Lab 5: Counting with SQL

Due date: Thursday, November 1, **midnight**.

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 second-graders in the school. Report just the number.

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

3. Find the total number of students in the class attended by `ROBBY PINNELL`.

4. Find the total number of students in `OTHA MOYER`'s class.

5. 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.

6. 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 25, 2007. Report just the number.

2. For each pastry flavor which is found in more than three types of pastries sold by the bakery, report the average price of an item of this flavor and the total number of different pastrties of this flavor on the menu. Sort the output in ascending order by the average price.

3. Find the total amount of money the bakery earned between October 10, 2007 and October 15, 2007 (inclusive). Report just the amount.

4. For each purchase made on October 31, 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.

5. For each purchase made by `MELLIE MCMAHAN` 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.

6. Find the number of `Chocolate`-flavored pastries (except for the `Chocolate Cake`) on the menu. Report just the number.

7. Find the total number of `Strawberry Cake`s purchased in October of 2007. Report just the number.

8. 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 engine displacement of a 6-cylinder car with 0 to 60 mph acceleration *better* than 15 (seconds) produced in the 1970s. Report just the numbers.

2. For each French car maker report the best milage per gallon of a car produced by it. Sort output in descending order by the best milage.

3. Find the weight of the lightest European car.

4. For each US car maker (reported by their short name), report the number of non 4-cylinder cars with 0 to 60 mph acceleration better than 14 seconds. 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 non-US car produced between 1971 and 1977 (inclusive)[1] Report the weight of this pile.

6. For each year in which `toyota` produced more than 2 models, report the best, the worst and the average gas milage of a `toyota` vehicle. Report results in chronological order.

CSU dataset

Here are the queries for the CSU dataset.

1. Report the average, the largest and the smallest number degrees granted on a CSU campus in 2000.

2. Report the number of years in which enrollment at `California Polytechnic State University - San Luis Obispo` has exceeded enrollment (for the same year) at `Fresno State University`.

3. Report the year, the first CSU campus was founded.

[1] "Every" = every car found in our dataset.

4. Report the average age of a CSU campus (as of this year).

5. For each campus that averaged more than $2500 in fees between 2000 and 2005, report the total cost of fees for this five year period. Sort in ascending order by fee.

6. 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.

7. For each campus report the total number of degrees granted between 1995 and 2004 (inclusively). Sort the output in descending order by the number of degrees.

8. 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).

9. 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.

10. Report all the disciplines which had non-zero graduate enrollment on all CSU campuses in 2004 offering these disciplines. For each discipline, show the total number of campuses on which it is offered to graduate students. Sort the output in ascending order by the number of campuses offering the programs.

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 marathon participants finished the race with a time better than $1:20:00$.

2. Find how many female runners from Massachusetts were among the first 75 finishers.

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 in their gender-age group (if a state did not have runners in top 50s, do not

output information about the state). Output in descending order by the computed number.

5. For each `CT` town with 3 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 `AHD` airport. Report just the number.
2. Find the total number of direct flights between `ATV` and `ALE`.
3. Find the pairs of airports which have more than two different direct non-stop flights between them. Report each pair exactly once (i.e., if a pair X, Y is reported, then Y, X does not need to be reported).
4. Find all airports with exactly 13 outgoing flights. Report airport code and the full name of the airport.
5. Find the number of airports from which airport `AHD` can be reached with exactly one transfer. (make sure to exclude `AHD` itself from the count). Report just the number.
6. Find the number of airports from which airport `AHD` can be reached with *at most* one transfer. (make sure to exclude `AHD` itself from the count). Report just the number.
7. For each airline report the total number of airports from which it has at least 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 four nights long in the `Harbinger but bequest` room included one adult and one child, and how long was the average stay in such cases. Report just the two numbers.
2. For each room report the total revenue and the average revenue per stay generated by stays in the room that originated in the months of June, July and August. Sort output in descending order by total revenue. (Output full room names).
3. Report the total number of reservations that commenced on Sundays. (*Hint:* look up the date of the *first* Sunday on the calendar).
4. Report the names of all visitors who stayed at the inn on three or more occasions. For each visitor, report the total number of nights spent in the inn. Sort alphabetically by last name.

5. Find the total number of single night stays of two adults with no kids in the rooms with maximum occupancy of two people. Report just the number.

6. For each room report the highest markup against the base price and the smallest markup (i.e., largest markdown). Report markups and markdowns both in absolute terms (absolute difference between the base price and the rate) and relative terms (percent change). Sort output in descending order by the absolute value of the largest markup.

7. For each room report the largest revenue from a single stay, the smallest revenue from a single stay, the total number of days the room was occupied[2] Output full names of rooms, sort in descending order by number of days of occupancy.

8. 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 red wine with a score of 96 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 white Santa Barbara County wines whose scores are 90 or above. Output in chronological order.

4. For each appellation that produced more than two Pinot Noir wines in 2008 report its name and county, the total number of Pinot Noir wines produced in 2008, the average price of a bottle of Pinot Noir from that year and the total (known) number of bottles produced[3]. Sort output in descending order by the number of wines.

[2] Count all days, including the days in 2011 for which the information in the database is available.

[3] Recall, one case is 12 bottles.

5. For each appellation inside Sonoma county compute the total (known)[4] 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).

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 lab05 lab05.ext
```

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