

Bogen MCTS Exam 70-432

Microsoft SQL Server 2008 – Implementation and Maintenance

Self-Paced Training Kit

Dette materiale er essensen, uddrag samt visse tilføjelser til ovennævnte bog

Indhold

Indhold

Bogen MCTS Exam 70-432	1
Microsoft SQL Server 2008 – Implementation and Maintenance	1
Self-Paced Training Kit	1
Overblik	7
Objekt hierarki	7
Databasens fysiske filer	7
Kap 1. Installing and Configuring SQL Server 2008	8
Lesson 1: Determining Hardware and Software Requirements	8
Minimum Hardware Requirements	8
Supported Operating Systems	9
Practice 1 Verify Minimum Requirements	9
Practice 2 Supporting Software Requirements	10
Lesson 2: Selecting SQL Server Editions	10
SQL Server Services	10
SQL Server Editions	10
Lesson 3: Installing and Configuring SQL Server Instances	11
Service Accounts	11
Collation Sequences	11
Authentication Modes	11
SQL Server Instances	11
SQL Server Configuration Manager	11
Practice 1 Creating Service Accounts	11
Practice 2 Install a SQL Server Instance	12
Installer en named SQL Server	13
Practice 3 Install the AdventureWorks Sample Database	13
Practice 3(Ekstra. Installation af Northwind)	14
How To Enable FILESTREAM Feature On SQL Server 2008	14
Lesson Review	15
Lesson 4: Configuring Database Mail	16
Practice 1 Configuring Database Mail	16
Send en mail	19
En trigger der sender en mail når der indsættes en række	19
Kap 2. Database Configuration and Maintenance	21
Lesson 1: Configuring Files and Filegroups	21
Files and Filegroups	21
Transaction Logs	21
Filestream	22
Databasen TempDB	22
Andre system-databaser	22
Practice 1 Creating Databases (44)	24

Lesson 2: Configuring Database Options	26
Database Options.....	26
Recovery Options	26
Backup Type	26
Recovery Model	26
Damaged Pages.....	27
Auto Options	27
Change Tracking	27
Access til databasen	27
Parameterization.....	28
Collation Sequences	28
Practice 1 Changing the Database Recovery Model (52).....	29
Lesson 3: Maintaining Database Integrity	29
Database Integrity Checks.....	29
Practice 1 Checking Database Integrity (56)	29
Lesson 1: Creating Tables.....	30
Schemas	30
Lesson 2: Implementing Constraints.....	31
Ekstra: Brug af Filestream	31
C#-koden	32
Kap 4. Designing SQL Server Indexes	34
Lesson 1: Index Architecture.....	34
Lesson 2: Designing Indexes.....	36
Practice: Creating Indexes (side 102).....	36
Lesson 3: Maintaining Indexes.....	36
Practice: Maintaining Indexes (side 106).....	36
Kap 5. Full Text Indexing.....	37
Lesson 1: Creating and Populating Full Text Indexes.....	37
Practice: Creating Full Text Indexes	37
Lesson 2: Querying Full Text Data.....	37
Practice: Querying with a Full Text Index	37
Lesson 3: Managing Full Text Indexes.....	38
Kap 6. Distributing and Partitioning Data	39
Lesson 1: Creating a Partition Function	39
Lesson 2: Creating Partition Scheme	40
Lesson 3: Creating Partitioned Tables and Indexes	40
Practice: Creating a Partitioned Function, Partition Scheme, Partition Table and Indexes (side 140, 144 og 146)	42
Lesson 4: Managing Partitions	44
Kap 7. Importing and Exporting Data	45
SQLCMD	45
Windows Command Batch Fil	46
Flytning og transformering af data	48
Oprettelse af script, der kopierer data	48
Data fra en tabel til en anden tabel	48
Lesson 1: Importing and Exporting Data.....	49

BCP	49
BULK Insert.....	50
The SQL Server Import and Export Wizard	51
Practice 1: Export Data Using BCP(side 167)	51
Practice Postnumre:.....	51
Practice 2: Exporting Tables (side 167)	52
Practice Extra: Her kommer så et rigtigt SSIS projekt udført i Visual Studio.....	53
Fra bogen: MS SQL Server 2005 Integration Services Step by Step	53
Kapitel 3	53
Øvelse Chap3A Creating a new Integration Services Project.	53
Opret Projekt.....	53
Tilføj Connection Managers	53
Connection Manager til en SQL Server database	53
Connection Manager til Excel-fil	53
Creating a Data Flow	54
Adding Data Adapters	54
Add source adapter.....	54
Add Destination Adapter	55
Udfør den oprettede package.....	55
Kap 8. Designing Policy Based Management	56
Lesson 1: Designing Policies.....	56
Facets	56
Conditions	56
Policy Targets	56
Policies	56
Policy Categories	56
Practice: Defining Policies and Checking for Compliance	56
Øvelse MS Official Course 6231A, Lab 2, Exercise 3: Creating a policy	59
Task 1: Create a policy to enforce naming conventions	59
Afprøvning.....	60
Kap 9. Backing up and Restoring a Database	62
Lesson 1: Backing up Databases	62
Backup Types.....	62
Full Backups.....	62
Differential Backups	62
Filegroup Backups	63
Partial Backups	63
Page Corruption	63
Maintenance Plans.....	63
Certificates and Master Keys	64
Validating a Backup.....	64
Backup kommando med dynamiske filnavne	64
Practice.....	65
Practice 1: Create a Compressed, Mirrored, Full Backup	65
Practice 2: Create a Transactional Log Backup	65
Practice 3: Create a Differential Backup	66

Lesson Review	66
Lesson 2: Restoring Databases.....	66
Practice 1: Purposely Damage a Database	66
Practice 2: Restore a Full Backup	67
Practice 3: Restore a Differential Backup	67
Practice 4: Restore a Transaction Log Backup	68
Lesson 3: Database Snapshots	69
Practice 1: Creating a Database Snapshot	69
Kap 10. Automating SQL Server.....	70
Job der indsætter rækker i en tabel.....	70
Kap 11. Designing SQL Server Security	72
SQL Server	72
Database	72
Rettigheder	73
Roller	73
Faste Server Roller	73
Database roller.....	74
Application Rolle	75
Brugerdefineret rolle	75
Grant, deny og revoke.....	75
System-views.....	76
Øvelser	76
Opret Windows brugere	76
Opret en SQL Server Login for en Windows bruger.....	76
Opret en SQL Server login for en ikke-Windows bruger.....	77
To connections.....	78
Afprøv Carol og giv rettigheder.....	78
Rollen public.....	80
Schema	81
Giv Carol et schema	81
Ny Login oprettet i SQL og med rettighed til en specifik table.....	82
Øvelser	83
Kap 12. Monitoring Microsoft SQL Server.....	90
Kap 13. Optimizing Performance.....	90
Kap 14. Failover Clustering	90
Kap 15. Database Mirroring.....	90
Kap 16. Log Shipping.....	90
Kap 17. Replication	90
Andre emner	91
Noter:	91

Alle øvelser i bogen er udført på Windows XP SP2.

En SQL Server Instans er en installation af SQL Server.

Overblik

Objekt hierarki

- Windows Server
 - SQL Server Default
 - SQL Server Named
 - Databaser
 - System
 - Master
 - Model
 - TempDB
 - MSDB
 - User-databaser
 - Schema
 - Database-objekter
 - Table
 - Extents
 - Pages
 - Rows

Databasens fysiske filer

Første datafil .mdf i filegroup Primary

Ekstra datafil(er) .ndf i evt. ekstra filegroup(s).

En eller flere transactional log filer.

Kap 1. Installing and Configuring SQL Server 2008

Lesson 1: Determining Hardware and Software Requirements

Minimum Hardware Requirements

| SQL Server 2008 | 32 - bit | 64 – bit |
|-----------------|--------------------------|---|
| Processor | Pentium III eller højere | Itanium, Opteron, Athelon eller Xeon/Pentium med EM64T support. |
| Processor Speed | 1.0 GHz | 1,6 GHz |
| Memory | 512 MB | 512 MB |

Supported Operating Systems

| | Windows Server 2008
Standard | Windows Server 2003
Standard SP2 | Windows XP Professional
SP2 | Windows Vista Home
Basic | Windows XP alle udgaver | Windows Server 2003
Small Business |
|------------|---------------------------------|-------------------------------------|--------------------------------|-----------------------------|-------------------------|---------------------------------------|
| Enterprise | X | X | | | | |
| Standard | X | X | | | | |
| Developer | X | X | X | X | | |
| Evaluation | X | X | X | X | | |
| Express | X | X | X | X | X | X |
| Workgroup | X | X | | | | |
| Compact | X | X | | | | |

Windows Server 2008 Core supporterer ikke SQL Server 2008, da der savnes .NET Framework til at supportere FILESTREAM, SPATIAL og DATE data typer mm.

SQL Server 2008 kræver

.NET Framework 3.5, som dog vil blive installeret automatisk hvis den mangler.

samt

Microsoft Data Access Components(MDAC) 2.8 SP1.

Shared Memory, Named Pipes eller TCP/IP networking support.

Internet Explorer 6 SP1 eller højere.

Practice 1 Verify Minimum Requirements

Side 5.

Højreklik My Computer/Denne Computer og vælg Egenskaber.

Under fanen Generelt check System for operativsystem og Computer for processor og RAM.

Practice 2 Supporting Software Requirements

Side 6.

Gå i Kontrolpanel/Tilføj og fjern programmer, og undersøg at de krævede versioner for .NET Framework og Internet Explorer er installeret.

I regedit: HKEY_LOCAL_MACHINE\Software\Microsoft\DataAccess – check at MDAC versionen i FullInstallVer er tilstrækkelig f.eks. 2.81.1132.0

Lesson 2: Selecting SQL Server Editions

SQL Server Services

- SQL Server
- SQL Server Agent
- SQL Server Browser
- SQL Full Text-Filter Daemon Launcher
- Service Broker
- SQL Server Integration Services
- SQL Server Reporting Services
- SQL Server Analysis Services

SQL Server Editions

| | |
|--------------|---|
| Enterprise | Har det hele. Flere instanser på samme server. |
| Standard | Lidt mindre. |
| Workgroup | Endnu mindre. |
| Express | Endnu mindre. Gratis. |
| Compact | Mindst. Ingen stored procedures, triggers, views |
| Developer | Samme som Enterprise. Må ikke bruges i produktion. |
| Evaluation | Samme som Enterprise. Gratis 180 dage. Må ikke bruges i produktion. |
| Professional | ??? |

Lesson 3: Installing and Configuring SQL Server Instances

Service Accounts

De rettigheder, som hver service har til ressourcerne i operativsystemet er afhængig af de service-accounts, som udfører dem.

Der behøves accounts for

- SQL Server Service (Database Engine)
- SQL Server Agent

Collation Sequences

Karaktersæt og understøttelse af sprog mht. hvordan karakterer behandles, sorteres og sammenlignes. Dette gælder også formater på datoer.

Collation kan sættes på instans, database, table og kolonne. Det kræves kun at blive sat på instansen, som vil defaulte til alle de andre niveauer, hvis det ikke udtrykkeligt overskrives.

Authentication Modes

- Windows Only (integrated security)
- Windows and SQL Server (mixed mode)

SQL Server Instances

Der kan installeres op til 50 instances på en enkelt maskine. En af dem kan være default og resten named instances.

Kun Enterprise kan installeres som flere instanser på en enkelt maskine. (Gælder dette mon så også Developer og Evaluation?)

SQL Server Configuration Manager

- Starte, stoppe, pause og restarte services
- Ændre service accounts og deres passwords (brug ikke Windows Service Control applet til dette)
- Managing startup mode for en service
- Konfigurere startop parametre for services

Practice 1 Creating Service Accounts

Det kan være en god ide at oprette disse brugere:

Side 22.

Gå ind i Denne Computer – Manage og opret to nye brugere SQL2008TK432DE og SQL2008TK432SQLAgent.

Passord Tjudyqmd7.

Det er sket nogle gange at service'ne ikke kunne starte.

I Computeradministration | systemværktøjer | Logbog | System kan det ses at det er logon fejl.

Ved at ændre password for de to Sql server brugere og så gå ind Tjenester og ændre password for brugerne der, kunne der startes. Det hjalp ikke at ændre password i Sql Server Configuration Manager.

Practice 2 Install a SQL Server Instance

Side 22.

Følg bogens vejledning for Practice 2 og installer en Default instans af SQL Server 2008.

Såsom:

Vælg at installere en default server, hvilket vil sige at den ikke får noget navn, men identificeres med computerens navn. Der kan senere installeres en named server.

Vælg at installere alt og fravælg derefter Analysis Services og Reporting Services. De bruges ikke i det følgende. Sørg for at Full-text Search er valgt.

Vælg user SQL2008TK432SQLAgent til SQL Server Agent

Og SQL2008TK432DE til resten.

Collation vælges Danish_Norwegian_CI_AS som evt er default.

Filestream enables under Database Engine Configuration, samme vindue som Authentication Mode sættes, under fanebladet Filestream.

Add current user som administrator.

Vælg Windows Authentication Mode. Der vælges Mixed Mode senere.

Analysis Service Configuration: Add Current user.

Reporting Services Configuration: Lad den være som default.

Errors and Usage reporting: Vælg intet.



Når installationen er færdig kan filerne findes i mappen

C:\Programmer\Microsoft SQL Server

Mapperne 80, 90 og 100 er fælles filer for alle serverne på computeren og hidrører henholdsvis fra versionerne SQL Server 2000, 2005 og 2008.

I mappen C:\Programmer\Microsoft SQL Server\100\Tools\Binn findes en del værktøjer såsom bcp.exe til at flytte data, og SQLCMD.EXE som er en prompt sql-fortolker.

SQL Server Management Studio er "C:\Programmer\Microsoft SQL Server\100\Tools\Binn\VSShell\Common7\IDE\Ssms.exe"

C:\Programmer\Microsoft SQL Server\MSSQL10.SQLEXPRESS er mappen for sqlexpress
C:\Programmer\Microsoft SQL Server\MSSQL10.MSSQLSERVER er mappen for den nyligt
installerede default server.
I mappen C:\Programmer\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA ligger
filerne for alle default serverens databaser.

Installer en named SQL Server

Installer evt. en server mere som får et navn, da der kun kan være en default server på en
computer. Installer på samme måde som den første bortset fra at den får et navn.
Denne installation går meget hurtigere end den første, da alle fælles ressourcer er installeret.

Practice 3 Install the AdventureWorks Sample Database

Side 25.

Gå ind på

<http://www.codeplex.com/MSFTDBProdSamples>

<http://msftdbprodsamples.codeplex.com/releases/view/45907>

og hent filen AdventureWorks2008_SR4.exe.

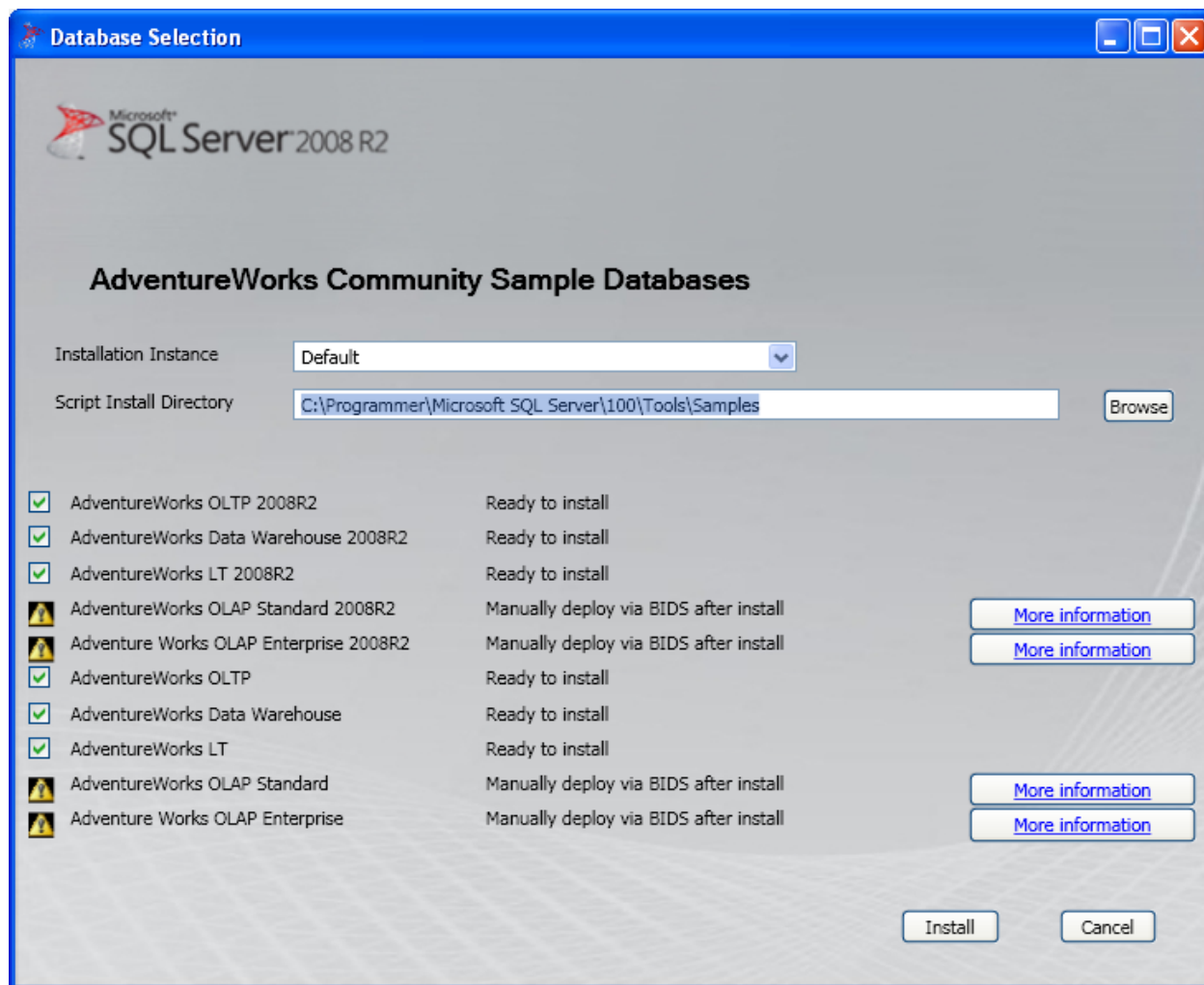
Husk at vælge Installation Instance hvis der er flere instanser. (Se skærbilledet næste side)

I modsætning til bogen, hvor der står at der skal hentes filen AdventureWorks2008*.msi, som ikke
er der, hentes filen AdventureWorks2008_SR4.exe og udføres. Den er meget stor. Jeg fik
databasen AdventureWorksLT2008 ud af det, som fandtes når jeg gik ind i Management Studio.
I bogen er det databasen AdventureWorksDW. Så måske skal det gøres om???
Anden gang fik jeg en flok databaser incl. AdventureWorksDW.
Den klager over at nogle af databaserne ikke kan installeres da forudsætningerne ikke er til stede,
bl.a. FileStream.

Der skal altså ikke udføres SQL restore kommandoer, som angivet i øvelsen.

Det hele blev hos mig lagt i C:\Programmer\Microsoft SQL Server\100\Tools\Samples
Så her fra kan det vel fjernes igen. Det fyldte 345 MB.

NB! Note that the AdventureWorks2008 database requires that full-text search and FILESTREAM is
enabled on your instance of SQL Server. If they are not, the installation will fail.



Hvis man har datafil og tilhørende logfil til AdventureWorks kan databasen tilføjes med Attach. Hvis man kun har datafilen kan det gøres med kommandoen (datafilen skal findes i den angivne sti):

```
CREATE DATABASE AdventureWorks2008R2 ON
(FILENAME = 'D:\Program Files\Microsoft SQL
Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\AdventureWorks2008R2_Data.mdf')
FOR ATTACH_REBUILD_LOG;
```

Practice 3(Ekstra. Installation af Northwind)

Kopier data- og log-fil for Northwind databasen ind mappen Data og attach Nortwind til databaseserveren.

Ved at højreklikke på Databases vælges Attach, der trykkes på Add og datafilen vælges.

How To Enable FILESTREAM Feature On SQL Server 2008

I bogens øvelser med installation af AdventureWorks står kommandoen

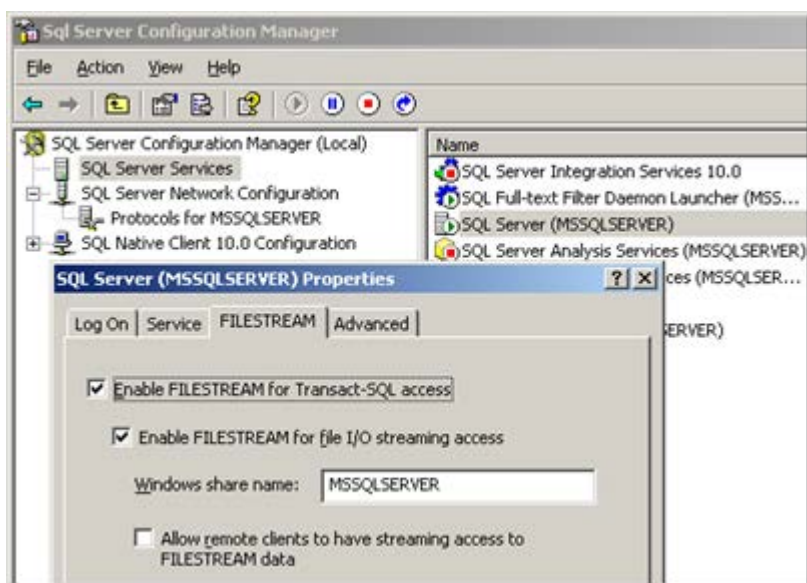
```
sp_configure 'filestream_access_level', 2;
reconfigure
```

Den fungerer fint, men vejledningen nedenunder, her siges at instansen ikke supporterer FILESTREAM, det gør tidligere versioner end 2008 ikke!!!???

FILESTREAM is a new feature of SQL Server 2008 that supports BLOB file storage on NTFS file shares managed by SQL Server. The FILESTREAM feature is disabled during the default SQL Server 2008 installation.

To enable FILESTREAM feature on SQL Server 2008:

1. Open SQL Server Configuration Manager (Start > Programs > Microsoft SQL Server 2008 > Configuration Tools > SQL Server Configuration Manager)
2. Navigate to the SQL Server Services node and select the SQL Server instance you want to modify (MSSQLSERVER)
3. Click the FILESTREAM tab and select the checkboxes to enable FILESTREAM and enter a share name for the files, as shown



Det samme findes under installationen, hvor FILESTREAM kan enables.

Det kan være nødvendigt at benytte SQL-kommandoen for enabling af FILESTREAM eller måske at

Lesson Review

1. Der spørges om hvad der skal gøres inden "FILESTREAM data can be stored". Det har historien ikke meldt noget om, men der menes nok kommandoen

```
EXEC sp_configure 'filestream_access_level', 2  
Reconfigure
```

2. svaret er nok SQL Server Configuration Manager.

Lesson 4: Configuring Database Mail

Database mail bruges fortrinsvis til at sende alerts til database-administrator om performance tilstande eller ændringer, der er sket på objekter.

Meddelelser i mails logges i Msdb databasen og styres af SQL Server Agent. Hvis denne ikke kører akkumuleres mailene i Msdb.

En mail-profil kan indeholde flere email-konti, sådan at der kan sendes til en anden hvis en mail-server ikke er til rådighed.

En profil kan være private, så kan andre brugere kun anvende den hvis de har fået granted access til den, eller den kan være public, så kan alle bruge den.

En profil kan sættes til at være default, så bruges den hvis der ikke er angivet en profil, når der sendes mail.

Practice 1 Configuring Database Mail

Udfør kommandoen:

```
exec sp_configure 'Database Mail XPs', 1;
go
Reconfigure with override;
go
```

Der kom en fejl om at 'Database Mail XPs' ikke var en eksisterende parameter eller at den var advanced

```
SELECT * FROM sys.configurations
ORDER BY name ;
go
```

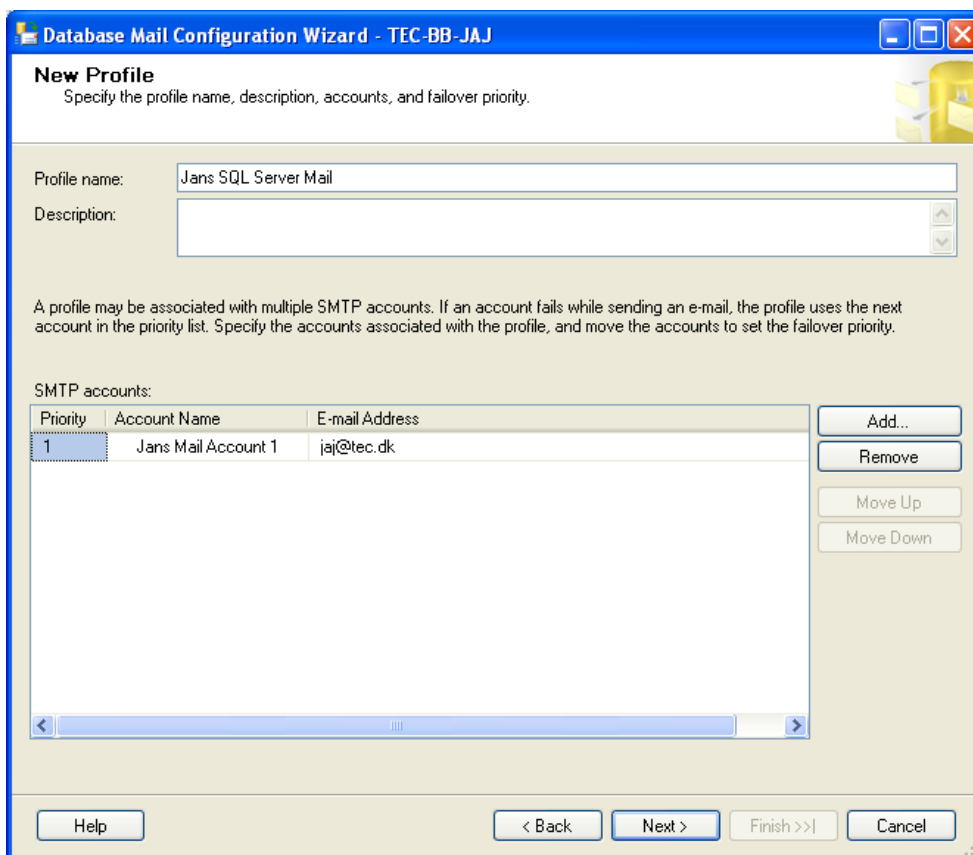
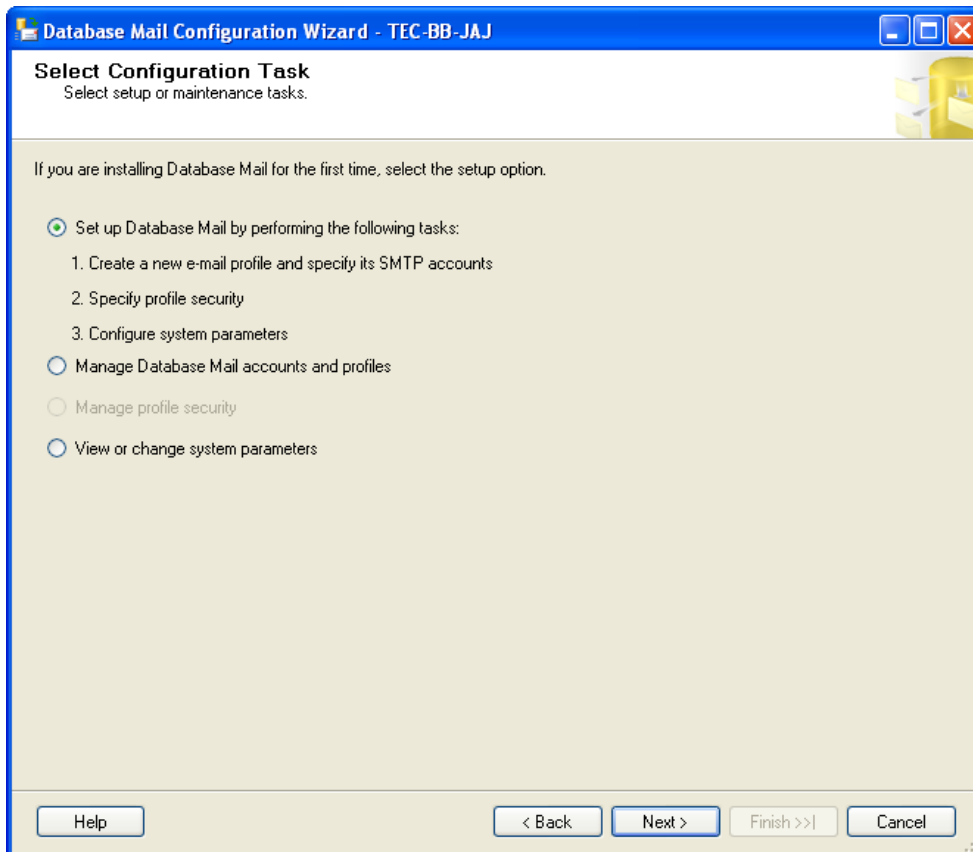
Her ses at det er en advanced. Så gør følgende:

```
exec sp_configure 'show advanced options', 1;
go
Reconfigure;
go

exec sp_configure 'Database Mail XPs', 1;
go
Reconfigure with override;
go
```

Så gik det godt.

Herefter højre-klikkes i Management | Database Mail. Vælg Configure Database Mail.



The screenshot shows the 'Database Mail Configuration Wizard - Manage Existing Account' window. The title bar reads 'Database Mail Configuration Wizard - TEC-BB-JAJUANSSQLSERVER'. The main heading is 'Manage Existing Account' with the instruction 'Choose the account to view, change, or delete.' Below this, the 'Account name' is set to 'Jans Sql Server mail' in a dropdown menu, with a 'Delete' button to its right. The 'Description' field is empty. Under the 'Outgoing mail server (SMTP)' section, the 'E-mail address' is 'jan@datamek.dk', 'Display name' is 'Jan SQL Server', 'Reply e-mail' is 'jaj@tec.dk', and 'Server name' is 'mail.datamek.dk'. The 'Port number' is set to '25'. There is an unchecked checkbox for 'This server requires a secure connection (SSL)'. The 'SMTP Authentication' section has three options: 'Windows Authentication using Database Engine service credentials' (unchecked), 'Basic authentication' (checked), and 'Anonymous authentication' (unchecked). Under 'Basic authentication', the 'User name' is 'jan', and both 'Password' and 'Confirm password' fields are masked with 'XXXXXX'. At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>|', and 'Cancel'.

Mail adresser på live.dk, hotmail.dk og msn.dk kan benytte smtp.live.com.
Gmail kan benytte smtp.gmail.com.

Vær opmærksom på SSL og at password skal genskrives hver gang man går ind i konfigurationen.

Send en mail

Ved hjælp af Stored Procedure `sp_send_dbmail` kan der sendes en mail, når der er oprettet en Database Mail Profile.

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name='JansDBMailProfile',
@recipients='jaj@tec.dk',
@subject='Test message',
@body='This is the body of the test message.
Congrates Database Mail Received By you Successfully.'
```

En trigger der sender en mail når der indsættes en række

```
create trigger trgFilmMail
on Film
for Insert
as
begin
    declare @Titel varchar(25)
    declare @Beskrivelse varchar(100)
    declare @MySubject varchar(40)
    declare @MyBody varchar(100)
    select @Titel=Titel, @Beskrivelse=Beskrivelse from Inserted
    set @MySubject = 'Ny Film: ' + @Titel
    set @MyBody = 'Beskrivelse af filmen ' + @Titel + ': ' + @Beskrivelse

    EXEC msdb.dbo.sp_send_dbmail
    @profile_name='JansDBMailProfile',
    @recipients='jaj@tec.dk',
    @subject = @MySubject,
    @body= @MyBody
end

go

insert into Film
values(7, 'Adams Æbler', 'En syret film om en psykopatisk ....', 2005)
```

Øvelser fra MS Official Course 6231A, Lab 1, Exercise 2

Task1:

Konfigurer SQL Server og Agent services til at starte automatisk. Der bruges SQL Server Configuration Manager og/eller Windows Tjenester.

Task2:

Opret et T-SQL script der begrænser antal connections til 10.

```
EXEC sp_configure 'show advanced options', '1';
GO
RECONFIGURE;
GO
EXEC sp_configure 'user connections', '10';
GO
RECONFIGURE;
GO
```

Task3:

Opret et SQL script som lister alle tabeller og de schema'er de ligger i, i den aktuelle database.

```
SELECT SCHEMA_NAME(schema_id) as SchemaName, name as Tablename
FROM sys.tables
```

Kap 2. Database Configuration and Maintenance

Lesson 1: Configuring Files and Filegroups

Files and Filegroups

Databaser består af to slags filer, data- og log-filer.

Datafiler tilhører Filegroups, som er et abstrakt begreb, som databaseobjekter kan lægges i. Create table, index og indexed views, har en ON klausul, hvor der angives hvilken filegroup objektet skal lægges i. Hvis den aktuelle filegroup har flere datafiler, vil de data, der lægges i database-objektet, blive fordelt i filerne, sådan at filerne bliver fulde samtidig. Dvs. at til to lige store filer, vil der løbende skrives lige meget data. Hvis de to filer ligger på hvert sit drev, vil trafikken mht. det aktuelle database-objekt, være ligeligt fordelt på de to drev.

```
create table MyTable
(
    Id int,
    Name varchar(25)
)
ON Filegroup1
```

Databehandling sker i en database-buffer, så inden der kan læses, ændres eller skrives data, må data læses fra datafil og op i memory. Når data i memory ændres, skrives der øjeblikkeligt til transactional-log-fil. Når der er brug for plads i memory, skrives der til datafil. Data der "mindst senest" er brugt, skrives til disk.

Bogen anbefaler at en ny database har følgende filer og filegroups:

En start-datafil, som også kaldes Primary File med extension .mdf i Primary filegroup. Denne datafil vil indeholde alle metadata. Den skal have en størrelse som modsvarer databasen Model. En nyinstalleret SQL Server 2008, hvor databasen Model ikke er ændret, kræver 3MB. En ekstra datafil med extension .ndf i MyFileGroup, som gøres Default, hvorved datafilen vil indeholde alle brugerdata, med mindre der angives en anden filegroup i oprettelsen af data-objekter.

En transactional log fil.

Disse tre filer bør ligge på hver sin disk.

Der kan så efter behov tilføjes flere datafiler og filegroups for at sprede trafikken yderligere på flere diske.

Kommandoen der sætter en filegroup til default:

```
Alter Database <db> modify filegroup <fg> default
```

Transaction Logs

Alle ændringer på data i en database registreres direkte i Transaction Log på disken, sådan at ændringerne kan gengøres hvis der sker noget med data.

Det er vigtigt disken med log-filen er hurtig, da databasen ikke kan ændres hurtigere end ændringerne registreres.

Selve de data-pages, der ændres i, skrives til disk efter behov eller ved CHECKPOINT.

Filestream

Filestream er nyt i SQL Server 2008. Dette betyder at et felt i en tabel kan være af typen Filestream, hvilket betyder at indholdet i feltet gemmes som en fil i operativsystemet. Dette kræver at der oprettes en filegroup, som har property CONTAINS FILEGROUP, hvilket gør at der oprettes en fil filestream.hdr, som bruges til at holde styr på de filer, der efterfølgende oprettes med data i.

Databasen TempDB

TempDB indeholder alle temporære tabeller og stored procedures, der benyttes af SQL Server. Den bruges også til temporære data, når der skal sorteres, grupperes, benyttes cursors og oprettes indexes.

Hver gang serveren startes op, oprettes der en ny TempDB, og der skal aldrig gemmes noget fra denne database, når serveren stoppes.

Der står i bogen at tempdb bruges meget mere end i tidligere version af SQL Server. Derfor kan man have brug for at lægge den på en disk for sig og evt. tilføje flere filer for at sprede den over flere diske.

Andre system-databaser

Master

Master databasen registrerer al system-level information for SQL Serveren. Dette inkluderer login konti, endpoints, linked servers og konfigurationer af systemet. Den holder også styr på alle bruger-databaser på serveren.

Serveren kan ikke starte op uden master databasen.

Der bør ikke oprettes bruger-database-objekter i master databasen. Så alle scripts bør indeholde kommandoen Use <database> på relevante steder, og der bør sørges for at den ønskede database er valgt når der i øvrigt udføres kommandoer, der opretter database-objekter.

Model

Model databasen danner template for alle oprettede databaser. Dens værdier i alle database-properties samt dens database-objekter vil findes i enhver ny database. Dette gælder også database-users, som måtte være oprettet i den.

Msdb

Msdb bruges af SQL Server Agent, Database Mail, Service Broker og andre services. Den registrerer automatiske jobs, alerts, data transformation packages og log shipping.

I msdb.dbo.backupset kan ses backup historik.

Practice 1 Creating Databases (44)

Følgende script udføres for at oprette en ny database.

Scriptet skal udføres i databasen Master.

Der er en PRIMARY filegroup til databasens oprettelse,
en filegroup fg1 til brugerdata, som sættes til default,
en filegroup Documents til filestream,

samt en logfil.

Angivelserne af filsti og filnavn må ikke indeholde linieskift. Er her vist wrapped.
Ret stien til så det passer på den aktuelle installation.

```
CREATE DATABASE TK432 ON
PRIMARY
(name = N'TK432_Data', filename = N'C:\Programmer\Microsoft SQL Server
\MSSQL10.JANSSQLSERVER\MSSQL\DATA\TK432.mdf',
size = 8MB, maxsize = unlimited, filegrowth = 16MB),

FILEGROUP fg1
( name = N'TK432_Data2', filename = N'C:\Programmer\Microsoft SQL Server
\MSSQL10.JANSSQLSERVER\MSSQL\DATA\TK432.ndf',
size = 8MB , maxsize = unlimited, filegrowth = 16MB),

FILEGROUP Documents CONTAINS FILESTREAM DEFAULT
(name = N'Documents', filename = N'C:\Programmer\Microsoft SQL Server
\MSSQL10.JANSSQLSERVER\MSSQL\DATA\TK432Documents')

LOG ON
(NAME = N'TK432_Log', FILEName = N'C:\Programmer\Microsoft SQL Server
\MSSQL10.JANSSQLSERVER\MSSQL\DATA\TK432.ldf' ,
size = 8MB, MAXSIZE = 2048GB, FILEGROWTH = 16MB)

go

ALTER DATABASE TK432
MODIFY FILEGROUP fg1
DEFAULT

go
```

Det foranstående N ved teksterne angiver at de er i Unicode. N står for National language character set. Dette betyder at der er angivet en NCHAR, NVARCHAR eller NTEXT type i stedet for CHAR, VARCHAR eller TEXT.

Hvis der kommer fejlmeddelelsen at SQL Serveren ikke har enabled FILESTREAM, selv om FILESTREAM er klikket af i egenskaber for SQL Server servicen, kan det være nødvendigt at udføre den stored procedure som udfører enablingen af FILESTREAM.

Hvis der kommer fejlmeddelelse på, at der er adgang nægtet for at oprette mappen TK432Documents, er det løsningen måske at sætte en anden konto på Sql Server service. En ganske simpel lokal bruger, som ikke har fået specielle rettigheder.

Lesson 2: Configuring Database Options

Database Options

- Recovery
- Auto options
- Change tracking
- Access
- Parameterization

Recovery Options

Full Recovery Model

Alt skrives i transaction log. Derfor kan der foretages recovery til point in time.

Bulk-logged Recovery Model

Bulk indsættelser i database logges kun bulk-vis. Derfor kan der ikke foretages recovery til point in time under en sådan indsættelse.

Det anbefales at skifte til Bulk-logged recovery model, når der skal bulk-loades, og umiddelbart efter skifte tilbage til Full recovery model, da der kan ske fejl.

Operationerne er:

- BCP
- Bulk insert
- Select ... into
- Create index
- Alter Index ... Rebuild

Simple Recovery Model

Der kan ikke udføres backup på transactional log-filer. Derfor kan der kun restores til seneste Full/Differential backup. Alt registreres i log-fil, men det er kun det der er sket siden sidste checkpoint, der kan bruges. Dette bruges til recovery ved genstart af server, samt rollback og readconsistency.

SQL kommando til ændring af Recovery Model:

```
alter database TK432  
set recovery {FULL | Bulk_logged | SIMPLE}
```

| Recovery | Backup Type | | | |
|----------|-------------|------|--------------|------------------------|
| | | Full | Differential | Transaction Log |
| | Full | Ja | Ja | Ja |
| | Bulk | Ja | Ja | Ja / nej, bulk minimal |

| | | | | |
|--------------|---------------|----|----|--------|
| Model | | | | logged |
| | Simple | Ja | Ja | Nej |

Damaged Pages

En database page er den mindste enhed, der kan flyttes imellem disk og databasebuffer i memory, og er på 8 kbyte.

Når der skrives en sådan til disk, kan der ske fejl, så den bliver korrupted. Derfor skrives der i dens header en checksum. Denne checksum sammenlignes med en beregnet checksum, når en sådan page læses igen. Så kan det ses om den er i orden.

En sådan fejl logges i Windows, og der indsættes en række i tabellen suspect_pages i databasen msdb.

I SQL Server 2005 kunne en sådan fejl kun repareres med restore af backup (behandles i kap. 9).

I SQL Server 2008 vil en database som er med i spejlings session, automatisk blive repareret med en page fra spejlingen. Der indsættes en log i view'et sys.dm_db_mirroring_auto_page_repair.

Ovenstående enables af PAGE_VERIFY CHECKSUM option.

Auto Options

Kommer senere.

Change Tracking

Handler om at der kan ændres i en række på disken af en bruger imellem at en anden bruger har læst den op i memory-buffer og at den skrives tilbage på disken.

Hvordan nu lige det. Bruger alle ikke den samme database-buffer???

Dette styres af et versionsnummer som bliver forøget hver gang en række ændres.

Access til databasen

Den normale mode, en database er i, er ONLINE, READ_WRITE og MULTIUSER.

Disse kan ændres og mulighederne er:

Status:

- ONLINE
- OFFLINE
- EMERGENCY

Ability to modify data:

- READ_ONLY
- READ_WRITE

User access:

- SINGLE_USER
- RESTRICTED_USER
- MULTI_USER

Alle disse kan ændres med kommandoen ALTER DATABASE.

Eksempel hvor alle brugere smides af og deres transaktioner rulles tilbage:

```
alter database TK432
set restricted_user
with rollback immediate
```

Og så tilbage til det normale:

```
alter database TK432
set multi_user
```

Parameterization

Når der i f.eks. C# oprettes en SQL-kommando, kan feltværdierne angives som parametre. Når en sådan kommando udføres flere gange, men med forskellige værdier, vil database-serveren benytte samme udførelsesplan og derved spare tid til at beregne planen.

Database option PARAMETERIZATION FORCED, kan åbenbart/måske??? sørge for at den selv opretter disse parametre, så der kan genbruges, selv om det ikke sker fra brugerens/programmets side.

Collation Sequences

Når en SQL Server installeres, kan der vælges hvilken collation den skal køre med. Dette kan ændres i den enkelte database i CREATE DATABASE og i ALTER DATABASE.

Brug af Windows collation definitioner:

```
alter database TK432
collate Danish_Greenlandic_100_CI_AS
```

Brug af SQL Server collation definitioner:

```
alter database TK432
collate SQL_Scandinavian_Cp850_CS_AS
```

Create database

```
create database TK432
```

```
...  
...  
...  
collate SQL_Scandinavian_Cp850_CS_AS
```

Der kan desuden defineres collation på field-niveau i en tabel, hvor typen er karakter-string.
Se i øvrigt side 11

Practice 1 Changing the Database Recovery Model (52)

```
alter database AdventureWorks  
set recovery FULL
```

Lesson 3: Maintaining Database Integrity

Database Integrity Checks

En datapage på 8 kbyte deles op i 16 * 512 bytes blokke. Nok for at operativsystemet har denne begrænsning når der læses og skrives til disk. Der kan ske den fejl, at ikke alle blokke bliver skrevet til disk.

Kommandoen DBCC checker integritet og udskriver fejl.

Practice 1 Checking Database Integrity (56)

```
dbcc checkdb ('TK432') with no_infomsgs, all_errormsgs
```

```
Command(s) completed successfully.
```

Så må alt jo være godt.

Kap 3. Tables

Lesson 1: Creating Tables

Tabeller er behandlet i SQL Undervisningsmateriale

Schemas

Et schema er et logisk område som database-objekter lægges i.

Hvis ikke andet er angivet, vil enhver bruger have schema dbo som default.

Der kan oprettes et nyt schema og gives til brugeren Jan. Men han vil stadig have dbo som default schema. Sådan at objekter lægges i schema dbo hvis de ikke kvalificeres med andet schema.

Brugeren Jan skal eksistere og kan så være logget ind på en anden Sql Server Management Studio.

Admin:

```
create schema JansSchema authorization Jan
```

Jan:

```
create table Navne(navn varchar(12))
```

```
insert into Navne values('Jan'), ('Rie')
```

```
select * from dbo.Navne
```

Output: Jan

Rie

Nu sættes JansSchema til at være brugeren Jan's default schema, og så lægges alt her, hvis der ikke kvalificeres med andet.

Admin:

```
alter user Jan with default_schema = JansSchema
```

Jan:

```
create table Navne(navn varchar(12))
```

```
insert into Navne values('Tob'), ('Kat')
```

```
select * from JansSchema.Navne
```

Output: Tob

Kat

Lesson 2: Implementing Constraints

Constraints er behandlet i SQL Undervisningsmateriale.

Ekstra: Brug af Filestream

For afprøve filestream udføres følgende.

I databasen TK432 er der jo oprettet en filegroup til at indeholde filestream.

Her oprettes en tabel Items hvor der skal gemmes jpg-billeder.

For at vise at det virker hentes billederne ud af tabellen og vises på skærmen i et C#-program.

Tabellen Items oprettes med en filestream kolonne. Dette kræver at der også oprettes en kolonne af typen ROWGUIDCOL, som formentlig giver hver række en unik id på hele SQL Serveren.

```
CREATE TABLE Items
(
    ItemID UNIQUEIDENTIFIER ROWGUIDCOL NOT NULL UNIQUE,
    ItemNumber VARCHAR(20),
    ItemDescription VARCHAR(50),
    ItemImage VARBINARY(MAX) FILESTREAM NULL
)
```

Billedet hentes ind i variablen @img

```
DECLARE @img AS VARBINARY(MAX)

SELECT @img = CAST(bulkcolumn AS VARBINARY(MAX))

    FROM OPENROWSET(BULK 'C:\Billeder\Einstein1.jpg',
        SINGLE_BLOB ) AS x
```

Og indsættes i tabellen

```
INSERT INTO Items (ItemID, ItemNumber, ItemDescription, ItemImage)

    SELECT NEWID(), '1', 'Einstein1', @img
```

Begge dele udføres i samme batch for at variablen @img kan leve.

Hvis dette gøres med filerne Einstein1.jpg, Einstein2.jpg, Einstein3.jpg og Einstein4.jpg, så er der i tabellen følgende.

```
select * from Items
```

Results

Messages

| | ItemID | ItemNumber | ItemDescription | ItemImage |
|---|--------------------------------------|------------|-----------------|---|
| 1 | B15472E2-C805-44BE-BFA3-FDA7D862CFE3 | 1 | Einstein1 | 0xFFD8FFE000104A46494600010100000100010000FFDB00... |
| 2 | 9308DA59-26DF-4BCC-9230-ECB4BA584FF | 2 | Einstein2 | 0xFFD8FFE000104A46494600010100000100010000FFDB00... |
| 3 | 75B9176D-A4C6-4280-AE84-C6F634AB1B7E | 3 | Einstein3 | 0xFFD8FFE000104A46494600010100000100010000FFDB00... |
| 4 | E4515A53-EF51-41C5-BE9F-0F0826822480 | 4 | Einstein4 | 0xFFD8FFE000104A46494600010100000100010000FFDB00... |

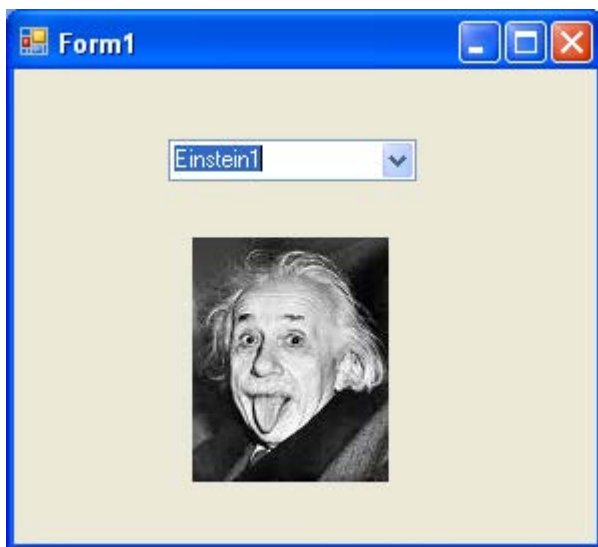
I følgende mappe ligger filen filestream.hdr, fordi TK321 databasen har en filegroup der kan indeholde filestream.

C:\Programmer\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\TK432Documents

I en undermappe til denne ligger der nu 4 filer, hver med databasens indhold af de 4 jpg-filerne.

C#-koden

I programmet oprettes en listboks med navnene på billederne og ved valg af et billede hentes informationen i filestream feltet i den aktuelle række i tabellen og konverteres til et billede som vises på skærmen.



Koden ses på næste side.

I Form1_Load() indsættes et item i listboksen for hver billede, så der kan vælges blandt dem.

```
public partial class Form1 : Form
{
    string conStr = "Data Source=.;Initial Catalog=TK432;"
        + "Integrated Security=True";

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection(conStr);
        SqlCommand cmd = new SqlCommand("select * from Items", con);
        try { con.Open(); } catch{}
        SqlDataReader result = cmd.ExecuteReader();
        while (result.Read())
        {
            ListItem li = new ListItem();
            li.value = result["ItemNumber"].ToString();
            li.text = result["ItemDescription"].ToString();
            lstImages.Items.Add(li);
        }
        try { con.Close(); }
        catch { }
    }
}
```

Brugeren vælger et billede i listboksen og i dens eventhandler SelectedIndexChanged() hentes det valgte billedes information fra tabellen og konverteres til et Image-objekt, der vises på skærm.

```
private void lstImages_SelectedIndexChanged(object sender, EventArgs e)
{
    string selected = ((ListItem)lstImages.SelectedItem).value;
    string sqlStr =
        "Select ItemImage from Items where ItemNumber = " + selected;
    SqlConnection con = new SqlConnection(conStr);
    SqlCommand cmd = new SqlCommand(sqlStr, con);
    try { con.Open(); }
    catch { }

    byte[] byteArray = (byte[])cmd.ExecuteScalar();
    Image im = Image.FromStream(new MemoryStream(byteArray));
    pictureBox1.Image = im;

    try { con.Close(); }
    catch { }
}
}
```

Kap 4. Designing SQL Server Indexes

Når der oprettes en primary key på en tabel, oprettes der en index af samme navn.

```
create table navne
(
    navn varchar(12) constraint navne_navn_PK primary key
)

select * from sys.indexes where name = 'navne_navn_PK'
```

Men der kan også oprettes andre indexes med kommandoen create index.

Lesson 1: Index Architecture

Et index er opbygget som en B-Tree. Hver node er en data-page og har lige så mange børn, som der er plads til index-rows i en data-page.

Flaskehalsen i en database er IO-trafikken til disk. Derfor er det vigtigt ikke at skulle indlæse flere data-pages end højest nødvendigt. Når først en data-page er indlæst, er det hurtigt at søge data igennem.

Et tænkt eksempel.

En tabel har 1.000.000 rækker på hver 800 bytes. I en data-page kan der være $8k/800 = 10$ rækker. Tabellen fylder således $1.000.000/10 = 100.000$ data-pages.

Hvis der udføres en "full table scan", dvs at der søges efter data uden at bruge index, vil alle data-pages skulle indlæses fra disk.

En række i en index på den aktuelle tabel skal principelt indeholde den kolonne, der er indekseret samt en refernce til adressen på den aktuelle række i tabellen. Dette tænkes at kunne være i en række af størrelsen 80 bytes.

En data-page i index'et kan altså indeholde $8k/80 = 100$ rækker.

Derfor vil hver node/data-page i index'et have 100 børn.

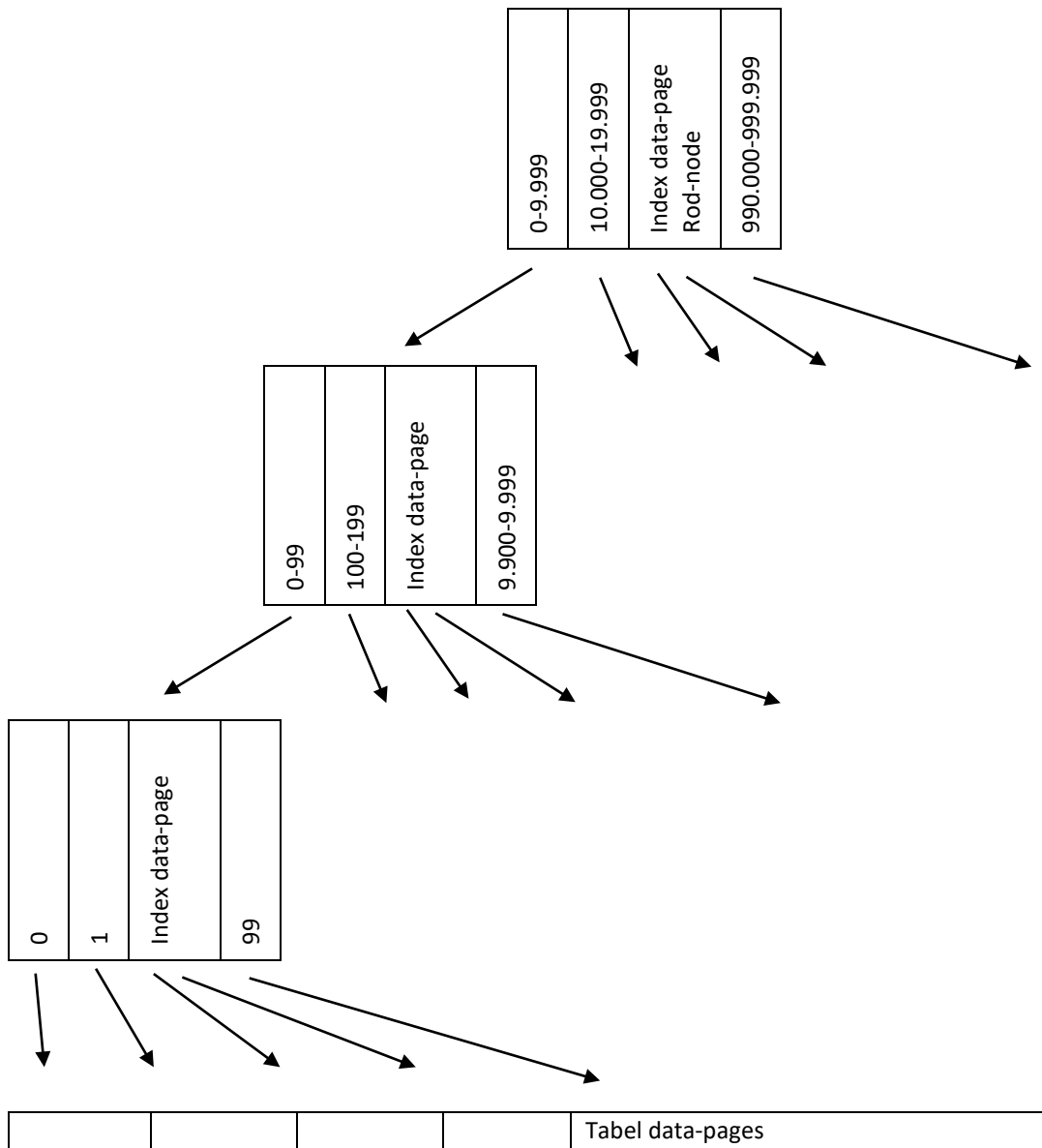
Rod-noden peger altså på 100 noder, som peger på $100*100 = 10.000$ noder, som peger på $100*100*100 = 1.000.000$ noder, som hver især peger på den fysiske data.

Der skal altså indlæses 3 data-pages af index'et, et i hvert niveau, for at finde ud af hvilken data-page fra tabellen, der skal indlæses.

Altså skal der indlæses 4 data-pages i alt for at finde en bestemt række i tabellen, mod 100.000 data-pages for at læse hele tabellen.

Se tegning på næste side.

Opbygningen af et index.



Lesson 2: Designing Indexes

Indexes kan være clustered eller non-clustered.

I en non-clustered index er der kun tabel-data fra kolonnen som er index'eret.

I en clustered index er alle data fra tabellen indeholdt i selve index'et.

Practice: Creating Indexes (side 102)

Lesson 3: Maintaining Indexes

Practice: Maintaining Indexes (side 106)

Kap 5. Full Text Indexing

Lesson 1: Creating and Populating Full Text Indexes

Practice: Creating Full Text Indexes

Practice 1: Create a Full Text Catalog

```
select * from AdventureWorks.Production.ProductDescription

alter database AdventureWorks
add filegroup AWFullTextFG
go

alter database AdventureWorks
add file(name=N'S AdventureWorks',
filename = N'C:\Programmer\Microsoft SQL Server\MSSQL10.MSSQLSERVER'
          + '\MSSQL\DATA\AdventureWorksFT.ndf')
to filegroup AWFullTextFG
go

use AdventureWorks
go

create fulltext catalog ProductsFTC
on filegroup AWFullTextFG
go
```

Practice 2: Create Full Text Index

```
select * from INFORMATION_SCHEMA.TABLE_CONSTRAINTS
where TABLE_NAME = 'ProductDescription'

CREATE FULLTEXT INDEX ON Production.ProductDescription(Description)
KEY INDEX PK_ProductDescription_ProductDescriptionID
ON ProductsFTC
WITH CHANGE_Tracking = AUTO
go
```

Lesson 2: Querying Full Text Data

Practice: Querying with a Full Text Index

```
SELECT Description
FROM Production.ProductDescription
go

SELECT ProductDescriptionID, Description
FROM Production.ProductDescription
WHERE FREETEXT(Description, N'bike')
go
```

```
SELECT a.ProductDescriptionID, a.Description, b.*
FROM Production.ProductDescription a
     INNER JOIN FREETEXTTABLE(Production.ProductDescription,
                             Description, N'bike') b ON a.ProductDescriptionID = b.[Key]
ORDER BY b.[Rank]
```

go

```
SELECT ProductDescriptionID, Description
FROM Production.ProductDescription
WHERE CONTAINS(Description, N'bike')
go
```

```
SELECT ProductDescriptionID, Description
FROM Production.ProductDescription
WHERE CONTAINS(Description, N'"bike*"')
go
```

Der er nogle flere forespørgsler i øvelsen. De skal alle analyseres lidt mere.

Lesson 3: Managing Full Text Indexes

Kap 6. Distributing and Partitioning Data

Ved at partitionere en tabel, kan dens indhold distribueres over flere filegroups/data-filer og dermed over flere diske.

Dette vil gøre at der hurtigt kan skrives data i en tabel.

I en partitioneret tabel opdeles en kolonne i forskellige områder/partitions, og værdierne for denne kolonne styrer så i hvilke filegroups rækkerne skal gemmes.

For at gøre dette må der oprettes en database med flere filegroups.

```
create database TestPartitionDB
on primary
    (name = primary_data, size = 3MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\TestPartition.mdf'),

filegroup fg1
    (name = fg1_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\Fg01.ndf'),

filegroup fg2
    (name = fg2_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\Fg02.ndf'),

filegroup fg3
    (name = fg3_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\Fg03.ndf'),

filegroup fg4
    (name = fg4_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\Fg04.ndf')

log on
    (name = db_log, size = 2MB, filegrowth = 10%,
     filename = 'D:\SQL Server 2008 Test\Kap 6a\TestPartitions_log.ldf');
```

Lesson 1: Creating a Partition Function

De områder/partitions en kolonne i en tabel opdeles styres af en såkaldt Partition Function, som har til opgave at definere grænseværdierne for områderne.

De angivne værdier er maksimumværdierne for hvert område.

Angivelsen 'right' betyder at områderne er inclusive max-værdierne. Ved at angive 'left' bliver det exclusive max-værdierne.

Når der defineres 3 grænseværdier, specificerer dette 4 partitions: Alle værdier under 10 inklusive negative, værdierne 10-20, værdierne 20-30 og alle værdier over 30.

```
use TestPartitionDB

create partition function
TestPartFunction(int)
as range right
for values(10,20,30)
```

Med følgende kommando kan de definerede partition områder ses.

```
select * from sys.partition_range_values
```

Lesson 2: Creating Partition Scheme

En Partition Scheme er en slags skema, hvor partitionerede tabeller og indexes kan lægges, og derved knyttes sammen med definitionerne i Partition Functions og der til hørende filegroups.

I oprettelsen af en Partition Schema knyttes netop partition-grænseværdier sammen med de filegroups, som data skal fordeles i.

```
create partition scheme
TestPartScheme
as partition TestPartFunction
to(fg1, fg2, fg3, fg4)
```

Med følgende kommando kan eksisterende Partition Schemes ses

```
select * from sys.partition_schemes
```

Lesson 3: Creating Partitioned Tables and Indexes

En Partitioned Table oprettes som andre tabeller med tilføjelsen 'on TestPartScheme(Id)', der lægger den i en Partitioned Scheme og angiver hvilket felt i tabellen, der skal partitioneres på. Dette felt skal være af samme type som angivet i den tilhørende Partitioned Function. Her typen int.

```
create table TestPartTable(
    Id int,
    FirstName varchar(20),
    LastName varchar(20)
)
on TestPartScheme(Id)
```

Der indsættes nu 49 rækker i tabellen. Rækkerne hentes fra en anden tabel i databasen AdventureWorks. Kolonnen Id får værdierne 1-49.

```
insert into TestPartTable
select BusinessEntityId, FirstName, LastName
```



```
from AdventureWorks2008.Person.Person
where BusinessEntityID < 50
```

Eller hvis databasen AdventureWorks2008 ikke findes.

```
insert into TestPartTable
select ContactID, FirstName, LastName
from AdventureWorks.Person.Contact
where ContactId < 50
```

I hvilke partitions rækker bliver lagt kan ses med følgende kommando.

```
select partition_number, rows
from sys.partitions
where object_id = object_id('TestPartTable')
```

| Partition_number | Rows |
|------------------|------|
| 1 | 9 |
| 2 | 10 |
| 3 | 10 |
| 4 | 20 |

Hvis 'right' ændres til 'left' i Partitioned Function vil resultatet blive følgende.

| Partition_number | Rows |
|------------------|------|
| 1 | 10 |
| 2 | 10 |
| 3 | 10 |
| 4 | 19 |

Practice: Creating a Partitioned Function, Partition Scheme, Partition Table and Indexes (side 140, 144 og 146)

Her følger den samlede kode.

Opret database med 13 ekstra filegroups.

```
create database Partitiontest
on primary
    (name = primary_data, size = 3MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\partitiontest.mdf'),

filegroup fg1
    (name = fg1_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg01.ndf'),
filegroup fg2
    (name = fg2_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg02.ndf'),
filegroup fg3
    (name = fg3_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg03.ndf'),
filegroup fg4
    (name = fg4_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg04.ndf'),
filegroup fg5
    (name = fg5_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg05.ndf'),
filegroup fg6
    (name = fg6_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg06.ndf'),
filegroup fg7
    (name = fg7_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg07.ndf'),
filegroup fg8
    (name = fg8_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg08.ndf'),
filegroup fg9
    (name = fg9_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg09.ndf'),
filegroup fg10
    (name = fg10_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg10.ndf'),
filegroup fg11
    (name = fg11_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg11.ndf'),
filegroup fg12
    (name = fg12_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg12.ndf'),
filegroup fg13
    (name = fg13_data, size = 2MB,
     filename = 'D:\SQL Server 2008 Test\Kap 6\fg13.ndf')

log on
    (name = db_log, size = 2MB, filegrowth = 10%,
     filename = 'D:\SQL Server 2008 Test\Kap 6\partitiontest_log.ldf');
```

Følgende skal udføres i den nye database PartitionTest.

Opret Partitioned Function.

```
create partition function partfunc(datetime) as
range right for values('2009-1-1', '2009-2-1', '2009-3-1', '2009-4-1',
                      '2009-5-1', '2009-6-1', '2009-7-1', '2009-8-1',
                      '2009-9-1', '2009-10-1', '2009-11-1', '2009-12-1')

select * from sys.partition_range_values
```

Opret Partitioned Schema.

```
create partition scheme partscheme as
partition partfunc to
(fg1, fg2, fg3, fg4, fg5, fg6, fg7, fg8, fg9, fg10, fg11, fg12, fg13)
go

select * from sys.partition_schemes
```

Opret tabel.

```
create table dbo.orders(
    OrderID int identity(1,1),
    OrderDate datetime not null,
    orderAmount money not null,
    constraint pk_orders primary key clustered (OrderDate, OrderID))
on partscheme(OrderDate)
```

Fyld tabellen med nogle datoer.

```
set nocount on
declare @month int, @day int
set @month = 1
set @day = 1

while @month <= 12
begin
    while @day <= 28
    begin
        insert dbo.orders(OrderDate, OrderAmount)
        select CAST(@month as varchar(2)) + '/' + CAST(@day as varchar(2))
        + '/2009', @day * 20

        set @day = @day + 1
    end
    set @day = 1
    set @month = @month + 1
end

go

select * from sys.partitions
where object_id = object_id('dbo.orders')
```

Lesson 4: Managing Partitions

Kommer senere.

Kap 7. Importing and Exporting Data

SQLCMD

(Hører egentlig ikke til her, men her er der brug for den)

SQLCMD er en kommandolinie-SQL-fortolker som kan udføres i en command-prompt.

SQLCMD er afløseren for Osql, som dog stadig findes.

2005 OK. Øvelsen laves med SQLCMD.

Når der åbnes en SQLCMD-prompt udføres angivne sql-kommandorer med 'go'.

Hjælp til syntaks: sqlcmd -?

-L: list servers

-S: specificerer hvilken instans(SQL Server) der skal forbindes til

-E: specificerer Windows authentication

-U: brugernavn

-P: password

-q: angiver at der kommer en efterfølgende SQL-kommando i anførselstegn, som skal udføres

-i: specificerer hvilken sql-script der skal udføres

-o: specificerer hvilken fil som output skal skrives i

-v: specificerer parametre til en sql-script

-A: DAC (Dedicated Administrator Connection) er en connection som kan være mulig når alt andet svigter

Følgende starter sqlcmd og connecter til default server. -E er default og kan udelades.

@servername er en supervariabel, der indeholder den aktuelle servers navn.

```
C:\>sqlcmd -E
```

```
1> select @@servername
```

```
2> go
```

```
-----  
MyLaptop
```

Følgende starter sqlcmd og connecter til den angivne server.

```
C:\>sqlcmd -S <pc-navn>\MyNamedInstance (kunne også være <pc-navn>\sqlexpress)
```

```
1> use northwind
```

```
2> go
```

```
Changed database context to 'Northwind'.
```

```
1> select name from sys.database_files
```

```
2> go
```

name

Northwind

Northwind_log

Eks. på afvikling af SQL-script og output gemmes i en tekstfil.

C:\>sqlcmd -E -i sqlquery1.sql -o tekst.fil

Opgaven i bogen indeholder:

sqlcmd -E

- select @@servername
- select @@version
- go
- exit

sqlcmd -L

- Viser alle servere.
-

sqlcmd -E -S <pc-navn>\MyNamedInstance

- select system_user (Windows-brugeren)
- go
- quit

At udføre en enkelt SQL-kommando med sqlcmd.

sqlcmd -Q "select * from AdventureWorks.HumanResources.Department"

-Q afslutter sqlcmd, -q forbliver i sqlcmd.

Windows Command Batch Fil

Med en .bat fil kan der udføres sql-scripts ved hjælp af SqlCmd.

CreateFilm.bat indeholder f.eks.:

```
@ECHO OFF
```

```
echo 'Tryk på en tast for at installere Film-database'
pause
```

```
sqlcmd -i FilmDB.sql
```

```
echo 'Nu er Film-databasen oprettet. Tryk på en tast'  
pause
```

Med switchene -U og -P kan scripts udføres med en bestemt database-bruger.

Flytning og transformering af data

Oprettelse af script, der kopierer data

Hvis man er lidt fingernem, kan man konstruere et script ud af en select-sætning, som kan bruges til at indsætte data fra en tabel i en anden tabel.

Hvis tabellen Person indeholder følgende data

| Id | Navn | Adr | Født |
|----|--------|-------------------------------|------------|
| 1 | Niels | Nygade 2900 Happiness | 1980-03-19 |
| 13 | Morten | Lyngby Hovedgade 2800 Lyngby | 1980-03-19 |
| 14 | Adem | Tranemosevej 7, 2600 Glostrup | 1980-04-23 |

vil følgende kommando

```
select 'insert into Person2 values(' + cast(Id as varchar) + ', ' +  
      + Navn + ', ' +  
      + Adr + ', ' +  
      + cast(Født as varchar) + ') '  
from Person
```

give følgende resultat

```
insert into Person2 values(1,'Niels','Nygade 2900 Happiness','1980-03-19')  
insert into Person2 values(13,'Morten','Lyngby Hovedgade 2800 Lyngby','1980-03-19')  
insert into Person2 values(14,'Adem','Tranemosevej 7, 2600 Glostrup','1980-04-23')
```

som så kan udføres og lægge data ind i en eksisterende tabel Person2.

Resultatet kan gemmes som et script og udføres når der er brug for det.

Data fra en tabel til en anden tabel

Select ... into

Følgende kommando opretter tabellen Department2 og lægger alt data ind i den fra tabellen Department.

Den oprettede tabel har ingen constraints.

```
select DepartmentID, Name, GroupName  
into AdventureWorks.HumanResources.Department2  
from AdventureWorks.HumanResources.Department
```

Insert ... select

Følgende kommando kræver at tabellen er oprettet


```
create table AdventureWorks.HumanResources.Department3
(
    DepartmentId int,
    Name varchar(30),
    GroupName varchar(40)
)
```

```
Insert into AdventureWorks.HumanResources.Department3
Select DepartmentID, Name, GroupName
From AdventureWorks.HumanResources.Department
```

Lesson 1: Importing and Exporting Data

BCP

BCP er et kommandolinie-program, som udføres i en command-prompt.

Det er både muligt at exportere data fra en tabel eller data fra en sql-forespørgsel til en tekstfil, og at importere data fra en tekstfil til en tabel.

I de følgende kommandoer er benyttet switchen -c, som angiver at filen får almindelige ASCII-karakterer. Der kan i stedet benyttes switchen -n, som angiver at teksten skrives med den karakertype, som er i den aktuelle tabel, hvilket kan gøre teksten i filen med eksporteret data delvist ulæseligt.

BCP import (in) kræver som input en tekstfil, hvor der er sat apostroffer omkring hvert felt-værdi. Dette vil den også producere ved export (out) .

Switchen -w gør at danske karakterer kommer til at virke. Eller -C -c.

-T angiver Windows Authentication

-U og -P angiver user og password ved SQL-Server Authentication

Export af tabel

Følgende eksempel exporterer alle data i tabellen Department til filen Department.txt.

Tabellen Department findes på serveren LapTop\JansSqlServer, databasen AdventureWorks og schema HumanResources.

Mappen C:\DBtest skal findes i forvejen.

-T angiver Windows authentication, -t angiver field terminering og -r angiver row terminering.

Switchen S angiver server.

Filen bliver altså kommasepareret.

```
C:\>bcp AdventureWorks.HumanResources.Department
out C:\DBtest\department.txt -c -S LapTop\JansSqlServer -T -t ',' -r '\n'
```

Import af tabel

Hvis den tidligere tabel, som er exporteret til en fil, skal importeres igen, skal tabellen som data lægges i eksistere i forvejen.

For at få en tabel magen til den exporterede Department, som der ikke ønskes ændret i, kan der laves en kopi af denne med følgende SQL-kommando, som vist oven for:

```
select *  
into AdventureWorks.HumanResources.Department2  
from AdventureWorks.HumanResources.Department
```

Nu haves tabellen Department2 med samme datatyper og indhold, men uden constraints. Indholdet kan evt. deletes.

Med følgende kommando importeres tekstfilen til den nye tabel.

```
C:\>bcp AdventureWorks.HumanResources.Department2  
in c:\DBtest\department.txt -c -S LapTop\JansSqlServer -T -t ',' -r '\n'
```

Med switchen -e kan der angives en error-fil, hvor data med fejl lægges. Dvs. data som hentes ind fra en kommasepareret fil, og som ikke er korrekte, og dermed ikke kan indsættes i den aktuelle table. Denne fil kan så rettes til og derefter indsættes data fra denne.

Export af query

Med en SQL select kommando og nøgleordet queryout kan resultatet af forespørgslen exporteres til en fil:

```
C:\>bcp "select Empno, FName, MInit, LName from H2TestSql.dbo.Emp" queryout c:\D  
Btest\Emp.txt -c -S LapTop\JansSqlServer -T -t ',' -r '\n'
```

BULK Insert

BULK Insert er en SQL-kommando. Dette har den fordel frem for BCP at kommandoen kan stå i en SQL-script.

Ulempen ved BULK Insert er at der kun kan importeres data, men det er jo også hovedsagen når der udføres script for at oprette en database.

Følgende opretter en tabel til postnumre og dernæst fyldes tabellen fra en kommasepareret fil

```
create table H2SqlMaj2011.dbo.PostNrBy  
(  
    PostNr int,  
    Byen nvarchar(25),  
    constraint PostNrBy_PostNr_PK primary key(PostNr)  
)  
go
```

```
Bulk insert H2SqlMaj2011.dbo.PostNrBy
from 'C:\Data Data Data\Databaser\Postnumre\PostNr.txt'
with
(
    codepage = 'ACP',
    batchsize = 250,
    datafiletype = 'widechar' eller 'char' ,
    fieldterminator = ',',
    rowterminator = '\n',
    maxerrors = 50,
    tablock )
```

The SQL Server Import and Export Wizard

BCP og BULK Insert udgør simple værktøjer til at importere og exportere data.

SQL Server Import and Export Wizard er indbygget i SQL Server Management Studio (SSMS) og er en lille del af SQL Server Integrated Services (SSIS).

Modsat BCP og BULK Insert behøver SSIS ikke nødvendigvis at have data fra en database som input eller output. Der kan f. eks. flyttes data fra en kommasepareret fil til en fil af typen Excel-regneark. Der kan også arbejdes med flere filer eller tabeller samtidig.

Practice 1: Export Data Using BCP(side 167)

Øvelsen afprøver det første eksempel af BCP ovenfor med henholdsvis switchen -c og -n. Afprøv desuden de andre eksempler ovenfor, både BCP og BULK Insert.

Practice Postnumre:

1. Opret tabellen PostNrBy med kolonnerne PostNr og Byen.
2. Med Bulk Insert indsættes data i tabellen PostNrBy fra filen PostNr.txt
3. Med BCP exporteres indholdet i tabellen PostNrBy til filen PostNrBCP.txt.
Her skulle der nu være apostroffer omkring værdierne.
4. Opret tabellen PostNrByBCP med kolonnerne PostNr og Byen.
5. Med BCP importeres indholdet i filen PostNrBCP.txt til tabellen PostNrByBCP.
6. Opret sql-scriptet CreatePostNr.sql indeholdende kommandoen fra punkt 4, som opretter tabellen.
7. Opret bat-filen CreatePostNr.cmd indeholdende kommandoen til udførelse af scriptet CreatePostNr.sql og dernæst kommandoen i punkt 5, som indsætter data vha. BCP. Indsæt kommandoen pause i slutningen, så vinduet ikke lukkes.

Ved problemer med at bruge æøå kan det hjælpe at skrive "CHCP 1252" i begyndelsen af bat-filen.

Practice 2: Exporting Tables (side 167)

Øvelsen eksporterer alle tabellerne i databasen AdventureWorks over i en ny database AdventureWorksTest vha. Import and Export Wizard.

1. Opret databasen AdventureWorksTest
`create database AdventureWorksTest`
2. Højreklik på AdventureWorks og vælg Tasks og Export Data.
3. På wizardens velkomstsider klik Next.
4. På siden Choose a Data Source, vælg den aktuelle server, og vælg Database: AdventureWorks. Klik Next.
5. På siden Choose a Destination, vælg den aktuelle server, og vælg Database: AdventureWorksTest. Tryk Next.
6. På siden Specify Table Copy or Query, vælg Copy data from one or more tables or views. Tryk Next.
7. På siden Select Source Tables and Views, vælg alle tabeller fra databasen AdventureWorks.
8. Marker den øverste tabel AWBuildVersion og tryk på Edit Mappings.
9. Læg mærke til mulighederne for create af destinations-tabellen. Tryk Cancel og Next.
10. Her skulle der vise sig nogle tabeller med typer, der ikke supporteres af C#.NET, som er grundlaget for flytningen af data. Men der er ingen problemer.
11. De tabeller der måtte give problemer skal klikkes af igen, så de ikke er med i transformationen. Men der er ingen problemer.
12. På siden Save and Run Package, vælges Run Immediately og evt. Save SSIS Package i SQL Server. Klik Next.
13. I Save SSIS Package angives Name til CopyAdventureWorks og tryk Next.
14. På siden Perform Actions vælges Finish.
15. På siden Performing Operation . . . ses udførelsen. Det tager lidt tid, da det er en stor database. Tryk Close.
16. Den gemte package kan findes og udføres igen ved at connecte til Server Type: Integration Services og samme Servare Name som før. Her kan den findes under Stored Packages | MSDB. Højreklik på CopyAdventureWorks og vælg Run Package og Execute. (Måske varer det lidt tid inden den figurerer der efter at den er oprettet.)

Practice Extra: Her kommer så et rigtigt SSIS projekt udført i Visual Studio

Fra bogen: MS SQL Server 2005 Integration Services Step by Step

Kapitel 3

Øvelse Chap3A Creating a new Integration Services Project.

Projektet kopierer indholdet fra tabellen Employee ud i Excel-fil.

(Originalen arbejder med Adventure-databasen, men her er det ændret til vores kendte Employee tabel)

Opret Projekt

Start Visual Studio

Vælg File | New | Project | Integration Services Project, giv den navnet "Chap03A min DB" eller "H2SSISProjekt" og gem den et passende sted.

I Solution Explorer højre-klikkes på Package.dtsx og Rename til CopyTablePackage.dtsx. Svar Ja til også at ændre package-objektet.

Tilføj Connection Managers

Connection Manager til en SQL Server database

Med midtervinduet tab Control Flow valgt, højre-klik i vinduet Connection Managers nederst i midtervinduet og vælg New OLE DB Connection . . .

Klik på New for at oprette en ny connection.

Provider: NativeOLE DB\SQL Native Client.

Server Name: Skriv localhost.

Use Windows Authentication.

Select Database name: H2Company (hvor tabellen Employee skal være)

Klik Test Connection og tryk OK og OK.

Connection Manager til Excel-fil

Opret en mappe Data under projektmappen "Chap03A min DB".

Højre-klik i vinduet Connection Managers og vælg New Connection . . .

Vælg EXCEL - Connection Manager for excel-files og klik Add.

I Excel File Path – Browse til den oprettede mappe, skriv i File Name: Employee og tryk Open.

Excel version: 95-2005

Klik First Row Has Column Names

Klik Ok.

Den oprettede Connection Manager får navnet "Excel Connection Manager". Højreklik og rename den til "Excel Connection Manager – Employee".

Creating a Data Flow

Klik tab Data Flow.

Klik på linket midt på siden, som fortæller at der ikke er nogen data flow valgt.

Dette gør at der oprettes en boks i Control Flow, der hedder Data Flow Task og som vises i dropdown listen i tab Data Flow.

Denne fyldes med de controls som nu kan lægges på tab Data Flow.

Ændre dens property Name til "Data Flow Task – Copy Employee". Dette kan gøres i begge tabs.

Adding Data Adapters

Data Adapters er Data Sources og Data Destinations.

Add source adapter

Med tab Data Flow valgt, træk fra Data Flow Sources i Toolbox en OLE DB Source til designvinduet.

Ændre dens Name i properties til OLE DB Source – Employee.

Den angiver en fejl som en lille rød plet fordi den ikke er tilknyttet en Connection Manager.

Dobbelt-klik på OLE DB Source – Employee, hvorved der åbnes en configuration properties editor.

Vælg fanen Connection Manager.

OLE DB connection manager: vælg localhost.H2Company (eller hvad den nu hedder).

Data Access mode: Table or View

Name of the Table or the View: dbo.Employee (eller dbo.Emp)

Klik Preview for at se data, og klik Close.

Klik på fanen Columns. Her kan der vælges hvilke kolonner der skal med fra tabellen, og hvilket navn de skal have i output. Alle er valgt. (Men man skal derind for at de bliver valgt).

Tryk OK.

Add Destination Adapter

Fra ToolBox i gruppen Data Flow Destinations trækkes en Excel Destination ind på designvinduet, og dens navn ændres til Excel Destination – Employee.

Dobbeltklik på den, hvorved der kommer en advarsel om at den ikke har noget input. Svar nej til at fortsætte.

Marker OLE DB Source – Employee og træk dens grønne pil hen på Excel Destination – Employee.

Dobbeltklik igen på Excel Destination – Employee.

Vælg fanen Connection Manager.

OLE DB Connection Manager: Excel Employee.

Data Access Mode: Table or view.

Name of the Excel sheet: Tryk New . . .

Ændre Create Table 'Excel Destination - Employee' til Create Table 'Employee',

(Hvis der havde været lange kolonner f.eks. NVARCHAR(256), skulle de ændres til NVARCHAR(50), da Excel connection manager tillader ikke lange kolonner)

Tryk OK.

Der er nu oprettet en tabel, som danner grundlaget for Excel-filen. Hvis man vil ændre denne tabel, må den forrige slettes med sql-kommandoen Drop Table. Dette kan gøres ved at trykke New, og så slette kommandoen Create Table og i stedet skrive Drop Table Employee.

Der kræves at source-tabellen har Unicode teksttyper, NVarchar og NChar. Hvis dette ændres efter at Data Flow Source og Destination er oprettet, skal der dobbeltklikkes på "OLE DB Source – Employee" og sige Yes til at ændre metadata.

Tryk Preview og se at der er tomt.

Læg mærke til at der ikke kan trykkes OK.

Tryk på fanen Mappings, hvorved der mappes imellem input og output-kolonner.

Tryk OK.

(Hvis der var angivet kortere kolonner, ville der er nu en advarsel om, at truncation kan forekomme.)

Udfør den oprettede package

Højre-klik på CopyTable.dtsx i Solution Explorer og vælg Execute Package. Når den er færdig stoppes debuggeren ved at klikke på linket eller i menuen Debug.

Filen Employee.xls er nu oprettet i mappen Data.

Kap 8. Designing Policy Based Management

Lesson 1: Designing Policies

Policy Based Management introducerer 5 objekter der bruges til at designe og checke for compliance:

- Facets
- Conditions
- Policy Targets
- Policies
- Policy Categories.

Facets

Facets er de egenskaber, der kan checkes på.

Conditions

Conditions er de betingelser, der kan sættes op. Ligner betingelserne i en WHERE-sætning.

Policy Targets

Policy Targets er de objekter der skal checkes på. F. eks. hvilken database der skal checkes.

Policies

Der tales om execution mode, som kan være:

- On demand
- On change, prevent
- On change, log only
- On schedule

Policy Categories

Categories bruges til at gruppere en eller flere policies.

Practice: Defining Policies and Checking for Compliance

Practice 1: Create a Condition

Der oprettes 4 conditions.

1-2:

I den aktuelle servers Management | Policy Management højreklikkes på Conditions og vælges New Condition.

3 (1. Cond.):

Name: Auto Shrink and Auto Close Disabled

Facet: Database

Expression: @AutoClose = False And @AutoShrink = False

4 (2. Cond.):

Name: Check Database Engine Surface Area

Facet: Surface Area Configuration

Expression: @AdHocRemoteQueriesEnabled = false

And @ClrIntegrationEnabled = false

And @OleAutomationEnabled = false

And @SqlMailEnabled = false

Forskelligt der normalt ikke er brug for, bl.a SQLMail, er erstattet af Database Mail.

5 (3. Cond.):

Name: Database not in Simple recovery model.

Facet: Database.

Expression: @RecoveryModel != Simple.

Simple vælges ikke, men skrives med fri tekst, men den checker at det er skrevet rigtigt. Ved fejl kan der ikke trykkes OK.

6-7(4. Cond.):

Name: Tables without primary keys

Facet: Table

Expression:

```
IsNull(ExecuteSql('Numeric',  
'Select 1 from sys.tables a  
inner join sys.indexes b on a.object_id = b.object_id  
where b.is_primary_key = 1  
and a.name = @@ObjectName  
and a.schema_id = schema_id(@@SchemaName)'), 0) = 1
```

Practice 2: Create a Condition for a Target Set

1:

Name: SQL Server 2005 or later.

Facet: Server

Expression: @VersionMajor >= 9

2:

Name: Online user databases

Facet: Database

Expression:

@IsAccessible = True

And @IsSystemObject = False

Practice 3: Create a Policy

Højreklik på Policies under serverens Management | Policy Management og vælg New Policy...

1.

Name: Check for auto shrink and auto close

Check condition: Vælg den tidligere oprettede "Auto Shrink and Auto Close Disabled"

Against targets: Online user databases

Evaluation mode: On demand

Server restriction : None

2.

Name: Check for surface area configuration

Check condition: Vælg: Check Database Engine Surface Area

Against targets: Intet

Evaluation mode: Vælg: On demand

Server restriction: Vælg: Server 2005 or later.

3:

Name: Check for Simple recovery model

Check condition: Vælg: Database not in Simple recovery model.

Against targets: Vælg: Online user databases

Evaluation mode: Vælg: On demand

Server restriction: None

4:

Name: Check tables for primary key

Check condition: Tables without primary keys

Against targets: Every table in Online user databases

Evaluation mode: On demand

Server restriction: None

Practice 4 Create a Policy Category

Højreklik på Policy Management og vælg Manage Categories.

Opret to Catories af navne "Instance Surface Area Best Practices"
og "Database Best Practices". Det indtastes bare to nye items. Tryk OK

Nu tilknytttes de enkelte Polices til disse Categories.

På Policies | Check for auto shrink and auto close – højreklikkes og vælges properties eller bare dobbeltklik.

I fanen Discription vælg Category: Database best practice. OK.

For Check For Simple Recovery Model Policy vælg Database best practices.

For Check For Surface Area Configuration Policy vælg Surface Area Best Practices.

For Check Tables For Primary Key Policy vælg Database best Practices.

Og så mangler man ligesom at afprøve det hele!

Øvelse MS Official Course 6231A, Lab 2, Exercise 3: Creating a policy

Task 1: Create a policy to enforce naming conventions

Der oprettes en Policy, som via en Condition gælder databasen TK432, og via en anden Condition kræver at alle Stored Procedures navne begynder med 'TK_'.

I denne sammenhæng er Database og Stored Procedure begrebet Facets.

Condition

I Management Studio, Object Explorer, gå i den aktuelle servers Management | Policy Management | Facets | Stored Procedure, højreklik på denne og vælg New Condition.

Udfyld følgende:

Name: 'TK Stored Condition'

Facet: Vælg Stored Procedure

Field: Vælg @Name

Operator: Vælg: LIKE

Value: 'TK_%'

Altså for Stored Procedures kræves : @Name LIKE 'TK_'

Fane Description : Description: 'Stored Procedures must begin with 'TK_''

Kik i Script og se at det udføres med en stored procedure.

Tryk OK

Ved at gøre det via Facets på en Stored Procedure, er denne valgt i den åbnede dialog. Ellers kunne man højre-klikke på Conditions og vælge New Condition, hvilket åbner den samme dialog.

Policy

Stadig i Policy Management højreklik på Policies og vælg New Policy.

Name: 'TK Procedure Names'

Check Condition: 'TK Stored Condition'

Against Target: Klik på Every foran Database og vælg New Condition

I New Condition:

Name: 'TK432 DB'

Facet: Database.

Expression: @Name = 'TK432'

I Description fanen: 'Apply only to TK432 Database'

Tryk OK.

Evaluation Mode: Vælg On change: Prevent.

Sever Restriction: None.

Enabled: Checked

Description: 'All procedure names in Database TK432 must begin with 'TK_''

Tryk OK.

Altså, på databasen TK432 skal stored procedures navne begynde med 'AW_'.

På databasen TK432 højreklikkes og der vælges Policies | Categories. Udfold <Default> og check at der står TK432

Afprøvning

```
create procedure spTest
as
select * from INFORMATION_SCHEMA.TABLES
```

Dette burde ikke kunne lade sig gøre og der kommer også følgende fejlmeddelelse.

```
Policy 'TK Procedure Names' has been violated by 'SQLSERVER:\SQL\TEC-BB-
JAJ\DEFAULT\Databases\TK432\StoredProcedures\dbo.spTest'.
This transaction will be rolled back.
Policy condition: '@Name LIKE 'TK_%''
Policy description: 'All Procedure names i Database TK432 must begin with 'TK_''
Additional help: '' : ''
Statement: 'create procedure spTest
as
select * from INFORMATION_SCHEMA.TABLES'.
Msg 3609, Level 16, State 1, Procedure sp_syspolicy_dispatch_event, Line 65
The transaction ended in the trigger. The batch has been aborted.
```

Så oprettes en stored procedure, som overholder reglerne.

```
create procedure TK_spTest
as
select * from INFORMATION_SCHEMA.TABLES
```

Command(s) completed successfully.

exec TK_spTest

| TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | TABLE_TYPE |
|---------------|--------------|------------|------------|
| TK432 | dbo | Table1 | BASE TABLE |
| TK432 | dbo | Table2 | BASE TABLE |
| TK432 | dbo | Table3 | BASE TABLE |

Kap 9. Backing up and Restoring a Database

Lesson 1: Backing up Databases

Backup Types

- Full
- Differential
- Transaction Log
- Filegroup

Full Backups

En full backup indeholder alle data-pages i den aktuelle database. Datapages der ikke indeholder data bliver dog udeladt.

En full backup er grundlaget for både differential backup og transaction backup. Disse kan altså ikke foretages før der er udført en full backup.

En full backup medtager også ændringer i databasen, som sker under udførelsen af full backup. Der foretages altså også backup af indholdet i transaction log i en full backup.

Transaction Log Backups

Transaction log backup svarer til incremental backup i Windows.

Enhver ændring i databasen får en linie i transaction log. Hver af disse linier får et unikt nummer tildelt, som kaldes Log Sequence Number (LSN). Nummereringen begynder ved 1 ved databasens oprettelse og tælles op til uendeligt.

Transaction log har to dele. En aktiv del og en inaktiv del.

Den aktive del indeholder ændringer til databasen, som der ikke er sket commit på endnu.

Den inaktive del indeholder ændringer, som der er sket commit på.

En backup af transaction log sker på den inaktive del. Når dette er udført frigives pladsen til at kunne overskrives.

Transaction log backup følger efter hinanden, sådan at alle der er foretaget siden seneste full/differential backup skal bruges ved en restore.

Differential Backups

Differential backup medtager alle ændringer i databasen, som er sket siden seneste full backup.

Hvis der foretages flere differential backup, vil den seneste indeholde alt hvad der ligger i forrige differential backup plus hvad der er sket siden.

Filegroup Backups

Det er muligt at foretage backup af en del af en database ved at udføre backup på en filegroup, idet der kan være flere filegroups i en database.

Der kan også foretages backup af en enkelt fil af flere filer i en filegroup, men det giver ikke nogen mening da database-tabeller vil være spredt over flere filer i samme filegroup.

Partial Backups

Partial Backups handler om at filegroups kan være read-only. Hvis der er sådanne, er det ikke nødvendigt at udføre backup på disse mere end én gang.

Derfor er det muligt at udføre backup af en database på kun skrivbare filegroups.

```
BACKUP DATABASE <database> READ_WRITE_FILEGROUPS . . .
```

Page Corruption

Når en data-page skrives til disk, skrives der i dens header en checksum af dens indhold.

Hvis dette skal bruges til noget, skal databasen sættes til at checke denne checksum, når data-pages indlæses fra disk. Dette gøres med kommandoen:

```
ALTER DATABASE <database> SET PAGE_VERIFY CHECKSUM
```

Databasen vil nu beregne en checksum af indholdet af hver data-page, som indlæses, og sammenligne denne med checksummen i headeren.

Hvis der er uoverensstemmelse imellem de to checksum, logges der en error ved at indsætte en række i en tabel i databasen MSDB. Hvis databasen har en mirror, hentes den tilsvarende page fra denne og skrives som erstatning for den korrupte page.

Hvis antallet af korrupte pages overstiger 1000, tages databasen offline.

Udførelsen af en backup kan sættes til at afbrydes eller ikke afbrydes når der backup'es en korrupt data-page.

Maintenance Plans

Maintenance Plans er en grafisk design af job-workflows, der udfører administrative funktioner, såsom:

- Backup
- Shrinking af databaser
- Rebuild af indexes
- Opdatering af statistikker
- Udførelse af konsistens checks

Certificates and Master Keys

Ligger bl.a. til grund for krypteringer. Disse kan der også udføres backup på. Backups af disse bør gemmes et andet fysisk sted end backups af databaserne.

Validating a Backup

Udfører en restore med option VERIFYONLY, hvilket gør at den ikke gennemføres, men kun checker at backup'en er valid.

Backup kommando med dynamiske filnavne

Med følgende script udføres en backup-kommando med tidspunktet - dato og tid - indsat i den oprettede fil.

Tidspunktet lægges i variabelen @time og ":" fjernes fra klokkeslettet, da det er en ulovlig karakter i filnavne.

I variabelen @file lægges sti og filnavn indsat tidspunktet, og denne bruges til backup-kommandoens filnavn.

Scriptet skal udføres samlet.

```
declare @time varchar(100)
set @time = cast(GETDATE() as varchar)
print @time

set @time = replace(@time, ':', '')
print @time

declare @file varchar(100)
set @file = 'C:\BackupDB\AdventureWorks_1a' + @time + '.bak'

print @file

backup database AdventureWorks
to disk = @file
with compression, init, format, checksum, stop_on_error

/*
Output:
Aug  7 2012 10:55AM
Aug  7 2012 1055AM
C:\BackupDB\AdventureWorks_1aAug  7 2012 1055AM.bak
Processed 184 pages for database 'AdventureWorks', file 'AdventureWorks' on file
1.
Processed 1 pages for database 'AdventureWorks', file 'AdventureWorks_log' on
file 1.
BACKUP DATABASE successfully processed 185 pages in 0.158 seconds (9.147
MB/sec).
*/
```


Practice

Følgende øvelser indeholder bl.a. backup af logfil. Dette kan ikke lade sig gøre hvis databasen er i recovery-mode Simple.

For at sætte databasen til recovery-mode Full udføres følgende kommando.

```
alter database AdventureWorks  
set recovery full
```

Eller højreklik på den aktuelle database i Obejct-explorer, vælg Properties, Options og sæt Recovery Model til Full.

Practice 1: Create a Compressed, Mirrored, Full Backup

Følgende kommando udfører en backup af databasen AdventureWorks.

De benyttede options er:

- compression: backup-filen kommer til at fylde mindre end database-filerne
- init: tidligere backups overskrives
- format: backup-headeren overskrives
- checksum: data-pages verificeres ved at beregne checksum og sammenligne med deres angivne checksum.
- Stop_on_error: backup'en afbrydes hvis der er fejl i en data-page

```
backup database AdventureWorks  
to disk = 'C:\BackupDB\AdventureWorks_1a.bak'  
mirror to disk = 'C:\BackupDB\AdventureWorks_1b.bak'  
with compression, init, format, checksum, stop_on_error
```

Message: BACKUP DATABASE successfully processed 21318 pages

Der blev skrevet 21318 pages * 8k = 170M til backup. Passer det med databasens størrelse, datafil og logfil?

Hvor meget fylder backup-filen. Er der sket en compression?

Logfilen bliver altså også backup'et!

Practice 2: Create a Transactional Log Backup

Da logfilen er tom pga. at der er udført en full backup, skal der lige indsættes en række i en tabel og derefter udføres en transacion log backup.

```
insert into AdventureWorks.HumanResources.Department  
(Name, GroupName)  
values ('Test1', 'Research and Development')
```

```
backup log AdventureWorks
```

```
to disk = 'C:\BackupDB\AdventureWorks_1.trn'
with compression, init, checksum, stop_on_error
```

Og så en gang til.

```
insert into AdventureWorks.HumanResources.Department
(Name, GroupName)
values('Test2', 'Research and Development')

backup log AdventureWorks
to disk = 'C:\BackupDB\AdventureWorks_2.trn'
with compression, init, checksum, stop_on_error
```

Practice 3: Create a Differential Backup

Der indsættes en række mere, og så foretages der en differential backup.

```
insert into AdventureWorks.HumanResources.Department
(Name, GroupName)
values('Test3', 'Research and Development')
```

Kommandoen for differential backup er den samme som for full backup, men tilføjet option differential.

```
backup database AdventureWorks
to disk = 'C:\BackupDB\AdventureWorks_1a.dif'
mirror to disk = 'C:\BackupDB\AdventureWorks_1b.dif'
with differential, compression, init, format, checksum, stop_on_error
```

Der haves nu filerne:

```
AdventureWorks_1a.bak
AdventureWorks_1b.bak
AdventureWorks_1a.dif
AdventureWorks_1b.dif
AdventureWorks_1.trn
AdventureWorks_2.trn
```

Lesson Review

Der spørges om begreberne NORECOVERY, NO_TRUNCATE og TRUNCATE_ONLY uden at begreberne har været behandlet i den foregående tekst.

Lesson 2: Restoring Databases

Practice 1: Purposely Damage a Database

Indsæt en enkelt række i database.

```
insert into AdventureWorks.HumanResources.Department  
(Name, GroupName)  
values ('Test4', 'Research and Development')
```

Fjern datafilen for databasen:

Åbn Sql Server Configuration Manager og stop servicen for den aktuelle instans. (Alternativt kan der udføres Detach eller Take Offline på databasen AdventureWorks)

I Windows explorer flyttes datafilen for databasen AdventureWorks, AdventureWorks_Data.mdf over i en anden mappe.

Start servicen igen og iagttag at det nu er umuligt at gøre noget i databasen i Management Studio.

Practice 2: Restore a Full Backup

Under restore af backup skal databasen Master være den aktive.

Før der restores skal der udføres backup på transaction log tail, sådan at der kan restores helt til det øjeblik hvor databasen fejlede.

```
backup log AdventureWorks  
to disk = 'C:\BackupDB\AdventureWorks_3.trn'  
with compression, init, no_truncate
```

Derefter kan full backup restores.

```
restore database AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_1a.bak'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Option STANDBY svarer til NORECOVERY, hvor der kan tilføjes flere differential eller transaction log backups, med den forskel at ved STANDBY kan der selectes fra databasen. Der kan ikke selectes fra databasen ved NORECOVERY.

Kig i databasens properties og se at den har status Standby.

Databasen er nu restored, men mangler de indsættelser, der blev udført efter full backup.

```
select * from AdventureWorks.HumanResources.Department
```

Practice 3: Restore a Differential Backup

Nu restores differential backup, som blev taget efter indsættelsen af department Test 3.

```
restore database AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_1a.dif'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Konstater at tabellen indeholder departments Test1, Test2 og Test3.

```
select * from AdventureWorks.HumanResources.Department
```

(Hvis det ønskes at afslutte her, som øvelsen i bogen gør, udføres der recovery på databasen med følgende kommando.

```
restore database AdventureWorks  
with recovery)
```

Men der var jo taget en transaction log backup (af tail) efter differential backup, så den kunne jo først udføres. Så er alt med i databasen også department Test4

```
restore log AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_3.trn'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Og derefter

```
restore database AdventureWorks  
with recovery
```

Practice 4: Restore a Transaction Log Backup

Hvis det skulle gøres med alle transaction log backup i stedet for at benytte differential backup, ville det således ud.

Bogen vil begynde med at restore full backup, men da databasen har været kørende i mellemtiden, skal der igen foretages en backup af transaction log tail.

```
backup log AdventureWorks  
to disk = 'C:\BackupDB\AdventureWorks_4.trn'  
with compression, init, no_truncate
```

Så kan der restores. Hold for hvert trin øje med indholdet i tabellen Department.

```
select * from AdventureWorks.HumanResources.Department
```

```
restore database AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_1a.bak'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Nu der der ingen af de ekstra indsatte departments

```
.  
restore log AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_1.trn'  
with standby = 'C:\BackupDB\AdventureWorks.stn'  
go
```

Nu er der department Test1.

```
restore log AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_2.trn'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Nu er der department Test2.

```
restore log AdventureWorks
```

```
from disk = 'C:\BackupDB\AdventureWorks_3.trn'  
with standby = 'C:\BackupDB\AdventureWorks.stn'
```

Nu er der department Test3 og Test4.

```
restore log AdventureWorks  
from disk = 'C:\BackupDB\AdventureWorks_4.trn'  
with recovery
```

Så er databasen kørende igen.

Check i databasens properties at status er Normal.

Lesson 3: Database Snapshots

Practice 1: Creating a Database Snapshot

Kap 10. Automating SQL Server

Job der indsætter rækker i en tabel

Som et lille eksempel vises hvordan der automatisk kan indsættes en række i en tabel hvert 10. sekund.

```
Create table JobTestTable(Text varchar(20), Time Datetime)
```

På SQL Server Agent | Jobs højreklikkes og vælges New Job.

Name: 'Insert into JobTestTable'

Owner: Vælg evt. sa, hvilket var nødvendigt hos mig da SQL Server åbenbart ikke kunne gennemskue om TEC\jaj var god nok til at starte jobbet. Måske er det nok at enable og så starter den når Schedule kommer.

Enable: Check

Vælg tab Steps.

Tryk New... og indtast:

Name: 'Insert row in JobTestTable'

Type: T-SQL

```
Insert into JobTestTable Values('Tid for indsættelse: ', getdate())
```

Tryk OK.

Vælg tab Schedule og tryk New...

Vælg Daily og hvert 10. sekund.

Tryk OK

Højreklik på det nyoprettede Job og vælg Start at Step...

Gå ind i Query Window og indtast følgende for at se de indsatte rækker.

```
select * from JobTestTable
```


Kap 11. Designing SQL Server Security

SQL Server

Enhver SQL Server har et login for Windows group BUILTIN\Administrators. BUILTIN\Administrators er via fixed server role (fast server rolle) sysadmin mappet til databasebrugeren dbo.

Der er også et SQL Server login der hedder sa, men denne er som default disabled. Hvis den enables skal den have sat et stærkt password. sa kan enables ved at gå ind i dens properties under tab Status. Kan også enables med SQL kommando. SQL Server login sa er ligeledes mappet til databasebrugeren dbo.

Enable login sa samt give den et password med SQL:

```
Alter login sa ENABLE
Alter login sa with password = 'P@ssw0rd'
Alter login sa DISABLE
```

Database

Enhver database har fra starten to brugere.

Brugeren dbo, ejer af schema dbo og med default schema dbo. Brugeren dbo er medlem af database rollen db_owner, som giver ret til at gøre alt i databasen.

Brugeren guest med default schema guest. Guest er som default disabled. Guest er som alle medlem af rollen public. Hvis et login ikke er mappet til en bruger på den database han forsøger at logge ind på, bliver han brugeren guest, hvis den er enablet. Guest kan enables og disables med følgende SQL kommandoer:

```
GRANT CONNECT TO guest;
DENY CONNECT TO guest;
```

Der er også INFORMATION_SCHEMA og sys, men det er ikke rigtige brugere og ikke mappet til et login.

Rettigheder

På både server, database, schema og databaseobjekt kan der gives rettigheder.

Rettigheder gives med nøgleordene GRANT, DENY og REVOKE.

Da rettigheder kan gives på forskellige niveauer, generelle på højt niveau og specifikke på lavt niveau, samt igennem roller og ved direkte angivelse, kan der være givet samme eller modstridende rettigheder flere gange til samme bruger. Her vil DENY vinde.

DENY er en negativ rettighed.

GRANT er en positiv rettighed.

REVOKE fjerner en GRANT eller en DENY.

Roller

Roller er ligesom Windows security groups, hvor der kan tildeles rettigheder. Server logins og database brugere kan så gøres til medlemmer af rollerne.

Faste Server Roller

En SQL Server login kan være medlem af en eller flere faste server roller. De egentlige rettigheder, som ligger i disse, er vist i skemaet herunde.

Den nederste, sysadmin, giver rettighed til at gøre alt i alle databaser på serveren.

| Fixed server role | Server-level permission |
|----------------------|---|
| bulkadmin | Granted: ADMINISTER BULK OPERATIONS |
| dbcreator | Granted: ALTER ANY DATABASE |
| diskadmin | Granted: ALTER RESOURCES |
| processadmin | Granted: ALTER ANY CONNECTION, ALTER SERVER STATE |
| securityadmin | Granted: ALTER ANY LOGIN |
| serveradmin | Granted: ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE |
| setupadmin | Granted: ALTER ANY LINKED SERVER |
| sysadmin | Granted with GRANT option: CONTROL SERVER |

De faste server roller kan ikke ændres, og der kan heller ikke oprettes nye. Så de er ikke særligt fleksible.

```
sp_addsrvrolemember 'TheLogin', 'sysadmin'
```

```
sp_dropsrvrolemember 'TheLogin', 'sysadmin'
```

Database roller

Database roller giver brugere rettigheder i databaser.

Der findes 3 slags database roller:

- Predefinerede/faste database roller
- Brugerdefinerede application roller
- Brugerdefinerede standard roller

De predefinerede/faste database roller indeholder følgende rettigheder.

| Fixed database role | Database-level permission |
|--------------------------|--|
| db_accessadmin | Granted: ALTER ANY USER, CREATE SCHEMA |
| db_accessadmin | Granted with GRANT option: CONNECT |
| db_backupoperator | Granted: BACKUP DATABASE, BACKUP LOG, CHECKPOINT |
| db_datareader | Granted: SELECT |
| db_datawriter | Granted: DELETE, INSERT, UPDATE |
| db_ddladmin | Granted: ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT, NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES |
| db_denydatareader | Denied: SELECT |
| db_denydatawriter | Denied: DELETE, INSERT, UPDATE |
| db_owner | Granted with GRANT option: CONTROL |
| db_securityadmin | Granted: ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION |

| | |
|--------------------|--|
| dbm_monitor | Granted: VIEW most recent status in Database Mirroring Monitor |
| | |

De predefinerede/faste database roller kan ikke ændres.

Medlemskab af rollerne kan tildeles og fjernes fra brugere med følgende stored procedures.

```
sp_addrolemember 'db_ddladmin', 'TheUser'  
sp_droprolemember 'db_ddladmin', 'TheUser'
```

Application Rolle

En application rolle giver rettigheder til en applicationen, som logger på serveren.

Brugerdefineret rolle

En brugerdefineret standard rolle er oprettet af databaseadministratoren. Til denne grantes rettigheder og brugere gøres til medlemmer af rollen.

```
create role NewRole  
  
grant select on TableName to NewRole  
  
sp_addrolemember 'NewRole', 'TheUser'
```

Grant, deny og revoke

De grundlæggende SQL permissions er grant, deny og revoke. Grant giver en rettighed. Deny nægter en rettighed. Revoke fjerner begge dele, som giver en neutral tilstand på det aktuelle niveau.

Rettigheder kan gives på server-niveau, på database-niveau, på schema-niveau og på database-objekt-niveau, f. eks. table.

Hvis der er givet modstridende tilladelser på forskellige niveau, som påvirker den samme rettighed, vil deny være den afgørende.

Hvis der som vist herunder er givet modstridende tilladelser på samme niveau, vil den sidste angivne være den afgørende.

```
deny select on TableName to TheUser  
grant select on TableName to TheUser  
revoke select on TableName to TheUser
```

System-views

Alle brugere kan læse fra system-views, men ser kun inform om de objekter, som de har rettigheder til.

F. eks. vil følgende kommando kun returnere rækker med de tabeller brugeren har rettigheder til.

```
select * from INFORMATION_SCHEMA.tables
```

Øvelser

I det følgende er der benyttet en database ved navnet SSEMDB.

Udskift databasenavnet SSEMDB med navnet på din egen testdatabase.

Opret Windows brugere

Der oprettes nogle Windows brugere og grupper, der kan logge ind på SQL-Serveren med Windows authentication.

Opret user Alice med password P@ssw0rd.

Opret user Bob med password P@ssw0rd.

Opret en SQL Server Login for en Windows bruger

For at give mulighed for at logge ind som en login på SQL Server 2008, skal der i Object Explorer højre-klikkes på den aktuelle Sql Server og vælges properties, og her under Security vælges SQL Server And Windows Authentication Mode. Server skal genstartes.

I Object Exploerer, den aktuelle Sql Server klik på Security og højre-klik på Logins. Vælg New Login. Vælg Windows authentication og ved Login Name tryk Search og find Alice, som er en Windows user.

Vælg Default database SSEMDB. Dette gør at når Alice logger ind, lander hun i databasen SSEMDB, men hun er ikke oprettet som user og vil så blive guest eller ingenting.

Tryk OK.

Det samme kan gøres med SQL:

```
use Master
create login "SELPACEDCPU\Alice" from Windows
Og fjernes igen
drop login "SELPACEDCPU\Alice"
```

Højre-klik på login Alice og vælg properties, eller dobbelt-klik på Alice, hvilket begge giver samme vindue, som ved oprettelse.

Vælg User Mapping og sæt flueben Map for SSEMDB, så bliver Alice oprettet som user i SSEMDB og får Default Schema dbo.

(Alternativt kan hun oprettes i SSEMDB som ny bruger, ved SSEMDB under Security at højre-klikke på Users og vælge New User...)

Opret en ny database-bruger med SQL:

```
use SSEMDB
CREATE USER Alice FOR LOGIN "SELPACEDCPU\Alice"
WITH DEFAULT_SCHEMA = dbo
```

Nu kan Alice ses som user i SSEMDB og i hendes properties kan ses at hun har Default Schema dbo.

Der kan også ses hvilke schemas hun ejer, hvilket er ingen, og det er ikke muligt at komme til at eje dbo. Der findes en række andre schemaer svarende til Fixed Database Roles.

Nyt schema til Alice.

På SSEMDB | Security | Schemas, højreklik og vælg New Schema.

Giv den navnet AliceSSEMDBSchema og Search efter Schema Owner og vælg Alice.

Tryk OK.

Opret nyt schema med SQL:

```
Create schema AliceSSEMDBSchema AUTHORIZATION Alice
```

Gå ind i Alice properties og se at hun nu er ejer af det nye AliceSSEMDBSchema.

Vælg det nye schema som default schema for Alice.

Default schema med SQL:

```
Alter user Alice
WITH DEFAULT_SCHEMA = AliceSSEMDBSchema
```

Login kan afprøves ved at logge ind i Windows som Alice og starte Management Studio og connecte med Windows Autication.

Men endnu nemmere ved at højreklikke på Management Studio og vælge Kør som...

Alice/password

Opret en SQL Server login for en ikke-Windows bruger

For at kunne logge ind med en SQL Server login, skal serveren sættes i Mixed Mode.

Højre-klik på den aktuelle server i Object Inspector og vælg Properties. Her vælges Security og der skiftes Server Authentication fra Windows Authenticon mode til SQL Server and Windows Authentication mode.

Genstart serveren for at dette kan få virkning.

I Object Explorer under default serverens Security mappe, højre-klik på Logins og vælg New Login.

Login Name: Carol.

Klik på SQL Server Authentication.

Password: P@ssw0rd.

Default Database: SSEMDB.

Tryk OK.

Med SQL:

```
create login Carol with password = 'P@ssw0rd', default_database = SSEMDB  
  
drop login Carol
```

To connections

For at afprøve Carol's muligheder, er det bedst at have to SQL Server Management Studio åbne. Den ene connected med en login, der har sysadmin rettigheder, f.eks. en Windows bruger fra BUILTIN\Administrators. Her giver administrator så Carol rettigheder. Den anden med Carol connected, hvor hun kan afprøve sine rettigheder. Der skiftes så i øvelsen imellem disse to Management Studio applikationer.

Alternativt kan der faktisk skiftes bruger, sådan at hvis man er connected som dbo, kan der skiftes til en svagere bruger og vendes tilbage til dbo igen.

Skiftet gælder kun database-useren, ikke den tilhørende server-login. Så hvis login Carol er medlem af serverrollen bulkadmin, får dette ikke virkning ved skift til user Carol.

Hvilken login er jeg

```
select SYSTEM_USER
```

Hvilken user er jeg

```
select CURRENT_USER
```

```
execute as user = 'Carol'  
go
```

```
. . . .
```

```
revert  
go
```

Der kan også afprøves ved en sql-prompt i en command-prompt.

```
C:\>sqlcmd -U sa -P P@ssw0rd (-i MyCreate.sql)  
1> exit
```

Afprøv Carol og giv rettigheder

Prøv at connecte med Carol/P@ssw0rd. Det går ikke godt.

Admin:

Gå igen ind i properties for Carol, vælg Mappings og under User Mapped To This Login vælg SSEMDB ved at klikke i checkboksen. Ingen default schema.

Ovenstående er det samme som at oprette en ny bruger i SSEMDB, ved i Object Explorer at højreklikke på Databases|SSEMDB|Security|Users, vælge New User., finde Login name: Carol og give User name: Carol.

Med SQL:

```
use SSEMDB
create user Carol for login Carol
```

Carol:

Prøv nu at logge ind som Carol. Det går godt, fordi Carol nu er blevet bruger i databasen SSEMDB.

Klik på New Query og læg mærke til at databasen SSEMDB bliver den aktive, fordi Carol har SSEMDB som default database.

Prøv at select fra en tabel f.eks. dbo.Customer.

```
select * from dbo.Customer
```

Dette giver en fejl da Carol ikke har nogen rettigheder.

Læg mærke til at tabellen er i schemaet dbo. Hvis et databaseobjekt ikke ligger i en brugers default schema kigges der efter det i schema'et dbo.

Admin:

Vælg user Carol properties og i Database Role Membership klik ud for db_datareader.

Carol:

Nu kan Carol selecte fra den førmtalte tabel.

```
select * from dbo.Customer
```

Admin:

Fjern db_datareader fra Carol og tilknyt hende igen til db_datareader i med følgende kommando

```
sp_addrolemember 'db_datareader', 'carol'
```

Fjernes igen med

```
sp_droprolemember 'db_datareader', 'carol'
```

Carol:

Afprøv at begge kommandoer virker.

```
select * from dbo.Customer
```

Medlemskab af database-rolle db_datareader giver læse-rettighed til hele databasen.

Public rolle.

Med rolle db_datareader fjernet fra Carol, prøv at give rolle public rettighed til at selecte på tabellen Customer

```
grant select on Customer to public
```

```
deny select on Customer to public
revoke select on Customer to public
```

Nu kan Carol læse i tabellen Customer. Fjern denne rettighed igen fra public.

Schema rettigheder.

Læse-rettighed begrænset til kun et bestemt schema, her schema dbo, kan gives med følgende kommando

```
grant select on schema::dbo to carol
```

Og fjernes igen med kommandoen

```
revoke select on schema::dbo to carol
```

Carol:

```
select * from dbo.Customer
```

Kan Carol oprette en table i schema dbo?

```
create table navne(navn varchar(34))
```

Nej, det kan hun ikke.

Admin:

Giv Carol lov til at create, drop og alter database-objekter i schema dbo.

Den første kommando er en generel rettighed, som ikke alene er nok. Den kan sammenlignes med en hovedkontakt:

```
grant create table to carol
```

Hvor næste kommando er en lokal kontakt til den egentlige rettighed

```
grant alter on schema::dbo to carol
```

Carol:

```
create table navne(navn varchar(34))
```

Admin:

Og fjern rettighederne igen

```
revoke create table to carol
revoke alter on schema::dbo to carol
```

Rollen public

Hvis der også oprettes brugeren Carol i databasen Northwind, viser det sig at det for Carol umiddelbart er muligt at selecte fra alle tabeller i denne database.

Det er fordi alle rettighederne til tabellerne er givet til rollen public, som alle brugere altid har.

Dette kan ses med de to følgende kommandoer

```
select * from information_schema.table_privileges
```



```
select * from information_schema.column_privileges
```

| GRANTOR | GRANTEE | TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | PRIVILEGE_TYPE | IS_GRANTABLE |
|---------|---------|---------------|--------------|------------|----------------|--------------|
| dbo | public | Northwind | dbo | Orders | DELETE | NO |
| dbo | public | Northwind | dbo | Orders | INSERT | NO |
| dbo | public | Northwind | dbo | Orders | REFERENCES | NO |
| dbo | public | Northwind | dbo | Orders | SELECT | NO |
| dbo | public | Northwind | dbo | Orders | UPDATE | NO |
| dbo | public | Northwind | dbo | Products | DELETE | NO |

...

| GRANTOR | GRANTEE | TABLE_CATALOG | TABLE_SCHEMA | TABLE_NAME | COLUMN_NAME | PRIVILEGE_TYPE | IS_GRANTABLE |
|---------|---------|---------------|--------------|------------|-------------|----------------|--------------|
| dbo | public | Northwind | dbo | Orders | OrderID | REFERENCES | NO |
| dbo | public | Northwind | dbo | Orders | OrderID | SELECT | NO |
| dbo | public | Northwind | dbo | Orders | OrderID | UPDATE | NO |

...

Følgende kommando gør at alle brugere på databasen kan selecte fra tabellen TableName.

```
grant select on TableName to public
```

Schema

Permissions der kan gives på schema.

Giv Carol et schema

Admin

Carol skal nu have sit eget schema med følgende kommando

```
create schema CarolSchema authorization Carol
```

For at hun kan create sin egen table i dette schema, skal hun stadigvæk have den generelle rettighed

```
grant create table to Carol
```

Hvis Carol stadig har dbo som default schema, skal hun kvalificere de tabeller, hun opretter, med CarolSchema, hvis tabellerne skal ligge her.

```
create table CarolSchema.Navne(Navn varchar(25))
```

```
drop table CarolSchema.Navne
```

Carol sættes nu til at have sit eget schema som default schema

```
alter user Carol with default_schema = CarolSchema
```

Carol kan nu oprette og benytte tabeller i sit eget schema uden at kvalificerede med schema-navnet

```
drop table Navne
```

```
create table Navne(Navn varchar(25))

insert into Navne values('Hans')
insert into Navne values('Jens')
insert into Navne values('Niels')
```

For lige at kontrollere at det var Carol's schema, der blev oprettet en table, selectes med kvalificering hendes schema

```
select * from CarolSchema.Navne
```

Hans
Jens
Niels

Ny Login oprettet i SQL og med rettighed til en specifik table

Admin

Opret en ny login Salsa med SQL og giv rettigheder til en bestemt table i dbo schema

```
drop user Salsa
create login Salsa with password='S&l$al23!'
create user Salsa for login Salsa

create table dbo.Carpets(name varchar(25), color varchar(20), price int)
insert into dbo.Carpets values('Flying', 'Red', 12500)
insert into dbo.Carpets values('Laying', 'Blue', 1000)
insert into dbo.Carpets values('Crouching', 'Yellow', 125)

grant select on dbo.Carpets to salsa
```

Salsa

kan nu selecte fra tabellen

```
select * from dbo.Carpets
```

Admin

giver nu rettighed til kun at selecte fra to af kolonnerne i tabellen

```
revoke select on dbo.carpets to salsa
grant select(name, price) on dbo.carpets to salsa
```

Salsa

kan nu ikke selecte alt fra tabellen

```
select * from dbo.Carpets
-error
```

Salsa må nu kun selecte fra de angivne kolonner

```
select name, price from dbo.carpets
```

Admin kan evt. fjerne rettigheden igen

```
revoke select(name, price) on dbo.carpets to salsa
```

Øvelser

1. Opret Sql Server-login Mads med default database MyDB og gør ham til medlem af serverrollen sysadmin.
Opret user Mads i MyDB mappet til login Mads
2. Opret Windows-login Alex med default database MyDB og tilsvarende user Alex i MyDB og giv ham rettighed til at læse alt i databasen (database-rolle).
3. Giv også user Alex rettighed til at indsætte (kun indsætte) i alle tabeller.
4. Giv user Alex rettighed til at oprette tabeller (i dbo-schema).
5. Opret login og tilhørende user Birthe i MyDB. Giv hende sit eget schema BirthesSchema.
Lad hende oprette tabellen BirthesTabel i sit eget schema.
Lad hende give Alex rettighed til at indsætte i tabellen.
Afprøv at Alex kan indsætte i tabellen.

Noter

At sætte en users default database til master går an, fordi guest default er enablet i master.

Object permissions.

| Object permission | Implied by object permission | Implied by schema permission |
|----------------------|------------------------------|------------------------------|
| ALTER | CONTROL | ALTER |
| CONTROL | CONTROL | CONTROL |
| DELETE | CONTROL | DELETE |
| EXECUTE | CONTROL | EXECUTE |
| INSERT | CONTROL | INSERT |
| RECEIVE | CONTROL | CONTROL |
| REFERENCES | CONTROL | REFERENCES |
| SELECT | RECEIVE | SELECT |
| TAKE OWNERSHIP | CONTROL | CONTROL |
| UPDATE | CONTROL | UPDATE |
| VIEW CHANGE TRACKING | CONTROL | VIEW CHANGE TRACKING |
| VIEW DEFINITION | CONTROL | VIEW DEFINITION |

Schema permissions.

| Schema permission | Implied by schema permission | Implied by database permission |
|----------------------|------------------------------|--------------------------------|
| CONTROL | CONTROL | CONTROL |
| TAKE OWNERSHIP | CONTROL | CONTROL |
| ALTER | CONTROL | ALTER ANY SCHEMA |
| EXECUTE | CONTROL | EXECUTE |
| INSERT | CONTROL | INSERT |
| DELETE | CONTROL | DELETE |
| UPDATE | CONTROL | UPDATE |
| SELECT | CONTROL | SELECT |
| REFERENCES | CONTROL | REFERENCES |
| VIEW CHANGE TRACKING | CONTROL | CONTROL |
| VIEW DEFINITION | CONTROL | VIEW DEFINITION |

Database permissions

| Database permission | Implied by database permission | Implied by server permission |
|---------------------------------------|--------------------------------|------------------------------|
| ALTER | CONTROL | ALTER ANY DATABASE |
| ALTER ANY APPLICATION ROLE | ALTER | CONTROL SERVER |
| ALTER ANY ASSEMBLY | ALTER | CONTROL SERVER |
| ALTER ANY ASYMMETRIC KEY | ALTER | CONTROL SERVER |
| ALTER ANY CERTIFICATE | ALTER | CONTROL SERVER |
| ALTER ANY CONTRACT | ALTER | CONTROL SERVER |
| ALTER ANY DATABASE AUDIT | ALTER | ALTER ANY SERVER AUDIT |
| ALTER ANY DATABASE DDL TRIGGER | ALTER | CONTROL SERVER |
| ALTER ANY DATABASE EVENT NOTIFICATION | ALTER | ALTER ANY EVENT NOTIFICATION |
| ALTER ANY DATASPACE | ALTER | CONTROL SERVER |
| ALTER ANY FULLTEXT CATALOG | ALTER | CONTROL SERVER |
| ALTER ANY MESSAGE TYPE | ALTER | CONTROL SERVER |
| ALTER ANY REMOTE SERVICE BINDING | ALTER | CONTROL SERVER |
| ALTER ANY ROLE | ALTER | CONTROL SERVER |
| ALTER ANY ROUTE | ALTER | CONTROL SERVER |
| ALTER ANY SCHEMA | ALTER | CONTROL SERVER |
| ALTER ANY SERVICE | ALTER | CONTROL SERVER |
| ALTER ANY SYMMETRIC KEY | ALTER | CONTROL SERVER |
| ALTER ANY USER | ALTER | CONTROL SERVER |
| AUTHENTICATE | CONTROL | AUTHENTICATE SERVER |
| BACKUP DATABASE | CONTROL | CONTROL SERVER |
| BACKUP LOG | CONTROL | CONTROL SERVER |
| CHECKPOINT | CONTROL | CONTROL SERVER |
| CONNECT | CONNECT REPLICATION | CONTROL SERVER |
| CONNECT REPLICATION | CONTROL | CONTROL SERVER |
| CONTROL | CONTROL | CONTROL SERVER |
| CREATE AGGREGATE | ALTER | CONTROL SERVER |
| CREATE ASSEMBLY | ALTER ANY ASSEMBLY | CONTROL SERVER |
| CREATE ASYMMETRIC KEY | ALTER ANY ASYMMETRIC KEY | CONTROL SERVER |
| CREATE CERTIFICATE | ALTER ANY CERTIFICATE | CONTROL SERVER |
| CREATE CONTRACT | ALTER ANY CONTRACT | CONTROL SERVER |

Database permissions fortsat...

| Database permission | Implied by database permission | Implied by server permission |
|--|---------------------------------------|-------------------------------|
| CREATE DATABASE | CONTROL | CREATE ANY DATABASE |
| CREATE DATABASE DDL EVENT NOTIFICATION | ALTER ANY DATABASE EVENT NOTIFICATION | CREATE DDL EVENT NOTIFICATION |
| CREATE DEFAULT | ALTER | CONTROL SERVER |
| CREATE FULLTEXT CATALOG | ALTER ANY FULLTEXT CATALOG | CONTROL SERVER |
| CREATE FUNCTION | ALTER | CONTROL SERVER |
| CREATE MESSAGE TYPE | ALTER ANY MESSAGE TYPE | CONTROL SERVER |
| CREATE PROCEDURE | ALTER | CONTROL SERVER |
| CREATE QUEUE | ALTER | CONTROL SERVER |
| CREATE REMOTE SERVICE BINDING | ALTER ANY REMOTE SERVICE BINDING | CONTROL SERVER |
| CREATE ROLE | ALTER ANY ROLE | CONTROL SERVER |
| CREATE ROUTE | ALTER ANY ROUTE | CONTROL SERVER |
| CREATE RULE | ALTER | CONTROL SERVER |
| CREATE SCHEMA | ALTER ANY SCHEMA | CONTROL SERVER |
| CREATE SERVICE | ALTER ANY SERVICE | CONTROL SERVER |
| CREATE SYMMETRIC KEY | ALTER ANY SYMMETRIC KEY | CONTROL SERVER |
| CREATE SYNONYM | ALTER | CONTROL SERVER |
| CREATE TABLE | ALTER | CONTROL SERVER |
| CREATE TYPE | ALTER | CONTROL SERVER |
| CREATE VIEW | ALTER | CONTROL SERVER |
| CREATE XML SCHEMA COLLECTION | ALTER | CONTROL SERVER |
| DELETE | CONTROL | CONTROL SERVER |
| EXECUTE | CONTROL | CONTROL SERVER |
| INSERT | CONTROL | CONTROL SERVER |
| REFERENCES | CONTROL | CONTROL SERVER |
| SELECT | CONTROL | CONTROL SERVER |
| SHOWPLAN | CONTROL | ALTER TRACE |
| SUBSCRIBE QUERY NOTIFICATIONS | CONTROL | CONTROL SERVER |
| TAKE OWNERSHIP | CONTROL | CONTROL SERVER |
| UPDATE | CONTROL | CONTROL SERVER |
| VIEW DATABASE STATE | CONTROL | VIEW SERVER STATE |
| VIEW DEFINITION | CONTROL | VIEW ANY DEFINITION |

Grant <permission> to <user>

Server permissions

| Server permission | Implied by server permission |
|---------------------------------|------------------------------|
| ADMINISTER BULK OPERATIONS | CONTROL SERVER |
| ALTER ANY CONNECTION | CONTROL SERVER |
| ALTER ANY CREDENTIAL | CONTROL SERVER |
| ALTER ANY DATABASE | CONTROL SERVER |
| ALTER ANY ENDPOINT | CONTROL SERVER |
| ALTER ANY EVENT NOTIFICATION | CONTROL SERVER |
| ALTER ANY LINKED SERVER | CONTROL SERVER |
| ALTER ANY LOGIN | CONTROL SERVER |
| ALTER ANY SERVER AUDIT | CONTROL SERVER |
| ALTER RESOURCES | CONTROL SERVER |
| ALTER SERVER STATE | CONTROL SERVER |
| ALTER SETTINGS | CONTROL SERVER |
| ALTER TRACE | CONTROL SERVER |
| AUTHENTICATE SERVER | CONTROL SERVER |
| CONNECT SQL | CONTROL SERVER |
| CONTROL SERVER | CONTROL SERVER |
| CREATE ANY DATABASE | ALTER ANY DATABASE |
| CREATE DDL EVENT NOTIFICATION | ALTER ANY EVENT NOTIFICATION |
| CREATE ENDPOINT | ALTER ANY ENDPOINT |
| CREATE TRACE EVENT NOTIFICATION | ALTER ANY EVENT NOTIFICATION |
| EXTERNAL ACCESS ASSEMBLY | CONTROL SERVER |
| SHUTDOWN | CONTROL SERVER |
| UNSAFE ASSEMBLY | CONTROL SERVER |
| VIEW ANY DATABASE | VIEW ANY DEFINITION |
| VIEW ANY DEFINITION | CONTROL SERVER |
| VIEW SERVER STATE | ALTER SERVER STATE |

Script der kaster logins af
Det er ikke muligt at slette en login, som er logget ind.
Følgende script disconnecter login Carol.

```
DECLARE @dbname varchar(50)
SET @dbname = 'SSEMDB'

DECLARE @strSQL varchar(255)
PRINT 'Killing Users'
PRINT '-----'

DROP table #tmpUsers

CREATE table #tmpUsers(
    spid int,
    eid int,
    status varchar(30),
    loginname varchar(50),
    hostname varchar(50),
    blk int,
    dbname varchar(50),
    cmd varchar(30),
    request_id int
)

INSERT INTO #tmpUsers EXEC SP_WHO

DECLARE LoginCursor CURSOR
READ_ONLY
FOR SELECT spid, dbname FROM #tmpUsers
WHERE dbname = @dbname
--eller
--WHERE loginname = 'Carol'

DECLARE @spid varchar(10)
DECLARE @dbname2 varchar(40)
OPEN LoginCursor

FETCH NEXT FROM LoginCursor INTO @spid, @dbname2
WHILE (@@fetch_status <> -1)
BEGIN
    IF (@@fetch_status <> -2)
    BEGIN
        PRINT 'Killing ' + @spid
        SET @strSQL = 'KILL ' + @spid
        EXEC (@strSQL)
    END
    FETCH NEXT FROM LoginCursor INTO @spid, @dbname2
END

CLOSE LoginCursor
DEALLOCATE LoginCursor
```


Supervariablen @@fetch_status returnerer hvordan den seneste FETCH NEXT gik og har følgende muligheder:

0: The FETCH statement was successful.

-1: The FETCH statement failed or the row was beyond the result set.

-2: The row fetched is missing

Kap 12. Monitoring Microsoft SQL Server

Kap 13. Optimizing Performance

Kap 14. Failover Clustering

Kap 15. Database Mirroring

Kap 16. Log Shipping

Kap 17. Replication

Andre emner

Sql-kommandoer med parametre.

```
SqlConnection con = new SqlConnection(
    "Data Source=.\JansSqlServer;Initial "
    + "Catalog=H2DB;Integrated Security=True");
con.Open();

string cmdStr = "Select * from Emp where EmpNo = @empno";

SqlParameter param = new SqlParameter("@empno", SqlDbType.Int);
param.Value = 7782;

SqlCommand cmd = new SqlCommand(cmdStr, con);
cmd.Parameters.Add(param);

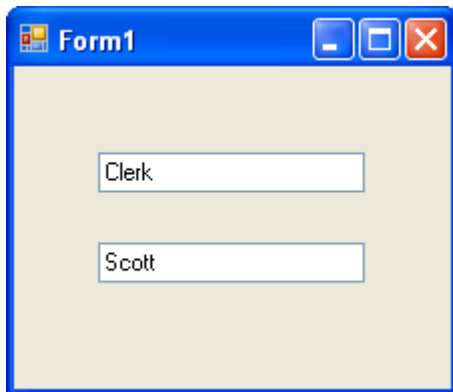
SqlDataReader result = cmd.ExecuteReader();

result.Read();
textBox1.Text = result["LName"].ToString();

result.Close();
param.Value = 7788;
result = cmd.ExecuteReader();

result.Read();
textBox2.Text = result["LName"].ToString();

con.Close();
```



Noter:

Databaserne master, model, tempdb og msdb er ikke omtalt samlet.
SqlCmd.exe er ikke beskrevet