University of California, Los Angeles
Department of Computer Science

Computer Science 143 **Prof. Ryan Rosario**

Homework 2

Due Tuesday, May 1, 2018 11:59pm ***via CCLE***

Please remember the following:

1. Homework is mostly graded on completion. We may grade a few parts, but it will never be the majority of the grade on the assignment. So try your best, and focus on solving the problems. Consider homework (and studying the solutions) as practice for the midterm.
2. Homework must be submitted digitally, on CCLE. We will not do any paper grading. You can use a text file, but if you use Word, a PDF is preferred rather than a DOC file.
3. If there are any exercises that are difficult to do digitally (such as diagrams or math), consider scanning your drawing or math, or using a graphics program (even a readable MS Paint is fine) or Equation Editor.
4. **For the sanity of the grader** we will ask you to run the queries and submit the result. You may lose points if you only provide a query.
5. Solutions will be posted.

Part 1: Text, Joins and Subqueries

For some reason, your instructor has been scraping the Caltrans website every 15 minutes or so, since 2015, to get road conditions on all of the highways within California. The data is written to MySQL. **Your version of the data is hourly, and only for 2017.**

A Caltrans highway conditions report looks like the following and contains conditions for individual stretches of highway ("area") typically representing a coarse area of the state: Northern, Southern, Central, Sierra Nevada etc.

```
SR 120
[IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA]
IS CLOSED FROM CRANE FLAT TO 5 MI WEST OF THE JCT OF US 395 /TIOGA PASS/
(TUOLUMNE, MONO CO) - FOR THE WINTER - MOTORISTS ARE ADVISED TO USE AN
ALTERNATE ROUTE

[YOSEMITE NAT'L PARK]
FOR YOSEMITE NAT'L PARK ROAD INFORMATION CALL 209-372-0200
```

The schema for the `caltrans` table looks like the following:

```
CREATE TABLE ‘caltrans‘ (
    reported TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    highway  VARCHAR(6) NOT NULL,
    area     VARCHAR(255),
    text     TEXT NOT NULL,
    hash     VARCHAR(32) NOT NULL
);
```

`reported` is the time the data was scraped, `highway` is the highway the status pertains to prefixed by its type (i.e. US101, SR1, I405), area refers to a particular part of the state or highway, and text is the update itself. Since we cannot use `text` as a primary key, a `hash` column was added.

US Highway (US)

California State Route (SR)

Interstate (I)

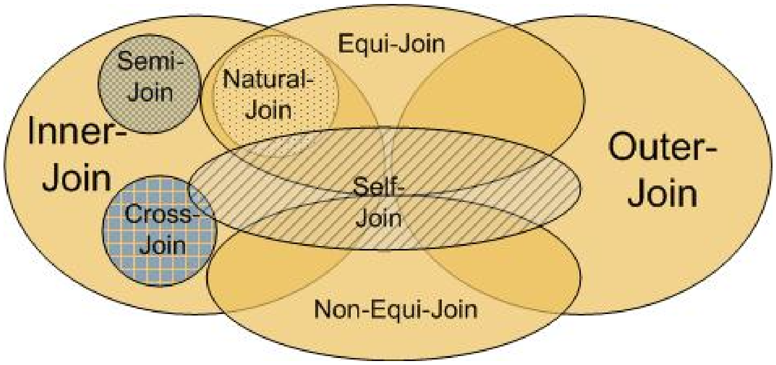
Exercises

(a) Write a query that returns a list of all the highways in 2017 that were closed due to snow at any point of the year, or were closed for the winter. Order them by `highway` and `area` and give us the top 20 results, both columns in descending order. (**Hint 1:** You don't need to do anything with dates to answer this question. **Hint 2:** Before writing a query, look at the data.)

(b) For each highway found in part (a), compute the percentage of days out of the year that the highway was closed. If a highway was closed for only a partial day, it counts as a full day. There are at least three ways to solve this problem. Try to use a method that involves a join, and a method that does not. Report the highway, area/stretch, the percentage of days it was closed in descending order by percentage, and only gives us the 5 highest percentages and the highways and areas they belong to. You may hardcode the number of days in the year (see the note below).

Note that not all of your responses with be perfect in theory because there were times when the instructor's script lost Internet connectivity, or the power went out. So, there may not be exactly 365 days in 2017, but we can just assume it for simplicity. You can also try to find the number of days represented in this dataset yourself if you wish, and use that number instead. Of course, if you want to write much more complicated queries, you can use a subquery to compute the number of days, but in this particular case it is a rabbit hole.

Part 2: Join Definitions



Exercises

(a) Your instructor almost included the above Venn Diagram in his lecture slides to show how different types of joins are related, but he noticed that it was wrong in at least one way. Explain at least one thing that is wrong about the diagram.

Part 3: More Joins and Subqueries

In Homework 1, we did several things with the Bird Scooter use case, but we did not have any data to practice writing queries with. Suppose we now have trip data in the following two tables:

1. `trip_starts`;
2. `trip_ends`;

Exercises

(a) Write a query that computes the elapsed time of each trip. If something happened and a trip end was not recorded, the elapsed time shall be 24 hours, per Bird's policy. Print your results as `trip_id`, `user_id`, and `trip_length`. Only show the first 5, without any special ordering.

(b) Write a query that computes the charge to the user for each trip. The charge is calculated as follows: $1 flat rate per trip plus 15 cents per minute. All fractional minutes are rounded up to the next minute. Assume we did not store the results of the query from part (a). Print the first 5 results (no ordering) as `trip_id`, `user_id` and `trip_charge`.

(c) Putting it all together: Suppose we bill the user at the end of the month rather than at the end of each trip. Write a query that computes the monthly charge for trips in March 2018 for each user assuming we did not store the results from parts (a) or (b). Print your results: `user_id` and `monthly_total` for the first five users (no ordering). In particular, how much does `user_id = 2` owe?

(d) In the solution set for Homework 1, it was mentioned that another way we can record starts and ends of trips was to use one table, 2 rows per trip: one row representing the start and a second row representing the end of the trip. We would then have an `ENUM` or `BIT` that specifies whether the row refers to a start or an end. If we wanted to use this one single table as the basis to charge users, what type of join would we need to compute?