Facit och kommentarer till tentamen 2024-08-13 i DA4001 Databasteknik

1. (a) Dataskyddsförordningen

   (b) Man behöver *informerat samtycke* för att få spara *personuppgifter*. Enligt media har EU kritiserat upplägget med att användaren blir ställd inför ett tveksamt val (min tolkning), men rimligen borde den verkliga kritiken vara att det inte är ett tillräckligt informerat val. Användaren ska informeras om *hur* personuppgifterna används, och det kan hända att det inte görs på ett ordentligt sätt. Lagen säger dock inget om att ta betalt för alternativet att slippa spårning.

   (c) Termen berättigat intresse används när organisationer/företag har anledning att använda personuppgifter och deras intresse trumfar de som personuppgifterna gäller. Ett exempel på det är när ett företag är skyldig att utföra kontroller som kräver personuppgifter. Ett exempel på det är banker som, oavsett avtal med kunder, måste genomföra övervakning som förhindrar penningtvätt.

   Om man ska hävda berättigat intresse måste ändamålen vara mycket tydligt formulerade, avgränsade och dokumenterade.

   IMY listar, förvånande nog, direktreklam som ett tänkbart berättigat intresse.

2. - Se kursboken.
   - Se kursboken.
   - Se kursboken.
   - En vy kan beskrivas som en virtuell tabell, dvs den är inte explicit lagrad i databasen, men den representerar resultatet av en utsökning med SELECT. En vy kan vara temporär eller permanent och används som om det vore en vanlig tabell.
   - En transaktion säkerställer att ändringar i databasen görs atomiskt och säkert. Inga ändringar i databasen har genomförts förrän transaktionen är fullständig (med ”COMMIT TRANSACTION”). Du är därför garanterad att tex insättningar av data, med komplicerade beroenden, är fullständigt genomförda efter transaktionen.

3. ER-diagram för schemat

4. SQL-uttryck.

   (a)
```
SELECT "Antal oavgjorda", COUNT(*) FROM GameResult
WHERE home_goals = away_goals;
```

```
SELECT COUNT(*)
  FROM Game AS G
    INNER JOIN GameResult AS GR ON GR.game_id == G.
        game_id
  WHERE G.league = "Allsvenskan"
    AND home_goals = away_goals;
```

(b)
```
SELECT arena, COUNT(*)
  FROM Team
  INNER JOIN Game ON Team.abbreviation = Game.
      home_team
  INNER JOIN GameResult ON GameResult.game_id =
      Game.game_id
  GROUP BY arena;
```

(d)
```
SELECT on_time, on_extra_time, 1.0 * on_extra_time
     /(on_time + on_extra_time) AS Andel
FROM (SELECT COUNT(*) AS on_time
        FROM Goals
        WHERE goal_minute <= 90),

      (SELECT COUNT(*) AS on_extra_time
        FROM Goals
        WHERE goal_minute > 90);
```

5. (a) Primärnycklar — tabell och rekommenderad primärnyckel:
   - Team: `abbreviation` (rimligt att anta att förkortningen är unik och alltid existerar).
   - League: `league_name` (ty samma ”organiser” kan tänkas ha flera ligor).
   - Game: `game_id`
   - GameResult: `game_id`
   - Player: `player_id`
   - Goals: `game_id`, `player_id`, `goal_minute` (flera mål per match och spelare, så mer än ett attribut behövs).

   (b) En match kan inte införas utan att det finns två lag som spelar den (man kan tänka sig andra antaganden: en final kan planeras innan man vet vilka finalisterna är) och då vill man kräva att `home_team` och `away_team` är finns i databasen. Det görs med **`FOREIGN KEY`** `( home_team, away_team)REFERENCES Team (abbreviation).`

   (c) Lägg in ett semantiskt villkor i GameResult: **`CHECK`** `home_goals >= 0` **`AND`** `away_goals >= 0`.

   (d) SQL-standarden tillåter att man skapar villkor i databasen (”assertions”) för integritetsvillkor av detta slag. SQLite stödjer dock inte den tekniken, utan kräver att man sätter villkor på tabellerna eller deras kolumner. Dessa villkor tillåter dock inte att man lägger in SELECT-uttryck hur som helst! Denna detalj tappade jag bort när jag skapade frågan, så ger poäng på ett generöst sätt här.

6. (a) Ett schema är på 2NF om
   - schemat är på 1NF och
   - ”varje icke-nyckelattribut är fullständigt funktionellt beroende av varje kandidatnyckel”.

   Man kan förklara det intuitivt med att attribut inte får vara funktionellt beroende av delnycklar. Primärnyckeln ska helst bestämma hela raden.

Exempel: Om man har alla attribut i tabellen Goals som primärnyckel och lägger till ett nytt attribut, spelarnummer, så är inte tabellen på 2NF längre. Spelarnummret är bestämt av `player_id`, inte av de andra attributen.

(b) För 3NF ska schemat vara på 2NF och inget icke-nyckelattribut vara fullständigt funktionellt beroende (FFB) av ett annat icke-nyckelattribut.

I det schema vi tittar på i denna tenta kan man titta på tabellen Game och lägga till ett attribut som är FFB av ett icke-nyckelattribut. Game bör ju ha `game_id` primärnyckel. Om vi lägger till attributet ”organiser” (från tabell League) så är organiser FFB av ”league”. Därmed bryter man mot 3NF.