

Løsningsforslag

Laget av Else Lervik.

Øving: SQL del 2

Dette løsningsforslaget bruker ikke NATURAL JOIN.

Årsaken er at denne operasjonen krever at de to kolonnene som koples har samme navn, det er fort gjort å glemme seg her.

OPPGAVE 1A

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

```
SELECT ordrehode.*, ordredetalj.delnr, ordredetalj.kvantum
FROM ordrehode, ordredetalj
WHERE ordrehode.levnr = 44
AND ordrehode.ordrenr = ordredetalj.ordrenr;
```

ORDRENR	DATO	LEVNR	DELNR	KVANTUM
13	13-SEP-86	44 p	51173	20
14	17-DEC-86	44 p	201	100
14	17-DEC-86	44 p	202	100
14	17-DEC-86	44 p	51173	30
15	03-JAN-87	44 p	201	100
15	03-JAN-87	44 p	202	100

OPPGAVE 1B

Finn navn og by ("LevBy") for leverandører som leverer del nummer 1.

```
SELECT levinfo.navn, levinfo.levby
FROM levinfo, prisinfo
WHERE levinfo.levnr = prisinfo.levnr
AND prisinfo.delnr = 1;
```

NAVN	LEVBY
Kontorekspressen AS	Oslo
Kontorutstyr AS	Ås

OPPGAVE 1C

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

```
SELECT levinfo.levnr, levinfo.navn, prisinfo.pris
FROM levinfo, prisinfo
WHERE levinfo.levnr = prisinfo.levnr
AND prisinfo.delnr = 201
AND prisinfo.pris = (
    SELECT MIN(pris)
    FROM prisinfo
    WHERE delnr = 201);
```

LEVNR NAVN	PRIS
44 Billig og Bra AS	1,6

OPPGAVE 1D

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

```
SELECT ordrehode.ordrenr, ordrehode.dato, ordredetalj.delnr, delinfo.beskrivelse,
ordredetalj.kvantum, prisinfo.pris, prisinfo.pris*ordredetalj.kvantum Totalpris
FROM ordrehode, ordredetalj, delinfo, prisinfo
WHERE ordrehode.ordrenr = ordredetalj.ordrenr AND
ordredetalj.delnr = prisinfo.delnr AND
prisinfo.levnr = ordrehode.levnr AND
ordredetalj.delnr = delinfo.delnr AND
ordrehode.ordrenr = 16;
```

ORDRENR	DATO	DELNR	BESKRIVELSE	KVANTUM
1,9	16 30.01.1987	201	Svarte kulepenner	50
6,5	16 30.01.1987	202	Blå kulepenner	50
,85	16 30.01.1987	1909	Skriveunderlag	10
,57	16 30.01.1987	51173	Binders	20
	11,4			

OPPGAVE 1E

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

```
SELECT delnr, levnr
FROM prisinfo
WHERE pris > (
SELECT pris
FROM prisinfo
WHERE katalognr = 'X7770');
```

DELNR	LEVNR
1	6
1	9
51200	6
3	82
4	82
3	6
4	6
202	6
3	9

OPPGAVE 1F

i) Tenk deg at tabellen levinfo skal deles i to. Sammenhengen mellom by og fylke skal tas ut av tabellen.

Det er unødvendig å lagre fylketilhørigheten 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 denne oppgaven. I resten av oppgavesettet antar du at du har den opprinnelige levinfo-tabellen.)

Hvorfor kan man finne på å gjøre noe sånt som dette?

Jo, man har en gammel database der datamodelleringen ikke ble nøye nok utført. Man glemte at fylke burde vært en egen entitetstype, og dermed en egen tabell. (Relasjonen levinfo tilfredsstiller ikke 3. normalform, vi har en transitiv avhengighet der.)

(Den observante student vil jo også si at man burde tatt ut bynavnet, og bare beholdt postnummeret i levinfo, av samme grunn.)

Man forsøker så å rette på dette i ettertid, etter at man har flere applikasjoner som jobber mot de eksisterende tabellene.

```
CREATE TABLE byfylke (  
  byen VARCHAR(20),  
  fylke VARCHAR(20),  
  CONSTRAINT byfylke_pk PRIMARY KEY(byen));
```

```
INSERT INTO byfylke (byen, fylke)  
  SELECT DISTINCT levby, fylke FROM levinfo;
```

```
SELECT * FROM byfylke;
```

BYEN	FYLKE
Oslo	Oslo
Trondheim	S-Trøndelag
Ål	Telemark
Ås	Østfold

```
CREATE TABLE levinfo2(  
  levnr INTEGER,  
  navn VARCHAR(20) NOT NULL,  
  adresse VARCHAR(20),  
  levby VARCHAR(20) NOT NULL,  
  postnr INTEGER NOT NULL,  
  CONSTRAINT levinfo2_pk PRIMARY KEY (levnr),  
  CONSTRAINT levinfo2_fk FOREIGN KEY (levby) REFERENCES byfylke(byen));
```

```
INSERT INTO levinfo2 (levnr, navn, adresse, levby, postnr)  
  SELECT levnr, navn, adresse, levby, postnr FROM levinfo;
```

```
SELECT * FROM levinfo2;
```

LEVNR	NAVN	ADRESSE	LEVBY
1234	6 Kontorekspressen AS	Skolegata 3	Oslo
7023	82 Kontordata AS	Åsveien 178	Trondheim
1456	9 Kontorutstyr AS	Villa Villekulla	Ås
1222	44 Billig og Bra AS	Aveny 56	Oslo

```

1456      12 Mister Office AS      Storgt 56      Ås
3345      81 Kontorbutikken AS      Gjennomveien 78  Ål

```

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 levview
AS
SELECT l.levnr,l.navn,l.adresse,l.levby,b.fylke,l.postnr
FROM levinfo2 l, byfylke b
WHERE l.levby = b.byen;

SELECT * FROM levview;

```

LEVNR	NAVN	ADRESSE	LEVBY	FYLKE
1234	6 Kontorekspressen AS	Skolegata 3	Oslo	Oslo
Trøndelag	82 Kontordata AS	Åsveien 178	Trondheim	S-
1456	9 Kontorutstyr AS	Villa Villekulla	Ås	Østfold
1222	44 Billig og Bra AS	Aveny 56	Oslo	Oslo
1456	12 Mister Office AS	Storgt 56	Ås	Østfold
Telemark	81 Kontorbutikken AS	Gjennomveien 78	Ål	
		3345		

ER VIEWET OPPDATERBART?

Kun relevant i Oracle. (Fungerer kanskje ikke i MySQL)

Dvs. kan vi endre de fysiske tabellen levinfo2 og fylke ved å sende endrings-setninger til levview?

Vi slår opp i datakatalogen:

```

SELECT column_name, updatable, insertable, deletable FROM user_updatable_columns
WHERE table_name = 'LEVVIEW';

```

Resultatet blir:

COLUMN_NAME	UPD	INS	DEL
LEVNR	YES	YES	YES
NAVN	YES	YES	YES
ADRESSE	YES	YES	YES
LEVBY	YES	YES	YES
FYLKE	NO	NO	NO
POSTNR	YES	YES	YES

Vi kan endre den tabellen der vi i viewet kun har en rad pr primærnøkkelverdi. Her vil det si tabellen levinfo2.

Vi kan ikke endre tabellen Fylke pga at vi kan ha mer enn én rad i viewet pr rad i tabellen Fylke.

OPPGAVE 1G

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.)

```
SELECT DISTINCT levby FROM levinfo WHERE levby NOT IN (
    SELECT levby FROM prisinfo, levinfo WHERE prisinfo.levnr = levinfo.levnr);
```

LEVBY

 Ål

OPPGAVE 1H

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

Trinn 1: Lag en virtuell tabell som viser hvem som kan levere hele eller deler av ordren:

```
CREATE VIEW kan_levere_til_ordren (lev_nr,del_nr,pris,total_pris)
AS
SELECT prisinfo.levnr, ordredetalj.delnr, pris, pris*kvantum
FROM prisinfo, ordredetalj
WHERE prisinfo.delnr = ordredetalj.delnr AND ordredetalj.ordrenr = 18;
```

```
SELECT * FROM kan_levere_til_ordren;
```

LEV_NR	DEL_NR	PRIS	TOTAL_PRIS
6	3	1199	2398
9	3	1050	2100
82	3	1299	2598
6	4	550	4400
82	4	899	7192

Trinn 2: Fra denne velger du så ut de leverandørene som kan levere like mange deler til ordren som ordren krever. Det vil si, kan levere hele ordren:

```
CREATE VIEW kan_levere_hele_ordren (lev_nr,total_pris)
AS
SELECT lev_nr,SUM(total_pris)
FROM kan_levere_til_ordren
GROUP BY lev_nr
HAVING COUNT(*)= (
    SELECT COUNT(*)
    FROM ordredetalj
    WHERE ordrenr=18);
```

```
SELECT * FROM kan_levere_hele_ordren;
```

LEV_NR	TOTAL_PRIS
6	6798
82	9790

Trinn 3: Til slutt finner du ut hvem som leverer billigst.

```
SELECT lev_nr, total_pris
FROM kan_levere_hele_ordren
WHERE total_pris = (
```

```
SELECT MIN(total_pris)
FROM kan_levere_hele_ordren);
```

LEV_NR	TOTAL_PRIS
6	6798

OPPGAVE 2A

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%' OR telefon NOT LIKE '2%';
```

FORLAG_ID	FORLAG_NAVN	ADRESSE	TELEFON
1	Tapir	Trondheim	73590000
2	Gyldendal	Oslo	22220000
3	Cappelen	Oslo	22200000
4	Universitetsforlaget	Oslo	23230000
5	Achehaug	Oslo	22000000
6	Oktober	Oslo	22002200
7	Tiden	Oslo	22232223

Vi ser at vi ikke får med det amerikanske forlaget der telefon-nummeret er NULL (tomt). Vi utvider setningen slik:

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

FORLAG_ID	FORLAG_NAVN	ADRESSE	TELEFON
1	Tapir	Trondheim	73590000
2	Gyldendal	Oslo	22220000
3	Cappelen	Oslo	22200000
4	Universitetsforlaget	Oslo	23230000
5	Achehaug	Oslo	22000000
6	Oktober	Oslo	22002200
7	Tiden	Oslo	22232223
8	Harper Collins	USA	

Dette gir samme resultat som SELECT * FROM forlag;

OPPGAVE 2B

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.

View i MySQL:

```
CREATE VIEW noen_forft AS
SELECT forfatter_id, fode_aar, YEAR(CURDATE()) AS til_aar FROM forfatter
WHERE fode_aar > 1900 AND dod_aar IS NULL
UNION
SELECT forfatter_id, fode_aar, dod_aar AS til_aar FROM forfatter
WHERE fode_aar IS NOT NULL AND dod_aar IS NOT NULL;
```

```
SELECT forfatter_id, fode_aar, til_aar, til_aar - fode_aar FROM noen_forft;
```

FORFATTER_ID	FODE_AAR	TIL_AAR	TIL_AAR-FODE_AAR
1	1937	1969	32
4	1942	2010	68
5	1961	2010	49
7	1859	1952	93
8	1948	2010	62

```
SELECT AVG(til_aar - fode_aar) FROM noen_forft;
```

```
AVG(TIL_AAR-FODE_AAR)
-----
                60
```

OPPGAVE 2C

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

Alternativ 1:

```
CREATE VIEW antall1 AS
  SELECT count(*) AS alle FROM forfatter;
```

```
CREATE VIEW antall2 AS
  SELECT count(*) AS noen FROM noen_forft;
```

```
SELECT * FROM antall1;
```

```
      ALLE
-----
      12
```

```
SELECT * FROM antall2;
```

```
      NOEN
-----
       5
```

```
SELECT antall2.noen*100/antall1.alle AS prosentandel FROM antall1, antall2;
```

```
PROSENTANDEL
-----
          41
```

1 rad er valgt.

Alternativ 2:

```
SELECT COUNT(noen_forft.forfatter_id)*100 / COUNT(forfatter.forfatter_id) AS
prosentandel
  FROM forfatter LEFT JOIN noen_forft ON (forfatter.forfatter_id =
noen_forft.forfatter_id);
```

```
PROSENTANDEL
-----
          41
```

1 rad er valgt.