#### 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.

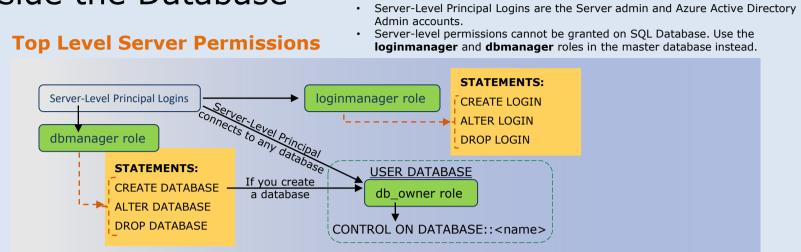
#### How to Read this Chart

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

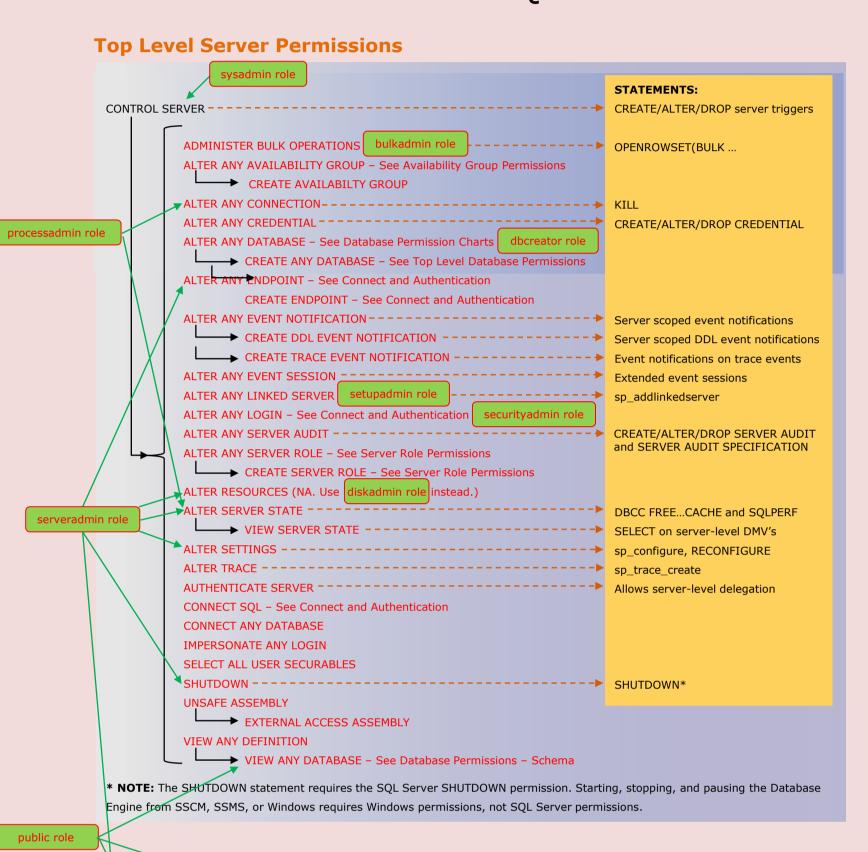
- from more than one type of higher scope.

  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 ‡ apply to SQL Server 2017 and Azure SQL Database
- Permissions in blue apply only to Azure SQL Database
   The newest permissions are underlined

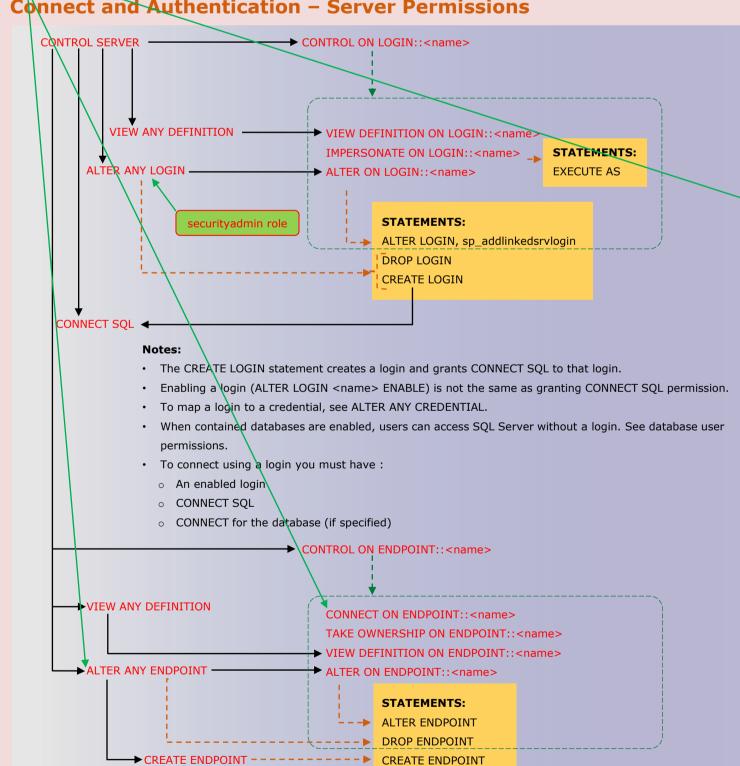
## Azure SQL Database Permissions Outside the Database Notes: Server-Level Principal Logins a Admin accounts.



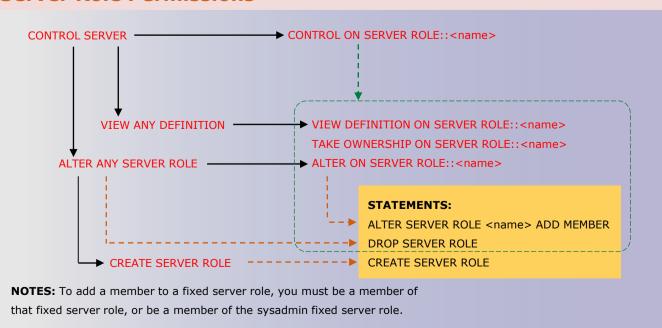
#### Server Level Permissions for SQL Server



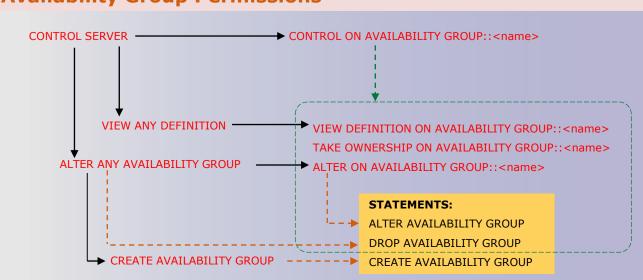
#### **Connect and Authentication – Server Permissions**



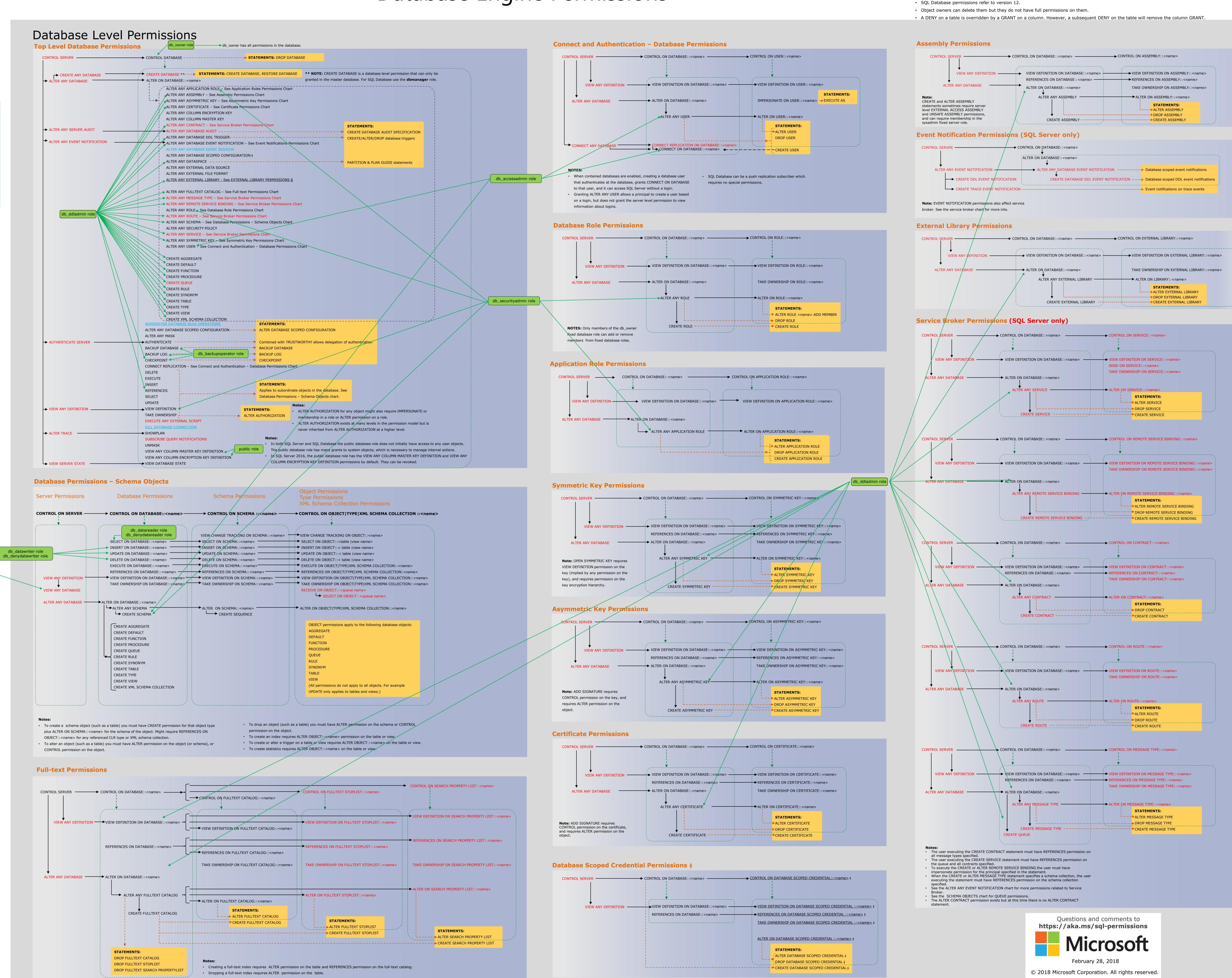
### **Server Role Permissions**



#### **Availability Group Permissions**



# Microsoft SQL Server 2017 and Azure SQL Database Database Engine Permissions



**NOTES:** • The CONTROL SERVER permission has all permissions on the instance of SQL Server or SQL Database.

However, it is sometimes possible to impersonate between roles and equivalent permissions.

• 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.)

Granting any permission on a securable allows VIEW DEFINITION on that securable. It is an implied permissions and it cannot be revoked,

The CONTROL DATABASE permission has all permissions on the database.

but it can be explicitly denied by using the DENY VIEW DEFINITION statement.