Analyzing the Impact of Business Hour Mismatch on Order Volume in the Food Delivery Industry: A Case Study of UEats and Ghub

By Abhigyan Dutta

abhigyandutta72@gmail.com

Queries:

1. Checking the Grubhub business hours(open time and close time):

```
WITH schedule_rules AS (
  -- Extract relevant schedule data (day, open and close time) for each virtual kitchen.
  SELECT
    vb_name,  -- Name or ID of the virtual kitchen.
    JSON_EXTRACT_SCALAR(value, '$.days_of_week[0]') AS day,  -- Extract the first day of the week from JSON
data.
    JSON_EXTRACT_SCALAR(value, '$.from') AS open_time,  -- Extract opening time from the schedule JSON.
    JSON_EXTRACT_SCALAR(value, '$.to') AS close_time  -- Extract closing time from the schedule JSON.
  FROM `arboreal-vision-339901.take_home_v2.virtual_kitchen_grubhub_hours`,
   UNNEST(JSON_EXTRACT_ARRAY(response,
     '$.availability_by_catalog.STANDARD_DELIVERY.schedule_rules')) AS value  -- Flatten the JSON array of
schedule rules.
)

SELECT
  vb_name AS Virtual_Restaurant_ID,
  ARRAY_AGG(
    STRUCT(
      day,  -- The day of the week.
      open_time,  -- The opening time.
      close_time  -- The closing time.
    )
  ) AS business_hours  -- Combine day, open, and close times into an array of business hours for each
restaurant.
FROM schedule_rules
GROUP BY Virtual_Restaurant_ID  -- Group by the virtual kitchen ID to get all business hours for each
kitchen.
LIMIT 5;  -- For simplicity, limit the result to the first 5 virtual restaurants.
```

2. Checking the Ubereats business hours(open time and close time):

```
-- Checking all fields of the UberEats virtual kitchen hours table
SELECT *
FROM `arboreal-vision-339901.take_home_v2.virtual_kitchen_ubereats_hours`
LIMIT 5;  -- Limit to 5 records for simplicity

-- Extracting UberEats start and end times from the nested JSON
SELECT
      JSON_EXTRACT(value, "$.regularHours.endTime") AS end_time,  -- Extract the end time from the regular
hours section of the JSON.
      JSON_EXTRACT(value, "$.regularHours.startTime") AS start_time  -- Extract the start time from the
regular hours section of the JSON.
FROM
  `arboreal-vision-339901.take_home_v2.virtual_kitchen_ubereats_hours`,  -- Use the UberEats virtual kitchen
hours table.
  UNNEST(JSON_QUERY_ARRAY(response, '$.data.menus.sections')) as value  -- Flatten the JSON array to access
the menus' sections.
```

3. Final query that checks the range between Virtual Restuarant Business Hours and ubereats business hours:

```
WITH grubhub_hours AS (
  SELECT
    JSON_EXTRACT(response, '$["slug"]') AS gh_slug,
    JSON_EXTRACT(response, '$["openHours"]') AS gh_open_hours
  FROM `arboreal-vision-339901.take_home_v2.virtual_kitchen_grubhub_hours`
),
ubereats_hours AS (
  SELECT
    JSON_EXTRACT(menu, '$[0]["key"]') AS ue_slug,
    JSON_EXTRACT(menu, '$[0]["sections"][0]["regularHours"][0]') AS ue_start_time,
    JSON_EXTRACT(menu, '$[0]["sections"][0]["regularHours"][1]') AS ue_end_time
  FROM `arboreal-vision-339901.take_home_v2.virtual_kitchen_ubereats_hours`
),
hours_joined AS (
  SELECT
    gh_slug,
    gh_open_hours,
    ue_slug,
    ue_start_time,
    ue_end_time
  FROM grubhub_hours
  JOIN ubereats_hours
    ON JSON_EXTRACT(gh_open_hours, '$[0]') = ue_slug
)
SELECT
  gh_slug,
  JSON_EXTRACT(gh_open_hours, '$[0]') AS gh_open_hours_string,
  ue_slug,
  ue_start_time,
  ue_end_time,
  CASE
    WHEN PARSE_TIMESTAMP('%I:%M %p', JSON_EXTRACT(gh_open_hours, '$[1]')) BETWEEN PARSE_TIMESTAMP('%I:%M %p',
ue_start_time) AND PARSE_TIMESTAMP('%I:%M %p', ue_end_time) THEN "In Range"
    WHEN ABS(TIMESTAMP_DIFF(PARSE_TIMESTAMP('%I:%M %p', JSON_EXTRACT(gh_open_hours, '$[1]')),
PARSE_TIMESTAMP('%I:%M %p', ue_start_time), MINUTE)) < 5 THEN "Out of Range with 5 mins difference"
    ELSE "Out of Range"
  END AS is_out_of_range
FROM hours_joined
```

Note:

Sample Python code that I used to check the Start time and the end time for ubereats(for part 2):

```
import json

# JSON response
response = '''
{
    "data": {
        "menuMapping": [
            {
                "menuType": "MENU_TYPE_FULFILLMENT_DELIVERY",
                "menuUUID": "26bd579e-5664-4f0a-8465-2f5eb5fbe705"
            },
```

```
            {
                "menuType": "MENU_TYPE_FULFILLMENT_PICK_UP",
                "menuUUID": "26bd579e-5664-4f0a-8465-2f5eb5fbe705"
            }
        ],
        "menuStructures": {
            "26bd579e-5664-4f0a-8465-2f5eb5fbe705": null
        },
        "menus": {
            "26bd579e-5664-4f0a-8465-2f5eb5fbe705": {
                "sections": [
                    {
                        "regularHours": {
                            "startTime": "08:00",
                            "endTime": "22:00"
                        }
                    },
                    {
                        "regularHours": {
                            "startTime": "09:00",
                            "endTime": "23:00"
                        }
                    }
                ]
            }
        }
    }
}
'''

# Parse the JSON response
response_data = json.loads(response)

# Extract values using list comprehension
menuUUID = response_data['data']['menuMapping'][0]['menuUUID']
sections = response_data['data']['menus'][menuUUID]['sections']

# Extract the start_time and end_time from each section
for section in sections:
    start_time = section['regularHours']['startTime']
    end_time = section['regularHours']['endTime']
    print(f"Start Time: {start_time}, End Time: {end_time}")
```

Output:

```
PS C:\LEARNING\01 PROJECTS\Detection-of-Neurodevelopment-Disorder-main> python -u "c:\LEARNING\Loop ai assesment\import json.py"
Start Time: 08:00, End Time: 22:00
Start Time: 09:00, End Time: 23:00
PS C:\LEARNING\01 PROJECTS\Detection-of-Neurodevelopment-Disorder-main>
```