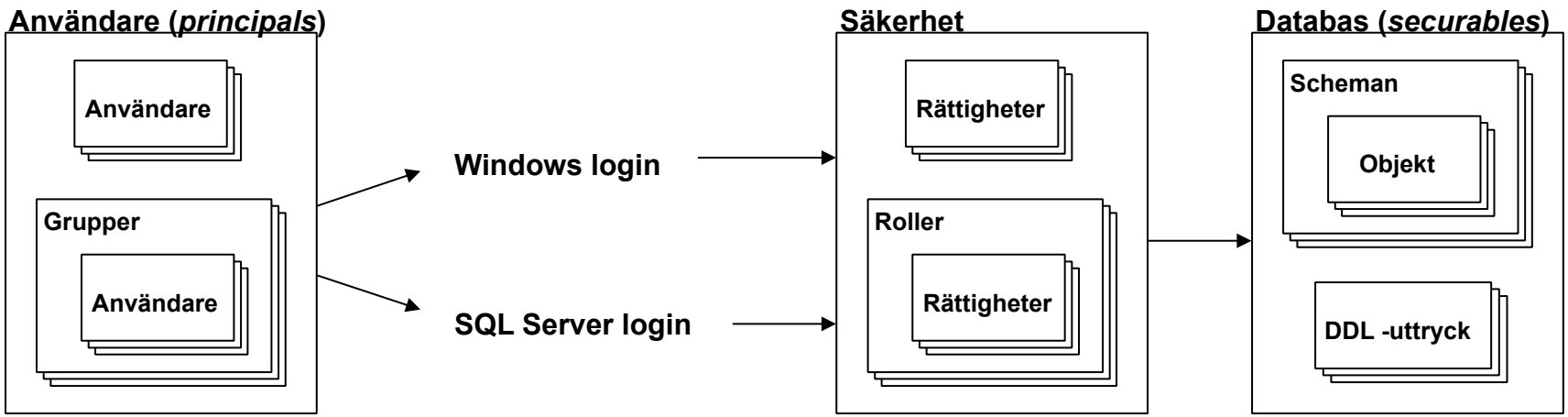
Säkerhet

1. Användare och login
2. Rättigheter
3. Roller
4. Applikationsroller

Kap. 18

1. Användare och login

- ***Användare*** **måste ha** ***login*** **till databasen**
- **SQL Server har två typer av** ***login*****:**
  - ***Windows login*** **- samma login som för** ***Windows PC -klienten***
  - ***SQL Server login*** **- specifikt login för** ***databasservern***
- **För** ***användare*** **så anges** ***rättigheter*** **för** ***objekt och uttryck*** **(*objekträttigheter*) och för** ***databasen*** **(*databasrättigheter*)**
- **Rättigheter kan samlas i** ***roller*** **(vilka tilldelas** ***användare*** **och** ***grupper*****)**



1.1. Skapa login

- **Ett *login* skapas genom `CREATE LOGIN` (både *SQL Server login* och *Windows login*)**
- **För *SQL Server login* så bör ett *”stark” lösenord anges***
  - **`MUST_CHANGE` tvingar användaren att ändra lösenordet vid första inloggningen**
- ***Default databas* och *språk* angives genom `DEFAULT_DATABASE` respektive `DEFAULT_LANGUAGE`**
- **Genom `CHECK_EXPIRATION` så måste att lösenordet ändras *regelbundet***
- **Genom `CHECK_POLICY` så måste ett *”starka” lösenord* används**

Syntax

```
CREATE LOGIN Login FROM WINDOWS                                  -- Windows login
   [WITH [DEFAULT_DATABASE = Databas]
        [,DEFAULT_LANGUAGE = Språk]]

CREATE LOGIN Login WITH PASSWORD = 'Lösenord' [MUST_CHANGE]  -- SQL Server login
   [WITH [DEFAULT_DATABASE = Databas]
        [,DEFAULT_LANGUAGE = Språk]
        [,CHECK_EXPIRATION = {ON|OFF}]
        [,CHECK_POLICY = {ON|OFF}]]
```

Notera

- **Vanligtvis används ett *verktyg* (t.ex. *Management Studio*) för att skapa login.**

1.2. Ändra och ta bort login

- **Login *tas bort genom* `DROP LOGIN`**
- **Genom `ALTER LOGIN` så kan *loginnamn*, *lösenord* och *inställningar ändras***
- ***Login* kan även *inaktiveras* (`DISABLE`) och *aktiveras* (`ENABLE`)**

Syntax

- `DROP LOGIN Login` -- Tar bort ett login
- `ALTER LOGIN Login {ENABLE|DISABLE}|[WITH [NAME = NyttLogin] ... ]` -- Ändrar Windows login
- `ALTER LOGIN Login {ENABLE|DISABLE}|[WITH [PASSWORD = 'NyttLösen' [OLD_PASSWORD = 'GammaltLösen'] [MUST_CHANGE]] [,NAME = NyttLogin] ... ]` -- Ändrar SQL Server login

1.3. Exempel login

- ***Skapar ett login med samma login som för en Windows -klient:***

```
CREATE LOGIN [Handelshögskolan\MathiasHatakka] FROM WINDOWS
```

- ***Tar bort det nyskapade loginnamnet:***

```
DROP LOGIN [Handelshögskolan\MathiasHatakka]
```

- ***Skapar ett SQL Server login istället (angivet ”svagt” lösenord):***

```
CREATE LOGIN MathiasHatakka WITH PASSWORD = 'Password'
      DEFAULT_DATABASE = Ekonomi
```

- ***Ändrar till ett korrekt SQL Server login med ”starkt” lösenord:***

```
ALTER LOGIN MathiasHatakka
      WITH PASSWORD = 'mha1234_Tmp'
      MUST_CHANGE,
      NAME = mha
```

- ***Inaktiverar ett login:***

```
ALTER LOGIN mha DISABLE
```

Ekonomi

Utbetalningar

UtlD	AnställdID*	Belopp	Månad
1	5	18700	Juli
2	5	1600	Juli
4	10	21200	Augusti
5	7	NULL	Juli
7	2	22500	Juli
8	7	19900	September
10	2	22500	Augusti
11	5	5600	September

1.4. Användare

- **Genom `CREATE USER` så skapas *användare för ett login***
- **För samma *användarnamn* som *loginnamn* så används `FOR LOGIN` -uttrycket**
- **Genom `ALTER USER` så ändras *användarnamnet***
- ***Defaultschema* kan anges för *användare* (istället för databasens *defaultschema:* `dbo`)**
- ***Användare* tas bort genom `DROP USER`**

Syntax

```
CREATE USER Användare                    -- Skapar en användare
   [{FOR|FROM} LOGIN Login]
   [WITH DEFAULT_SCHEMA = Schema]

ALTER USER Användare WITH                -- Ändrar en användare
   [NAME = NyttAnvändareNamn]
   [, DEFAULT_SCHEMA = Schema]

DROP USER Användare                      -- Tar bort en användare
```

Notera

- ***Användare* skapas för den aktiva databasen!**

1.5. Scheman

- ***Scheman* skapas genom `CREATE SCHEMA`**
  - ***Tabeller* och *vyer* kan skapas i samband med ett scheman**
- **Objekt tilldelas ett schema genom att inkludera *schemanamet* framför *objektnamnet***
- **Objekt *flyttas* mellan scheman genom `ALTER SCHEMA`**
  - ***Det går inte att flytta objekt som andra objekt (vyer och funktioner) är beroende av!***
  - **Vid flytt av objekt så försvinner alla *rättigheter***
- ***Scheman* tas bort genom `DROP SCHEMA` (måste dock vara *tomt*)**

Syntax

- ```
  CREATE SCHEMA Schema [AUTHORIZATION Ägare]      -- Skapar ett schema
     [TabellDefinition] ...
     [VyDefinition] ...
     [GrantUttryck] ...
     [RevokeUttryck] ...
     [DenyUttryck] ...
  ```
- ```
  ALTER SCHEMA Schema TRANSFER Securable          -- Ändrar ett schema
  ```
- ```
  DROP SCHEMA Schema                              -- Tar bort ett schema
  ```
  ```

1.6. Exempel för användare och scheman

- ***Skapa en användare med samman användarnamn som loginnamn:***

```
CREATE USER mha FOR LOGIN mha
```

- ***Ändrar användarnamnet:***

```
ALTER USER mha WITH NAME = Mathias
```

- ***Skapar ett schema för alla ”Ekonomi” -objekt i databasen:***

```
CREATE SCHEMA Ekonomiskt
```

- ***Flyttar en tabell från defaultschemat till det nyskapade schemat:***

```
ALTER SCHEMA Ekonomiskt
      TRANSFER dbo.Utbetalningar
```

- ***Ändrar en användara och sätter ett defaultschema:***

```
ALTER USER Mathias
      WITH DEFAULT_SCHEMA = Ekonomiskt
```

Ekonomi

Utbetalningar

UtlD	AnställdID*	Belopp	Månad
1	5	18700	Juli
2	5	1600	Juli
4	10	21200	Augusti
5	7	NULL	Juli
7	2	22500	Juli
8	7	19900	September
10	2	22500	Augusti
11	5	5600	September

2. Rättigheter

- ***Rättigheter* sättas för *tabeller*, *vyer*, *lagrade procedurer* och *funktioner***
  - ***Beviljas* genom `GRANT`**
  - ***Upphävs* genom `REVOKE`**
- **För *tabeller*, *vyer* och *funktioner med tabellretur* kan `SELECT`, `UPDATE` och `REFERENCES` -rättigheter sättas på attributnivå (kolumn).**
- **Rättigheter *beviljas till* (`TO`) en *principal* (*användare* eller *roller*) och *upphävs från* (`FROM`)**
- **Genom `WITH GRANT OPTION` så får en *principal rättigheter att utfärda rättigheter***
- ***Upphävs* rättigheter med `CASCADE` så upphävs även en *principals* alla satta rättigheter (dvs., alla rättigheter som denna specifika användare satt på objektet)**

Syntax

- ```
  GRANT {ALL|Rättigheter [, ...]}              -- Beviljar rättigheter
  ON [Schema.]Objekt [(Attribut [, ...])]
  TO Principal                                 -- Till användare eller roll
  [WITH GRANT OPTION]
  ```
- ```
  REVOKE [GRANT OPTION FOR] {ALL|Rättigheter [, ...]} -- Upphäver rättigheter
  ON [Schema.]Objekt [(Attribut [, ...])]
  FROM Principal                                       -- För användare eller roll
  [CASCADE]
  ```

2.1. Typer av rättigheter och exempel

Typ	Gäller för
SELECT -frågor	Tabeller, vyer och funktioner med tabellretur
Funktionsfrågor	Tabeller, vyer och funktioner med tabellretur
EXECUTE	Lagrade procedurer och funktioner med singelvärdesretur
REFERNECES	Tabeller, vyer och funktioner
ALL	Tabeller, vyer, lagrad procedurer och funktioner

Exempel

- ***Beviljar rättigheter för att utföra alla typer av funktionsfrågor:***

```
GRANT INSERT, UPDATE, DELETE
   ON Ekonomiskt.Utbetalningar
   TO Mathias
```

- ***Upphäver alla rättigheter:***

```
REVOKE ALL
   ON Ekonomiskt.Utbetalningar
   FROM Mathias
```

2.2. Rättigheter för scheman

- **Även rättigheter för *scheman beviljas* med `GRANT` och *upphävs* med `REVOKE`**
- **`ON` -uttrycket anger *klassnamnet* (`SCHEMA`) och *scope behörigheten* (`::`)**

Syntax

- ```
  GRANT Rättigheter [, ...]                    -- Beviljar rättigheter för scheman
  ON SCHEMA :: Schema
  TO Principal                                 -- Till användare eller roll
  [WITH GRANT OPTION]
  ```
- ```
  REVOKE [GRANT OPTION FOR] Rättigheter [, ...]   -- Upphäver rättigheter
  ON SCHEMA :: Schema
  FROM Principal                                  -- För användare eller roll
  [CASCADE]
  ```

Exempel

- ***Beviljar rättigheter för att ta ändra (ALTER) ett schema:***

```
GRANT ALTER
   ON SCHEMA :: Ekonomiskt
   TO Mathias
```

2.3. Rättigheter för DDL -uttryck

- **`GRANT` och `REVOKE` kan även anges för DDL -uttryck**
  - **Inget `ON` -uttrycket används!**

Syntax

- **`GRANT`** {**`ALL`**|rättigheter [, ...]} -- Beviljar rättigheter för DDL -uttryck
  `TO` *principal* -- Till användare eller grupp
  [**`WITH GRANT OPTION`**]
- **`REVOKE`** {**`ALL`**|rättigheter [, ...]} -- Upphäver rättigheter
  `FROM` *principal* -- För användare eller grupp
  [**`CASCADE`**]

Exempel

- ***Beviljar rättigheter för att lägga till tabeller och vyer:***

```
GRANT CREATE TABLE, CREATE VIEW
   TO Mathias
```

- ***Upphäver rättigheterna för alla DDL -uttryck:***

```
REVOKE ALL
   FROM Mathias
```

3. Roller

- **En *roll* är en samling *rättigheter* som kan tilldelas till en grupp av användare**
- **Det finns tre typer av *roller*:**
  - ***Fixerade serverroller* - definierar roller på *servernivå***
  - ***Fixerade databasroller* - definierar roller på *databasnivå***
  - ***Användaredefinierade roller (server och databas)* - sätter rollerna för användare**

Syntax

```
ALTER SERVER ROLE role_name
{
        ADD MEMBER server_principal
        DROP MEMBER server_principal
        WITH NAME = new_role_name
}
```

```
ALTER SERVER ROLE dbcreator ADD MEMBER mha;
ALTER SERVER ROLE dbcreator DROP MEMBER mha;
```

Fixerade serverroller

- **sysadmin** -- Kan utföra alla typer av operationer i en server.
- **securityadmin** -- Kan lägga till rättigheter, login och användare.
- **dbcreator** -- Kan skapa, ändra och ta bort databaser.

3.1. Databasroller

- ***Fixerade databasroller*** **tilldelas automatiskt till nyskapade databaser**
- ***Medlemmar*** **läggs till och tas bort från de** ***fixerade databasrollerna*** **(dock kan inte** ***rollerna*** **tas bort)**

Syntax

```
ALTER ROLE role_name
{
        ADD MEMBER server_principal
        DROP MEMBER server_principal
        WITH NAME = new_role_name
}
```

```
ALTER ROLE db_datawriter ADD MEMBER mha;
ALTER ROLE db_datawriter DROP MEMBER mha;
```

Fixerade databasroller (ett urval)

- **db_owner** -- Har alla rättigheter.
- **db_accessadmin** -- Kan skapa och ta bort login.
- **db_securityadmin** -- Kan hantera rättigheter och roller.
- **db_ddladmin** -- Kan utfärda DDL -uttryck.
- **db_datawriter** -- Kan lägga till, ändra och ta bort data från databasen.
- **db_datareader** -- Kan endast hämta data från databasen.

3.2. Användaredefinierade roller

- ***Användaredefinierad roller*** **skapas genom** **`CREATE ROLE`**
  - **Skapas med ett *unikt namn* och en *ägare***
- ***Användaredefinierad roller*** **tas bort genom** **`DROP ROLE`** **(fungerar ej för *server-* och *databasroller*)**
  - **Alla *medlemmar* av rollen måste tas bort innan rollen kan tas bort!**

Syntax:

```
CREATE ROLE role_name [AUTHORIZATION Ägare]
DROP ROLE role_name
```

Exempel:

```
CREATE ROLE Utbetalningar_admin;

GRANT INSERT, UPDATE
ON Utbetalningar
TO Utbetalningar_admin;

ALTER ROLE Utbetalningar_admin ADD MEMBER mha;
ALTER ROLE Utbetalningar_admin ADD MEMBER tpmc;

ALTER ROLE db_datareader ADD MEMBER
```

3.2. Användaredefinierade roller

Exempel:

```
CREATE ROLE Utbetalningar_admin;

GRANT INSERT, UPDATE, DELETE
ON Utbetalningar
TO Utbetalningar_admin;

ALTER ROLE Utbetalningar_admin ADD MEMBER mha;
ALTER ROLE Utbetalningar_admin ADD MEMBER tpmc;

ALTER ROLE db_datareader ADD MEMBER Utbetalningar_admin;

---
ALTER ROLE Utbetalningar_admin DROP MEMBER mha;
ALTER ROLE Utbetalningar_admin DROP MEMBER tpmc;
DROP ROLE Utbetalningar_admin;
```

3.4. Information om databasroller

- **Information om *roller* återfås genom *systemprocedurerna* `sp_HelpSrvRole` och `sp_HelpRole`**
- **Information om *medlemmarna av en roll* återfås *genom systemproceduren* `sp_HelpRoleMember` och `sp_HelpRoleMember`**

Syntax

- **`sp_HelpRole`** `[[@rolename =] 'Databasroll']`
- **`sp_HelpRoleMember`** `[[@rolename =] 'Databasroll']`

Exempel

- ***Vilka medlemmar finns det (fanns det innan den togs bort) i rollen GruppUtbetalningar:***

```
EXEC sp_HelpRoleMember GruppUtbetalningar
```

- ***Alla roller i den aktiva databasen:***

```
EXEC sp_HelpRole
```

3.5. Neka rättigheter

- **Då rättigheter kan tilldelade både genom *användarnamnet* och genom *roller* så kan det uppstå *konflikter***
  - **Som en lösning på detta finns `DENY` -uttrycket**
- **Rättighet som är `DENY` för *användare* kan inte bli `GRANTED` genom en *roll***

Syntax

- ```
  DENY {ALL|Rättigheter [, ...]}                -- Neka rättigheter till objekt
  ON [Schema.]Objekt [(Attribut [, ...])]
  TO Principal                                  -- För användare
  [CASCADE]
  ```
- ```
  DENY Rättigheter [, ...]                      -- Neka rättigheter till scheman
  ON SCHEMA :: Schema
  TO Principal                                  -- För användare
  [CASCADE]
  ```
- ```
  DENY {ALL|Rättigheter [, ...]}      -- Neka rättigheter till DDL -uttryck
  TO Principal                        -- För användare
  [CASCADE]
  ```
  ```

3.6. Exempel

- ***Säkerställer att en användare inte får ”högre rättigheter” genom en roll:***

```
CREATE ROLE Utbetalningar_admin;


      GRANT INSERT, UPDATE, DELETE
      ON Utbetalningar
      TO Utbetalningar_admin;


      ALTER ROLE Utbetalningar_admin ADD MEMBER mha;
      ALTER ROLE Utbetalningar_admin ADD MEMBER tpmc;



      DENY DELETE
      ON Utbetalningar
      TO tpmc;
```

4. Applikationsroller

- **Genom *applikationsroller* kan rättigheter tilldelas *användare* som gäller när användaren ansluter mot databasen genom en *applikation***
  - ***Ex.:* anslutning genom ADO.NET i en C# -applikation**
- **Skapas genom `CREATE APPLICATION ROLE`**
- **Tas bort genom `DROP APPLICATION ROLE`**

Syntax

- ```
  CREATE APPLICATION ROLE role_name WITH PASSWORD 'Lösenord'
     [, DEFAULT_SCHEMA = Schema]
  ```
- ```
  DROP APPLICATION ROLE role_name
  ```

Exempel

- ***Skapar en applikationsroll genom vilken det endast går att hämta data:***

```
CREATE APPLICATION ROLE AppUtbetalningar WITH PASSWORD 'ekonomi1337',
      DEFAULT_SCHEMA = Ekonomiskt

-- Sätter rättigheter till applikationsrollen
   GRANT SELECT
   ON Utbetalningar
   TO AppUtbetalningar
```

4.1. Använda applikationsroller

- ***Applikationsroller*** **aktiveras med** ***systemproceduren*** **`sp_SetAppRole`**
  - **Inaktiveras med `sp_UnsetAppRole` (exekveras automatiskt då kopplingen stängs ner)**
- **Vid aktivering av** ***applikationsroller*** **ignoreras de ”normala” rättigheterna**
- **För att explicit inaktivera en** ***applikationsroll*** **så måste rollen ha aktiverats med en** ***cookie***

Syntax

- **`sp_SetAppRole`** `[@rolename =] 'Databasroll',`
  `[@password =] 'Lösenord'`
  `[, [@fCreateCookie =] {True|False}}]`
  `[, [@cookie =] @cookie OUTPUT]`
- **`sp_UnsetAppRole`** `@cookie`

Exempel

- ***Använder applikationsroll för att hämta data (förutsätt att användaren inte har SELECT rättigheter):***

```
SELECT * FROM Utbetalningar;                          -- Fungerar inte!
EXEC sp_SetAppRole AppUtbetalningar, ekonomi1337;     -- Aktiverar rollen
SELECT * FROM Utbetalningar;                          -- Fungerar!
```