

Emnekode: DB1102

Emnenavn: Databaser

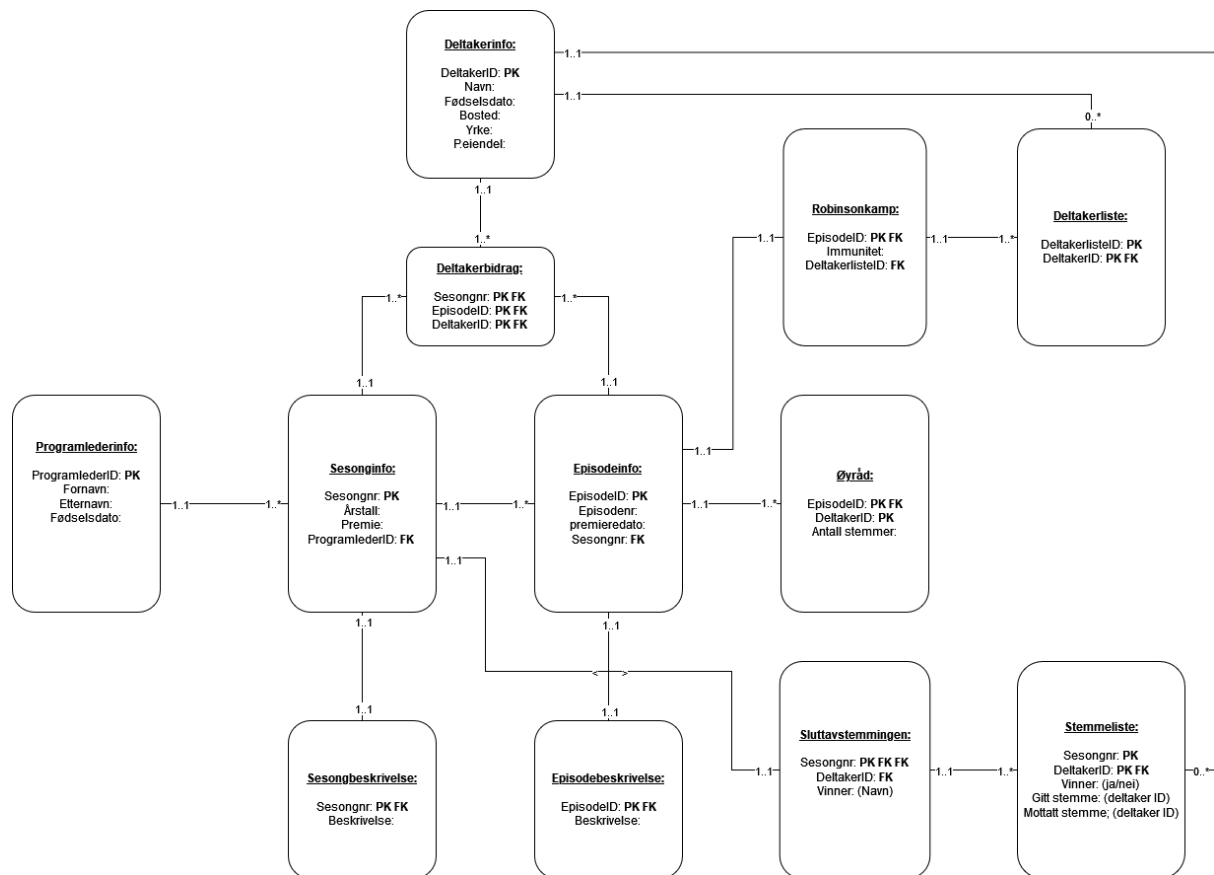
Innleveringsdato: 8/12/2021

Individuell hjemmeeksamen

Høgskolen Kristiania

Semester for høsten 2021

## Oppgave 1 – modellering



## Kommentar:

Jeg har aldri sett TV programmet Robinsonekspedisjonen så jeg vet ikke hvordan; Robinsonkamp, Øyråd og/eller finalestemming fungerer.

Jeg har tolket oppgaven slik at Robinsonkamp gjøres mellom to eller flere enkle individer og ikke alle deltagere ikke kommer å gjøre en Robinsonkamp.

Øyråd er en egen ting fra slutt avstemmingen.

Finalestemming gjøres av alle som er på slutten, men ikke alle som var med på starten.

Håper dette klare opp hvorfor jeg har skrevet mellom Deltakerinfo – Deltakerliste har 1..1 – 0..\* og Deltakerinfo – Stemmeliste har 1..1 – 0..\*.


## Oppgave 2 – SQL

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

```
SELECT navn, MAX(innbyggertall) AS innbyggertall
```

```
FROM kommune
```

```
ORDER BY innbyggertall DESC;
```



The screenshot shows a SQL query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The result table has two columns: 'navn' and 'innbyggertall'. The first row is 'Oslo' with a population of 697010.

	navn	innbyggertall
▶	Oslo	697010

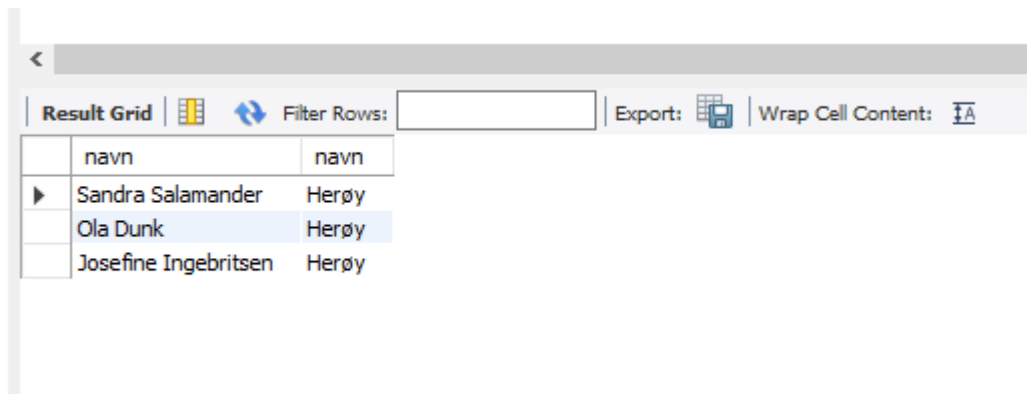
b) Lag en spørring som gir informasjon om hvilke registrerte spillere som bor i en kommune som heter Herøy.

```
SELECT s.navn, k.navn
```

```
FROM spiller AS s LEFT JOIN kommune AS k
```

```
ON s.kommuneid = k.kommuneid
```

```
where k.navn = "Herøy";
```



The screenshot shows a SQL query result grid. The toolbar includes 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The result table has two columns: 's.navn' and 'k.navn'. The first three rows are 'Sandra Salamander', 'Ola Dunk', and 'Josefine Ingebritsen', all from 'Herøy'.

	s.navn	k.navn
▶	Sandra Salamander	Herøy
	Ola Dunk	Herøy
	Josefine Ingebritsen	Herøy

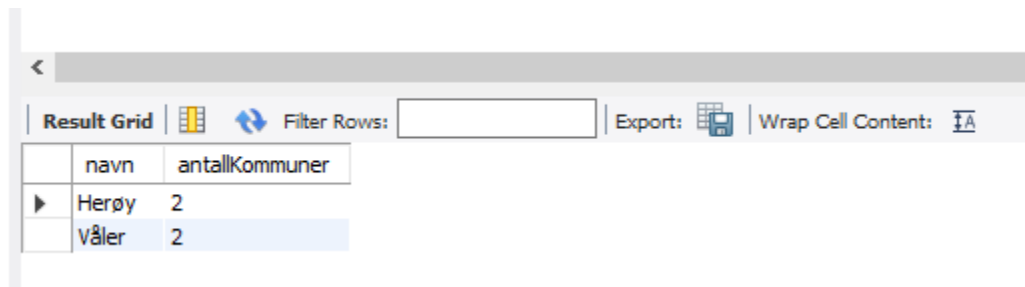
c) Lag en spørring som gir navn på alle registrerte kommuner som har samme navn, men ligger i forskjellige fylker.

```
SELECT navn, COUNT(navn) AS antallKommuner
```

```
FROM kommune
```

```
GROUP BY navn
```

```
HAVING COUNT(fylke) > 1;
```



	navn	antallKommuner
▶	Herøy	2
	Våler	2

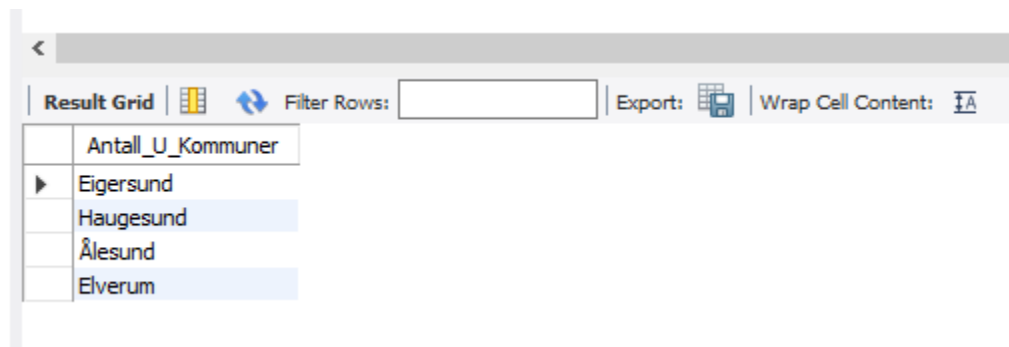
d) Lag en spørring som viser hvor mange registrerte kommuner som har en liten 'u' i navnet sitt. Navngi kolonnen i svaret: Antall U-Kommuner.

Kommentar: Får ikke kjørt spørring med ønsket navn, nytt navn blir Antall\_U\_Kommuner.

```
SELECT navn AS Antall_U_Kommuner
```

```
FROM kommune
```

```
WHERE navn LIKE "%u%";
```



	Antall_U_Kommuner
▶	Eigersund
	Haugesund
	Ålesund
	Elverum

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 k.navn, k.kommuneid, count(v.spillernr) AS totalVinn
```

```
FROM kommune AS k LEFT JOIN spiller as s
```

```
ON k.kommuneid = s.kommuneid
```

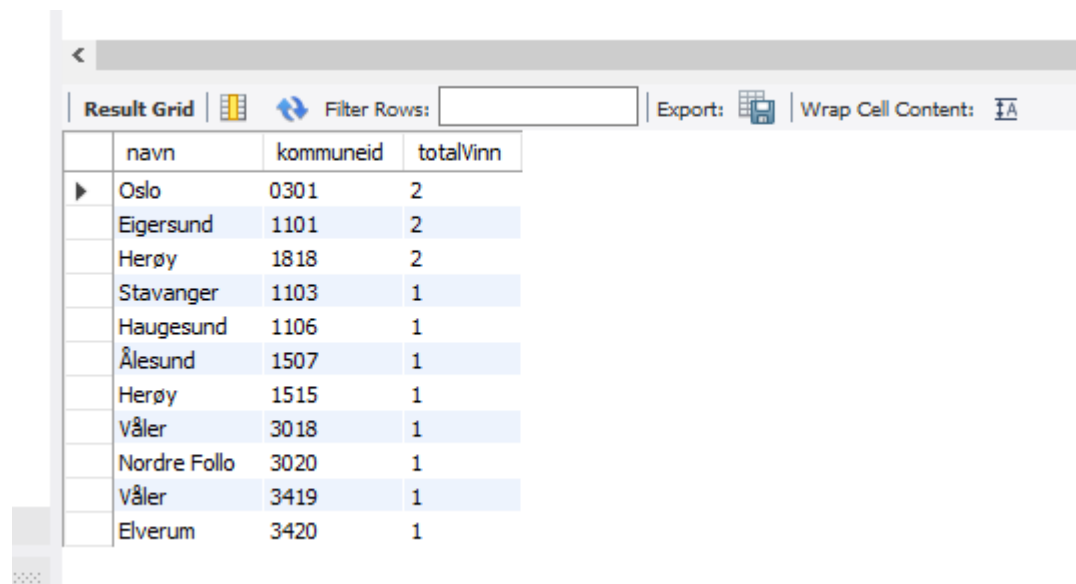
```
LEFT JOIN vinner AS v
```

```
ON s.spillernr = v.spillernr
```

```
GROUP BY k.kommuneid
```

```
HAVING MAX(v.spillernr)
```

```
ORDER BY totalVinn DESC;
```



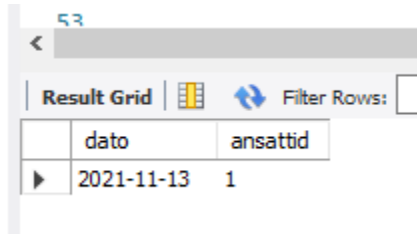
	navn	kommuneid	totalVinn
►	Oslo	0301	2
	Eigersund	1101	2
	Herøy	1818	2
	Stavanger	1103	1
	Haugesund	1106	1
	Ålesund	1507	1
	Herøy	1515	1
	Våler	3018	1
	Nordre Follo	3020	1
	Våler	3419	1
	Elverum	3420	1

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 dato, ansattid

FROM trekning

WHERE utbetaling = 0;



	dato	ansattid
▶	2021-11-13	1

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.

Kommentar: Uklart beskrevet, men mener du "...spillere som har vunnet flere..."?

Jeg kommer å løse oppgaven hvor det er spillere som har vunnet mer en kun en gang.

SELECT s.navn, COUNT(v.spillernr) AS antallVinn, k.navn AS kommune

FROM spiller as s LEFT JOIN vinner AS v

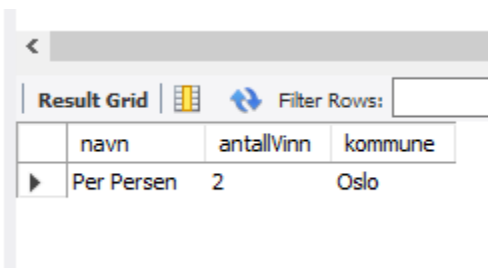
ON s.spillernr = v.spillernr

LEFT JOIN kommune AS k

ON s.kommuneid = k.kommuneid

GROUP BY s.navn

HAVING COUNT(v.spillernr) > 1;



	navn	antallVinn	kommune
▶	Per Persen	2	Oslo

/\*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 kommune
```

```
ADD km2 int;
```

```
UPDATE kommune SET km2 = 454 WHERE navn = "oslo";
```

```
UPDATE kommune SET km2 = 433 WHERE navn = "Eigersund";
```

```
UPDATE kommune SET km2 = 71 WHERE navn = "Stavanger";
```

```
UPDATE kommune SET km2 = 73 WHERE navn = "Haugesund";
```

```
UPDATE kommune SET km2 = 632 WHERE navn = "Ålesund";
```

```
UPDATE kommune SET km2 = 120 WHERE navn = "Herøy";
```

```
UPDATE kommune SET km2 = 64 WHERE navn = "Herøy";
```

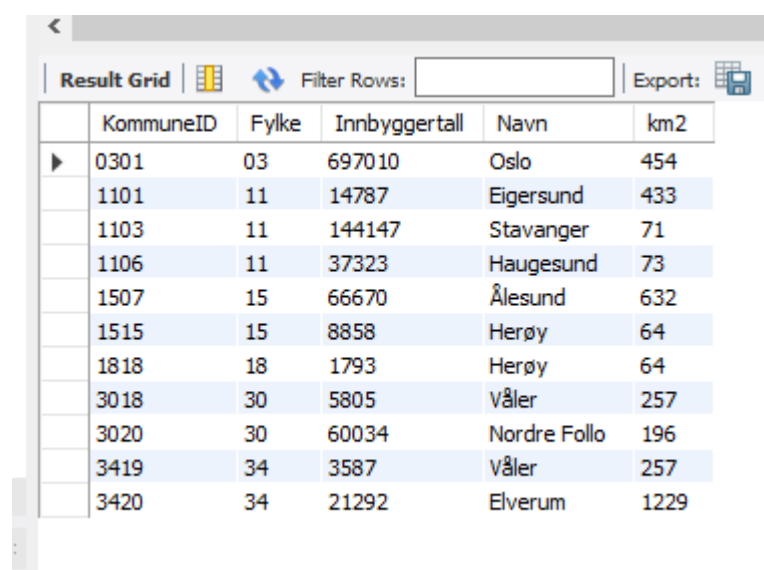
```
UPDATE kommune SET km2 = 705 WHERE navn = "Våler";
```

```
UPDATE kommune SET km2 = 196 WHERE navn = "Nordre Follo";
```

```
UPDATE kommune SET km2 = 257 WHERE navn = "Våler";
```

```
UPDATE kommune SET km2 = 1229 WHERE navn = "Elverum";
```

```
SELECT * FROM kommune;
```



	KommuneID	Fylke	Innbyggertall	Navn	km2
▶	0301	03	697010	Oslo	454
	1101	11	14787	Eigersund	433
	1103	11	144147	Stavanger	71
	1106	11	37323	Haugesund	73
	1507	15	66670	Ålesund	632
	1515	15	8858	Herøy	64
	1818	18	1793	Herøy	64
	3018	30	5805	Våler	257
	3020	30	60034	Nordre Follo	196
	3419	34	3587	Våler	257
	3420	34	21292	Elverum	1229

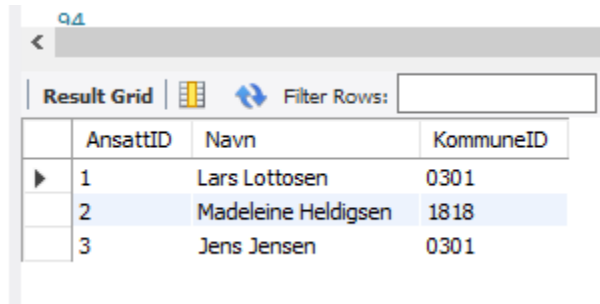
i) Det har vært en ny trekning. Legg inn følgende informasjon i databasen: Trekningen ble avholdt 4. desember 2021. Det var nøyaktig 11 millioner i utbetaling.

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).

```
INSERT INTO ansatt (ansattid, navn, kommuneID)
```

```
VALUE (3, "Jens Jensen", "0301");
```

```
SELECT * FROM ansatt;
```

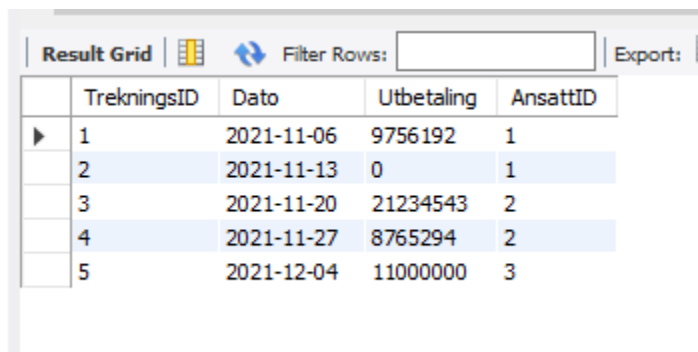


	AnsattID	Navn	KommuneID
▶	1	Lars Lottosen	0301
	2	Madeleine Heldigsen	1818
	3	Jens Jensen	0301

```
INSERT INTO trekning (trekningsid, dato, utbetaling, ansattid)
```

```
VALUE (5, "2021-12-4", 11000000, 3);
```

```
SELECT * FROM trekning;
```



	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



```
INSERT INTO spiller (spillernr, navn, adresse, kommuneid)
```

```
VALUE (14, "Lars Andersen", "Lilliveien 56", "1507");
```

```
INSERT INTO vinner (spillernr, trekningsid)
```

```
VALUE (14 ,5);
```

```
INSERT INTO spiller (spillernr, navn, adresse, kommuneid)
```

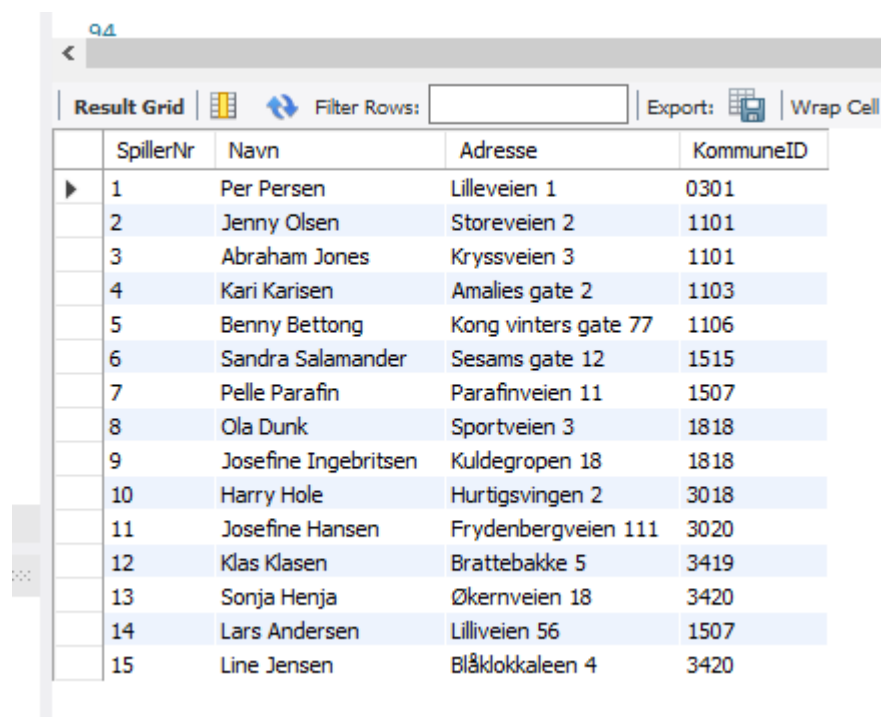
```
VALUE (15, "Line Jensen", "Blåklukkaleen 4", "3420");
```

```
INSERT INTO vinner (spillernr, trekningsid)
```

```
VALUE (15 ,5);
```

```
SELECT * FROM spiller;
```

```
SELECT * FROM vinner;
```





The screenshot shows a database application interface. At the top, there is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell'. Below the toolbar is a table with 5 columns: 'SpillerNr', 'Navn', 'Adresse', and 'KommuneID'. The table contains 15 rows of data, numbered 1 to 15. The last two rows (14 and 15) correspond to the data inserted in the SQL statements above.

	SpillerNr	Navn	Adresse	KommuneID
▶	1	Per Persen	Lilleveien 1	0301
	2	Jenny Olsen	Storeveien 2	1101
	3	Abraham Jones	Kryssveien 3	1101
	4	Kari Karisen	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	Hurtigsvingen 2	3018
	11	Josefine Hansen	Frydenbergveien 111	3020
	12	Klas Klasen	Brattekakke 5	3419
	13	Sonja Henja	Økernveien 18	3420
	14	Lars Andersen	Lilliveien 56	1507
	15	Line Jensen	Blåklukkaleen 4	3420

9.4

<

Result Grid   Filter Rows:

	SpillerNr	TrekningsID
▶	1	1
	2	1
	3	1
	4	1
	5	3
	6	3
	7	3
	8	3
	1	4
	9	4
	10	4
	11	4
	12	4
	13	4
	14	5
	15	5

## Oppgave 3 – Normalisering

a) Slik tabellen fremstår, hvilke svakheter og begrensninger har denne tabellen?

Tabellen sin svakhet og begrensinger er;

- Den mikser utstyr og kundenr som PK, som gjør at man får mye repeterende data.
- Hvis samme kunde leier samme produkt til et senere tidspunkt vill ikke PK være unik.

b) 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.

Leiekvittering	UtstyrID	Utlån	Innlevert	Kundenr
1	37	12/01/2021	03/01/2021	8
2	24	02/02/2021	03/02/2021	11
3	15	22/03/2021	22/04/2021	9
4	11	06/11/2021	06/11/2021	13
5	15	16/12/2021		12
6	24	16/12/2021		12
7	37	16/12/2021		12
UtstyrID	Utstyr		Kundenr	Kundenavn
11	Batteridrill		8	Morten Hansen
15	Sementblander		9	Lene Hansen
24	Tilhenger		11	Hans Hansen
37	Batteridrill		12	Andre Jensen
			13	Kari Normann

Kommentar: De kolonnene som er fetet er primær nøkkel.

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

I min nye tabell jeg har lagt inn i oppgave c) så skal den blitt overført til 3. normal form.

Dette kan begrunnes med at ingen av mine primær nøkler i alle tre tabeller har ingen transitiv (indirekte) avhengighet til primærnøkkelen min i forskjell til original tabellen.