SQL Queries:

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

—-

```
WITH cte_conversion AS
 (SELECT g.uid, min(dt) AS dt, SUM(COALESCE(spent, 0)) AS spend,
 CASE WHEN SUM(COALESCE(spent, 0)) > 0 THEN 'converted'
  ELSE 'not_converted' END AS conversion
 FROM groups g
 LEFT JOIN activity a
 ON g.uid = a.uid
 GROUP BY 1)

SELECT c.uid, join_dt, dt, COALESCE (u.country, 'Unknown') AS country,
COALESCE (u.gender, 'Unknown') AS gender,
COALESCE (g.device, 'Unknown') AS device, g.group, c.spend, c.conversion
FROM cte_conversion c
LEFT JOIN groups g
ON c.uid = g.uid
LEFT JOIN users u
ON c.uid = u.id
```

—-

```
WITH cte_conversion AS (
    SELECT
        g.uid,
        MIN(dt) AS dt,
        SUM(COALESCE(spent, 0)) AS spend,
        CASE WHEN SUM(COALESCE(spent, 0)) > 0 THEN 'converted' ELSE 'not_converted'
END AS conversion
    FROM
        groups g
    LEFT JOIN
        activity a ON g.uid = a.uid
    GROUP BY
        1
)

SELECT
```

```
    c.uid,
    join_dt,
    dt,
    COALESCE(u.country, 'Unknown') AS country,
    COALESCE(u.gender, 'Unknown') AS gender,
    COALESCE(g.device, 'Unknown') AS device,
    g.group,
    c.spend,
    c.conversion,
    -- Calculate conversion rate for Group A
    CASE WHEN COUNT(CASE WHEN g.group = 'A' THEN 1 END) > 0
         THEN COUNT(CASE WHEN c.conversion = 'converted' AND g.group = 'A' THEN 1
END) * 100.0 / COUNT(CASE WHEN g.group = 'A' THEN 1 END)
         ELSE 0 END AS conversion_rate_A,
    -- Calculate conversion rate for Group B
    CASE WHEN COUNT(CASE WHEN g.group = 'B' THEN 1 END) > 0
         THEN COUNT(CASE WHEN c.conversion = 'converted' AND g.group = 'B' THEN 1
END) * 100.0 / COUNT(CASE WHEN g.group = 'B' THEN 1 END)
         ELSE 0 END AS conversion_rate_B
FROM
    cte_conversion c
LEFT JOIN
    groups g ON c.uid = g.uid
LEFT JOIN
    users u ON c.uid = u.id
GROUP BY
    1, 2, 3, 4, 5, 6, 7, 8, 9;



—-

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

—--

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, total_spent,
converted
FROM user_activity_group;
```

—---

```
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";
```

—---

```
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";
```

—---

```
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;
```

—---

Calculating the user conversion rate for the control and treatment groups:

```sql
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";
```

Data for Hypothesis testing:

```sql
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, total_spent,
converted
FROM user_activity_group;
```

Calculating the average amount spent per user for the control andtreatment groups, including users who did not convert query:

```sql
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";
- **Statistical Tests Used:** For Conversion Rate used Two-sample z-test for a difference in proportions, for Average Amount used Two-sample t-test for a difference in means.
- **Results Overview:** Average Amount- The p-value, standing at 0.9438, indicates a lack of statistical significance, thereby leading to the retention of the null hypothesis. Consequently, there exists no discernible variance in the mean expenditure between the two groups under examination. Conversion Rate- The obtained p-value of 0.0001 attains statistical significance, leading to the rejection of the null hypothesis. This implies a discernible distinction in the conversion rates between the two groups.
```

Preparing for advaced tasks:

```sql
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;
```
```