Documentation for question 1

Overview

This document outlines the process of querying user revenue and group participation based on order data. Initially, the query was designed to filter for orders completed in the last 30 days. However, due to the absence of any completed orders within that timeframe, the query was adjusted to extend the date range to the last 60 days.

Initial Query: Last 30 Days

The first query aimed to retrieve the top 10 users who contributed the highest revenue in the last 30 days. The query included the following components:

Data Sources: Orders, Users, Groups, Group Carts, Group Deals, Products, Product Names, and Categories.

Filters: Orders created within the last 30 days and marked as "COMPLETED".

Query to Check Orders in the Last 30 Days

Before executing the main query, the following SQL command was used to check if there were any completed orders in the last 30 days:

sql

Copy

```
SELECT
    *
FROM
    orders
WHERE
    created_at >= NOW() - INTERVAL '30 days'
    AND status = 'COMPLETED';  -- Ensure 'COMPLETED' is the correct status
```

Outcome of the 30-Day Check

Upon executing the query to check for completed orders in the last 30 days, it was determined that there were no completed orders within that timeframe, resulting in zero rows returned.

Main Query Example

The main query designed to analyze user contributions in the last 30 days is as follows:

sql

Copy

```
WITH UserRevenue AS (
    SELECT
        gc.user_id,
        SUM(o.total_amount) AS total_revenue,
        COUNT(DISTINCT g.id) AS group_count
    FROM
        orders o
    JOIN
        groups_carts gc ON o.groups_carts_id = gc.id
    JOIN
        groups g ON gc.group_id = g.id
    WHERE
        o.created_at >= NOW() - INTERVAL '30 days'
        AND o.status = 'COMPLETED'
    GROUP BY
        gc.user_id
),
UserCategories AS (
    SELECT
        u.id AS user_id,
        ARRAY_AGG(DISTINCT c.name) AS categories
    FROM
```

```
        users u
    JOIN
        groups_carts gc ON u.id = gc.user_id
    JOIN
        groups g ON gc.group_id = g.id
    JOIN
        group_deals gd ON g.group_deals_id = gd.id
    JOIN
        products p ON gd.product_id = p.id
    JOIN
        product_names pn ON p.name_id = pn.id
    JOIN
        categories c ON pn.category_id = c.id
    GROUP BY
        u.id
)
SELECT
    u.user_id,
    u.total_revenue,
    u.group_count,
    uc.categories
FROM
    UserRevenue u
LEFT JOIN
    UserCategories uc ON u.user_id = uc.user_id
ORDER BY
    u.total_revenue DESC
LIMIT 10;
```

Updated Query: Last 60 Days

To ensure more comprehensive data retrieval, the query was modified to filter for orders completed in the last 60 days. This adjustment aimed to capture any relevant transactions that may have occurred outside the initial 30-day window.

Updated Query Example

sql

Copy

```
WITH UserRevenue AS (
    SELECT
        gc.user_id,
        SUM(o.total_amount) AS total_revenue,
        COUNT(DISTINCT g.id) AS group_count
    FROM
        orders o
    JOIN
        groups_carts gc ON o.groups_carts_id = gc.id
    JOIN
        groups g ON gc.group_id = g.id
    WHERE
        o.created_at >= NOW() - INTERVAL '60 days'
        AND o.status = 'COMPLETED'
    GROUP BY
        gc.user_id
),
UserCategories AS (
    SELECT
        u.id AS user_id,
        ARRAY_AGG(DISTINCT c.name) AS categories
    FROM
        users u
    JOIN
        groups_carts gc ON u.id = gc.user_id
    JOIN
        groups g ON gc.group_id = g.id
    JOIN
        group_deals gd ON g.group_deals_id = gd.id
    JOIN
        products p ON gd.product_id = p.id
    JOIN
        product_names pn ON p.name_id = pn.id
    JOIN
        categories c ON pn.category_id = c.id
    GROUP BY
        u.id
)
SELECT
    u.user_id,
    u.total_revenue,
    u.group_count,
    uc.categories
FROM
    UserRevenue u
LEFT JOIN
    UserCategories uc ON u.user_id = uc.user_id
ORDER BY
    u.total_revenue DESC
LIMIT 10;
```

Outcome

By extending the date range to the last 60 days, the updated query allows for a broader analysis of user contributions and group participation. This adjustment is expected to yield more results, providing insights into user behavior and product categories.

Conclusion

The transition from a 30-day to a 60-day query reflects a necessary adaptation in data retrieval strategies to ensure comprehensive analysis. Future queries may continue to adjust the date range based on data availability and business needs.