

Facit och kommentarer till tentamen 2025-12-17 i DA4001

1.

- (a) SQL-injektion är att lura ett datorprogram att köra annan SQL-kod än den tänkta, typiskt via ett formulär eller en textruta som inte blivit validerad.
- (b) En kandidatnyckel är en minimal supernyckel, vilket i sin tur är ett eller flera attribut som unikt identifierar en tupel/rad i en relation/-tabell. Om man tar bort ett attribut från en kandidatnyckel så är det inte längre en nyckel.
- (c) En entitetstyp är en kategori av saker i en datamodell som man vill lagra information om.
- (d) Ett index är en datastruktur som gör att man kan slå upp (lokalisera) attribut (och kombinationer av attribut) snabbt, samt iterera över dem snabbt, utan att behöva gå igenom hela tabeller.
- (e) Det tar minne att ha ett index, vilket kan vara slösaktigt, och det krävs datorberäkningar för att underhålla indexet när man lägger till, tar bort, eller ändrar data.

- (f)

1	3	1	5
1	3	2	6
2	4	1	5
2	4	2	6
- (g) **INSERT INTO** Table2 **VALUES** (3, 7);

2.

```
-- a
SELECT COUNT(*) FROM family;

-- b
SELECT family, count(*)
FROM familymembers
GROUP BY family;

-- c
WITH familyinfo AS (
    SELECT fam_acc, COUNT(*) AS N
    FROM family AS F
    INNER JOIN familymembers AS FM ON F.fam_acc=FM.
        family
    GROUP BY F.fam_acc
)
SELECT fam_acc, N
FROM familyinfo
WHERE N = (SELECT MAX(N) FROM familyinfo);

-- d Lista alla proteiner som finns i samma familjer
    som STF1_HUMAN
```

```

SELECT DISTINCT FM2.protein, FM1.family -- We will get
    repetitions
FROM familymembers AS FM1
INNER JOIN familymembers AS FM2 ON FM1.family == FM2.
    family
WHERE FM1.protein='STF1_MOUSE' AND FM1.protein <> FM2
    .protein;

-- Använd FM1 för att ta reda på vilka familjer sö
    kproteinet finns i.
-- Med FM2 listar vi proteinerna för de funna
    familjerna.

-- e
WITH RECURSIVE protlist(accession, family) AS (
    SELECT 'STF1_HUMAN', 'NHR5'
    UNION
    SELECT DISTINCT FM2.protein, FM1.family -- We
        will get repetitions
    FROM protlist AS PL
    INNER JOIN familymembers AS FM1 ON PL.accession
        = FM1.protein
    INNER JOIN familymembers AS FM2 ON FM1.family ==
        FM2.family
    WHERE FM1.protein <> FM2.protein
)
SELECT DISTINCT accession FROM protlist;

```

3.

- (a) Du har rätt att kräva att företag raderar alla data om dig, förutsatt att det inte inskränker på tex skyldigheter till myndigheter eller krav från dig om uppdrag.
- (b) Banker har skyldighet att förebygga bedrägerier och det är rättslig grund för att lagra viss information om deras kunder.
- (c) Till exempel genetisk information och hälsa, men man kan också nämna sexuell läggning, fackligt deltagande, religion, etniskt ursprung.
- (d) Du har rätt till insyn i automatiska beslut. När beslut fattas automatiskt måste den personuppgiftsansvariga tala om för dig att beslutet är automatiserat, ge dig rätt att få beslutet granskat av en riktig person, och ge dig möjlighet att bestrida beslutet.

4.

- (a) Romben representerar en sambandstyp. Attributet "name" finns i två entitetstyper, så det döps om.
- (b) Lämpliga relationer:
 - Plant: (plantname, placement, species, plantURL)
 - Param: (paramname, description, unit)
 - Observation: (plantname, paramname, day, time, value)

- (c) För *Plant* är både *species* och *plantURL* ickenyckelattribut, men *species* bestämmer värdet på *plantURL*. Det tillåts inte av 3NF som kräver att ickenyckelattribut inte får vara funktionellt beroende av andra ickenyckelattribut.
- (d) Ta bort *plantURL* från *Plant* och skapa en ny relation som parar *species* och *plantURL*.

5. Here is an attempt.

```

CREATE TABLE Plant (
  plantname VARCHAR(20) PRIMARY KEY,
  placement VARCHAR(20) NOT NULL, --Right, Bo?
  species    VARCHAR(20) NOT NULL, --Bo knows this
  plantURL   VARCHAR(20)
);

CREATE TABLE Param (
  paramname  VARCHAR(20) PRIMARY KEY,
  description VARCHAR(20),
  unit       VARCHAR(5) -- Expecting cm, joule, etc
);

CREATE TABLE Observation (
  plantname VARCHAR(20),
  paramname VARCHAR(20),
  obsday    DATE CHECK (day > '1990-01-10'),
            -- Bo bought his first computer this day
  obstime   TIME,
  val       DOUBLE
  PRIMARY KEY (day, t) -- Questionable choice, but
                    that is what Bo asked for,
  FOREIGN KEY plantname REFERENCES Plant(plantname),
  FOREIGN KEY paramname REFERENCES Param(paramname),
);

```