Microsoft SQL Server 2017 and Azure SQL Database
Database Engine Permissions

Permission Syntax
Most permission statements have the format :

AUTHORIZATION PERMISSION ON SECURABLE::NAME TO PRINCIPAL

- AUTHORIZATION must be GRANT, REVOKE or DENY.
- PERMISSION is listed in the charts below.
- ON SECURABLE::NAME is the server, server object, database, or database object and its name. (ON SECURABLE::NAME is omitted for server-wide and database-wide permissions.)
- PRINCIPAL is the login, user, or role which receives or loses the permission. Grant permissions to roles whenever possible.

Sample grant statement: GRANT UPDATE ON OBJECT::Production.Parts TO PartsTeam
Denying a permission at any level, overrides a related grant.
To remove a previously granted permission, use REVOKE, not DENY.

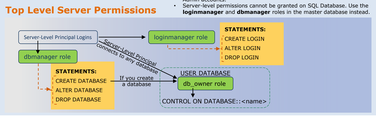
How to Read this Chart
- Most of the more granular permissions are included in more than one higher level scope permission. So permissions can be inherited from more than one type of higher scope.
- Black, green, and purple arrows and boxes point to subordinate permissions that are included in the scope of higher a level permission.
- Brown arrows and boxes indicate some of the statements that can use the permission.
- Permissions in black apply to both SQL Server 2016 and Azure SQL Database
- Permissions in red apply only to SQL Server 2016 and later
- Permissions marked with § apply only to SQL Server 2017
- Permissions marked with ǂ apply to SQL Server 2017 and Azure SQL Database
- Permissions in blue apply only to Azure SQL Database
- The newest permissions are underlined

NOTES:
- The CONTROL SERVER permission has all permissions on the instance of SQL Server or SQL Database.
- The CONTROL DATABASE permission has all permissions on the database.
- Permissions do not imply role memberships and role memberships do not grant permissions. (E.g. CONTROL SERVER does not imply membership in the sysadmin fixed server role. Membership in the db_owner role does not grant the CONTROL DATABASE permission.) However, it is sometimes possible to impersonate between roles and equivalent permissions.
- Granting any permission on a securable allows VIEW DEFINITION on that securable. It is an implied permissions and it cannot be revoked, but it can be explicitly denied by using the DENY VIEW DEFINITION statement.
- SQL Database permissions refer to version 12.
- Object owners can delete them but they do not have full permissions on them.
- A DENY on a table is overridden by a GRANT on a column. However, a subsequent DENY on the table will remove the column GRANT.

Azure SQL Database Permissions Outside the Database

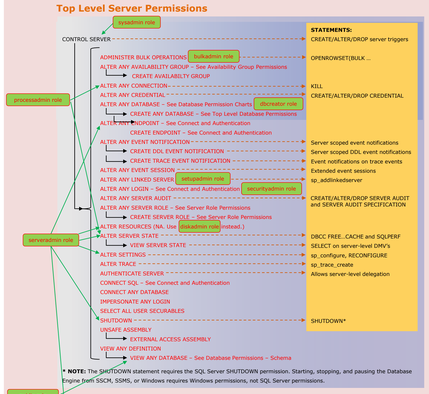
Notes:
- Server-Level Principal Logins are the Server admin and Azure Active Directory Admin accounts
- Server-level permissions cannot be granted on SQL Database. Use the **loginmanager** and **dbmanager** roles in the master database instead.

Top Level Server Permissions



Server Level Permissions for SQL Server

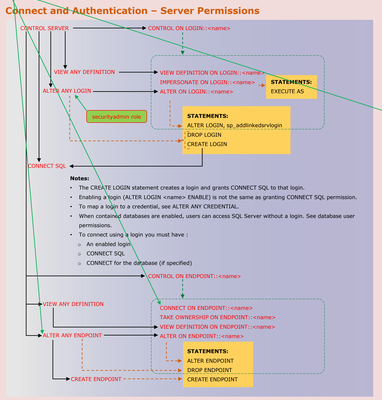
Top Level Server Permissions



* **NOTE:** The SHUTDOWN statement requires the SQL Server SHUTDOWN permission. Starting, stopping, and pausing the Database Engine from SSCM, SSMS, or Windows requires Windows permissions, not SQL Server permissions.

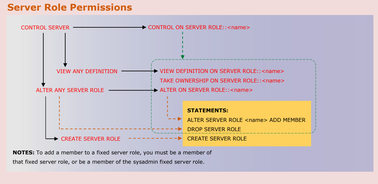
public role

Connect and Authentication – Server Permissions



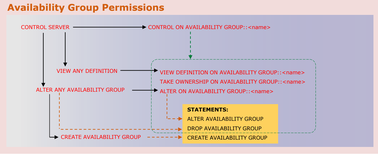
Notes:
- The CREATE LOGIN statement creates a login and grants CONNECT SQL to that login.
- Enabling a login (ALTER LOGIN <name> ENABLE) is not the same as granting CONNECT SQL permission.
- To map a login to a credential, see ALTER ANY CREDENTIAL.
- When contained databases are enabled, users can access SQL Server without a login. See database user permissions.
- To connect using a login you must have :
  - An enabled login
  - CONNECT SQL
  - CONNECT for the database (if specified)

Server Role Permissions



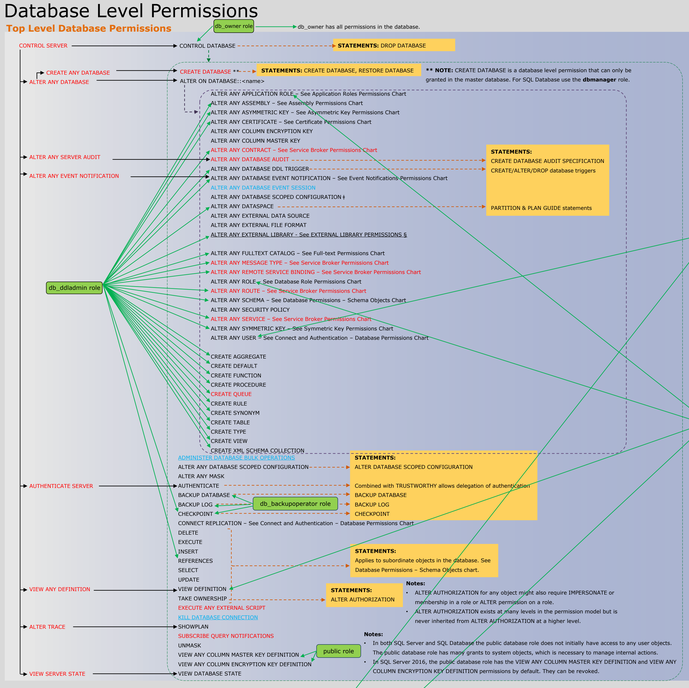
NOTES: To add a member to a fixed server role, you must be a member of
that fixed server role, or be a member of the sysadmin fixed server role.

Availability Group Permissions



Database Level Permissions

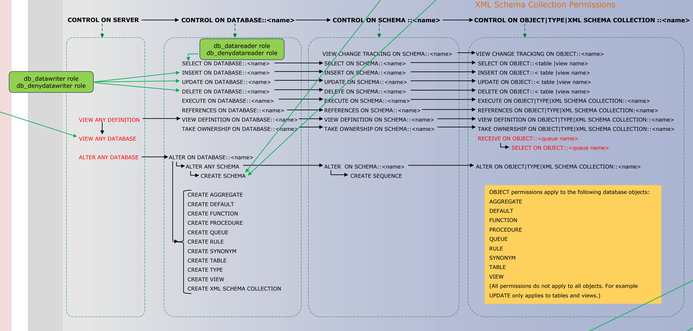
Top Level Database Permissions



Notes:
- ALTER AUTHORIZATION for any object might also require IMPERSONATE or membership in a role or ALTER permission on a role.
- ALTER AUTHORIZATION exists at many levels in the permission model but is never inherited from ALTER AUTHORIZATION at a higher level.

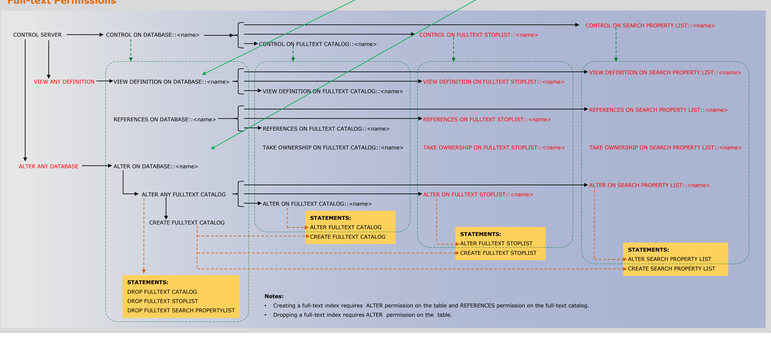
Notes:
- In both SQL Server and SQL Database the public database role does not initially have access to any user objects. The public database role has many grants to system objects, which is necessary to manage internal actions.
- In SQL Server 2016, the public database role has the VIEW ANY COLUMN MASTER KEY DEFINITION and VIEW ANY COLUMN ENCRYPTION KEY DEFINITION permissions by default. They can be revoked.

Database Permissions – Schema Objects



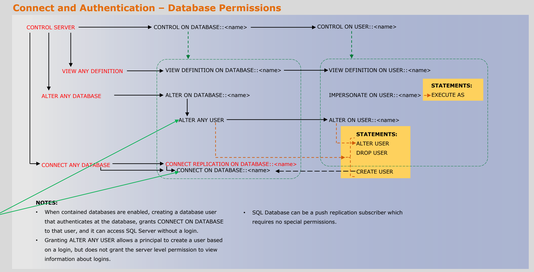
Notes:
- To create a schema object (such as a table) you must have CREATE permission for that object type plus ALTER ON SCHEMA::<name> for the schema of the object. Might require REFERENCES ON OBJECT::<name> for any referenced CLR type or XML schema collection.
- To alter an object (such as a table) you must have ALTER permission on the object (or schema), or CONTROL permission on the object.
- To drop an object (such as a table) you must have ALTER permission on the schema or CONTROL permission on the object.
- To create an index requires ALTER OBJECT::<name> permission on the table or view.
- To create or alter a trigger on a table or view requires ALTER OBJECT::<name> on the table or view.
- To create statistics requires ALTER OBJECT::<name> on the table or view.

Full-text Permissions



Notes:
- Creating a full-text index requires ALTER permission on the table and REFERENCES permission on the full-text catalog.
- Dropping a full-text index requires ALTER permission on the table.

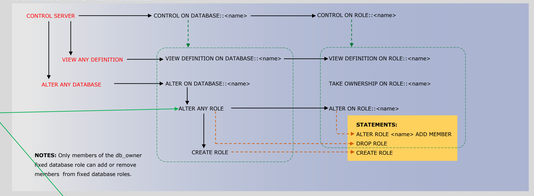
Connect and Authentication – Database Permissions



db_accessadmin role

NOTES:
- When contained databases are enabled, creating a database user that authenticates at the database, grants CONNECT ON DATABASE to that user, and it can access SQL Server without a login.
- Granting ALTER ANY USER allows a principal to create a user based on a login, but does not grant the server level permission to view information about logins.
- SQL Database can be a push replication subscriber which requires no special permissions.

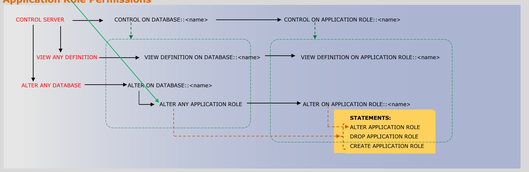
Database Role Permissions



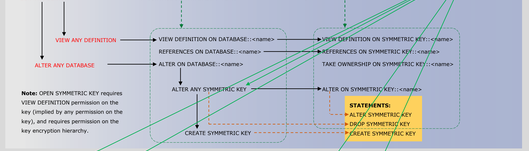
db_securityadmin role

NOTES: Only members of the db_owner fixed database role can add or remove members from fixed database roles.

Application Role Permissions



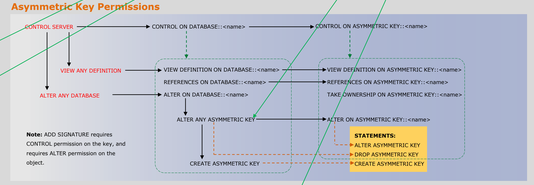
Symmetric Key Permissions



Note: OPEN SYMMETRIC KEY requires VIEW DEFINITION permission on the key (implied by any permission on the key), and requires permission on the key encryption hierarchy.

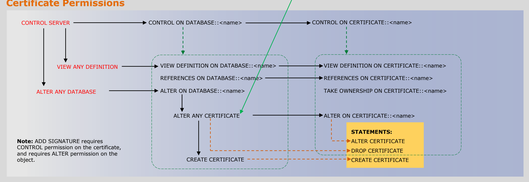
db_ddladmin role

Asymmetric Key Permissions



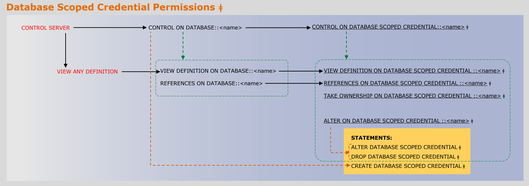
Note: ADD SIGNATURE requires CONTROL permission on the key, and requires ALTER permission on the object.

Certificate Permissions

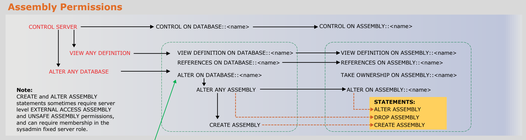


Note: ADD SIGNATURE requires CONTROL permission on the certificate, and requires ALTER permission on the object.

Database Scoped Credential Permissions ǂ



Assembly Permissions

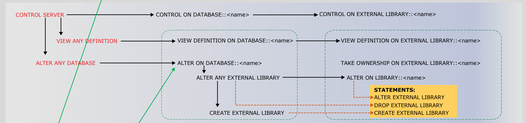


Note:
CREATE and ALTER ASSEMBLY statements sometimes require server level permissions ACCESS ASSEMBLY and UNSAFE ASSEMBLY permissions, and can require membership in the sysadmin fixed server role.

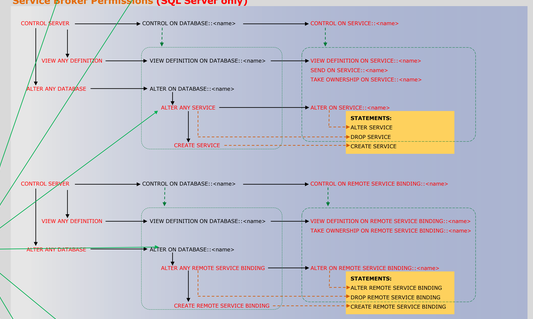
Event Notification Permissions (SQL Server only)

Note: EVENT NOTIFICATION permissions also affect service broker. See the service broker chart for more info.

External Library Permissions



Service Broker Permissions (SQL Server only)



Notes:
- The user executing the CREATE CONTRACT statement must have REFERENCES permission on all message types specified.
- The user executing the CREATE SERVICE statement must have REFERENCES permission on the queue and all contracts specified.
- To execute the CREATE or ALTER REMOTE SERVICE BINDING the user must have REFERENCES permission on the service specified in the statement.
- When the CREATE or ALTER MESSAGE TYPE statement specifies a schema collection, the user executing the statement must have REFERENCES permission on the schema collection specified.
- See the ALTER ANY EVENT NOTIFICATION chart for more permissions related to Service Broker.
- See the SCHEMA OBJECTS chart for QUEUE permissions.
- The ALTER CONTRACT permission exists but at this time there is no ALTER CONTRACT statement.

Questions and comments to
https://aka.ms/sql-permissions

Microsoft

February 28, 2018

© 2018 Microsoft Corporation. All rights reserved.