>ALLOW_SINGLE_EVENT_LOSS. Events can be lost from the session. Single events are dropped only when all event buffers are full. Losing single events when buffers are full allows for acceptable SQL Server performance characteristics, while minimizing the loss in the processed event stream.</p> <p>ALLOW_MULTIPLE_EVENT_LOSS. Full event buffers can be lost from the session. The number of events lost depends on the memory size allocated to the session, the partitioning of the memory, and the size of the events in the buffer. This option minimizes performance impact on the server when event buffers are quickly filled. However, large numbers of events can be lost from the session.</p> <p>NO_EVENT_LOSS. No event loss is allowed. This option ensures that all events raised are retained. Using this option forces all the tasks that fire events to wait until space is available in an event buffer. This may lead to detectable performance degradation while the event session is active.</p>
max_dispatch_latency	int	The amount of time, in milliseconds, that events will be buffered in memory before they are served to session targets. Valid values are from 1 to 2147483648, and -1. A value of -1 indicates that dispatch latency is infinite. Is nullable.
max_memory	int	The amount of memory allocated to the session for event buffering. The default value is 4 MB. Is nullable.
max_event_size	int	The amount of memory set aside for events that do not fit in event session buffers. If max_event_size exceeds the calculated buffer size, two additional buffers of max_event_size are allocated to the event session. Is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
memory_partition_mode	nchar(1)	<p>The location in memory where event buffers are created. The default partition mode is G. Is not nullable. memory_partition_mode is one of:</p> <p>G - NONE</p> <p>C - PER_CPU</p> <p>N - PER_NODE</p>
memory_partition_mode_desc	sysname	<p>The default is NONE. Is not nullable. Is one of the following:</p> <p>NONE. A single set of buffers are created within a SQL Server instance.</p> <p>PER_CPU. A set of buffers is created for each CPU.</p> <p>PER_NODE. A set of buffers is created for each non-uniform memory access (NUMA) node.</p>
track_causality	bit	<p>Enable or disable causality tracking. If set to 1 (ON), tracking is enabled and related events on different server connections can be correlated. The default setting is 0 (OFF). Is not nullable.</p>
startup_state	bit	<p>Value determines whether or not session is started automatically when the server starts. The default is 0. Is not nullable. Is one of:</p> <p>0 (OFF). The session does not start when the server starts.</p> <p>1 (ON). The event session starts when the server starts.</p>

Permissions

Requires VIEW SERVER STATE permission on the server.

See Also



[Catalog Views \(Transact-SQL\)](#)

[Extended Events Catalog Views \(Transact-SQL\)](#)

[Extended Events](#)

sys.server_event_session_actions (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each action on each event of an event session.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
event_id	int	The ID of the event. This ID is unique within the event session object. Is not nullable.
name	sysname	The name of the action. Is nullable.
package	sysname	The name of the event package that contains the event. Is nullable.
module	sysname	The name of the module that contains the event. Is nullable.

Permissions

Requires VIEW SERVER STATE permission on the server.

Remarks

This view has the following relationship cardinalities.

From	To	Relationship
sys.server_event_session_actions.event_session_id	sys.sys.server_event_sessions.event_session_id	Many to one
sys.server_event_session_actions.event_id	sys.server_event_session_events.event_session_id	Many to one
sys.server_event_session_actions.event_session_id	sys.server_event_session_events.event_id	

See Also

[Catalog Views \(Transact-SQL\)](#)

[Extended Events Catalog Views \(Transact-SQL\)](#)

[Extended Events](#)

sys.server_event_session_events (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event in an event session.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
event_id	int	The ID of the event. This ID is unique within an event session object. Is not nullable.
name	sysname	The name of the event. Is not nullable.
package	sysname	The name of the event package that contains the event. Is not nullable.
module	sysname	The name of the module that contains the event. Is not nullable.
predicate	nvarchar(3000)	The predicate expression that is applied to the event. Is nullable.
predicate_xml	nvarchar(3000)	The XML predicate expression that is applied to the event. Is nullable.

Permissions

Requires VIEW SERVER STATE permission on the server.

Remarks

This view has the following relationship cardinalities.

From	To	Relationship
sys.server_event_session_events.event_session_id	sys.server_event_sessions.event_session_id	Many to one

See Also


[Catalog Views \(Transact-SQL\)](#)

[Extended Events Catalog Views \(Transact-SQL\)](#)

[Extended Events](#)

sys.server_event_session_fields (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each customizable column that was explicitly set on events and targets.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
object_id	int	The ID of the object this field is associated with. Is not nullable.
name	sysname	The name of the field. Is not nullable.
value	sql_variant	The value of the field. Is not nullable.

Permissions

Requires VIEW SERVER STATE permission on the server.

Remarks

This view has the following relationship cardinalities.



From	To	Relationship
sys.server_event_session_actions.event_session_id	sys.server_event_sessions.event_session_id	Many to one
sys.server_event_session_actions.event_id	sys.server_event_session_events.event_session_id	Many to one
sys.server_event_session_actions.object_id	sys.server_event_session_events.event_id	
sys.server_event_session_actions.event_session_id		
sys.server_event_session_actions.event_session_id	sys.server_event_session_targets.event_session_id	Many to one
sys.server_event_session_actions.object_id	sys.server_event_session_targets.target_id	

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.server_event_session_targets (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event target for an event session.

COLUMN NAME	DATA TYPE	DESCRIPTION
event_session_id	int	The ID of the event session. Is not nullable.
target_id	int	The ID of the target. ID is unique within the event session object. Is not nullable.
name	sysname	The name of the event target. Is not nullable.
package	sysname	The name of the event package that contains the event target. Is not nullable.
module	sysname	The name of the module that contains the event target. Is not nullable.

Permissions

Requires VIEW SERVER STATE permission on the server.

Remarks

This view has the following relationship cardinalities.

From	To	Relationship
sys.server_event_session_targets.event_session_id	sys.server_event_sessions.event_session_id	Many to one

See Also



[Catalog Views \(Transact-SQL\)](#)

[Extended Events Catalog Views \(Transact-SQL\)](#)

[Extended Events](#)

External Operations Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Here are the catalog views of external data objects, such as data sources, file formats, and tables. Use these views to query the state of items such as Hadoop tables used in PolyBase operations.





[sys.external_tables](#) (Transact-SQL)

[sys.external_data_sources](#) (Transact-SQL)

[sys.external_file_formats](#) (Transact-SQL)

sys.external_tables (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each external table in the current database.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .	
max_column_id_used	int	Maximum column ID ever used for this table.	
uses_ansi_nulls	bit	Table was created with the SET ANSI_NULLS database option ON.	
data_source_id	int	Object ID for the external data source.	
file_format_id	int	For external tables over a HADOOP external data source, this is the Object ID for the external file format.	
location	nvarchar(4000)	For external tables over a HADOOP external data source, this is the path of the external data in HDFS.	
reject_type	tinyint	For external tables over a HADOOP external data source, this is the way rejected rows are counted when querying external data.	VALUE – the number of rejected rows. PERCENTAGE – the percentage of rejected rows.
reject_value	float	For external tables over a HADOOP external data source: For <i>reject_type</i> = value, this is the number of row rejections to allow before failing the query. For <i>reject_type</i> = percentage, this is the percentage of row rejections to allow before failing the query.	

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
reject_sample_value	int	For <i>reject_type</i> = percentage, this is the number of rows to load, either successfully or unsuccessfully, before calculating the percentage of rejected rows.	NULL if reject_type = VALUE.
distribution_type	int	For external tables over a SHARD_MAP_MANAGER external data source, this is the data distribution of the rows across the underlying base tables.	0 – Sharded 1 – Replicated 2 – Round robin
distribution_desc	nvarchar(120)	For external tables over a SHARD_MAP_MANAGER external data source, this is the distribution type displayed as a string.	
sharding_column_id	int	For external tables over a SHARD_MAP_MANAGER external data source and a sharded distribution, this is the column ID of the column that contains the sharding key values.	
remote_schema_name	sysname	For external tables over a SHARD_MAP_MANAGER external data source, this is the schema where the base table is located on the remote databases (if different from the schema where the external table is defined).	
remote_object_name	sysname	For external tables over a SHARD_MAP_MANAGER external data source, this is the name of the base table on the remote databases (if different from the name of the external table).	

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[sys.external_file_formats](#) (Transact-SQL)

[sys.external_data_sources](#) (Transact-SQL)

[CREATE EXTERNAL TABLE](#) (Transact-SQL)

sys.external_data_sources (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each external data source in the current database for SQL Server, SQL Database, and SQL Data Warehouse.

Contains a row for each external data source on the server for Parallel Data Warehouse.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
data_source_id	int	Object ID for the external data source.	
name	sysname	Name of the external data source.	
location	nvarchar(4000)	The connection string, which includes the protocol, IP address, and port for the external data source.	
type_desc	nvarchar(255)	Data source type displayed as a string.	HADOOP, RDBMS, SHARD_MAP_MANAGER, RemoteDataArchiveTypeExt DataSource
type	tinyint	Data source type displayed as a number.	0 - HADOOP 1 - RDBMS 2 - SHARD_MAP_MANAGER 3 - RemoteDataArchiveTypeExt DataSource
resource_manager_location	nvarchar(4000)	For type HADOOP, the IP and port location of the Hadoop resource manager. This is used for submitting a job on a Hadoop data source. NULL for other types of external data sources.	
credential_id	int	The object ID of the database scoped credential used to connect to the external data source.	

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
database_name	sysname	For type RDBMS, the name of the remote database. For type, SHARD_MAP_MANAGER, the name of the shard map manager database. NULL for other types of external data sources.	
shard_map_name	sysname	For type SHARD_MAP_MANAGER, the name of the shard map. NULL for other types of external data sources.	

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[sys.external_file_formats](#) (Transact-SQL)

[sys.external_tables](#) (Transact-SQL)

[CREATE EXTERNAL DATA SOURCE](#) (Transact-SQL)

sys.external_file_formats (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each external file format in the current database for SQL Server, SQL Database, and SQL Data Warehouse.

Contains a row for each external file format on the server for Parallel Data Warehouse.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
file_format_id	int	Object ID for the external file format.	
name	sysname	Name of the file format. In SQL Server and SQL Data Warehouse, this is unique for the database. In Parallel Data Warehouse, this is unique for the server.	
format_type	tinyint	The file format type.	DELIMITEDTEXT, RCFILE, ORC, PARQUET
field_terminator	nvarchar(10)	For format_type = DELIMITEDTEXT, this is the field terminator.	
string_delimiter	nvarchar(10)	For format_type = DELIMITEDTEXT, this is the string delimiter.	
date_format	nvarchar(50)	For format_type = DELIMITEDTEXT, this is the user-defined date and time format.	
use_type_default	bit	For format_type = DELIMITED TEXT, specifies how to handle missing values when PolyBase is importing data from HDFS text files into SQL Data Warehouse.	0 – store missing values as the string 'NULL'. 1 – store missing values as the column default value.
serde_method	nvarchar(255)	For format_type = RCFILE, this is the serialization/deserialization method.	

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
row_terminator	nvarchar(10)	For format_type = DELIMITEDTEXT, this is the character string that terminates each row in the external Hadoop file.	Always '\n'.
encoding	nvarchar(10)	For format_type = DELIMITEDTEXT, this is the encoding method for the external Hadoop file.	Always 'UTF8'.
data_compression	nvarchar(255)	The data compression method for the external data.	<p>For format_type = DELIMITEDTEXT:</p> <ul style="list-style-type: none"> - 'org.apache.hadoop.io.compress.DefaultCodec' - 'org.apache.hadoop.io.compress.GzipCodec' <p>For format_type = RCFILE:</p> <ul style="list-style-type: none"> - 'org.apache.hadoop.io.compress.DefaultCodec' <p>For format_type = ORC:</p> <ul style="list-style-type: none"> - 'org.apache.hadoop.io.compress.DefaultCodec' - 'org.apache.hadoop.io.compress.SnappyCodec' <p>For format_type = PARQUET:</p> <ul style="list-style-type: none"> - 'org.apache.hadoop.io.compress.GzipCodec' - 'org.apache.hadoop.io.compress.SnappyCodec'

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[sys.external_data_sources \(Transact-SQL\)](#)

[sys.external_tables \(Transact-SQL\)](#)

[CREATE EXTERNAL FILE FORMAT \(Transact-SQL\)](#)

Filestream and FileTable Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section describes the catalog views related to the FileTable feature.

[sys.database_filestream_options](#) (Transact-SQL)

Displays information about the level of non-transactional access to FILESTREAM data in FileTables that is enabled. Contains one row for each database in the SQL Server instance.

[sys.filetable_system_defined_objects](#) (Transact-SQL)

Displays a list of the system-defined objects that are related to FileTables. Contains one row for each system-defined object.


[sys.filetables](#) (Transact-SQL)

Returns a row for each FileTable. Inherits from **sys.tables**.

For more information about FileTables, see [FileTables \(SQL Server\)](#).

sys.database_filestream_options (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays information about the level of non-transactional access to FILESTREAM data in FileTables that is enabled. Contains one row for each database in the SQL Server instance.

For more information about FileTables, see [FileTables \(SQL Server\)](#).

COLUMN	TYPE	DESCRIPTION
database_id	int	The ID of the database. This value is unique within the SQL Server instance.
directory_name	nvarchar(255)	The database-level directory for all FileTable namespaces.
non_transacted_access	tinyint	<p>The level of non-transactional access to FILESTREAM data that is enabled. The level of access is set by the NON_TRANSACTED_ACCESS option of the CREATE DATABASE or ALTER DATABASE statement.</p> <p>This setting has one of the following values:</p> <p>0 – Not enabled. This is the default value. This level is set by providing the value OFF for the NON_TRANSACTED_ACCESS option.</p> <p>1 – Read-only access. This level is set by providing the value READ_ONLY for the NON_TRANSACTED_ACCESS option.</p> <p>3 – Full access. This level is set by providing the value FULL for the NON_TRANSACTED_ACCESS option.</p> <p>5 - In transition to READONLY</p> <p>6 – In transition to OFF</p>





COLUMN	TYPE	DESCRIPTION
non_transacted_access_desc	nvarchar(60)	<p>The description of the level of non-transactional access identified in non_transacted_access.</p> <p>This setting has one of the following values:</p> <p>NONE – This is the default value.</p> <p>READ_ONLY</p> <p>FULL</p> <p>IN_TRANSITION_TO_READ_ONLY</p> <p>IN_TRANSITION_TO_OFF</p>

See Also

[Enable the Prerequisites for FileTable](#)

sys.filetable_system_defined_objects (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays a list of the system-defined objects that are related to FileTables. Contains one row for each system-defined object.

When you create a FileTable, related objects such as constraints and indexes are created at the same time. You cannot alter or drop these objects; they disappear only when the FileTable itself is dropped.

For more information about FileTables, see [FileTables \(SQL Server\)](#).

COLUMN	DATA TYPE	DESCRIPTION
object_id	int	Object ID of the system-defined object related to a FileTable. References the object in sys.objects .
parent_object_id	int	Object ID of the parent FileTable. References the object in sys.objects .





See Also

[Create, Alter, and Drop FileTables](#)

[Manage FileTables](#)

sys.filetables (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each FileTable in SQL Server 2017. For more information about FileTables, see [FileTables \(SQL Server\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id		Object identification number. Is unique within a database. For more information, sys.objects (Transact-SQL) .
is_enabled	bit	1 = FileTable is in 'enabled' state.
directory_name	varchar(255)	Name of the root directory for a FileTable.
filename_collation_id		Is the collation identifier defined for the FileTable
filename_collation_name		Is the collation name defined for the FileTable.





See Also

[Manage FileTables](#)

[FileTables \(SQL Server\)](#)

Full-Text Search and Semantic Search Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section describes the catalog views that provide information about full-text indexes and semantic indexes.

Full-Text Search Catalog Views

[sys.fulltext_catalogs](#)

Contains a row for each full-text catalog.

[sys.fulltext_document_types](#)

Returns a row for each document type that is available for full-text indexing operations. Each row represents the **IFilter** interface that is registered in the instance of SQL Server.

[sys.fulltext_index_catalog_usages](#)

Returns a row for each full-text catalog to full-text index reference.

[sys.fulltext_index_columns](#)

Contains a row for each column that is part of a full-text index.

[sys.fulltext_index_fragments](#)

Contains a row for each full-text index fragment in every table that contains a full-text index.

[sys.fulltext_indexes](#)

Contains a row per full-text index of a tabular object.

[sys.fulltext_languages](#)

Contains one row per language whose word breakers are registered with SQL Server. Each row displays the LCID and name of the language.

[sys.fulltext_stoplists](#)

Contains a row per full-text stoplist in the database.

[sys.fulltext_stopwords](#)

Contains a row per stopword for all stoplists in the database.

[sys.fulltext_system_stopwords](#)

Provides access to the system stoplist.

[sys.registered_search_properties](#) (Transact-SQL)

Contains a row for each search property contained by any search property list on the current database.

[sys.registered_search_property_lists](#) (Transact-SQL)

Contains a row for each search property list on the current database.

Semantic Search Catalog Views

[sys.fulltext_semantic_language_statistics_database](#) (Transact-SQL)

Returns a row about the semantic language statistics database installed on the current instance of SQL Server.

[sys.fulltext_semantic_languages \(Transact-SQL\)](#)

Returns a row for each language whose statistics model is registered with the instance of SQL Server. When a language model is registered, that language is enabled for semantic indexing.

See Also



[System Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Full-Text Search and Semantic Search Dynamic Management Views and Functions \(Transact-SQL\)](#)

sys.fulltext_catalogs (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each full-text catalog.

NOTE

The following columns will be removed in a future release of SQL Server: **data_space_id**, **file_id**, and **path**. Do not use these columns in new development work, and modify applications that currently use any of these columns as soon as possible.

COLUMN NAME	DATA TYPE	DESCRIPTION
fulltext_catalog_id	int	ID of the full-text catalog. Is unique across the full-text catalogs in the database.
name	sysname	Name of the catalog. Is unique within the database.
path	nvarchar(260)	Name of the catalog directory in the file system.
is_default	bit	The default full-text catalog. True = Is default. False = Is not default.
is_accent_sensitivity_on	bit	Accent-sensitivity setting of the catalog. True = Is accent-sensitive. False = Is not accent-sensitive.
data_space_id	int	Filegroup where this catalog was created.
file_id	int	File ID of the full-text file associated with the catalog.
principal_id	int	ID of the database principal that owns the full-text catalog.
is_importing	bit	Indicates whether the full-text catalog is being imported: 1 = The catalog is being imported. 2 = The catalog is not being imported.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

[CREATE FULLTEXT CATALOG \(Transact-SQL\)](#)

[ALTER FULLTEXT CATALOG \(Transact-SQL\)](#)

[DROP FULLTEXT CATALOG \(Transact-SQL\)](#)

sys.fulltext_document_types (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each document type that is available for full-text indexing operations. Each row represents the IFilter interface that is registered in the instance of SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
document_type	sysname	The file extension of the supported document type. This value can be used to identify the filter that will be used during full-text indexing of columns of type varbinary(max) or image .
class_id	uniqueidentifier	GUID of the IFilter class that supports file extension.
path	nvarchar(260)	The path to the IFilter DLL. The path is only visible to members of the serveradmin fixed server role.
version	sysname	Version of the IFilter DLL.
manufacturer	sysname	Name of the IFilter manufacturer. Note: Only documents with the manufacturer as Microsoft are supported on SQL Database.

Permissions


The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.fulltext_index_catalog_usages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each full-text catalog to full-text index reference.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the full-text indexed table. Is unique within the database.
index_id	int	ID of full-text index.
fulltext_catalog_id	int	ID of full-text catalog.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

[Data Spaces \(Transact-SQL\)](#)

sys.fulltext_index_columns (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column that is part of a full-text index.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object of which this is part.
column_id	int	ID of the column that is part of the full-text index.
type_column_id	int	ID of the type column that stores the user-supplied document file extension —".doc", ".xls", and so forth—of the document in a given row. The type column is specified only for columns whose data requires filtering during full-text indexing. NULL if not applicable. For more information, see Configure and Manage Filters for Search .
language_id	int	LCID of language whose word breaker is used to index this full-text column. 0 = Neutral. For more information, see sys.fulltext_languages (Transact-SQL) .
statistical_semantics	int	1 = This column has statistical semantics enabled in addition to full-text indexing.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.fulltext_index_fragments (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

A fulltext index uses internal tables called *full-text index fragments* to store the inverted index data. This view can be used to query the metadata about these fragments. This view contains a row for each full-text index fragment in every table that contains a full-text index.

COLUMN NAME	DATA TYPE	DESCRIPTION
table_id	int	Object ID of the table that contains the full-text index fragment.
fragment_object_id	int	Object ID of the internal table associated with the fragment.
fragment_id	int	Logical ID of the full-text index fragment. This is unique across all fragments for this table.
timestamp	timestamp	Timestamp associated with the fragment creation. The timestamps of more recent fragments are larger than the timestamps of older fragments.
data_size	int	Logical size of the fragment in bytes.
row_count	int	Number of individual rows in the fragment.
status	int	Status of the fragment, one of: 0 = Newly created and not yet used 1 = Being used for insert during fulltext index population or merge 4 = Closed. Ready for query 6 = Being used for merge input and ready for query 8 = Marked for deletion. Will not be used for query and merge source. A status of 4 or 6 means that the fragment is part of the logical full-text index and can be queried; that is, it is a <i>queryable</i> fragment.

Remarks

The sys.fulltext_index_fragments catalog view can be used to query the number of fragments comprising a full-text

index. If you are experiencing slow full-text query performance, you can use `sys.fulltext_index_fragments` to query for the number of queryable fragments (status = 4 or 6) in the full-text index, as follows:

```
SELECT table_id, status FROM sys.fulltext_index_fragments
WHERE status=4 OR status=6;
```

If many queryable fragments exist, Microsoft recommends that you reorganize the full-text catalog that contains the full-text index to merge the fragments together. To reorganize a of full-text catalog use [ALTER FULLTEXT CATALOG](#)*catalog_name* REORGANIZE. For example, to reorganize a full-text catalog named `ftCatalog` in the `AdventureWorks2012` database, enter:

```
USE AdventureWorks2012;
GO
ALTER FULLTEXT CATALOG ftCatalog REORGANIZE;
GO
```

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Populate Full-Text Indexes](#)

sys.fulltext_indexes (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per full-text index of a tabular object.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this full-text index belongs.
unique_index_id	int	ID of the corresponding unique, non-full-text index that is used to relate the full-text index to the rows.
fulltext_catalog_id	int	ID of the full-text catalog in which the full-text index resides.
is_enabled	bit	1 = Full-text index is currently enabled.
change_tracking_state	char(1)	State of change-tracking. M = Manual A = Auto O = Off
change_tracking_state_desc	nvarchar(60)	Description of the state of change-tracking. MANUAL AUTO OFF
has_crawl_completed	bit	Last crawl (population) that the full-text index has completed.
crawl_type	char(1)	Type of the current or last crawl. F = Full crawl I = Incremental, timestamp-based crawl U = Update crawl, based on notifications P = Full crawl is paused.

COLUMN NAME	DATA TYPE	DESCRIPTION
crawl_type_desc	nvarchar(60)	Description of the current or last crawl type. FULL_CRAWL INCREMENTAL_CRAWL UPDATE_CRAWL PAUSED_FULL_CRAWL
crawl_start_date	datetime	Start of the current or last crawl. NULL = None.
crawl_end_date	datetime	End of the current or last crawl. NULL = None.
incremental_timestamp	binary(8)	Timestamp value to use for the next incremental crawl. NULL = None.
stoplist_id	int	ID of the stoplist that is associated with this full-text index.
data_space_id	int	Filegroup where this full-text index resides.
property_list_id	int	ID of the search property list that is associated with this full-text index. NULL indicates that no search property list is associated with the full-text index. To obtain more information about this search property list, use the sys.registered_search_property_lists (Transact-SQL) catalog view.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

Examples

The following example uses a full-text index on the `HumanResources.JobCandidate` table of the `AdventureWorks2012` sample database. The example returns the object ID of the table, the search property list ID, and the stoplist ID of the stoplist used by the full-text index.

NOTE

For the code example that creates this full-text index, see the "Examples" section of [CREATE FULLTEXT INDEX \(Transact-SQL\)](#).





```
USE AdventureWorks2012;  
GO  
SELECT object_id, property_list_id, stoplist_id FROM sys.fulltext_indexes  
       where object_id = object_id('HumanResources.JobCandidate');  
GO
```

See Also

[sys.fulltext_index_fragments \(Transact-SQL\)](#)
[sys.fulltext_index_columns \(Transact-SQL\)](#)
[sys.fulltext_index_catalog_usages \(Transact-SQL\)](#)
[Object Catalog Views \(Transact-SQL\)](#)
[Catalog Views \(Transact-SQL\)](#)
[Create and Manage Full-Text Indexes](#)
[DROP FULLTEXT INDEX \(Transact-SQL\)](#)
[CREATE FULLTEXT INDEX \(Transact-SQL\)](#)
[ALTER FULLTEXT INDEX \(Transact-SQL\)](#)

sys.fulltext_languages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains one row per language whose word breakers are registered with SQL Server. Each row displays the LCID and name of the language. When word breakers are registered for a language, its other linguistic resources—stemmers, noise words (stopwords), and thesaurus files—become available to full-text indexing/querying operations. The value of **name** or **lcid** can be specified in the full-text queries and full-text index Transact-SQL statements.

COLUMN	DATA TYPE	DESCRIPTION
lcid	int	Microsoft Windows locale identifier (LCID) for the language.
name	sysname	Is either the value of the alias in sys.syslanguages corresponding to the value of lcid or the string representation of the numeric LCID.

Values Returned for Default Languages

The following table shows values for the languages whose word breakers are registered by default.

LANGUAGE	LCID
Arabic	1025
Bengali (India)	1093
British English	2057
Bulgarian	1026
Catalan	1027
Chinese (Hong Kong SAR, PRC)	3076
Chinese (Macao SAR)	5124
Chinese (Singapore)	4100
Croatian	1050
Czech	1029
Danish	1030

LANGUAGE	LCID
Dutch	1043
English	1033
French	1036
German	1031
Applies to: SQL Server 2012 through SQL Server 2017. Greek	1032
Gujarati	1095
Hebrew	1037
Hindi	1081
Icelandic	1039
Indonesian	1057
Italian	1040
Japanese	1041
Kannada	1099
Korean	1042
Latvian	1062
Lithuanian	1063
Malay - Malaysia	1086
Malayalam	1100
Marathi	1102
Neutral	0
Norwegian (Bokmål)	1044
Applies to: SQL Server 2012 through SQL Server 2017. Polish	1045
Portuguese (Brazil)	1046
Portuguese (Portugal)	2070

LANGUAGE	LCID
Punjabi	1094
Romanian	1048
Russian	1049
Serbian (Cyrillic)	3098
Serbian (Latin)	2074
Simplified Chinese	2052
Slovak	1051
Slovenian	1060
Spanish	3082
Swedish	1053
Tamil	1097
Telugu	1098
Thai	1054
Traditional Chinese	1028
Applies to: SQL Server 2012 through SQL Server 2017.	1055
Turkish	
Ukrainian	1058
Urdu	1056
Vietnamese	1066

Remarks

To update the list of languages registered with full-text search, use [sp_fulltext_service](#)'**update_languages**'.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[sp_fulltext_load_thesaurus_file](#) (Transact-SQL)

[sp_fulltext_service](#) (Transact-SQL)

[Configure and Manage Word Breakers and Stemmers for Search](#)
[Configure and Manage Thesaurus Files for Full-Text Search](#)
[Configure and Manage Stopwords and Stoplists for Full-Text Search](#)
[Upgrade Full-Text Search](#)

sys.fulltext_semantic_language_statistics_database (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row about the semantic language statistics database installed on the current instance of SQL Server.

You can query this view to find out about the semantic language statistics component required for semantic processing.

Column name	Type	Description
database_id	int	ID of the database, unique within an instance of SQL Server.
register_date	datetime	Date the database was registered for semantic processing.
registered_by	int	ID of the server principal that registered the database for semantic processing.
version	nvarchar(128)	The latest version information specific to the semantic language statistics database.

General Remarks

For more information, see [Install and Configure Semantic Search](#).

Metadata

For information about the languages that are supported for semantic indexing, query the catalog view [sys.fulltext_semantic_languages](#) (Transact-SQL).

Security

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

Examples

The following example shows how to query **sys.fulltext_semantic_language_statistics_database** to get information about the semantic language statistics database registered on the current instance of SQL Server.





```
SELECT * FROM sys.fulltext_semantic_language_statistics_database;  
GO
```

See Also

[Install and Configure Semantic Search](#)

sys.fulltext_semantic_languages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each language whose statistics model is registered with the instance of SQL Server. When a language model is registered, that language is enabled for semantic indexing.

This catalog view is similar to [sys.fulltext_languages \(Transact-SQL\)](#).

Column name	Type	Description
lcid	int	Microsoft Windows locale identifier (LCID) for the language.
name	sysname	Is either the value of the alias in sys.syslanguages (Transact-SQL) corresponding to the value of lcid , or the string representation of the numeric LCID.

General Remarks

For more information, see [Install and Configure Semantic Search](#).

Metadata

For more information about the semantic language statistics database that is installed to support semantic indexing, query the catalog view [sys.fulltext_semantic_language_statistics_database \(Transact-SQL\)](#).

Security

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

Examples

The following example shows how to query **sys.fulltext_semantic_languages** to get information about all the language models registered for semantic indexing on the current instance of SQL Server.





```
SELECT * FROM sys.fulltext_semantic_languages;  
GO
```

See Also

[Install and Configure Semantic Search](#)

sys.fulltext_stoplists (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per full-text stoplist in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
stoplist_id	int	ID of the stoplist, unique within the database.
name	sysname	Name of the stoplist.
create_date	datetime	Date that stoplist was created.
modify_date	datetime	Date that stoplist was last modified using any ALTER statement.
Principal_id	int	ID of the database principal that owns the stoplist.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[sys.fulltext_system_stopwords \(Transact-SQL\)](#)

[sys.fulltext_stopwords \(Transact-SQL\)](#)

[Configure and Manage Stopwords and Stoplists for Full-Text Search](#)





[CREATE FULLTEXT STOPLIST \(Transact-SQL\)](#)

[ALTER FULLTEXT STOPLIST \(Transact-SQL\)](#)

[DROP FULLTEXT STOPLIST \(Transact-SQL\)](#)

sys.fulltext_stopwords (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per stopword for all stoplists in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
stoplist_id	int	ID of the stoplist to which stopword belongs. This ID is unique within the database.
stopword	nvarchar(64)	The term to be considered for a stop-word match.
language	sysname	Is either the value of the alias in sys.fulltext_languages corresponding to the value of the locale identifier (LCID), or is the string representation of the numeric LCID.
language_id	int	LCID used for word breaking.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)





[Configure and Manage Stopwords and Stoplists for Full-Text Search](#)

[sys.fulltext_stoplists \(Transact-SQL\)](#)

[sys.fulltext_system_stopwords \(Transact-SQL\)](#)

sys.fulltext_system_stopwords (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides access to the system stoplist.

COLUMN NAME	DATA TYPE	DESCRIPTION
stopword	nvarchar(64)	The term that is considered for a stop-word match.
language_id	int	Locale identifier (LCID) of the language. This LCID is used for word breaking.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[sys.fulltext_stoplists \(Transact-SQL\)](#)

[sys.fulltext_stopwords \(Transact-SQL\)](#)

[Configure and Manage Stopwords and Stoplists for Full-Text Search](#)

sys.registered_search_properties (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each search property contained by any search property list on the current database.

COLUMN NAME	DATA TYPE	DESCRIPTION
property_list_id	int	ID of the search property list to which this property belongs.
property_set_guid	uniqueidentifier	Globally unique identifier GUID that identifies the property set to which the search property belongs.
property_int_id	int	Integer that identifies this search property within the property set. property_int_id is unique within the property set.
property_name	nvarchar(64)	Name that uniquely identifies this search property in the search property list. Note: To search on a property, specify this property name in the CONTAINS predicate.
property_description	nvarchar(512)	Description of the property.

COLUMN NAME	DATA TYPE	DESCRIPTION
property_id	int	<p>Internal property ID of the search property within the search property list identified by the property_list_id value.</p> <p>When a given property is added to a given search property list, the Full-Text Engine registers the property and assigns it an internal property ID that is specific to that property list. The internal property ID, which is an integer, is unique to a given search property list. If a given property is registered for multiple search property lists, a different internal property ID might be assigned for each search property list.</p> <p>Note: The internal property ID is distinct from the property integer identifier that is specified when adding the property to the search property list. For more information, see Search Document Properties with Search Property Lists.</p> <p>To view all property-related content in the full-text index: sys.dm_fts_index_keywords_by_property (Transact-SQL)</p>

Remarks

For more information about search property lists, see [Search Document Properties with Search Property Lists](#).

Permissions

Visibility of the metadata for search properties is limited to those that are in search property lists that you either own or on which you have been granted some REFERENCE permission.

NOTE

The search property list owner can grant REFERENCE or CONTROL permissions on the list. Users with CONTROL permission can also grant REFERENCE permission to other users.

Examples

The following example lists all of the metadata for registered search properties.

```
USE AdventureWorks2012;
GO
SELECT * FROM sys.registered_search_properties;
GO
```





See Also

[ALTER FULLTEXT INDEX \(Transact-SQL\)](#)

[sys.fulltext_indexes \(Transact-SQL\)](#)

sys.registered_search_property_lists (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each search property list on the current database.

COLUMN NAME	DATA TYPE	DESCRIPTION
property_list_id	int	ID of the property list.
name	sysname	Name of the property list.
create_date	datetime	Date the property list was created.
modify_date	datetime	Date the property list was last modified by any ALTER statement.
principal_id	int	Owner of the property list.

Remarks

For more information, see [Search Document Properties with Search Property Lists](#).

Permissions

Visibility of the metadata in search property lists is limited to those that you either own or on which you have been granted some REFERENCE permission.

NOTE

The search property list owner can grant REFERENCE or CONTROL permissions on the list. Users with CONTROL permission can also grant REFERENCE permission to other users.

Examples

The following example displays the ID and name of the search property lists in the AdventureWorks2012 database.

```
USE AdventureWorks2012;
GO
SELECT property_list_id, name FROM sys.registered_search_property_lists;
GO
```




See Also

[ALTER FULLTEXT INDEX \(Transact-SQL\)](#)

[sys.fulltext_indexes \(Transact-SQL\)](#)

Linked Servers Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

This section contains the following catalog views.

sys.linked_logins	sys.servers
sys.remote_logins	


See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.linked_logins (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per linked-server-login mapping, for use by RPC and distributed queries from local server to the corresponding linked server.

COLUMN NAME	DATA TYPE	DESCRIPTION
server_id	int	ID of the server in sys.servers .
local_principal_id	int	Server-principal to whom mapping applies. 0 = wildcard or public.
uses_self_credential	bit	If 1, mapping indicates session should use its own credentials; otherwise, 0 indicates that session uses the name and password that are supplied.
remote_name	sysname	Remote user name to use when connecting. Password is also stored, but not exposed in catalog view interfaces.
modify_date	datetime	Date the linked login was last changed.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Linked Servers Catalog Views \(Transact-SQL\)](#)

sys.remote_logins (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per remote-login mapping. This catalog view is used to map incoming local logins that claim to be coming from a corresponding server to an actual local login.

COLUMN NAME	DATA TYPE	DESCRIPTION
server_id	int	ID of the server in sys.servers . This name is supplied by the connection from the "remote" server.
remote_name	sysname	Login name that the connection will supply to be mapped. If NULL, the login name that is specified in the connection is used.
local_principal_id	int	ID of the server principal to whom the login is mapped. If 0, the remote login is mapped to the login with the same name.
modify_date	datetime	Date the linked login was last changed.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).



See Also

[Linked Servers Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.servers (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per linked or remote server registered, and a row for the local server that has **server_id** = 0.

COLUMN NAME	DATA TYPE	DESCRIPTION
server_id	int	Local ID of linked server.
name	sysname	When server_id = 0, this is the server name. When server_id > 0, this is the local name of linked server.
product	sysname	Product name of the linked server. "SQL Server" indicates this is another instance of SQL Server.
provider	sysname	OLE DB provider name for connecting to linked server.
data_source	nvarchar(4000)	OLE DB data source connection property.
location	nvarchar(4000)	OLE DB location connection property. NULL if none.
provider_string	nvarchar(4000)	OLE DB provider-string connection property. Is NULL unless the caller has the ALTER ANY LINKED SERVER permission.
catalog	sysname	OleDb catalog connection property. NULL if none.
connect_timeout	int	Connect time-out in seconds, 0 if none.
query_timeout	int	Query time-out in seconds, 0 if none.
is_linked	bit	0 = Is an old-style server added by using sp_addserver , with different RPC and distributed-transaction behavior. 1 = Standard linked server.
is_remote_login_enabled	bit	RPC option is set enabling incoming remote logins for this server.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_rpc_out_enabled	bit	Outgoing (from this server) RPC is enabled.
is_data_access_enabled	bit	Server is enabled for distributed queries.
is_collation_compatible	bit	Collation of remote data is assumed to be compatible with local data if no collation information is available.
uses_remote_collation	bit	If 1, use the collation reported by the remote server; otherwise, use the collation specified by the next column.
collation_name	sysname	Name of collation to use, or NULL if just use local.
lazy_schema_validation	bit	If 1, schema validation is not checked at query startup.
is_system	bit	This server can be accessed only by the internal system.
is_publisher	bit	Server is a replication Publisher.
is_subscriber	bit	Server is a replication Subscriber.
is_distributor	bit	Server is a replication Distributor.
is_nonsql_subscriber	bit	Server is a non-SQL Server replication Subscriber.
is_remote_proc_transaction_promotion_enabled	bit	If 1, calling a remote stored procedure starts a distributed transaction and enlists the transaction with MS DTC. For more information, see sp_serveroption (Transact-SQL) .
modify_date	datetime	Date that server information was last changed.

Permissions

The value in **provider_string** is always NULL unless the caller has the ALTER ANY LINKED SERVER permission.

Permissions are not required to view the local server (**server_id** = 0).

When you create a linked or remote server, SQL Server creates a default login mapping to the **public** server role. This means that by default, all logins can view all linked and remote servers. To restrict visibility to these servers, remove the default login mapping by executing [sp_droplinkedserverlogin](#) and specifying NULL for the *locallogin* parameter.

If the default login mapping is deleted, only users that have been explicitly added as a linked login or remote login can view the linked or remote servers for which they have a login. To view all linked and remote servers after the default login mapping is deleted requires the following permissions:

- ALTER ANY LINKED SERVER or ALTER ANY LOGIN ON SERVER
- Membership in the **setupadmin** or **sysadmin** fixed server roles

See Also

[Catalog Views \(Transact-SQL\)](#)





[Linked Servers Catalog Views \(Transact-SQL\)](#)

[sp_addlinkedsrvlogin \(Transact-SQL\)](#)

[sp_addremotelogin \(Transact-SQL\)](#)

Object Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.





sys.allocation_units	sys.objects
sys.assembly_modules	sys.parameters
sys.check_constraints	sys.partitions
sys.columns	sys.periods
sys.computed_columns	sys.procedures
sys.default_constraints	sys.sequences
sys.events	sys.service_queues
sys.event_notifications	sys.sql_dependencies
sys.extended_procedures	sys.sql_expression_dependencies
sys.foreign_key_columns	sys.sql_modules
sys.foreign_keys	sys.stats
sys.function_order_columns	sys.stats_columns
sys.hash_indexes	sys.synonyms
sys.identity_columns	sys.table_types
sys.index_columns	sys.tables
sys.trigger_event_types	sys.key_constraints
sys.trigger_events	sys.masked_columns
sys.triggers	sys.memory_optimized_tables_internal_attributes
sys.views	sys.numbered_procedure_parameters
sys.numbered_procedures	

See Also

[System Catalog Views](#) [System Views \(Transact-SQL\)](#)

sys.all_columns (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows the union of all columns belonging to user-defined objects and system objects.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this column belongs.
name	sysname	Name of the column. Is unique within the object.
column_id	int	ID of the column. Is unique within the object. Column IDs might not be sequential.
system_type_id	tinyint	ID of the system-type of the column.
user_type_id	int	ID of the type of the column as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length (in bytes) of the column. -1 = Column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml . For text columns, the max_length value will be 16 or the value set by sp_tableoption 'text in row'.
precision	tinyint	Precision of the column if numeric-based; otherwise, 0.
scale	tinyint	Scale of the column if numeric-based; otherwise, 0.
collation_name	sysname	Name of the collation of the column if character-based; otherwise, NULL.
is_nullable	bit	1 = Column is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_ansi_padded	bit	1 = Column uses ANSI_PADDING ON behavior if character, binary, or variant. 0 = Column is not character, binary, or variant.
is_rowguidcol	bit	1 = Column is a declared ROWGUIDCOL.
is_identity	bit	1 = Column has identity values
is_computed	bit	1 = Column is a computed column.
is_filestream	bit	1 = Column is declared to use filestream storage.
is_replicated	bit	1 = Column is replicated.
is_non_sql_subscribed	bit	1 = Column has a non- SQL Server subscriber.
is_merge_published	bit	1 = Column is merge-published.
is_dts_replicated	bit	1 = Column is replicated by using SSIS.
is_xml_document	bit	1 = Content is a complete XML document. 0 = Content is a document fragment, or the column data type is not XML.
xml_collection_id	int	Non-zero if the column's data type is xml and the XML is typed. The value will be the ID of the collection containing the column's validating XML schema namespace 0 = no XML schema collection.
default_object_id	int	ID of the default object, regardless of whether it is a stand-alone sys.sp_bindefault , or an in-line, column-level DEFAULT constraint. The parent_object_id column of an inline column-level default object is a reference back to the table itself. 0 = No default.

COLUMN NAME	DATA TYPE	DESCRIPTION
rule_object_id	int	<p>ID of the stand-alone rule bound to the column by using sys.sp_bindrule.</p> <p>0 = No stand-alone rule.</p> <p>For column-level CHECK constraints, see sys.check_constraints (Transact-SQL).</p>
is_sparse	bit	<p>1 = Column is a sparse column. For more information, see Use Sparse Columns.</p>
is_column_set	bit	<p>1 = Column is a column set. For more information, see Use Column Sets.</p>
generated_always_type	tinyint	<p>Applies to: SQL Server 2016 through SQL Server 2017.</p> <p>The numeric value representing the type of column:</p> <p>0 = NOT_APPLICABLE</p> <p>1 = AS_ROW_START</p> <p>2 = AS_ROW_END</p>
generated_always_type_desc	nvarchar(60)	<p>Applies to: SQL Server 2016 through SQL Server 2017.</p> <p>The text description of the type of column:</p> <p>NOT_APPLICABLE</p> <p>AS_ROW_START</p> <p>AS_ROW_END</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)





[sys.columns \(Transact-SQL\)](#)

[sys.system_columns \(Transact-SQL\)](#)

[sys.computed_columns \(Transact-SQL\)](#)

sys.all_objects (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows the UNION of all schema-scoped user-defined objects and system objects.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Object name.
object_id	int	Object identification number. Is unique within a database.
principal_id	int	<p>ID of the individual owner if different from the schema owner. By default, schema-contained objects are owned by the schema owner. However, another owner can be specified by using the ALTER AUTHORIZATION statement to change ownership.</p> <p>Is NULL if there is no alternative individual owner.</p> <p>Is NULL if the object type is one of the following:</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>TA = Assembly (CLR) trigger</p> <p>TR = SQL trigger</p> <p>UQ = UNIQUE constraint</p>
schema_id	int	<p>ID of the schema that contains the object.</p> <p>For all schema scoped system objects that are included with SQL Server, this value is always in (schema_id('sys'), schema_id('INFORMATION_SCHEMA')).</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
parent_object_id	int	<p>ID of the object to which this object belongs.</p> <p>0 = Not a child object.</p>
type	char(2)	<p>Object type:</p> <p>AF = Aggregate function (CLR)</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>FN = SQL scalar function</p> <p>FS = Assembly (CLR) scalar-function</p> <p>FT = Assembly (CLR) table-valued function</p> <p>IF = SQL inline table-valued function</p> <p>IT = Internal table</p> <p>P = SQL Stored Procedure</p> <p>PC = Assembly (CLR) stored-procedure</p> <p>PG = Plan guide</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>RF = Replication-filter-procedure</p> <p>S = System base table</p> <p>SN = Synonym</p> <p>SQ = Service queue</p> <p>TA = Assembly (CLR) DML trigger</p> <p>TF = SQL table-valued-function</p> <p>TR = SQL DML trigger</p> <p>TT = Table type</p> <p>U = Table (user-defined)</p> <p>UQ = UNIQUE constraint</p> <p>V = View</p> <p>X = Extended stored procedure</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	Description of the object type. AGGREGATE_FUNCTION CHECK_CONSTRAINT DEFAULT_CONSTRAINT FOREIGN_KEY_CONSTRAINT SQL_SCALAR_FUNCTION CLR_SCALAR_FUNCTION CLR_TABLE_VALUED_FUNCTION SQL_INLINE_TABLE_VALUED_FUNCTION INTERNAL_TABLE SQL_STORED_PROCEDURE CLR_STORED_PROCEDURE PLAN_GUIDE PRIMARY_KEY_CONSTRAINT RULE REPLICATION_FILTER_PROCEDURE SYSTEM_TABLE SYNONYM SERVICE_QUEUE CLR_TRIGGER SQL_TABLE_VALUED_FUNCTION SQL_TRIGGER TABLE_TYPE USER_TABLE UNIQUE_CONSTRAINT VIEW EXTENDED_STORED_PROCEDURE
create_date	datetime	Date the object was created.

COLUMN NAME	DATA TYPE	DESCRIPTION
modify_date	datetime	Date the object was last modified by using an ALTER statement. If the object is a table or a view, modify_date also changes when a clustered index on the table or view is created or modified.
is_ms_shipped	bit	Object created by an internal SQL Server component.
is_published	bit	Object is published.
is_schema_published	bit	Only the schema of the object is published.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[sys.objects \(Transact-SQL\)](#)

[sys.system_objects \(Transact-SQL\)](#)

sys.all_parameters (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows the union of all parameters that belong to user-defined or system objects.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this parameter belongs.
name	sysname	Name of parameter. Is unique within the object. If the object is a scalar function, the parameter name is an empty string in the row representing the return value.
parameter_id	int	ID of parameter. Is unique within the object. If the object is a scalar function, parameter_id = 0 represents the return value.
system_type_id	tinyint	ID of the system type of the parameter.
user_type_id	int	ID of the type of the parameter as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length of the parameter, in bytes. -1 = Column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml .
precision	tinyint	Precision of the parameter if it is numeric-based; otherwise, 0.
scale	tinyint	Scale of the parameter if it is numeric-based; otherwise, 0.
is_output	bit	1 = Parameter is output (or return); otherwise, 0.
is_cursor_ref	bit	1 = Parameter is a cursor reference parameter.

COLUMN NAME	DATA TYPE	DESCRIPTION
has_default_value	bit	<p>1 = Parameter has a default value.</p> <p>SQL Server only maintains default values for CLR objects in this catalog view; therefore, this column will always have a value of 0 for Transact-SQL objects. To view the default value of a parameter in a Transact-SQL object, query the definition column of the sys.sql_modules catalog view, or use the OBJECT_DEFINITION system function.</p>
is_xml_document	bit	<p>1 = Content is a complete XML document.</p> <p>0 = Content is a document fragment or the data type of the column is not xml.</p>
default_value	sql_variant	<p>If has_default_value is 1, the value of this column is the value of the default for the parameter; otherwise, NULL.</p>
xml_collection_id	int	<p>Is the ID of the XML schema collection used to validate the parameter.</p> <p>Nonzero if the data type of the parameter is xml and the XML is typed.</p> <p>0 = There is no XML schema collection, or the parameter is not XML.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)





[Querying the SQL Server System Catalog FAQ](#)

[sys.parameters \(Transact-SQL\)](#)

[sys.system_parameters \(Transact-SQL\)](#)

sys.all_sql_modules (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the union of **sys.sql_modules** and **sys.system_sql_modules**.

The view returns a row for each natively compiled, scalar user-defined function. For more information, see [Scalar User-Defined Functions for In-Memory OLTP](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object of the containing object. Is unique within a database.
definition	nvarchar(max)	SQL text that defines this module. NULL = Encrypted
uses_ansi_nulls	bit	Module was created with SET ANSI_NULLS ON.
uses_quoted_identifier	bit	Module was created with SET QUOTED_IDENTIFIER ON.
is_schema_bound	bit	Module was created with the SCHEMABINDING option.
uses_database_collation	bit	1 = Schema-bound module definition depends on the default-collation of the database for correct evaluation; otherwise, 0. Such a dependency prevents changing the default collation of the database.
is_recompiled	bit	Procedure was created using the WITH RECOMPILE option.
null_on_null_input	bit	Module was declared to produce a NULL output on any NULL input.
execute_as_principal_id	int	ID of the EXECUTE AS database principal. NULL by default or if EXECUTE AS CALLER. ID of the specified principal if EXECUTE AS SELF or EXECUTE AS <principal>. -2 = EXECUTE AS OWNER.

COLUMN NAME	DATA TYPE	DESCRIPTION
uses_native_compilation	bit	<p>Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>0 = not natively compiled</p> <p>1 = is natively compiled</p> <p>The default value is 0.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)





[sys.sql_modules \(Transact-SQL\)](#)

[sys.system_sql_modules \(Transact-SQL\)](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.all_views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows the UNION of all user-defined and system views.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
is_replicated	bit	1 = View is replicated.
has_replication_filter	bit	1 = View has a replication filter.
has_opaque_metadata	bit	1 = VIEW_METADATA option specified for view. For more information, see CREATE VIEW (Transact-SQL) .
has_unchecked_assembly_data	bit	1 = Table contains persisted data that depends on an assembly whose definition changed during the last ALTER ASSEMBLY. Resets to 0 after the next successful DBCC CHECKDB or DBCC CHECKTABLE.
with_check_option	bit	1 = WITH CHECK OPTION was specified in the view definition.
is_date_correlation_view	bit	1 = View was created automatically by the system to store correlation information between datetime columns. Creation of this view was enabled by setting DATE_CORRELATION_OPTIMIZATION to ON .

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)


[DBCC CHECKDB \(Transact-SQL\)](#)

[ALTER ASSEMBLY \(Transact-SQL\)](#)

[DBCC CHECKTABLE \(Transact-SQL\)](#)

sys.allocation_units (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

Contains a row for each allocation unit in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
allocation_unit_id	bigint	ID of the allocation unit. Is unique within a database.
type	tinyint	Type of allocation unit: 0 = Dropped 1 = In-row data (all data types, except LOB data types) 2 = Large object (LOB) data (text , ntext , image , xml , large value types, and CLR user-defined types) 3 = Row-overflow data
type_desc	nvarchar(60)	Description of the allocation unit type: DROPPED IN_ROW_DATA LOB_DATA ROW_OVERFLOW_DATA
container_id	bigint	ID of the storage container associated with the allocation unit. If type = 1 or 3, container_id = sys.partitions.hobt_id. If type is 2, then container_id = sys.partitions.partition_id. 0 = Allocation unit marked for deferred drop
data_space_id	int	ID of the filegroup in which this allocation unit resides.
total_pages	bigint	Total number of pages allocated or reserved by this allocation unit.
used_pages	bigint	Number of total pages actually in use.

COLUMN NAME	DATA TYPE	DESCRIPTION
data_pages	bigint	<p>Number of used pages that have:</p> <p>In-row data</p> <p>LOB data</p> <p>Row-overflow data</p> <p>Note that the value returned excludes internal index pages and allocation-management pages.</p>

NOTE

When you drop or rebuild large indexes, or drop or truncate large tables, the Database Engine defers the actual page deallocations, and their associated locks, until after the transaction commits. Deferred drop operations do not release allocated space immediately. Therefore, the values returned by `sys.allocation_units` immediately after dropping or truncating a large object may not reflect the actual disk space available.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[sys.partitions \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.assembly_modules (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each function, procedure or trigger that is defined by a common language runtime (CLR) assembly. This catalog view maps CLR stored procedures, CLR triggers, or CLR functions to their underlying implementation. Objects of type TA, AF, PC, FS, and FT have an associated assembly module. To find the association between the object and the assembly, you can join this catalog view to other catalog views. For example, when you create a CLR stored procedure, it is represented by one row in **sys.objects**, one row in **sys.procedures** (which inherits from **sys.objects**), and one row in **sys.assembly_modules**. The stored procedure itself is represented by the metadata in **sys.objects** and **sys.procedures**. References to the procedure's underlying CLR implementation are found in **sys.assembly_modules**.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	Object identification number of the SQL object. Is unique within a database.
assembly_id	int	ID of the assembly from which this module was created.
assembly_class	sysname	Name of the class within the assembly that defines this module.
assembly_method	sysname	Name of the method within the assembly_class that defines this module. NULL for aggregate functions (AF).
null_on_null_input	bit	Module was declared to produce a NULL output for any NULL input.
execute_as_principal_id	int	ID of the database principal under which the context execution occurs, as specified by the EXECUTE AS clause of the CLR function, stored procedure, or trigger. NULL = EXECUTE AS CALLER. This is the default. ID of the specified database principal = EXECUTE AS SELF, EXECUTE AS <i>user_name</i> , or EXECUTE AS <i>login_name</i> . -2 = EXECUTE AS OWNER.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.check_constraints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is a CHECK constraint, with **sys.objects.type** = 'C'.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
is_disabled	bit	CHECK constraint is disabled.
is_not_for_replication	bit	CHECK constraint was created with the NOT FOR REPLICATION option.
is_not_trusted	bit	CHECK constraint has not been verified by the system for all rows.
parent_column_id	int	0 indicates a table-level CHECK constraint. Non-zero value indicates that this is a column-level CHECK constraint defined on the column with the specified ID value.
definition	nvarchar(max)	SQL expression that defines this CHECK constraint.
uses_database_collation	bit	1 = The constraint definition depends on the default collation of the database for correct evaluation; otherwise, 0. Such a dependency prevents changing the database default collation.
is_system_named	bit	1 = Name was generated by system. 0 = Name was supplied by the user.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.column_store_dictionaries (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each dictionary used in xVelocity memory optimized columnstore indexes. Dictionaries are used to encode some, but not all data types, therefore not all columns in a columnstore index have dictionaries. A dictionary can exist as a primary dictionary (for all segments) and possibly for other secondary dictionaries used for a subset of the column's segments.

COLUMN NAME	DATA TYPE	DESCRIPTION
hobt_id	bigint	ID of the heap or B-tree index (hobt) for the table that has this columnstore index.
column_id	int	ID of the columnstore column starting with 1. The first column has ID = 1, the second column has ID = 2, etc.
dictionary_id	int	There can be two kinds of dictionaries, global and local, associated with a column segment. A dictionary_id of 0 represents the global dictionary that is shared across all column segments (one for each row group) for that column.
version	int	Version of the dictionary format.
type	int	Dictionary type: 1 – Hash dictionary containing int values 2 – Not used 3 – Hash dictionary containing string values 4 – Hash dictionary containing float values For more information about dictionaries, see Columnstore Indexes Guide .
last_id	int	The last data ID in the dictionary.
entry_count	bigint	Number of entries in the dictionary.
on_disc_size	bigint	Size of dictionary in bytes.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Indicates the partition ID. Is unique within a database.

Permissions

All columns require at least VIEW DEFINITION permission on the table. The following columns return null unless the user also has **SELECT** permission: last_id, entry_count, data_ptr.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[sys.columns \(Transact-SQL\)](#)

[sys.all_columns \(Transact-SQL\)](#)

[sys.computed_columns \(Transact-SQL\)](#)





[Columnstore Indexes Guide](#)

[Columnstore Indexes Guide](#)

[sys.column_store_segments \(Transact-SQL\)](#)

sys.column_store_row_groups (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2014)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides clustered columnstore index information on a per-segment basis to help the administrator make system management decisions. **sys.column_store_row_groups** has a column for the total number of rows physically stored (including those marked as deleted) and a column for the number of rows marked as deleted. Use **sys.column_store_row_groups** to determine which row groups have a high percentage of deleted rows and should be rebuilt.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	The id of the table on which this index is defined.
index_id	int	ID of the index for the table that has this columnstore index.
partition_number	int	ID of the table partition that holds row group row_group_id. You can use partition_number to join this DMV to sys.partitions.
row_group_id	int	The row group number associated with this row group. This is unique within the partition. -1 = tail of an in-memory table.
delta_store_hobt_id	bigint	The hobt_id for OPEN row group in the delta store. NULL if the row group is not in the delta store. NULL for the tail of an in-memory table.
state	tinyint	ID number associated with the state_description. 0 = INVISIBLE 1 = OPEN 2 = CLOSED 3 = COMPRESSED 4 = TOMBSTONE

COLUMN NAME	DATA TYPE	DESCRIPTION
state_description	nvarchar(60)	<p>Description of the persistent state of the row group:</p> <p>INVISIBLE – A hidden compressed segment in the process of being built from data in a delta store. Read actions will use the delta store until the invisible compressed segment is completed. Then the new segment is made visible, and the source delta store is removed.</p> <p>OPEN – A read/write row group that is accepting new records. An open row group is still in rowstore format and has not been compressed to columnstore format.</p> <p>CLOSED – A row group that has been filled, but not yet compressed by the tuple mover process.</p> <p>COMPRESSED – A row group that has filled and compressed.</p>
total_rows	bigint	Total rows physically stored in the row group. Some may have been deleted but they are still stored. The maximum number of rows in a row group is 1,048,576 (hexadecimal FFFFF).
deleted_rows	bigint	Total rows in the row group marked deleted. This is always 0 for DELTA row groups.
size_in_bytes	bigint	Size in bytes of all the data in this row group (not including metadata or shared dictionaries), for both DELTA and COLUMNSTORE rowgroups.

Remarks

Returns one row for each columnstore row group for each table having a clustered or nonclustered columnstore index.

Use **sys.column_store_row_groups** to determine the number of rows included in the row group and the size of the row group.

When the number of deleted rows in a row group grows to a large percentage of the total rows, the table becomes less efficient. Rebuild the columnstore index to reduce the size of the table, reducing the disk I/O required to read the table. To rebuild the columnstore index use the **REBUILD** option of the **ALTER INDEX** statement.

The updateable columnstore first inserts new data into an **OPEN** rowgroup, which is in rowstore format, and is also sometimes referred to as a delta table. Once an open rowgroup is full, its state changes to **CLOSED**. A closed rowgroup is compressed into columnstore format by the tuple mover and the state changes to **COMPRESSED**. The tuple mover is a background process that periodically wakes up and checks whether there are any closed rowgroups that are ready to compress into a columnstore rowgroup. The tuple mover also deallocates any rowgroups in which every row has been deleted. Deallocated rowgroups are marked as **TOMBSTONE**. To run tuple

mover immediately, use the **REORGANIZE** option of the **ALTER INDEX** statement.

When a columnstore row group has filled, it is compressed, and stops accepting new rows. When rows are deleted from a compressed group, they remain but are marked as deleted. Updates to a compressed group are implemented as a delete from the compressed group, and an insert to an open group.

Permissions

Returns information for a table if the user has **VIEW DEFINITION** permission on the table.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example joins the **sys.column_store_row_groups** table to other system tables to return information about specific tables. The calculated `PercentFull` column is an estimate of the efficiency of the row group. To find information on a single table remove the comment hyphens in front of the **WHERE** clause and provide a table name.

```
SELECT i.object_id, object_name(i.object_id) AS TableName,
i.name AS IndexName, i.index_id, i.type_desc,
CSRowGroups.*,
100*(total_rows - ISNULL(deleted_rows,0))/total_rows AS PercentFull
FROM sys.indexes AS i
JOIN sys.column_store_row_groups AS CSRowGroups
    ON i.object_id = CSRowGroups.object_id
   AND i.index_id = CSRowGroups.index_id
--WHERE object_name(i.object_id) = '<table_name>'
ORDER BY object_name(i.object_id), i.name, row_group_id;
```

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[sys.columns \(Transact-SQL\)](#)

[sys.all_columns \(Transact-SQL\)](#)

[sys.computed_columns \(Transact-SQL\)](#)





[Columnstore Indexes Guide](#)

[sys.column_store_dictionaries \(Transact-SQL\)](#)

[sys.column_store_segments \(Transact-SQL\)](#)

sys.column_store_segments (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2014)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each column segment in a columnstore index. There is one column segment per column per rowgroup. For example, a table with 10 rowgroups and 34 columns returns 340 rows.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Indicates the partition ID. Is unique within a database.
hobt_id	bigint	ID of the heap or B-tree index (hobt) for the table that has this columnstore index.
column_id	int	ID of the columnstore column.
segment_id	int	ID of the rowgroup. For backward compatibility, the column name continues to be called <code>segment_id</code> even though this is the rowgroup ID. You can uniquely identify a segment using <code><hobt_id, partition_id, column_id></code> , <code><segment_id></code> .
version	int	Version of the column segment format.
encoding_type	int	Type of encoding used for that segment: 1 = <code>VALUE_BASED</code> - non-string/binary with no dictionary (very similar to 4 with some internal variations) 2 = <code>VALUE_HASH_BASED</code> - non-string/binary column with common values in dictionary 3 = <code>STRING_HASH_BASED</code> - string/binary column with common values in dictionary 4 = <code>STORE_BY_VALUE_BASED</code> - non-string/binary with no dictionary 5 = <code>STRING_STORE_BY_VALUE_BASED</code> - string/binary with no dictionary All encodings take advantage of bit-packing and run-length encoding when possible.
row_count	int	Number of rows in the row group.

COLUMN NAME	DATA TYPE	DESCRIPTION
has_nulls	int	1 if the column segment has null values.
base_id	bigint	Base value id if encoding type 1 is being used. If encoding type 1 is not being used, base_id is set to 1.
magnitude	float	Magnitude if encoding type 1 is being used. If encoding type 1 is not being used, magnitude is set to 1.
primary_dictionary_id	int	A value of 0 represents the global dictionary. A value of -1 indicates that there is no global dictionary created for this column.
secondary_dictionary_id	int	A non-zero value points to the local dictionary for this column in the current segment (i.e. the rowgroup). A value of -1 indicates that there is no local dictionary for this segment.
min_data_id	bigint	Minimum data id in the column segment.
max_data_id	bigint	Maximum data id in the column segment.
null_value	bigint	Value used to represent nulls.
on_disk_size	bigint	Size of segment in bytes.

Remarks

The following query returns information about segments of a columnstore index.

```
SELECT i.name, p.object_id, p.index_id, i.type_desc,
       COUNT(*) AS number_of_segments
FROM sys.column_store_segments AS s
INNER JOIN sys.partitions AS p
      ON s.hobt_id = p.hobt_id
INNER JOIN sys.indexes AS i
      ON p.object_id = i.object_id
WHERE i.type = 5 OR i.type = 6
GROUP BY i.name, p.object_id, p.index_id, i.type_desc ;
GO
```

Permissions

All columns require at least **VIEW DEFINITION** permission on the table. The following columns return null unless the user also has **SELECT** permission: has_nulls, base_id, magnitude, min_data_id, max_data_id, and null_value.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user

has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[sys.columns \(Transact-SQL\)](#)

[sys.all_columns \(Transact-SQL\)](#)





[sys.computed_columns \(Transact-SQL\)](#)

[Columnstore Indexes Guide](#)

[sys.column_store_dictionaries \(Transact-SQL\)](#)

sys.columns (Transact-SQL)

11/27/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each column of an object that has columns, such as views or tables. The following is a list of object types that have columns:

- Table-valued assembly functions (FT)
- Inline table-valued SQL functions (IF)
- Internal tables (IT)
- System tables (S)
- Table-valued SQL functions (TF)
- User tables (U)
- Views (V)

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this column belongs.
name	sysname	Name of the column. Is unique within the object.
column_id	int	ID of the column. Is unique within the object. Column IDs might not be sequential.
system_type_id	tinyint	ID of the system type of the column.
user_type_id	int	ID of the type of the column as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length (in bytes) of the column. -1 = Column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml . For text columns, the max_length value will be 16 or the value set by sp_tableoption 'text in row'.

COLUMN NAME	DATA TYPE	DESCRIPTION
precision	tinyint	Precision of the column if numeric-based; otherwise, 0.
scale	tinyint	Scale of column if numeric-based; otherwise, 0.
collation_name	sysname	Name of the collation of the column if character-based; otherwise, NULL.
is_nullable	bit	1 = Column is nullable.
is_ansi_padded	bit	1 = Column uses ANSI_PADDING ON behavior if character, binary, or variant. 0 = Column is not character, binary, or variant.
is_rowguidcol	bit	1 = Column is a declared ROWGUIDCOL.
is_identity	bit	1 = Column has identity values
is_computed	bit	1 = Column is a computed column.
is_filestream	bit	1 = Column is a FILESTREAM column.
is_replicated	bit	1 = Column is replicated.
is_non_sql_subscribed	bit	1 = Column has a non-SQL Server subscriber.
is_merge_published	bit	1 = Column is merge-published.
is_dts_replicated	bit	1 = Column is replicated by using SSIS.
is_xml_document	bit	1 = Content is a complete XML document. 0 = Content is a document fragment or the column data type is not xml .
xml_collection_id	int	Nonzero if the data type of the column is xml and the XML is typed. The value will be the ID of the collection containing the validating XML schema namespace of the column. 0 = No XML schema collection.

COLUMN NAME	DATA TYPE	DESCRIPTION
default_object_id	int	<p>ID of the default object, regardless of whether it is a stand-alone object sys.sp_bindefault, or an inline, column-level DEFAULT constraint. The parent_object_id column of an inline column-level default object is a reference back to the table itself.</p> <p>0 = No default.</p>
rule_object_id	int	<p>ID of the stand-alone rule bound to the column by using sys.sp_bindrule.</p> <p>0 = No stand-alone rule. For column-level CHECK constraints, see sys.check_constraints (Transact-SQL).</p>
is_sparse	bit	<p>1 = Column is a sparse column. For more information, see Use Sparse Columns.</p>
is_column_set	bit	<p>1 = Column is a column set. For more information, see Use Sparse Columns.</p>
generated_always_type	tinyint	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>Identifies when the column value is generated (will always be 0 for columns in system tables):</p> <p>0 = NOT_APPLICABLE</p> <p>1 = AS_ROW_START</p> <p>2 = AS_ROW_END</p> <p>For more information, see Temporal Tables (Relational databases).</p>
generated_always_type_desc	nvarchar(60)	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>Textual description of <code>generated_always_type</code>'s value (always NOT_APPLICABLE for columns in system tables)</p> <p>NOT_APPLICABLE</p> <p>AS_ROW_START</p> <p>AS_ROW_END</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
encryption_type	int	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>Encryption type:</p> <p>1 = Deterministic encryption</p> <p>2 = Randomized encryption</p>
encryption_type_desc	nvarchar(64)	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>Encryption type description:</p> <p>RANDOMIZED</p> <p>DETERMINISTIC</p>
encryption_algorithm_name	sysname	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>Name of encryption algorithm.</p> <p>Only AEAD_AES_256_CBC_HMAC_SHA_512 is supported.</p>
column_encryption_key_id	int	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>ID of the CEK.</p>
column_encryption_key_database_name	sysname	<p>Applies to: SQL Server 2016 through SQL Server 2017, SQL Database.</p> <p>The name of the database where the column encryption key exists if different than the database of the column. NULL if the key exists in the same database as the column.</p>
is_hidden	bit	<p>Applies to: SQL Server 2017 through SQL Server 2017, SQL Database.</p> <p>Indicates if the column is hidden:</p> <p>0 = regular, not-hidden, visible column</p> <p>1 = hidden column</p>
is_masked	bit	<p>Applies to: SQL Server 2017 through SQL Server 2017, SQL Database.</p> <p>Indicates if the column is masked by a dynamic data masking:</p> <p>0 = regular, not-masked column</p> <p>1 = column is masked</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[System Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)





[Querying the SQL Server System Catalog FAQ](#)

[sys.all_columns \(Transact-SQL\)](#)

[sys.system_columns \(Transact-SQL\)](#)

sys.computed_columns (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column found in **sys.columns** that is a computed-column.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Inherited columns>		The sys.computed_columns view returns all columns in the sys.columns view. It also returns the additional columns described below. For a description of the columns that the sys.computed_columns view inherits from sys.columns , see sys.columns (Transact-SQL) . The value of the is_computed column is always set to 1 in the sys.computed_columns view.
definition	nvarchar(max)	SQL text that defines this computed-column.
uses_database_collation	bit	1 = The column definition depends on the default collation of the database for correct evaluation; otherwise, 0. Such a dependency prevents changing the database default collation.
is_persisted	bit	Computed column is persisted.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.default_constraints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is a default definition (created as part of a CREATE TABLE or ALTER TABLE statement instead of a CREATE DEFAULT statement), with **sys.objects.type** = D.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
parent_column_id	int	ID of the column in parent_object_id to which this default belongs.
Definition	nvarchar(max)	SQL expression that defines this default.
is_system_named	bit	1 = Name was generated by system. 0 = Name was supplied by the user.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns the definition of the DEFAULT constraint that is applied to the `VacationHours` column of the `HumanResources.Employee` table.

```
USE AdventureWorks2012;
GO
SELECT d.definition
FROM sys.default_constraints AS d
INNER JOIN sys.columns AS c
ON d.parent_column_id = c.column_id
WHERE d.parent_object_id = OBJECT_ID(N'HumanResources.Employee', N'U')
AND c.name = 'VacationHours';
```

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.events (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each event for which a trigger or event notification fires. These events represent the event types that are specified when the trigger or event notification is created by using [CREATE TRIGGER](#) or [CREATE EVENT NOTIFICATION](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the trigger or event notification. This value, together with type , uniquely identifies the row.
type	int	Event that causes the trigger to fire.
type_desc	nvarchar(60)	Description of the event that causes the trigger to fire.
is_trigger_event	bit	1 = Trigger event. 0 = Notification event.
event_group_type	int	Event group on which the trigger or event notification is created, or null if not created on an event group.
event_group_type_desc	nvarchar(60)	Description of the event group on which the trigger or event notification is created, or null if not created on an event group.

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.event_notifications (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each object that is an event notification, with **sys.objects.type** = EN.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Event notification name.
object_id	int	Object identification number. Is unique within a database.
parent_class	tinyint	Class of parent. 0 = Database 1 = Object or Column
parent_class_desc	nvarchar(60)	DATABASE OBJECT_OR_COLUMN
parent_id	int	Non-zero ID of the parent object. 0 = The parent class is the database.
create_date	datetime	Date created.
modify_date	datetime	Always equals create_date .
service_name	nvarchar(256)	Name of the target service to which the notification is sent.
broker_instance	nvarchar(128)	Broker instance to which the notification is sent.
principal_id	int	ID of the database principal that owns this event notification.
creator_sid	varbinary(85)	SID of the login who created the event notification. Is NULL if the FAN_IN option is not specified.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.event_notification_event_types (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event or event group on which an event notification can fire.

COLUMN NAME	DATA TYPE	DESCRIPTION
type	int	Type of event or event group that causes an event notification to fire.
type_name	nvarchar(128)	Name of an event or event group. This can be specified in the FOR clause of a CREATE EVENT NOTIFICATION statement.
parent_type	int	Type of event group that is the parent of the event or event group.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.extended_procedures (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is an extended stored procedure, with **sys.objects.type** = X. Because extended stored procedures are installed into the **master** database, they are only visible from that database context. Selecting from the **sys.extended_procedures** view in any other database context will return an empty result set.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
dll_name	nvarchar(260)	Name, including path, of the DLL for this extended stored procedure.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.external_libraries (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2017)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Supports the management of package libraries related to external runtimes such as R or Python.

sys.external_libraries

The catalog view sys.external_libraries lists a row for each external library that has been uploaded into the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
external_library_id	int	ID of the external library object.
name	sysname	Name of the external library. Is unique within the database per owner.
principal_id	int	ID of the principal that owns this external library.
language	sysname	Name of the language or runtime that supports the external library. Valid values are 'R'. Additional runtimes might be added in future.
scope	int	0 for public scope; 1 for private scope
scope_desc	varchar(7)	Indicates whether the package is public or private

See also





[sys.external_library_files](#)

[CREATE EXTERNAL LIBRARY](#)

[Package management for SQL Server R Services](#)

sys.external_library_files (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2017)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Lists a row for each file that makes up an external library.

COLUMN NAME	DATA TYPE	DESCRIPTION
external_library_id	int	ID of the external library object.
content	varbinary(max)	Content of the external library file artifact.
platform	tinyint	ID of the host platform on which SQL Server is installed.
platform_desc	nvarchar(60)	Name of the host platform. Valid values are 'WINDOWS', 'LINUX'.

See also





[sys.external_libraries](#)

[CREATE EXTERNAL LIBRARY](#)

[Package management for SQL Server Machine Learning Service](#)

sys.foreign_keys (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per object that is a FOREIGN KEY constraint, with **sys.object.type** = F.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
referenced_object_id	int	ID of the referenced object.
key_index_id	int	ID of the key index within the referenced object.
is_disabled	bit	FOREIGN KEY constraint is disabled.
is_not_for_replication	bit	FOREIGN KEY constraint was created by using the NOT FOR REPLICATION option.
is_not_trusted	bit	FOREIGN KEY constraint has not been verified by the system.
delete_referential_action	tinyint	The referential action that was declared for this FOREIGN KEY when a delete happens. 0 = No action 1 = Cascade 2 = Set null 3 = Set default
delete_referential_action_desc	nvarchar(60)	Description of the referential action that was declared for this FOREIGN KEY when a delete occurs: NO_ACTION CASCADE SET_NULL SET_DEFAULT

COLUMN NAME	DATA TYPE	DESCRIPTION
update_referential_action	tinyint	<p>The referential action that was declared for this FOREIGN KEY when an update happens.</p> <p>0 = No action</p> <p>1 = Cascade</p> <p>2 = Set null</p> <p>3 = Set default</p>
update_referential_action_desc	nvarchar(60)	<p>Description of the referential action that was declared for this FOREIGN KEY when an update happens:</p> <p>NO_ACTION</p> <p>CASCADE</p> <p>SET_NULL</p> <p>SET_DEFAULT</p>
is_system_named	bit	<p>1 = Name was generated by the system.</p> <p>0 = Name was supplied by the user.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.foreign_key_columns (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column, or set of columns, that comprise a foreign key.

COLUMN NAME	DATA TYPE	DESCRIPTION
constraint_object_id	int	ID of the FOREIGN KEY constraint.
constraint_column_id	int	ID of the column, or set of columns, that comprise the FOREIGN KEY (<i>1..n</i> where n=number of columns).
parent_object_id	int	ID of the parent of the constraint, which is the referencing object.
parent_column_id	int	ID of the parent column, which is the referencing column.
referenced_object_id	int	ID of the referenced object, which has the candidate key.
referenced_column_id	int	ID of the referenced column (candidate key column).

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.function_order_columns (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row per column that is a part of an **ORDER** expression of a common language runtime (CLR) table-valued function.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object (CLR table-valued function) the order is defined on.
order_column_id	int	ID of the order column. order_column_id is unique only within object_id . order_column_id represents the position of this column in the ordering.
column_id	int	ID of the column in object_id . column_id is unique only within object_id .
is_descending	bit	1 = order column has a descending sort direction. 0 = order column has an ascending sort direction.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.hash_indexes (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows the current hash indexes and the hash index properties. Hash indexes are supported only on [In-Memory OLTP \(In-Memory Optimization\)](#).

The sys.hash_indexes view contains the same columns as the sys.indexes view and an additional column named **bucket_count**. For more information about the other columns in the sys.hash_indexes view, see [sys.indexes \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.indexes (Transact-SQL) .
bucket_count	int	Count of hash buckets for hash indexes. For more information about the bucket_count value, including guidelines for setting the value, see CREATE TABLE (Transact-SQL) .

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.. For more information, see [Metadata Visibility Configuration](#).

Examples

```
SELECT object_name([object_id]) AS 'table_name', [object_id],  
       [name] AS 'index_name', [type_desc], [bucket_count]  
FROM sys.hash_indexes  
WHERE OBJECT_NAME([object_id]) = 'T1';
```





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.identity_columns (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column that is an identity column.

The **sys.identity_columns** view inherits rows from the **sys.columns** view. The **sys.identity_columns** view returns the columns in the **sys.columns** view, plus the **seed_value**, **increment_value**, **last_value**, and **is_not_for_replication** columns. For more information, see [Catalog Views \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
<columns inherited from sys.columns>		The sys.identity_columns view returns all columns in the sys.columns view. It also returns the additional columns described below. For a description of the columns that the sys.identity_columns view inherits from sys.columns , see sys.columns (Transact-SQL) .
seed_value	sql_variant	Seed value for this identity column. The data type of the seed value is the same as the data type of the column itself.
increment_value	sql_variant	Increment value for this identity column. The data type of the seed value is the same as the data type of the column itself.
last_value	sql_variant	Last value generated for this identity column. The data type of the seed value is the same as the data type of the column itself.
is_not_for_replication	bit	Identity column is declared NOT FOR REPLICATION.

NOTE

To create an automatically incrementing number that can be used in multiple tables or that can be called from applications without referencing any table, see [Sequence Numbers](#).

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.indexes (Transact-SQL)

11/21/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per index or heap of a tabular object, such as a table, view, or table-valued function.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this index belongs.
name	sysname	Name of the index. name is unique only within the object. NULL = Heap
index_id	int	ID of the index. index_id is unique only within the object. 0 = Heap 1 = Clustered index > 1 = Nonclustered index
type	tinyint	Type of index: 0 = Heap 1 = Clustered 2 = Nonclustered 3 = XML 4 = Spatial 5 = Clustered columnstore index. Applies to: SQL Server 2014 through SQL Server 2017. 6 = Nonclustered columnstore index. Applies to: SQL Server 2012 through SQL Server 2017. 7 = Nonclustered hash index. Applies to: SQL Server 2014 through SQL Server 2017.

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	<p>Description of index type:</p> <p>HEAP</p> <p>CLUSTERED</p> <p>NONCLUSTERED</p> <p>XML</p> <p>SPATIAL</p> <p>CLUSTERED COLUMNSTORE - Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>NONCLUSTERED COLUMNSTORE - Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>NONCLUSTERED HASH : NONCLUSTERED HASH indexes are supported only on memory-optimized tables. The sys.hash_indexes view shows the current hash indexes and the hash properties. For more information, see sys.hash_indexes (Transact-SQL). Applies to: SQL Server 2014 through SQL Server 2017.</p>
is_unique	bit	<p>1 = Index is unique.</p> <p>0 = Index is not unique.</p> <p>Always 0 for clustered columnstore indexes.</p>
data_space_id	int	<p>ID of the data space for this index. Data space is either a filegroup or partition scheme.</p> <p>0 = object_id is a table-valued function or in-memory index.</p>
ignore_dup_key	bit	<p>1 = IGNORE_DUP_KEY is ON.</p> <p>0 = IGNORE_DUP_KEY is OFF.</p>
is_primary_key	bit	<p>1 = Index is part of a PRIMARY KEY constraint.</p> <p>Always 0 for clustered columnstore indexes.</p>
is_unique_constraint	bit	<p>1 = Index is part of a UNIQUE constraint.</p> <p>Always 0 for clustered columnstore indexes.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
fill_factor	tinyint	<p>> 0 = FILLFACTOR percentage used when the index was created or rebuilt.</p> <p>0 = Default value</p> <p>Always 0 for clustered columnstore indexes.</p>
is_padded	bit	<p>1 = PADINDEX is ON.</p> <p>0 = PADINDEX is OFF.</p> <p>Always 0 for clustered columnstore indexes.</p>
is_disabled	bit	<p>1 = Index is disabled.</p> <p>0 = Index is not disabled.</p>
is_hypothetical	bit	<p>1 = Index is hypothetical and cannot be used directly as a data access path. Hypothetical indexes hold column-level statistics.</p> <p>0 = Index is not hypothetical.</p>
allow_row_locks	bit	<p>1 = Index allows row locks.</p> <p>0 = Index does not allow row locks.</p> <p>Always 0 for clustered columnstore indexes.</p>
allow_page_locks	bit	<p>1 = Index allows page locks.</p> <p>0 = Index does not allow page locks.</p> <p>Always 0 for clustered columnstore indexes.</p>
has_filter	bit	<p>1 = Index has a filter and only contains rows that satisfy the filter definition.</p> <p>0 = Index does not have a filter.</p>
filter_definition	nvarchar(max)	<p>Expression for the subset of rows included in the filtered index.</p> <p>NULL for heap or non-filtered index.</p>
auto_created	bit	<p>1 = Index was created by the automatic tuning.</p> <p>0 = Index was created by the user.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns all indexes for the table `Production.Product` in the AdventureWorks2012 database.

```
SELECT i.name AS index_name
      ,i.type_desc
      ,is_unique
      ,ds.type_desc AS filegroup_or_partition_scheme
      ,ds.name AS filegroup_or_partition_scheme_name
      ,ignore_dup_key
      ,is_primary_key
      ,is_unique_constraint
      ,fill_factor
      ,is_padded
      ,is_disabled
      ,allow_row_locks
      ,allow_page_locks
FROM sys.indexes AS i
INNER JOIN sys.data_spaces AS ds ON i.data_space_id = ds.data_space_id
WHERE is_hypothetical = 0 AND i.index_id <> 0
AND i.object_id = OBJECT_ID('Production.Product');
GO
```

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[sys.index_columns \(Transact-SQL\)](#)

[sys.xml_indexes \(Transact-SQL\)](#)

[sys.objects \(Transact-SQL\)](#)

[sys.key_constraints \(Transact-SQL\)](#)

[sys.filegroups \(Transact-SQL\)](#)





[sys.partition_schemes \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.index_columns (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per column that is part of a **sys.indexes** index or unordered table (heap).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object the index is defined on.
index_id	int	ID of the index in which the column is defined.
index_column_id	int	ID of the index column. index_column_id is unique only within index_id .
column_id	int	ID of the column in object_id . 0 = Row Identifier (RID) in a nonclustered index. column_id is unique only within object_id .
key_ordinal	tinyint	Ordinal (1-based) within set of key-columns. 0 = Not a key column, or is an XML index, a columnstore index, or a spatial index. Note: An XML or spatial index cannot be a key because the underlying columns are not comparable, meaning that their values cannot be ordered.
partition_ordinal	tinyint	Ordinal (1-based) within set of partitioning columns. A clustered columnstore index can have at most 1 partitioning column. 0 = Not a partitioning column.
is_descending_key	bit	1 = Index key column has a descending sort direction. 0 = Index key column has an ascending sort direction, or the column is part of a columnstore or hash index.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_included_column	bit	<p>1 = Column is a nonkey column added to the index by using the CREATE INDEX INCLUDE clause, or the column is part of a columnstore index.</p> <p>0 = Column is not an included column.</p> <p>Columns implicitly added because they are part of the clustering key are not listed in sys.index_columns.</p> <p>Columns implicitly added because they are a partitioning column are returned as 0.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns all indexes and index columns for the table `Production.BillofMaterials`.

```
USE AdventureWorks2012;
GO
SELECT i.name AS index_name
      ,COL_NAME(ic.object_id,ic.column_id) AS column_name
      ,ic.index_column_id
      ,ic.key_ordinal
      ,ic.is_included_column
FROM sys.indexes AS i
INNER JOIN sys.index_columns AS ic
  ON i.object_id = ic.object_id AND i.index_id = ic.index_id
WHERE i.object_id = OBJECT_ID('Production.BillofMaterials');
```

Here is the result set.

index_name	column_name	index_column_id	key_ordinal	
is_included_column				
-----	-----	-----	-----	

AK_BillofMaterials_ProductAssemblyID_ComponentID_StartDate	ProductAssemblyID	1	1	0
AK_BillofMaterials_ProductAssemblyID_ComponentID_StartDate	ComponentID	2	2	0
AK_BillofMaterials_ProductAssemblyID_ComponentID_StartDate	StartDate	3	3	0
PK_BillofMaterials_BillofMaterialsID	BillofMaterialsID	1	1	0
IX_BillofMaterials_UnitMeasureCode	UnitMeasureCode	1	1	0
(5 row(s) affected)				

See Also

- [Object Catalog Views \(Transact-SQL\)](#)
- [Catalog Views \(Transact-SQL\)](#)
- [sys.indexes \(Transact-SQL\)](#)

[sys.objects \(Transact-SQL\)](#)





[CREATE INDEX \(Transact-SQL\)](#)

[sys.columns \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

index_resumable_operations (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2017)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

sys.index_resumable_operations is a system view that monitors and checks the current execution status for resumable Index rebuild.

Applies to: SQL Server 2017 and Azure SQL Database

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this index belongs (not nullable).
index_id	int	ID of the index (not nullable). index_id is unique only within the object.
name	sysname	Name of the index. name is unique only within the object.
sql_text	nvarchar(max)	DDL T-SQL statement text
last_max_dop	smallint	Last MAX_DOP used (default = 0)
partition_number	int	Partition number within the owning index or heap. For non-partitioned tables and indexes or in case all partitions are being rebuild the value of this column is NULL.
state	tinyint	Operational state for resumable index: 0=Running 1=Pause
state_desc	nvarchar(60)	Description of the operational state for resumable index (running or Paused)
start_time	datetime	Index operation start time (not nullable)
last_pause_time	datetime	Index operation last pause time (nullable). NULL if operation is running and never paused.
total_execution_time	int	Total execution time from start time in minutes (not nullable)
percent_complete	real	Index operation progress completion in % (not nullable).

COLUMN NAME	DATA TYPE	DESCRIPTION
page_count	bigint	Total number of index pages allocated by the index build operation for the new and mapping indexes (not nullable).

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Example

List all resumable index rebuild operations that are in the PAUSE state.

```
SELECT * FROM sys.index_resumable_operations WHERE STATE = 1;
```

See Also

[ALTER INDEX \(Transact-SQL\)](#)

[Catalog views \(Transact-SQL\)](#) [Object catalog views \(Transact-SQL\)](#) [sys.indexes \(Transact-SQL\)](#) [sys.index_columns \(Transact-SQL\)](#)

[sys.xml_indexes \(Transact-SQL\)](#)

[sys.objects \(Transact-SQL\)](#)

[sys.key_constraints \(Transact-SQL\)](#)





[sys.filegroups \(Transact-SQL\)](#)

[sys.partition_schemes \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.internal_partitions (Transact-SQL)

11/16/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each rowset that tracks internal data for columnstore indexes on disk-based tables. These rowsets are internal to columnstore indexes and track deleted rows, rowgroup mappings, and delta store rowgroups. They track data for each for each table partition; every table has at least one partition. SQL Server re-creates the rowsets each time it rebuilds the columnstore index.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Partition ID for this partition. This is unique within a database.
object_id	int	Object ID for the table that contains the partition.
index_id	int	Index ID for the columnstore index defined on the table. 1 = clustered columnstore index 2 = nonclustered columnstore index
partition_number	int	The partition number. 1 = first partition of a partitioned table, or the single partition of a nonpartitioned table. 2 = second partition, and so on.
internal_object_type	tinyint	Rowset objects that track internal data for the columnstore index. 2 = COLUMN_STORE_DELETE_BITMAP 3 = COLUMN_STORE_DELTA_STORE 4 = COLUMN_STORE_DELETE_BUFFER 5 = COLUMN_STORE_MAPPING_INDEX

COLUMN NAME	DATA TYPE	DESCRIPTION
internal_object_type_desc	nvarchar(60)	<p>COLUMN_STORE_DELETE_BITMAP – This bitmap index tracks rows that are marked as deleted from the columnstore. The bitmap is for every rowgroup since partitions can have rows in multiple rowgroups. The rows are that are still physically present and taking up space in the columnstore.</p> <p>COLUMN_STORE_DELTA_STORE – Stores groups of rows, called rowgroups, that have not been compressed into columnar storage. Each table partition can have zero or more deltastore rowgroups.</p> <p>COLUMN_STORE_DELETE_BUFFER – For maintaining deletes to updateable nonclustered columnstore indexes. When a query deletes a row from the underlying rowstore table, the delete buffer tracks the deletion from the columnstore. When the number of deleted rows exceed 1048576, they are merged back into the delete bitmap by background Tuple Mover thread or by an explicit Reorganize command. At any given point in time, the union of the delete bitmap and the delete buffer represents all deleted rows.</p> <p>COLUMN_STORE_MAPPING_INDEX – Used only when the clustered columnstore index has a secondary nonclustered index. This maps nonclustered index keys to the correct rowgroup and row ID in the columnstore. It only stores keys for rows that move to a different rowgroup; this occurs when a delta rowgroup is compressed into the columnstore, and when a merge operation merges rows from two different rowgroups.</p>
Row_group_id	int	ID for the deltastore rowgroup. Each table partition can have zero or more deltastore rowgroups.
hobt_id	bigint	ID of the internal rowset object. This is a good key for joining with other DMVs to get more information about the physical characteristics of the internal rowset.
rows	bigint	Approximate number of rows in this partition.

COLUMN NAME	DATA TYPE	DESCRIPTION
data_compression	tinyint	The state of compression for the rowset: 0 = NONE 1 = ROW 2 = PAGE
data_compression_desc	nvarchar(60)	The state of compression for each partition. Possible values for rowstore tables are NONE, ROW, and PAGE. Possible values for columnstore tables are COLUMNSTORE and COLUMNSTORE_ARCHIVE.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

General Remarks

SQL Server re-creates new columnstore internal indexes each time it creates or rebuilds a columnstore index.

Examples

A. View all of the internal rowsets for a table

This example returns all of the internal columnstore rowsets for a table. You can also use the `hobt_id` to find more information about the specific rowset.

```
SELECT i.object_id, i.index_id, i.name, p.hobt_id, p.internal_object_type_id, p.internal_object_type_desc
FROM sys.internal_partitions AS p
JOIN sys.indexes AS i
on i.object_id = p.object_id
WHERE p.object_id = OBJECT_ID ( '<table name' ) ;
```

See Also


[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.internal_tables (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each object that is an internal table. Internal tables are automatically generated by SQL Server to support various features. For example, when you create a primary XML index, SQL Server automatically creates an internal table to persist the shredded XML document data. Internal tables appear in the **sys** schema of every database and have unique, system-generated names that indicate their function, for example, **xml_index_nodes_2021582240_32001** or **queue_messages_1977058079**.

Internal tables do not contain user-accessible data, and their schema are fixed and unalterable. You cannot reference internal table names in Transact-SQL statements. For example, you cannot execute a statement such as `SELECT * FROM <sys.internal_table_name>`. However, you can query catalog views to see the metadata of internal tables.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
internal_type	tinyint	Type of the internal table: 201 = queue_messages 202 = xml_index_nodes 203 = fulltext_catalog_freelist 205 = query_notification 206 = service_broker_map 207 = extended_indexes (such as a spatial index) 208 = filestream_tombstone 209 = change_tracking 210 = tracked_committed_transactions

COLUMN NAME	DATA TYPE	DESCRIPTION
internal_type_desc	nvarchar(60)	<p>Description of the type of internal table:</p> <p>QUEUE_MESSAGES</p> <p>XML_INDEX_NODES</p> <p>FULLTEXT_CATALOG_FREELIST</p> <p>FULLTEXT_CATALOG_MAP</p> <p>QUERY_NOTIFICATION</p> <p>SERVICE_BROKER_MAP</p> <p>EXTENDED_INDEXES</p> <p>FILESTREAM_TOMBSTONE</p> <p>CHANGE_TRACKING</p> <p>TRACKED_COMMITTED_TRANSACTION S</p>
parent_id	int	<p>ID of the parent, regardless of whether it is schema-scoped or not. Otherwise, 0 if there is no parent.</p> <p>queue_messages = object_id of queue</p> <p>xml_index_nodes = object_id of the xml index</p> <p>fulltext_catalog_freelist = fulltext_catalog_id of the full-text catalog</p> <p>fulltext_index_map = object_id of the full-text index</p> <p>query_notification, or service_broker_map = 0</p> <p>extended_indexes = object_id of an extended index, such as a spatial index</p> <p>object_id of the table for which table tracking is enabled = change_tracking</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
parent_minor_id	int	<p>Minor ID of the parent.</p> <p>xml_index_nodes = index_id of the XML index</p> <p>extended_indexes = index_id of an extended index, such as a spatial index</p> <p>0 = queue_messages, fulltext_catalog_freelist, fulltext_index_map, query_notification, service_broker_map, or change_tracking</p>
lob_data_space_id	int	Non-zero value is the ID of data space (filegroup or partition-scheme) that holds the large object (LOB) data for this table.
filestream_data_space_id	int	Reserved for future use.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Remarks

Internal tables are placed on the same filegroup as the parent entity. You can use the catalog query shown in Example F below to return the number of pages internal tables consume for in-row, out-of-row, and large object (LOB) data.

You can use the [sp_spaceused](#) system procedure to return space usage data for internal tables. **sp_spaceused** reports internal table space in the following ways:

- When a queue name is specified, the underlying internal table associated with the queue is referenced and its storage consumption is reported.
- Pages that are used by the internal tables of XML indexes, spatial indexes, and full-text indexes are included in the **index_size** column. When a table or indexed view name is specified, the pages for the XML indexes, spatial indexes, and full-text indexes for that object are included in the columns **reserved** and **index_size**.

Examples

The following examples demonstrate how to query internal table metadata by using catalog views.

A. Show internal tables that inherit columns from the sys.objects catalog view

```
SELECT * FROM sys.objects WHERE type = 'IT';
```

B. Return all internal table metadata (including that which is inherited from sys.objects)

```
SELECT * FROM sys.internal_tables;
```

C. Return internal table columns and column data types

```
SELECT SCHEMA_NAME(itab.schema_id) AS schema_name
      , itab.name AS internal_table_name
      , typ.name AS column_data_type
      , col.*
FROM sys.internal_tables AS itab
JOIN sys.columns AS col ON itab.object_id = col.object_id
JOIN sys.types AS typ ON typ.user_type_id = col.user_type_id
ORDER BY itab.name, col.column_id;
```

D. Return internal table indexes

```
SELECT SCHEMA_NAME(itab.schema_id) AS schema_name
      , itab.name AS internal_table_name
      , idx.*
FROM sys.internal_tables AS itab
JOIN sys.indexes AS idx ON itab.object_id = idx.object_id
ORDER BY itab.name, idx.index_id;
```

E. Return internal table statistics

```
SELECT SCHEMA_NAME(itab.schema_id) AS schema_name
      , itab.name AS internal_table_name
      , s.*
FROM sys.internal_tables AS itab
JOIN sys.stats AS s ON itab.object_id = s.object_id
ORDER BY itab.name, s.stats_id;
```

F. Return internal table partition and allocation unit information

```
SELECT SCHEMA_NAME(itab.schema_id) AS schema_name
      , itab.name AS internal_table_name
      , idx.name AS heap_or_index_name
      , p.*
      , au.*
FROM sys.internal_tables AS itab
JOIN sys.indexes AS idx
--      JOIN to the heap or the clustered index
  ON itab.object_id = idx.object_id AND idx.index_id IN (0,1)
JOIN   sys.partitions AS p
  ON p.object_id = idx.object_id AND p.index_id = idx.index_id
JOIN   sys.allocation_units AS au
--      IN_ROW_DATA (type 1) and ROW_OVERFLOW_DATA (type 3) => JOIN to partition's Hobt
--      else LOB_DATA (type 2) => JOIN to the partition ID itself.
ON au.container_id =
  CASE au.type
    WHEN 2 THEN p.partition_id
    ELSE p.hobt_id
  END
ORDER BY itab.name, idx.index_id;
```

G. Return internal table metadata for XML indexes

```
SELECT t.name AS parent_table
      ,t.object_id AS parent_table_id
      ,it.name AS internal_table_name
      ,it.object_id AS internal_table_id
      ,xi.name AS primary_XML_index_name
      ,xi.index_id as primary_XML_index_id
FROM sys.internal_tables AS it
JOIN sys.tables AS t
      ON it.parent_id = t.object_id
JOIN sys.xml_indexes AS xi
      ON it.parent_id = xi.object_id
      AND it.parent_minor_id = xi.index_id
WHERE it.internal_type_desc = 'XML_INDEX_NODES';
GO
```

H. Return internal table metadata for Service Broker queues

```
SELECT q.name AS queue_name
      ,q.object_id AS queue_id
      ,it.name AS internal_table_name
      ,it.object_id AS internal_table_id
FROM sys.internal_tables AS it
JOIN sys.service_queues AS q ON it.parent_id = q.object_id
WHERE it.internal_type_desc = 'QUEUE_MESSAGES';
GO
```

I. Return internal table metadata for all Service Broker services

```
SELECT *
FROM tempdb.sys.internal_tables
WHERE internal_type_desc = 'SERVICE_BROKER_MAP';
GO
```





See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

sys.key_constraints (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is a primary key or unique constraint. Includes **sys.objects.type** PK and UQ.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
unique_index_id	int	ID of the corresponding unique index in the parent object that was created to enforce this constraint.
is_system_named	bit	1 = Name was generated by system. 0 = Name was supplied by the user.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.masked_columns (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Use the **sys.masked_columns** view to query for table-columns that have a dynamic data masking function applied to them. This view inherits from the **sys.columns** view. It returns all columns in the **sys.columns** view, plus the **is_masked** and **masking_function** columns, indicating if the column is masked, and if so, what masking function is defined. This view only shows the columns on which there is a masking function applied.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this column belongs.
name	sysname	Name of the column. Is unique within the object.
column_id	int	ID of the column. Is unique within the object. Column IDs might not be sequential.
sys.masked_columns returns many more columns inherited from sys.columns .	various	See sys.columns (Transact-SQL) for more column definitions.
is_masked	bit	Indicates if the column is masked. 1 indicates masked.
masking_function	nvarchar(4000)	The masking function for the column.

Remarks

Permissions

This view returns information about tables where the user has some sort of permission on the table or if the user has the VIEW ANY DEFINITION permission.

Example

The following query joins **sys.masked_columns** to **sys.tables** to return information about all masked columns.

```
SELECT tbl.name as table_name, c.name AS column_name, c.is_masked, c.masking_function
FROM sys.masked_columns AS c
JOIN sys.tables AS tbl
    ON c.object_id = tbl.object_id
WHERE is_masked = 1;
```





See Also

[Dynamic Data Masking](#)

[sys.columns \(Transact-SQL\)](#)

sys.memory_optimized_tables_internal_attributes (Transact-SQL)

11/16/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each internal memory-optimized table used for storing user memory-optimized tables. Each user table corresponds to one or more internal tables. A single table is used for the core data storage. Additional internal tables are used to support features such as temporal, columnstore index and off-row (LOB) storage for memory-optimized tables.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the user table. Internal memory-optimized tables that exist to support a user table (such as off-row storage or deleted rows in case of Hk/Columnstore combinations) have the same object_id as their parent.
xtp_object_id	bigint	In-Memory OLTP object ID corresponding to the internal memory-optimized table that is used to support the user table. It is unique within the database and it can change over the lifetime of the object.
type	int	Type of internal table. 0 => DELETED_ROWS_TABLE 1 => USER_TABLE 2 => DICTIONARIES_TABLE 3 => SEGMENTS_TABLE 4 => ROW_GROUPS_INFO_TABLE 5 => INTERNAL OFF-ROW DATA TABLE 252 => INTERNAL_TEMPORAL_HISTORY_TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	<p>Description of the type</p> <p>DELETED_ROWS_TABLE -> Internal table tracking deleted rows for a columnstore index</p> <p>USER_TABLE -> Table containing the in-row user data</p> <p>DICTIONARIES_TABLE -> Dictionaries for a columnstore index</p> <p>SEGMENTS_TABLE -> Compressed segments for a columnstore index</p> <p>ROW_GROUPS_INFO_TABLE -> Metadata about compressed row groups of a columnstore index</p> <p>INTERNAL OFF-ROW DATA TABLE -> Internal table used for storage of an off-row column. In this case, minor_id reflects the column_id.</p> <p>INTERNAL_TEMPORAL_HISTORY_TABLE -> Hot tail of the disk-based history table. Rows inserted into the history are inserted into this internal memory-optimized table first. There is a background task that asynchronously moves rows from this internal table to the disk-based history table.</p>
minor_id	int	<p>0 indicates a user or internal table</p> <p>Non-0 indicates the ID of a column stored off-row. Joins with column_id in sys.columns.</p> <p>Each column stored off-row has a corresponding row in this system view.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

A. Returning all columns that are stored off-row

The following T-SQL script illustrates a table with multiple large non-LOB columns and a single LOB column:

```
CREATE TABLE dbo.LargeTableSample
(
    Id    int IDENTITY PRIMARY KEY NONCLUSTERED,
    C1    nvarchar(4000),
    C2    nvarchar(4000),
    C3    nvarchar(4000),
    C4    nvarchar(4000),
    Misc  nvarchar(max)
) WITH (MEMORY_OPTIMIZED = ON);
GO
```

The following query shows all columns that are stored off-row, along with their sizes. A size of -1 indicates a LOB

column. All LOB columns are stored off-row.

```
SELECT
    QUOTENAME(SCHEMA_NAME(o.schema_id)) + N'.' + QUOTENAME(OBJECT_NAME(moa.object_id)) AS 'table',
    c.name AS 'column',
    c.max_length
FROM sys.memory_optimized_tables_internal_attributes moa
    JOIN sys.columns c ON moa.object_id = c.object_id AND moa.minor_id=c.column_id
    JOIN sys.objects o on moa.object_id=o.object_id
WHERE moa.type=5;
```

B. Returning memory consumption of all columns that are stored off-row

To get more details about the memory consumption of off-row columns you can use the following query, which shows the memory consumption of all internal tables and their indexes that are used to store the off-row columns:

```
SELECT
    QUOTENAME(SCHEMA_NAME(o.schema_id)) + N'.' + QUOTENAME(OBJECT_NAME(moa.object_id)) AS 'table',
    c.name AS 'column',
    c.max_length,
    mc.memory_consumer_desc,
    mc.index_id,
    mc.allocated_bytes,
    mc.used_bytes
FROM sys.memory_optimized_tables_internal_attributes moa
    JOIN sys.columns c ON moa.object_id = c.object_id AND moa.minor_id=c.column_id
    JOIN sys.dm_db_xtp_memory_consumers mc ON moa.xtp_object_id=mc.xtp_object_id
    JOIN sys.objects o on moa.object_id=o.object_id
WHERE moa.type=5;
```

C. Returning memory consumption of columnstore indexes on memory-optimized tables

Use the following query to show the memory consumption of columnstore indexes on memory-optimized tables:

```
SELECT
    QUOTENAME(SCHEMA_NAME(o.schema_id)) + N'.' + QUOTENAME(OBJECT_NAME(moa.object_id)) AS 'table',
    i.name AS 'columnstore index',
    SUM(mc.allocated_bytes) / 1024 as [allocated_kb],
    SUM(mc.used_bytes) / 1024 as [used_kb]
FROM sys.memory_optimized_tables_internal_attributes moa
    JOIN sys.indexes i ON moa.object_id = i.object_id AND i.type in (5,6)
    JOIN sys.dm_db_xtp_memory_consumers mc ON moa.xtp_object_id=mc.xtp_object_id
    JOIN sys.objects o on moa.object_id=o.object_id
WHERE moa.type IN (0, 2, 3, 4)
GROUP BY o.schema_id, moa.object_id, i.name;
```

Use the following query break down the memory consumption across internal structures used for columnstore indexes on memory-optimized tables:

```
SELECT
    QUOTENAME(SCHEMA_NAME(o.schema_id)) + N'.' + QUOTENAME(OBJECT_NAME(moa.object_id)) AS 'table',
    i.name AS 'columnstore index',
    moa.type_desc AS 'internal table',
    mc.index_id AS 'index',
    mc.memory_consumer_desc,
    mc.allocated_bytes / 1024 as [allocated_kb],
    mc.used_bytes / 1024 as [used_kb]
FROM sys.memory_optimized_tables_internal_attributes moa
    JOIN sys.indexes i ON moa.object_id = i.object_id AND i.type in (5,6)
    JOIN sys.dm_db_xtp_memory_consumers mc ON moa.xtp_object_id=mc.xtp_object_id
    JOIN sys.objects o on moa.object_id=o.object_id
WHERE moa.type IN (0, 2, 3, 4)
```

sys.module_assembly_usages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each module-to-assembly reference.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	Object identification number of the SQL object. Is unique within a database.
assembly_id	int	ID of the assembly from which this module was created.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.numbered_procedures (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each SQL Server stored procedure that was created as a numbered procedure. This does not show a row for the base (number = 1) stored procedure. Entries for the base stored procedures can be found in views such as **sys.objects** and **sys.procedures**.

IMPORTANT

Numbered procedures are deprecated. Use of numbered procedures is discouraged. A DEPRECATION_ANNOUNCEMENT event is fired when a query that uses this catalog view is compiled.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object of the stored procedure.
procedure_number	smallint	Number of this procedure within the object, 2 or greater.
definition	nvarchar(max)	The SQL Server text that defines this procedure. NULL = encrypted.

NOTE

XML and CLR parameters are not supported for numbered procedures.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.numbered_procedure_parameters (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each parameter of a numbered procedure. When you create a numbered stored procedure, the base procedure is number 1. All subsequent procedures have numbers 2, 3, and so forth.

sys.numbered_procedure_parameters contains the parameter definitions for all subsequent procedures, numbered 2 and greater. This view does not show parameters for the base stored procedure (number = 1). The base stored procedure is similar to a nonnumbered stored procedure. Therefore, its parameters are represented in [sys.parameters \(Transact-SQL\)](#).

IMPORTANT

Numbered procedures are deprecated. Use of numbered procedures is discouraged. A DEPRECATION_ANNOUNCEMENT event is fired when a query that uses this catalog view is compiled.

NOTE

XML and CLR parameters are not supported for numbered procedures.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this parameter belongs.
procedure_number	smallint	Number of this procedure within the object, 2 or greater.
name	sysname	Name of the parameter. Is unique within procedure_number .
parameter_id	int	ID of the parameter. Is unique within the procedure_number .
system_type_id	tinyint	ID of the system type of the parameter
user_type_id	int	ID of the type, as defined by user, of the parameter.
max_length	smallint	Maximum length of the parameter in bytes. -1 = Column data type is varchar(max), nvarchar(max), or varbinary(max).
precision	tinyint	Precision of the parameter if numeric-based; otherwise, 0.

COLUMN NAME	DATA TYPE	DESCRIPTION
scale	tinyint	Scale of the parameter if numeric-based; otherwise, 0.
is_output	bit	1 = Parameter is output or return; otherwise, 0
is_cursor_ref	bit	1 = Parameter is a cursor-reference parameter.

NOTE

XML and CLR parameters are not supported for numbered procedures.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.objects (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each user-defined, schema-scoped object that is created within a database, including natively compiled scalar user-defined function.

For more information, see [Scalar User-Defined Functions for In-Memory OLTP](#).

NOTE

sys.objects does not show DDL triggers, because they are not schema-scoped. All triggers, both DML and DDL, are found in [sys.triggers](#). sys.triggers supports a mixture of name-scoping rules for the various kinds of triggers.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Object name.
object_id	int	Object identification number. Is unique within a database.
principal_id	int	<p>ID of the individual owner, if different from the schema owner. By default, schema-contained objects are owned by the schema owner. However, an alternate owner can be specified by using the ALTER AUTHORIZATION statement to change ownership.</p> <p>Is NULL if there is no alternate individual owner.</p> <p>Is NULL if the object type is one of the following:</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>TA = Assembly (CLR-integration) trigger</p> <p>TR = SQL trigger</p> <p>UQ = UNIQUE constraint</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
schema_id	int	<p>ID of the schema that the object is contained in.</p> <p>Schema-scoped system objects are always contained in the sys or INFORMATION_SCHEMA schemas.</p>
parent_object_id	int	<p>ID of the object to which this object belongs.</p> <p>0 = Not a child object.</p>
type	char(2)	<p>Object type:</p> <p>AF = Aggregate function (CLR)</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>FN = SQL scalar function</p> <p>FS = Assembly (CLR) scalar-function</p> <p>FT = Assembly (CLR) table-valued function</p> <p>IF = SQL inline table-valued function</p> <p>IT = Internal table</p> <p>P = SQL Stored Procedure</p> <p>PC = Assembly (CLR) stored-procedure</p> <p>PG = Plan guide</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>RF = Replication-filter-procedure</p> <p>S = System base table</p> <p>SN = Synonym</p> <p>SO = Sequence object</p> <p>U = Table (user-defined)</p> <p>V = View</p> <p>Applies to: SQL Server 2012 through SQL Server 2017.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
		<p>SQ = Service queue</p> <p>TA = Assembly (CLR) DML trigger</p> <p>TF = SQL table-valued-function</p> <p>TR = SQL DML trigger</p> <p>TT = Table type</p> <p>UQ = UNIQUE constraint</p> <p>X = Extended stored procedure</p> <p>Applies to: SQL Server 2016 through SQL Server 2017, Azure SQL Database, Azure SQL Data Warehouse, Parallel Data Warehouse.</p> <p>ET = External Table</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	<p>Description of the object type:</p> <p>AGGREGATE_FUNCTION</p> <p>CHECK_CONSTRAINT</p> <p>CLR_SCALAR_FUNCTION</p> <p>CLR_STORED_PROCEDURE</p> <p>CLR_TABLE_VALUED_FUNCTION</p> <p>CLR_TRIGGER</p> <p>DEFAULT_CONSTRAINT</p> <p>EXTENDED_STORED_PROCEDURE</p> <p>FOREIGN_KEY_CONSTRAINT</p> <p>INTERNAL_TABLE</p> <p>PLAN_GUIDE</p> <p>PRIMARY_KEY_CONSTRAINT</p> <p>REPLICATION_FILTER_PROCEDURE</p> <p>RULE</p> <p>SEQUENCE_OBJECT</p> <p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>SERVICE_QUEUE</p> <p>SQL_INLINE_TABLE_VALUED_FUNCTION</p> <p>SQL_SCALAR_FUNCTION</p> <p>SQL_STORED_PROCEDURE</p> <p>SQL_TABLE_VALUED_FUNCTION</p> <p>SQL_TRIGGER</p> <p>SYNONYM</p> <p>SYSTEM_TABLE</p> <p>TABLE_TYPE</p> <p>UNIQUE_CONSTRAINT</p> <p>USER_TABLE</p> <p>VIEW</p>

COLUMN NAME	DATA TYPE	VIEW
		DESCRIPTION
create_date	datetime	Date the object was created.
modify_date	datetime	Date the object was last modified by using an ALTER statement. If the object is a table or a view, modify_date also changes when a clustered index on the table or view is created or altered.
is_ms_shipped	bit	Object is created by an internal SQL Server component.
is_published	bit	Object is published.
is_schema_published	bit	Only the schema of the object is published.

Remarks

You can apply the [OBJECT_ID](#), [OBJECT_NAME](#), and [OBJECTPROPERTY\(\)](#) built-in functions to the objects shown in sys.objects.

There is a version of this view with the same schema, called [sys.system_objects](#), that shows system objects. There is another view called [sys.all_objects](#) that shows both system and user objects. All three catalog views have the same structure.

In this version of SQL Server, an extended index, such as an XML index or spatial index, is considered an internal table in sys.objects (type = IT and type_desc = INTERNAL_TABLE). For an extended index:

- name is the internal name of the index table.
- parent_object_id is the object_id of the base table.
- is_ms_shipped, is_published and is_schema_published columns are set to 0.

Related useful system views

Subsets of the objects can be viewed by using system views for a specific type of object, such as:

- [sys.tables](#)
- [sys.views](#)
- [sys.procedures](#)

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

A. Returning all the objects that have been modified in the last N days

Before you run the following query, replace `<database_name>` and `<n_days>` with valid values.

```

USE <database_name>;
GO
SELECT name AS object_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.objects
WHERE modify_date > GETDATE() - <n_days>
ORDER BY modify_date;
GO

```

B. Returning the parameters for a specified stored procedure or function

Before you run the following query, replace `<database_name>` and `<schema_name.object_name>` with valid names.

```

USE <database_name>;
GO
SELECT SCHEMA_NAME(schema_id) AS schema_name
      ,o.name AS object_name
      ,o.type_desc
      ,p.parameter_id
      ,p.name AS parameter_name
      ,TYPE_NAME(p.user_type_id) AS parameter_type
      ,p.max_length
      ,p.precision
      ,p.scale
      ,p.is_output
FROM sys.objects AS o
INNER JOIN sys.parameters AS p ON o.object_id = p.object_id
WHERE o.object_id = OBJECT_ID('<schema_name.object_name>')
ORDER BY schema_name, object_name, p.parameter_id;
GO

```

C. Returning all the user-defined functions in a database

Before you run the following query, replace `<database_name>` with a valid database name.

```

USE <database_name>;
GO
SELECT name AS function_name
      ,SCHEMA_NAME(schema_id) AS schema_name
      ,type_desc
      ,create_date
      ,modify_date
FROM sys.objects
WHERE type_desc LIKE '%FUNCTION%';
GO

```

D. Returning the owner of each object in a schema.

Before you run the following query, replace all occurrences of `<database_name>` and `<schema_name>` with valid names.

```

USE <database_name>;
GO
SELECT 'OBJECT' AS entity_type
      ,USER_NAME(OBJECTPROPERTY(object_id, 'OwnerId')) AS owner_name
      ,name
FROM sys.objects WHERE SCHEMA_NAME(schema_id) = '<schema_name>'
UNION
SELECT 'TYPE' AS entity_type
      ,USER_NAME(TYPEPROPERTY(SCHEMA_NAME(schema_id) + '.' + name, 'OwnerId')) AS owner_name
      ,name
FROM sys.types WHERE SCHEMA_NAME(schema_id) = '<schema_name>'
UNION
SELECT 'XML SCHEMA COLLECTION' AS entity_type
      ,COALESCE(USER_NAME(xsc.principal_id),USER_NAME(s.principal_id)) AS owner_name
      ,xsc.name
FROM sys.xml_schema_collections AS xsc JOIN sys.schemas AS s
      ON s.schema_id = xsc.schema_id
WHERE s.name = '<schema_name>';
GO

```

See Also

[Catalog Views \(Transact-SQL\)](#)

[sys.all_objects \(Transact-SQL\)](#)

[sys.system_objects \(Transact-SQL\)](#)

[sys.triggers \(Transact-SQL\)](#)





[Object Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[sys.internal_tables \(Transact-SQL\)](#)

sys.parameters (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each parameter of an object that accepts parameters. If the object is a scalar function, there is also a single row describing the return value. That row will have a **parameter_id** value of 0.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this parameter belongs.
name	sysname	Name of the parameter. Is unique within the object. If the object is a scalar function, the parameter name is an empty string in the row representing the return value.
parameter_id	int	ID of the parameter. Is unique within the object. If the object is a scalar function, parameter_id = 0 represents the return value.
system_type_id	tinyint	ID of the system type of the parameter.
user_type_id	int	ID of the type of the parameter as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length of the parameter, in bytes. Value = -1 when the column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml .
precision	tinyint	Precision of the parameter if numeric-based; otherwise, 0.
scale	tinyint	Scale of the parameter if numeric-based; otherwise, 0.
is_output	bit	1 = Parameter is OUTPUT or RETURN; otherwise, 0.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_cursor_ref	bit	1 = Parameter is a cursor-reference parameter.
has_default_value	bit	1 = Parameter has default value. SQL Server only maintains default values for CLR objects in this catalog view; therefore, this column has a value of 0 for Transact-SQL objects. To view the default value of a parameter in a Transact-SQL object, query the definition column of the sys.sql_modules catalog view, or use the OBJECT_DEFINITION system function.
is_xml_document	bit	1 = Content is a complete XML document. 0 = Content is a document fragment, or the data type of the column is not xml .
default_value	sql_variant	If has_default_value is 1, the value of this column is the value of the default for the parameter; otherwise, NULL.
xml_collection_id	int	Non-zero if the data type of the parameter is xml and the XML is typed. The value is the ID of the collection containing the validating XML schema namespace of the parameter. 0 = No XML schema collection.
is_readonly	bit	1 = Parameter is READONLY; otherwise, 0.
is_nullable	bit	1 = Parameter is nullable. (the default). 0 = Parameter is not nullable, for more efficient execution of natively-compiled stored procedures.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)





[Querying the SQL Server System Catalog FAQ](#)

[sys.all_parameters \(Transact-SQL\)](#)

[sys.system_parameters \(Transact-SQL\)](#)

sys.partitions (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each partition of all the tables and most types of indexes in the database. Special index types such as Full-Text, Spatial, and XML are not included in this view. All tables and indexes in SQL Server contain at least one partition, whether or not they are explicitly partitioned.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Indicates the partition ID. Is unique within a database.
object_id	int	Indicates the ID of the object to which this partition belongs. Every table or view is composed of at least one partition.
index_id	int	Indicates the ID of the index within the object to which this partition belongs. 0 = heap 1 = clustered index 2 or greater = nonclustered index
partition_number	int	Is a 1-based partition number within the owning index or heap. For non-partitioned tables and indexes, the value of this column is 1.
hobt_id	bigint	Indicates the ID of the data heap or B-tree that contains the rows for this partition.
rows	bigint	Indicates the approximate number of rows in this partition.
filestream_filegroup_id	smallint	Applies to: SQL Server 2012 through SQL Server 2017. Indicates the ID of the FILESTREAM filegroup stored on this partition.

COLUMN NAME	DATA TYPE	DESCRIPTION
data_compression	tinyint	<p>Indicates the state of compression for each partition:</p> <p>0 = NONE 1 = ROW 2 = PAGE 3 = COLUMNSTORE : Applies to: SQL Server 2012 through SQL Server 2017 4 = COLUMNSTORE_ARCHIVE : Applies to: SQL Server 2014 through SQL Server 2017</p> <p>Note: Full text indexes will be compressed in any edition of SQL Server.</p>
data_compression_desc	nvarchar(60)	<p>Indicates the state of compression for each partition. Possible values for rowstore tables are NONE, ROW, and PAGE. Possible values for columnstore tables are COLUMNSTORE and COLUMNSTORE_ARCHIVE.</p>

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also





[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.periods (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each table for which periods have been defined.

COLUMN HEADER	DATA TYPE	DESCRIPTION
period_type	sysname	Name of the period
period_type_desc	tinyint	The numeric value representing the type of period: 1 = system-time period
object_id	nvarchar(60)	The text description of the type of column: SYSTEM_TIME_PERIOD
object_id	int	The id of the table containing the period_type column
start_column_id	int	The id of the column that defines the lower period boundary
end_column_id	int	The id of the column that defines the upper period boundary

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[System Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[sys.all_columns \(Transact-SQL\)](#)

[sys.system_columns \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[Temporal Tables](#)

sys.plan_guides (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

Contains a row for each plan guide in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
plan_guide_id	int	Unique identifier of the plan guide in the database.
name	sysname	Name of the plan guide.
create_date	datetime	Date and time the plan guide was created.
modify_date	Datetime	Date the plan guide was last modified.
is_disabled	bit	1 = Plan guide is disabled. 0 = Plan guide is enabled.
query_text	nvarchar(max)	Text of the query on which the plan guide is created.
scope_type	tinyint	Identifies the scope of the plan guide. 1 = OBJECT 2 = SQL 3 = TEMPLATE
scope_type_desc	nvarchar(60)	Description of scope of the plan guide. OBJECT SQL TEMPLATE
scope_object_id	Int	object_id of the object defining the scope of the plan guide, if the scope is OBJECT. NULL if the plan guide is not scoped to OBJECT.

COLUMN NAME	DATA TYPE	DESCRIPTION
scope_batch	nvarchar(max)	Batch text, if scope_type is SQL. NULL if batch type is not SQL. If NULL and scope_type is SQL, the value of query_text applies.
parameters	nvarchar(max)	The string defining the list of parameters associated with the plan guide. NULL = No parameter list is associated with the plan guide.
hints	nvarchar(max)	The OPTION clause hints associated with the plan guide.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Catalog Views \(Transact-SQL\)](#)

[sp_create_plan_guide \(Transact-SQL\)](#)

[sp_create_plan_guide_from_handle \(Transact-SQL\)](#)

sys.procedures (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is a procedure of some kind, with **sys.objects.type** = P, X, RF, and PC.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL)
is_auto_executed	bit	1 = Procedure is auto-executed at the server startup; otherwise, 0. Can only be set for procedures in the master database.
is_execution_replicated	bit	Execution of this procedure is replicated.
is_repl_serializable_only	bit	Replication of the procedure execution is done only when the transaction can be serialized.
skips_repl_constraints	bit	During execution, the procedure skips constraints marked NOT FOR REPLICATION.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.sequences (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each sequence object in a database.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits all columns from sys.objects .
start_value	sql_variant NOT NULL	The starting value for the sequence object. If the sequence object is restarted by using ALTER SEQUENCE it will restart at this value. When the sequence object cycles it proceeds to the minimum_value or maximum_value , not the start_value .
increment	sql_variant NOT NULL	The value that is used to increment the sequence object after each generated value.
minimum_value	sql_variant NULL	The minimum value that can be generated by the sequence object. After this value is reached, the sequence object will either return an error when trying to generate more values or restart if the CYCLE option is specified. If no MINVALUE has been specified, this column returns the minimum value supported by the sequence generator's data type.
maximum_value	sql_variant NULL	The maximum value that can be generated by the sequence object. After this value is reached the sequence object will either start returning an error when trying to generate more values or restart if the CYCLE option is specified. If no MAXVALUE has been specified this column returns the maximum value supported by the sequence object's data type.
is_cycling	bit NOT NULL	Returns 0 if NO CYCLE has been specified for the sequence object and 1 if CYCLE has been specified.
is_cached	bit NOT NULL	Returns 0 if NO CACHE has been specified for the sequence object and 1 if CACHE has been specified.

COLUMN NAME	DATA TYPE	DESCRIPTION
cache_size	int NULL	Returns the specified cache size for the sequence object. This column contains NULL if the sequence was created with the NO CACHE option or if CACHE was specified without specifying a cache size. If the value specified by the cache size is larger than the maximum number of values that can be returned by the sequence object, that unobtainable cache size is still displayed.
system_type_id	tinyint NOT NULL	ID of the system type for sequence object's data type.
user_type_id	int NOT NULL	ID of the data type for the sequence object as defined by the user.
precision	tinyint NOT NULL	Max precision of the data type.
scale	tinyint NOT NULL	Max scale of the type. Scale is returned together with precision to give users complete metadata. Scale is always 0 for sequence objects because only integer types are allowed.
current_value	sql_variant NOT NULL	The last value obligated. That is, the value returned from the most recent execution of the NEXT VALUE FOR function or the last value from executing the sp_sequence_get_range procedure. Returns the START WITH value if the sequence has never been used.
is_exhausted	bit NOT NULL	0 indicates that more values can be generated from the sequence. 1 indicates that the sequence object has reached the MAXVALUE parameter and the sequence is not set to CYCLE. The NEXT VALUE FOR function returns an error until the sequence is restarted by using ALTER SEQUENCE.

Permissions

In SQL Server 2005 and later versions, the visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Sequence Numbers](#)

[CREATE SEQUENCE \(Transact-SQL\)](#)

[ALTER SEQUENCE \(Transact-SQL\)](#)

[DROP SEQUENCE \(Transact-SQL\)](#)

[NEXT VALUE FOR \(Transact-SQL\)](#)

`sp_sequence_get_range` (Transact-SQL)

sys.server_assembly_modules (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each assembly module for the server-level triggers of type TA. This view maps assembly triggers to the underlying CLR implementation. You can join this relation to **sys.server_triggers**. The assembly must be loaded into the **master** database. The tuple (object_id) is the key for the relation.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	This is a FOREIGN KEY reference back to the object upon which this assembly module is defined.
assembly_id	int	ID of the assembly from which this module was created. The assembly must be loaded into the master database.
assembly_class	sysname	Name of the class within assembly that defines this module.
assembly_method	sysname	Name of the method within the class that defines this module. Is NULL for aggregate functions (AF).
execute_as_principal_id	int	ID of the EXECUTE AS server principal. NULL by default or if EXECUTE AS CALLER. ID of the specified principal if EXECUTE AS SELF EXECUTE AS <principal>. -2 = EXECUTE AS OWNER.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).


See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

sys.server_events (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each event for which a server-level event-notification or server-level DDL trigger fires. The columns **object_id** and **type** uniquely identify the server event.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the server-level event notification or server-level DDL trigger to fire.
type	int	Type of the event that causes the event notification or DDL trigger to fire.
type_desc	nvarchar(60)	Description of the event that causes the DDL trigger or event notification to fire.
event_group_type	int	Event group on which the trigger or event notification is created, or null if not created on an event group.
event_group_type_desc	nvarchar(60)	Description of the event group on which the trigger or event notification is created, or null if not created on an event group

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.server_event_notifications (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each server-level event notification object.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Server event notification name. Is unique across all server-level event notifications.
object_id	int	Object identification number. Is unique within the master database.
parent_class	tinyint	Class of parent. Is always 100 = Server.
parent_class_desc	nvarchar(60)	Description of class of parent. Is always SERVER.
parent_id	int	Is always 0.
create_date	datetime	Date created.
modify_date	datetime	Date object was last modified by using an ALTER statement.
service_name	nvarchar(256)	Name of the target service to which the notification is sent.
broker_instance	nvarchar(128)	The service broker where the named target service is defined.
creator_sid	varbinary(85)	SID of the login executing the statement that creates the event notification. NULL if WITH FAN_IN is not specified in the event notification definition.
principal_id	int	ID of the server principal that owns this.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.server_sql_modules (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains the set of SQL modules for server-level triggers of type TR. You can join this relation to sys.server_triggers. The tuple (object_id) is the key of the relation.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	This is a FOREIGN KEY reference back to the server-level trigger where this module is defined.
definition	nvarchar(max)	SQL text that defines this module. NULL = Encrypted.
uses_ansi_nulls	bit	Module was created with ANSI NULLS set option set to ON.
uses_quoted_identifier	bit	Module was created with QUOTED IDENTIFIER set option set to ON.
execute_as_principal_id	int	ID of the EXECUTE AS server principal. NULL by default or if EXECUTE AS CALLER ID of the specified principal if EXECUTE AS SELF EXECUTE AS principal-2 = EXECUTE AS OWNER.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.server_triggers (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains the set of all server-level DDL triggers with object_type of TR or TA. In the case of CLR triggers, the assembly must be loaded into the **master** database. All server-level DDL trigger names exist in a single, global scope.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the trigger.
object_id	int	ID of the object.
parent_class	tinyint	Class of parent. Is always: 100 = Server
parent_class_desc	nvarchar(60)	Description of class of parent. Is always: SERVER.
parent_id	int	Always 0 for triggers on the SERVER.
type	char(2)	Object type: TA = Assembly (CLR) trigger TR = SQL trigger
type_desc	nvarchar(60)	Description of the class of the object type. CLR_TRIGGER SQL_TRIGGER
create_date	datetime	Date the trigger was created.
modify_date	datetime	Date the trigger was last modified by using an ALTER statement.
is_ms_shipped	bit	Trigger created on behalf of the user by an internal SQL Server component.
is_disabled	bit	1 = Trigger is disabled.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.server_trigger_events (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each event for which a server-level (synchronous) trigger fires.

COLUMN NAME	DATA TYPE	DESCRIPTION
inherited columns		Inherits all columns from sys.server_events .
is_first	bit	Trigger is marked to be the first to fire for this event.
is_last	bit	Trigger is marked to be the last to fire for this event.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.sql_dependencies (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each dependency on a referenced entity as referenced in the Transact-SQL expression or statements that define some other referencing object.

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use [sys.sql_expression_dependencies](#) instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
class	tinyint	Identifies the class of the referenced entity: 0 = Object or column (non-schema-bound references only) 1 = Object or column (schema-bound references) 2 = Types (schema-bound references) 3 = XML Schema collections (schema-bound references) 4 = Partition function (schema-bound references)
class_desc	nvarchar(60)	Description of class of referenced entity: OBJECT_OR_COLUMN_REFERENCE_NON_SCHEMA_BOUND OBJECT_OR_COLUMN_REFERENCE_SCHEMA_BOUND TYPE_REFERENCE XML_SCHEMA_COLLECTION_REFERENCE PARTITION_FUNCTION_REFERENCE
object_id	int	ID of the referencing object.
column_id	int	If the referencing ID is a column, ID of referencing column; otherwise, 0.

COLUMN NAME	DATA TYPE	DESCRIPTION
referenced_major_id	int	<p>ID of the referenced entity, interpreted by value of class, according to:</p> <p>0, 1 = Object ID of object or column.</p> <p>2 = Type ID.</p> <p>3 = XML Schema collection ID.</p>
referenced_minor_id	int	<p>Minor-ID of the referenced entity, interpreted by value of class, as shown in the following.</p> <p>When class =:</p> <p>0, referenced_minor_id is a column ID; or if not a column, it is 0.</p> <p>1, referenced_minor_id is a column ID; or if not a column, it is 0.</p> <p>Otherwise, referenced_minor_id = 0.</p>
is_selected	bit	Object or column is selected.
is_updated	bit	Object or column is updated.
is_select_all	bit	Object is used in SELECT * clause (object-level only).

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also





[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.sql_expression_dependencies (Transact-SQL)

11/21/2017 • 7 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each by-name dependency on a user-defined entity in the current database. This includes dependences between natively compiled, scalar user-defined functions and other SQL Server modules. A dependency between two entities is created when one entity, called the *referenced entity*, appears by name in a persisted SQL expression of another entity, called the *referencing entity*. For example, when a table is referenced in the definition of a view, the view, as the referencing entity, depends on the table, the referenced entity. If the table is dropped, the view is unusable.

For more information, see [Scalar User-Defined Functions for In-Memory OLTP](#).

You can use this catalog view to report dependency information for the following entities:

- Schema-bound entities.
- Non-schema-bound entities.
- Cross-database and cross-server entities. Entity names are reported; however, entity IDs are not resolved.
- Column-level dependencies on schema-bound entities. Column-level dependencies for non-schema-bound objects can be returned by using [sys.dm_sql_referenced_entities](#).
- Server-level DDL triggers when in the context of the master database.

COLUMN NAME	DATA TYPE	DESCRIPTION
referencing_id	int	ID of the referencing entity. Is not nullable.
referencing_minor_id	int	Column ID when the referencing entity is a column; otherwise 0. Is not nullable.
referencing_class	tinyint	Class of the referencing entity. 1 = Object or column 12 = Database DDL trigger 13 = Server DDL trigger Is not nullable.
referencing_class_desc	nvarchar(60)	Description of the class of referencing entity. OBJECT_OR_COLUMN DATABASE_DDL_TRIGGER SERVER_DDL_TRIGGER Is not nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_schema_bound_reference	bit	<p>1 = Referenced entity is schema-bound.</p> <p>0 = Referenced entity is non-schema-bound.</p> <p>Is not nullable.</p>
referenced_class	tinyint	<p>Class of the referenced entity.</p> <p>1 = Object or column</p> <p>6 = Type</p> <p>10 = XML schema collection</p> <p>21 = Partition function</p> <p>Is not nullable.</p>
referenced_class_desc	nvarchar(60)	<p>Description of class of referenced entity.</p> <p>OBJECT_OR_COLUMN</p> <p>TYPE</p> <p>XML_SCHEMA_COLLECTION</p> <p>PARTITION_FUNCTION</p> <p>Is not nullable.</p>
referenced_server_name	sysname	<p>Name of the server of the referenced entity.</p> <p>This column is populated for cross-server dependencies that are made by specifying a valid four-part name. For information about multipart names, see Transact-SQL Syntax Conventions (Transact-SQL).</p> <p>NULL for non-schema-bound entities for which the entity was referenced without specifying a four-part name.</p> <p>NULL for schema-bound entities because they must be in the same database and therefore can only be defined using a two-part (<i>schema.object</i>) name.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
referenced_database_name	sysname	<p>Name of the database of the referenced entity.</p> <p>This column is populated for cross-database or cross-server references that are made by specifying a valid three-part or four-part name.</p> <p>NULL for non-schema-bound references when specified using a one-part or two-part name.</p> <p>NULL for schema-bound entities because they must be in the same database and therefore can only be defined using a two-part (<i>schema.object</i>) name.</p>
referenced_schema_name	sysname	<p>Schema in which the referenced entity belongs.</p> <p>NULL for non-schema-bound references in which the entity was referenced without specifying the schema name.</p> <p>Never NULL for schema-bound references because schema-bound entities must be defined and referenced by using a two-part name.</p>
referenced_entity_name	sysname	<p>Name of the referenced entity. Is not nullable.</p>
referenced_id	int	<p>ID of the referenced entity. The value of this column is never NULL for schema-bound references. The value of this column is always NULL for cross-server and cross-database references.</p> <p>NULL for references within the database if the ID cannot be determined. For non-schema-bound references, the ID cannot be resolved in the following cases:</p> <p>The referenced entity does not exist in the database.</p> <p>The schema of the referenced entity depends on the schema of the caller and is resolved at run time. In this case, <i>is_caller_dependent</i> is set to 1.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
referenced_minor_id	int	<p>ID of the referenced column when the referencing entity is a column; otherwise 0. Is not nullable.</p> <p>A referenced entity is a column when a column is identified by name in the referencing entity, or when the parent entity is used in a SELECT * statement.</p>
is_caller_dependent	bit	<p>Indicates that schema binding for the referenced entity occurs at runtime; therefore, resolution of the entity ID depends on the schema of the caller. This occurs when the referenced entity is a stored procedure, extended stored procedure, or a non-schema-bound user-defined function called in an EXECUTE statement.</p> <p>1 = The referenced entity is caller dependent and is resolved at runtime. In this case, referenced_id is NULL.</p> <p>0 = The referenced entity ID is not caller dependent.</p> <p>Always 0 for schema-bound references and for cross-database and cross-server references that explicitly specify a schema name. For example, a reference to an entity in the format <code>EXEC MyDatabase.MySchema.MyProc</code> is not caller dependent. However, a reference in the format <code>EXEC MyDatabase..MyProc</code> is caller dependent.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
is_ambiguous	bit	<p>Indicates the reference is ambiguous and can resolve at run time to a user-defined function, a user-defined type (UDT), or an xquery reference to a column of type xml.</p> <p>For example, assume that the statement</p> <pre>SELECT Sales.GetOrder() FROM Sales.MySales</pre> <p>is defined in a stored procedure. Until the stored procedure is executed, it is not known whether <code>Sales.GetOrder()</code> is a user-defined function in the <code>Sales</code> schema or column named <code>Sales</code> of type UDT with a method named <code>GetOrder()</code>.</p> <p>1 = Reference is ambiguous.</p> <p>0 = Reference is unambiguous or the entity can be successfully bound when the view is called.</p> <p>Always 0 for schema bound references.</p>

Remarks

The following table lists the types of entities for which dependency information is created and maintained. Dependency information is not created or maintained for rules, defaults, temporary tables, temporary stored procedures, or system objects.

ENTITY TYPE	REFERENCING ENTITY	REFERENCED ENTITY
Table	Yes*	Yes
View	Yes	Yes
Filtered index	Yes**	No
Filtered statistics	Yes**	No
Transact-SQL stored procedure***	Yes	Yes
CLR stored procedure	No	Yes
Transact-SQL user-defined function	Yes	Yes
CLR user-defined function	No	Yes
CLR trigger (DML and DDL)	No	No
Transact-SQL DML trigger	Yes	No

ENTITY TYPE	REFERENCING ENTITY	REFERENCED ENTITY
Transact-SQL database-level DDL trigger	Yes	No
Transact-SQL server-level DDL trigger	Yes	No
Extended stored procedures	No	Yes
Queue	No	Yes
Synonym	No	Yes
Type (alias and CLR user-defined type)	No	Yes
XML schema collection	No	Yes
Partition function	No	Yes

* A table is tracked as a referencing entity only when it references a Transact-SQL module, user-defined type, or XML schema collection in the definition of a computed column, CHECK constraint, or DEFAULT constraint.

** Each column used in the filter predicate is tracked as a referencing entity.

*** Numbered stored procedures with an integer value greater than 1 are not tracked as either a referencing or referenced entity.

Permissions

Requires VIEW DEFINITION permission on the database and SELECT permission on sys.sql_expression_dependencies for the database. By default, SELECT permission is granted only to members of the db_owner fixed database role. When SELECT and VIEW DEFINITION permissions are granted to another user, the grantee can view all dependencies in the database.

Examples

A. Returning entities that are referenced by another entity

The following example returns the tables and columns referenced in the view `Production.vProductAndDescription`. The view depends on the entities (tables and columns) returned in the `referenced_entity_name` and `referenced_column_name` columns.

```
USE AdventureWorks2012;
GO
SELECT OBJECT_NAME(referencing_id) AS referencing_entity_name,
       o.type_desc AS referencing_description,
       COALESCE(COL_NAME(referencing_id, referencing_minor_id), '(n/a)') AS referencing_minor_id,
       referencing_class_desc,
       referenced_server_name, referenced_database_name, referenced_schema_name,
       referenced_entity_name,
       COALESCE(COL_NAME(referenced_id, referenced_minor_id), '(n/a)') AS referenced_column_name,
       is_caller_dependent, is_ambiguous
FROM sys.sql_expression_dependencies AS sed
INNER JOIN sys.objects AS o ON sed.referencing_id = o.object_id
WHERE referencing_id = OBJECT_ID(N'Production.vProductAndDescription');
GO
```

B. Returning entities that reference another entity

The following example returns the entities that reference the table `Production.Product`. The entities returned in the `referencing_entity_name` column depend on the `Product` table.

```
USE AdventureWorks2012;
GO
SELECT OBJECT_SCHEMA_NAME ( referencing_id ) AS referencing_schema_name,
       OBJECT_NAME(referencing_id) AS referencing_entity_name,
       o.type_desc AS referencing_description,
       COALESCE(COL_NAME(referencing_id, referencing_minor_id), '(n/a)') AS referencing_minor_id,
       referencing_class_desc, referenced_class_desc,
       referenced_server_name, referenced_database_name, referenced_schema_name,
       referenced_entity_name,
       COALESCE(COL_NAME(referenced_id, referenced_minor_id), '(n/a)') AS referenced_column_name,
       is_caller_dependent, is_ambiguous
FROM sys.sql_expression_dependencies AS sed
INNER JOIN sys.objects AS o ON sed.referencing_id = o.object_id
WHERE referenced_id = OBJECT_ID(N'Production.Product');
GO
```

C. Returning cross-database dependencies

The following example returns all cross-database dependencies. The example first creates the database `db1` and two stored procedures that reference tables in the databases `db2` and `db3`. The `sys.sql_expression_dependencies` table is then queried to report the cross-database dependencies between the procedures and the tables. Notice that NULL is returned in the `referenced_schema_name` column for the referenced entity `t3` because a schema name was not specified for that entity in the definition of the procedure.

```
CREATE DATABASE db1;
GO
USE db1;
GO
CREATE PROCEDURE p1 AS SELECT * FROM db2.s1.t1;
GO
CREATE PROCEDURE p2 AS
    UPDATE db3..t3
    SET c1 = c1 + 1;
GO
SELECT OBJECT_NAME (referencing_id),referenced_database_name,
       referenced_schema_name, referenced_entity_name
FROM sys.sql_expression_dependencies
WHERE referenced_database_name IS NOT NULL;
GO
USE master;
GO
DROP DATABASE db1;
GO
```





See Also

[sys.dm_sql_referenced_entities \(Transact-SQL\)](#)

[sys.dm_sql_referencing_entities \(Transact-SQL\)](#)

sys.sql_modules (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each object that is an SQL language-defined module in SQL Server, including natively compiled scalar user-defined function. Objects of type P, RF, V, TR, FN, IF, TF, and R have an associated SQL module. Stand-alone defaults, objects of type D, also have an SQL module definition in this view. For a description of these types, see the **type** column in the [sys.objects](#) catalog view.

For more information, see [Scalar User-Defined Functions for In-Memory OLTP](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object of the containing object. Is unique within a database.
definition	nvarchar(max)	SQL text that defines this module. NULL = Encrypted.
uses_ansi_nulls	bit	Module was created with SET ANSI_NULLS ON. Will always be = 0 for rules and defaults.
uses_quoted_identifier	bit	Module was created with SET QUOTED_IDENTIFIER ON.
is_schema_bound	bit	Module was created with SCHEMABINDING option. Always contains a value of 1 for natively compiled stored procedures.
uses_database_collation	bit	1 = Schema-bound module definition depends on the default-collation of the database for correct evaluation; otherwise, 0. Such a dependency prevents changing the database's default collation.
is_recompiled	bit	Procedure was created WITH RECOMPILE option.
null_on_null_input	bit	Module was declared to produce a NULL output on any NULL input.

COLUMN NAME	DATA TYPE	DESCRIPTION
execute_as_principal_id	Int	<p>ID of the EXECUTE AS database principal.</p> <p>NULL by default or if EXECUTE AS CALLER.</p> <p>ID of the specified principal if EXECUTE AS SELF or EXECUTE AS <principal>.</p> <p>-2 = EXECUTE AS OWNER.</p>
uses_native_compilation	bit	<p>Applies to: SQL Server 2014 through SQL Server 2014.</p> <p>0 = not natively compiled</p> <p>1 = is natively compiled</p> <p>The default value is 0.</p>

Remarks

The SQL expression for a DEFAULT constraint, object of type D, is found in the [sys.default_constraints](#) catalog view. The SQL expression for a CHECK constraint, object of type C, is found in the [sys.check_constraints](#) catalog view.

This information is also described in [sys.dm_db_uncontained_entities](#) (Transact-SQL).

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns the name, type, and definition of each module in the current database.

```
SELECT sm.object_id, OBJECT_NAME(sm.object_id) AS object_name, o.type, o.type_desc, sm.definition
FROM sys.sql_modules AS sm
JOIN sys.objects AS o ON sm.object_id = o.object_id
ORDER BY o.type;
GO
```

See Also

[Catalog Views \(Transact-SQL\)](#)





[Object Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.stats (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each statistics object that exists for the tables, indexes, and indexed views in the database in SQL Server. Every index will have a corresponding statistics row with the same name and ID (**index_id** = **stats_id**), but not every statistics row has a corresponding index.

The catalog view [sys.stats_columns](#) provides statistics information for each column in the database. For more information about statistics, see [Statistics](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which these statistics belong.
name	sysname	Name of the statistics. Is unique within the object.
stats_id	int	ID of the statistics. Is unique within the object.
auto_created	bit	Indicates whether the statistics were automatically created by SQL Server. 0 = Statistics were not automatically created by SQL Server. 1 = Statistics were automatically created by SQL Server.
user_created	bit	Indicates whether the statistics were created by a user. 0 = Statistics were not created by a user. 1 = Statistics were created by a user.
no_recompute	bit	Indicates whether the statistics were created with the NORECOMPUTE option. 0 = Statistics were not created with the NORECOMPUTE option. 1 = Statistics were created with the NORECOMPUTE option.

COLUMN NAME	DATA TYPE	DESCRIPTION
has_filter	bit	<p>0 = Statistics do not have a filter and are computed on all rows.</p> <p>1 = Statistics have a filter and are computed only on rows that satisfy the filter definition.</p>
filter_definition	nvarchar(max)	<p>Expression for the subset of rows included in filtered statistics.</p> <p>NULL = Non-filtered statistics.</p>
is_temporary	bit	<p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>Indicate whether the statistics is temporary. Temporary statistics support Always On availability groups secondary databases that are enabled for read-only access.</p> <p>0 = The statistics is not temporary.</p> <p>1 = The statistics is temporary.</p>
is_incremental	bit	<p>Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>Indicate whether the statistics are created as incremental statistics.</p> <p>0 = The statistics are not incremental.</p> <p>1 = The statistics are incremental.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following examples returns all the statistics and statistics columns for the `HumanResources.Employee` table.

```
USE AdventureWorks2012;
GO
SELECT s.name AS statistics_name
      ,c.name AS column_name
      ,sc.stats_column_id
FROM sys.stats AS s
INNER JOIN sys.stats_columns AS sc
    ON s.object_id = sc.object_id AND s.stats_id = sc.stats_id
INNER JOIN sys.columns AS c
    ON sc.object_id = c.object_id AND c.column_id = sc.column_id
WHERE s.object_id = OBJECT_ID('HumanResources.Employee');
```

See Also





[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.stats_columns (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column that is part of **sys.stats** statistics.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object of which this column is part.
stats_id	int	ID of the statistics of which this column is part.
stats_column_id	int	1-based ordinal within set of stats columns.
column_id	int	ID of the column from sys.columns .

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[sys.stats \(Transact-SQL\)](#)

sys.synonyms (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each synonym object that is **sys.objects.type** = SN.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.objects >		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
base_object_name	nvarchar(1035)	Fully quoted name of the object to which the user of this synonym is redirected.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.system_columns (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column of system objects that have columns.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this column belongs.
name	sysname	Name of the column. Is unique within the object.
column_id	int	ID of the column. Is unique within the object. Column IDs might not be sequential.
system_type_id	tinyint	ID of the system-type of the column
user_type_id	int	ID of the type of the column as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length (in bytes) of column. -1 = Column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml . For text columns, the max_length value will be 16 or the value set by sp_tableoption 'text in row'.
precision	tinyint	Precision of the column if numeric-based; otherwise, 0.
scale	tinyint	Scale of the column if numeric-based; otherwise, 0.
collation_name	sysname	Name of the collation of the column if character-based; otherwise, NULL.
is_nullable	bit	1 = Column is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_ansi_padded	bit	1 = Column uses ANSI_PADDING ON behavior if character, binary, or variant. 0 = Column is not character, binary, or variant.
is_rowguidcol	bit	1 = Column is a declared ROWGUIDCOL.
is_identity	bit	1 = Column has identity values.
is_computed	bit	1 = Column is a computed column.
is_filestream	bit	1 = Column is declared to use filestream storage.
is_replicated	bit	1 = Column is replicated.
is_non_sql_subscribed	bit	1 = Column has a non- SQL Server subscriber.
is_merge_published	bit	1 = Column is merge-published.
is_dts_replicated	bit	1 = Column is replicated by using SSIS.
is_xml_document	bit	1 = Content is a complete XML document. 0 = Content is a document fragment, or the column data type is not xml .
xml_collection_id	int	Non-zero if the column data type is xml and the XML is typed. The value will be the ID of the collection containing the validating XML schema namespace of the column. 0 = No XML schema collection.
default_object_id	int	ID of the default object, regardless of whether it is a stand-alone sys.sp_bindefault , or an inline, column-level DEFAULT constraint. The parent_object_id column of an inline column-level default object is a reference back to the table itself. Or, 0 if there is no default.
rule_object_id	int	ID of the stand-alone rule bound to the column by using sys.sp_bindrule . 0 = No stand-alone rule. For column-level CHECK constraints, see sys.check_constraints (Transact-SQL) .

COLUMN NAME	DATA TYPE	DESCRIPTION
is_sparse	bit	1 = Column is a sparse column. For more information, see Use Sparse Columns .
is_column_set	bit	1 = Column is a column set. For more information, see Use Column Sets .
generated_always_type	tinyint	The numeric value representing the type of column: 0 = NOT_APPLICABLE 1 = AS_ROW_START 2 = AS_ROW_END
generated_always_type_desc	nvarchar(60)	The text description of the type of column: NOT_APPLICABLE AS_ROW_START AS_ROW_END Applies to: SQL Server 2016 through SQL Server 2017.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)





[sys.columns \(Transact-SQL\)](#)

[sys.all_columns \(Transact-SQL\)](#)

[sys.computed_columns \(Transact-SQL\)](#)

sys.system_objects (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for all schema-scoped system objects that are included with Microsoft SQL Server. All system objects are contained in the schemas named sys or INFORMATION_SCHEMA.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Object name.
object_id	int	Object identification number. Is unique within a database.
principal_id	int	<p>ID of the individual owner if different from the schema owner. By default, schema-contained objects are owned by the schema owner. However, another owner can be specified by using the ALTER AUTHORIZATION statement to change ownership.</p> <p>Is NULL if there is no other individual owner.</p> <p>Is NULL if the object type is one of the following:</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>TA = Assembly (CLR) trigger</p> <p>TR = SQL trigger</p> <p>UQ = UNIQUE constraint</p>
schema_id	int	<p>ID of the schema that the object is contained in.</p> <p>For all schema-scoped system objects that included with SQL Server, this value will always be in (schema_id('sys'), schema_id('INFORMATION_SCHEMA'))</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
parent_object_id	int	<p>ID of the object to which this object belongs.</p> <p>0 = Not a child object.</p>
type	char(2)	<p>Object type:</p> <p>AF = Aggregate function (CLR)</p> <p>C = CHECK constraint</p> <p>D = DEFAULT (constraint or stand-alone)</p> <p>F = FOREIGN KEY constraint</p> <p>FN = SQL scalar function</p> <p>FS = Assembly (CLR) scalar-function</p> <p>FT = Assembly (CLR) table-valued function</p> <p>IF = SQL inline table-valued function</p> <p>IT = Internal table</p> <p>P = SQL Stored Procedure</p> <p>PC = Assembly (CLR) stored-procedure</p> <p>PG = Plan guide</p> <p>PK = PRIMARY KEY constraint</p> <p>R = Rule (old-style, stand-alone)</p> <p>RF = Replication-filter-procedure</p> <p>S = System base table</p> <p>SN = Synonym</p> <p>SQ = Service queue</p> <p>TA = Assembly (CLR) DML trigger</p> <p>TF = SQL table-valued-function</p> <p>TR = SQL DML trigger</p> <p>TT = Table type</p> <p>U = Table (user-defined)</p> <p>UQ = UNIQUE constraint</p> <p>V = View</p> <p>X = Extended stored procedure</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	Description of the object type. AGGREGATE_FUNCTION CHECK_CONSTRAINT DEFAULT_CONSTRAINT FOREIGN_KEY_CONSTRAINT SQL_SCALAR_FUNCTION CLR_SCALAR_FUNCTION CLR_TABLE_VALUED_FUNCTION SQL_INLINE_TABLE_VALUED_FUNCTION INTERNAL_TABLE SQL_STORED_PROCEDURE CLR_STORED_PROCEDURE PLAN_GUIDE PRIMARY_KEY_CONSTRAINT RULE REPLICATION_FILTER_PROCEDURE SYSTEM_TABLE SYNONYM SERVICE_QUEUE CLR_TRIGGER SQL_TABLE_VALUED_FUNCTION SQL_TRIGGER TABLE_TYPE USER_TABLE UNIQUE_CONSTRAINT VIEW EXTENDED_STORED_PROCEDURE
create_date	datetime	Date the object was created.

COLUMN NAME	DATA TYPE	DESCRIPTION
modify_date	datetime	Date the object was last modified by using an ALTER statement. If the object is a table or a view, modify_date also changes when a clustered index on the table or view is created or altered.
is_ms_shipped	bit	Object is created by an internal Microsoft SQL Server component.
is_published	bit	Object is published.
is_schema_published	bit	Only the schema of the object is published.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

sys.system_parameters (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each system object that has parameters.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this parameter belongs.
name	sysname	Name of the parameter. Is unique within the object. If the object is a scalar function, the parameter name is an empty string in the row representing the return value.
parameter_id	int	ID of the parameter. Is unique within the object. If the object is a scalar function, parameter_id = 0 represents the return value.
system_type_id	tinyint	ID of the system type of the parameter.
user_type_id	int	ID of the type of the parameter as defined by the user. To return the name of the type, join to the sys.types catalog view on this column.
max_length	smallint	Maximum length of the parameter, in bytes. Value will be -1 for when column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml .
precision	tinyint	Precision of the parameter if numeric-based; otherwise, 0.
scale	tinyint	Scale of the parameter if numeric-based; otherwise, 0.
is_output	bit	1 = Parameter is output (or return); otherwise, 0.
is_cursor_ref	bit	1 = Parameter is a cursor-reference parameter.

COLUMN NAME	DATA TYPE	DESCRIPTION
has_default_value	bit	<p>1 = Parameter has default value.</p> <p>SQL Server only maintains default values for CLR objects in this catalog view; therefore, this column will always have a value of 0 for Transact-SQL objects. To view the default value of a parameter in a Transact-SQL object, query the definition column of the sys.sql_modules catalog view, or use the OBJECT_DEFINITION system function.</p>
is_xml_document	bit	<p>1 = Content is a complete XML document.</p> <p>0 = Content is a document fragment or the data type of the column is not xml.</p>
default_value	sql_variant	<p>If has_default_value is 1, the value of this column is the value of the default for the parameter; otherwise NULL.</p>
xml_collection_id	int	<p>Non-zero if the data type of the parameter is xml and the XML is typed. The value is the ID of the collection that contains the validating XML schema namespace for the parameter.</p> <p>0 = There is no XML schema collection.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)





[Querying the SQL Server System Catalog FAQ](#)

[sys.parameters \(Transact-SQL\)](#)

[sys.all_parameters \(Transact-SQL\)](#)

sys.system_sql_modules (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row per system object that contains an SQL language-defined module. System objects of type FN, IF, P, PC, TF, V have an associated SQL module. To identify the containing object, you can join this view to [sys.system_objects](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	Object identification number of the containing object, unique within a database.
definition	nvarchar(max)	SQL text that defines this module.
uses_ansi_nulls	bit	1 = Module was created with the SET ANSI_NULLS database option ON. Always returns 1.
uses_quoted_identifier	bit	1 = Module was created with SET QUOTED_IDENTIFIER ON. Always returns 1.
is_schema_bound	bit	0 = Module was not created with the SCHEMABINDING option. Always returns 0.
uses_database_collation	bit	0 = Module does not depend on the default collation of the database. Always returns 0.
is_recompiled	bit	0 = Procedure was not created by using the WITH RECOMPILE option. Always returns 0.
null_on_null_input	bit	0 = Module was not created to produce a NULL output on any NULL input. Always returns 0.
execute_as_principal_id	int	Always returns NULL

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user

has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[sys.sql_modules](#) (Transact-SQL)

[sys.all_sql_modules](#) (Transact-SQL)

[Catalog Views](#) (Transact-SQL)

[Object Catalog Views](#) (Transact-SQL)

sys.system_views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each system view that is shipped with SQL Server 2017. All system views are contained in the schemas named **sys** or **INFORMATION_SCHEMA**.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
is_replicated	bit	1 = View is replicated.
has_replication_filter	bit	1 = View has a replication filter.
has_opaque_metadata	bit	1 = VIEW_METADATA option specified for view. For more information, see CREATE VIEW (Transact-SQL) .
has_unchecked_assembly_data	bit	1 = Table contains persisted data that depends on an assembly whose definition changed during the last ALTER ASSEMBLY. Will be reset to 0 after the next successful DBCC CHECKDB or DBCC CHECKTABLE.
with_check_option	bit	1 = WITH CHECK OPTION was specified in the view definition.
is_date_correlation_view	bit	1 = View was created automatically by the system to store correlation information between datetime columns. Creation of this view was enabled by setting DATE_CORRELATION_OPTIMIZATION to ON.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)
[Catalog Views \(Transact-SQL\)](#)
[DBCC CHECKDB \(Transact-SQL\)](#)
[DBCC CHECKTABLE \(Transact-SQL\)](#)
[ALTER ASSEMBLY \(Transact-SQL\)](#)

sys.table_types (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays properties of user-defined table types in SQL Server. A table type is a type from which table variables or table-valued parameters could be declared. Each table type has a **type_table_object_id** that is a foreign key into the [sys.objects](#) catalog view. You can use this ID column to query various catalog views, in a way that is similar to an **object_id** column of a regular table, to discover the structure of the table type such as its columns and constraints.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.types (Transact-SQL) .
type_table_object_id	int	Object identification number. This number is unique within a database.
is_memory_optimized	bit	<p>Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>The following are the possible values:</p> <p>0 = is not memory optimized</p> <p>1 = is memory optimized</p> <p>A value of 0 is the default value.</p> <p>Table types are always created with DURABILITY = SCHEMA_ONLY. Only the schema is persisted on disk.</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Object Catalog Views \(Transact-SQL\)](#)

[Use Table-Valued Parameters \(Database Engine\)](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.tables (Transact-SQL)

11/21/2017 • 5 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each user table in SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
lob_data_space_id	int	<p>A nonzero value is the ID of the data space (filegroup or partition scheme) that holds the large object binary (LOB) data for this table. Examples of LOB data types include varbinary(max), varchar(max), geography, or xml.</p> <p>0 = The table does not LOB data.</p>
filestream_data_space_id	int	<p>Is the data space ID for a FILESTREAM filegroup or a partition scheme that consists of FILESTREAM filegroups.</p> <p>To report the name of a FILESTREAM filegroup, execute the query</p> <pre>SELECT FILEGROUP_NAME (filestream_data_space_id) FROM sys.tables</pre> <p>.</p> <p>sys.tables can be joined to the following views on filestream_data_space_id = data_space_id.</p> <ul style="list-style-type: none">- sys.filegroups- sys.partition_schemes- sys.indexes- sys.allocation_units- sys.fulltext_catalogs- sys.data_spaces- sys.destination_data_spaces- sys.master_files- sys.database_files- backupfilegroup (join on filegroup_id)

COLUMN NAME	DATA TYPE	DESCRIPTION
max_column_id_used	int	Maximum column ID ever used by this table.
lock_on_bulk_load	bit	Table is locked on bulk load. For more information, see sp_tableoption (Transact-SQL) .
uses_ansi_nulls	bit	Table was created with the SET ANSI_NULLS database option ON.
is_replicated	bit	1 = Table is published using snapshot replication or transactional replication.
has_replication_filter	bit	1 = Table has a replication filter.
is_merge_published	bit	1 = Table is published using merge replication.
is_sync_tran_subscribed	bit	1 = Table is subscribed using an immediate updating subscription.
has_unchecked_assembly_data	bit	1 = Table contains persisted data that depends on an assembly whose definition changed during the last ALTER ASSEMBLY. Will be reset to 0 after the next successful DBCC CHECKDB or DBCC CHECKTABLE.
text_in_row_limit	int	<p>The maximum bytes allowed for text in row.</p> <p>0 = Text in row option is not set. For more information, see sp_tableoption (Transact-SQL).</p>
large_value_types_out_of_row	bit	1 = Large value types are stored out-of-row. For more information, see sp_tableoption (Transact-SQL) .
is_tracked_by_cdc	bit	1 = Table is enabled for change data capture. For more information, see sys.sp_cdc_enable_table (Transact-SQL) .
lock_escalation	tinyint	<p>The value of the LOCK_ESCALATION option for the table:</p> <p>0 = TABLE</p> <p>1 = DISABLE</p> <p>2 = AUTO</p>
lock_escalation_desc	nvarchar(60)	A text description of the lock_escalation option for the table. Possible values are: TABLE, AUTO, and DISABLE.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_filetable	bit	<p>Applies to: SQL Server 2012 through SQL Server 2017 and Azure SQL Database.</p> <p>1 = Table is a FileTable.</p> <p>For more information about FileTables, see FileTables (SQL Server).</p>
durability	tinyint	<p>Applies to: SQL Server 2014 through SQL Server 2017 and Azure SQL Database.</p> <p>The following are possible values:</p> <p>0 = SCHEMA_AND_DATA</p> <p>1 = SCHEMA_ONLY</p> <p>The value of 0 is the default value.</p>
durability_desc	nvarchar(60)	<p>Applies to: SQL Server 2014 through SQL Server 2017 and Azure SQL Database.</p> <p>The following are the possible values:</p> <p>SCHEMA_ONLY</p> <p>SCHEMA_AND_DATA</p> <p>The value of SCHEMA_AND_DATA indicates that the table is a durable, in-memory table. SCHEMA_AND_DATA is the default value for memory optimized tables. The value of SCHEMA_ONLY indicates that the table data will not be persisted upon restart of the database with memory optimized objects.</p>
is_memory_optimized	bit	<p>Applies to: SQL Server 2014 through SQL Server 2017 and Azure SQL Database.</p> <p>The following are the possible values:</p> <p>0 = not memory optimized.</p> <p>1 = is memory optimized.</p> <p>A value of 0 is the default value.</p> <p>Memory optimized tables are in-memory user tables, the schema of which is persisted on disk similar to other user tables. Memory optimized tables can be accessed from natively compiled stored procedures.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
temporal_type	tinyint	<p>Applies to: SQL Server 2016 through SQL Server 2017 and Azure SQL Database.</p> <p>The numeric value representing the type of table:</p> <p>0 = NON_TEMPORAL_TABLE</p> <p>1 = HISTORY_TABLE</p> <p>2 = SYSTEM_VERSIONED_TEMPORAL_TABLE</p>
temporal_type_desc	nvarchar(60)	<p>Applies to: SQL Server 2016 through SQL Server 2017 and Azure SQL Database.</p> <p>The text description of the type of table:</p> <p>NON_TEMPORAL_TABLE</p> <p>HISTORY_TABLE</p> <p>SYSTEM_VERSIONED_TEMPORAL_TABLE</p>
history_table_id	int	<p>Applies to: SQL Server 2016 through SQL Server 2017 and Azure SQL Database.</p> <p>When temporal_type IN (2, 4) returns object_id of the table that maintains historical data, otherwise returns NULL.</p>
is_remote_data_archive_enabled	bit	<p>Applies to: SQL Server 2016 through SQL Server 2017 and Azure SQL Database</p> <p>Indicates whether the table is Stretch-enabled.</p> <p>0 = The table is not Stretch-enabled.</p> <p>1 = The table is Stretch-enabled.</p> <p>For more info, see Stretch Database.</p>
is_external	bit	<p>Applies to: SQL Server 2016 through SQL Server 2017, Azure SQL Database, and Azure SQL Data Warehouse.</p> <p>Indicates table is an external table.</p> <p>0 = The table is not an external table.</p> <p>1 = The table is an external table.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
history_retention_period	int	Applies to: Azure SQL Database. The numeric value representing duration of the temporal history retention period in units specified with history_retention_period_unit.
history_retention_period_unit	int	Applies to: Azure SQL Database. The numeric value representing type of temporal history retention period unit. -1 :INFINITE 3: DAY 4: WEEK 5: MONTH 6: YEAR
history_retention_period_unit_desc	nvarchar(10)	Applies to: Azure SQL Database. The text description of type of temporal history retention period unit. INFINITE DAY WEEK MONTH YEAR

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns all of the user tables that do not have a primary key.

```
SELECT SCHEMA_NAME(schema_id) AS schema_name
       ,name AS table_name
FROM sys.tables
WHERE OBJECTPROPERTY(object_id,'TableHasPrimaryKey') = 0
ORDER BY schema_name, table_name;
GO
```

The following example shows how related temporal data can be exposed.

Applies to: SQL Server 2016 through SQL Server 2017 and Azure SQL Database.

```
SELECT T1.object_id, T1.name as TemporalTableName, SCHEMA_NAME(T1.schema_id) AS TemporalTableSchema,  
T2.name as HistoryTableName, SCHEMA_NAME(T2.schema_id) AS HistoryTableSchema,  
T1.temporal_type_desc  
FROM sys.tables T1  
LEFT JOIN sys.tables T2  
ON T1.history_table_id = T2.object_id  
ORDER BY T1.temporal_type desc
```

The following example shows how information on temporal history retention can be exposed.

Applies to: Azure SQL Database.

```
SELECT DB.is_temporal_history_retention_enabled, SCHEMA_NAME(T1.schema_id) AS TemporalTableSchema,  
T1.name as TemporalTableName, SCHEMA_NAME(T2.schema_id) AS HistoryTableSchema, T2.name as HistoryTableName,  
T1.history_retention_period, T1.history_retention_period_unit_desc  
FROM sys.tables T1  
OUTER APPLY (select is_temporal_history_retention_enabled from sys.databases where name = DB_NAME()) DB  
LEFT JOIN sys.tables T2  
ON T1.history_table_id = T2.object_id WHERE T1.temporal_type = 2
```

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[DBCC CHECKDB \(Transact-SQL\)](#)

[DBCC CHECKTABLE \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

[In-Memory OLTP \(In-Memory Optimization\)](#)

sys.trigger_event_types (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each event or event group on which a trigger can fire.

COLUMN NAME	DATA TYPE	DESCRIPTION
type	int	Type of event or event group that causes a trigger to fire.
type_name	nvarchar(64)	Name of an event or event group. This can be specified in the FOR clause of a CREATE TRIGGER statement.
parent_type	int	Type of event group that is the parent of the event or event group.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.trigger_events (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row per event for which a trigger fires.

NOTE

sys.trigger_events does not apply to event notifications.

COLUMN NAME	DATA TYPE	DESCRIPTION
<Columns inherited from sys.events >	Not applicable	Inherits the object_id , type , type_desc columns from sys.events .
is_first	bit	Trigger is marked to be the first to fire for this event.
is_last	bit	Trigger is marked to be the last to fire for this event.
event_group_type	int	Event group on which the trigger is created, or null if not created on an event group.
event_group_type_desc	nvarchar(60)	Description of the event group on which the trigger is created, or null if not created on an event group.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

sys.triggers (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object that is a trigger, with a type of TR or TA. DML trigger names are schema-scoped and, therefore, are visible in **sys.objects**. DDL trigger names are scoped by the parent entity and are only visible in this view.

The **parent_class** and **name** columns uniquely identify the trigger in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Trigger name. DML trigger names are schema-scoped. DDL trigger names are scoped with respect to the parent entity.
object_id	int	Object identification number. Is unique within a database.
parent_class	tinyint	Class of the parent of the trigger. 0 = Database, for the DDL triggers. 1 = Object or column for the DML triggers.
parent_class_desc	nvarchar(60)	Description of the parent class of the trigger. DATABASE OBJECT_OR_COLUMN
parent_id	int	ID of the parent of the trigger, as follows: 0 = Triggers that are database-parented triggers. For DML triggers, this is the object_id of the table or view on which the DML trigger is defined.
type	char(2)	Object type: TA = Assembly (CLR) trigger TR = SQL trigger

COLUMN NAME	DATA TYPE	DESCRIPTION
type_desc	nvarchar(60)	Description of object type. CLR_TRIGGER SQL_TRIGGER
create_date	datetime	Date the trigger was created.
modify_date	datetime	Date the object was last modified by using an ALTER statement.
is_ms_shipped	bit	Trigger created on behalf of the user by an internal SQL Server component.
is_disabled	bit	Trigger is disabled.
is_not_for_replication	bit	Trigger was created as NOT FOR REPLICATION.
is_instead_of_trigger	bit	1 = INSTEAD OF triggers 0 = AFTER triggers.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Security Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each view object, with **sys.objects.type** = V.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL)
is_replicated	bit	1 = View is replicated.
has_replication_filter	bit	1 = View has a replication filter.
has_opaque_metadata	bit	1 = VIEW_METADATA option specified for view. For more information, see CREATE VIEW (Transact-SQL) .
has_unchecked_assembly_data	bit	1 = View contains persisted data that depends on an assembly whose definition changed during the last ALTER ASSEMBLY. Resets to 0 after the next successful DBCC CHECKDB or DBCC CHECKTABLE.
with_check_option	bit	1 = WITH CHECK OPTION was specified in the view definition.
is_date_correlation_view	bit	1 = View was created automatically by the system to store correlation information between datetime columns. Creation of this view was enabled by setting DATE_CORRELATION_OPTIMIZATION to ON.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[ALTER ASSEMBLY \(Transact-SQL\)](#)




[DBCC CHECKDB \(Transact-SQL\)](#)

[DBCC CHECKTABLE \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

Partition Function Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

This section contains the following catalog views.

sys.partition_functions	sys.partition_range_values
sys.partition_parameters	





See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.partition_functions (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each partition function in SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the partition function. Is unique within the database.
function_id	int	Partition function ID. Is unique within the database.
type	char(2)	Function type. R = Range
type_desc	nvarchar(60)	Function type. RANGE
fanout	int	Number of partitions created by the function.
boundary_value_on_right	bit	For range partitioning. 1 = Boundary value is included in the RIGHT range of the boundary. 0 = LEFT.
is_system		Applies to: SQL Server 2012 through SQL Server 2017. 1 = Object is used for full-text index fragments. 0 = Object is not used for full-text index fragments.
create_date	datetime	Date the function was created.
modify_date	datetime	Date the function was last modified using an ALTER statement.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Partition Function Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[sys.partition_range_values \(Transact-SQL\)](#)

[sys.partition_parameters \(Transact-SQL\)](#)

sys.partition_parameters (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each parameter of a partition function.

COLUMN NAME	DATA TYPE	DESCRIPTION
function_id	int	ID of the partition function to which this parameter belongs.
parameter_id	int	ID of the parameter. Is unique within the partition function, beginning with 1.
system_type_id	tinyint	ID of the system type of the parameter. Corresponds to the system_type_id column of the sys.types catalog view.
max_length	smallint	Maximum length of the parameter in bytes.
precision	tinyint	Precision of the parameter if numeric-based; otherwise, 0.
scale	tinyint	Scale of the parameter if numeric-based; otherwise, 0.
collation_name	sysname	Name of the collation of the parameter if character-based; otherwise, NULL.
user_type_id	int	ID of the type. Is unique within the database. For system data types, user_type_id = system_type_id .

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Partition Function Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[sys.partition_functions \(Transact-SQL\)](#)

[sys.partition_range_values \(Transact-SQL\)](#)

sys.partition_range_values (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each range boundary value of a partition function of type R.

COLUMN NAME	DATA TYPE	DESCRIPTION
function_id	int	ID of the partition function for this range boundary value.
boundary_id	int	ID (1-based ordinal) of the boundary value tuple, with left-most boundary starting at an ID of 1.
parameter_id	int	ID of the parameter of the function to which this value corresponds. The values in this column correspond with those in the parameter_id column of the sys.partition_parameters catalog view for any particular function_id .
value	sql_variant	The actual boundary value.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Partition Function Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[sys.partition_functions \(Transact-SQL\)](#)

[sys.partition_parameters \(Transact-SQL\)](#)

Policy-Based Management Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Policy-Based Management has the following views for displaying information about Policy-Based Management policies, conditions, expressions, groups, and filters. These views are in the msdb database and are owned by the dbo schema.

syspolicy_conditions	syspolicy_policy_group_subscriptions
syspolicy_policies	syspolicy_policy_groups
syspolicy_policy_execution_history	syspolicy_system_health_state
syspolicy_policy_execution_history_details	

See Also

[Administer Servers by Using Policy-Based Management](#)

syspolicy_conditions (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each Policy-Based Management condition in the instance of SQL Server. syspolicy_conditions belongs to the dbo schema in the msdb database. The following table describes the columns in the syspolicy_conditions view.

COLUMN NAME	DATA TYPE	DESCRIPTION
condition_id	int	Identifier of this condition. Each condition represents a collection of one or more condition expressions.
name	sysname	Name of the condition.
date_created	datetime	Date and time the condition was created.
description	nvarchar(max)	Description of the condition. The description column is optional and can be NULL.
created_by	sysname	Login that created the condition.
modified_by	sysname	Login that most recently modified the condition. Is NULL if never modified.
date_modified	datetime	Date and time the condition was created. Is NULL if never modified.
is_name_condition	smallint	Specifies whether the condition is a naming condition. 0 = The condition expression does not contain the @Name variable. 1 = The condition expression contains the @Name variable.
facet	nvarchar(max)	Name of the facet that the condition is based on.
Expression	nvarchar(max)	Expression of the facet states.
obj_name	sysname	The object name assigned to @Name if the condition expression contains this variable.

Remarks

When you are troubleshooting Policy-Based Management, query the syspolicy_conditions view to determine who created or last changed the condition.

Permissions

Requires membership in the PolicyAdministratorRole role in the msdb database.


See Also

[Administer Servers by Using Policy-Based Management](#)

[Policy-Based Management Views \(Transact-SQL\)](#)

syspolicy_policies (Transact-SQL)

11/16/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each Policy-Based Management policy in the instance of SQL Server. syspolicy_policies belongs to the dbo schema in the msdb database. The following table describes the columns in the syspolicy_policies view.

COLUMN NAME	DATA TYPE	DESCRIPTION
policy_id	int	Identifier of the policy.
name	sysname	Name of the policy.
condition_id	int	ID of the condition enforced or tested by this policy.
root_condition_id	int	For internal use only.
date_created	datetime	Date and time the policy was created.
execution_mode	int	<p>Evaluation mode for the policy. Possible values are as follows:</p> <p>0 = On demand</p> <p>This mode evaluates the policy when directly specified by the user.</p> <p>1 = On change: prevent</p> <p>This automated mode uses DDL triggers to prevent policy violations.</p> <p>2 = On change: log only</p> <p>This automated mode uses event notification to evaluate a policy when a relevant change occurs and logs policy violations.</p> <p>4 = On schedule</p> <p>This automated mode uses a SQL Server Agent job to periodically evaluate a policy. The mode logs policy violations.</p> <p>Note: The value 3 is not a possible value.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
policy_category	int	ID of the Policy-Based Management policy category that this policy belongs to. Is NULL if it is the default policy group.
schedule_uid	uniqueidentifier	When the execution_mode is On schedule, contains the ID of the schedule; otherwise, is NULL.
description	nvarchar(max)	Description of the policy. The description column is optional and can be NULL.
help_text	nvarchar(4000)	The hyperlink text that belongs to help_link.
help_link	nvarchar(2083)	The additional help hyperlink that is assigned to the policy by the policy creator.
object_set_id	int	ID of the object set that the policy evaluates.
is_enabled	bit	Indicates whether the policy is currently enabled (1) or disabled (0).
job_id	uniqueidentifier	When the execution_mode is On schedule, contains the ID of the SQL Server Agent job that runs the policy.
created_by	sysname	Login that created the policy.
modified_by	sysname	Login that most recently modified the policy. Is NULL if never modified.
date_modified	datetime	Date and time the policy was created. Is NULL if never modified.

Remarks

When you are troubleshooting Policy-Based Management, query the [syspolicy_conditions](#) view to determine whether the policy is enabled. This view also displays who created or last changed the policy.

Permissions





Requires membership in the PolicyAdministratorRole role in the msdb database.

See Also

[Administer Servers by Using Policy-Based Management](#)
[Policy-Based Management Views \(Transact-SQL\)](#)

syspolicy_policy_execution_history (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays the time when policies were executed, the result of each execution, and details about errors if any occurred. The following table describes the columns in the syspolicy_policy_execution_history view.

COLUMN NAME	DATA TYPE	DESCRIPTION
history_id	bigint	Identifier of this record. Each record indicates a policy and one time that it was initiated.
policy_id	int	Identifier of the policy.
start_date	datetime	Date and time this policy tried to run.
end_date	datetime	Time this policy finished running.
result	bit	Success or failure of the policy. 0 = Failure, 1 = Success.
exception_message	nvarchar(max)	Message generated by the exception if one occurred.
exception	nvarchar(max)	Description of the exception if one occurred.

Remarks

The [syspolicy_policy_execution_history_details](#) view contains related details about the targets of the policy and the condition expressions tested.

Permissions





Requires membership in the PolicyAdministratorRole role in the msdb database.

See Also

[Administer Servers by Using Policy-Based Management](#)
[Policy-Based Management Views \(Transact-SQL\)](#)

syspolicy_policy_execution_history_details (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays the condition expressions that were executed, the targets of the expressions, the result of each execution, and details about errors if any occurred. The following table describes the columns in the syspolicy_execution_history_details view.

COLUMN NAME	DATA TYPE	DESCRIPTION
detail_id	bigint	Identifier of this record. Each record represents the attempt to evaluate or enforce one condition expression in a policy. If applied to multiple targets, each condition will have a detail record for each target.
history_id	bigint	Identifier of the history event. Each history event represents one try to execute a policy. Because a condition can have several condition expressions and several targets, a history_id can create several detail records. Use the history_id column to join this view to the syspolicy_policy_execution_history view.
target_query_expression	nvarchar(max)	Target of the policy and syspolicy_policy_execution_history view.
execution_date	datetime	Date and time that this detail record was created.
result	bit	Success or failure of this target and condition expression evaluation: 0 (success) or 1 (failure).
result_detail	nvarchar(max)	Result message. Only available if provided by the facet.
exception_message	nvarchar(max)	Message generated by the exception if one occurred.
exception	nvarchar(max)	Description of the exception if one occurred.

Remarks

When you are troubleshooting Policy-Based Management, query the syspolicy_policy_execution_history_details

view to determine which target and condition expression combinations failed, when they failed, and review related errors.

The following query combines the `syspolicy_policy_execution_history_details` view with the `syspolicy_policy_execution_history_details` and `syspolicy_policies` views to display the name of the policy, the name of the condition, and details about failures.

```
SELECT Pol.name AS Policy,
Cond.name AS Condition,
PolHistDet.target_query_expression,
PolHistDet.execution_date,
PolHistDet.result,
PolHistDet.result_detail,
PolHistDet.exception_message,
PolHistDet.exception
FROM msdb.dbo.syspolicy_policies AS Pol
JOIN msdb.dbo.syspolicy_conditions AS Cond
    ON Pol.condition_id = Cond.condition_id
JOIN msdb.dbo.syspolicy_policy_execution_history AS PolHist
    ON Pol.policy_id = PolHist.policy_id
JOIN msdb.dbo.syspolicy_policy_execution_history_details AS PolHistDet
    ON PolHist.history_id = PolHistDet.history_id
WHERE PolHistDet.result = 0 ;
```

Permissions

Requires membership in the PolicyAdministratorRole role in the msdb database.



See Also

[Administer Servers by Using Policy-Based Management](#)

[Policy-Based Management Views \(Transact-SQL\)](#)

syspolicy_policy_categories (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each Policy-Based Management policy category in the instance of SQL Server. Policy categories help you organize policies when you have many policies. The following table describes the columns in the syspolicy_policy_groups view.

COLUMN NAME	DATA TYPE	DESCRIPTION
policy_category_id	int	Identifier of the policy category.
name	sysname	Name of the policy category.
mandate_database_subscriptions	bit	Indicates whether the policy category applies to all databases in an instance without an explicit subscription (1) or the policy category must be applied to a database by using an explicit subscription (0).

Remarks

Displays a list of Policy-Based Management policy groups.

Permissions


Requires membership in the PolicyAdministratorRole role in the msdb database.

See Also

[Administer Servers by Using Policy-Based Management](#)
[Policy-Based Management Views \(Transact-SQL\)](#)

syspolicy_policy_category_subscriptions (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each Policy-Based Management subscription in the instance of SQL Server. Each row describes a target and policy category pair. The following table describes the columns in the syspolicy_policy_group_subscriptions view.

COLUMN NAME	DATA TYPE	DESCRIPTION
policy_category_subscription_id	int	Identifier of this record.
target_type	sysname	Type of database object that is the target of this subscription.
target_object	sysname	Name of the target object.
policy_category_id	int	ID of the policy category that is applied to the target.

Remarks

This view shows the targets that are subscribed to policy categories.

Permissions

Requires membership in the PolicyAdministratorRole role in the msdb database.

See Also

[Administer Servers by Using Policy-Based Management](#)

[Policy-Based Management Views \(Transact-SQL\)](#)

sypolicy_system_health_state (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Displays one row for each Policy-Based Management policy and target query expression combination. Use the sypolicy_system_health_state view to programmatically check the policy health of the server. The following table describes the columns in the sypolicy_system_health_state view.

COLUMN NAME	DATA TYPE	DESCRIPTION
health_state_id	bigint	Identifier of the policy health state record.
policy_id	int	Identifier of the policy.
last_run_date	datetime	Date and time the policy was last run.
target_query_expression_with_id	nvarchar(400)	The target expression, with values assigned to identity variables, that defines the target against which the policy is evaluated.
target_query_expression	nvarchar(max)	The expression that defines the target against which the policy is evaluated.
result	bit	Health state of this target with regard to the policy: 0 = Failure 1 = Success

Remarks

The sypolicy_system_health_state view displays the most recent health state of target query expression for each active (enabled) policy. The SQL Server Management Studio Object Explorer and Object Explorer Details page aggregates policy health from this view to show the critical health state.

Permissions





Requires membership in the PolicyAdministratorRole role in the msdb database.

See Also

[Administer Servers by Using Policy-Based Management](#)
[Policy-Based Management Views \(Transact-SQL\)](#)

Resource Governor Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains Resource Governor information for the following catalog views. These views are optimized for performance and utility. When possible, use these catalog views to access Resource Governor metadata.

sys.resource_governor_configuration (Transact-SQL)	sys.resource_governor_resource_pools (Transact-SQL)
sys.resource_governor_external_resource_pools (Transact-SQL)	sys.resource_governor_workload_groups (Transact-SQL)


See Also

[Resource Governor Related Dynamic Management Views \(Transact-SQL\)](#)

[Resource Governor](#)

sys.resource_governor_configuration (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the stored Resource Governor state.

COLUMN NAME	DATA TYPE	DESCRIPTION
classifier_function_id	int	The ID of the classifier function as it is stored in the metadata. Is not nullable. Note This function is used to classify new sessions and uses rules to route the workload to the appropriate workload group. For more information, see Resource Governor .
is_enabled	bit	Indicates the current state of the Resource Governor: 0 = Resource Governor is not enabled. 1 = Resource Governor is enabled. Is not nullable.
max_outstanding_io_per_volume	int	Applies to: SQL Server 2014 through SQL Server 2017. The maximum number of outstanding I/O per volume.

Remarks

The catalog view displays the Resource Governor configuration as stored in metadata. To see the in-memory configuration use the corresponding dynamic management view.

Permissions

Requires VIEW ANY DEFINITION permission to view contents, requires CONTROL SERVER permission to change contents.

Examples

The following example shows how to get and compare the stored metadata values and the in-memory values of the Resource Governor configuration.

```
USE master;
GO
-- Get the stored metadata.
SELECT
object_schema_name(classifier_function_id) AS 'Classifier UDF schema in metadata',
object_name(classifier_function_id) AS 'Classifier UDF name in metadata'
FROM
sys.resource_governor_configuration;
GO
-- Get the in-memory configuration.
SELECT
object_schema_name(classifier_function_id) AS 'Active classifier UDF schema',
object_name(classifier_function_id) AS 'Active classifier UDF name'
FROM
sys.dm_resource_governor_configuration;
GO
```

See Also

[Resource Governor Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[sys.dm_resource_governor_configuration \(Transact-SQL\)](#)

[Resource Governor](#)

sys.resource_governor_external_resource_pools (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Applies to: SQL Server 2016 R Services (In-Database) and SQL Server 2017 Machine Learning Services (In-Database)

Returns the stored external resource pool configuration in SQL Server. Each row of the view determines the configuration of a pool.

COLUMN NAME	DATA TYPE	DESCRIPTION
pool_id	int	Unique ID of the resource pool. Is not nullable. Note: May be renamed in the future.
name	sysname	Name of the resource pool. Is not nullable.
max_cpu_percent	int	Maximum average CPU bandwidth allowed for all requests in the resource pool when there is CPU contention. Is not nullable.
max_memory_percent	int	Percentage of total server memory that can be used by requests in this resource pool. Is not nullable. The effective maximum depends on the pool minimums. For example, max_memory_percent can be set to 100, but the effective maximum is lower.
max_processes	int	Maximum number of concurrent external processes. The default value, 0, specifies no limit. Is not nullable.
version	bigint	Internal version number.

Permissions

Requires VIEW SERVER STATE permission.

See also

[Resource governance for machine learning in SQL Server](#)

[Resource Governor Catalog Views \(Transact-SQL\)](#)

[sys.dm_resource_governor_resource_pools \(Transact-SQL\)](#)

[Resource Governor](#)

[sys.dm_resource_governor_resource_pool_affinity \(Transact-SQL\)](#)

[external scripts enabled Server Configuration Option](#)

[ALTER EXTERNAL RESOURCE POOL \(Transact-SQL\)](#)

sys.resource_governor_resource_pools (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the stored resource pool configuration in SQL Server. Each row of the view determines the configuration of a pool.

COLUMN NAME	DATA TYPE	DESCRIPTION
pool_id	int	Unique ID of the resource pool. Is not nullable.
name	sysname	Name of the resource pool. Is not nullable.
min_cpu_percent	int	Guaranteed average CPU bandwidth for all requests in the resource pool when there is CPU contention. Is not nullable.
max_cpu_percent	int	Maximum average CPU bandwidth allowed for all requests in the resource pool when there is CPU contention. Is not nullable.
min_memory_percent	int	Guaranteed amount of memory for all requests in the resource pool. This is not shared with other resource pools. Is not nullable.
max_memory_percent	int	Percentage of total server memory that can be used by requests in this resource pool. Is not nullable. The effective maximum depends on the pool minimums. For example, max_memory_percent can be set to 100, but the effective maximum is lower.
cap_cpu_percent	int	Applies to: SQL Server 2012 through SQL Server 2017. Hard cap on the CPU bandwidth that all requests in the resource pool will receive. Limits the maximum CPU bandwidth to the specified level. The allowed range for value is from 1 through 100.

COLUMN NAME	DATA TYPE	DESCRIPTION
min_iops_per_volume	int	<p>Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>The minimum I/O operations per second (IOPS) per volume setting for this pool. 0 = no reservation. Cannot be null.</p>
max_iops_per_volume	int	<p>Applies to: SQL Server 2014 through SQL Server 2017.</p> <p>The maximum I/O operations per second (IOPS) per volume setting for this pool. 0 = unlimited. Cannot be null.</p>

Remarks

The catalog view displays the stored metadata. To see the in-memory configuration, use the corresponding dynamic management view, [sys.dm_resource_governor_resource_pools \(Transact-SQL\)](#).

Permissions

Requires VIEW ANY DEFINITION permission to view contents, requires CONTROL SERVER permission to change contents.

See Also

[Resource Governor Catalog Views \(Transact-SQL\)](#)




[sys.dm_resource_governor_resource_pools \(Transact-SQL\)](#)

[Resource Governor](#)

[sys.resource_governor_external_resource_pools \(Transact-SQL\)](#)

sys.resource_governor_workload_groups (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the stored workload group configuration in SQL Server. Each workload group can subscribe to one and only one resource pool.

COLUMN NAME	DATA TYPE	DESCRIPTION
group_id	int	Unique ID of the workload group. Is not nullable.
name	sysname	Name of the workload group. Is not nullable.
importance	sysname	Note: Importance only applies to workload groups in the same resource pool. Is the relative importance of a request in this workload group. Importance is one of the following, with MEDIUM being the default: LOW, MEDIUM, HIGH. Is not nullable.
request_max_memory_grant_percent	int	Maximum memory grant, as a percentage, for a single request. The default value is 25. Is not nullable. Note: If this setting is higher than 50 percent, large queries will run one at a time. Therefore, there is greater risk of getting an out-of-memory error while the query is running.
request_max_cpu_time_sec	int	Maximum CPU use limit, in seconds, for a single request. The default value, 0, specifies no limit. Is not nullable. Note: For more information, see CPU Threshold Exceeded Event Class .
request_memory_grant_timeout_sec	int	Memory grant time-out, in seconds, for a single request. The default value, 0, uses an internal calculation based on query cost. Is not nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
max_dop	int	Maximum degree of parallelism for the workload group. The default value, 0, uses global settings. Is not nullable. Note: This setting will override the query option maxdop .
group_max_requests	int	Maximum number of concurrent requests. The default value, 0, specifies no limit. Is not nullable.
pool_id	int	ID of the resource pool that this workload group uses.
external_pool_id	int	Applies to: SQL Server 2016 through SQL Server 2017. ID of the external resource pool that this workload group uses.

Remarks

The catalog view displays the stored metadata. To see the in-memory configuration, use the corresponding dynamic management view, [sys.dm_resource_governor_workload_groups \(Transact-SQL\)](#).

The stored and in-memory configuration can be different if the Resource Governor configuration has been changed but the ALTER RESOURCE GOVERNOR RECONFIGURE statement has not been applied.

Permissions

Requires VIEW ANY DEFINITION permission to view contents, requires CONTROL SERVER permission to change contents.

See Also





[sys.dm_resource_governor_workload_groups \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Resource Governor Catalog Views \(Transact-SQL\)](#)

Query Store Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

In this Section

- [sys.database_query_store_options](#) (Transact-SQL)
- [sys.query_context_settings](#) (Transact-SQL)
- [sys.query_store_plan](#) (Transact-SQL)
- [sys.query_store_query](#) (Transact-SQL)
- [sys.query_store_query_text](#) (Transact-SQL)
- [sys.query_store_wait_stats](#) (Transact-SQL)
 - [sys.query_store_runtime_stats](#) (Transact-SQL)
- [sys.query_store_runtime_stats_interval](#) (Transact-SQL)





See Also

[Query Store Stored Procedures](#) (Transact-SQL)

[Monitoring Performance By Using the Query Store](#)

sys.database_query_store_options (Transact-SQL)

11/21/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the Query Store options for this database.

Applies to: SQL Server (SQL Server 2016 through SQL Server 2017), SQL Database.

COLUMN NAME	DATA TYPE	DESCRIPTION
desired_state	smallint	Indicates the desired operation mode of Query Store, explicitly set by user. 0 = OFF 1 = READ_ONLY 2 = READ_WRITE
desired_state_desc	nvarchar(64)	Textual description of the desired operation mode of Query Store: OFF READ_ONLY READ_WRITE
actual_state	smallint	Indicates the operation mode of Query Store. In addition to list of desired states required by the user, actual state can be an error state. 0 = OFF 1 = READ_ONLY 2 = READ_WRITE 3 = ERROR
actual_state_desc	nvarchar(64)	<p>Textual description of the actual operation mode of Query Store. OFF READ_ONLY READ_WRITE ERROR</p> <p>There are situations when actual state is different from the desired state:</p> <p>Query Store may operate in read-only mode even if read-write was specified by the user. For example, that might happen if the database is in read-only mode or if Query Store size exceeded the quota.</p> <p>Extremely rarely, Query Store can end up in ERROR state because of internal errors. If this happens, Query Store could be recovered by executing sp_query_store_consistency_check stored procedure within the affected database.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
readonly_reason	int	<p>When the desired_state_desc is READ_WRITE and the actual_state_desc is READ_ONLY, readonly_reason returns a bit map to indicate why the Query Store is in readonly mode.</p> <p>1 – database is in read-only mode</p> <p>2 – database is in single-user mode</p> <p>4 – database is in emergency mode</p> <p>8 – database is secondary replica (applies to Always On and Azure SQL Database geo-replication). This value can be effectively observed only on readable secondary replicas</p> <p>65536 – the Query Store has reached the size limit set by the MAX_STORAGE_SIZE_MB option.</p> <p>131072 - The number of different statements in Query Store has reached the internal memory limit. Consider removing queries that you do not need or upgrading to a higher service tier to enable transferring Query Store to read-write mode. Only applies to SQL Database.</p> <p>262144 – Size of in-memory items waiting to be persisted on disk has reached the internal memory limit. Query Store will be in read-only mode temporarily until the in-memory items are persisted on disk. Only applies to SQL Database.</p> <p>524288 – Database has reached disk size limit. Query Store is part of user database, so if there is no more available space for a database, that means that Query Store cannot grow further anymore. Only applies to SQL Database.</p> <p>To switch the Query Store operations mode back to read-write, see Verify Query Store is Collecting Query Data Continuously section of Best Practice with the Query Store.</p>
current_storage_size_mb	bigint	Size of Query Store on disk in megabytes.

COLUMN NAME	DATA TYPE	DESCRIPTION
flush_interval_seconds	bigint	<p>Defines period for regular flushing of Query Store data to disk. Default value is 900 (15 min).</p> <p>Change by using the</p> <pre>ALTER DATABASE <database> SET QUERY_STORE (DATA_FLUSH_INTERVAL_SECONDS = <interval>)</pre> <p>statement.</p>
interval_length_minutes	bigint	<p>The statistics aggregation interval. Arbitrary values are not allowed . Use one of the following: 1, 5, 10, 15, 30, 60, and 1440 minutes. The default value is 60 minutes.</p>
max_storage_size_mb	bigint	<p>Maximum disk size for the Query Store. Default value is 100 MB. For SQL Database Premium edition, default is 1Gb and for SQL Database Basic edition, default is 10Mb.</p> <p>Change by using the</p> <pre>ALTER DATABASE <database> SET QUERY_STORE (MAX_STORAGE_SIZE_MB = <size>)</pre> <p>statement.</p>
stale_query_threshold_days	bigint	<p>Number of days that queries with no policy settings are kept in Query Store. Default value is 30. Set to 0 to disable the retention policy. For SQL Database Basic edition, default is 7 days.</p> <p>Change by using the</p> <pre>ALTER DATABASE <database> SET QUERY_STORE (CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = <value>))</pre> <p>statement.</p>
max_plans_per_query	bigint	<p>Limits the maximum number of stored plans. Default value is 200. If the maximum value is reached, Query Store stops capturing new plans for that query. Setting to 0 removes the limitation with regards to the number of captured plans.</p> <p>Change by using the</p> <pre>ALTER DATABASE<database> SET QUERY_STORE (MAX_PLANS_PER_QUERY = <n>)</pre> <p>statement.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
query_capture_mode	smallint	<p>The currently active query capture mode:</p> <p>1 = ALL - all queries are captured. This is the default configuration value for SQL Server (SQL Server 2016 through SQL Server 2017).</p> <p>2 = AUTO - capture relevant queries based on execution count and resource consumption. This is the default configuration value for SQL Database.</p> <p>3 = NONE - stop capturing new queries. Query Store will continue to collect compile and runtime statistics for queries that were captured already. Use this configuration cautiously since you may miss to capture important queries.</p>
query_capture_mode_desc	nvarchar(60)	<p>Textual description of the actual capture mode of Query Store:</p> <p>ALL (default for SQL Server 2016)</p> <p>AUTO (default for SQL Database)</p> <p>NONE</p>
size_based_cleanup_mode	smallint	<p>Controls whether cleanup will be automatically activated when total amount of data gets close to maximum size:</p> <p>1 = OFF – size based cleanup won't be automatically activated.</p> <p>2 = AUTO - size based cleanup will be automatically activated when size on disk reaches 90% of max_storage_size_mb. This is the default configuration value.</p> <p>Size based cleanup removes the least expensive and oldest queries first. It stops at approximately 80% of max_storage_size_mb.</p>
size_based_cleanup_mode_desc	smallint	<p>Textual description of the actual size-based cleanup mode of Query Store:</p> <p>OFF</p> <p>AUTO (default)</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
wait_stats_capture_mode	smallint	Controls whether Query Store performs capture of wait statistics: 0 = OFF 1 = ON
wait_stats_mode_capture_desc	nvarchar(60)	Textual description of the actual wait statistics capture mode: OFF ON (default)

Permissions





Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.query_context_settings \(Transact-SQL\)](#)
[sys.query_store_plan \(Transact-SQL\)](#)
[sys.query_store_query \(Transact-SQL\)](#)
[sys.query_store_query_text \(Transact-SQL\)](#)
[sys.query_store_runtime_stats \(Transact-SQL\)](#)
[sys.query_store_wait_stats \(Transact-SQL\)](#)
[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)
[Monitoring Performance By Using the Query Store](#)
[Catalog Views \(Transact-SQL\)](#)
[sys.fn_stmt_sql_handle_from_sql_stmt \(Transact-SQL\)](#)
[Query Store Stored Procedures \(Transact-SQL\)](#)

sys.query_context_settings (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the semantics affecting context settings associated with a query. There are a number of context settings available in SQL Server that influence the query semantics (defining the correct result of the query). The same query text compiled under different settings may produce different results (depending on the underlying data).

COLUMN NAME	DATA TYPE	DESCRIPTION
context_settings_id	bigint	Primary key. This value is exposed in Showplan XML for queries.
set_options	varbinary(8)	Bit mask reflecting state of several SET options. For more information, see sys.dm_exec_plan_attributes (Transact-SQL).
language_id	smallint	The id of the language. For more information, see sys.syslanguages (Transact-SQL).
date_format	smallint	The date format. For more information, see SET DATEFORMAT (Transact-SQL).
date_first	tinyint	The date first value. For more information, see SET DATEFIRST (Transact-SQL).

COLUMN NAME	DATA TYPE	DESCRIPTION
status	varbinary(2)	<p>Bitmask field that indicates type of query or context in which query was executed.</p> <p>Column value can be combination of multiple flags (expressed in hexadecimal):</p> <p>0x0 – regular query (no specific flags)</p> <p>0x1 - query that was executed through one of the cursor APIs stored procedures</p> <p>0x2 – query for notification</p> <p>0x4 – internal query</p> <p>0x8 – auto parameterized query without universal parameterization</p> <p>0x10 – cursor fetch refresh query</p> <p>0x20 - query that is being used in cursor update requests</p> <p>0x40 - initial result set is returned when a cursor is opened (Cursor Auto Fetch)</p> <p>0x80 – encrypted query</p> <p>0x100 – query in context of row-level security predicate</p>
required_cursor_options	int	Cursor options specified by the user such as the cursor type.
acceptable_cursor_options	int	Cursor options that SQL Server may implicitly convert to in order to support the execution of the statement.

COLUMN NAME	DATA TYPE	DESCRIPTION
merge_action_type	smallint	<p>The type of trigger execution plan used as the result of a MERGE statement.</p> <p>0 indicates a non-trigger plan, a trigger plan that does not execute as the result of a MERGE statement, or a trigger plan that executes as the result of a MERGE statement that only specifies a DELETE action.</p> <p>1 indicates an INSERT trigger plan that runs as the result of a MERGE statement.</p> <p>2 indicates an UPDATE trigger plan that runs as the result of a MERGE statement.</p> <p>3 indicates a DELETE trigger plan that runs as the result of a MERGE statement containing a corresponding INSERT or UPDATE action.</p> <p>For nested triggers run by cascading actions, this value is the action of the MERGE statement that caused the cascade.</p>
default_schema_id	int	ID of the default schema, which is used to resolve names that are not fully qualified.
is_replication_specific	bit	Used for replication.
is_contained	varbinary(1)	1 indicates a contained database.

Permissions





Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)
[sys.query_store_plan \(Transact-SQL\)](#)
[sys.query_store_query \(Transact-SQL\)](#)
[sys.query_store_query_text \(Transact-SQL\)](#)
[sys.query_store_runtime_stats \(Transact-SQL\)](#)
[sys.query_store_wait_stats \(Transact-SQL\)](#)
[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)
[Monitoring Performance By Using the Query Store](#)
[Catalog Views \(Transact-SQL\)](#)
[Query Store Stored Procedures \(Transact-SQL\)](#)
[sys.fn_stmt_sql_handle_from_sql_stmt \(Transact-SQL\)](#)

sys.query_store_plan (Transact-SQL)

11/21/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about each execution plan associated with a query.

COLUMN NAME	DATA TYPE	DESCRIPTION
plan_id	bigint	Primary key.
query_id	bigint	Foreign key. Joins to sys.query_store_query (Transact-SQL) .
plan_group_id	bigint	ID of the plan group. Cursor queries typically require multiple (populate and fetch) plans. Populate and fetch plans that are compiled together are in the same group. 0 means plan is not in a group.
engine_version	nvarchar(32)	Version of the engine used to compile the plan in 'major.minor.build.revision' format.
compatibility_level	smallint	Database compatibility level of the database referenced in the query.
query_plan_hash	binary(8)	MD5 hash of the individual plan.
query_plan	nvarchar(max)	Showplan XML for the query plan.
is_online_index_plan	bit	Plan was used during an online index build.
is_trivial_plan	bit	Plan is a trivial plan (output in stage 0 of query optimizer).
is_parallel_plan	bit	Plan is parallel.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_forced_plan	bit	Plan is marked as forced when user executes stored procedure sys.sp_query_store_force_plan . Forcing mechanism <i>does not guarantee</i> that exactly this plan will be used for the query referenced by query_id . Plan forcing causes query to be compiled again and typically produces exactly the same or similar plan to the plan referenced by plan_id . If plan forcing does not succeed, force_failure_count is incremented and last_force_failure_reason is populated with the failure reason.
is_natively_compiled	bit	Plan includes natively compiled memory optimized procedures. (0 = FALSE, 1 = TRUE).
force_failure_count	bigint	Number of times that forcing this plan has failed. It can be incremented only when the query is recompiled (<i>not on every execution</i>). It is reset to 0 every time is_plan_forced is changed from FALSE to TRUE .
last_force_failure_reason	int	Reason why plan forcing failed. 0: no failure, otherwise error number of the error that caused the forcing to fail 8637: ONLINE_INDEX_BUILD 8683: INVALID_STARJOIN 8684: TIME_OUT 8689: NO_DB 8690: HINT_CONFLICT 8691: SETOPT_CONFLICT 8694: DQ_NO_FORCING_SUPPORTED 8698: NO_PLAN 8712: NO_INDEX 8713: VIEW_COMPILE_FAILED <other value>: GENERAL_FAILURE

COLUMN NAME	DATA TYPE	DESCRIPTION
last_force_failure_reason_desc	nvarchar(128)	<p>Textual description of last_force_failure_reason_desc.</p> <p>ONLINE_INDEX_BUILD: query tries to modify data while target table has an index that is being built online</p> <p>INVALID_STARJOIN: plan contains invalid StarJoin specification</p> <p>TIME_OUT: Optimizer exceeded number of allowed operations while searching for plan specified by forced plan</p> <p>NO_DB: A database specified in the plan does not exist</p> <p>HINT_CONFLICT: Query cannot be compiled because plan conflicts with a query hint</p> <p>DQ_NO_FORCING_SUPPORTED: Cannot execute query because plan conflicts with use of distributed query or full-text operations.</p> <p>NO_PLAN: Query processor could not produce query plan because forced plan could not be verified to be valid for the query</p> <p>NO_INDEX: Index specified in plan no longer exists</p> <p>VIEW_COMPILE_FAILED: Could not force query plan because of a problem in an indexed view referenced in the plan</p> <p>GENERAL_FAILURE: general forcing error (not covered with reasons above)</p>
count_compiles	bigint	Plan compilation statistics.
initial_compile_start_time	datetimeoffset	Plan compilation statistics.
last_compile_start_time	datetimeoffset	Plan compilation statistics.
last_execution_time	datetimeoffset	Last execution time refers to the last end time of the query/plan.
avg_compile_duration	float	Plan compilation statistics.
last_compile_duration	bigint	Plan compilation statistics.

Plan forcing limitations

Query Store has a mechanism to enforce Query Optimizer to use certain execution plan. However, there are some limitations that can prevent a plan to be enforced.

First, if the plan contains following constructions:

- Insert bulk statement.
- Insert bulk statement.
- Reference to an external table
- Distributed query or full-text operations
- Use of Global queries
- Cursors
- Invalid star join specification

Second, when objects that plan relies on, are no longer available:

- Database (if Database, where plan originated, does not exist anymore)
- Index (no longer there or disabled)

Finally, problems with the plan itself:

- Not legal for query
- Query Optimizer exceeded number of allowed operations
- Incorrectly formed plan XML

Permissions

Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)

[sys.query_store_query \(Transact-SQL\)](#)

[sys.query_store_query_text \(Transact-SQL\)](#)

[sys.query_store_runtime_stats \(Transact-SQL\)](#)

[sys.query_store_wait_stats \(Transact-SQL\)](#)

[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)





[Monitoring Performance By Using the Query Store](#)

[Catalog Views \(Transact-SQL\)](#)

[Query Store Stored Procedures \(Transact-SQL\)](#)

sys.query_store_query (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the query and its associated overall aggregated runtime execution statistics.

COLUMN NAME	DATA TYPE	DESCRIPTION
query_id	bigint	Primary key.
query_text_id	bigint	Foreign key. Joins to sys.query_store_query_text (Transact-SQL)
context_settings_id	bigint	Foreign key. Joins to sys.query_context_settings (Transact-SQL) .
object_id	bigint	ID of the database object that the query is part of (stored procedure, trigger, CLR UDF/UDAgg, etc.). 0 if the query is not executed as part of a database object (ad-hoc query).
batch_sql_handle	varbinary(64)	ID of the statement batch the query is part of. Populated only if query references temporary tables or table variables.
query_hash	binary(8)	MD5 hash of the individual query, based on the logical query tree. Includes optimizer hints.
is_internal_query	bit	The query was generated internally.
query_parameterization_type	tinyint	Kind of parameterization: 0 – None 1 – User 2 – Simple 3 – Forced
query_parameterization_type_desc	nvarchar(60)	Textual description for the parameterization type.
initial_compile_start_time	datetimeoffset	Compile start time.
last_compile_start_time	datetimeoffset	Compile start time.

COLUMN NAME	DATA TYPE	DESCRIPTION
last_execution_time	datetimeoffset	Last execution time refers to the last end time of the query/plan.
last_compile_batch_sql_handle	varbinary(64)	Handle of the last SQL batch in which query was used last time. It can be provided as input to sys.dm_exec_sql_text (Transact-SQL) to get the full text of the batch.
last_compile_batch_offset_start	bigint	Information that can be provided to sys.dm_exec_sql_text along with last_compile_batch_sql_handle.
last_compile_batch_offset_end	bigint	Information that can be provided to sys.dm_exec_sql_text along with last_compile_batch_sql_handle.
count_compiles	bigint	Compilation statistics.
avg_compile_duration	float	Compilation statistics in microseconds.
last_compile_duration	bigint	Compilation statistics in microseconds.
avg_bind_duration	float	Binding statistics in microseconds.
last_bind_duration	bigint	Binding statistics.
avg_bind_cpu_time	float	Binding statistics.
last_bind_cpu_time	bigint	Binding statistics.
avg_optimize_duration	float	Optimization statistics in microseconds.
last_optimize_duration	bigint	Optimization statistics.
avg_optimize_cpu_time	float	Optimization statistics in microseconds.
last_optimize_cpu_time	bigint	Optimization statistics.
avg_compile_memory_kb	float	Compile memory statistics.
last_compile_memory_kb	bigint	Compile memory statistics.
max_compile_memory_kb	bigint	Compile memory statistics.
is_clouddb_internal_query	bit	Always 0 in SQL Server on-premises.

Permissions

Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)

[sys.query_store_plan \(Transact-SQL\)](#)

[sys.query_store_query_text \(Transact-SQL\)](#)

[sys.query_store_wait_stats \(Transact-SQL\)](#)

[sys.query_store_runtime_stats \(Transact-SQL\)](#)

[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)

[Monitoring Performance By Using the Query Store](#)





[Catalog Views \(Transact-SQL\)](#)

[Query Store Stored Procedures \(Transact-SQL\)](#)

[sys.fn_stmt_sql_handle_from_sql_stmt \(Transact-SQL\)](#)

sys.query_store_query_text (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains the Transact-SQL text and the SQL handle of the query.

COLUMN NAME	DATA TYPE	DESCRIPTION
query_text_id	bigint	Primary key.
query_sql_text	nvarchar(max)	SQL text of the query, as provided by the user. Includes whitespaces, hints and comments.
statement_sql_handle	varbinary(64)	SQL handle of the individual query.
is_part_of_encrypted_module	bit	Query text is a part of an encrypted module.
has_restricted_text	bit	Query text contains a password or other unmentionable words.

Permissions

Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)

[sys.query_store_plan \(Transact-SQL\)](#)

[sys.query_store_query \(Transact-SQL\)](#)

[sys.query_store_runtime_stats \(Transact-SQL\)](#)

[sys.query_store_wait_stats \(Transact-SQL\)](#)

[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)

[Monitoring Performance By Using the Query Store](#)





[Catalog Views \(Transact-SQL\)](#)

[Query Store Stored Procedures \(Transact-SQL\)](#)

[sys.fn_stmt_sql_handle_from_sql_stmt \(Transact-SQL\)](#)

sys.query_store_runtime_stats (Transact-SQL)

11/21/2017 • 6 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the runtime execution statistics information for the query.

COLUMN NAME	DATA TYPE	DESCRIPTION
runtime_stats_id	bigint	Identifier of the row representing runtime execution statistics for the plan_id , execution_type and runtime_stats_interval_id . It is unique only for the past runtime statistics intervals. For currently active interval there may be multiple rows representing runtime statistics for the plan referenced by plan_id , with the execution type represented by execution_type . Typically, one row represents runtime statistics that are flushed to disk, while other(s) represent in-memory state. Hence, to get actual state for every interval you need to aggregate metrics, grouping by plan_id , execution_type and runtime_stats_interval_id .
plan_id	bigint	Foreign key. Joins to sys.query_store_plan (Transact-SQL).
runtime_stats_interval_id	bigint	Foreign key. Joins to sys.query_store_runtime_stats_interval (Transact-SQL).
execution_type	tinyint	Determines type of query execution: 0 – Regular execution (successfully finished) 3 – Client initiated aborted execution 4 - Exception aborted execution
execution_type_desc	nvarchar(128)	Textual description of the execution type field: 0 – Regular 3 – Aborted 4 - Exception
first_execution_time	datetimeoffset	First execution time for the query plan within the aggregation interval.

COLUMN NAME	DATA TYPE	DESCRIPTION
last_execution_time	datetimeoffset	Last execution time for the query plan within the aggregation interval.
count_executions	bigint	Total count of executions for the query plan within the aggregation interval.
avg_duration	float	Average duration for the query plan within the aggregation interval (reported in microseconds) .
last_duration	bigint	Last duration for the query plan within the aggregation interval (reported in microseconds).
min_duration	bigint	Minimum duration for the query plan within the aggregation interval (reported in microseconds).
max_duration	bigint	Maximum duration for the query plan within the aggregation interval (reported in microseconds).
stdev_duration	float	Duration standard deviation for the query plan within the aggregation interval (reported in microseconds).
avg_cpu_time	float	Average CPU time for the query plan within the aggregation interval (reported in microseconds).
last_cpu_time	bigint	Last CPU time for the query plan within the aggregation interval (reported in microseconds).
min_cpu_time	bigint	Minimum CPU time for the query plan within the aggregation interval (reported in microseconds).
max_cpu_time	bigint	Maximum CPU time for the query plan within the aggregation interval (reported in microseconds).
stdev_cpu_time	float	CPU time standard deviation for the query plan within the aggregation interval (reported in microseconds).
avg_logical_io_reads	float	Average number of logical IO reads for the query plan within the aggregation interval. (expressed as a number of 8KB pages read).
last_logical_io_reads	bigint	Last number of logical IO reads for the query plan within the aggregation interval. (expressed as a number of 8KB pages read).

COLUMN NAME	DATA TYPE	DESCRIPTION
min_logical_io_reads	bigint	Minimum number of logical IO reads for the query plan within the aggregation interval. (expressed as a number of 8KB pages read).
max_logical_io_reads	bigint	Maximum number of logical IO reads for the query plan within the aggregation interval.(expressed as a number of 8KB pages read).
stdev_logical_io_reads	float	Number of logical IO reads standard deviation for the query plan within the aggregation interval. (expressed as a number of 8KB pages read).
avg_logical_io_writes	float	Average number of logical IO writes for the query plan within the aggregation interval.
last_logical_io_writes	bigint	Last number of logical IO writes for the query plan within the aggregation interval.
min_logical_io_writes	bigint	Minimum number of logical IO writes for the query plan within the aggregation interval.
max_logical_io_writes	bigint	Maximum number of logical IO writes for the query plan within the aggregation interval.
stdev_logical_io_writes	float	Number of logical IO writes standard deviation for the query plan within the aggregation interval.
avg_physical_io_reads	float	Average number of physical IO reads for the query plan within the aggregation interval (expressed as a number of 8KB pages read).
last_physical_io_reads	bigint	Last number of physical IO reads for the query plan within the aggregation interval (expressed as a number of 8KB pages read).
min_physical_io_reads	bigint	Minimum number of physical IO reads for the query plan within the aggregation interval (expressed as a number of 8KB pages read).
max_physical_io_reads	bigint	Maximum number of physical IO reads for the query plan within the aggregation interval (expressed as a number of 8KB pages read).

COLUMN NAME	DATA TYPE	DESCRIPTION
stdev_physical_io_reads	float	Number of physical IO reads standard deviation for the query plan within the aggregation interval (expressed as a number of 8KB pages read).
avg_clr_time	float	Average CLR time for the query plan within the aggregation interval (reported in microseconds).
last_clr_time	bigint	Last CLR time for the query plan within the aggregation interval (reported in microseconds).
min_clr_time	bigint	Minimum CLR time for the query plan within the aggregation interval (reported in microseconds).
max_clr_time	bigint	Maximum CLR time for the query plan within the aggregation interval (reported in microseconds).
stdev_clr_time	float	CLR time standard deviation for the query plan within the aggregation interval (reported in microseconds).
avg_dop	float	Average DOP (degree of parallelism) for the query plan within the aggregation interval.
last_dop	bigint	Last DOP (degree of parallelism) for the query plan within the aggregation interval.
min_dop	bigint	Minimum DOP (degree of parallelism) for the query plan within the aggregation interval.
max_dop	bigint	Maximum DOP (degree of parallelism) for the query plan within the aggregation interval.
stdev_dop	float	DOP (degree of parallelism) standard deviation for the query plan within the aggregation interval.
avg_query_max_used_memory	float	Average memory grant (reported as the number of 8 KB pages) for the query plan within the aggregation interval. Always 0 for queries using natively compiled memory optimized procedures.

COLUMN NAME	DATA TYPE	DESCRIPTION
last_query_max_used_memory	bigint	Last memory grant (reported as the number of 8 KB pages) for the query plan within the aggregation interval. Always 0 for queries using natively compiled memory optimized procedures.
min_query_max_used_memory	bigint	Minimum memory grant (reported as the number of 8 KB pages) for the query plan within the aggregation interval. Always 0 for queries using natively compiled memory optimized procedures.
max_query_max_used_memory	bigint	Maximum memory grant (reported as the number of 8 KB pages) for the query plan within the aggregation interval. Always 0 for queries using natively compiled memory optimized procedures.
stdev_query_max_used_memory	float	Memory grant standard deviation (reported as the number of 8 KB pages) for the query plan within the aggregation interval. Always 0 for queries using natively compiled memory optimized procedures.
avg_rowcount	float	Average number of returned rows for the query plan within the aggregation interval.
last_rowcount	bigint	Number of returned rows by the last execution of the query plan within the aggregation interval.
min_rowcount	bigint	Minimum number of returned rows for the query plan within the aggregation interval.
max_rowcount	bigint	Maximum number of returned rows for the query plan within the aggregation interval.
stdev_rowcount	float	Number of returned rows standard deviation for the query plan within the aggregation interval.
avg_log_bytes_used	float	Average number of bytes in the database log used by the query plan, within the aggregation interval. Applies only to Azure SQL Database .
last_log_bytes_used	bigint	Number of bytes in the database log used by the last execution of the query plan, within the aggregation interval. Applies only to Azure SQL Database .

COLUMN NAME	DATA TYPE	DESCRIPTION
min_log_bytes_used	bigint	Minimum number of bytes in the database log used by the query plan, within the aggregation interval. Applies only to Azure SQL Database .
max_log_bytes_used	bigint	Maximum number of bytes in the database log used by the query plan, within the aggregation interval. Applies only to Azure SQL Database .
stdev_log_bytes_used	float	Standard deviation of the number of bytes in the database log used by a query plan, within the aggregation interval. Applies only to Azure SQL Database .

Permissions

Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)

[sys.query_store_plan \(Transact-SQL\)](#)

[sys.query_store_query \(Transact-SQL\)](#)

[sys.query_store_query_text \(Transact-SQL\)](#)

[sys.query_store_wait_stats \(Transact-SQL\)](#)

[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)





[Monitoring Performance By Using the Query Store](#)

[Catalog Views \(Transact-SQL\)](#)

[Query Store Stored Procedures \(Transact-SQL\)](#)

sys.query_store_wait_stats (Transact-SQL)

11/16/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2017)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the wait information for the query.

COLUMN NAME	DATA TYPE	DESCRIPTION
wait_stats_id	bigint	Identifier of the row representing wait statistics for the plan_id, runtime_stats_interval_id, execution_type and wait_category. It is unique only for the past runtime statistics intervals. For currently active interval there may be multiple rows representing wait statistics for the plan referenced by plan_id, with the execution type represented by execution_type and the wait category represented by wait_category. Typically, one row represents wait statistics that are flushed to disk, while other(s) represent in-memory state. Hence, to get actual state for every interval you need to aggregate metrics, grouping by plan_id, runtime_stats_interval_id, execution_type and wait_category.
plan_id	bigint	Foreign key. Joins to sys.query_store_plan (Transact-SQL) .
runtime_stats_interval_id	bigint	Foreign key. Joins to sys.query_store_runtime_stats_interval (Transact-SQL) .
wait_category	tinyint	Wait types are categorized using the table below, and then wait time is aggregated across these wait categories. Different wait categories require a different follow up analysis to resolve the issue, but wait types from the same category lead to very similar troubleshooting experiences, and providing the affected query on top of waits would be the missing piece to complete the majority of such investigations successfully.
wait_category_desc	nvarchar(128)	For textual description of the wait category field please review the table below.

COLUMN NAME	DATA TYPE	DESCRIPTION
execution_type	tinyint	Determines type of query execution: 0 – Regular execution (successfully finished) 3 – Client initiated aborted execution 4 - Exception aborted execution
execution_type_desc	nvarchar(128)	Textual description of the execution type field: 0 – Regular 3 – Aborted 4 - Exception
total_query_wait_time_ms	bigint	Total CPU wait time for the query plan within the aggregation interval and wait category (reported in microseconds).
avg_query_wait_time_ms	float	Average wait duration for the query plan per execution within the aggregation interval and wait category (reported in milliseconds).
last_query_wait_time_ms	bigint	Last wait duration for the query plan within the aggregation interval and wait category (reported in milliseconds).
min_query_wait_time_ms	bigint	Minimum CPU wait time for the query plan within the aggregation interval and wait category (reported in milliseconds).
max_query_wait_time_ms	bigint	Maximum CPU wait time for the query plan within the aggregation interval and wait category (reported in milliseconds).
stdev_query_wait_time_ms	float	Query wait duration standard deviation for the query plan within the aggregation interval and wait category (reported in milliseconds).

Wait categories mapping table

"%" is used as a wildcard

INTEGER VALUE	WAIT CATEGORY	WAIT TYPES INCLUDE IN THE CATEGORY
0	Unknown	Unknown
1	CPU	SOS_SCHEDULER_YIELD

INTEGER VALUE	WAIT CATEGORY	WAIT TYPES INCLUDE IN THE CATEGORY
2	Worker Thread	THREADPOOL
3	Lock	LCK_M_%
4	Latch	LATCH_%
5	Buffer Latch	PAGELATCH_%
6	Buffer IO	PAGEIOLATCH_%
7	Compilation*	RESOURCE_SEMAPHORE_QUERY_COMPILE
8	SQL CLR	CLR%, SQLCLR%
9	Mirroring	DBMIRROR%
10	Transaction	XACT%, DTC%, TRAN_MARKLATCH_%, MSQL_XACT_%, TRANSACTION_Mutex
11	Idle	SLEEP_%, LAZYWRITER_SLEEP, SQLTRACE_BUFFER_FLUSH, SQLTRACE_INCREMENTAL_FLUSH_SLEEP, SQLTRACE_WAIT_ENTRIES, FT_IFTS_SCHEDULER_IDLE_WAIT, XE_DISPATCHER_WAIT, REQUEST_FOR_DEADLOCK_SEARCH, LOGMGR_QUEUE, ONDEMAND_TASK_QUEUE, CHECKPOINT_QUEUE, XE_TIMER_EVENT
12	Preemptive	PREEMPTIVE_%
13	Service Broker	BROKER_% (but not BROKER_RECEIVE_WAITFOR)
14	Tran Log IO	LOGMGR, LOGBUFFER, LOGMGR_RESERVE_APPEND, LOGMGR_FLUSH, LOGMGR_PMM_LOG, CHKPT, WRITELOGF
15	Network IO	ASYNC_NETWORK_IO, NET_WAITFOR_PACKET, PROXY_NETWORK_IO, EXTERNAL_SCRIPT_NETWORK_IOF
16	Parallelism	CXPACKET, EXCHANGE

INTEGER VALUE	WAIT CATEGORY	WAIT TYPES INCLUDE IN THE CATEGORY
17	Memory	RESOURCE_SEMAPHORE, CMEMTHREAD, CMEMPARTITIONED, EE_PMOLOCK, MEMORY_ALLOCATION_EXT, RESERVED_MEMORY_ALLOCATION_EX T, MEMORY_GRANT_UPDATE
18	User Wait	WAITFOR, WAIT_FOR_RESULTS, BROKER_RECEIVE_WAITFOR
19	Tracing	TRACEWRITE, SQLTRACE_LOCK, SQLTRACE_FILE_BUFFER, SQLTRACE_FILE_WRITE_IO_COMPLETI ON, SQLTRACE_FILE_READ_IO_COMPLETIO N, SQLTRACE_PENDING_BUFFER_WRITER S, SQLTRACE_SHUTDOWN, QUERY_TRACEOUT, TRACE_EVTNOTIFF
20	Full Text Search	FT_RESTART_CRAWL, FULLTEXT GATHERER, MSSEARCH, FT_METADATA_MUTEX, FT_IFTSHC_MUTEX, FT_IFTSISM_MUTEX, FT_IFTS_RWLOCK, FT_COMPROWSET_RWLOCK, FT_MASTER_MERGE, FT_PROPERTYLIST_CACHE, FT_MASTER_MERGE_COORDINATOR, PWAIT_RESOURCE_SEMAPHORE_FT_P ARALLEL_QUERY_SYNC
21	Other Disk IO	ASYNC_IO_COMPLETION, IO_COMPLETION, BACKUPIO, WRITE_COMPLETION, IO_QUEUE_LIMIT, IO_RETRY
22	Replication	SE_REPL_%, REPL_%, HADR_% (but not HADR_THROTTLE_LOG_RATE_GOVER NOR), PWAIT_HADR_%, REPLICA_WRITES, FCB_REPLICA_WRITE, FCB_REPLICA_READ, PWAIT_HADRSIM
23	Log Rate Governor	LOG_RATE_GOVERNOR, POOL_LOG_RATE_GOVERNOR, HADR_THROTTLE_LOG_RATE_GOVERN OR, INSTANCE_LOG_RATE_GOVERNOR

****Compilation*** wait category is currently not supported.

Permissions

Requires the **VIEW DATABASE STATE** permission.





See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)
[sys.query_store_plan \(Transact-SQL\)](#)
[sys.query_store_query \(Transact-SQL\)](#)
[sys.query_store_query_text \(Transact-SQL\)](#)
[sys.query_store_runtime_stats_interval \(Transact-SQL\)](#)
[Monitoring Performance By Using the Query Store](#)
[Catalog Views \(Transact-SQL\)](#)
[Query Store Stored Procedures \(Transact-SQL\)](#)

sys.query_store_runtime_stats_interval (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the start and end time of each interval over which runtime execution statistics information for a query has been collected.

COLUMN NAME	DATA TYPE	DESCRIPTION
runtime_stats_interval_id	bigint	Primary key.
start_time	datetimeoffset	Start time of the interval.
end_time	datetimeoffset	End time of the interval.
comment	nvarchar(32)	Always NULL.

Permissions

Requires the **VIEW DATABASE STATE** permission.

See Also

[sys.database_query_store_options \(Transact-SQL\)](#)

[sys.query_context_settings \(Transact-SQL\)](#)

[sys.query_store_plan \(Transact-SQL\)](#)

[sys.query_store_query \(Transact-SQL\)](#)

[sys.query_store_query_text \(Transact-SQL\)](#)

[sys.query_store_runtime_stats \(Transact-SQL\)](#)

[sys.query_store_wait_stats \(Transact-SQL\)](#)





[Monitoring Performance By Using the Query Store](#)

[Catalog Views \(Transact-SQL\)](#)

[Query Store Stored Procedures \(Transact-SQL\)](#)

Scalar Types Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.assembly_types	sys.types
------------------------------------	---------------------------

See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.assembly_types (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each user-defined type that is defined by a CLR assembly. The following **sys.assembly_types** appear in the list of inherited columns (see [sys.types \(Transact-SQL\)](#)) after **rule_object_id**.

COLUMN NAME	DATA TYPE	DESCRIPTION
assembly_id	int	ID of the assembly from which this type was created.
assembly_class	sysname	Name of the class within the assembly that defines this type.
is_binary_ordered	bit	Sorting the bytes of this type is equivalent to sorting using comparison operators on the type.
is_fixed_length	bit	Length of the type is always the same as max_length.
prog_id	nvarchar(40)	ProgID of the type as exposed to COM.
assembly_qualified_name	nvarchar(4000)	Assembly qualified type name. The name is in a format suitable to be passed to Type.GetType().

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[Scalar Types Catalog Views \(Transact-SQL\)](#)

sys.types (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each system and user-defined type.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the type. Is unique within the schema.
system_type_id	tinyint	ID of the internal system-type of the type.
user_type_id	int	ID of the type. Is unique within the database. For system data types, user_type_id = system_type_id .
schema_id	int	ID of the schema to which the type belongs.
principal_id	int	ID of the individual owner if different from schema owner. By default, schema-contained objects are owned by the schema owner. However, an alternate owner can be specified by using the ALTER AUTHORIZATION statement to change ownership. NULL if there is no alternate individual owner.
max_length	smallint	Maximum length (in bytes) of the type. -1 = Column data type is varchar(max) , nvarchar(max) , varbinary(max) , or xml . For text columns, the max_length value will be 16.
precision	tinyint	Max precision of the type if it is numeric-based; otherwise, 0.
scale	tinyint	Max scale of the type if it is numeric-based; otherwise, 0.
collation_name	sysname	Name of the collation of the type if it is character-based; other wise, NULL.
is_nullable	bit	Type is nullable.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_user_defined	bit	1 = User-defined type. 0 = SQL Server system data type.
is_assembly_type	bit	1 = Implementation of the type is defined in a CLR assembly. 0 = Type is based on a SQL Server system data type.
default_object_id	int	ID of the stand-alone default that is bound to the type by using sp_bindefault . 0 = No default exists.
rule_object_id	int	ID of the stand-alone rule that is bound to the type by using sp_bindrule . 0 = No rule exists.
is_table_type	bit	Indicates the type is a table.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Scalar Types Catalog Views \(Transact-SQL\)](#)

[ALTER AUTHORIZATION \(Transact-SQL\)](#)

[OBJECTPROPERTY \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.type_assembly_usages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row per type to assembly reference.

COLUMN NAME	DATA TYPE	DESCRIPTION
user_type_id	int	ID of the type To return the name of the type, join to the sys.types catalog view on this column.
assembly_id	int	ID of the assembly

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Scalar Types Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.column_type_usages (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each column that is of user-defined type.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this column belongs.
column_id	int	ID of the column. Is unique within the object.
user_type_id	int	ID of the user-defined type. To return the name of the type, join to the sys.types catalog view on this column.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).

See Also

[Scalar Types Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.parameter_type_usages (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each parameter that is of user-defined type.

NOTE

This view does not return rows for parameters of numbered procedures.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object to which this parameter belongs.
parameter_id	int	ID of the parameter. Is unique within the object.
user_type_id	int	ID of the user-defined type. To return the name of the type, join to the sys.types catalog view on this column.

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).





See Also

[Scalar Types Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

Security Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Security information is exposed in catalog views that are optimized for performance and utility. When possible, use the following catalog views to access catalog metadata.

Database-Level Views

sys.database_permissions	sys.database_scoped_credentials
sys.database_principals	sys.master_key_passwords
sys.database_role_members	sys.user_token

Server-Level Views

sys.credentials	sys.server_principals
sys.login_token	sys.server_role_members
sys.securable_classes	sys.sql_logins
sys.server_permissions	sys.system_components_surface_area_configuration

Encryption Views

sys.asymmetric_keys	sys.cryptographic_providers
sys.certificates	sys.key_encryptions
sys.column_encryption_key_values	sys.openkeys
sys.column_encryption_keys	sys.security_policies (Transact-SQL)
sys.column_master_key_definitions	sys.security_predicates (Transact-SQL)
sys.crypt_properties	sys.symmetric_keys

SQL Server Audit Views

sys.server_audits	sys.server_file_audits
sys.server_audit_specifications	sys.server_audit_specifications_details
sys.database_audit_specifications	sys.audit_database_specification_details

Permissions


The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Security Center for SQL Server Database Engine and Azure SQL Database Security-Related Dynamic Management Views and Functions \(Transact-SQL\)](#)

sys.asymmetric_keys (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each asymmetric key.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the key. Is unique within the database.
principal_id	int	ID of the database principal that owns the key.
asymmetric_key_id	int	ID of the key. Is unique within the database.
pvt_key_encryption_type	char(2)	How the key is encrypted. NA = Not encrypted MK = Key is encrypted by the master key PW = Key is encrypted by a user-defined password SK = Key is encrypted by service master key.
pvt_key_encryption_type_desc	nvarchar(60)	Description of how the private key is encrypted. NO_PRIVATE_KEY ENCRYPTED_BY_MASTER_KEY ENCRYPTED_BY_PASSWORD ENCRYPTED_BY_SERVICE_MASTER_KEY
thumbprint	varbinary(32)	SHA-1 hash of the key. The hash is globally unique.
algorithm	char(2)	Algorithm used with the key. 1R = 512-bit RSA 2R = 1024-bit RSA 3R = 2048-bit RSA

COLUMN NAME	DATA TYPE	DESCRIPTION
algorithm_desc	nvarchar(60)	Description of the algorithm used with the key. RSA_512 RSA_1024 RSA_2048
key_length	int	Bit length of the key.
sid	varbinary(85)	Login SID for this key. For Extensible Key Management keys this value will be NULL.
string_sid	nvarchar(128)	String representation of the login SID of the key. For Extensible Key Management keys this value will be NULL.
public_key	varbinary(max)	Public key.
attested_by	nvarchar(260)	System use only.
provider_type	nvarchar(120)	Type of cryptographic provider: CRYPTOGRAPHIC PROVIDER = Extensible Key Management keys NULL = Non-Extensible Key Management keys
cryptographic_provider_guid	uniqueidentifier	GUID for the cryptographic provider. For non-Extensible Key Management keys this value will be NULL.
cryptographic_provider_algid	sql_variant	Algorithm ID for the cryptographic provider. For non-Extensible Key Management keys this value will be NULL.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Security Catalog Views \(Transact-SQL\)](#)

[Extensible Key Management \(EKM\)](#)





[Catalog Views \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

[CREATE ASYMMETRIC KEY \(Transact-SQL\)](#)

sys.certificates (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each certificate in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the certificate. Is unique within the database.
certificate_id	int	ID of the certificate. Is unique within the database.
principal_id	int	ID of the database principal that owns this certificate.
pvt_key_encryption_type	char(2)	How the private key is encrypted. NA = There is no private key for the certificate MK = Private key is encrypted by the master key PW = Private key is encrypted by a user-defined password SK = Private key is encrypted by the service master key.
pvt_key_encryption_type_desc	nvarchar(60)	Description of how the private key is encrypted. NO_PRIVATE_KEY ENCRYPTED_BY_MASTER_KEY ENCRYPTED_BY_PASSWORD ENCRYPTED_BY_SERVICE_MASTER_KEY
is_active_for_begin_dialog	bit	If 1, this certificate is used to initiate encrypted service dialogs.
issuer_name	nvarchar(442)	Name of certificate issuer.
cert_serial_number	nvarchar(64)	Serial number of certificate.
sid	varbinary(85)	Login SID for this certificate.

COLUMN NAME	DATA TYPE	DESCRIPTION
string_sid	nvarchar(128)	String representation of the login SID for this certificate
subject	nvarchar(4000)	Subject of this certificate.
expiry_date	datetime	When certificate expires.
start_date	datetime	When certificate becomes valid.
thumbprint	varbinary(32)	SHA-1 hash of the certificate. The SHA-1 hash is globally unique.
attested_by	nvarchar(260)	System use only.
pvt_key_last_backup_date	datetime	The date and time the certificate's private key was last exported.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Security Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

[CREATE CERTIFICATE \(Transact-SQL\)](#)

sys.column_encryption_keys (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about column encryption keys (CEKs) created with the [CREATE COLUMN ENCRYPTION KEY](#) statement. Each row represents a CEK.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	The name of the CMK.
column_encryption_key_id	int	ID of the CEK.
create_date	datetime	Date the CEK was created.
modify_date	datetime	Date the CEK was last modified.

Permissions

Requires the **VIEW ANY COLUMN ENCRYPTION KEY** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[ALTER COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[DROP COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[CREATE COLUMN MASTER KEY \(Transact-SQL\)](#)





[Security Catalog Views \(Transact-SQL\)](#)

[Always Encrypted \(Database Engine\)](#)

[sys.column_encryption_key_values \(Transact-SQL\)](#)

sys.column_encryption_key_values (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about encrypted values of column encryption keys (CEKs) created with either the [CREATE COLUMN ENCRYPTION KEY](#) or the [ALTER COLUMN ENCRYPTION KEY \(Transact-SQL\)](#) statement. Each row represents a value of a CEK, encrypted with a column master key (CMK).

COLUMN NAME	DATA TYPE	DESCRIPTION
column_encryption_key_id	int	ID of the CEK in the database.
column_master_key_id	int	ID of the column master key that was used to encrypt the CEK value.
encrypted_value	varbinary(8000)	CEK value encrypted with the CMK specified in column_master_key_id.
encryption_algorithm_name	sysname	Name of an algorithm used to encrypt the CEK value. Name of the encryption algorithm used to encrypt the value. The algorithm for the system providers must be RSA_OAEP .

Permissions

Requires the **VIEW ANY COLUMN ENCRYPTION KEY** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[ALTER COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[DROP COLUMN ENCRYPTION KEY \(Transact-SQL\)](#)

[CREATE COLUMN MASTER KEY \(Transact-SQL\)](#)

[Security Catalog Views \(Transact-SQL\)](#)

[sys.column_encryption_keys \(Transact-SQL\)](#)





[sys.column_master_keys \(Transact-SQL\)](#)

[sys.columns \(Transact-SQL\)](#)

[Always Encrypted \(Database Engine\)](#)

sys.column_master_keys (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each database master key added by using the [CREATE MASTER KEY](#) statement. Each row represents a single column master key (CMK).

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	The name of the CMK.
column_master_key_id	int	ID of the column master key.
create_date	datetime	Date the column master key was created.
modify_date	datetime	Date the column master key was last modified.
key_store_provider_name	sysname	Name of the provider for the column master key store that contains the CMK. Allowed values are: MSSQL_CERTIFICATE_STORE – If the column master key store is a Certificate Store. A user-defined value, if the column master key store is of a custom type.
key_path	nvarchar(4000)	A column master key store-specific path of the key. The format of the path depends on the column master key store type. Example: <div>'CurrentUser/Personal/'<thumbprint></div> For a custom column master key store, the developer is responsible for defining what a key path is for the custom column master key store.

Permissions

Requires the **VIEW ANY COLUMN MASTER KEY** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE COLUMN MASTER KEY \(Transact-SQL\)](#)





Security Catalog Views (Transact-SQL)

Always Encrypted (Database Engine)

sys.column_encryption_key_values (Transact-SQL)

sys.credentials (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each server-level credential.

COLUMN NAME	DATA TYPE	DESCRIPTION
credential_id	int	ID of the credential. Is unique in the server.
name	sysname	Name of the credential. Is unique in the server.
credential_identity	nvarchar(4000)	Name of the identity to use. This will generally be a Windows user. It does not have to be unique.
create_date	datetime	Time at which the credential was created.
modify_date	datetime	Time at which the credential was last modified.
target_type	nvarchar(100)	Type of credential. Returns NULL for traditional credentials, CRYPTOGRAPHIC PROVIDER for credentials mapped to a cryptographic provider. For more information about external key management providers, see Extensible Key Management (EKM) .
target_id	int	ID of the object that the credential is mapped to. Returns 0 for traditional credentials and non-0 for credentials mapped to a cryptographic provider. For more information about external key management providers, see Extensible Key Management (EKM) .

Remarks

For database-level credentials, see [sys.database_scoped_credentials](#).

Permissions

Requires either `VIEW ANY DEFINITION` permission or `ALTER ANY CREDENTIAL` permission. In addition, the principal must not be denied `VIEW ANY DEFINITION` permission.

See Also

[sys.database_scoped_credentials](#)
[Credentials \(Database Engine\)](#)
[Security Catalog Views \(Transact-SQL\)](#)
[Principals \(Database Engine\)](#)
[CREATE CREDENTIAL \(Transact-SQL\)](#)

sys.crypt_properties (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each cryptographic property associated with a securable.

COLUMN NAME	DATA TYPE	DESCRIPTION
class	tinyint	Identifies class of thing on which property exists. 1 = Object or column
class_desc	nvarchar(60)	Description of the class of thing on which property exists. OBJECT_OR_COLUMN
major_id	int	ID of thing on which property exists, interpreted according to class
thumbprint	varbinary(32)	SHA-1 hash of the certificate or asymmetric key used.
crypt_type	char(4)	Encryption type. SPVC = Encrypted by certificate private key SPVA = Encrypted by asymmetric private key CPVC = Counter signature by certificate private key CPVA = Counter signature by asymmetric key
crypt_type_desc	nvarchar(60)	Description of encryption type. SIGNATURE BY CERTIFICATE SIGNATURE BY ASYMMETRIC KEY COUNTER SIGNATURE BY CERTIFICATE COUNTER SIGNATURE BY ASYMMETRIC KEY
crypt_property	varbinary(max)	Signed or encrypted bits.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Security Catalog Views \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

[Securables](#)

[CREATE CERTIFICATE \(Transact-SQL\)](#)


[CREATE SYMMETRIC KEY \(Transact-SQL\)](#)

[CREATE ASYMMETRIC KEY \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.cryptographic_providers (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each registered cryptographic provider.

COLUMN NAME	DATA TYPE	DESCRIPTION
provider_id	int	Identification number of the cryptographic provider.
name	sysname	Name of the cryptographic provider.
guid	uniqueidentifier	Unique provider GUID.
version	nvarchar(50)	Version of the provider in the format 'aa.bb.cccc.dd'.
dll_path	nvarchar(512)	Path to DLL that implements the Extensible Key Management (EKM) Application Program Interface (API).
is_enabled	bit	Whether the provider is enabled on the server or not. 0 = not enabled (default) 1 = enabled

Remarks

The **sys.cryptographic_providers** view is visible to the public.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Security Catalog Views \(Transact-SQL\)](#)





[Encryption Hierarchy](#)

[Extensible Key Management \(EKM\)](#)

[CREATE CRYPTOGRAPHIC PROVIDER \(Transact-SQL\)](#)

sys.database_audit_specifications (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the database audit specifications in a SQL Server audit on a server instance. For more information, see [SQL Server Audit \(Database Engine\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
Name	sysname	Name of the auditing specification.
database_specification_id	int	ID of the database specification.
create_date	datetime	Date the audit specification was created.
modified_date	datetime	Date the audit specification was last modified.
is_state_enabled	bit	Audit specification state: 0 – DISABLED 1 –ENABLED
audit_GUID	uniqueidentifier	GUID for the audit that contains this specification. Used during enumeration of member database audit specifications during database attach/startup.

Remarks

If a database is in ready-only mode, the SQL Server Audit feature cannot add Database Audit Specifications.

Permissions

Principals with the **ALTER ANY DATABASE AUDIT** or **VIEW DEFINITION** permissions, the dbo role, and members of the db_owners fixed database role have access to this catalog view. In addition, the principal must not be denied **VIEW DEFINITION** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE SERVER AUDIT \(Transact-SQL\)](#)

[ALTER SERVER AUDIT \(Transact-SQL\)](#)





[DROP SERVER AUDIT \(Transact-SQL\)](#)

[CREATE SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)
[DROP SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)
[CREATE DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)
[ALTER DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)
[DROP DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)
[ALTER AUTHORIZATION \(Transact-SQL\)](#)
[sys.fn_get_audit_file \(Transact-SQL\)](#)
[sys.server_audits \(Transact-SQL\)](#)
[sys.server_file_audits \(Transact-SQL\)](#)
[sys.server_audit_specifications \(Transact-SQL\)](#)
[sys.server_audit_specification_details \(Transact-SQL\)](#)
[sys.database_audit_specification_details \(Transact-SQL\)](#)
[sys.dm_server_audit_status \(Transact-SQL\)](#)
[sys.dm_audit_actions \(Transact-SQL\)](#)
[sys.dm_audit_class_type_map \(Transact-SQL\)](#)
[Create a Server Audit and Server Audit Specification](#)

sys.database_audit_specification_details (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the database audit specifications in a SQL Server audit on a server instance for all databases. For more information, see [SQL Server Audit \(Database Engine\)](#). For a list of all audit_action_id's and their names, query [sys.dm_audit_actions \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
database_specification_id	int	ID of the audit specification.
audit_action_id	int	ID of the audit action.
audit_action_name	Sysname	Name of audit action or audit action group
Class	int	Identifies class of object which is being audited.
class_desc	Nvarchar(60)	Description of class of object which is being audited: - SCHEMA - TABLE
major_id	int	Major ID of object being audited, such as a Table ID of a Table Audit action.
minor_id	Int	Secondary ID of object that is being audited, interpreted according to class, such as the column ID of a Table Audit action.
audited_principal_id	int	Principal that is being audited.
audited_result	Nvarchar(60)	Audit action results: - SUCCESS AND FAILURE - SUCCESS - FAILURE
is_group	Bit	Shows whether the object is a group: 0 - Not a group 1 - Group

Permissions

Principals with the **ALTER ANY DATABASE AUDIT** or **VIEW DEFINITION** permissions, the **dbo** role, and members of the **db_owners** fixed database role have access to this catalog view. In addition, the principal must not be denied **VIEW DEFINITION** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE SERVER AUDIT \(Transact-SQL\)](#)

[ALTER SERVER AUDIT \(Transact-SQL\)](#)

[DROP SERVER AUDIT \(Transact-SQL\)](#)

[CREATE SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[DROP SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[CREATE DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[DROP DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER AUTHORIZATION \(Transact-SQL\)](#)

[sys.fn_get_audit_file \(Transact-SQL\)](#)

[sys.server_audits \(Transact-SQL\)](#)

[sys.server_file_audits \(Transact-SQL\)](#)

[sys.server_audit_specifications \(Transact-SQL\)](#)

[sys.server_audit_specification_details \(Transact-SQL\)](#)

[sys.database_audit_specifications \(Transact-SQL\)](#)

[sys.dm_server_audit_status \(Transact-SQL\)](#)





[sys.dm_audit_actions \(Transact-SQL\)](#)

[sys.dm_audit_class_type_map \(Transact-SQL\)](#)

[Create a Server Audit and Server Audit Specification](#)

sys.database_credentials (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each database scoped credential in the database.

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use [sys.database_scoped_credentials](#) instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
credential_id	int	ID of the database scoped credential. Is unique in the database.
name	sysname	Name of the database scoped credential. Is unique in the database.
credential_identity	nvarchar(4000)	Name of the identity to use. This will generally be a Windows user. It does not have to be unique.
create_date	datetime	Time at which the database scoped credential was created.
modify_date	datetime	Time at which the database scoped credential was last modified.
target_type	nvarchar(100)	Type of database scoped credential. Returns NULL for database scoped credentials.
target_id	int	ID of the object that the database scoped credential is mapped to. Returns 0 for database scoped credentials

Permissions

Requires `CONTROL` permission on the database.

See Also

[Credentials \(Database Engine\)](#)

[CREATE DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)

[ALTER DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)





[DROP DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)

[CREATE CREDENTIAL \(Transact-SQL\)](#)

[sys.credentials \(Transact-SQL\)](#)

sys.database_scoped_credentials (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each database scoped credential in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
credential_id	int	ID of the database scoped credential. Is unique in the database.
name	sysname	Name of the database scoped credential. Is unique in the database.
credential_identity	nvarchar(4000)	Name of the identity to use. This will generally be a Windows user. It does not have to be unique.
create_date	datetime	Time at which the database scoped credential was created.
modify_date	datetime	Time at which the database scoped credential was last modified.
target_type	nvarchar(100)	Type of database scoped credential. Returns <code>NULL</code> for database scoped credentials.
target_id	int	ID of the object that the database scoped credential is mapped to. Returns 0 for database scoped credentials

Permissions

Requires `CONTROL` permission on the database.

See Also

[Credentials \(Database Engine\)](#)

[CREATE DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)

[ALTER DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)





[DROP DATABASE SCOPED CREDENTIAL \(Transact-SQL\)](#)

[CREATE CREDENTIAL \(Transact-SQL\)](#)

[sys.credentials \(Transact-SQL\)](#)

sys.database_permissions (Transact-SQL)

11/21/2017 • 5 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for every permission or column-exception permission in the database. For columns, there is a row for every permission that is different from the corresponding object-level permission. If the column permission is the same as the corresponding object permission, there is no row for it and the permission applied is that of the object.

IMPORTANT

Column-level permissions override object-level permissions on the same entity.

COLUMN NAME	DATA TYPE	DESCRIPTION
class	tinyint	<p>Identifies class on which permission exists.</p> <p>0 = Database 1 = Object or Column 3 = Schema 4 = Database Principal 5 = Assembly - Applies to: SQL Server 2008 through SQL Server 2017. 6 = Type 10 = XML Schema Collection - Applies to: SQL Server 2008 through SQL Server 2017. 15 = Message Type - Applies to: SQL Server 2008 through SQL Server 2017. 16 = Service Contract - Applies to: SQL Server 2008 through SQL Server 2017. 17 = Service - Applies to: SQL Server 2008 through SQL Server 2017. 18 = Remote Service Binding - Applies to: SQL Server 2008 through SQL Server 2017. 19 = Route - Applies to: SQL Server 2008 through SQL Server 2017. 23 = Full-Text Catalog - Applies to: SQL Server 2008 through SQL Server 2017. 24 = Symmetric Key - Applies to: SQL Server 2008 through SQL Server 2017. 25 = Certificate - Applies to: SQL Server 2008 through SQL Server 2017. 26 = Asymmetric Key - Applies to: SQL Server 2008 through SQL Server 2017.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
class_desc	nvarchar(60)	<p>Description of class on which permission exists.</p> <p>DATABASE</p> <p>OBJECT_OR_COLUMN</p> <p>SCHEMA</p> <p>DATABASE_PRINCIPAL</p> <p>ASSEMBLY</p> <p>TYPE</p> <p>XML_SCHEMA_COLLECTION</p> <p>MESSAGE_TYPE</p> <p>SERVICE_CONTRACT</p> <p>SERVICE</p> <p>REMOTE_SERVICE_BINDING</p> <p>ROUTE</p> <p>FULLTEXT_CATALOG</p> <p>SYMMETRIC_KEYS</p> <p>CERTIFICATE</p> <p>ASYMMETRIC_KEY</p>
major_id	int	<p>ID of thing on which permission exists, interpreted according to class. Usually, the major_id is simply the kind of ID that applies to what the class represents.</p> <p>0 = The database itself</p> <p>>0 = Object-IDs for user objects</p> <p><0 = Object-IDs for system objects</p>
minor_id	int	<p>Secondary-ID of thing on which permission exists, interpreted according to class. Often, the major_id is zero, because there is no subcategory available for the class of object. Otherwise, it is the Column-ID of a table.</p>
grantee_principal_id	int	<p>Database principal ID to which the permissions are granted.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
grantor_principal_id	int	Database principal ID of the grantor of these permissions.
type	char(4)	Database permission type. For a list of permission types, see the next table.
permission_name	nvarchar(128)	Permission name.
state	char(1)	Permission state: D = Deny R = Revoke G = Grant W = Grant With Grant Option
state_desc	nvarchar(60)	Description of permission state: DENY REVOKE GRANT GRANT_WITH_GRANT_OPTION

Database Permissions

The following types of permissions are possible.

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
AADS	ALTER ANY DATABASE EVENT SESSION	DATABASE
AAMK	ALTER ANY MASK	DATABASE
AEDS	ALTER ANY EXTERNAL DATA SOURCE	DATABASE
AEFF	ALTER ANY EXTERNAL FILE FORMAT	DATABASE
AL	ALTER	APPLICATION ROLE, ASSEMBLY, ASYMMETRIC KEY, CERTIFICATE, CONTRACT, DATABASE, FULLTEXT CATALOG, MESSAGE TYPE, OBJECT, REMOTE SERVICE BINDING, ROLE, ROUTE, SCHEMA, SERVICE, SYMMETRIC KEY, USER, XML SCHEMA COLLECTION
ALAK	ALTER ANY ASYMMETRIC KEY	DATABASE
ALAR	ALTER ANY APPLICATION ROLE	DATABASE

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
ALAS	ALTER ANY ASSEMBLY	DATABASE
ALCF	ALTER ANY CERTIFICATE	DATABASE
ALDS	ALTER ANY DATASPACE	DATABASE
ALED	ALTER ANY DATABASE EVENT NOTIFICATION	DATABASE
ALFT	ALTER ANY FULLTEXT CATALOG	DATABASE
ALMT	ALTER ANY MESSAGE TYPE	DATABASE
ALRL	ALTER ANY ROLE	DATABASE
ALRT	ALTER ANY ROUTE	DATABASE
ALSB	ALTER ANY REMOTE SERVICE BINDING	DATABASE
ALSC	ALTER ANY CONTRACT	DATABASE
ALSK	ALTER ANY SYMMETRIC KEY	DATABASE
ALSM	ALTER ANY SCHEMA	DATABASE
ALSV	ALTER ANY SERVICE	DATABASE
ALTG	ALTER ANY DATABASE DDL TRIGGER	DATABASE
ALUS	ALTER ANY USER	DATABASE
AUTH	AUTHENTICATE	DATABASE
BADB	BACKUP DATABASE	DATABASE
BALO	BACKUP LOG	DATABASE
CL	CONTROL	APPLICATION ROLE, ASSEMBLY, ASYMMETRIC KEY, CERTIFICATE, CONTRACT, DATABASE, FULLTEXT CATALOG, MESSAGE TYPE, OBJECT, REMOTE SERVICE BINDING, ROLE, ROUTE, SCHEMA, SERVICE, SYMMETRIC KEY, TYPE, USER, XML SCHEMA COLLECTION
CO	CONNECT	DATABASE
CORP	CONNECT REPLICATION	DATABASE
CP	CHECKPOINT	DATABASE

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
CRAG	CREATE AGGREGATE	DATABASE
CRAK	CREATE ASYMMETRIC KEY	DATABASE
CRAS	CREATE ASSEMBLY	DATABASE
CRCF	CREATE CERTIFICATE	DATABASE
CRDB	CREATE DATABASE	DATABASE
CRDF	CREATE DEFAULT	DATABASE
CRED	CREATE DATABASE DDL EVENT NOTIFICATION	DATABASE
CRFN	CREATE FUNCTION	DATABASE
CRFT	CREATE FULLTEXT CATALOG	DATABASE
CRMT	CREATE MESSAGE TYPE	DATABASE
CRPR	CREATE PROCEDURE	DATABASE
CRQU	CREATE QUEUE	DATABASE
CRRL	CREATE ROLE	DATABASE
CRRT	CREATE ROUTE	DATABASE
CRRU	CREATE RULE	DATABASE
CRSB	CREATE REMOTE SERVICE BINDING	DATABASE
CRSC	CREATE CONTRACT	DATABASE
CRSK	CREATE SYMMETRIC KEY	DATABASE
CRSM	CREATE SCHEMA	DATABASE
CRSN	CREATE SYNONYM	DATABASE
CRSO	Applies to: SQL Server 2012 through SQL Server 2017. CREATE SEQUENCE	DATABASE
CRSV	CREATE SERVICE	DATABASE
CRTB	CREATE TABLE	DATABASE
CRTY	CREATE TYPE	DATABASE

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
CRVW	CREATE VIEW	DATABASE
CRXS	Applies to: SQL Server 2008 through SQL Server 2017. CREATE XML SCHEMA COLLECTION	DATABASE
DABO	ADMINISTER DATABASE BULK OPERATIONS	DATABASE
DL	DELETE	DATABASE, OBJECT, SCHEMA
EAES	EXECUTE ANY EXTERNAL SCRIPT	DATABASE
EX	EXECUTE	ASSEMBLY, DATABASE, OBJECT, SCHEMA, TYPE, XML SCHEMA COLLECTION
IM	IMPERSONATE	USER
IN	INSERT	DATABASE, OBJECT, SCHEMA
RC	RECEIVE	OBJECT
RF	REFERENCES	ASSEMBLY, ASYMMETRIC KEY, CERTIFICATE, CONTRACT, DATABASE, FULLTEXT CATALOG, MESSAGE TYPE, OBJECT, SCHEMA, SYMMETRIC KEY, TYPE, XML SCHEMA COLLECTION
SL	SELECT	DATABASE, OBJECT, SCHEMA
SN	SEND	SERVICE
SPLN	SHOWPLAN	DATABASE
SUQN	SUBSCRIBE QUERY NOTIFICATIONS	DATABASE
TO	TAKE OWNERSHIP	ASSEMBLY, ASYMMETRIC KEY, CERTIFICATE, CONTRACT, DATABASE, FULLTEXT CATALOG, MESSAGE TYPE, OBJECT, REMOTE SERVICE BINDING, ROLE, ROUTE, SCHEMA, SERVICE, SYMMETRIC KEY, TYPE, XML SCHEMA COLLECTION
UP	UPDATE	DATABASE, OBJECT, SCHEMA

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
VW	VIEW DEFINITION	APPLICATION ROLE, ASSEMBLY, ASYMMETRIC KEY, CERTIFICATE, CONTRACT, DATABASE, FULLTEXT CATALOG, MESSAGE TYPE, OBJECT, REMOTE SERVICE BINDING, ROLE, ROUTE, SCHEMA, SERVICE, SYMMETRIC KEY, TYPE, USER, XML SCHEMA COLLECTION
VWCK	VIEW ANY COLUMN ENCRYPTION KEY DEFINITION	DATABASE
VWCM	VIEW ANY COLUMN MASTER KEY DEFINITION	DATABASE
VWCT	VIEW CHANGE TRACKING	TABLE, SCHEMA
VWDS	VIEW DATABASE STATE	DATABASE

Permissions

Any user can see their own permissions. To see permissions for other users, requires VIEW DEFINITION, ALTER ANY USER, or any permission on a user. To see user-defined roles, requires ALTER ANY ROLE, or membership in the role (such as public).

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

A: Listing all the permissions of database principals

The following query lists the permissions explicitly granted or denied to database principals.

IMPORTANT

The permissions of fixed database roles do not appear in sys.database_permissions. Therefore, database principals may have additional permissions not listed here.

```
SELECT pr.principal_id, pr.name, pr.type_desc,
       pr.authentication_type_desc, pe.state_desc, pe.permission_name
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id;
```

B: Listing permissions on schema objects within a database

The following query joins sys.database_principals and sys.database_permissions to sys.objects and sys.schemas to list permissions granted or denied to specific schema objects.

```
SELECT pr.principal_id, pr.name, pr.type_desc,  
       pr.authentication_type_desc, pe.state_desc,  
       pe.permission_name, s.name + '.' + o.name AS ObjectName  
FROM sys.database_principals AS pr  
JOIN sys.database_permissions AS pe  
      ON pe.grantee_principal_id = pr.principal_id  
JOIN sys.objects AS o  
      ON pe.major_id = o.object_id  
JOIN sys.schemas AS s  
      ON o.schema_id = s.schema_id;
```

See Also

[Securables](#)





[Permissions Hierarchy \(Database Engine\)](#)

[Security Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.database_principals (Transact-SQL)

11/21/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each security principal in a SQL Server database.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of principal, unique within the database.
principal_id	int	ID of principal, unique within the database.
type	char(1)	Principal type: A = Application role C = User mapped to a certificate E = External user from Azure Active Directory G = Windows group K = User mapped to an asymmetric key R = Database role S = SQL user U = Windows user X = External group from Azure Active Directory group or applications
type_desc	nvarchar(60)	Description of principal type. APPLICATION_ROLE CERTIFICATE_MAPPED_USER EXTERNAL_USER WINDOWS_GROUP ASYMMETRIC_KEY_MAPPED_USER DATABASE_ROLE SQL_USER WINDOWS_USER EXTERNAL_GROUPS

COLUMN NAME	DATA TYPE	DESCRIPTION
default_schema_name	sysname	Name to be used when SQL name does not specify a schema. Null for principals not of type S, U, or A.
create_date	datetime	Time at which the principal was created.
modify_date	datetime	Time at which the principal was last modified.
owning_principal_id	int	ID of the principal that owns this principal. All principals except Database Roles must be owned by dbo .
sid	varbinary(85)	SID (Security Identifier) of the principal. NULL for SYS and INFORMATION SCHEMAS.
is_fixed_role	bit	If 1, this row represents an entry for one of the fixed database roles: db_owner, db_accessadmin, db_datareader, db_datawriter, db_ddladmin, db_securityadmin, db_backupoperator, db_denypdatareader, db_denydatawriter.
authentication_type	int	<p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>Signifies authentication type. The following are the possible values and their descriptions.</p> <p>0 : No authentication 1 : Instance authentication 2 : Database authentication 3 : Windows Authentication</p>
authentication_type_desc	nvarchar(60)	<p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>Description of the authentication type. The following are the possible values and their descriptions.</p> <p>NONE : No authentication INSTANCE : Instance authentication DATABASE : Database authentication WINDOWS : Windows Authentication</p>
default_language_name	sysname	<p>Applies to: SQL Server 2012 through SQL Server 2017.</p> <p>Signifies the default language for this principal.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
default_language_lcid	int	Applies to: SQL Server 2012 through SQL Server 2017. Signifies the default LCID for this principal.
allow_encrypted_value_modifications	bit	Applies to: SQL Server 2016 through SQL Server 2017, SQL Database. Suppresses cryptographic metadata checks on the server in bulk copy operations. This enables the user to bulk copy data encrypted using Always Encrypted, between tables or databases, without decrypting the data. The default is OFF.

Remarks

The *PasswordLastSetTime* properties are available on all supported configurations of SQL Server, but the other properties are only available when SQL Server is running on Windows Server 2003 or later and both CHECK_POLICY and CHECK_EXPIRATION are enabled. See [Password Policy](#) for more information.

Permissions

Any user can see their own user name, the system users, and the fixed database roles. To see other users, requires ALTER ANY USER, or a permission on the user. To see user-defined roles, requires ALTER ANY ROLE, or membership in the role.

Examples

A: Listing all the permissions of database principals

The following query lists the permissions explicitly granted or denied to database principals.

IMPORTANT

The permissions of fixed database roles do not appear in sys.database_permissions. Therefore, database principals may have additional permissions not listed here.

```
SELECT pr.principal_id, pr.name, pr.type_desc,
       pr.authentication_type_desc, pe.state_desc, pe.permission_name
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id;
```

B: Listing permissions on schema objects within a database

The following query joins sys.database_principals and sys.database_permissions to sys.objects and sys.schemas to list permissions granted or denied to specific schema objects.


```

SELECT pr.principal_id, pr.name, pr.type_desc,
       pr.authentication_type_desc, pe.state_desc,
       pe.permission_name, s.name + '.' + o.name AS ObjectName
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id
JOIN sys.objects AS o
      ON pe.major_id = o.object_id
JOIN sys.schemas AS s
      ON o.schema_id = s.schema_id;

```

Examples: Azure SQL Data Warehouse and Parallel Data Warehouse

C: Listing all the permissions of database principals

The following query lists the permissions explicitly granted or denied to database principals.

IMPORTANT

The permissions of fixed database roles do not appear in `sys.database_permissions`. Therefore, database principals may have additional permissions not listed here.

```

SELECT pr.principal_id, pr.name, pr.type_desc,
       pr.authentication_type_desc, pe.state_desc, pe.permission_name
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id;

```

D: Listing permissions on schema objects within a database

The following query joins `sys.database_principals` and `sys.database_permissions` to `sys.objects` and `sys.schemas` to list permissions granted or denied to specific schema objects.

```

SELECT pr.principal_id, pr.name, pr.type_desc,
       pr.authentication_type_desc, pe.state_desc,
       pe.permission_name, s.name + '.' + o.name AS ObjectName
FROM sys.database_principals AS pr
JOIN sys.database_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id
JOIN sys.objects AS o
      ON pe.major_id = o.object_id
JOIN sys.schemas AS s
      ON o.schema_id = s.schema_id;

```

See Also

[Principals \(Database Engine\)](#)

[sys.server_principals \(Transact-SQL\)](#)





[Security Catalog Views \(Transact-SQL\)](#)

[Contained Database Users - Making Your Database Portable](#)

[Connecting to SQL Database By Using Azure Active Directory Authentication](#)

sys.database_role_members (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each member of each database role. Database users, application roles, and other database roles can be members of a database role. To add members to a role, use the [ALTER ROLE](#) statement with the `ADD MEMBER` option. Join with [sys.database_principals](#) to return the names of the `principal_id` values.

COLUMN NAME	DATA TYPE	DESCRIPTION
<code>role_principal_id</code>	<code>int</code>	Database principal ID of the role.
<code>member_principal_id</code>	<code>int</code>	Database principal ID of the member.

Permissions

Any user can view their own role membership. To view other role memberships requires membership in the `db_securityadmin` fixed database role or `VIEW DEFINITION` on the database.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Example

The following query returns the members of the database roles.

```
SELECT DP1.name AS DatabaseRoleName,
       isnull (DP2.name, 'No members') AS DatabaseUserName
FROM sys.database_role_members AS DRM
RIGHT OUTER JOIN sys.database_principals AS DP1
    ON DRM.role_principal_id = DP1.principal_id
LEFT OUTER JOIN sys.database_principals AS DP2
    ON DRM.member_principal_id = DP2.principal_id
WHERE DP1.type = 'R'
ORDER BY DP1.name;
```

See Also

[Security Catalog Views \(Transact-SQL\)](#)

[Principals \(Database Engine\)](#)

[Catalog Views \(Transact-SQL\)](#)

[ALTER ROLE \(Transact-SQL\)](#)

[sys.server_role_members \(Transact-SQL\)](#)

sys.key_encryptions (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each symmetric key encryption specified by using the ENCRYPTION BY clause of the CREATE SYMMETRIC KEY statement.

COLUMN NAMES	DATA TYPES	DESCRIPTION
key_id	int	ID of the encrypted key.
thumbprint	varbinary(32)	SHA-1 hash of the certificate with which the key is encrypted, or the GUID of the symmetric key with which the key is encrypted.
crypt_type	char(4)	Type of encryption: ESKS = Encrypted by symmetric key ESKP, ESP2, or ESP3 = Encrypted by password EPUC = Encrypted by certificate EPUA = Encrypted by asymmetric key ESKM = Encrypted by master key
crypt_type_desc	nvarchar(60)	Description of encryption type: ENCRYPTION BY SYMMETRIC KEY ENCRYPTION BY PASSWORD (Beginning with SQL Server 2017, includes a version number for use by CSS.) ENCRYPTION BY CERTIFICATE ENCRYPTION BY ASYMMETRIC KEY ENCRYPTION BY MASTER KEY Note: Windows DPAPI is used to protect the service master key.
crypt_property	varbinary(max)	Signed or encrypted bits.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Security Catalog Views \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

[CREATE SYMMETRIC KEY \(Transact-SQL\)](#)

sys.login_token (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for every server principal that is part of the login token.


COLUMN NAME	DATA TYPE	DESCRIPTION
principal_id	int	ID of the principal. This value is unique within server.
sid	varbinary(85)	Security identifier of the principal. If this is a Windows principal, sid = Windows SID. If the login is mapped to a certificate, sid = GUID from the certificate.
name	nvarchar(128)	Name of the principal. This value is unique within server.
type	nvarchar(128)	Description of principal type. All types are mapped to sid . The value can be one of the following: SQL LOGIN WINDOWS LOGIN WINDOWS GROUP SERVER ROLE LOGIN MAPPED TO CERTIFICATE LOGIN MAPPED TO ASYMMETRIC KEY CERTIFICATE ASYMMETRIC KEY
usage	nvarchar(128)	Indicates the principal participates in the evaluation of GRANT or DENY permissions, or serves as an authenticator. This value can be one of the following: GRANT OR DENY DENY ONLY AUTHENTICATOR

See Also

[sys.user_token \(Transact-SQL\)](#)
[sys.server_principals \(Transact-SQL\)](#)
[sys.database_principals \(Transact-SQL\)](#)
[Principals \(Database Engine\)](#)

sys.master_key_passwords (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each database master key password added by using the **sp_control_dbmasterkey_password** stored procedure. The passwords that are used to protect the master keys are stored in the credential store. The credential name follows this format: ##DBMKEY_<database_family_guid>_<random_password_guid>##. The password is stored as the credential secret. For each password added by using **sp_control_dbmasterkey_password**, there is a row in **sys.credentials**.

Each row in this view shows a **credential_id** and the **family_guid** of a database the master key of which is protected by the password associated with that credential. A join with **sys.credentials** on the **credential_id** will return useful fields, such as the **create_date** and credential name.

COLUMN NAME	DATA TYPE	DESCRIPTION
credential_id	int	ID of the credential to which the password belongs. This ID is unique within the server instance.
family_guid	uniqueidentifier	Unique ID of the original database at creation. This GUID remains the same after the database is restored or attached, even if the database name is changed. If automatic decryption by the service master key fails, SQL Server uses the family_guid to identify credentials that may contain the password used to protect the database master key.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[sp_control_dbmasterkey_password \(Transact-SQL\)](#)

[Security Catalog Views \(Transact-SQL\)](#)

[CREATE SYMMETRIC KEY \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

sys.openkeys (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view returns information about encryption keys that are open in the current session.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	ID of the database that contains the key.
database_name	sysname	Name of the database that contains the key.
key_id	int	ID of the key. The ID is unique within the database.
key_name	sysname	Name of the key. Unique within the database.
key_guid	varbinary	GUID of the key. Unique within the database.
opened_date	datetime	Date and time when the key was opened.
status	int	1 if the key is valid in metadata. 0 if the key is not found in metadata.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Encryption Hierarchy](#)

[OPEN SYMMETRIC KEY \(Transact-SQL\)](#)

sys.securable_classes (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a list of securable classes

COLUMN NAME	DATA TYPE	DESCRIPTION
class_desc	sysname	Name of the class.
class	int	Numerical designation of the class.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns the securable classes supported by this instance of SQL Server.





```
SELECT * FROM sys.securable_classes ORDER BY class;
```

See Also

[Securables](#)

sys.security_policies (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each security policy in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the security policy, unique within the database.
object_id	int	ID of the security policy.
principal_id	int	ID of the owner of the security policy, as registered to the database. NULL if the owner is determined via the schema.
schema_id	int	ID of the schema where the object resides.
parent_object_id	int	ID of the object to which the policy belongs. Must be 0.
type	varchar(2)	Must be SP .
type_desc	nvarchar(60)	SECURITY_POLICY .
create_date	datetime	UTC date the security policy was created.
modify_date	datetime	UTC date the security policy was last modified.
is_ms_shipped	bit	Always false.
is_enabled	bit	Security policy specification state: 0 = disabled 1 = enabled
is_not_for_replication	bit	Policy was created with the NOT FOR REPLICATION option.
uses_database_collation	bit	Uses the same collation as the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_schemabinding_enabled	bit	<p>Schemabinding state for the security policy:</p> <p>0 or NULL = enabled</p> <p>1 = disabled</p>

Permissions

Principals with the **ALTER ANY SECURITY POLICY** permission have access to all objects in this catalog view as well as anyone with **VIEW DEFINITION** on the object.

See Also

[Row-Level Security](#)

[sys.security_predicates \(Transact-SQL\)](#)

[CREATE SECURITY POLICY \(Transact-SQL\)](#)





[Security Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

[Principals \(Database Engine\)](#)

sys.security_predicates (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each security predicate in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the security policy that contains this predicate.
security_predicate_id	int	Predicate ID within this security policy.
target_object_id	int	ID of the object on which the security predicate is bound.
predicate_definition	nvarchar(max)	Fully qualified name of the function that will be used as a security predicate, including the arguments. Note that the <code>schema.function</code> name may be normalized (i.e. escaped) as well as any other element in the text for consistency. For example: <pre>[dbo].[fn_securitypredicate] ([wing], [startTime], [endTime])</pre>
predicate_type	int	The type of predicate used by the security policy: 0 = FILTER PREDICATE 1 = BLOCK PREDICATE
predicate_type_desc	nvarchar(60)	The type of predicate used by the security policy: FILTER BLOCK
operation	int	The type of operation specified for the predicate: NULL = all applicable operations 1 = AFTER INSERT 2 = AFTER UPDATE 3 = BEFORE UPDATE 4 = BEFORE DELETE

COLUMN NAME	DATA TYPE	DESCRIPTION
operation_desc	nvarchar(60)	<p>The type of operation specified for the predicate:</p> <p>NULL</p> <p>AFTER INSERT</p> <p>AFTER UPDATE</p> <p>BEFORE UPDATE</p> <p>BEFORE DELETE</p>

Permissions

Principals with the **ALTER ANY SECURITY POLICY** permission have access to all objects in this catalog view as well as anyone with **VIEW DEFINITION** on the object.

See Also

[Row-Level Security](#)

[sys.security_policies](#) (Transact-SQL)

[CREATE SECURITY POLICY](#) (Transact-SQL)





[Security Catalog Views](#) (Transact-SQL)

[Catalog Views](#) (Transact-SQL)

[Principals](#) (Database Engine)

sys.server_audits (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each SQL Server audit in a server instance. For more information, see [SQL Server Audit \(Database Engine\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
audit_id	int	ID of the audit.
name	sysname	Name of the audit.
audit_guid	uniqueidentifier	GUID for the audit that is used to enumerate audits with member Server Database audit specifications during server start-up and database attach operations.
create_date	datetime	UTC date the audit was created.
modify_date	datetime	UTC date the audit was last modified.
principal_id	int	ID of the owner of the audit, as registered to the server.
type	char(2)	Audit type: SL – NT Security event log AL – NT Application event log FL – File on file system
type_desc	nvarchar(60)	SECURITY LOG APPLICATION LOG FILE
on_failure	tinyint	On Failure to write an action entry: 0 – Continue 1 – Shutdown server instance 2 – Fail operation

COLUMN NAME	DATA TYPE	DESCRIPTION
on_failure_desc	nvarchar(60)	On Failure to write an action entry: CONTINUE SHUTDOWN SERVER INSTANCE FAIL_OPERATION
is_state_enabled	tinyint	0 – Disabled 1 - Enabled
queue_delay	int	Maximum time, in milliseconds, to wait before writing to disk. If 0, the audit will guarantee a write before an event can continue.
predicate	nvarchar(3000)	The predicate expression that is applied to the event.

Permissions

Principals with the **ALTER ANY SERVER AUDIT** or **VIEW ANY DEFINITION** permission have access to this catalog view. In addition, the principal must not be denied **VIEW ANY DEFINITION** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE SERVER AUDIT \(Transact-SQL\)](#)

[ALTER SERVER AUDIT \(Transact-SQL\)](#)

[DROP SERVER AUDIT \(Transact-SQL\)](#)

[CREATE SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[DROP SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[CREATE DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[DROP DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER AUTHORIZATION \(Transact-SQL\)](#)

[sys.fn_get_audit_file \(Transact-SQL\)](#)

[sys.server_file_audits \(Transact-SQL\)](#)

[sys.server_audit_specifications \(Transact-SQL\)](#)

[sys.server_audit_specification_details \(Transact-SQL\)](#)

[sys.database_audit_specifications \(Transact-SQL\)](#)

[sys.database_audit_specification_details \(Transact-SQL\)](#)

[sys.dm_server_audit_status \(Transact-SQL\)](#)





[sys.dm_audit_actions \(Transact-SQL\)](#)

[sys.dm_audit_class_type_map \(Transact-SQL\)](#)

[Create a Server Audit and Server Audit Specification](#)

sys.server_audit_specifications (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the server audit specifications in a SQL Server audit on a server instance. For more information on SQL Server Audit, see [SQL Server Audit \(Database Engine\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
name	Sysname	Name of the server specification.
server_specification_id	Int	ID of the server_specification .
create_date	Datetime	Date the audit server specification was created.
modified_date	Datetime	Date the audit server specification was last modified.
is_state_enabled	tinyint	Audit specification state: 0 – DISABLED 1 –ENABLED
audit_GUID	uniqueidentifier	GUID for the audit that contains this specification. Used during enumeration of member server audit specifications during server startup.

Permissions

Principals with the **ALTER ANY SERVER AUDIT** or **VIEW ANY DEFINITION** permission have access to this catalog view. In addition, the principal must not be denied **VIEW ANY DEFINITION** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE SERVER AUDIT \(Transact-SQL\)](#)

[ALTER SERVER AUDIT \(Transact-SQL\)](#)

[DROP SERVER AUDIT \(Transact-SQL\)](#)

[CREATE SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[DROP SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[CREATE DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)





[DROP DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER AUTHORIZATION \(Transact-SQL\)](#)

[sys.fn_get_audit_file \(Transact-SQL\)](#)
[sys.server_audits \(Transact-SQL\)](#)
[sys.server_file_audits \(Transact-SQL\)](#)
[sys.server_audit_specification_details \(Transact-SQL\)](#)
[sys.database_audit_specifications \(Transact-SQL\)](#)
[sys.database_audit_specification_details \(Transact-SQL\)](#)
[sys.dm_server_audit_status \(Transact-SQL\)](#)
[sys.dm_audit_actions \(Transact-SQL\)](#)
[sys.dm_audit_class_type_map \(Transact-SQL\)](#)
[Create a Server Audit and Server Audit Specification](#)

sys.server_audit_specification_details (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about the server audit specification details (actions) in a SQL Server audit on a server instance. For more information, see [SQL Server Audit \(Database Engine\)](#). For a list of all audit_action_id's and their names, query [sys.dm_audit_actions \(Transact-SQL\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
server_specification_id	int	ID of the audit server specification
audit_action_id	int	ID of the audit action
audit_action_name	sysname	Name of group or name of audit action
class	tinyint	Reserved
class_desc	nvarchar(60)	Reserved
major_id	int	Reserved
minor_id	int	Reserved
audited_principal_id	int	Reserved
audited_result	nvarchar(60)	Audited result: - SUCCESS AND FAILURE - SUCCESS - FAILURE
is_group	bit	Whether the audited object is a group: 0 - Not a group 1 - Group

Permissions

Principals with the **ALTER ANY SERVER AUDIT** or **VIEW ANY DEFINITION** permission have access to this catalog view. In addition, the principal must not be denied **VIEW ANY DEFINITION** permission.





The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

Security Catalog Views (Transact-SQL)
CREATE SERVER AUDIT (Transact-SQL)
ALTER SERVER AUDIT (Transact-SQL)
DROP SERVER AUDIT (Transact-SQL)
CREATE SERVER AUDIT SPECIFICATION (Transact-SQL)
ALTER SERVER AUDIT SPECIFICATION (Transact-SQL)
DROP SERVER AUDIT SPECIFICATION (Transact-SQL)
CREATE DATABASE AUDIT SPECIFICATION (Transact-SQL)
ALTER DATABASE AUDIT SPECIFICATION (Transact-SQL)
DROP DATABASE AUDIT SPECIFICATION (Transact-SQL)
ALTER AUTHORIZATION (Transact-SQL)
sys.fn_get_audit_file (Transact-SQL)
sys.server_audits (Transact-SQL)
sys.server_file_audits (Transact-SQL)
sys.server_audit_specifications (Transact-SQL)
sys.database_audit_specifications (Transact-SQL)
sys.database_audit_specification_details (Transact-SQL)
sys.dm_server_audit_status (Transact-SQL)
sys.dm_audit_actions (Transact-SQL)
sys.dm_audit_class_type_map (Transact-SQL)
Create a Server Audit and Server Audit Specification

sys.server_file_audits (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains extended information about the file audit type in a SQL Server audit on a server instance. For more information, see [SQL Server Audit \(Database Engine\)](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
audit_id	int	ID of the audit.
name	sysname	Name of the audit.
audit_guid	uniqueidentifier	GUID of the audit.
create_date	datetime	UTC date when the file audit was created.
modify_date	datetime	UTC date when the file audit was last modified.
principal_id	int	ID of the owner of the audit as registered on the server.
type	char(2)	Audit type: 0 = NT Security event log 1 = NT Application event log 2 = File on file system
type_desc	nvarchar(60)	Audit type description.
on_failure	tinyint	On Failure condition: 0 = Continue 1 = Shut down server instance 2 = Fail operation
on_failure_desc	nvarchar(60)	On Failure to write an action entry: CONTINUE SHUTDOWN SERVER INSTANCE FAIL OPERATION

COLUMN NAME	DATA TYPE	DESCRIPTION
is_state_enabled	tinyint	0 = Disabled 1 = Enabled
queue_delay	int	Suggested maximum time, in milliseconds, to wait before writing to disk. If 0, the audit will guarantee a write before the event can continue.
predicate	nvarchar(8000)	Predicate expression that is applied to the event.
max_file_size	bigint	Maximum size, in megabytes, of the audit: 0 = Unlimited/Not applicable to the type of audit selected.
max_rollover_files	int	Maximum number of files to use with the rollover option.
max_files	int	Maximum number of files to use without the rollover option.
reserved_disk_space	int	Amount of disk space to reserve per file.
log_file_path	nvarchar(260)	Path to where audit is located. File path for file audit, application log path for application log audit.
log_file_name	nvarchar(260)	Base name for the log file supplied in the CREATE AUDIT DDL. An incremental number is added to the base_log_name file as a suffix to create the log file name.

Permissions

Principals with the **ALTER ANY SERVER AUDIT** or **VIEW ANY DEFINITION** permission have access to this catalog view. In addition, the principal must not be denied **VIEW ANY DEFINITION** permission.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[CREATE SERVER AUDIT \(Transact-SQL\)](#)

[ALTER SERVER AUDIT \(Transact-SQL\)](#)

[DROP SERVER AUDIT \(Transact-SQL\)](#)

[CREATE SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)


[DROP SERVER AUDIT SPECIFICATION \(Transact-SQL\)](#)

[CREATE DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)

[ALTER DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)
[DROP DATABASE AUDIT SPECIFICATION \(Transact-SQL\)](#)
[ALTER AUTHORIZATION \(Transact-SQL\)](#)
[sys.fn_get_audit_file \(Transact-SQL\)](#)
[sys.server_audits \(Transact-SQL\)](#)
[sys.server_file_audits \(Transact-SQL\)](#)
[sys.server_audit_specifications \(Transact-SQL\)](#)
[sys.database_audit_specifications \(Transact-SQL\)](#)
[sys.database_audit_specification_details \(Transact-SQL\)](#)
[sys.dm_server_audit_status \(Transact-SQL\)](#)
[sys.dm_audit_actions \(Transact-SQL\)](#)
[sys.dm_audit_class_type_map \(Transact-SQL\)](#)
[Create a Server Audit and Server Audit Specification](#)

sys.server_permissions (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each server-level permission.

COLUMN NAME	DATA TYPE	DESCRIPTION
class	tinyint	Identifies class of thing on which permission exists. 100 = Server 101 = Server-principal 105 = Endpoint
class_desc	nvarchar(60)	Description of class on which permission exists. One of the following values: SERVER SERVER_PRINCIPAL ENDPOINT
major_id	int	ID of the securable on which permission exists, interpreted according to class. For most, this is just the kind of ID that applies to what the class represents. Interpretation for non-standard is as follows: 100 = Always 0
minor_id	int	Secondary ID of thing on which permission exists, interpreted according to class.
grantee_principal_id	int	Server-principal-ID to which the permissions are granted.
grantor_principal_id	int	Server-principal-ID of the grantor of these permissions.
type	char(4)	Server permission type. For a list of permission types, see the next table.
permission_name	nvarchar(128)	Permission name.

COLUMN NAME	DATA TYPE	DESCRIPTION
state	char(1)	Permission state: D = Deny R = Revoke G = Grant W = Grant With Grant option
state_desc	nvarchar(60)	Description of permission state: DENY REVOKE GRANT GRANT_WITH_GRANT_OPTION

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
ADBO	ADMINISTER BULK OPERATIONS	SERVER
AL	ALTER	ENDPOINT, LOGIN
ALCD	ALTER ANY CREDENTIAL	SERVER
ALCO	ALTER ANY CONNECTION	SERVER
ALDB	ALTER ANY DATABASE	SERVER
ALES	ALTER ANY EVENT NOTIFICATION	SERVER
ALHE	ALTER ANY ENDPOINT	SERVER
ALLG	ALTER ANY LOGIN	SERVER
ALLS	ALTER ANY LINKED SERVER	SERVER
ALRS	ALTER RESOURCES	SERVER
ALSS	ALTER SERVER STATE	SERVER
ALST	ALTER SETTINGS	SERVER
ALTR	ALTER TRACE	SERVER
AUTH	AUTHENTICATE SERVER	SERVER
CL	CONTROL	ENDPOINT, LOGIN
CL	CONTROL SERVER	SERVER

PERMISSION TYPE	PERMISSION NAME	APPLIES TO SECURABLE
CO	CONNECT	ENDPOINT
COSQ	CONNECT SQL	SERVER
CRDB	CREATE ANY DATABASE	SERVER
CRDE	CREATE DDL EVENT NOTIFICATION	SERVER
CRHE	CREATE ENDPOINT	SERVER
CRTE	CREATE TRACE EVENT NOTIFICATION	SERVER
IM	IMPERSONATE	LOGIN
SHDN	SHUTDOWN	SERVER
TO	TAKE OWNERSHIP	ENDPOINT
VW	VIEW DEFINITION	ENDPOINT, LOGIN
VWAD	VIEW ANY DEFINITION	SERVER
VWDB	VIEW ANY DATABASE	SERVER
VWSS	VIEW SERVER STATE	SERVER
XA	EXTERNAL ACCESS	SERVER

Permissions

Any user can see their own permissions. To see permissions for other logins, requires VIEW DEFINITION, ALTER ANY LOGIN, or any permission on a login. To see user-defined server roles, requires ALTER ANY SERVER ROLE, or membership in the role.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following query lists the permissions explicitly granted or denied to server principals.

IMPORTANT

The permissions of fixed server roles do not appear in sys.server_permissions. Therefore, server principals may have additional permissions not listed here.

```
SELECT pr.principal_id, pr.name, pr.type_desc,  
       pe.state_desc, pe.permission_name  
FROM sys.server_principals AS pr  
JOIN sys.server_permissions AS pe  
     ON pe.grantee_principal_id = pr.principal_id;
```

See Also

[Security Catalog Views \(Transact-SQL\)](#)

[Securables](#)





[Catalog Views \(Transact-SQL\)](#)

[Permissions \(Database Engine\)](#)

[Permissions Hierarchy \(Database Engine\)](#)

sys.server_principals (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for every server-level principal.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the principal. Is unique within a server.
principal_id	int	ID number of the Principal. Is unique within a server.
sid	varbinary(85)	SID (Security-IDentifier) of the principal. If Windows principal, then it matches Windows SID.
type	char(1)	Principal type: S = SQL login U = Windows login G = Windows group R = Server role C = Login mapped to a certificate K = Login mapped to an asymmetric key
type_desc	nvarchar(60)	Description of the principal type: SQL_LOGIN WINDOWS_LOGIN WINDOWS_GROUP SERVER_ROLE CERTIFICATE_MAPPED_LOGIN ASYMMETRIC_KEY_MAPPED_LOGIN
is_disabled	int	1 = Login is disabled.
create_date	datetime	Time at which the principal was created.
modify_date	datetime	Time at which the principal definition was last modified.

COLUMN NAME	DATA TYPE	DESCRIPTION
default_database_name	sysname	Default database for this principal.
default_language_name	sysname	Default language for this principal.
credential_id	int	ID of a credential associated with this principal. If no credential is associated with this principal, credential_id will be NULL.
owning_principal_id	int	The principal_id of the owner of a server role. NULL if the principal is not a server role.
is_fixed_role	bit	Returns 1 if the principal is one of the fixed server roles. For more information, see Server-Level Roles .

Permissions

Any login can see their own login name, the system logins, and the fixed server roles. To see other logins, requires ALTER ANY LOGIN, or a permission on the login. To see user-defined server roles, requires ALTER ANY SERVER ROLE, or membership in the role.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following query lists the permissions explicitly granted or denied to server principals.

IMPORTANT

The permissions of fixed server roles do not appear in sys.server_permissions. Therefore, server principals may have additional permissions not listed here.

```
SELECT pr.principal_id, pr.name, pr.type_desc,
       pe.state_desc, pe.permission_name
FROM sys.server_principals AS pr
JOIN sys.server_permissions AS pe
      ON pe.grantee_principal_id = pr.principal_id;
```

See Also

[Security Catalog Views \(Transact-SQL\)](#)





[Catalog Views \(Transact-SQL\)](#)

[Principals \(Database Engine\)](#)

[Permissions Hierarchy \(Database Engine\)](#)

sys.server_role_members (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each member of each fixed and user-defined server role.

COLUMN NAME	DATA TYPE	DESCRIPTION
role_principal_id	int	Server-Principal ID of the role.
member_principal_id	int	Server-Principal ID of the member.

To add or remove server role membership, use the [ALTER SERVER ROLE \(Transact-SQL\)](#) statement.

Permissions

Logins can view their own server role membership and can view the principal_id's of the members of the fixed server roles. To view all server role membership requires the **VIEW DEFINITION ON SERVER ROLE** permission or membership in the **securityadmin** fixed server role.

For more information, see [Metadata Visibility Configuration](#).

Examples

The following example returns the names and id's of the roles and their members.

```
SELECT sys.server_role_members.role_principal_id, role.name AS RoleName,
       sys.server_role_members.member_principal_id, member.name AS MemberName
FROM sys.server_role_members
JOIN sys.server_principals AS role
    ON sys.server_role_members.role_principal_id = role.principal_id
JOIN sys.server_principals AS member
    ON sys.server_role_members.member_principal_id = member.principal_id;
```

See Also

[Catalog Views \(Transact-SQL\)](#)





[Security Catalog Views \(Transact-SQL\)](#)

[Server-Level Roles](#)

[Principals \(Database Engine\)](#)

sys.sql_logins (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for every SQL Server authentication login.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	--	Inherits from sys.server_principals .
is_policy_checked	bit	Password policy is checked.
is_expiration_checked	bit	Password expiration is checked.
password_hash	varbinary(256)	Hash of SQL login password. Beginning with SQL Server 2012, stored password information is calculated using SHA-512 of the salted password.

For a list of columns that this view inherits, see [sys.server_principals \(Transact-SQL\)](#).

Remarks

To view both SQL Server authentication logins and Windows authentication logins, see [sys.server_principals \(Transact-SQL\)](#).

When contained database users are enabled, connections can be made without logins. To identify those accounts, see [sys.database_principals \(Transact-SQL\)](#).

Permissions

Any SQL Server authentication login can see their own login name, and the sa login. To see other logins, requires ALTER ANY LOGIN, or a permission on the login.

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)





[Security Catalog Views \(Transact-SQL\)](#)

[Password Policy](#)

[Principals \(Database Engine\)](#)

sys.symmetric_keys (Transact-SQL)

11/21/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for every symmetric key created with the CREATE SYMMETRIC KEY statement.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the key. Unique within the database.
principal_id	int	ID of the database principal who owns the key.
symmetric_key_id	int	ID of the key. Unique within the database.
key_length	int	Length of the key in bits.
key_algorithm	char(2)	Algorithm used with the key: R2 = RC2 R4 = RC4 D = DES D3 = Triple DES DT = TRIPLE_DES_3KEY DX = DESX A1 = AES 128 A2 = AES 192 A3 = AES 256 NA = EKM Key

COLUMN NAME	DATA TYPE	DESCRIPTION
algorithm_desc	nvarchar(60)	Description of the algorithm used with the key: RC2 RC4 DES Triple_DES TRIPLE_DES_3KEY DESX AES_128 AES_192 AES_256 NULL (Extensible Key Management algorithms only)
create_date	datetime	Date the key was created.
modify_date	datetime	Date the key was modified.
key_guid	uniqueidentifier	Globally unique identifier (GUID) associated with the key. It is auto-generated for persisted keys. GUIDs for temporary keys are derived from the user-supplied pass phrase.
key_thumbprint	sql_variant	SHA-1 hash of the key. The hash is globally unique. For non-Extensible Key Management keys this value will be NULL.
provider_type	nvarchar(120)	Type of cryptographic provider: CRYPTOGRAPHIC PROVIDER = Extensible Key Management keys NULL = Non-Extensible Key Management keys
cryptographic_provider_guid	uniqueidentifier	GUID for the cryptographic provider. For non-Extensible Key Management keys this value will be NULL.
cryptographic_provider_algid	sql_variant	Algorithm ID for the cryptographic provider. For non-Extensible Key Management keys this value will be NULL.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Remarks

The RC4 algorithm is deprecated. This feature will be removed in a future version of Microsoft SQL Server. Do not use this feature in new development work, and modify applications that currently use this feature as soon as possible.

NOTE

The RC4 algorithm is only supported for backward compatibility. New material can only be encrypted using RC4 or RC4_128 when the database is in compatibility level 90 or 100. (Not recommended.) Use a newer algorithm such as one of the AES algorithms instead. In SQL Server 2012 material encrypted using RC4 or RC4_128 can be decrypted in any compatibility level.

Clarification regarding DES algorithms:

- DESX was incorrectly named. Symmetric keys created with ALGORITHM = DESX actually use the TRIPLE DES cipher with a 192-bit key. The DESX algorithm is not provided. This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.
- Symmetric keys created with ALGORITHM = TRIPLE_DES_3KEY use TRIPLE DES with a 192-bit key.
- Symmetric keys created with ALGORITHM = TRIPLE_DES use TRIPLE DES with a 128-bit key.

See Also

[Catalog Views \(Transact-SQL\)](#)

[Extensible Key Management \(EKM\)](#)




[Security Catalog Views \(Transact-SQL\)](#)

[Encryption Hierarchy](#)

[CREATE SYMMETRIC KEY \(Transact-SQL\)](#)

sys.system_components_surface_area_configuration (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for each executable system object that can be enabled or disabled by a surface area configuration component. For more information, see [Surface Area Configuration](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
component_name	sysname	Component name. This will have the keyword collation, Latin1_General_CI_AS_KS_WS. Cannot be NULL.
database_name	sysname	Database that contains the object. This will have the keyword collation, Latin1_General_CI_AS_KS_WS. Must be one of the following: master msdb mssqlsystemresource
schema_name	sysname	Schema that contains the object. This will have the keyword collation, Latin1_General_CI_AS_KS_WS. Cannot be NULL.
object_name	sysname	Name of the object. This will have the keyword collation, Latin1_General_CI_AS_KS_WS. Cannot be NULL.
state	tinyint	0 = Disabled 1 = Enabled

COLUMN NAME	DATA TYPE	DESCRIPTION
type	char(2)	<p>Object type. Can be one of the following:</p> <p>P = SQL_STORED_PROCEDURE</p> <p>PC = CLR_STORED_PROCEDURE</p> <p>FN = SQL_SCALAR_FUNCTION</p> <p>FS = CLR_SCALAR_FUNCTION</p> <p>FT = CLR_TABLE_VALUED_FUNCTION</p> <p>IF = SQL_INLINE_TABLE_VALUED_FUNCTION</p> <p>TF = SQL_TABLE_VALUED_FUNCTION</p> <p>X = EXTENDED_STORED_PROCEDURE</p>
type_desc	nvarchar(60)	Friendly name description of the object type.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also





[Catalog Views \(Transact-SQL\)](#)

[Security Catalog Views \(Transact-SQL\)](#)

[sp_configure \(Transact-SQL\)](#)

sys.user_token (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row for every database principal that is part of the user token in SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
principal_id	int	ID of the principal. The value is unique within database.
sid	varbinary(85)	Security identifier of the principal if the principal is defined external to the database. For example, this can be a SQL Server login, Windows login, Windows Group login, or a login mapped to a certificate, otherwise, this value is NULL.
name	nvarchar (128)	Name of the principal. The value is unique within database.
type	nvarchar (128)	Description of principal type. All types are mapped to sid . The value can be one of the following: SQL USER WINDOWS LOGIN WINDOWS GROUP ROLE APPLICATION ROLE DATABASE ROLE USER MAPPED TO CERTIFICATE USER MAPPED TO ASYMMETRIC KEY CERTIFICATE ASYMMETRIC KEY

COLUMN NAME	DATA TYPE	DESCRIPTION
usage	nvarchar (128)	<p>Indicates the principal participates in the evaluation of GRANT or DENY permissions, or serves as an authenticator.</p> <p>This value can be one of the following:</p> <p>GRANT OR DENY</p> <p>DENY ONLY</p> <p>AUTHENTICATOR</p>

See Also

[sys.login_token \(Transact-SQL\)](#)

[sys.server_principals \(Transact-SQL\)](#)

[sys.database_principals \(Transact-SQL\)](#)

[Principals \(Database Engine\)](#)

Service Broker Catalog Views (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views:

sys.conversation_endpoints	sys.service_contract_usages
sys.conversation_groups	sys.service_contracts (Transact-SQL)
sys.conversation_priorities (Transact-SQL)	sys.service_message_types
sys.message_type_xml_schema_collection_usages (Transact-SQL)	sys.service_queue_usages
sys.remote_service_bindings	sys.service_queues (Transact-SQL)
sys.routes	sys.services
sys.service_contract_message_usages	sys.transmission_queue

See Also

[Catalog Views \(Transact-SQL\)](#)


[System Views \(Transact-SQL\)](#)

[sys.service_broker_endpoints \(Transact-SQL\)](#)

[sys.service_queues \(Transact-SQL\)](#)

sys.conversation_endpoints (Transact-SQL)

11/16/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Each side of a Service Broker conversation is represented by a conversation endpoint. This catalog view contains a row per conversation endpoint in the database.

COLUMN NAME	DATA TYPE	DESCRIPTION
conversation_handle	uniqueidentifier	Identifier for this conversation endpoint. Not NULLABLE.
conversation_id	uniqueidentifier	Identifier for the conversation. This identifier is shared by both participants in the conversation. This together with the is_initiator column is unique within the database. Not NULLABLE.
is_initiator	tinyint	Whether this endpoint is the initiator or the target of the conversation. Not NULLABLE. 1 = Initiator 0 = Target
service_contract_id	int	Identifier of the contract for this conversation. Not NULLABLE.
conversation_group_id	uniqueidentifier	Identifier for the conversation group this conversation belongs to. Not NULLABLE.
service_id	int	Identifier for the service for this side of the conversation. Not NULLABLE.
lifetime	datetime	Expiration date/time for this conversation. Not NULLABLE.
state	char(2)	The current state of the conversation. Not NULLABLE. One of: SO Started outbound. SQL Server processed a BEGIN CONVERSATION for this conversation, but no messages have yet been sent. SI Started inbound. Another instance started a new conversation with SQL Server, but SQL Server has not yet completely received the first message. SQL Server may create the conversation in this state if the first message is fragmented or SQL Server receives

COLUMN NAME	DATA TYPE	DESCRIPTION
		<p>messages out of order. However, SQL Server might create the conversation in the CO (conversing) state if the first transmission received for the conversation contains the entire first message.</p> <p>CO Conversing. The conversation is established, and both sides of the conversation may send messages. Most of the communication for a typical service takes place when the conversation is in this state.</p> <p>DI Disconnected inbound. The remote side of the conversation has issued an END CONVERSATION. The conversation remains in this state until the local side of the conversation issues an END CONVERSATION. An application might still receive messages for the conversation. Because the remote side of the conversation has ended the conversation, an application cannot send messages on this conversation. When an application issues an END CONVERSATION, the conversation moves to the CD (Closed) state.</p> <p>DO Disconnected outbound. The local side of the conversation has issued an END CONVERSATION. The conversation remains in this state until the remote side of the conversation acknowledges the END CONVERSATION. An application cannot send or receive messages for the conversation. When the remote side of the conversation acknowledges the END CONVERSATION, the conversation moves to the CD (Closed) state.</p> <p>ER Error. An error has occurred on this endpoint. The error message is placed in the application queue. If the application queue is empty, this indicates that the application already consumed the error message.</p> <p>CD Closed. The conversation endpoint is no longer in use.</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
state_desc	nvarchar(60)	Description of endpoint conversation state. This column is NULLABLE. One of: STARTED_OUTBOUND STARTED_INBOUND CONVERSING DISCONNECTED_INBOUND DISCONNECTED_OUTBOUND CLOSED ERROR
far_service	nvarchar(256)	Name of the service on the remote side of conversation. Not NULLABLE.
far_broker_instance	nvarchar(128)	The broker instance for the remote side of the conversation. NULLABLE.
principal_id	int	Identifier of the principal whose certificate is used by the local side of the dialog. Not NULLABLE.
far_principal_id	int	Identifier of the user whose certificate is used by the remote side of the dialog. Not NULLABLE.
outbound_session_key_identifier	uniqueidentifier	Identifier for outbound encryption key for this dialog. Not NULLABLE.
inbound_session_key_identifier	uniqueidentifier	Identifier for inbound encryption key for this dialog. Not NULLABLE.
security_timestamp	datetime	Time at the local session key was created. Not NULLABLE.
dialog_timer	datetime	The time at which the conversation timer for this dialog sends a DialogTimer message. Not NULLABLE.
send_sequence	bigint	Next message number in the send sequence. Not NULLABLE.
last_send_tran_id	binary(6)	Internal transaction ID of last transaction to send a message. Not NULLABLE.
end_dialog_sequence	bigint	The sequence number of the End Dialog message. Not NULLABLE.

COLUMN NAME	DATA TYPE	DESCRIPTION
receive_sequence	bigint	Next message number expected in message receive sequence. Not NULLABLE.
receive_sequence_frag	int	Next message fragment number expected in message receive sequence. Not NULLABLE.
system_sequence	bigint	The sequence number of the last system message for this dialog. Not NULLABLE.
first_out_of_order_sequence	bigint	The sequence number of the first message in the out of order messages for this dialog. Not NULLABLE.
last_out_of_order_sequence	bigint	The sequence number of the last message in the out of order messages for this dialog. Not NULLABLE.
last_out_of_order_frag	int	Sequence number of the last message in the out of order fragments for this dialog. Not NULLABLE.
is_system	bit	1 if this is a system dialog. Not NULLABLE.
priority	tinyint	The conversation priority that is assigned to this conversation endpoint. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.conversation_groups (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row for each conversation group.


COLUMN NAME	DATA TYPE	DESCRIPTION
conversation_group_id	uniqueidentifier	Identifier for the conversation group. Not NULLABLE.
service_id	int	Identifier of the service for conversations in this group. Not NULLABLE.
is_system	bit	Indicates whether this is a system instance or not. NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.conversation_priorities (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each conversation priority created in the current database, as shown in the following table:

COLUMN NAME	DATA TYPE	DESCRIPTION
Priority_id	int	A number that uniquely identifies the conversation priority. Not NULLABLE.
name	sysname	Name of the conversation priority. Not NULLABLE.
service_contract_id	int	The identifier of the contract that is specified for the conversation priority. This can be joined on the service_contract_id column in sys.service_contracts. NULLABLE.
local_service_id	int	The identifier of the service that is specified as the local service for the conversation priority. This column can be joined on the service_id column in sys.services. NULLABLE.
remote_service_name	nvarchar(256)	The name of the service that is specified as the remote service for the conversation priority. NULLABLE.
priority	tinyint	The priority level that is specified in this conversation priority. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Examples

The following example lists the conversation priorities by using joins to show the contract and local service names.

```
SELECT scp.name AS priority_name,  
       ssc.name AS contract_name,  
       ssvc.name AS local_service_name,  
       scp.remote_service_name,  
       scp.priority AS priority_level  
FROM sys.conversation_priorities AS scp  
     INNER JOIN sys.service_contracts AS ssc  
       ON scp.service_contract_id = ssc.service_contract_id  
     INNER JOIN sys.services AS ssvc  
       ON scp.local_service_id = ssvc.service_id  
ORDER BY priority_name, contract_name,  
         local_service_name, remote_service_name;
```

See Also

[ALTER BROKER PRIORITY \(Transact-SQL\)](#)

[CREATE BROKER PRIORITY \(Transact-SQL\)](#)




[DROP BROKER PRIORITY \(Transact-SQL\)](#)

[sys.services \(Transact-SQL\)](#)

[sys.service_contracts \(Transact-SQL\)](#)

sys.message_type_xml_schema_collection_usages (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view returns a row for each service message type that is validated by an XML schema collection.


COLUMN NAME	DATA TYPE	DESCRIPTION
message_type_id	int	The ID of the service message type. Not NULLABLE.
xml_collection_id	int	The ID of the collection containing the validating XML schema namespace. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.remote_service_bindings (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row per remote service binding.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of this remote service binding. Not NULLABLE.
remote_service_binding_id	int	ID of this remote service binding. Not NULLABLE.
principal_id	int	ID of the database principal that owns this remote service binding. NULLABLE.
remote_service_name	nvarchar(256)	Name of the remote service that this binding applies to. NULLABLE.
service_contract_id	int	ID of the contract that this binding applies to. A value of 0 is a wildcard that means this binding applies to all contracts for the service. Not NULLABLE.
remote_principal_id	int	ID for the user specified in the remote service binding. Service Broker uses a certificate owned by this user for communicating with the specified service on the specified contracts. NULLABLE.
is_anonymous_on	bit	This remote service binding uses ANONYMOUS security. The identity of the user that begins the conversation is not provided to the target service. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.routes (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog views contains one row per route. Service Broker uses routes to locate the network address for a service.

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the route, unique within the database. Not NULLABLE.
route_id	int	Identifier for the route. Not NULLABLE.
principal_id	int	Identifier for the database principal that owns the route. NULLABLE.
remote_service_name	nvarchar(256)	Name of the remote service. NULLABLE.
broker_instance	nvarchar(128)	Identifier of the broker that hosts the remote service. NULLABLE.
lifetime	datetime	The date and time when the route expires. Notice that this value does not use the local time zone. Instead, the value shows the expiration time for UTC. NULLABLE.
address	nvarchar(256)	Network address to which Service Broker sends messages for the remote service. NULLABLE.
mirror_address	nvarchar(256)	Network address of the mirroring partner for the server specified in the address. NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.service_contract_message_usages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row per (contract, message type) pair.

COLUMN NAME	DATA TYPE	DESCRIPTION
service_contract_id	int	Identifier of the contract using the message type. Not NULLABLE.
message_type_id	int	Identifier of the message type used by the contract. Not NULLABLE.
is_sent_by_initiator	bit	Message type can be sent by the conversation initiator. Not NULLABLE.
is_sent_by_target	bit	Message type can be sent by the conversation target. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.service_contract_usages (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row per (service, contract) pair.





COLUMN NAME	DATA TYPE	DESCRIPTION
service_id	int	Identifier of the service using the contract. Not NULLABLE.
service_contract_id	int	Identifier of the contract used by the service. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.service_contracts (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row for each contract in the database.


COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the contract, unique within the database. Not NULLABLE.
service_contract_id	int	Identifier of the contract. Not NULLABLE.
principal_id	int	Identifier for the database principal that owns this contract. NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.service_message_types (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row per message type registered in the service broker.


COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of message type, unique within the database. Not NULLABLE.
message_type_id	int	Identifier of the message type, unique within the database. Not NULLABLE.
principal_id	int	Identifier for the database principal that owns this message type. NULLABLE.
validation	char(2)	Validation done by Broker prior to sending messages of this type. Not NULLABLE. One of: N = None X = XML E = Empty
validation_desc	nvarchar(60)	Description of the validation done by Broker prior to sending messages of this type. NULLABLE. One of: NONE XML EMPTY
xml_collection_id	int	For validation that uses an XML schema, the identifier for the schema collection used. Otherwise, NULL.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.service_queue_usages (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view returns a row for each reference between service and service queue. A service can only be associated with one queue. A queue can be associated with multiple services.

COLUMN NAME	DATA TYPE	DESCRIPTION
service_id	int	Identifier of the service. Unique within the database. Not NULLABLE.
service_queue_id	int	Identifier of the service queue used by the service. Not NULLABLE.

Permissions





The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[sys.services](#) (Transact-SQL)

sys.service_queues (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each object in the database that is a service queue, with **sys.objects.type** = SQ.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		For a list of columns that this view inherits, see sys.objects (Transact-SQL) .
max_readers	smallint	Maximum number of the concurrent readers allowed in the queue.
activation_procedure	nvarchar(776)	Three-part name of the activation procedure.
execute_as_principal_id	int	ID of the EXECUTE AS database principal. NULL by default or if EXECUTE AS CALLER. ID of the specified principal if EXECUTE AS SELF EXECUTE AS <principal>. -2 = EXECUTE AS OWNER.
is_activation_enabled	bit	1 = Activation is enabled.
is_receive_enabled	bit	1 = Receive is enabled.
is_enqueue_enabled	bit	1 = Enqueue is enabled.
is_retention_enabled	bit	1 = Messages are retained until dialog end.
is_poison_message_handling_enabled	bit	Applies to: SQL Server 2012 through SQL Server 2017. 1 = Poison message handling is enabled.

Permissions





The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

sys.services (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row for each service in the database.



COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Case-sensitive name of service, unique within the database. Not NULLABLE.
service_id	int	Identifier of the service. Not NULLABLE.
principal_id	int	Identifier for the database principal that owns this service. NULLABLE.
service_queue_id	int	Object id for the queue that this service uses. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

sys.transmission_queue (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This catalog view contains a row for each message in the transmission queue, as shown in the following table:

COLUMN NAME	DATA TYPE	DESCRIPTION
conversation_handle	uniqueidentifier	Identifier for the conversation that this message belongs to. Not NULLABLE.
to_service_name	nvarchar(256)	Name of the service that this message is to. NULLABLE.
to_broker_instance	nvarchar(128)	Identifier of the broker that hosts the service that this message is to. NULLABLE.
from_service_name	nvarchar(256)	Name of the service that this message is from. NULLABLE.
service_contract_name	nvarchar(256)	Name of the contract that the conversation for this message follows. NULLABLE.
enqueue_time	datetime	Time at which the message entered the queue. This value uses UTC regardless of the local time zone for the instance. Not NULLABLE.
message_sequence_number	bigint	Sequence number of the message. Not NULLABLE.
message_type_name	nvarchar(256)	Message type name for the message. NULLABLE.
is_conversation_error	bit	Whether this message is an error message. 0 = Not an error message. 1 = Error message. Not NULLABLE.

COLUMN NAME	DATA TYPE	DESCRIPTION
is_end_of_dialog	bit	<p>Whether this message is an end of conversation message. Not NULLABLE.</p> <p>0 = Not an end of conversation message.</p> <p>1 = End of conversation message.</p> <p>Not NULLABLE.</p>
message_body	varbinary(max)	The body of this message. NULLABLE.
transmission_status	nvarchar(4000)	<p>The reason this message is on the queue. This is generally an error message explaining why sending the message failed. If this is blank, the message has not been sent yet.</p> <p>NULLABLE.</p>
priority	tinyint	The priority level that is assigned to this message. Not NULLABLE.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

Server-wide Configuration Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

This section contains the following catalog views.

sys.configurations	sys.trace_columns
sys.time_zone_info	sys.trace_events
sys.traces	sys.trace_event_bindings
sys.trace_categories	sys.trace_subclass_values





See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.configurations (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each server-wide configuration option value in the system.

COLUMN NAME	DATA TYPE	DESCRIPTION
configuration_id	int	Unique ID for the configuration value.
name	nvarchar(35)	Name of the configuration option.
value	sql_variant	Configured value for this option.
minimum	sql_variant	Minimum value for the configuration option.
maximum	sql_variant	Maximum value for the configuration option.
value_in_use	sql_variant	Running value currently in effect for this option.
description	nvarchar(255)	Description of the configuration option.
is_dynamic	bit	1 = The variable that takes effect when the RECONFIGURE statement is executed.
is_advanced	bit	1 = The variable is displayed only when the show advancedoption is set.

For a list of all server configuration options, see [Server Configuration Options \(SQL Server\)](#).

NOTE

For database-level configuration options, see [ALTER DATABASE SCOPED CONFIGURATION \(Transact-SQL\)](#). To configure Soft-NUMA, see [Soft-NUMA \(SQL Server\)](#).

Permissions

Requires membership in the **public** role. For more information, see [Metadata Visibility Configuration](#).





See Also

[Server-wide Configuration Catalog Views \(Transact-SQL\)](#)

[Catalog Views \(Transact-SQL\)](#)

sys.time_zone_info (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns information about supported time zones. All time zones installed on the computer are stored in the following registry hive:

```
KEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones .
```

COLUMN NAME	DATA TYPE	DESCRIPTION
name	sysname	Name of the time zone in Windows standard format. For example, Cen. Australia Standard Time or Central European Standard Time .
current_utc_offset	nvarchar(12)	Current offset to UTC. For example, +01:00 or -07:00 .
is_currently_dst	bit	True if currently observing daylight savings time.

See Also

[GETUTCDATE \(Transact-SQL\)](#)





[AT TIME ZONE \(Transact-SQL\)](#)

[Date and Time Data Types and Functions \(Transact-SQL\)](#)

[Server-wide Configuration Catalog Views \(Transact-SQL\)](#)

sys.traces (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The **sys.traces** catalog view contains the current running traces on the system. This view is intended as a replacement for the **fn_trace_getinfo** function.

For a complete list of supported trace events, see [SQL Server Event Class Reference](#).

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
id	int	Trace ID.
status	int	Trace status: 0 = stopped 1 = running
path	nvarchar(260)	Path of the trace file. This value is null when the trace is a rowset trace.
max_size	bigint	Maximum trace file size limit in megabytes (MB). This value is null when the trace is a rowset trace.
stop_time	datetime	Time to stop the running trace.
max_files	int	Maximum number of rollover files. This value is null if the Max number is not set.
is_rowset	bit	1 = rowset trace.
is_rollover	bit	1 = rollover option is enabled.
is_shutdown	bit	1 = shutdown option is enabled.
is_default	bit	1 = default trace.
buffer_count	int	Number of in-memory buffers used by the trace.
buffer_size	int	Size of each buffer (KB).

COLUMN NAME	DATA TYPE	DESCRIPTION
file_position	bigint	Last trace file position. This value is null when the trace is a rowset trace.
reader_spid	int	Rowset trace reader session ID. This value is null when the trace is a file trace.
start_time	datetime	Trace start time.
last_event_time	datetime	Time the last event fired.
event_count	bigint	Total number of events that occurred.
dropped_event_count	int	Total number of events dropped.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.trace_categories \(Transact-SQL\)](#)

[sys.trace_columns \(Transact-SQL\)](#)





[sys.trace_events \(Transact-SQL\)](#)

[sys.trace_event_bindings \(Transact-SQL\)](#)

[sys.trace_subclass_values \(Transact-SQL\)](#)

sys.trace_categories (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Similar event classes are grouped by a category. Each row in the **sys.trace_categories** catalog view identifies a category that is unique across the server. These categories do not change for a given version of the SQL Server Database Engine.

For a complete list of supported trace events, see [SQL Server Event Class Reference](#).

IMPORTANT! This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
category_id	smallint	Unique ID of this category. This column is also in the sys.trace_events catalog view.
name	nvarchar(128)	Unique name of this category. This parameter is not localized.
type	tinyint	Category type: 0 = Normal 1 = Connection 2 = Error

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.traces \(Transact-SQL\)](#)

[sys.trace_columns \(Transact-SQL\)](#)





[sys.trace_events \(Transact-SQL\)](#)

[sys.trace_event_bindings \(Transact-SQL\)](#)

[sys.trace_subclass_values \(Transact-SQL\)](#)

sys.trace_columns (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The **sys.trace_columns** catalog view contains a list of all trace event columns. These columns do not change for a given version of the SQL Server Database Engine.

For a complete list of supported trace events, see [SQL Server Event Class Reference](#).

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
trace_column_id	smallint	Unique ID of this column.
name	nvarchar(128)	Unique name of this column. This parameter is not localized.
type_name	nvarchar(128)	Data type name of this column.
max_size	int	Maximum data size of this column in bytes.
is_filterable	bit	Indicates whether the column can be used in filter specification. 0 = false 1 = true
is_repeatabe	bit	Indicates whether the column can be referenced in the "repeated column" data. 0 = false 1 = true
is_repeated_base	bit	Indicates whether this column is used as a unique key for referencing repeated data. 0 = false 1 = true

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.traces \(Transact-SQL\)](#)

[sys.trace_categories \(Transact-SQL\)](#)





[sys.trace_events \(Transact-SQL\)](#)

[sys.trace_event_bindings \(Transact-SQL\)](#)

[sys.trace_subclass_values \(Transact-SQL\)](#)

sys.trace_events (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The **sys.trace_events** catalog view contains a list of all SQL trace events. These trace events do not change for a given version of the SQL Server Database Engine.

IMPORTANT! This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

For more information about these trace events, see [SQL Server Event Class Reference](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
trace_event_id	smallint	Unique ID of the event. This column is also in the sys.trace_event_bindings and sys.trace_subclass_values catalog views.
category_id	smallint	Category ID of the event. This column is also in the sys.trace_categories catalog view.
name	nvarchar(128)	Unique name of this event. This parameter is not localized.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.traces \(Transact-SQL\)](#)

[sys.trace_categories \(Transact-SQL\)](#)





[sys.trace_columns \(Transact-SQL\)](#)

[sys.trace_event_bindings \(Transact-SQL\)](#)

[sys.trace_subclass_values \(Transact-SQL\)](#)

sys.trace_event_bindings (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The **sys.trace_event_bindings** catalog view contains a list of all possible usage combinations of events and columns. For each event listed in the **trace_event_id** column, all available columns are listed in the **trace_column_id** column. Not all available columns are populated each time a given event occurs. These values do not change for a given version of the SQL Server Database Engine.

For a complete list of supported trace events, see [SQL Server Event Class Reference](#).

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
trace_event_id	smallint	ID of the trace event. This column is also in the sys.trace_events catalog view.
trace_column_id	smallint	ID of the trace column. This column is also in the sys.trace_columns catalog view.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.traces \(Transact-SQL\)](#)

[sys.trace_categories \(Transact-SQL\)](#)





[sys.trace_columns \(Transact-SQL\)](#)

[sys.trace_events \(Transact-SQL\)](#)

[sys.trace_subclass_values \(Transact-SQL\)](#)

sys.trace_subclass_values (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

The **sys.trace_subclass_values** catalog view contains a list of named column values. These subclass values do not change for a given version of the SQL Server Database Engine.

For a complete list of supported trace events, see [SQL Server Event Class Reference](#).

IMPORTANT

This feature will be removed in a future version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature. Use Extended Event catalog views instead.

COLUMN NAME	DATA TYPE	DESCRIPTION
trace_event_id	smallint	ID of the trace event. This parameter is also in the sys.trace_events catalog view.
trace_column_id	smallint	ID of the trace column used for enumeration. This parameter is also in the sys.trace_columns catalog view.
subclass_name	nvarchar(128)	Meaning of the column value.
subclass_value	smallint	Column value.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[sys.traces \(Transact-SQL\)](#)

[sys.trace_categories \(Transact-SQL\)](#)





[sys.trace_columns \(Transact-SQL\)](#)

[sys.trace_events \(Transact-SQL\)](#)

[sys.trace_event_bindings \(Transact-SQL\)](#)

Spatial Data Catalog Views

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section describes the catalog views that provide information about spatial data.

[sys.spatial_index_tessellations](#) (Transact-SQL)

Represents the information about the tessellation scheme and parameters of each of the spatial indexes.

[sys.spatial_indexes](#) (Transact-SQL)

Represents the main index information of the spatial indexes.

[sys.spatial_reference_systems](#) (Transact-SQL)

Lists the supported spatial reference systems (SRIDs).

sys.spatial_index_tessellations (Transact-SQL)

11/21/2017 • 3 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Represents the information about the tessellation scheme and parameters of each of the spatial indexes.

NOTE

For information about tessellation, see [Spatial Indexes Overview](#).

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the object on which the index is defined. Each (object_id, index_id) pair has a corresponding entry in sys.spatial_indexes .
index_id	int	ID of the spatial index in which the indexed column is defined
tessellation_scheme	sysname	Name of the tessellation scheme, one of: GEOMETRY_GRID, GEOGRAPHY_GRID
bounding_box_xmin	float(53)	X-coordinate of the lower-left corner of the bounding box, one of: NULL = Not applicable for a given tessellation scheme (such as GEOGRAPHY_GRID) <i>n</i> = If tessellation_scheme is GEOMETRY_GRID, the x-min coordinate value. Note: The coordinates defined by the bounding box parameters are interpreted for each object according to its Spatial Reference Identifier (SRID) .
bounding_box_ymin	float(53)	Y-coordinate of the lower-left corner of the bounding box, one of: NULL = Not applicable for a given tessellation scheme (such as GEOGRAPHY_GRID) <i>n</i> = If tessellation_scheme is GEOMETRY_GRID, the y-min coordinate value
bounding_box_xmax	float(53)	X-coordinate of the upper-right corner of the bounding box, one of: NULL = Not applicable for a given tessellation scheme (such as GEOGRAPHY_GRID) <i>n</i> = If tessellation_scheme is GEOMETRY_GRID, the x-max coordinate value

COLUMN NAME	DATA TYPE	DESCRIPTION
bounding_box_ymax	float(53)	Y-coordinate of upper-right corner of the bounding box, one of: NULL = Not applicable for a given tessellation scheme (such as GEOGRAPHY_GRID) <i>n</i> = If tessellation_scheme is GEOMETRY_GRID, the y-max coordinate value
level_1_grid	smallint	Grid density for the top-level grid. If tessellation_scheme is GEOMETRY_GRID or GEOGRAPHY_GRID, one of: 16 = 4 by 4 grid (LOW) 64 = 8 by 8 grid (MEDIUM) 256 = 16 by 16 grid (HIGH) NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_1_grid_desc	nvarchar(60)	Grid density for the top-level grid, one of: LOW MEDIUM HIGH NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_2_grid	smallint	Grid density for the 2nd-level grid. If tessellation_scheme is GEOMETRY_GRID or GEOGRAPHY_GRID, one of: 16 = 4 by 4 grid (LOW) 64 = 8 by 8 grid (MEDIUM) 256 = 16 by 16 grid (HIGH) NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_2_grid_desc	nvarchar(60)	Grid density for the 2nd-level grid, one of: LOW MEDIUM HIGH NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_3_grid	smallint	Grid density for the 3rd-level grid. If tessellation_scheme is GEOMETRY_GRID or GEOGRAPHY_GRID, one of: 16 = 4 by 4 grid (LOW) 64 = 8 by 8 grid (MEDIUM) 256 = 16 by 16 grid (HIGH) NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.

COLUMN NAME	DATA TYPE	DESCRIPTION
level_3_grid_desc	nvarchar(60)	Grid density for the 3rd-level grid, one of:LOW MEDIUM HIGH NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_4_grid	smallint	Grid density for the 4th-level grid. If tessellation_scheme is GEOMETRY_GRID or GEOGRAPHY_GRID, one of: 16 = 4 by 4 grid (LOW)64 = 8 by 8 grid (MEDIUM) 256 = 16 by 16 grid (HIGH) NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
level_4_grid_desc	nvarchar(60)	Grid density for the 4th-level grid, one of:< LOW MEDIUM HIGH NULL = Not applicable for given spatial index type or tessellation scheme. NULL is returned when new SQL Server 11 tessellation is used.
cells_per_object	int	Number of cells per spatial object, one of: If tessellation_scheme is GEOMETRY_GRID or GEOGRAPHY_GRID, <i>n</i> = number of cells per object NULL = Not applicable for given spatial index type or tessellation scheme

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[Object Catalog Views \(Transact-SQL\)](#)

[Spatial Indexes Overview](#)

[sys.objects \(Transact-SQL\)](#)





[sys.spatial_indexes \(Transact-SQL\)](#)

[sys.indexes \(Transact-SQL\)](#)

[sys.index_columns \(Transact-SQL\)](#)

sys.spatial_indexes (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Represents the main index information of the spatial indexes.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.indexes .
spatial_index_type	tinyint	Type of spatial index: 1 = Geometric spatial index 2 = Geographic spatial index
spatial_index_type_desc	nvarchar(60)	Type description of spatial index: GEOMETRY = geometric spatial index GEOGRAPHY = geographic spatial index
tessellation_scheme	sysname	Name of tessellation scheme: GEOMETRY_GRID, GEOMETRY_AUTO_GRID, GEOGRAPHY_GRID, GEOGRAPHY_AUTO_GRID Note: For information about tessellation schemes, see Spatial Indexes Overview .
<inherited columns>		Inherits columns from sys.indexes . The inherited columns has_filter and filter_definition appear after the columns that are specific to spatial indexes.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

See Also

[sys.objects](#) (Transact-SQL)

[sys.spatial_index_tessellations](#) (Transact-SQL)

[sys.indexes](#) (Transact-SQL)

[sys.index_columns](#) (Transact-SQL)

sys.spatial_reference_systems (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Lists the spatial reference systems (SRIDs) supported by SQL Server.

COLUMN NAME	DATA TYPE	DESCRIPTION
spatial_reference_id	int	The SRID supported by SQL Server.
authority_name	nvarchar(128)	The authority of the SRID.
authorized_spatial_reference_id	int	The SRID given by the authority named in authority_name .
well_known_text	nvarchar(4000)	The WKT representation of the SRID.
unit_of_measure	nvarchar(128)	The name of the unit of measure.
unit_conversion_factor	float	The length of the unit of measure in meters.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission.

SQL Data Warehouse and Parallel Data Warehouse Catalog Views

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This topic lists the SQL Data Warehouse and Parallel Data Warehouse catalog views.

All SQL Data Warehouse and Parallel Data Warehouse catalog views begin with **sys.pdw**.

SQL Data Warehouse and Parallel Data Warehouse Catalog Views

The following catalog views apply to both SQL Data Warehouse and Parallel Data Warehouse:

[sys.pdw_column_distribution_properties](#) (Transact-SQL)

[sys.pdw_database_mappings](#) (Transact-SQL)

[sys.pdw_distributions](#) (Transact-SQL)

[sys.pdw_index_mappings](#) (Transact-SQL)

[sys.pdw_loader_backup_run_details](#) (Transact-SQL)

[sys.pdw_loader_backup_runs](#) (Transact-SQL)

[sys.pdw_nodes_column_store_dictionaries](#) (Transact-SQL)

[sys.pdw_nodes_column_store_row_groups](#) (Transact-SQL)

[sys.pdw_nodes_column_store_segments](#) (Transact-SQL)

[sys.pdw_nodes_columns](#) (Transact-SQL)

[sys.pdw_nodes_indexes](#) (Transact-SQL)

[sys.pdw_nodes_partitions](#) (Transact-SQL)

[sys.pdw_nodes_pdw_physical_databases](#) (Transact-SQL)

[sys.pdw_nodes_tables](#) (Transact-SQL)

[sys.pdw_replicated_table_cache_state](#) (Transact-SQL)

[sys.pdw_table_distribution_properties](#) (Transact-SQL)

[sys.pdw_table_mappings](#) (Transact-SQL)

Parallel Data Warehouse Catalog Views

The following catalog views apply to Parallel Data Warehouse only:

[sys.pdw_diag_event_properties](#) (Transact-SQL)

[sys.pdw_diag_events](#) (Transact-SQL)

[sys.pdw_diag_sessions](#) (Transact-SQL)

[sys.pdw_health_alerts \(Transact-SQL\)](#)

[sys.pdw_health_component_groups \(Transact-SQL\)](#)

[sys.pdw_health_component_properties \(Transact-SQL\)](#)

[sys.pdw_health_component_status_mappings \(Transact-SQL\)](#)

[sys.pdw_health_components \(Transact-SQL\)](#)





[sys.pdw_loader_run_stages \(Transact-SQL\)](#)

See Also

[Catalog Views \(Transact-SQL\)](#)

sys.pdw_column_distribution_properties (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds distribution information for columns.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	ID of the object to which the column belongs.	
column_id	int	ID of the column.	
distribution_ordinal	tinyint	Ordinal (1-based) within set of distribution.	0 = Not a distribution column. 1 = SQL Data Warehouse is using this column to distribute the parent table.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_database_mappings (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Maps the **database_ids** of databases to the physical name used on Compute nodes, and provides the **principal id** of the database owner on the system. Join **sys.pdw_database_mappings** to **sys.databases** and **sys.pdw_nodes_pdw_physical_databases**.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
physical_name	nvarchar(36)	The physical name for the database on the Compute nodes. physical_name and database_id form the key for this view.	
database_id	int	The object ID for the database. See sys.databases (Transact-SQL) . physical_name and database_id form the key for this view.	

Examples: Parallel Data Warehouse

The following example joins sys.pdw_database_mappings to other system tables to show how databases are mapped.

```
SELECT DB.database_id, DB.name, Map.*, Phys.*
FROM sys.databases AS DB
JOIN sys.pdw_database_mappings AS Map
    ON DB.database_id = Map.database_id
JOIN sys.pdw_nodes_pdw_physical_databases AS Phys
    ON Map.physical_name = Phys.physical_name
ORDER BY DB.database_id, Phys.pdw_node_id;
```

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)





[sys.pdw_index_mappings \(Transact-SQL\)](#)

[sys.pdw_table_mappings \(Transact-SQL\)](#)

[sys.pdw_nodes_pdw_physical_databases \(Transact-SQL\)](#)

sys.pdw_diag_event_properties (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds information about which properties are associated with diagnostic events.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
event_name	nvarchar(255)	Name of the specific diagnostics event.	
property_name	nvarchar(255)	Name of a property of the event.	

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_diag_events (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds information about events that can be included in diagnostic sessions on the system.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
name	nvarchar(255)	Name of the specific diagnostics event.	
source	nvarchar(255)	Source of the event (engine, general, dms, etc.)	
is_enabled	bit	Whether the event is being published.	

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_diag_sessions (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds information regarding the various diagnostic sessions that have been created on the system.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
name	nvarchar(255)	Name of the diagnostics session. Key for this view.	
xml_data	nvarchar(4000)	XML payload describing the session.	
is_active	bit	Flag indicating whether the flag is active.	
host_address	nvarchar(255)	Address of the machine hosting the session definition (Control node).	
principal_id	int	ID of the user that created the session at the database level.	
database_id	int	ID of the database that is the scope of the diagnostic session.	

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_distributions (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds information about the distributions on the appliance. It lists one row per appliance distribution.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
distribution_id	int	Unique numeric id associated with the distribution. Key for this view.	1 to the number of Compute nodes in appliance multiplied by the number of distributions per Compute node.
pdw_node_id	int	ID of the node this distribution is on.	See pdw_node_id in sys.dm_pdw_nodes (Transact-SQL) .
name	nvarchar(32)	String identifier associated with the distribution, used as a suffix on distributed tables.	String composed of 'A-Z','a-z','0-9','_','-'. .
position	int	Position of the distribution within a node respective to other distributions on that node.	1 to the number of distributions per node.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_health_alerts (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Stores properties for the different alerts that can occur on the system; this is a catalog table for alerts.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
alert_id	int	Unique identifier of the alert. Key for this view.	NOT NULL
component_id	int	ID of the component this alert applies to. The component is a general component identifier, such as "Power Supply," and is not specific to an installation. See sys.pdw_health_components (Transact-SQL) .	NOT NULL
alert_name	nvarchar(255)	Name of the alert.	NOT NULL
state	nvarchar(32)	State of the alert.	NOT NULL Possible values: 'Operational' 'NonOperational' 'Degraded' 'Failed'
severity	nvarchar(32)	Severity of the alert.	NOT NULL Possible values: 'Informational' 'Warning' 'Error'





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
type	nvarchar(32)	Type of alert.	NOT NULL Possible values: StatusChange - The device status has changed. Threshold - A value has exceeded the threshold value.
description	nvarchar(4000)	Description of the alert.	NOT NULL
condition	nvarchar(255)	Used when type = Threshold. Defines how the alert threshold is calculated.	NULL
status	nvarchar(32)	Alert status	NULL
condition_value	bit	Indicates whether the alert is allowed to occur during system operation.	NULL Possible values 0 - alert is not generated. 1 - alert is generated.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_health_component_groups (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Stores information about logical groupings of components and devices.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
group_id	int	Unique identifier for components and devices. Key for this view.	NOT NULL
group_name	nvarchar(255)	Logical group name for the components and devices.	NOT NULL

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_health_component_properties (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Stores properties that describe a device. Some properties show device status and some properties describe the device itself.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
property_id	int	Unique identifier of the property of a component. property_id and component_id form the key for this view.	NOT NULL
component_id	int	The ID of the component. See sys.pdw_health_components (Transact-SQL) . property_id and component_id form the key for this view.	NOT NULL
property_name	nvarchar(255)	Name of the property.	NOT NULL
physical_name	nvarchar(32)	Property name as defined by the manufacturer.	NOT NULL
is_key	bit	Determines whether the device instance is unique or not unique.	NOT NULL 0 - Device instance is unique. 1 - Device instance is not unique.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_health_component_status_mappings (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Defines the mapping between the SQL Data Warehouse component statuses and the manufacturer-defined component names.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
property_id	int	Unique identifier of the property. property_id, component_id, and physical_name form the key for this view.	NOT NULL
component_id	int	The ID of the component. See sys.pdw_health_components (Transact-SQL) . property_id, component_id, and physical_name form the key for this view.	NOT NULL
physical_name	nvarchar(32)	Property name as defined by the manufacturer. property_id, component_id, and physical_name form the key for this view.	NOT NULL
logical_name	nvarchar(255)	Property name as defined by SQL Data Warehouse.	NOT NULL 0 - Device instance is unique. 1 - Device instance is not unique.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_health_components (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Stores information about all components and devices that exist in the system. These include hardware, storage devices, and network devices.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
component_id	int	Unique identifier of a component or device. Key for this view.	NOT NULL
group_id	Int	The logical component group to which this component belongs. See sys.pdw_health_components (Parallel Data Warehouse) .	NOT NULL
component_name	nvarchar(255)	Name of the component.	NOT NULL

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_index_mappings (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Maps the logical indexes to the physical name used on Compute nodes as reflected by a unique combination of **object_id** of the table holding the index and the **index_id** of a particular index within that table.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	The object ID for the logical table on which this index exists. See sys.objects (Transact-SQL) . physical_name and object_id form the key for this view.	
index_id	nvarchar(32)	The ID for the index. See sys.indexes (Transact-SQL) .	
physical_name	nvarchar(36)	The name of the index in the databases on the Compute nodes. physical_name and object_id form the key for this view.	

See Also





[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

[sys.pdw_table_mappings \(Transact-SQL\)](#)

[sys.pdw_database_mappings \(Transact-SQL\)](#)

sys.pdw_loader_backup_run_details (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse





Contains further detailed information, beyond the information in [sys.pdw_loader_backup_runs \(Transact-SQL\)](#), about ongoing and completed backup and restore operations in SQL Data Warehouse and about ongoing and completed backup, restore, and load operations in Parallel Data Warehouse. The information persists across system restarts.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
run_id	int	Unique identifier for a specific backup or restore run. run_id and pdw_node_id form the key for this view.	
pdw_node_id	int	Unique identifier of an appliance node for which this record holds details. run_id and pdw_node_id form the key for this view.	See node_id in sys.dm_pdw_nodes (Transact-SQL) .
status	nvarchar(16)	The current status of the run.	'CANCELLED', 'COMPLETED', 'FAILED', 'QUEUED', 'RUNNING'
start_time	datetime	Time at which the operation started on this particular node.	
end_time	datetime	Time at which the operation ended on this particular node, if any.	
total_elapsed_time	int	Total time the operation has been running on this particular node.	If total_elapsed_time exceeds the maximum value for an integer (24.8 days in milliseconds), it will cause materialization failure due to overflow. The maximum value in milliseconds is equivalent to 24.8 days.
progress	int	Progress of the operation expressed as a percentage.	0 to 100

See Also

sys.pdw_loader_backup_runs (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about ongoing and completed backup and restore operations in SQL Data Warehouse, and about ongoing and completed backup, restore, and load operations in Parallel Data Warehouse. The information persists across system restarts.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
run_id	int	Unique identifier for a specific backup, restore, or load run. Key for this view.	
name	nvarchar(255)	Null for load. Optional name for backup or restore.	
submit_time	datetime	Time the request was submitted.	
start_time	datetime	Time the operation started.	
end_time	datetime	Time the operation completed, failed, or was cancelled.	
total_elapsed_time	int	Total time elapsed between start_time and current time, or between start_time and end_time for completed, cancelled, or failed runs.	If total_elapsed_time exceeds the maximum value for an integer (24.8 days in milliseconds), it will cause materialization failure due to overflow. The maximum value in milliseconds is equivalent to 24.8 days.
operation_type	nvarchar(16)	The load type.	'BACKUP', 'LOAD', 'RESTORE'





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
mode	nvarchar(16)	The mode within the run type.	For operation_type = BACKUP DIFFERENTIAL FULL For operation_type = LOAD APPEND RELOAD UPSERT For operation_type = RESTORE DATABASE HEADER_ONLY
database_name	nvarchar(255)	Name of the database that is the context of this operation	
table_name	nvarchar(255)	Information not available.	
Principal_id	int	ID of the user requesting the operation.	
session_id	nvarchar(32)	ID of the session performing the operation.	See session_id in sys.dm_pdw_exec_sessions (Transact-SQL) .
request_id	nvarchar(32)	ID of the request performing the operation. For loads, this is the current or last request associated with this load..	See request_id in sys.dm_pdw_exec_requests (Transact-SQL) .
status	nvarchar(16)	Status of the run.	'CANCELLED','COMPLETED','FAILED','QUEUED','RUNNING'
progress	int	Percentage completed.	0 to 100
command	nvarchar(4000)	Full text of the command submitted by the user.	Will be truncated if longer than 4000 characters (counting spaces).
rows_processed	bigint	Number of rows processed as part of this operation.	
rows_rejected	bigint	Number of rows rejected as part of this operation.	
rows_inserted	bigint	Number of rows inserted into the database table(s) as part of this operation.	

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_loader_run_stages (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains information about ongoing and completed load operations in Parallel Data Warehouse. The information persists across system restarts.





Column Name	Data Type	Description	Range
run_id	int	Unique identifier of a loader run.	
stage	nvarchar(30)	The current stage for the run.	'CREATE_STAGING', 'DMS_LOAD', 'LOAD_INSERT', 'LOAD_CLEANUP'
request_id	nvarchar(32)	ID of the request running this stage.	
status	nvarchar(16)	Status of this phase.	
start_time	datetime	Time at which the stage was started.	
end_time	datetime	Time at which the stage ended, if any.	NULL if not started or in progress.
total_elapsed_time	int	Total time this stage spent (or spent so far) running.	If total_elapsed_time exceeds the maximum value for an integer (24.8 days in milliseconds), it will cause materialization failure due to overflow. The maximum value in milliseconds is equivalent to 24.8 days.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_nodes_column_store_dictionaries (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each dictionary used in columnstore indexes. Dictionaries are used to encode some, but not all data types, therefore not all columns in a columnstore index have dictionaries. A dictionary can exist as a primary dictionary (for all segments) and possibly for other secondary dictionaries used for a subset of the column's segments.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Indicates the partition ID. Is unique within a database.
hobt_id	bigint	ID of the heap or B-tree index (hobt) for the table that has this columnstore index.
column_id	int	ID of the columnstore column.
dictionary_id	int	Id of the dictionary.
version	int	Version of the dictionary format.
type	int	Dictionary type: 1 – Hash dictionary containing int values 2 – Not used 3 – Hash dictionary containing string values 4 – Hash dictionary containing float values
last_id	int	The last data id in the dictionary.
entry_count	bigint	Number of entries in the dictionary.
on_disc_size	bigint	Size of dictionary in bytes.
pdw_node_id	int	Unique identifier of a SQL Data Warehouse node.

Permissions

Requires VIEW SERVER STATE permission.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)





[CREATE COLUMNSTORE INDEX \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_segments \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_row_groups \(Transact-SQL\)](#)

sys.pdw_nodes_column_store_row_groups (Transact-SQL)

11/16/2017 • 4 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Provides clustered columnstore index information on a per-segment basis to help the administrator make system management decisions in SQL Data Warehouse. **sys.pdw_nodes_column_store_row_groups** has a column for the total number of rows physically stored (including those marked as deleted) and a column for the number of rows marked as deleted. Use **sys.pdw_nodes_column_store_row_groups** to determine which row groups have a high percentage of deleted rows and should be rebuilt.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of the underlying table. This is the physical table on the Compute node, not the object_id for the logical table on the Control node. For example, object_id does not match with the object_id in sys.tables. To join with sys.tables, use sys.pdw_index_mappings.
index_id	int	ID of the clustered columnstore index on <i>object_id</i> table.
partition_number	int	ID of the table partition that holds row group <i>row_group_id</i> . You can use <i>partition_number</i> to join this DMV to sys.partitions.
row_group_id	int	ID of this row group. This is unique within the partition.
dellta_store_hobt_id	bigint	The hobt_id for delta row groups, or NULL if the row group type is not delta. A delta row group is a read/write row group that is accepting new records. A delta row group has the OPEN status. A delta row group is still in rowstore format and has not been compressed to columnstore format.
state	tinyint	ID number associated with the state_description. 1 = OPEN 2 = CLOSED 3 = COMPRESSED

COLUMN NAME	DATA TYPE	DESCRIPTION
state_desc	nvarchar(60)	<p>Description of the persistent state of the row group:</p> <p>OPEN – A read/write row group that is accepting new records. An open row group is still in rowstore format and has not been compressed to columnstore format.</p> <p>CLOSED – A row group that has been filled, but not yet compressed by the tuple mover process.</p> <p>COMPRESSED – A row group that has filled and compressed.</p>
total_rows	bigint	Total rows physically stored in the row group. Some may have been deleted but they are still stored. The maximum number of rows in a row group is 1,048,576 (hexadecimal FFFF).
deleted_rows	bigint	<p>Number of rows physically stored in the row group that are marked for deletion.</p> <p>Always 0 for DELTA row groups.</p>
size_in_bytes	int	Combined size, in bytes, of all the pages in this row group. This size does not include the size required to store metadata or shared dictionaries.
pdw_node_id	int	Unique id of a SQL Data Warehouse node.
distribution_id	int	Unique id of the distribution.

Remarks

Returns one row for each columnstore row group for each table having a clustered or nonclustered columnstore index.

Use **sys.pdw_nodes_column_store_row_groups** to determine the number of rows included in the row group and the size of the row group.

When the number of deleted rows in a row group grows to a large percentage of the total rows, the table becomes less efficient. Rebuild the columnstore index to reduce the size of the table, reducing the disk I/O required to read the table. To rebuild the columnstore index use the **REBUILD** option of the **ALTER INDEX** statement.

The updateable columnstore first inserts new data into an **OPEN** rowgroup, which is in rowstore format, and is also sometimes referred to as a delta table. Once an open rowgroup is full, its state changes to **CLOSED**. A closed rowgroup is compressed into columnstore format by the tuple mover and the state changes to **COMPRESSED**. The tuple mover is a background process that periodically wakes up and checks whether there are any closed rowgroups that are ready to compress into a columnstore rowgroup. The tuple mover also deallocates any rowgroups in which every row has been deleted. Deallocated rowgroups are marked as **RETIRED**. To run tuple mover immediately, use the **REORGANIZE** option of the **ALTER INDEX** statement.

When a columnstore row group has filled, it is compressed, and stops accepting new rows. When rows are deleted from a compressed group, they remain but are marked as deleted. Updates to a compressed group are implemented as a delete from the compressed group, and an insert to an open group.

Permissions

Requires **VIEW SERVER STATE** permission.

Examples: SQL Data Warehouse and Parallel Data Warehouse

The following example joins the **sys.pdw_nodes_column_store_row_groups** table to other system tables to return information about specific tables. The calculated `PercentFull` column is an estimate of the efficiency of the row group. To find information on a single table remove the comment hyphens in front of the WHERE clause and provide a table name.

```
SELECT IndexMap.object_id,
       object_name(IndexMap.object_id) AS LogicalTableName,
       i.name AS LogicalIndexName, IndexMap.index_id, NI.type_desc,
       IndexMap.physical_name AS PhyIndexNameFromIMap,
       CSRowGroups.*,
       100*(ISNULL(deleted_rows,0))/total_rows AS PercentDeletedRows
FROM sys.tables AS t
JOIN sys.indexes AS i
     ON t.object_id = i.object_id
JOIN sys.pdw_index_mappings AS IndexMap
     ON i.object_id = IndexMap.object_id
     AND i.index_id = IndexMap.index_id
JOIN sys.pdw_nodes_indexes AS NI
     ON IndexMap.physical_name = NI.name
     AND IndexMap.index_id = NI.index_id
JOIN sys.pdw_nodes_column_store_row_groups AS CSRowGroups
     ON CSRowGroups.object_id = NI.object_id
     AND CSRowGroups.pdw_node_id = NI.pdw_node_id
     AND CSRowGroups.index_id = NI.index_id
--WHERE t.name = '<table_name>'
ORDER BY object_name(i.object_id), i.name, IndexMap.physical_name, pdw_node_id;
```

The following SQL Data Warehouse example counts the rows per partition for clustered column stores as well as how many rows are in Open, Closed, or Compressed Row groups:

```
SELECT
    s.name AS [Schema Name]
  ,t.name AS [Table Name]
  ,rg.partition_number AS [Partition Number]
  ,SUM(rg.total_rows) AS [Total Rows]
  ,SUM(CASE WHEN rg.State = 1 THEN rg.Total_rows ELSE 0 END) AS [Rows in OPEN Row Groups]
  ,SUM(CASE WHEN rg.State = 2 THEN rg.Total_Rows ELSE 0 END) AS [Rows in Closed Row Groups]
  ,SUM(CASE WHEN rg.State = 3 THEN rg.Total_Rows ELSE 0 END) AS [Rows in COMPRESSED Row Groups]
FROM sys.pdw_nodes_column_store_row_groups rg
JOIN sys.pdw_nodes_tables pt
     ON rg.object_id = pt.object_id AND rg.pdw_node_id = pt.pdw_node_id AND pt.distribution_id = rg.distribution_id
JOIN sys.pdw_table_mappings tm
     ON pt.name = tm.physical_name
INNER JOIN sys.tables t
     ON tm.object_id = t.object_id
INNER JOIN sys.schemas s
     ON t.schema_id = s.schema_id
GROUP BY s.name, t.name, rg.partition_number
ORDER BY 1, 2
```

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)





[CREATE COLUMNSTORE INDEX \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_segments \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_dictionaries \(Transact-SQL\)](#)

sys.pdw_nodes_column_store_segments (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each column in a columnstore index.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	bigint	Indicates the partition ID. Is unique within a database.
hobt_id	bigint	ID of the heap or B-tree index (hobt) for the table that has this columnstore index.
column_id	int	ID of the columnstore column.
segment_id	int	ID of the column segment.
version	int	Version of the column segment format.
encoding_type	int	Type of encoding used for that segment.
row_count	int	Number of rows in the row group.
has_nulls	int	1 if the column segment has null values.
base_id	bigint	Base value id if encoding type 1 is being used. If encoding type 1 is not being used, base_id is set to 1.
magnitude	float	Magnitude if encoding type 1 is being used. If encoding type 1 is not being used, magnitude is set to 1.
primary_dictionary_id	int	Id of primary dictionary.
secondary_dictionary_id	int	Id of secondary dictionary. Returns -1 if there is no secondary dictionary.
min_data_id	bigint	Minimum data id in the column segment.
max_data_id	bigint	Maximum data id in the column segment.

COLUMN NAME	DATA TYPE	DESCRIPTION
null_value	bigint	Value used to represent nulls.
on_disk_size	bigint	Size of segment in bytes.
pdw_node_id	int	Unique identifier of a SQL Data Warehouse note.

Examples: Azure SQL Data Warehouse and Parallel Data Warehouse

The following query returns information about segments of a columnstore index.

```
SELECT i.name, p.object_id, p.index_id, i.type_desc,
       COUNT(*) AS number_of_segments
FROM sys.column_store_segments AS s
INNER JOIN sys.partitions AS p
    ON s.hobt_id = p.hobt_id
INNER JOIN sys.indexes AS i
    ON p.object_id = i.object_id
WHERE i.type = 6
GROUP BY i.name, p.object_id, p.index_id, i.type_desc ;
```

Join sys.pdw_nodes_column_store_segments with other system tables to determine the row count and on-disk size of the segments.

```
SELECT o.name, css.hobt_id, css.column_id, css.pdw_node_id, css.row_count, css.on_disk_size
FROM sys.pdw_nodes_column_store_segments AS css
JOIN sys.pdw_nodes_partitions AS pnp
    ON css.partition_id = pnp.partition_id
JOIN sys.pdw_nodes_tables AS part
    ON pnp.object_id = part.object_id
    AND pnp.pdw_node_id = part.pdw_node_id
JOIN sys.pdw_table_mappings AS TMap
    ON part.name = TMap.physical_name
JOIN sys.objects AS o
    ON TMap.object_id = o.object_id
ORDER BY css.hobt_id, css.column_id;
```

Permissions

Requires **VIEW SERVER STATE** permission.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)





[CREATE COLUMNSTORE INDEX \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_row_groups \(Transact-SQL\)](#)

[sys.pdw_nodes_column_store_dictionaries \(Transact-SQL\)](#)

sys.pdw_nodes_columns (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Shows columns for user-defined tables and user-defined views.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	ID of the object to which this column belongs.	
name	sysname	Name of the column. Unique in object.	
column_id	int	ID of the column. Unique in object.	
system_type_id	tinyint	ID of the system type of the column.	
user_type_id	int	ID of the type of the column as defined by the user.	
max_length	smallint	Maximum length (in bytes) of the column.	Includes -1 (not valid) for unsupported column types.
precision	tinyint	Precision of the column if numeric-based; otherwise, 0.	
scale	tinyint	Scale of column if numeric-based; otherwise, 0.	
collation_name	sysname	Name of the collation of the column if character-based; otherwise, NULL.	
is_nullable	bit	1 = Column is nullable.	
is_ansi_padded	bit	1 = Column uses ANSI_PADDING ON behavior if character, binary, or variant.	Always 0.
is_rowguidcol	bit	1 = Column is a declared ROWGUIDCOL.	Always 0.
is_identity	bit	1 = Column has identity values.	Always 0.
is_computed	bit	1 = Column is a computed column.	Always 0.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
is_filestream	bit	1 = Column is a FILESTREAM column.	Always 0.
is_replicated	bit	1 = Column is replicated.	Always 0.
is_non_sql_subscribed	bit	1 = Column has a non-SQL subscriber.	Always 0.
is_merge_published	bit	1 = Column is merge-published.	Always 0.
is_dts_replicated	bit	1 = Column is replicated by using SSIS.	Always 0.
is_xml_document	bit	1 = Content is a complete XML document.	Always 0.
xml_collection_id	int	0 = No XML schema collection.	Always 0.
default_object_id	int	ID of the default object; 0 = No default.	Always 0.
rule_object_id	int	ID of the stand-alone rule bound to the column. 0 = No stand-alone rule.	Always 0.
is_sparse	bit	1 = Column is a sparse column.	Always 0.
is_column_set	bit	1 = Column is a column set.	Always 0.
pdw_node_id	int	Unique identifier of a SQL Data Warehouse node.	NOT NULL

Permissions





Requires CONTROL SERVER permission.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)
[sys.all_columns \(Transact-SQL\)](#)

sys.pdw_nodes_indexes (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns indexes for SQL Data Warehouse.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	id of the object to which this index belongs.	
name	sysname	Name of the index. Name is unique only within the object. NULL = Heap	
index_id	int	id of the index. index_id is unique only within the object. 0 = Heap 1 = Clustered index > 1 = Non-clustered index	
type	tinyint	Type of index: 0 = Heap 1 = Clustered 2 = Non-clustered 5 = Clustered xVelocity memory optimized columnstore index	
type_desc	nvarchar(60)	Description of index type: HEAP CLUSTERED NONCLUSTERED CLUSTERED COLUMNSTORE	
is_unique	bit	0 = Index is not unique.	Always 0.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
data_space_id	int	id of the data space for this index. Data space is either a filegroup or partition scheme. 0 = object_id is a table-valued function.	
ignore_dup_key	bit	0 = IGNORE_DUP_KEY is OFF.	Always 0.
is_primary_key	bit	1 = Index is part of a PRIMARY KEY constraint.	Always 0.
is_unique_constraint	bit	1 = Index is part of a UNIQUE constraint.	Always 0.
fill_factor	tinyint	> 0 = FILLFACTOR percentage used when the index was created or rebuilt. 0 = Default value	Always 0.
is_padded	bit	0 = PADINDEX is OFF.	Always 0.
is_disabled	bit	1 = Index is disabled. 0 = Index is not disabled.	
is_hypothetical	bit	0 = Index is not hypothetical.	Always 0.
allow_row_locks	bit	1 = Index allows row locks.	Always 1.
allow_page_locks	bit	1 = Index allows page locks.	Always 1.
has_filter	bit	0 = Index does not have a filter.	Always 0.
filter_definition	nvarchar(max)	Expression for the subset of rows included in the filtered index.	Always NULL.
pdw_node_id	int	Unique identifier of a SQL Data Warehouse node.	NOT NULL

Permissions





Requires CONTROL SERVER permission.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_nodes_partitions (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each partition of all the tables, and most types of indexes in a SQL Data Warehouse database. All tables and indexes contain at least one partition, whether or not they are explicitly partitioned.

COLUMN NAME	DATA TYPE	DESCRIPTION
partition_id	<code>bigint</code>	id of the partition. Is unique within a database.
object_id	<code>int</code>	id of the object to which this partition belongs. Every table or view is composed of at least one partition.
index_id	<code>int</code>	id of the index within the object to which this partition belongs.
partition_number	<code>int</code>	1-based partition number within the owning index or heap. For SQL Data Warehouse, the value of this column is 1.
hobt_id	<code>bigint</code>	ID of the data heap or B-tree that contains the rows for this partition.
rows	<code>bigint</code>	Approximate number of rows in this partition.
data_compression	<code>int</code>	Indicates the state of compression for each partition: 0 = NONE 1 = ROW 2 = PAGE 3 = COLUMNSTORE
data_compression_desc	<code>nvarchar(60)</code>	Indicates the state of compression for each partition. Possible values are NONE, ROW, and PAGE.
pdw_node_id	<code>int</code>	Unique identifier of a SQL Data Warehouse node.

Permissions

Requires CONTROL SERVER permission.

Examples: Azure SQL Data Warehouse and Parallel Data Warehouse

Example A: Display rows in each partition within each distribution

Applies to: SQL Data Warehouse, Parallel Data Warehouse

To display the number of rows in each partition within each distribution, use [DBCC PDW_SHOWPARTITIONSTATS \(SQL Server PDW\)](#) .

Example B: Uses system views to view rows in each partition of each distribution of a table

Applies to: SQL Data Warehouse

This query returns the number of rows in each partition of each distribution of the table `myTable` .





```
SELECT o.name, pnp.index_id, pnp.partition_id, pnp.rows,
       pnp.data_compression_desc, pnp.pdw_node_id
FROM sys.pdw_nodes_partitions AS pnp
JOIN sys.pdw_nodes_tables AS NTables
    ON pnp.object_id = NTables.object_id
AND pnp.pdw_node_id = NTables.pdw_node_id
JOIN sys.pdw_table_mappings AS TMap
    ON NTables.name = TMap.physical_name
    AND substring(TMap.physical_name,40, 10) = pnp.distribution_id
JOIN sys.objects AS o
    ON TMap.object_id = o.object_id
WHERE o.name = 'myTable'
ORDER BY o.name, pnp.index_id, pnp.partition_id;
```

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_nodes_pdw_physical_databases (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each physical database on a compute node. Aggregate physical database information to get detailed information about databases. To combine information, join the `sys.pdw_nodes_pdw_physical_databases` to the `sys.pdw_database_mappings` and `sys.databases` tables.

COLUMN NAME	DATA TYPE	DESCRIPTION
database_id	int	The object ID for the database. Note that this value is not same as a database_id in the sys.databases (Transact-SQL) view.
physical_name	sysname	The physical name for the database on the Shell/Compute nodes. This value is same as a value in the physical_name column in the sys.pdw_database_mappings (Transact-SQL) view.
pdw_node_id	int	Unique numeric id associated with the node.

Examples: Azure SQL Data Warehouse and Parallel Data Warehouse

A. Returning

The following query returns the name and ID of each database in master, and the corresponding database name on each compute node.

```
SELECT D.database_id AS DBID_in_master, D.name AS UserDatabaseName,
PD.pdw_node_id AS NodeID, DM.physical_name AS PhysDBName
FROM sys.databases AS D
JOIN sys.pdw_database_mappings AS DM
    ON D.database_id = DM.database_id
JOIN sys.pdw_nodes_pdw_physical_databases AS PD
    ON DM.physical_name = PD.physical_name
ORDER BY D.database_id, PD.pdw_node_ID;
```

B. Using sys.pdw_nodes_pdw_physical_databases to gather detailed object information

The following query shows information about indexes and includes useful information about the database the objects belong to objects in the database.

```

SELECT D.name AS UserDatabaseName, D.database_id AS DBIDinMaster,
DM.physical_name AS PhysDBName, PD.pdw_node_id AS NodeID,
IU.object_id, IU.index_id, IU.user_seeks, IU.user_scans, IU.user_lookups, IU.user_updates
FROM sys.databases AS D
JOIN sys.pdw_database_mappings AS DM
    ON D.database_id = DM.database_id
JOIN sys.pdw_nodes_pdw_physical_databases AS PD
    ON DM.physical_name = PD.physical_name
JOIN sys.dm_pdw_nodes_db_index_usage_stats AS IU
    ON PD.database_id = IU.database_id
ORDER BY D.database_id, IU.object_id, IU.index_id, PD.pdw_node_ID;

```

C. Using sys.pdw_nodes_pdw_physical_databases to determine the encryption state

The following query provides encryption state of the AdventureWorksPDW2012 database.

```

WITH dek_encryption_state AS
(
    SELECT ISNULL(db_map.database_id, dek.database_id) AS database_id, encryption_state
    FROM sys.dm_pdw_nodes_database_encryption_keys AS dek
        INNER JOIN sys.pdw_nodes_pdw_physical_databases AS node_db_map
            ON dek.database_id = node_db_map.database_id AND dek.pdw_node_id = node_db_map.pdw_node_id
        LEFT JOIN sys.pdw_database_mappings AS db_map
            ON node_db_map.physical_name = db_map.physical_name
        INNER JOIN sys.dm_pdw_nodes AS nodes
            ON nodes.pdw_node_id = dek.pdw_node_id
    WHERE dek.encryptor_thumbprint <> 0x
)
SELECT TOP 1 encryption_state
FROM dek_encryption_state
WHERE dek_encryption_state.database_id = DB_ID('AdventureWorksPDW2012 ')
ORDER BY (CASE encryption_state WHEN 3 THEN -1 ELSE encryption_state END) DESC;

```

See Also





[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

[sys.databases \(Transact-SQL\)](#)

[sys.pdw_database_mappings \(Transact-SQL\)](#)

sys.pdw_nodes_tables (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains a row for each table object that a principal either owns or on which the principal has been granted some permission.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
<inherited columns>		For a list of columns that this view inherits, see sys.objects .	
lob_data_space_id	int		Always 0.
filestream_data_space_id	int	Data space ID for a FILESTREAM filegroup or Information not available.	NULL
max_column_id_used	int	Maximum column ID used by this table.	
lock_on_bulk_load	bit	Table is locked on bulk load.	TBD
uses_ansi_nulls	bit	Table was created with the SET ANSI_NULLS database option ON.	1
is_replicated	bit	1 = Table is published using replication.	0; replication is not supported.
has_replication_filter	bit	1 = Table has a replication filter.	0
is_merge_published	bit	1 = Table is published using merge replication.	0; not supported.
is_sync_tran_subscribed	bit	1 = Table is subscribed using an immediate updating subscription.	0; not supported.
has_unchecked_assembly_data	bit	1 = Table contains persisted data that depends on an assembly whose definition changed during the last ALTER ASSEMBLY. Will be reset to 0 after the next successful DBCC CHECKDB or DBCC CHECKTABLE.	0; no CLR support.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
text_in_row_limit	int	0 = Text in row option is not set.	Always 0.
large_value_types_out_of_row	bit	1 = Large value types are stored out-of-row.	Always 0.
is_tracked_by_cdc	bit	1 = Table is enabled for change data capture	Always 0; no CDC support.
lock_escalation	tinyint	The value of the LOCK_ESCALATION option for the table: 2 = AUTO	Always 2.
lock_escalation_desc	nvarchar(60)	A text description of the lock_escalation option.	Always 'AUTO'.
pdw_node_id	int	Unique identifier of a SQL Data Warehouse node.	NOT NULL

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_replicated_table_cache_state (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns the state of the cache associated with a replicated table by **object_id**.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	The object ID for the table. See sys.objects (Transact-SQL) . object_id is the key for this view.	
state	nvarchar(40)	The replicated table cache state for this table.	'NotReady','Ready'

Example

This example joins sys.pdw_replicated_table_cache_state with sys.tables to retrieve the table name and the state of the replicated table cache.





```
SELECT t.[name], p.[object_id], p.[state]
FROM sys.pdw_replicated_table_cache_state p
JOIN sys.tables t ON t.object_id = p.object_id
```

Next steps

For a list of all the catalog views for SQL Data Warehouse and Parallel Data Warehouse, see [SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#).

sys.pdw_table_distribution_properties (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Holds distribution information for tables.





COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
object_id	int	ID of the table for which thee properties were specified.	
distribution_policy	tinyint	0 = UNDEFINED 1 = NONE 2 = HASH 3 = REPLICATE 4 = ROUND_ROBIN	REPLICATE only applies to Parallel Data Warehouse.
distribution_policy_desc	nvarchar(60)	UNDEFINED, NONE, HASH, REPLICATE, SEGMENTED_HEAP	SQL Data Warehouse returns either HASH or REPLICATE.

See Also

[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

sys.pdw_table_mappings (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Ties user tables to internal object names by **object_id**.

COLUMN NAME	DATA TYPE	DESCRIPTION	RANGE
physical_name	nvarchar(36)	The physical name for the table. physical_name and object_id form the key for this view.	
object_id	int	The object ID for the table. See sys.objects (Transact-SQL) . physical_name and object_id form the key for this view.	

See Also





[SQL Data Warehouse and Parallel Data Warehouse Catalog Views](#)

[sys.pdw_index_mappings \(Transact-SQL\)](#)

[sys.pdw_database_mappings \(Transact-SQL\)](#)

Stretch Database Catalog Views - sys.remote_data_archive_databases

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each remote database that stores data from a Stretch-enabled local database.





COLUMN NAME	DATA TYPE	DESCRIPTION
remote_database_id	int	The auto-generated local identifier of the remote database.
remote_database_name	sysname	The name of the remote database.
data_source_id	int	The data source used to connect to the remote server

See Also

[Stretch Database](#)

Stretch Database Catalog Views - sys.remote_data_archive_tables

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2016)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Contains one row for each remote table that stores data from a Stretch-enabled local table.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	The object ID of the Stretch-enabled local table.
remote_database_id	int	The auto-generated local identifier of the remote database.
remote_table_name	sysname	The name of the table in the remote database that corresponds to the Stretch-enabled local table.
filter_predicate	nvarchar(max)	<p>The filter predicate, if any, that identifies rows in the table to be migrated. If the value is null, the entire table is eligible to be migrated.</p> <p>For more info, see Enable Stretch Database for a table and Select rows to migrate by using a filter predicate.</p>
migration_direction	tinyint	<p>The direction in which data is currently being migrated. The available values are the following.</p> <p>1 (outbound) 2 (inbound)</p>
migration_direction_desc	nvarchar(60)	<p>The description of the direction in which data is currently being migrated. The available values are the following.</p> <p>outbound (1) inbound (2)</p>
is_migration_paused	bit	Indicates whether migration is currently paused.





COLUMN NAME	DATA TYPE	DESCRIPTION
is_reconciled	bit	<p>Indicates whether the remote table and the SQL Server table are in sync.</p> <p>When the value of is_reconciled is 1 (true), the remote table and the SQL Server table are in sync, and you can run queries that include the remote data.</p> <p>When the value of is_reconciled is 0 (false), the remote table and the SQL Server table are not in sync. Recently migrated rows have to be migrated again. This occurs when you restore the remote Azure database, or when you delete rows manually from the remote table. Until you reconcile the tables, you can't run queries that include the remote data. To reconcile the tables, run sys.sp_rda_reconcile_batch.</p>

See Also

[Stretch Database](#)

XML Schemas (XML Type System) Catalog Views (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

This section contains the following catalog views.

sys.column_xml_schema_collection_usage	sys.parameter_xml_schema_collection_usages
sys.selective_xml_index_paths	sys.xml_schema_attributes
sys.xml_schema_model_groups	sys.xml_schema_component_placements
sys.xml_schema_collections	sys.xml_schema_components
sys.xml_schema_namespaces	sys.xml_schema_elements
sys.xml_schema_types	sys.xml_schema_facets
sys.xml_schema_wildcard_namespaces	sys.xml_indexes
sys.xml_schema_wildcards	



See Also

[Catalog Views \(Transact-SQL\)](#)

[System Views \(Transact-SQL\)](#)

sys.column_xml_schema_collection_usages (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row for each column that is validated by an XML schema.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	The ID of the object to which this column belongs.
column_id	int	The ID of the column. Is unique within the object.
xml_collection_id	int	The ID of the collection that contains the validating XML schema namespace of the column.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.parameter_xml_schema_collection_usages (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse
 Parallel Data Warehouse

Returns a row for each parameter that is validated by an XML schema.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	The ID of the object to which this parameter belongs.
parameter_id	int	The ID of the parameter. Is unique within the object.
xml_collection_id	int	The ID of the XML schema collection that contains the validating XML schema namespace of the parameter.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.selective_xml_index_paths (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2012)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Available beginning in SQL Server 2012 Service Pack 1, each row in sys.selective_xml_index_paths represents one promoted path for particular selective xml index.

If you create a selective xml index on xmlcol of table T using following statement,

```
CREATE SELECTIVE XML INDEX sxi1 ON T(xmlcol)
FOR ( path1 = '/a/b/c' AS XQUERY 'xs:string',
      path2 = '/a/b/d' AS XQUERY 'xs:double'
    )
```

There will be two new rows in sys.selective_xml_index_paths corresponding to the index sxi1.

COLUMN NAME	DATA TYPE	DESCRIPTION
object_id	int	ID of table with XML column.
index_id	int	Unique id of the selective xml index.
path_id	int	Promoted XML path id.
path	nvarchar(4000)	Promoted path. For example, '/a/b/c/d/e'.
name	sysname	Path name.
path_type	tinyint	0 = XQUERY 1 = SQL
path_type_desc	sysname	Based on path_type value 'XQUERY' or 'SQL'.
xml_component_id	int	Unique ID of the XML schema component in the database.
xquery_type_description	nvarchar(4000)	Name of the specified xsd type.
is_xquery_type_inferred	bit	1 = type is inferred.
xquery_max_length	smallint	Max length (in character of xsd type).
is_xquery_max_length_inferred	bit	1 = maximum length is inferred.
is_node	bit	0 = node() hint not present. 1 = node() optimization hint applied.

COLUMN NAME	DATA TYPE	DESCRIPTION
-------------	-----------	-------------

system_type_id	tinyint	ID of the system type of the column.
user_type_id	tinyint	ID of the user type of the column.
max_length	smallint	Max Length (in bytes) of the type. -1 = Column data type is varchar(max), nvarchar(max), varbinary(max), or xml.
precision	tinyint	Maximum precision of the type if it is numeric-based. Otherwise 0.
scale	tinyint	Maximum scale of the type if it is numeric-based. Otherwise, 0.
collation_name	sysname	Name of the collation of the type if it is character-based. Otherwise, NULL.
is_singleton	bit	0 = SINGLETON hint not present. 1 = SINGLETON optimization hint applied.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_attributes (Transact-SQL)

11/21/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema component that is an attribute, **symbol_space** of **A**.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	--	Inherits from sys.xml_schema_components .
is_default_fixed	bit	1 = The default value is a fixed value. This value cannot be overridden in an XML instance. 0 = The default value is not a fixed value for the attribute. (default)
must_be_qualified	bit	1 = The attribute must be explicitly namespace qualified. 0 = The attribute may be implicitly namespace qualified. (default)
default_value	nvarchar (4000)	Default value of the attribute. Is NULL if a default value is not supplied.

Permissions





The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)
[Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_component_placements (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per placement for XML schema components.

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_component_id	int	ID of the XML schema component that owns this placement.
placement_id	int	ID of the placement. This is unique within the owning XML schema component.
placed_xml_component_id	int	ID of the placed XML schema component.
is_default_fixed	bit	1 = The default value is a fixed value. This value cannot be overridden in an XML instance. 0 = The value can be overridden. (default)
min_occurrences	int	Minimum number of placed component occurs.
max_occurrences	int	Maximum number of placed component occurs.
default_value	nvarchar (4000)	Default value if one is supplied. Is NULL if a default value is not supplied.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_components (Transact-SQL)

11/27/2017 • 2 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per component of an XML schema. The pair (**collection_id**, **namespace_id**) is a compound foreign key to the containing namespace. For named components, the values for **symbol_space**, **name**, **scoping_xml_component_id**, **is_qualified**, **xml_namespace_id**, **xml_collection_id** are unique.

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_component_id	int	Unique ID of the XML schema component in the database.
xml_collection_id	int	ID of the XML schema collection that contains the namespace of this component.
xml_namespace_id	int	ID of the XML namespace within the collection.
is_qualified	bit	<p>1 = This component has an explicit namespace qualifier.</p> <p>0 = This is a locally scoped component. In this case, the pair, namespace_id, collection_id, refers to the "no namespace" targetNamespace.</p> <p>For wildcard components this value will be equal to 1.</p>
name	nvarchar (4000)	Unique name of the XML schema component. Is NULL if the component is unnamed.
symbol_space	char(1)	<p>Space in which this symbol name is unique, based on kind:</p> <p>N = None</p> <p>T = Type</p> <p>E = Element</p> <p>M = Model-Group</p> <p>A = Attribute</p> <p>G = Attribute-Group</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
symbol_space_desc	nvarchar (60)	<p>Description of space in which this symbol name is unique, based on kind:</p> <p>NONE</p> <p>TYPE</p> <p>ELEMENT</p> <p>MODEL_GROUP</p> <p>ATTRIBUTE</p> <p>ATTRIBUTE_GROUP</p>
kind	char(1)	<p>Kind of XML schema component.</p> <p>N = Any Type (special intrinsic component)</p> <p>Z = Any Simple Type (special intrinsic component)</p> <p>P = Primitive Type (intrinsic types)</p> <p>S = Simple Type</p> <p>L = List Type</p> <p>U = Union Type</p> <p>C = Complex Simple Type (derived from Simple)</p> <p>K = Complex Type</p> <p>E = Element</p> <p>M = Model-Group</p> <p>W = Element-Wildcard</p> <p>A = Attribute</p> <p>G = Attribute-Group</p> <p>V = Attribute-Wildcard</p>

COLUMN NAME	DATA TYPE	DESCRIPTION
kind_desc	nvarchar (60)	<p>Description of the kind of XML schema component:</p> <p>ANY_TYPE</p> <p>ANY_SIMPLE_TYPE</p> <p>PRIMITIVE_TYPE</p> <p>SIMPLE_TYPE</p> <p>LIST_TYPE</p> <p>UNION_TYPE</p> <p>COMPLEX_SIMPLE_TYPE</p> <p>COMPLEX_TYPE</p> <p>ELEMENT</p> <p>MODEL_GROUP</p> <p>ELEMENT_WILDCARD</p> <p>ATTRIBUTE</p> <p>ATTRIBUTE_GROUP</p> <p>ATTRIBUTE_WILDCARD</p>
derivation	char(1)	<p>Derivation method for derived types:</p> <p>N = None (not derived)</p> <p>X = Extension</p> <p>R = Restriction</p> <p>S = Substitution</p>
derivation_desc	nvarchar (60)	<p>Description of derivation method for derived types:</p> <p>NONE</p> <p>EXTENSION</p> <p>RESTRICTION</p> <p>SUBSTITUTION</p>
base_xml_component_id	int	ID of the component from which this component is derived. NULL if there is none.
scoping_xml_component_id	int	Unique ID of the scoping component. NULL if there is none (global scope).

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).


See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_elements (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema component that is a Type, **symbol_space** of **E**.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>	--	Inherits columns from sys.xml_schema_components .
is_default_fixed	bit	1 = Default value is a fixed value. This value cannot be overridden in XML instance. 0 = Default value is not a fixed value for the element. (default).
is_abstract	bit	1 = Element is abstract and cannot be used in an instance document. A member of the substitution group of the element must appear in the instance document. 0 = Element is not abstract. (default).
is_nillable	bit	1 = Element is nillable. 0 = Element is not nillable. (default)
must_be_qualified	bit	1 = Element must be explicitly namespace qualified. 0 = Element may be implicitly namespace qualified. (default)
is_extension_blocked	bit	1 = Replacement with an instance of an extension type is blocked. 0 = Replacement with extension type is allowed. (default)
is_restriction_blocked	bit	1 = Replacement with an instance of a restriction type is blocked. 0 = Replacement with restriction type is allowed. (default)
is_substitution_blocked	bit	1 = Instance of a substitution group cannot be used. 0 = Replacement with substitution group is permitted. (default)

COLUMN NAME	DATA TYPE	DESCRIPTION
is_final_extension	bit	<p>1 = Replacement with an instance of an extension type is disallowed.</p> <p>0 = Replacement in an instance of an extension type is allowed. (default)</p>
is_final_restriction	bit	<p>1 = Replacement with an instance of a restriction type is disallowed.</p> <p>0 = Replacement in an instance of a restriction type is allowed. (default)</p>
default_value	nvarchar (4000)	Default value of the element. NULL if a default value is not supplied.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_facets (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per facet (restriction) of an xml-type definition (corresponds to **sys.xml_types**).

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_component_id	int	ID of XML component (type) to which this facet belongs.
facet_id	int	ID (1-based ordinal) of facet, unique within component-id.
kind	char(2)	Kind of facet: LG = Length LN = Minimum Length LX = Maximum Length PT = Pattern (regular expression) EU = Enumeration IN = Minimum Inclusive value IX = Maximum Inclusive value EN = Minimum Exclusive value EX = Maximum Exclusive value DT = Total Digits DF = Fraction Digits WS = White Space normalization

COLUMN NAME	DATA TYPE	DESCRIPTION
kind_desc	nvarchar (60)	Description of kind of facet: LENGTH MINIMUM_LENGTH MAXIMUM_LENGTH PATTERN ENUMERATION MINIMUM_INCLUSIVE_VALUE MAXIMUM_INCLUSIVE_VALUE MINIMUM_EXCLUSIVE_VALUE MAXIMUM_EXCLUSIVE_VALUE TOTAL_DIGITS FRACTION_DIGITS WHITESPACE_NORMALIZATION
is_fixed	bit	1 = Facet has a fixed, prespecified value. 0 = No fixed value. (default)
value	nvarchar (4000)	Fixed, pre-specified value of the facet.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).





See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_indexes (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns one row per XML index.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.indexes .
using_xml_index_id	int	NULL = Primary XML index. Nonnull = Secondary XML index. Nonnull is a self-join reference to the primary XML index.
secondary_type	char(1)	Type description of secondary index: P = PATH secondary XML index V = VALUE secondary XML index R = PROPERTY secondary XML index NULL = Primary XML index
secondary_type_desc	nvarchar(60)	Type description of secondary index: PATH = PATH secondary XML index VALUE = VALUE secondary XML index PROPERTY = PROPERTY secondary xml indexes. NULL = Primary XML index
xml_index_type	tinyint	Index type: 0 = Primary XML index 1 = Secondary XML index 2 = Selective XML index 3 = Secondary selective XML index

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_index_type_description	nvarchar(60)	Description of index type: PRIMARY_XML Secondary XML Index Selective XML Index Secondary Selective XML index
path_id	int	NULL for all XML indexes except secondary selective XML index. Else, the ID of the promoted path over which the secondary selective XML index is built. This value is the same value as path_id from sys.selective_xml_index_paths system view.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[Object Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_model_groups (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema component that is a Model-Group, **symbol_space** of **M**..

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.xml_schema_components .
compositor	char(1)	Compositor kind of group: A = XSD <all> Group C = XSD <choice> Group S = XSD <sequence> Group
compositor_desc	nvarchar (60)	Description of compositor kind of group: XSD_ALL_GROUP XSD_CHOICE_GROUP XSD_SEQUENCE_GROUP

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_collections (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema collection. An XML schema collection is a named set of XSD definitions. The XML schema collection itself is contained in a relational schema, and it is identified by a schema-scoped Transact-SQL name. The following tuples are unique: xml_collection_id, and schema_id and name.

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_collection_id	int	ID of the XML schema collection. Unique within the database.
schema_id	int	ID of the relational schema that contains this XML schema collection.
principal_id	int	ID of the individual owner if different from the schema owner. By default, schema-contained objects are owned by the schema owner. However, an alternate owner may be specified by using the ALTER AUTHORIZATION statement to change ownership. NULL = No alternate individual owner.
name	sysname	Name of the XML schema collection.
create_date	datetime	Date the XML schema collection was created.
modify_date	datetime	Date the XML schema collection was last altered.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

[Querying the SQL Server System Catalog FAQ](#)

sys.xml_schema_namespaces (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XSD-defined XML namespace. The following tuples are unique: **collection_id**, **namespace_id**, and **collection_id**, and **name**.

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_collection_id	int	ID of the XML schema collection that contains this namespace.
name	nvarchar(4000)	Name of XML namespace. Blank name indicates no target namespace.
xml_namespace_id	int	1-based ordinal that uniquely identifies the XML namespace in the database.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).



See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_types (Transact-SQL)

11/27/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema component that is a Type, **symbol_space** of **T**.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.xml_schema_components .
is_abstract	bit	1 = Type is an abstract type. All instances of an element of this type must use xsi:type to indicate a derived type that is not abstract. 0 = Type is not abstract. (default)
allows_mixed_content	bit	1 = Mixed content is allowed 0 = Mixed content is not allowed. (default)
is_extension_blocked	bit	1 = Replacement with an extension of the type is blocked in instances when the block attribute on the complexType definition or the blockDefault attribute of the ancestor <schema> element information item is set to "extension" or "#all". 0 = Replacement with extension is not blocked.
is_restriction_blocked	bit	1 = Replacement with a restriction of the type is blocked in instances when the block attribute on the complexType definition or the blockDefault attribute of the ancestor <schema> element information item is set to "restriction" or "#all". 0 = Replacement with restriction is not blocked. (default)
is_final_extension	bit	1 = Derivation by extension of the type is blocked when the final attribute on the complexType definition or the finalDefault attribute of the ancestor <schema> element information item is set to "extension" or "#all". 0 = Extension is allowed. (default)

COLUMN NAME	DATA TYPE	DESCRIPTION
is_final_restriction	bit	<p>1 = Derivation by restriction of the type is blocked when the final attribute on the simple or complexType definition or the finalDefault attribute of the ancestor <schema> element information item is set to "restriction" or "#all".</p> <p>0 = Restriction is allowed. (default)</p>
is_final_list_member	bit	<p>1 = This simple type cannot be used as the item type in a list.</p> <p>0 = This type is a complex type, or it can be used as list item type. (default)</p>
is_final_union_member	bit	<p>1 = This simple type cannot be used as the member type of a union type.</p> <p>0 = This type is a complex type. or it can be used as union member type. (default)</p>

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_wildcard_namespaces (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per enumerated namespace for an XML schema wildcard.

COLUMN NAME	DATA TYPE	DESCRIPTION
xml_component_id	int	ID of the XML schema component (wildcard) to which this applies.
namespace	nvarchar(4000)	Name or URI of the namespace that is used by the XML wildcard.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)

sys.xml_schema_wildcards (Transact-SQL)

11/16/2017 • 1 min to read • [Edit Online](#)

THIS TOPIC APPLIES TO:  SQL Server (starting with 2008)  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

Returns a row per XML schema component that is an Attribute-Wildcard (**kind** of **V**) or Element-Wildcard (**kind** of **W**), both with **symbol_space** of **N**.

COLUMN NAME	DATA TYPE	DESCRIPTION
<inherited columns>		Inherits columns from sys.xml_schema_components .
process_content	char(1)	Indicates how contents are processed. S = Strict validation (must validate) L = Lax validation (validate if possible) P = Skip validation
process_content_desc	nvarchar(60)	Description of how contents are processed: STRICT_VALIDATION LAX_VALIDATION SKIP_VALIDATION
disallow_namespaces	bit	0 = Namespaces enumerated in sys.xml_schema_wildcard_namespaces are the only ones allowed. 1 = Namespaces are the only ones disallowed.

Permissions

The visibility of the metadata in catalog views is limited to securables that a user either owns or on which the user has been granted some permission. For more information, see [Metadata Visibility Configuration](#).

See Also

[Catalog Views \(Transact-SQL\)](#)

[XML Schemas \(XML Type System\) Catalog Views \(Transact-SQL\)](#)