CIS4331 Spring 2019 Lab 6
Summary Queries Using Aggregation Functions

1. Objectives

This lab will help you to

- Learn how to summarize data using aggregation functions
- Enhance your skills to query data from one or more tables

2. Tasks to Complete

Complete the questions about data summaries on the **MGS Database** included in the later part of this document. These queries use the tables in **user mgs**. This lab is **required to submit by 11:50pm, Sunday, Feb. 24**.

NOTE: the links to online Oracle SQL Language references are available in the Modules\Resources folder on Canvas.

NOTE: The scripts you need in this practice can be downloaded from the item "*Murach Book Example Code"* in the Modules\Resources folder on Canvas.

3. Submission Requirements

Please place your SQL statements in a text file with the extension .sql. Mark each query based on the question number. Write your FULL name on the first page.

Then submit this SQL script file by attaching it to the link **Lab 6** in folder **Assignments\Labs** on Canvas.

Remember that only the last task is **required to submit by 11:59pm, Sunday, Feb. 24**.

Summary queries on MGS Database

1. Print the total number of orders and their total tax amount.
   Use headings ORDER_COUNT, TOTAL_TAX in the query result.

2. How many orders are paid by Visa cards, and what are the total tax amount of these orders? Use a single query to get both answers.
   Use headings VISA_ORDER_COUNT, VISA_TOTAL_TAX in the query result.

3. For each type of credit cards, print the total number of orders paid by this card type, and the total tax amount of these orders.
   Use headings CARD_ORDER_COUNT, CARD_TOTAL_TAX in the query result.

4. Rewrite the query in Question 3 such that only credit cards with at least 2 orders are printed.
   Use headings HOT_CARD_ORDER_COUNT, HOT_CARD_TOTAL_TAX in the query result.
   HINT: need to use HAVING clause.

5. Rewrite the query in Question 4 such that only orders dated in March 2012 are included in the query result. In another word, for each type of credit cards with at least 2 orders dated in March 2012 and paid by this card type, print March 2012 as the selection date, the count and total tax amount of such orders for this credit card type.
   Use headings SELECTION_DATE, HOT_CARD_ORDER_COUNT, HOT_CARD_ TOTAL_TAX in the query result.
   HINT: need to add a constant value in SELECT clause, use a pattern for desired orders, and use both WHERE clause and HAVING clause in the query.

6. For each product category, print how many products are in this category, the highest listing price of products in this category, and the lowest listing price of products in this category.
   Use headings CATEGORY_NAME, PRODUCT_COUNT, HIGHEST_LISTING, LOWEST_LISTING in the query result.
   HINT: need to use a JOIN and GROUP BY clause in your summary query.

7. Rewrite the query in Question 6 such that only products with listing prices more than $300 are included in the query result. In another word, for each product category, print how many products with listing prices more than $300 are in this category, the highest listing of such products, and the lowest listing of such products in this category.
   Use headings CATEGORY_NAME, PRODUCT_COUNT_OVER$300, HIGHEST_LISTING_OVER$300, LOWEST_LISTING_OVER$300 in the query result.
   HINT: think about which clause to add: a WHERE clause or HAVING clause?

8. Rewrite the query in Question 6 such that only categories with at least 3 products are included in the query result. In another word, for each category with at least 3 products,

print how many products are in this category, the highest listing of products in this category, and the lowest listing of products in this category.
Use headings CATEGORY_NAME, HOT_CAT_PRODUCT_COUNT, HOT_CAT_HIGHEST_LISTING, HOT_CAT_LOWEST_LISTING in the query result.
HINT: think about which clause to add: a WHERE clause or HAVING clause?