```
-- Loading Dataset into Database

LOAD DATA  INFILE "C:/Program Files/apnaa_1.csv"
INTO TABLE apnaaa_1
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n' IGNORE 1 LINES;




-- Query to view all data from the table apnaaa_1

SELECT * FROM apnaaa_1;

-- Adding a column for selling price and calculating it based on
placed_gmv and quantity

ALTER TABLE apnaaa_1
ADD COLUMN selling_price DOUBLE;

UPDATE apnaaa_1
SET selling_price = placed_gmv / quantity;



-- Total placed GMV and quantity sold across the dataset

SELECT sum(placed_gmv), sum(quantity)
FROM apnaaa_1;



-- Monthly total sales to identify sales trends over time

SELECT
    DATE(order_date) AS order_date,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    DATE(order_date)
ORDER BY
    DATE(order_date);
```

```sql
-- Top 10 SKUs by total sales to identify best-performing products

SELECT
    sku_id,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    sku_id
ORDER BY total_sales
    DESC
LIMIT 10;




-- User spending analysis to understand customer value and purchasing
behavior

SELECT
    user_id,
    COUNT(order_id) AS total_orders,
    ROUND(SUM(placed_gmv), 2) AS total_spent
FROM
    apnaaa_1
GROUP BY
    user_id
ORDER BY total_orders
    DESC;




-- Total quantity sold and sales per warehouse to evaluate warehouse
performance

SELECT
    warehouse_name,
    SUM(quantity) AS total_quantity,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    warehouse_name
ORDER BY total_sales
    DESC;
```

```sql
-- Average selling price and average order value across all transactions

SELECT
    AVG(selling_price) AS average_selling_price,
    AVG(placed_gmv) AS average_order_value
FROM
    apnaaa_1;




-- Daily sales analysis per SKU to track performance over time

SELECT
    DATE(order_date) AS order_date,
    sku_id,
    SUM(placed_gmv) AS total_sales
FROM
    apnaaa_1
GROUP BY
    DATE(order_date), sku_id
ORDER BY
    DATE(order_date), total_sales DESC;




-- Monthly sales analysis to track revenue trends by month

SELECT
    DATE_FORMAT(order_date, '%Y-%m') AS month,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    month
ORDER BY
    month;
```

```sql
-- Warehouse performance analysis based on quantity and sales figures

SELECT
    warehouse_name,
    SUM(quantity) AS total_quantity,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    warehouse_name
ORDER BY warehouse_name, total_sales
    DESC;




-- Monthly sales and quantity analysis per warehouse to identify seasonal
trends

SELECT
    warehouse_name,
    DATE_FORMAT(order_date, '%Y-%m') AS order_month,
    SUM(quantity) AS total_quantity,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY warehouse_name,
    order_month
ORDER BY warehouse_name,
    order_month;



-- Customer segmentation analysis based on total GMV spent by users

SELECT
    SUM(CASE WHEN total_gmv >= 100 THEN 1 ELSE 0 END) AS
    low_value_count,
    SUM(CASE WHEN total_gmv BETWEEN 100 AND 500 THEN 1 ELSE 0 END) AS
medium_value_count,
    SUM(CASE WHEN total_gmv > 500 THEN 1 ELSE 0 END) AS high_value_count
FROM (
    SELECT
        user_id,
        SUM(placed_gmv) AS total_gmv
    FROM apnaaa_1
    GROUP BY user_id
) AS sub;
```

```sql
-- Yearly and monthly sales trends analysis to assess revenue growth over time

SELECT
    YEAR(order_date) AS order_year,
    MONTH(order_date) AS order_month,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    YEAR(order_date), MONTH(order_date)
ORDER BY order_year,
    order_month;



-- Total quantity, sales, and order count analysis by day of the week for comprehensive daily performance evaluation

SELECT
    DAYOFWEEK(order_date) AS day_of_week,
    COUNT(order_id) AS total_orders
    SUM(quantity) AS total_quantity,
    ROUND(SUM(placed_gmv), 2) AS total_sales
FROM
    apnaaa_1
GROUP BY
    day_of_week
ORDER BY
    day_of_week;
```