# **MySQL databse kommandoer**

Dette er en SQL øvelse. Lever inn både SQL setningen du lager og resultatet.

MySQL workbench: se kap 30 i refman-8.0

**Oppgave 1**: Lag SQL setninger som:

1. Oppretter en ny database oblig1.

CREATE database oblig1;

1. Lager tabellen Film i kapittel 2, oppg. 1, i databasen oblig1

CREATE TABLE Film (

FilmNr INTEGER,

Tittel VARCHAR(100) NOT NULL,

År SMALLINT,

Land VARCHAR(50),

Sjanger VARCHAR(50),

Alder SMALLINT,

Tid SMALLINT,

Pris DECIMAL(8, 2),

CONSTRAINT FilmPN PRIMARY KEY (FilmNr)

) Engine=InnoDB;

1. Setter inn dataene i figur 2.16 i filmtabellen

INSERT INTO Film (FilmNr, Tittel, År, Land, Sjanger, Alder, Tid, Pris) VALUES

( 1, 'Casablanca', 1942, 'USA', 'Drama', 15, 102, '149.00'),

( 2, 'Fort Apachea', 1948, 'USA', 'Western', 15, 127, NULL),

( 3, 'Apocalypse Now', 1979, 'USA', 'Action', 18, 155, '123.00'),

( 4, 'Streets of Fire', 1984, 'USA', 'Action', 15, 93, NULL),

( 5, 'High Noon', 1952, 'USA', 'Western', 15, 85, '123.00'),

( 6, 'Cinema Paradiso', 1988, 'Italia', 'Komedie', 11, 123, NULL),

( 7, 'Asterix hos britene', 1988, 'Frankrike', 'Tegnefilm', 7, 78, '149.00'),

( 8, 'Veiviseren', 1987, 'Norge', 'Action', 15, 96, '87.00'),

( 9, 'Salmer fra kjøkkenet', 2002, 'Norge', 'Komedie', 7, 80, '149.00'),

(10, 'Anastasia', 1997, 'USA', 'Tegnefilm', 7, 94, '123.00'),

(11, 'La Grande bouffe', 1973, 'Frankrike', 'Drama', 15, 129, ' 87.00'),

(12, 'The Blues Brothers', 1980, 'USA', 'Komedie', 11, 124, '135.00'),

(13, 'Beatles: Help', 1965, 'Storbritania', 'Musikk', 11, 144, NULL

);

1. Henter Tittel, Sjanger og Pris for filmer som er produsert i 1988 eller seinere, sortert synkende på pris (=dyrest først)

SELECT Tittel, Sjanger, Pris FROM Film WHERE År >= 1988 ORDER BY Pris DESC;

1. Henter alle kolonner for filmer som ikke er til salgs (mangler pris), sortert på alder og sjanger.

SELECT \* FROM Film WHERE Pris IS NULL ORDER BY Alder, Sjanger;

1. Finner antall filmer som er til salgs i hver sjanger, og summen av prisen

SELECT Sjanger, COUNT(\*) AntallTilSalgs, SUM(Pris) AS SumPris FROM Film   
 WHERE Pris IS NOT NULL GROUP BY Sjanger ORDER BY Sjanger;

1. Setter inn en ny rad. Finn på data selv.

INSERT INTO Film (FNr, Tittel, År, Land, Sjanger, Alder, Tid, Pris)

VALUES (14, 'Avatar', 2008, 'USA', 'Drama', 13, 180, NULL);

1. Korrigerer tittelen på filmen «High Noon». Den heter egentlig «High Moon»  
   UPDATE Film SET Tittel = 'High Moon' WHERE Tittel = 'High Noon';
2. Legger på 10% på prisen på alle Action filmer  
   UPDATE Film set Pris = Pris\*1.1 WHERE Sjanger='Action';
3. Sletter filmen Anastasia  
   DELETE FROM Film WHERE Tittel = 'Anastasia';

**Oppgave 2**: Videobutikken har også en tabell Kunde. Se kap. 3, oppgave 4. Lag SQL setninger som:

1. Lager tabellen Kunde.  
   Kundetabellen skal inneholde et kundenummer (som er primærnøkkel), fornavn og etternavn, adresse og postnummer.

CREATE TABLE Kunde (

KundeNr INTEGER,

Fornavn VARCHAR(50) NOT NULL,

Etternavn VARCHAR(50) NOT NULL,

Adresse VARCHAR(100) NOT NULL,

PostNr CHAR(4) NOT NULL,

CONSTRAINT KundePN PRIMARY KEY (KundeNr)

) Engine=InnoDB;

1. Setter inn minst 3 rader med data i tabellen Kunde. En av dem heter Kari Mo.

INSERT INTO Kunde (KundeNr, Fornavn, Etternavn, Adresse, PostNr) VALUES

(5002, 'Paal', 'Aass', 'Lindemans gate79', '1711'),

(5007, 'Joakim', 'Laursen', 'Thomas Heftyes gate 39', '0015'),

(5009, 'Kari', 'Mo', 'Solheimgata 81', '0654');

**Noen stamkunder får lov til å leie uten å betale kontant. De får en faktura isteden.**

1. Lag en fakturatabell. Finn selv ut hvilke felter som bør være med i tabellen. Skriv noen få ord om hva de forskjellige feltene skal inneholde og en begrunnelse for hvorfor feltet skal kunne ha NULL eller ikke. Er det noen felter du vil ha standardverdier (default-verdier) i? Begrunn typen du har valgt for de forskjellige feltene. Hva velger du som PK? Bruker du FK?

CREATE TABLE Faktura (

FakturaNr INT,

KundeNr INT,

FilmNr INT,

Beløp DECIMAL(8,2) NOT NULL,

Dato DATE,

TekstTilKunden VARCHAR(200),

CONSTRAINT FakturaPK PRIMARY KEY (FakturaNr),

CONSTRAINT KundeNrFK FOREIGN KEY (KundeNr) REFERENCES Kunde (KundeNr),

CONSTRAINT FilmNrFK FOREIGN KEY (FilmNr) REFERENCES Film(FilmNr)

) Engine=InnoDB;  
Hva vil du gjøre for å få flere filmer på en faktura?  
Ser at forhold mellom Faktura og Film er M:N, lager ny tabell FakturaFilm:

CREATE TABLE FakturaFilm (

FakturaNr INT,

FilmNr INT,

Beløp DECIMAL(8,2) NOT NULL,

Dato DATE,

CONSTRAINT FakturaFilmPK PRIMARY KEY (FakturaNr, FilmNr),

CONSTRAINT FilmNrFK FOREIGN KEY (FilmNr) REFERENCES Film(FilmNr),

CONSTRAINT FakturaNrFK FOREIGN KEY (FakturaNr) REFERENCES Faktura(FakturaNr)

) Engine=InnoDB;  
Benytter ikke FakturaFilm-tabell i det videre.

1. Legger inn data i faktura tabellen. Legg inn minst 2 fakturaer for Kari Mo

INSERT INTO Faktura (FakturaNr, KundeNr, FilmNr, Beløp, Dato, TekstTilKunden) VALUES

(10,5002, 1, 200, '2021-09-10','Casablanca'),

(11,5009, 7, 180, '2021-09-20','Asterix hos britene'),

(12,5009, 4, 150, '2021-10-03','Streets of Fire')

;

1. Hent alle Kari Mo sine fakturaer.  
   SELECT \* FROM Faktura F

INNER JOIN Kunde K ON F.Kundenr=K.KundeNr

WHERE K.Fornavn='Kari' AND K.Etternavn='Mo';

1. Legger inn en forretningsregel, slik at fakturabeløpet ikke kan være negativt, og ikke større enn kr 10.000,- (CONSTRAINT)  
   ALTER TABLE Faktura ADD  
    CONSTRAINT CHECK(Beløp > 0 AND Beløp < 10000);
2. Prøv å legge inn en faktura rad med «ulovlig» pris. Hva skjer.  
   INSERT INTO Faktura (FakturaNr, KundeNr, FilmNr, Beløp, TekstTilKunden)   
    VALUES (15, 5009, 4, 11111, "Apocalypse Now - 123kr, CasaBlanca - 149kr");  
   Innsettingen godtas. Har for gammel MySQL.
3. Korriger SQLen din så prisen blir «lovlig og kjør SQLen på nytt

INSERT INTO Faktura (FakturaNr, KundeNr, FilmNr, Beløp, TekstTilKunden)   
 VALUES (15, 5009, 4, 123, "Apocalypse Now - 123kr, CasaBlanca - 149kr");

**NB! 2f, 2g og 2h fungerer kanskje ikke, men prøv. De krever en forholdvis ny utgave av MySQL.**

**Oppgave 3**

Finn dokumentasjonen på den innebygde funksjonen SUBSTRING\_INDEX på mariadb.com

1. Kopier beskrivelse av funksjonen til besvarelsen

<https://mariadb.com/kb/en/library/substring_index/>  


1. Vis at du kan bruke funksjonen

Denne returnerer strengen: Dette er, vet jeg

SELECT SUBSTRING\_INDEX('Dette er, vet jeg, en vanskelig avggjørelse',',',2)   
 AS AltFørAndreKomma;

Finn dokumentasjonen på den innebygde funksjonen SUBSTRING\_INDEX på mariadb.com

Kopier beskrivelse av funksjonen til besvarelsen

Returns the substring from string *str* before count occurrences of the delimiter *delim*. If *count* is positive, everything to the left of the final delimiter (counting from the left) is returned. If *count* is negative, everything to the right of the final delimiter (counting from the right) is returned. SUBSTRING\_INDEX() performs a case-sensitive match when searching for *delim*.

1. Vis at du kan bruke funksjonen

SELECT SUBSTRING\_INDEX('1-2-3-4','-', 1); - gir 1

SELECT SUBSTRING\_INDEX('1-2-3-4','-', 2); - gir 1-2

SELECT SUBSTRING\_INDEX('1-2-3-4','-', -3); gir 2-3-4  
  
dette kan brukes der man har nettsider hvor man ikke vil ha med .com eller annet info som ikke er relevant som like vel er med i dataen man behandler.

**Oppgave 4**

MySQL har en system database (schema) som heter mysql. Utfør «USE mysql;» så kan du jobbe mot denne basen. (Eller: finn denne databasen i **phpMyAdmin**.)

* 1. Hent alla data fra tabellen DB
  2. Finn fram til dokumentasjon for denne systemdatabasen
  3. Hent data fra et par av de andre tabellene

Løsningsforslag:

For å se systemdatabasene i MySQL: Edit | Preferences

Velg SQL-Editor. Under Sidebar, kryss av for «Show Metadata and Internal Schemas»

For å få informasjon om en database (schema): Velg en base i baseoversikten (schema) til venstre i Workbench. Da kommer det opp en skiftnøkkel og en i, Trykk på ien.

Et par eksempelr på hva systemdatabasen mysql inneholder:

* Tabellen mysql.user inneholder om informasjon om hver bruker, dvs. hva slags rettigheter brukeren har etc.
* Tabellen help\_topic inneholder en kort forklaring for hver emne, samt en url til hjelpsystemet til en server et sted (MariaDB sin server).

En annen systemdatabase er information\_schema

* Tabellen columns: Inneholder informasjon om alle kolonnene i databasene som du har opprettet.

Info:

For å se systemdatabasene i object-browseren i Workbench:

Edit | Preferences | SQL Editor | Show Metadata Schemata

Every MySQL is shipped with default system schemas/databases. Those are:

* **mysql** - is the system database that contains tables that store information required by the MySQL server
* **information\_schema** - provides access to database metadata (kap 24 i refman-5.7)
* **performance\_schema** - is a feature for monitoring MySQL Server execution at a low level (kap 26 i refman-8.0)
* **sys** - a set of objects that helps **DBA**s and developers interpret data collected by the Performance Schema (kap 27 i refman-8.0)