Lab 6: Full Power of SQL

Due date: Thursday, May 19, **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 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.

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.

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

STUDENT database

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

1. Find the grade with the largest number of students. Report the grade and the number of students in it.
2. Find the classroom with the largest number of students. Report the classroom and the teacher (first name, last name).
3. Find all classrooms where the number of students is less than the average number of students per class. Report the classroom number and the first and the last name of the teacher. Sort the output by classroom.
4. Find all grades in which the number of students is greater than the number of students in fifth 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.
6. Find all teachers who teach fewer students than `KIRK MARROTTE`. Report first and last names of the teachers.

BAKERY database

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

1. Find all customers who did not purchase any `Vanilla`-flavored items from the bakery. Report first and last name of the customer.
2. Find the total amount of money spent by `CHARLENE MESDAQ` on bakery purchases in October.
3. For each customer of the bakery find the total number of purchases (i.e., reciepts associated with that customer) and the total amount of money spent. Report the first and the last name of the customer, the number of purchases and the total amount; sort output in descending order by the total amount of money spent.
4. Find the total revenue of the bakery from chocolate-flavored goods. Report just the total revenue number.
5. Find the type of baked good (food type, flavor) responsible for highest total revenue.
6. Find the least popular (by number of purchases) item. Report the item (food, flavor).

7. Find the day of the highest revenue in the month of October.

8. For every customer who made 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.

9. 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).

10. Output the names of all customers who made multiple purchases (more than one receipt) on the earliest 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. Find the average gas milage for French cars produced between 1975 and 1980.

2. For each European country report the average gas milage, average acceleration and average engine displacement for the cars produced by the country's automakers in 1976.

3. Find all models in the database with the best overall acceleration[1]. Report the full name of the car, the year it was produced and the acceleration.

4. Find the automaker whose 1977 models had the best average gas milage. Report the automaker, the number of models it produced in 1977 and the average gas milage.

5. For each year find the automakes 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.

6. Find the least fuel-efficient 4-cylinder model. Report the full name of the car, the year it was produced and the home country of its maker.

7. For each US car maker find the lightest car it produced. Output the name of the car maker, the full name of the car, the year it was produced and its weight.

[1]Remember, the smaller acceleration, the better it is.

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

CSU database

Here are the queries for the CSU dataset.

1. For `San Jose State University` report all disciplines where total enrollment (undergraduate+graduate) in 2004 has exceeded 1000 students. Output the full name of the discipline. Exclude `'Undeclared'` majors from the output.

2. For each campus in `Los Angeles` county report the percentage of students studying `Engineering` in 2004. (Use the `TotalEnrollment_AY` column for the total enrollment numbers and the undergraduate enrollment for the discipline). Output the full name of the campus, and the computer percentage presented in the column named `"Eng_Percentage"`. Sort output by percentage.

3. For each campus in `Los Angeles` county report the percentage of students studying `Engineering` or `Computer and Info. Sciences` in 2004. (Use the `TotalEnrollment_AY` column for the total enrollment numbers). Output the full name of the campus, and the computer percentage presented in the column named `"Eng_CS_Percentage"`. Sort output by percentage.

4. Find the campus with the largest enrollment in 2004. Output the name of the campus and the total undergraduate enrollment.

5. 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.

6. Find the university with the best (smallest) faculty-to-student ratio in 2004. Report the name of the campus, total undergraduate enrollment, faculty FTE and the faculty-to-student ratio.

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

8. For each year between 1995 and 1999 (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 $n \geq 1995$ 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 $n-1$.

9. For each year between 1995 and 2004 (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.

10. 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 and faculty FTE numbers). Output campus name, year and the ratio in alphabetical order by campus name.

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 the most 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[2]. 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. For each room, report the total revenue the room has generated off of the reservations and the percentage of the overall hotel revenue the room reservations account for. Sort the rooms in descending order by the percentage.

6. Find the least kid-friendly 2-bed room (i.e., the room in which the smallest number of kids stayed). Report the name of the room, the number of kids who stayed in it and the decor style.

7. For each room report whether it is occupied or unoccupied on May 19, 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.

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

8. To simplify accounting, the hotel counts the revenue for a hotel stay for the month during which the hotel stay commenced. Using this accounting mechanism report the average, the highest and the lowest monthly revenues.

9. For each room report how many reservations were made for the lowest 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.

10. Report the name of the month during which the revenue generated from reservations of `'Harbinger but bequest'` room was the largest (among all monthly revenues for this room).

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 states with fewer race participants than New Hampshire (`NH`). Report each state and the number of race participants from it.

3. Find all towns in Massachussets (`MA`) which had more runners in the race than the entire state of Missouri (`MO`). Report just the names of towns.

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

5. 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.

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

WINE dataset

1. Find the county with the largest number of AVAs in it. Report just the county name.

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

3. For each vintage year report the highest scoring wine (or wines). Report year, grape, winery, wine name and the score. Output results in chronological order.

4. Report the grape with the largest number of high-ranked wines (score of 90 and above).

5. Find the wine responsible for highest sales revenue. Report the vintage year, winery, wine name, score and the computed revenue.

6. Find the highest-ranked cheap (price does not exceed $20) red wine from Sonoma County. Report grape, winery, wine name, appelation, score and price.

7. Find if there are any 2007 Zinfandels that scored better than any 2008 Cabernet Sauvingnon. report winery, wine name, appellation, score and price.

8. 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.

9. Find how many cases were produced of the most expensive white wine from Santa Barbara County.

10. Find the winery with the largest total number of wines in the database and report the name of the winery, total number of wines and the total sales revenue that their wines generate.

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 `lab6.ext`, where *i* stands for the initial of your first name, and *lastname* is your last name. E.g., if I were submitting this file, the name would be `lab5-adekhtyar.zip` or `lab4-adekhtyar.tar.gz`.

The archive shall contain seven directories: `CARS`, `CSU`, `BAKERY`, `INN`, `STUDENTS`, `WINE` and `MARATHON`.

Each directory shall contain the following SQL scripts:

- Database creation (`<DATABASE>-setup.sql`), database population (`<DATABASE>-insert.sql`) and database cleanup (`<DATABASE>-cleanup.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>-queries.sql` (e.g., `CARS-queries.sql`.

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