Miscellaneous Topics – Intermediate SQL Queries

This assignment is designed to review a variety of topics and concepts, from indexing, query optimization and regular expressions (RegEx) to window functions and common table expressions (CTEs). In an effort to adequately represent these topics, a number of different databases on both MySQL Community Server and SQL Server Express platforms will be used. All of the databases are briefly described and each question is worth 0.25 points. All the SQL code must be entered in `MySQL7_MiscTopics.sql` file, separated by the existing comments, and each problem should be run in Workbench or SQL Server Management Studio (only for the last problem) independently. The results should be copy/pasted or exported into a single Excel file `Assign8_MiscTopics.xlsx`, one result per sheet.

The Internet Movie Database (IMDB) contains information about most films ever made. The first part of the assignment is based on the IMDB database, attached to the assignment on Canvas in a zipped format, that needs to be installed on your local MySQL server.

1. Use an ALTER TABLE statement to add an appropriate index to minimize the number of examined rows. Use EXPLAIN to compare the number of rows examined, and then remove the index.
   ```
   SELECT * FROM movies_sample
   WHERE release_year BETWEEN 2000 AND 2010
     AND moviename LIKE 'T%';
   ```
2. Optimize the following query assuming the database is in its original state. My initial cost was 80809.599, yours will likely be different.
   ```
   SELECT moviename, release_year
   FROM movies_sample
   WHERE movieid = 476084;
   SHOW STATUS LIKE 'last_query_cost';
   ```
3. You are a fan of James Bond 007 movies. You wonder how many movies out there have a similar title in the sense that they end with a space and three digits. Write a SQL statement that uses a regular expression to search the movies_sample table for all movies that end with the described pattern. Show the movie id, title and release year. (901 rows)
4. In the query for the previous exercise, which 3-digit combination was the most popular in movie titles? Show the top-10 3-digit combinations (must be exactly 3-digit) based on the number of movie titles that end with a space and those 3 digits. Show the 3-digit number, and the number of times it is used in the movie title, order results by number of times from high to low. (366 rows)
5. We want find out how many movies about wars (at least in the literal sense). Using regular expression to find all movies that have a word "wars" in the title. Notice that wars must appear alone as a stand-alone word, it cannot be embedded in other words such as "Warsaw". Show the movie id, title, and release year in the result. (440 rows)

The next two questions use the Order Entry database from the first four assignments. You can use the `OrdEntry_All.sql` file attached to the assignment in Canvas to recreate and repopulate the database.

6. Use three (3) CTEs to find the best and worst customer in terms of total sales generated. Use custFull and custSales parameters for the first CTE named SalesByCust. The BestCust and WorstCust CTEs should have the same 2 parameters as well as custDsgn parameter that will designate the customer as "Best" or "Worst". Use UNION ALL to get both the best and worst customer at the end. (2 rows)
7. Use a recursive CTE to reconstruct the employee supervisory hierarchy. Use employee numbers, first and last name, supervisor number and employee level. The end result must show the employee's full name, level, as well as supervisor's full name. This is a relatively standard problem that should be easy to google, and is also discussed in some detail in the presentation. (7 rows)

The Sakila sample database is one of the most comprehensive MySQL sample databases, and it may already be installed on your local server. The next four (4) questions are based on the Sakila database, attached to the assignment on Canvas in a zipped format, that needs to be installed (if necessary) on your local MySQL server.

8. We are looking for our best customers. Create Customer_Rentals CTE to list the store_id, customer_id, first_name, last_name, and number of rentals by that customer at each store. Then select all the columns from the CTE and include an additional column for the rank of that particular customer at that particular store such that the customer with the most rentals at store 1 earns rank 1 and the customer with the most rentals at store 2 earns rank 1. Include only customers with 35 rentals or more. You must use RANK() window function with an appropriate partition. Then add another column with sequential ranks using DENSE_RANK() function. (37 rows)
9. List the title, store_id, a copy number (starting at 1 for each film and store), and email address of the customer who had possession of that copy on Aug 1, 2005 for all copies of films in the Music category. If no customers had possession of a copy on that particular date, display Not rented for the email address. You must use ROW_NUMBER window function for copy numbers. (232 rows)
10. List the title and rating of all horror movies on inventory for rent at store 1. Include an additional column named Available that indicates, with a Yes or No, whether at least 1 copy of that movie was available for rent in store 1 on Aug 1, 2005. (38 rows)
11. We are curious about our best customers and their favorite movies. List the customer email, film title, and number of times that customer has rented that title. Include only films ranked #1 or #2 for that customer (by number of times rented). In addition, only include in the list those customers that have rented 2 or more films multiple times each. For example, customer Thelma Murray should appear in the list 4 times: she rented one movie 3 times and 3 other movies 2 times each. Customer Yolanda Weaver, on the other hand, should not appear in the list: she only rented 1 move twice. (60 rows)

The last exercise in the assignment need to be completed using the local installation of Microsoft's SQL Server Express as described on the second half of page 2 in the software installation PDF (see `01_Softw_Install.pdf` in `01_Intro_DBMS\Lectures` folder). The AdventureWorks database is a comprehensive SQL Server sample database. Unless it came installed with the SQL Server Express, you must download the zipped file and install the database.

12. Explore the Product and BillOfMaterials tables. A bill of materials (BOM) lists all of the component products needed to make a finished product. For example: `SELECT * FROM BillOfMaterials WHERE ProductAssemblyID = 775` will list the ProductID's of component products needed to build finished ProductID 775 (Mountain-100 Black, 38). These component products, in turn, have their own BOMs. Final, finished products (such as ProductID 775 above) have a ProductAssemblyID of NULL. Note that BOMs have an end date, which you might need to take into consideration! List the ProductID, name, color, and quantity of all products required to build finished ProductId 775. (90 rows total, including one row for ProductID 775 in the final result). Note: SQL Server does not use the RECURSIVE keyword.

Submission: You must submit `MySQL8_MiscTopics.sql` SQL script, and the `Assgn8_MiscTopics.xlsx` Excel file on Canvas by the designated due date.