

Øving 4

a. List ut all informasjon (ordrehode og ordredetalj) om ordrer for leverandør nr 44.

```
SELECT *
FROM ordrehode oh
LEFT JOIN ordredetalj od
ON oh.ordrenr = od.ordrenr
WHERE levnr = 44;
```

b. Finn navn og by (“LevBy”) for leverandører som kan levere del nummer 1.

```
SELECT li.navn, li.levby
FROM prisinfo pi
LEFT JOIN levinfo li
ON li.levnr = pi.levnr
WHERE pi.delnr = 1;
```

c. Finn nummer, navn og pris for den leverandør som kan levere del nummer 201 til billigst pris.

```
SELECT li.levnr, li.navn, pi.pris
FROM prisinfo pi
LEFT JOIN levinfo li
ON li.levnr = pi.levnr
WHERE pi.delnr = 201
ORDER BY pi.pris
ASC
LIMIT 1;
```

d. Lag fullstendig oversikt over ordre nr 16, med ordrenr, dato, delnr, beskrivelse, kvantum, (enhets-)pris og beregnet beløp (=pris*kvantum).

Her har jeg brukt NATURAL JOIN i stedet for en equijoin for kortere spørringer, men de kunne like gjerne skrives som JOIN ON a.egenskap = b.egenskap for hver av egenskapene.

```
SELECT ordrenr, dato, delnr, beskrivelse, kvantum, pris AS stykkpris, kvantum * pris AS beløp
FROM prisinfo
NATURAL JOIN ordredetalj
NATURAL JOIN ordrehode
NATURAL JOIN delinfo
WHERE ordrenr = 16;
```

e. Finn delnummer og leverandørnummer for deler som har en pris som er høyere enn prisen for del med katalognr X7770.

```
SELECT delnr, levnr
FROM prisinfo pi
WHERE pi.pris > (
    SELECT pris
```

```

FROM prisinfo
WHERE katalognr = "X7770"
);

```

f.

i) Tenk deg at tabellen levinfo skal deles i to. Sammenhengen mellom by og fylke skal tas ut av tabellen. Det er unødvendigå lagre fylkestilhørigheten for hver forekomst av by. Lag én ny tabell som inneholder byer og fylker. Fyll denne med data fra levinfo. Lag også en tabell som er lik levinfo unntatt kolonnen Fylke. (Denne splittingen av tabellen levinfo gjelder bare i denneoppgaven. I resten av oppgavesettet antar du at du har den opprinnelige levinfo-tabellen.)

Jeg lager nye tabeller og overfører all relevant data fra levinfo.

```

CREATE TABLE byer(
levby VARCHAR(20) NOT NULL,
fylke VARCHAR(20) NOT NULL,
CONSTRAINT pk_byer PRIMARY KEY(levby)
);

```

```

INSERT INTO byer (levby, fylke)
SELECT levby, fylke
FROM levinfo
GROUP BY levby;

```

```

CREATE TABLE levinfo_nofylke(
levnr INTEGER,
navn VARCHAR(20) NOT NULL,
adresse VARCHAR(20) NOT NULL,
levby VARCHAR(20) NOT NULL,
postnr INTEGER NOT NULL,
CONSTRAINT levinfo_nofylke_pk PRIMARY KEY(levnr));

```

```

INSERT INTO levinfo_nofylke (levnr,navn,adresse,levby,postnr)
SELECT levnr,navn,adresse,levby,postnr
FROM levinfo;

```

ii) Lag en virtuell tabell (view) slik at brukerne i størst mulig grad kan jobbe på samme måte mot de to nye tabellene som den gamle. Prøv ulike kommandoer mot tabellen (select, update, delete, insert). Hvilke begrensninger, hvis noen, har brukerne i forhold til tidligere?

```

CREATE VIEW levinfo_medfylke AS
SELECT li.*, b.fylke
FROM levinfo_nofylke li
LEFT JOIN byer b
ON li.levby = b.levby;

```

Select fungerer akkurat som før, ingen merkbar forskjell.

```

MariaDB [idatt2103_ov4]> SELECT * FROM levinfo_medfylke WHERE fylke = "Oslo";
+-----+-----+-----+-----+-----+-----+

```

```

| levnr | navn                | adresse      | levby | postnr | fylke |
+-----+-----+-----+-----+-----+-----+
| 6     | Kontorekspressen AS | Skolegata 3  | Oslo  | 1234   | Oslo  |
| 44    | Billig og Bra AS    | Aveny 56     | Oslo  | 1222   | Oslo  |
+-----+-----+-----+-----+-----+
2 rows in set (0.051 sec)

```

```

MariaDB [idatt2103_ov4]> SELECT * FROM levinfo_medfylke WHERE levnr > 20;
+-----+-----+-----+-----+-----+-----+
| levnr | navn                | adresse      | levby  | postnr | fylke      |
+-----+-----+-----+-----+-----+-----+
| 44    | Billig og Bra AS    | Aveny 56     | Oslo   | 1222   | Oslo       |
| 81    | Kontorbutikken AS  | Gjennomveien 78 | Ål     | 3345   | Telemark   |
| 82    | Kontordata AS      | Åsveien 178   | Trondheim | 7023   | S-Trøndelag |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

Update fungerer akkurat som før når man endrer egenskaper som ikke er fylke/by.

Om man bruker UPDATE til å endre en verdi i kolonnen “by”, vil byen endre seg, og fylket vil automatisk følge etter, og bli satt til det fylket som hører til den nye byen, gitt i byer. Dette er ønsket oppførsel, og gjør oppdateringen lettere.

```

+-----+-----+-----+-----+-----+-----+
| levnr | navn                | adresse      | levby  | postnr | fylke      |
+-----+-----+-----+-----+-----+-----+
| 6     | Kontorekspressen AS | Skolegata 3  | Oslo   | 1234   | Oslo       |
| 9     | Kontorutstyr AS    | Villa Villekulla | Ås     | 1456   | Østfold    |
| 12    | Mister Office AS   | Storgt 56     | Ås     | 1456   | Østfold    |
| 44    | Billig og Bra AS    | Pilestredet 4 | Oslo   | 1222   | Oslo       |
| 81    | Kontorbutikken AS  | Gjennomveien 78 | Ål     | 3345   | Telemark   |
| 82    | Kontordata AS      | Åsveien 178   | Trondheim | 7023   | S-Trøndelag |
+-----+-----+-----+-----+-----+

```

```

MariaDB [idatt2103_ov4]> UPDATE levinfo_medfylke SET levby = "Oslo" WHERE levnr = 82;
Query OK, 1 row affected (0.011 sec)
Rows matched: 1  Changed: 1  Warnings: 0

```

```

+-----+-----+-----+-----+-----+-----+
| levnr | navn                | adresse      | levby  | postnr | fylke      |
+-----+-----+-----+-----+-----+-----+
| 6     | Kontorekspressen AS | Skolegata 3  | Oslo   | 1234   | Oslo       |
| 9     | Kontorutstyr AS    | Villa Villekulla | Ås     | 1456   | Østfold    |
| 12    | Mister Office AS   | Storgt 56     | Ås     | 1456   | Østfold    |
| 44    | Billig og Bra AS    | Pilestredet 4 | Oslo   | 1222   | Oslo       |
| 81    | Kontorbutikken AS  | Gjennomveien 78 | Ål     | 3345   | Telemark   |
| 82    | Kontordata AS      | Åsveien 178   | Oslo   | 7023   | Oslo       |
+-----+-----+-----+-----+-----+

```

Et problem dukker derimot opp dersom man prøver å endre “fylke” direkte. Om man endrer på “fylke” i den nye viewet/vinduet, vil vi endre på tabellen `byeri` stedet for å endre på tabellen `levinfo_nofylke`. Altså kan man ved å endre på “fylke” på levnr 9 i eksempelet over fra “Østfold” til “Vestfold”, vil det også endre på levnr 12.

levnr	navn	adresse	levby	postnr	fylke
6	Kontorekspressen AS	Skolegata 3	Oslo	1234	Oslo
9	Kontorutstyr AS	Villa Villekulla	Ås	1456	Østfold
12	Mister Office AS	Storgt 56	Ås	1456	Østfold
44	Billig og Bra AS	Pilestredet 4	Oslo	1222	Oslo
81	Kontorbutikken AS	Gjennomveien 78	Ål	3345	Østfold
82	Kontordata AS	Åsveien 178	Oslo	7023	Oslo

```
MariaDB [idatt2103_ov4]> UPDATE levinfo_medfylke SET fylke = "Vestfold" WHERE levnr = 9;
Query OK, 1 row affected (0.011 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

levnr	navn	adresse	levby	postnr	fylke
6	Kontorekspressen AS	Skolegata 3	Oslo	1234	Oslo
9	Kontorutstyr AS	Villa Villekulla	Ås	1456	Vestfold
12	Mister Office AS	Storgt 56	Ås	1456	Vestfold
44	Billig og Bra AS	Pilestredet 4	Oslo	1222	Oslo
81	Kontorbutikken AS	Gjennomveien 78	Ål	3345	Østfold
82	Kontordata AS	Åsveien 178	Oslo	7023	Oslo

De to tabellene i viewet henger ikke sammen med en triviell relasjon som en-til-en, så om en prøver å slette en tuppel får man en feilmelding.

```
MariaDB [idatt2103_ov4]> DELETE FROM levinfo_medfylke WHERE levnr = 44;
ERROR 1288 (HY000): The target table levinfo_medfylke of the DELETE is not updatable
```

Inserts fungerer heller ikke i mitt tilfelle, da tabellen bruker OUTER JOIN som ikke er tillatt i følge Dokumentasjonen.

```
MariaDB [idatt2103_ov4]> INSERT INTO levinfo_medfylke (levnr, navn, adresse, levby, postnr) VALUES (8
ERROR 1471 (HY000): The target table levinfo_medfylke of the INSERT is not insertable-into
```

Alle endringer som INSERT, UPDATE og DELETE kan fortstatt gjøres uten problem i de opprinnelige tabellene, de som vinduet baserer seg på.

g. Anta at en vurderer å slette opplysningene om de leverandørene som ikke er representert i Prisinfo-tabellen. Finn ut hvilke byer en i tilfelle ikke får leverandør i. (Du skal ikke utføre slettingen.) (Tips: Svaret skal bli kun én by, "Ål".)

```
SELECT DISTINCT levby FROM levinfo WHERE levby NOT IN (SELECT levby FROM prisinfo LEFT JOIN levinfo
... men hvorfor er SELECT DISTINCT levby FROM levinfo WHERE levnr NOT IN (SELECT levnr
FROM prisinfo); feil?
```

h. Finn leverandørnummer for den leverandør som kan levere ordre nr 18 til lavest totale beløp (vanskelig).

```
delnr, kvantum i ordren: > SELECT delnr, kvantum FROM ordredetalj WHERE ordrenr = 18;
```

Alle leverandører som kan levere delnr(minst en av) i ordre 18: > SELECT * FROM prisinfo RIGHT JOIN ordredetalj ON prisinfo.delnr = ordredetalj.delnr WHERE ordredetalj.ordrenr = 18;

Antall delnr (individuelle deltyper) hver leverandør kan levere: > SELECT levnr, COUNT(*) FROM kan_levere_deler GROUP BY levnr;

Finn alle levnr fra kan_levere_deler som kan levere like mange delnr som det er i ordre 18. > SELECT levnr FROM kan_levere_deler GROUP BY levnr HAVING (COUNT() = (SELECT COUNT() FROM ordredetalj WHERE ordrenr = 18));

Finn totalpris per del, fra leverandørene som kan levere alle delene > SELECT kld.levnr, pris*kvantum FROM kan_levere_deler kld RIGHT JOIN kan_levere_alle kla ON kld.levnr = kla.levnr;

Finn totalpris for hele ordren per leverandør: > SELECT kld.levnr, SUM(pris*kvantum) AS totalpris FROM kan_levere_deler kld RIGHT JOIN kan_levere_alle kla ON kld.levnr = kla.levnr GROUP BY kld.levnr ORDER BY totalpris;

-- levnr og prisinfo for alle deler i ordre 18 hos alle leverandører som har delen

```
CREATE VIEW kan_levere_deler AS
SELECT pi.levnr, pi.delnr, pi.pris, kvantum
FROM ordredetalj od LEFT JOIN prisinfo pi ON od.delnr = pi.delnr
WHERE od.ordrenr = 18;
```

-- levnr og totalsum for alle leverandører som har alle delene

```
CREATE VIEW kan_levere_alle AS
SELECT levnr, SUM(pris*kvantum) AS totalpris
FROM kan_levere_deler
GROUP BY levnr HAVING (
COUNT(*) = (SELECT COUNT(*) FROM ordredetalj WHERE ordrenr = 18)
);
```

-- Hent ut kun leverandøren med den laveste totalprisen

```
SELECT * FROM kan_levere_alle ORDER BY totalpris ASC LIMIT 1;
```

Oppgave 2: SQL med NULL-verdier

a. Sett opp en SELECT-setning som er UNION mellom alle forlag med Oslo-nummer (telefonnummer begynner med 2) og alle som ikke er Oslo-nummer. Får du med forlaget med NULL-verdi på telefonnummer? Hvis ikke, utvid unionen med en mengde til.

```
SELECT *
FROM forlag
WHERE telefon LIKE '2%'
UNION
SELECT *
FROM forlag
WHERE telefon NOT LIKE '2%'
UNION
SELECT *
FROM forlag
WHERE telefon IS NULL;
```

b. Sett opp SQL-setninger som finner gjennomsnittlig alder på forfattere der fødselsåret er oppgitt. For forfattere der dødsåret ikke er oppgitt, skal du kun ta med de som er født etter 1900.

Kan løses med view, enkelt å lese:

```
CREATE VIEW forfatter_fra_til AS
SELECT forfatter_id, fornavn, etternavn, fode_aar AS fra, dod_aar AS til
  FROM forfatter
  WHERE fode_aar IS NOT NULL AND dod_aar IS NOT NULL

UNION

SELECT forfatter_id, fornavn, etternavn, fode_aar AS fra, YEAR(current_date()) as til
  FROM forfatter
  WHERE fode_aar IS NOT NULL AND dod_aar IS NULL AND fode_aar > 1900;

-- eksempel: se alle aldre:
SELECT forfatter_id, fornavn, etternavn, til-fra AS alder FROM forfatter_fra_til;

-- finn gjennomsnittsalderen:
SELECT SUM(til-fra)/COUNT(*) FROM forfatter_fra_til;
```

Kan også løses uten view, med en lang spørring.

```
SELECT SUM(alder) / COUNT(*) FROM (
  SELECT dod_aar - fode_aar AS alder
  FROM forfatter
  WHERE fode_aar IS NOT NULL AND dod_aar IS NOT NULL

  UNION

  SELECT YEAR(current_date()) - fode_aar AS alder
  FROM forfatter
  WHERE fode_aar IS NOT NULL AND dod_aar IS NULL AND fode_aar > 1900
) AS forfatter_alder;
```

Uttrykkene summerer de fem forfatterne som stemmer med kravene, og gir resultatet 68.0 år.

c. Sett opp SQL-setninger som finner hvor stor andel av forfatterne som ble med i beregningene under b).

```
SELECT COUNT(forfatter_alder.forfatter_id), COUNT(forfatter.forfatter_id) FROM
(
  -- forfattere som blir telt med i gjennomsnittet
  SELECT * FROM forfatter
  WHERE fode_aar IS NOT NULL
  AND ( dod_aar IS NOT NULL OR fode_aar > 1900 )
) AS forfatter_alder
RIGHT JOIN forfatter
ON forfatter.forfatter_id = forfatter_alder.forfatter_id;
```

Gir resultatet

```
+-----+-----+
```

COUNT(forfatter_alder.forfatter_id)	COUNT(forfatter.forfatter_id)
5	12

Alternativ løsning som bruker vinduet opprettet tidligere, viser andel som prosent:

```
SELECT CONCAT(100 * COUNT(fft.forfatter_id) / COUNT(f.forfatter_id), '%') AS prosent_aldersberegning
FROM forfatter_fra_til fft
RIGHT JOIN forfatter f
ON fft.forfatter_id = f.forfatter_id;
```

prosent_aldersberegning
41.6667%