

Step 1.

Find the number of taxi rides for each taxi company for November 15-16, 2017. Sort the results by the *trips\_amount* field in descending order.

Code

```
SELECT
    c.company_name,
    COUNT(t.trip_id) AS trips_amount
FROM trips t
JOIN cabs c ON t.cab_id = c.cab_id
WHERE t.start_ts BETWEEN '2017-11-15 00:00:00' AND '2017-11-16 23:59:59'
GROUP BY c.company_name
ORDER BY trips_amount DESC;
```

Result Sample

company_name	trips_amount
Flash Cab	19558
Taxi Affiliation Services	11422
Medallion Leasin	10367
Yellow Cab	9888
Taxi Affiliation Service Yellow	9299
Chicago Carriage Cab Corp	9181
City Service	8448
Sun Taxi	7701
Star North Management LLC	7455

Step 2.

Find the number of rides for every taxi companies whose name contains the words "Yellow" or "Blue" for November 1-7, 2017. Group the results by the company name.

### Code

```
SELECT
    c.company_name,
    COUNT(t.trip_id) AS trips_amount
FROM trips t
JOIN cabs c ON t.cab_id = c.cab_id
WHERE t.start_ts BETWEEN '2017-11-01 00:00:00' AND '2017-11-07 23:59:59'
    AND (c.company_name LIKE '%Yellow%' OR c.company_name LIKE '%Blue%')
GROUP BY c.company_name;
```

### Results

company_name	trips_amount
Blue Diamond	6764
Blue Ribbon Taxi Association Inc.	17675
Taxi Affiliation Service Yellow	29213
Yellow Cab	33668

### Step 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. Sort the result in descending order by *trips\_amount*.

### Code

```
SELECT
CASE
    WHEN c.company_name IN ('Flash Cab','Taxi Affiliation Services')
    THEN c.company_name
    ELSE 'Other'
    END AS company,
COUNT (t.trip_id) AS trips_amount
FROM trips t
JOIN cabs c ON t.cab_id = c.cab_id
WHERE t.start_ts BETWEEN '2017-11-01 00:00:00' AND '2017-11-07 23:59:59'
GROUP BY company
ORDER BY trips_amount DESC;
```

### Results

company	trips_amount
Other	335771
Flash Cab	64084
Taxi Affiliation Services	37583

#### Step 4.

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

#### Code

```
SELECT
    neighborhood_id,
    name
FROM
    neighborhoods
where
    name LIKE '%Hare' OR name LIKE 'Loop';
```

#### Results

neighborhood_id	name
50	Loop
63	O'Hare

#### Step 5.

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

**Code**

```
SELECT
    ts,
    CASE
        WHEN w.description LIKE '%rain%' OR w.description LIKE '%storm%' THEN 'Bad'
        ELSE 'Good'
    END AS weather_conditions
FROM weather_records w;
```

**Results Sample**

ts	weather_conditions
2017-11-01 00:00:00	Good
2017-11-01 01:00:00	Good
2017-11-01 02:00:00	Good
2017-11-01 03:00:00	Good
2017-11-01 04:00:00	Good
2017-11-01 05:00:00	Good
2017-11-01 06:00:00	Good
2017-11-01 07:00:00	Good
2017-11-01 08:00:00	Good

**Step 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
    t.start_ts,
    CASE
        WHEN w.description LIKE '%rain%' OR w.description LIKE '%storm%' THEN 'Bad'
        ELSE 'Good'
    END AS weather_conditions,
    t.duration_seconds
FROM trips t
    JOIN weather_records w ON t.start_ts = w.ts
WHERE
    t.pickup_location_id = 50
    AND t.dropoff_location_id = 63
    AND EXTRACT(DOW FROM t.start_ts) = 6
ORDER BY t.trip_id;
```

**Results Sample**

start_ts	weather_conditions	duration_seconds
2017-11-25 12:00:00	Good	1380
2017-11-25 16:00:00	Good	2410
2017-11-25 14:00:00	Good	1920
2017-11-25 12:00:00	Good	1543
2017-11-04 10:00:00	Good	2512
2017-11-11 07:00:00	Good	1440
2017-11-11 04:00:00	Good	1320
2017-11-04 16:00:00	Bad	2969
2017-11-18 11:00:00	Good	2280

