

Utsatt eksamen IN2090 Høsten 2022

Utsatt skriftlig eksamen i IN2090

2022 HØST

Tid: Tirsdag 17. januar 09:00-13:00

Sted: Silurveien

Tillatte hjelpemidler: Alle trykte og skrevne hjelpemidler tillatt.

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

Håndtegning:

På denne eksamen er det oppgaver som krever håndtegning. Du bruker skisseark du får utdelt. Det er anledning til å bruke flere ark per oppgave. Se instruksjon for utfylling av skisseark i lenken under oppgavelinjen.

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: Prosjekter (15)

Du er ansatt i et firma som har vokst mye, og trenger nå bedre oversikt over ansatte, prosjekter og ressurser. De ønsker derfor å lage en database over dette, og du er derfor blitt bedt om å lage en ER-modell for databasen. Etter diskusjon med ledelsen har dere kommet frem til at følgende vil gjelde:

- En ansatt er identifisert med en egen person-ID (eller Pid), og har ellers et navn og en mengde titler (slik som “MSc”, “Dr”, “Advokatfullmektig”, osv.).
- En ansatt kan jobbe på et prosjekt og har da en stilling knyttet til prosjektet (slik som “prosjektkoordinator”, “økonomiansvarlig”, osv.). En ansatt kan

være knyttet til mange prosjekter og et prosjekt kan ha mange ansatte. Alle prosjekter må derimot ha minst én ansatt.

- Et prosjekt har et unikt navn. Prosjekter går også over en periode som består av en startdato og en sluttdato. Prosjektet har derfor en varighet, men denne kan naturligvis utledes fra start- og sluttdatoen på perioden.
- Et prosjekt kan også ha en bestemt ansatt som leder. Et prosjekt kan høyst ha én leder og en ansatt kan kun være leder for maks ett prosjekt. En ansatt trenger derimot ikke være leder for et prosjekt og et prosjekt trenger heller ikke ha en leder.
- Et prosjekt kan ha en beskrivelse. En beskrivelse er unikt identifisert av en dokument-ID (eller Did) og har samme navn som prosjektet det er knyttet til (navnet på dokumentet kan altså utledes fra navnet på prosjektet det hører til). Beskrivelsen har også en mengde emneord knyttet til seg. En beskrivelse hører til nøyaktig ett prosjekt. Et prosjekt trenger ikke ha noen beskrivelse og kan høyst ha én beskrivelse.
- Et prosjekt kan deles inn i milepæler, hvor en milepæl har et navn som er unikt innenfor prosjektet den er knyttet til. En milepæl har også en frist for når den må være utført. Et prosjekt kan ha mange milepæler.
- En milepæl har også alltid en ansatt som er ansvarlig for dens gjennomføring. En milepæl kan ha flere ansvarlige ansatte, men en ansatt kan kun være ansvarlig for ett prosjekt, men trenger derimot ikke være ansvarlig for noe prosjekt.
- En ansatt kan bruke en ressurs for et prosjekt. En ressurs har et unikt navn og har ellers en kostnad knyttet til seg. Gitt en ansatt og et prosjekt kan de bruke mange ressurser; gitt et prosjekt og en ressurs kan det derimot kun være én ansatt; mens gitt en ressurs og en ansatt kan det være mange prosjekter. Hver ressurs må derimot brukes av minst én ansatt og ett prosjekt.

Lag derfor en ER-modell som inneholder informasjonen over.

- I denne oppgaven skal du svare med digital håndtegning. Bruk eget skisseark (utdelt). Se instruksjon for utfylling av skisseark i lenken under oppgavelinjen.
- Tegn og skriv tydelig.
- 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.

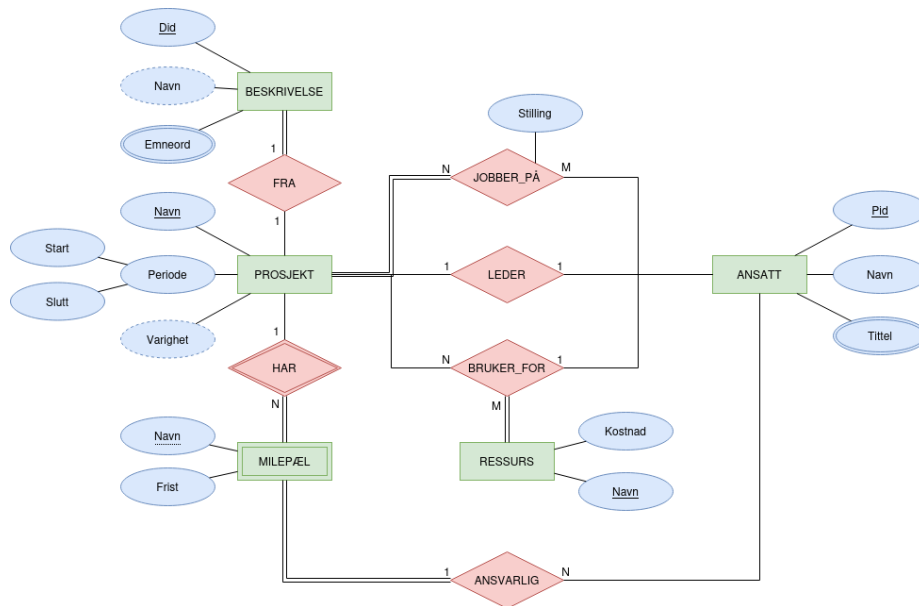


Figure 1: Prosjekt-modell

Løsningsforslag

Sensorveiledning

- -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 for å bruke identifiserende 1:1:N-ternær-relasjon med total deltakelse fra svak KOMMENTAR for siste del

Følgende beskjed ble gitt til alle kandidater under eksamen:

Det er en skrivefeil i oppgave 1.1. I nest siste punkt skal det stå "milepæl" hvor det nå står "prosjekt".

1.2 ER: Romvesner (5)

HUFF (Departementet for Hemmelige, Utenomjordiske og Farefulle Fenomener) har omsider fått kontakt med noen romvesner de er svært interessert i å kommunisere med. Romvesnene kommuniserer utelukkende med ER-diagrammer, og

har sendt diagrammet under.

Hvilke av disse setningene er sanne i henhold til ER-diagrammet. Kryss av “JA” dersom du mener setningen er modellert av ER-diagrammet, og “NEI” hvis ikke.

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

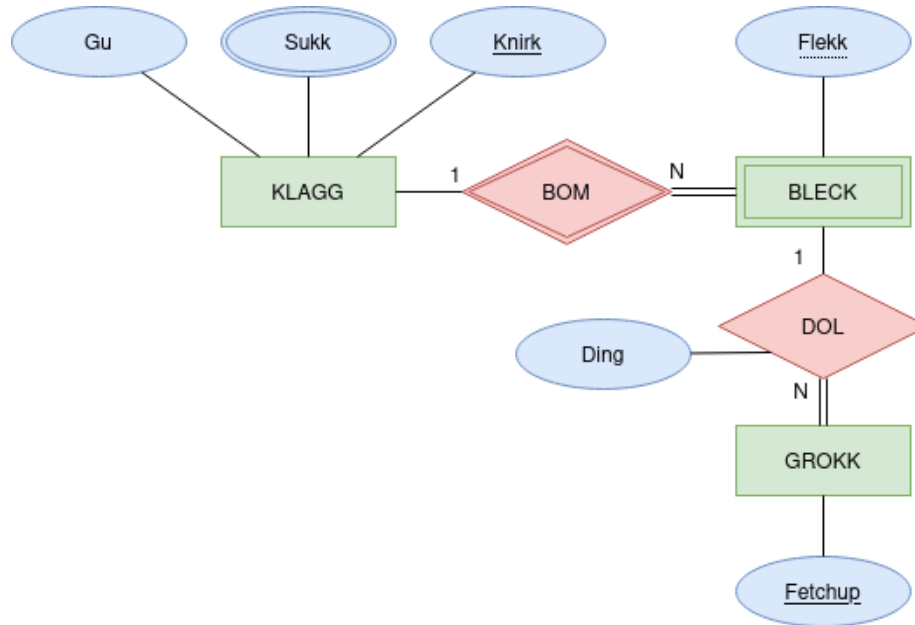


Figure 2: romvesen-modell

1. En KLAGG har minst ett Sukk
2. I DOL-relasjonen kan det være forskjellige Ding-verdier for samme BLECK
3. En KLAGG kan være relatert til mange forskjellige BLECK
4. Hvis det finnes minst én GROKK i databasen, så er det minst én KLAGG også
5. En BLECK kan ikke identifiseres uten å vite dens DOL-relaterte GROKK

Løsningsforslag

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

Automatisk rettet.

1.3 Realisering: Bestillinger (10)

Du er ansatt i et firma som skal lage en ny database over firmaets kunder, produkter og bestillinger. Domene-ekspertene har laget følgende ER-modell:

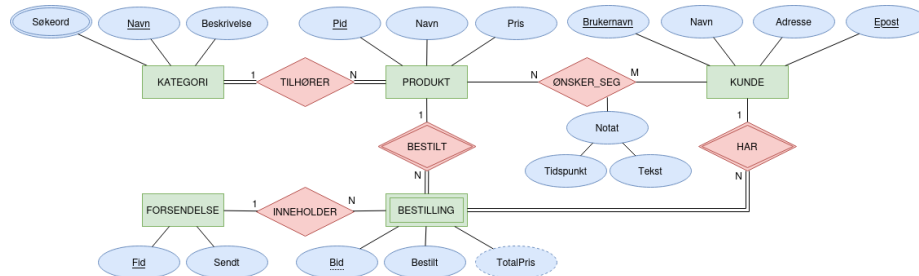


Figure 3: ER-modell over bestillinger

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. Gjør det tydelig hvilke attributter som utgjør kandidatnøkler og fremmenøkler. Dersom det er flere kandidatnøkler, marker også valgt primærnøkkel. F.eks. bruk understreking for å markere kandidatnøkler, fet skrift for primærnøkler, og 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

Bruker her KN, PN og FN som forkortelse for henholdsvis *kandidatnøkler*, *primærnøkkel*, og *fremmednøkkel*.

Vi starter med å realisere de vanlige entitetene:

Kategori(Navn, Beskrivelse)

- KN: {Navn}
- PN: {Navn}

Produkt(Pid, Navn, Pris)

- KN: {Pid}
- PN: {Pid}

Kunde(Brukernavn, Navn, Adresse, Epost)

- KN: {Brukernavn}, {Epost}
- PN: {Brukernavn}

Forsendelse(Fid, Sendt)

- KN: {Fid}
- PN: {Fid}

Her velger vi Brukernavn som primærnøkkel for Kunde.

Så realiserer vi de svake entiteten, og dens to identifiserende relasjoner:

Bestilling(Bid, Pid, Brukernavn, Bestilt)

- KN: {Bid, Pid, Brukernavn}
- PN: {Bid, Pid, Brukernavn}
- FN: Pid -> Produkt(Pid)
Brukernavn -> Kunde(Brukernavn)

Det er ingen 1:1-relasjoner, så realiserer derfor 1:N-relasjonene, hvor vi velger å realisere TILHØRER inn i Produkt siden alle produkter har TILHØRER nøyaktig én KATEGORI og blir dermed aldri NULL. Vi velger også å realisere INNEHOLDER som fremmednøkkel i BESTILLING, ettersom det er naturlig at de fleste bestillinger er del av en forsendelse. Vi oppdaterer da Bestilling og Produkt til:

Produkt(Pid, Navn, Pris, Kategori)

- KN: {Pid}
- PN: {Pid}
- FN: Kategori -> Kategori(Navn)

Bestilling(Bid, Pid, Brukernavn, Bestilt, Forsendelse)

- KN: {Bid, Pid, Brukernavn}
- PN: {Bid, Pid, Brukernavn}
- FN: Pid -> Produkt(Pid)
Brukernavn -> Kunde(Brukernavn)
Forsendelse -> Forsendelse(Fid)

Så realiserer vi M:N-relasjonen som egen relasjon:

ØnskerSeg(Produkt, Kunde, Tidspunkt, Tekst)

- KN: {Produkt, Kunde}
- PN: {Produkt, Kunde}
- FN: Produkt -> Produkt(Pid)
Kunde -> Kunde(Brukernavn)

Til slutt realiserer vi flerverdi-attributten Sørkeord:

Sørkeord(Kategori, Sørkeord)

- KN: {Kategori, Sørkeord}
- PN: {Kategori, Sørkeord}
- FN: Kategeori -> Kategori(Navn)

Hele skjemaet blir da:

Kategori(Navn, Beskrivelse)

- KN: {Navn}
- PN: {Navn}

Kunde(Brukernavn, Navn, Adresse, Epost)

- KN: {Brukernavn}, {Epost}
- PN: {Brukernavn}

Forsendelse(Fid, Sendt)

- KN: {Fid}
- PN: {Fid}

Produkt(Pid, Navn, Pris, Kategori)

- KN: {Pid}
- PN: {Pid}
- FN: Kategori -> Kategori(Navn)

Bestilling(Bid, Pid, Brukernavn, Bestilt, Forsendelse)

- KN: {Bid, Pid, Brukernavn}
- PN: {Bid, Pid, Brukernavn}
- FN: Pid -> Produkt(Pid)
Brukernavn -> Kunde(Brukernavn)
Forsendelse -> Forsendelse(Fid)

ØnskerSeg(Produkt, Kunde, Tidspunkt, Tekst)

- KN: {Produkt, Kunde}
- PN: {Produkt, Kunde}
- FN: Produkt -> Produkt(Pid)
Kunde -> Kunde(Brukernavn)

Søkeord(Kategori, Søkeord)

- KN: {Kategori, Søkeord}
- PN: {Kategori, Søkeord}
- FN: Kategori -> Kategori(Navn)

Sensorveiledning

- 0.5 for hver vanlige entitet
- 3 poeng for den svake entiteten (inkludert de identifiserende relasjonene)
- 1 poeng for hver relasjon, og 1.5 for hver relasjon med attributt
- 0.5 for multi-valued attributten, utledbar attributt, og sammensatt nøkkel

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
- -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

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 kun ha primærnøkkel i GROUP BY men andre attributter (fra samme tabell) i SELECT (støttes av PostgreSQL)
- Ingen trekk for enkle syntaksfeil man enkelt ville funnet ut av om man hadde mulighet til å kjøre spørringen

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, oppgi hvor mange rader det er i resultatet. 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 spiller
WHERE navn LIKE '%d%';
```

2.

```
SELECT *
FROM spiller
WHERE sid = 5 OR rating = NULL;
```

3.


```

SELECT DISTINCT spilltype
FROM parti
    JOIN turnering USING (tid)
WHERE vinner = 'remis';

```

4.

```

SELECT tid_brukt
FROM trekk
WHERE tur >= 10
LIMIT 4;

```

5.

```

SELECT spilltype
FROM turnering
WHERE startdato > '2022-01-01'
INTERSECT ALL
SELECT spilltype
FROM turnering
WHERE tid != 4;

```

6.

```

SELECT *
FROM spiller AS s
    LEFT OUTER JOIN parti AS p
        ON (s.sid = p.sort)
WHERE vinner IS NULL;

```

Løsningsforslag

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

Automatisk rettet.

2.2 Åpninger (5)

Skriv en spørring som finner `pid` og tiden brukt på alle trekk som enten er `'e4'` eller `'c3'` og som er på første tur (altså tur 1).

Løsningsforslag

```
SELECT pid, tid Brukt
FROM trekk
WHERE tur = 1 AND
      (trekk = 'e4' OR trekk = 'c3');
```

2.3 Skakk-matt vinnere (5)

Skriv en SQL-spørring som finner navn på alle spillere som har vunnet med sorte brikker i turneringen med navn 'Skakk-matt'. Svaret skal kun inneholde unike navn.

Løsningsforslag

```
SELECT DISTINCT s.navn
FROM spiller AS s
      JOIN part AS p ON (s.sid = p.sort)
      JOIN turnering AS t USING (tid)
WHERE t.navn = 'Skakk-matt' AND
      p.vinner = 'sort';
```

2.4 Korte partier (5)

Skriv en SQL-spørring som finner pid, vinner og antall trekk for hvert parti, men kun for de partiene med færre enn (eller lik) 10 trekk (spørringen skal også ha med de som ikke har noen trekk med antall lik 0).

Løsningsforslag

```
SELECT p.pid, p.vinner, count(t.pid) AS antall
FROM parti AS p
      LEFT OUTER JOIN trekk AS t ON (p.pid = t.pid)
GROUP BY p.pid, p.vinner
HAVING count(t.pid) <= 10;
```

2.5 Ny spiller (8)

Skriv en SQL-kommando som setter inn en ny spiller med navn 'Mona', som har sid-verdi lik 1 pluss høyeste sid-verdi allerede i spiller-tabellen, og som har rating lik gjennomsnittsratingen til alle spillerene allerede i spiller-tabellen som har spilt minst ett parti.

Løsningsforslag

```
INSERT INTO spiller(sid, navn, rating)
SELECT
```

```

max(s.sid)+1 AS sid,
'Mona' AS navn,
(SELECT avg(s2.rating)
 FROM spiller AS s2
 WHERE EXISTS (
   SELECT 1
   FROM parti
   WHERE s2.sid = hvit OR
         s2.sid = sort
 )
) AS rating
FROM spiller AS s;

-- eller
INSERT INTO spiller(sid, navn, rating)
VALUES (
  (SELECT max(sid)+1 FROM spiller),
  'Mona',
  (SELECT avg(rating)
   FROM spiller AS s
   WHERE EXISTS (
     SELECT 1
     FROM parti
     WHERE s.sid = hvit OR
           s.sid = sort
   )
  )
);

```

2.6 Populære åpninger (10)

Skriv en spørring som finner det mest populære åpningstrekket (altså det vanligste trekket for tur lik 1) for hver spilltype (altså 'lynsjakk', 'hurtigsjakk', osv.). Resultatet skal også inneholde det mest populære trekket for partier som ikke er tilknyttet noen turnering, og du kan da la `spilltype` være 'uavhengig' i resultatet.

Merk: Det kan finnes flere spilltyper i databasen enn de som er listet i eksempel-dataene.

Løsningsforslag

```

WITH
trekk_per_stype AS (
  SELECT t.spilltype, tr.trekk, count(*) AS antall
  FROM turnering AS t
       JOIN parti AS p USING (tid)

```

```

        JOIN trekk AS tr USING (pid)
    WHERE tr.tur = 1
    GROUP BY t.spilltype, tr.trekk
    UNION -- kan også ha UNION ALL her
    SELECT 'uavhengig' AS spilltype, tr.trekk, count(*) AS antall
    FROM parti AS p
        JOIN trekk AS tr USING (pid)
    WHERE p.tid IS NULL AND
        tr.tur = 1
    GROUP BY tr.trekk
),
spilletyper AS (
    SELECT DISTINCT spilltype
    FROM trekk_per_stype
)
SELECT
    st.spilltype,
    (SELECT t.trekk
     FROM trekk_per_stype AS t
     WHERE st.spilltype = t.spilltype
     ORDER BY t.antall DESC
     LIMIT 1
    )
FROM spilletyper AS st;

```

Sensorveiledning

3 Relasjonsmodellen og normalformer (25)

3.1 Nøkler (7) (flervalg)

Gitt følgende relasjon

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

med følgende FDer:

1. $AB \rightarrow D$
2. $E \rightarrow G$
3. $D \rightarrow G$
4. $F \rightarrow A$

Angi for hvert attributt hvorvidt det er et nøkkelattributt (JA) eller ikke (NEI). Hvert riktige svar gir 1 poeng, mens hvert gale svar gir -1 poeng. Ubesvart gir alltid 0 poeng. Laveste poengsum for hele oppgaven er 0.

Er attributtet et nøkkelattributt i R?

Attributt	JA	NEI
A		
B		
C		
D		
E		
F		
G		

Løsningsforslag

Attributt	JA	NEI
A		X
B	X	
C	X	
D		X
E	X	
F	X	
G		X

Automatisk rettet.

3.2 Normalformer (8) (flervalg)

Gitt følgende relasjon:

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

med kandidatnøkler AB og BC .

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
$B \rightarrow E$				
$AB \rightarrow C$				
$C \rightarrow A$				
$E \rightarrow D$				

Løsningsforslag

FD	1NF	2NF	3NF	BCNF
$B \rightarrow E$	X			
$AB \rightarrow C$				X
$C \rightarrow A$			X	
$E \rightarrow D$		X		

Automatisk rettet.

3.3 Tapsfri dekomposisjon (10)

Gitt følgende relasjon:

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

med kandidatnøkkel ACE , samt følgende FDer:

1. $AC \rightarrow B$
2. $B \rightarrow D$
3. $E \rightarrow FG$

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

Løsningsforslag

Starter med å splitt siste FD opp i to:

1. $AC \rightarrow B$
2. $B \rightarrow D$
3. $E \rightarrow F$
4. $E \rightarrow G$

Ser at FD 1 bryter med BCNF siden AC ikke er en supernøkkel, så dekomponerer mhp. denne: Får da $S_1(A, B, C, D)$ med AC som kandidatnøkkel og FDene 1 og 2, og $S_2(A, C, E, F, G)$ med kandidatnøkkel ACE og FDene 3 og 4.

- For $S_1(A, B, C, D)$ så bryter ikke FD 1 med BCNF, men FD 2 bryter, så dekomponerer videre mhp. denne: Får da $S_{11}(B, D)$ med FD 2 og kandidatnøkkel B , og $S_{12}(A, B, C)$ med FD 1 og dermed kandidatnøkkel AC . Ingen av disse bryter med BCNF.
- For $S_2(A, C, E, F, G)$ så bryter FD 3 med BCNF, så dekomponerer mhp. denne: Får da $S_{21}(E, F, G)$ med FDene 3 og 4 og kandidatnøkkel E , og $S_{22}(A, C, E)$ uten noen FDer og kandidatnøkkel ACE . Begge disse er dermed på BCNF, og vi er ferdige.

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

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