



- ✓ DDL Kommandon
- ✓ CREATE/ALTER/DROP Database
- ✓ CREATE /ALTER/DROP Table
- ✓ ALTER/ADD/DROP Column
- ✓ CREATE /ALTER/DROP Index
- ✓ Shrink med DBCC

Chapter 3, 6, 8 – delar av.

Beginning SQL Server 2008 for Developers



# CREATE DATABASE

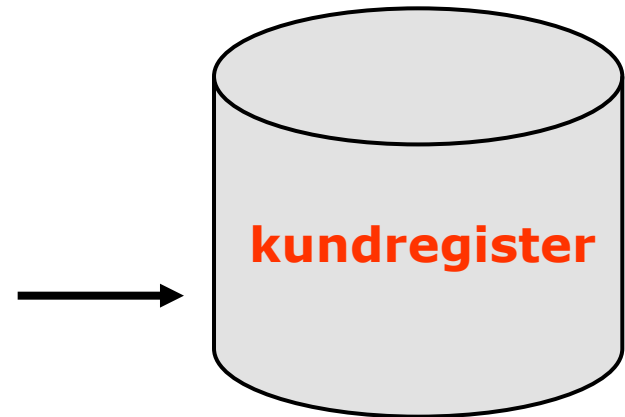
Själva databasen är kontainer för tabellerna. Vi måste alltså först och främst skapa vår databas. Det görs i SQL med kommandot CREATE DATABASE.

## Syntax:

```
CREATE DATABASE databasnamn
```

## Exempel:

```
CREATE DATABASE kundregister
```



Beroende på databashanterare finns det sedan en uppsjö med olika alternativ för att sätta rättigheter, välja fysisk placering av databasfilen, bestämma hur loggning ska skötas, hur stor och hur mycket databasen får växa med mera, med mera. Här får du ta hjälp av respektive databashanterares manual för att se hur du ska gå till väga.



## Exempel som gäller för MS SQL Server:

```
CREATE DATABASE kundregister ON PRIMARY
(
NAME='kundregister',
FILENAME='E:\Data\kundregister.mdf',
SIZE=6144KB,
MAXSIZE=UNLIMITED,
FILEGROWTH=2048KB
)
LOG ON
(
NAME='kundregister_log',
FILENAME='D:\Data\kundregister_log.ldf',
SIZE=1024KB,
MAXSIZE=UNLIMITED,
FILEGROWTH=10%
)
```



# CREATE TABLE

För att skapa tabeller i vår databas använder vi SQL-kommandot CREATE TABLE.

## Syntax:

```
CREATE TABLE tabellnamn  
(  
    kolumn1 datatyp,  
    kolumn2 datatyp,  
    kolumn3 datatyp ...  
)
```

Tabellnamn som börjar med  
# anger lokal temporärtabell  
## anger global temporärtabell

Observera paranteser  
och inte måsvingar.

## Exempel:

```
CREATE TABLE kund  
(  
    fornamn varchar(25),  
    efternamn varchar(30),  
    alder tinyint  
)
```



fornamn	efternamn	alder



## CREATE TABLE FORTS.

Förutom att enbart ange datatyper kan vi även tala om vilket/vilka fält som ska vara primärnyckel, vilka fält som inte får vara null, vilket fält som ska ha räknare, constraints med mera.

Funktion	DDL kod
PRIMARY KEY (ett fält)	<code>kol datatyp PRIMARY KEY</code>
INDEX PK	<code>kol datatyp PRIMARY KEY CLUSTERED</code>
NOT NULL	<code>kol datatyp NOT NULL</code>
UNIQUE	<code>kol datatyp UNIQUE</code>
IDENTITY	<code>kol datatyp IDENTITY(startvärde, uppräkning)</code>
DF CONSTRAINT	<code>kol datatyp CONSTRAINT [DF-namn] DEFAULT (0)</code>
CK CONSTRAINT	<code>kol datatyp CONSTRAINT [CK-namn] CHECK (len([fält])&gt;=(0))</code>
SAMMANSATT Pk	<code>PRIMARY KEY CLUSTERED (fält1,fält2)</code>
BERÄKNANDE	<code>Förfallodat AS DATEADD(day,Betvillkor,Datum)</code>

Pk och UNIQUE kan inte användas på samma kolumn. Pk är UNIQUE!  
Sammansatt PK kan inte sättas förrän fälten har skapats!



## EXEMPEL CREATE TABLE

Nedan skapas en typisk tabell med SQL. Exemplet är gjort för SQL Server.

### Exempel:

```
CREATE TABLE member
(
    memberID INT PRIMARY KEY NOT NULL IDENTITY(1,1),
    Firstname varchar(50) NOT NULL,
    Lastname varchar(50) NOT NULL,
    Age tinyint NOT NULL CONSTRAINT [CK_mem] CHECK ( [age])>(0)),
    Orgnr char(11) UNIQUE NOT NULL,
    Comment varchar(200)
)
```

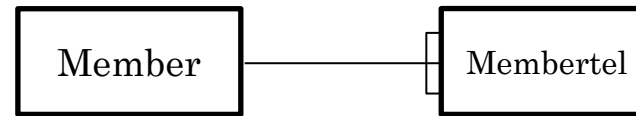
Flera constraint på samma fält:

```
Age tinyint NOT NULL
CONSTRAINT [CK_mem] CHECK ([age])>(0))
CONSTRAINT [DF_mem] DEFAULT = (1)
```

memberID	Firstname	Lastname	Age	Orgnr	Comment
1	Karl	Larsson	86	556323-9988	Börjar bli lite till åren ko...
2	Nisse	Nilsson	45	459812-6541	Har inte erlagt medlems...
3	Janne	Jansson	12	654513-2356	Är han inte lite ung för d...
4	Kalle	Anka	56	473225-6547	Han heter ju som ankan i...



## 1:N RELATION, SKAPA BARNTABELLEN



```
CREATE TABLE membertel
(
  membertelID INT PRIMARY KEY CLUSTERED NOT NULL IDENTITY(1,1) ,
  Kontakt varchar(50) NOT NULL,
  memberID INT NOT NULL
)
```

### Om sammansatt nyckel:

```
CREATE TABLE membertel
(
  memberID INT NOT NULL
  membertelID INT NOT NULL IDENTITY(1,1) ,
  Kontakt varchar(50) NOT NULL,
  PRIMARY KEY CLUSTERED (memberID,membertelid)
)
```



## ALTER TABLE

ALTER TABLE används då vi vill förändra en existerande tabell. Syntaxen för hur detta görs skiljer mycket från databashanterare till databashanterare. Denna slide behandlar MS SQL Server.

### Syntax, ändra kolumn (SQL Server):

```
ALTER TABLE tabellnamn  
ALTER COLUMN kolumnnamn ...
```

### Syntax, addera kolumn (SQL Server):

```
ALTER TABLE tabellnamn  
ADD kolumnnamn ...
```

### Syntax, radera kolumn(er) (SQL Server):

```
ALTER TABLE tabellnamn  
DROP kolumnnamn1, kolumnnamn2 ...
```

Syntaxen efter ALTER COLUMN följer samma regler som när en ny tabell skapas.





## EXEMPEL ALTER TABLE

Lägger till fältet Rabatt i tabellen Artikel

**Exempel:**

```
ALTER TABLE Artikel  
  ADD Rabatt Decimal(2,2) NULL
```

Tar bort fältet Rabatt i tabellen Artikel

**Exempel:**

```
ALTER TABLE Artikel  
  DROP COLUMN Rabatt
```



## LÄGGA TILL EN CHECK CONSTRAINT O DEFAULT CONSTRAINT

### Exempel:

```
CREATE TABLE member
(
  memberID INT PRIMARY KEY NOT NULL IDENTITY(1,1),
  Firstname varchar(50) NOT NULL,
  Lastname varchar(50) NOT NULL,
  Age tinyint NOT NULL CONSTRAINT [DF_member_age] DEFAULT (0),
  Orgnr char(11) UNIQUE NOT NULL,
  Comment varchar(200)
)
GO

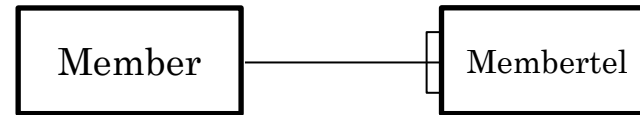
ALTER TABLE
ADD CONSTRAINT [CK_member] CHECK ((len([orgnr])=(11)))
GO
```

Radera en constraint:

```
ALTER TABLE member
  DROP CONSTRAINT [CK_member]
GO
```



## SKAPA EN FOREIGN KEY CONSTRAINT



### Exempel:

```
CREATE TABLE membertel
(  membertelID INT PRIMARY KEY NOT NULL IDENTITY(1,1),
  Kontakt varchar(50) NOT NULL,
  memberID INT NOT NULL,
  CONSTRAINT [FK_Membertel_Member] FOREIGN KEY ([Memberid])
  REFERENCES [member] ([Memberid])
  ON DELETE CASCADE
)
GO
```

NOACTION är default

### Ändring av tabell:

```
ALTER TABLE [MemberTel]
ADD CONSTRAINT [FK_Membertel_Member] FOREIGN KEY [Memberid])
REFERENCES [member] ([Memberid])
ON DELETE CASCADE
ON UPDATE NOACTION
GO
```



## CREATE INDEX

För att indexera på en kolumn använder vi följande kommando:

### Syntax:

```
CREATE INDEX indexnamn ON tabell  
(kolumnnamn1, kolumnnamn2...)
```

Om vi vet att vi ofta gör sökningar på Produktnamn i vår databas (WHERE Produktnamn=...) kan det vara lämpligt att skapa ett index mot denna kolumn.

ProduktID	Artnr	Produktnamn	typID
1	3443	Stolen Gunnar	12
2	5423	Bordet Jan	21
3	6534	Pallen Gun	12
4	6545	Kassen Jarl	2
5	9875	Kvitto	

### Exempel:

```
CREATE INDEX pnamn ON Produkt  
(Produktnamn) ASC
```

Förhoppningsvis kommer våra sökningar att gå fortare i framtiden.

```
CREATE UNIQUE NONCLUSTERED INDEX pnamn ON Produkt  
(Produktnamn) ASC
```



## ANVÄNDARDEFINIERADE DATATYPER

### SQL Server tillåter användardefinierade datatyper

- Synonymer för existerande standard-datatyper
- Ger enklare och mer konsistent hantering av kolumner av samma typ i flera tabeller. T ex Personnr, Postnr kolumner får samma datatyp
- Kan inte tas bort (drop) eller ändras när de väl är använda.

```
CREATE TYPE PersNR  
FROM char(11) NOT NULL;
```

### Vi har nu fått en ny datatyp som heter PersNr

- Kan användas i alla tabeller som använder en personnummerkolumn.
- Konsistent lagring! Alla kolumner med personnummer lagras på samma sätt och blir kompatibla.

Man kan inte räkna med att alla databashanterare stödjer användardefinierade datatyper och bör inte användas om man eftersträvar portabilitet.



# DROP

Med kommandot **DROP** kan vi radera objekt såsom databaser, tabeller, fält, index m fl.

## Syntax, radera en databas:

```
DROP DATABASE databasnamn
```

## Syntax, radera en tabell:

```
DROP TABLE tabellnamn
```

## Syntax, radera ett index:

```
DROP INDEX tabellnamn.indexnamn
```

Du kan inte ta bort en databas om det finns en aktiv uppkoppling mot den.

**Var försiktig med ovanstående kommandon, du får ingen varning i SQL!**

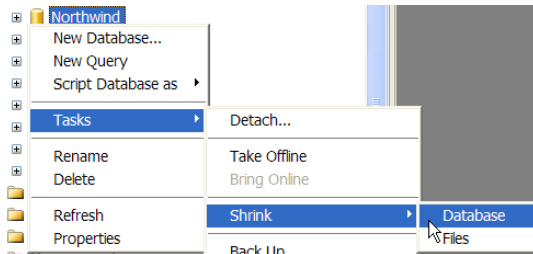


## KRYMPA DATABASER - SHRINK

När data raderas behåller filen sin storlek! - Slöseri med diskutrymme!

Fragmenterad databasfil:

### Krympa databaser med Management Studio

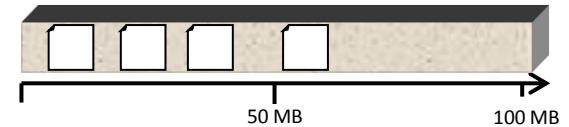


### Krympa filer med TSQL

DBCC SHRINKDATABASE (Faktura, 5)

DBCC = Database Console Commands

Före SHRINK:



Efter SHRINK

Olika sätt  
att krympa  
filer

**DEFAULT**



**NOTRUNCATE**



**TRUNCATEONLY**



**EMPTYFILE**



Flyttar data till annan fil i filgruppen.  
Endast tomma filer kan tas bort.

-- Man kan krympa (SHRINK) både databasen och transaktionsloggen

-- Krymper Faktura och lämnar 5% kvar