SQL Notes

Task 1: Trip Count per Company on Specific Dates

Objective:
Get the number of trips made by each cab company between November 15 and 16, 2017.

Key Concepts:

- `INNER JOIN` used to combine `cabs` and `trips` on `cab_id`.
- `CAST(trips.start_ts AS date)` extracts the date from timestamp for filtering.
- `GROUP BY` groups trips by company.
- `ORDER BY trips_amount DESC` sorts by trip count, descending.

Task 2: Compare Yellow and Blue Companies

Objective:
Get the trip counts from Nov 1–7, 2017, specifically for companies with "Yellow" or "Blue" in their names.

Key Concepts:

- `LIKE '%%Yellow%%'` and `LIKE '%%Blue%%'` filters company names.
- `UNION ALL` combines two filtered queries without removing duplicates.
- Same `JOIN`, `WHERE`, and `GROUP BY` structure as Task 1.

Task 3: Categorizing Company Types

Objective:
Group trip counts from Nov 1–7, 2017 into three categories:

- Flash Cab
- Taxi Affiliation Services
- Other companies

Key Concepts:

- `CASE` used for custom categorization.
- Trips grouped using alias `company`.

Task 4: Neighborhood Name Filter

Objective:
Select neighborhood details where the name:

- Ends with "Hare" (`LIKE '%Hare'`)
- Or is exactly "Loop"

Key Concepts:

- `LIKE` operator for partial string matching.
- Simple filtering from `neighborhoods` table.

Task 5: Weather Categorization

Objective:
Classify weather conditions as:

- Bad if `description` contains "rain" or "storm"

- Good otherwise

Key Concepts:

- `CASE` for custom classification.
- String pattern matching with `LIKE`.

Task 6: Weather Conditions for Specific Route on Saturdays

Objective: Get weather conditions and trip details for trips:

- Starting at location 50, ending at 63
- On Saturdays (`EXTRACT(DOW) = 6`)

Key Concepts:

- `EXTRACT(DOW)` extracts day of the week (6 = Saturday).
- Nested `SELECT` used to apply weather condition logic.
- Join between trips and weather records based on timestamp (`ts`).

SQL Queries

Task 1:

```
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;
```

Task 2:

```
SELECT
        cabs.company_name as 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 company_name
    UNION ALL
    SELECT
        cabs.company_name as 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 company_name;
```

Task 3:

```
SELECT
  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
JOIN
  trips
ON
  cabs.cab_id = trips.cab_id
WHERE
  CAST(trips.start_ts AS DATE) BETWEEN '2017-11-01' AND '2017-11-07'
GROUP BY
  company
ORDER BY
  trips_amount DESC;
```

Task 4:

```
Select
  neighborhood_id,
  name
From
  neighborhoods
Where
  name Like '%Hare' OR name like 'Loop';
```

Task 5:

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

```
        ELSE 'Good'
            END AS weather_conditions
FROM
    weather_records;
```

Task 6:

```
SELECT
    start_ts,
    T.weather_conditions,
    duration_seconds
FROM
    trips
INNER JOIN (
    SELECT
        ts,
        CASE
            WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad'
            ELSE 'Good'
        END AS weather_conditions
    FROM
        weather_records
) T ON T.ts = trips.start_ts
WHERE
    pickup_location_id = 50 AND dropoff_location_id = 63 AND EXTRACT (DOW from
trips.start_ts) = 6
ORDER BY trip_id
```
