Raymond Lin
304937942
CS 143

Homework 2 Writeup
Part 1
a)

```
SELECT DISTINCT highway, area
FROM hw2.caltrans
WHERE condition LIKE '% CLOSED % SNOW %' OR condition LIKE '% CLOSED %
WINTER %'
ORDER BY highway, area
LIMIT 20;
```

```
 highway |                    area
---------+-----------------------------------------------
 I5      | IN THE NORTHERN CALIFORNIA AREA
 SR108   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR120   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR130   | IN THE CENTRAL CALIFORNIA AREA
 SR138   | IN THE SOUTHERN CALIFORNIA AREA
 SR158   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR168   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR172   | IN THE NORTHERN CALIFORNIA AREA
 SR18    | IN THE SOUTHERN CALIFORNIA AREA
 SR2     | IN THE SOUTHERN CALIFORNIA AREA
 SR20    | IN THE NORTHERN CALIFORNIA AREA
 SR203   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR267   | IN THE NORTHERN CALIFORNIA AREA
 SR270   | IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA
 SR3     | IN THE NORTHERN CALIFORNIA AREA
 SR33    | IN THE SOUTHERN CALIFORNIA AREA
 SR330   | IN THE SOUTHERN CALIFORNIA AREA
 SR38    | IN THE SOUTHERN CALIFORNIA AREA
 SR4     | IN THE CENTRAL CALIFORNIA AREA
 SR88    | IN THE CENTRAL CALIFORNIA & SIERRA NEVADA
```

b)

```
SELECT
        stretch,
```

```
        closed_days * 100 / (COUNT(DISTINCT EXTRACT (DOY FROM
hw2.caltrans.reported)))::double precision AS percentage
    FROM (
            SELECT
                COUNT(DISTINCT EXTRACT (DOY FROM reported))::double
precision AS closed_days,
                (highway, area) AS stretch
            FROM hw2.caltrans
            WHERE (condition LIKE '% CLOSED % SNOW %' OR condition LIKE
'% CLOSED % WINTER %')
            GROUP BY stretch
        ) sq,
        hw2.caltrans
    WHERE sq.stretch = stretch
    GROUP BY stretch, closed_days
    ORDER BY percentage DESC
    LIMIT 5;

                stretch                |    percentage
    -----------------------------------------------------------+------------------
     (SR89,"IN THE NORTHERN CALIFORNIA AREA & SIERRA NEVADA") |
73.780487804878
     (SR120,"IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA") |
68.5975609756098
     (SR203,"IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA") |
68.2926829268293
     (SR108,"IN THE CENTRAL CALIFORNIA AREA & SIERRA NEVADA") |
61.890243902439
     (SR4,"IN THE CENTRAL CALIFORNIA AREA")          | 60.9756097560976
```

Part 2

a)

Cross-join should not be a subset of inner-join. Cross-join is the cartesian product of all tuples in two relations.

As a result, the number of values in the set generated by a cross-join will always be greater than or equal to the

number of values in the set generated by an innter-join. Thus, inner-join should actually be a subset of the cross-join.

Part 3

a)

```
SELECT
        trip_start.trip_id,
        trip_start.user_id,
        CEIL(EXTRACT(EPOCH FROM age(COALESCE(hw2.trip_end.time,
hw2.trip_start.time + make_interval(days := 1)), hw2.trip_start.time)) / 60)
                AS trip_length
FROM hw2.trip_start LEFT OUTER JOIN hw2.trip_end
        ON hw2.trip_end.trip_id = hw2.trip_start.trip_id
        AND hw2.trip_end.user_id = hw2.trip_start.user_id
LIMIT 5;
```

trip_id	user_id	trip_length
0	20685	2
2	34808	3
3	25463	1440
4	26965	2
5	836	1

-note: trip_length in minutes

b)

```
SELECT
        trip_start.trip_id,
        trip_start.user_id,
        1 + 0.15*(CEIL(EXTRACT(EPOCH FROM
age(COALESCE(hw2.trip_end.time, hw2.trip_start.time + make_interval(days := 1)),
hw2.trip_start.time)) / 60))
                AS trip_charge
FROM hw2.trip_start LEFT OUTER JOIN hw2.trip_end
        ON hw2.trip_end.trip_id = hw2.trip_start.trip_id
        AND hw2.trip_end.user_id = hw2.trip_start.user_id
LIMIT 5;
```

trip_id	user_id	trip_charge
0	20685	1.3
2	34808	1.45
3	25463	217
4	26965	1.3
5	836	1.15

c)

```
SELECT
    trip_start.user_id,
    SUM (
        CASE WHEN
            (1 + 0.15*
                CEIL
                (
                    (
                    EXTRACT
                        (
                        EPOCH FROM age
                            (
                            COALESCE(hw2.trip_end.time,
hw2.trip_start.time + make_interval(days := 1)), hw2.trip_start.time
                            )
                        ) / 60
                    )
                )
            ) > 100 THEN 100
            ELSE
            (1 + 0.15*
                CEIL
                (
                    (
                    EXTRACT
                        (
                        EPOCH FROM age
                            (
                            COALESCE(hw2.trip_end.time,
hw2.trip_start.time + make_interval(days := 1)), hw2.trip_start.time
                            )
                        ) / 60
                    )
                )
            )
        END
        )
        AS monthly_charge
FROM hw2.trip_start LEFT OUTER JOIN hw2.trip_end
```

```
        ON hw2.trip_end.trip_id = hw2.trip_start.trip_id
        AND hw2.trip_end.user_id = hw2.trip_start.user_id
    WHERE (EXTRACT(MONTH FROM hw2.trip_start.time) IN (3)) AND
(EXTRACT(YEAR FROM hw2.trip_start.time) IN (2018))
    GROUP BY trip_start.user_id
    LIMIT 5;
```

```
 user_id | monthly_charge
---------+----------------
       0 |          105.5
       1 |           4.05
       2 |         314.05
       3 |           11.9
       4 |         210.55
```

For user_id = 2, the monthly charge is $314.05.

d)

We could use a self left-outer join on user_id, trip_id, and where the enum bits are opposite (which means one represents the start and the other represents the end). Then, the start records whose corresponding end records had been lost, will have a NULL value for end, as it is a left outer join.