UiT
THE ARCTIC
UNIVERSITY
OF NORWAY

Faculty of science and technology

EXAMINATION QUESTION PAPER

Exam in:	INF-2700 Database Systems
Date:	Wednesday 22.02.2017
Time:	Kl 09:00 - 13:00
Place:	Theorifagbygget, Hus 3, 3.416
Approved aids:	None
Type of sheets (squares/lines):	Digital exam
Number of pages incl. cover page:	6
Contact person during the exam:	Weihai Yu
Phone:	41429077

NB! It is not allowed to submit scratch paper along with the answer sheets. If you do submit scratch paper, it will not be evaluated.

Question 1 (40%)

Below are some database tables with example data for an online show application.

- People

pid	name
p01	anna
p02	jan
p03	hanna
p04	ole
p05	tom

- Shows

sid	title
s01	dance
s02	talk
s03	concert

- Performers

sid	**pid**
s01	p01
s01	p02
s02	p03
s03	p01
s03	p04

- Watchers

sid	**pid**
s01	p01
s01	p03
s01	p05
s03	p02
s03	p05

The *primary keys* of the tables are in **bold** text.

Foreign key in Performers:

- sid: references sid of Shows
- pid: references pid of People

Foreign key in Watchers:

- sid: references sid of Shows
- pid: references pid of People

Write queries to find the required information.

Queries 1–5 must be formulated in *both relational algebra and SQL*.

Queries 6–10 need only be formulated in *SQL*.

Note: In the result tables of your SQL queries, there should be *no* identical (duplicate) rows.

Relational algebra *and* SQL (1–5):

1. The titles of all shows.
   The result for the example database is:

title
dance
talk
concert

2. Names of all performers.
   The result for the example database is:

name
anna
jan
hanna
ole

3. Names of the watchers of the 'dance' show.
   The result for the example database is:

name
anna
hanna
tom

4. Titles of shows that nobody watches
   The result for the example database is:

title
talk

5. Titles of shows that are watched by some of their own performers.
   Display also the the performers who watch the show.
   The result for the example database is:

title	name
dance	anna

SQL *only* (6–10):

6. Number of showss

   The result for the example database is:

numberOfShows
3

7. Titles of shows and the number of their watchers, in descending order of the numbers.

   You do not have to display the shows that are not watched.

   The result for the example database is:

title	numberOfWatchers
dance	3
concert	2

8. Titles and performers of solos.

   A *solo* is a show performed by a single person.

   The result for the example database is:

title	name
talk	hanna

9. Titles of shows that have more watchers than performers.

   Display also the numbers of performers and wathers.

   The result for the example database is:

title	numberOfPerformers	numberOfWathers
dance	2	3

10. Names of people who watched all shows performed by 'anna'.

    The result for the example database is:

name
tom

Question 2 (20%)

Now consider the physical data organization of the database in Question 1.

Suppose that our online show application gets very popular. There are millions of shows. Some shows are watched by millions of people, while some are only watched by very few people.

We consider the file organization of table `Watchers`. We focus now on the operation to find the watchers of a given show.

Should we organize the file based on hash or B^+-tree?

Answer the following questions.

1. What is the primary performance overhead of database systems in general?
2. Describe how table `Watchers` is organized with hash on `sid`.
3. Describe how table `Watchers` is organized as a B^+-tree with `sid` as the search key.
4. Discuss which file organization you would choose for table `Watchers`.
   Focus on the performance of the operation to find the watchers of a given `sid`.
   Take into account the fact that the numbers of watchers of shows can vary from very few to millions.

Question 3 (20%)

Answer the following questions. Please explain the relevant concepts while answering the questions.

1. What is a relation schema in *Boyce-Codd Normal Form* (BCNF)?
2. Given the relation schema $R(A, B, C), F = \{A \to B, B \to C\}$.
   Is the schema in BCNF?
3. Given the relation schema $R'(A, B, C), F' = \{A \to B, B \to A, B \to C\}$.
   Is the schema in BCNF?
4. What is a *lossless decomposition* of a relation schema?
5. Use one of the schemas R or R' above to discuss why a lossless BCNF decomposition is useful.

Question 4 (20%)

1. What is an *ACID transaction*?
2. Describe a concurrency control mechanism based on *timestamp ordering*.
3. Discuss what this concurrency control mechanism achieves.

–END–