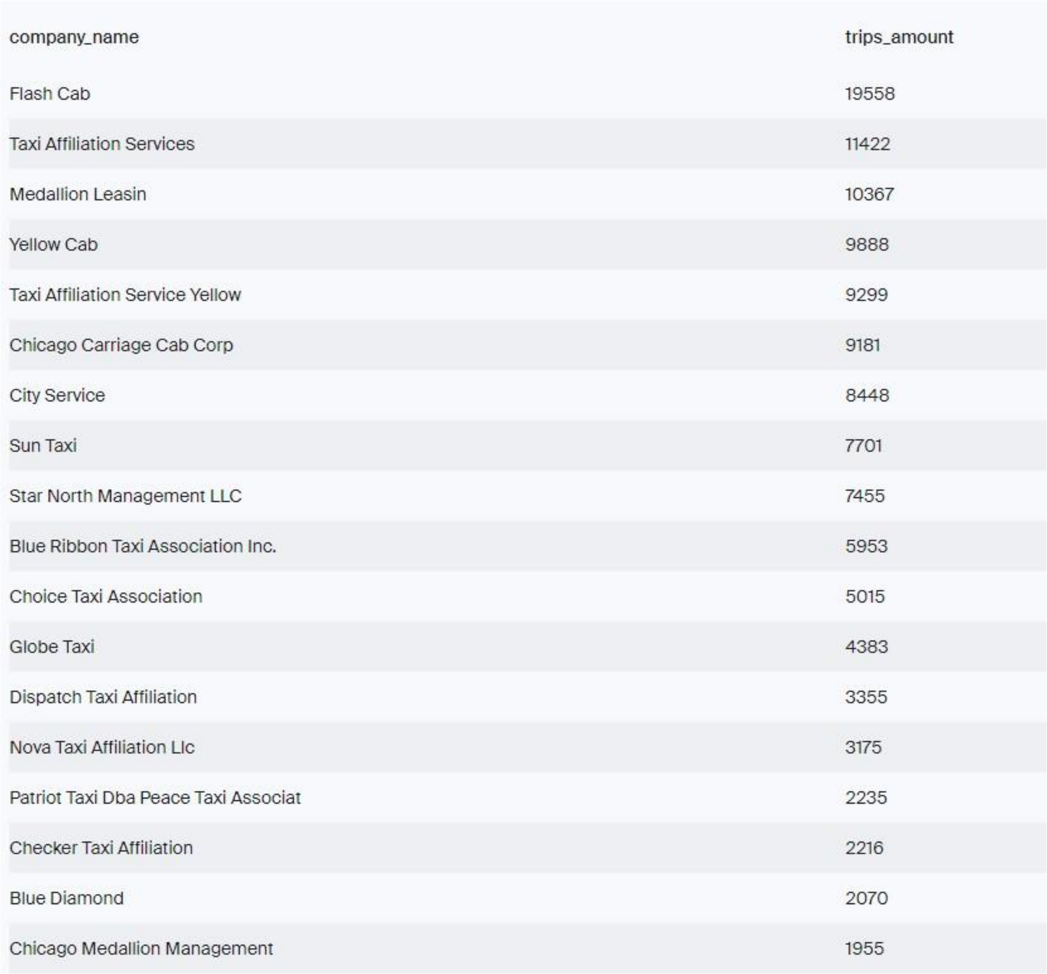
Goal 1: Find the number of taxi rides for each taxi company for November 15-16, 2017. Sort the results by amount of trips in descending order.

Code:

```
SELECT
    cabs.company_name AS company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    cabs
    INNER JOIN trips ON trips.cab_id = cabs.cab_id
WHERE
    trips.start_ts::date BETWEEN '2017-11-15' AND '2017-11-16'
GROUP BY      company_name ORDER BY
    trips_amount DESC;
```

Results:



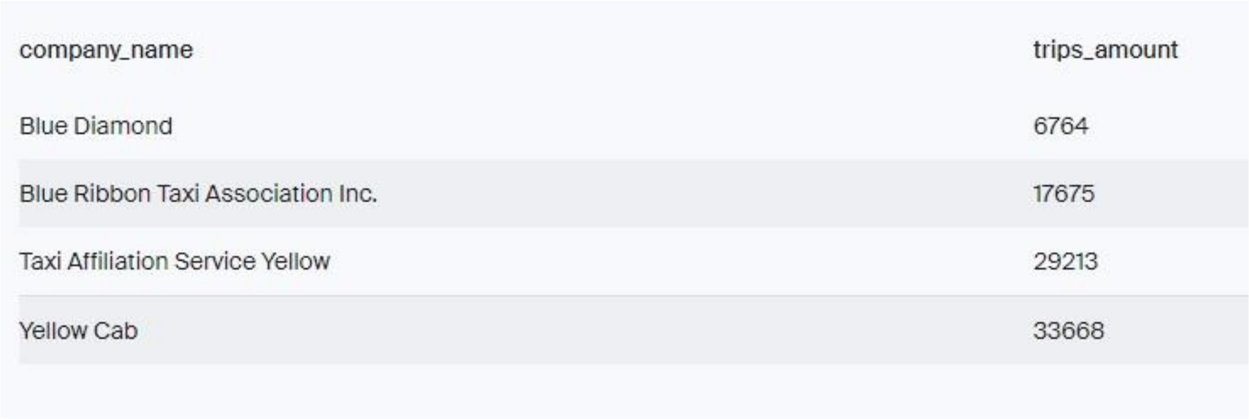
company_name	trips_amount
Flash Cab	19558
Taxi Affiliation Services	11422
Medallion Leasin	10367
Yellow Cab	9888
Taxi Affiliation Service Yellow	9299
Chicago Carriage Cab Corp	9181
City Service	8448
Sun Taxi	7701
Star North Management LLC	7455
Blue Ribbon Taxi Association Inc.	5953
Choice Taxi Association	5015
Globe Taxi	4383
Dispatch Taxi Affiliation	3355
Nova Taxi Affiliation Llc	3175
Patriot Taxi Dba Peace Taxi Associat	2235
Checker Taxi Affiliation	2216
Blue Diamond	2070
Chicago Medallion Management	1955

Goal 2: Find the number of rides for every taxi company whose name contains the words "Yellow" or "Blue" for November 1-7, 2017. Group the results by company name.

Code:

```
SELECT
    cabs.company_name AS company_name,
    COUNT(trips.trip_id) AS
trips_amount FROM      cabs
    INNER JOIN trips ON trips.cab_id = cabs.cab_id
WHERE
    (cabs.company_name LIKE '%Yellow%'
    OR cabs.company_name LIKE '%Blue%')
    AND trips.start_ts::date BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY      company_name;
```

Results:



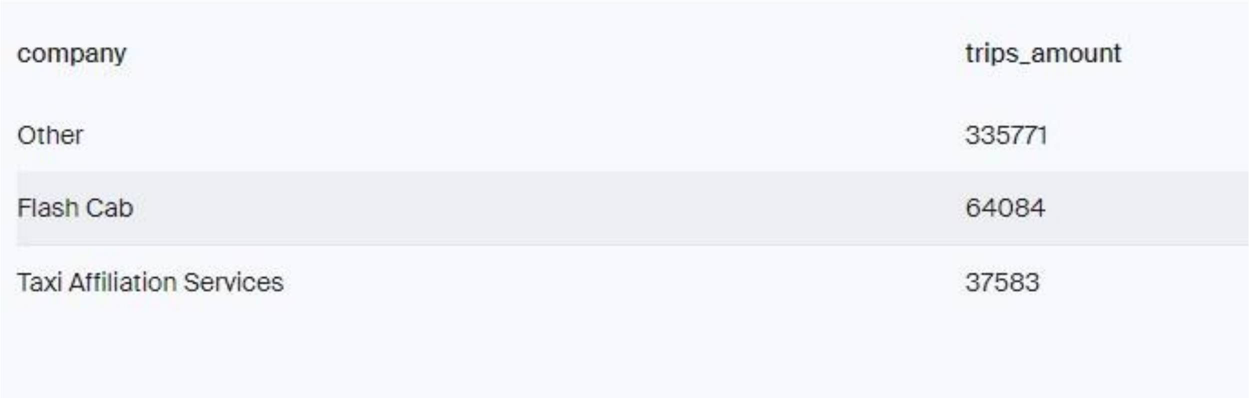
company_name	trips_amount
Blue Diamond	6764
Blue Ribbon Taxi Association Inc.	17675
Taxi Affiliation Service Yellow	29213
Yellow Cab	33668

Goal 3: Find the number of rides for the two most popular companies. 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 the amount of trips.

Code:

```
SELECT
    CASE
        WHEN cabs.company_name = 'Flash Cab' THEN 'Flash Cab'
        WHEN cabs.company_name = 'Taxi Affiliation Services' THEN
'Taxi Affiliation Services'
        ELSE 'Other'
    END AS company,
    COUNT(trips.trip_id) AS
trips_amount FROM      cabs
    INNER JOIN trips ON trips.cab_id = cabs.cab_id WHERE
    trips.start_ts::date BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY      company ORDER BY
    trips_amount DESC;
```

Results:



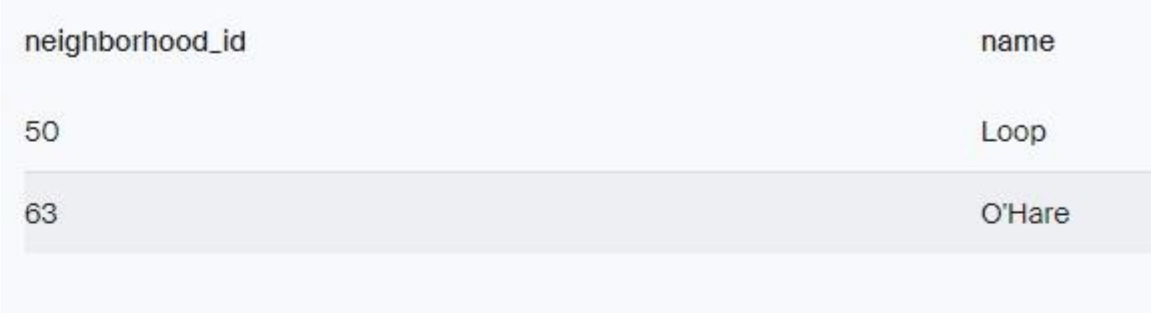
company	trips_amount
Other	335771
Flash Cab	64084
Taxi Affiliation Services	37583

Goal 4: Retrieve the identifiers of the O'Hare and Loop neighborhoods.

Code:

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

Results:



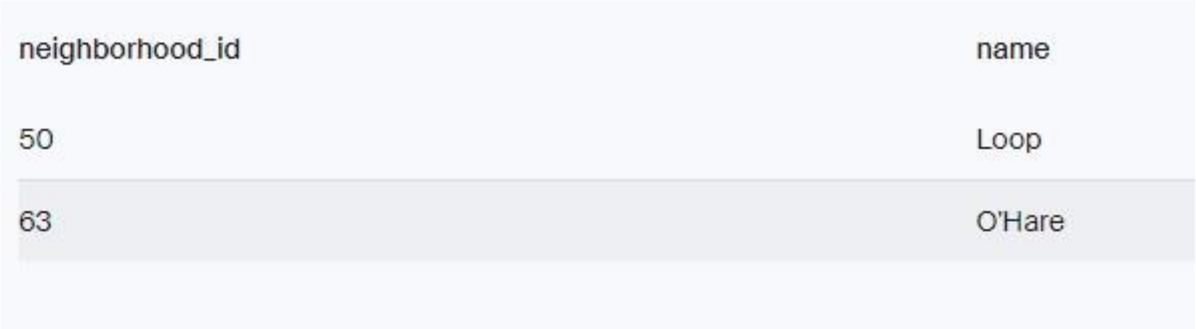
neighborhood_id	name
50	Loop
63	O'Hare

Goal 5: For each hour, retrieve the weather condition records. Break all hours into two groups: Bad if the description field contains the words rain or storm, and Good for others.

Code:

```
SELECT
    DATE_TRUNC('hour', ts::timestamp) AS date_and_hour,
    CASE
        WHEN description LIKE '%rain%' THEN 'Bad'
        WHEN description LIKE '%storm%' THEN 'Bad'
        ELSE 'Good'
    END AS weather_conditions
FROM      weather_records
GROUP BY      date_and_hour,
weather_conditions ORDER BY
date_and_hour;
```

Results:



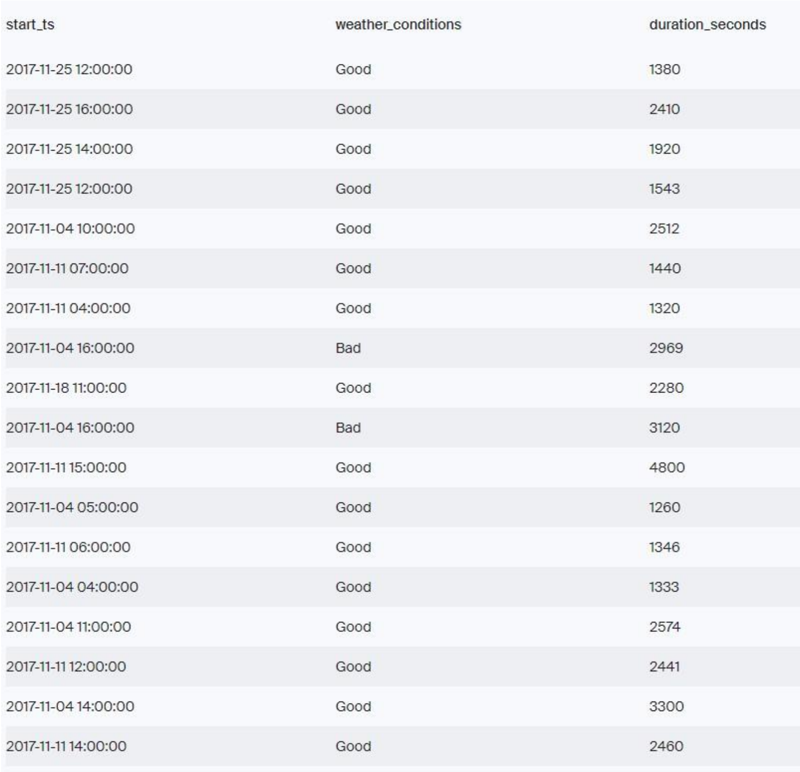
neighborhood_id	name
50	Loop
63	O'Hare

Goal 6: Retrieve all the rides that started in the Loop on a Saturday and ended at O'Hare. Get the weather conditions for each ride. Also, retrieve the duration of each ride. Ignore rides for which data on weather conditions is not available. Sort by trip id.

Code:

```
SELECT
    trips.start_ts AS 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 AS duration_seconds
FROM      trips
    INNER JOIN weather_records ON weather_records.ts = trips.start_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;
```

Results:



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
2017-11-04 16:00:00	Bad	3120
2017-11-11 15:00:00	Good	4800
2017-11-04 05:00:00	Good	1260
2017-11-11 06:00:00	Good	1346
2017-11-04 04:00:00	Good	1333
2017-11-04 11:00:00	Good	2574
2017-11-11 12:00:00	Good	2441
2017-11-04 14:00:00	Good	3300
2017-11-11 14:00:00	Good	2460