1. Can a user show up more than once in the `activity` table? In this case yes, multiple users are reported as visiting the site twice ;

```
SELECT uid, COUNT(uid)
FROM activity
GROUP BY uid
HAVING COUNT(uid) > 1;
```

2. What type of join should we use to join the `users` table to the `activity` table? Inner join as that way we only get matching data ;

```
SELECT *

FROM activity

INNER JOIN users

ON users.id = activity.uid;
```

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

By using the ISNULL function where can scan to see if any values in a column are empty/null. By doing the following I was able to realise that there are not Null data points in the spent column on the activity table.

```
SELECT *

FROM activity

WHERE spent ISNULL;


The Function to fill in NULL values would be;


COALESCE(column, fill_value)
```

```
COALESCE(u.id, 0)
```

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

   To find the first date I did the following;

   ```
   SELECT dt

   FROM activity

   ORDER BY dt ASC;
   ```

   This showed me the smallest values for date, in this instance when the experiment started. 2023-01-25

   To find when it ends I used the same code, but I changed the Ascending aspect to Descending to find that largest value, in this case the most recent date. Which was 2023-02-06

   ```
   SELECT dt

   FROM activity

   ORDER BY dt DESC;
   ```

5. How many total users were in the experiment?

   To find this is used the 'COUNT()' function on the 'id' column from the users table. I also use distinct to make sure I don't have any duplicate id numbers, which can

skew my understanding of the data. This gives me the total amount of entries in the ‘id’ column, which is 48943.

```
SELECT DISTINCT COUNT(id)

FROM users;
```

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

   Because we are using PostgreSQL we have to put the word Group in quotation marks. This is because PostgreSQL reserves the word ‘Group’ for its function.

   To count the number of individuals in each group I have two queries;

   To count A I used the following;

```
SELECT "group", COUNT(*)

FROM groups

WHERE "group" = 'A'

GROUP BY "group";
```

The output of this informed me that 24343 users took part in group A.

To count B I used the following;

```
SELECT "group", COUNT(*)

FROM groups

WHERE "group" = 'B'

GROUP BY "group";
```

To which I found the amount of users in group B totaled 24600. This group is slightly larger than group A but applying the law of large numbers this shouldn't be too much of a problem.

7. What was the conversion rate of all users?

To calculate conversion rates you have to take the number of 'conversions' and divide that by the total number of users of the website that are tracked across the same time period.

First I got the total number of users and the total number of users who 'converted' i.e bought something.

```
SELECT

    COUNT(DISTINCT id) AS total_users,

    COUNT(DISTINCT activity.uid) AS total_converted,

    ROUND((COUNT(DISTINCT activity.uid)*1.0 / COUNT(DISTINCT id) * 1.0 *
100),2) || '%' AS conversion_rate

FROM users

LEFT JOIN activity ON users.id = activity.uid

LEFT JOIN groups on activity.uid = groups.uid;
```

This gives me the following value 4.28%.

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

```
WITH converted_users AS (
    SELECT g.group, COUNT(DISTINCT a.uid) AS activity_user_count
    FROM activity AS a
    JOIN groups AS g ON g.uid = a.uid
    GROUP BY g.group
),

all_users AS (
    SELECT "group", COUNT (DISTINCT uid) AS group_user_count
    FROM groups
    GROUP BY "group"
)

SELECT converted_users.group,
       converted_users.activity_user_count,
       all_users.group_user_count,
       (converted_users.activity_user_count * 1.0) /
(all_users.group_user_count * 1.0) * 100 AS conversion_ratio
FROM converted_users
JOIN all_users ON all_users.group = converted_users.group
```

Group A 3.92%

Group B 4.63%

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

```
SELECT
    "group",
    ROUND(AVG(SUM(spent::numeric)) OVER(PARTITION BY "group") / COUNT(DISTINCT g.uid),2) AS avg_spent
FROM groups g
FULL JOIN activity a USING(uid)
GROUP BY "group";
```

Hi Will. You are calculating the average by the number of rows in activity. But there are some users that have more than one purchase, so you're averages are wrong.

You have to include users.id and group by it. In that way you limit the possibility to count users more than once.

A: 3.37

B:3.38

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

Extract the analysis dataset

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
  COALESCE(u.id, 0) AS user_id,
  COALESCE(a.uid, 0) AS activity_uid,
  COALESCE(g.uid, 0) AS group_uid,
  COALESCE(u.country, '') AS country,
  COALESCE(u.gender, '') AS gender,
  COALESCE(g.device, '') AS device,
  COALESCE(g.group, '') AS group_name,
  COALESCE(SUM(a.spent), 0) AS total_spent
FROM users u
FULL OUTER JOIN activity a ON a.uid = u.id
FULL OUTER JOIN groups g ON g.uid = u.id
GROUP BY 1,2,3,4,5,6,7
order by 1 ASC;
```

48943 Rows