# *SQL project*

### Description of the data

A database with info on taxi rides in Chicago:

neighborhoods table: data on city neighborhoods

* name: name of the neighborhood
* neighborhood\_id: neighborhood code

cabs table: data on taxis

* cab\_id: vehicle code
* vehicle\_id: the vehicle's technical ID
* company\_name: the company that owns the vehicle

trips table: data on rides

* trip\_id: ride code
* cab\_id: code of the vehicle operating the ride
* start\_ts: date and time of the beginning of the ride (time rounded to the hour)
* end\_ts: date and time of the end of the ride (time rounded to the hour)
* duration\_seconds: ride duration in seconds
* distance\_miles: ride distance in miles
* pickup\_location\_id: pickup neighborhood code
* dropoff\_location\_id: dropoff neighborhood code

weather\_records table: data on weather

* record\_id: weather record code
* ts: record date and time (time rounded to the hour)
* temperature: temperature when the record was taken
* description: brief description of weather conditions, e.g. "light rain" or "scattered clouds"

A screenshot of a computer

Description automatically generated

**Task 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.

**Code:**

SELECT

cabs.company\_name AS company\_name,

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 '2017-11-15' AND '2017-11-16'

GROUP BY

cabs.company\_name

ORDER BY

trips\_amount DESC;

**Task 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.

**Code:**

SELECT

cabs.company\_name AS company\_name,

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

FROM

trips

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

WHERE

(cabs.company\_name LIKE '%Yellow%' OR cabs.company\_name LIKE '%Blue%')

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

GROUP BY

cabs.company\_name;

**Task 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*.

**Code:**

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;

**Task 4**: Retrieve the identifiers of the O'Hare and Loop neighborhoods from the *neighborhoods* table.

**Code:**

SELECT

name,

neighborhood\_id

FROM

neighborhoods

WHERE

name = 'O''Hare'

OR name = 'Loop';

**Task 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*.

**Code:**

SELECT

ts,

CASE

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

WHEN weather\_records.description Like '%storm%' THEN 'Bad'

ELSE 'Good'

END AS weather\_conditions

FROM

weather\_records;

**Task 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.*

**Code:**

SELECT

trips.start\_ts,

CASE

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

WHEN weather\_records.description Like '%storm%' THEN 'Bad'

ELSE 'Good'

END AS weather\_conditions,

trips.duration\_seconds

FROM

trips

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

WHERE

pickup\_location\_id = 50

AND dropoff\_location\_id = 63

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

ORDER BY

trips.trip\_id;