

Tentamen 2005-12-21

DATABASE TECHNOLOGY - 1MB025

Date Wednesday 21 December, 2005
Time 08:00-13:00
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 simplify 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) meta data

Answer: 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) primary index

Answer: Primärindex består av en ordnad fil av dataposter med 2 fält. Första fältet är av samma typ som ordningsfältet (indexeringsfältet) för datafilen och det andra fältet är en pekare till ett datablock (blockpekare). Primärindex är ett gles index då det har en indexpost för varje block i data-filen. Primärindex kräver mycket mindre plats än motsvarande datafil och kan utnyttjas för att snabba upp sökningen av dataposter i datafilen med avseende på indexeringsfältet.

(c) referential integrity (sv. referensintegritet)

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

(d) full functional dependency (sv. fullt-funktionellt beroende)

Answer:

Ett funktionellt beroende så att X bestämmer Y , $X \rightarrow Y$, existerar då om för varje par av tupler $t_1, t_2 \in r(R)$ och för alla $r(R)$ följande gäller: om $t_1[X] = t_2[X]$ så gäller att $t_1[Y] = t_2[Y]$

Fullt funktionellt beroende anger att för ett funktionellt beroende gäller att det inte finns någon delmängd attribut $A \subseteq X$ så att $(X - \{A\}) \rightarrow Y$.

Här gäller att R är ett relationsschema och $r(R)$ är en instans av schemat R med attributen A_1, \dots, A_n och $X, Y \subseteq \{A_1, \dots, A_n\}$.

Alltså ett fullt funktionellt beroende är ett funktionellt beroende som inte innehåller något onödigt attribut i determinanten (vänsterledet i beroendet).

2. Data models - three schema architecture:

4 pts

Explain and give examples on what is meant by the concepts *physical* and *logical data independence*.

Answer: The three-schema architecture introduces a multi-level architecture where each level represents one abstraction level - in 1978 the "standard" architecture (ANSI/SPARC architecture) for databases was introduced. It consists of 3 levels where each level introduces one abstraction layer and has a schema that describes how representations should be mapped to the next lower abstraction level:

1) The internal level or internal schema - describes storage structures and access paths for the physical database. Abstraction level: files, index files etc. Is usually defined through the data definition language (DDL) of the DBMS.

2) Conceptual level or conceptual schema - an abstract description of the physical database. Constitute one, for all users, common basic model of the logical content of the database. This abstraction level corresponds to "the real world":

object, characteristics, relationships between objects etc. The schema is created in the DDL according to a specific data model.

3) External level, external schemas, or views - a typical DB has several users with varying needs, demands, access privileges etc. External schemas describes different views of the conceptual database with respect to what different user groups would like to/are allowed to see. Some DBMSs have a specific language for view definitions (else the DDL is used).

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. Physical database design - indexing:

4 pts

(a) Describe the structure of B+ trees

Answer: Ett B+-träd består av en balanserad trädstruktur med en ordnad mängd noder med sökvärden: en rot, inre noder och lövnoder. Ordnings-talet p på trädet avgör storleken på trädet där inre noder har som minst $p/2$ och som mest p trädpekare (utom roten).

Vidare har en inre nod med q st pekare (where $q < p$) $q - 1$ st sökvärden. Interna noder består av en ordnad mängd av pekare p och sökvärden k enligt $\langle p_1, k_1, p_2, k_2, \dots, p_{q-1}, k_{q-1}, p_q \rangle$, där $k_1 < k_2 < \dots < k_q$. För alla sökvärden x i ett delträd pekat på från p_i gäller att $k_{i-1} < x \leq k_i$ för $1 < i < q$; $x \leq k_i$ för $i = 1$; och $k_{i-1} < x$ för $i = q$.

Lövnoderna har strukturen $\langle \langle pr_1, k_1 \rangle, \langle pr_2, k_2 \rangle, \dots, \langle pr_{q-1}, k_{q-1} \rangle, pr_{next} \rangle$ där $q < p$, varje pr_i är en datapekare (eller datablock-pekare) och pr_{next} pekar vidare till nästa lövnod. Löven finns alla på samma nivå och innehåller minst $p/2$ värden. (svaret kan ges en mycket mer kortfattat form mha figurer).

(b) Describe the principle for retrieval of data records by using a B+ tree index.

4. SQL and relational algebra:

4 pts

Assume that we have a literature database where there are two relations (tables) with the following schemas:

```
BOOK(BID, BNAME)
CHAPTER(CID, CNAME, LENGTH, BOID)
```

, where xID's represents keys.

(a) Formulate a query in relational algebra that retrieves book id, book name, chapter id, chapter name and the length of the chapters for the book "Guide Uppsala". (2p)

(b) Formulate an SQL query that retrieves the book id, book name, and the number of chapters for each book, i.e. how many chapters each book consists of. (2p)

Answer: $\pi_{\langle BID, BNAME, CID, CNAME, LENGTH \rangle}$
($\sigma_{BNAME='GuideUppsala'}(BOOKX_{\langle BID=BOID \rangle}CHAPTER)$)

```
SELECT B.BID, B.BNAME, COUNT(*) AS NO_OF_ CHAPTERS
FROM BOOK B, CHAPTER C
WHERE B.PID = C.BOID
GROUP BY BID, BNAME
```

5. **Concurrency control (sv. samtidighetskontroll):**

4 pts

- (a) Describe the principles for lock management (sv. låshantering) for transactions controlled by a *two-phase locking protocol* (sv. två-fas låsningsprotokoll).

Answer: En transaktion sägs följa ett två-fas låsningsprotokoll om alla låsningsoperationer föregår den första upplåsningsoperationen (unlock) i transaktionen. Alltså en sådan transaktion genomgår en expanderande fas där nya lås kan utfärdas men inga lås kan släppas; och en krympande fas där existerande lås kan låsas upp men inga nya lås kan erhållas.

- (b) What important properties can be stated by transaction schedules controlled by two-phase locking protocols?

Answer: Två-fas låsningsprotokoll garanterar serialiserbara transaktions-scheman men garanterar ej frihet från deadlocks.

6. **Object-oriented and object-relational databases:**

4 pts

- (a) What is the purposes of the three main kinds of user-definable database extensibility mechanisms available in an *object-relational* database system (sv. 'objektrelationella databaser')?
- (b) Which of the above extensibility mechanisms are lacking or weak in a first generation object-oriented database system, i.e. in *object stores* (sv. 'objektlager')?

7. **Query optimization:**

4 pts

- (a) What is *selectivity* (sv. 'selektivitet') and how is it used in cost-based query optimization?
- (b) What is the worst case complexity of cost-based query optimization and how does one avoid this cost?
- (c) Why does cost-based query optimization pay off despite its complexity?
- (d) How are operators in execution plans different from relational algebra operators? Give examples.

8. **ODBC:**

4 pts

- (a) What is ODBC?
- (b) What is the difference between JDBC and ODBC?

- (c) What is an ODBC driver (sv. 'drivrutin')?
- (d) What is the purpose of the ODBC driver manager?

Good luck and a Merry, Merry Christmas!

/ Kjell och Tore