Øving 4: Normalisering og SQL, del 2

# Oppgave 1 Normalisering

## a)

SELECT \* FROM ordredetalj JOIN ordrehode ON ordredetalj.ordrenr = ordrehode.ordrenr AND ordredetalj.levnr = 44

## b)

SELECT levinfo.navn, levinfo.levby FROM prisinfo JOIN levinfo ON prisinfo.levnr = levinfo.levnr AND prisinfo.delnr = 1

## c)

SELECT levinfo.levnr, levinfo.navn, levinfo.adresse, prisinfo.pris FROM prisinfo JOIN levinfo ON prisinfo.levnr = levinfo.levnr AND prisinfo.delnr = 201 ORDER BY prisinfo.pris LIMIT 1

## d)

SELECT OH.ordrenr, OH.dato, DI.delnr, DI.beskrivelse, OD.kvantum, PI.pris, OD.kvantum \* PI.pris AS "Beløp" FROM ordrehode AS OH JOIN ordredetalj AS OD ON OH.ordrenr = OD.ordrenr AND OH.ordrenr = 16 JOIN delinfo AS DI ON OD.delnr = DI.delnr JOIN prisinfo AS PI ON OD.delnr = PI.delnr

## e)

SELECT delnr, levnr FROM prisinfo WHERE pris > (SELECT pris FROM prisinfo WHERE katalognr = "X7770")

## f) i)

CREATE TABLE byer(

id INTEGER PRIMARY KEY AUTO\_INCREMENT,

by\_navn VARCHAR(20) NOT NULL UNIQUE,

fylke\_navn VARCHAR(20) NOT NULL);

INSERT INTO byer (by\_navn, fylke\_navn) SELECT DISTINCT(levby), fylke FROM levinfo;

CREATE TABLE ny\_levinfo(

levnr INTEGER PRIMARY KEY,

navn VARCHAR(20) NOT NULL,

adresse VARCHAR(20) NOT NULL,

levby INTEGER NOT NULL,

postnr INTEGER NOT NULL,

CONSTRAINT ny\_levinfo\_fk FOREIGN KEY(levby) REFERENCES byer(id));

INSERT INTO ny\_levinfo SELECT levnr, navn, adresse, byer.id as levby, postnr FROM levinfo JOIN byer ON levinfo.levby = byer.by\_navn

## f) ii)

CREATE VIEW levinfo\_view AS

SELECT levnr, navn, adresse, byer.by\_navn, byer.fylke\_navn, postnr

FROM ny\_levinfo JOIN byer ON ny\_levinfo.levby = byer.id;

Kan slette rader og oppdatere rader så lenge man holder seg til en tabell. Kan ikke sette inn.

## g)

SELECT \* FROM levinfo AS L LEFT JOIN prisinfo AS P ON L.levnr = P.levnr GROUP BY L.levby HAVING P.delnr IS NULL

## h)

SELECT levnr, SUM(pris\*kvantum)as SUM FROM prisinfo

JOIN ordredetalj ON prisinfo.delnr = ordredetalj.delnr AND ordredetalj.ordrenr = 18

GROUP by levnr HAVING COUNT(levnr) = (SELECT COUNT(delnr) FROM ordredetalj WHERE ordrenr = 18)

ORDER BY SUM(pris\*kvantum) LIMIT 1;

# Oppgave 2 SQL

## a)

SELECT \* FROM forlag WHERE telefon LIKE '2%'

UNION

SELECT \* FROM forlag WHERE telefon NOT LIKE '2%'

UNION

SELECT \* FROM forlag WHERE telefon LIKE '2%'

UNION

SELECT \* FROM forlag WHERE telefon IS NULL

## b)

SELECT AVG(IFNULL(dod\_aar, YEAR(CURRENT\_DATE)) - fode\_aar) AS "Gjennomsnittlig alder" FROM `forfatter` WHERE fode\_aar IS NOT NULL AND (fode\_aar > 1900 OR dod\_aar IS NOT NULL)

## c)

SELECT COUNT(IFNULL(dod\_aar, YEAR(CURRENT\_DATE)) - fode\_aar) / (SELECT COUNT(\*) FROM forfatter) AS "Andel" FROM forfatter WHERE fode\_aar IS NOT NULL AND (fode\_aar > 1900 OR dod\_aar IS NOT NULL)