



Security



SQL Server Security

- Access is managed at two levels
 - Server-Level: Access to the server - Logins
 - Database-Level: Access to a database - Users
- Principals: Entities that can request Server resources
 - Example: Database User, Database Role
- Securable: Entity that can be secured with permissions
 - Example: Table, Column, Stored Procedure, etc
- Permission Types
 - Tables - SELECT, INSERT, UPDATE, DELETE
 - Stored Procedures - EXECUTE, VIEW DEFINITION
 - Objects (tables, databases, etc) - ALTER, DROP



CREATE LOGIN Statement

- Creates a new SQL Server login
- Example

```
CREATE LOGIN genetA  
WITH PASSWORD = 'password' MUST_CHANGE  
, DEFAULT_DATABASE = myDatabase  
, CHECK_EXPIRATION = OFF  
, CHECK_POLICY = OFF
```



CREATE USER Statement

- Adds a user to the current database.

- Example

```
CREATE LOGIN TestLogin WITH PASSWORD = 'Password';  
CREATE USER TestUser FOR LOGIN TestLogin
```

- ALTER USER

- ☐ Renames a database user

```
ALTER USER userName WITH NAME = newUser_name
```

- ☐ Example

```
ALTER USER TestUser WITH NAME = TestUserX
```

- DROP USER user_name

- ☐ Removes a user from the current database.



CREATE ROLE

- Creates a new database role in the current database
- Roles provide a better way to manage permissions.
- Syntax

```
CREATE ROLE role_name
```

- A member added to a role inherits the permissions of the role
- Example 2

```
CREATE ROLE buyers ;  
EXEC sp_addrolemember 'buyers' , 'testUser'
```



Permissions

- Assumption is no access unless granted
- Permissions are given by using GRANT, DENY, and REVOKE statements
- Server Access Level
 - Grant Access
 - Create Database
- Database Access Level
 - Query (Select)
 - Modify data
 - Create objects



GRANT Statement

- Give user privileges on an object
- Grants permissions on a securable to a principal.
- Granting a permission removes DENY or REVOKE of that permission on the specified securable

- Examples:

```
GRANT SELECT ON Customers  
TO testUser
```

```
GRANT SELECT (FirstName) ON Employee  
TO buyers
```

```
GRANT INSERT ON Orders  
TO testUser
```

```
GRANT DELETE, UPDATE ON Customers  
TO testUser
```



DENY Statement

- Denies a permission to a principal
- Remove user privileges on an object
- Prevents the principal from inheriting the permission through its group or role memberships
- Examples:
 - DENY SELECT ON Customers to testUser
 - DENY INSERT ON Customers TO testUser
 - DENY DELETE, UPDATE ON Customers TO testUser



REVOKE Statement

- Removes a previously granted or denied permission

- Examples:

REVOKE SELECT on customers to test

REVOKE INSERT ON Orders to test

REVOKE DELETE , UPDATE ON Customers to test

REVOKE SELECT (FirstName) ON Employee FROM testUser

REVOKE EXECUTE ON spUpdateEmployeeHireInfo
FROM Recruiting11;

REVOKE REFERENCES (EmployeeID) ON tblEmployee
FROM testUser