

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.

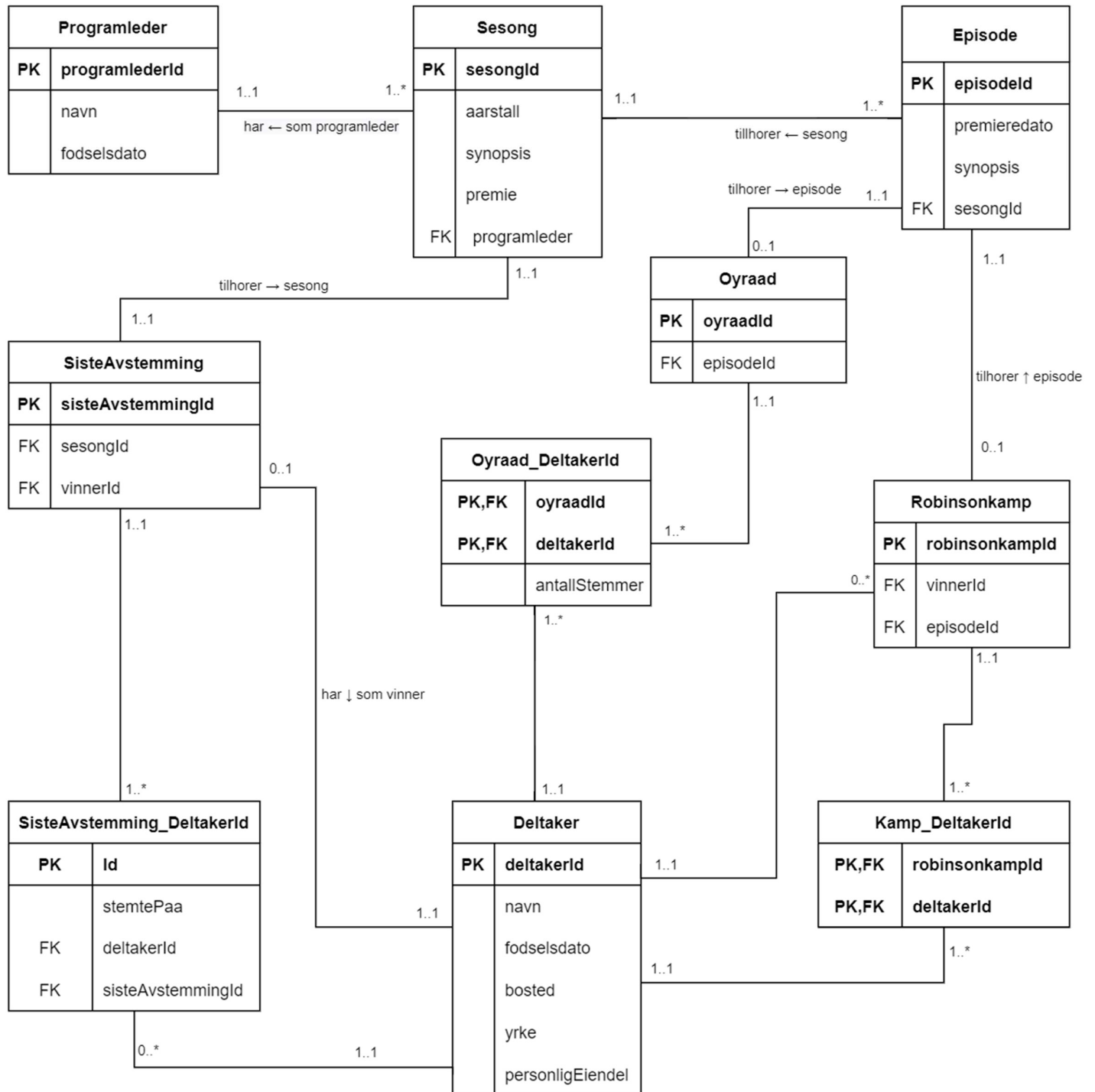
Oppgave: Tegn en modell for din foreslåtte løsning. Du kan selv velge om du vil benytte kråkefot eller UML notasjon. Velger du kråkefot trenger du ikke skille mellom identifiserende og

ikkeidentifiserende 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.



- a) Jeg går ut ifra at det kun er en kamp, samt øyråd per episode. Videre har jeg valgt å skrive uten norske bokstaver.
- b) På SisteAvstemming_DeltakerId har jeg valgt å ikke bruke en sammensatt primærnøkkel, på den måten kan en deltaker stemme flere ganger. Videre bruker jeg 0..* ettersom jeg ser for meg at det er mulig en deltaker ikke får mulighet til å være med på siste avstemming.

Oppgave 2 – SQL (50%)

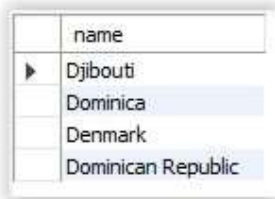
I eksamensoppgaven vil du finne et SQL-script (lotto.sql) som opprettet en database (lotto) og fyller tabeller med data. Kjør SQL-scriptet mot din egen MySQL-server. Deretter løser du oppgavene som beskrevet nedenfor. Svaret på hver deloppgave er todelt:

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%';
```



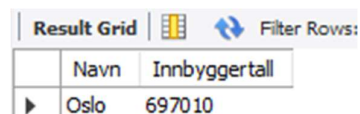
	name
▶	Djibouti
	Dominica
	Denmark
	Dominican Republic

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.

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

1.

```
select Navn, Innbyggertall as Innbyggertall  
from kommune  
where Innbyggertall = (select max(Innbyggertall) from kommune);
```



	Navn	Innbyggertall
▶	Oslo	697010

2.

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

```
select spiller.* , kommune.Navn as KommuneNavn  
from spiller  
inner join kommune ON kommune.KommuneID = spiller.KommuneID  
where kommune.Navn = "Herøy";
```

	SpillerNr	Navn	Adresse	KommuneID	KommuneNavn
	6	Sandra Salamander	Sesams gate 12	1515	Herøy
	8	Ola Dunk	Sportveien 3	1818	Herøy
▶	9	Josefine Ingebritsen	Kuldegropen 18	1818	Herøy

1.

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

1.

```
select Navn
from kommune
group by Navn
having (count(Navn) > 1);
```

	Navn
►	Herøy
	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.**

1.

```
select sum(Navn like '%u%') as 'Antall U-Kommuner'
from kommune;
```

Result Grid	
	Antall U-Kommuner
►	4

2.

- 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.**

1.

```
select fylke.navn, count(fylke.navn) as antallVinnere
from vinner
join spiller on vinner.spillerNr = spiller.spillerNr
join kommune on spiller.kommuneID = kommune.kommuneID
join fylke on kommune.fylke = fylke.fylkeID
group by fylke.fylkeID
order by antallVinnere desc;
```

Result Grid		
	navn	antallVinnere
►	Rogaland	4
	Oslo	2
	Møre og Romsdal	2
	Nordland	2
	Viken	2
	Innlandet	2

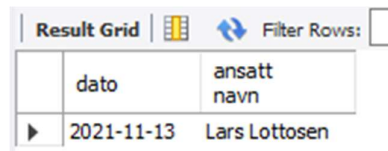
2.

Dette var det nærmeste jeg fikk, prøvde å få den til å bare vise Rogaland, men fikk bare feilmeldinger og merkelige svar.

- 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.**

1.

```
select dato, ansatt.navn as 'ansatt navn'
from trekning
left join vinner on trekning.TrekningsID = vinner.trekningsID
join ansatt on trekning.ansattID = ansatt.ansattID
where SpillerNr is null;
```



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid has two columns: 'dato' and 'ansatt navn'. The first row contains the values '2021-11-13' and 'Lars Lottosen'.

	dato	ansatt navn
▶	2021-11-13	Lars Lottosen

2.

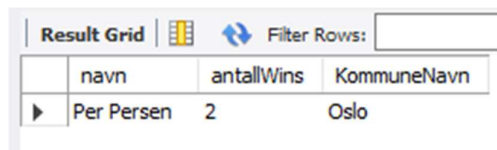
- g) **Lag en spørring som viser navn på spillere har vunnet flere enn en gang, hvor mange ganger de har vunnet, og hvilken kommune de bor i.**

1.

```
select spiller.navn, count(vinner.SpillerNr) as antallWins, kommune.navn as
KommuneNavn
from vinner
```

```
inner join spiller on vinner.SpillerNr = spiller.spillerNr
inner join kommune on kommune.kommuneID = spiller.kommuneID
```

```
group by vinner.spillerNr
having count(vinner.SpillerNr) > 1;
```



The screenshot shows a 'Result Grid' with a 'Filter Rows' button. The grid has three columns: 'navn', 'antallWins', and 'KommuneNavn'. The first row contains the values 'Per Persen', '2', and 'Oslo'.

	navn	antallWins	KommuneNavn
▶	Per Persen	2	Oslo

2.

- 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.**

1.

```
alter table kommune
add Areal decimal(6,2);
```

```
update kommune set areal = 454.09 where navn = "Oslo";
update kommune set areal = 432.48 where navn = "Eigersund";
update kommune set areal = 262.52 where navn = "Stavanger";
update kommune set areal = 72.68 where navn = "Haugesund";
```

```

update kommune set areal = 632.42 where navn = "Ålesund";
update kommune set areal = 119.52 where navn = "Herøy";
update kommune set areal = 256.96 where navn = "Våler";
update kommune set areal = 203.00 where navn = "Nordre Follo";
update kommune set areal = 1229.28 where navn = "Elverum";

```

```
select * from kommune;
```

	KommuneID	Fylke	Innbyggertall	Navn	Areal
▶	0301	03	697010	Oslo	454.09
	1101	11	14787	Eigersund	432.48
	1103	11	144147	Stavanger	262.52
	1106	11	37323	Haugesund	72.68
	1507	15	66670	Ålesund	632.42
	1515	15	8858	Herøy	119.52
	1818	18	1793	Herøy	119.52
	3018	30	5805	Våler	256.96
	3020	30	60034	Nordre Follo	203.00
	3419	34	3587	Våler	256.96
	3420	34	21292	Elverum	1229.28
*	NULL	NULL	NULL	NULL	NULL

2.

- 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).**

1.

```

insert into lotto.ansatt
values (3, 'Jens Jensen', '0301');

```

```

insert into lotto.trekning
values (5, '2021-12-4', 11000000, 3);

```

```

insert into lotto.spiller
values (14, 'Lars Andersen', 'Lilliveien 56', '1507'),
(15, 'Line Jensen', 'Blåklukkaleen 4', '3420');

```

```

insert into lotto.vinner
values ('14', '5'),
('15', '5');

```

2.

```
select * from ansatt;
```

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

```
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

```
select * from spiller;
```

	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	Brattebakke 5	3419
	13	Sonja Henja	Økernveien 18	3420
	14	Lars Andersen	Lilliveien 56	1507
	15	Line Jensen	Blåkløkkealeen 4	3420


```
select * from vinner;
```

	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

- 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.

1. Hvis vi går ut ifra at utbetaling i trekning tabellen allerede er fordelt på antall vinnere

```
create view fylkeSum as
select fylke.navn, sum(Trekning.utbetaling) as sum
from vinner
join trekning on vinner.TrekningsID = trekning.TrekningsID
join spiller on vinner.spillerNr = spiller.spillerNr
join kommune on spiller.kommuneID = kommune.kommuneID
join fylke on kommune.fylke = fylke.fylkeID
group by fylke.navn
order by sum desc;

select * from fylkeSum;
```

Result Grid		
	navn	sum
▶	Rogaland	50503119
	Møre og Romsdal	42469086
	Nordland	29999837
	Oslo	18521486
	Viken	17530588
	Innlandet	17530588

- 2.

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?
- Det er mye unødvendig bruk av 'string' variabler som tar mye mer plass enn int.
 - Mye repetisjon av navn.
 - Hvis en f.eks må endre på kundens navn, blir man nødt til å gjøre det flere ganger.
- 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.

Svar:

KundeNr (PK)	KundeNavn
8	Morten Hansen
9	Lene Jenssen
11	Hans Hansen
12	Andre Jensen
13	Kari Normann

UtstyrId (PK)	Utstyr
11	Batteridrill
15	Sementblander
24	Tilhenger
37	Batteridrill

LånId (PK)	KundeNr (FK)	UtstyrId (FK)	Utlånt	Innlevert
1	8	37	2021-01-12	2021-01-13
2	11	24	2021-02-02	2021-02-03
3	9	15	2021-03-22	2021-04-22
4	13	11	2021-11-06	2021-11-08

5	12	15	2021-12-06	
6	12	24	2021-12-06	
7	12	37	2021-12-06	

Jeg har her valgt å sortere lån etter utlånsdato og ikke rekkefølgen til den opprinnelige tabellen, som er sortert etter KundeNr.

Videre bruker jeg LånId istedenfor en sammensatt primærnøkkel ettersom jeg antar at en kunde kan låne samme utstyr flere ganger. (kan for så vidt bruke en supernøkkel istedenfor)

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

- a Ingen elementer er avhengige av noe annet enn sin primærnøkkel.