

# Tentamen 2004-08-16

## DATABASTEKNIK - 1DL116, 1MB025

Datum .....Måndagen den 16 Augusti, 2004  
Tid .....14:00-19:00  
Jourhavande lärare ...Kjell Orsborn, tel. 471 11 54 eller 070 425 06 91  
Hjälpmedel .....miniräknare

### Anvisningar:

- Läs igenom hela skrivningen och notera eventuella oklarheter innan du börjar lösa uppgifterna. Förutom anvisningarna på skrivningsomslaget så gäller följande:
  - Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng och oklara formuleringar kan dessutom misstolkas.
  - Antaganden utöver de som står i uppgiften måste anges. Gjorda antaganden får förstås inte förändra den givna uppgiften.
  - Skriv endast på en sida av papperet och använd ett nytt papper för varje uppgift för att underlätta rättning och minska risken för missförstånd.
- För godkänt krävs det cirka 50% av maxpoäng.

## 1. Databasterminologi:

2p

Förklara följande begrepp i ett databassammanhang.

- (a) meta-data

Svar: Meta data, or the database schema, include data about data, i.e. a description of the database stored in the system catalog. Meta-data consist of information about structure of files, type and storage format of each data item, various constraints on the data and other types of information about data such as authorization privileges and access statistics. For the relational model this include descriptions of the relation names, attribute names, data types, primary keys, secondary keys, foreign keys, other constraints, views, storage structures and indexes, and security and authorization information.

- (b) normalisering (eng. normalization)

Svar:

Normalisering är en formell metod för att analysera relationscheman som baseras på nycklar och funktionella beroenden mellan attribut. Relations-scheman kan genom en serie tester normaliseras till någon av de normal-former som existerar (1nf, 2nf, 3nf, bcnf, etc.).

Om ett schema ej uppfyller kraven för en viss nf så skapas nya delschema genom att dela upp attributen i originalschemat så att dessa nya delschema kan uppfylla villkoren. Avsikten är bland annat att undvika partiell eller transitiva funktionella beroenden som kan orsaka olika uppdateringsproblem inkluderande: insättnings-, borttagnings-, och modifieringsproblem. Fördelar med normalisering är att man kan undvika uppdateringsproblem, redundans, och nullvärden.

En nackdel med normalisering är att man får flera relationer att hålla reda på som kan försämra effektiviteten för vissa frågor då fler join-operationer kan behövas.

## 2. Datamodeller:

4p

Hur representeras ER-modellens begrepp entitetstyp och attribut (eng. entity type och attribute) i följande datamodeller?

- (a) relations-datamodell (2p)

Svar: I relationsmodellen representeras en entitetstyp som en relation (tabell) och entitetstypens attribut som attribut i relationen (en kolumn i tabellen).

- (b) objekt-orienterad datamodell (2p)

Svar:

I en objekt-orienterad datamodell representeras en entitetstyp som en klass (eller typ) och attribut som attribut hos klassen.

## 3. SQL:

4p

Antag att vi i en databas har tre relationer (tabeller) med följande relations-scheman:

```
PRODUKT(PID, NAMN, PRIS)
DETALJ(DID, NAMN, TYP, VIKT)
OMFATTAR(PID, DID, ANTAL)
```

, där xID betecknar nycklar. Det finns alltså flera olika produkter med olika pris som består av ett antal detaljer av olika typ och med olika vikt. OMFATTAR är här en flera-till-flera relation.

- (a) Skapa en vy i SQL som omfattar all information från de 3 tabellerna. (2p)

Svar:

```
CREATE VIEW TOTPRODUKT AS SELECT P.PID, P.NAMN, P.PRIS,  
    D.DID, D.NAMN,D.TYP,D.VIKT,O.ANTAL  
FROM PRODUKT P, DETALJ D DETALJ, OMFATTAR O  
WHERE P.PID = O.PID AND  
    D.DID = O.DID;
```

- (b) Formulera följande fråga i SQL: "Vad är totala vikten på varje produkt?"  
(om den anses bestå av de ingående detaljernas sammanlagda vikt). (2p)

Svar:

```
SELECT P.PID, P.NAMN, SUM(D.VIKT * D.ANTAL)  
FROM PRODUKT P, DETALJ D, OMFATTAR O  
WHERE P.PID = O.PID AND  
    D.DID = O.DID  
GROUP BY P.PID;
```

```
SELECT PID, NAMN, SUM(VIKT * ANTAL)  
FROM TOTPRODUKT  
GROUP BY PID;
```

#### 4. Join-operationer:

2p

- (a) Vad är en equijoin (1p)

Svar:

En equi-join är en sammansättning (join), alltså en sammansättning av två relationer till en ny relation i beaktande av ett sammansättningsvillkor, där sammansättningsvillkoret består av ett likhetsvillkor.

- (b) Vad är skillnaden mellan en equijoin och en naturlig join? (1p)

Svar:

Skillnaden mellan en equijoin och en naturlig join är att vid naturlig join så elimineras det andra av de redundanta attributen som ingår i join-villkoret. I en equi-join görs ingen eliminering.

#### 5. Transaktioner och samtidighetskontroll (eng. concurrency control):

4p

- (a) Vilken egenskap garanteras för transaktionerna i ett transaktionsschema som följer ett två-fas låsningsprotokoll (eng. two-phase locking protocol)?  
(endast namnet på egenskapen är ej tillräckligt för full poäng) (2p)

Svar: serialiserbarhet (eng. serializable)

- (b) Vilken egenskap kan normalt ej garanteras för ett transaktionsschema som följer ett två-fas låsningsprotokoll? (endast namnet på egenskapen är ej tillräckligt för full poäng) (2p)

Svar: frihet från dödlig låsning (eng. dead-lock)

#### 6. Frågeoptimering:

5p

- (a) Vad är ett “oklustrat index” (unclustered index)? (1 p)
- (b) Vad är “kostnadsbaserad optimering”? (1 p)
- (c) Vilka är de tre vanligaste join-metoderna? (3 p)

6a: Ett oklustrat index är ett index vars nycklar har annan sorteringsordning än raderna i tabellen.

6b: Kostnadsbaserad optimering optimerar databasfrågor genom att uppskatta kostnaden att utföra olika möjliga frågeexekveringsplaner. Den exekveringsplan med lägst uppskattad kostnad väljs för att utföra frågan. Kostnadsuppskattningarna baseras på statistik om databasen. Kostnadsbaserad optimering används av relationsdatabasoptimerare.

6c: Nested loop join

Sort merge join

Hash join

#### 7. XML-databaser:

4p

- (a) Vad skiljer DTD från XMLSchema? (2 p)
- (b) Vilka är de två viktigaste frågespråken för XML? (2 p)

7a: Både DTD och XMLSchema beskriver innehåll i XML-filer. XMLSchema är betydligt mer semantiskt rikt genom att man har understöd för många olika datatyper, arv och relationer mellan entiteter. DTDn har bara datatypen sträng och är mer likt en grammatik där man beskriver layout av tillåtna dokument. DTDn uttrycks i annat språk än XML, medan XMLSchema uttrycks i XML.

7b: XQuery och Xpath

#### 8. Multimedia-databaser:

4p

- (a) Vilka fördelar har “objekt-relationella” databaser för att lagra multi-media data? (2 p)
- (b) Vad är en BLOB och vad används den till? (1 p)
- (c) Varför är RAID bra för att lagra multi-media data i databaser? (1 p)

8a: Man kan lagra komplexa objekt mot vars struktur man kan ställa effektiva optimerade databasfrågor. Man kan indexera innehållet i objekten. Komplexa objekt kan t.ex. vara filmer kan vara representerade som objekt bestående av scener bestående av bilder innehållande aktörer.

8b: BLOB = Binary Large Object är en datatyp för att lagra stora bitsträngar i relationsdatabaser utan att veta något om själva innehållet i BLOBen (t.ex. foton). Man kan således inte ställa frågor som filtrerar baserat på innehållet i en BLOB.

8c: RAID = Redundant Array of Inexpensive Discs tillåter att filer ligger lagrade i bitar spridda över flera diskar så att man snabbt kan läsa dem parallellt. Överföringstiden av t.ex. bilder blir då snabbare än om allt ligger på en disk.

#### 9. Databas API:n:

3p

- (a) Beskriv skillnaden mellan JDBC och ODBC. (1 p)
- (b) Beskriv arkitekturen av ODBC. Illustrera med bild. (2 p)

9a: JDBC = Java DataBase Connectivity är ett gränssnitt mellan programmeringsspråket Java och SQL-baserade relationella databaser. ODBC = Open DataBase Connectivity är ett programmeringsspråksoberoende gränssnitt SQL-gränssnitt.

9b: Se kompendium on-line.

Lycka till!

/ Kjell och Tore