

-- 26/09/2025

Task 4: Aggregate Functions and Grouping

USE ecommerce_db;

-- Total number of users

```
SELECT COUNT(*) AS total_users
FROM Users;
```

```
total_users
      12
```

-- Total stock available across all products

```
SELECT SUM(stock) AS total_stock
FROM Products;
```

```
total_stock
      354
```

-- Average product price

```
SELECT AVG(price) AS avg_price
FROM Products;
```

```
avg_price
    9899.3
```

-- Cheapest and costliest product overall

```
SELECT MIN(price) AS cheapest_product, MAX(price) AS costliest_product
FROM Products;
```

```
cheapest_product  costliest_product
              299              55000
```

-- Total revenue from all payments

```
SELECT SUM(amount) AS total_revenue
FROM Payments;
```

```
total_revenue
    102891
```

-- Revenue per payment method with min, max, avg

```
SELECT method,
       SUM(amount) AS total_revenue,
       MIN(amount) AS min_payment,
       MAX(amount) AS max_payment,
       AVG(amount) AS avg_payment
FROM Payments
GROUP BY method;
```

method	total_revenue	min_payment	max_payment	avg_payment
Net Banking	56000	56000	56000	56000
Debit Card	6998	6998	6998	6998
Cash on Delivery	998	998	998	998
UPI	11297	11297	11297	11297
Credit Card	27598	27598	27598	27598

-- Orders summary

```
SELECT COUNT(order_id) AS total_orders,  
       MIN(order_date) AS first_order,  
       MAX(order_date) AS last_order  
FROM Orders;
```

total_orders	first_order	last_order
5	24-09-2025 16:50	24-09-2025 16:50

-- Number of products in each category

```
SELECT c.category_name AS category_name, COUNT(p.product_id) AS product_count  
FROM Categories c  
JOIN Products p ON c.category_id = p.category_id  
GROUP BY c.category_name;
```

category_name	product_count
Electronics	3
Fashion & Clothing	2
Books	1
Home Appliances	1
Sports	1
Toys	1
Beauty & Health	1

-- Total stock per category

```
SELECT c.category_name AS category_name, SUM(p.stock) AS total_stock  
FROM Categories c  
JOIN Products p ON c.category_id = p.category_id  
GROUP BY c.category_name;
```

category_name	total_stock
Electronics	74
Fashion & Clothing	110
Books	50
Home Appliances	15
Sports	20
Toys	35
Beauty & Health	50

-- Average price per category

```
SELECT c.category_name AS category_name, AVG(p.price) AS avg_price  
FROM Categories c  
JOIN Products p ON c.category_id = p.category_id  
GROUP BY c.category_name;
```

category_name	avg_price
Electronics	27333
Fashion & Clothing	899
Books	299
Home Appliances	8500
Sports	4999
Toys	899
Beauty & Health	499

-- Min and Max price per category

```
SELECT c.category_name AS category_name, MIN(p.price) AS min_price, MAX(p.price) AS max_price
FROM Categories c
JOIN Products p ON c.category_id = p.category_id
GROUP BY c.category_name;
```

category_name	min_price	max_price
Electronics	1999	55000
Fashion & Clothing	499	1299
Books	299	299
Home Appliances	8500	8500
Sports	4999	4999
Toys	899	899
Beauty & Health	499	499

-- Customer-wise sales summary (uses all aggregates)

```
SELECT u.name,
       COUNT(o.order_id) AS total_orders,
       SUM(p.amount) AS total_spent,
       MIN(p.amount) AS smallest_payment,
       MAX(p.amount) AS largest_payment,
       AVG(p.amount) AS avg_payment
FROM Users u
JOIN Orders o ON u.user_id = o.user_id
JOIN Payments p ON o.order_id = p.order_id
GROUP BY u.name
ORDER BY total_spent DESC;
```

name	total_orders	total_spent	smallest_payment	largest_payment	avg_payment
Aditya Patil	1	56000	56000	56000	56000
Tushar Patil	1	27598	27598	27598	27598
Ganesh Kadam	1	11297	11297	11297	11297
Bhushan Chavan	1	6998	6998	6998	6998
Dinesh Jadhav	1	998	998	998	998

-- Categories with more than 2 products

```
SELECT c.category_name, COUNT(p.product_id) AS product_count
FROM Categories c
JOIN Products p ON c.category_id = p.category_id
GROUP BY c.category_name
HAVING COUNT(p.product_id) > 2;
```

category_name	product_count
Electronics	3

-- Payment methods with total revenue above 50,000

```
SELECT method, SUM(amount) AS total_revenue
FROM Payments
GROUP BY method
HAVING SUM(amount) > 50000;
```

method	total_revenue
Net Banking	56000

```
-- Categories with average price greater than 1000
SELECT c.category_name, AVG(p.price) AS avg_price
FROM Categories c
JOIN Products p ON c.category_id = p.category_id
GROUP BY c.category_name
HAVING AVG(p.price) > 1000;
```

category_name	avg_price
Electronics	27333
Home Appliances	8500
Sports	4999

```
-- Customers who spent at least 10,000
SELECT u.name, SUM(p.amount) AS total_spent
FROM Users u
JOIN Orders o ON u.user_id = o.user_id
JOIN Payments p ON o.order_id = p.order_id
GROUP BY u.name
HAVING SUM(p.amount) >= 10000;
```

name	total_spent
Aditya Patil	56000
Ganesh Kadam	11297
Tushar Patil	27598