

Obligatorisk Oppgave 1

Karianne Glent s375078
06.02.2023

Innholdsliste

Oppgavesett 1	2
Oppgave 1.....	2
a).....	2
b).....	2
c)	3
Oppgave 2 (<i>W3School</i>)	3
SQL Select	3
SQL Where.....	3
SQL order by	4
SQL Insert	5
SQL NULL	5
SQL Update.....	6
SQL Delete	6
SQL Functions	7
SQL Like	7
SQL Wildcards.....	8
Oppgave 3: (Bruk wikipedia eller andre kilder og finn informasjon om).....	9
a) Hva er en database?.....	9
b) Hva er en relasjonsdatabase?	9
Oppgave 4.....	10
Gjør SQL oppgavene til film-oppgaven i kapittel 2 i boka skriftlig	10
Oppgavesett 2	13
Oppgave 1.....	13
Oppgave 2.....	14
Oppgave 3.....	14
Oppgave 4.....	15
Oppgavesett 3	18
Oppgave 2.....	27

Oppgavesett i DATA1500 Databaser

Oppgavesett 1

Oppgave 1

a) Regn om 10011011 fra binært til 10-tallsystem.

$$\begin{array}{r} 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \\ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \\ 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^7 \\ 1 + 2 + 8 + 16 + 128 \\ = \underline{\underline{155}} \\ * \text{ Potenser multiplisert med } 0 = 0 \end{array}$$

10011011 = 155

b) Skriv 537 binært.

$$\begin{array}{rcl} 537 : 2 = 268 & \text{rest } 1 \\ 268 : 2 = 134 & \text{rest } 0 \\ 134 : 2 = 67 & \text{rest } 0 \\ 67 : 2 = 33 & \text{rest } 1 \\ 33 : 2 = 16 & \text{rest } 1 \\ 16 : 2 = 8 & \text{rest } 0 \\ 8 : 2 = 4 & \text{rest } 0 \\ 4 : 2 = 2 & \text{rest } 0 \\ 2 : 2 = 1 & \text{rest } 0 \\ 1 : 2 = 0 & \text{rest } 1 \end{array}$$

537 = 1000011001

c) Skriv DATABASE binært.

Bokstav	ASCII Code	Binært
D	068	01000100
A	065	01000001
T	084	01010100
A	065	01000001
B	066	01000010
A	065	01000001
S	083	01010011
E	069	01000101

Database =

0100010001000001010101000100000101000010010000010101001101000101

Oppgave 2 (*W3School*)

SQL Select

1. Insert the missing statement to get all the columns from the Customers table.

SELECT * FROM Customers;

2. Select all the different values from the Country column in the Customers table.

SELECT DISTINCT Country FROM Customers;

3. Write a statement that will select the City column from the Customers table:

SELECT City FROM Customers;

SQL Where

1. Select all records where the City column has the value “Berlin”.

SELECT*FROM Customers

WHERE City = 'Berlin';

2. Use the NOT keyword to select all records where City is NOT “Berlin”.

```
SELECT * FROM Customers  
WHERE NOT City = 'Berlin';
```

3. Select all records where the CustomerID column has the value 32.

```
SELECT * FROM Customers  
WHERE CustomerID = 32;
```

4. Select all records where the City column has the value “Berlin” and the PostalCode column has the value 12209.

```
SELECT * FROM Customers  
WHERE City = 'Berlin'  
AND PostalCode = 12209;
```

5. Select all records where the City column has the value ‘Berlin’ or ‘London’.

```
SELECT * FROM Customers  
WHERE City = 'Berlin'  
OR City = 'London';
```

SQL order by

1. Select all records from the Customers table, sort the result alphabetically by the column City.

```
SELECT * FROM Customers  
ORDER BY City;
```

2. Select all records from the Customers table, sort the result reversed alphabetically by the column City.

```
SELECT * FROM Customers  
ORDER BY City DESC
```

3. Select all records from the Customers table, sort the result alphabetically, first by the column Country, then, by the column City.

```
SELECT * FROM Customers
```

```
ORDER BY Country, City;
```

SQL Insert

1. Insert a new record in the Customers table.

```
INSERT INTO Customers (
```

```
CustomerName,
```

```
Address,
```

```
City,
```

```
PostalCode,
```

```
Country)
```

```
VALUES (
```

```
'Hekkan Burger',
```

```
'Gateveien 15',
```

```
'Sandnes',
```

```
'4306',
```

```
'Norway');
```

SQL NULL

1. Select all records from the Customers where the PostalCode column is empty.

```
SELECT * FROM Customers
```

```
WHERE PostalCode IS NULL;
```

2. Select all records from the Customers where the PostalCode column is NOT empty.

```
SELECT * FROM Customers
```

```
WHERE PostalCode IS NOT NULL;
```

SQL Update

1. Update the City column of all records in the Customers table.

UPDATE CUSTOMERS

SET City = 'Oslo';

2. Set the value of the City columns to ‘Oslo’, but only the one where the Country column has the value “Norway”.

UPDATE Customers

SET City = 'Oslo'

WHERE Country = 'Norway';

3. Update the City value and the Country value

UPDATE Customers

SET City = 'Oslo',

Country = 'Norway'

WHERE CustomerID = 32;

SQL Delete

1. Delete all the records from the Customers table where the Country value is “Norway”.

DELETE FROM Customers

WHERE Country = 'Norway*';

2. Delete all the records from the Customers table.

DELETE FROM Customers;

SQL Functions

- 1. Use the MIN function to select the record with the smallest value of the Price column**

```
SELECT MIN(Price)
```

```
FROM Products;
```

- 2. Use an SQL function to select the record with the highest value of the Price column.**

```
SELECT MAX(Price)
```

```
FROM Products;
```

- 3. Use the correct function to return the number of records that have the Price value set to 18.**

```
SELECT COUNT(*)
```

```
FROM Products
```

```
WHERE Price = 18;
```

- 4. Use an SQL function to calculate the average price of all products.**

```
SELECT AVG(Price)
```

```
FROM Products;
```

- 5. Use an SQL function to calculate the sum of all the Price column values in the Products table.**

```
SELECT SUM(Price)
```

```
FROM Products;
```

SQL Like

- 1. Select all records where the value of the City column starts with the letter "a".**

```
SELECT * FROM Customers
```

```
WHERE City LIKE 'a%';
```

2. Select all records where the value of the City column ends with the letter "a".

```
SELECT * FROM Customers  
WHERE City LIKE '%a';
```

3. Select all records where the value of the City column contains the letter "a".

```
SELECT * FROM Customers  
WHERE City LIKE '%a%';
```

4. Select all records where the value of the City column starts with letter "a" and ends with the letter "b".

```
SELECT * FROM Customers  
WHERE City LIKE 'a%b';
```

5. Select all records where the value of the City column does NOT start with the letter "a".

```
SELECT * FROM Customers  
WHERE City NOT LIKE 'a%';
```

SQL Wildcards

1. Select all records where the second letter of the City is an "a".

```
SELECT * FROM Customers  
WHERE City LIKE '_a%'
```

2. Select all records where the first letter of the City is an "a" or a "c" or an "s".

```
SELECT * FROM Customers  
WHERE City LIKE '[acs]%'
```

3. Select all records where the first letter of the City starts with anything from an "a" to an "f".

```
SELECT * FROM Customers  
WHERE City LIKE '[a-f]%'
```

4. Select all records where the first letter of the City is NOT an "a" or a "c" or an "f".

```
SELECT * FROM Customers
```

```
WHERE City LIKE '[!acf]%';
```

Oppgave 3: (Bruk wikipedia eller andre kilder og finn informasjon om)

a) Hva er en database?

En database er en samling av strukturert informasjon eller data, som er lagret på et elektronisk medium som f.eks. en datamaskin (Bratbergsengen, Store Norske Leksikon, 2019) (Oracle, 2023). Selve samlingen av dataene er organisert etter en bestemt modell. Dette gjør at store mengder data kan bli tatt hånd om på en effektiv måte samtidig som det er motsetningsfritt og sikkert (Wikipedia, 2022).

b) Hva er en relasjonsdatabase?

En relasjonsdatabase er en database hvor data er organisert i tabeller (Bratbergsengen, Store Norske Leksikon, 2019). Dataene blir lagret i rader, som er fordelt på kolonner. Hver rad vil inneholde data for én enhet. Hver rad har også sin egen unike nøkkel som differensierer dem fra andre rader (Wikipedia, 2019).

Oppgave 4

Gjør SQL oppgavene til film-oppgaven i kapittel 2 i boka skriftlig (skriv SQL-svarene du tror er riktig inn i svardokumentet).

FNr	Tittel	År	Land	Sjanger	Alder	Tid	Pris
1	Casablanca	1942	USA	Drama	15	102	149.00
2	Fort Apache	1948	USA	Western	15	127	
3	Apocalypse Now	1979	USA	Action	18	155	123.00
4	Streets of Fire	1984	USA	Action	15	93	
5	High Noon	1952	USA	Western	15	85	123.00
6	Cinema Paradiso	1988	Italia	Komedie	11	123	
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	133	135.00
13	Beatles: Help	1965	Storbritania	Musikk	11	144	

Figur 2.15: Tabellen Film

1. Videobutikken Blanca både leier ut og selger filmer. Tabellen Film vist i figur 2.15, inneholder data om de videofilmene butikkene fører. FNr inneholder et unikt filmnummer og er valgt som primærnøkkel. De øvrige kolonnene inneholder filmtittel, produksjonsår, produksjonsland, sjanger, aldersgrense, spilletid i minutter og salgspris. Merk at noen filmer er rene utleiefilmer, og disse har ikke salgspris (inneholder et nullmerke).

Skriv SQL-spøringer for å vise følgende opplysninger fram filmtabellen:

a) All informasjon om filmer produsert i 1988.

```
SELECT * FROM Film
```

```
WHERE År = 1988;
```

b) Tittel på amerikanske filmer produsert på 1980-tallet.

```
SELECT FNr, Tittel  
FROM Film  
WHERE Land = 'USA' AND År BETWEEN 1980 AND 1989;
```

c) Komedier med aldersgrense under 10 år og spilletid under 130 minutter.

```
SELECT * FROM Film  
WHERE Sjanger = 'Komedie' AND Alder < 10 AND Tid < 130;
```

d) Tittel på alle action- og westernfilmer.

```
SELECT FNr, Tittel  
FROM Film  
WHERE Sjanger = 'Action' OR SJANGER = 'Western';
```

e) Alle produksjonsland, sortert og uten gjentakelser.

```
SELECT DISTINCT Land  
FROM Film  
ORDER BY Land;
```

f) Korteste og lengste spilletid innen hver sjanger.

```
SELECT Sjanger, MIN(Tid) AS Korteste, MAX(Tid) AS Lengste  
FROM Film  
ORDER BY Sjanger;
```

g) Antall filmer som ikke er til salgs.

```
SELECT COUNT(*) AS IkkeTilSalgs  
FROM Film  
WHERE Pris IS NULL;
```

h) Antall filmer under 100 kr.

```
SELECT COUNT(Pris) AS FilmerUnder100Kr  
FROM Film  
WHERE Pris < 100;
```

i) Filmer med tittel som slutter på ‘now’.

```
SELECT * FROM Film  
WHERE Tittel LIKE '%now';
```

j) Gjennomsnittspris for sjangre med flere enn 2 filmer.

```
SELECT Sjanger, AVG(Pris) AS GjennomsnittsPris  
FROM Film  
GROUP BY Sjanger  
HAVING COUNT (*) > 2;
```

k) Differansen mellom dyreste og billigste film innen hver sjanger.

```
SELECT Sjanger, MAX(Pris)-MIN(Pris) AS Differansen  
FROM Film  
GROUP BY Sjanger;
```

l) Totalt antall filmer og antall filmer til salgs, fordelt på produksjonsland.

```
SELECT Land, COUNT(*) AS TotaltAntall, COUNT(Pris) AS AntallTilSalgs  
FROM Film  
GROUP BY Land;
```

m) Antall år siden utgivelse for filmer eldre enn 50 år. Tips: Vedlegg A lister navn på noen datofunksjoner. Du trenger kanskje en funksjon for å finne dagens dato, og en funksjon for å trekke ut årstallet fra en dato.

```
SELECT FNr, YEAR(CURDATE()) - År AS AntallÅr  
FROM Film  
WHERE YEAR(CURDATE()) - År > 50;
```

Oppgavesett 2

Oppgave 1

Gjør mer på tutorialen på <https://www.w3schools.com/sql/default.asp>.
Gjør Exercises fra Create Table til SQL Dates.

Write the correct SQL statement to create a new table called Persons.

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255)
);
```

Write the correct SQL statement to delete a table called Persons.

```
DROP TABLE Persons;
```

Use the TRUNCATE statement to delete all data inside a table.

```
TRUNCATE TABLE Persons;
```

Add a column of type DATE called Birthday.

```
ALTER TABLE Persons
ADD Birthday DATE;
```

Delete the column Birthday from the Persons table.

```
ALTER TABLE Persons
DROP COLUMN Birthday;
```

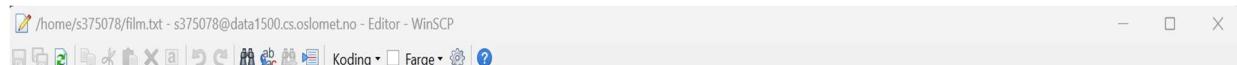
Oppgave 2

```
MariaDB [s375078]> show tables;
+-----+
| Tables_in_s375078 |
+-----+
| DEPT
| EMP
+-----+
2 rows in set (0,000 sec)

MariaDB [s375078]> select * from EMP;
+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB   | MGR   | HIREDATE | SAL    | COMM   | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7369  | SMITH   | CLERK | 7902  | 1980-12-17 | 800    | NULL   | 20     |
| 7499  | ALLEN   | SALESMAN | 7698  | 1981-02-18 | 1600   | 300    | 30     |
| 7521  | WARD    | SALESMAN | 7698  | 1981-02-22 | 1250   | 500    | 30     |
| 7566  | JONES   | MANAGER | 7839  | 1981-04-02 | 2975   | NULL   | 20     |
| 7654  | MARTIN  | SALESMAN | 7698  | 1981-09-28 | 1250   | 1400   | 30     |
| 7698  | BLAKE   | MANAGER | 7893  | 1981-05-01 | 2850   | NULL   | 30     |
| 7782  | CLARK   | MANAGER | 7839  | 1981-06-09 | 2450   | NULL   | 10     |
| 7788  | SCOTT   | ANALYST | 7566  | 1981-11-09 | 3000   | NULL   | 20     |
| 7839  | KING    | PRESIDENT | NULL  | 1981-11-17 | 5000   | NULL   | 10     |
| 7844  | TURNER  | SALESMAN | 7698  | 1981-09-08 | 1500   | 0      | 30     |
| 7876  | ADAMS   | CLERK   | 7788  | 1981-09-23 | 1100   | NULL   | 20     |
| 7900  | JAMES   | CLERK   | 7698  | 1981-12-03 | 950    | NULL   | 30     |
| 7902  | FORD    | ANALYST | 7566  | 1981-12-03 | 3000   | NULL   | 20     |
| 7934  | MILLER  | CLERK   | 7782  | 1982-01-23 | 1300   | NULL   | 10     |
+-----+-----+-----+-----+-----+-----+-----+-----+
14 rows in set (0,000 sec)
```

Oppgave 3

Script for opprettelse av Film tabellen og Film tabellen



```
/home/s375078/film.txt - s375078@data1500.cs.oslomet.no - Editor - WinSCP
drop table if exists Film;
SET FOREIGN_KEY_CHECKS=0;
create table Film (FnR int primary key, Tittel varchar(20), År int, Land varchar(15), Sjanger varchar(10), Alder int, Tid int, Pris decimal(5,2));
insert into Film values('1','Casablanca','1942','USA','Drama','15','102','149.00'),
('2','Fort Apache','1942','USA','Western','15','127',NULL),
('3','Apocalypse Now','1979','USA','Action','18','155','123.00'),
('4','Streets of fire','1984','USA','Western','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','Tegnfilm','7','78','149.00'),
('8','Veiviseren','1987','Norge','Action','15','96','87.00'),
('9','Salmen fra kjækknenet','2002','Norge','Komedie','7','80','149.00'),
('10','Anastasia','1997','USA','Tegnfilm','7','94','123.00'),
('11','La Grande bouffe','1973','Frankrike','Drama','15','129','87.00'),
('12','The Blues Brothers','1980','USA','Komedie','11','133','135.00'),
('13','Beatles: Help','1965','Storbritannia','Musikk','11','144',NULL);
```

```
MariaDB [s375078]> SELECT * FROM Film;
+---+-----+-----+-----+-----+-----+-----+-----+
| FnR | Tittel | År | Land | Sjanger | Alder | Tid | Pris |
+---+-----+-----+-----+-----+-----+-----+-----+
| 1 | Casablanca | 1942 | USA | Drama | 15 | 102 | 149.00 |
| 2 | Fort Apache | 1942 | USA | Western | 15 | 127 | NULL |
| 3 | Apocalypse Now | 1979 | USA | Action | 18 | 155 | 123.00 |
| 4 | Streets of fire | 1984 | USA | Western | 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 | Tegnfilm | 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 | 133 | 135.00 |
| 13 | Beatles: Help | 1965 | Storbritania | Musikk | 11 | 144 | NULL |
+---+-----+-----+-----+-----+-----+-----+-----+
13 rows in set (0,000 sec)
```

Oppgave 4

a)

```
MariaDB [s375078]> SELECT * FROM Film
    -> WHERE År = 1988;
+---+-----+-----+-----+-----+-----+-----+
| FnR | Tittel | År | Land | Sjanger | Alder | Tid | Pris |
+---+-----+-----+-----+-----+-----+-----+-----+
| 6 | Cinema Paradiso | 1988 | Italia | Komedie | 11 | 123 | NULL |
| 7 | Asterix hos britene | 1988 | Frankrike | Tegnfilm | 7 | 78 | 149.00 |
+---+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0,001 sec)
```

b)

```
MariaDB [s375078]> SELECT FnR, Tittel
    -> FROM Film
    -> WHERE Land = 'USA' AND År BETWEEN 1980 AND 1989;
+---+-----+
| FnR | Tittel |
+---+-----+
| 4 | Streets of fire |
| 12 | The Blues Brothers |
+---+-----+
2 rows in set (0,000 sec)
```

c)

```
MariaDB [s375078]> SELECT * FROM Film
    -> WHERE Sjanger = 'Komedie' AND Alder < 10 AND Tid < 130;
+---+-----+-----+-----+-----+-----+-----+
| FnR | Tittel | År | Land | Sjanger | Alder | Tid | Pris |
+---+-----+-----+-----+-----+-----+-----+-----+
| 9 | Salmer fra kjøkkenet | 2002 | Norge | Komedie | 7 | 80 | 149.00 |
+---+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,000 sec)
```

d)

```
MariaDB [s375078]> SELECT * FROM Film
-> WHERE Sjanger = 'Komedie' AND Alder < 10 AND Tid < 130;
+-----+-----+-----+-----+-----+-----+
| FnR | Tittel           | År    | Land   | Sjanger | Alder | Tid    | Pris   |
+-----+-----+-----+-----+-----+-----+
|   9 | Salmer fra kjøkkenet | 2002 | Norge  | Komedie  |     7 | 80    | 149.00 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0,000 sec)
```

e)

```
MariaDB [s375078]> SELECT DISTINCT Land
-> FROM Film
-> ORDER BY Land;
+-----+
| Land      |
+-----+
| Frankrike |
| Italia    |
| Norge     |
| Storbritania |
| USA       |
+-----+
5 rows in set (0,001 sec)
```

f)

```
MariaDB [s375078]> SELECT Sjanger,MIN(Tid) AS Korteste,MAX(Tid) AS Lengste
-> FROM Film
-> ORDER BY Sjanger;
+-----+-----+-----+
| Sjanger | Korteste | Lengste |
+-----+-----+-----+
| Drama   |      78 |     155 |
+-----+-----+-----+
1 row in set (0,000 sec)
```

g)

```
MariaDB [s375078]> SELECT COUNT(*) AS IkkeTilSalgs
-> FROM Film
-> WHERE Pris IS NULL
-> ;
+-----+
| IkkeTilSalgs |
+-----+
|          4   |
+-----+
1 row in set (0,000 sec)
```

h)

```
MariaDB [s375078]> SELECT COUNT(Pris) AS FilmerUnder100Kr
  -> FROM Film
  -> WHERE Pris<100;
+-----+
| FilmerUnder100Kr |
+-----+
|          2 |
+-----+
1 row in set (0,000 sec)
```

i)

```
MariaDB [s375078]> SELECT * FROM Film
  -> WHERE Tittel LIKE '%now';
+-----+-----+-----+-----+-----+-----+-----+
| FnR | Tittel      | År    | Land | Sjanger | Alder | Tid   | Pris  |
+-----+-----+-----+-----+-----+-----+-----+
|  3  | Apocalypse Now | 1979 | USA   | Action   |     18 | 155  | 123.00 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,000 sec)
```

j)

```
MariaDB [s375078]> SELECT Sjanger,AVG(Pris) AS GjennomsnittsPris
  -> FROM Film
  -> GROUP BY Sjanger
  -> HAVING COUNT(*)>2;
+-----+-----+
| Sjanger | GjennomsnittsPris |
+-----+-----+
| Komedie  |      142.000000 |
| Western  |      123.000000 |
+-----+-----+
2 rows in set (0,001 sec)
```

k)

```
MariaDB [s375078]> SELECT Sjanger,MAX(Pris)-MIN(Pris) AS Differansen
  -> FROM Film
  -> GROUP BY Sjanger
  -> ;
+-----+-----+
| Sjanger | Differansen |
+-----+-----+
| Action   |      36.00 |
| Drama    |      62.00 |
| Komedie  |      14.00 |
| Musikk   |      NULL  |
| Tegnefilm|      0.00  |
| Tegnfilm |      0.00  |
| Western  |      0.00  |
+-----+-----+
7 rows in set (0,000 sec)
```

l)

```
MariaDB [s375078]> SELECT Land,COUNT(*) AS TotaltAntall,COUNT(Pris) AS AntallTilSalgs
-> FROM Film
-> GROUP BY Land;
+-----+-----+-----+
| Land | TotaltAntall | AntallTilSalgs |
+-----+-----+-----+
| Frankrike | 2 | 2 |
| Italia | 1 | 0 |
| Norge | 2 | 2 |
| Storbritania | 1 | 0 |
| USA | 7 | 5 |
+-----+-----+-----+
5 rows in set (0,000 sec)
```

m)

```
MariaDB [s375078]> SELECT FNr,YEAR(CURDATE())-År AS AntallÅr
-> FROM Film
-> WHERE YEAR(CURDATE())-År>50;
+-----+
| FNr | AntallÅr |
+-----+
| 1 | 81 |
| 2 | 81 |
| 5 | 71 |
| 13 | 58 |
+-----+
4 rows in set (0,000 sec)
```

Oppgavesett 3

1. List opp etternavn, avdeling og lønn til alle ansatte i EMP tabellen med lønn mellom 1000 og 2000.

```
MariaDB [s375078]> SELECT * FROM EMP
-> WHERE SAL BETWEEN 1000 AND 2000;
+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME | JOB | MGR | HIREDATE | SAL | COMM | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7499 | ALLEN | SALESMAN | 7698 | 1981-02-18 | 1600 | 300 | 30 |
| 7521 | WARD | SALESMAN | 7698 | 1981-02-22 | 1250 | 500 | 30 |
| 7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250 | 1400 | 30 |
| 7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500 | 0 | 30 |
| 7876 | ADAMS | CLERK | 7788 | 1981-09-23 | 1100 | NULL | 20 |
| 7934 | MILLER | CLERK | 7782 | 1982-01-23 | 1300 | NULL | 10 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0,000 sec)
```

2. List de ulike jobbtypene som finnes.

```
MariaDB [s375078]> SELECT JOB
    -> FROM EMP
    -> GROUP BY JOB;
+-----+
| JOB      |
+-----+
| ANALYST  |
| CLERK    |
| MANAGER   |
| PRESIDENT |
| SALESMAN  |
+-----+
5 rows in set (0,000 sec)
```

3. List ansattnr, navn, jobb, lønn og avdelingsnr for ansatte i avdeling 10 og 30.

```
MariaDB [s375078]> SELECT EMPNO,ENAME,JOB,SAL,DEPTNO
    -> FROM EMP
    -> WHERE DEPTNO=10 OR DEPTNO=30;
+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB      | SAL     | DEPTNO |
+-----+-----+-----+-----+-----+
| 7499  | ALLEN  | SALESMAN | 1600   | 30     |
| 7521  | WARD   | SALESMAN | 1250   | 30     |
| 7654  | MARTIN | SALESMAN | 1250   | 30     |
| 7698  | BLAKE  | MANAGER   | 2850   | 30     |
| 7782  | CLARK  | MANAGER   | 2450   | 10     |
| 7839  | KING   | PRESIDENT | 5000   | 10     |
| 7844  | TURNER | SALESMAN | 1500   | 30     |
| 7900  | JAMES  | CLERK    | 950    | 30     |
| 7934  | MILLER | CLERK    | 1300   | 10     |
+-----+-----+-----+-----+-----+
9 rows in set (0,000 sec)
```

4. Vis ansatte som ble rekruttert i 1982.

```
SELECT * FROM EMP
```

```
WHERE YEAR(HIREDATE)=1982;
```

```
MariaDB [s375078]> SELECT * FROM EMP
    -> WHERE YEAR(HIREDATE)=1982;
+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB      | MGR    | HIREDATE  | SAL     | COMM   | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7934  | MILLER | CLERK    | 7782  | 1982-01-23 | 1300   | NULL   | 10     |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0,000 sec)
```

5. List ansatte som med navn som inneholder TH eller AR.

```
MariaDB [s375078]> SELECT * FROM EMP
-> WHERE ENAME LIKE '%TH%' OR ENAME LIKE '%AR%';
+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME   | JOB      | MGR    | HIREDATE | SAL     | COMM    | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+
| 7369  | SMITH    | CLERK    | 7902   | 1980-12-17 | 800    | NULL    | 20    |
| 7521  | WARD     | SALESMAN | 7698   | 1981-02-22 | 1250   | 500    | 30    |
| 7654  | MARTIN   | SALESMAN | 7698   | 1981-09-28 | 1250   | 1400   | 30    |
| 7782  | CLARK    | MANAGER  | 7839   | 1981-06-09 | 2450   | NULL    | 10    |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0,000 sec)
```

6. List ansattnr og navn sortert på navn.

```
MariaDB [s375078]> SELECT EMPNO,ENAME
-> FROM EMP
-> GROUP BY ENAME;
+-----+-----+
| EMPNO | ENAME   |
+-----+-----+
| 7876  | ADAMS   |
| 7499  | ALLEN   |
| 7698  | BLAKE   |
| 7782  | CLARK   |
| 7902  | FORD    |
| 7900  | JAMES   |
| 7566  | JONES   |
| 7839  | KING    |
| 7654  | MARTIN  |
| 7934  | MILLER  |
| 7788  | SCOTT   |
| 7369  | SMITH   |
| 7844  | TURNER  |
| 7521  | WARD    |
+-----+-----+
14 rows in set (0,000 sec)
```

7. Finn navn, jobb, lønn og kommisjon til alle ansatte som ikke har noen sjef.

```
MariaDB [s375078]> SELECT ENAME,JOB,SAL,COMM
-> FROM EMP
-> WHERE MGR IS NULL;
+-----+-----+-----+-----+
| ENAME | JOB      | SAL     | COMM    |
+-----+-----+-----+-----+
| KING  | PRESIDENT | 5000   | NULL    |
+-----+-----+-----+-----+
1 row in set (0,000 sec)
```

8. List alle selgere i synkende rekkefølge på kommisjon delt på lønn.

```
MariaDB [s375078]> SELECT * FROM EMP
-> WHERE JOB='SALESMAN'
-> ORDER BY COMM/SAL DESC;
+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB    | MGR   | HIREDATE | SAL    | COMM   | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+
| 7654  | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250  | 1400  | 30    |
| 7521  | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250  | 500   | 30    |
| 7499  | ALLEN  | SALESMAN | 7698 | 1981-02-18 | 1600  | 300   | 30    |
| 7844  | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500  | 0     | 30    |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0,000 sec)
```

9. Finn årlig kompensasjon til selgerne basert på månedlig lønn og månedlig kommisjon.

```
MariaDB [s375078]> SELECT EMPNO,ENAME,JOB, (SAL+COMM)*12 AS ÅrligSalg
-> FROM EMP
-> WHERE JOB='SALESMAN';
+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB      | ÅrligSalg |
+-----+-----+-----+-----+
| 7499  | ALLEN  | SALESMAN | 22800   |
| 7521  | WARD   | SALESMAN | 21000   |
| 7654  | MARTIN | SALESMAN | 31800   |
| 7844  | TURNER | SALESMAN | 18000   |
+-----+-----+-----+-----+
4 rows in set (0,000 sec)
```

10. Finn alle selgere i avdeling 30 med lønn større eller lik £1500.

```
MariaDB [s375078]> SELECT * FROM EMP
-> WHERE JOB='SALESMAN' AND SAL>=1500 AND DEPTNO=30;
+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME  | JOB      | MGR   | HIREDATE | SAL    | COMM   | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+
| 7499  | ALLEN  | SALESMAN | 7698 | 1981-02-18 | 1600  | 300   | 30    |
| 7844  | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500  | 0     | 30    |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0,000 sec)
```

11. Finn antall MANAGER i EMP tabellen.

```
MariaDB [s375078]> SELECT COUNT(*)
-> FROM EMP
-> WHERE JOB='MANAGER';
+-----+
| COUNT(*) |
+-----+
|      3   |
+-----+
1 row in set (0,000 sec)
```

12. Finn gjennomsnittlig årlig lønn + kommisjon for selgerne.

```
MariaDB [s375078]> SELECT AVG(SAL)*12+AVG(COMM)*12
-> FROM EMP
-> WHERE JOB='SALESMAN';
+-----+
| AVG(SAL)*12+AVG(COMM)*12 |
+-----+
|      23400.0000   |
+-----+
1 row in set (0,000 sec)
```

13. Finn differansen mellom høyeste og laveste lønn.

```
MariaDB [s375078]> SELECT MAX(SAL)-MIN(SAL) AS SalgDifferanse
-> FROM EMP;
+-----+
| SalgDifferanse |
+-----+
|      4200   |
+-----+
1 row in set (0,000 sec)
```

14. Finn det lengste avdelingsnavnet (funksjonen length()).

```
MariaDB [s375078]> SELECT DNAME, LENGTH(DNAME) FROM DEPT WHERE DEPTNO=
-> SELECT DEPTNO FROM DEPT ORDER BY LENGTH(DNAME) DESC LIMIT 1;
+-----+-----+
| DNAME      | LENGTH(DNAME)  |
+-----+-----+
| ACCOUNTING |          10  |
+-----+-----+
1 row in set (0,001 sec)
```

15. Finn antall ansatte i avdeling 30 som har fått kommisjon.

```
MariaDB [s375078]> SELECT COUNT(*)
-> FROM EMP
-> WHERE DEPTNO=30 AND COMM>0;
+-----+
| COUNT(*) |
+-----+
|      3   |
+-----+
1 row in set (0,000 sec)
```

16. Finn navn og lønn til alle ansatte i Chicago.

```
MariaDB [s375078]> SELECT ENAME, SAL, LOC
-> FROM EMP,DEPT
-> WHERE EMP.DEPTNO=DEPT.DEPTNO
-> AND LOC='CHICAGO';
+-----+-----+-----+
| ENAME | SAL  | LOC    |
+-----+-----+-----+
| ALLEN | 1600 | CHICAGO |
| WARD  | 1250 | CHICAGO |
| MARTIN | 1250 | CHICAGO |
| BLAKE  | 2850 | CHICAGO |
| TURNER | 1500 | CHICAGO |
| JAMES  | 950  | CHICAGO |
+-----+-----+-----+
6 rows in set (0,000 sec)
```

**17. List avdelingsnr, avdelingsnavn, jobb og etternavn med avdelingsnr i
stigende rekkefølge.**

```
MariaDB [s375078]> SELECT EMP.DEPTNO, DNAME, JOB, ENAME
-> FROM EMP,DEPT
-> WHERE EMP.DEPTNO = DEPT.DEPTNO
-> ORDER BY DEPTNO
-> ;
+-----+-----+-----+-----+
| DEPTNO | DNAME     | JOB      | ENAME   |
+-----+-----+-----+-----+
|    10  | ACCOUNTING | MANAGER  | CLARK   |
|    10  | ACCOUNTING | PRESIDENT | KING    |
|    10  | ACCOUNTING | CLERK    | MILLER  |
|    20  | RESEARCH   | CLERK    | SMITH   |
|    20  | RESEARCH   | MANAGER  | JONES   |
|    20  | RESEARCH   | ANALYST  | SCOTT   |
|    20  | RESEARCH   | CLERK    | ADAMS   |
|    20  | RESEARCH   | ANALYST  | FORD    |
|    30  | SALES     | SALESMAN | ALLEN   |
|    30  | SALES     | SALESMAN | WARD    |
|    30  | SALES     | SALESMAN | MARTIN  |
|    30  | SALES     | MANAGER  | BLAKE   |
|    30  | SALES     | SALESMAN | TURNER  |
|    30  | SALES     | CLERK    | JAMES   |
+-----+-----+-----+-----+
14 rows in set (0,000 sec)
```

18. List alle avdelinger som ikke har noen ansatte.

```
MariaDB [s375078]> SELECT DNAME
    -> FROM DEPT
    -> LEFT JOIN EMP ON DEPT.DEPTNO=EMP.DEPTNO WHERE JOB IS NULL;
+-----+
| DNAME      |
+-----+
| OPERATIONS |
+-----+
1 row in set (0,000 sec)
```

19. List alle avdelinger som har ansatte.

```
MariaDB [s375078]> SELECT DISTINCT DNAME
    -> FROM DEPT
    -> LEFT JOIN EMP ON DEPT.DEPTNO=EMP.DEPTNO
    -> WHERE JOB IS NOT NULL;
+-----+
| DNAME      |
+-----+
| ACCOUNTING |
| RESEARCH   |
| SALES      |
+-----+
3 rows in set (0,000 sec)
```

20. Finn alle ansatte som tjener mer enn JONES.

```
MariaDB [s375078]> SELECT * FROM EMP
    -> WHERE SAL>(SELECT SAL FROM EMP WHERE ENAME='JONES');
+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPNO | ENAME | JOB      | MGR    | HIREDATE | SAL    | COMM   | DEPTNO |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 7788  | SCOTT | ANALYST | 7566  | 1981-11-09 | 3000  | NULL   | 20     |
| 7839  | KING   | PRESIDENT | NULL   | 1981-11-17 | 5000  | NULL   | 10     |
| 7902  | FORD   | ANALYST | 7566  | 1981-12-03 | 3000  | NULL   | 20     |
+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0,000 sec)
```

21. List opp ansatte som tjener mer enn sjefen sin.

```
MariaDB [s375078]> SELECT E1.ENAME,E1.SAL,E2.ENAME,E2.SAL
    -> FROM EMP AS E1,EMP AS E2
    -> WHERE E1.MGR=E2.EMPNO AND E1.SAL>E2.SAL;
+-----+-----+-----+-----+
| ENAME | SAL  | ENAME | SAL  |
+-----+-----+-----+-----+
| SCOTT | 3000 | JONES | 2975 |
| FORD  | 3000 | JONES | 2975 |
+-----+-----+-----+-----+
2 rows in set (0,000 sec)
```

22. List opp navn og jobb for ansatte som har samme jobb som JONES.

```
MariaDB [s375078]> SELECT E1.ENAME,E1.JOB,E2.ENAME,E2.JOB
-> FROM EMP AS E1,EMP AS E2
-> WHERE E2.ENAME='JONES' AND E1.JOB=E2.JOB;
+-----+-----+-----+
| ENAME | JOB      | ENAME | JOB      |
+-----+-----+-----+
| JONES | MANAGER  | JONES | MANAGER  |
| BLAKE | MANAGER  | JONES | MANAGER  |
| CLARK | MANAGER  | JONES | MANAGER  |
+-----+-----+-----+
3 rows in set (0,000 sec)
```

23. Finn alle ansatt i avdeling 10 som har samme jobb som noen i avdeling 30.

```
MariaDB [s375078]> SELECT E1.ENAME,E1.JOB,E1.DEPTNO,E2.ENAME,E2.DEPTNO
-> FROM EMP AS E1,EMP AS E2
-> WHERE E1.DEPTNO=10 AND E2.DEPTNO=30 AND E1.JOB=E2.JOB;
+-----+-----+-----+-----+-----+
| ENAME | JOB      | DEPTNO | ENAME | DEPTNO |
+-----+-----+-----+-----+
| CLARK | MANAGER  |      10 | BLAKE |      30 |
| MILLER | CLERK    |      10 | JAMES  |      30 |
+-----+-----+-----+-----+
2 rows in set (0,000 sec)
```

24. List navn og jobb til ansatte som har samme jobb og lik lønn som FORD.

```
MariaDB [s375078]> SELECT E1.ENAME,E1.JOB
-> FROM EMP AS E1,EMP AS E2
-> WHERE E2.ENAME='FORD'
-> AND E1.JOB=E2.JOB
-> AND E1.SAL=E2.SAL;
+-----+
| ENAME | JOB      |
+-----+
| SCOTT | ANALYST |
| FORD  | ANALYST |
+-----+
2 rows in set (0,000 sec)
```

25. List navn, jobb, avdeling og lønn for ansatte som har samme jobb som JONES og større eller lik lønn som FORD.

```
MariaDB [s375078]> SELECT DISTINCT E3.ENAME,E3.SAL,E3.JOB
-> FROM EMP AS E1,EMP AS E2,EMP AS E3
-> WHERE E1.ENAME='JONES'
-> AND E3.JOB=E1.JOB
-> OR E2.ENAME='FORD'
-> AND E3.SAL>E2.SAL;
+-----+-----+-----+
| ENAME | SAL   | JOB      |
+-----+-----+-----+
| JONES | 2975 | MANAGER |
| BLAKE | 2850 | MANAGER |
| CLARK | 2450 | MANAGER |
| KING  | 5000 | PRESIDENT |
+-----+-----+-----+
4 rows in set (0,001 sec)
```

26. Finn alle ansatte i avdeling 10 som har samme jobb som noen i SALES avdelingen.

```
MariaDB [s375078]> SELECT EMP1.DEPTNO,EMP1.ENAME
-> FROM DEPT AS DEPT2,EMP AS EMP1,EMP AS EMP2
-> WHERE DEPT2.DNAME='SALES' AND EMP2.DEPTNO=DEPT2.DEPTNO
-> AND EMP1.DEPTNO=10
-> AND EMP1.JOB=EMP2.JOB;
+-----+-----+
| DEPTNO | ENAME  |
+-----+-----+
|     10 | CLARK  |
|     10 | MILLER |
+-----+-----+
2 rows in set (0,000 sec)
```

27. Finn ansatte i Chicago som har samme jobb som ALLEN, sorter navnene i stigende rekkefølge.

```
MariaDB [s375078]> SELECT E1.EMPNO,E1.ENAME,E1.DEPTNO
-> FROM EMP AS E1,EMP AS E2,DEPT AS D1
-> WHERE E2.ENAME='ALLEN'
-> AND D1.LOC='CHICAGO'
-> AND E2.DEPTNO=D1.DEPTNO
-> AND E1.JOB=E2.JOB
-> GROUP BY E1.ENAME ASC;
+-----+-----+-----+
| EMPNO | ENAME  | DEPTNO |
+-----+-----+-----+
|  7499 | ALLEN  |    30  |
|  7654 | MARTIN |    30  |
|  7844 | TURNER |    30  |
|  7521 | WARD   |    30  |
+-----+-----+-----+
4 rows in set (0,000 sec)
```

28. Finn alle ansatte som tjener mer enn gjennomsnittet for ansatte i sin avdeling.

```
MariaDB [s375078]> SELECT E1.ENAME,E1.JOB,E1.SAL
-> FROM EMP AS E1
-> JOIN EMP AS E2
-> ON E1.SAL = E2.SAL
-> WHERE E1.SAL > (SELECT AVG(SAL)
-> FROM EMP
-> ORDER BY DEPTNO);
+-----+-----+-----+
| ENAME | JOB      | SAL    |
+-----+-----+-----+
| JONES | MANAGER   | 2975   |
| BLAKE | MANAGER   | 2850   |
| CLARK | MANAGER   | 2450   |
| SCOTT | ANALYST   | 3000   |
| FORD  | ANALYST   | 3000   |
| KING  | PRESIDENT | 5000   |
| SCOTT | ANALYST   | 3000   |
| FORD  | ANALYST   | 3000   |
+-----+-----+-----+
8 rows in set (0,000 sec)
```

Oppgave 2

1. Ta utgangspunkt i eksemplardatabasen til Hobbyhuste, se vedlegg B.

a) Anta vi har 161 varer plassert i 21 kategorier. Hvor mange rader gir spøringer `SELECT * FROM Vare, Kategori?`

Vi får en krysskoblet tabell, med alle rader fra Vare og alle rader fra Kategori. Det er 21 kategorier og 161 varer som multiplisert sammen blir 3381 rader.

b) Hvor mange rader vil en likekobling av tabellene Vare og Kategori med hensyn på KatNr inneholde? Skriv SQL-Koden. Hva skjer hvis noen av varene ikke er plassert i en kategori?

Med hensyn til KatNr knytter spørringen hver vare til sin kategori. Da blir det 161 rader.

```
MariaDB [s375078]> SELECT *
-> FROM Vare INNER JOIN Kategori ON Vare.KatNr = Kategori.katNr;
```

Hvis en vare ikke er plassert i en kategori vil man få NULL i kolonnen KatNr.

c) Vis alle ordrelinjer, men ta dessuten med varenavn (Betegnelse) og ordredato i utskriften.

```
MariaDB [s375078]> SELECT Ordrelinje.*,Vare.Betegnelse,Ordre.Ordredato
-> FROM Ordrelinje,Ordre,Vare
-> WHERE Ordrelinje.OrdreNr=Ordre.OrdreNr
-> AND Ordrelinje.VNr=Vare.VNr;
```

d) Utvid SQL-koden for oppgave 1c med en ny kolonne som viser totalbeløp for hver ordrelinje.

```
MariaDB [s375078]> SELECT OL.*,V.Betegnelse,O.Ordredato,OL.Antall*OL.PrisPrEnhet AS Beløp  
-> FROM Ordrelinje AS OL,Ordre AS O,Vare AS V  
-> WHERE OL.OrdreNr=O.OrdreNr  
-> AND OL.VNR = V.VNr;
```

e) Vis samlet beløp hver kunde har handlet for.

```
MariaDB [s375078]> SELECT K.KNr,K.Fornavn,K.Etternavn,  
-> SUM(OL.Antall*OL.PrisPrEnhets) AS Beløp  
-> FROM Ordrelinje AS OL,Ordre AS O,Kunde AS K  
-> WHERE OL.OrdreNr=O.OrdreNr  
-> AND K.KNr=O.KNr  
-> GROUP BY K.KNr,K.Fornavn,K.Etternavn;  
+-----+-----+-----+-----+  
| KNr | Fornavn | Etternavn | Beløp |  
+-----+-----+-----+-----+  
| 5002 | Paal | Aass | 7063.00 |  
| 5007 | Joakim | Laursen | 30197.50 |  
| 5009 | Laurits | Eckhoff | 8769.20 |  
| 5011 | aashild | SAEtran | 7450.00 |  
| 5022 | Torgrim | ostbo | 9166.90 |  
| 5025 | Malvin | Khan | 3534.50 |  
| 5028 | Sidsel | Gulli | 5071.50 |  
| 5039 | Katrine | Eilertsen | 16364.80 |  
| 5042 | Skjalg | Tengesdal | 24820.50 |  
| 5043 | Gunn Iren | aanestad | 5620.50 |  
| 5049 | Khalid | Rue | 130821.50 |  
| 5071 | Jann | Skjelvik | 12950.50 |  
| 5079 | Ine | Kraft | 32310.50 |  
| 5081 | Valter | Grimsmo | 25852.00 |  
| 5082 | Alexandra | Saleh | 52379.00 |  
| 5087 | Maj | Elton | 17557.50 |  
| 5091 | Agnethe | Wessel | 97061.30 |  
| 5092 | Erland | Troan | 16356.50 |  
| 5093 | Morten | Lindland | 18838.00 |  
| 5102 | Kaja | Hovik | 6605.50 |  
| 5119 | Thale | Evenrud | 15200.50 |
```

f) Prøv å utvide SQL-koden for oppgave 1e med en ny kolonne som viser antall ordrer for hver kunde. Hva er problemet?

```
MariaDB [s375078]> SELECT KNr,COUNT(*) AS AntallOrdre  
-> FROM Ordre  
-> GROUP BY KNr;  
+-----+  
| KNr | AntallOrdre |  
+-----+  
| 5002 | 5 |  
| 5007 | 9 |  
| 5009 | 6 |  
| 5011 | 5 |  
| 5022 | 6 |  
| 5025 | 5 |  
| 5028 | 5 |  
| 5039 | 6 |  
| 5042 | 6 |  
| 5043 | 5 |  
| 5049 | 5 |
```

g) Vis samlet beløp pr. ordre.

```
MariaDB [s375078]> SELECT OrdreNr,SUM(Antall*PrisPrEnhet) AS SumPrOrdre  
-> FROM Ordrelinje  
-> GROUP BY OrdreNr;□
```

h) Lag en vareliste som for hver vare vise antall enheter på lager og samlet lagerverdi for denne varen.

```
MariaDB [s375078]> SELECT VNr,Antall,Antall*Pris AS LagerVerdi  
-> FROM Vare;
```

i) Finn samlet verdi av varelageret.

```
MariaDB [s375078]> SELECT SUM(Antall*Pris) AS SumVareLager  
-> FROM Vare;  
+-----+  
| SumVareLager |  
+-----+  
| 2856730.00 |  
+-----+  
1 row in set (0,000 sec)
```

j) Finn ut hvor mye hver varekategori har solgt for. Lag en sortert liste med bestselgerne først, og få med navn på kategori i utskriften.

```
MariaDB [s375078]> SELECT Kategori.KatNr,Kategori.Navn,SUM(Ordrelinje.Antall*  
PrisPrEnhett) AS SamletPris  
-> FROM Vare,Ordrelinje,Kategori  
-> WHERE Vare.VNr=Ordrelinje.VNr AND Vare.KatNr=Kategori.KatNr  
-> GROUP BY Kategori.KatNr,Kategori.Navn  
-> ORDER BY SUM(Ordrelinje.Antall*PrisPrEnhett) DESC;  
+-----+-----+-----+  
| KatNr | Navn | SamletPris |  
+-----+-----+-----+  
| 1 | Hageutstyr | 6125596.30 |  
| 12 | Hagemobler | 1276253.50 |  
| 3 | Keramikk | 388107.50 |  
| 14 | Busker | 349611.00 |  
| 17 | Dekorasjoner | 327931.30 |  
| 4 | Konfekt og marsipan | 244158.00 |  
| 16 | Blomsterloker | 216593.00 |  
| 11 | Leker | 139967.00 |  
| 21 | Gjodsel | 94167.50 |  
| 10 | Boker | 69549.00 |  
| 18 | Gronnsaksfro | 68167.70 |  
| 13 | Dukker og nisser | 49367.40 |  
| 7 | Fiske | 32119.50 |  
| 2 | Hobbymalning | 30304.60 |  
| 20 | Hobbypakker | 19647.00 |  
| 15 | Blomsterfro | 14539.30 |  
| 6 | Tekstil, som og strikking | 4368.00 |  
+-----+-----+-----+  
17 rows in set (0,025 sec)
```

k) Vis alle postnumre der det enten bor en ansatt eller en kunde. Hvordan få med steder der det bor både en ansatt og en kunde? Hva kan du gjøre for å få med navn på poststeder?

```
MariaDB [s375078]> SELECT DISTINCT Poststed.PostNr  
-> FROM Ansatt,Kunde,Poststed  
-> WHERE Ansatt.PostNr=Poststed.PostNr OR Kunde.PostNr=Poststed.PostNr;  
+-----+  
| PostNr |  
+-----+  
| 0015 |  
| 0032 |  
| 0122 |  
| 0171 |  
| 0201 |  
| 0208 |  
| 0214 |  
| 0250 |  
| 0255 |  
| 0270 |  
| 0330 |  
| 0406 |  
| 0445 |  
| 0452 |  
| 0455 |  
| 0501 |
```

Referanser

Bratbergsengen, K. (2019, 4 13). *Store Norske Leksikon*. Hentet 1 25, 2023 fra relasjonsdatabase:
<https://snl.no/relasjonsdatabase>

Bratbergsengen, K. (2019, 12 12). *Store Norske Leksikon*. Hentet fra Database:
<https://snl.no/database>

Oracle. (2023). *Oracle*. Hentet 1 2023 fra What is a database:
<https://www.oracle.com/database/what-is-database/>

Universitetsforlaget. (2020). *Universitetsforlaget*. Hentet fra Kap4losning:
https://www.dbsys.info/Databasesystemer/2_Losninger/kap4losning.html

Wikipedia. (2019, 8 20). *Wikipedia*. Hentet 1 24, 2023 fra Relasjonsdatabase:
<https://no.wikipedia.org/wiki/Relasjonsdatabase>

Wikipedia. (2022, 1 13). *Wikipedia*. Hentet 1 25, 2023 fra Database:
<https://no.wikipedia.org/wiki/Database>