

SQL INTERVIEW PREPARATION PART 2

Table Structure

id: int64
time_id: datetime64[ns]
user_id: object
customer_id: object
client_id: object
event_type: object
event_id: int64

Question: Most Popular Client ID: Select the most popular client_id based on a count of the number of users who have at least 50% of their events from the following list: 'video call received', 'video call sent', 'voice call received', 'voice call sent'.

Solution:

```
WITH EventSummary AS (  
    SELECT  
        user_id,  
        client_id,  
        COUNT(CASE WHEN event_type IN ('video call received', 'video call sent', 'voice call  
received', 'voice call sent') THEN 1 END) AS special_event_count,  
        COUNT(*) AS total_event_count  
    FROM events_table  
    GROUP BY user_id, client_id  
)  
  
QualifiedUsers AS (  
    SELECT  
        client_id,  
        user_id  
    FROM EventSummary  
    WHERE special_event_count >= 0.5 * total_event_count  
)
```

```

ClientUserCount AS (
    SELECT
        client_id,
        COUNT(DISTINCT user_id) AS user_count
    FROM QualifiedUsers
    GROUP BY client_id
)
SELECT client_id
FROM ClientUserCount
ORDER BY user_count DESC
LIMIT 1;

```

Detailed Explanation of the query:

This query identifies the **most popular client_id**, based on the **number of users** who have at least **50% of their events** related to specific types (video call received, video call sent, voice call received, voice call sent). Here's the breakdown:

Step 1: Event Summary (EventSummary CTE)

```

WITH EventSummary AS (
    SELECT
        user_id,
        client_id,
        COUNT(CASE WHEN event_type IN ('video call received', 'video call sent', 'voice call received', 'voice call sent') THEN 1 END) AS special_event_count,
        COUNT(*) AS total_event_count
    FROM events_table
    GROUP BY user_id, client_id
)

```

- **Purpose:** Calculate the total events (total_event_count) and the count of special events (special_event_count) for each user and client combination.
- **Key Elements:**
 - COUNT(CASE WHEN ...): Counts the events that match the specific event types.

- COUNT(*): Counts all events for the user_id and client_id.
- GROUP BY user_id, client_id: Groups events for each user_id and client_id.

Step 2: Filter Qualified Users (QualifiedUsers CTE)

QualifiedUsers AS (

SELECT

client_id,

user_id

FROM EventSummary

WHERE special_event_count >= 0.5 * total_event_count

)

- **Purpose:** Identify users who have at least **50% of their events** as the special event types.
- **Key Elements:**
 - WHERE special_event_count >= 0.5 * total_event_count: Filters users whose ratio of special events to total events is at least 50%.
 - Only the user_id and client_id of such users are retained.

Step 3: Count Users per Client (ClientUserCount CTE)

ClientUserCount AS (

SELECT

client_id,

COUNT(DISTINCT user_id) AS user_count

FROM QualifiedUsers

GROUP BY client_id

)

- **Purpose:** Count the number of unique users who meet the 50% threshold for each client_id.
- **Key Elements:**
 - COUNT(DISTINCT user_id): Ensures each user is counted only once per client_id.
 - GROUP BY client_id: Aggregates data at the client_id level.

Step 4: Retrieve the Most Popular Client

SELECT client_id

```
FROM ClientUserCount

ORDER BY user_count DESC

LIMIT 1;
```

- **Purpose:** Identify the client with the highest number of users meeting the criteria.
- **Key Elements:**
 - ORDER BY user_count DESC: Sorts clients by their user count in descending order.
 - LIMIT 1: Returns only the top client.

Query Flow:

1. **EventSummary:** Calculates event counts (total and special) for each user-client pair.
2. **QualifiedUsers:** Filters users who meet the 50% threshold of special events.
3. **ClientUserCount:** Aggregates the count of such users per client_id.
4. **Final Selection:** Identifies the client_id with the highest number of users.

Example:

Input Table (events_table):

user_id	client_id	event_type	event_id
1	A	video call received	101
1	A	video call sent	102
1	A	browse	103
2	B	voice call sent	201
2	B	voice call received	202
3	A	browse	301

Step-by-Step Output:

1. **EventSummary:**

user_id	client_id	special_event_count	total_event_count
1	A	2	3
2	B	2	2
3	A	0	1

2. **QualifiedUsers:**

client_id	user_id
-----------	---------

A	1
---	---

B	2
---	---

3. **ClientUserCount:**

client_id	user_count
-----------	------------

A	1
---	---

B	1
---	---

4. **Final Output:**

client_id

A

In this case, both clients have an equal user count, but sorting by other criteria (like client_id) may be necessary in case of ties.

Key Learnings:

- **Common Table Expressions (CTEs):** Break down complex logic into readable and reusable components.
- **Window Functions:** Use for aggregate and partition-based calculations.
- **Filtering Logic:** Helps derive insights based on specific conditions.
- **Ranking and Aggregation:** Useful for identifying top-performing entities.