SQL INTERVIEW PREPARATION PART 2.1

Table Structure

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

Question 1: **Desktop-Only Users: Write a query that returns the company (customer_id column) with the highest number of users that use desktop only.**

Solution:

```sql
WITH DeviceUsage AS (
   SELECT
     user_id,
     customer_id,
     COUNT(DISTINCT client_id) AS distinct_clients
   FROM events_table
   WHERE client_id = 'desktop'
   GROUP BY user_id, customer_id
)
SELECT customer_id, COUNT(user_id) AS desktop_only_users
FROM DeviceUsage
WHERE distinct_clients = 1
GROUP BY customer_id
ORDER BY desktop_only_users DESC
LIMIT 1;
```

Query Explanation:

The query identifies the **company (customer_id)** with the **highest number of "desktop-only" users**, where "desktop-only" users are those who use **only the desktop client**.

Steps:

1. **DeviceUsage CTE:**
   - Counts the number of distinct client_id values per user in each company (customer_id) for desktop users.
   - Filters only desktop users.
2. **Final Query:**
   - Filters out users who used **only the desktop client** (distinct_clients = 1).
   - Counts these users for each company.
   - Sorts companies by the number of desktop-only users in descending order.
   - Returns the top company.

Example Input Table (events_table):

user_id	customer_id	client_id	event_type
1	A	desktop	login
2	A	desktop	purchase
3	A	mobile	browse
4	B	desktop	browse
5	B	desktop	login
6	B	desktop	purchase
7	C	mobile	login

Query Output:

1. **DeviceUsage CTE:**

user_id	customer_id	distinct_clients
1	A	1
2	A	1
3	A	2
4	B	1
5	B	1
6	B	1

2. **Final Output:**

customer_id	**desktop_only_users**
B	3

Result: The top company is **B** with **3 desktop-only users**.

Question 2: **Bottom Companies by Mobile Usage: Write a query that returns a list of the bottom 2 companies by mobile usage. Company is defined in the customer_id column. Mobile usage is defined as the number of events registered on a client_id == 'mobile'. Order the result by the number of events ascending. In the case where there are multiple companies tied for the bottom ranks (rank 1 or 2), return all the companies. Output the customer_id and number of events.**

Solution:

```
WITH MobileUsage AS (
  SELECT
    customer_id,
    COUNT(*) AS mobile_event_count
  FROM events_table
  WHERE client_id = 'mobile'
  GROUP BY customer_id
),
RankedCompanies AS (
  SELECT
    customer_id,
    mobile_event_count,
    DENSE_RANK() OVER (ORDER BY mobile_event_count ASC) AS rank
  FROM MobileUsage
)
SELECT customer_id, mobile_event_count
FROM RankedCompanies
WHERE rank <= 2;
```

Query Explanation:

The query finds the **bottom 2 companies (customer_id) with the lowest mobile event counts** (client_id = 'mobile).

Steps:

1. **MobileUsage CTE:**
   - Counts the total mobile events (mobile_event_count) for each company.
2. **RankedCompanies CTE:**
   - Assigns a **rank** to companies based on their mobile event count in **ascending order** using DENSE_RANK.
3. **Final Query:**
   - Filters and retrieves companies with a rank of **1 or 2** (bottom two companies).

Example Input Table (events_table):

customer_id	user_id	client_id	event_type
A	1	mobile	login
A	2	mobile	browse
B	3	desktop	purchase
C	4	mobile	login
C	5	mobile	browse
D	6	mobile	login

Query Output:

1. **MobileUsage CTE:**

customer_id	mobile_event_count
A	2
C	2
D	1

2. **RankedCompanies CTE:**

customer_id	mobile_event_count	rank
D	1	1
A	2	2
C	2	2

3. **Final Output:**

customer_id	mobile_event_count
D	1
A	2
C	2

Result: The query returns **D, A, C**, since they are tied for the bottom 2 ranks.

<u>Question 3:</u> **Exclusive Users per Client: Write a query that returns a number of users who are exclusive to only one client. Output the client_id and number of exclusive users.**

<u>Solution:</u>

```
WITH ClientUsage AS (
  SELECT
    user_id,
    client_id,
    COUNT(DISTINCT client_id) AS client_count
  FROM events_table
  GROUP BY user_id, client_id
)
SELECT client_id, COUNT(user_id) AS exclusive_user_count
FROM ClientUsage
WHERE client_count = 1
GROUP BY client_id;
```

Query Explanation:

This query identifies the number of **exclusive users** for each client_id. A user is considered exclusive if they use **only one client** (e.g., only mobile or only desktop).

Steps:

1. **ClientUsage CTE:**
   - Counts the distinct number of clients (client_count) each user has interacted with.
2. **Final Query:**
   - Filters users with client_count = 1 (exclusive users).
   - Groups by client_id and counts the number of such exclusive users (exclusive_user_count).

Example Input Table (events_table):

user_id	client_id	event_type
1	mobile	login
1	mobile	browse
2	desktop	purchase
3	desktop	login
3	mobile	browse
4	desktop	login

Query Output:

1. **ClientUsage CTE:**

user_id	client_id	client_count
1	mobile	1
2	desktop	1
3	desktop	2
4	desktop	1

2. **Final Output:**

client_id	**exclusive_user_count**
mobile	1
desktop	2

Result: The query identifies exclusive users for each client (mobile: 1, desktop: 2).