REVOKE (Transact-SQL)

07/26/2017 • 5 minutes to read • Contributors

In this article

Syntax
Arguments
Remarks
Permissions
Securable-specific Syntax
See Also

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

Removes a previously granted or denied permission.

Transact-SQL Syntax Conventions

Syntax

Copy

```
-- Syntax for SQL Server and Azure SQL Database

-- Simplified syntax for REVOKE
REVOKE [ GRANT OPTION FOR ]
      {
        [ ALL [ PRIVILEGES ] ]
        |
                 permission [ ( column [ ,...n ] ) ] [ ,...n ]
      }
      [ ON [ class :: ] securable ]
      { TO | FROM } principal [ ,...n ]
      [ CASCADE] [ AS principal ]
```

FEEDBACK

```
-- Syntax for Azure SQL Data Warehouse and Parallel Data Warehouse

REVOKE
    <permission> [ ,...n ]
    [ ON [ <class_type> :: ] securable ]
    [ FROM | TO ] principal [ ,...n ]
    [ CASCADE ]
[;]

<permission> ::=
{ see the tables below }

<class_type> ::=
{
      LOGIN
    | DATABASE
    | OBJECT
    | ROLE
    | SCHEMA
    | USER
}
```

Arguments

GRANT OPTION FOR
Indicates that the ability to grant the specified permission will be revoked. This is required when you are using the CASCADE argument.

> **Important**
>
> If the principal has the specified permission without the GRANT option, the permission itself will be revoked.

ALL
Applies to: SQL Server 2008 through SQL Server 2017

This option does not revoke all possible permissions. Revoking ALL is equivalent to revoking the following permissions.

- If the securable is a database, ALL means BACKUP DATABASE, BACKUP LOG, CREATE DATABASE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE RULE, CREATE TABLE, and CREATE VIEW.
- If the securable is a scalar function, ALL means EXECUTE and REFERENCES.

- If the securable is a table-valued function, ALL means DELETE, INSERT, REFERENCES, SELECT, and UPDATE.
- If the securable is a stored procedure, ALL means EXECUTE.
- If the securable is a table, ALL means DELETE, INSERT, REFERENCES, SELECT, and UPDATE.
- If the securable is a view, ALL means DELETE, INSERT, REFERENCES, SELECT, and UPDATE.

> **Note**
>
> The REVOKE ALL syntax 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. Revoke specific permissions instead.

PRIVILEGES
Included for ISO compliance. Does not change the behavior of ALL.

permission
Is the name of a permission. The valid mappings of permissions to securables are described in the topics listed in Securable-specific Syntax later in this topic.

column
Specifies the name of a column in a table on which permissions are being revoked. The parentheses are required.

class
Specifies the class of the securable on which the permission is being revoked. The scope qualifier **::** is required.

securable
Specifies the securable on which the permission is being revoked.

TO | FROM *principal*
Is the name of a principal. The principals from which permissions on a securable can be revoked vary, depending on the securable. For more information about valid combinations, see the topics listed in Securable-specific Syntax later in this topic.

CASCADE
Indicates that the permission that is being revoked is also revoked from other principals to which it has been granted by this principal. When you are using the CASCADE argument, you must also include the GRANT OPTION FOR argument.

> **Caution**

A cascaded revocation of a permission granted WITH GRANT OPTION will revoke both GRANT and DENY of that permission.

AS *principal*
Use the AS principal clause to indicate that you are revoking a permission that was granted by a principal other than you. For example, presume that user Mary is principal_id 12 and user Raul is principal 15. Both Mary and Raul grant a user named Steven the same permission. The sys.database_permissions table will indicate the permissions twice but they will each have a different grantor_prinicpal_id value. Mary could revoke the permission using the `AS RAUL` clause to remove Raul's grant of the permission.

The use of AS in this statement does not imply the ability to impersonate another user.

Remarks

The full syntax of the REVOKE statement is complex. The syntax diagram above was simplified to draw attention to its structure. Complete syntax for revoking permissions on specific securables is described in the topics listed in Securable-specific Syntax later in this topic.

The REVOKE statement can be used to remove granted permissions, and the DENY statement can be used to prevent a principal from gaining a specific permission through a GRANT.

Granting a permission removes DENY or REVOKE of that permission on the specified securable. If the same permission is denied at a higher scope that contains the securable, the DENY takes precedence. However, revoking the granted permission at a higher scope does not take precedence.

> ⚠ **Caution**
>
> A table-level DENY does not take precedence over a column-level GRANT. This inconsistency in the permissions hierarchy has been preserved for backward compatibility. It will be removed in a future release.

The sp_helprotect system stored procedure reports permissions on a database-level securable

The REVOKE statement will fail if CASCADE is not specified when you are revoking a permission from a principal that was granted that permission with GRANT OPTION specified.

Permissions

Principals with CONTROL permission on a securable can revoke permission on that securable. Object owners can revoke permissions on the objects they own.

Grantees of CONTROL SERVER permission, such as members of the sysadmin fixed server role, can revoke any permission on any securable in the server. Grantees of CONTROL permission on a database, such as

members of the db_owner fixed database role, can revoke any permission on any securable in the database. Grantees of CONTROL permission on a schema can revoke any permission on any object within the schema.

Securable-specific Syntax

The following table lists the securables and the topics that describe the securable-specific syntax.

Securable	Topic
Application Role	REVOKE Database Principal Permissions (Transact-SQL)
Assembly	REVOKE Assembly Permissions (Transact-SQL)
Asymmetric Key	REVOKE Asymmetric Key Permissions (Transact-SQL)
Availability Group	REVOKE Availability Group Permissions (Transact-SQL)
Certificate	REVOKE Certificate Permissions (Transact-SQL)
Contract	REVOKE Service Broker Permissions (Transact-SQL)
Database	REVOKE Database Permissions (Transact-SQL)
Endpoint	REVOKE Endpoint Permissions (Transact-SQL)
Database Scoped Credential	REVOKE Database Scoped Credential (Transact-SQL)
Full-text Catalog	REVOKE Full-Text Permissions (Transact-SQL)
Full-Text Stoplist	REVOKE Full-Text Permissions (Transact-SQL)
Function	REVOKE Object Permissions (Transact-SQL)
Login	REVOKE Server Principal Permissions (Transact-SQL)
Message Type	REVOKE Service Broker Permissions (Transact-SQL)
Object	REVOKE Object Permissions (Transact-SQL)
Queue	REVOKE Object Permissions (Transact-SQL)
Remote Service Binding	REVOKE Service Broker Permissions (Transact-SQL)
Role	REVOKE Database Principal Permissions (Transact-SQL)

Route	REVOKE Service Broker Permissions (Transact-SQL)
Schema	REVOKE Schema Permissions (Transact-SQL)
Search Property List	REVOKE Search Property List Permissions (Transact-SQL)
Server	REVOKE Server Permissions (Transact-SQL)
Service	REVOKE Service Broker Permissions (Transact-SQL)
Stored Procedure	REVOKE Object Permissions (Transact-SQL)
Symmetric Key	REVOKE Symmetric Key Permissions (Transact-SQL)
Synonym	REVOKE Object Permissions (Transact-SQL)
System Objects	REVOKE System Object Permissions (Transact-SQL)
Table	REVOKE Object Permissions (Transact-SQL)
Type	REVOKE Type Permissions (Transact-SQL)
User	REVOKE Database Principal Permissions (Transact-SQL)
View	REVOKE Object Permissions (Transact-SQL)
XML Schema Collection	REVOKE XML Schema Collection Permissions (Transact-SQL)

See Also

Permissions Hierarchy (Database Engine)
DENY (Transact-SQL)
GRANT (Transact-SQL)
sp_addlogin (Transact-SQL)
sp_adduser (Transact-SQL)
sp_changedbowner (Transact-SQL)
sp_dropuser (Transact-SQL)
sp_helprotect (Transact-SQL)
sp_helpuser (Transact-SQL)

ⓘ **Note**

The feedback system for this content will be changing soon. Old comments will not be carried over. If content within a comment thread is important to you, please save a copy. For more information on the upcoming change, we invite you to read our blog post.

0 Comments

Sign in

0 people listening

+ Follow

Post comment as...