

Løsningsforslag – eksamen IN2090

Høsten 2019

Dato: 06.12.2019

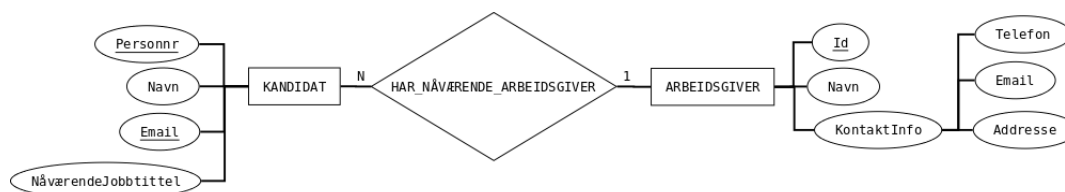
1 Modellering (40)

1.1 Modellering: Kandidater (7)

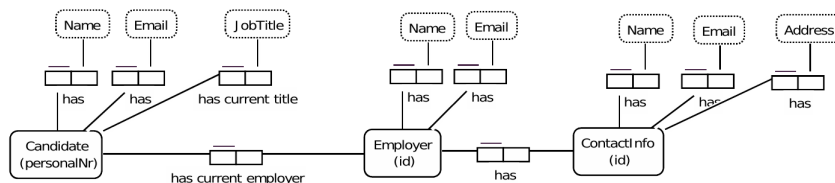
Løsningsforslag:

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

ER:



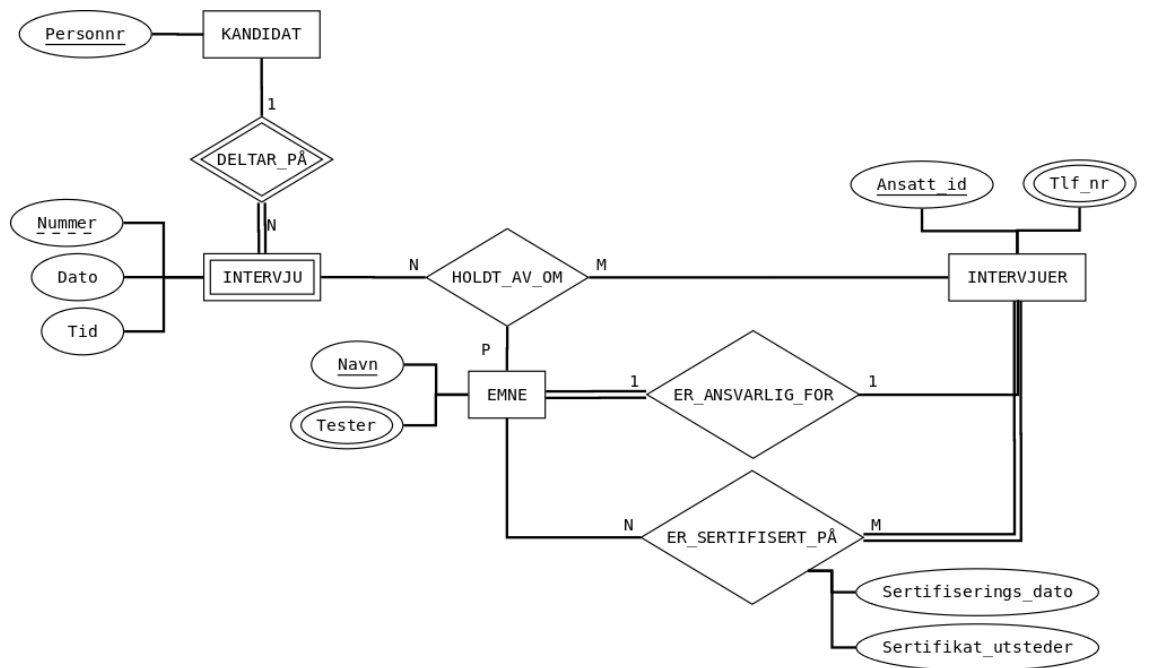
ORM:



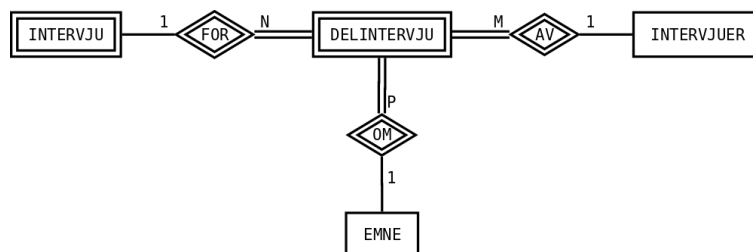
1.2 Modellering: Intervjuer (18)

Løsningsforslag:

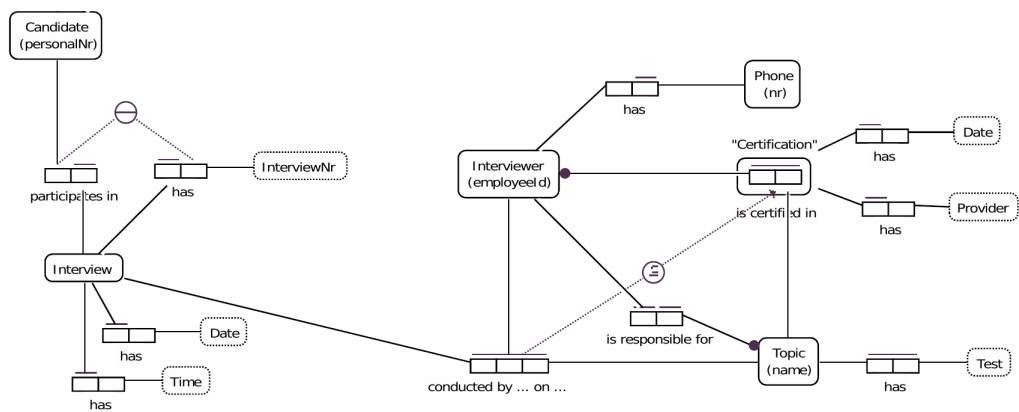
ER:



Alternativt kan den ternære relasjonen modelleres slik:

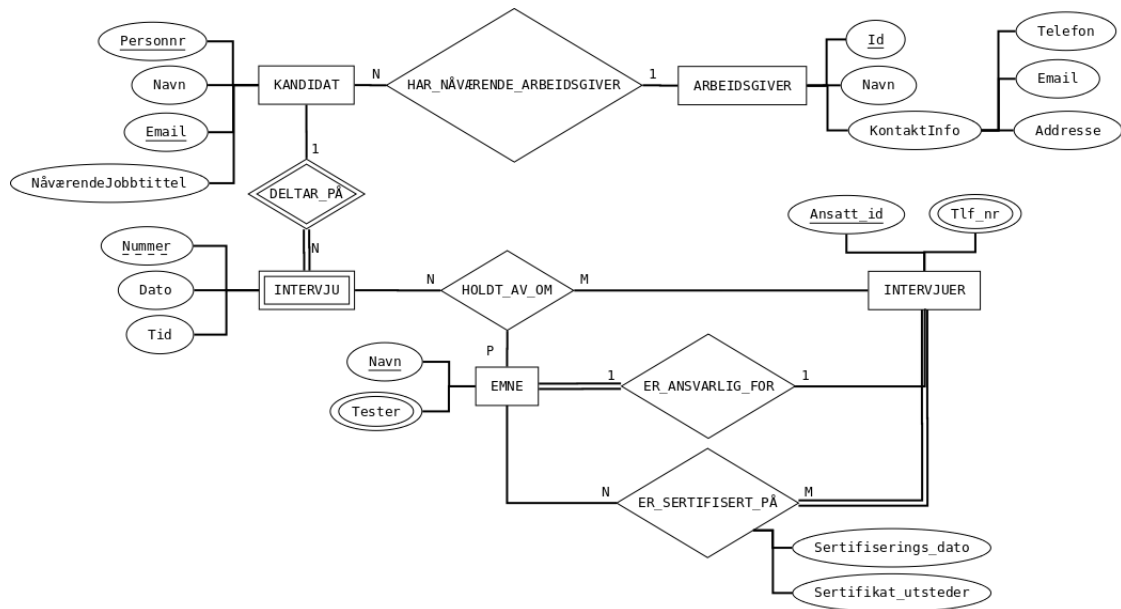


ORM:

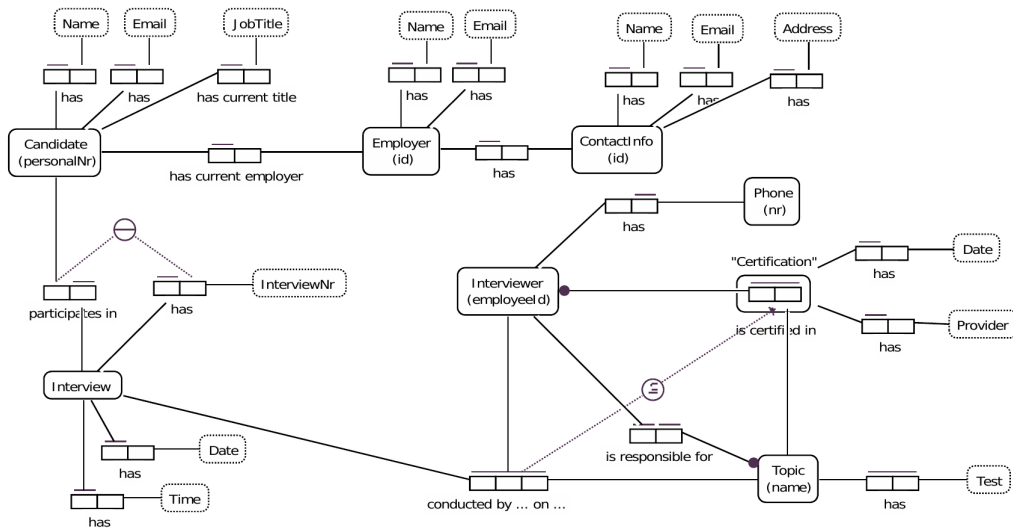


Løsningsforslag: (Hele modellen)

ER:



ORM:



1.3 Realisering (15)

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(Pid)
 Offices(Pid) → Publisher(Pid)

Vi har her 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)

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)

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)

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)

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)

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)

Løsningsforslag:

```

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

```

2.7 Nyeste album (5)

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)

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)

Automatisk rettet.

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)

Automatisk rettet.

Løsningsforslag:

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