Lab 5: Counting with SQL

Due date: Thursday, May 12, **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 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.

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 fifth grade. Report just the number.

2. Find the total number of classrooms in which kindergarden students study. Report just the number.

3. Find the total number of students in BILLIE KRIENER's class. Report just the number.

4. Report the names of teachers who have more than five 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. Sort the output by the number of classrooms in descending order, then by class in ascending order.

BAKERY dataset

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

1. Find how many different customers made purchases on October 12, 2007. Report just the number.

2. Find the total amount of money KIP ARNN spent at the bakery on October 20, 2007.

3. For each purchase made on October 12, 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 on October 12, 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 the number of `Ganache Cookies` sold by the bakery during October of 2007. report just the number.

6. Report the total number of tarts, cakes and cookies purchased during the month of October. Output the type of pastry and the total. Sort the output in alphabetical order by type of pastry.

7. For each customer, report the total number of purchases they made and the total number of individual pastries they bought. 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 engine displacement of a 6-cylinder car produced in the 1970s. Report just the numbers.
2. For each year in the 1980s, report the best 0 to 60 mph acceleration for cars made in Japan. Sort output by year in ascending order.
3. Find the weight of the heaviest US-made car.
4. For each US car maker, report the number of non 4-cylinder cars produced between 1975 and 1980 (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 which `ford` produced more than 6 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. Report the names of all countries that in the 1970s produced more than 10 cars that had gas milage better than 25 mpg. 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 a CSU campus in 2002.
2. Report the number of years in which enrollment in `California Polythechnic State University - San Luis Obispo` has exceeded enrollment (for the same year) in `Fresno State University`.
3. Report the latest year when a CSU campus was founded.
4. For each `Los Angeles` county CSU campus report the average 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 with between 10,000 and 20,000 students enrolled in 2004, report the number of disciplines for which the campus had non-zero undergraduate enrollment. Sort the output in alphabetical order by the name of the campus.

7. For each county with more than one university campus, 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. 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 textbfbest, i.e., the **fastest** time is the smallest one!

1. Find how many female runners ran faster than one hour and thirty minutes. Report just the number.

2. Find how many runners in 50-59 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 50 (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 Connecticut 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 `ALX` airport. Report just the number.

2. Find all airports with exactly 15 outgoing flights. Report airport code and the full name of the airport.

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

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

5. 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.

INN dataset

1. Count how many seven-night stays occurred at the inn during the month of January (of 2010) and report the number (the entire stay must occur in the month of January).

2. Find the total revenue of the inn for all available reservations.

3. Report the total number of reservations that commenced on Fridays. (*Hint:* look up the date of the *first* Friday 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 with two beds report the total number of reservations originated in 2010 and the total numbers of adults and kids (as separate columns) who stayed in the rooms. Report both the room code and the full name of the room. Sort the output in alphabetical order by room code.

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* - note the the last reservation in each room *may* and *will* 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 of cases of all wines rated 95 points of above (for all vintage years combined).

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) and the total number of wines with that score. Sort by the wine score.

3. For each year, report the total number of white wines whose scores are 88 or above. Output in chronological order.

4. For each appellation that produced more than two Cabernet Sauvignon wines in 2007 report its name and county, the total number of Cabernet Sauvignon wines produced in 2008 and the total (known) number of cases.

5. For each grape variety compute the total (known)[1] 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).

6. Compute production volumes (in the total number of bottles) by California regions (areas, e.g., Central Coast) for 2006. 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`.

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

Submit using `handin`:

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

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