

Løsningsforslag – eksamen IN2090

Høsten 2019

Dato: 06.12.2019

1 Modellering (40)

Sensorveiledning:

Generelt

- 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)
- 2 for manglende nøkkel-attributt, multivalued-attributt, el.
- 1 for manglende/feil plassering av vanlig attributt
- 2 for manglende markering av svak entitet eller identifiserende relasjon

1.1 Modellering: Kandidater (7)

Sensorveiledning:

Detaljert poengfordeling:

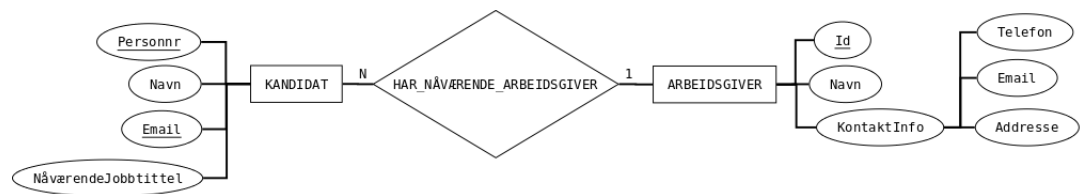
- 2 for KANDIDAT
- 2 for relasjonen
- 3 for ARBEIDSGIVER

Greit om JobbTittel plasseres på relasjonen JOBBER_FOR (altså ingen trekk).

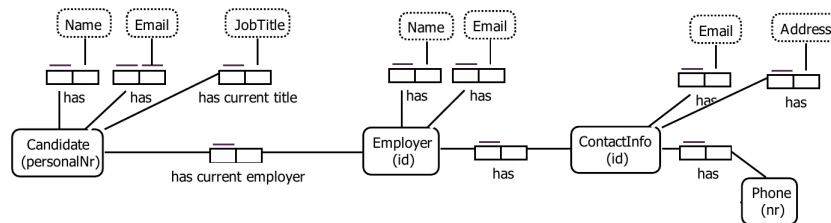
Løsningsforslag:

Merk: Adresse på kandidater ble ikke nevnt i den engelske versjonen. Ingen trekk dersom denne mangler.

ER:



ORM:



1.2 Modellering: Intervjuer (18)

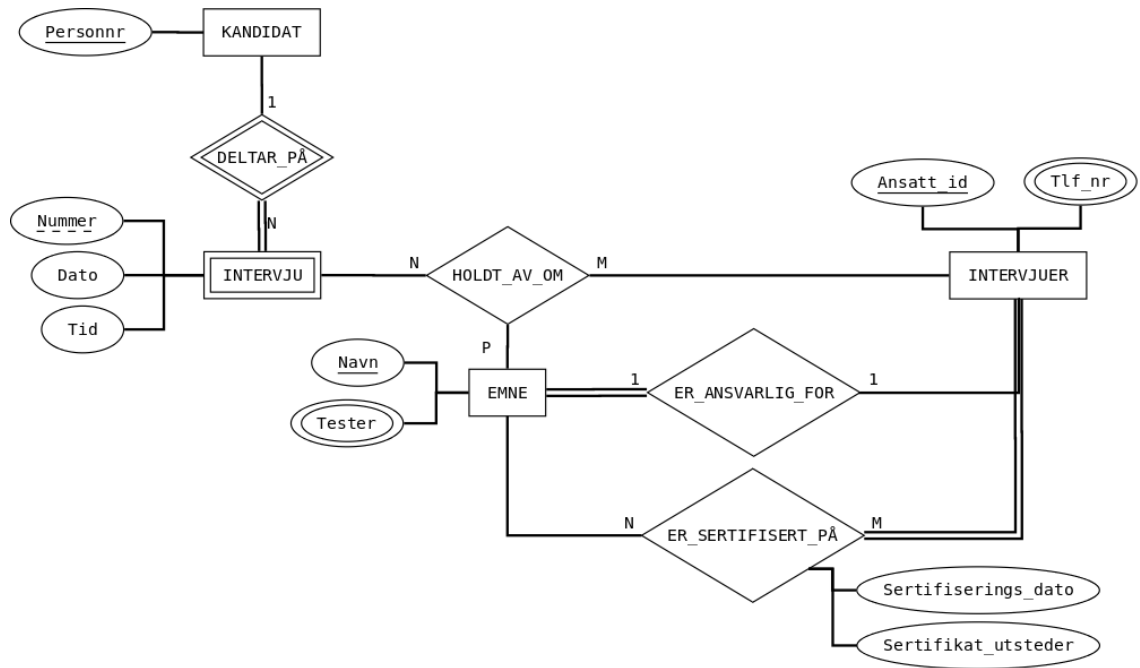
Sensorveiledning:

Detaljert poengfordeling:

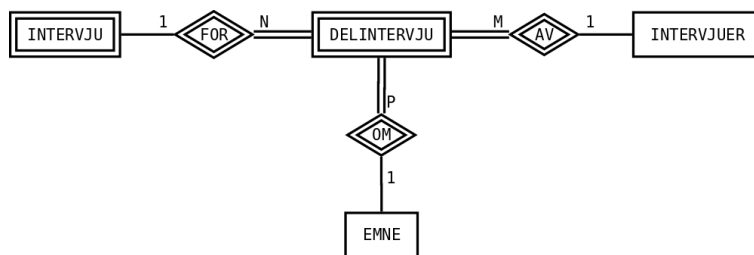
- 4 for INTERVJU med DELTAR_PÅ-relasjonen
(Merk: Ikke sagt eksplisitt hvor mange intervjuer en kandidat kan delta på, både 1 og N godtas.)
- 4 for den ternære relasjonen HOLDT_AV_OM eller alternativ modellering (se under)
- 2 for hver av EMNE og INTERVJUER (med attributter)
(Merk: Ansatt_id ikke sagt at skal være unik, ikke trekk for om unik/ikke unik.)
- 3 for ER_ANSVARLIG_FOR med kardinaliteter
- 3 for ER_SERTIFISERT_PÅ med kardinaliteter

Løsningsforslag:

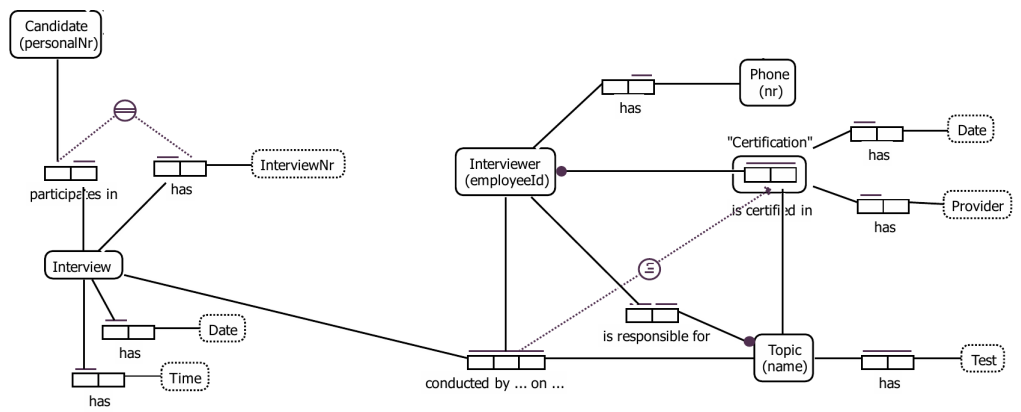
ER:



Alternativt kan den ternære relasjonen modelleres slik:

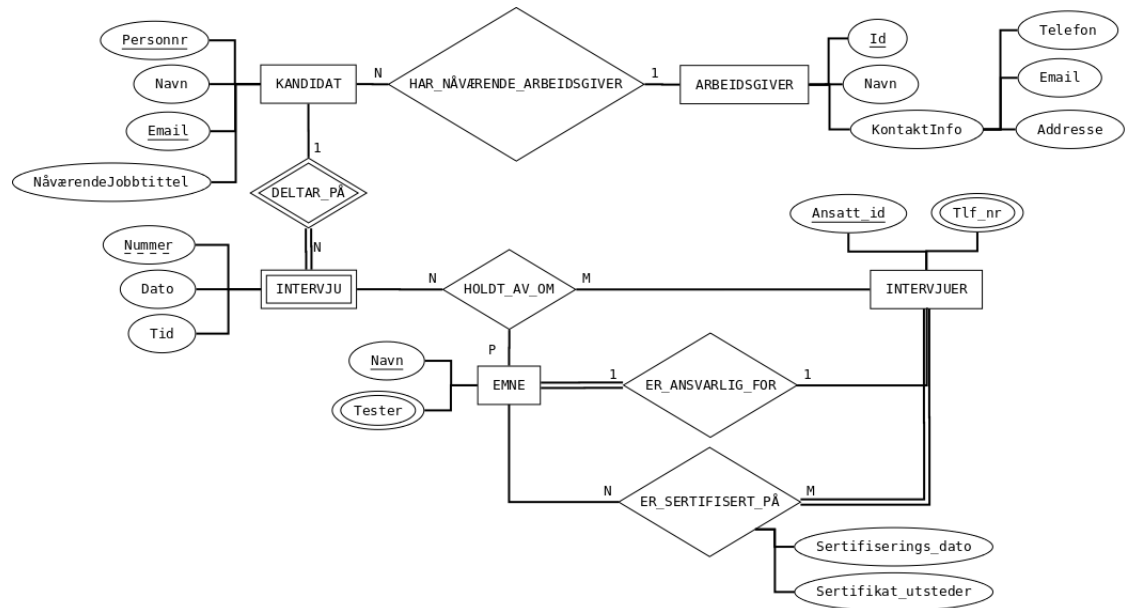


ORM:

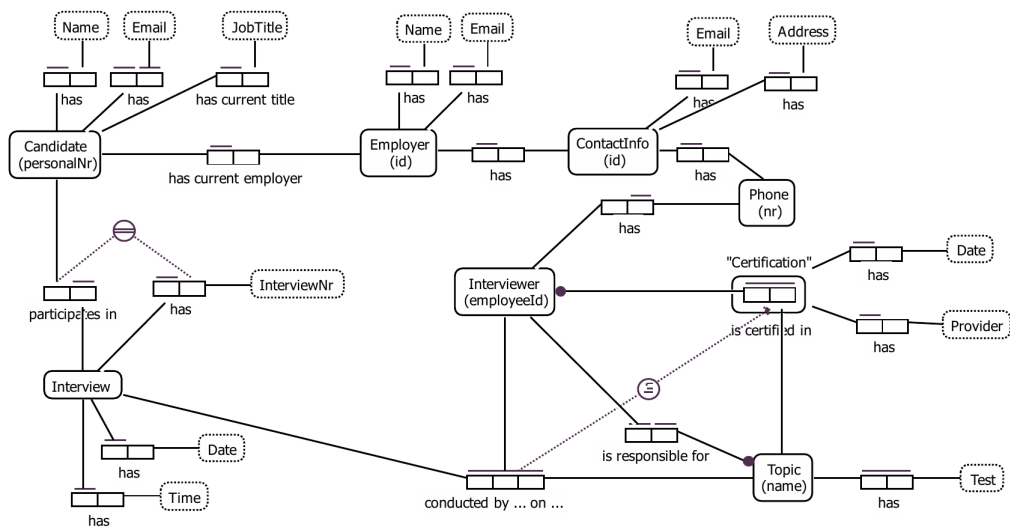


Løsningsforslag: (Hele modellen)

ER:



ORM:



1.3 Realisering (15)

Sensorveiledning:

Poengfordeling:

- 2 for alle (ikke-svake) entiteter
- 3 for CHAPTER
- 1 for mange-til-mange-relasjoner
- 2 for en-til-en -og en-til-mange-relasjoner

Trekk:

- 0.5 for manglende eller feil plassering av attributt
- 1 for manglende kandidat- eller primærnøkkel-markering (men kun -0.5 dersom har primærnøkkel-markering men ikke kandidatnøkkel-markering)
- 1 for manglende fremmednøkkel-markering
- 1 for færre eller flere attributter i nøkkel enn nødvendig
- 1 for manglende forklaring på valg av realisering av en-til-en og en-til-mange-relasjoner
- 2 for å realisere CONTAINS dobbelt (en for svake CHAPTER og en for relasjonen selv)

Løsningsforslag:

Realiseringen gir følgende databaseskjema:

Book(ISBN, WebAddr, Title, ContactAuth, ContactEmail, Publisher, Year)
Author(Id, FName, LName, Phone)
Publisher(Pid)
Chapter(ISBN, ChNr, Title)
BookAuthor(ISBN, Aid)
AuthorPublisherContract(Aid, Pid, Amount)
Offices(Pid, Office)

hvor vi har følgende fremmednøkler:

Book(ContactAuth) → Author(Id)
Book(Publisher) → Publisher(Pid)
Chapter(ISBN) → Book(ISBN)
BookAuthor(ISBN) → Book(ISBN)
BookAuthor(Aid) → Author(Id)
AuthorPublisherContract(Aid) → Author(Id)
AuthorPublisherContract(Pid) → Publisher(Id)
Offices(Pid) → Publisher(Id)

Av de to kandidatnøklerne for BOOK har vi valgt ISBN som primærnøkkel da dette virker mest naturlig. Vi har videre valgt å legge HAS_MAIN_CONTACT-relasjonen inn i relasjonen *Book*, ettersom det er naturlig å anta at de fleste bøker har en hovedkontantperson. Vi har også valgt å legge IS_PUBLISHED_BY inn i *Book*, ettersom det er naturlig å anta at de fleste bøker har et forlag/publisher.

2 SQL (50)

2.1 Skranker og SQL (10)

Sensorveiledning:

-1.25 for galt svar

1.25 for riktig svar

Merk: Skrivefeil i engelsk versjon, men denne skrivefeilen ble opplyst om til alle som tok eksamen.

Rettes automatisk. 0 poeng er laveste totale poengsum for oppgaven.

Løsningsforslag:

	Lovlig	Ulovlig
INSERT INTO dyr VALUES (7, 'Mona', 1, 0);		X
INSERT INTO dyr VALUES (6, 'Timmy', NULL, 2);	X	
INSERT INTO art VALUES (4, 'Flue', 'Innsekt');		X
UPDATE dyr SET mor_til = 3 WHERE did = 0;	X	
UPDATE dyr SET mor_til = 4 WHERE mor_til IS NULL;		X
DELETE FROM dyr WHERE navn = 'Mons';		X
DELETE FROM dyr WHERE did = 3;	X	
DROP TABLE art CASCADE;	X	

2.2 Band etter 2000 (5)

Sensorveiledning:

For denne og de neste oppgavene:

- 2 for manglende %
- 1 for = fremfor **LIKE**
- 2 for **AND** i stedet for **OR**
- 1 for manglende eller ekstra kolonner

Ingen trekk for sammenlikning direkte med årstall, håndtere datoer som strenger, el. (INF1300 og tidligere IN2090 gjennomgikk ikke håndtering av datoer i detalj.)

Løsningsforslag:

```
SELECT navn, startet
FROM band
WHERE startet >= '2000-01-01' OR
      navn LIKE '%King%';
```

Også greit om antar at etter 2000 betyr startet >= '2001-01-01'.

2.3 Timer Pop-musikk fra 90s (5)

Sensorveiledning:

- 2 for manglende /3600.0 (men OK om ikke tar med desimalet)
- 2 for manglende uttrykk i **WHERE**
- 2 for sjangerid = 0 el.

- 1 for `count` i stedet for `sum`
- 2 for manglende aggregering
- 2 for manglende/feil tabeller i joins
- 2 for meningsløs `GROUP BY` (f.eks. på `album.navn`)
- 2 for `NATURAL JOIN`
- 1 for å finne antall timer per band (med `GROUP BY`)

Ingen trekk for sammenlikning direkte med årstall, håndtere datoer som strenger, el. (INF1300 og tidligere IN2090 gjennomgikk ikke håndtering av datoer i detalj.)

Løsningsforslag:

```
SELECT sum(s.spilletid)/3600.0 AS timer
  FROM sjanger AS sj
       INNER JOIN band AS b USING (sjangerID)
       INNER JOIN album AS a USING (bandID)
       INNER JOIN sang AS s USING (albumID)
 WHERE sj.navn = 'Pop' AND
       b.startet >= '1990-01-01' AND
       b.startet < '2000-01-01';
```

2.4 Personer født på interessant dato (5)

Sensorveiledning:

- 1 for unødvendige duplikater (f.eks. `UNION ALL` i union av joins)
- 2 om spørringen bruker kryssprodukt av person, band, album og `OR`
- 3 for å finne personer født på samme dato som bandet personen spiller i ble startet eller ga ut et album (joins og `OR`)

Løsningsforslag:

```
SELECT navn
  FROM person
 WHERE født IN (
   SELECT startet
   FROM band
   UNION
   SELECT utgitt
```



```
    FROM album
);
```

eller med OR:

```
SELECT navn
  FROM person
 WHERE født IN (
    SELECT startet
    FROM band
  ) OR født IN (
    SELECT utgitt
    FROM album
  );
```

eller med union av joins:

```
SELECT p.navn
  FROM person AS p
        INNER JOIN band AS b ON (p.født = b.startet)
UNION
SELECT p.navn
  FROM person AS p
        INNER JOIN album AS a ON (p.født = a.utgitt);
```

2.5 Sanger per band (5)

Sensorveiledning:

- 1 for * i `count`
- 2 for `INNER JOIN` men ellers korrekt
- 2 for manglende `HAVING` eller `GROUP BY`

Løsningsforslag:

```
SELECT b.bandID, b.navn, count(s.sangID) AS antall_sanger
  FROM band AS b
        LEFT OUTER JOIN album AS a USING (bandID)
        LEFT OUTER JOIN sang AS s USING (albumID)
 GROUP BY b.bandID, b.navn -- b.navn ikke nødvendig (i PostgreSQL)
HAVING count(s.sangID) < 3;
```

2.6 Slett tomme album (5)

Sensorveiledning:

- 1 for `DROP` fremfor `DELETE`
- 2 for (outer) joins i `FROM` (men ellers riktig)
- 4 for inner join i `FROM` med `NOT NULL`-sjekk el.
- 5 poeng for riktig tilpassning av spørring fra forrige oppgave (selvom spørringen fra forrige oppgave er feil, trekker ikke for følgefeil)

Løsningsforslag:

```
DELETE FROM album
WHERE albumID NOT IN (
  SELECT albumID
  FROM sang
);
```

2.7 Nyeste album (5)

Sensorveiledning:

- 0.5 for å ikke navngi kolonnene (fikse navnkonflikten med `navn`)
- 2 for manglende `LIMIT` eller `ORDER BY`-klausul
- 2 for ikke å telle antall sanger
- 1 for manglende `GROUP BY` (men teller sanger)
- 0.5 for manglende kolonner i `GROUP BY` (Merk: `a.navn`, `a.utgitt` ikke nødvendig)
- 1 for manglende `DESC` og -0.5 for eksplisitt `ASC`

Løsningsforslag:

```
CREATE VIEW nyeste_album(navn, band, utgitt, antall_sanger) AS
SELECT a.navn, b.navn, a.utgitt, count(*)
FROM album AS a
      INNER JOIN band AS b USING (bandID)
      INNER JOIN sang AS s USING (albumID)
GROUP BY a.albumID, a.navn, b.navn, a.utgitt
ORDER BY a.utgitt DESC
LIMIT 10;
```

2.8 Super-album (10)

Sensorveiledning:

- 5 for riktig delspørring som finner alle super-album
- 5 for riktig delspørring som finner antallet super-album per band
- 3 for manglende antall album
- 2 for manglende `GROUP BY`

Løsningsforslag:

```
WITH
  super_album AS (
    SELECT albumID
    FROM sang
    GROUP BY albumID
    HAVING sum(spilletid) > 3600
  )
SELECT b.bandID, b.navn, count(*) AS antall
FROM band AS b
  INNER JOIN album AS a USING (bandID)
  INNER JOIN super_album AS sa USING (albumID)
GROUP BY b.bandID, b.navn;
```

3 Relasjonsmodellen (10)

3.1 FDer (6)

Sensorveiledning:

På 1):

- 1 for avhuk på attributt som ikke er med i tillukningen
- 0.4 for avhuk på attributt som er med i tillukningen
- 0 er minimal poengsum

På 2):

- 1 for avhuk på attributt som ikke er med i tillukningen
- 2 for avhuk på attributt som er med i tillukningen
- 0 er minimal poengsum

På 3):

-1 for avhuk på attributt som ikke er med i alle kandidatnøkler

1 for avhuk på attributt som er med i alle kandidatnøkler

Automatisk rettet. 0 poeng er laveste totale poengsum for oppgaven.

Løsningsforslag:

1. Hva er tillukningen til $\{A\}$.

Løsning: $\{A\}^+ = \{A, B, C, D, F\}$.

2. Hva er tillukningen til $\{B\}$.

Løsning: $\{B\}^+ = \{B\}$.

3. Hvilke attributter må være med i alle kandidatnøkler:

Løsning: $\{A, E\}$

3.2 Normalformer (4)

Sensorveiledning:

-1 for gal normalform

1 for riktig normalform

Automatisk rettet. 0 poeng er laveste totale poengsum for oppgaven.

Løsningsforslag:

	BCNF	3NF	2NF	1NF
$A, B \rightarrow F$	X			
$A, D \rightarrow E$			X	
$B \rightarrow D$				X
$A \rightarrow C$				X