Uppsala University
Department of Information Technology
Kjell Orsborn, Tore Risch

Final Exam 2006-12-20

DATABASE TECHNOLOGY - 1MB025, 1MD026, 1DL116, 1DL124

```
Date ......................................... Wednesday, Dec 20, 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 guidelines 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:** 2 pts

   Explain the following database concepts:

   (a) entity integrity constraint
   entitetsintegritet (eng. entity integrity) Answer: To preserve entity integrity, guaranteeing that all tuples in a relation can be uniquely identified, it is required that no primary key is assigned a NULL value.

   (b) participation constraint
   Answer: The participation constraint states if an entity has to be a member of a relationship type or not. Total participation states that all entities has to have at least one relationship of the type and partial participation means that not all entities must have a relationship of the type.

2. **Database system architecture:** 4 pts

   Describe the three-schema architecture for database management systems and explain how it supports different forms of 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 se. Some DBMSÕs 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. **Enhanced entity-relationship modeling:** 6 pts

   Explain the concepts specialization, generalization and aggregation (sv. specialisering, generalisering och aggregering) within enhanced entity-relationship modeling.

Answer:

Specialization is a process to conceptually refine a general entity type called a superclass by specifying a set of subclasses. The subclasses are created by identifying some distinguishing characteristics among subsets of entities of the superclass that is the basis to form the subclasses. Possible example ...

Generalization is a process to specify a superclass by identifying a number of common characteristics among a set of (sub)classes. These characteristics can be extracted and defined to form the attributes in a common superclass, where these characteristics can be inherited by the subclasses. Possible example ...

Aggregation is an abstraction concept to group entities into composite objects from their components. In three cases can aggregation be related to the EER model. The 1st case is an aggregation of attribute values of an object to form the whole object. The 2nd case is the representation of an aggregation relationship using an ordinary relationship. The 3rd case is not explicitly supported in EER but involve the possibility to combine related objects using a particular relationship instance into a higher-level aggregate object. Possible examples ...

4. **Normalization:** 4 pts

Explain:

(a) *partial* functional dependency (sv. *partiellt* funktionellt beroende) and (2pts)
(b) *transitive* functional dependency (sv. *transitivt* funktionellt beroende). (2pts)

Answer:

A *partial* functional dependency is a functional dependency, $X \to Y$ where some attribute $A \in X$ can be removed from X and the dependency still holds, i.e. for some $A \in X, (X - \{A\}) \to Y$.

A *transitive* functional dependency is a functional dependency, $X \to Y$ where there is a set of non-prime attributes Z and both $X \to Z$ and $Z \to Y$ hold.

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

Describe the basic principles of *external* hashing and how it can be used to store and retrieve data records in files.

Answer:

Hashing for disk files is called external hashing. The hash function maps a key into a relative bucket number. A table in the file header converts the bucket number into a block address (see Figure 13.9 in Elmasri/Navathe). A typical hash function has the following form: $h(K) = K \bmod M$, where M is the number of buckets that the file is divided into. Overflow buckyes and chaining can be used to solve bucket overflows. The insert a record in the file, the hash function is applied to the hash field of the record and the bucket number where to insert the record is returned. To search for a record with a specfic value of the hash field works similarly by applying the hash function to the value and getting the bucket number where the record is stored in return.

6. **Query optimization:** 4 pts

(a) What is selectivity and why is it needed in cost-based query optimization? (1p)

(b) What is the worst case complexity of cost-based query optimization and how does one avoid this cost as user? (1p)

(c) Why does cost-based query optimization pay off despite its complexity? (1p)

(d) Give examples of two operators in an 'execution plan' that are not in the relational algebra. (1p)

Answer:

6a: Selectivity is the probability that a predicate is true. I.e. it measures how selective a predicate is. It is used to estimate the cost of executing a query or a part of a query.

6b: Cost-based query optimization is exponential over the number of joins in a query. The cost is avoided by pre-compiling queries (prepare).

6c: The number of joins in a query is much smaller than the size of the databases (the number of rows there).

6d:

```
Index scan
Sorting
Segment scan
Stop after n
Remove duplicates
Hash join
Merge join
Nested loop join
```

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

(a) What three kinds of user-definable database extensibility mechanisms are available in an object-relational database system? (2pts)

(b) Which of the above extensibility mechanisms are lacking or weak in a simple object-oriented database system (i.e. in an object store)? (2pts)

Answer:

7a:
User defined data types
User defined indexes
User defined query optimization

7b:
User defined query optimization

8. **Data warehouses:** 4 pts

A university needs to analyze the quality of its education and wants to use data warehousing technology. They want to summarize grades, courses, and students using a data cube containing grades of students per course, year, and department.

(a) Define a star schema to represent the data cube in a relational database. (2pts).

(b) Define a data qube query to summarize the average grades per department over years 2002-2006. (2pts)

Answer:

8a:
`course(cid,cname, year, dept, ...)` dimension
`student(ssn,sname, ...)` dimension
`grade(ssn, cid, score)` fact table

8b:

```
select course.dept,course.year,avg(grade.score)
from grade, student, course
where course.year >= 2002
  and course.year <= 2006
  and grade.ssn = student.ssn
  and grade.cid = course.cid
group by dept,year
with cube
```

Good Luck and Merry, Merry Christmas!

/ Kjell och Tore