Zuber Database

1. Calculate the number of taxi rides for each taxi company on November 15-16, 2017. Name the result `trips_amount` and display it with `company_name`, sorted by `trips_amount` in descending order.

Code:

```
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 '2017-11-15' 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

2. Calculate the number of rides for taxi companies with names containing "Yellow" or "Blue" from November 1-7, 2017. Name the result `trips_amount` and group the results by `company_name` field.

Code:

```
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 '2017-11-1' AND '2017-11-7'  and (cabs.company_name like'%Blue%' or cabs.company_name like'%Yellow%')
GROUP BY company_name
```

Result

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

3. For November 2017, find the number of rides for "Flash Cab" and "Taxi Affiliation Services". Find the number of rides for these two companies and name the resulting variable `trips_amount` and group the rest of the companies as "Other", Group the data by taxi company names. sort the results by `trips_amount` in descending order.

Code:

```
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 '2017-11-1' AND '2017-11-7'
GROUP BY company
ORDER BY trips_amount DESC;
```

Result

company	trips_amount
Other	335771
Flash Cab	64084
Taxi Affiliation Services	37583

4. Retrieve the neighborhood identifiers for O'Hare and the Loop from the `neighborhoods` table.

Code:

```
select * from  neighborhoods
where name like '%Hare%' or name like 'Loop%'
```

Result

neighborhood_id	name
50	Loop
63	O'Hare

5. For each hour, get the weather condition from the `weather_records` table. break statement into two categorize conditions as "Bad" if description contains (rain or storm), and "Good" for others. Include the `date`, `hour` (ts), and the weather condition field `weather_conditions`.

Code:

```
select
ts as date_and_hour ,
case
   when description like '%rain%' or description like '%storm%' then 'Bad'Else 'Good' END AS weather_conditions
from weather_records
```

Result

date_and_hour	weather_conditions
2017-11-01 00:00:00	Good
2017-11-01 01:00:00	Good
2017-11-01 02:00:00	Good
2017-11-01 03:00:00	Good
2017-11-01 04:00:00	Good
2017-11-01 05:00:00	Good
2017-11-01 06:00:00	Good
2017-11-01 07:00:00	Good
2017-11-01 08:00:00	Good
2017-11-01 09:00:00	Good
2017-11-01 10:00:00	Good
2017-11-01 11:00:00	Good
2017-11-01 12:00:00	Good
2017-11-01 13:00:00	Good

6. Retrieve all Saturday rides from the Loop (`neighborhood_id: ` 50) to O'Hare (`neighborhood_id: ` 63) and their weather conditions. Calculate the ride durations and exclude any rides without weather data.

Code:

```
select
start_ts,
case
   when description like '%rain%' or description like '%storm%' then 'Bad'Else 'Good' END AS weather_conditions,
duration_seconds
from weather_records
inner join trips on trips.start_ts=weather_records.ts
where
trips.pickup_location_id = '50' AND
trips.dropoff_location_id = '63' and
extract(dow from trips.start_ts)=6
order by  trip_id
```

Result

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