UNIVERSITY OF TORONTO AT SCARBOROUGH

FINAL EXAMINATION Fall 2016

CSCC43H – Introduction to Databases
Instructor – Sina Meraji

Duration – 3 hours

No aids allowed

The final exam is worth 50% of your final mark. In order to pass the course, you must earn at least 40% on the test. Please answer all questions in the space provided. You may use the blank pages dispersed throughout the exam for rough work. Try to be **concise** in your answers.

Good luck!

Last Name	
First Name	
Student Number	

Marks

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Total
RA	SQL	Embedded SQL	ER Model/ DB Design	Design Theory	XML Query Languages	Open Questions	True/ False	
12	**15**	**10**	**15**	**10**	**10**	**8**	**20**	**100**

(left blank)

[12 Marks] Question 1 (Relational Algebra)
Assume the following relational schema that represents customers renting movies:

Movies (<u>MovieID</u>, Title, Director, Studio)
Customer (<u>CustomerID</u>, Name, Address, Age, Balance)
Rented (<u>CustomerID</u>, <u>MovieID</u>)

Answer the following queries in **Relational Algebra**.

[3marks] Q1: Find the titles of the movies directed by "Tim Burton" or produced by the "Universal" studio.

[3marks] Q2: Find the title of movies that have been rented by at least two customers of age 40.

[3marks] Q3: Find the customerID(s) of the customers over 50 years old that have the minimum balance.

[3marks] Q4: Find the pairs of names of customers that have rented the movie 'Godfather'.

[15 Marks] Question 2 (SQL Queries)

Assume the following relational schema that represents drinking habits:

Beers (<u>beername</u>, manufacturer)
Bars (<u>barname</u>, address, license)
Drinkers (<u>drinkername</u>, address, phone)
Likes (<u>drinkername</u>, <u>beername</u>)
Sells (<u>barname</u>, <u>beername</u>, price)
Frequents (<u>drinkername</u>, <u>barname</u>)

The following foreign key constraints hold:

- Likes (drinkername) $\subseteq$ Drinkers(drinkername)
- Likes (beername) $\subseteq$ Beers(beername)
- Sells (barname) $\subseteq$ Bars(barname)
- Sells (beername) $\subseteq$ Beers(beername)
- Frequents(drinkername) $\subseteq$ Drinkers(drinkername)
- Frequents (barname) $\subseteq$ Bars(barname)

Answer the following queries in **SQL**.

[3 Marks] Q1: Find the name of drinkers whose phone has exchange "333", for example the phone number “647-***333***-8787” would qualify.

[3 Marks] Q2: Find the name of bars that serve "Stella" beer at the same price that "Joe's Bar" charges for beer with name "Bud".

[3 Marks] Q3: Find the name and manufacturer of beers that customer “John” likes.

[3 Marks] Q4: Find the name of beer(s) sold for the lowest price.

[3 Marks] Q5: Find the address of bars that have more than 50 frequent drinkers.

[10 Marks] Question 3 (Embedded SQL)

[7 Marks] Part A

Consider the existence of a table "my*Cars*" in your PostgreSQL database "*car*". Write server side code (java/jdbc) that connects to the database, and submits SQL queries in order to determine whether the table *myCars* is populated with some tuples. If there are tuples in *myCars* then delete all of them; if there are no tuples drop the table *myCars* from the database. The script finally closes the database connection.

Notes:

- Your code should work correctly for any database instance.
- Marking will be tolerant on java syntax errors, so it's fine if you miss the exact name of a java method.

[3 Marks] Part B

The following JDBC code snippet illustrates the use of a *commit* and *rollback* object in a transaction. Assume that the empty table **Owner (oid, fname, lname)** exists in the database before execution of the code.

```
…
try{
   //Assume a valid connection object conn
   conn.setAutoCommit(false);

   Statement stmt = conn.createStatement();

   // Submit a well-formed SQL statement
   String SQL = "INSERT INTO Owner VALUES (1, 'Tony', 'Montana')";
   stmt.executeUpdate(SQL);

   //Submit a malformed SQL statement that breaks
   String SQL = "INSERTED VALUES (2, 'Michael', Corleone')";
   stmt.executeUpdate(SQL);

   // If there is no error.
   conn.commit();
}catch(SQLException e){
   // If there is any error rollback.
   conn.rollback();
}
…
```

Q: What is the state of the owner table after execution of the code above (fill in rows)?

[15 Marks] Question 4 (E/R and DB Design)

General Information

A university needs to develop an electronic system in order to administer information about who works in which research project. Here is the information that is provided to you by the university to help you design the database of the system:

- Professors have an SIN, a name, a rank, and a research specialty.
- Projects have a project number, a sponsor name (e.g., NSERC), a starting date, an ending date, and a budget.
- Graduate students have an SIN, a name and a degree program (e.g., M.S. or Ph.D.).
- Each project is managed by one professor (known as the project's principal investigator).
- Professors can manage and/or work on multiple projects.
- In each project one or more professors are involved (known as the project's co-investigators).
- In each project one or more graduate students are involved (known as the project's research assistants).
- When graduate students work on a project, a professor must supervise their work on the project. Graduate students can work on multiple projects, in which case they will have a (potentially different) supervisor for each one.
- Departments have a department number, a department name, and a main office.
- Departments have a professor (known as the chairman) who runs the department.
- Professors are associated with one or more departments, and for each department that they are associated with, a time percentage is associated with their job.
- Graduate students belong to one department in which they are working on their degree.

Part A [8 Marks]

Design an Entity-Relationship Model (ER Diagram) for the electronic system. Use entities (rectangles), relationships (diamonds) and attributes (ovals) to model the information above. Clearly indicate primary keys (underlined attributes) and the cardinalities with which an entity participates in a relationship (express the participation of an entity to a relationship with a pair of (minimum, maximum) or alternatively, you can use the arrow notation of the textbook).

Part B [7 Marks]

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

[10 Marks] Question 5 (Normal Forms)

Consider the following functional dependencies over the attribute set ABCDEFGH:

$$FD = \{ A \rightarrow E,\ BE \rightarrow D,\ AD \rightarrow BE,\ BDH \rightarrow E,\ AC \rightarrow E,$$

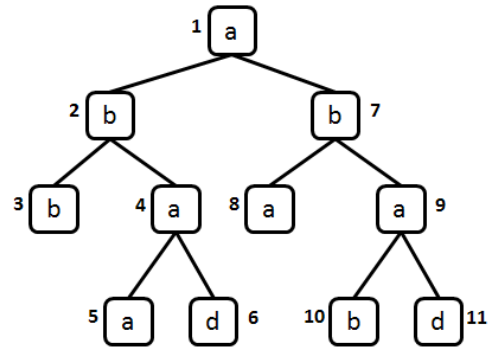
$$F \rightarrow A,\ E \rightarrow B,\ D \rightarrow H,\ BG \rightarrow F,\ CD \rightarrow A\}$$

Find a minimal basis for this set of functional dependencies.

[10 Marks] Question 6 (XML Query Languages)

Assume the xml tree on the side:

For each of the following XPath expression, write which nodes are selected (use the number of nodes, comma-separated).



a) //b

b) //*[.//d]

c) //*[count(.//*)=count(ancestor::*)]

d) //c[position() = last()]

e) //*[preceding-sibling::b]

[8 Marks] Question 7 (Open Questions)

[4 Marks] Q1: Briefly explain how the closure algorithm works?

[4 Marks] Q2: Briefly explain the 3rd normal form and BCNF

[20 Marks] Question 8 (True/False Statements)

For each of the following statements, indicate whether they are *true* or *false*. Please clearly *circle* or underline the answer you think is correct. The marking scheme is as follows:

- **+2**: for correct answer
- **-1**: for incorrect answer
- **0**: for not answering

The minimum mark for this question is 0 (never below).

Answer		**Statement**
TRUE	**FALSE**	Assume a relation **R**(X, Y, Z). In relational algebra, it holds that: $\sigma_{A \vee B}(R) = \sigma_A(R) \cup \sigma_B(R)$
TRUE	**FALSE**	Assume relations **R**(X, Y, Z) and **S**(X, Y, Z). In relational algebra, it holds that: $\sigma_A(R \cup S) = \sigma_A(R) \cup \sigma_A(S)$
TRUE	**FALSE**	In relational algebra **(*bag-semantics*)** the following holds: {a, a, b, b, c, c} ∩ {b, b, c, d} = {c}
TRUE	**FALSE**	In relational algebra **(*bag-semantics*)** the following holds: {a, a, b, b, c} ∪ {b, b, c, d} = {a, a, b, b, c, d}
TRUE	**FALSE**	In relational algebra (*bag-semantics*) the following holds: {a, a, b, b, c} - {b, b, c, d} = {a, a, d}
TRUE	**FALSE**	Prepared statements typically give up efficiency for the sake of security (i.e., there is a trade-off between efficiency and security).
TRUE	**FALSE**	In SQL, it holds that: TRUE AND NULL → NULL
TRUE	**FALSE**	Defining an *index* on an attribute of a relation that is frequently *updated* is a good idea.
TRUE	**FALSE**	Attribute-based checks *cannot* be used to express and enforce foreign-key constraints.
TRUE	**FALSE**	If **|T|** is the join cardinality of an equi-join between two relations **R** and **S**, then: $\|T\| >= \|R\|*\|S\|$

(left blank)

(left blank)