Case Study 1:

Q1. Calculate the Total order quantity, GMV and Net sales, where GMV is the summation of all the order values, and Net sale is the values after the discount.

Solution:

```
WITH
  table_order1 AS (
  SELECT
    a.OrderId AS OrderId,
    a.ItemPrice AS ItemPrice,
    a.QuantityOrdered AS QuantityOrdered,
    a.PromotionDiscount AS PromotionDiscount,
    b.Purchasedate AS Purchasedate,
    b.OrderStatus AS OrderStatus,
    SPLIT(PromotionDiscount,'"')[OFFSET(7)] AS Discount
  FROM
    `statfinity_sql_case.Order1` a
  LEFT JOIN
    `statfinity_sql_case.Order2` b
  ON
    a.OrderId=b.OrderId)

SELECT
  SUM(table_order1.QuantityOrdered) AS Total_order_quantity,
  SUM(table_order1.ItemPrice) AS GMV,
  SUM((1-(CAST(table_order1.Discount AS float64)/100))*table_order1.ItemPrice) AS
Net_Sales
FROM
  table_order1
WHERE
  OrderStatus="Shipped"
  AND QuantityOrdered>0
```

Q2. Calculate Total order quantity, GMV and Net sales for the month of October 2022 where the order status is not equal to "Canceled".

Solution:

```
WITH
  table_order1 AS (
  SELECT
    a.OrderId AS OrderId,
    a.ItemPrice AS ItemPrice,
    a.QuantityOrdered AS QuantityOrdered,
    a.PromotionDiscount AS PromotionDiscount,
    b.Purchasedate AS Purchasedate,
    b.OrderStatus AS OrderStatus,
    SPLIT(PromotionDiscount,'"')[OFFSET(7)] AS Discount
  FROM
    `statfinity_sql_case.Order1` a
  LEFT JOIN
    `statfinity_sql_case.Order2` b
  ON
    a.OrderId=b.OrderId)

SELECT
  SUM(table_order1.QuantityOrdered) AS Total_order_quantity,
  SUM(table_order1.ItemPrice) AS GMV,
  SUM((1-(CAST(table_order1.Discount AS float64)/100))*table_order1.ItemPrice) AS
Net_Sales
FROM
  table_order1
WHERE
  OrderStatus!="Canceled"
  AND QuantityOrdered>0
  AND EXTRACT(month
  FROM
    table_order1.Purchasedate)=8
  AND EXTRACT(year
  FROM
    table_order1.Purchasedate)=2022
```

Case Study 2:

Q1. Calculate the number of users who installed the app date wise.

Solution:

```
SELECT
  user_first_seen_date AS installation_date,
  COUNT(DISTINCT(user_id)) AS Number_of_users
FROM
  `statfinity_sql_case.User1`
GROUP BY
  user_first_seen_date
ORDER BY
  user_first_seen_date
```

Q2. Average number of days users use the app.

Solution:

```
WITH
  table_user1 AS (
  SELECT
    DISTINCT(a.user_id) AS user_id,
    COUNT(DISTINCT(b.Date)) AS days
  FROM
    `statfinity_sql_case.User1` a
  LEFT JOIN
    `statfinity_sql_case.User2` b
  ON
    a.user_id=b.user_id
  GROUP BY
    a.user_id)

SELECT
  AVG(table_user1.days) AS Average_days
FROM
  table_user1
```

Q3. Retention percentage for Day-1, Day-3 and Day-7.

Solution:

```
WITH
  table_user1 AS (
  SELECT
    DISTINCT(a.user_id) AS user_id,
    COUNT(DISTINCT(b.Date)) AS days
  FROM
    `statfinity_sql_case.User1` a
  LEFT JOIN
    `statfinity_sql_case.User2` b
  ON
    a.user_id=b.user_id
  GROUP BY
    a.user_id)

SELECT
  ROUND(SUM(CASE
        WHEN table_user1.days>=1 THEN 1
      ELSE
      0
    END
      )/COUNT(table_user1.user_id)*100,2) AS Day_1_retention,
  ROUND(SUM(CASE
        WHEN table_user1.days>=3 THEN 1
      ELSE
      0
    END
      )/COUNT(table_user1.user_id)*100,2) AS Day_3_retention,
  ROUND(SUM(CASE
        WHEN table_user1.days>=7 THEN 1
      ELSE
      0
    END
      )/COUNT(table_user1.user_id)*100,2) AS Day_7_retention
FROM
  table_user1
```