Lab 5: Simple Queries Part II

**Due date:** Thursday, February 19, **midnight**!.

**Note: Lab 6** will be assigned on February 20.

# 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 for each of the information needs outlined below. The information needs can be addressed with either a simple SELECT statement (i.e., a SELECT statement without grouping, aggregation and nested subqueries) and/or with the UNION operation, or with a SELECT statement that uses some aggregation. Please note, that some queries in this lab may require you to immitate the intersection operation with other available means. Each information need **must be met** with a **single** SQL statement.

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:** 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:** This assignment does not use the MARATHON and AIRLINES datasets. You do not need to submit your MARATHON and AIRLINES directories with the assignment.

### STUDENTS dataset

For the STUDENTS dataset, write an SQL script `STUDENTS-lab5.sql` containing SQL statements answering the following information requests.

1. Find all classmates of `JODY NETZEL`. Report first and last name. Do not report `JODY NETZEL`. Sort output in alpabetical order by last name.
2. Find all first-grade students who are NOT taught by OTHA MOYER. Report their first and last names in alphabetical order by last name.
3. Report the total number of third-graders in the school.
4. Find and report the number of students taught by `ALFREDA SUGAI`.

### BAKERY dataset

Write an SQL script `BAKERY-lab5.sql` containing SQL statements answering the following information requests.

**Note:** Your queries must match exactly the wording of the information need. For example, if you are asked to find the price of an `Apricot Tart`, the following query

```
SELECT price
FROM goods
WHERE CODE = '90-APR-PF';
```

is considered to be incorrect because nowhere in the query was the code `'90-APR-PF'` mentioned. (This is especially important when you are expected to produce a join of two or more tables, but instead look up the foreign key value and use it verbatim in the query. Such queries will be marked as incorrect on the spot).

1. Find all dates in the first half of October of 2007 (October 1 to October 15 inclusive) on which one customer made multiple purchases. Report each date exactly once, output dates sorted in ascending order.

2. Find all customers who purchased, during the same trip to the bakery, two different `Croissant`s. Report first and last names of the customers in alphabetical order by their last name.

3. Find all customers who did not make a purchase between October 14 and October 19, 2007 (inclusively). Report their first and last names sorted alphabetically by last name.

4. Find all days on which *either* `ALMETA DOMKOWSKI` made a purchase, *or* someone purchased a `Gongolais Cookie`. Sort dates in chronological order. Each date shall appear exactly once.

5. Report the total amount of money `NATACHA STENZ` spent at the bakery during the month of October, 2007.

6. Report the total amount of money spent by bakery customers in October 2007 on `Cookie`s.

### CARS dataset

Here are the queries for the CARS dataset. Name the SQL scrips `CARS-lab5.sql`

1. Find all cars made after 1980 with gas mileage better than the 1982 `honda civic`. Report full name of the car, year it was made and the name of the manufacturer. Sort output in descending order by gas mileage.

2. Find the average, maximum and minimum horsepower for 4-cylinder vehicles manufactured by `chrysler` between 1971 and 1976 inclusively.

3. Find how many cars produced in 1971 had better acceleration than a 1972 `volvo 145e (sw)`. Report just the number.

4. Find how many different car manufacturers produced a vehicle heavier than 4000 lbs.

### CSU dataset

Here are the queries for the CSU dataset. Name the SQL scrips `CSU-lab5.sql`

1. Report the total number of degrees granted by `California Polytechnic State University-San Luis Obispo` in the period between 1995 and 2000 (inclusively).

2. Find the largest, the smallest and the average fee on a CSU campus in 2005.

3. Report the average student to faculty (use student FTE to faculty FTE ratio) ratio in 2004 among the campuses where 2004 enrollment (FTE numbers) was greater than 15000.

4. Report all years in which *either* there were more than 17000 students (NOT FTEs) on California Polytechnic State University-San Luis Obispo campus, or California Polytechnic State University-San Luis Obispo graduated (gave degrees) to more than 3500 students. Report years in chronological order, with each year reported once.

### INN dataset

For the INN dataset, create a SQL script file `INN-lab5.sql` with SQL queries for the following information needs. (When no year is supplied in the query descriptions below, assume 2010).

1. Find all rooms that were occupied on all three of the following dates: May 15, 2010, August 18, 2010 and December 12, 2010. Report just the full name of the room and the room code. Sort output in alpabetical order by room name.
2. Find the names of all people[1] staying at the inn at the same time as `HERBERT FRYDAY`. Sort the output in alphabetical order by last name.
3. Find the number of August reservations (both checkin and checkout dates are in August) where two adults are staying with two children.
4. Find the average number of nights of stay in the `'Interim but salutary'` room for all reservations that commenced May 1, 2010 or later and ended before August 31, 2010.

## WINE dataset

Create a SQL script `WINE-lab5.sql` containing SQL statements representing the following information needs.

1. List all 2006 vintage wines from Napa (any appellation within the county) whose total revenue exceeds that of the 2006 `'Appelation Series'`[2] Paso Robles Zinfandel from `'Rosenblum'` winery. For each wine report grape, winery and name, score and revenue. Order by revenue.
2. Find all wines produced in the same vintage year as the `Tor Chardonnay`, which have both the higher score and the higher production.
3. Find the average score of a `Paso Robles Zinfandel.` ;
4. Find the total revenue from all red wines made by `Kosta Browne`.
5. Find the average number of cases of a `Pinor Noir` produced from grapes sourced from the `Central Coast`.

[1] We only know the names of the people who made the reservations, so only those names are subject to the query.

[2] There is a typo there. Let it be for now.

6. Report the overall number of different red wines produced in `Russian River Valley` during the year when this AVA had a wine with a score of 98.

### KATZENJAMMER dataset

Create a SQL script `KATZENJAMMER-lab5.sql` containing SQL statements representing the following information needs.

1. Find the number of times Turid played bass balalaika on Katzenjammer songs.
2. Find the number of times Anne-Marit was positioned center stage while Marianne was playing drums.
3. Find the number of times Solveig sang lead while Marianne was performing out front (left, right or center stage).
4. Find the total number of different instruments Solveig played on Katzenjammer songs.
5. List all the instruments that both Marianne and Turid played on (possibly different) Katzenjammer songs.
6. Find how many different performers played guitar.
7. Find on how many songs at least two performers played the same instrument.

## 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 `lab4.ext` where `ext` is one of the extensions above. The archive shall contain seven directories: `CARS`, `CSU`, `INN BAKERY`, `STUDENTS`, `WINE`, `KATZENJAMMER`.

Each directory shall contain the following SQL scripts:

- Database creation script. (e.g., `CARS-setup.sql`). Use the scripts from Lab 2/Lab 4 submissions.
- Table creation script. Use `<DATASET>-insert.sql` (e.g., `CARS-insert.sql`) file from Lab 4 submission.
- The cleanup script (e.g., `CARS-cleanup.sql`). Use the scripts from Lab 2/Lab 4.
- **NEW script.** One script per database, containing all SQL statements and any `SQL*plus` statements needed for formatting. Name the script (as specified above) `<DATASET>-lab5.sql` (e.g., `CARS-lab5.sql`.

Submit using handin:

Section 01:

```
$ handin dekhtyar lab05-01 <file>
```

Section 04:

```
$ handin dekhtyar lab05-04 <file>
```