

JOBSHEET 7

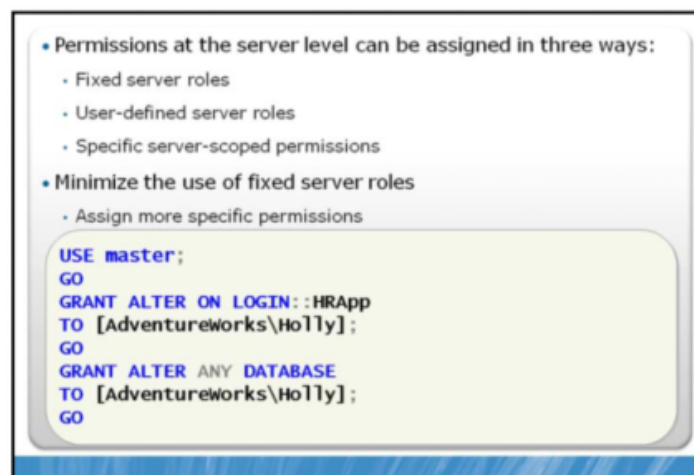
SERVER AND DATABASE ROLES

1.1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengkonfigurasi Server Roles baik fix server-roles maupun user-defined server-roles
2. Mengkonfigurasi Database Roles baik fix database-roles maupun user-defined database-roles
3. Mengenal entitas principals dan memberikan hak akses dengan statement GRANT terhadap object tertentu

1.2 Menggunakan Server Roles



Setelah login telah diautentikasi oleh server, dan dipetakan ke pengguna basis data, Anda harus menentukan permission untuk login tersebut. Permission dapat diberikan secara langsung ke login / database user. Dalam modul ini, Anda akan melihat bagaimana permission ditentukan secara langsung, dan bagaimana permission diberlakukan untuk banyak pengguna, *roles* dapat digunakan dalam hal ini.

Mengumpulkan set of permissions ke dalam satu role, akan lebih mudah dikontrol daripada memberikan GRANT akses kepada tiap users

1.2.1 Langkah-langkah Percobaan

1. Buka SQL Server Management Studio dan melakukan Connect to Server sesuai Authentication masing-masing.
2. Setelah itu buat query baru dengan cara. Klik File → New → Query with Current Connection.

3. Buatlah sebuah login untuk kita gunakan sebagai test

```
USE master;
GO

CREATE LOGIN Mod10Login WITH PASSWORD = 'Pa$$w0rd', CHECK_POLICY = OFF;
GO
```

4. Tampilkan semua fixed server roles menggunakan T-SQL. Perhatikan bahwa roles merupakan bagian server principals.

```
SELECT * FROM sys.server_principals WHERE type = 'R';
GO
```

5. Add Mod10Login to the serveradmin role dan juga bulkadmin role menggunakan T-SQL

```
ALTER SERVER ROLE bulkadmin ADD MEMBER Mod10Login;
GO

ALTER SERVER ROLE serveradmin ADD MEMBER Mod10Login;
GO
```

6. Tampilkan anggota dari server roles menggunakan T-SQL

```
SELECT r.name AS RoleName,
       p.name AS PrincipalName
FROM sys.server_role_members AS srm
INNER JOIN sys.server_principals AS r
ON srm.role_principal_id = r.principal_id
INNER JOIN sys.server_principals AS p
ON srm.member_principal_id = p.principal_id;
GO
```

7. Hapus Mod10Login dari serveradmin role menggunakan T-SQL

```
ALTER SERVER ROLE serveradmin DROP MEMBER Mod10Login;
GO
```

8. Tambahkan user-defined server-role baru menggunakan T-SQL

```
CREATE SERVER ROLE srv_documenters;
GO
```

9. Tampilkan daftar server permissions yang sudah diberi hak akses saat ini

```
SELECT p.name AS PrincipalName,
       sp.permission_name AS PermissionName,
       class_desc AS ClassDescription,
       Major_id AS MajorID
FROM sys.server_permissions AS sp
INNER JOIN sys.server_principals AS p
ON sp.grantee_principal_id = p.principal_id
ORDER BY p.name, sp.permission_name;
GO
```

1.2 Menggunakan Fix Database Roles

- Permissions at the database level can be assigned in three ways:
 - Fixed database roles
 - User-defined database roles
 - Specific database-scoped permissions
- Minimize the use of fixed database roles
 - Assign more specific permissions

```
USE Adventureworks;  
GO  
GRANT CREATE TABLE TO HRManager;  
GO  
GRANT VIEW DEFINITION TO James;  
GO
```

Ada 3 cara menentukan permissions di database level:

- Fixed database roles mirip dengan fixed server roles, terlepas dari objects & scope yang berbeda yaitu di database level
- User-defined database roles
- Database-scoped permissions

Role	Description
db_owner	Perform any configuration and maintenance activities on the DB and can drop it
db_securityadmin	Modify role membership and manage permissions
db_accessadmin	Add or remove access to the DB for logins
db_backupoperator	Back up the DB
db_ddladmin	Run any DDL command in the DB
db_datawriter	Add, delete, or change data in all user tables
db_datareader	Read all data from all user tables
db_denydatawriter	Cannot add, delete, or change data in user tables
db_denydatareader	Cannot read any data in user tables

Ada beberapa fixed database roles yang sudah ada dan bias kita berikan roles tersebut ke user

1.2.1 Langkah-langkah Percobaan

1. Gunakan MarketDev database
2. Tampilkan database roles menggunakan T-SQL. Perhatikan bahwa roles dianggap sebagai database principals. Perhatikan juga bahwa db_owner ditampilkan di dalam list

```
USE MarketDev;  
GO  
  
SELECT * FROM sys.database_principals WHERE type = 'R';  
GO
```

3. Tambahkan Mod10Login user ke db_backupoperator role

```
ALTER ROLE db_backupoperator ADD MEMBER Mod10Login;  
GO
```

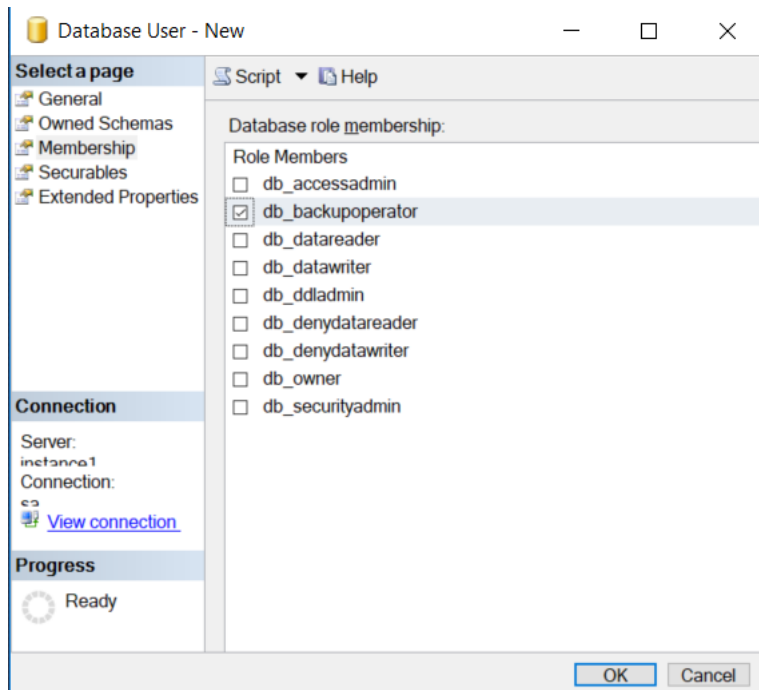
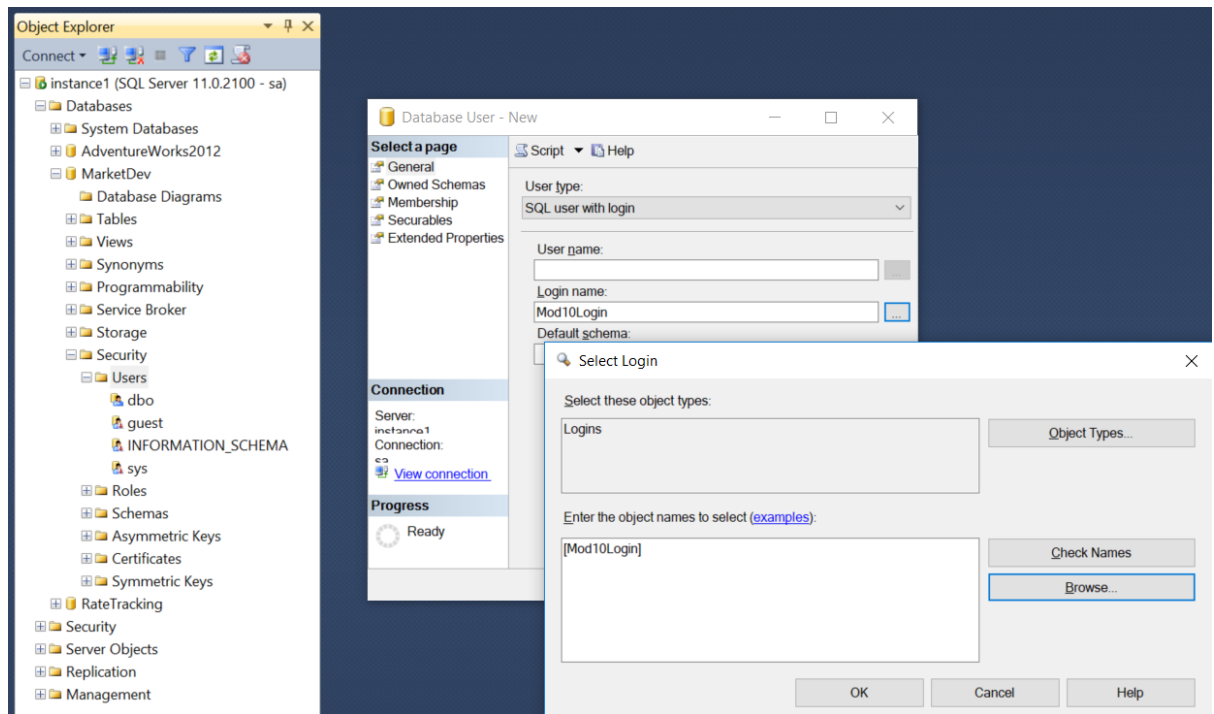
4. Tampilkan members dari database roles

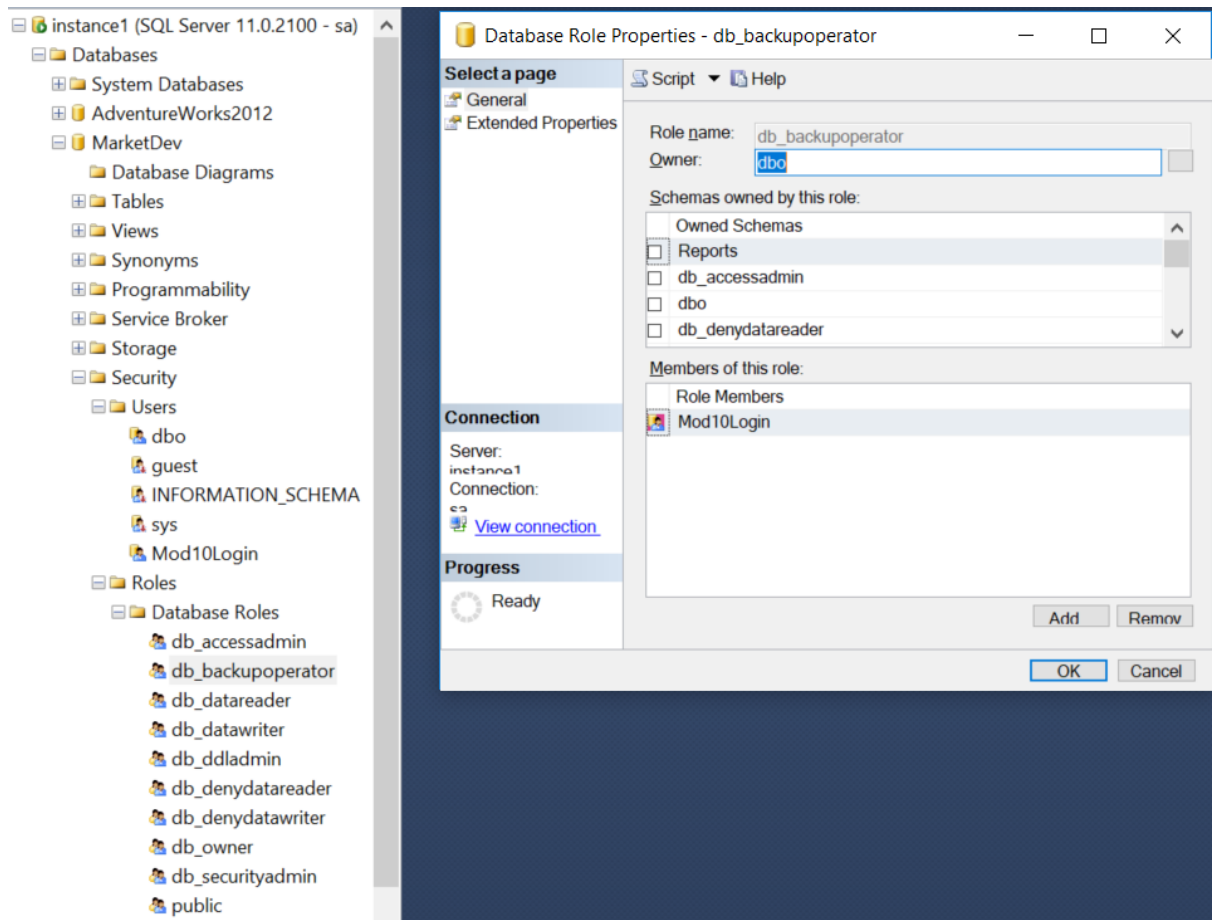
```
SELECT r.name AS RoleName,  
       p.name AS PrincipalName  
FROM sys.database_role_members AS drm  
INNER JOIN sys.database_principals AS r  
ON drm.role_principal_id = r.principal_id  
INNER JOIN sys.database_principals AS p  
ON drm.member_principal_id = p.principal_id;  
GO
```

5. Untuk menghapus Mod10Login dari db_backupoperator role gunakan query

```
ALTER ROLE db_backupoperator DROP MEMBER Mod10Login;  
GO
```

6. Lakukan yang hal yang sama dengan menggunakan GUI





1.3 Menggunakan User-Defined Database Roles

- Database roles can be created, modified, and dropped
 - CREATE ROLE statement to create
 - Roles have owners
 - Permissions are granted to role
 - Role permissions are inherited by role members

```

USE MarketDev;
GO

CREATE ROLE MarketingReaders
AUTHORIZATION dbo;
GO

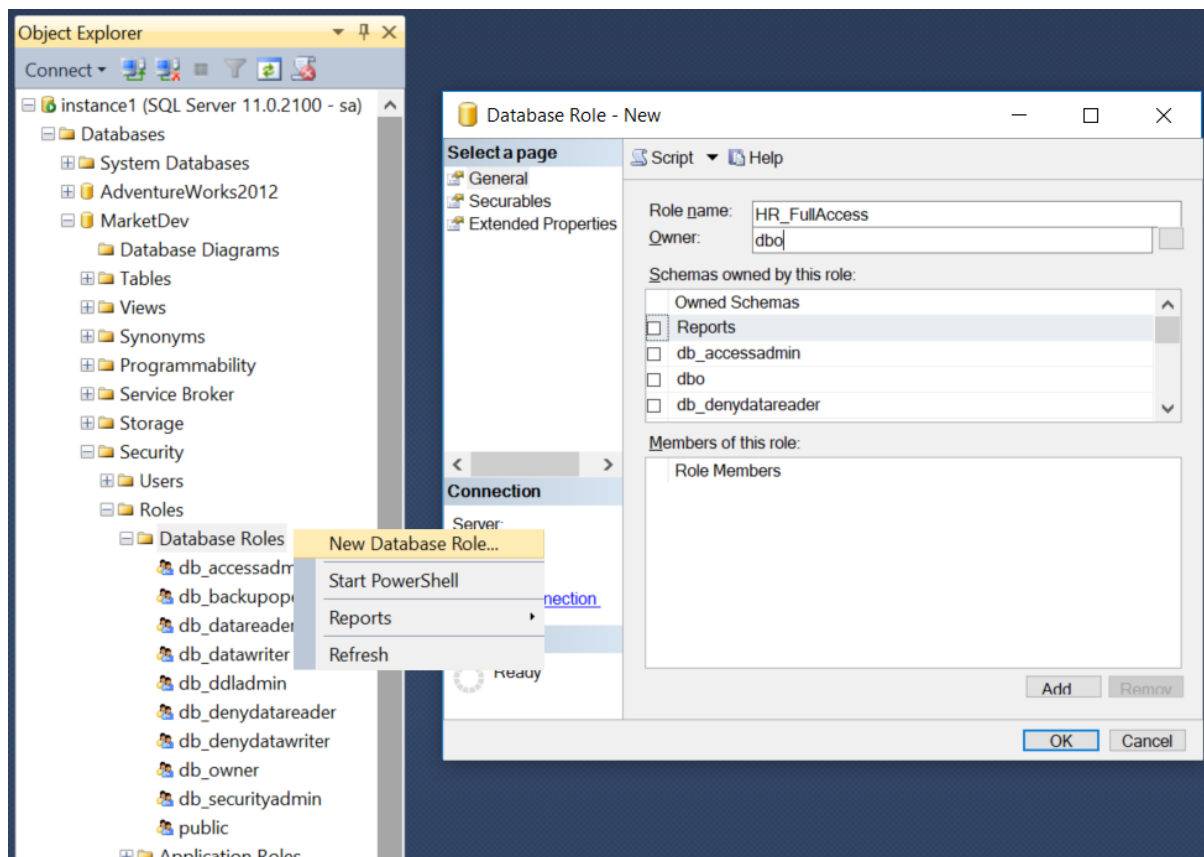
GRANT SELECT ON SCHEMA::Marketing
TO MarketingReaders;
GO

```

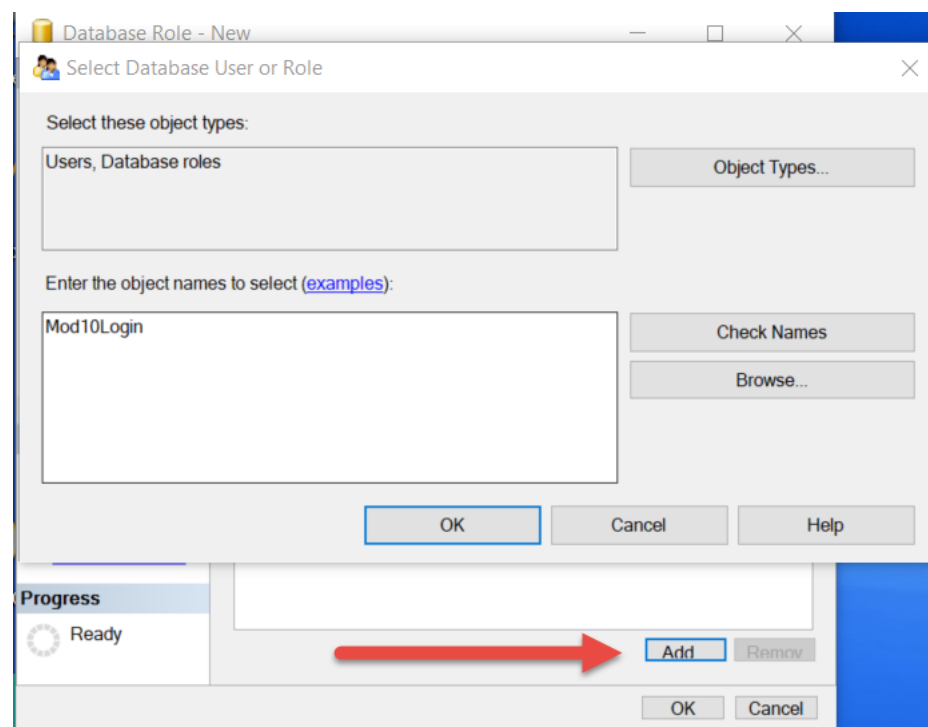
User-defined database roles bisa dibuat dengan menggunakan CREATE ROLE statement

1.3.1 Langkah-langkah Percobaan

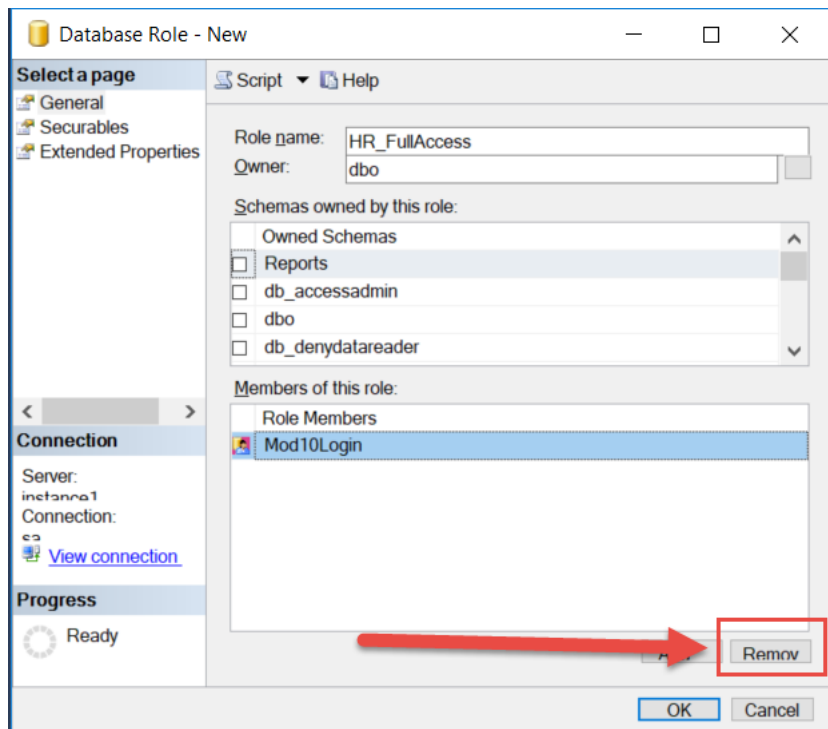
1. Untuk menambahkan User-Defined database roles menggunakan GUI lakukan



Untuk menambahkan user ke dalam Role, pilih Add



Hapus lagi User Mod10Login dari database-roles dengan pilih Remove



2. Gunakan MarketDev database. Buatlah role HR_LimitedAccess

```
USE MarketDev;  
GO
```

```
CREATE ROLE HR_LimitedAccess AUTHORIZATION dbo;  
GO
```

3. Tambahkan user Mod10Login ke role tersebut

```
ALTER ROLE HR_LimitedAccess ADD MEMBER Mod10Login;  
GO
```

4. Tampilkan list dari database roles menggunakan T-SQL

```
SELECT r.name AS RoleName,  
       p.name AS PrincipalName  
FROM sys.database_role_members AS drm  
INNER JOIN sys.database_principals AS r  
ON drm.role_principal_id = r.principal_id  
INNER JOIN sys.database_principals AS p  
ON drm.member_principal_id = p.principal_id;  
GO
```

5. Untuk menghapus kembali user dari Role dan juga menghapus Role, bisa menggunakan T-SQL berikut

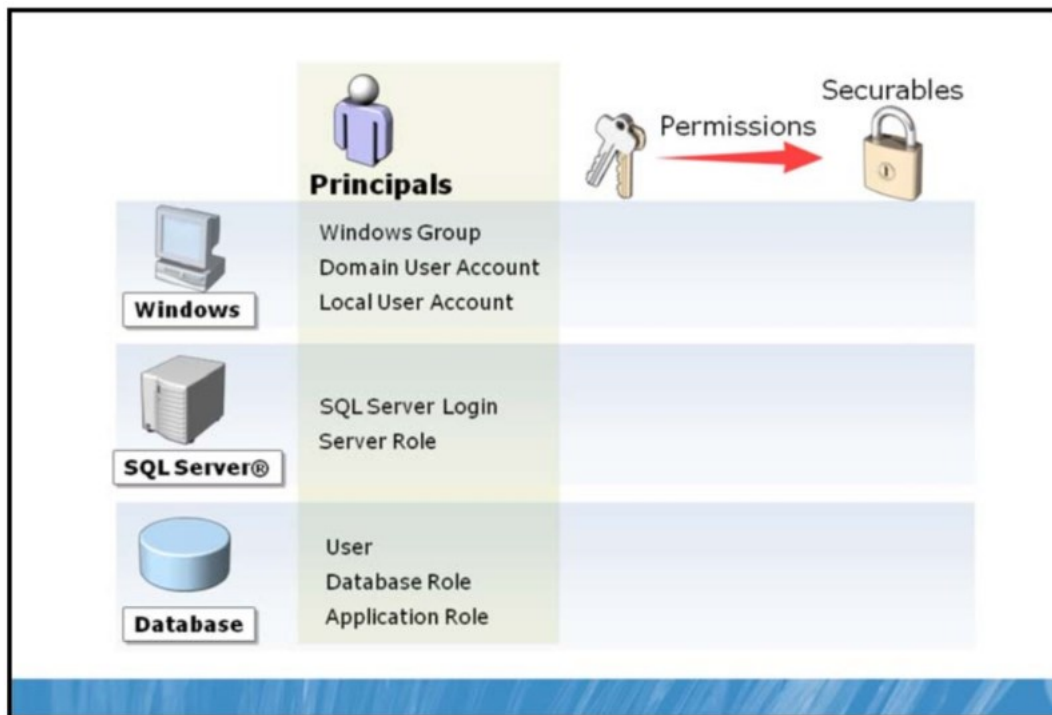

```

ALTER ROLE HR_LimitedAccess DROP MEMBER Mod10Login;
GO

DROP ROLE HR_FullAccess;
DROP ROLE HR_LimitedAccess;
GO

```

1.4 Memberikan Authorisasi User-Access ke Objects



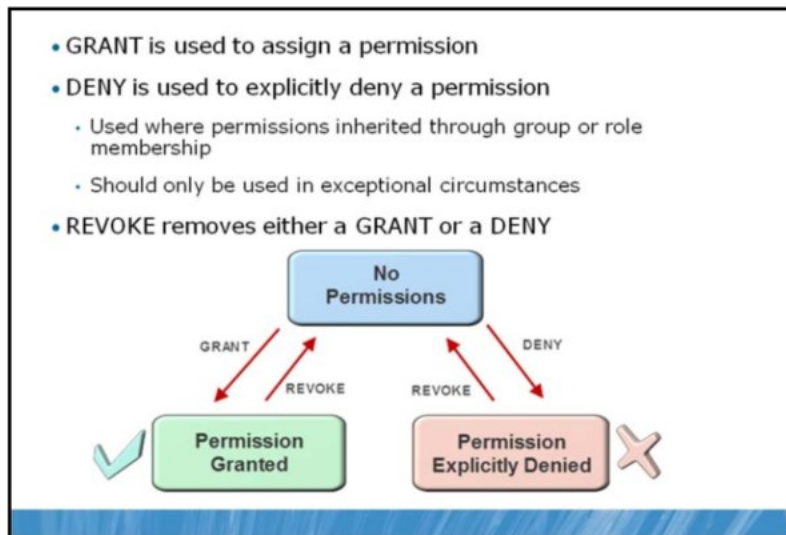
Dalam Authorisasi dikenal istilah Principals yaitu entiti yang bisa melakukan request dan bisa diberikan / granted ke SQL Server resources

Dalam Windows® level, principals termasuk users dan groups

Dalam SQL Server level, fixed dan user-defined server roles adalah juga merupakan principal yang berisi principals lain

Resource yang bisa diberikan control-access meliputi, scope:

- Server
- Database
- Schema



Permission bisa kita control menggunakan perintah GRANT, DENY dan REVOKE

1.4.1 Langkah-langkah Percobaan

1. Gunakan MarketDev database. Tampilkan daftar semua server principals

```
USE MarketDev;
GO

SELECT * FROM sys.server_principals;
GO
```

2. Tampilkan daftar semua database principals yang bisa kita beri hak akses

```
SELECT * FROM sys.database_principals;
GO
```

3. Tambahkan user Mod11Login. Dan berikan hak SELECT di table Marketing.Product,

```
CREATE LOGIN Mod11Login WITH PASSWORD = 'Pa$$w0rd', CHECK_POLICY = OFF;
GO

USE MarketDev;
GO

CREATE USER Mod11Login FOR LOGIN Mod11Login;
GO

GRANT SELECT ON Marketing.Product TO Mod11Login;
GO
```

4. Rubah execution context menggunakan user Mod11Login dan lakukan SELECT dari Table Marketing.Product & table Marketing.PostalCode. Catat bila ada kesalahan dan laporkan analisa anda.

```
EXECUTE AS USER = 'Mod11Login';
GO

SELECT * FROM Marketing.Product;
GO

SELECT * FROM Marketing.PostalCode;
GO
```

5. Tambahkan GRANT SELECT ke column StateCode dan CityName, dengan menggunakan context Administrator

```
REVERT;
GO

GRANT SELECT ON Marketing.PostalCode
(StateCode, CityName)
TO Mod11Login;
GO
```

6. Cobalah untuk query SELECT column StateCode dan CityName dan juga coba query SELECT all columns dari Marketing.PostalCode. Catat bila ada kesalahan dan laporkan analisa anda.

```
EXECUTE AS USER = 'Mod11Login';
GO

SELECT DISTINCT StateCode, CityName FROM Marketing.PostalCode;
GO

SELECT * FROM Marketing.PostalCode;
GO
```

1.5 Latihan Praktikum

1. Lakukan praktikum di atas dan buat laporan proses dan hasil di screen shot disertai penjelasannya.