**15V**

# Lektion 1

Tirsdag den 23. august 2016

**Litteratur:** [SQL Server 2005 Express in 24 hours] page 181 - 191
A quick guide to Transact-SQL

**Emner:**

Præsentation af fag og hinanden

Gennemgang af lektionsplan.

Transact-SQL programming
- Hvad er det og hvor kan man bruge det
- Hvad er en BATCH og hvad er et script
- Variable erklæring og brug
- IF ELSE
- BEGIN END
- CASE
- WHILE
- PRINT
- CONVERT
- EXECUTE
- Cursors
- Table-variable

Scripts kontra Management-tool

**Opgaver:** Exercise 1.1 – 1.8 (vi når givetvis ikke alle opgaver)

**Læsning til næste gang:**

Overheads om nye ting i SQL Server 2005/2008/2012/2014/2016

**Bemærkninger:**

## Exercise 1.1

Denne øvelse er en lille repetitionsopgave i SQL. Øvelsen skal laves **uden computer.**

Du vil på klassen få udleveret nogle stumper SQL og skal for hver angive, om resultatet vil blive

- En parse/kompileringsfejl
- En runtime fejl
- En logisk fejl (dvs SQL kører, men resultatet vil være forkert)
- At SQL'en virker og angiv da, hvad resultatet vil være

Alle SQL-stumperne kører på eksempeldatabasen, der vil være vedhæftet i overblik og i create tables + inserts.

## Exercise 1.2

Herunder er vist tabeldefinitionen for en tabel kunde og derefter tre forskellige definitioner af en fakturatabel.

```
create table kunde
(
kundeid int primary key,
kundenavn varchar(20),
postnr char(4)
)
insert into kunde values(1,'Ib','8270'),(2,'Bo','8000'),(3,'Claus','8240'),
(4,'Dan','8000'),(5,'Elo','8270'),(6,'Frede','8270')
--
-- 3 forskellige definitioner af faktura
--
-- definition 1
create table faktura
(
fakturaid int primary key,
kundeid int foreign key references kunde,
fakturadato date
)
-- definition 2
create table faktura
(
fakturaid int primary key,
kundeid int foreign key references kunde on delete cascade on update cascade,
fakturadato date
)
-- definition 3
create table faktura
(
fakturaid int primary key,
kundeid int foreign key references kunde on delete set null,
fakturadato date
)
insert into faktura values(1,5,'2016.07.09')
insert into faktura values(2,2,'2016.01.09')
insert into faktura values(3,5,'2016.01.13')
insert into faktura values(4,null,'2016.01.13')
```

Udfyld skemaet med hvad der vil ske hvis du kører de fire nævnte SQL-sætninger med de tre definitioner af fakturetabellen

| | Definition 1 | Definition 2 | Definition 3 |
|---|---|---|---|
| `insert into faktura values(5,9,'2016.01.13')` | | | |
| `update faktura set kundeid = 23 where fakturaid = 1` | | | |
| `delete from kunde where kundeid = 2` | | | |
| `update kunde set kundeid = 13 where kundeid = 5` | | | |

## Exercise 1.3

Betragt denne sql-sætning (på kunde-tabellen fra opgave 1.2).

```
select postnr,count(*)
from kunde
where kundenavn not like '%u%'
group by postnr
having count(*) > 1
```

Hvad laver den?

Som du kan se, er der en betingelse i både where-delen og i having-delen. Hvilken rolle spiller de to betingelser.

## Exercise 1.4

Denne opgave tager udgangspunkt i eksempeldatabasen

Find den højeste løn, nogen i person-tabellen har.

Find den næsthøjeste løn, der findes i person-tabellen.

Find den tredje-højeste løn.

Find navnet på den, der har den højeste løn.

## Exercise 1.5

Make a table persons with attributes name and age.

Insert a least 5 records with different ages

Make a batch, that can print the number of persons with ages in the interval 0-9, 10-19, 20-29 and so on.

The printning from the batch could look like:

| age | no. of persons |
|---|---|
| 0-9 | 2 |
| 10-19 | 1 |
| 20-29 | 0 |
| ... | |
| ... | |

Change the batch, so that it stops, when there are no more persons with an age above the values all-ready presented.

Hint WHILE

## Exercise 1.6

This shows a Batch with a cursor.
What does the batch do?

Can you make the same result in a much more easy way?

Right now the batch assumes, that the table Person is not empty. Can you remove the need for that assumption

```
declare @counter int
declare @totalsalary int
set @counter = 0
set @totalsalary = 0
declare p cursor
for select salary from person
declare @salary int
open p
fetch p into @salary
while @@fetch_status != -1
begin
  set @totalsalary = @totalsalary + @salary
  set @counter = @counter +1
  fetch p into @salary
end
close p
deallocate p
select @totalsalary/@counter
```

## Exercise 1.7

A system holds the grades for the students, we will use the old grades in Denmark 00, 03, 5, 6, 7, 8, 9, 10, 11 and 13. A student always has 5 grades.

In these old days a student would pass the high-school exam if the following two rules were both OK

- The average should be at least 5.5
- The sum of the two lower grades plus the average of the rest should be at least 13.

Make a list showing the students name and if he/she had passed.

On fronter you can find a script with the tables, some inserts and the head of a function (exercise1.5).

## Exercise 1.8

The database stores its own data in so-called system tables. If you want to see a list of the tables you have created yourself – try the SQL query

select name from sysobjects where type = ‘u’ and category = 0

Use this query to make a batch, that shows the name and number of records for each table.(Hint! you must use cursors and execute)