

# ER-modellering Bibliotek

# Introduktion

Dokumentet innehåller en ER-modellering av ett bibliotek.

## Konceptuell modellering

### 1. Beskrivning

Biblioteket lagrar böcker och har en databas för alla böcker.

En bok har ISBN, titel och författare.

Varje bok är kategoriserad i en eller flera kategorier.

En kategori kan ha flera böcker.

En kategori har ett namn, en beskrivning och ett id.

Uppgifter om författaren lagras också, såsom namn, ålder och födelseland.

Författaren kan skriva flera böcker.

Varje bok kan ha en eller flera författare.

En utgivare ger ut böcker.

Varje bok har en utgivare.

En utgivare har ett id, namn och land.

### 2. Entiteter (med attribut)

- bok (ISBN, titel och författare)
- kategori (namn, beskrivning och id)
- författare (namn, ålder och födelseland)
- utgivare (id, namn och land)

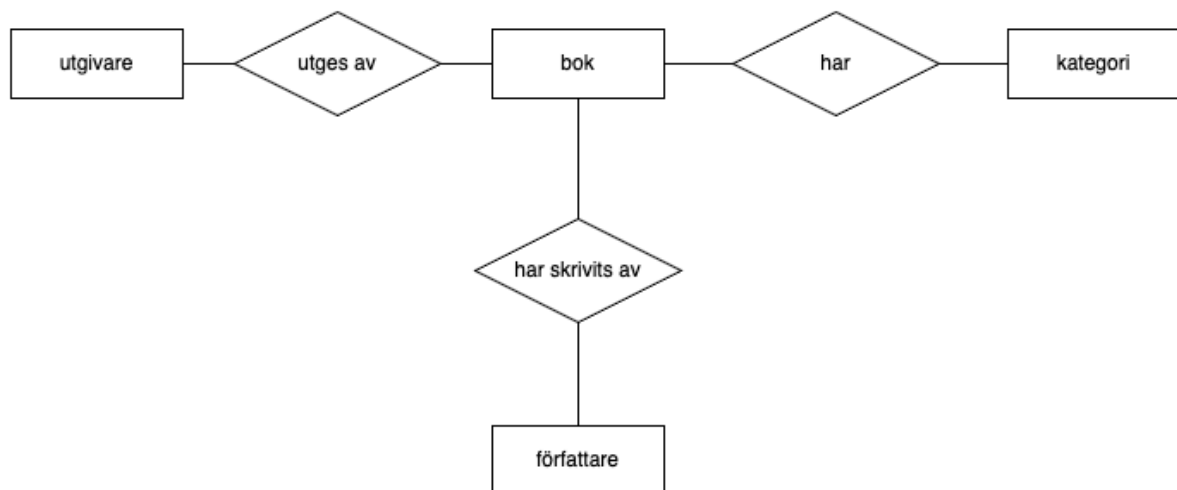
### 3. Relationer

	bok	kategori	författare	utgivare
bok		har en eller flera kategorier	har en eller flera författare	har en utgivare
kategori	kan ha flera			

	böcker			
författare	kan skriva flera böcker			
utgivare	ger ut böcker			

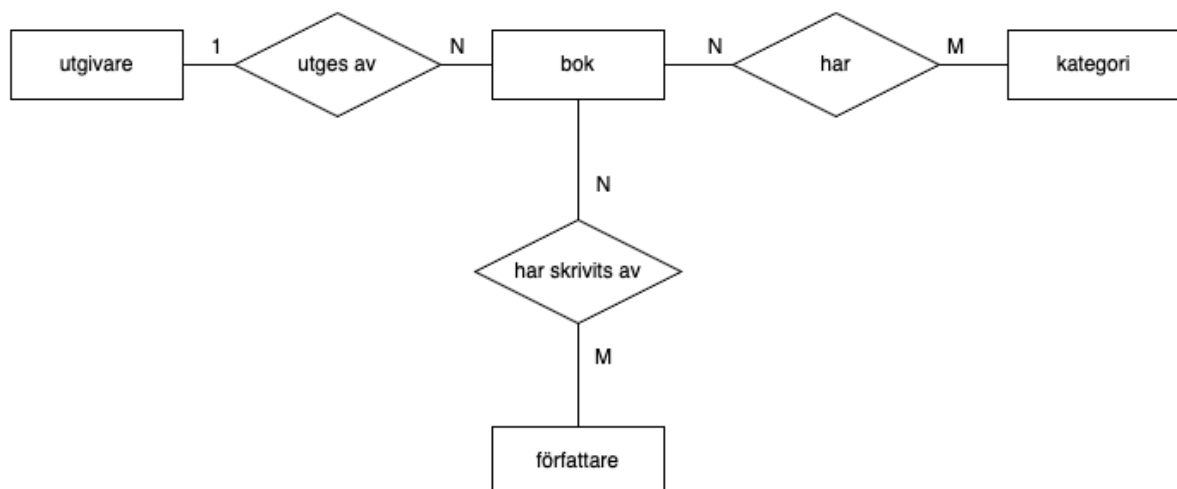
#### 4. ER-diagram med entiteter och relationer

Fördelen med att spara ner varje steg är att det är enklare att lägga till en entitet i detta diagram.



#### 5. ER-diagram med kardinalitet

Jämför för relationerna i matrisen och dubbelkolla med texten i steg 1.



## 6. ER-diagram med alla attribut samt kandidatnycklar

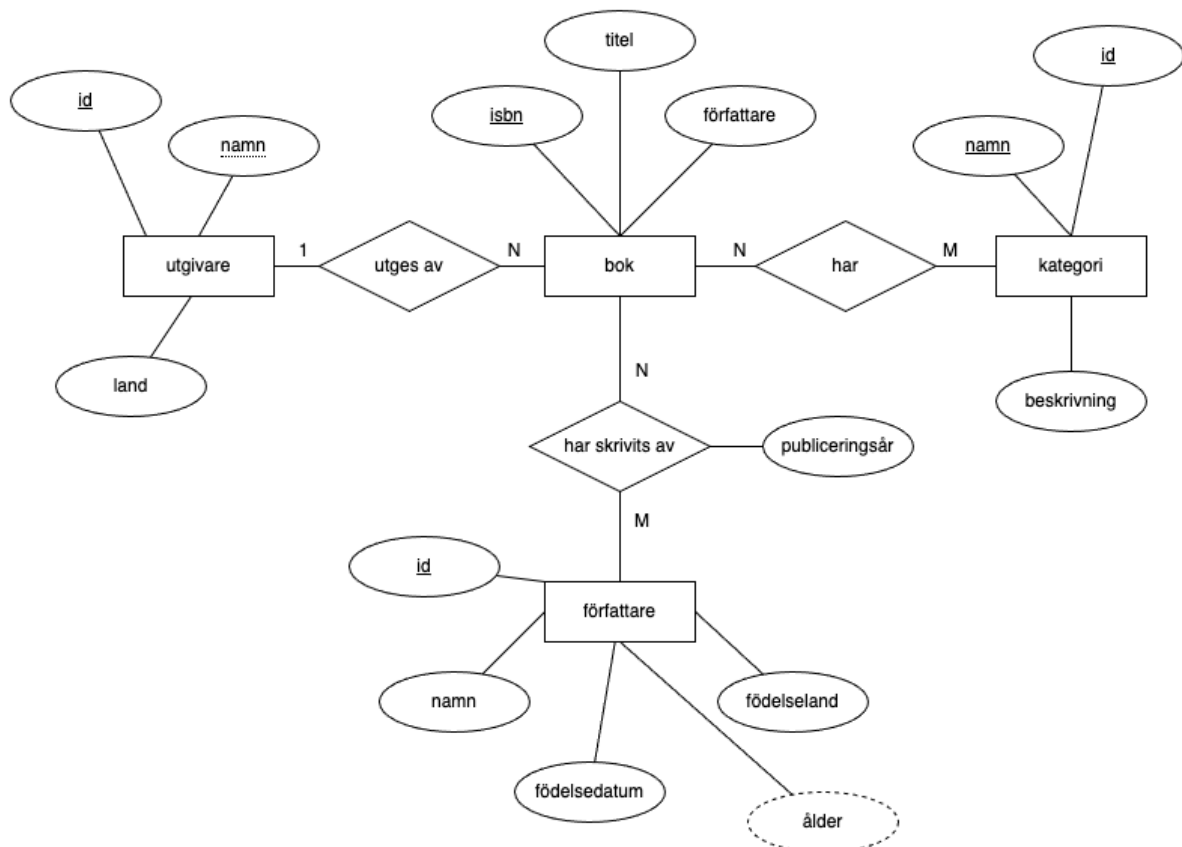
Kandidatnyckel - ett attribut som kan peka ut en rad unikt och är kandidat till att vara primärnyckel. Därför kan vi markera flera om vi vill. Varje entitet måste ha minst 1 kandidatnyckel.

Lägga till andra attribut?

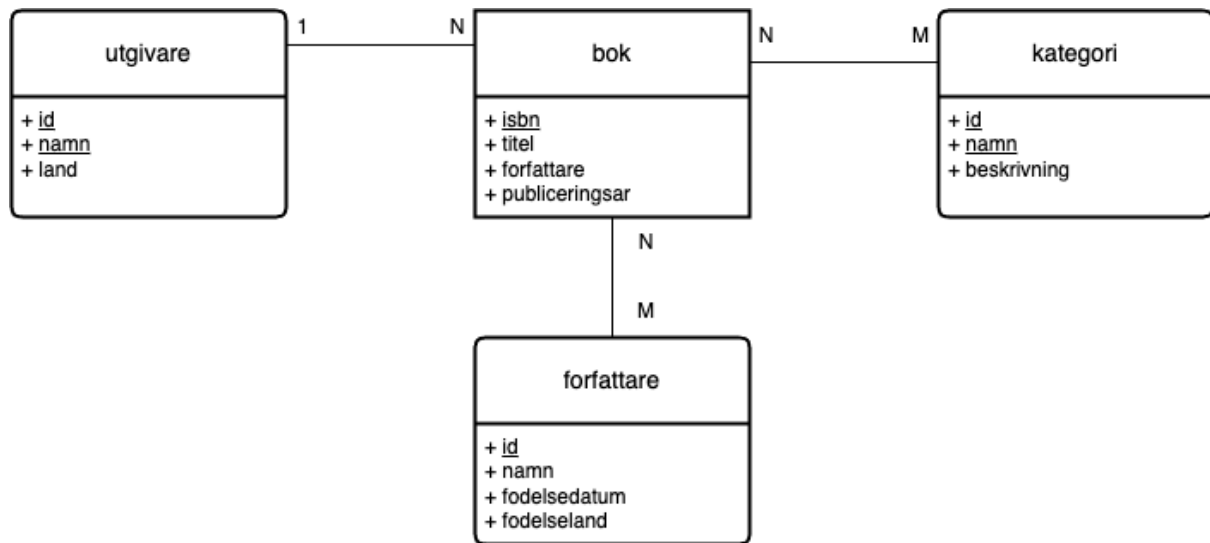
- lägger till födelsedatum på författare istället för ålder som är "derived", dvs kan räknas eller tas fram från andra attribut.
- lägger till ett attribut på en relation, publiceringsår

Resonera kring varje entitet om vilka som ska vara kandidatnycklar.

- utgivare, räcker id eller ska vi namn också men jag vet inte så antingen väntar jag med att markera den eller markerar den med en sträckad linje för att markera att den är svag
- bok, isbn är unikt men räcker det eller behöver jag ett id
- kategori, id naturligtvis och så måste väl namnet kategorin också vara det
- författare, id



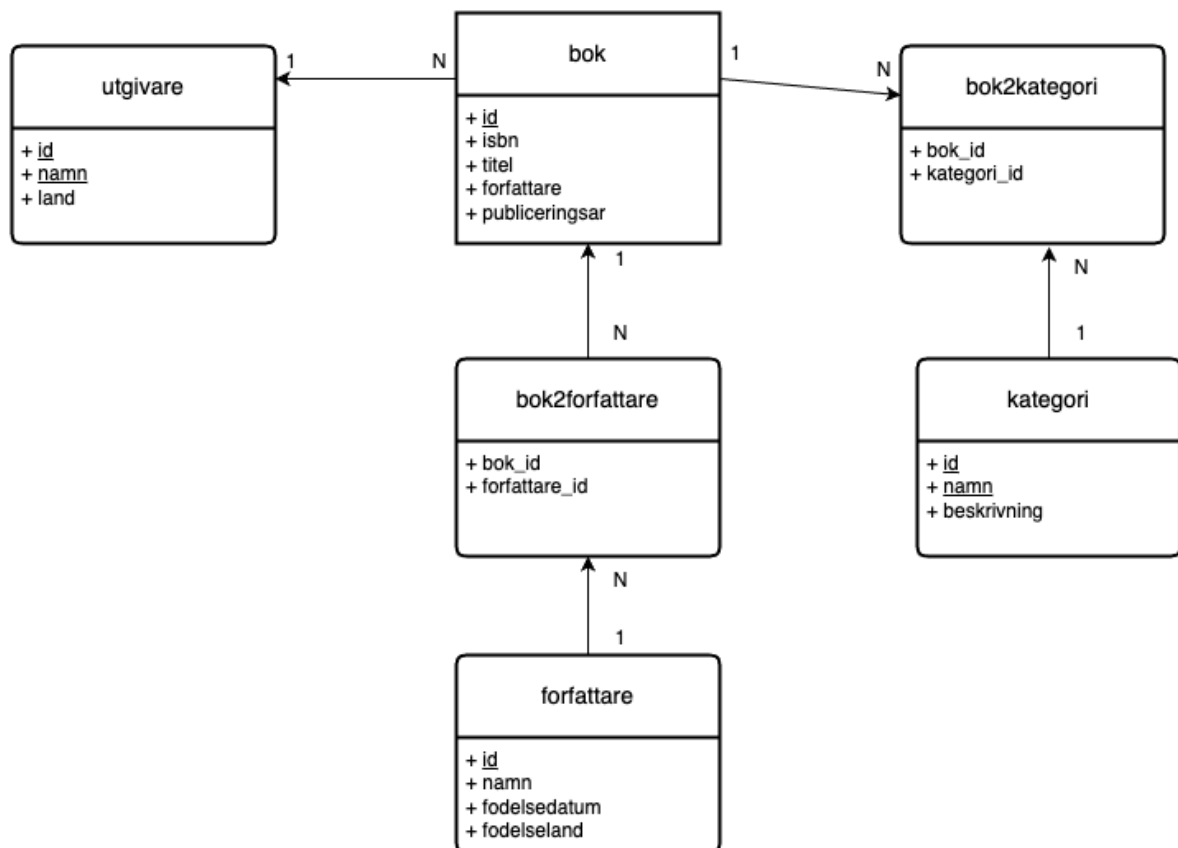
Alternativ visualisering av steg 6 genom entities i draw.io.



# Logisk modellering

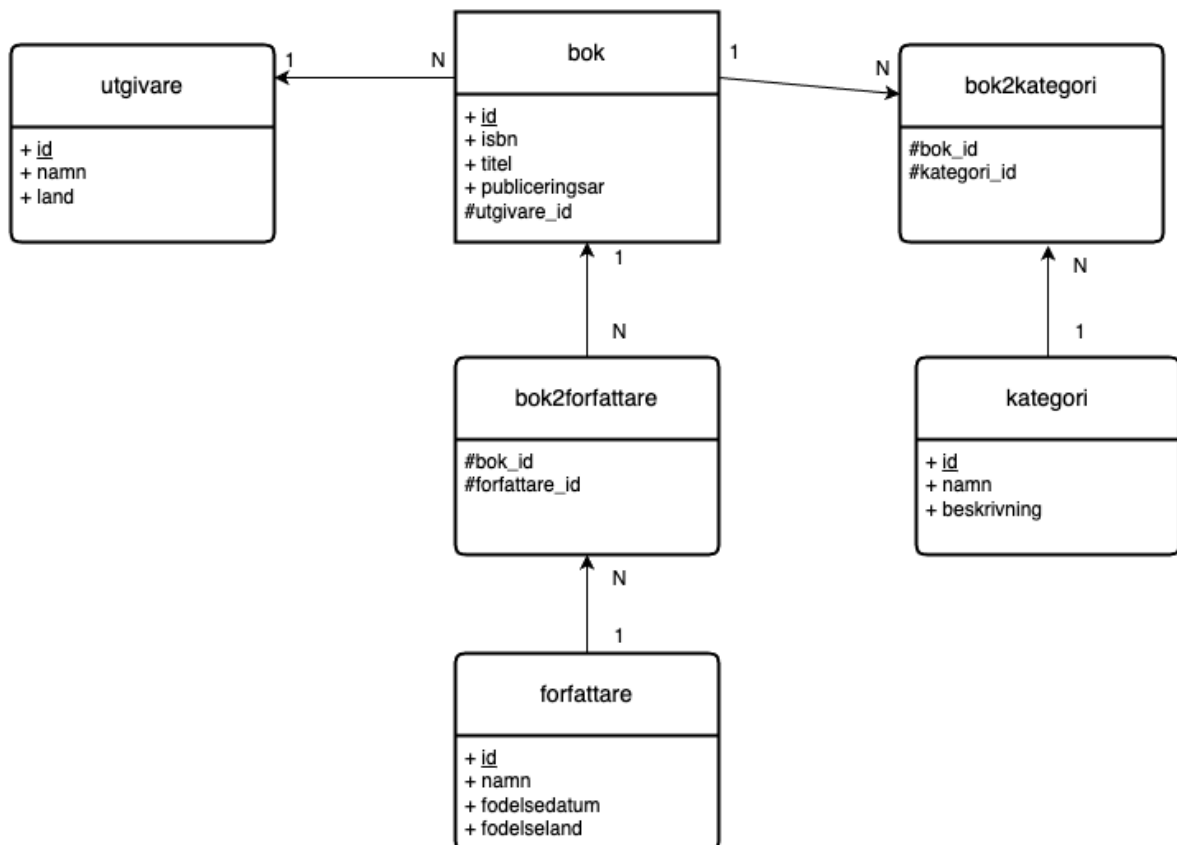
## 7. ER-diagram enligt relationsmodellen

Fortsätt på alternativ 2 från steg 6. Fixa många-till-många med kopplingstabeller mellan bok och författare samt bok och kategori. Antingen kopplar vi bok med isbn eller gör vi ett id på bok och jag väljer ett id.



## 8. ER-diagram med primära och främmande attribut samt kompletterande attribut

Så där då ska vi titta på nycklar. Vi ju förslag på kandidatnycklar från steg 6. Attributen i kopplingstabellerna blir främmande nycklar. Enligt kokboken ska de markeras med #. De andra understrukna attributen blir våra primärnycklar.



# Fysisk modellering del 2

## 9. SQL DDL tabeller

Alla tabeller finns i Appendix A.

## 10. Funktioner (API)

Dessa funktioner ska databasen stödja:

- Visa alla författare och dess information
- Visa alla författare födda i ett visst land
- Visa de böcker som ett förlag har gett ut
- Visa böckerna och hur de är kategoriserade
- Visa böckerna en författare har skrivit

Dessa vyer finns i databasen:



# Appendix A SQL DDL

```
DROP TABLE IF EXISTS utgivare;
DROP TABLE IF EXISTS kategori;
DROP TABLE IF EXISTS forfattare;
DROP TABLE IF EXISTS bok;
DROP TABLE IF EXISTS bok2forfattare;
DROP TABLE IF EXISTS bok2kategori;

CREATE TABLE utgivare
(
    id INTEGER NOT NULL,
    namn VARCHAR(120) NOT NULL,
    land VARCHAR(120),

    PRIMARY KEY (id)
);

CREATE TABLE kategori
(
    id INTEGER NOT NULL,
    namn VARCHAR(40) NOT NULL,
    beskrivning VARCHAR(255),

    PRIMARY KEY (id),
    UNIQUE (name)
);

CREATE TABLE forfattare
(
    id INTEGER NOT NULL,
    namn VARCHAR(120) NOT NULL,
    fodelsedatum DATE,
    fodelseland VARCHAR(120),

    PRIMARY KEY (id)
);

CREATE TABLE bok
(
    id INTEGER NOT NULL,
    isbn VARCHAR(15),
    titel VARCHAR(120) NOT NULL,
    publiceringsar INTEGER DEFAULT YEAR(NOW()),
    utgivare_id INTEGER,

    PRIMARY KEY (id),
    FOREIGN KEY (utgivare_id) REFERENCES utgivare (id)
);

CREATE TABLE bok2forfattare
```

```
(
    bok_id INTEGER NOT NULL,
    forfattare_id INTEGER NOT NULL,

    PRIMARY KEY (bok_id, forfattare_id),
    FOREIGN KEY (bok_id) REFERENCES bok (id),
    FOREIGN KEY (forfattare_id) REFERENCES forfattare (id)
);
```

CREATE TABLE bok2kategori

```
(
    bok_id INTEGER NOT NULL,
    kategori_id INTEGER NOT NULL,

    PRIMARY KEY (bok_id, kategori_id),
    FOREIGN KEY (bok_id) REFERENCES bok (id),
    FOREIGN KEY (kategori_id) REFERENCES kategori (id)
);
```