Løsningsforslag

Øving: Normalisering og SQL del 1

Oppgave 1 Normalisering

Tabellen med 13 kolonner ser altså slik ut:

leieforhold(kunde_id, kunde_navn, kunde_adresse, kunde_tlf, eiendom_id, eiendom_adresse, eier_id, eier_navn, eier_adresse, eier_tlf, fra_uke, til_uke, pris)

Siden en person kun kan leie en eiendom av gangen (dvs. i et gitt tidsrom), får vi følgende kandidatnøkler:

- (kunde_id, fra_uke)
- (kunde_id, til_uke)
- (eiendom_id, fra_uke)
- (eiendom_id, til_uke)

fra_uke og til_uke identifiserer en bestemt uke et bestemt år, dvs attributtet må ha verdier som gir både ukenummeret og årstallet.

Vi velger (kunde_id, fra_uke) som primærnøkkel.

Problemer med denne tabellen:

Innlegging av data:

Det er f.eks. ikke er mulig å registrere en eiendom uten samtidig å registrere kunde_id og eiendom _id. M.a.o. vi kan ikke legge inn en eiendom i databasen før vi har et leieforhold på den. Dette er selvfølgelig uholdbart.

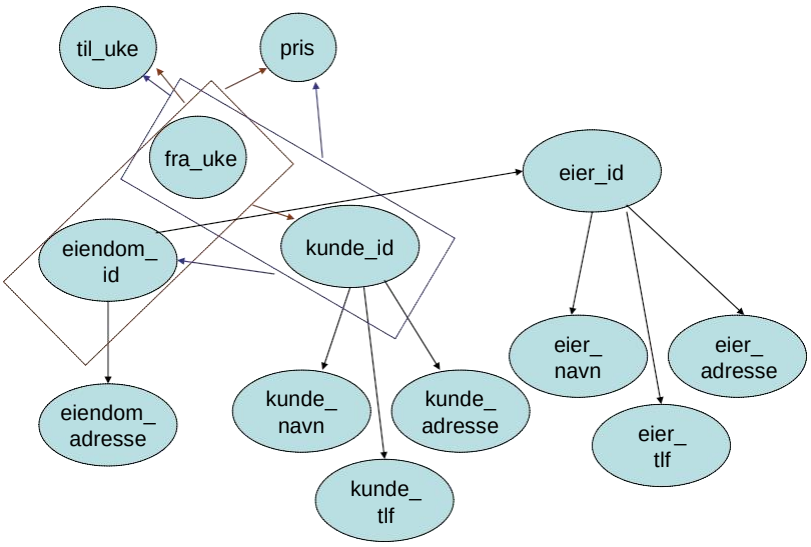
Endring av data:

En update-setning vil enkelt sørge for at alle aktuelle data blir forandret dersom f.eks. en kunde skifter adresse, men det er selvfølgelig helt unødvendig å lagre kundens adresse i tilknytning til hvert eneste leieforhold kunden har. Tilsvarende for data knyttet til eier og eiendom.

Sletting av data:

Her har vi problem motsatt av det vi har ved innlegging av data. Det er ikke mulig å fjerne alle leieforholdene knyttet til en eiendom uten å fjerne eiendommen. Kanskje det ikke er noe problem, kanskje det er sånn vi vil ha det? Men kanskje dette er den eneste eiendommen denne eieren har. Da forsvinner eieren også!

Følgende diagram viser funksjonelle avhengigheter mellom attributtene:



Vi kan bruke til_uke på samme måte som fra_uke. Det vil si at vi har enda flere determinanter enn de som framgår av figuren. Vi kan sette opp følgende tabell over determineringer:

Determinant	**bestemmer entydig**
eiendom_id	eiendom_adresse, eier_id
eiendom_id, fra_uke	til_uke, pris, kunde_id
eiendom_id, til_uke	fra_uke, pris, kunde_id
kunde_id, fra_uke	til_uke, pris, eiendom_id
kunde_id, til_uke	fra_uke, pris, eiendom_id
eier_id	eier_navn, eier_adresse, eier_tlf
kunde_id	kunde_navn, kunde_adresse, kunde_tlf

Vi skal sette opp relasjoner på BCNF direkte fra figuren og tabellen over. Enhver determinant skal bli kandidatnøkkel i våre nye relasjoner (her er primærnøklene understreket):

eiendom(<u>eiendom_id</u>, eiendom_adresse, eier_id)
eier(<u>eier_id</u>, eier_navn, eier_adresse, eier_tlf)
kunde(<u>kunde_id</u>, kunde_navn, kunde_adresse, kunde_tlf)
leieforhold(<u>eiendom_id, fra_uke</u>, til_uke, pris, kunde_id)

I siste relasjon har vi følgende alternative nøkler (som også er determinanter): (eiendom_id, til_uke), (kunde_id, til_uke) og (kunde_id, fra_uke).

Kan vi løse oppgaven ved å gjennomføre prosessen 1NF --> 2NF --> 3NF? Alle tre kravene på side 93 i boka er oppfylt. Dermed må BCNF benyttes.

Oppgave 2 SQL

Datainnhold:

```
select * from poststed;
select * from borettslag;
select * from bygning;
select * from andelseier;
select * from leilighet;
```

POSTSTED:

POSTNR	POSTSTED
2020	Skedsmokorset
6408	Aureosen
7033	Trondheim
7020	Trondheim

BORETTSLAG:

BOLAG_NAVN	BOLAG_ADR	ETABL_AAR	POSTNR
Tertitten	Åsveien 100	1980	7020
Sisiken	Brurød	1990	7033
Lerken	Storgt 5	2000	6408

BYGNING:

BYGN_ID	BYGN_ADR	ANT_ETASJER	BOLAG_NAVN	POSTNR
1	Åsveien 100a	3	Tertitten	7020
2	Åsveien 100b	3	Tertitten	7020
3	Åsveien 100c	6	Tertitten	7020
4	Storgt 10	2	Sisiken	7020
5	Åsveien 100	1	Tertitten	7020

ANDELEIER:

AND_EIER_NR	FORNAVN	ETTERNAVN	TELEFON	ANSIENNITET	BOLAG_NAVN
1	Even	Trulsbo	56667743	3	Tertitten
2	Anna	Olsen	45674588	10	Tertitten
3	Ingrid	Olsen	45785388	8	Tertitten
4	Arne	Torp	78565388	7	Tertitten
5	Arne	Martinsen	78555388	4	Sisiken

LEILIGHET:

LEIL_NR	ANT_ROM	ANT_KVM	ETASJE	BYGN_ID	AND_EIER_NR
1	5	110	3	1	1
2	5	110	3	1	2
3	2	50	1	3	3
5	5	110	1	1	4

////////// OPPGAVER MED LØNINGSFORSLAG ////////////////////////////////

Noen steder er det satt opp flere alternative løsninger. Merk at det kan finnes enda flere løsninger som gir riktig svar.

1. Finn alle borettslag etablert i årene 1975-1985.

```
SELECT *
FROM borettslag
WHERE etabl_aar BETWEEN 1975 AND 1985;
```

```
BOLAG_NAVN           BOLAG_ADR                                ETABL_AAR
POSTNR
-------------------- ---------------------------------------- ---------- --------
--
Tertitten            Åsveien 100                                    1980
7020
```

2.Skriv ut en liste over andelseiere. Listen skal ha linjer som ser slik ut:
"fornavn etternavn, ansiennitet: ansiennitet år".
Listen skal være sortert på ansiennitet, de med lengst ansiennitet øverst.

MySQL:

```
SELECT CONCAT(fornavn, ' ', etternavn, ',  ansiennitet: ', ansiennitet, ' År')
FROM andelseier
ORDER BY ansiennitet DESC;
```

```
FORNAVN||''||ETTERNAVN||',ANSIENNITET:'||ANSIENNITET||'ÅR.'
--------------------------------------------------------------------
Anna Olsen, ansiennitet: 10 år.
Ingrid Olsen, ansiennitet: 8 år.
Arne Torp, ansiennitet: 7 år.
Arne Martinsen, ansiennitet: 4 år.
Even Trulsbo, ansiennitet: 3 år.
```

3. I hvilket år ble det eldste borettslaget etablert?

```
SELECT MIN(etabl_aar) "Etabl.år eldste lag"
FROM borettslag;
```

```
Etabl.år eldste lag
-------------------
               1980
```

4. Finn adressene til alle bygninger som inneholder leiligheter med minst tre rom.

```
Alternativ 1:
SELECT DISTICT bygn_adr
FROM bygning, leilighet
WHERE bygning.bygn_id = leilighet.bygn_id AND ant_rom >= 3;

Alternativ 2:
SELECT DISTICT bygn_adr
FROM bygning JOIN leilighet ON (bygning.bygn_id = leilighet.bygn_id)
WHERE ant_rom >= 3;
```

```
BYGN_ADR
----------------------------------------
Åsveien 100a
```

5. Finn antall bygninger i borettslaget "Tertitten".

```
SELECT count(*) "Antall bygninger"
FROM bygning
WHERE bolag_navn = 'Tertitten';
```

```
Antall bygninger
----------------
               4
```

6. Lag en liste som viser antall bygninger i hvert enkelt borettslag.
Listen skal være sortert på borettslagsnavn. Husk at det kan finnes borettslag
uten bygninger.

```
SELECT borettslag.bolag_navn, COUNT(bygning.bolag_navn) "Antall bygninger"
FROM borettslag LEFT JOIN bygning ON (borettslag.bolag_navn = bygning.bolag_navn)
GROUP BY borettslag.bolag_navn ORDER BY borettslag.bolag_navn;
```

```
BOLAG_NAVN           Antall bygninger
-------------------- ----------------
Lerken                              0
Sisiken                             1
Tertitten                           4
```

7. Finn antall leiligheter i borettslaget "Tertitten".

```
Alternativ 1:
SELECT COUNT(*) "Antall leiligheter"
FROM bygning JOIN leilighet ON (bygning.bygn_id = leilighet.bygn_id)
WHERE bygning.bolag_navn = 'Tertitten';

Alternativ 2:
SELECT COUNT(*) "Antall leiligheter"
FROM bygning, leilighet
WHERE bygning.bygn_id = leilighet.bygn_id
AND bygning.bolag_navn = 'Tertitten';
```

```
Antall leiligheter
------------------
                 4
```

8. Hvor høyt kan du bo i borettslaget "Tertitten"?

```
Alternativ 1:
SELECT max(etasje) "Max etasje"
FROM leilighet
WHERE bygn_id IN
(SELECT bygn_id FROM bygning WHERE bolag_navn = 'Tertitten');

Alternativ 2:
SELECT max(etasje) "Max etasje"
FROM bygning JOIN leilighet ON (bygning.bygn_id = leilighet.bygn_id)
WHERE bygning.bolag_navn = 'Tertitten';

Alternativ 3:
SELECT max(etasje) "Max etasje"
FROM leilighet JOIN bygning ON (leilighet.bygn_id = bygning.bygn_id)
WHERE bolag_navn = 'Tertitten';
```

```
Max etasje
----------
         3
```

9. Finn navn og nummer til andelseiere som ikke har leilighet.

```
SELECT fornavn, etternavn, and_eier_nr
FROM andelseier
WHERE and_eier_nr NOT IN (SELECT and_eier_nr FROM leilighet);
```

```
FORNAVN                        ETTERNAVN                      AND_EIER_NR
------------------------------ ------------------------------ -----------
Arne                           Martinsen                                5
```

10. Finn antall andelseiere pr borettslag, sortert etter antallet. Husk at det kan finnes borettslag uten andelseiere.

```
SELECT borettslag.bolag_navn "Borettslag navn", COUNT(andelseier.and_eier_nr)
"Antall andelseiere"
FROM borettslag LEFT JOIN andelseier ON (borettslag.bolag_navn =
andelseier.bolag_navn)
GROUP BY borettslag.bolag_navn
ORDER BY "Antall andelseiere" DESC;
```

```
Borettslag navn      Antall andelseiere
-------------------- ------------------
Tertitten                             4
Sisiken                               1
Lerken                                0
```

11. Skriv ut en liste over alle andelseiere. For de som har leilighet, skal leilighetsnummeret skrives ut.

```
SELECT andelseier.and_eier_nr, fornavn, etternavn, leil_nr
FROM andelseier LEFT JOIN leilighet ON andelseier.and_eier_nr =
leilighet.and_eier_nr;
```

```
AND_EIER_NR FORNAVN                        ETTERNAVN                LEIL_NR
----------- ------------------------------ -------------------- ----------
          1 Even                           Trulsbo                       1
          2 Anna                           Olsen                         2
          3 Ingrid                         Olsen                         3
          4 Arne                           Torp                          4
          5 Arne                           Martinsen
```

12. Hvilke borettslag har leiligheter med eksakt 4 rom?

```
SELECT DISTINCT bolag_navn
FROM bygning, leilighet
WHERE bygning.bygn_id = leilighet.bygn_id AND ant_rom = 4;
```

```
Ingen rader valgt.
```

13. Skriv ut en liste over antall andelseiere pr postnr og poststed, begrenset til de som bor i leiligheter tilknyttet et borettslag. Husk at postnummeret til disse er postnummeret til bygningen de bor i, og ikke postnummeret til borettslaget. Du trenger ikke ta med poststeder med 0 andelseiere.

Vi kopler sammen tre tabeller: Leilighet, Bygning, Poststed
(Det er tiltstrekkelig med tre tabeller pga and_eier_nr er med i tabellen leilighet)

Alternativ 1:

```
SELECT p.postnr, p.poststed, COUNT(*) "Antall andelseiere"
FROM leilighet l, bygning b, poststed p
WHERE l.bygn_id = b.bygn_id AND b.postnr = p.postnr
GROUP BY p.postnr, p.poststed;
```

Alternativ 2:

```
SELECT poststed.postnr, poststed.poststed, COUNT(*) "Antall andelseiere"
FROM leilighet JOIN bygning ON (leilighet.bygn_id = bygning.bygn_id)
JOIN poststed ON (bygning.postnr = poststed.postnr)
GROUP BY poststed.postnr, poststed.poststed;
```

```
    POSTNR POSTSTED             Antall andelseiere
---------- -------------------- ------------------
      7020 Trondheim                             4
```

Ekstraoppgave:
Hva hvis vi vil ha ut poststeder med 0 andelseiere?

```
SELECT poststed.postnr, poststed.poststed, COUNT(leilighet.and_eier_nr) "Antall
andelseiere"
FROM leilighet  JOIN bygning ON (leilighet.bygn_id = bygning.bygn_id) RIGHT JOIN
poststed ON (bygning.postnr = poststed.postnr)
GROUP BY poststed.postnr, poststed.poststed;
```

Merk at vi ikke kan bruke count(*) pga at NULL-verdier da blir talt med.

```
    POSTNR POSTSTED             Antall andelseiere
---------- -------------------- ------------------
      2020 Skedsmokorset                         0
      6408 Aureosen                              0
      7020 Trondheim                             4
      7033 Trondheim                             0
```