Uppsala University
Department of Information Technology
Kjell Orsborn, Tore Risch

Tentamen 2007-08-22
DATABASE TECHNOLOGY - 1DL116, 1MB025, 1DL124

```
Date ......................................... Wednesday 22 Aug, 2007
Time ...................................................... 09:00-14:00
Teacher on duty ....... Kjell Orsborn, phone 471 11 54 or 070 425 06 91

Exam aids ................................................. calculator
```

Instructions:

- Read through the complete exam and note any unclear directives before you start solving the questions. The following guide lines hold:
  - Write clear and neat answers! Answers that cannot be read can obviously not result in any points and unclear formulations can be misunderstood.
  - Assumptions outside of what is stated in the question must be explained. Any assumptions made should not alter the given question.
  - Write your answer on only one side of the paper and use a new paper for each new question to simplyfy the correction process and to avoid possible misunderstandings.
- A passing grade requires about 50% of the maximum number of points.

1. **Database terminology:** 4 pts

   Explain the following database concepts:

   (a) transaction (sv. transaktion)

   Answer: A transaction is a logical unit of database processing that is performed in its entirety or not at all.

   (b) secondary index (sv. sekundärindex)

   Answer: Sekundärindex är en ordnad fil av dataposter med 2 fält där första fältet är av samma typ som som indexeringsfältet, dvs vilket fält som helst i datafilen. Andra fältet är en blockpekare. Indexeringsfältet kan vara ett icke-nyckelfält eller ett sekundärnyckelfält och datafilen ej sorterad efter indexeringsfältet. Sekundärindex kan vara glesa eller täta. Index ger en avsevärd effetivisering vid sökning av dataposter. Vid updatering av datafilen måste också tillhörande index uppdateras vilket medför en viss ökad kostnad för dessa operationer.

   (c) recovery (sv. "återhämtning")

   Answer: The process of reconstructing a database back to the last consistent state before a transaction failure.

   (d) BCNF

   Answer: Boyce-Codd's Normal Form states that a relation should, in addition to fulfilling 1st normal form, fulfil that all determinants should be candidate keys. i.e. all non-trivial full functional dependencies should originate from a candidate key.

2. **Data models and the Three-schema architecture:** 4 pts

   Explain and give examples of what is meant by the two concepts:

   (a) *physical* data independence (sv. fysiskt dataoberoende) (2 pts)

   (b) *logical* data independence (sv. logiskt dataoberoende) (2 pts)

   Answer: Fysiskt dataoberoende är förmågan att kunna förändra det interna schemat utan att behöva förändra det konceptuella (ej heller det externa) schemat. Ett exempel kan vara att man vill definera ett index över ett attribut i en tabell för att snabba upp sökningen av vid frågor som inbegriper detta attribut. Dessa frågor behöver alltså inte definieras om för att utnyttja detta index.

   Physical data independence: the possibility to change the internal schema without influencing the conceptual schema. e.g. the effects of a physical reorganization of the database, such as adding an access path, is eliminated.

   Logical data independence: the possibility to change the conceptual schema without influencing the external schemas (views). e.g. add another field to a conceptual schema.

3. **Data integrity (sv. dataintegritet):** 4 pts

   (a) Explain the term referential integrity (sv. referensintegritet) within the relational data model (2 pts)

   Svar: Referensintegritet kräver att om en tupel i en relation refererar till en annan relation så måste den referera till en existerande tupel.

(b) Let $r1(R1)$ and $r2(R2)$ be relations with primary keys (sv. primärnycklar) $K1$ and $K2$ respectively. The attribute fk of the relational schema $R2$ is a foreign key (sv. främmade nyckel) referencing $K1$ of the relational schema $R1$. What tests must be made in order to preserve the referential integrity constraint (sv. bivillkor) during an UPDATE operation. (2pts)

Answer: There are two cases:

Consider relationship set R between entity sets E_1 and E_2. The relational schema for R includes the primary keys K_1 of E_1 and K_2 of E_2. Then K_1 and K_2 form foreign keys on the relational schemas for E_1 and E_2 respectively.

Case 1: If a tuple t_2 is updated in relation r_2 and the update modifiesvalues for the foreign key α, then a test similar to the insert case is made. Let $t_2^{'}$ denote the new value of tuple t_2. The system must ensure that: $t_2^{'}[\alpha] \in \Pi_{K_1}(r_1)$

Case 2: If a tuple t_1 is updated in r_1, and the update modifies values for the primary key (K_1), then a test similar to the delete case is made. The system must compute $\sigma_{\alpha=t_1[K_1]}(r_2)$ using the old value of t_1 (the value before the update is applied). If this set is not empty, the update may be rejected as an error, or the update may be cascaded to the tuples in the set (cascading update), or the tuples in the set may be deleted (cascading delete).

4. **SQL:** 4 pts

Assume that we have a product database consisting of two relations (tables) with the following schemas:

```
PRODUCT(PID,PNAME)
COMPONENT(CID,CNAME,WEIGHT,COST,PID)
```

, where xID's represent keys.

(a) Formulate a query in relational algebra that retrieves the product id and name, the component id and name, and the weight of the product named "Transporter v0.9b". (2pts)

(b) Formulate an SQL query that retrieves the product id, name and the number of components that *each* product consists of. (2pts)

Svar: $\pi_{<PID,PNAME,CID,CNAME,WEIGHT>}$ $(\sigma_{PNAME='Transporterv0.9b'}(PRODUCT \text{x}_{<PID=PID>} COMPONENT))$

```
SELECT P.PID,P.PNAME, COUNT(*) AS NO_OF_COMPONENTS
FROM PRODUCT P, COMPONENT C
WHERE P.PID = C.PID
GROUP BY PID,PNAME

%\begin{verbatim}
%SELECT P.PID,P.PNAME, COUNT(*) AS NO_OF_PARTS
%FROM PRODUCT P, PART C
%WHERE P.PID = C.PID
%GROUP BY PID,PNAME
```

5. **Physical database design:** 4 pts

Explain the organization and functionality of hash-files (hash-filer). The answer should include how to retrieve a data record (sv. datapost) with regard to a specific search key (sv. söknyckel) of the hash-file.

Svar: En hash-fil består av ett statiskt eller dynamiskt antal datablock som hanteras av olika typer av hashningstekniker. Hash-filer hanterar adressering av dataposter till datablock genom att applicera en hash-funktion till hash-fältet (dvs sökfältet) vilken returnerar adressen till ett datablock för insättning eller återsökning av dataposten. En vanlig form av hashfunktion har formen h(f(p)) = f(p) mod M, där hash-funktionen h(f(p)) tillhandahåller addressen för det datablock där dataposten p skall lagras genom att beräkna hashfältet f(p) modulo (mod) antalet datablock M. Man hittar alltså var (i vilket block) en datapost finns för en specifik söknyckel genom att beräkna hash-funktionen för nyckeln som ger adressen till blocket.

6. **Query optimization:** 4pts

(a) How is selectivity (sv. selektivitet) measured? Why is it very important in cost-based query optimization (sv. kostnadsbaserad frågeoptimering)? (1 pt)

(b) What is the worst case complexity of a cost-based query optimizer? (1 pt)

(c) In what language are optimized SQL queries, i.e. execution plans (sv. exekveringsplaner), expressed? (1 pt)

(d) How does the query interpreter (sv. frågeinterpretator) handle very large intermediate results (sv. mellanresultat) produced in an execution plan? (1 pt)

7. **Database APIs:** 4pts

(a) What is the difference between JDBC and ODBC? (1 pt)

(b) What does the "O"in ODBC abbreviate? (1 pt)

(c) How does JDBC handle very large query results? (1 pt)

(d) How can one avoid the high cost of query optimization when using JDBC (1 pt)

8. **Data Warehouses:** 4pts

(a) What does OLAP stand for? (1 pt)

(b) What is a star schema (sv. stjärnschema) and when do they occur? (1 pt)

(c) What is the data cube operator in modern SQL and what is it used for? (1 pt)

(d) Why are data warehouses normally stored in a separate DBMS from an operational database (sv. produktionsdatabas) ? (1 pt)

Good luck and have a great summer!

/ Kjell och Tore