## The Zuber Database

Zuber is a new ride-sharing company that's launching in Chicago. Your task is to find patterns in the available information. You want to understand passenger preferences and the impact of external factors on rides.

You'll study a database, analyze data from competitors, and investigate the impact of weather on ride frequency.

## Description of the data

**Tables:**

*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

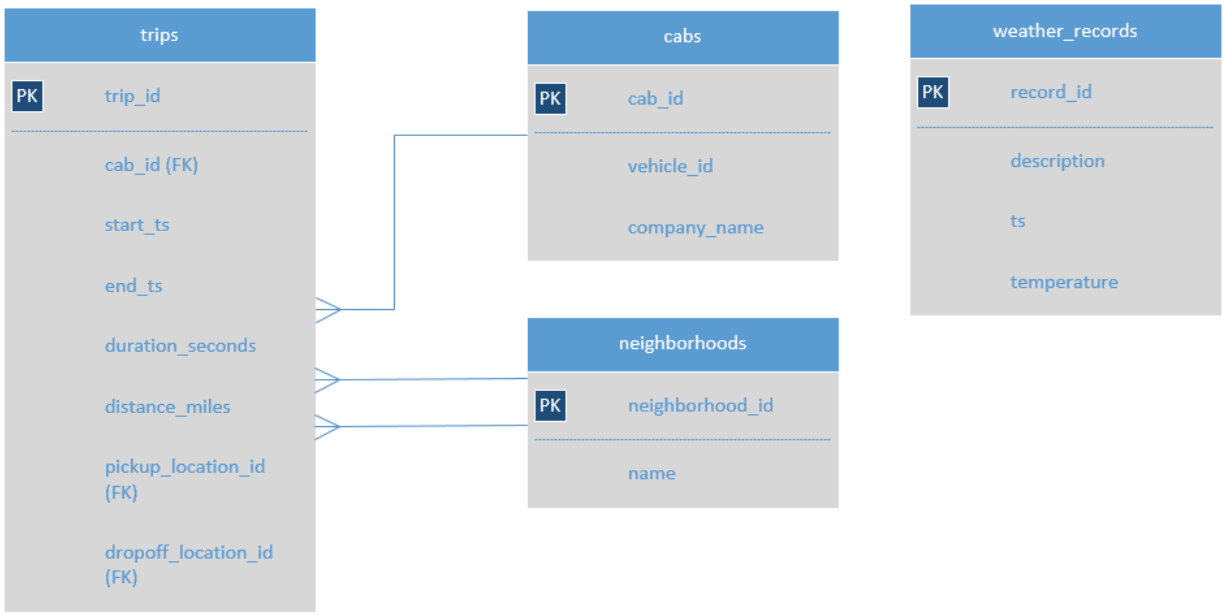
*trips table: data on rides*

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



## Tasks, Solutions and Results

**Task 1 / 6**

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

**Result**

| 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 |

**Task 2 / 6**

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.

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

**Result**

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

**Task 3 / 6**

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 company_name != 'Flash Cab'
    AND company_name != 'Taxi Affiliation Services'
    THEN 'Other'

    WHEN company_name = 'Flash Cab'
    THEN 'Flash Cab'

    WHEN company_name = 'Taxi Affiliation Services'
    THEN 'Taxi Affiliation Services'

    END AS company,
    COUNT(trip_id) AS trips_amount
FROM
    trips JOIN cabs ON trips.cab_id = cabs.cab_id
WHERE
    start_ts::date BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY
    company
ORDER BY
    trips_amount DESC;
```

**Result**

| company | trips_amount |
|---|---|
| Other | 335771 |
| Flash Cab | 64084 |
| Taxi Affiliation Services | 37583 |

**Task 4 / 6**

- [x] 4. Retrieve the identifiers of the O'Hare and Loop neighborhoods from the *neighborhoods* table.

```sql
SELECT
    neighborhood_id,
    name
FROM
    neighborhoods
WHERE
    name LIKE 'O_____' OR name LIKE 'Loop';
```

**Result**

| neighborhood_id | name |
|---|---|
| 50 | Loop |
| 63 | O'Hare |

**Task 5 / 6**

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*.

```sql
SELECT
    ts,
    CASE WHEN description LIKE '%rain%' OR description LIKE '%storm%'
    THEN 'Bad'

    ELSE
        'Good'

    END AS weather_conditions
FROM
    weather_records
GROUP BY
    weather_conditions,
    ts;
```

**Result**

| ts | weather_conditions |
|---|---|
| 2017-11-29 06:00:00 | Good |
| 2017-11-25 04:00:00 | Good |
| 2017-11-20 17:00:00 | Good |
| 2017-11-18 21:00:00 | Bad |
| 2017-11-15 10:00:00 | Bad |
| 2017-11-30 00:00:00 | Good |
| 2017-11-12 13:00:00 | Bad |
| 2017-11-03 10:00:00 | Good |
| 2017-11-12 07:00:00 | Bad |
| 2017-11-29 12:00:00 | Good |
| 2017-11-18 06:00:00 | Good |
| 2017-11-28 21:00:00 | Good |
| 2017-11-27 07:00:00 | Good |
| 2017-11-20 06:00:00 | Good |
| 2017-11-12 21:00:00 | Good |

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
    ts,
    CASE WHEN description LIKE '%rain%' OR description LIKE '%storm%'
    THEN 'Bad'

    ELSE
        'Good'

    END AS weather_conditions
FROM
    weather_records
GROUP BY
    weather_conditions,
    ts;
```

Result

| ts | weather_conditions |
|---|---|
| 2017-11-29 06:00:00 | Good |
| 2017-11-25 04:00:00 | Good |
| 2017-11-20 17:00:00 | Good |
| 2017-11-18 21:00:00 | Bad |
| 2017-11-15 10:00:00 | Bad |
| 2017-11-30 00:00:00 | Good |
| 2017-11-12 13:00:00 | Bad |
| 2017-11-03 10:00:00 | Good |
| 2017-11-12 07:00:00 | Bad |
| 2017-11-29 12:00:00 | Good |
| 2017-11-18 06:00:00 | Good |
| 2017-11-28 21:00:00 | Good |
| 2017-11-27 07:00:00 | Good |
| 2017-11-20 06:00:00 | Good |
| 2017-11-12 21:00:00 | Good |