**1.Conversion Rate Of All Users:**

with S1 AS (SELECT users.id, SUM(activity.spent),

CASE WHEN SUM(activity.spent)>0 THEN 1

ELSE 0 END AS converted

FROM users

LEFT JOIN activity

ON users.id=activity.uid

GROUP BY users.id)

SELECT AVG(converted) AS conversion\_rate

FROM S1;

**2.Conversion Rate Of A and B:**

with S1 AS (SELECT users.id,SUM(activity.spent), groups.group As Test\_Group,

CASE WHEN SUM(activity.spent)>0 THEN 1

ELSE 0 END AS converted

FROM users

LEFT JOIN activity

ON users.id=activity.uid

LEFT JOIN groups

ON users.id=groups.uid

GROUP BY users.id, groups.group)

SELECT S1.Test\_Group,AVG(converted) AS conversion\_rate

FROM S1

GROUP BY s1.Test\_Group

ORDER BY s1.Test\_Group;

**3.Extracting The Analysis Dataset:**

SELECT users.id,users.country,users.gender,groups.device,groups.group,

CASE WHEN SUM(activity.spent) > 0 THEN 1

ELSE 0 END AS converted,SUM(COALESCE(activity.spent,0))

FROM users

LEFT JOIN groups

ON users.id=groups.uid

LEFT JOIN activity

ON users.id=activity.uid

GROUP BY users.id,groups.device,groups.group;

**4.Minimum And Maximum Join Dates:**

SELECT MIN(join\_dt),MAX(join\_dt)

FROM groups;

**5.Novelty Effect:**

with S1 AS (SELECT groups.group, activity.dt AS date,SUM(CASE WHEN (activity.spent) > 0 THEN 1

ELSE 0 END) AS converted\_users,avg(COALESCE(activity.spent,0)) AS spent

FROM activity

Left join

Groups

on activity.uid = groups.uid

GROUP BY groups.group, activity.dt)

SELECT Count(DISTINCT(groups.uid)) AS total\_users, S1.group, date, converted\_users, spent

FROM

groups

LEFT JOIN

S1

on S1.date = groups.join\_dt and S1.group = groups.group

group by S1.date, S1.group,S1.converted\_users, S1.spent;

**6.Total Users In Experiment:**

SELECT COUNT(DISTINCT id)

FROM users;

**7.Users In Control And Treatment Group:**

SELECT COUNT(uid),"group"

FROM groups

GROUP BY "group";

**8.Left Join Users And Activity Table:**

SELECT \* FROM users

LEFT JOIN activity

ON users.id=activity.uid;

**9.User ShowUp More Than Once:**

SELECT uid,count(uid)

FROM activity

GROUP BY uid

HAVING COUNT(uid)>1;