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

Lab 8
Full Power of SQL

Due date: May 22$^{\text{th}}$.

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 in each of the information needs outlined below. The information needs may be quite complex and to address them, the use of aggregation, grouping, nested queries, or their combinations may be required.

NOTE: If you have to do floating point comparison, ensure that the floats are within **0.0001** of each other. This is especially relevant in the CSU dataset.

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.

Each information needs needs to be addressed with a *single SQL statement*, but the statement can have multiple levels of nesting, grouping and aggregation, and use the `UNION` operation.

Note: In this lab, we use only seven databases. There are no queries for the AIRLINES database.

BAKERY database

1. Find the customer(s) who spent the most on pastries in October of 2007. Report first and last name.
2. Find the type of baked good (food type, flavor) responsible for highest total revenue.
3. Find the most popular (by number of purchases) item. Report the item (food, flavor).
4. Find the day of the highest revenue in the month of October.

5. For every customer who DID NOT make a purchase on the day of the highest revenue, report the total number of purchases (overall) the customer made, and the total amount of those purchases. Order the output by the total amount of purchases.

6. For every customer report the item they spent the most money on. Report customer name (first, last) and the name (flavor, food) of the item, as well as the total amount of money spent. Sort in alphabetical order by last name of the customer. (Note: some customers may have spent the same (largest) amount of money on two or more different items. All such items shall be displayed).

7. Output the names of all customers who made multiple purchases (more than one receipt) on the latest day in October on which they made a purchase. Report names (first, last) of the customers and the earliest day in October on which they made a purchase, sorted in chronological order.

CARS database

1. Report all vehicles with the best acceleration. For each vehicle, report its full name and the year of production.

2. Among the vehicles with the best acceleration, report the most powerful (horsepower) one. Report full name and the year of production.

3. Find the automaker that produced multiple vehicles in 1976, whose 1976 vehicles had the best average gas milage. Report the automaker, the number of vehicle models it produced in 1976 and the average gas milage. (Note: exclude automakers with a single vehicle in 1976 from consideration completely).

4. For each year find the automakers whose models for that year had the best average gas milage. Report the year, the automaker, the number of models produced that year, and the average gas milage. Present the output in chronological order.

5. Find the most fuel-efficient 8-cylinder model. Report the full name of the car, the year it was produced and the home country of its maker.

6. Find the difference in gas milage between the most fuel-efficient 8-cylinder model and the least fuel-efficient 4-cylinder model. Report just the number.

7. For each country report the number of 4-cylinder models its companies have produced in the 1970s which have higher horsepower than some 8-cylinder model also produced in the 1970s. (Note, the 8-cylinder model can come from any country and any company).

CSU database

1. Find the campus with the largest enrollment in 1957. Output the name of the campus and the total undergraduate enrollment.

2. Find the university that granted the largest total number of degrees over the entire recorded history. Report the name of the university and the total number of degrees.

3. Find the university with the best (smallest) faculty-to-student ratio in 2002. Report the name of the campus, total undergraduate enrollment, faculty FTE, and the faculty-to-student ratio.

4. Find the university with the largest percentage of the undergraduate student body in the engineering discipline in 2004. Output the name of the campus and the percent of the engineering students on campus.

5. For each year between 2000 and 2004 (inclusive), report the campus with the highest relative increase in enrollement from previous year. Output the year and the campus name.

   Note: if a university started accepting students in year $n \geq 2000$ for the first time, information about this university need not be captured in the process of determining the campus with the best relative increase in enrollment for year n. That is: only consider a campus in year n if it enrolled students in year $n - 1$.

   Note: See the note at the top about floating point comparison.

6. For each year between 1997 and 2003 (inclusive), find the university with the best (highest) total degrees granted to total enrollment (use enrollment numbers) ratio. Report the years, the names of the campuses, and the ratios in chronological order.

7. For each university with an undergraduate engineering program in 2004 (i.e., with a non-zero number of engineering undergraduates), report the year of the lowest student-to-faculty ratio (use enrollment FTE and faculty FTE numbers). Output campus name, year and the ratio in alphabetical order by campus name.

INN database

1. Find the least popular room in the hotel. The least popular room is the room that had seen the lowest number of reservations (Note: if there is a tie for the least popular room status, report all the least popular rooms). Report the full name of the room, the room code and the number of reservations.

2. Find the room that has been occupied the most based on the reservations in the database[1]. Report the room name, room code and the number of days it was occupied.

3. Find the most expensive reservation(s) made. Report the room name (full), dates of stay, last name of the person who made the reservation, daily rate, and the total amount paid.

4. For each room, report the most expensive reservation. Report the full room name, dates of stay, last name of the person who made the reservation, daily rate, and the total amount paid. Sort the output in descending order by total amount paid.

5. Find the best month (i.e., month with the highest total revenue). Report the month (as the full month name, e.g. 'September'), the total number of reservations and the revenue. For the purposes of the query, count the entire revenue of a stay that commenced in one month and ended in another towards the earlier month. (e.g., a September 29 - October 3 stay is counted as September stay for the purpose of revenue computation).

[1] No need to limit the number of occupied days to 2010.

6. For each room report whether it is occupied or unoccupied on October 22, 2010. Report the full name of the room, the room code, and put either `'Occupied'` or `'Empty'` depending on whether the room is occupied on that day. (the room is occupied if there is someone staying the night of May 19, 2010. It is NOT occupied if there is a checkout on this day, but no checkin). Output in alphabetical order by room code.

   (Hint: You can do an ORDER BY after a UNION.)

7. For each room report how many reservations were made for the most expensive rate for that room. Report full room name and the appropriate number of reservations. Sort the output in ascending order by the number of reservations.

MARATHON database

1. Find the state with the largest number of participants.

2. Find all towns in Massachussets (`MA`) which fielded more female runners than male runners for the race. Report the names of towns.

3. Find all people from `SOUTHBORO, MA` who ran better than at least one runner from the state of Missouri. Output the name (first, last) of each runner, their hometown, and state and the overall place in the race.

4. Find all towns in Massachussets (`MA`) all female runners from which had better than the average pace in the race. Report town names.

STUDENTS database

1. Find the grade(s) with the largest number of classrooms. Report the grade and the number of classrooms in it.

2. Find the classroom(s) with the smallest number students. Report the classroom and the teacher (first name, last name).

3. Find how many classrooms have the number of students that exceeds the average number of students per class. Report just the number.

4. Find all grades in which the number of students is smaller than the number of students in fourth grade. Report just the grades.

5. Find all pairs of grades with the same number of students in them. Report each pair only once. Report both grades and the number of students.

WINE dataset

1. Find the grape(s) that grow(s) in the largest number of appellations. Report grape name, color and the number of appellations it grows in.

2. Find the most popular white grape (i.e., the grape that is used to make the largest number of white wines in the database). Report the name of the grape.

3. Report the grape with the largest number of high-ranked wines (score of 93 and above).

4. Report the appellation responsible for the largest number of high-ranked wines (score of 93 and above). Report just the name of the appellation.

5. Find the high-ranked wine (score of 93 or above) responsible for highest sales revenue. Report the vintage year, winery, wine name, score, and the computed revenue.

6. Find if there are any 2008 Chardonnays that scored better than any 2007 Syrah. Report winery, wine name, appellation, score, and price.

7. Two California AVAs, Carneros and Dry Creek Valley have a bragging rights contest every year: the AVA that produces the highest-ranked wine among all the wines produced in both AVAs wins. Based on the data in the database, output (as a single tuple) the number of vintage years each AVA has won between 2005 and 2009 (you want the output to look like a score of a game between the two AVAs. Only the vintage years where one AVA won count - vintages when both AVAs had the same highest score should not be counted).

8. Find how many cases were produced of the most expensive red wine from San Luis Obispo county.

Submission Instructions

Please, follow these instructions exactly. Up to 20% of the Lab 7 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 BAKERY dataset, the file should be "BAKERY-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 lab07 BAKERY-query.sql CARS-query.sql CSU-query.sql INN-query.sql MARATHON-query.sql STUDENTS-query.sql WINE-query.sql [README.txt]