

# Tentamen 2004-04-14

## DATABASTEKNIK - 1DL116, 1MB025

Datum .....Onsdagen den 14 April, 2004  
Tid .....8:00-13:00  
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. **Database terminology:** 4pts

Concisely explain the following concepts (in a database context):

(a) primary key

Answer: A primary key is a minimal super key that is chosen among the candidate keys to become the key of a relation. A minimal super key consists of a minimal subset of the attributes of a relation that uniquely identifies all tuples in the relation. 1p

(b) 3NF

Answer: A relation schema is in 3NF if it is in 2NF and if no non-key attribute in the schema is fully functional dependent of any other non-key attribute. 1p

2. **Data models:** 4pts

Explain, and give examples of, what is meant by the two concepts *physical* and *logical data independence* that can be accomplished through the three-schema architecture.

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 definiera 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.

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. **Relational model - integrity constraints:** 4pts

Explain in the context of the relational model the following concepts:

(a) entity integrity

Answer: Entitetsvillkor uttrycker att ingen primärnyckel får anta värdet NULL så att alla tupler i en relation kan identidieras unikt. 2p

(b) referential integrity

Answer: Referensintegritet kräver att om en tupel i en relation refererar till en annan relation så måste den referera till en existerande tupel. 2p

4. **Relational algebra and SQL:** 2pts

Express the following query in two variants, with and without using a nested subquery, with the help of the relational schema below:

Find the names of all warehouses that have greater areas than some warehouse located in Uppsala.

```
warehouse (wh-name,wh-city,area)
```

Answer:

```
select distinct T.wh-name
from warehouse as T, warehouse as S
where T.area > S.area and S.wh-city = "Uppsala"
```

```
select wh-name
from warehouse
where area > some (select area
                   from warehouse
                   where wh-city = "Uppsala")
```

**5. Transactions:**

4pts

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

Answer: To preserve the integrity of data, the DBMS must ensure ACID properties:

Atomicity (atomic or indivisible): a logic processing unit (all operations of the transaction) is carried out in its whole or not at all.

Consistency (preservation): a correct execution of a transaction in isolation should preserve the consistency of the database (from one consistent state to another).

Isolation: Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions. The updates of a transaction shall be isolated from other transactions until after the commit point.

Durability (or permanency): If a transaction completes successfully, the changes it has made to the database must persist and should not be lost in a later system failure.

**6. Security and Authorization:**

4pts

- (a) How is authorization specified in modern relational databases?
- (b) Why are views useful for authorization?
- (c) When can a user transfer authorization rights to another user?
- (d) What is 'access matrix'?

Answer:

a) by GRANT/REVOKE SQL commands that associates access rights with tables and views.

b) Views make it possible to specify access rights covering subsets of tables or combinations of tables. E.g. dynamic access rights can be specified where the accessible data depends on row values in tables.

c) A user can transfer access rights if those were granted to him using the annotation WITH GRANT OPTION.

d) The access matrix is a 2D system object  $S \times O \rightarrow R$  stating what access rights  $R$  a subject  $S$  has for object  $O$ . Subjects are e.g. users, accounts, roles. Objects are e.g. tables, views. Rights are e.g. READ, WRITE, UPDATE.

**7. Object-Oriented and Object-Relational Databases:** 4pts

(a) What are the three most important kinds of user-definable database extensibility mechanisms available in an object-relational database system? (3 p)

(b) Which one of the above extensibility mechanisms is lacking in an object-oriented kind of database system (an 'object store')? (1 p)

Answer:

a) User-defined functions (methods).

User-defined storage representations.

User-defined query optimization methods.

b) User-defined query optimization methods

**8. Active Databases:** 4pts

(a) What are the kinds of problems where ECA rules should *not* be used? Motivate why not. (2p)

(b) Give an example of a problem where ECA rules should be used. Motivate why. (2p)

Answer:

a) Where queries can be used instead

Where updates can be used instead

Where stored procedures can be used instead

b) Problems where a certain action should always be executed whenever a certain kind of update of the database is made. The action should be independent on who makes the update.

For example, a travel agent might maintain a waiting list and whenever someone (e.g. other agent) cancels a suitable booking the system should automatically book someone from the waiting list.

**9. Query Processing:** 4pts

We have a table

PERSONS(SSN, NAME)

SSN is key and the table is clustered on SSN. There is a B-tree index on NAME. Given the query

```
select SSN from PERSON where NAME = "KALLE"
```

- (a) What two execution plans are possible? (1 p)  
(b) Give exact formula stating which plan is faster in terms of parameters of the physical representation of the table in the database. When are they equally fast? (3p)

Answer:

a)

```
project(SSN,select(RowScan(PERSON),NAME="KALLE"))  
project(SSN,IndexScan(PERSON.NAME="KALLE"))
```

b)

N number of rows in PERSON  
B number of rows per table block  
I number of rows per index block  
d distinct values of NAME,  $d \leq N$ ,  
Assume m rows match KALLE,  $m = N/d$  on average.

Cost of reading block from disk is 1.  
Number of blocks in table is  $N/B$ .  
Therefore depth of index is  $\log_I(d)$  ( $\log_I = I$ :th logarithm)

Plan 1:  $N/B$  (must scan it all since there can be  $> 1$  KALLE)  
Plan 2:  $\log_I(d) + m = \log_I(d) + N/d$   
(assume one KALLE in index, m in rows)

Equally fast when  $N/B = \log_I(d) + N/d$   
This can happen when m is large, i.e d small and  
thus many persons in the table are called KALLE.

Good luck!

/ Kjell och Tore