Lab 5: Counting with SQL

Due date: Thursday, May 15, **at the beginning of the lab period**.

Lab Assignment

Assignment Preparation

This is an individual lab. Each student has to complete all work required in the lab, and submit all required materials **exactly as specified** in this assignment.

The assignment will involve writing SQL queries for different information needs (quesions asked in English) for each of the five course datasets.

The Task

You are to write and debug (to ensure correct output) the SQL queries that return information as requested in each of the information needs outlined below. Each information need in this lab can be represented by either a single SELECT statement (possibly including aggregate operations, GROUP BY and HAVING clauses), or by a number of SELECT statements combined using a combination of MINUS, UNION and INTERSECT operators.

For this assignment, you will prepare one SQL script for each database. In addition to SQL statements you may need to include some SQL*plus formatting instructions to ensure that your output looks good. In particular, every row of every resulting table must be printed in a single line. If that means changing the size of the line - do it. Similarly, there should not be awkward pagination of the answers - change page size as needed.

STUDENT dataset

For STUDENT dataset, write an SQL script containing SQL statements answering the following information requests.

1. Find the total number of students who attend 3d grade. Report just the number.

2. Find the total number of students in KIRK MAROTTE's class. Report just the number.

3. For each classroom, report the total number of students attending it.

4. Report the names of teachers who have more than 7 students in their classes.

5. For each grade, report the number of classrooms in which it is taught.

BAKERY dataset

Write an SQL script containing SQL statements answering the following information requests.

1. Find how many different customers made purchases on October 14, 2007. Report just the number.

2. For each type of food output the average price on an item of that type.

3. For each purchase made on October 7, 2007 output the receipt number, total number of items purchased and total price.

4. For each purchase made on October 7, 2007 output the receipt number and the prices of the most expensive and the least expensive item.

5. Find all purchases made between October 1 and October 6, 2007 (including these two dates), which consisted of five items (notice that five is the maximum number of items per receipt in the BAKERY dataset).

CARS dataset

1. Find the average engine displacement of an 8-cylinder car produced in the 1970s. Report just the number.

2. For each year, report the average gas milage of US-made cars.

3. Find the weight of the heaviest non-US made car.

4. For each US car maker, report the number of 4-cylinder cars produced between 1975 and 1979 (inclusive).

5. For each US car maker, report the best and worst acceleration for a car it produced in 1977. (Remember, smaller accelerations are better).

CSU dataset

Here are the queries for the CSU dataset.

1. Report the average, the maximum and the minimum enrollment among the CSU campuses in 2002.
2. Report the number of years in which enrollment in `California Polythechnic State University - San Luis Obispo` has exceeded 14,000.
3. Report the earliest year when a CSU campus was founded.
4. For each campus which averaged more than $2000 in fees from 1996 to 2000 (inclusive), report the total amount of fees for that period. Sort your output by the total amount of fees.
5. For each campus for which data exists for more than 40 years, report the average, the maximum and the minimum enrollment (for all years). Sort your output by average enrollment.

MARATHON dataset

For this dataset, all times must be output in the same format as in the original dataset (in the file `marathon.csv`).

Note: please remember that the textbfbest, i.e., the **fastest** time is the smallest one!

1. Find how many athletes ran faster than one hour and thirty minutes. Report just the number.
2. Find how many female runners are in top 100 finishers. Report just the number.
3. For each gender/age group, report total number of runners in the group, the overall place of the best runner in the group and the best time shown by the runner in the group.
4. For each state, whose representatives participated in the marathon report the number of runners from it, who finished in top 20 within their age/gender group.
5. For each Massachussets town with 5 or more participants in the race, report the average time of its resident runners in the race. Output the results sorted by the average time.

AIRLINES dataset

1. Find the total number of flights originating at the `ANY` airport. Report just the number.

2. Find all airports with exactly 10 outgoing flights. Report airport code and the full name of the airport.

3. Find the number of airports from which airport `KKI` can be reached with exactly one transfer. (make sure to exclude `KKI` itself from the count). Report just the number.

4. Find the number of airports from which airport `KKI` can be reached with *at most* one transfer. (make sure to exclude `KKI` itself from the count). Report just the number.

5. For each airline, report the number of flights between two airports whose names are `Municipal` . Count a pair of flights between two such airports as one flight, (i.e., a flight from airport A to airport B and a return flight by the same company from B to A count as one flight in this query).

Submission Instructions

You must submit all your files in a single archive. Accepted formats are `gzipped tar` (`.tar.gz`) or `zip` (`.zip`). The file you are submitting must be named `lab4-ilastname.ext`, where *i* stands for the initial of your first name, and *lastname* is your last name. E.g., if I were submitting this file, the name would be `lab5-adekhtyar.zip` or `lab4-adekhtyar.tar.gz`.

The archive shall contain six directories: `AIRLINES`, `CARS`, `CSU`, `BAKERY`, `STUDENTS` and `MARATHON`.

Each directory shall contain the following SQL scripts:

- Database creation (`<DATABASE>-setup.sql`), database population (`<DATABASE>-insert.sql`) and database cleanup (`<DATABASE>-insert.sql`) scripts from Lab 4.

- **NEW script.** One script per database, containing all SQL statements and any SQL*plus statements needed for formatting. Name the script `<DATASET>-queries.sql` (e.g., `CARS-queries.sql`.

Note: Please do not use any `spool` commands in your scripts.