Assignment 1

Multi-Table and Complex Queries

Due: Wednesday February 13, 5 p.m.

In this assignment you will construct a number of queries against the *Calgary Weather Database* and the *Mountain Database*.

The Calgary Weather Database

Refer to Assignment 1 for the description of the Calgary Weather Database.

The Mountain Database

The mountain database records information about the mountains of the world such as elevation, location, and some interesting information about the mountain.

The logical structure of the database is:

MOUNTAIN(mountain_id, mountain_name, range_id, elevation, latitude, longitude, notes)

FOREIGN KEY range_id → MOUNTAIN_RANGE(range_id)

MOUNTAIN_RANGE(range_id, range_name, range_continent, notes)

FOREIGN KEY range_continent → CONTINENT(cont_id)

CONTINENT (cont_id, name, area, highest_point, notes)

FOREIGN KEY highest_point → MOUNTAIN(mountain_id)

The database is loaded and available to you in the `mountain` database on the class MySQL instance.

To complete this assignment, you will need to fully understand the structure and contents of the database. You will need to investigate the data, find out what is in the table, how it is formatted and what the columns mean. Do not include your investigative queries in your submission.

All columns in this database have comments associated with them. To view the comments you can use the MySQL command:

```
SHOW CREATE TABLE table_name;
```

This will also show you the primary and foreign keys.

Use the

```
SHOW TABLES;
```

```
DESCRIBE table;
```

and

```
SELECT * FROM ...
```

commands to determine table structure and contents.

You will need to look up MySQL functions and syntax that have not been covered in class or labs.

You may need to look up information about some of the data.

Your Submission

- Your submission should consist of two text files. `A2.sql`, and a log file, `A2.log`, submitted through the Blackboard assignment submission drop box.
- Your SQL statements should be neatly formatted according to the standards shown in class. All queries should have upper case keywords (`SELECT`, `WHERE`, `IN`, `HAVING` etc.) and lower case table and column names, each clause on a separate line, and appropriate, consistent indenting.
- Your file should be commented. As shown in Lab 1, there *MUST* be a block at the top of the file that includes your name, the date, and the course and assignment number.

  In addition, each query should have a comment block labelling it and explaining what it does. Include any assumptions that you may have made while creating the query.
- Each query must be answered with a single SELECT statement. No fair getting procedural.
- You *must* use column and table aliases where appropriate. Results should display with a readable, correct and complete column heading. Where appropriate, units should be included.

Queries

Part I. The Caglary Weather Database

1. `(3 marks)` In Assignment 1 you wrote a query to list the "number of days below -30". Using that query as a basis, write a query to find the average, minimum and maximum number of "days below -30" per year?

2. `(6 marks)` Which day had the coldest minimum temperature and which day had the warmest maximum temperature? List the day and the temperature. This needs to be done with a single query and should return a single line.

3. `(4 marks)` List all of the days that have the same: minimum temperature, maximum temperature, and total precipitation. Only include days with more than 0 total precipitation. Your result should look something like:

```
+-------------+-------------+--------------+------------+
| cw_max_temp | cw_min_temp | total_precip | cw_date    |
+-------------+-------------+--------------+------------+
|       -16.7 |       -22.2 |          1.3 | 1967-01-16 |
|       -16.7 |       -22.2 |          1.3 | 1971-12-21 |
|       -15.6 |       -23.9 |          0.3 | 1962-01-16 |
|       -15.6 |       -23.9 |          0.3 | 1974-03-06 |
|       -13.3 |       -20.6 |          2.5 | 1975-12-04 |
|       -13.3 |       -20.6 |          2.5 | 1976-02-27 |
|       -11.1 |       -18.3 |          0.3 | 1966-12-02 |
|       -11.1 |       -18.3 |          0.3 | 1969-02-06 |
...
```

   Order the result by maximum temperature, minimum temperature, total precipitation and date. There should be no duplicate rows.

4. `(3 marks)` Repeat the question above using a different query technique. Your two queries should return the an identical result. If not, one of them is wrong!

5. `(5 marks)` I ride a bike for transportation. Often people, cyclists and non, will remark that Calgary weather is too inclement for year round bike riding. I always reply that there are a few days where the weather is nasty, but that there are far more good days than bad.

   I need you to find out the truth. Write a query that catagorizes days as "Good", "Poor" or "Stay Home" riding days, based on the conditions, and counts how many there are of each.

   Extreme heat or cold, high winds, heavy snowfall or heavy rain make for poor riding.

"Stay Home" a high less than -20 and more than 10 cm of snow.

"Poor": A low temperature below -20 or a high temperature above +30 or wind gusts over 70 kph or more than 12 cm of snow or more than 50 mm rain.

All other days are "Good".

Use a UNION to create the query. The categories are mutually exclusive so the total "Good" plus the total "Poor" should equal the total number of days in the database.

Your query should display the category, the number of day in that category and the average number of days per year in that category.

Part II. The Mountain Database

1. `(4 marks)` How many mountains are in each mountain range in the database? Show the range name, the continent and the number of mountains. Order the result from the highest to lowest number of mountains, then alphabetically by the name of the mountain range.

2. `(2 marks)` Alter your query to include ranges that do not have any mountains.

3. `(2 marks)` Rewrite the query above using a different technique.

4. `(5 marks)` Which continent has the highest average height of its peaks?

5. `(6 marks)` To answer this question you will need to create what is called a cross tab query. You want to count the high, medium and low peaks on each of three continents, Europe, North America and Asia.

   The end result will be something like this:

```
+----------+------------+--------+------+
| Category | N. America | Europe | Asia |
+----------+------------+--------+------+
| HIGH     |          1 |      1 |    9 |
| MEDIUM   |          6 |      8 |    3 |
| LOW      |         20 |      0 |    0 |
+----------+------------+--------+------+
```

   Low mountains are less than 10,000 feet. Medium mountains are between 10,000 and 20,000 feet and high mountains are more than 20,000 feet.

   The mountain database includes a function, `ft_to_m()` which converts its argument to meters, from feet.

Grading

The marks for each query are given; marks vary based on the complexity of the answer. Queries will be marked on whether or not they produce the correct answer and if appropriate techniques were used.

In addition there will be marks for the following:

- File submission. You should submit two files as specified above. 2 marks.
- SQL script comments. The script is well commented. 4 marks.
- Formating of SQL queries. This should be with consistent use of indents and capitalization. Aliases should be used as appropriate to make the script and the output readable. 4 marks.

The queries are worth 35 and the formatting is worth 10 for a total of 45.