Lab 5: Simple Queries

Due date: May 8$^{\text{th}}$.

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

NOTE: You **MUST** provide a comment on the line before each SELECT statement. The comment must denote the number of each query. Use "`-- Q#`", where "`#`" is the number of the query. If for any reason you are missing a query, just leave out the query and the comment. Improperly labeling you queries can confuse the grading script.

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.

AIRLINES dataset

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 `AXX` 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 `AXX` airport by `Northwest`. 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 `Frontier` and `JetBlue`. 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). For each pair, report both airport codes.

5. Find all airports served by ALL five of the airlines listed below: `Delta`, `Frontier`, `USAir`, `UAL` and `Southwest`. Report just the airport codes, sorted in alphabetical order.

6. Find all airports that are served by at least three `Southwest` flights. Report just the three-letter codes of the airports — each code exactly once, in alphabetical order.

BAKERY dataset

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 chocolate-flavored items on the menu whose price is under $5.00. 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. Find the prices of the following items:

   - any cookie priced above $1.10;
   - any lemon-flavored items;
   - any apple-flavored item except for the pie.

   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 customers who made a purchase on October 3, 2007. Report the name of the customer (first, last). Sort the output in alphabetical order by the customer's last name. Each customer name must appear at most once.

4. Find all different cakes purchased on October 4, 2007. Each cake (flavor, food) is to be listed once. Sort output in alphabetical order by the cake flavor.

5. List all pastries purchased by `ARIANE CRUZEN` on October 25, 2007. For each pastry, specify its flavor and type, as well as the price. Output the pastries in the order in which they appear on the receipt (each pastry needs to appear the number of times it was purchased).

6. Find all types of cookies purchased by `KIP ARNN` during the month of October of 2007. Report each cookie type (flavor, food type) exactly once in alphabetical order by flavor.

CARS dataset

1. Find all models[1] produced by Chrysler (`'chrysler'`) stored in the database. Report just the names of the models in alphabetical order.

2. Find all Reanults (`'renault'`) in the database. For each, report the make and the year. Sort output by year.

3. Find all cars produced by `Volvo` between 1975 and 1980 (inclusive). Report the make of the car and the year it was produced, sort output in ascending order by the year.

4. Find all cars produced in 1981 that are heavier than 3000 lbs, with acceleration *better* than 17 seconds. Report the name of the car and the name of the automaker.

5. Find all non-European car makers which produced at least one light (weight less than 2000lbs) car between 1979 and 1981 (inclusively). Output the full name of the company and its home country. Each company should be reported just once.

6. For each `mazda` released after 1978, compute the ratio between the weight of the car and its number of horsepowers. Report the full name of the car, the year it was produced, and the ratio sorted in descending order by the ratio.

7. Report the names of all countries other than `'usa'` which produced at least one 6-cylinder car.

CSU dataset

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. Report all campuses that were created in 1947. Output only the name of campus, sort by campus id.

2. For each year between 1994 and 2000 (inclusive) report the number of students who graduated from `California State University-San Marcos`. Output the year and the number of degrees granted. Sort output by year.

3. Report undergraduate and graduate enrollments (as two numbers) in `'Mathematics'`, `'Engineering'` and `'Computer and Info. Sciences'` disciplines for both Polytechnic universities of the CSU system in 2004. Output the name of the campus, the discipline, and the number of graduate and undergraduate students enrolled. Sort output by campus name, and by discipline for each campus.

---

[1]Just models, not makes.

4. For each university with more than 20,000 full-time equivalent students (FTE) enrolled in 2003, report the student to faculty ratio that year. Report name of campus, the full-time equivalent enrollment, number of faculty, and the students-to-faculty ratio [2]. Sort output in descending order by students-to-faculty ratio. For this query, give each column a meaningful heading.

5. For each `Los Angeles` county CSU campus report the percentage of students who graduated in 2002. Use the total number of enrolled students (NOT full-time equivalents) for this query. Report campus name and the percent (presented as a number between 0 and 100). Sort output in descending order by the percentage of graduates.

6. Find all disciplines and campuses where graduate enrollment in 2004 was at least three times higher than undergraduate enrollment. Report campus names and discipline names. Sort output by campus name, then by discipline name in alphabetical order.

7. Report the total amount of money collected from student fees (use the full-time equivalent enrollment for computations) at `'Fresno 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 and the two computed numbers sorted chronologically by year.

INN dataset

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 `modern` rooms with a base price below $160 and two beds. Report room names and codes in alphabetical order by the code.

2. Find all August 2010 reservations (a.k.a., all reservations that both start AND end in August) for the 'Convoke and sanguine' 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 February 6, 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 `GRANT KNERIEN` 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. For each reservation that starts on December 31, 2010 report the room name, nightly rate, number of nights spent, and the total amount of money paid. Sort output in descending order by the number of nights stayed.

6. 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

---

[2] Hint: it should be a number greater than 1.

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.

7. List the duration of each stay in the 'Immutable before decorum' room that commenced in the month of November 2010. Report the reservation code, the check in date and the duration (number of nights) of stay. Sort in chronological order by the check in date.

MARATHON dataset

Note: Unless specified, leave all temporal types in their default format.

1. Report the time, pace and the overall place of TEDDY BRASEL.

2. Report names (first, last), times, overall places as well as places in their gender-age group for all female runners from QUNICY, MA. Sort output by overall place in the race.

3. Find the results for all 34-year old female runners from Connecticut (CT). For each runner, output name (first, last), town, and the running time. Sort by time.

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.

STUDENTS dataset

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, and by classroom within each grade.

3. Find all teachers who teach in kindergarten. Report first and last name of the teachers and the room number. Sort the output by room number.

4. Find all students taught by KIRK MARROTTE. Output first and last names of students sorted in alphabetical order by their last name.

5. For each teacher, 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. Report the teacher's name (first and last) and the grade.

WINE dataset

1. List all AVAs located in Monterey county. Output just the names of the AVA appellations and sort them in alphabetical order.

2. List all white grape varieties for which at least one wine of the 2008 vintage is rated at 90 points or above in the database. Each grape variety needs to be reported once. Sort the output in alphabetical order.

3. List all Sonoma county appellations for which the database contains at least one rating for a `'Grenache'`. For each appellation list its name and county. Sort output in alphabetical order by county, then by appellation name. Report each appellation once.

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 `Carlisle` (name of the winery) `Syrah` compute the total revenue assuming that all the wine sold at the specified price. Report the name of the wine, its vintage wine score and overall revenue. Sort in descending order by revenue. Exclude NULL values.

6. Find all AVAs with Santa Barbara county, which produced at least one red wine and at least one white wine in the same vintage, both rated above 90. Report the names of AVAs in alphabetical order.

7. 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 Sauvingnon`, and a bottle of `Kistler`'s `McCrea Vineyard` 2006 `Chardonnay`. Report just the one number.

Submission Instructions

Please, follow these instructions exactly. Up to 20% of the Lab 5 grade will be assigned for conformance to the assignment specifications, **including the submission instructions**.

Be sure to properly comment your queries (see the comment in "The Task" section.

Please, **name your files exactly as requested** (including capitalization). Correct submission simplifies grading, and ensures its correctness.

There should be one file per dataset named: "¡DATASET¿-query.sql". E.g. for the AIRLINES dataset, the file should be "AIRLINES-query.sql". (See the handin command for proper naming.)

Please include your name and Cal Poly email address in all files you are submitting. If you are submitting code/scripts, include, at the beginning of the file a few comment lines with this information. Files that cannot be authenticated by observing their content will result in penalties assessed for your work.

No archives are expected for this lab. You are expected to handin one file per dataset and an optional README.txt including any comments about this lab. Submit your files using the following `handin` command:

$ handin eaugusti lab05 AIRLINES-query.sql BAKERY-query.sql CARS-query.sql CSU-query.sql INN-query.sql MARATHON-query.sql STUDENTS-query.sql WINE-query.sql [README.txt]