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 AS company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    cabs
    LEFT JOIN trips ON trips.cab_id = cabs.cab_id
WHERE
    trips.start_ts::date >= '2017-11-15'
    AND trips.start_ts::date <= '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 AS company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    cabs
    LEFT JOIN trips ON trips.cab_id = cabs.cab_id
WHERE
    cabs.company_name LIKE '%Blue%'
    AND trips.start_ts::date >= '2017-11-1'
    AND trips.start_ts::date <= '2017-11-7'
GROUP BY
    cabs.company_name
UNION
SELECT
    cabs.company_name AS company_name,
    COUNT(trips.trip_id) AS trips_amount
FROM
    cabs
    LEFT JOIN trips ON trips.cab_id = cabs.cab_id
WHERE
    cabs.company_name LIKE '%Yellow%'
    AND trips.start_ts::date >= '2017-11-1'
    AND trips.start_ts::date <= '2017-11-7'
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
    DISTINCT 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 >= '2017-11-1'
    AND trips.start_ts::date <= '2017-11-7'
GROUP BY
    company
ORDER BY
    trips_amount DESC;
```

####

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

```
SELECT
    *
FROM
    neighborhoods
WHERE
    name LIKE '%Hare%'
    OR name LIKE '%Loop%'
    AND name <> 'West 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.

```
SELECT
    DISTINCT DATE_TRUNC('hour', ts) AS ts,
/* For each hour, retrieve the weather condition records from the weather_records
table */
    CASE
        WHEN description LIKE '%rain%' THEN
        'Bad'
        WHEN description LIKE '%storm%' THEN
        'Bad'
        ELSE 'Good'
        END AS weather_conditions
    /* 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.  */
FROM
    weather_records
GROUP BY
    description,
    ts
ORDER BY
    ts ASC;
```

####

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.

```
SELECT
    trips.start_ts,
    weather_records.weather_conditions,
    trips.duration_seconds
FROM
    trips
    INNER JOIN (SELECT
            DISTINCT DATE_TRUNC('hour', ts) AS ts,
                    CASE
                        WHEN description LIKE '%rain%' THEN
                        'Bad'
                        WHEN description LIKE '%storm%' THEN
                        'Bad'
                        ELSE 'Good'
                        END AS weather_conditions
                    FROM
                        weather_records)
                        AS weather_records ON weather_records.ts = trips.start_ts
WHERE
```

```
    pickup_location_id = 50
    AND EXTRACT(isodow FROM start_ts) = 6
    AND dropoff_location_id = 63
    /*...on a Saturday...Extract! */
ORDER BY
    trips.trip_id;
```