

Databaser - Arbeidskrav 02

Relasjonsmodellen, del 2

Oppgave 1a

Lag en SQL-spørring som utfører operasjonene seleksjon og projeksjon på tabellen Bok.

```
1 SELECT utgitt_aar
2 FROM bok
3 WHERE utgitt_aar > 1990;
```

PROJECTION er her `utgitt_aar`.

SELECTION er filteret satt på `utgitt_aar > 1990`.

Generelt syntaks:

```
1 SELECT Col1, Col2
2 FROM [Table1]
3 WHERE [Col1] > Result
```

Oppgave 1b

Lag en SQL-spørring som utfører operasjonen produkt (cross join) på tabellene Forlag og Bok.
Beskriv resultatet med egne ord.

```
1 SELECT *
2 FROM forlag
3 CROSS JOIN bok;
```

Funksjonen `CROSS JOIN` fletter sammen kolonnene fra de to tabellene som velges i spørringen.

Oppgave 1c

Lag SQL-spørringer som utfører operasjonene likhetsforening (equijoin) og naturlig forening (natural join) på tabellene Forlag og Bok.
Hva forteller resultatet?

EQUI JOIN

```
1 SELECT *
2 FROM forlag
3 JOIN bok
4 ON forlag.forlag_id = bok.forlag_id;
```

`EQUI JOIN` fletter sammen de to tabellene og bruker elementer som matcher i henhold til kriteriet gitt i `ON`-utsagnet. Den setter kun inn kolonner som matcher i begge tabellene. Vi kan dermed f.eks flette sammen to tabeller hvor Tabell 1 har en kolonne med ett navn som inneholder samme informasjon som en kolonne i en Tabell 2, men som har et annet navn kolonnenavn.

NATURAL JOIN

```
1 SELECT *
2 FROM forlag
3 NATURAL JOIN bok;
```

`NATURAL JOIN` fletter sammen de to tabellene, og kombinerer like kolonner.

Oppgave 1d

Finn eksempler på attributter eller kombinasjoner av attributter som er unionkompatible. Hvilke relasjonsoperasjoner krever at operandene er unionkompatible? Sett opp SQL-spørringer som utfører disse operasjonene, et eksempel på hver. Beskriv med egne ord hva spørringene gir deg svaret på.

For at attributter skal være unionkompatible må vi hente kolonner som inneholder like datatyper, i tillegg til at de må være i samme rekkefølge.

F.eks kan vi hente ut `tittel` og `bok_id`, som begge

```
1 SELECT tittel FROM bok
2 UNION
3 SELECT bok_id from bok_forfatter;
```

Resultatet av denne spørringen blir tittelen på bøker.

Alternativt:

```
1 SELECT fornavn, etternavn FROM forfatter
2 UNION
3 SELECT fornavn, etternavn FROM konsulent;
```

Resultatet her blir en tabell som kombinerer kolonnene `fornavn` og `etternavn` fra tabellene `forfatter` og `konsulent`.

Prøvde å finne en måte å kombinere boktittel, bok-ID og forfatter, for å vite dette i en tabell, men fikk ikke det til.

Oppgave 2a

Bruk SQL til å finne fornavn og etternavn til alle forfatterne sortert alfabetisk på etternavn. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```
1 SELECT etternavn, fornavn
2 FROM forfatter
3 ORDER BY etternavn, fornavn;
```

`ORDER BY` sorterer alfabetisk med `etternavn` som første filter, og `fornavn` som andre filter. Hvis det da er to oppføringer med `etternavn` som er like, vil disse to videresorteres etter `fornavn`.

Oppgave 2b

Bruk SQL til å finne eventuelle forlag (`forlag_id` er nok) som ikke har gitt ut bøker. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```
1 SELECT *
2 FROM forlag
3 WHERE NOT EXISTS (
4     SELECT 1
5     FROM bok
6     WHERE forlag.forlag_id = bok.forlag_id
7 );
```

Her sjekker `WHERE` om `forlag_id` fra `forlag` eksisterer i tabellen `bok`. Den ytre delen (`FROM` og `WHERE NOT EXISTS`) henter ut radene fra tabellen `forlag` som ikke har en tilsvarende verdi i tabellen `bok`.

Resultat:

forlag_id	forlag_navn	adresse	telefon
1	Tapir	Trondheim	73590000
4	Universitetsforlaget	Oslo	23230000
6	Oktober	Oslo	22002200

Uttrykt mer generelt:

```
1 SELECT *
2 FROM tabell1
3 WHERE NOT EXISTS (
4     SELECT 1
5     FROM tabell2
6     WHERE tabell1.id = tabell2.id
7 );
```

Oppgave 2c

Bruk SQL til å finne forfattere som er født før 1900. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```
1 SELECT etternavn, fornavn, fode_aar
2 FROM forfatter
3 WHERE fode_aar < 1900;
```

Oppgave 2d

Bruk SQL til å finne navn og adresse til forlaget som har gitt ut boka 'Sult'. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```
1 SELECT forlag.forlag_navn, forlag.adresse
2 FROM bok
3 JOIN forlag ON bok.forlag_id = forlag.forlag_id
4 WHERE bok.tittel = 'Sult';
```

Bruker `SELECT` for å velge tabell.kolonne `forlag.forlag_navn` og `forlag.adresse`. `FROM` tabell `bok` skal vi bruk `JOIN` til å flette sammen tabellene ut fra å matche kolonnene `forlag_id` fra begge tabellene. Filtrerer den nye tabelleb etter verdien `Sult` og får ut:

forlag_navn	adresse
Gyldendal	Oslo

Lager altså en ny tabell og filtrerer på denne.

Oppgave 2e

Bruk SQL til å finne titlene på bøkene som Hamsun har skrevet. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```
1 SELECT bok.tittel
2 FROM bok
3 JOIN bok_forfatter ON bok.bok_id = bok_forfatter.bok_id
4 JOIN forfatter ON bok_forfatter.forfatter_id = forfatter.forfatter_id
5 WHERE forfatter.etternavn = 'Hamsund';
```

Bruker "Hamsund" i spørringen, siden det er en skrivefeil i datasettet.

Fletter sammen tabellene `bok` og `bok_forfatter` ut fra den felles kolonnen `bok_id`. Deretter flettes den nye tabellen med tabellen `forfatter` ut fra kolonnen `forfatter_id`. Til slutt filtreres verdiene basert på verdien "Hamsund" i kolonnen `forfatter.etternavn`.

Vi kan også bruke et wildcard-søk, med `LIKE`, hvor `%` fungerer som wildcard-tegnet:

```

1 SELECT bok.tittel
2 FROM bok
3 JOIN bok_forfatter ON bok.bok_id = bok_forfatter.bok_id
4 JOIN forfatter ON bok_forfatter.forfatter_id = forfatter.forfatter_id
5 WHERE forfatter.etternavn LIKE 'Hamsun%';

```

Dette vil returnere alle oppføringer som begynner med `Hamsun`, og kan være en god ide om man ikke finner det man forventer å finne.

Oppgave 2f

Bruk SQL til å finne informasjon om bøker og forlagene som har utgitt dem. Én linje i oversikten skal inneholde bokas tittel og utgivelsesår, samt forlagets navn, adresse og telefonnummer. Forlag som ikke har gitt ut noen bøker skal også med i listen. Hvilken eller hvilke operasjoner fra relasjonsalgebraen brukte du?

```

1 SELECT bok.tittel, bok.utgitt_aar, forlag.forlag_navn, forlag.adresse,
   forlag.telefon
2 FROM forlag
3 LEFT JOIN bok ON forlag.forlag_id = bok.forlag_id;

```

`LEFT JOIN`-operasjonen tilsvarer operasjonen `NATURAL JOIN` kombinert med `UNION` for å inkludere de ikke-matchende radene fra venstre tabell.

Output-tabell:

tittel	utgitt_aar	forlag_navn	adresse	telefon
<i>NULL</i>	<i>NULL</i>	Tapir	Trondheim	73590000
Markens grøde	1917	Gyldendal	Oslo	22220000
Victoria	1898	Gyldendal	Oslo	22220000
Sult	1890	Gyldendal	Oslo	22220000
Benoni	1908	Gyldendal	Oslo	22220000
Rosa	1908	Gyldendal	Oslo	22220000
Et skritt	1997	Gyldendal	Oslo	22220000
Den femte	1996	Gyldendal	Oslo	22220000
Villspor	1995	Gyldendal	Oslo	22220000
Silkeridderen	1994	Gyldendal	Oslo	22220000
Den hvite hingsten	1992	Gyldendal	Oslo	22220000
Hunder	1992	Gyldendal	Oslo	22220000
Rebecca	1981	Cappelen	Oslo	22200000

tittel	utgitt_aar	forlag_navn	adresse	telefon
Klosterkrønike	1982	Cappelen	Oslo	22200000
Universet	1988	Cappelen	Oslo	22200000
Nålen	1978	Cappelen	Oslo	22200000
Se terapeuten	1998	Cappelen	Oslo	22200000
Sa mor	1996	Cappelen	Oslo	22200000
Jubel	1995	Cappelen	Oslo	22200000
Tatt av kvinnen	1993	Cappelen	Oslo	22200000
Supernaiv	1996	Cappelen	Oslo	22200000
NULL	NULL	Universitetsforlaget	Oslo	23230000
Gutter er gutter	1995	Aschehaug	Oslo	22000000
Bridget Jones	1995	Aschehaug	Oslo	22000000
NULL	NULL	Oktober	Oslo	22002200

Slutt.