Lab 5: Counting with SQL

Due date: Thursday, November 3, **during 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 for 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.

General Note: In queries that start with the phrase "For each YYY report ...", you are expected to include the column representing YYY in your output. For example, the query "For each grade report the sum of all classrooms" should result in a query that outputs two columns: `GRADE` and `SUM(CLASSROOM)`. This applies to all datasets and all upcoming labs as well.

STUDENT dataset

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

1. Find the total number of teachers in the school. 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 the class attended by `TOBIE SAADE`.
4. Report the names of teachers who have between four and six (inclusive) students in their classes. Sort the output in alphabetical order by last name of the teacher.
5. For each grade, report the number of classrooms in which it is taught and the toal number of students in the grade. Sort the output by the number of classrooms in descending order, then by grade in ascending order.

BAKERY dataset

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

1. Find how many different purchases were made on October 17, 2007. Report just the number.
2. Find the total amount of money the bakery earned between October 3, 2007 and October 7, 2007 (inclusive). Report just the amount.
3. For each purchase made on October 23, 2007 output the name of the customer (first, last) who made the purchase, total number of items purchased and the total amount of money paid. Sort the output in descending order by the total amount of money paid.
4. For each purchase made by `JULIET LOGAN` output the receipt number, the date of purchase, the total number of items purchased and the prices of the most expensive and the least expensive item. Sort the output by date of purchase.
5. Find the number of `Apple`-flavored pastries (except for the `Apple Pie`) sold by the bakery during October of 2007. Report just the number.
6. For each pastry type (e.g., `Pie`, `Tart`, etc…) report the number of pastries of this type on the menu, and the average price. Sort the output alphabetically by pastry type.

7. For each customer, report the total number of purchases they made, the total number of individual pastries they bought and the total amount of money they spent. Sort the output in descending order by the number of purchases. Output both first and last name of each customer.

CARS dataset

1. Find the average, the smallest and the largest weight of a 4-cylinder car produced in the 1970s. Report just the numbers.
2. For each Japanese car maker report the best 0 to 60 mph acceleration of a car produced by it. Sort output by year in ascending order.
3. Find the weight of the heaviest US-made car.
4. For each US car maker (reported by their short name), report the number of non 4-cylinder cars with gas milage better than 20 MPG produced between 1975 and 1980 (inclusive). Sort the output in descending order by the number of cars reported.
5. A junkyard is planning to stack on top of each other every single French car produced between 1971 and 1977 (inclusive)[1] Report the weight of this pile.
6. For each year in which `chrysler` produced more than 5 models, report the heaviest, the lightest and the average weight of a `chrysler` vehicle. Report results in chronological order.
7. For each country[2] report the total number of automakers that produced at least one car in 1977.
8. Report the names of all countries that in the 1970s produced more than 8 cars that had acceleration of 11 seconds or better. Report just the names of the countries in alphabetical order.

CSU dataset

Here are the queries for the CSU dataset.

1. Report the average, the largest and the smallest number of faculty employed on the `California Polytechnic State Univeristy-San Luis Obispo` campus.
2. Report the number of years in which enrollment at `California State University-Los Angeles` has exceeded enrollment (for the same year) at `Long Beach State University`.

[1]"Every" = every car found in our dataset.
[2]Which had at least one car in 1977.

3. Report the latest year when a CSU campus was founded.
4. For each `Los Angeles` county CSU campus report the average, the lowest and the highest fee over the period of time reflected in the database (1996 – 2006). Report the campuses in descending order by the average fee.
5. For each campus for which data exists for more than 60 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 number of disciplines for which the campus had non-zero graduate enrollment. Sort the output in alphabetical order by the name of the campus. (This query should exclude campuses that had no graduate enrollment at all).
7. For each county with more than one university campus, report the total number of campuses, the total enrollment (use student FTE) and the average faculty FTE among the county's campuses in 2004.
8. Report all the disciplines which had non-zero graduate enrollment on all CSU campuses in 2004 offering these disciplines. Report just the names of the disciplines sorted in alphabetical order.

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 **best**, i.e., the **fastest** time is the smallest one!

1. Find how many female runners from Massachusetts ran faster than one hour and thirty minutes. Report just the number.
2. Find how many runners in 40-49 age group are 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 100 (if a state did not have runners in top 50, do not output information about the state). Output in descending order by the computed number.
5. For each Rhode Island town with 7 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 `ANN` airport. Report just the number.

2. Find all airports with exactly 17 outgoing flights. Report airport code and the full name of the airport.

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

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

5. For each airline report the total number of airports from which it has more than one outgoing flight. Report the full name of the airline and the number of airports computed. Report the results sorted by the number of airports in descending order.

INN dataset

1. Count how many stays over two nights long in the `Frugal not apropos` room included two adults and no children, and how many total days were in those stays. Report just the two numbers.

2. For each room report the total revenue generated by stays in the room for the whole year of 2010 (i.e., all reservations in the database). Sort output in descending order by revenue. (Output full room names).

3. Report the total number of reservations that commenced on Sturdays. (*Hint:* look up the date of the *first* Saturday on the calendar).

4. Report the names of all visitors who stayed at the inn on three or more occasions. Sort alphabetically by last name.

5. For each room report the average number of nights per reservation. Output full names of rooms, sort in descending order by the average.

6. For each room report how many nights in 2010 the room was occupied. Report the room code, the full name of the room and the number of occupied nights. Sort in descending order by occupied nights. (Note: it has to be *number of nights in 2010* - the last reservation in each room *may* and *will* can go beyond December 31, 2010, so the "extra" nights in 2011 need to be deducted).

   Note/Hint: This is almost an extra credit problem. While multiple solutions are possible, my solution uses SQL's `SIGN()` built-in function which returns -1 for negative numbers, +1 for positive numbers and 0 for 0.

WINE dataset

1. Report the total number and the highest, the lowest and the average prices for a bottle of a white wine with a score of 94 or above.
2. For each wine score value above 88 report average price, the cheapest price and the most expensive price for a bottle of wine with that score (for all vintage years combined), the total number of wines with that score and the total number of cases produced. Sort by the wine score.
3. For each year, report the total number of red Napa wines whose scores are 92 or above. Output in chronological order.
4. For each appellation that produced more than two Zinfandel wines in 2008 report its name and county, the total number of Zinfandel wines produced in 2008, the average price of a bottle of Zinfandel from that year and the total (known) number of cases. Sort output in descending order by the number of wines.
5. For each county compute the total (known)[3] sales volume that it can generate. Sort the output in descending order by the total sales volume. (Note: recall what a case of wine is). Exclude Calaveras county and any appellations that are larger than a single county (their county is listed as `'N/A'`) from your report.
6. Compute production volumes (in the total number of bottles) by California regions (areas, e.g., Central Coast) for 2008. Report the region name, the number of different wines produced and the total number of bottles. Sort by the total number of bottles in descending order. Exclude wines labeled `'California'` or with no area information available.

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

[3] Recall, that information about production volumes for some wines is not available.

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

Submit using `handin`:

```
handin dekhtyar lab05 lab05.ext
```