Facit och kommentarer till tentamen 2026-02-11 i DA4001

1.
```
-- a. Högsta registrerade årsinkomsten
SELECT MAX(yearly_income)
  FROM characteristics;

-- b.
SELECT A.fname, A.lname, C.yearly_income
  FROM anonymization AS A
  INNER JOIN characteristics AS C
    ON A.userid = C.anonymous_user;

-- c.
SELECT C.address_area, COUNT(UI.interest_key)
  FROM characteristics AS C
  INNER JOIN user_interests AS UI
    ON C.anonymous_user = UI.anonymous_user
  GROUP BY C.address_area;


-- d.
SELECT AVG(yearly_income)
  FROM characteristics AS C
  INNER JOIN user_interests AS UI
    ON C.anonymous_user = UI.anonymous_user
  INNER JOIN interests AS I
    ON UI.interest_key = I.interest_key
  WHERE interest_desc = 'Golf';

-- e.
CREATE VIEW shared_interests AS
  SELECT DISTINCT F.*
    FROM friendship AS F
    INNER JOIN user_interests AS UI1
      ON UI1.anonymous_user = F.user1
    INNER JOIN user_interests AS UI2
      ON UI2.anonymous_user = F.user2
  WHERE F.strength >= 0.5
    AND UI1.interest_key = UI2.interest_key;

-- f.
SELECT A.fname, A.lname, I.interest_desc
  FROM anonymization AS A
  OUTER JOIN user_interests AS UI
    ON A.userid = UI.anonymous_user
  INNER JOIN interests AS I
    ON UI.interest_key = I.interest_key
  ;
```

2. (a) Ett exempel på referensvillkor är i tabellen characteristics:

```
FOREIGN KEY anonymous_user REFERENCES
    anonymization(userid)
```

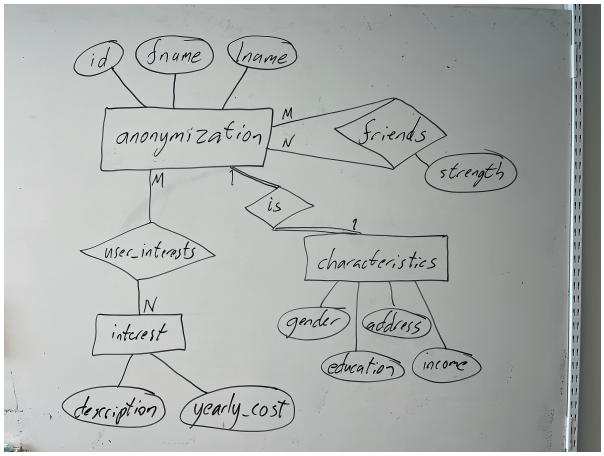
(b) Flera av attributen borde man kräva har värden, dvs NULL är inte tillåtet. Vi kan tex ändra första tabellen till:

```
CREATE TABLE anonymization (
       userid CHAR(10) PRIMARY KEY,
       fname  VARCHAR(50) NOT NULL,
       lname  VARCHAR(50) NOT NULL,
       );
```

(c) Semantiska integritetsvillkor innebär att vi kodar in kunskap om den verksamhet som vi modellerar som går utöver "praktiska" krav som NOT NULL. Ett enkelt sådant är i *characteristics*:

```
yearly_income INTEGER CHECK (yearly_income
    >= 0)
```

3. Ett ER-diagram baserat på databasschemat:



4. Kandidatnycklar:

```
anonymization: userid och (fname, lname)
characteristics: anonymous_user
user_interests: (anonymous_user, interest_key)
friendship: (user1, user2)
```

(a) För att schemat ska vara på BCNF måste alla determinanter, dvs de attribut som bestämmer andra attribut och kombinationer av attribut ($A \rightarrow B$) också vara kandidatnycklar. Det gäller trivialt i alla relationer, men notera särskilt att i anonymization, där det finns två kandidatnycklar, bestämmer de varandra.

(b) Om vi ändrar schemat så att en tabell inte är på 1NF så bryter vi mot BCNF. Ett exempel är att lägga till telefonnummer i anonymization och hävda att vi får lista flera telefonnummer där så länge de är komma-separerade.

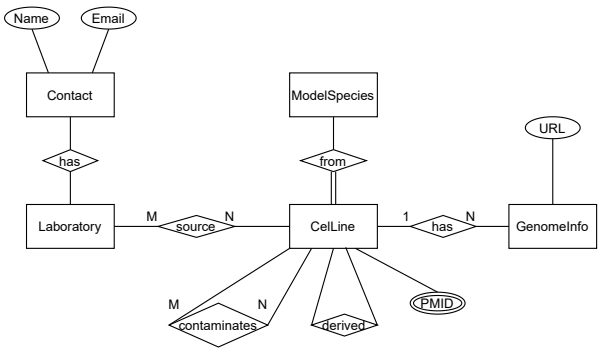
5. Det är en databas med personuppgifter och då krävs det rättslig grund för lagringen. Det finns flera möjliga grunder och för marknadsföring kan man till exempel hävda ”berättigat intresse”, men eftersom det lagras mer än bara kontaktuppgifter så krävs det antagligen informerat samtycke från de inblandade.

   När organisationen inte behöver uppgifterna längre, eller användaren drar tillbaka sitt samtycke, så bör uppgifterna raderas.

   I schemat kan man läsa att ”gender” är en lagrad uppgift. Det kan tolkas som en känslig personuppgift och får inte lagras. Det är dessutom svårt att hävda att organisationen *behöver* den uppgiften för marknadsföring, så min tolkning är att den uppgiften bryter mot dataskyddsförordningen.

6. I databasmodellering, och specifikt ER-modellering, innebär objektifiering av en sambandstyp att man omvandlar en sambandstyp till en egen entitet. Syftet kan vara att koppla attribut eller ytterligare relationer direkt till ett samband, eller att man vill göra sambandet lite tydligare/viktigare.

7. ER-diagram för AlbanoMed:



8. Jag gör ett databasschema (se figur 1) utan finesser och fokuserar bara på relationerna.

   Det är praktiskt att införa identifierare för viktiga entiteter. Med unika identifierare behöver man inte fundera på om, tex, namnet på en cellinje eller ett laboratorium är unikt.

   Jag försöker följa kursbokens kokboksmetod.

   - De centrala entitetstyperna `CelLine` och `Laboratory` blir genast relationer/tabeller.
   - Sambandstypen mellan `CelLine` och `Laboratory` är M-N, så vi inför en relation, `CelLineSource`, för att länka ihop entiteter från dessa.
   - Kontaktinformationen är underspecificerad i ER-modellen. Jag väljer att tänka att man kan ha flera kontakter på ett labb, så implementerar stöd för en 1-N-relation. Relationen `Contact` får därför ett fält `lab_id` som länkar till labbet.
   - `GenomeInfo` har en 1-N-relation till cellinjer. Rekommendationen är då att man skapar en relation `GenomeInfo` som har ett referensattribut till cellinjens tabell.

- M-N-relationen `contaminates` länkar två cellinjer. Det finns ju riktning på det kontamination, så jag försöker uttrycka det som att en `contaminant` kontaminerar en viss cellinje som ges av `cell_line_id` .
- M-N-relationen `derived` fungerar förstås på exakt samma sätt som `contaminates`.
- Vi vill hålla reda på artinformation för källan till cellinjen. Även om modellen inte uttrycker det så väl som den borde, så kan ju en art ge upphov till flera cellinjer. Därför bör artinformationen ligga i en separat relation. Utan attribut så är det i och för sig inte ett självklart val; man kunde ju låta artnamnet vara enda attributet, men för att ”framtidssäkra” schemat så blir det enklare att lägga till attribut till arter om man låter dem ligga i en egen tabell.
- Litteraturen ligger som ett flervärt attribut i modellen. Det behöver då en egen tabell, med hänvisning till en artikel (`pmid`) och en cellinje. Det blir automatiskt så att en artikel kan diskutera flera cellinjer och varje cellinje kan refereras i flera artiklar utan problem.

```
CREATE TABLE CelLine (
  cell_line_name VARCHAR(50),
  cell_line_id INTEGER,
  species_id INTEGER,
);

CREATE TABLE Laboratory (
   lab_name VARCHAR(50),
   lab_id INTEGER
);

CREATE TABLE CelLineSource (
   cell_line_id INTEGER,
   lab_id INTEGER,
   (cell_line_id, lab_id)
);

CREATE TABLE Contact (
  name VARCHAR(50),
  email VARCHAR(50),
  lab_id INTEGER
);

CREATE TABLE GenomeInfo (
   url VARCHAR(100),
   cell_line_id INTEGER
);

CREATE TABLE Contaminates (
   contaminant INTEGER,
   cell_line_id INTEGER,
    (contaminant, cell_line_id)
);

CREATE TABLE Derived (
   origin INTEGER,
   cell_line_id INTEGER,
    (origin, cell_line_id)
);

CREATE TABLE ModelSpecies (
   species_id INTEGER,
   species_name VARCHAR(20)
);

CREATE TABLE Literature (
   pmid INTEGER,
   cell_line_name VARCHAR(50),
    (pmid, cell_line_name)
);
```

Figur 1: Databasschema för fråga 8.