Introduction to Databases

IT University of Copenhagen

January 16, 2006

This exam consists of 5 problems with a total of 16 questions. The weight of each problem is stated. You have 4 hours to answer all 16 questions. The complete assignment consists of 6 numbered pages (including this page), *plus an answer sheet to be used for several of the questions.*

If you cannot give a complete answer to a question, try to give a partial answer. You may choose to write your answer in Danish or English. Write only on the front of sheets, and remember to write your CPR-number on each page. Please start your answer to each question at the top of a *new* page. Please order and number the pages before handing in.

GUW refers to *Database Systems – The Complete Book* by Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom, 2002.

All written aids are allowed / Alle skriftlige hjælpemidler er tilladt.

1 Database design (25%)

The academic world is an interesting example of international cooperation and exchange. This problem is concerned with modeling of a database that contains information on researchers, academic institutions, and collaborations among researchers. A researcher can either be employed as a professor or a lab assistant. There are three kinds of professors: Assistant, associate, and full professors. The following should be stored:

- For each researcher, his/her name, year of birth, and current position (if any).
- For each institution, its name, country, and inauguration year.
- For each institution, the names of its schools (*e.g. School of Law, School of Business, School of Computer Science,...*). A school belongs to exactly one institution.
- An employment history, including information on all employments (start and end date, position, and what school).
- Information about co-authorships, i.e., which researchers have co-authered a research paper. The titles of common research papers should also be stored.
- For each researcher, information on his/her highest degree (BSc, MSc or PhD), including who was the main supervisor, and at what school.
- For each professor, information on what research projects (title, start date, and end date) he/she is involved in, and the total amount of grant money for which he/she was the main applicant.

a) Draw an E/R diagram for the data set described above. Make sure to indicate all cardinality constraints specified above. The E/R diagram should not contain redundant entity sets, relationships, or attributes. Also, use relationships whenever appropriate. If you need to make any assumptions, include them in your answer.

b) Convert your E/R diagram from question a) into relations, and write SQL statements to create the relations. You may make any reasonable choice of data types. Remember to include any constraints that follow from the description of the data set or your E/R diagram, including primary key and foreign key constraints.

2 Normalization (15%)

We consider the following relation:

`Articles(ID,title,journal,issue,year,startpage,endpage,TR-ID)`

It contains information on articles published in scientific journals. Each article has a unique ID, a title, and information on where to find it (name of journal, what issue, and on which pages). Also, if results of an article previously appeared in a “technical report” (TR), the ID of this technical report can be specified. We have the following information on the attributes:

- For each journal, an issue with a given number is published in a single year.
- The `endpage` of an article is never smaller than the `startpage`.
- There is never (part of) more than one article on a single page.

The following is an instance of the relation:

ID	title	journal	issue	year	startpage	endpage	TR-ID
42	Cuckoo Hashing	JAlg	51	2004	121	133	87
33	Deterministic Dictionaries	JAlg	41	2001	69	85	62
33	Deterministic Dictionaries	JAlg	41	2001	69	85	56
39	Dictionaries in less space	SICOMP	31	2001	111	133	47
57	P vs NP resolved	JACM	51	2008	1	3	99
77	What Gödel missed	SICOMP	51	2008	1	5	98
78	What Gödel missed	Nature	2222	2008	22	22	98

a) Based on the above, indicate for each of the following sets of attributes whether it is a key for `Articles` or not. Use the answer sheet of the exam for your answer.

1. {`ID`}; 2. {`ID,TR-ID`}; 3. {`ID,title,TR-ID`}
4. {`title`}; 5. {`title,year`}; 6. {`startpage,journal,issue`}

If you wish, you may additionally write a brief explanation for each answer, which will be taken into account, but is not necessary to get full points.

b) Based on the above, indicate for each of the following potential functional dependencies, whether it is indeed an FD or not. Use the answer sheet of the exam for your answer.

1. `ID` → `title`; 2. `startpage` → `endpage`; 3. `journal issue` → `year`
4. `title` → `ID`; 5. `ID` → `startpage endpage journal issue`; 6. `TR-ID` → `ID`

If you wish, you may additionally write a brief explanation for each answer, which will be taken into account, but is not necessary to get full points.

c) Based on a) and b), perform normalization into BCNF, and state the resulting relations.

3 SQL and relational algebra (35%)

We consider again the relation `Articles` from problem 2.

a) Indicate for each of the following expressions whether it is a valid SQL statement or not. A valid statement, as described in GUW, should be accepted by a standard SQL interpreter, whereas an invalid statement should result in an error message. Use the answer sheet of the exam for your answer.

1. `SELECT * FROM Articles WHERE endpage-startpage>10;`
2. `SELECT * FROM Articles WHERE endpage-startpage<0;`
3. `SELECT SUM(name) FROM Articles;`
4. `SELECT AVG(year) FROM Articles WHERE title LIKE 'C%';`
5. `SELECT COUNT(*) FROM Articles GROUP BY year;`
6. `SELECT year,COUNT(*) FROM Articles WHERE COUNT(*)>10 GROUP BY year;`

b) Indicate for each of the following queries, how many tuples would be returned if it was run on the instance of `Articles` from problem 2. Use the answer sheet of the exam for your answer.

1. `SELECT ID FROM Articles WHERE year<2006;`
2. `SELECT DISTINCT ID FROM Articles WHERE year<2006;`
3. `SELECT AVG(year) FROM Articles GROUP BY journal;`
4. `SELECT ID FROM Articles WHERE title LIKE '%d';`

Consider the relations `Authors(auID,name)` and `Authoring(articleID,authorID)`, containing information on names of authors, and who is authoring which papers, respectively.

c) Write an SQL query that returns for each article, its ID, title and the number of authors.

d) Write an SQL query that returns the titles of articles authored by '`Robert Tarjan`'.

e) Write an SQL query that returns the number of co-authors of '`Robert Tarjan`'. (I.e., the number of authors who have written at least one article together with him.)

f) Write SQL statements that correspond to the following two relational algebra expressions. Duplicate elimination should be performed.

1. $\pi_{\texttt{title,authorID}}(\sigma_{\text{year}=2005}(\texttt{Articles}))$
2. $\gamma_{\texttt{year,COUNT(ID)}}(\texttt{Articles})$

4 Efficiency and transactions (15%)

Consider the following six queries on `Articles` from problem 2:

```
1.  SELECT title FROM Articles WHERE year=2005;
2.  SELECT title FROM Articles WHERE endpage=100;
3.  SELECT title FROM Articles WHERE year>1995 AND year<2000;
4.  SELECT title FROM Articles WHERE journal='JACM' AND issue=55;
5.  SELECT title FROM Articles WHERE issue=55 AND journal='JACM';
6.  SELECT title FROM Articles WHERE endpage-startpage>50;
```

a) Indicate which of the above queries would likely be faster (based on the knowledge you have from the course), if *all* of the following indexes were created. Use the answer sheet of the exam for your answer.

```
CREATE INDEX Idx1 ON Articles(year,startpage);
CREATE INDEX Idx2 ON Articles(startpage,endpage);
CREATE INDEX Idx3 ON Articles(journal,issue,year);
```

In the following we consider the below transactions on the `Authors(auID,name)` relation.

Time	User A	User B
1	`INSERT INTO Authors VALUES (42,'Donald Knuth');`	
2		`INSERT INTO Authors VALUES (43,'Guy Threepwood');`
3		`DELETE FROM Authors WHERE name LIKE 'Don%';`
4		`INSERT INTO Authors VALUES (44,'Donald E. Knuth');`
5	`DELETE FROM Authors WHERE name LIKE 'Guy%';`	
6	`COMMIT;`	
7		`COMMIT;`

b) Suppose that `Authors` is initially empty, that the transactions are run at isolation level `READ COMMITTED`, and that the commands are issued in the order indicated above. What is the content of `Authors` after the execution?

c) Suppose that `Authors` is initially empty. What are the possible contents of `Authors` after each *serial* execution of the two transactions?

5 Constraints (10%)

Suppose that the Authoring relation of problem 3 relation was created as follows:

```
CREATE TABLE Authoring(
  articleID INT REFERENCES Article(ID) ON DELETE SET NULL,
  authorID INT REFERENCES Author(ID) ON DELETE CASCADE
)
```

a) Indicate which of the following statements are true, and which are not. Use the answer sheet of the exam for your answer.

1. If we try to delete a tuple from Authoring, the tuple is not deleted. Instead, articleID is set to NULL.
2. If we delete a tuple from Authoring, any tuples in Author referred to by this tuple are also deleted.
3. If we delete a tuple from Article, some attributes of Authoring may have their values set to NULL.
4. If we try to insert a tuple into Author, with an ID that is not referred to in Authoring, the operation is rejected.
5. If we try to insert a tuple into Authoring, with an ID that does not exist in Author, the operation is rejected.

b) Write CHECK constraints for Articles of Problem 2 that ensure the following:

1. Values of the journal attribute does *not* start with 'Journal'.
2. The value of the endpage attribute is never smaller than that of startpage.
3. The value of year is given in full (e.g. 1999 is not abbreviated as 99). You may assume that year is of type integer, and that there are no articles more than 200 years old.

Answer sheet (to be handed in)

Name		**Page number**	
CPR		**Total pages**	

Instructions. For all questions except 3.b (which asks for numbers), you must place exactly one X in each column. Note that the grading will be done in a way such that random answering does not pay. For example, two correct answers and one incorrect answer will be worth the same as one correct answer and two question marks.

Question 2.a	**1**	**2**	**3**	**4**	**5**	**6**
Key						
Not a key						
?						

Question 2.b	**1**	**2**	**3**	**4**	**5**	**6**
FD						
Not an FD						
?						

Question 3.a	**1**	**2**	**3**	**4**	**5**	**6**
Valid						
Invalid						
?						

Question 3.b	**1**	**2**	**3**	**4**
Number of tuples				

Question 4.a	**1**	**2**	**3**	**4**	**5**	**6**
Faster						
Same						
?						

Question 5.a	**1**	**2**	**3**	**4**	**5**
True					
False					
?					