Globox SQL

```
/*
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 as user_country,
      u.gender as gender,
      g.device as ios_or_android,
      CASE WHEN g.group = 'A' THEN 'Control Group' ELSE 'Treatment Group' END AS
test_group,
      SUM(COALESCE(a.spent, 0)) AS sum_money_spent,
      CASE WHEN a.spent IS NOT NULL then 1 ELSE 0 END as conv_or_not
FROM users AS u
LEFT JOIN groups AS g
 ON g.uid = u.id
LEFT JOIN activity AS a
          ON a.uid = u.id
GROUP BY user_id, user_country, gender, ios_or_android, test_group, conv_or_not
ORDER BY sum_money_spent DESC
;

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

SELECT uid as user_id, COUNT(spent) AS many_purchase
FROM activity
GROUP BY user_id
ORDER BY many_purchase DESC
;
---Yes, one user can spent money more than once on the app

/*
Q-2
What type of join should we use to join the users table to the activity table?
*/
---one to many , left join

SELECT *
```

```sql
FROM users
LEFT JOIN activity
ON users.id = activity.uid
;

/*
Q-3
What SQL function can we use to fill in NULL values?
*/
--coalesce / case when

/*
Q-4
What are the start and end dates of the experiment?
*/

SELECT MIN(join_dt) AS start_date_g,
                     MAX(join_dt) AS end_date_g,
    MIN(dt) AS start_date_a,
                     MAX(dt) AS end_date_a
FROM groups
LEFT JOIN activity
ON activity.uid = groups.uid
;
---start date = 25/01/2023
---end date = 06/02/2023


/*
Q5
How many total users were in the experiment?
*/
SELECT count(DISTINCT (id))
FROM users
;
--- 48,943 uniuqe users


/*
Q6
How many users were in the control and treatment groups?
*/
--- The control group is group A - therefor there is a 24,434 users in the control group
--- The treatment group is group B - there is 24600 users in the treatment group
```

```
SELECT g.group, COUNT(DISTINCT(uid))
FROM groups AS g
GROUP BY g.group
;

/*
Q7
What was the conversion rate of all users?
*/
SELECT COUNT(DISTINCT(a.uid)) AS num_users_buy,
                        COUNT (DISTINCT(u.id)) AS num_all_users,
     (COUNT(DISTINCT(a.uid)) / CAST(COUNT (DISTINCT(u.id)) AS DECIMAL(10,5))) *100
AS conversion_rate
FROM users AS u
LEFT JOIN activity AS a
        ON a.uid = u.id
;
--- the conversion rate for all users together is 4.27%

/*
Q8
What is the user conversion rate for the control and treatment groups?
*/
SELECT g.group,
                        COUNT(DISTINCT(a.uid)) AS num_users_buy,
                        COUNT (DISTINCT(g.uid)) AS num_all_users,
     (COUNT(DISTINCT(a.uid)) / CAST(COUNT (DISTINCT(g.uid)) AS DECIMAL(10,5))) *100
AS conversion_rate
FROM groups AS g
LEFT JOIN activity AS a
        ON a.uid = g.uid
GROUP BY g.group
;
--- the conversion rates: group A (control group) = 3.92%
---                  group B (treatment group) = 4.63%

/*
Q9
What is the average amount spent per user for the control and treatment groups,
including users who did not convert?
*/

SELECT g.group,
    ROUND(AVG(COALESCE(a.spent, 0)),2) AS avg_spent
FROM groups AS g
LEFT JOIN activity AS a
        ON a.uid = g.uid
GROUP BY g.group
;

--- if we include the customers that didnt bought then:
--- control group (group A) - 3.366
--- treatment group (group B) - 3.379

/*
Q10
Why does it matter to include users who did not convert when calculating the
average amount spent per user?
*/
--- Because users that not spent are necesary to our experiment, we consider them as spent 0.
--- The average should calculating all users and not just the users that bought something.
--- we can see that if we calculating the avg without those users -
--- the numbers are different and extremly higer
```