Lab 6: Complex Queries

Due date: Tuesday, November 6, **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 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.

STUDENT dataset

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

1. Find the total number of students in the third grade. Report just the number.
2. Find the total number of classrooms in which the fourth grade students are taught. Report just the number.
3. For each classroom, report the number of students studying in it.
4. For each grade, report the number of students studying in it.
5. Find the grade with the largest number of students. Report the grade and the number of students in it.
6. Find the classroom with the largest number of students. Report the classroom and the teacher (first name, last name).

BAKERY dataset

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

1. Find the total number of purchases of `Truffle Cake`.
2. Find the total amount of money spent by `CORETTA DUKELOW` 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). Report the first and the last name of the customer and the number of purchases.
4. Find the largest number of purchases a single customer made during the month of October.
5. Find the smallest number of items purchased by a single customer during the month of October (among the listed customers of the bakery).
6. Find the largest, the smallest and the average amount of money spent by a single customer on bakery purchases during the month of October. Report the three numbers.
7. For all customers with 10 or more separate purchases (reciepts), output the customers' names (first, last) and total number of purchased items and the total amount of money spent.
8. Find the total revenue of the bakery from chocolate-flavored goods.
9. For each type of baked good sold find the total revenue from it.
10. Find the type of baked good responsible for highest total revenue.

CARS dataset

1. Find the total number of car models in the database produced in Japan.
2. Find the total number of models produced by `Ford Motor Company` found in the database.
3. Find the average gas milage for French cars produced between 1974 and 1978.
4. For each country report the average gas milage, average acceleration and average engine displacement for the cars produced by the country's automakers in 1975.
5. Find the country whose 1975 models cars had the best average gas milage.
6. Find all models in the database with the best overall acceleration. Report the full name of the car, the year it was produced and the acceleration.
7. Find the most fuel-efficient 8-cylinder model. Report he full name of the car, the year it was produced and the home country of its maker.
8. For each car maker find the heaviest car it produced. Output the name of the car maker, the full name of the car and its weight.
9. 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 dataset

Here are the queries for the CSU dataset.

1. Find the total undergraduate enrollment in the CSU system in 2005. Report just the number.
2. For each year report the total undergraduate enrollemnt and the total number of CSU campuses which enrolled students that year.
3. For each discipline report average undergraduate and graduate enrollment in 2004 among the CSU campuses (which enrolled students for that discipline).
4. Find the campus with the largest enrollment in 2006. 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 campus with at least 30 years worth of enrollment data report the years of highest and lowest enrollment (over the entire recorded enrollment history). Output the full name of the campus, the year of best (highest) recorded enrollment and the year of worst (lowest) recorded enrollment.
9. For each year from 1960 on 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 $n \geq 1960$ 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 $n - 1$.

MARATHON dataset

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

1. For each gender/age group report the total number of race participants.
2. For each state which had its residents participate in the race report the number of its residents who won in their age/gender group.
3. For each town in Rhode Island report the highest overall place in the race for its resident.
4. For each age, find the race participant with the best overall time in the race. Report the age, the name of the participant and their time and overall place in the race. Sort the output by age.
5. Find all people who ran better than the best runner from the state of Connecticut. Output the name (first, last) of the runner, their hometown and state and the overall place in the race.

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-ilastname.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 `lab6-adekhtyar.zip` or `lab6-adekhtyar.tar.gz`.

The archive shall contain five directories: `CARS`, `CSU`, `BAKERY`, `STUDENTS` and `MARATHON`.

Each directory shall contain the following SQL scripts:

- Database creation script. (e.g., `CARS-setup.sql`). Use the scripts from Lab 2 and (for MARATHON) Lab 3 submissions.
- Table creation scripts (e.g.,`CARS-countries.sql`). Use the scripts from Lab 2/Lab 3. You are welcome to `cat` all scripts together into one big script. If you do, name it `<DATASET>-insert.sql` (e.g., `CARS-insert.sql`).
- The cleanup script (e.g., `CARS-cleanup.sql`). Use the scripts from Lab 2 and Lab 3.
- **NEW script.** One script per database, containing all SQL statements and any SQL*plus statements needed for formatting. Name the script `<DATASET>-query.sql` (e.g., `CARS-query.sql`.