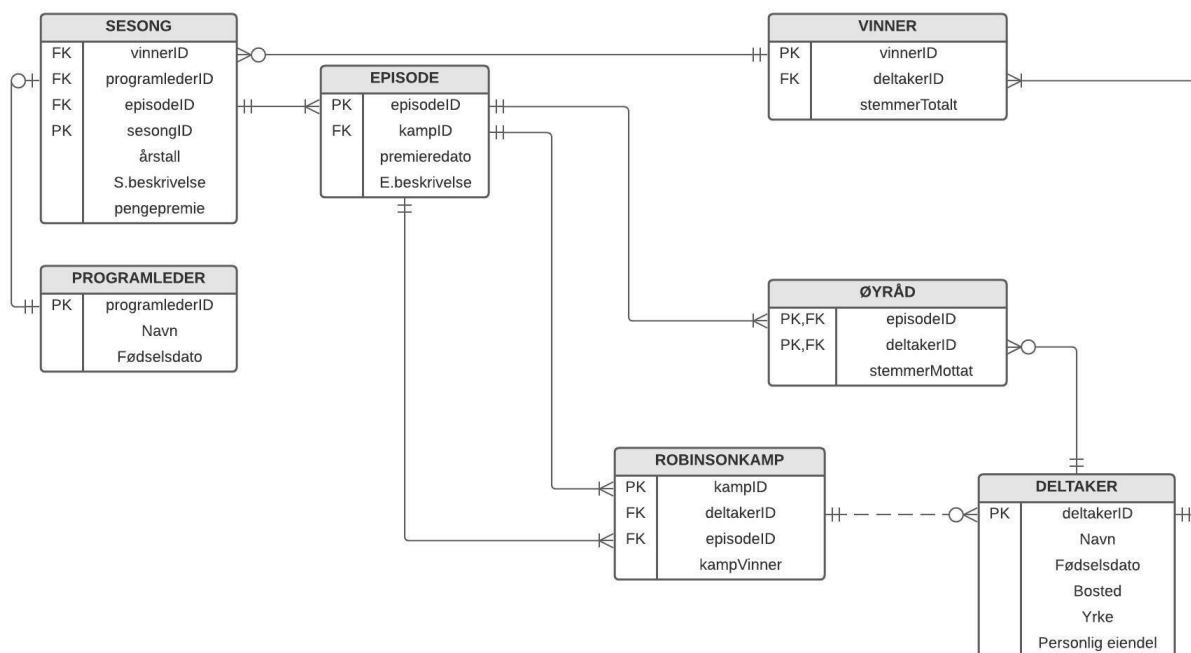


## Oppgave 1 - Modellering



Jeg tolker oppgaven 1 på følgende måte: etter min forståelse av oppgaven har jeg valgt å legge til syv entiteter og brukt kråkefot notasjon til å sette relasjoner mellom enhver entitet.

Vi starter ut i fra “sesong” hvor vi har en entitet for programleder for seg selv med en-til-en forhold. En programleder kan delta i en eller ingen sesong, og siden vi har blitt oppgitt 3 forskjellige programledere kan vi finne ut når de har deltatt etter sesonger.

Episode har to “hendelser” entiteter som er koplet på. Det er satt opp en kobling entitet mellom episode og deltaker for å unngå mange-til-mange forhold.

Robinson kamp har en attributt kamp vinner som jeg tenker at det var fornuftig å ha med slik at vi har oversikt hvem som vant kampen. Øyråd har en attributt som lagrer antall mottatte stemmer fra kandidater. Til slutt har vi en vinner entitet med kobling mot sesong og deltaker hvor attributten stemmer totalt mottatt definerer sesongens vinner.





C)

SELECT

f.navn AS Fylke, k.navn AS Kommune

FROM

fylke f

JOIN

kommune k ON f.FylkeID = k.Fylke;

```
38  -- /*
39      c) Lag en spørring som gir navn
40      på alle registrerte kommuner
41      som har samme navn,
42      men ligger i forskjellige fylker.
43  -- */
44  •  SELECT
45      f.navn AS Fylke, k.navn AS Kommune
46  FROM
47      fylke f
48      JOIN
49      kommune k ON f.FylkeID = k.Fylke;
50
51  -- Bruker en JOIN på fylke og kommune tabellen hvor de har fylkeID til felles
52
53
```

100% 79:51

Result Grid Filter Rows: Search Export:

Fylke	Kommune
Innlandet	Våler
Innlandet	Elverum
Møre og Romsdal	Ålesund
Møre og Romsdal	Herøy
Nordland	Herøy
Oslo	Oslo
Rogaland	Eigersund
Rogaland	Stavanger
Rogaland	Haugesund
Viken	Våler
Viken	Nordre Follo

**D)**

SELECT

Navn AS 'Antall U-Kommuner'

FROM

kommune

WHERE

Navn LIKE '%u%';

```
47
48  /*
49      d) Lag en spørring som viser
50      hvor mange registrerte kommuner
51      som har en liten 'u' i navnet sitt.
52      Navngi kolonnen i svaret: Antall U-Kommuner.
53  */
```

```
54 • SELECT Navn AS 'Antall U-Kommuner'
55     FROM kommune
56     WHERE Navn LIKE '%u%';
```

57

58 -- Brukte wildcard % forran og bak for å finne navnene som inneholdt "u" |

100% 75:58

Result Grid



Filter Rows:

Q Search

Export:



Antall U-Kommuner

Eigersund

Haugesund

Ålesund

Elverum

**E)**

```
SELECT count(k.fylke) antall, f.navn Fylke
FROM kommune k
JOIN fylke f
ON k.fylke = f.fylkeid
group by fylke;
```

**F)**

```
SELECT DISTINCT a.navn AS Trekkansvarlig, t.dato AS DatoTrek, t.utbetaling AS Premie,  
t.trekningsid AS TrekningsID  
FROM   ansatt a  
JOIN   trekning t  
ON     a.ansattid = t.ansattid  
LEFT JOIN vinner v  
ON     t.trekningsid = v.trekningsidw  
WHERE  t.utbetaling = 0;
```

```
93  /*  
94      f) Lag en spørring som viser hvilke treknings  
95      som ikke har hatt noen vinnere.  
96      Resultatet skal vise trekningsens dato,  
97      og navnet på hvem som var trekningsansvarlig.  
98  */  
99  
100  SELECT DISTINCT a.navn AS Trekkansvarlig, t.dato AS DatoTrek, t.utbetaling AS Premie, t.trekningsid AS TrekningsID  
101  FROM   ansatt a  
102  JOIN   trekning t  
103  ON     a.ansattid = t.ansattid  
104  LEFT JOIN vinner v  
105  ON     t.trekningsid = v.trekningsid  
106  WHERE  t.utbetaling = 0;  
107  
108  -- brukte to joins hvor den første kobler ansatt og trekning det viser hvem som trakk for den dagen  
109  -- brukte en left join for beholde alle data  
110
```

100% 45:109 1 error found

Result Grid Filter Rows: Search Export:

Trekkansvarlig	DatoTrek	Premie	trekningsID
Lars Lottosen	2021-11-13	0	2

g)

```
SELECT k.navn Kommune, s.navn Navn, v.spillernr SpillerNr, count(v.trekningsid) as  
GangerVunnet  
FROM vinner v  
RIGHT JOIN spiller s  
ON v.spillerNr = s.spillerNr  
LEFT JOIN kommune k  
ON s.kommuneId = k.kommuneId  
GROUP BY spillernr  
HAVING gangervunnet > 1;
```

```
112  
113  -- /*  
114      g) Lag en spørring som viser  
115      navn på spillere har vunnet flere enn en gang,  
116      hvor mange ganger de har vunnet,  
117      og hvilken kommune de bor i.  
118  -- */  
119  • SELECT k.navn Kommune, s.navn Navn, v.spillernr SpillerNr, count(v.trekningsid) as GangerVunnet  
120     FROM vinner v  
121     RIGHT JOIN spiller s  
122     ON v.spillerNr = s.spillerNr  
123     LEFT JOIN kommune k  
124     ON s.kommuneId = k.kommuneId  
125     GROUP BY spillernr  
126     HAVING gangervunnet > 1;  
127
```

100% 29:122 1 error found

Result Grid Filter Rows: Search Export:

Kommune	Navn	SpillerNr	GangerVunnet
Oslo	Per Persen	1	2



**H)**

ALTER TABLE kommune

ADD Areal DECIMAL(10,2) DEFAULT NULL;

UPDATE kommune SET Areal = 454 WHERE KommuneID = 0301;

UPDATE kommune SET Areal = 435.5 WHERE KommuneID = 1101;

UPDATE kommune SET AREAL = 71.35 WHERE KommuneID = 1103;

UPDATE kommune SET AREAL = 72.68 WHERE KommuneID = 1106;

UPDATE kommune SET AREAL = 632.4 WHERE KommuneID = 1507;

UPDATE kommune SET AREAL = 119.5 WHERE KommuneID = 1515;

UPDATE kommune SET AREAL = 64.4 WHERE KommuneID = 1818;

UPDATE kommune SET AREAL = 257 WHERE KommuneID = 3018;

UPDATE kommune SET AREAL = 203 WHERE KommuneID = 3020;

UPDATE kommune SET AREAL = 257 WHERE KommuneID = 3419;

UPDATE kommune SET AREAL = 1229 WHERE KommuneID = 3420;

```

133  /*
134  h) Legg inn en ny kolonne Areal i kommunetabellen. Legg inn fornuftige verdier i den nye kolonnen for de eksisterende kommunene.
135  Velg datatype du selv mener er passende. Arealet skal oppgis i antall kvadratkilometer, med to desimaler.
136  */
137  ALTER TABLE kommune
138  ADD Areal DECIMAL(10,2) DEFAULT NULL;
139  UPDATE kommune SET Areal = 454 WHERE KommuneID = 0301;
140  UPDATE kommune SET Areal = 435.5 WHERE KommuneID = 1101;
141  UPDATE kommune SET AREAL = 71.35 WHERE KommuneID = 1103;
142  UPDATE kommune SET AREAL = 72.68 WHERE KommuneID = 1106;
143  UPDATE kommune SET AREAL = 632.4 WHERE KommuneID = 1507;
144  UPDATE kommune SET AREAL = 119.5 WHERE KommuneID = 1515;
145  UPDATE kommune SET AREAL = 64.4 WHERE KommuneID = 1818;
146  UPDATE kommune SET AREAL = 257 WHERE KommuneID = 3018;
147  UPDATE kommune SET AREAL = 203 WHERE KommuneID = 3020;
148  UPDATE kommune SET AREAL = 257 WHERE KommuneID = 3419;
149  UPDATE kommune SET AREAL = 1229 WHERE KommuneID = 3420;
150
151  -- https://no.wikipedia.org/wiki/Fylkesnummer ---
152  SELECT * FROM KOMMUNE;

```

75% 57:142 1 error found

Result Grid Filter Rows: Search Edit: Export/Import:

KommuneID	Fylke	Innbyggertall	Navn	Areal
0301	03	697010	Oslo	454.00
1101	11	14787	Eigersund	435.50
1103	11	144147	Stavanger	71.35
1106	11	37323	Haugesund	72.68
1507	15	66670	Ålesund	632.40
1515	15	8858	Herøy	119.50
1818	18	1793	Herøy	64.40
3018	30	5805	Våler	257.00
3020	30	60034	Nordre Follo	203.00
3419	34	3587	Våler	257.00
3420	34	21292	Elverum	1229.00
NULL	NULL	NULL	NULL	NULL

Kilde: <https://no.wikipedia.org/wiki/Fylkesnummer>

**D)**

INSERT INTO ANSATT

(AnsattID, Navn, KommuneID) VALUES

(3,'Jens Jensen', '0301');

INSERT INTO trekning

(

TrekningID,

Dato,

Utbetaling,

AnsattID

) VALUES

(

5,

'2021-12-04',

11000000,

3

);

	TrekningID	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

**Oppgave 3 - Normalisering****A)**

<b>Kundenr</b>	<b>UtstyrId</b>	<b>Kundenavn</b>	<b>Utstyr</b>	<b>Utlånt</b>	<b>Innlevert</b>
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

Tabellene fremstår som normalisert i andre form med en del data redundans. Tabellen har også to primærnøkler som blir en sammensatt nøkkel. Dette vil danne delvis avhengighet hvor attributter som kundenavn, utstyr, utlånt og innlevert er transitiv avhengig av nøkkelen. Vi tar i utgangspunktet der kundeNr er determinant for kundenavn, utstyrId er determinant av utstyr, men utlånt og innlevert er funksjonell avhengig av utstyrId, og er indirekte avhengig(transitiv) av kundenr. Prosessen med normalisering er å unngå transitiv avhengighet. Vi får også problemer underveis når vi skal legge til, oppdatere, eller slette data fra tabellen.

**B)**

kundeNr	utstyrId	utlånID
8	37	1
9	15	2
11	24	3
12	15	4
12	24	5
12	37	6
13	11	7

utlånID	utlånt	innlevert
1	2021-01-12	2021-01-13
2	2021-03-22	2021-04-22
3	2021-02-02	2021-02-03
4	2021-12-06	
5	2021-12-06	
6	2021-12-06	
7	2021-11-06	2021-11-08

kundeNr	kundeNavn
8	Morten Hansen
9	Lene Jenssen
11	Hans Hansen
12	Andre Jensen
12	Andre Jensen
12	Andre Jensen
13	Kari Normann

utstyrId	utstyr
37	Batteridrill
15	Sementblander
24	Tilhenger
15	Sementblander
24	Tilhenger
37	Batteridrill
11	Batteridrill

Får å få tabellen vår til tredje normalform må vi gjøre den først til andre normalform. Her har vi lagt til en ny nøkkel for utlåns data med utlånId som primærnøkkel. I leier tabellen ser vi også utlånId er attributt som er funksjonell avhengig av både utstyrId og kundeNr. Kundenr har kun navn som ikke er avhengig av andre primærnøkkel. UtstyrId har kun utstyr med minimalt andre attributter og ikke avhengig av andre primærnøkkel. Når betingelser på andre normalform oppfylles er du på tredje normalform.

c)

Jeg vil tolke at tabellen er på tredje normalform hvor hver av entitet har en attributt(er) som er funksjonell avhengig av primærnøkkel sin og ikke transitiv avhengig av andre attributter. Vi kan også oppdatere data for tabellen uten å påvirke andre tabeller.