Lab Assignment 1
Released January 21, Due February 4 by 23:59

Dr. Evgenia Ternovska

Important Note: *Students must work individually on this, and other CMPT 354 assignments. You may not discuss the specific questions in this assignment, nor their solutions with any other student. You may not provide or use any solution, in whole or in part, to or by another student. Students are not allowed to publish tests, assignments and their solutions (e.g., put them on an open public repository). You are encouraged to discuss the general concepts involved in the questions in the context of completely different problems. If you are in doubt as to what constitutes acceptable discussion, please ask!*

In this lab assignment, you will develop a database for an Archaeology museum that stores and displays historic artifacts. Each artifact is identified by an artifact number (e.g., CL892), and has its dimensions and weight. Each artifact is made of a specific material – clay, glass, bronze, fabric, gold, etc., and has a recommended optimal storage humidity. A number of conservation and restorations technicians work at the museum. You need to store the name, Social Insurance Number (SIN), address, phone number, and salary of each technician, and specify if she or he is conservationist or restorer, or both. Each technician is an expert on one or more artifact materials (e.g. silver and gold), and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded. There are also museum guides (who are, essentially, educators) working at the museum. Museum guides must have a bi-annual examination to ensure their knowledge is up-to-date. For each guide, you must store the date of the most recent exam. All museum employees (including technicians and guides) belong to a union. You must store the union membership number of each employee. You can assume that each employee is uniquely identified by the Social Insurance Number. The museum conducts a number of tests that are used periodically to ensure that artifacts are stored properly. Each test has a Ministry of Culture test number, a name, and a maximum possible score. The Ministry requires the museum to keep track of each time that a given artifact is inspected by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score that the artifact storage conditions received on the test.

More specifically, you should accomplish the following tasks:

1. Design a database schema to model the domain of the museum as described above. You can use as many relations as you deem necessary. Determine suitable data types for each attribute.

   Specify the keys (and foreign keys, if needed) for relations. It is up to you to decide if there should be any keys, and which attribute (set of attributes) should form a key. Provide the definition of the schema in a simple text file named *LastNameFirstNameSchemaDefinition.txt*, where each relation is defined in the form:

   ```
   RelationName(attribute_1 datatype, attribute_2 datatype, ..., attribute_7,...)

   Keys of RelationName: {attribute_1, attribute_2}, {attribute_7}...
   ```

After defining the schema, include in the file *LastNameFirstNameSchemaDefinition.txt* a short paragraph explaining the schema you created. Focus on explaining what information each of the relations holds, how you chose the data types of the attributes, and how you picked the key constraints. The paragraph should be no longer than 200 words.

2. Write the data definition commands for creating the schema you chose. The commands should generate the tables, as well as the key constraints.

3. Insert at least 10 records into your database using the INSERT INTO command.

Include the commands for tasks 2 and 3 in a text file named *LastNameFirstNameSchema.txt*. You should submit both

- *LastNameFirstNameSchemaDefinition.txt* and
- *LastNameFirstNameSchema.txt.*

The lab assignment report should be submitted through the CourSys system. Your PostgreSQL database should be located in your CMPT354 account in CSIL and your report submitted through CourSys.

Keys, Primary Keys, Foreign Keys in SQL

The following information is for your reference. You don't need to use all of these constructs, just the ones that are needed.

Keys are probably the most basic and very essential database constraints. Collectively the value(s) of the attributes of the key are unique, and thereby uniquely identify the tuple (row) of the relation (table). Any update that effects the values of the key, or violate the key integrity will result in an error state for the DBMS.

```
CREATE TABLE Account (
accnum  VARCHAR(12) UNIQUE,
branch  VARCHAR(30),
custid  VARCHAR(10),
balance NUMERIC(14,2)
);
```

The following insertion gives an error:

```
INSERT INTO Account VALUES
(1, 'London',    'cust1', 100),
(1, 'Edinburgh', 'cust3', 200);
```

The following insertion succeeds:

```
INSERT INTO Account VALUES
(NULL, 'London',    'cust1', 100),
(NULL, 'Edinburgh', 'cust3', 200);
```

Keys consisting of more than one attribute must be declared using a different syntax.

```
CREATE TABLE Movies (
m_title    VARCHAR(30),
m_director VARCHAR(30),
m_year     SMALLINT,
m_genre    VARCHAR(30),
UNIQUE (m_title,m_year)
);
```

This declares the set {m_title,m_year} as a key for Movies.

Primary Keys are essentially UNIQUE + NOT NULL.

```
CREATE TABLE Account (
accnum  VARCHAR(12) PRIMARY KEY,
branch  VARCHAR(30),
custid  VARCHAR(10),
balance NUMERIC(14,2)
);
```

same as

```
CREATE TABLE Account (
accnum  VARCHAR(12) NOT NULL UNIQUE,
branch  VARCHAR(30),
custid  VARCHAR(10),
balance NUMERIC(14,2)
);
```

Foreign keys in SQL.

```
CREATE TABLE Customer (
custid  VARCHAR(10) PRIMARY KEY
name    VARCHAR(20),
city    VARCHAR(30),
address VARCHAR(30)
);
```

```
CREATE TABLE Account (
accnum  VARCHAR(12),
branch  VARCHAR(30),
custid  VARCHAR(10) REFERENCES Customer(custid),
balance NUMERIC(14,2)
);
```

Every value for attribute `custid` in Account must appear among the values of the *key* `custid` in Customer.

General syntax (useful for declaring compound foreign keys):

```
CREATE TABLE <table1> (
<attr> <type>,
...
<attr> <type>,
FOREIGN KEY (<list1>)
REFERENCES <table2>(<list2>)
);
```

where

- `<list1>` and `<list2>` are lists
  with the *same number* of attributes
- attributes in `<list1>` are from table `<table1>`
- attributes in `<list2>` are *unique* in `<table2>`