SQL Task – 3

GROUP BY :

1. Find the total sold_quantity of products for each category.

```
SELECT category, SUM(sold_quantity) as total_sold
FROM products
GROUP BY category;
```

2. Find the average price of products for each brand**.**

```
SELECT brand, AVG(price) as avg_price
FROM products
GROUP BY brand;
```

3. Count how many products are available in each category.

```
SELECT category, COUNT(*) as product_count
FROM products
GROUP BY category;
```

4. Find the maximum rating of products for each category.

```
SELECT category, MAX(rating) as max_rating
FROM products
GROUP BY category;
```

5. Find the minimum price of products for each brand.

```
SELECT brand, MIN(price) as min_price
FROM products
GROUP BY brand;
```

HAVING :

1. Find all categories where the total sold_quantity is greater than 1000.

```
SELECT category, SUM(sold_quantity) as total_sold
FROM products
GROUP BY category
```

```
HAVING SUM(sold_quantity) > 1000;
```

2. Find brands where the average price of products is more than 10000.

```
SELECT brand, AVG(price) as avg_price
FROM products
GROUP BY brand
HAVING AVG(price) > 10000;
```

3. Find categories where the maximum rating is greater than 4.7.

```
SELECT category, MAX(rating) as max_rating
FROM products
GROUP BY category
HAVING MAX(rating) > 4.7
```

4. Find brands where the count of products is more than 2.

```
SELECT brand, COUNT(*) as product_count
FROM products
GROUP BY brand
HAVING COUNT(*) > 2;
```

5. Find categories where the sum of stock is less than 200.

```
SELECT category, SUM(stock) as total_stock
FROM products
GROUP BY category
HAVING SUM(stock) < 200;
```

ORDER BY :

1. List all products ordered by price ascending.

```
SELECT * FROM products
ORDER BY price ASC;
```

2. Show all products ordered by sold_quantity descending.

```
SELECT * FROM products
ORDER BY sold_quantity DESC;
```

3. Display products ordered by rating descending and then price ascending.

```
SELECT * FROM products
ORDER BY rating DESC, price ASC;
```

4. List all products ordered by category ascending and sold_quantity descending.

```
SELECT * FROM products
ORDER BY category ASC, sold_quantity DESC;
```

5. Show all products ordered by brand alphabetically.

```
SELECT * FROM products
ORDER BY brand ASC;
```

LIMIT :

1. Show the first 5 products in the table.

```
SELECT * FROM products
LIMIT 5;
```

2. Show the top 3 highest-priced products.

```
SELECT * FROM products
ORDER BY price DESC
LIMIT 3;
```

3. Show the first 5 products when ordered by rating descending.

```
SELECT * FROM products
ORDER BY rating DESC
LIMIT 5;
```

4. Show 2 products after skipping the first 5 rows.

```
SELECT * FROM products
LIMIT 2 OFFSET 5;
```

5. Display the top 10 products with highest sold_quantity.

```
SELECT * FROM products
ORDER BY sold_quantity DESC
LIMIT 10;
```

ASC / DESC

1. Show all products sorted by price ascending.

   ```
   SELECT * FROM products
   ORDER BY price ASC;
   ```

2. Show all products sorted by rating descending.

   ```
   SELECT * FROM products
   ORDER BY rating DESC;
   ```

3. Display products sorted by sold_quantity descending, then price ascending.

   ```
   SELECT * FROM products
   ORDER BY sold_quantity DESC, price ASC;
   ```

4. List products sorted by category ascending, then sold_quantity descending.

   ```
   SELECT * FROM products
   ORDER BY category ASC, sold_quantity DESC;
   ```

5. Show all products sorted by brand descending.

   ```
   SELECT * FROM products
   ORDER BY brand DESC;
   ```