

Eksamen DB1102 H2021

Hjemmeeksamen 7.desember - 8. desember 2021.

Teller 100% av karakteren i emnet. Vurdering: Bestått / Ikke bestått.

Alle hjelpemidler tillatt. Ikke tillatt å samarbeide med andre.

Når du leverer eksamen, skal du samle besvarelsen din i ett pdf-dokument. Dokumentet inneholder svar på alle oppgavene.

Oppgave 1 – modellering (30%)

Robinsonekspedisjonen er endelig tilbake på TV-skjermen. Det begynner å bli så mange sesonger nå, at vi trenger en database som kan holde orden på alle episodene, deltakerne og noen hendelser som har skjedd. Hvis du ikke vet hva Robinsonekspedisjonen er, så er det et av de første realityprogrammene på norsk TV. Et kjapt Google-søk vil gi deg mer informasjon.

Vi må ha oversikt over alle sesongene; årstall og hvem som var programleder. Det har vært tre programledere i den norske versjonen (og det er den norske versjonen vi modellerer): Nils Ove Oftebro, Christer Falck og Silje Torp. For en sesong må vi vite hva premien var for å vinne. Vi må også kunne lagre en tekstlig, overordnet beskrivelse av sesongen som helhet. For programlederne ønsker vi å vite når de er født, og hva de heter.

Vi ønsker å lagre data om hver enkelt episode. En episode hører naturlig nok til en sesong. Vi ønsker å lagre premieredato for alle episodene. Vi trenger nok også å kunne lagre en beskrivelse av episoden.

La oss ikke glemme deltakerne. Vi ønsker å lagre navn, fødselsdato, bosted og yrke (da de var med i Robinson) og hvilken personlig eiendel de valgte å ha med på øya.

En episode har gjerne noen faste hendelser, blant annet:

- En Robinsonkamp der deltakerne i kampen kan vinne immunitet i neste øyråd.
- Øyråd: En avstemming der den med flest stemmer blir stemt ut av ekspedisjonen.

Robinsonkamp: Robinsonkampene er individuelle. Vi ønsker å lagre hvem som deltok i kampen, hvilken episode kampen foregikk og hvem som vant immunitet.

Øyråd: Vi ønsker å vite hvilke deltakere som mottok stemmer i Øyråd. Eks: Kåre Hansen 3 stemmer, Else Jensen 1 stemme og Balder Haukedal 4 stemmer. Vi har ikke mulighet til å lagre hvem som stemte på hvem, for det er som oftest ukjent. Vi må også vite i hvilken episode øyrådet ble avholdt.

En sesong avsluttes med en siste avstemming for å kåre en vinner. Noen av deltakerne stemmer på den de mener fortjener seieren, og den med flest stemmer vinner ekspedisjonen. Vi ønsker å ha oversikt over alle disse avsluttende avstemningene som har vært i Robinsonekspedisjonene; hvem som ga sin stemme til hvilken kandidat. For her kan du

forutsette at vi kjenner til hvem som stemte på hvem. I siste sesong fikk en av deltakerne anledning til å gi to stemmer. Du må derfor ta høyde for at en deltaker kan gi flere stemmer i denne avstemmingen.

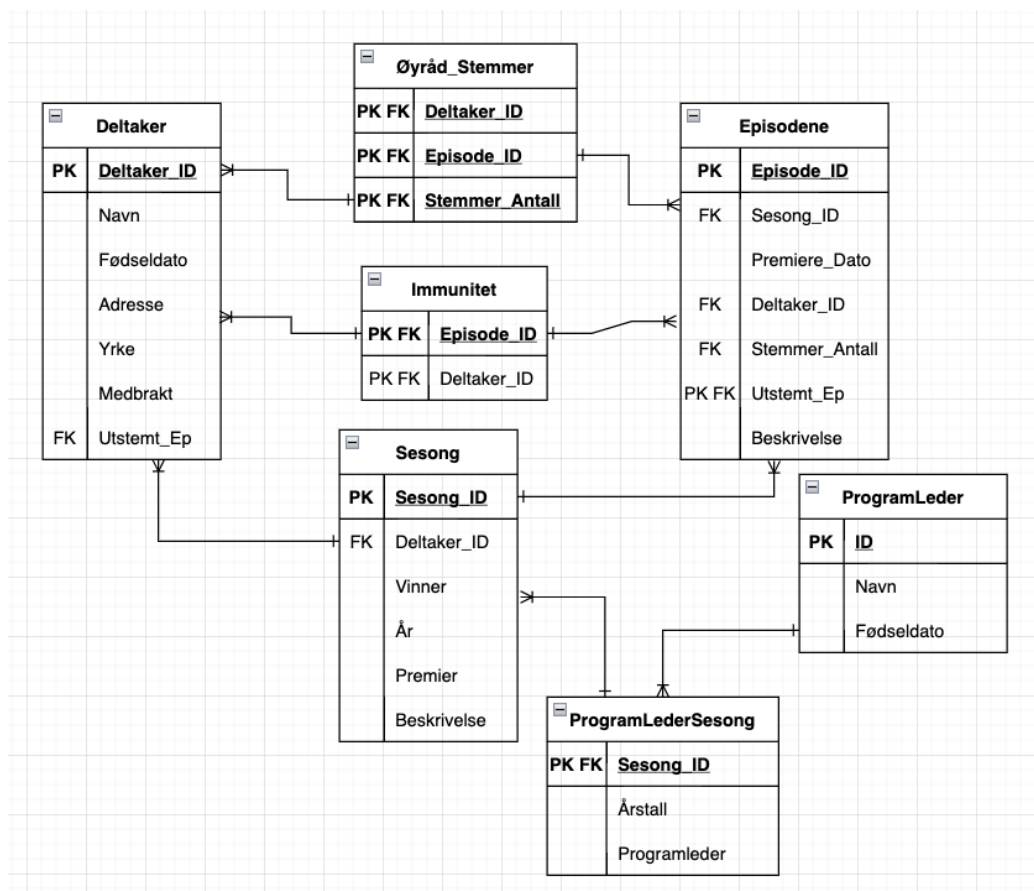
Oppgave: Tegn en modell for din foreslåtte løsning. Du kan selv velge om du vil benytte kråkefot eller UML notasjon. Velger du kråkefot trenger du ikke skille mellom identifiserende og ikke- identifiserende forhold. (UML notasjon har uansett ikke skille på dette.) Modellen din skal inneholde:

- Entitetene og deres attributter.
- Primærnøkler og fremmednøkler.
- Relasjonene mellom entitetene.
- Multiplisiteten (deltagelse og kardinalitet) for relasjonene.
- Hvis nødvendig, koblingsentiteter.

Hvis du synes noe er uklart, så gjør dine egne antagelser. Husk i så fall å gjøre rede for disse.

Vi anbefaler å benytte et program, som for eksempel Lucidchart, til å tegne modellen. Du kan også velge å tegne for hånd og lime inn bildet av tegningen din, men det kan da være vanskeligere for sensor å tyde den.

Løsning:



PS:

- 1) Ved avstemming i tabell «Øyråd_Stemmer» vil deltaker som oppnår flest stemmer resultere i utstemning. Vedkommende velger hvem som får immunitet i neste episode. Jeg har derfor valgt å legge til en Attributt; Utmeldt.
- 2) Ved sesongavslutning gjennomføres en sluttavstemning for å finne en vinner der alle deltagerne i sesongen kan avgi stemme. I tabellen «Sesong» Jeg har laget en attributt «Vinner» der deltager og deres stemmer for deltagerkandidatene registreres. Den deltager som får mest stemmer (MAX) blir vinner.
- 3) «Premier» er en tekst som beskriver premien.
- 4) «Beskrivelse» er en tekst som beskriver henholdsvis episodene og sesongene.

Oppgave 2 – SQL (50%)

I eksamensoppgaven vil du finne et SQL-script (lotto.sql) som opprettet en database (lotto) og fyller tabeller med data. Kjør SQL-scriptet mot din egen MySQL-server. Deretter løser du oppgavene som beskrevet nedenfor. Svaret på hver deloppgave er todelt:

1. SQL som løser oppgaven.
2. En skjermdump som viser resultatet ditt når SQL er kjørt.

Eksempel:

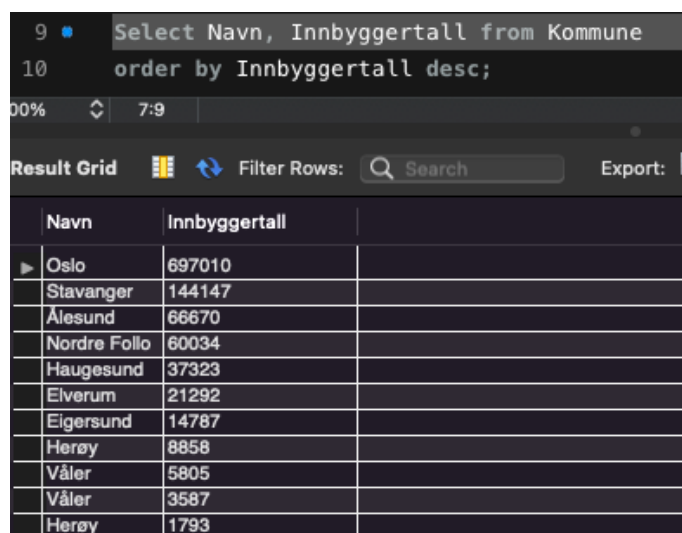
Hvis jeg hadde hatt en oppgave fra world-databasen: «Hent ut navn på alle land som begynner på bokstaven D», så ville et svar kunne sett slik ut:

```
SELECT name from country WHERE name LIKE 'D%';
```

Men du skal altså benytte **lotto-databasen** som er vedlagt eksamensoppgaven. Husk at det er bedre å gjøre et forsøk på å besvare en oppgave enn å svare blankt.

1. a) Lag en spørring som gir informasjon om hvilken av de registrerte kommunene som har størst innbyggertall, og hva denne kommunen heter.

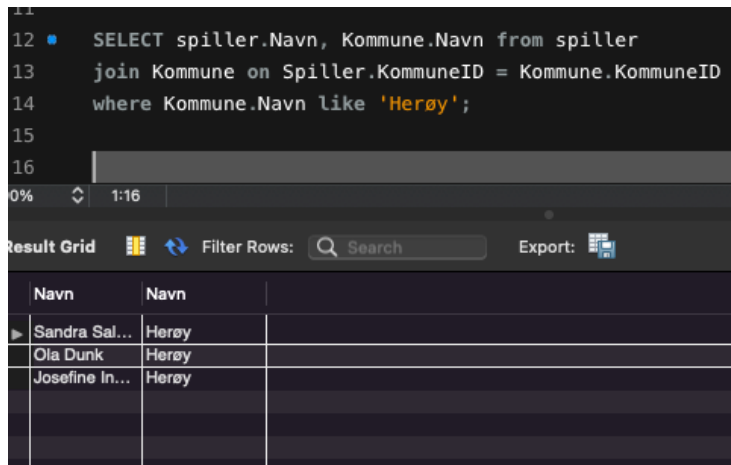
Select Navn, Innbyggertall from Kommune
order by Innbyggertall desc;



Navn	Innbyggertall
Oslo	697010
Stavanger	144147
Ålesund	66670
Nordre Follo	60034
Haugesund	37323
Elverum	21292
Eigersund	14787
Herøy	8858
Våler	5805
Våler	3587
Herøy	1793

2. b) Lag en spørring som gir informasjon om hvilke registrerte spillere som bor i en kommune som heter Herøy.

```
SELECT spiller.Navn, Kommune.Navn from spiller
join Kommune on Spiller.KommuneID = Kommune.KommuneID
where Kommune.Navn like 'Herøy';
```



The screenshot shows a SQL query editor with the following query:

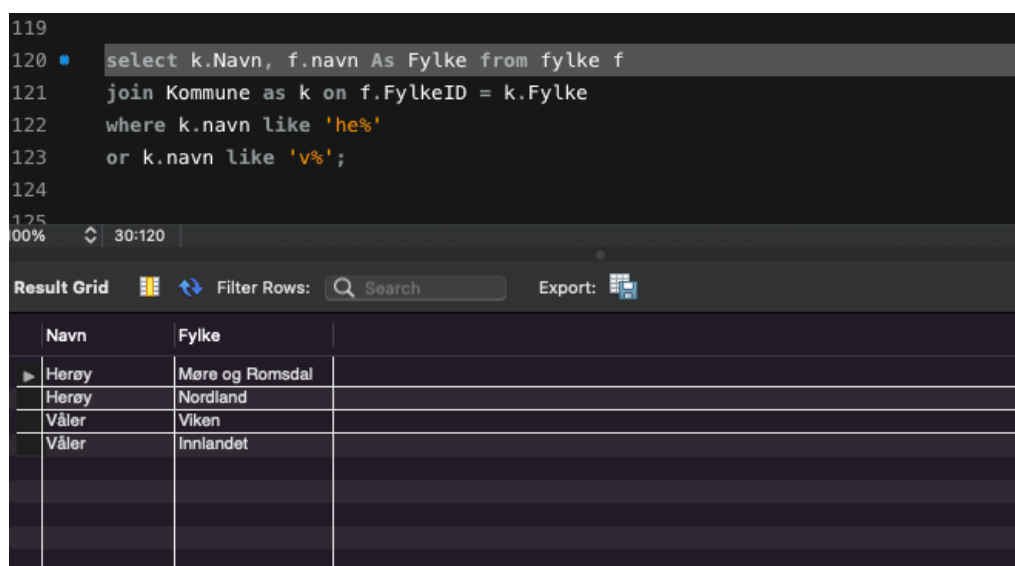
```
SELECT spiller.Navn, Kommune.Navn from spiller
join Kommune on Spiller.KommuneID = Kommune.KommuneID
where Kommune.Navn like 'Herøy';
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'Navn' and 'Navn'. The results are as follows:

Navn	Navn
Sandra Sal...	Herøy
Ola Dunk	Herøy
Josefine In...	Herøy

3. c) Lag en spørring som gir navn på alle registrerte kommuner som har samme navn, men ligger i forskjellige fylker.

```
Select k.Navn, f.navn As Fylke from fylke f
join Kommune as k on f.FylkeID = k.Fylke
where k.navn like 'he%'
or k.navn like 'v%';
```



The screenshot shows a SQL query editor with the following query:

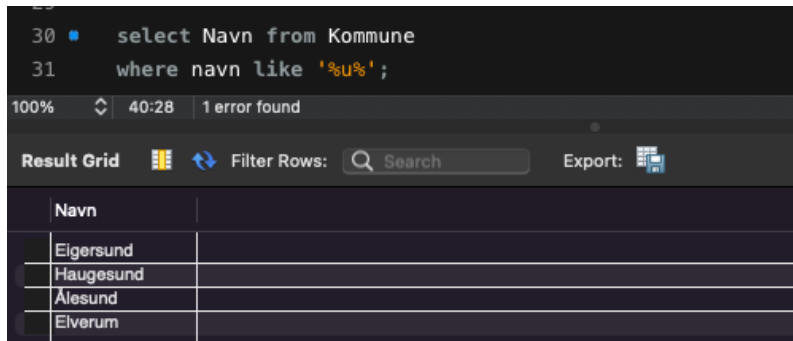
```
select k.Navn, f.navn As Fylke from fylke f
join Kommune as k on f.FylkeID = k.Fylke
where k.navn like 'he%'
or k.navn like 'v%';
```

Below the query editor is a 'Result Grid' showing the results of the query. The grid has two columns: 'Navn' and 'Fylke'. The results are as follows:

Navn	Fylke
Herøy	Møre og Romsdal
Herøy	Nordland
Våler	Viken
Våler	Innlandet

4. d) Lag en spørring som viser hvor mange registrerte kommuner som har en liten 'u' i navnet sitt. Navngi kolonnen i svaret: Antall U-Kommuner.

```
select Navn from Kommune
where navn like '%u%';
```

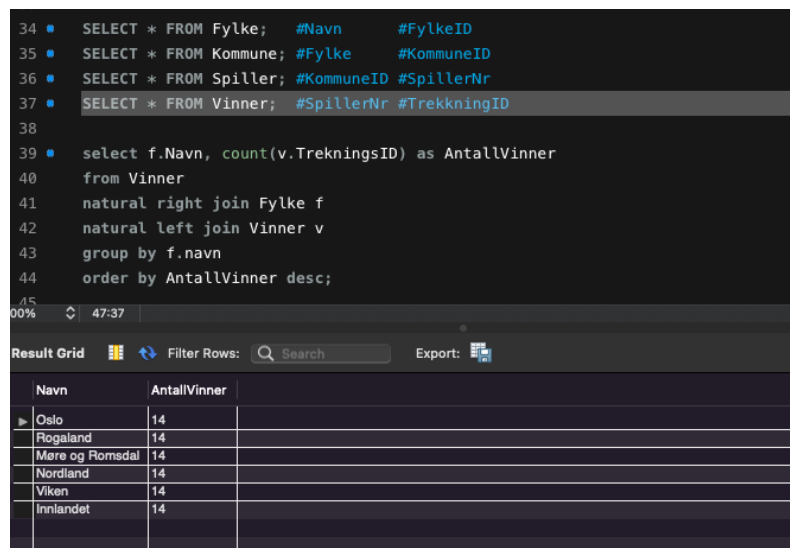


Navn
Eigersund
Haugesund
Ålesund
Elverum

5. e) Lag en spørring som viser hvilket fylke som samlet sett har hatt flest premievinnere til nå. Resultatet skal vise fylket, og antall vinnere. Hvis noen har vunnet flere ganger, så skal de telles for hver gang de vinner.

```
SELECT * FROM Fylke;      #Navn      #FylkeID
SELECT * FROM Kommune;    #Fylke    #KommuneID
SELECT * FROM Spiller;    #KommuneID #SpillerNr
SELECT * FROM Vinner;     #SpillerNr #TrekkningsID
```

```
select f.Navn, count(v.TrekkningsID) as AntallVinner
from Vinner
natural right join Fylke f
natural left join Vinner v
group by f.navn
order by AntallVinner desc;
```

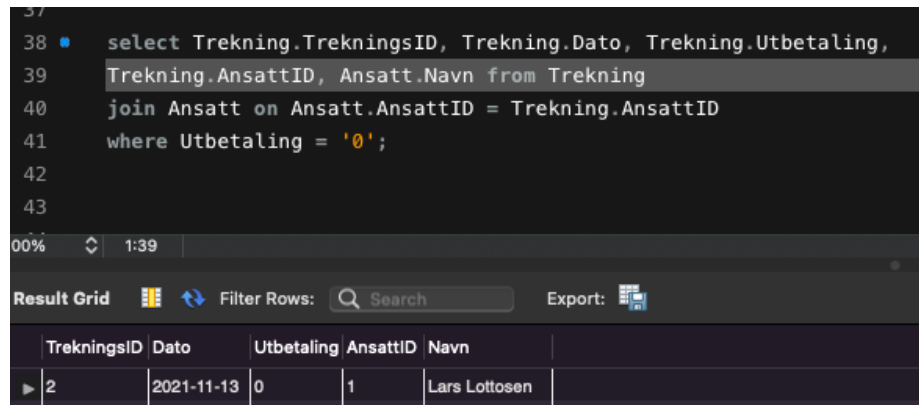


Navn	AntallVinner
Oslo	14
Rogaland	14
Møre og Romsdal	14
Nordland	14
Viken	14
Innlandet	14

Kommentar: Jeg er ikke lykket å koble tabellene siden det finnes ikke noe identifikatorer mellom fylke og spiller vinner.

6. f) Lag en spørring som viser hvilke trekninger som ikke har hatt noen vinnere. Resultatet skal vise trekningens dato, og navnet på hvem som var trekningsansvarlig.

```
Select Trekning.TrekningsID, Trekning.Dato, Trekning.Utbetaling,  
Trekning.AnsattID, Ansatt.Navn from Trekning  
join Ansatt on Ansatt.AnsattID = Trekning.AnsattID  
where Utbetaling = '0';
```



The screenshot shows a SQL query editor with the following query:

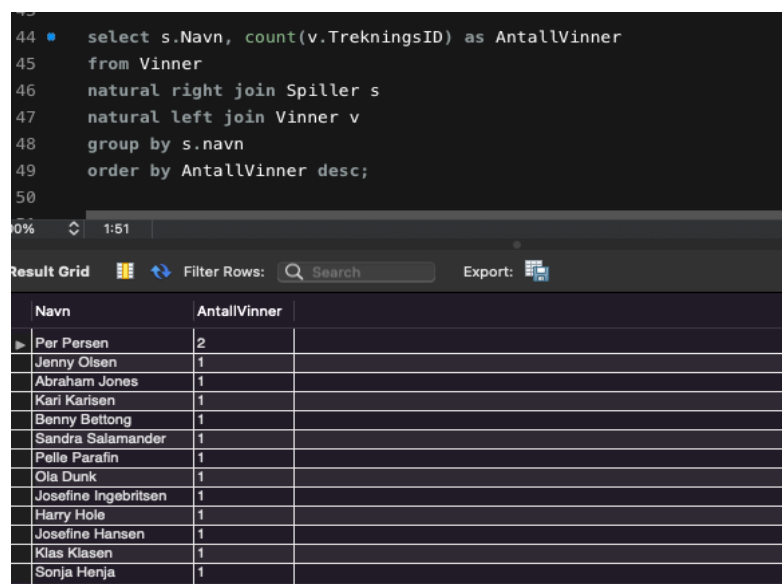
```
37  
38 select Trekning.TrekningsID, Trekning.Dato, Trekning.Utbetaling,  
39 Trekning.AnsattID, Ansatt.Navn from Trekning  
40 join Ansatt on Ansatt.AnsattID = Trekning.AnsattID  
41 where Utbetaling = '0';  
42  
43
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns: TrekningsID, Dato, Utbetaling, AnsattID, and Navn. The results are as follows:

TrekningsID	Dato	Utbetaling	AnsattID	Navn
2	2021-11-13	0	1	Lars Lottosen

7. g) Lag en spørring som viser navn på spillere har vunnet flere enn en gang, hvor mange ganger de har vunnet, og hvilken kommune de bor i.

```
select s.Navn, count(v.TrekningsID) as AntallVinner  
from Vinner  
natural right join Spiller s  
natural left join Vinner v  
group by s.navn  
order by AntallVinner desc;
```



The screenshot shows a SQL query editor with the following query:

```
44 select s.Navn, count(v.TrekningsID) as AntallVinner  
45 from Vinner  
46 natural right join Spiller s  
47 natural left join Vinner v  
48 group by s.navn  
49 order by AntallVinner desc;  
50
```

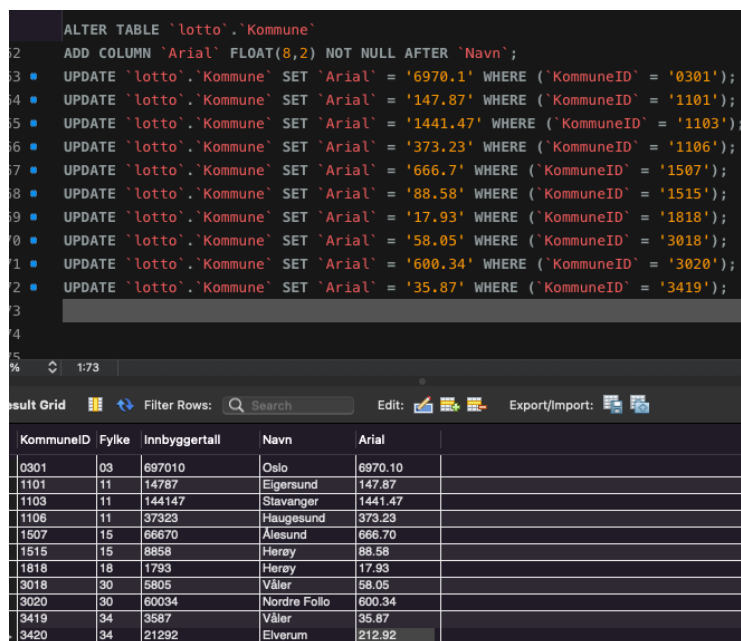
Below the query editor is a "Result Grid" showing the results of the query. The grid has columns: Navn and AntallVinner. The results are as follows:

Navn	AntallVinner
Per Persen	2
Jenny Olsen	1
Abraham Jones	1
Kari Karlsen	1
Benny Bettong	1
Sandra Salamander	1
Pelle Parafin	1
Ola Dunk	1
Josefine Ingebritsen	1
Harry Hole	1
Josefine Hansen	1
Klas Klasen	1
Sonja Henja	1

Kommentar: Jeg fikk ikke til attributtet « Kommune navn » det ble « error » hver gang.

8. h) Legg inn en ny kolonne Areal i kommunetabellen. Legg inn fornuftige verdier i den nye kolonnen for de eksisterende kommunene. Velg datatype du selv mener er passende. Arealet skal oppgis i antall kvadratkilometer, med to desimaler.

```
ALTER TABLE `lotto`.`Kommune`  
ADD COLUMN `Areal` FLOAT(8,2) NOT NULL AFTER `Navn`;  
UPDATE `lotto`.`Kommune` SET `Areal` = '6970.1' WHERE (`KommuneID` = '0301');  
UPDATE `lotto`.`Kommune` SET `Areal` = '147.87' WHERE (`KommuneID` = '1101');  
UPDATE `lotto`.`Kommune` SET `Areal` = '1441.47' WHERE (`KommuneID` = '1103');  
UPDATE `lotto`.`Kommune` SET `Areal` = '373.23' WHERE (`KommuneID` = '1106');  
UPDATE `lotto`.`Kommune` SET `Areal` = '666.7' WHERE (`KommuneID` = '1507');  
UPDATE `lotto`.`Kommune` SET `Areal` = '88.58' WHERE (`KommuneID` = '1515');  
UPDATE `lotto`.`Kommune` SET `Areal` = '17.93' WHERE (`KommuneID` = '1818');  
UPDATE `lotto`.`Kommune` SET `Areal` = '58.05' WHERE (`KommuneID` = '3018');  
UPDATE `lotto`.`Kommune` SET `Areal` = '600.34' WHERE (`KommuneID` = '3020');  
UPDATE `lotto`.`Kommune` SET `Areal` = '35.87' WHERE (`KommuneID` = '3419');
```



The screenshot shows a database management interface. The top section displays the executed SQL commands, which include an ALTER TABLE statement to add a new column 'Areal' and several UPDATE statements to populate it for various municipalities. Below the SQL editor, a 'Result Grid' is visible, showing the data for the 'Kommune' table. The grid has columns for 'KommuneID', 'Fylke', 'Innbyggertall', 'Navn', and 'Areal'. The data is as follows:

KommuneID	Fylke	Innbyggertall	Navn	Areal
0301	03	697010	Oslo	6970.10
1101	11	14787	Eigersund	147.87
1103	11	144147	Stavanger	1441.47
1106	11	37323	Haugesund	373.23
1507	15	66670	Ålesund	666.70
1515	15	8858	Herøy	88.58
1818	18	1793	Herøy	17.93
3018	30	5805	Våler	58.05
3020	30	60034	Nordre Follo	600.34
3419	34	3587	Våler	35.87
3420	34	21292	Elverum	212.92

9. i) Det har vært en ny trekning. Legg inn følgende informasjon i databasen: Trekningen ble avholdt 4. desember 2021. Det var nøyaktig 11 millioner i utbetaling. Det var en ny trekningsansvarlig: Jens Jensen, som bor i Oslo. Det var to vinnere som delte utbetalingen: Lars Andersen, som bor i Ålesund (Lilliveien 56) og Line Jensen som bor på Elverum (Blåklukkaleen 4).

```

77 • UPDATE `lotto`.`Kommune` SET `Areal` = '212.92' WHERE (`KommuneID` = '3420');
78 • INSERT INTO `lotto`.`Ansatt` (`AnsattID`, `Navn`) VALUES ('3', 'Jens Jensen');
79 • INSERT INTO `lotto`.`Trekning` (`TrekningsID`, `Dato`, `Utbetaling`)
80   VALUES ('5', '2021-12-04', '11000000');
81 • UPDATE `lotto`.`Trekning` SET `AnsattID` = '3' WHERE (`TrekningsID` = '5');
82 • INSERT INTO `lotto`.`Spiller` (`SpillerNr`, `Navn`, `Adresse`)
83   VALUES ('14', 'Lars Andersen', 'Lilliveien 56');
84 • INSERT INTO `lotto`.`Spiller` (`SpillerNr`, `Navn`, `Adresse`)
85   VALUES ('15', 'Line Jensen', 'Blåklukkaleen 4');
86 • UPDATE `lotto`.`Spiller` SET `KommuneID` = '1507' WHERE (`SpillerNr` = '14');
87 • UPDATE `lotto`.`Spiller` SET `KommuneID` = '3420' WHERE (`SpillerNr` = '15');
88 • INSERT INTO `lotto`.`Vinner` (`SpillerNr`, `TrekningsID`) VALUES ('14', '5');
89 • INSERT INTO `lotto`.`Vinner` (`SpillerNr`, `TrekningsID`) VALUES ('15', '5');
90
91
92

```

TrekningsID	Dato	Utbetaling	AnsattID
1	2021-11-06	9756192	1
2	2021-11-13	0	1
3	2021-11-20	21234543	2
4	2021-11-27	8765294	2
5	2021-12-04	11000000	3
NULL	NULL	NULL	NULL

AnsattID	Navn	KommuneID
1	Lars Lottosen	0301
2	Madeleine Heldigsen	1818
3	Jens Jensen	0301
NULL	NULL	NULL

SpillerNr	TrekningsID
5	3
6	3
7	3
8	3
1	4
9	4
10	4
11	4
12	4
13	4
14	5
15	5
NULL	NULL

SpillerNr	Navn	Adresse	KommuneID
4	Kari Karlsen	Amalies gate 2	1103
5	Benny Bettong	Kong vinters gate 77	1106
6	Sandra Salamander	Sesams gate 12	1515
7	Pelle Parafin	Parafinveien 11	1507
8	Ola Dunk	Sportveien 3	1818
9	Josefine Ingebritsen	Kuldegropen 18	1818
10	Harry Hole	Hurtigvingen 2	3018
11	Josefine Hansen	Frydenbergveien 111	3020
12	Klas Klasen	Brattebakke 5	3419
13	Sonja Henja	Økernveien 18	3420
14	Lars Andersen	Lilliveien 56	1507
15	Line Jensen	Blåklukkaleen 4	3420
NULL	NULL	NULL	NULL

10. j) (Vanskelig) Lag et view som viser hvilke fylker som har vunnet hvor mye penger. Viewet skal inneholde fylkets navn, og totale utbetalinger til fylkets spillere, sortert slik at fylket som har vunnet mest kommer først.

```
138 CREATE OR REPLACE VIEW FylkesVinner AS(  
139     SELECT f.navn, t.Utbetaling, s.Spiller  
140     FROM fylke f  
141     JOIN Vinner v ON v.SpillerNr = s.SpillerNr  
142     ORDER BY FylkesVinner);  
143 Resultat (SELECT *):  
144  
145  
146  
147  
148  
149  
150  
151
```

100% 21:143 1 error found

Action Output

Time	Action
------	--------

Kommentar: Jeg vet at jeg er på riktig spor men langt fra mål.

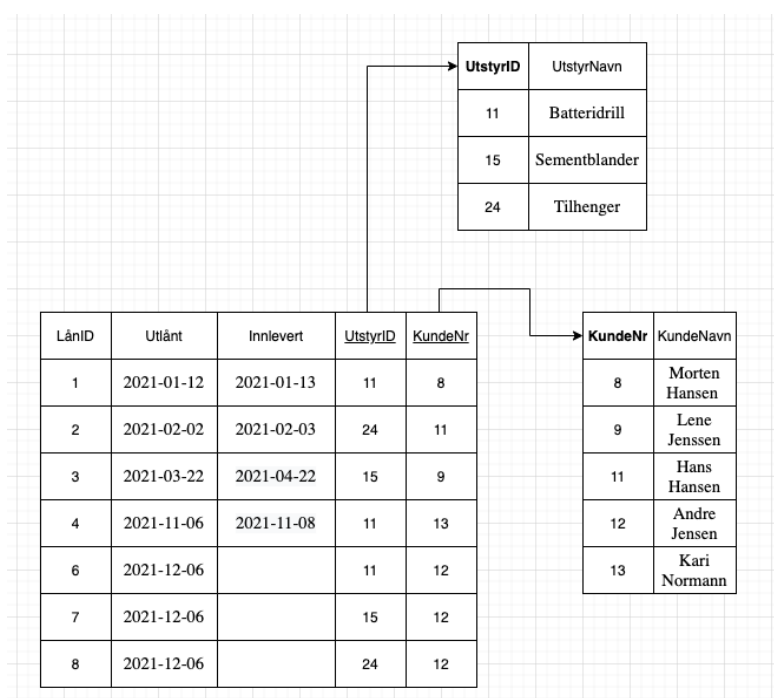
Oppgave 3 – Normalisering (20%)

En bedrift leier ut diverse verktøy og utstyr til privatmarkedet. De har en tabell i en database som holder oversikt over utleien. Kolonnenavn i uthevet (bold) er primærnøkler. Et utdrag av tabellen

vises nedenfor:

Kundenr	UtstyrId	Kundenavn	Utstyr	Utlånt	Innlevert
8	37	Morten Hansen	Batteridrill	2021-01-12	2021-01-13
9	15	Lene Jenssen	Sementblander	2021-03-22	2021-04-22
11	24	Hans Hansen	Tilhenger	2021-02-02	2021-02-03
12	15	Andre Jensen	Sementblander	2021-12-06	
12	24	Andre Jensen	Tilhenger	2021-12-06	
12	37	Andre Jensen	Batteridrill	2021-12-06	
13	11	Kari Normann	Batteridrill	2021-11-06	2021-11-08

- a) (5%) Slik tabellen fremstår, hvilke svakheter og begrensninger har denne tabellen?
- i) Det er redundans i tabellen med redundant informasjon øker muligheten å for feil og vi kan få motstridene informasjon ved oppdatering (inkonsistent).¹
Dessuten bruker vi unødvendig lagring plass.²
Tabellen kan sette på normalform og vil bruke marginalt mere tid i søkeprosessen, men det er insignifikant i forhold til de problematiske søk slik som tabellen nå er.
- b) (10%) Normaliser tabellen til 3. normalform. Gjør rede for egne antagelser om dataene der du trenger det. Du kan velge å introdusere nye kolonner om ønskelig.



C) (5%) Begrunn hvorfor løsningen din oppfyller kravene til 3. normalform.

Det fin ingen transitivet eller partielle avhengigheter i tabellene i figur 3 b) som vis ovenfor og tabellene måter derfor kravene til 3NF. Som vist har jeg laget 3 tabeller der kunde nummer og utstyr nummer er PK og FK. Jeg opprettet to tabeller siden det var to primærnøkler i tabellen (KundeNr og UtstyrID).

Jeg tok utgangspunkt i at det ikke finnes fler enn ett utstyr av hver UtstyrID. Det kan derfor ikke lånes ut mer enn ett UtstyrID av gangen.

PK er merket med bold skrift, FK er merket med understrek.

Kilder:

- 1) Database-systemer 5. utgaver Bjørn Kristoffersen side 48
- 2) Database-systemer 5. utgaver Bjørn Kristoffersen side 94
- 3) Database-systemer 5. utgaver Bjørn Kristoffersen side 242
- 4) https://kristiania.instructure.com/courses/6545/files/722662?module_item_id=256268