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

Problem Statement:

- Ensure store hours are consistent across UberEats and Grubhub to avoid mismatched customer experiences and operational issues.
- UberEats hours are the ground truth, and Grubhub hours will be compared against them.
- **Goal**: Identify if Grubhub hours are: In Range, Out of Range with 5 mins difference OR Out of Range.

Schema Used:
UberEats
arboreal-vision-339901.take_home_v2.virtual_kitchen_ubereats_hours
Grubhub
arboreal-vision-339901.take_home_v2.virtual_kitchen_grubhub_hours
Join Key: (b_name, vb_name)

Assumptions:
- Latest timestamp per store (slug) is used in both datasets.
- daysBitArray is parsed as Monday - Sunday (index 0 = Monday).
- Time format: assumed as "%H:%M" for UberEats and "%H:%M:%E*S" for Grubhub.

SQL Solution:

```
-- Step 1: Latest UberEats records per store
WITH latest_ubereats AS (
  SELECT *
  FROM (
     SELECT *, ROW_NUMBER() OVER (PARTITION BY b_name, vb_name ORDER BY timestamp DESC) AS rn
    FROM arboreal-vision-339901.take_home_v2.virtual_kitchen_ubereats_hours
  )
  WHERE rn = 1
),
-- Step 2: Flatten UberEats business hours and apply daysBitArray logic
ubereats_hours_flat AS (
  SELECT
    b_name,
    vb_name,
    slug AS ubereats_slug,
    PARSE_TIME('%H:%M', JSON_EXTRACT_SCALAR(hour, '$.startTime')) AS ue_open,
    PARSE_TIME('%H:%M', JSON_EXTRACT_SCALAR(hour, '$.endTime')) AS ue_close,
    day_index
  FROM latest_ubereats,
   UNNEST(REGEXP_EXTRACT_ALL(TO_JSON_STRING(response), r'"regularHours":\[(\{.*?\})\]')) AS hours_block,
    UNNEST([STRUCT(JSON_EXTRACT_ARRAY('[' || hours_block || ']', '$') AS hour_array)]) AS ha,
    UNNEST(ha.hour_array) AS hour,
    UNNEST(GENERATE_ARRAY(0, 6)) AS day_index,
```

```
  UNNEST([STRUCT(JSON_EXTRACT_ARRAY(hour, '$.daysBitArray') AS days_array)]) AS temp
 WHERE JSON_EXTRACT_SCALAR(days_array[OFFSET(day_index)], '$') = 'true'
),
-- Step 3: Latest Grubhub records per store
latest_grubhub AS (
 SELECT *
 FROM (
    SELECT *, ROW_NUMBER() OVER (PARTITION BY b_name, vb_name ORDER BY timestamp
DESC) AS rn
  FROM arboreal-vision-339901.take_home_v2.virtual_kitchen_grubhub_hours
 )
 WHERE rn = 1
),
-- Step 4: Flatten Grubhub hours and parse times using %H:%M:%E*S
grubhub_hours_flat AS (
 SELECT
  gh.b_name,
  gh.vb_name,
  gh.slug AS grubhub_slug,

  CASE LOWER(JSON_EXTRACT_SCALAR(day_name_array_item, '$'))
    WHEN 'monday' THEN 0
    WHEN 'tuesday' THEN 1
    WHEN 'wednesday' THEN 2
    WHEN 'thursday' THEN 3
    WHEN 'friday' THEN 4
    WHEN 'saturday' THEN 5
    WHEN 'sunday' THEN 6
  END AS day_index,

  SAFE.PARSE_TIME('%H:%M:%E*S', TRIM(JSON_EXTRACT_SCALAR(schedule_rule, '$.from'))) AS
gh_open,
    SAFE.PARSE_TIME('%H:%M:%E*S', TRIM(JSON_EXTRACT_SCALAR(schedule_rule, '$.to'))) AS
gh_close

 FROM latest_grubhub AS gh,
  UNNEST(JSON_EXTRACT_ARRAY(response,
'$.availability_by_catalog.STANDARD_DELIVERY.schedule_rules')) AS schedule_rule,
  UNNEST(JSON_EXTRACT_ARRAY(schedule_rule, '$.days_of_week')) AS day_name_array_item
 WHERE
  JSON_EXTRACT_SCALAR(schedule_rule, '$.from') IS NOT NULL
  AND JSON_EXTRACT_SCALAR(schedule_rule, '$.to') IS NOT NULL
),

-- Step 5: Join UberEats and Grubhub hours by store and day
joined_hours AS (
 SELECT
  gh.grubhub_slug,
  ue.ubereats_slug,
  gh.b_name,
  gh.vb_name,
```

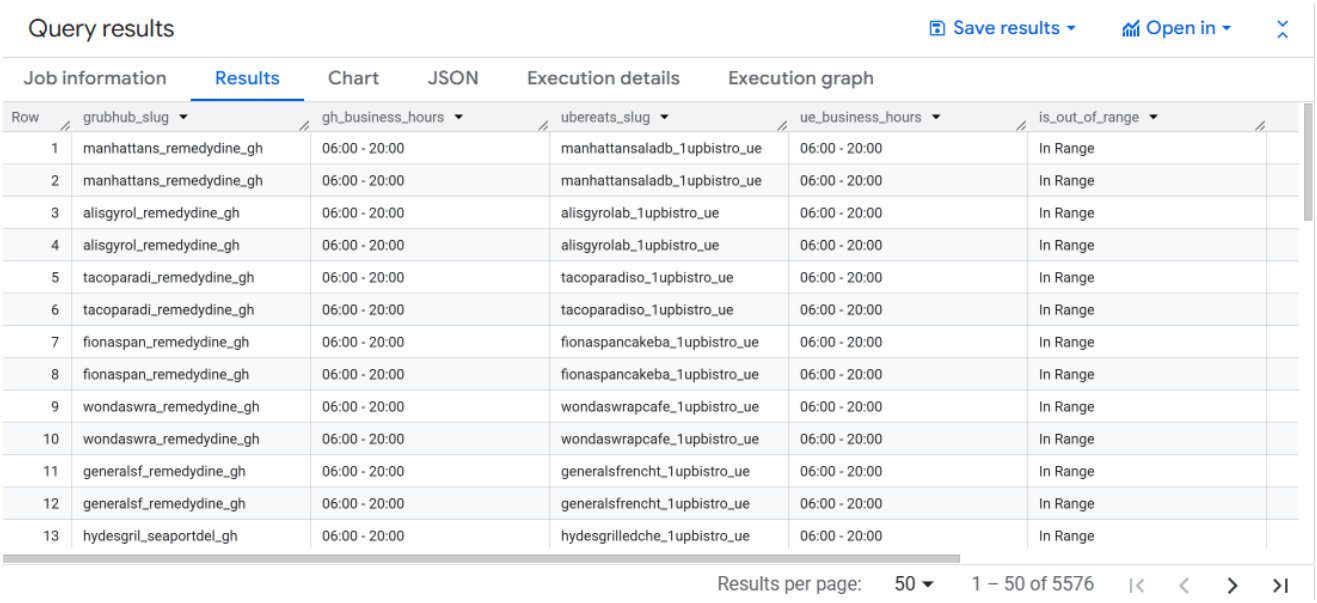
```
    gh.day_index,
    gh.gh_open,
    gh.gh_close,
    ue.ue_open,
    ue.ue_close
  FROM grubhub_hours_flat AS gh
  INNER JOIN ubereats_hours_flat AS ue
    ON LOWER(gh.b_name) = LOWER(ue.b_name)
    AND LOWER(gh.vb_name) = LOWER(ue.vb_name)
    AND gh.day_index = ue.day_index
)

-- Step 6: Final Output
SELECT
  grubhub_slug,
    CONCAT(FORMAT_TIME('%H:%M', gh_open), ' - ', FORMAT_TIME('%H:%M', gh_close)) AS gh_business_hours,
  ubereats_slug,
    CONCAT(FORMAT_TIME('%H:%M', ue_open), ' - ', FORMAT_TIME('%H:%M', ue_close)) AS ue_business_hours,
  CASE
    WHEN gh_open = ue_open AND gh_close = ue_close THEN 'In Range'
    WHEN ABS(TIME_DIFF(gh_open, ue_open, MINUTE)) <= 5
          AND ABS(TIME_DIFF(gh_close, ue_close, MINUTE)) <= 5 THEN 'Out of Range with 5 mins difference'
    ELSE 'Out of Range'
  END AS is_out_of_range
FROM joined_hours
ORDER BY b_name, day_index;
```

Business Hours Comparison Criteria:

- **“In Range”**: Indicates that the opening and closing times for both UberEats and Grubhub are exactly the same.
- **“Out of Range with 5 Minutes Difference”**: Applied when the opening and/or closing times between UberEats and Grubhub differ by up to 5 minutes.
- **“Out of Range”**: Used when the difference in opening and/or closing times between UberEats and Grubhub exceeds 5 minutes.

Final Output Sample:


Query results

Save results | Open in

Job information | Results | Chart | JSON | Execution details | Execution graph

Row	grubhub_slug	gh_business_hours	ubereats_slug	ue_business_hours	is_out_of_range
1	manhattans_remedydine_gh	06:00 - 20:00	manhattansaladb_1upbistro_ue	06:00 - 20:00	In Range
2	manhattans_remedydine_gh	06:00 - 20:00	manhattansaladb_1upbistro_ue	06:00 - 20:00	In Range
3	alisgyrol_remedydine_gh	06:00 - 20:00	alisgyrolab_1upbistro_ue	06:00 - 20:00	In Range
4	alisgyrol_remedydine_gh	06:00 - 20:00	alisgyrolab_1upbistro_ue	06:00 - 20:00	In Range
5	tacoparadi_remedydine_gh	06:00 - 20:00	tacoparadiso_1upbistro_ue	06:00 - 20:00	In Range
6	tacoparadi_remedydine_gh	06:00 - 20:00	tacoparadiso_1upbistro_ue	06:00 - 20:00	In Range
7	fionaspan_remedydine_gh	06:00 - 20:00	fionaspancakeba_1upbistro_ue	06:00 - 20:00	In Range
8	fionaspan_remedydine_gh	06:00 - 20:00	fionaspancakeba_1upbistro_ue	06:00 - 20:00	In Range
9	wondaswra_remedydine_gh	06:00 - 20:00	wondaswrapcafe_1upbistro_ue	06:00 - 20:00	In Range
10	wondaswra_remedydine_gh	06:00 - 20:00	wondaswrapcafe_1upbistro_ue	06:00 - 20:00	In Range
11	generalsf_remedydine_gh	06:00 - 20:00	generalsfrencht_1upbistro_ue	06:00 - 20:00	In Range
12	generalsf_remedydine_gh	06:00 - 20:00	generalsfrencht_1upbistro_ue	06:00 - 20:00	In Range
13	hydesgril_seaportdel_gh	06:00 - 20:00	hydesgrilledche_1upbistro_ue	06:00 - 20:00	In Range

Results per page: 50 1 – 50 of 5576

Business Value:

- Proactively monitor store configurations.
- Prevent service disruptions and Service Level Agreements violations.
- Build trust through consistent availability across platforms.