Cal Poly Spring 2013	**CPE/CSC 365** Introduction to Database Systems	Alexander Dekhtyar Eriq Augustine

Lab 6
Counting with SQL

Due date: May 15$^{\text{th}}$.

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 (questions asked in English) for each of the 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 UNION.

NOTE: You **MUST** provide a comment on the line before each SELECT statement. The comment must denote the number of each query. Use "`-- Q#`", where "`#`" is the number of the query. If for any reason you are missing a query, just leave out the query and the comment. Improperly labeling you queries can confuse the grading script.

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.

AIRLINES dataset

1. Find all airports with exactly 19 outgoing flights. Report airport code and the full name of the airport sorted in alphabetical order by the code.

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

3. For all airports with exactly 19 outgoing flights, find the number of airports that can be reached using exactly one transfer (make sure to exclude the $A \rightarrow B \rightarrow A$ flights from consideration). Output the airport code and the count. Sort output in descending order by the total number of airports. Keep in mind that, flightNo is not unique by itself.

Note: Getting this query right without resorting to nested queries is hard, but possible. Don't use nested queries.

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

5. 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.

BAKERY dataset

1. 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 pastries of this flavor on the menu. Sort the output in ascending order by the average price.

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

3. For each purchase made by `CHARLENE MESDAQ` output the receipt number, the date of purchase, the total number of items purchased and the amount paid. Sort in descending order by the amount paid.

4. For each day of the week of October 8, 2007 (Monday to Sunday) report the total number of purchases (receipts), the total number of pastries purchased and the overall daily revenue. Report results in chronological order.

5. Report all days on which more than five cakes were purchased, sorted in chronological order.

CARS dataset

1. For each Japanese car maker (reported by their short name) report the best mileage per gallon of a car produced by it and the average acceleration. Sort output in ascending order by the best mileage.

2. For each US car maker (reported by their short name), report the number of 4-cylinder cars that are lighter than 4000 lbs with 0 to 60 mph acceleration better than 14 seconds. Sort the output in descending order by the number of cars reported.

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

4. For each year when US-manufactured cars averaged less than 100 horsepowers, report the best and the highest and the lowest engine displacement number. Sort in chronological order.

CSU dataset

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

2. For each campus for which data exists for more than 60 years, report the minimum, average, maximum enrollment (for all years). Sort your output by average enrollment.

3. For each campus in LA and Orange counties report the total number of degrees granted between 1998 and 2002 (inclusively). Sort the output in descending order by the number of degrees.

4. For each campus that had more than 20000 enrolled students in 2004, 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).

5. For each year between 2002 and 2004 (inclusively), report the best, the worst, and the average student-to-faculty ratio among the CSU campuses. Use FTE number for the enrolled students. Sort results chronologically.

INN dataset

1. For each room, report the total revenue for all stays and the average revenue per stay generated by stays in the room that originated in the months of September, October and November. Sort output in descending order by total revenue. (Output full room names).

2. Report the total number of reservations that commenced on Fridays and the total revenue they brought in. (*Hint:* look up the date of the *first* Friday on the calendar, or other useful DATE/TIME functions).

3. For each day of the week, report the total number of reservations commenced on it and the total revenue these reservations brought. You can report days of week as numbers $1-7$, with 1 representing *Sunday*. The output should be sorted by the day number.

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

MARATHON dataset

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

1. 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 overall place of the worst runner in the group. Output result sorted by age group and sorted by gender (F followed by M) within each age group.

2. Report the total number of gender/age groups for which both the first and the second place runners (within the group) hail from the same state. Report just the number.

3. For each full minute, report the total number of runners whose pace was between that number of minutes and the next. (That is, how many runners ran the marathon at a pace between 5 and 6 minutes, how many - at a pace between 6 and 7 minutess, and so on). Order by the minute.

4. For each state, whose representatives participated in the marathon, report the number of runners from it who finished in top 10 in their gender-age group (if a state did not have runners in top 10s, 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).

STUDENTS dataset

1. Report the names of teachers (first, last) who have between three and five (inclusive) students in their classes. Sort the output in alphabetical order by last name of the teacher. Sort output in alphabetical order by teacher's last name.

2. For each grade, report the number of classrooms in which it is taught and the total number of students in the grade. Sort the output by the number of classrooms in descending order, then by grade in ascending order.

3. For each fourth-grade classroom, report the total number of students. Sort output in the descending order by the number of students.

4. For each kindergarten classroom, report the student (last name) who is the first (alphabetically) on the class roster. Sort output by classroom.

WINE dataset

1. 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.

2. For each year, report the total number of red Sonoma County wines whose scores are 90 or above. Output in chronological order.

3. For each appellation that produced more than two Cabernet Sauvingnon wines in 2007 report its name and county, the total number of Cabernet Sauvingnon wines produced in 2008, the average price of a bottle of Cabernet Sauvingnon from that vintage, and the total (known) number of bottles produced[1]. Sort output in descending order by the number of wines.

4. For each appellation inside Central Coast compute the total (known)[2] sales volume that it can generate for the wines produced in 2008. Sort the output in descending order by the total sales volume. (Note: recall what a case of wine is).

5. For each county in the database, report the score of the highest ranked 2009 red wine. Exclude wines that do not have a county of origin (`'N/A'`). Sort output in descending order by the best score.

[1] Recall, one case is 12 bottles.
[2] Recall, that information about production volumes for some wines is not available.

Submission Instructions

Please, follow these instructions exactly. Up to 20% of the Lab 6 grade will be assigned for conformance to the assignment specifications, **including the submission instructions**.

Be sure to properly comment your queries (see the comment in "The Task" section.

Please, **name your files exactly as requested** (including capitalization). Correct submission simplifies grading, and ensures its correctness.

There should be one file per dataset named: "¡DATASET¿-query.sql". E.g. for the AIRLINES dataset, the file should be "AIRLINES-query.sql". (See the handin command for proper naming.)

Please include your name and Cal Poly email address in all files you are submitting. If you are submitting code/scripts, include, at the beginning of the file a few comment lines with this information. Files that cannot be authenticated by observing their content will result in penalties assessed for your work.

No archives are expected for this lab. You are expected to handin one file per dataset and an optional README.txt including any comments about this lab. Submit your files using the following `handin` command:

$ handin eaugusti lab06 AIRLINES-query.sql BAKERY-query.sql CARS-query.sql CSU-query.sql INN-query.sql MARATHON-query.sql STUDENTS-query.sql WINE-query.sql [README.txt]