

Tentamen 2006-04-19

DATABASE TECHNOLOGY - 1MB025

Date Wednesday 19 April, 2006
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 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) relational schema (sv. relationsschema)
- (b) secondary index (sv. sekundärindex)
- (c) deadlock (sv. "dödlig låsning")
- (d) BCNF

2. **Data models:**

4 pts

Explain the concepts primary key (sv. primärnyckel) of the relational data model and object identifier (sv. objektidentifierare) of an object data model and their most important properties.

Answer: En primärnyckel är en minimal supernyckel, utvald bland kandidatnycklarna att utgöra nyckel för en relation. En minimal supernyckel består av en minimal delmängd av relationens attribut som unikt identifierar alla tupler i relationen.

En objektidentifierare är en unik och ofta logisk och systemgenererad identifierare som används för att unikt identifiera objekt under hela dess existens, samt för att hantera referenser mellan objekt.

De huvudsakliga skillnaderna mellan en P.N. och en OID är att en P.N. unikt identifierar tupler (rader) i en tabell medan en OID unikt identifierar objekt i hela databasen. Vidare så är en OID alltid systemgenererad, ändrar aldrig värde och används internt för att referera till objekt. En P.N. utgörs normalt av en eller flera attribut (kolumner) från en relation, de väljs av databaskonstruktören och kan följaktligen ändras eller korrigeras.

3. **Conceptual modeling:**

4 pts

In enhanced entity-relationship modeling (sv. utökad entitets-relations-modellering) one can specify certain constraints for a specialization and generalization (sv. specialisering och generalisering). Explain in this context the following concepts:

- (a) disjointness constraint
- (b) completeness constraint

Answer:

- (a) disjointness constraint innebär att subklasserna till en superklass kan specificeras till att vara *disjunkta*, dvs att en instans kan endast vara medlem av en av dessa subklasser, eller *överlappande* (eng. overlapping), dvs disjunkthetsvillkoret krävs så att instanser tillåts vara medlemmar till fler än en av subklasserna.
- (b) completeness constraint specificerar klassificeringen till att vara *total* eller *partiell*. Om klassificeringen är total så måste instanserna till superklasserna vara medlem i någon subclass och för en partiell klassificering behöver ej instanserna till superklassen vara medlem av någon subclass.

Notera att villkoren a), b) är ortogonala, dvs de är oberoende av varann så att alla kombinationer av dessa kan existera.

4. **Transactions:** 4 pts

Describe the properties that one would like transactions to fulfill in a database context (hint: ACID).

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

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

6. **Object-oriented and object-relational databases:** 4 pts

- (a) Describe the three main kinds of user-definable database extensibility mechanisms available in an *object-relational* database system (sv. 'objektrelational databas')? (3pts)
- (b) What is swizzling and what is it used for? (1pt)

7. **Query optimization:** 4 pts

- (a) What is *selectivity* (sv. 'selektivitet') and how is it used in a cost-based query optimizer? (1pt)
- (b) What is the worst case complexity of cost-based query optimization? (1pt)
- (c) Why does cost-based query optimization pay off despite its complexity? (1pt)
- (d) How are operators in execution plans different from relational algebra operators? Give examples. (1pt)

8. **Database APIs:** 4 pts

- (a) What is JDBC? (1 pt)
- (b) How does JDBC handle the high cost of dynamic query optimization? (1 pt)
- (c) What does 'O' in 'ODBC' stand for? (1 pt)
- (d) How does JDBC handle queries that return very large result sets? (1 pt)

Good luck!

/ Kjell och Tore