### **Question:**

The task involves comparing the business hours of restaurants on **UberEats** and **Grubhub** platforms. Specifically, the goal is to:

* Extract the business hours for both platforms.
* Compare the business hours for the same restaurant across both platforms.
* Identify whether the business hours on **Grubhub** fall within the business hours on **UberEats**, or if they are **out of range** (either completely or with a slight 5-minute difference).

### **Approach:**

1. **Extract Business Hours for Both Platforms:**
   * **UberEats:** The business hours are stored within a nested JSON structure. We need to extract the start time and end time for the restaurant’s regular hours from the JSON response.
   * **Grubhub:** The business hours are stored in a simpler JSON structure, which provides a from and to field representing the start and end times, respectively.
2. **Join Data on Restaurant Information:**
   * Both UberEats and Grubhub provide a **business name** (b\_name) and a **virtual brand name** (vb\_name) to uniquely identify each restaurant. To compare the business hours of the same restaurant on both platforms, we join the two datasets based on these fields.
3. **Compare Business Hours:**
   * After extracting the business hours for both platforms, the next step is to compare:
     + If **Grubhub's** business hours are entirely within the range of **UberEats'** business hours, the result will be "In Range".
     + If **Grubhub's** business hours fall outside of **UberEats'** hours, it will be marked as "Out of Range".
     + A special case is handled where there is a **slight 5-minute difference** between the business hours, in which case the result will be "Out of Range with 5 mins difference".
4. **Return the Result:**
   * The result will return the **Grubhub slug**, **Grubhub business hours**, **UberEats slug**, and **UberEats business hours** along with the status of the comparison (whether the business hours are "In Range", "Out of Range", or "Out of Range with 5 mins difference").

### **Breakdown of the Code:**

1. **Ubereats CTE (Common Table Expression):**
   * In the Ubereats CTE, we extract the **start time** and **end time** for the UberEats restaurant's business hours from the nested JSON. The b\_name and vb\_name are selected to help join the data later with Grubhub.
   * The selected fields are:
     + ue\_slug: The unique identifier for the UberEats restaurant.
     + Ubereats\_starttime: The start time of the restaurant's business hours.
     + Ubereats\_endtime: The end time of the restaurant's business hours.
     + restaurant\_info: A structured combination of business name and virtual brand name for the join.
2. **Grubhub CTE:**
   * Similarly, in the Grubhub CTE, we extract the **start time** (from) and **end time** (to) for Grubhub’s business hours. This data is much simpler to extract as it is directly available in the JSON response.
   * The selected fields are:
     + gh\_slug: The unique identifier for the Grubhub restaurant.
     + Grubhub\_starttime: The start time of Grubhub's business hours.
     + Grubhub\_endtime: The end time of Grubhub's business hours.
     + restaurant\_info: The structured combination of business name and virtual brand name to ensure proper data matching.
3. **Main Query - Joining the Data:**
   * The main query combines both **Ubereats** and **Grubhub** using an INNER JOIN based on the restaurant\_info. This ensures that only the records for the same restaurant on both platforms are compared.
4. **Comparing Business Hours:**
   * The CASE statement compares the business hours for **Grubhub** and **UberEats**:
     + If **Grubhub's** start time is greater than or equal to **UberEats'** start time and **Grubhub's** end time is less than or equal to **UberEats'** end time, it’s marked as **"In Range"**.
     + If **Grubhub's** start time is earlier than **UberEats'** start time or **Grubhub's** end time is later than **UberEats'** end time, it’s marked as **"Out of Range"**.
     + A special case where there is a **5-minute difference** in the hours is flagged as **"Out of Range with 5 mins difference"**.
5. **Final Output:**
   * The final output returns:
     + gh\_slug and gh\_business\_hours for **Grubhub**.
     + ue\_slug and ue\_business\_hours for **UberEats**.
     + A new column is\_out\_of\_range that indicates whether the business hours are in range, out of range, or within 5 minutes difference.

### **Conclusion:**

The solution efficiently compares the business hours between **UberEats** and **Grubhub** using SQL queries that handle data extraction, transformation, and comparison. By structuring the data extraction process and using logical conditions to compare the business hours, we were able to provide meaningful insights about whether the times are aligned or not. This method ensures that even with slight time discrepancies, the solution remains accurate and practical.