

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science
APRIL 2012S EXAMINATIONS

PLEASE HAND IN

CSC 343 H1S
Instructor: Horton
Duration — 3 hours
Examination Aids: None

Student Number:

Family Name(s):

Given Name(s):

Do not turn this page until you have received the signal to start.
In the meantime, please fill in the identification section above and
read the instructions below.

In the SQL questions, you are welcome to use views.

Comments are not required, although they may help us mark your answers.

There is a blank page at the end for rough work. If you want any of it
marked, indicate that clearly there, as well as in the question itself.

1: _____/12

2: _____/ 6

3: _____/ 6

4: _____/10

5: _____/12

6: _____/ 1

7: _____/11

8: _____/ 8

9: _____/ 8

10: _____/ 6

11: _____/16

TOTAL: _____/96

Question 1. [12 MARKS]

We used the schema below many times in lecture. A course's cName attribute gives its full name, such as "Introduction to Databases", while its CID is a value like "csc343". The WR attribute indicates whether or not the course satisfies a writing requirement.

Relations

- Students(SID, surName, campus)
- Courses(CID, cName, WR)
- Offerings(OID, CID, term, instructor)
- Took(SID, OID, grade)

Integrity constraints

- Offerings[CID] \subseteq Courses[CID]
- Took[SID] \subseteq Students[SID]
- Took[OID] \subseteq Offerings[OID]

Write the following queries in Relational Algebra. Use only the basic operators $\Pi, \sigma, \bowtie, \times, \cap, \cup, -, \rho$.

1. Report all instructors who have taught two or more different courses (with different CID values) in the term "fall 2011" but none in the term "winter 2012".

2. Report the following for every course: the course CID and name, the maximum grade ever given in that course (in any offering of it), and the minimum grade ever given in that course (in any offering of it).

Question 2. [6 MARKS]

The following SQL can be input into postgresql without errors. Suppose we have done so.

```
CREATE TABLE B (  
    five INT,  
    six INT,  
    seven INT UNIQUE  
);  
  
CREATE TABLE A (  
    one INT CHECK (one > 0),  
    two INT CHECK (two > 0),  
    three INT CHECK (three > 0),  
    four INT CONSTRAINT ok CHECK (  
        four < one + two + three  
    ),  
    PRIMARY KEY (one, two)  
);  
  
INSERT INTO B VALUES  
(51, 52, 53),  
(29, 28, 27),  
(44, 44, 44);  
  
INSERT INTO A VALUES  
(1, 2, 29, 1),  
(1, 5, 44, 11),  
(8, 6, 51, 51);
```

1. Suppose we were to do this:

```
insert into A values (4, 5, 29, null);
```

Would this cause an error? YES No

Explain why or why not.

2. Suppose we were to do this:

```
insert into B values (5, 6, null), (5, 6, null);
```

Would this cause an error? YES No

Explain why or why not.

3. Suppose we were to do this:

```
insert into A values (2, null, 6, 18);
```

Would this cause an error? YES No

Explain why or why not.

Question 3. [6 MARKS]

Consider the following schema:

```
create table A (
    num int,
    name text primary key
);

create table B (
    p int,
    q int
);
```

Answer the questions below about various pairs of queries on this schema.

1. Query 1:

```
select num
from A
where num > any (select q from B);
```

Query 2:

```
select num
from A
where not exists
    (select * from B where num <= q);
```

Do the queries always give the same result? YES NO

If they do, explain why. If they do not, give a very small example of data that demonstrates this: draw table A and table B, the result table for Query 1 and the result table for Query 2.

2. Query 1:

```
select num
from A
where num in (select q from B);
```

Query 2:

```
select num
from A, B
where num = q;
```

Do the queries always give the same result? YES NO

If they do, explain why. If they do not, give a very small example of data that demonstrates this: draw table A and table B, the result table for Query 1 and the result table for Query 2.

3. Query 1:

```
select name
from A, B
where num = p;
```

Query 2:

```
select distinct name
from A, B
where num = p;
```

Do the queries always give the same result? YES NO

If they do, explain why. If they do not, give a very small example of data that demonstrates this:
draw table A and table B, the result table for Query 1 and the result table for Query 2.

Question 4. [10 MARKS]

Below is a SQL version of the schema we saw in Question 1. It is a little different in than it separates the CID attribute into two attributes: dept and cNum.

```
create table Students(  
    sID integer primary key,  
    surName varchar(15) not null,  
    campus text);  
  
create table Courses(  
    cNum integer,                -- E.g., 343  
    name varchar(40) not null,   -- E.g., 'Introduction to Databases'  
    dept text,                   -- E.g., 'CSC'  
    wr boolean,  
    primary key (cNum, dept));  
  
create table Offerings(  
    oID integer primary key,  
    cNum integer,  
    dept text,  
    term integer not null,  
    instructor varchar(40),  
    foreign key (cNum, dept) references Courses);  
  
create table Took(  
    sID integer,  
    oID integer,  
    grade integer,  
    primary key (sID, oID),  
    foreign key (sID) references Students,  
    foreign key (oID) references Offerings);
```

Write the following SQL queries.

1. For each course, report the course number and department, the instructor who has taught the most offerings of that course, and the number of offerings that he or she taught. If there is a tie, report them all. If a course was never taught, report null for the instructor and number.

Here is the schema again, for reference:

```
create table Students(  
    sID integer primary key,  
    surName varchar(15) not null,  
    campus text);  
  
create table Courses(  
    cNum integer,           -- E.g., 343  
    name varchar(40) not null, -- E.g., 'Introduction to Databases'  
    dept text,             -- E.g., 'CSC'  
    wr boolean,  
    primary key (cNum, dept));  
  
create table Offerings(  
    oID integer primary key,  
    cNum integer,  
    dept text,  
    term integer not null,  
    instructor varchar(40),  
    foreign key (cNum, dept) references Courses);  
  
create table Took(  
    sID integer,  
    oID integer,  
    grade integer,  
    primary key (sID, oID),  
    foreign key (sID) references Students,  
    foreign key (oID) references Offerings);
```

2. Let's say that a course has level "junior" if its cNum is between 100 and 299 inclusive, and has level "senior" if its cNum is between 300 and 499 inclusive. Report the average grade, across all departments and course offerings, for all junior courses and for all senior courses. Report your answer in a table that looks like this:

level	levelavg
junior	
senior	

Each average should be an average of the individual student grades, not an average of the course averages.

Question 5. [12 MARKS]

Suppose the following XML file is called `another.xml`.

```
<?xml version="1.0" standalone="yes"?>
<a p="hello">
  <b x="1" y="5"/>
  <c n="100">
    <d real="true">no way</d>
    <d real="false">yes way</d>
    <d real="false">possibly</d>
  </c>
  <b x="3" y="1"/>
  <b x="2" y="6"/>
  <c n="52">
    <d real="false">truly</d>
  </c>
  <c n="50">
    <d real="true">really</d>
    <d real="true">actually</d>
  </c>
</a>
```

Each of the queries in this question runs without errors. Show the output. We will not be grading the whitespace in your answer, so format it as you wish. Note: There are five queries on three pages.

1. `fn:doc("another.xml")/a[2]`

2. `fn:doc("another.xml")/a/c[@n<55]/d`

```
3. let $d := fn:doc("another.xml")
   let $thing := $d/a/c
   let $other := $thing/d
   return <answer>{$other}</answer>
```

```
4. let $d := fn:doc("another.xml")
   return
   <outer>
     { for $me in $d/a/b[@x < 2]/@y
       for $you in $d/a/c
       where $you/@n < 100
       return <huh whatever="{ $me}"> { $you/d} </huh>
     }
   </outer>
```

For convenience, here is the contents of file `another.xml` again:

```
<?xml version="1.0" standalone="yes"?>
<a p="hello">
  <b x="1" y="5"/>
  <c n="100">
    <d real="true">no way</d>
    <d real="false">yes way</d>
    <d real="false">possibly</d>
  </c>
  <b x="3" y="1"/>
  <b x="2" y="6"/>
  <c n="52">
    <d real="false">truly</d>
  </c>
  <c n="50">
    <d real="true">really</d>
    <d real="true">actually</d>
  </c>
</a>
```

Show the output of this final query.

```
5. let $d := fn:doc("another.xml")
   return
   <both>
     <one> {$d//c[d>"t"]} </one>
     <two> {$d//c/d[.>"t"]} </two>
   </both>
```

Question 6. [1 MARK]

For one point, make me laugh. If you write anything here you will get this point. (And you can use the rest of the page for rough work. If you want any of your rough work marked, make sure that is very clearly indicated.)

Question 7. [11 MARKS]

Suppose we have a file called `recipes.dtd` containing this:

```
<!ELEMENT RECIPES (RECIPE)+>
<!ELEMENT RECIPE (INGREDIENTS, STEPS)>
<!ATTLIST RECIPE name CDATA #REQUIRED>
<!ATTLIST RECIPE type CDATA #IMPLIED>
<!ELEMENT INGREDIENTS (INGREDIENT)+>
<!ELEMENT INGREDIENT (NAME, QUANTITY)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT QUANTITY EMPTY>
<!ATTLIST QUANTITY amount CDATA #REQUIRED>
<!ATTLIST QUANTITY units CDATA #REQUIRED>
<!ELEMENT STEPS (STEP)+>
<!ELEMENT STEP (#PCDATA)>
```

Part (a) [1 MARK]

Define a small xml file that could be successfully validated with respect to this DTD and includes one recipe, with one ingredient and one step.

Part (b) [10 MARKS]

Write the following queries in XQuery.

1. Return the name of all recipes in file `recipes.xml` that include an ingredient named “oranges” and an ingredient named “bacon”.
2. For each recipe in file `recipes.xml`, return the recipe name and the total amount of “butter” and “oil” in its ingredient list. In your total, include only amounts whose unit is “ml”. (Yes, this will vastly understate the total for some recipes, but it makes the question much easier!)

Question 8. [8 MARKS]

Consider the relation R on attributes $ABCDE$, with the following functional dependencies:

$$AB \rightarrow DE, \quad C \rightarrow E, \quad D \rightarrow C, \quad E \rightarrow A.$$

Use the BCNF decomposition algorithm to produce a schema in BCNF.

Label your steps and show your rough work (it will be marked). At the bottom of the next page, write your final answer.

Below, write your final answer. Include the relations and their functional dependencies.

Question 9. [8 MARKS]

Consider relation $R(A, B, C, D, E, G)$ with functional dependencies F .

$$F = \{C \rightarrow AD, \ G \rightarrow B, \ AC \rightarrow D, \ CG \rightarrow BAE\}$$

Part (a) [2 MARKS]

Compute all keys for R .

Part (b) [4 MARKS]

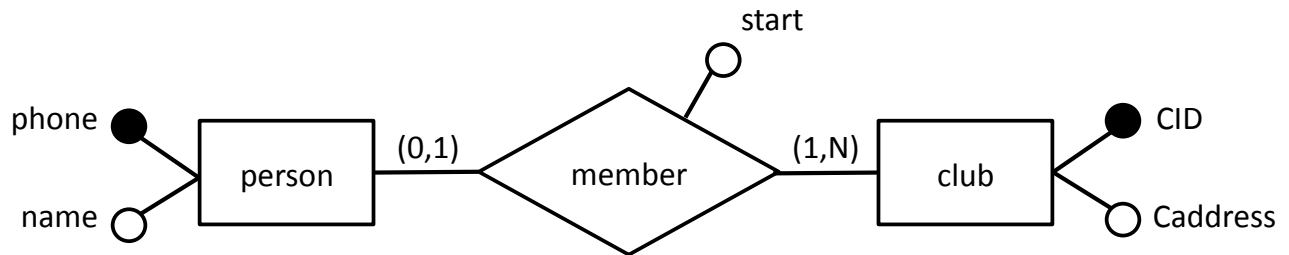
Compute a minimal basis for F .

Part (c) [2 MARKS]

Using the minimal basis from part (b), employ the 3NF synthesis algorithm to obtain a lossless and dependency-preserving decomposition of relation R into a collection of relations that are in 3NF. Don't forget to ensure that a key is included in one of the relations.

Question 10. [6 MARKS]

Consider the following ER diagram:



Which of these cardinalities is possible? **Don't guess.** There is 1 mark for each correct answer and -0.5 for each incorrect answer. The minimum mark is 0.

person	member	club	Is it possible?	
5	0	8	YES	No
5	7	8	YES	No
5	0	5	YES	No
5	10	5	YES	No
11	3	4	YES	No
11	9	4	YES	No

Question 11. [16 MARKS]

In this question you will design a schema for a database about the music industry. This is the information that needs to be represented:

- A musical group has a name (which must be unique), a genre of music (*e.g.* hip hop or classical), and one or more musicians who are members of the group.
- A musician has an unique ID, a name, and a date of birth.
- A musical group may be represented by an “agent”.
- An agent has unique ID, a name, and a phone number. An agent may represent any number of groups, including none.
- A venue is a place where music is played, such as the Mod Club, or the Air Canada Centre. It has a unique ID, a name, and a capacity (the number of people it can hold).
- Agents book musical groups to play. Each booking is for a particular group, in a particular venue, on a particular date, and for a particular fee.

Note: A group can be booked to play by an agent who doesn’t represent them. You don’t have to do anything to prevent this.

Part (a) [8 MARKS]

Design an Entity-Relationship Model (ER Diagram) for this information. Clearly indicate primary keys by using solid circles, or by underlining attribute names, as the textbook does. Show the cardinalities with which an entity participates in a relationship with a pair of the form (minimum, maximum). Alternatively, you can use the arrow notation of the textbook.

Part (b) [8 MARKS]

Translate your Entity-Relationship Model (ER Diagram) into a logical model (DB Schema). For each relation in your schema, provide its name, attributes and keys (underlined attributes).

*[Use the space below for rough work. This page will **not** be marked, unless you clearly indicate the part of your work that you want us to mark.]*

Total Marks = 96