PROG2220: S.Q.L. (MySQL)
Assignment 4

Assignment Type: INDIVIDUAL

Due Date: Week 11 (start of class)

Note: Review the output files provided (XXA04*.out). *Display the question headers in your output files.*

Topics:

- **Tasks 1 and 2: Summary Queries** (Group Functions) and **Subqueries**
  - Important: Use subquery only when it is explicitly required.
- **Task 3: Create DB, Create/Alter Tables, Indexes**

Warm Up: Textbook Exercises

Tasks 1 and 2: Do *all* the **Chapter 6** textbook exercises (pages 184-185) and **Chapter 7** textbook exercises (pages 212-213).

Task 3: Do *all* the **Chapter 11** textbook exercises (page 351). Compare your solution to the textbook exercise solutions under G:\mysql\ex_solutions. Do not submit your textbook exercise solution.

Task 1. Chapter 6 My Guitar Shop (MGS) Database

Save your solution to **XXA04Task1.sql**. Redirect your output to **XXA04Task1.out**.

Assumption: You have MGS database created from G:\mysql\mgs_ex_starts\create_my_guitar_shop.sql (part of Lab 2).

Q1. MGS Exercise 6-1 [3 points]

Write a SELECT statement that returns these columns:

- The count of the number of orders in the Orders table
- The sum of the tax_amount columns in the Orders table

Q2. MGS Exercise 6-2 [4 points]

Write a SELECT statement that returns one row for each category that has products with these columns:

- The category_name column from the Categories table
- The count of the products in the Products table
- The list price of the most expensive product in the Products table

Sort the result set so the category with the most products appears first.

Q3. MGS Exercise 6-6 [3 points]

Write a SELECT statement that answers this question: What is the total amount ordered for each product? Return these columns:

- The product name from the Products table
- The total amount for each product in the Order_Items (Hint: You can calculate the total amount by subtracting the discount amount from the item price and then multiplying it by the quantity)

Use the WITH ROLLUP operator to include a row that gives the grand total.

Q4. MGS Exercise 7-3 [3 points]

Write a SELECT statement that returns the category_name column from the Categories table. Return one row for each category that has never been assigned to any product in the Products table. To do that, **use a subquery** and include the **NOT EXISTS** operator as part of your solution.

Task 2. Software Expert (SWE) Database

Save your solution to **XXA04Task2.sql**. Redirect your output to **XXA04Task2.out**.

Assumption: You have SWE database created from G:\mysql\swexpert\swexpert.sql (part of Lab 3).

Q1. SWE Exercise 1 [2 points]

Display the average evaluation score for consultant ID (EVALUATEE_ID) 105. Round the retrieved value to one decimal place.

Q2. SWE Exercise 2 [2 points]

Count the number of consultants who are certified in skill ID 1.

Q3. SWE Exercise 3 [4 points]

List the first and last name of every consultant who has ever worked on a project with consultant **Mark Myers**. Include **Mark Myers** in the result set. **Use a subquery.** You must also use the consultant's full name in your query.

Q4. SWE Exercise 4 [5 points]

Use the UNION operator to generate a result set consisting of two columns: project ID and name. The result must include all the projects with completed evaluations AND projects managed by consultant with last name that starts with 'Z'. **Use a subquery.** Remove any duplicate results.

Task 3. Software Expert (SWE) Database

Save your solution to **XXA04Task3.sql**. Redirect your standard output to **XXA04Task3.out** and your standard error to **XXA04Task3.err** using the following as a template:

```
-- mysql -u root -p --force < C:\mysql\XXA04Task3.sql 1>
C:\mysql\XXA04Task3.out 2> C:\mysql\XXA04Task3.err
```

The template above will redirect both standard output (port 1) and standard error (port 2) from console to files. The --force option will continue to execute the SQL scripts even when an error occurs.

Assumption: You have SWE database created from G:\mysql\swexpert\swexpert.sql (part of Lab 3).

Q1. SWE Exercise 1 [4 points]

a. Write an ALTER TABLE statement that adds a new 'total_days' column to the **project_consultant** table. This new column should have a default value of 0 (zero).

b. Update the new column with the difference of ROLL OFF and ROLL ON dates.

c. Display all the contents of **project_consultant** table.

d. Drop the 'total_days' column.

Q2. SWE Exercise 2 [4 points]

a. Include statements to drop the table if it already exists.

b. Create a new **evaluation_audit** table with these columns:

- audit_id (primary key, auto increment): integer
- audit_e_id : integer (do not allow nulls)
- audit_score: integer (allow nulls)
- audit_user: text (length of 20; allow nulls)

c. Insert a new row for consultant (e_id) 100 with a score of 90.

d. Display all the contents of **evaluation_audit** table.

Q3. SWE Exercise 3 [5 points]

Modify **evaluation_audit** table as follows:

a. Write a single ALTER TABLE statement that will disallow null values for the audit_user column.

b. Write an ALTER TABLE statement that adds a new 'audit_date' column.

c. Insert another row for consultant 100 with a score of 95 along with the current user and current date. *Hint: Use the USER() and SYSDATE() functions.*

d. Display all the contents of **evaluation_audit** table.

e. Write a *negative test case* by inserting a row for consultant 100 with a score of 99 with an unknown user and date. This will show an error message in the ERR file.

Q4. SWE Exercise 4 [1 point]

Write a *negative test case* by inserting a new row with an unknown (NULL) skill_id to the **project_skill** table.

Q5. SWE Exercise 5 [2 points]

Write a *negative test case* by deleting a row from the consultant table that will violate a foreign key constraint.

Assignment Submissions

Reminder: All printouts must be stapled and submitted **in the correct sequence**!

1. Download and print the **PROG2220_CoverPage.pdf** (PROG2220 Assignment Cover Page and Standards Marking Sheet) posted in eConestoga. **All the sections** of the Cover page must be filled
2. A printout of **A4Marking.pdf**
3. A printout of **XXA04Task1.sql**, where XX is your initials in upper case letters
4. A printout of **XXA04Task1.out**
5. A printout of **XXA04Task2.sql**
6. A printout of **XXA04Task2.out**
7. A printout of **XXA04Task3.sql**
8. A printout of **XXA04Task3.out**
9. A printout of **XXA04Task3.err**