Lab 4: Simple Queries

Due date: Friday, May 12, **midnight**!.

Note: Lab 5 will be released on May 12 in class.

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 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[1].

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 and further SQL labs you will be working with the instructor's versions of the databases from Labs 2 and 3. The read-only

[1]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.

versions of these databases are available on the MySQL server as databases with the following names:

```
airlines
bakery
cars
csu
inn
katzenjammer
marathon
students
wine
```

Each of you has been granted the `SELECT` priveleges on each of these databases (contact the instructor if this is not the case). Your queries must be written for the tables in these databases and must run properly on them and produce correct output.

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

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. Use indentation where possible.

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 111. 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 classroom in descending order.

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

4. Find all students taught by `OTHA MOYER`. Output first and last names of students sorted in alphabetical order by their last name.

5. For each teacher teaching grades K through 3, 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 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. Report 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 Reanults (`'renault'`) in the database. For each, report the name and the year. Sort output by year.
2. Find all cars produced by `Volvo` between 1977 and 1981 (inclusive). Report the name of the car and the year it was produced, sort output in ascending order by the year.
3. Report all Asian automakers. Output the full name of the automaker and the country of origin sorted alphabetically by the full name of the automaker.
4. Find all non-four cylinder cars produced in 1980 that have a better 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 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 `saab` 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.

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. Report all campuses from Los Angeles county. Output the full name of campus in alphabetical order.
2. For each year between 1994 and 2000 (inclusive) report the number of students who graduated from `California Maritime Academy` 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 the number of undergraduate students enrolled. Sort output by campus name, and by discipline for each campus.

4. Report graduate enrollments in 2004 in `'Agriculture'` and `'Biological Sciences'` for any university that offers graduate studies in both disciplines. Report one line per university (with the two grad. enrollment numbers in separate columns), sort universities in descending order by the number of `'Agriculture'` graduate students.

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

6. 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, the two computed numbers sorted chronologically by year.

7. Find all campuses where enrollment in 2003 (use the FTE numbers), was higher than the 2003 enrollment in `'San Jose State University'`. Report the name of campus, the 2003 enrollment number, the number of faculty teaching that year, and the student-to-faculty ratio. Sort output in ascending order by student-to-faculty ratio.

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 `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 July 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 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 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.

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 `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).
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 alphebetical 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 excactly once, in alphabetical order.

WINE dataset

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

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. 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.

KATZENJAMMER dataset

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

1. Report, in order, the tracklist for `'Le Pop'`. 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 `'Mother Superior'`. Output the first name of each performer and the instrument, sort alphabetically by the first name.

3. List all instruments played by Anne-Marit 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 ukalele playing (by any of the performers). Report song titles in alphabetical order.

5. Find all instruments Turid 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 all songs where the lead vocalist is not positioned center stage. For each song, report the name, the name of the lead vocalist (first name) and her position on the stage. Output results in alphabetical order by the song. (Note: if a song had more than one lead vocalist, you may see multiple rows returned for that song. This is the expected behavior).

7. 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)

8. 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).

Submission Instructions

You must submit nine `<DATASET>-info.sql` files. In addition, submit a simple `README` file with your name and email address.

There is no need to submit any other files.

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.

When unpacked, your archive must place the nine `<DATASET>-info.sql` files into the current directory (i.e., in the root of your `handin` directory for Lab 4 submission). No other subdirectories are needed.

Submit using handin:

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