

EXAMINATION QUESTION PAPER

Exam in:	INF-2700 Database Systems
Date:	Thursday 01.03.2018
Time:	09:00 - 13:00
Place:	Bodega/K1.04
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 a messaging archive application.

- Users

Uid	Name	Country
u1	Eva	Sweden
u2	Ole	Norway
u3	Ida	Norway
u4	Eva	Norway
u5	Ed	USA

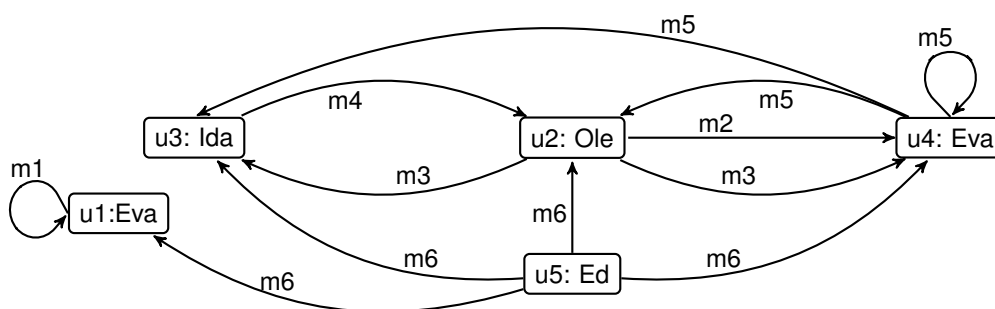
- Messages

Mid	Author	Content	Date
m1	u1	test new account	2018-02-28
m2	u2	Are you ready?	2018-02-28
m3	u2	Good luck!	2018-03-01
m4	u3	thanks! you, too.	2018-03-01
m5	u4	Thank you! I need it.	2018-03-01
m6	u5	You win!	2018-03-01

- Recipients

Mid	Uid	Read
m1	u1	true
m2	u4	false
m3	u3	true
m3	u4	true
m4	u2	false
m5	u2	false
m5	u3	false
m5	u4	false
m6	u1	true
m6	u2	true
m6	u3	false
m6	u4	false

The figure illustrates the association between messages and users.



In the tables, the *primary keys* of the tables are in **bold** text.

Foreign key in Messages:

- Author: **references** Uid of Users.

Foreign keys in Recipients:

- Mid: **references** Mid of Messages.
- Uid: **references** Uid of Users.

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. All countries of messaging users.

The result for the example database is:

Country
Sweden
Norway
USA

2. Content of all messages on 2018-03-01

The result for the example database is:

Content
Good luck!
thanks! to you, too.
Thank you! I need it.
You win!

3. Dates and content of messages with at least one recipient who is not the author himself/herself.

The result for the example database is:

Date	Content
2018-02-28	Are you ready?
2018-03-01	Good luck!
2018-03-01	thanks! to you too.
2018-03-01	Thank you! I need it.
2018-03-01	You win!

4. Dates and content of messages that are *only* sent to the author himself/herself.

The result for the example database is:

Date	Content
2018-02-28	test new account

5. Names and Countries of people who did not send any message on 2018-03-01.

The result for the example database is:

Name	Country
Eva	Sweden

SQL *only* (6–10):

6. Number of different countries.

The result for the example database is:

NumberOfCountries
3

7. Content, author names and dates of unread messages to “Eva” from “Norway”. List in the ascending order of the dates.

The result for the example database is:

Content	Name	Date
Are you ready?	Ole	2018-02-28
Thank you! I need it.	Eva	2018-03-01
You win!	Ed	2018-03-01

8. List of `Mids` of messages and the numbers of recipients of the messages, in descending order of the numbers.

The result for the example database is:

Mid	NumberOfRecipients
m6	4
m5	3
m3	2
m1	1
m2	1
m4	1

9. Messages on 2018-03-01 with at least 3 recipients. List the `Mids` and the numbers of recipients of the messages.

The result for the example database is:

Mid	NumberOfRecipients
m5	3
m6	4

10. Names and Countries of people who sent a message to everybody except himself/herself.

The result for the example database is:

Name	Country
Ed	USA

Question 2 (20%)

Now consider the physical data organization for the message archiving database.

When the application becomes popular, there are a huge amount of messages archived.

We decide to organize the table Recipients with hash on `Uid`. In addition, there is a hash index on `Mid`.

Answer the following questions.

When discussing query performance, you should make reasonable assumptions of data sizes.

1. Sketch how Recipients data are organized with a figure and some brief description.
2. What is the primary performance overhead of database systems in general?
3. Given the `Uid` value of a user, how to find the `Mids` of the messages sent to the user?
What is the performance overhead of the search?
4. Given the `Mid` value of a message, how to find the `Uids` of recipients of the message?
What is the performance overhead of the search?

Question 3 (20%)

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

1. What is *functional dependency* $X \rightarrow Y$ of a relation instance r ?
For the example instance of table `Users` in Question 1, check if the following functional dependencies are satisfied.
 - a) $Name \rightarrow Country$
 - b) $Country \rightarrow Name$
 - c) $\{Name, Country\} \rightarrow Uid$
 - d) $Uid \rightarrow \{Name, Country\}$
2. What is a *superkey* of a relation schema?
Can you define a superkey using functional dependencies?
3. Given a relation schema $R = ABCD$ with functional dependencies $\{AB \rightarrow CD, D \rightarrow B\}$.
Is AB a super key of R ? Is AC a super key of R ? Is AD a super key of R ?
Why and why not?
4. What is a *candidate key* of a relation schema?
Can you find *all* candidate keys of the above schema R ?
5. Why is the concept of functional dependency useful in the design of a database schema?
Explain with the schema R .

Question 4 (20%)

1. What is an *ACID transaction*?
2. What is a *log* for transaction processing?
How is a log organized?
What are the operations on a log?
What are the performance costs of these operations?
3. What is *write-ahead logging* (WAL)?
4. Describe an implementation of write-ahead logging.
5. Describe briefly how to *commit* a transaction with a single disk write.
Explain why it works.

—END—