

## Lab 4: Simple Queries

**Due date:** Monday, October 24, **midnight!**.

**Note:** Lab 5 will be released on October 21 (Friday), however, it's official start is after Lab 4 is due..

## 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 (questions asked in English) for each of the nine 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 a simple SELECT statement (i.e., a SELECT statement without grouping, aggregation and nested subqueries) and/or with UNION statements<sup>1</sup>.

Each information need **must be met** with a **single** SQL statement. DO NOT use grouping (GROUP BY) and aggregation for these queries.

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

---

<sup>1</sup>MySQL does not support ANSI SQL INTERSECT (intersection) and EXCEPT (set difference) operations. We will discuss how to perform these operations using nested SQL queries later in the course.

**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.

## STUDENTS dataset

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

1. Find all students who study in classroom 102. For each student list first and last name. Sort the output by the last name of the student.
2. For each classroom report the grade that is taught in it. Report just the classroom number and the grade number. Sort output by grade in ascending order, then by classroom in ascending order.
3. Find all students taught by BILLIE KRIENER. Output first and last names of students sorted in alphabetical order by their last name.
4. Are there any students who share the same first name. Report all pairs of such students (if found). Each pair must be reported in a single row of output. The output shall contain the first and the last names of both students.
5. For each teacher teaching grades 3 through 6, report the grade (s)he teaches. Each name has to be reported exactly once. Sort the output by grade and alphabetically by teacher's last name for each grade.

## BAKERY dataset

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

**Note: Here, and everywhere else** 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 cakes on the menu. For each item output the flavor, the name (food type) of the item, and the price. Sort your output in descending order by price.
2. Report the prices of the following items:
  - Any chocolate-flavored item priced between \$1 and \$2<sup>2</sup>
  - All meringues
  - All eclairs except the vanilla-flavored one.

Output the flavor, the name (food type) and the price of each pastry. Sort the output in alphabetical order by the flavor and then the name.

3. Find all items purchased by TERRELL ARZT on October 13, 2007. Output the flavor and the name (food type) of each item and its price. Organize the output in alphabetical order by the type of food, then by the flavor. Make sure that you report *each* individual item purchased.
4. Find all different eclairs purchased on October 4, 2007. Each eclair (flavor, food) is to be listed once. Sort output in alphabetical order by the eclair flavor.
5. Find all items on the menu that are cheaper than the one item RAYFORD SOPKO bought on October 27, 2007. For each item report its description (food, flavor) and price. Sort the output in descending order by price.

## CARS dataset

1. Find all Japanese car makers and report their full names in alphabetical order.
2. Report all years for which the database has information about at least one fiat. Each year needs to be reported exactly once, years should be printed in ascending order.
3. For each car produced by Volkswagen between 1980 and 1982 report the year it was produced and its horsepower and milage per gallon. Report the cars in descending order by the horsepower. Identify each car by its full name.
4. Find all non-four cylinder cars produced in 1979 that have fuel economy better than 20 MPG and that accelerate to 60 mph faster than in 15 seconds. Report the name of the car and the full name of the automaker.

---

<sup>2</sup>Unless otherwise specified all "between" instructions are assumed to be inclusive, i.e., in this particular case, the query output shall include all items priced exactly \$1.00 and exactly \$2.00, if such items exist in the database.

5. For each French car released between 1975 and 1978, compute the ratio between the weight of the car and its number of horsepower. Report the full name of the car, the year it was produced and the ratio sorted in descending order by the ratio.

## CSU dataset

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

**Note:** See note for the **BAKERY** dataset. For this dataset, you must use the name of the campus in the query, whenever the name is provided. It is **incorrect** to replace the name of the campus with the campus id number.

1. Find the fees paid by students at **California Polytechnic State University-San Luis Obispo** in 2000. Report just the amount.
2. For each year between 1960 and 1969 (inclusive) report the number of students who were enrolled at **California State University-Los Angeles** (report the actual number of students, NOT the FTEs). Output the year and the number of enrolled students sorted by year.
3. For each university<sup>3</sup> with more than 17,000 enrolled FTE students in 2003 report the FTE enrollment, the number of faculty and the student-to-faculty ratio. Sort the output in ascending order by the student-to-faculty ratio<sup>4</sup>.
4. Find all disciplines and campuses where graduate enrollment in 2004 was at least three times higher than undergraduate enrollment (and the undergraduate enrollment was non-zero). Report campus names and discipline names. Sort output by campus name, then by discipline name in alphabetical order.
5. Report the total amount of money collected from student fees (use the full-time equivalent enrollment for computations) at '**Sonoma State University**' for each year between 2002 and 2004 inclusively, and the amount of money collected from student fees per one full-time equivalent faculty. Output the year, the two computed numbers sorted chronologically by year.
6. Find all years when the number of graduates from **California Polytechnic State University-San Luis Obispo** (i.e., the number of degrees granted) was more than the combined number of degrees granted by of **California State University-East Bay** and **Sonoma State University**. Report the year, the number of degrees granted by Cal Poly and the combined number of degrees granted by CSU East Bay and Sonoma State. Output results in chronological order.

---

<sup>3</sup>Any time you see "For each university" in the specification of an information need, you must identify the universities in the output by the full name of the campus.

<sup>4</sup>Use student FTE numbers in ratio computation.

## INN dataset

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

**Note:** If the full name of a room is provided in the question, you cannot replace with with a three-letter code in the text of the query.

1. Find all rooms with two Queen beds with the base price of over \$160 per night. Report room names and the decor style in alphabetical order by the room code.
2. Find all October reservations (a.k.a., all reservations that both start AND end in October) for the 'Thrift and accolade' room where the rate was smaller than the base for the room. For each reservation report the last name of the person who reserved it, checkin and checkout dates, the total number of people staying and the daily rate. Output reservations in chronological order.
3. Find all rooms occupied on October 24, 2010. Report full name of the room, the check-in and checkout dates of the reservation. Sort output in alphabetical order by room name.
4. For each stay of GLEN DONIGAN in the hotel, calculate the total amount of money, he paid. Report reservation code, checkin and checkout dates, room name (full) and the total amount of money the stay cost. Sort output in chronological order by the day of arrival.
5. Report all reservations in rooms with double beds that contained four adults. For each reservation report its code, the full name and the code of the room, check-in and check out dates. Report reservations in chronological order, and sorted by the three-letter room code (in alphabetical order) for any reservations that commenced on the same day.

## MARATHON dataset

For this dataset, all times must be shown in the output in the same format as in the original dataset (in the file `marathon.csv`). Also, please give time and pace columns in your output appropriate column headers (in your SQL commands). The information needs are below. Name the file `MARATHON-info.sql`.

1. Report the overall place and the run time of the person who took the second place in the 50-59 age group for men.
2. Report names (first, last), times, overall places as well as places in their gender-age group for all male runners from CAMBRIDGE, MA. Sort output by overall place in the race.

3. Find all female runners from Massachusetts who were faster than the male runner who took the 50th place in the 40–49 age group. For each runner report her full name (first, last), age group, place in the race, place in her age group, and home town. Sort results in ascending order by overall place.
4. Find all duplicate bibs in the race. Report just the bib numbers. Sort in ascending order of the bib number. Each duplicate bib number must be reported exactly once.
5. List all runners who took first place and second place in their respective age/gender groups. For age group, output name (first, last) and age for both the winner and the runner up (in a single row). Order the output by gender, then by age group.

### AIRLINES dataset

For the AIRLINES dataset, create a SQL script file `AIRLINES-info.sql` with SQL queries for the following information needs. You may not substitute numeric codes for airlines in the place of airline names in the queries below. You may use three-letter airport abbreviations whenever they are used in the questions.

1. Find all airlines that have at least one flight out of KKI airport. Report the full name and the abbreviation of each airline. Report each name only once. Sort the airlines in alphabetical order.
2. Find all destinations served from the KKI airport by UAL. Report flight number, airport code and the full name of the airport. Sort in ascending order by flight number.
3. Find all *other* destinations that are accessible from AXX on only **Northwest** flights with **exactly one** change-over. Report pairs of flight numbers, airport codes for the final destinations, and full names of the airports sorted in alphabetical order by the airport code.
4. Report all pairs of airports served by both USAir and AirTran. Each pair must be reported exactly once (if a pair X,Y is reported, than a pair Y,X is redundant and should not be reported).
5. Find all airports that are served by at least four flights to different airports from the same airline. Report just the three-letter codes of the airports — each code exactly once, in alphabetical order. Also report the abbreviated name of the airline (note: if an airport has more than one airline with four flights that match the condition of the query, the airport will be reported twice)

## WINE dataset

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

1. List all AVAs located in Santa Barbara county. Output just the names of the AVA appellations and sort them in alphabetical order.
2. Find all *other* wines produced by the winery that made the only **Viognier** in the database. For each wine report the grape, the name the vintage and the score. Sort output in ascending order by vintage, and alphabetically by name of wine for the wines of the same vintage.
3. A wine collector rates the wines using a specially designed score computed using the following formula:

$$\text{wineScore}(\text{wine}) = \frac{20 \cdot (\text{Score} - 88) + 10}{\text{Price}} \cdot \frac{1200}{\text{Cases}}.$$

Compute the collector's wine ratings for each 2008 **Grenache**. Output the winery and the name of the wine, price, score and number of cases, and the collector's rating. Sort in descending order by the rating.

4. List all vintage years in which at least one **Zinfandel** from Sonoma County (any appellation) scored above 92. Each year needs to be reported once. Sort in chronological order.
5. A case of wine is 12 bottles. For each **Cabernet Sauvignon** with a score of 96 or higher compute the total revenue assuming that all the wine sold at the specified price. Report the winery, name of the wine, its vintage, wine score and overall revenue. Sort in descending order by revenue. Exclude NULL values.
6. Compute the total price of a bottle of **Kosta Browne's Koplen Vineyard 2008 Pinot Noir**, two bottles of **Darioush's 2007 Darius II Cabernet Sauvignon** and a bottle of **Kistler's McCrea Vineyard 2006 Chardonnay**. Report just the one number.

## KATZENJAMMER dataset

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

1. Report, in order, the tracklist for 'Rockland'. Output just the names of the songs in the order in which they occur on the album.
2. List the instruments each performer plays on 'Shine Like Neon Rays'. Output the first name of each performer and the instrument, sort alphabetically by the first name.

3. List all instruments played by Turid at least once during the performances. Report the instruments in alphabetical order (each instrument needs to be reported exactly once).
4. Find all songs that featured ukulele playing by Solveig. Report song titles in alphabetical order.
5. Find all instruments Marianne ever played on the songs where she sang lead vocals. Report the names of instruments in alphabetical order (each instrument needs to be reported exactly once).
6. Find a song on which Anne-Marit played three different instruments. Report the name of the song. (The name of the song shall be reported exactly once)
7. In the order of columns right - center - back - left, report the positioning of the band during 'A Bar In Amsterdam'. (just one record needs to be returned with four columns containing the first names of the performers who were staged at the specific positions during the song).
8. For each song on which Turid played `balalaika bass` and Anne-Marit was positioned on the left side of the stage<sup>5</sup> report the lead vocalist. Report the song title and the first name of the vocalist. Sort in alphabetical order by the song title.

## 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 nine directories: AIRLINES, CARS, CSU, INN BAKERY, STUDENTS, MARATHON, KATZENJAMMER, and WINE.

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. The contents of the file may be the merged contents of all `<DATASET>-build-<table>.sql` scripts (in appropriate order), or, a list of the following SQL commands:

```
source <DATASET>-build-<table>.sql
```

for each table in the dataset. For example, for the INN dataset, the `INN-insert.sql` file can be:

---

<sup>5</sup>Here, and everywhere, stage positions are from the perspective of the band facing the audience.

```
--- CSC 365
--- Lab 4
--- your name here
```

```
source WINE-build-grapes.sql
source WINE-build-appellations.sql
source WINE-build-wine.sql
```

- The `<DATASET>-build-<table>.sql` files from Lab 2/Lab 3, if your `<DATASET>-insert.sql` file needs them.
- The cleanup script (e.g., `CARS-cleanup.sql`). Use the scripts from Lab 2/Lab 3.
- **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>-info.sql` (e.g., `CARS-info.sql`).

**Note.** If you want to update the INSERT statements in your `<DATASET>-build-<table>.sql` files (some of you wanted to be able to insert multiple tuples in a single INSERT statement), feel free to do it now.

Submit using handin:

```
$ handin dekhtyar lab04 <file>
```