1.

Print the *company\_name* field. Find the number of taxi rides for each taxi company for November 15-16, 2017, name the resulting field *trips\_amount* and print it, too. Sort the results by the *trips\_amount* field in descending order.

SELECT cabs.company\_name,

COUNT(trips.trip\_id) AS trips\_amount

FROM cabs

INNER JOIN trips ON trips.cab\_id = cabs.cab\_id

WHERE CAST(trips.start\_ts AS date) BETWEEN '2017-11-15' AND '2017-11-16'

GROUP BY company\_name

ORDER BY trips\_amount DESC;

2.

Find the number of rides for every taxi companies whose name contains the words "Yellow" or "Blue" for November 1-7, 2017. Name the resulting variable *trips\_amount.* Group the results by the *company\_name* field.

SELECT

trips.start\_ts,

CASE

WHEN weather\_records.description LIKE '%rain%' OR weather\_records.description LIKE '%storm%' THEN 'Bad'

ELSE 'Good'

END AS weather\_conditions,

trips.duration\_seconds

FROM

trips

JOIN

weather\_records ON trips.start\_ts = weather\_records.ts

WHERE

trips.pickup\_location\_id = 50

AND trips.dropoff\_location\_id = 63

AND EXTRACT(DOW FROM trips.start\_ts) = 6

ORDER BY

trips.trip\_id;

3.

For November 1-7, 2017, the most popular taxi companies were Flash Cab and Taxi Affiliation Services. Find the number of rides for these two companies and name the resulting variable *trips\_amount.* Join the rides for all other companies in the group "Other." Group the data by taxi company names. Name the field with taxi company names *company*. Sort the result in descending order by *trips\_amount*.

SELECT

CASE

WHEN cabs.company\_name LIKE '%Flash Cab%' THEN 'Flash Cab'

WHEN cabs.company\_name LIKE '%Taxi Affiliation Services%' THEN 'Taxi Affiliation Services'

ELSE 'Other'

END AS company,

COUNT(trips.trip\_id) AS trips\_amount

FROM

trips

INNER JOIN

cabs ON cabs.cab\_id = trips.cab\_id

WHERE

CAST(trips.start\_ts AS DATE) BETWEEN '2017-11-01' AND '2017-11-07'

GROUP BY

company

ORDER BY

trips\_amount DESC;

4.

Retrieve the identifiers of the O'Hare and Loop neighborhoods from the *neighborhoods* table.

SELECT

name,

neighborhood\_id

FROM

neighborhoods

WHERE

name = 'O''Hare'

OR name = 'Loop';

5.

For each hour, retrieve the weather condition records from the *weather\_records* table. Using the CASE operator, break all hours into two groups: Bad if the *description* field contains the words rain or storm, and Good for others. Name the resulting field *weather\_conditions*. The final table must include two fields: date and hour (*ts*) and *weather\_conditions*.

SELECT

ts,

CASE

WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'

ELSE 'Good'

END AS weather\_conditions

FROM

weather\_records;

6.

Retrieve from the *trips* table all the rides that started in the Loop (*pickup\_location\_id:* 50) on a Saturday and ended at O'Hare (*dropoff\_location\_id*: 63). Get the weather conditions for each ride. Use the method you applied in the previous task. Also, retrieve the duration of each ride. Ignore rides for which data on weather conditions is not available.

The table columns should be in the following order:

* *start\_ts*
* *weather\_conditions*
* *duration\_seconds*

Sort by *trip\_id.*

SELECT

trips.start\_ts,

CASE

WHEN weather\_records.description LIKE '%rain%' OR weather\_records.description LIKE '%storm%' THEN 'Bad'

ELSE 'Good'

END AS weather\_conditions,

trips.duration\_seconds

FROM

trips

JOIN

weather\_records ON trips.start\_ts = weather\_records.ts

WHERE

trips.pickup\_location\_id = 50

AND trips.dropoff\_location\_id = 63

AND EXTRACT(DOW FROM trips.start\_ts) = 6

ORDER BY

trips.trip\_id;