Lab 5: Counting with SQL

Due date: Thursday, May 13, **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 the STUDENT dataset, write an SQL script containing SQL statements answering the following information requests.

1. Find the total number of students who attend fourth grade. Report just the number.
2. Find the total number of classrooms in which fourth grade students study. Report just the number.
3. Find the total number of students in LEIA TARRING's class. Report just the number.
4. For each classroom, report the total number of students attending it. Order your output in descending order by the count.
5. Report the names of teachers who have more than six students in their classes.
6. 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 23, 2007. Report just the number.
2. For each flavor of food output the average price on an item with that flavor and the total number of different items with the flavor. Output the results in ascending order by average price.
3. For each purchase made on October 23, 2007 output the receipt number, total number of items purchased and the total amount of money paid. Sort the output in ascending order by the total amount of money paid.
4. For each purchase made on October 23, 2007 output the receipt number and the prices of the most expensive and the least expensive item. Sort the output by receipt number.
5. Find all purchases made between October 10 and October 17, 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). Output the receipt number and the date of each purchase. Sort the output in ascending order by date.
6. Find the number of `Strawberry Cakes` sold by the bakery during October of 2007. report just the number.
7. For each customer, report the total number of purchases they made and the total number of individual pastries they bought. Sort the output alpabetically by the last name of the customer. Output both first and last name of each customer.

CARS dataset

1. Find the average gas milage of an 8-cylinder car produced in the 1970s. Report just the number.
2. For each year, report the average engine displacement of US-made cars. Sort output by year in ascending order.
3. Find the weight of the lightest european car.
4. For each non-US car maker, report the number of 4-cylinder cars produced between 1970 and 1977 (inclusive). Sort the output in descending order by the total number 4-cylinder cars.
5. For each US car maker, report the best and worst acceleration for a car it produced in 1975 and the total number of models produced that year. (Remember, smaller accelerations are better). Output results in alphabetical order by car maker.
6. For each year in each `gm` produced more than 8 models, report the heaviest, the lightest and the average weight of a `ford` model. Report results in chronological order.
7. For each year between 1976 and 1982 (inclusively) and for each US carmaker report the number of models they produced that year (as stored in the database). Report results in chronological order, sort carmakers alphabetically within each year.
8. For each country report the total number models produced in 1979 (if a country did not produce a model that year, it need not be shown).

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 2000.
2. Report the number of years in which enrollment in `California Polythechnic State University - San Luis Obispo` has exceeded 15,000.
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.

6. For each campus report the total number of degrees granted in the period from 1995 to 2004 (inclusively). Sort the output in descending order by the total number of degrees granted.

7. For each county with more than one university campus, report the total enrollment (use student FTE) and the average faculty FTE among the county's campuses in 2004.

8. For each discipline show the number of campuses with students enrolled in it in 2004. Sort the output by the total number of campuses for each discipline.

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 twenty 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. Output result sorted by age group and sorted by gender (F followed by M) within each age 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. Output in descending order by the computed number.

5. For each Rhode Island town with 5 or more participants in the race, report the average time of its resident runners in the race *computed in seconds*. Output the results sorted by the average time (best average time first)

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). (The query should return tuples only for those airlines that have such flights).

6. For each airline report the total number of airports in which it operates. An airline operates in an airport if there is at least one outgoing flight for this airline from the airport. Report the results sorted by the total number of airports in descending order.

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 `lab5.ext`, where `ext` is one of the extensions above.

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>-lab5.sql` (e.g., `CARS-lab5.sql`.

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

Submit using `handin`:

```
handin dekhtyar-grader lab05-365 lab05.ext
```