

Eksamen IN2090H21

Skriftlig hjemmeeksamen i IN2090

2021 HØST

Varighet: 15. desember 09:00 til 15. desember 13:00.

Det er viktig at du leser denne forsiden nøye før du starter.

Generell informasjon:

- Viktige beskjeder under eksamen blir gitt direkte fra faglærer på emnets semesterside eller i Canvas, om emnet ditt bruker det. Det er derfor viktig at du sjekker emnets semesterside/Canvas-rommet jevnlig.
- Besvarelsen din skal reflektere ditt eget, selvstendige arbeid og skal være et resultat av din egen læring og arbeidsinnsats.
- Alle hjelpemidler er tillatt ved skriftlig hjemmeeksamen. Dersom du gjengir tekst fra bøker, nettartikler eller lignende, så må det henvises til disse kildene i besvarelsen for å unngå mistanke om ulovlig tekstlikhet. Dette gjelder også dersom du oversetter tekst fra andre språk.
- Du er selv ansvarlig for å sørge for at eksamensbesvarelsen din ikke er tilgjengelig for andre under eksamenstiden, hverken fysisk eller digitalt.
- Husk at besvarelsen skal være anonym, du skal ikke oppgi hverken ditt eller medstudenters navn.
- Om du vil trekke deg fra eksamen, trykk på hamburgermeny oppe til høyre i Inspera og velg “Jeg vil trekke meg”.

Samarbeid under eksamen:

Det er ikke tillatt å samarbeide eller kommunisere med andre under eksamen. Samarbeid og kommunikasjon vil bli betraktet som forsøk på fusk. Det blir gjort plagiatskontroll av alle innleverte eksamener der tekstlikhet mellom besvarelser blir sjekket. Om du bruker notater som er utarbeidet i samarbeid med andre før eksamen, kan dette gi treff i en plagiatsjekk. Slik tekstlikhet kan bli betraktet som lav selvstendighet eller forsøk på fusk. Unngå derfor klipp/lim fra notater laget i samarbeid med andre.

Fusk:

Les om hva som regnes som fusk på UiOs nettsider.

Digital håndtegning/filopplasting:

Du har fått 30 min. ekstratid til opplasting av filer (f.eks. digital håndtegning). Sjekk hvordan du leverer inn digital håndtegning.

Kontaktinfo:

Brukerstøtte eksamen.

Digital oppklaringsrunde:

Oppklaringsrunde/trøsterunde (hvor man kan stille spørsmål underveis i eksamen) vil foregå på Zoom, og starte én time etter at eksamen har startet. Dersom dere har spørsmål vil dere da gå inn på Zoom-møtet (lenke vil bli publisert på semestersiden før eksamen), og bli satt i venterom. Deretter vil jeg ta inn én og én etter tur. Etttersom det er mange studenter i dette kurset er det viktig at man er forberedt og har klar alle spørsmål når dere kommer inn i venterommet. Dere kan naturligvis fortsette å jobbe med oppgavene mens dere er i venterommet.

Struktur:

Eksamen består av 3 deler (maksimal poengsum i parentes):

- Modellering og realisering (30)
- SQL (45)
- Relasjonsmodellen, normalformer og dekomposisjon (25)

1 Modellering (30)

1.1 ER: Hoteller (15)

I denne oppgaven skal du lage en ER-modell for en database for et hotell som blant annet skal lagre informasjon om gjester, rom og bestillinger.

- Modellen kan enten lages med et egnet program på egen laptop (f.eks. Dia eller diagrams.net), eller via håndtegning på papir. Bruk dine foretrukne verktøy til dette, f.eks. bilde av papirark fra mobil. Send bildet fra mobilen til deg selv på mail. Åpne bildet fra mailen på din “eksamensmaskin” og lagre bildet på denne maskinen. Herfra kan du laste opp fila til eksamensoppgaven. Det er viktig at tegningene er leselige. Husk å spesifisere hvilke tegninger som tilhører hvilke oppgaver. Det er studentens ansvar å dobbeltsjekke at bildefil er lastet opp og er leselig. MN anbefaler at du tar vare på originaltegninger til sensur har falt. Du kan også bruke nettbrett/elektronisk penn til å levere håndtegning. Det vil bli gitt 30 min. ekstratid på eksamen som følge av dette.
- Du må gjerne dele modellen opp i deler (f.eks. over flere sider), men sørg for at det fremkommer tydelig hvilke deler av modellen som besvarer hvilke deler av oppgaven.

- Du må gjerne inkludere kommentarer i modellene. Dersom det er uklarheter eller tvetydigheter i oppgavebeskrivelsen, bruk sunn fornuft og skriv en kommentar til modellene dine om hvilke antagelser og tolkninger du eventuelt gjør.
- Tegn og skriv tydelig.

Modellen skal inneholde informasjonen gitt under.

- Hotellet har mange rom som systemet skal lagre følgende informasjon om: Et unikt romnummer, som består av etasjen rommet ligger på samt dørrnummeret (f.eks. kan et romnummer være 305 hvor 3 er etasjen og 05 er dørrnummeret). Hvert rom har også en type (f.eks. 'dobbelrom', 'enkeltrom', 'suite', osv.), og et antall senger. Antall senger kan utledes fra romtypen. Hvert rom har også en pris per natt knyttet til seg.
- Hotellet bruker adgangskort til rommene, og hvert rom kan ha mange adgangskort, og hvert kort fungerer på nøyaktig ett rom. Kortene har et unikt kortnummer og en kode. Hvert rom må ha minst ett adgangskort.
- Databasen skal også lagre informasjon om gjestene på hotellet. Om gjestene ønsker hotellet å lagre navn, samt en epostadresse og et telefonnummer. Både eposten og telefonnummeret er unikt for hver gjest. En gjest kan være tilknyttet organisasjoner, f.eks. firmaer eller offentlige organer. Hver organisasjon har et unikt navn og en rabatt (uttrykt ved et enkelt desimaltall, f.eks. 0.1) på hotellet. Hver gjest kan tilhøre flere organisasjoner og en organisasjon kan ha mange gjester som tilhører den.
- En annen sentral ting som systemet skal lagre informasjon om er bestillinger. En bestilling gjøres på et rom for et bestemt tidspunkt. Kombinasjonen av bestillingens tidspunkt og rommet bestillingen er på er unikt. Et rom kan derimot ha mange bestillinger, men kun dersom de har forskjellige tidspunkt. I tillegg ønsker vi å lagre hvorvidt bestillingen inkluderer frokost eller ikke. Bestillinger har også en pris, men denne kan utledes fra prisen på rommet og lengden på bestillingen.
- En bestilling er for minst én gjest, men kan være for mange gjester, og en gjest kan være med på mange bestillinger. I tillegg har hver bestilling nøyaktig én gjest som er bestillingens kontaktperson. Her ønsker vi også å lagre den foretrukne kontaktmetoden (f.eks. 'epost' eller 'telefon'). Merk at én gjest kan godt være kontaktperson på flere bestillinger.
- Til slutt ønsker vi å lagre informasjon om hvilke gjester som nå bor på hvilke rom, og adgangskortene de har fått. Disse dataene representerer kun informasjon som er sant i øyeblikket, og ikke historisk informasjon. Altså kan en gjest kun bo på ett rom, men hvert rom kan ha mange gjester (for eksempel kan jo to personer bo på et dobbeltrom samtidig). Hver gjest får utlevert høyst ett adgangskort, og et adgangskort kan høyst gis ut til én gjest.

Løsningsforslag

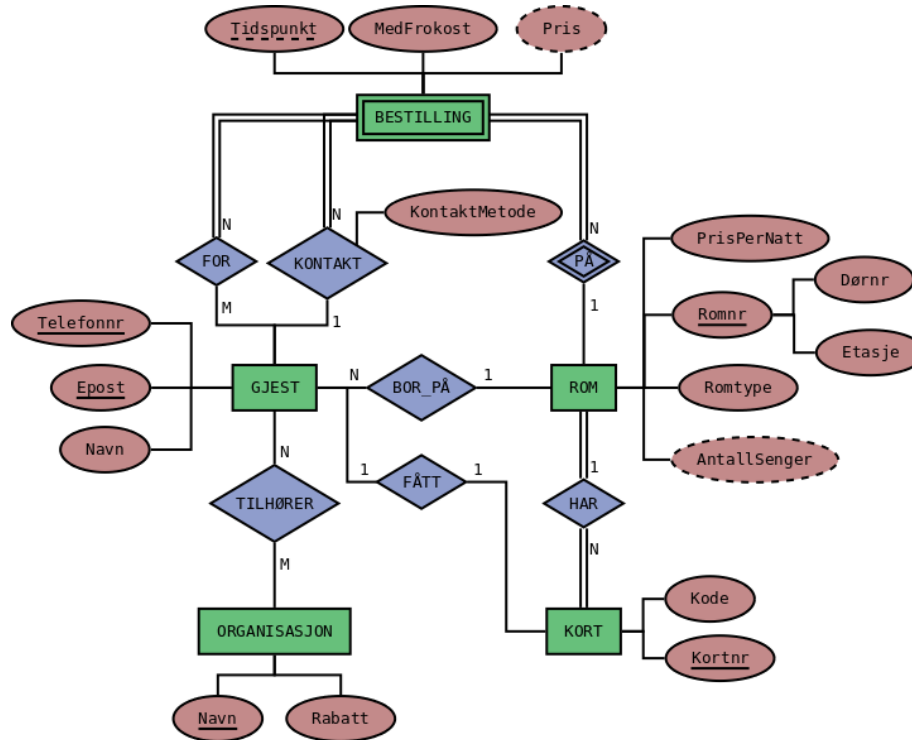


Figure 1: Hotell-modell

Sensorveiledning

- 1 poeng for GJEST, ORGANISASJON og KORT (med attributter)
- 2 poeng for ROM (med attributter)
- 3 poeng for BESTILLING (med attributter og identifiserende relasjon)
- 1 poeng for hver relasjon, men 2 poeng for KONTAKT med attributt (enten på relasjonen selv eller på bestillingen)
- -0.5 for feil plassering av total deltakelse eller riktig kardinalitet, men feil plassering
- -1 for feil kardinalitet eller manglende total deltakelse
- -2 for manglende kardinalitet på hele relasjonen
- -1 for manglende attributt-markeringer (nøkkel eller multi-valued)
- -1 for manglende/feil plassering av vanlig attributt

- -2 for både manglende markering av svak entitet og identifiserende relasjon (-1 for kun å mangle én av dem)

1.2 ER: Hus (5)

Hvilke av disse setningene er sanne i henhold til følgende ER-diagram. Kryss av “JA” dersom du mener setningen er modellert av ER-diagrammet, og “NEI” hvis ikke. Merk at modellen ikke trenger å stemme overens med din intuisjon om domenet.

Hvert riktig svar gir 1 poeng, mens hvert uriktige svar gir -1 poeng. Total poengsum for hele oppgaven vil ikke være lavere enn 0.

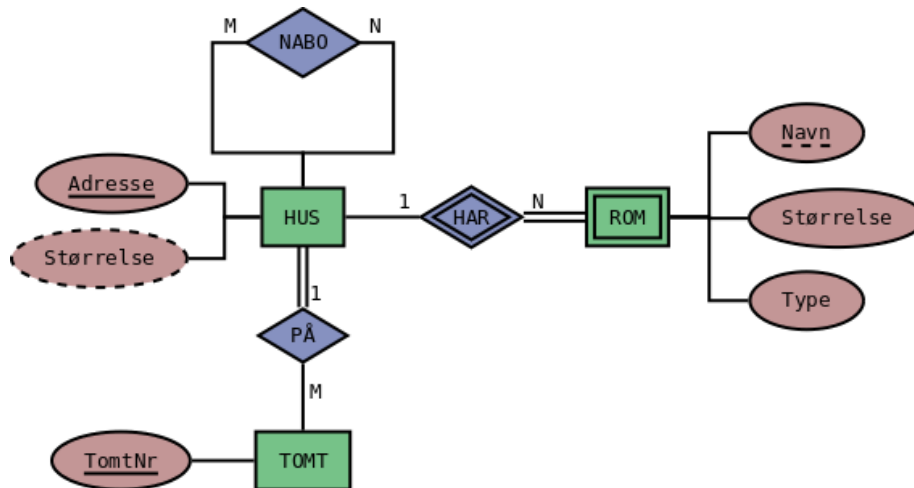


Figure 2: ER-modell

1. Et hus har minst ett rom.
2. Et hus kan være på mange tomter.
3. Hvert hus har minst én nabo.
4. Et hus kan ha mange rom med samme type.
5. Alle hus har en adresse.

Løsningsforslag

1. Nei
2. Ja
3. Nei
4. Ja
5. Ja

Automatisk rettet.

1.3 Realisering: Styrer (10)

Under er det en modell som beskriver styrer til land og kommuner (slik som et valgt Storting for Norge eller et bystyre for Oslo). Hvert styre må enten være et styre for en kommune, da et bystyre, eller for et land, da et parlament (men dette kan jo ikke uttrykkes i ER). Videre består hvert styre av politiske partier, hver med et gitt antall representanter. Styret velger en regjering (som også består av politiske partier). Merk at ett styre kan ende opp med å måtte velge flere regjeringer, dersom den første f.eks. velger å gå av, mister tillit, el. Tilsvarende kan en regjering ende opp med å sitte over flere styre dersom den blir gjenvalgt.

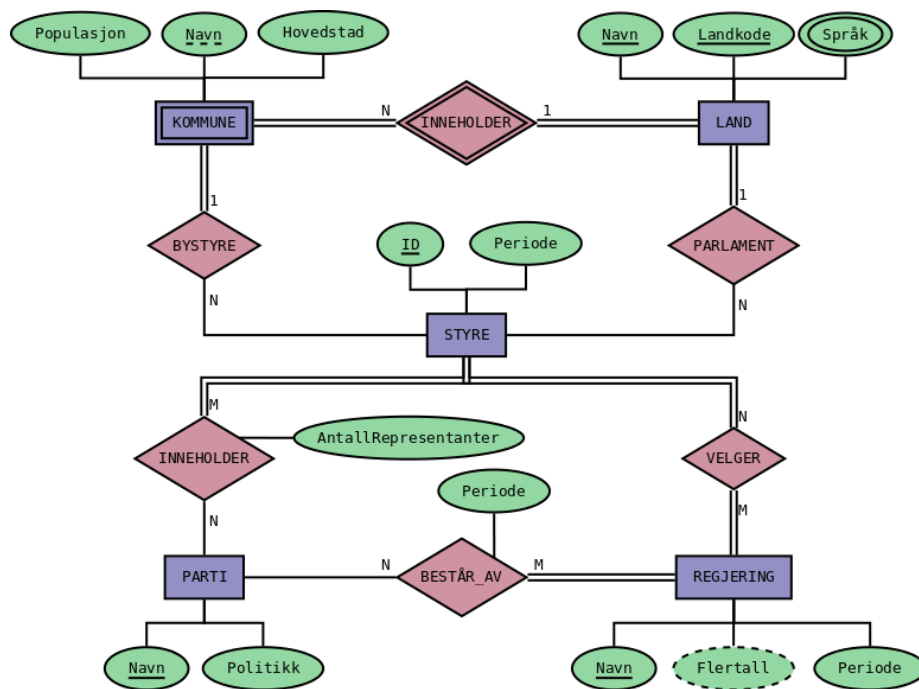


Figure 3: ER-modell over styrer

Realiser modellen til et relasjonelt databaseskjema. Det resulterende databaseskjemaet skal være korrekt (samsvare med modellen), effektivt (unngå overflødige tabeller og kolonner med mange NULL-verdier), og tydelig (bør være lett å forstå). Bruk algoritmen for realisering for å lage et slikt skjema, og forklar eventuelle valg du tar underveis. For hver relasjon, oppgi relasjonens navn, og navnet til hvert attributt. Du trenger ikke oppgi datatyper for attributtene, og du skal ikke bruke SQL i denne oppgaven. Bruk understreking for å markere kandidatnøkler. Dersom det er flere kandidatnøkler, bruk fet skrift for å markere valgt primærnøkkel. Bruk piler for å markere fremmednøkler (f.eks. "T(A) -> S(B)" for å uttrykke at relasjon T sin attributt A er en fremmednøkkel som peker på relasjon S sin B attributt).

Om du ønsker å ha bilde større eller i en egen tab, høyreklikk på bildet og velg “Se bildet/View Image”.

Løsningsforslag

- *Land*(landkode, *navn*)
- *Styre*(*id*, *periode*)
- *Parti*(navn, *politikk*)
- *Regjering*(navn, *periode*)
- *Kommune*(navn, *land*, *populasjon*, *hovedstad*)
- *Bystyre*(*by*, *land*, *styre*)
- *Parlament*(*land*, *styre*)
- *Inneholder*(*styre*, *parti*, *antallRepresentanter*)
- *Velger*(*styre*, *regjering*)
- *BestårAv*(*parti*, *regjering*, *periode*)
- *Språk*(*land*, *språk*)

hvor

- *Kommune*(*land*) → *Land*(*navn*)
- *Bystyre*(*by*, *land*) → *Kommune*(*navn*, *land*)
- *Bystyre*(*styre*) → *Styre*(*id*)
- *Parlament*(*land*) → *Land*(*navn*)
- *Parlament*(*styre*) → *Styre*(*id*)
- *Inneholder*(*styre*) → *Styre*(*id*)
- *Inneholder*(*parti*) → *Parti*(*navn*)
- *Velger*(*styre*) → *Styre*(*id*)
- *Velger*(*regjering*) → *Regjering*(*navn*)
- *BestårAv*(*styre*) → *Styre*(*id*)
- *BestårAv*(*parti*) → *Parti*(*navn*)
- *Språk*(*land*) → *Land*(*navn*)

Valg:

- *landkode* som primærnøkkel for *Land*
- lar *BYSTYRE* og *PARLAMENT* bli nye relasjoner, siden dersom vi representerer disse med fremmednøkler vil det alltid blir én NULL-verdi for hvert tuppel i *Styre*-relasjonen

Alternativ løsning: Kan eventuelt velge å realisere *BYSTYRE* og *PARLAMENT* inn i *Styre* ved å erstatte relasjonene *Bystyre* og *Parlament* med to ekstra fremmednøkler i *Styre*:

- *Styre*(*id*, *periode*, *bystyreby*, *bystyreland*, *parlament*)

hvor

- *Styre*(*bystyreby*, *bystyreland*) → *Kommune*(*navn*, *land*)
- *Styre*(*parlament*) → *Land*(*navn*)

Merk at for hvert tuppel i *Styre* vil enten *bystyreby* og *bystyreland* eller *parlament* være NULL.

Sensorveiledning

- 0.5 for hver vanlige entitet
- 1.5 poeng for den svake entiteten (inkludert den identifiserende relasjonen)
- 1 poeng for hver relasjon, og 1.5 for hver relasjon med attributt
- 0.5 for multi-valued attributten

Trekk:

- -0.5 for manglende attributt
- -0.5 for manglende markering av kandidatnøkler og primærnøkkel
- -0.5 for manglende fremmednøkkelmarkering
- -0.5 for manglende forklaring på valg (eller -2 om ingen forklaring på hele oppgaven)
- -0.5 for å realisere den identifiserende relasjonen dobbelt

2 SQL (45)

Oppgavene i denne delen er om SQL. Alle oppgavene vil bruke det samme databaseskjemaet, og en beskrivelse av dette er vedlagt som PDF for hver oppgave.

Du må gjerne inkludere kommentarer i spørringene (gjøres med “–” i SQL). Dersom det er uklarheter eller tvetydigheter i oppgavebeskrivelsen, bruk sunn fornuft og skriv en kommentar om hvilke antagelser og tolkinger du gjør.

Sensorveiledning

Følgende beskjed om skrivefeil i databaseskjema ble lagt ut for studentene under eksamen:

Kolonnen “oppdaget” i tabellene stjerne, planet og måne skal ha type int (inneholder kun et årstall), ikke date slik det står i CREATE-setningene i PDFen. Eksempeldataene er derimot riktige da.

Generelt:

- -1 for **NATURAL JOIN** som joiner på mer enn tiltenkt
- -1 for manglende/feil tabeller i joins i **FROM**
- -0.5 for manglende kolonner i **SELECT**
- -1 for manglende begrensning i **WHERE**
- Ingen trekk for å bruke **GROUP BY** på navn (de er også markert som **UNIQUE**)
- Ingen trekk for å kun ha primærnøkkel i **GROUP BY** men andre attributter (fra samme tabell) i **SELECT** (støttes av PostgreSQL)

2.1 Flervalg (12)

Anta at databasen inneholder nøyaktig de dataene som er vist i eksempel-dataene i den vedlagte PDFen. For hver spørring under, hvor mange rader er det i svaret? Hvert riktig svar gir 2 poeng, mens hvert uriktige svar gir -0.5 poeng. Blankt svar/ubesvart gir 0 poeng. Laveste poengsum på hele oppgaven er 0.

1.

```
SELECT *
FROM planet
WHERE oppdaget = NULL;
```

2.

```
SELECT *
FROM planet AS p, måne AS m
WHERE p.oppdaget = 2017 AND
      m.oppdaget = 2019;
```

3.

```
SELECT oppdaget FROM planet
WHERE oppdaget >= 2017
UNION
SELECT oppdaget FROM måne
WHERE oppdaget >= 2017;
```

4.

```
SELECT sid, count(*)
FROM planet
GROUP BY sid
HAVING count(*) > 1;
```

5.

```
SELECT *
FROM observator NATURAL JOIN stjerne
WHERE type = 'satellitt';
```

6.

```
SELECT DISTINCT type
FROM observator
WHERE type = 'satellitt';
```

Løsningsforslag

Spørring	0	1	2	3	4
1	X				

Spørring	0	1	2	3	4
2			X		
3				X	
4			X		
5	X				
6		X			

Automatisk rettet.

2.2 Store stjerner (5)

Skriv en SQL-spørring som finner navn, masse og lysstyrke på alle stjerner som har masse større enn 200 eller lysstyrke større enn 10.

Løsningsforslag

```
SELECT navn, masse, lysstyrke
FROM stjerne
WHERE masse > 200 OR
      lysstyrke > 10;
```

2.3 Planeter i solsystemet (5)

Skriv en SQL-spørring som finner navn og masse på planeter som er i samme solsystem som planeten med navn 'Tellus'. Sorter resultatet etter masse fra minst til størst.

Løsningsforslag

```
SELECT p2.navn, p2.masse
FROM planet AS p1
      JOIN planet AS p2 USING (sid)
WHERE p1.navn = 'Tellus'
ORDER BY p2.masse;
```

Sensorveiledning

Følgende oppklarende beskjed ble lagt ut under eksamen:

To planeter er i samme solsystem om de går i bane rundt samme stjerne.

2.4 Pund til kilo (5)

Det viser seg at observatoriet med navn 'BESS' konsekvent har rapportert massen på alle sine observasjoner av stjerner gjort før '1990' i pund fremfor

kilo. Skriv derfor en SQL-kommando som oppdaterer massen til alle stjerner observert før dette året av dette observatoriet til kilogram. Merk at ett pund er 0.45 kilogram, altså blir ny masse lik 0.45 ganger tidligere masse.

Løsningsforslag

```
UPDATE stjerne
SET masse = 0.45 * masse
WHERE oppdaget < 1990 AND oid = (
    SELECT oid
    FROM observator
    WHERE navn = 'BESS'
);
```

2.5 Universet (8)

Skriv en SQL-kommando som lager et VIEW med navn `universet` som inneholder to kolonner: det totale antallet astronomiske objekter i universet (altså totalt antall stjerner, planeter og måner som er i databasen), samt den totale massen til universet (altså den totale massen til alle disse objektene i databasen).

Løsningsforslag

```
CREATE VIEW universet(antall, masse) AS
WITH
    alle_objekter AS ( -- Massen til alle objekter i universet
        SELECT masse FROM stjerne
        UNION ALL
        SELECT masse FROM planet
        UNION ALL
        SELECT masse FROM måne
    )
SELECT count(*) AS antall, sum(masse) AS masse
FROM alle_objekter;
```

2.6 Store solsystemer (10)

Et solsystem er en mengde astronomiske objekter som går i bane rundt en stjerne, altså alle planetene som går i bane rundt stjernen samt alle disse planetenes måner.

Skriv en SQL-spørring som finner navn på alle stjerner som er en del av et stort solsystem, hvor et stort solsystem er et solsystem med mer enn 10 planeter eller som har total masse (på både stjernen, planetene og månene til sammen) på mer enn 400.

Løsningsforslag

```
WITH
solsystemer AS (
  -- Knytter sid og stjernens navn til hvert
  -- objekts masse i stjernens solsystem
  SELECT sid, navn AS stjerne, masse
  FROM stjerne
  UNION ALL
  SELECT s.sid, s.navn AS stjerne, p.masse
  FROM stjerne AS s
       JOIN planet AS p USING (sid)
  UNION ALL
  SELECT s.sid, s.navn AS stjerne, m.masse
  FROM stjerne AS s
       JOIN planet AS p USING (sid)
       JOIN måne AS m USING (pid)
)
SELECT stjerne
FROM solsystemer
GROUP BY sid, stjerne
HAVING sum(masse) > 400
UNION ALL
SELECT s.navn AS stjerne
FROM stjerne AS s
     JOIN planet AS p USING (sid)
GROUP BY s.sid, s.navn
HAVING count(*) > 10
```

3 Relasjonsmodellen og normalformer (25)

3.1 Relasjonsalgebra (5)

Bruk samme skjema som for SQL-oppgavene over, og skriv en SQL-spørring som gir samme resultat (samme tupler/rader) som relasjonsalgebra-uttrykket under:

$$\pi_{navn} (observerator \bowtie_{oid=soid} \pi_{soid} (\rho_{oid \rightarrow soid} (\sigma_{lysstyrke > 50} (stjerne))))$$

Løsningsforslag

```
SELECT o.navn
FROM observator AS o
     JOIN stjerne AS s USING (oid)
WHERE s.lysstyrke > 50;
```

Alternativ og mer direkte:

```

SELECT navn
FROM observator
      JOIN (SELECT oid AS soid
            FROM stjerne
            WHERE lysstyrke > 50) AS t ON oid = soid;

```

3.2 Normalformer (10) (flervalg)

Gitt følgende relasjon:

$R(A, B, C, D, E, F, G)$

med kandidatnøkler $BCFG$ og $BDFG$.

For hver av FDene under, anta at FDen gjelder for relasjonen R over, og bruk algoritmen for å finne normalform til å avgjøre hvilken normalform FDen (alene) tilsier at R er på.

Hvert riktige svar gir 2 poeng, hvert uriktige svar gir -0.5 poeng, ubesvart gir 0 poeng, men total poengsum på hele oppgaven vil ikke bli lavere enn 0.

FD	1NF	2NF	3NF	BCNF
$BD \rightarrow C$				
$AC \rightarrow D$				
$ABC \rightarrow E$				
$DEF \rightarrow C$				
$CFG \rightarrow A$				

Løsningsforslag

FD	1NF	2NF	3NF	BCNF
$BD \rightarrow C$			X	
$AC \rightarrow D$			X	
$ABC \rightarrow E$		X		
$DEF \rightarrow C$			X	
$CFG \rightarrow A$	X			

Automatisk rettet.

3.3 Tapsfri dekomposisjon (10)

Gitt følgende relasjon:

$R(A, B, C, D, E, F)$

med kandidatnøkkel ABD , og følgende FDer:

1. $AB \rightarrow C$
2. $B \rightarrow E$
3. $D \rightarrow E$
4. $D \rightarrow F$

Dekomponer R tapsfritt til BCNF. Vis stegene du gjør og list opp nøkler og FDer på alle relasjonene underveis.

Løsningsforslag

$AB \rightarrow C$ bryter med BCNF, siden AB ikke er en supernøkkel. Finner tillukningen til AB :

$$AB^+ = ABCE$$

Dekomponerer R til $S_1(A, B, C, E)$ og $S_2(A, B, D, F)$.

For $S_1(A, B, C, E)$ gjelder FD 1 og 2, og har dermed eneste kandidatnøkkel AB . FD 1 bryter da ikke med BCNF siden AB er en supernøkkel, men FD 2 bryter med BCNF siden B ikke er en supernøkkel. Dekomponerer derfor med hensyn på denne:

$$B^+ = BE$$

Dekomponer S_1 til $S_{11}(B, E)$ og $S_{12}(A, B, C)$. For S_{11} gjelder bare FD 2, og er dermed på BCNF og har eneste kandidatnøkkel B . For S_{22} gjelder kun FD 1, og er dermed også på BCNF og har eneste kandidatnøkkel AB .

$S_2(A, B, D, F)$ har kun FD 4. Vi ser at $ABD^+ = ABDF$ og er dermed også (eneste) kandidatnøkkel for S_2 . Siden D ikke er en supernøkkel bryter dermed FDen med BCNF og vi dekomponerer med $D^+ = DF$: $S_{21}(A, B, D)$ og $S_{22}(D, F)$. For S_{21} gjelder ingen FDer og den er dermed på BCNF (med ABD som eneste kandidatnøkkel), mens for S_{22} holder kun FD 4, og får dermed eneste kandidatnøkkel D , og bryter dermed ikke med BCNF.

Altså kan R dekomponeres tapsfritt til $S_{11}(B, E)$, $S_{12}(A, B, C)$, $S_{21}(A, B, D)$ og $S_{22}(D, F)$.

Sensorveiledning

- 5 poeng for riktig første dekomponering (inkludert nøkler og FDer på de nye relasjonene)
- 2.5 poeng for hver av de to andre dekomponeringene (inkludert nøkler og FDer på de nye relasjonene)
- -1 poeng for manglende/feil FD på relasjon
- -1 poeng for feil utregning av tillukning
- -1 poeng for manglende/feil kandidatnøkler
- -3 poeng for ingen forklaringer