SQL Queries

Query 1 for Hypothesis Testing:

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

Query 2 for Novelty Effect:

```
SELECT
    sub.groups,
    sub.date,
    sub.average_amount_spent,
    sub.paid_users,
    sub.total_spent,
    sub.total_users,
    CASE WHEN sub.total_users > 0 THEN sub.paid_users * 1.0 / sub.total_users ELSE 0 END AS conversion_rate,
    sub.total_spent / sub.paid_users AS converted_average_spent
FROM (
    SELECT
        g.group AS groups,
        g.join_dt AS date,
        SUM(a.spent) / COUNT(DISTINCT u.id) AS average_amount_spent,
        COUNT(DISTINCT CASE WHEN a.spent > 0 THEN a.uid END) AS paid_users,
        SUM(COALESCE(a.spent, 0)) AS total_spent,
        COUNT(DISTINCT u.id) AS total_users
    FROM users as u
    INNER JOIN "groups" g ON g.uid = u.id
    LEFT JOIN activity a ON u.id = a.uid
    GROUP BY g.group, g.join_dt
) AS sub
ORDER BY sub.date;
```

Query 3 for Novelty Effect:

```
SELECT
    sub.date_converted,
    CASE
        WHEN (sub.date_converted - sub.date_registered) = 0 THEN 'Same day'
        WHEN (sub.date_converted - sub.date_registered) = 1 THEN '1 day later'
        ELSE (sub.date_converted - sub.date_registered) | | ' days'
    END AS days_since_conversion,
    sub.date_registered,
    sub.groups,
    sub.user_id,
    a.spent AS purchase_amount
FROM (
    SELECT
        g.group AS groups,
        g.join_dt AS date_registered,
        a.uid AS user_id,
        a.dt AS date_converted
    FROM
        users as u
        INNER JOIN "groups" g ON g.uid = u.id
        LEFT JOIN activity a ON u.id = a.uid
    WHERE
        a.spent > 0
) AS sub
LEFT JOIN activity a ON sub.user_id = a.uid AND sub.date_converted = a.dt
ORDER BY
    sub.user_id DESC, a.dt;
```

Query 4 for Power Analysis:

```
SELECT
   u.id,
   g.join_dt,
   COALESCE(SUM(a.spent), 0) AS total_spent
FROM
   users u
   LEFT JOIN activity a ON u.id = a.uid
   INNER JOIN "groups" g ON u.id = g.uid
GROUP BY
   u.id,
   g.join_dt;
```

Query 5 for Power Analysis(Cumulative Users):

```
SELECT
    DISTINCT(join_dt) AS "date",
    COUNT(uid) OVER (ORDER BY join_dt) AS cumulative_users
FROM
    "groups"
ORDER BY
    "date";
```