Innehåll Security

- SQL Injektions
- Säkerhetssystemet
- Schema
- Login
- Användare
- Roller
- User
- Applikationsanvändare AppUser
- Backup av databas
- Restore / Recovery av databas
- Flytta/Kopiera en databas, Detach/Attach
- Scripta en databas
- Maintenance Plan

Chapter 4 och 7

Beginning SQL Server 2008 for Developers

SQL Injektions (1 av 5)

Så här kan det se ut vid normal inloggning från ett formulär:

En inloggning via ett formulär som ser ut som ovan genererar en SELECT sats enligt nedan för att kontrollera om den aktuella användaren med lösenord har tillgång till applikationen (eg finns):

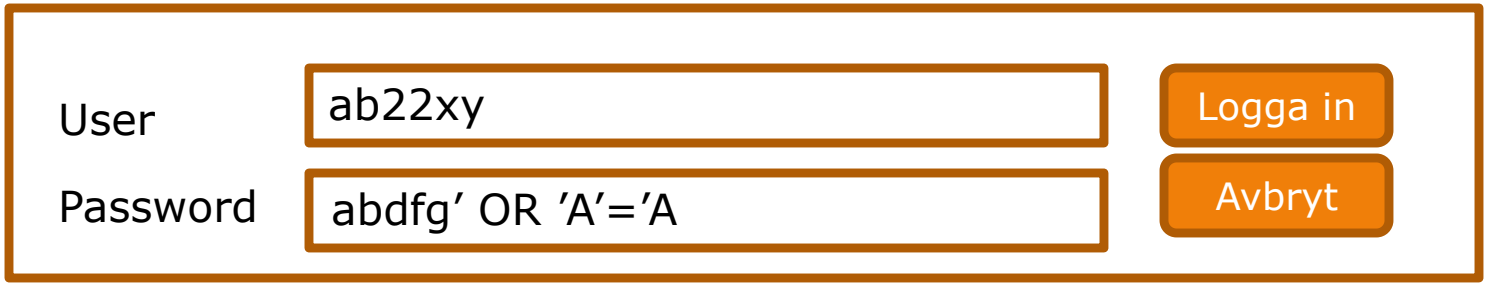
	uid	usr	pwd
▶	1	ab22xy	apQ123a
	2	aa22yx	QWn981L

```
SELECT Usr, pwd
FROM [User]
WHERE Usr='ab22xy' AND pwd='apQ123a'
```

Eftersom allt stämmer så bör inloggningen kunna ske utan problem?

SQL Injektions (2 av 5)

Vad händer här då? Nu skickar användaren vid inloggningstillfället med lite mer än det är tänkt. Användaren skickar in skadlig SQL kod:



Hur ser då SQL satsen ut som skapas utifrån den inmatning användaren gör:

	uid	usr	pwd
▶	1	ab22xy	apQ123a
	2	aa22yx	QWn981L

Injekted

```
SELECT Usr, pwd
FROM [User]
WHERE Usr='ab22xy' AND pwd='abdfg' OR 'A'='A'
```

Testa

Kommer användaren att bli inloggad?

I SQL-satsen

SQL Injektions (3 av 5)

Om användaren istället skickar in följande – vad händer då?

Hur ser då SQL satsen ut som skapas utifrån den inmatning användaren gör:

	uid	usr	pwd
▶	1	ab22xy	apQ123a
	2	aa22yx	QWn981L

Injekted

```
SELECT Usr, pwd
FROM [User]
WHERE Usr='ab22xy' AND pwd='abdfg'; DROP TABLE Kund;--'
```

I SQL-satsen

Kommer användaren att bli inloggad?
Händer det något annat?

SQL Injektions (4 av 5)

Åtgärd:

- ✓ Ersätt alla apostrofer ’ (strängavslut) med exempelvis citattecken ” innan du skapar SQL- satsen.

```
1 Ursprunget är:      abdfg’; DROP TABLE Kund;--’
```

```
2. Som då blir:       abdfg”; DROP TABLE Kund;--
```

I ASP.NET finns inbyggt skydd mod SQL injektions om du använder parameteriserade frågor eller lagrade procedurer.

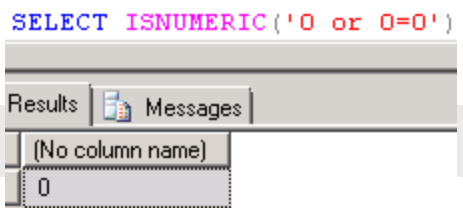
```
3. Och SQL satsen blir:

SELECT Usr, pwd
FROM User
WHERE Usr=’ab22xy’ AND pwd=’abdfg”; DROP TABLE Kund;--’
```

@pwd blir slutligen endast **en** sträng utan SQL funktionalitet.

- ✓ Kontrollera att numeriska data är enbart numeriska. ISNUMERIC(@pwd) True = 1, False =0

```
WHERE Usr=’ab22xy’ AND pwd=0 OR 0=0
```



SQL Injektions (5 av 5)

BOOKS ON LINE

Home | Library | Learn | Downloads | Support | Community | Sign in | United Kingdom - English

Search MSDN with Bing

- MSDN Library
- Servers and Enterprise Development
- SQL Server
- SQL Server 2008 R2
- Product Documentation
- SQL Server 2008 R2 Books Online
- Database Engine
- Security and Protection
  - Threat and Vulnerability Mitigation (Databa
  - Threat and Vulnerability Matrix (Database
  - SQL Injection

Community Content

Prevent from SQL Injection Attac...
Use always stored procedure inst...

History of SQL Injection...
After 10 years, sql injection is...

More...

SQL Injection

msdn

SQL Server 2008 R2 | Other Versions | 12 out of 25 rated this helpful | Rate this topic

SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.

The primary form of SQL injection consists of direct insertion of code into user-input variables that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed.

The injection process works by prematurely terminating a text string and appending a new command. Because the inserted command may have additional strings appended to it before it is executed, the malefactor terminates the injected string with a comment mark "--". Subsequent text is ignored at execution time.

The following script shows a simple SQL injection. The script builds an SQL query by concatenating hard-coded strings together with a string entered by the user:

```
var Shipcity;
ShipCity = Request.form ("ShipCity");
var sql = "select * from OrdersTable where ShipCity = '" + ShipCity + "'";
```

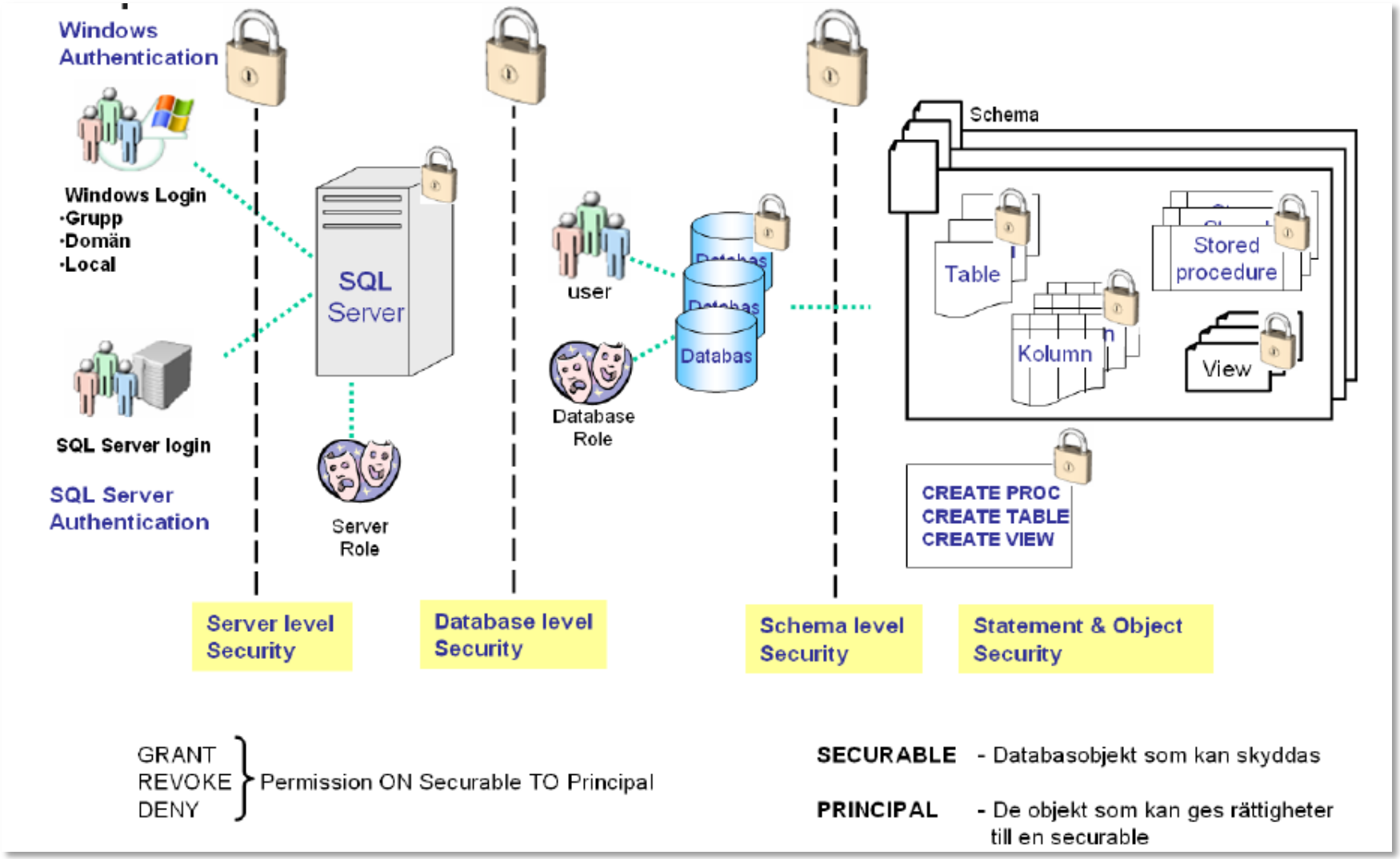
The user is prompted to enter the name of a city. If she enters `Redmond`, the query assembled by the script looks similar to the following:

```
SELECT * FROM OrdersTable WHERE ShipCity = 'Redmond'
```

However, assume that the user enters the following:

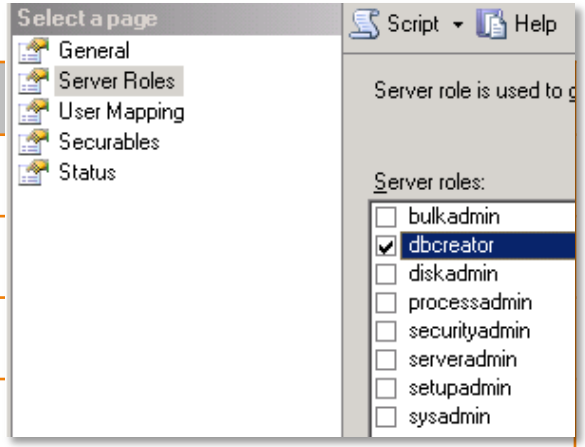
```
Redmond'; drop table OrdersTable--
```

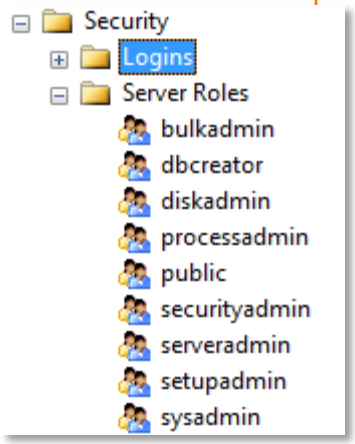
Säkerhetssystemet



Roller – Server, statiska

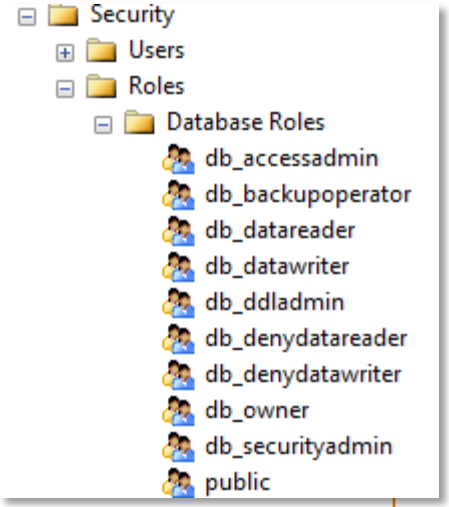
Roll	Får utföra
bulkadmin	Får köra BULK INSERT kommando
dbcreator	Create, Alter, Drop och Restore
diskadmin	Administrera filer på disk
processadmin	Radera login via T-SQL kod
securityadmin	Skapa och sköta inloggning med SQL Server Authentication
serveradmin	Adminstrera server. Starta o Stoppa, ändra egenskaper för SQL Server
setupadmin	Utföra arbete som har att göra med flerserver-användning
sysadmin	Högsta behörighet, Kan utföra allt





Ägare av en databas blir den som skapar databasen. Om annan user ska vara ägare måste ägaren eller sysadmin byta ägare på databasen.

Roller – Databas, dynamiska

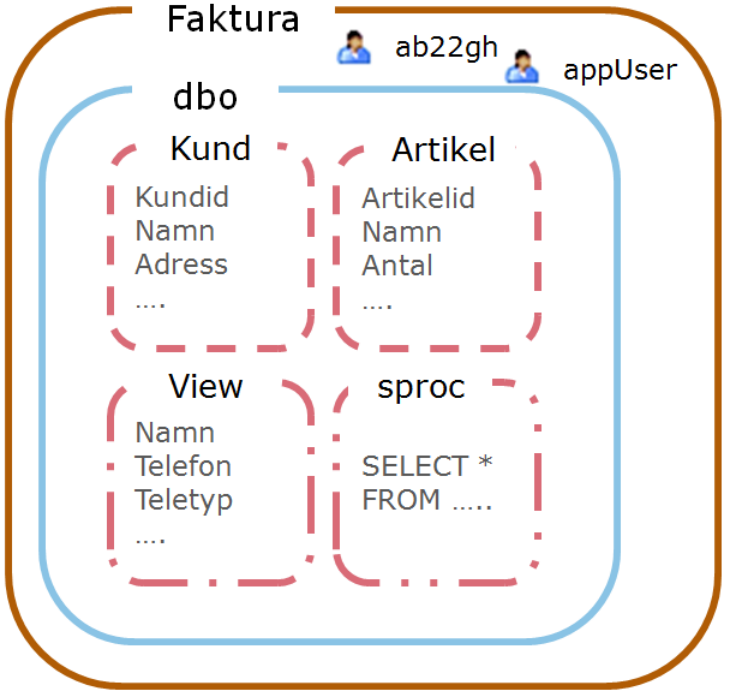


Roll	Får utföra
db_accessadmin	Får tilldela åtkomst för en login
db_backupoperator	Får ta och hantera backup
db_datareader	Får läsa data från alla användartabeller
db_datawriter	Får skriva data till alla användartabeller
db_ddladmin	Får köra SQL DDL kommandon
db_denydatareader	Får inte läsa data från användartabeller
db_denydatawriter	Får inte skriva/ändra data i användartabeller
db_owner	Ägare av databasen – får göra allt med databasen
db_securityadmin	Får ändra roller och rättigheter
public	Får se alla objekt som är skapade som Public

Schema (1 av 2)

Struktur

1. En server kan ha en eller flera databaser
2. En databas kan ha ett eller flera scheman
3. Ett schema kan innehålla ett eller flera objekt



Namnsättning

Server.Databas.Schema.Objekt
falken.faktura.dbo.kund

I ett schema kan det finnas olika användare med olika eller samma roller.

Ägare

Ett schema ägs av en User
Ett schema äger alla objekt i schemat

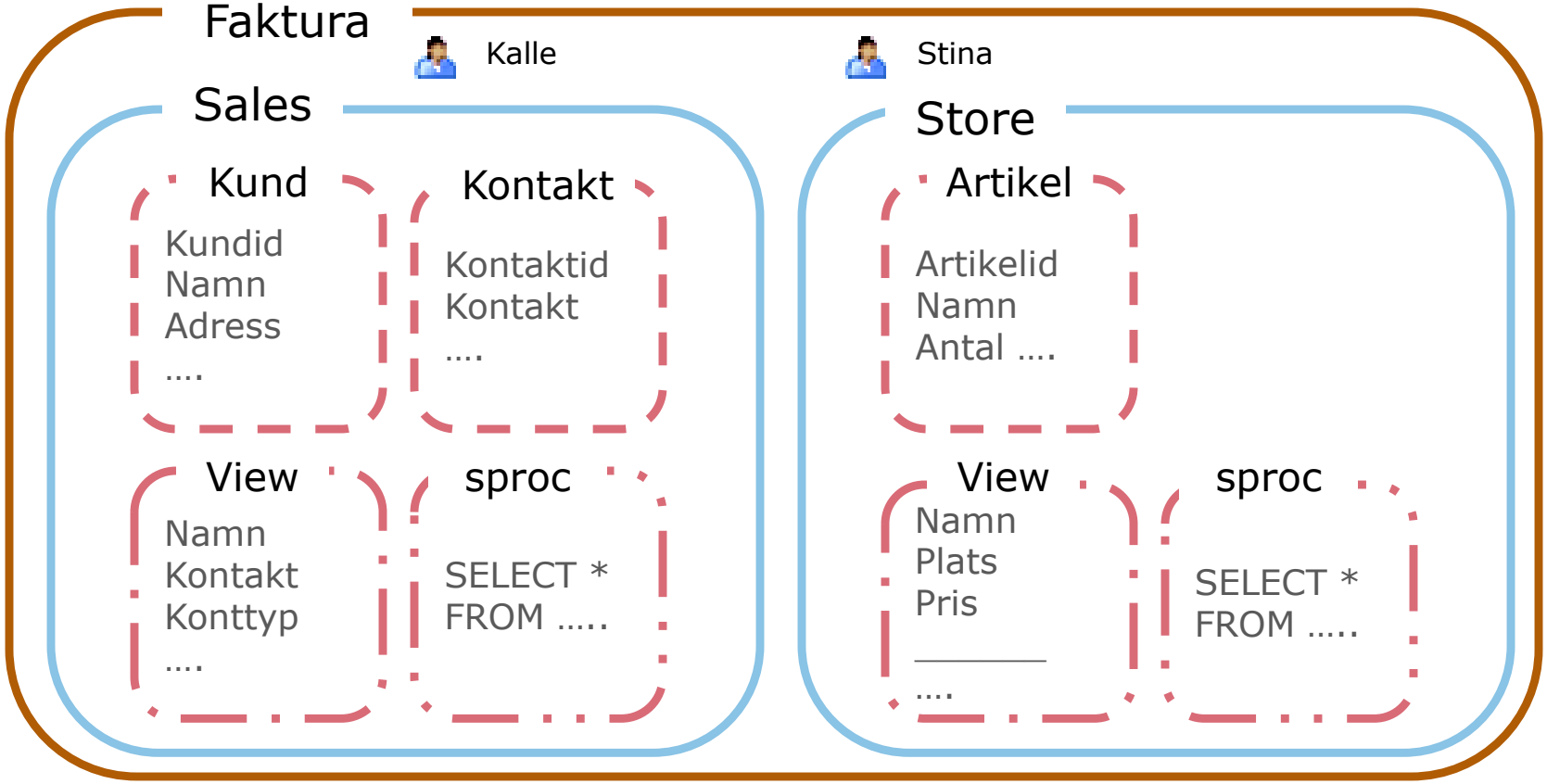
Standardschema i en databas är dbo och det ägs av den som äger databasen. Normalt är det den som skapat databasen (dbcreator).

Ett objekt i ett schema kan ha samma namn som ett objekt i ett annat schema i samma databas.

Schema (2 av 2)

Exempel

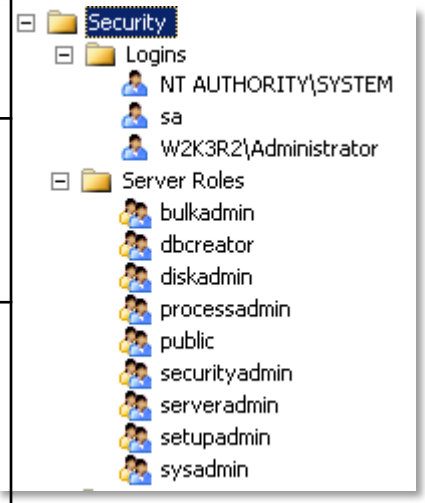
Försäljningsavdelningen ska ha inte kunna skapa (lägga till) nya artiklar. Det är förrådsavdelningen som sköter det. Förrådsavdelningen ska inte kunna arbeta med fakturor.



Att visa kundens namn ur schema sales: **SELECT namn FROM sales.kund;**

Login (1 av 6)

Begrepp	Innebörd
Login	För att komma åt SQL Server så måste man kunna logga in. Ett login konto måste skapas och tilldelas rättigheter.
User	När ett login tilldelas rättigheter i en databas blir login en user i databasen. Där bestäms också vad en användare får göra i databasen.
Roller	En roll beskriver vad ett login får utföra. Är login en dbcreater så får den som loggat in skapa databaser och blir ägare av den eller de databaser som skapats med det aktuella login.



Login (2 av 6)

Det finns två metoder på hur ett login kontrolleras, authenticate.

Windows Authentication
är den säkraste metoden. Kontrollen sker via Windows inloggningssystem vilket rekommenderas.
Login uppgifter hämtas från Windows. Användaren måste alltså finnas i Windows.

SQL Server Authentication
är den metod där kontrollen enbart sker i SQL Server. Användaren behöver inte finnas i Windows.

I kommande exempel visas hur man skapar en användare med SQL Server Authentication. Det fungerar ungefär på samma sätt med Windows Authentication. Skillnaden är att man hämtar användaren från Windows inloggningsmiljö istället för att själv ange login.

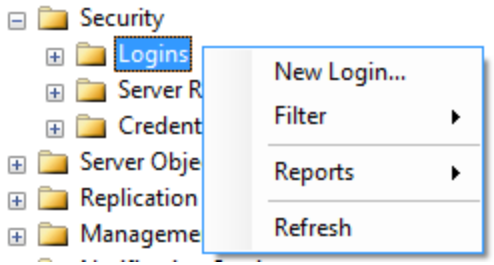
Login (3 av 6)

Varje användare som ska komma åt databasen måste ha ett login. Flera användare kan dela på ett och samma login vilket görs utifrån den roll som användaren har. Exempelvis så har man en gemensam användare för anslutning via ett webbgränssnitt, en webanvändare.

När du ska skapa en ny login så ska du vara inloggad som administratör.

Befintliga inloggningskonton hittar du under Security och Login.

För att skapa ett nytt login högerklickar du på Login och väljer *New Login…*



Innan du börjar. Högerklicka på Serverobjekt och välj Properties. Välj Security och se till att du har *SQL Server and Windows Authentification mode* markerat.

Login (4 av 6)

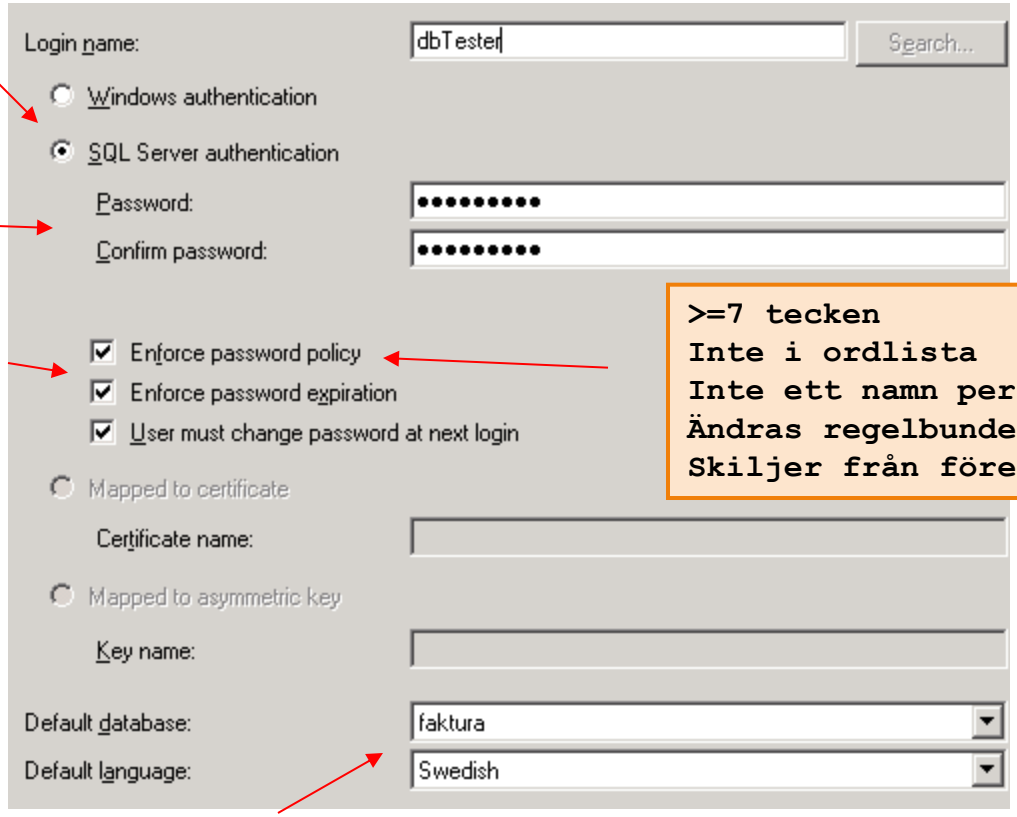
Att skapa en inloggning med SQL Server authentication:

1 Ange ett namn för login

2 Markera valet SQL Server authentication

3 Ange lösenordet två ggr

4 Bestäm alternativen




```
>=7 tecken
Inte i ordlista
Inte ett namn person/user
Ändras regelbundet
Skiljer från föregående
```

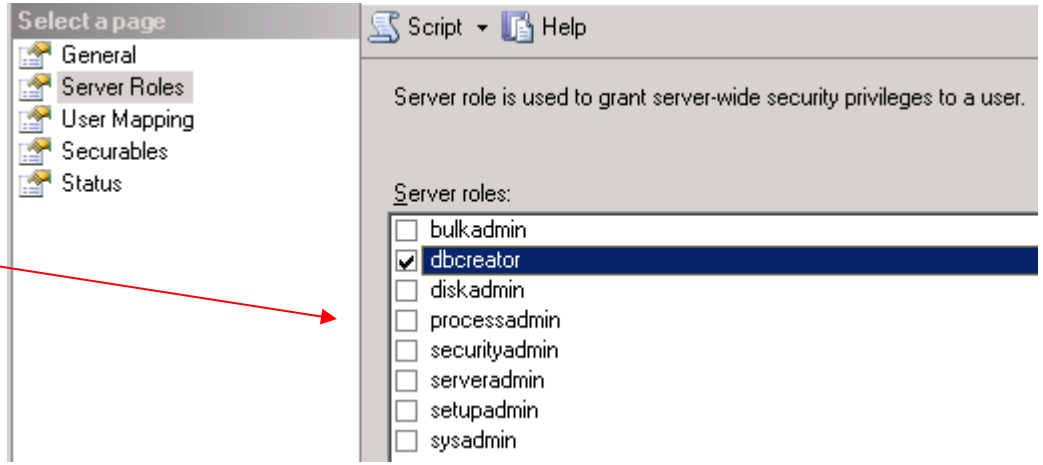
[Password Policy Books OnLine]

5 Ange default databas (master) och ev default språk

Login (5 av 6)

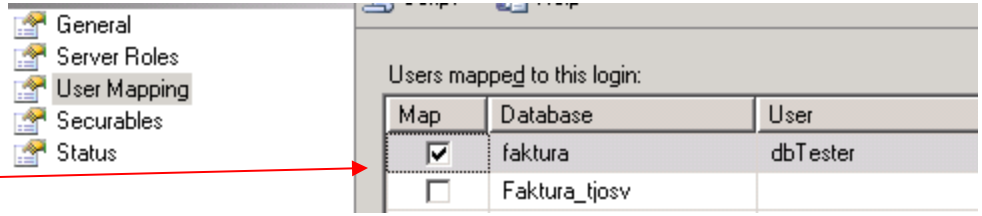
6 Välj fliken Server Roles

7 Markera roll/roller på servernivå

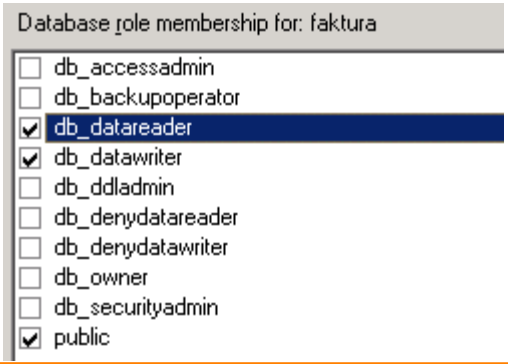


8 Välj fliken User Mapping

9 Markera databas (om finns)



10 Markera roll/roller på den valda databasen

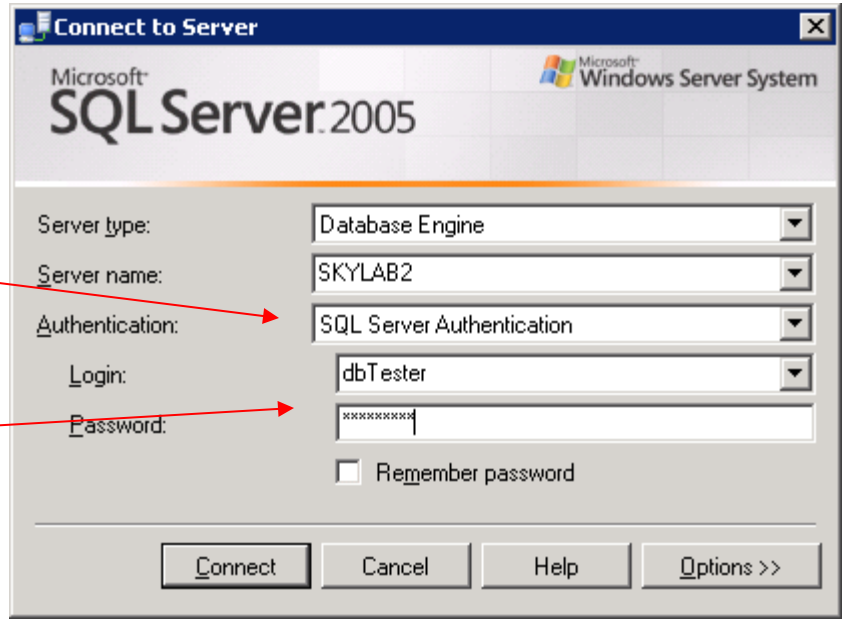


Login (6 av 6)

11 Avsluta med OK och det är klart att testa.
Välj File, Disconnect Object Explorer
Välj File, Connect Object Explorer

12 Ändra till
SQL Server Authentication

13 Ange Login och Password
Välj Connect

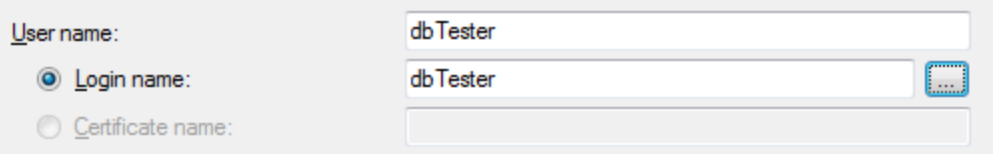


Kontrollera att ditt nya login har de roller som du angett. Kan du öppna databaser som login inte har rättighet till? Vad händer när du skapar en ny databas? Vem blir ägare av databasen? Högerklicka på databasen och kontrollera under fliken General.

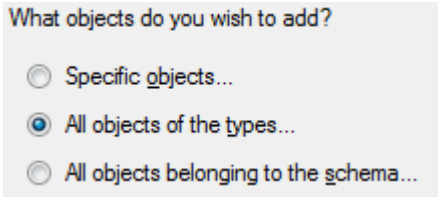
Skapa en User

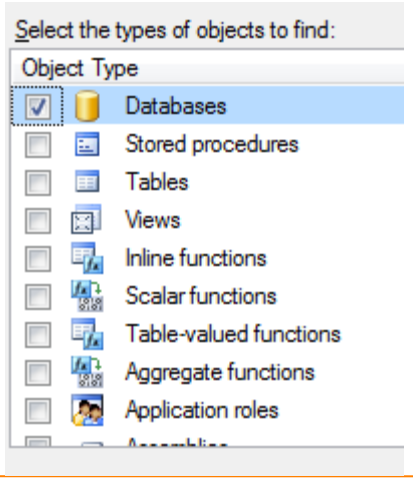
När du skapar ett login i SQL Server så kan du tilldela den åtkomst och rättigheter i din databas. Det kan du gör direkt när du skapar login eller som här.

- Öppna Security mappen i databasen och högerklicka därefter på Users. Välj New User…



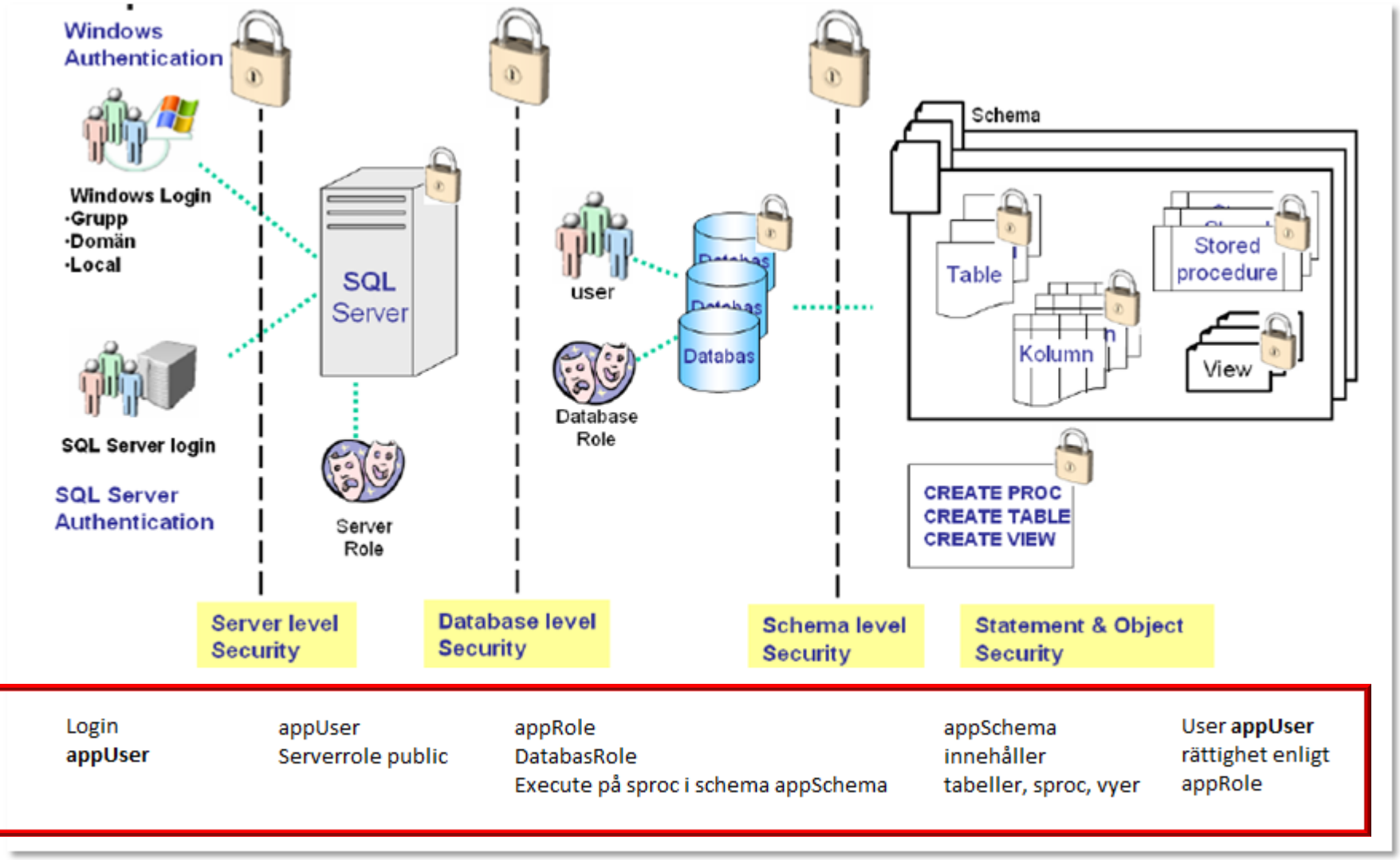
- Ange Login som ska vara användare.
- Välj sedan fliken Securable och klicka på Add…
- Markera All objekts of the types… Välj OK





- Här kan du välja vilka objekt som användaren ska få rättigheter för. Välj exempelvis Databas och OK.
- Slutligen bestämmer du vad användaren får göra i det valda objektet.

Säkerhetssystemet Praktiskt



Login **appUser**	appUser Serverrole public	appRole DatabasRole Execute på sproc i schema appSchema	appSchema innehåller tabeller, sproc, vyer	User **appUser** rättighet enligt appRole

Applikationsanvändaren AppUser utan eget schema

Förutsättning:
Vi behöver en användare som får logga in med en applikation. Användaren ska enbart ha möjlighet att köra våra lagrade procedurer och därmed endast kunna göra det som den lagrade proceduren är satt att göra.

Antag att det finns ett Login som heter appUser som du kan knyta till din databas. appUser kan logga in men kan inte göra något om du inte tilldelar den en användare med en roll!

1. Skapa en Database Roles (appRoles) med dbo som owner.
   Knyt in dbo Schema under Securable
   Markera Execute under Permissions
2. Skapa en User i din databas som använder loginet appUser.
   Se till att User har dbo Schema som default schema
   Knyt in Database Roles på User.
3. Kontrollera i Schema så att det är korrekt….

Applikationsanvändaren AppUser med eget schema

Förutsättning:
Vi behöver en användare som får logga in med en applikation. Användaren ska enbart ha möjlighet att köra våra lagrade procedurer och därmed endast kunna göra det som den lagrade proceduren är satt att göra.

Antag att det finns ett Login som heter appUser som du kan knyta till din databas. appUser kan logga in men kan inte göra något om du inte tilldelar den en användare med en roll!

1. Skapa ett schema i din databas. Det gör du under Security och Schema. dbo som owner.
2. Skapa en Database Roles (appRoles) med dbo som owner.
   Knyt in ditt Schema under Securable
   Markera Execute under Permissions
3. Skapa en User i din databas som använder loginet appUser.
   Se till att User har ditt nya schema som default schema
   Knyt in Database Roles på User.
4. Kontrollera i Schema så att det är korrekt….

Backup / Restore / Recovery

Du bör se till att din databas blir säkerhetskopierad efter ett visst mönster. Det är bara en tidsfråga tills du får ett fel i någon form.
Felet kan vara en hårddisk som går sönder, en brand, vattenskador, stöld, kraschade tabeller, hackade data, felaktigt handhavande eller annat som kan skada din databas.

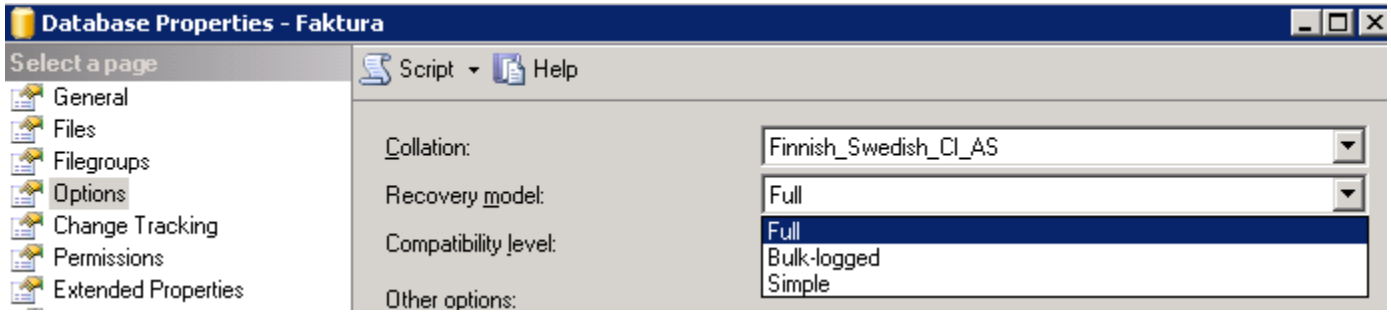
Det finns två typer av hårddiskar. De som gått sönder och de som ännu inte gått sönder.

Kommando	Innebörd
BACKUP	Säkerhetskopierar databas med alla objekt. Styrs av vald Recovery modell på databasen.
RESTORE	Återläser från den backup som skapats tidigare
RECOVERY	Återställer databasen till ett visst skick - tidsbundet. Kan vara tidsangivet.

Backup genomförs normalt i MS SQL Server. Den Backup som sker av skivenheten och dess filer är mindre lyckad för databaser.

Backup förberedelse

När du skapar din databas kan du välja vilken Recovery model som ska användas. Du kan också byta senare genom att högerklicka på din databas. Välj Properties och fliken Options.



Nivå	Innebörd
Full	Backup kan tas på både databas och log. Både full och differentiell Använd denna typen på stora databaser.
Simple	Backup kan tas enbart på databas. Endast full backup och differentiell. Används på mindre databaser.
Bulk-logged	Backup tas i sk Bulkformat.

Om du inte kan ändra från ex Simple till Full: Kör följande DDL kommando:

```
ALTER DATABASE [din_databas_namn] SET RECOVERY FULL;
```

Backup-Strategi

Beroende på hur affärskritiska data är så kan strategin för backup variera.

Typ	Kommentar
OLAP	**On Line Analysis Processing**. Data uppdateras inte så ofta - data analyseras.
OLTP	**On Line Transactions Processing**. Data uppdateras och ändras ofta vilket kräver större exakthet i återläsningen. Man vill tappa så lite som möjligt.

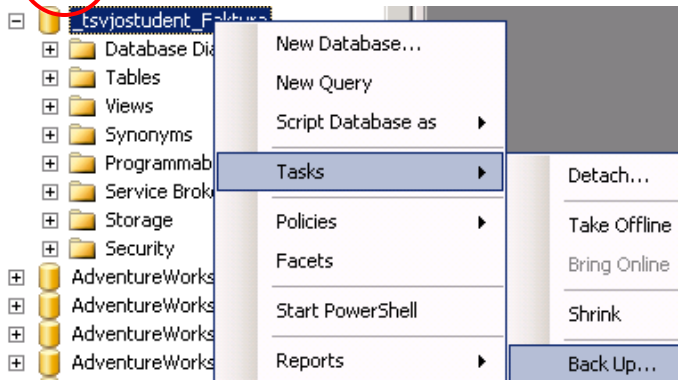
Backup typ	Större databaser (Full)	Mindre databaser (Simple)
Full	Flera ggr/vecka, Varje dag?	Minst 1ggr /vecka
Differentiell	1- flera ggr/dag	Beror på uppdateringsfrekvens
Log	1- flera ggr/dag	

Testa att göra återläsningar ibland så du ser att dina backuper är korrekta.

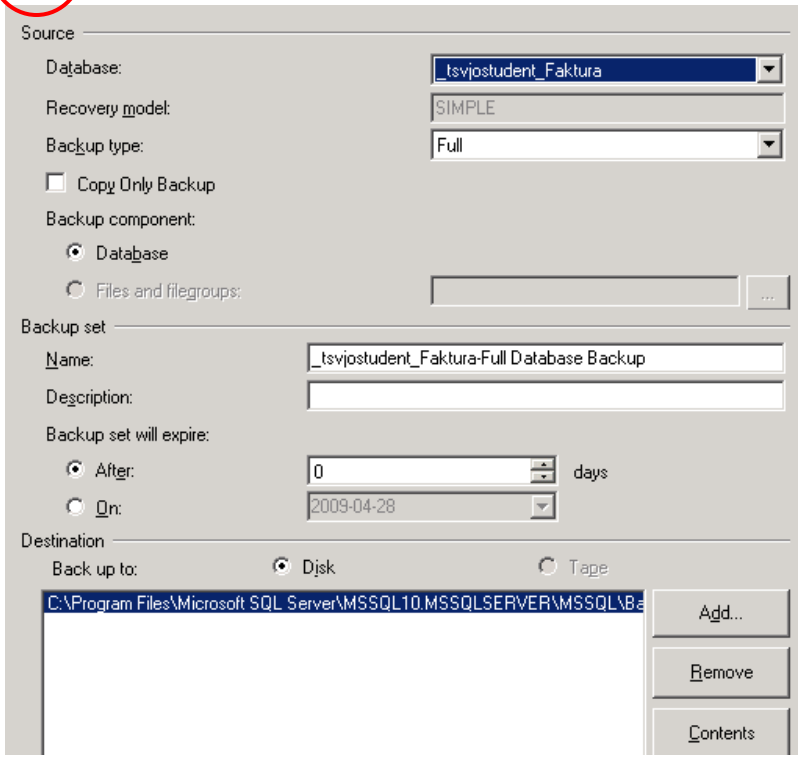
Använd olika media vid backupptagning – rullande. Ex månad/vecka/dag.

Backup

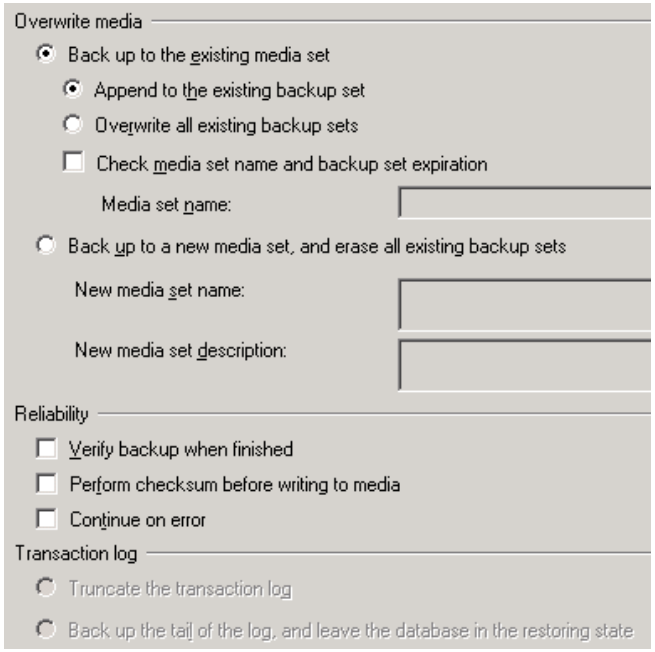
1 Högerklicka på din databas



2 Välj Backup Type

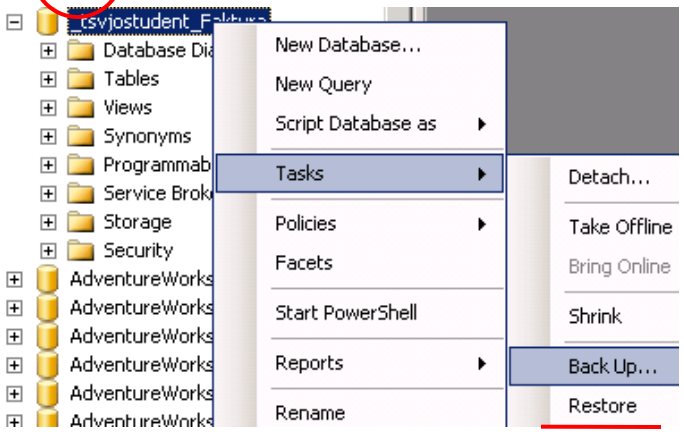


3 Gör dina val!

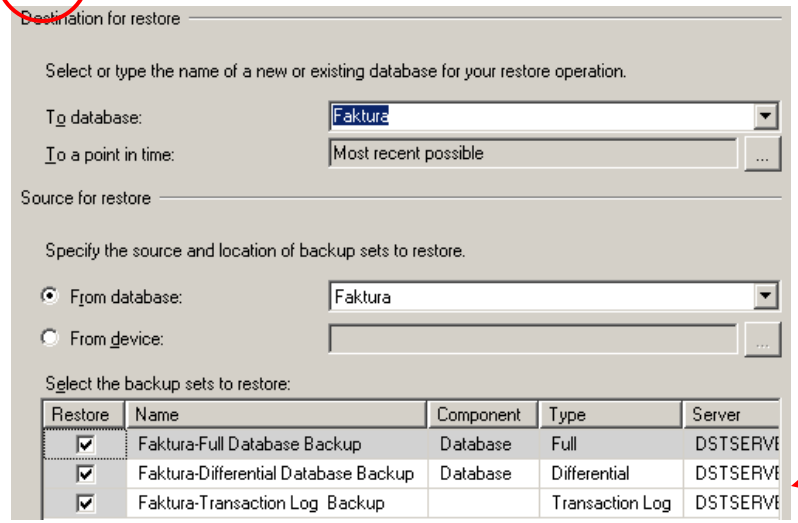


Restore / Recovery

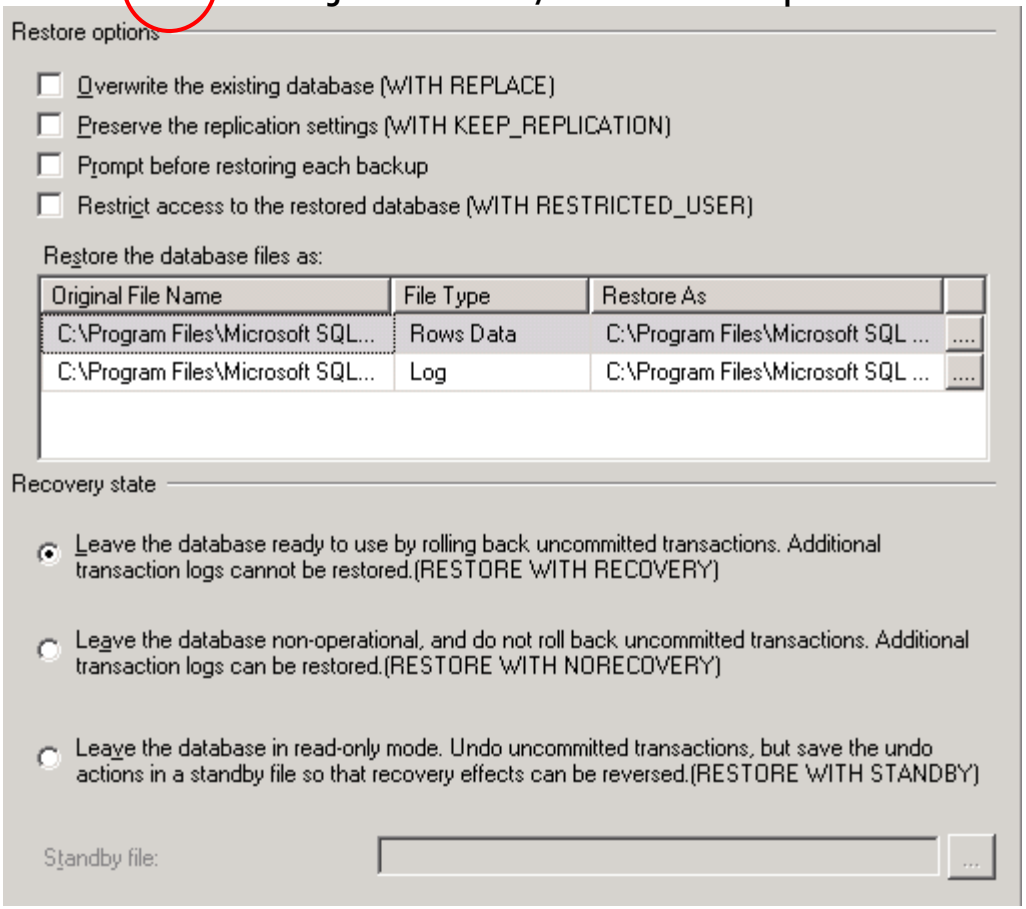
1 Högerklicka på din databas



2 Återställning ska ske till/av



3 Välj Restore / Recover options



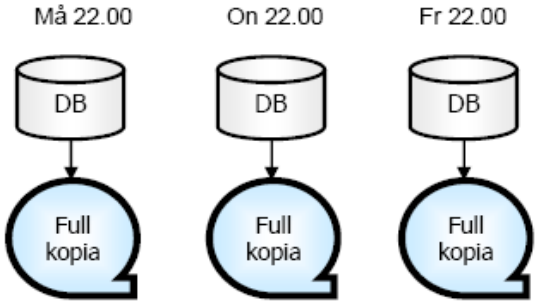
Det finns en Full, en Differentiell backup på databasen och en backup på logfilen.

Samspelet mellan Databas och Log vid Backup

Återhämtning (Recovery) med "Simple Model"

Till vilken punkt kan vi återhämta databasen om databasen kraschar kl 15.00 på Onsdag? _____[1]

Måndag kl 22



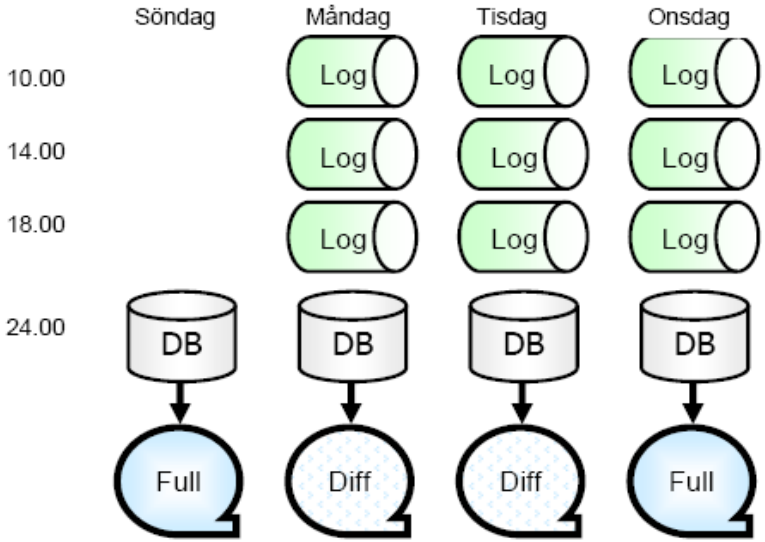
Återhämtning med "Full Model"

Antag att databasen krashar Onsdag kl 17.00.
Till vilken punkt kan man recover? _____[2]

Onsdag kl 17

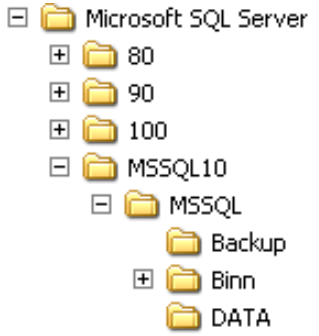
Vad gäller om transaktionsloggen är skadad? _____[3]

Onsdag kl 14



Backup på fil / Flytta databas

Om du vill ta en kopia på databasen eller om du vill flytta databasen utan att göra Backup eller Scripta den så kan du kopiera/flytta den när den är frånkopplad (Detach). Annars kan du inte kopiera/flytta databasen.

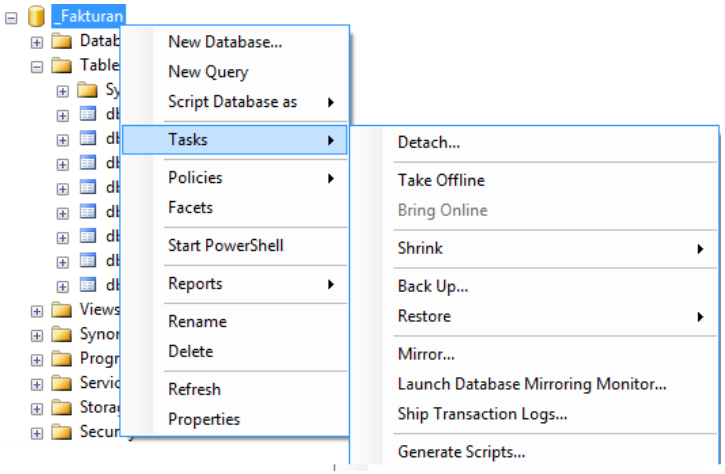


- ✓ Gör en Detach av databasen. Först ska du se till att ingen använder databasen.
  Ingen har tillgång till databasen när du gjort Detach.
- ✓ Öppna Utforskaren och kopiera/flytta databasen.
  Databasen finns normalt under C:\Program files\….\Data
- ✓ Anslut databasen med Attach. Peka ut den nya platsen och markera databasen. Nu kan användarna få tillgång till databasen igen.

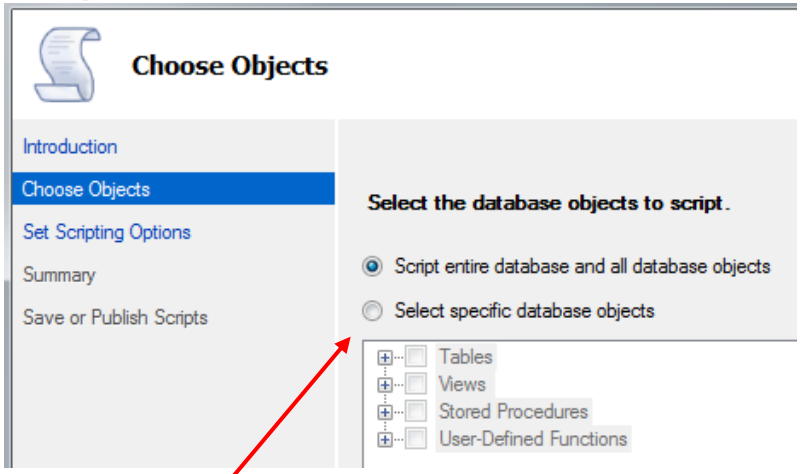
Scripta en databas (1 av 3)

Om en databas exempelvis ska flyttas till en kund så kan det vara lämpligt att scripta den och sedan ta med scriptet för installation hos kund.

1 Högerklicka på din databas



2 Välj vad som ska skriptas



Vill du ha alla objekt i databasen scriptade?

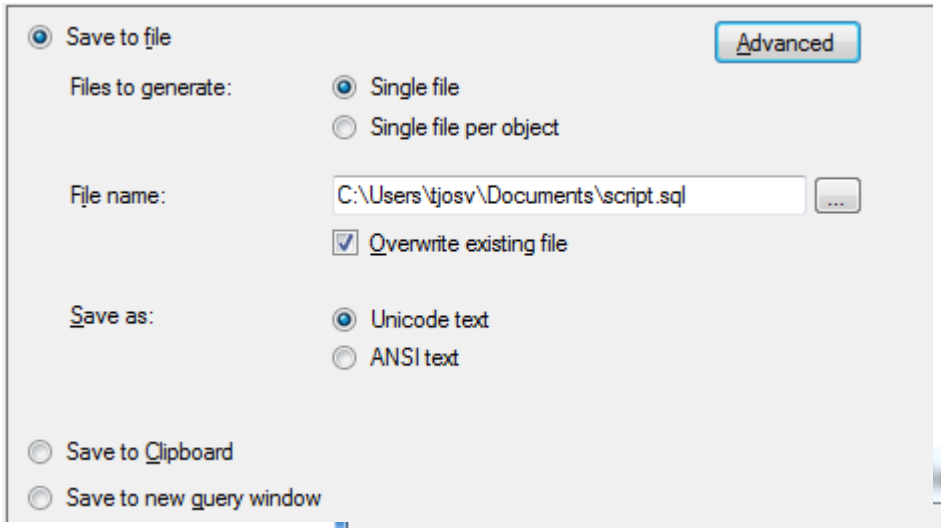
Scripta en databas (2 av 3)

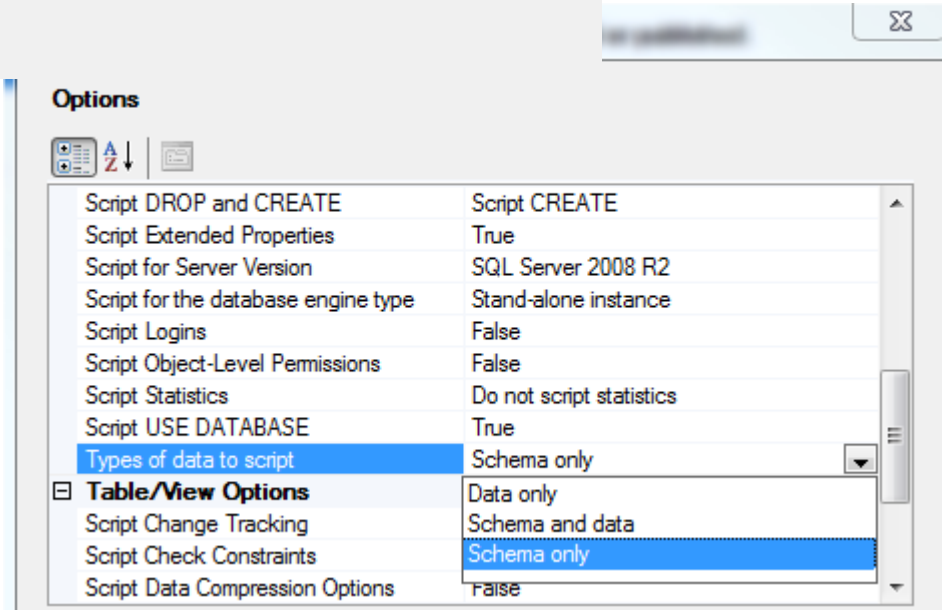
3

Bestäm var o hur för scriptet.

Under Advanced kan du detaljbestämma vad scriptet ska innehålla.

I SQL 2008 kan även data scriptas.



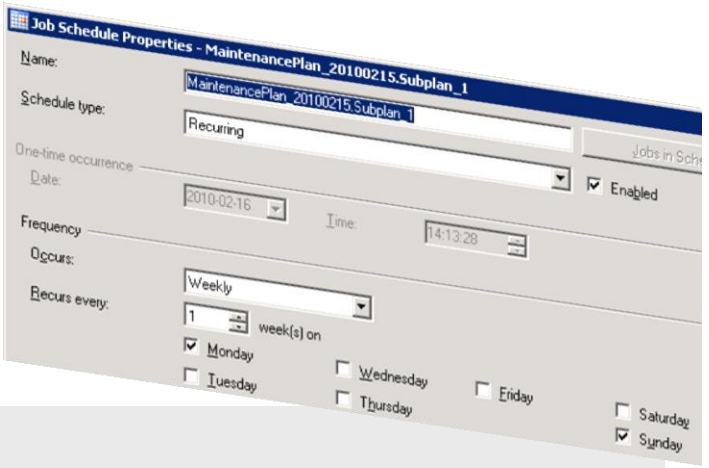


Scripta en databas (3 av 3)

4 Så här kan en del av resultatet ser ut:

```
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Telef
BEGIN
CREATE TABLE [dbo].[Telefon](
    [TelId] [int] IDENTITY(1,1) NOT NULL,
    [KundId] [int] NOT NULL,
    [TelTypId] [int] NOT NULL,
    [Telnummer] [varchar](20) NULL,
 CONSTRAINT [PK_Telefon] PRIMARY KEY CLUSTERED
(
    [TelId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[Telef
BEGIN
CREATE TABLE [dbo].[TelefonTyp](
    [TelTypId] [int] IDENTITY(1,1) NOT NULL,
    [Teltyp] [varchar](20) NULL,
 CONSTRAINT [PK_TelefonTyp] PRIMARY KEY CLUSTERED
(
    [TelTypId] ASC
)WITH (PAD_INDEX  = OFF, IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

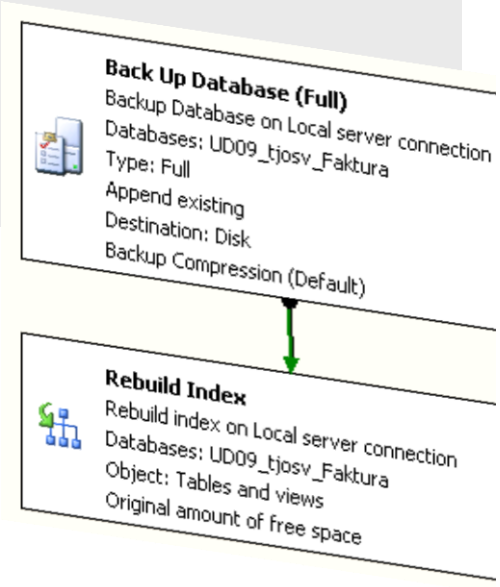
Maintenance Plan



Med hjälp av en Maintenance Plan (underhållsplan) kan du automatisera backuper och andra arbeten för din databas.

Om Management Plan inte är tillgänglig kan du konfigurera om SQL Server.
Kör följande script:

```
sp_configure 'show advanced options', 1
GO
RECONFIGURE;
GO
sp_configure 'Agent XPs', 1
GO
RECONFIGURE
GO
```



I Objekt Explorer ska du nu hitta Maintenance Plan under noden Management.

SQL Agent måste vara startad.

Genomgången av detta sker under lektionstid och visas inte detaljerat här. Se boken sid 220.