

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

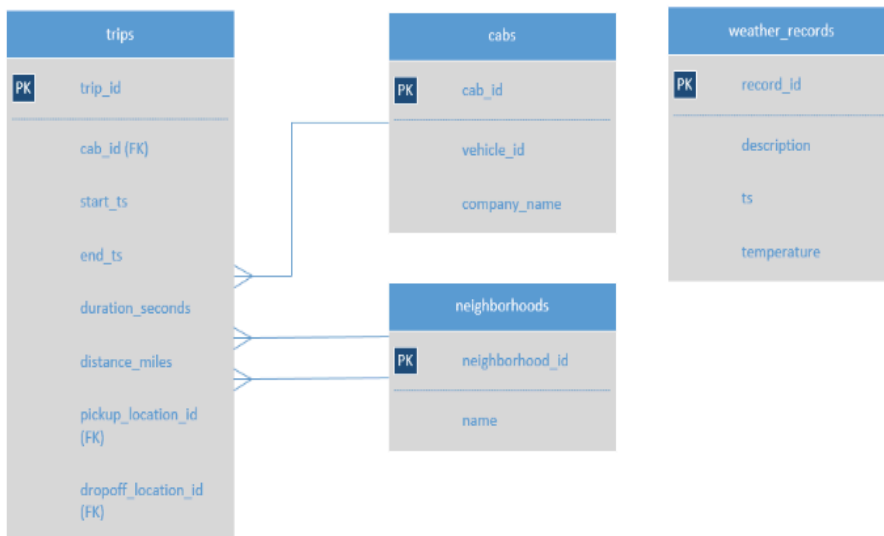
trips table: data on rides

- **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"

Table scheme



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.

```

1  select
2  cabs.company_name as company,
3  count (trips.trip_id) as trips_amount
4  from
5  cabs
6  inner join
7  trips on trips.cab_id = cabs.cab_id
8  where
9  cast(trips.start_ts as date) between '2017-11-15' and
10 '2017-11-16'
11 group by
12 company_name
13 order by
14 trips_amount desc;
15
16

```

Result

company	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

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.

```
1  SELECT
2      cabs.company_name,
3      COUNT(trips.trip_id) AS trips_amount
4  FROM
5      trips
6  JOIN cabs ON trips.cab_id = cabs.cab_id
7  WHERE
8      CAST(start_ts AS DATE) BETWEEN '2017-11-01' AND '2017-11-07'
9      AND (cabs.company_name LIKE '%Yellow%' OR cabs.company_name LIKE '%Blue%')
10 GROUP BY
11     cabs.company_name;
```

Result

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

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*.

```
1  SELECT
2      CASE
3          WHEN cabs.company_name = 'Flash Cab' THEN 'Flash Cab'
4          WHEN cabs.company_name = 'Taxi Affiliation Services' THEN 'Taxi Affiliation Services'
5          ELSE 'Other'
6      END AS company,
7      COUNT(trips.trip_id) AS trips_amount
8  FROM
9      trips
10 JOIN
11     cabs ON trips.cab_id = cabs.cab_id
12 WHERE
13     CAST(trips.start_ts AS DATE) BETWEEN '2017-11-01' AND '2017-11-07'
14 GROUP BY
15     company
16 ORDER BY
17     trips_amount DESC;
```

Result

company	trips_amount
Other	335771
Flash Cab	64084
Taxi Affiliation Services	37583

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

```
1  SELECT
2  neighborhood_id,
3  name
4  FROM
5  neighborhoods
6  WHERE
7  name LIKE '%Hare' OR name LIKE 'Loop'
8
```

Result

neighborhood_id	name
50	Loop
63	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*.

```

1  SELECT
2  ts,
3  CASE
4  WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'
5  ELSE 'Good'
6  END AS weather_conditions
7  FROM
8  weather_records;

```

Result	
2017-11-01 21:00:00	Good
2017-11-01 22:00:00	Good
2017-11-01 23:00:00	Good
2017-11-02 00:00:00	Good
2017-11-02 01:00:00	Good
2017-11-02 02:00:00	Good
2017-11-02 03:00:00	Bad
2017-11-02 04:00:00	Bad
2017-11-02 05:00:00	Bad
2017-11-02 06:00:00	Bad
2017-11-02 07:00:00	Bad
2017-11-02 08:00:00	Good
2017-11-02 09:00:00	Good
2017-11-02 10:00:00	Good
2017-11-02 11:00:00	Good

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*.

```

1  select
2  trips.start_ts,
3  trips.duration_seconds,
4  case
5  WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'
6  ELSE 'Good'
7  END AS weather_conditions
8  from
9  trips
10 join weather_records on weather_records.ts = trips.start_ts
11 where
12 trips.pickup_location_id = '50' and
13 trips.dropoff_location_id = '63' and
14 extract (dow from trips.start_ts) = '6'
15

```

Result

start_ts	duration_seconds	weather_conditions
2017-11-25 12:00:00	1380	Good
2017-11-25 16:00:00	2410	Good
2017-11-25 14:00:00	1920	Good
2017-11-25 12:00:00	1543	Good
2017-11-04 10:00:00	2512	Good
2017-11-11 07:00:00	1440	Good
2017-11-11 04:00:00	1320	Good
2017-11-04 16:00:00	2969	Bad
2017-11-18 11:00:00	2280	Good
2017-11-04 16:00:00	3120	Bad
2017-11-11 15:00:00	4800	Good

