Lab 7: Full Power of SQL

Due date: Monday, June 4, 10:00am.

Note: Lab 8 will most likely be assigned on Firday, June 2.

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 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.

The information need must be addressed with a single SELECT statement. This statement can unclude multiple levels of nesting, grouping and aggregation, and `UNION` subqueries. You can use any SQL feature learned in the course, or discovered by you individually. This includes the `JOIN` syntax, and any other features MySQL SQL offers. However, I advise caution: *some* ways of addressing the assigned information needs using features we have not discussed in class are considered illegal, as they may not be guaranteed to always produce the correct result. If in doubt, consult the instructor.

For this assignment and further SQL labs you will be working with the instructor's versions of the databases from Labs 2 and 3. The read-only versions of these databases are available on the MySQL server as databases with the following names:

```
airlines
bakery
cars
csu
inn
katzenjammer
marathon
students
wine
```

Each of you has been granted the `SELECT` priveleges on each of these databases (contact the instructor if this is not the case). Your queries must be written for the tables in these databases and must run properly on them and produce correct output.

You will prepare one SQL script for each database. Please note: every row of every resulting table must be printed in a single line.

NOTE: Please provide a comment in front of each SQL statement in each of your files. The simplest comment can just state the query number (e.g., `"-- Q3."`) for this particular database. This is very useful for the situations when for one reason or another you elected not to implement a query.

Note: Please, make your queries human-readable. This means ensuring that all your queries fit 80-character lines (so that your files could be printed), and breaking the queries into multiple lines to improve readability. Use indentation where possible.

File names. Name the files containing your SQL queries for this assignment `<DATASET>-queries.sql`. For example, for the `CARS` dataset, name your file `CARS-queries.sql`.

STUDENTS database

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

1. Find the teacher(s) who teach(es) the largest number of students. Report the name of the teacher(s) (first and last) and the number of students in their class.

2. Find the grade(s) with the largest number of students whose last names start with letters `'A'`,`'B'`,`'C'`. Report the grade and the number of such students [1].

[1] We did not pay much attention to the `LIKE` operator in our class, but you can see how it works on the example of this query.

3. Find all classrooms which have fewer students in them than the average number of students in a classroom in the school. Report the classroom numbers in ascending order. Report the number of student in each classroom.

4. Find all pairs of classrooms with the same number of students in them. Report each pair only once. Report both classrooms and the number of students. Sort output in ascending order by the number of students in the classroom.

5. For each grade with more than one classroom, report the last name of the teacher who teachers the classroom with the largest number of students in the grade. Output results in ascending order by the grade.

BAKERY database

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

1. Find the customer(s) who spent the most on pastries in October of 2007. Report first and last name.

2. Find the customers who never purchased a twist (`'Twist'`) (in October of 2007). Report their first and last names in alphabetical order by last name.

3. Find the type of baked good (food type, flavor) responsible for highest total revenue.

4. Find the most popular (by number of pastries sold) item. Report the item (food, flavor) and its total number of sales.

5. Find the day of the highest revenue in the month of October, 2007.

6. Find the best-selling item (by number of purchases) on the day of the highest revenue in October of 2007.

7. for every type of `Cake` report the customer(s) who purchased it the largest number of times during the month of October 2007. Report the name of the pastry (flavor, food type), the name of the customer (first, last), and the number of purchases made. Sort output in descending order on the number of purchases, then in alphabetical order by last name of the customer.

8. Output the names of all customers who did not make a purchase between October 5 and October 11 (inclusive) of 2007. Output first and last names in alphabetical order by last name.

9. 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.

10. Find out if the sales of `Chocolate`-flavored items (in terms of revenue) or the sales of `Croissants` (of all flavors) were higher in October of 2007. Output the word `Chocolate`, if the sales of `Chocolate`-flavored items had higher revenue, or the word `Croissant` if the sales of `Croissants` had higher revenue.

    Note: This can be done in a number of ways. One way involved the `CASE...WHEN` clause inside the `SELECT` clause, but there are ways of building the output without the use of any "exotic" features.

CARS database

When writing SQL queries for the information needs below, please think carefully for each of the attributes of a vehicle, what consititutes "best", and what constitutes "worst" (e.g., is "best acceleration" the highest acceleration or the lowest? is "best gas mileage" the highest gas mileage or the lowest?)

1. Report all vehicles with the best gas mileage. For each vehicle, report its full name and the year of production.
2. Among the vehicles with the gas mileage, report the one with the best acceleration. Report full name and the year of production.
3. For each country report the automaker with the largest number of cars in the database. Report the name of the country, the short name of the automaker. Output in alphabetical order by country.
4. For each year find the automakers whose models for that year were (on average) the heaviest. Report the year, the automaker, the number of models produced that year and the average acceleration. Exclude any automakers that produced only one car in a particular year from consideration for that year. Present the output in chronological order.
5. 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.
6. For each year between 1972 and 1976 (inclusively) find if US automakers or all other automakers produced more different cars. Report, in chronological order either `'us'` or `'rest of the world'` for each year, depending on whether the US automakers had more different cars produced, or fewer. Report `'tie'` in case of a tie[2].

[2] If you discover that no ties exist in the years considered, you are allowed not to include the logic for reporting the output `'tie'` into your query. But if the ties do exist, this logic must be there, and the output must be correct.

CSU database

Here are the queries for the CSU dataset.

1. Find the campus with the largest enrollment in 2000. Output the name of the campus and the total undergraduate enrollment.
2. Find the university that granted the largest total number of degrees per year over its entire recorded history. Report the name of the university and the total number of degrees.
3. Find the university with the best (smallest) student-to-faculty ratio in 2003. Report the name of the campus and the student-to-faculty ratio. Use FTE numbers for the enrollment.
4. Find the university with the largest percentage of the undergraduate student body in the `Computer and Info. Sciences` in 2004. Output the name of the campus and the percent of the social sciences students on campus.
5. 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.
6. For each campus report the year of the best student-to-faculty ratio, together with the ratio itself. Sort output in alphabetical order by campus name (use FTE numbers to compute the ratios).
7. For each year (for which the data is available) report the total number of campuses in which student-to-faculty ratio became worse (i.e. *more* students per faculty) as compared to the previous year. Report in chronological order by campus.

INN database

1. Find the most popular room in the hotel. The most popular room is the room that had seen the largest number of reservations (Note: if there is a tie for the most popular room status, report all such 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 largest number of days based on the reservations in the database[3]. Report the room name, room code and the number of days it was occupied.
3. 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.

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

4. Find the best month (i.e., month with the highest total revenue). Report the month, 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).

5. For each room report whether it is occupied or unoccupied on July 4, 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 July, 2010. It is NOT occupied if there is a checkout on this day, but no checkin). Output in alphabetical order by room code.

MARATHON database

For this dataset, all times must be output in the same format as in the original dataset (in the file `marathon.csv`).

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

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

3. A "pace category" of runners consists of all runners who showed a pace that had the same full minute value. For example all runners who ran at the pace of 8 minutes and NN seconds where $NN \leq 59$ belong to the same pace category. For each pace category report the gender-age group with the largest number of runners in it. (Gender-age groups are reported as a pair of gender, and age bracket). Report the results sorted in ascending order by the full minute of the pace.

   Note: In other words, answer the query "Among all the runners who ran at the pace between N and $N+1$ minutes, find the most common gender-age group they belong to" for each possible $[N, N+1]$ interval.

4. For each state report the gender-age group with the largest number of participants. Output state, gender-age group and the number of runners in the group in alphabetical order by the state code. Report only information for the states where the largest number of participants in a gender-age group is greater than one.

5. Find the thirtyth fastest female runner. Report her overall place in the race and the full name.

   Note: This must be done using a single SQL query that DOES NOT use the `LIMIT` clause. Think of how this can be done. Think of what it means for a row to represent the thirtyth fastest (female) runner.

6. For each town in New Hampshire report the total number of male and the total number of female runners. Both numbers shall be reported on the same line. If no runners of a specfic gender from the town participated in the marathon, report `NULL`.

   Note: What type of an operation is called for here? With what data?

WINE dataset

1. Find the most popular red grape (i.e., the grape that is used to make the largest number of white wines) in San Luis Obispo County. Report the name of the grape.
2. Report the grape with the largest number of high-ranked wines (wines ranked 93 or higher).
3. Report the appellation responsible for the largest number of high-ranked wines (score of 93 and above). Report just the name of the appellation.
4. Find if there are any 2008 Zinfandels that scored better than all 2007 Grenaches. Report winery, wine name, appellation, score and price.
5. 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).
6. Find how many cases were produced of the most expensive red wine from Napa county.

KATZENJAMMER dataset

1. Report the first name of the performer who never played accordion.
2. Report, in alphabetical order (if more than one song returned), the titles of all instrumental compositions performed by Katzenjammer ("instrumental composition" means no vocals).
3. Report the title (or titles) of the song(s) that involved the largest number of instruments played by all performers combined (if multiple songs, report the titles in alphabetical order).

4. Find the favorite instrument of each performer. Report the first name of the performer, the name of the instrument and the number of songs the performer played the instrument on. Sort in alphabetical order by the first name.

5. Find all instruments that ONLY Anne-Marit played. Report instruments in alphabetical order.

6. Report the first name of the performer who played the largest number of different instruments.

7. Who spent the most time performing in the center of the stage (in terms of number of songs on which she was positioned there)? Return just the first name of the performer.

8. Which instrument(s) was/were played on the largest number of songs? Report just the names of the instruments (note, you are counting number of songs on which an intrument was played, make sure to not count two different performers playing same instrument on the same song twice).

Submission Instructions

Submit one script per database, containing all SQL statements. Name the script `<DATASET>-queries.sql` (e.g., `CARS-queries.sql`. Submit all files (and a `README` file in a single archive named `lab07.zip` or `lab07.tar.gz`. When unpacked, the files shall be placed in the root of your `handin` directory.

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

Submit:

```
$handin dekhtyar lab07 <file>
```