### **Created Database & Tables**

	#		Time	Action	Message
0		1	13:13:55	CREATE DATABASE ecommerce	1 row(s) affected
0		2	13:13:56	USE ecommerce	0 row(s) affected
0		3	13:14:18	${\sf CREATE\ TABLE\ customer\_id\ INT\ PRIMARY\ KEY\ AUTO\_INCREMENT}, {\sf customer\_name\ VARCHAR} (100), {\sf email\ VARCHAR} (100$	0 row(s) affected
0		4	13:14:21	${\sf CREATE\ TABLE\ product\_id\ INT\ PRIMARY\ KEY\ AUTO\_INCREMENT}, product\_name\ VARCHAR (100), category\ VARCHAR (50),\dots$	0 row(s) affected
0		5	13:14:23	CREATE TABLE orders ( order_id INT PRIMARY KEY AUTO_INCREMENT, customer_id INT, order_date DATE, total_amount DECIMAL	0 row(s) affected
0		6	13:14:25	CREATE TABLE order_items ( order_item_id INT PRIMARY KEY AUTO_INCREMENT, order_id INT, product_id INT, quantity INT, sub	0 row(s) affected

### **Customer Table**

	customer_id	customer_name	email	country
•	1	Amit Patel	amit@gmail.com	India
	2	Sarah Smith	sarah@gmail.com	USA
	3	Mohammed Ali	ali@yahoo.com	UAE
	NULL	NULL	NULL	NULL

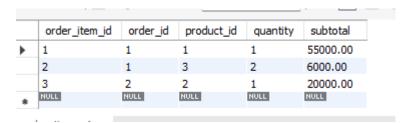
#### **Products Table**

	product_id	product_name	category	price
•	1	Laptop	Electronics	55000.00
	2	Smartphone	Electronics	20000.00
	3	Shoes	Fashion	3000.00
	NULL	NULL	NULL	NULL

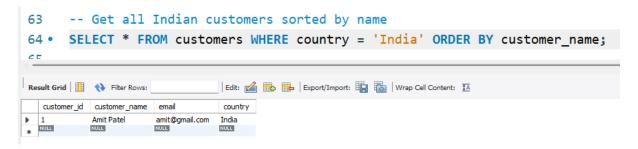
### Order Table

	order_id	customer_id	order_date	total_amount
•	1	1	2024-04-01	75000.00
	2	2	2024-04-02	20000.00
	NULL	NULL	NULL	NULL

## Order\_items Table



→ Select, Where, Groupby, Orderby queries



```
-- Total orders and amount spent by each customer
67 • SELECT customer_id, COUNT(order_id) AS total_orders, SUM(total_amount) AS total_spent
    FROM orders GROUP BY customer_id;
Export: Wrap Cell Content: IA
 customer_id total_orders total_spent
             75000.00
    1 20000.00
 70
        -- Monthly order summary with total revenue
        SELECT
 71 •
             DATE FORMAT(order date, '%Y-%m') AS order month,
 72
             COUNT(order id) AS total orders,
 73
             SUM(total amount) AS total revenue
 74
 75
        FROM orders GROUP BY order_month ORDER BY order_month;
 76
Export: Wrap Cell Content: 1A
    order_month total_orders total_revenue
 2024-04
                     95000.00
            2
  → JOINS (INNER, LEFT, RIGHT)
       -- INNER JOIN: Orders with customer names
      SELECT o.order_id, c.customer_name, o.order_date, o.total_amount
79
       FROM orders o
       INNER JOIN customers c ON o.customer id = c.customer id;
80
21
Export: Wrap Cell Content: $\frac{1}{4}$
  order_id customer_name order_date total_amount
                  2024-04-01 75000.00
        Amit Patel
 2
        Sarah Smith 2024-04-02 20000.00
```

```
-- LEFT JOIN: All customers with or without orders
82
      SELECT c.customer name, o.order id
83 •
      FROM customers c
84
      LEFT JOIN orders o ON c.customer_id = o.customer_id;
85
26
Export: Wrap Cell Content: TA
  customer_name order_id
  Amit Patel
           1
  Sarah Smith
           NULL
  Mohammed Ali
      -- RIGHT JOIN: All orders with customer names
87
88 • SELECT o.order_id, c.customer_name
      FROM orders o
89
      RIGHT JOIN customers c ON o.customer id = c.customer id;
90
                           Export: Wrap Cell Content: IA
order_id customer_name
  1
       Amit Patel
       Sarah Smith
  NULL
       Mohammed Ali
  → Subqueries
 92
      -- Customers who spent more than average total order amount
      SELECT customer_name FROM customers
 SELECT customer id
 95
          FROM orders
 96
          GROUP BY customer_id
 97
          HAVING SUM(total amount) > (
 98
 99
              SELECT AVG(total_amount) FROM orders
          )
100
      );
101
                          Export: Wrap Cell Content: IA
 customer_name
Amit Patel
```

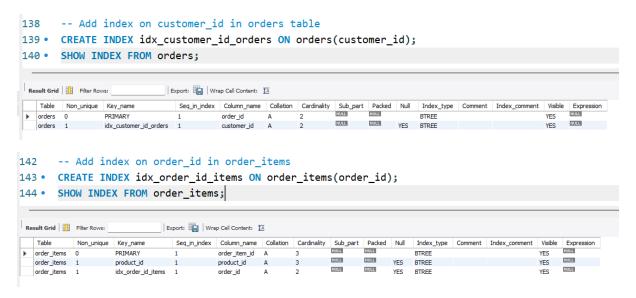
2

20000.000000

```
103
      -- Find customers who placed the highest value single order
104 • SELECT customer_name
105
      FROM customers
SELECT customer_id
107
          FROM orders
108
          WHERE total_amount = (
109
               SELECT MAX(total_amount) FROM orders
110
111
           )
112
      );
Export: Wrap Cell Content: IA
   customer_name
Amit Patel
 → Aggregate Functions (Sum, Average, Minimum, Maximum)
114
       -- Average order amount per customer
115 • SELECT customer_id, AVG(total_amount) AS avg_order_value
       FROM orders
116
       GROUP BY customer_id;
117
                            Export: Wrap Cell Content: $\frac{1}{2}
customer_id avg_order_value
          75000.000000
  1
```

```
119
        -- Total quantity sold per product
        SELECT p.product_name, SUM(oi.quantity) AS total_quantity_sold
120 •
121
        FROM order_items oi
122
        JOIN products p ON oi.product_id = p.product_id
123
        GROUP BY p.product_id;
Export: Wrap Cell Content: $\frac{1}{2}A
   product_name total_quantity_sold
  Laptop
   Shoes
  Smartphone
125 ·
          SELECT
                MAX(total amount) AS highest order,
126
                MIN(total_amount) AS lowest_order
127
          FROM orders;
128
  Result Grid | Filter Rows:
                                         Export: Wrap Cell Content: $\overline{1}{4}
     highest_order lowest_order
   75000.00
                20000.00
   → Views
130 • CREATE VIEW customer_summary AS
131 SELECT c.customer_id, c.customer_name, COUNT(o.order_id) AS total_orders, SUM(o.total_amount) AS total_spent
    FROM customers c
133 LEFT JOIN orders o ON c.customer_id = o.customer_id
134 GROUP BY c.customer_id;
          select * from customer_summary;
136 •
 Export: Wrap Cell Content:
     customer_id customer_name total_orders total_spent
    1
               Amit Patel
                                       75000.00
    2
                                       20000.00
               Sarah Smith
    3
               Mohammed Ali
customer summary 16 ×
```

#### → Indexes



→ Case statement for conditional logic

```
147 • SELECT
148
         customer_name,
149
         total_amount,
150 ⊖
         CASE
           WHEN total_amount >= 1000 THEN 'High Value'
151
152
           WHEN total_amount >= 500 THEN 'Medium Value'
153
           ELSE 'Low Value'
154
         END AS order category
       FROM orders
155
156
       JOIN customers ON orders.customer_id = customers.customer_id;
                             Export: Wrap Cell Content: IA
customer_name total_amount order_category
  Amit Patel
           75000.00 High Value
  Sarah Smith 20000.00 High Value
```

# → Date based analysis

## → Ranking