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.

```
SELECT
    cabs.company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    cabs
JOIN
    trips 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.

```
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 '2017-11-01' AND '2017-11-07'
    AND cabs.company_name LIKE '%Yellow%'
GROUP BY
    cabs.company_name

UNION ALL

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 '2017-11-01' AND '2017-11-07'
   AND cabs.company_name LIKE '%Blue%'
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*.

```
SELECT
   CASE
      WHEN cabs.company_name IN ('Flash Cab', 'Taxi Affiliation Services') THEN
cabs.company_name
      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
   CAST(trips.start_ts AS DATE) BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY
   CASE
      WHEN cabs.company_name IN ('Flash Cab', 'Taxi Affiliation Services') THEN
cabs.company_name
      ELSE 'Other'
   END
ORDER BY
   trips_amount DESC;
```

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

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

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.

```
SELECT
   ts,
   CASE
      WHEN description LIKE '%rain%' OR 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.

```
SELECT
   trips.start_ts,
   CASE
      WHEN weather_records.description ILIKE '%rain%' OR weather_records.description ILIKE
'%storm%' THEN 'Bad'
      ELSE 'Good'
   END AS weather_conditions,
   trips.duration_seconds
FROM
```

```
    trips
INNER JOIN
    weather_records
ON
    DATE(trips.start_ts) = DATE(weather_records.ts) AND EXTRACT(HOUR FROM
trips.start_ts) = EXTRACT(HOUR FROM weather_records.ts)
WHERE
    trips.pickup_location_id = 50
    AND trips.dropoff_location_id = 63
    AND EXTRACT(DOW FROM trips.start_ts) = 6 -- Saturday
ORDER BY
    trips.trip_id;
```