Jeisiany da Cruz Ferrugini
Processo Seletivo – Data Science

Projeto em SQL

Para responder às questões foi utilizado o **Standard SQL no BigQuery.**

Mesmo não sendo necessário executar as consultas no Google BigQuery, decidi utilizar para melhor análise e compreensão.

1)

Para responder essa pergunta, resolvi dividi-la em 3 partes para melhor compreensão:

Parte 1: Dias da semana (1 a 7, sendo 1 = domingo) ordenados pelo faturamento médio de corrida (média do *total_amount*):

- Entrada:

```
SELECT EXTRACT(DAYOFWEEK from pickup_datetime) AS day_of_week,

       ROUND(AVG(total_amount),2) as avg_day_amount

FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`

GROUP BY EXTRACT(DAYOFWEEK from pickup_datetime)

ORDER BY avg_day_amount DESC;
```

- Saída:

Linha	day_of_week	avg_day_amount
1	1	40.92
2	2	40.8
3	6	40.05
4	4	39.83
5	3	39.74
6	7	38.69
7	5	37.4

Parte 2: Visualizar o dia (data) com maior faturamento médio e o dia da semana referente a essa data:

- Entrada:

```
SELECT EXTRACT(DATE from pickup_datetime) AS date,

       EXTRACT(DAYOFWEEK from pickup_datetime) AS day_of_week,

       AVG(total_amount) as avg_day_amount

FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`

GROUP BY EXTRACT(DATE from pickup_datetime), EXTRACT(DAYOFWEEK from pickup_datetime)

ORDER BY avg_day_amount DESC

LIMIT 1000;
```

- Saída:

Linha	date	day_of_week	avg_day_amount
1	2018-01-04	5	59.253333333
2	2018-12-19	4	51.903636364
3	2018-08-09	5	47.1375
4	2018-09-02	1	46.9624
5	2018-03-26	2	46.538823529
6	2018-12-09	1	46.258823529
7	2018-03-16	6	45.6945
8	2018-05-23	4	44.560625
9	2018-09-03	2	44.553333333
10	2018-10-27	7	44.47
11	2018-08-21	3	44.004117647
12	2018-03-15	5	43.960666667
13	2018-02-25	1	43.501666667
14	2018-11-06	3	43.152790698
15	2018-06-13	4	42.94875

Parte 3 (Final): Exibir o faturamento dos dias (datas) referentes ao dia da semana com maior faturamento médio:

Foi possível analisar na **Parte 2** que o dia da semana que tem o maior faturamento médio é a quinta-feira (5), assim para exibir o faturamento de todas as quintas-feira será necessário realizar ***subquerys***, para isso foi utilizado o WITH para emular uma tabela temporária.

- Entrada:

```
WITH avg_amount AS (

      SELECT EXTRACT(DATE FROM pickup_datetime) AS date,

             EXTRACT(DAYOFWEEK FROM pickup_datetime) AS day_of_week,

             ROUND(AVG(total_amount),2) AS avg_total_amount,

             SUM(total_amount) AS sum_total_amount

      FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`

      GROUP BY date, day_of_week

      ORDER BY avg_total_amount DESC

      LIMIT 1000

), max_avg_amount AS (

      SELECT day_of_week

      FROM avg_amount

      WHERE avg_total_amount = (SELECT MAX(avg_total_amount) FROM avg_amount)

)

SELECT date, day_of_week, sum_total_amount, avg_total_amount

FROM avg_amount

WHERE day_of_week = (SELECT day_of_week FROM max_avg_amount)

ORDER BY sum_total_amount DESC
```

- Saída:

Linha	date	day_of_week	sum_total_amount	avg_total_amount
1	2018-11-29	5	1933.38	34.52
2	2018-09-13	5	873.04	39.68
3	2018-08-09	5	754.2	47.14
4	2018-03-15	5	659.41	43.96
5	2018-04-05	5	504.11	31.51
6	2018-01-25	5	397.75	39.78
7	2018-07-05	5	225.66	32.24
8	2018-01-04	5	177.76	59.25
9	2018-05-03	5	172.81	34.56
10	2018-12-20	5	130	21.67
11	2018-05-24	5	118.42	39.47

2)

```sql
SELECT passenger_count,
       COUNT(passenger_count) AS trips_count,
       ROUND(AVG(total_amount),2) AS avg_amount_trip,
       ROUND(AVG(total_amount / passenger_count),2) AS avg_per_passenger
FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`
WHERE passenger_count BETWEEN 1 AND 5
GROUP BY passenger_count
ORDER BY passenger_count
```

Linha	passenger_count	trips_count	avg_amount_trip	avg_per_passenger
1	1	716	39.83	39.83
2	2	151	39.45	19.73
3	3	32	34.65	11.55
4	4	25	36.74	9.19
5	5	50	37.98	7.6

3)

```
SELECT date, trips_day,
       LAG(date, 1)
         OVER (ORDER BY date) AS previous_day,
       ROUND((((trips_day - LAG(trips_day, 1) OVER (ORDER BY date)) /
(LAG(trips_day, 1) OVER (ORDER BY date))) * 100),2) AS perc_diff
FROM (
  SELECT EXTRACT(DATE FROM pickup_datetime) AS date,
       COUNT(EXTRACT(DATE FROM pickup_datetime)) AS trips_day,
  FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`
  WHERE passenger_count BETWEEN 1 AND 5
  GROUP BY date
  ORDER BY date
)
ORDER BY date
```

Linha	date	trips_day	previous_day	perc_diff
1	2018-01-04	3	*null*	*null*
2	2018-01-05	12	2018-01-04	300.0
3	2018-01-15	15	2018-01-05	25.0
4	2018-01-16	3	2018-01-15	-80.0
5	2018-01-25	10	2018-01-16	233.33
6	2018-01-26	8	2018-01-25	-20.0
7	2018-02-04	17	2018-01-26	112.5
8	2018-02-05	6	2018-02-04	-64.71
9	2018-02-14	22	2018-02-05	266.67
10	2018-02-24	15	2018-02-14	-31.82
11	2018-02-25	16	2018-02-24	6.67

4)

Para essa questão **não** foi levado em consideração apenas as **corridas válidas (1 a 5 passageiros)**.

a) Cálculo da média de gorjeta do grupo das **corridas que houveram pedágios e que possui até 3 passageiros**:

```
SELECT AVG(tip_amount) AS avg_tip_amount_g1
FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`
WHERE passenger_count <= 3 AND tolls_amount > 0
```

Linha	avg_tip_amount_g1
1	6.235597015

b) Cálculo da média de gorjeta do grupo que **não houveram pedágios e que possuem 4 ou mais passageiros**:

```
SELECT AVG(tip_amount) AS avg_tip_amount_g2
FROM `projeto-sql-monks.taxi_ride.tlc_yellow_trips_2018`
WHERE passenger_count >= 4 AND tolls_amount = 0
```

Linha	avg_tip_amount_g2
1	3.422985075

É possível perceber que o grupo 1 possui uma média de gorjetas de ≈ 82,17% maior que o grupo 2, no qual não paga pedágio e possuem mais passageiros por corrida.

Projeto em Análise de Dados