CSCI 392 – SQL Homework Part 1

Before completing these exercises, download the SQL files to create the university database (shortlinks.hope.edu/392-create-university-db) and insert the data (shortlinks.hope.edu/392-insert-university-data). Open these files in SQL Server Management Studio and execute them.

Part 1 – SQL Queries

Complete each of these exercises by writing SQL queries and executing them. Store your queries in a single file, and denote the beginning of each question using comments (e.g. /* Question 1*/). Be sure the solutions are in order in the file, or you will be marked down.

Put the SQL statement

```
use banking;
```

at the top of the file, before question 1, to select the correct database.

Before question 3, put the SQL statement:

```
use "first.last";
```

(replace first.last with your Vanderplex user name)

You should connect to the server using your SQL server account (firstname.lastname, student ID #). **Upload the file to Moodle when it is complete.**

The exercises below start on page 171 of the textbook:

Question 3.8. For part (b), do not include Smith in the results.

Question 3.15. Here's a hint for part (a): another way of thinking about exists/not exists is that they are testing whether or not a set is empty. For part (c), you should include branches in Brooklyn in the results, as long as they have assets greater than some **other** branch in Brooklyn.

Question 3.2. You should create the grade_points table in your database and fill it with data. For part (c), if a student has not taken any courses, your query should display a value of **null** for the GPA. You may not find any students like this; you can insert one of your own in order to test that your query works correctly.

Question 3.11. For part (b), assume that Spring is the very first semester of any calendar year. For part (d), **include the name of the department in the results,** in addition to displaying the minimum per-department salary.

Question 3.12. For part (b) use **Fall** in place of *Autumn*.

Part 2 – Create table statements

Before getting started on this part of the assignment, you should review the documentation on the `create table` statement in the textbook on page 126. You can find SQL Server specific documentation at shortlinks.hope.edu/392-create-table-syntax. You will need to do a bit of parsing of the syntax specification there if you reference this documentation.

For this part of the assignment, you should create two SQL files. The first should be named `create_videostore_schema.sql` and the second should be named `drop_videostore_schema.sql`.

In the `create_videostore_schema.sql` file you should write `create table` statements to create the tables needed to create the following tables and relationships from the attached ER diagram:

- `Customer`
- `ZipCode`
- `Area`
- `Store`
- `Prefers` – for this one, name the column something other than `Order`, since this is a reserved word in SQL
- `Reserves`

Define appropriate primary key and foreign key constraints for each table. Design the tables to minimize the number of tables created while still avoiding null values wherever possible.

Here are a couple more requirements:

- The ID fields for the `Area`, `Customer`, and `Store` tables should be auto-generated identity columns with a seed value of 1000 and an increment value of 1 (see shortlinks.hope.edu/392-identity-columns for details on using identity columns).
- In addition to defining an appropriate primary key for the `Store` table, you should also add a **table constraint** that ensures the values for the `Order` attribute are unique for each customer (that is, for a particular customer, no two rows should have the same value for the `Order` attribute).

The `drop_videostore_schema.sql` file should contain a sequence of `drop table if exists` statements that drop the tables created in the `create_videostore_schema.sql` file. Note that you need to drop any tables that refer to another table using a foreign key before dropping the referenced table. You can find documentation on the drop table statement at shortlinks.hope.edu/392-drop-table-syntax

You should be able to run your create schema script followed by your drop schema script in succession, leaving your database in the state it started.