SQL Project

Customer Purchase Behaviour Analysis for E-Commerce

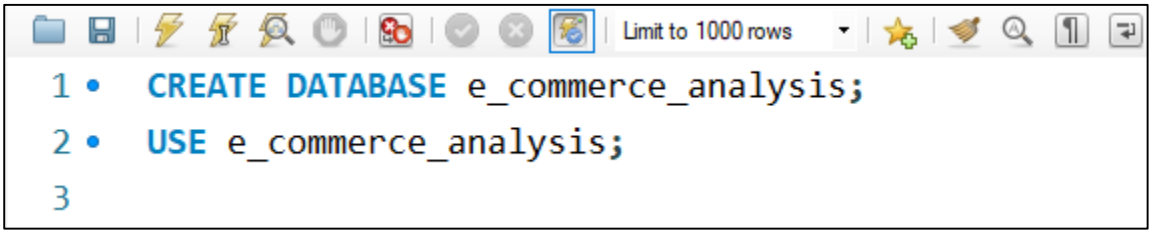
Problem Statement

Objective: To analyse customer purchase behaviour on an e-commerce platform to identify trends, patterns, and insights that can help improve marketing strategies, optimize inventory, and enhance customer experience.

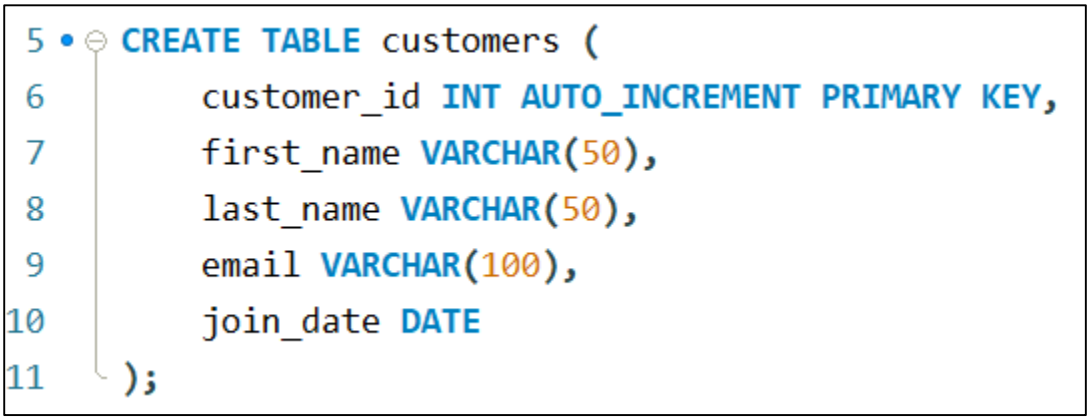
Problem Statement: The e-commerce platform is experiencing fluctuating sales and low customer retention rates. The goal is to analyse customer purchase data to understand buying patterns, identify high-value customers, determine the impact of marketing campaigns, and provide actionable recommendations to boost sales and customer loyalty.

Start

Create Database



Creating Tables “Customers”



Creating Tables “Products”

```sql
CREATE TABLE products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    product_name VARCHAR(100),
    category VARCHAR(50),
    price DECIMAL(10, 2)
);
```

Creating Tables “Orders”

```sql
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10, 2),
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
```

Creating Tables “Order_Items”

```sql
CREATE TABLE order_items (
    order_item_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    price DECIMAL(10, 2),
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

Inserting data

```
42 •  INSERT INTO customers (first_name, last_name, email, join_date) VALUES
43    ('John', 'Doe', 'johndoe@gmail.com', '2023-01-10'),
44    ('Jane', 'Smith', 'janesmith@gmail.com', '2023-02-15'),
45    ('Alice', 'Johnson', 'alicejohnson@gmail.com', '2023-03-20'),
46    ('Bob', 'Brown', 'bobbrown@gmail.com', '2023-04-25'),
47    ('Carol', 'Williams', 'carolwilliams@gmail.com', '2023-05-30'),
48    ('David', 'Wilson', 'davidwilson@gmail.com', '2023-06-10'),
49    ('Emma', 'Moore', 'emmamoore@gmail.com', '2023-07-15'),
50    ('Frank', 'Taylor', 'franktaylor@gmail.com', '2023-08-20'),
51    ('Grace', 'Anderson', 'graceanderson@gmail.com', '2023-09-25'),
52    ('Henry', 'Thomas', 'henrythomas@gmail.com', '2023-10-30'),
53    ('Ivy', 'Jackson', 'ivyjackson@gmail.com', '2023-11-05'),
54    ('Jack', 'White', 'jackwhite@gmail.com', '2023-12-10'),
55    ('Kathy', 'Harris', 'kathyharris@gmail.com', '2024-01-15'),
56    ('Liam', 'Martin', 'liammartin@gmail.com', '2024-02-20'),
57    ('Mia', 'Thompson', 'miathompson@gmail.com', '2024-03-25'),
58    ('Noah', 'Garcia', 'noahgarcia@gmail.com', '2024-04-30'),
59    ('Olivia', 'Martinez', 'oliviamartinez@gmail.com', '2024-05-15'),
60    ('Paul', 'Robinson', 'paulrobinson@gmail.com', '2024-06-20'),
61    ('Quinn', 'Clark', 'quinnclark@gmail.com', '2024-07-25'),
62    ('Rita', 'Rodriguez', 'ritarodriguez@gmail.com', '2024-08-30');
```

```
64 •  select * from customers;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

customer_id	first_name	last_name	email	join_date
1	John	Doe	johndoe@gmail.com	2023-01-10
2	Jane	Smith	janesmith@gmail.com	2023-02-15
3	Alice	Johnson	alicejohnson@gmail.com	2023-03-20
4	Bob	Brown	bobbrown@gmail.com	2023-04-25
5	Carol	Williams	carolwilliams@gmail.com	2023-05-30
6	David	Wilson	davidwilson@gmail.com	2023-06-10
7	Emma	Moore	emmamoore@gmail.com	2023-07-15
8	Frank	Taylor	franktaylor@gmail.com	2023-08-20
9	Grace	Anderson	graceanderson@gmail.com	2023-09-25
10	Henry	Thomas	henrythomas@gmail.com	2023-10-30
11	Ivy	Jackson	ivyjackson@gmail.com	2023-11-05
12	Jack	White	jackwhite@gmail.com	2023-12-10
13	Kathy	Harris	kathyharris@gmail.com	2024-01-15
14	Liam	Martin	liammartin@gmail.com	2024-02-20
15	Mia	Thompson	miathompson@gmail.com	2024-03-25
16	Noah	Garcia	noahgarcia@gmail.com	2024-04-30
17	Olivia	Martinez	oliviamartinez@gmail.com	2024-05-15
18	Paul	Robinson	paulrobinson@gmail.com	2024-06-20
19	Quinn	Clark	quinnclark@gmail.com	2024-07-25
20	Rita	Rodriguez	ritarodriguez@gmail.com	2024-08-30
NULL	NULL	NULL	NULL	NULL

Inserting data

```sql
INSERT INTO products (product_name, category, price) VALUES
('Smartphone', 'Electronics', 699.99),
('Laptop', 'Electronics', 1199.99),
('Headphones', 'Accessories', 149.99),
('Keyboard', 'Accessories', 79.99),
('Office Chair', 'Furniture', 249.99),
('Monitor', 'Electronics', 299.99),
('Mouse', 'Accessories', 49.99),
('Desk Lamp', 'Furniture', 89.99),
('Webcam', 'Electronics', 89.99),
('Smartwatch', 'Electronics', 199.99),
('Printer', 'Electronics', 129.99),
('Tablet', 'Electronics', 349.99),
('Bluetooth Speaker', 'Accessories', 79.99),
('External Hard Drive', 'Electronics', 139.99),
('USB Flash Drive', 'Accessories', 29.99),
('Gaming Mouse', 'Accessories', 89.99),
('Ergonomic Chair', 'Furniture', 399.99),
('Desk Organizer', 'Furniture', 39.99),
('Wireless Charger', 'Accessories', 59.99),
('Portable SSD', 'Electronics', 199.99);
```

```sql
select * from products;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

product_id	product_name	category	price
1	Smartphone	Electronics	699.99
2	Laptop	Electronics	1199.99
3	Headphones	Accessories	149.99
4	Keyboard	Accessories	79.99
5	Office Chair	Furniture	249.99
6	Monitor	Electronics	299.99
7	Mouse	Accessories	49.99
8	Desk Lamp	Furniture	89.99
9	Webcam	Electronics	89.99
10	Smartwatch	Electronics	199.99
11	Printer	Electronics	129.99
12	Tablet	Electronics	349.99
13	Bluetooth Sp...	Accessories	79.99
14	External Hard...	Electronics	139.99
15	USB Flash Drive	Accessories	29.99
16	Gaming Mouse	Accessories	89.99
17	Ergonomic Chair	Furniture	399.99
18	Desk Organizer	Furniture	39.99
19	Wireless Char...	Accessories	59.99
20	Portable SSD	Electronics	199.99
NULL	NULL	NULL	NULL

Inserting data

```
92 •  INSERT INTO orders (customer_id, order_date, total_amount) VALUES
93    (1, '2024-01-15', 799.98),
94    (2, '2024-02-20', 1299.98),
95    (3, '2024-03-22', 329.98),
96    (4, '2024-04-30', 79.99),
97    (5, '2024-05-31', 249.99),
98    (6, '2024-06-10', 299.98),
99    (7, '2024-07-15', 169.98),
100   (8, '2024-08-20', 359.98),
101   (9, '2024-09-25', 119.98),
102   (10, '2024-10-30', 149.99),
103   (11, '2024-11-05', 249.99),
104   (12, '2024-12-10', 449.98),
105   (13, '2024-01-15', 79.99),
106   (14, '2024-02-20', 349.99),
107   (15, '2024-03-25', 89.99),
108   (16, '2024-04-30', 119.98),
109   (17, '2024-05-15', 79.99),
110   (18, '2024-06-20', 299.99),
111   (19, '2024-07-25', 89.99),
112   (20, '2024-08-30', 149.99);
```

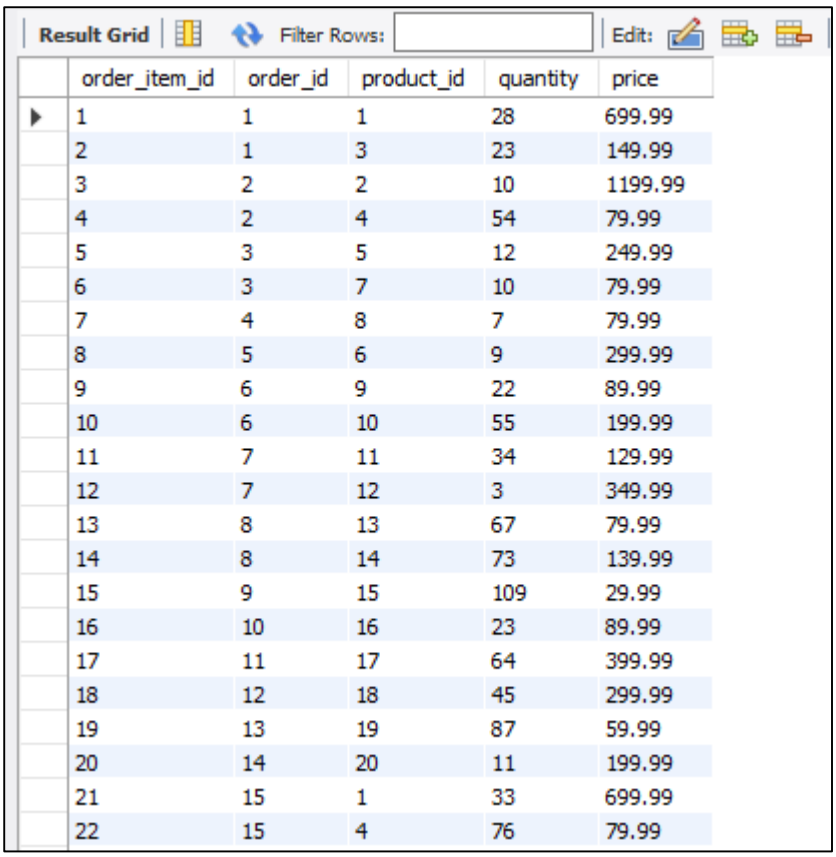
```
115 •  select * from orders;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

order_id	customer_id	order_date	total_amount
1	1	2024-01-15	799.98
2	2	2024-02-20	1299.98
3	3	2024-03-22	329.98
4	4	2024-04-30	79.99
5	5	2024-05-31	249.99
6	6	2024-06-10	299.98
7	7	2024-07-15	169.98
8	8	2024-08-20	359.98
9	9	2024-09-25	119.98
10	10	2024-10-30	149.99
11	11	2024-11-05	249.99
12	12	2024-12-10	449.98
13	13	2024-01-15	79.99
14	14	2024-02-20	349.99
15	15	2024-03-25	89.99
16	16	2024-04-30	119.98
17	17	2024-05-15	79.99
18	18	2024-06-20	299.99
19	19	2024-07-25	89.99
20	20	2024-08-30	149.99
NULL	NULL	NULL	NULL

Inserting data

```
118 •  INSERT INTO order_items (order_id, product_id, quantity, price) VALUES
119     (1, 1, 28, 699.99),
120     (1, 3, 23, 149.99),
121     (2, 2, 10, 1199.99),
122     (2, 4, 54, 79.99),
123     (3, 5, 12, 249.99),
124     (3, 7, 10, 79.99),
125     (4, 8, 7, 79.99),
126     (5, 6, 9, 299.99),
127     (6, 9, 22, 89.99),
128     (6, 10, 55, 199.99),
129     (7, 11, 34, 129.99),
130     (7, 12, 3, 349.99),
131     (8, 13, 67, 79.99),
132     (8, 14, 73, 139.99),
133     (9, 15, 109, 29.99),
134     (10, 16, 23, 89.99),
135     (11, 17, 64, 399.99),
136     (12, 18, 45, 299.99),
137     (13, 19, 87, 59.99),
138     (14, 20, 11, 199.99),
```


Result Grid | Filter Rows: | Edit:

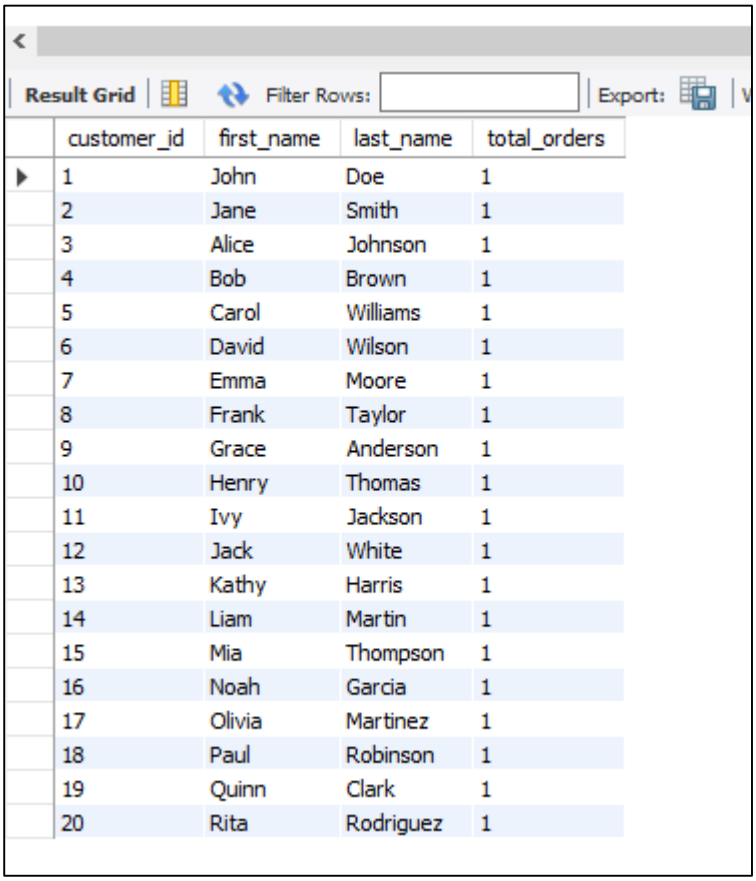
order_item_id	order_id	product_id	quantity	price
1	1	1	28	699.99
2	1	3	23	149.99
3	2	2	10	1199.99
4	2	4	54	79.99
5	3	5	12	249.99
6	3	7	10	79.99
7	4	8	7	79.99
8	5	6	9	299.99
9	6	9	22	89.99
10	6	10	55	199.99
11	7	11	34	129.99
12	7	12	3	349.99
13	8	13	67	79.99
14	8	14	73	139.99
15	9	15	109	29.99
16	10	16	23	89.99
17	11	17	64	399.99
18	12	18	45	299.99
19	13	19	87	59.99
20	14	20	11	199.99
21	15	1	33	699.99
22	15	4	76	79.99

Job for Business Analyst

1. Customer Purchase Frequency

Question: Determine the number of orders placed by each customer.

```
151 •   SELECT
152         c.customer_id,
153         c.first_name,
154         c.last_name,
155         COUNT(o.order_id) AS total_orders
156     FROM
157         customers c
158     JOIN
159         orders o ON c.customer_id = o.customer_id
160     GROUP BY
161         c.customer_id, c.first_name, c.last_name
162     ORDER BY
163         total_orders DESC;
```



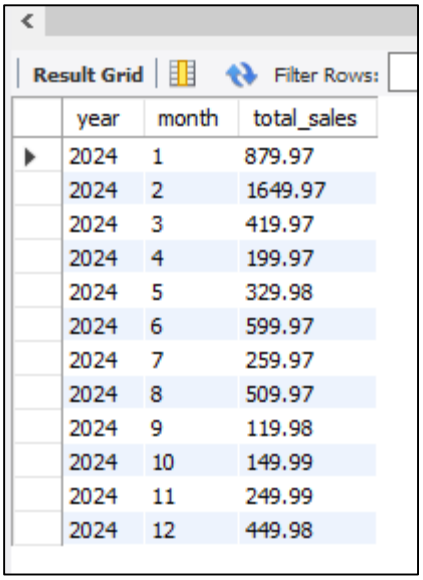
Result Grid | Filter Rows: | Export:

customer_id	first_name	last_name	total_orders
1	John	Doe	1
2	Jane	Smith	1
3	Alice	Johnson	1
4	Bob	Brown	1
5	Carol	Williams	1
6	David	Wilson	1
7	Emma	Moore	1
8	Frank	Taylor	1
9	Grace	Anderson	1
10	Henry	Thomas	1
11	Ivy	Jackson	1
12	Jack	White	1
13	Kathy	Harris	1
14	Liam	Martin	1
15	Mia	Thompson	1
16	Noah	Garcia	1
17	Olivia	Martinez	1
18	Paul	Robinson	1
19	Quinn	Clark	1
20	Rita	Rodriguez	1

2. Monthly Sales Performance

Question: Calculate the total sales amount for each month.

```
166 •  SELECT
167        YEAR(o.order_date) AS year,
168        MONTH(o.order_date) AS month,
169        SUM(o.total_amount) AS total_sales
170    FROM
171        orders o
172    GROUP BY
173        YEAR(o.order_date), MONTH(o.order_date)
174    ORDER BY
175        year, month;
```



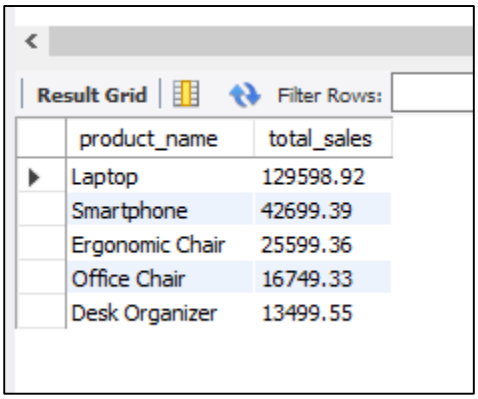
Result Grid | Filter Rows:

year	month	total_sales
2024	1	879.97
2024	2	1649.97
2024	3	419.97
2024	4	199.97
2024	5	329.98
2024	6	599.97
2024	7	259.97
2024	8	509.97
2024	9	119.98
2024	10	149.99
2024	11	249.99
2024	12	449.98

3. Top Selling Products

Question: Identify the top 5 products based on total sales.

```
178 •  SELECT
179        p.product_name,
180        SUM(oi.quantity * oi.price) AS total_sales
181    FROM
182        order_items oi
183    JOIN
184        products p ON oi.product_id = p.product_id
185    GROUP BY
186        p.product_name
187    ORDER BY
188        total_sales DESC
189    LIMIT 5;
```



Result Grid | Filter Rows:

product_name	total_sales
Laptop	129598.92
Smartphone	42699.39
Ergonomic Chair	25599.36
Office Chair	16749.33
Desk Organizer	13499.55

4. Customer Segmentation by Spending

Question: Segment customers into different spending categories (Low, Medium, High).

```sql
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(o.total_amount) AS total_spent,
    CASE
        WHEN SUM(o.total_amount) <= 500 THEN 'Low'
        WHEN SUM(o.total_amount) BETWEEN 501 AND 1000 THEN 'Medium'
        ELSE 'High'
    END AS spending_category
FROM
    orders o
JOIN
    customers c ON o.customer_id = c.customer_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_spent DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_id	first_name	last_name	total_spent	spending_category
2	Jane	Smith	1299.98	High
1	John	Doe	799.98	Medium
12	Jack	White	449.98	Low
8	Frank	Taylor	359.98	Low
14	Liam	Martin	349.99	Low
3	Alice	Johnson	329.98	Low
18	Paul	Robinson	299.99	Low
6	David	Wilson	299.98	Low
5	Carol	Williams	249.99	Low
11	Ivy	Jackson	249.99	Low
7	Emma	Moore	169.98	Low
10	Henry	Thomas	149.99	Low
20	Rita	Rodriguez	149.99	Low
9	Grace	Anderson	119.98	Low
16	Noah	Garcia	119.98	Low
15	Mia	Thompson	89.99	Low
19	Quinn	Clark	89.99	Low
4	Bob	Brown	79.99	Low
13	Kathy	Harris	79.99	Low
17	Olivia	Martinez	79.99	Low

5. Average Order Value

Question: Calculate the average order value for each customer.

```
199 •  SELECT
200        c.customer_id,
201        c.first_name,
202        c.last_name,
203        AVG(o.total_amount) AS avg_order_value
204    FROM
205        orders o
206    JOIN
207        customers c ON o.customer_id = c.customer_id
208    GROUP BY
209        c.customer_id, c.first_name, c.last_name
210    ORDER BY
211        avg_order_value DESC;
```

Result Grid | Filter Rows: | Export:

customer_id	first_name	last_name	avg_order_value
2	Jane	Smith	1299.980000
1	John	Doe	799.980000
12	Jack	White	449.980000
8	Frank	Taylor	359.980000
14	Liam	Martin	349.990000
3	Alice	Johnson	329.980000
18	Paul	Robinson	299.990000
6	David	Wilson	299.980000
5	Carol	Williams	249.990000
11	Ivy	Jackson	249.990000
7	Emma	Moore	169.980000
10	Henry	Thomas	149.990000
20	Rita	Rodriguez	149.990000
9	Grace	Anderson	119.980000
16	Noah	Garcia	119.980000
15	Mia	Thompson	89.990000
19	Quinn	Clark	89.990000
4	Bob	Brown	79.990000
13	Kathy	Harris	79.990000
17	Olivia	Martinez	79.990000

6. Product Sales by Category

Question: Determine the total sales for each product category.

```
198 •  SELECT
199        p.category,
200        SUM(oi.quantity * oi.price) AS total_sales
201    FROM
202        order_items oi
203    JOIN
204        products p ON oi.product_id = p.product_id
205    GROUP BY
206        p.category
207    ORDER BY
208        total_sales DESC;
```

Result Grid | Filter Rows:

category	total_sales
Electronics	216065.90
Furniture	63447.29
Accessories	40094.52

7. Sales by Product Over Time

Question: Track the total sales of each product over the past 12 months.

```
200 •  SELECT
201        p.product_name,
202        DATE_FORMAT(o.order_date, '%Y-%m') AS month,
203        SUM(oi.quantity * oi.price) AS monthly_sales
204    FROM
205        order_items oi
206    JOIN
207        orders o ON oi.order_id = o.order_id
208    JOIN
209        products p ON oi.product_id = p.product_id
210    WHERE
211        o.order_date >= DATE_SUB(CURDATE(), INTERVAL 12 MONTH)
212    GROUP BY
213        p.product_name, month
214    ORDER BY
215        p.product_name, month;
```

Result Grid | Filter Rows:

product_name	month	monthly_sales
Bluetooth Speaker	2024-08	5359.33
Desk Lamp	2024-04	559.93
Desk Lamp	2024-08	7039.12
Desk Organizer	2024-12	13499.55
Ergonomic Chair	2024-11	25599.36
External Hard Drive	2024-08	10219.27
Gaming Mouse	2024-10	2069.77
Headphones	2024-01	3449.77
Headphones	2024-04	3449.77
Keyboard	2024-02	4319.46
Keyboard	2024-03	6079.24
Laptop	2024-02	11999.90
Laptop	2024-04	117599.02
Monitor	2024-05	2699.91
Monitor	2024-06	10199.66
Mouse	2024-03	799.90
Mouse	2024-07	6079.24
Office Chair	2024-03	2999.88
Office Chair	2024-05	13749.45
Portable SSD	2024-02	2199.89
Printer	2024-07	4419.66
Smartphone	2024-01	19599.72

8. Top Customers by Spending

Question: Find the top 10 customers by total spending.

```sql
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(o.total_amount) AS total_spent
FROM
    orders o
JOIN
    customers c ON o.customer_id = c.customer_id
GROUP BY
    c.customer_id, c.first_name, c.last_name
ORDER BY
    total_spent DESC
LIMIT 10;
```

customer_id	first_name	last_name	total_spent
2	Jane	Smith	1299.98
1	John	Doe	799.98
12	Jack	White	449.98
8	Frank	Taylor	359.98
14	Liam	Martin	349.99
3	Alice	Johnson	329.98
18	Paul	Robinson	299.99
6	David	Wilson	299.98
5	Carol	Williams	249.99
11	Ivy	Jackson	249.99

9. Order Value Distribution

Question: Determine the distribution of order values into different ranges (e.g., Low, Medium, High).

```sql
SELECT
    CASE
        WHEN total_amount <= 100 THEN 'Low'
        WHEN total_amount BETWEEN 101 AND 500 THEN 'Medium'
        ELSE 'High'
    END AS order_value_range,
    COUNT(order_id) AS order_count
FROM
    orders
GROUP BY
    order_value_range;
```

order_value_range	order_count
High	2
Medium	13
Low	5