SQL QUERIES

Can a user show up more than once in the activity table?

Yes. Since the activity table contains a row per day for each purchase made by a user, a user can appear multiple times on different days or even the same day if they make purchases on separate occasions.

What type of join should we use to join the users table to the activity table?

We should use an INNER JOIN if we are only interested in users who have made a purchase (i.e., those present in both the users and activity tables). However, if we want to include all users regardless of whether they've made a purchase, a LEFT JOIN (with users as the left table) would be appropriate.

What SQL function can we use to fill in NULL values?

The SQL function COALESCE can be used to fill in NULL values. This function returns the first non-null value in a list.

What are the start and end dates of the experiment?

```
SELECT
   MIN(dt) AS start_date,
   MAX(dt) AS end_date
FROM activity;
```

How many total users were in the experiment?

```
SELECT COUNT(DISTINCT id) AS total_users
FROM users;
```

What was the conversion rate of all users?

```
SELECT
   (SELECT COUNT(DISTINCT uid)
    FROM activity
    WHERE spent > 0) /
   CAST((SELECT COUNT(DISTINCT id)
        FROM users) AS FLOAT) AS conversion_rate
```

What is the user conversion rate for the control and treatment groups?

```
1.SELECT
   u.group,
   COUNT(DISTINCT CASE WHEN a.spent > 0 THEN u.uid END) /
   CAST(COUNT(DISTINCT u.uid) AS FLOAT) AS conversion_rate
FROM users u
```

```
LEFT JOIN activity a ON u.uid = a.uid
GROUP BY u.group;
```

```
2.WITH cte AS (
  SELECT uid, "group", SUM(spent) AS total_spent
  FROM groups
  LEFT JOIN activity USING (uid)
  GROUP BY uid, "group"
)
SELECT "group", ROUND(AVG(converted) * 100, 2) AS conversion_rate
FROM (
  SELECT uid, "group", COALESCE(total_spent, 0) AS total_spent,
    CASE
      WHEN total_spent IS NOT NULL THEN 1
      ELSE 0
    END AS converted
  FROM cte
) AS subquery
GROUP BY "group";
```

What is the average amount spent per user for the control and treatment groups, including users who did not convert?

```
SELECT
    u.group,
    COALESCE(SUM(a.spent), 0) / COUNT(DISTINCT u.uid) AS avg_amount_spent
FROM users u
LEFT JOIN activity a ON u.uid = a.uid
GROUP BY u.group;
```

```
.SELECT
    u.id AS user_id,
    u.country,
    u.gender,
    a.device AS user_device,
    g.group AS test_group,
    g.join_dt AS join_date,
    CASE WHEN a.spent > 0 THEN 1 ELSE 0 END AS converted,
    COALESCE(SUM(a.spent), 0) AS total_spent
FROM
    users u
LEFT JOIN activity a ON u.id = a.uid
LEFT JOIN groups g ON u.id = g.uid
GROUP BY
    u.id,
    u.country,
    u.gender,
    a.device,
    g.group,
    g.join_dt,
    CASE WHEN a.spent > 0 THEN 1 ELSE 0 END;
```

Why does it matter to include users who did not convert when calculating the average amount spent per user?

In order to measure the impact on total revenue (amount spent), we cannot only average the users who converted because there could have been fewer users who converted in the treatment.

Finally, here's the SQL query to extract the analysis dataset as requested:

1. Write a SQL query that returns: the user ID, the user’s country, the user’s gender, the user’s device type, the user’s test group, whether or not they converted (spent > $0), and how much they spent in total ($0+).

```
SELECT
   u.id AS user_id,
   u.country,
   u.gender,
   u.device,
   u.group AS test_group,
   CASE
      WHEN SUM(a.spent) > 0 THEN 'Converted'
      ELSE 'Not Converted'
   END AS conversion_status,
   COALESCE(SUM(a.spent), 0) AS total_spent
FROM users u
LEFT JOIN activity a ON u.uid = a.uid
GROUP BY u.id, u.country, u.gender, u.device, u.group;
```