

GloBox - A/B test

SQL Queries

1. Understanding the Database

1. Can a user show up more than once in the activity table? Yes or no, and why?

Yes, because 139 users have accessed and purchased more than once during experiment period

Query:

```
SELECT COUNT(*)- COUNT(DISTINCT uid) AS id_duplicates  
FROM activity;
```

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

Left join and Inner join will give the same result and will be useful in case the analysis will focus on the people which are recorded into activity group. To have a look at all the users(also those who did not interact during the experiment period) need a right join.

Query:

```
SELECT  
    activity.uid,  
    activity.spent,  
    activity.device,  
    users.country,  
    users.gender  
FROM activity  
LEFT JOIN users
```

ON activity.uid = users.id;

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

Coalesce function

Query:

```
SELECT
    uid,
    COALESCE(device,'Unknown') as device
FROM groups;
```

4. What are the start and end dates of the experiment?

Start: 2023-01-25, End: 2023-02-06

Query:

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

or

SELECT dt
FROM activity
ORDER BY dt DESC;
```

5. How many total users were in the experiment?

48943

Query:

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

6. How many users were in the control and treatment groups?

A: 24343, B: 24600

Query:

```
SELECT
    "group",
    COUNT(uid) AS tot_user
FROM groups
GROUP BY "group";
```

7. What was the conversion rate of all users?

4.28

Query:

```
WITH user_activity AS (
    SELECT us.id,
    CASE
    WHEN SUM(ac.spent) > 0 THEN 1
    ELSE 0
    END AS converted
    FROM users AS us
    LEFT JOIN activity AS ac
    ON us.id = ac.uid
    GROUP BY us.id
)
SELECT
    COUNT(DISTINCT id) AS total_users,
    SUM(converted) AS total_converted,
    ROUND(SUM(converted) * 100.0 / COUNT(DISTINCT id), 2) AS conversion_rate
FROM user_activity;
```

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

A_CvR: 3.92, B_CvR: 4.63

Query:

```
WITH user_activity_group AS (  
    SELECT us.id,  
           g.group,  
           CASE  
               WHEN SUM(ac.spent) > 0 THEN 1  
               ELSE 0  
           END AS converted  
    FROM users AS us  
    LEFT JOIN activity AS ac ON us.id = ac.uid  
    JOIN groups AS g ON g.uid = us.id  
    GROUP BY us.id, g.group  
)  
SELECT "group",  
       COUNT(DISTINCT id) AS total_users,  
       SUM(converted) AS total_converted,  
       ROUND(SUM(converted) * 100.0 / COUNT(DISTINCT id), 2) AS conversion_rate  
FROM user_activity_group  
GROUP BY "group";
```

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

A: 3.37, B: 3.39

Query:

```
WITH user_activity_group AS (  
    SELECT us.id, ac.spent, g.group  
    FROM users AS us  
    LEFT JOIN activity AS ac ON us.id = ac.uid  
    JOIN groups AS g ON g.uid = us.id  
    GROUP BY us.id, ac.spent, g.group
```

```
)  
SELECT "group",  
ROUND(SUM(spent) /COUNT(DISTINCT id),2) AS avg_amt_spent_per_user  
FROM user_activity_group  
GROUP BY "group";
```

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

Including non-converting users helps ensure that your data is representative of the entire user base. Excluding non-converters might introduce bias and skew the results toward only those users who are more likely to convert. This can lead to misleading conclusions about the effectiveness of the change being tested.

2. Extracting data for Analysis

1. Data for Hypothesis testing

Query:

```
WITH user_activity_group AS (  
  
    SELECT us.id, us.country, us.gender, ac.device, g.group,  
  
    COALESCE(ROUND(SUM(spent),2),0) AS total_spent,  
  
    CASE  
  
    WHEN COALESCE(ROUND(SUM(ac.spent),2),0) > 0 THEN 1  
  
    ELSE 0  
  
    END AS converted  
  
    FROM users AS us  
  
    LEFT JOIN activity AS ac ON us.id = ac.uid  
  
    JOIN groups AS g ON g.uid = us.id  
  
    GROUP BY us.id, us.country, us.gender, ac.device, g.group  
  
    )  
  
    SELECT id, country, gender, device AS device_type, "group" as test_group, amount_spent,  
converted  
  
    FROM user_activity_group;
```

2. Data for Advanced Analysis

Query:

```
WITH user_activity_group AS (
```

```
SELECT us.id, us.country, us.gender, ac.device, g.group, g.join_dt AS join_date, ac.dt AS  
activity_date,
```

```
COALESCE(ROUND(SUM(spent),2),0) AS total_spent,
```

```
CASE
```

```
WHEN COALESCE(ROUND(SUM(ac.spent),2),0) > 0 THEN 1
```

```
ELSE 0
```

```
END AS converted
```

```
FROM users AS us
```

```
LEFT JOIN activity AS ac ON us.id = ac.uid
```

```
JOIN groups AS g ON g.uid = us.id
```

```
GROUP BY us.id, us.country, us.gender, ac.device, g.group, ac.dt, g.join_dt, ac.dt
```

```
)
```

```
SELECT id, country, gender, device AS device_type, total_spent, "group" as test_group,  
converted, join_date, activity_date
```

```
FROM user_activity_group;
```