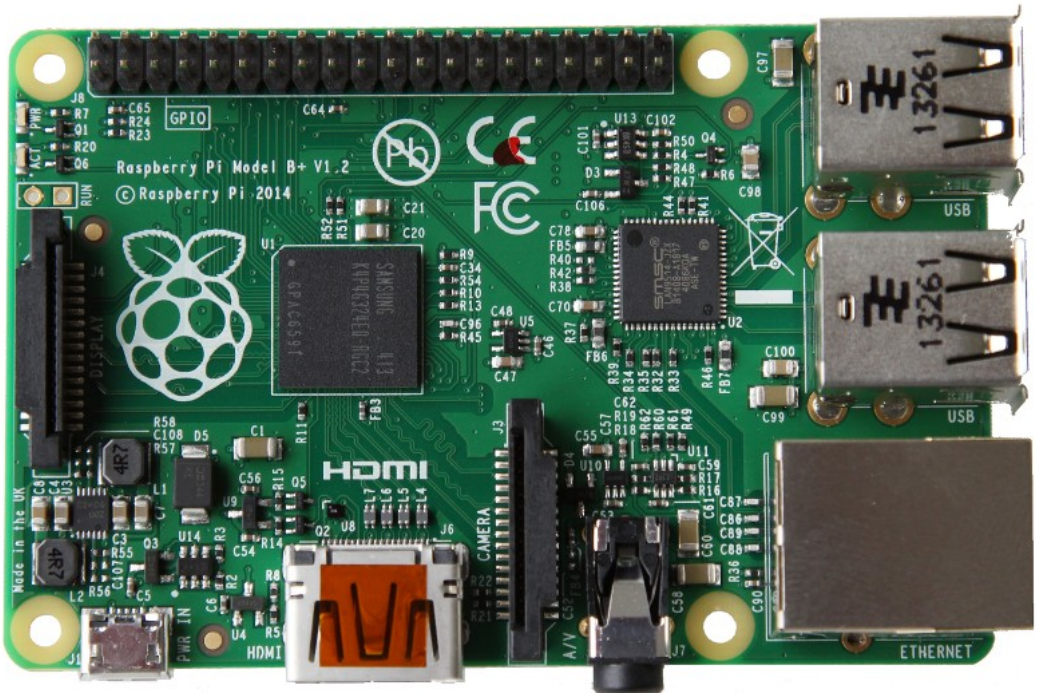
Introduktion til SQL



Henrik Kressner
Marts 2019
Denne og andre kan findes på: https://synkro.dk/bog

Indholdsfortegnelse

1. Indledning....2
2. Hvad er en database....3
3. SQL basale kommandoer....5
3.1 Søg i en SQL database....6
3.3 Indsæt en ny record....7
3.4 Fjern en record....7
3.5 Rette i en record....7
3.6 Opgaver....8
4. Betingede operationer....9
4.1 Udvidet søgeteknik....9
4.2 Rette en record....11
4.3 Fjern en record....11
4.4 Sortering....12
4.5 Statistiske funktioner....13
4.6 Opgaver....14
Bilag A....15

1. Indledning

SQL er et programmerings sprog til brug for databaser der blev standardiseret sidst i 1980'erne, det er stadig det mest udbredte værktøj til håndtering af databaser.

Dette kompendie giver en kortfattet introduktion til SQL. Eksemplerne er udført med mariadb på en Raspberry Pi. Der kan være mindre justeringer i hvordan du logger ind på databaseserveren, men når du først er inde, er det lige meget hvilken version du bruger, og hvilken platform du kører på, bare det er SQL i CLI.

Forudsætninger: Du skal kunne begå dig i et CLI.

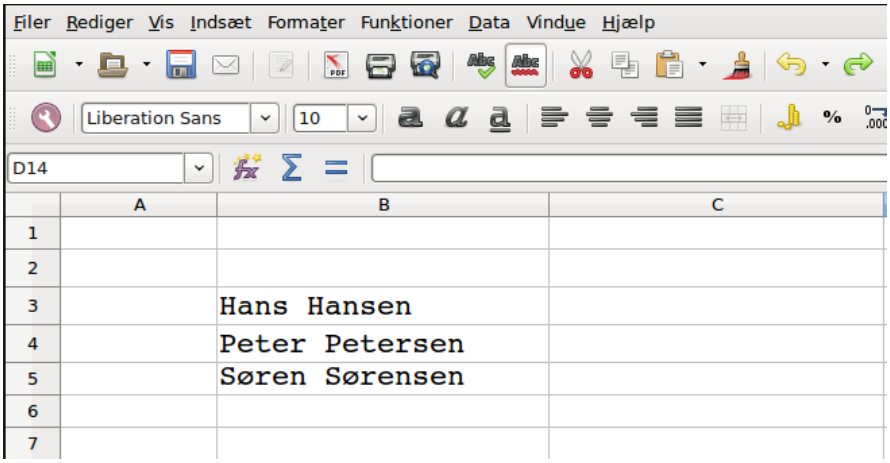
Hvis du vil have de samme eksempler som her, kan du gå til bilag A og se hvordan du opretter databasen.

Fejlmeldinger, kritik og forslag er velkommen på nedenstående kontaktadresse.

Henrik Kressner
kressner@synkro.dk

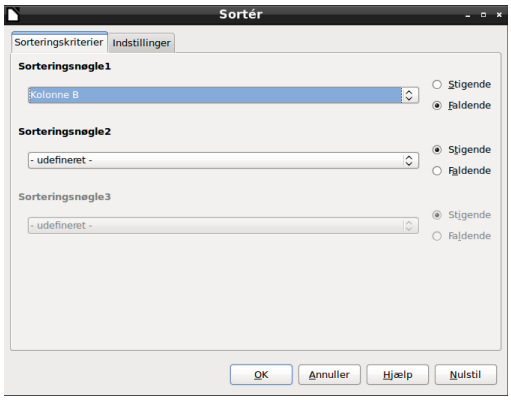
2. Hvad er en database

En database er en samling af data, lad os se på et eksempel. Vi kan benytte et regneark som en database:



Figur 2.1

I dette tilfælde bruger vi kolonne B til en liste over personnavne. Når vi har data inde i en database kan vi begynde at operere på dem, vi kan eksempelvis sortere dem alfabetisk. I LibreOffice gøres det ved først at markere kolonne B, det er jo der vores data er, og derefter vælge menupunkt Data → Sorter, så kommer denne menu:



Figur 2.2

På menuen kan man vælge om man vil sortere faldene eller stigende, og da vi har valgt kolonne B som sorteringsnøgle, vil kolonne B blive sorteret alfabetisk, når vi trykker på OK.

Dette eksempel er en ”flad” database, hvilket medfører vi eksempelvis ikke kan sortere på efternavn, for at kunne det skal vi oprette en record.

En record er en samling af data, der er altså en relation mellem de enkelte data i en record. En database der kan håndtere relationer, kaldes en relationsdatabase.

På figur 2.3 har vi ændret databasen, vi har nu en record per person, det betyder at fornavn og efternavn har hver sin kolonne, men data ”hører sammen” vandret.

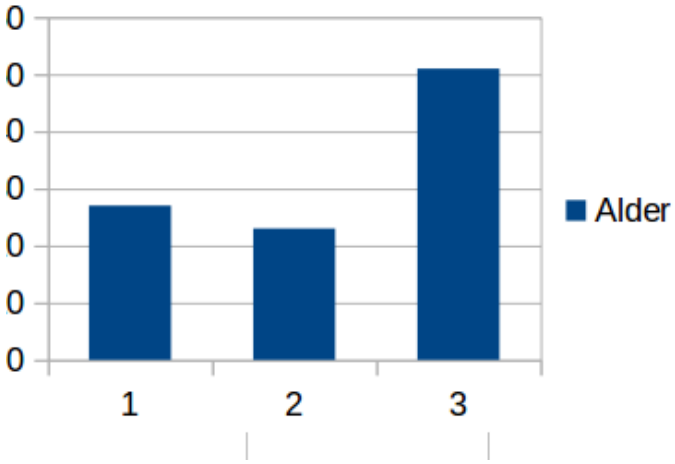
B	C
Fornavn	**Efternavn**
Søren	Sørensen
Peter	Petersen
Hans	Hansen

Figur 2.3

Det betyder vi nu kan sortere på både fornavn og efternavn, vi kan endda sortere på efternavn, og hvis to personer har samme efternavn, kan vi lade fornavnet bestemme sorteringen.

Vi kan udvidde med en kolonne for alder, og endda generere et diagram over aldersfordelingen i databasen, i LibreOffice gøres det ved at vælge menupunkt Indsæt → Objekt → Diagram.

B	C	D
Fornavn	**Efternavn**	**Alder**
Søren	Sørensen	27
Peter	Petersen	23
Hans	Hansen	51



Figur 2.4

I ovenstående eksempel vil filnavnet også være navnet på databasen. Regnearkets navn, i dette tilfælde ark1, vil være navnet på den ene tabel vi bruger i den database, hver linje er en record.

Alt sammen nemt og smart, men der er en hage, det er ikke nemt at dele data, det kræver en databaseserver. Mariedb er en databaseserver, den logger vi ind på i næste afsnit.

3. SQL basale kommandoer

Du logger ind på en mariadb databaseserver ved at skrive:

```
$ mariadb -u <brugernavn> -p
```

Herefter vil databaseserveren bede dig indtaste password, hvis du taster det korrekt, og slutter med at trykke enter, bliver du logget ind. Hvis dit bruger navn er alfaalfa, og dit password er alfaalfa, vil din loginsekvens se således ud.

```
$ mariadb -u alfaalfa -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.1.37-MariaDB-0+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye
root@MuleMaster:~# mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.1.37-MariaDB-0+deb9u1 Raspbian 9.0

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Herefter går vi ud fra der er en database på serveren kaldet Firma[1], i den database er der en tabel over de ansatte, som vi har kaldt ansatte. Inde i databasen er en tabel med en record for hver ansat, bestående af fornavn, efternavn og alder.

1 For generering af databasen, se Bilag A.

3.1 Søg i en SQL database

Efter at være logget ind skal man vælge den database man ønsker at arbejde med, i dette tilfælde hedder den Firma:

```
MariaDB [(none)]> use Firma;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [Firma]>
```

Herefter opererer vi udelukkende på den database der hedder Firma. (Bemærk: Der er forskel på små og store bogstaver) I den database har vi en tabel der hedder ansatte, vi kan vælge at se alt hvad vi ved om de ansatte ved at skrive:

```
MariaDB [Firma]> select * from ansatte;
+------+-------------+-----------+-------+
| id   | fornavn     | efternavn | alder |
+------+-------------+-----------+-------+
|    1 | Frederiksen | Frederik  |    19 |
|    2 | Hans        | Hansen    |    22 |
|    3 | Birte       | Birtesen  |    20 |
|    4 | Tulle       | Tullesen  |    23 |
|    5 | Jens        | Jensen    |    22 |
|    6 | Arne        | Arnesen   |    20 |
|    7 | Nina        | Ninasen   |    21 |
|    8 | Poul        | Poulsen   |    23 |
+------+-------------+-----------+-------+
8 rows in set (0.00 sec)

MariaDB [Firma]>
```

Vi har lavet en database forespørgelse med SQL kommandoen ”select”. Stjernen (asterix eller *) betyder alt (i dette tilfælde, ”alle kolonner”), resten af linjen betyder vi vil læse fra den tabel der hedder ansatte.

Kolonnen kaldet id, bruges til entydigt at betegne hver record, det er normalt et fortløbende tal.

3.3 Indsæt en ny record

Vi kan indsætte en ny record i databasen med insert kommandoen, det kan gøres således:

```
MariaDB [Firma]> insert into ansatte values (9, "Pia", "Piasen", 25);
Query OK, 1 row affected (0.01 sec)

MariaDB [Firma]>
```

Hvis du herefter tester ved at skrive:

```
MariaDB [Firma]>  select * from ansatte;
```

Kan vi se Pia er kommet ind som den sidste (den niende) record i databasen.

3.4 Fjern en record

Vi kan fjerne en record med SQL kommandoen delete, det kan se således ud:

```
MariaDB [Firma]> delete from ansatte where id = 4;
Query OK, 1 row affected (0.01 sec)

MariaDB [Firma]>
```

Her bruger vi id til at finde lige præcis den record vil vil rette, det er den sikreste metode.

3.5 Rette i en record

Hvis vi nu finder ud af Pia ikke hedder Piasen til efternavn, men Hansen, kan vi rette hendes record med SQL kommandoen update. Det kan se således ud:

```
MariaDB [Firma]> update ansatte set efternavn = "Hansen"
where id = 9;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [Firma]>
```

Igen kan vi tjekke ved at skrive:

```
MariaDB [Firma]>  select * from ansatte;
```

Som det ses af eksemplet kan vi dele linjen hvis den bliver for lang, ved at trykke enter, det er først når der er sat et semikolon og trykket enter, at databasen forsøger at udføre kommandoen,

3.6 Opgaver

3.6.1. Opret dig selv i databasen, tjek ved at liste databasen.

3.6.2 Slet Arne fra databasen, tjek ved at liste databasen.

3.6.3 Opret Arne igen, men nu med efternavnet Mortensen, tjek ved at liste databasen.

3.6.3 Ret fejlen i record nummer 1, så fornavn og efternavn kommer i de rette felter, tjek ved at liste databasen.

4. Betingede operationer

SQL kommandoer kan sættes sammen til mere komplekse udtryk. Det gøres med SQL kommandoen where og ved at bruge logiske operatorer:

>	Større end
>=	Større end og lig med
<	Mindre end
<=	Mindre end og log med
<>	Forskellig fra
=	Lig med
OR	Enten eller
AND	Både og

4.1 Udvidet søgeteknik

Hvis vi kun vil se efternavnet på alle i databasen kan det gøres således:

```
MariaDB [Firma]> select efternavn from ansatte;
+-----------+
| efternavn |
+-----------+
| Frederik  |
| Hansen    |
| Birtesen  |
| Tullesen  |
| Jensen    |
| Arnesen   |
| Ninasen   |
| Poulsen   |
+-----------+
8 rows in set (0.00 sec)

MariaDB [Firma]>
```

Vi kan søge på flere felter ved at liste dem adskilt af komme i søgningen:

```
MariaDB [Firma]> select efternavn, alder from ansatte;
```

Resultatet kommer ud i den rækkefølge vi beder om i select. I dette tilfælde kommer kolonnen efternavn først, og alder sidst. Hvis vi bytter om på rækkefølgen i forespørgelsen, kommer de ud i modstat rækkefølge.

Vi kan søge alle der hedder Hans til fornavn ved at skrive:

```
MariaDB [Firma]> select * from ansatte where fornavn = "Hans";
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
+------+---------+-----------+-------+
1 row in set (0.00 sec)
```

Her bruger vi operatoren lig med (=) til at fortælle databasen, vi vil se alle der hedder Hans til fornavn.

Vi kunne i stedet vælge kun at se dem der er ældre end 21 år:

```
MariaDB [Firma]> select * from ansatte where alder > 21;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
|    5 | Jens    | Jensen    |    22 |
|    8 | Poul    | Poulsen   |    23 |
|    9 | Pia     | Hansen    |    25 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

Vi kan søge på flere forskellige personnavne ved at bruge or (eller) operatoren. Hvis vi vil se alle med fornavn Hans eller Jens kan vi skrive:

```
MariaDB [Firma]> select * from ansatte where fornavn = "Hans"
or fornavn = "Jens";
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
|    5 | Jens    | Jensen    |    22 |
+------+---------+-----------+-------+
2 rows in set (0.01 sec)
```

Igen har vi trykket enter fordi linien blev for lang, men kommandoen bliver først udført når vi sætter at semikolon, efterfulgt af et tryk på enter.

4.2 Rette en record

Vi kunne også udvælge en record ved at gøre brug af and operatoren til at præcisere hvem det er vi vil rette, det kan se således ud:

```
MariaDB [Firma]> update ansatte set efternavn = "Hansen" where  fornavn = "Pia"
and efternavn = "Piasen";
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

4.3 Fjern en record

Vi kan også bruge where og et logisk udtryk, til at udpege den record vi vil slette, det kan se således ud:

```
MariaDB [Firma]> delete from ansatte where fornavn = "Jens" and efternavn =
"Jensen";
Query OK, 1 row affected (0.03 sec)
```

Vi kunne også være lidt mere grov og slette alle under 20 år:

```
MariaDB [Firma]> delete from ansatte where alder <= 20;
Query OK, 3 rows affected (0.02 sec)

MariaDB [Firma]> select * from ansatte;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
|    7 | Nina    | Ninasen   |    21 |
|    8 | Poul    | Poulsen   |    23 |
|    9 | Pia     | Hansen    |    25 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

Bemærk: Når du arbejder i SQL som vi gør her, altså på kommandolinien (CLI), er der ingen fortrydelsesfunktioner og ingen der spørger om du mener det. Der findes naturligvis GUI værktøjer der hjælper med det, men de bliver udviklet ved hjælp af det vi arbejder med her.

Det er altså op til databaseprogrammøren at udvikle et brugerinderface med faciliteter som, ”Mener du virkelig du vil slette” og ”Vil du gendanne det du slettede for lidt siden”.

4.4 Sortering

Vi kan sortere output ved at tilføje SQL kommandoen ”order by”. Hvis vi vil sortere på efternavn vil det se således ud:

```
MariaDB [Firma]> select * from ansatte order by efternavn;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
|    9 | Pia     | Hansen    |    25 |
|    7 | Nina    | Ninasen   |    21 |
|    8 | Poul    | Poulsen   |    23 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

Nu har vi to der hedder Hansen til efternavn, hvis vi sekundært vil sortere på fornavn vil det se således ud:

```
MariaDB [Firma]> select * from ansatte order by efternavn, fornavn;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    2 | Hans    | Hansen    |    22 |
|    9 | Pia     | Hansen    |    25 |
|    7 | Nina    | Ninasen   |    21 |
|    8 | Poul    | Poulsen   |    23 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

Eller vi kan sortere efter alder, laveste alder først:

```
MariaDB [Firma]> select * from ansatte order by alder ASC;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    7 | Nina    | Ninasen   |    21 |
|    2 | Hans    | Hansen    |    22 |
|    8 | Poul    | Poulsen   |    23 |
|    9 | Pia     | Hansen    |    25 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

Eller vi kan sortere efter alder, laveste alder sidst:

```
MariaDB [Firma]> select * from ansatte order by alder DESC;
+------+---------+-----------+-------+
| id   | fornavn | efternavn | alder |
+------+---------+-----------+-------+
|    9 | Pia     | Hansen    |    25 |
|    8 | Poul    | Poulsen   |    23 |
|    2 | Hans    | Hansen    |    22 |
|    7 | Nina    | Ninasen   |    21 |
+------+---------+-----------+-------+
4 rows in set (0.00 sec)
```

4.5 Statistiske funktioner

SQL Indeholder som standard følgende statistiske funktioner:

Avg:	Giver den gennemsnitlige (average) værdi.
Count:	Tæller antal.
Max:	Visere maksimalværdien.
Min:	Viser mindste værdien.
Sum:	Viser summen.

For at finde gennemsnittet af et talsæt bruge vi AVG kommandoen. Vi kan bruge den til at finde den gennemsnitlige alder i vores Firmadatabase, ved at skrive:

```
MariaDB [Firma]> select avg(alder) from ansatte;
+------------+
| avg(alder) |
+------------+
|    22.7500 |
+------------+
1 row in set (0.00 sec)
```

Vi kan forespørge på flere parametre ved at adskille dem med et komma, og samtidig regne på dem, det kan se således ud:

```
MariaDB [Firma]> select min(alder), max(alder), avg(alder) from ansatte;
+------------+------------+------------+
| min(alder) | max(alder) | avg(alder) |
+------------+------------+------------+
|         21 |         25 |    22.7500 |
+------------+------------+------------+
1 row in set (0.00 sec)
```

Resultatet kommer ud i den rækkefølge vi beder om, fordi vi starter med `min(alder)` kommer den kolonne ud først.

4.6 Opgaver

4.6.1 Udvid databasen med mindst 5 personer.

4.6.2 List alle der hedder Hansen til efternavn.

4.6.3 List alle der hedder Hansen til efternavn, output skal være alder og fornavn, i den rækkefølge.

4.6.4 Fjern en af de record du oprettede under 4.6.1.

4.6.5 Lav Nina's efternavn om til Schandorf

4.6.6 List alle sorteret baglæns på efternavn. (Med baglæns menes Å kommer først, A til sidst.)

4.6.7 Beregn gennemsnitsalderen på alle der hedder Hansen til efternavn.

4.6.8 List alder hvis alder er større end gennemsnitsalderen.

Bilag A

Hvis du vil bruge den samme database som dette kompendie bruger, kan det gøres det ved at skabe en tekstfil med navn create.sql og give den følgende indhold.

```
CREATE DATABASE Firma;

CREATE USER "alfaalfa"@"localhost" IDENTIFIED BY "alfaalfa";
GRANT ALL PRIVILEGES ON Firma . ansatte TO "alfaalfa"@"localhost";
CREATE USER "betabeta"@"localhost" IDENTIFIED BY "betabeta";
GRANT ALL PRIVILEGES ON Firma . ansatte TO "betabeta"@"localhost";
CREATE USER "cetaceta"@"localhost" IDENTIFIED BY "cetaceta";
GRANT ALL PRIVILEGES ON Firma . ansatte TO "cetaceta"@"localhost";
CREATE USER "detadeta"@"localhost" IDENTIFIED BY "detadeta";
GRANT ALL PRIVILEGES ON Firma . ansatte TO "detadeta"@"localhost";

USE Firma;
DROP TABLE IF EXISTS ansatte;
CREATE TABLE ansatte (id INT, fornavn VARCHAR(20), efternavn VARCHAR(30), alder INT);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (1, 'Frederiksen', 'Frederik', 19);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (2, 'Hans', 'Hansen', 22);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (3, 'Birte', 'Birtesen', 20);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (4, 'Tulle', 'Tullesen', 23);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (5, 'Jens', 'Jensen', 22);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (6, 'Arne', 'Arnesen', 20);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (7, 'Nina', 'Ninasen', 21);
INSERT INTO ansatte (id, fornavn, efternavn, alder) VALUES (8, 'Poul', 'Poulsen', 23);
SELECT * FROM ansatte;
```

create.sql

Når filen er gemt logger du ind i databasen med administrator rettigheder og skriver:

```
MariaDB [(none)]> source create.sql;
```

Efter lidt tid vil database og bruger blive oprettet, data lagt ind og hele databasen skrevet ud på skærmen.

Hele databasen og brugere kan slettes ved at eksekvere filen ved navn delete.sql med følgende indhold:

```
drop user "alfaalfa"@"localhost";
drop user "betabeta"@"localhost";
drop user "cetaceta"@"localhost";
drop user "detadeta"@"localhost";

drop database Firma;
```

delete.sql

Den kan efterfølgende eksekveres ved som administrator at skrive:

```
MariaDB [(none)]> source delete.sql;
```