```sql
CREATE DATABASE E_COMMERCE;
USE E_COMMERCE;

-- create tables
CREATE TABLE Categories (
    category_id INT PRIMARY KEY,
    category_name VARCHAR(50)
);

CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    name VARCHAR(100),
    category_id INT,
    price DECIMAL(10,2),
    stock_quantity INT,
    added_date DATE,
    FOREIGN KEY (category_id) REFERENCES Categories(category_id)
);
```

```sql
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone_number VARCHAR(20),
    address VARCHAR(200),
    registration_date DATE
);

CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_id INT,
    order_date DATE,
    total_amount DECIMAL(10,2),
    status VARCHAR(20),
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
```

```sql
CREATE TABLE Order_Items (
    order_item_id INT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    subtotal DECIMAL(10,2),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

CREATE TABLE Payments (
    payment_id INT PRIMARY KEY,
    order_id INT,
    payment_date DATE,
    payment_method VARCHAR(20),
    payment_status VARCHAR(20),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);


CREATE TABLE Shipping (
    shipping_id INT PRIMARY KEY,
    order_id INT,
    shipping_date DATE,
    delivery_date DATE,
    shipping_status VARCHAR(30),
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);
```

```sql
-- Insert values
INSERT INTO Categories VALUES
(1,'Electronics'),
(2,'Clothing'),
(3,'Books'),
(4,'Home'),
(5,'Toys');

INSERT INTO Products VALUES
(1,'Mobile',1,15000,50,'2024-01-10'),
(2,'Laptop',1,55000,20,'2024-02-05'),
(3,'T-Shirt',2,500,100,'2024-01-15'),
(4,'Jeans',2,1200,60,'2024-02-20'),
(5,'Novel',3,300,80,'2024-03-01'),
(6,'Cookware Set',4,2500,30,'2024-01-25'),
(7,'Toy bike',5,900,40,'2024-02-10'),
(8,'BhagvatGeeta',3,1800,60,'2024-02-20');


INSERT INTO Customers VALUES
(1,'Rahul Sharma','rahul@gmail.com','9876543210','Ahmedabad','2022-05-10'),
(2,'Priya Patel','priya@gmail.com','9876543211','Surat','2023-01-15'),
(3,'Amit Verma','amit@gmail.com','9876543212','Vadodara','2021-07-20'),
(4,'Neha Singh','neha@gmail.com','9876543213','Rajkot','2024-03-01'),
(5,'Karan Mehta','karan@gmail.com','9876543214','Mumbai','2023-11-05'),
(6,'Riya Desai','riya@gmail.com','9876543215','Pune','2022-08-18'),
(7,'Arjun Shah','arjun@gmail.com','9876543216','Delhi','2024-01-10'),
(8,'Simran Kaur','simran@gmail.com','9876543217','Chandigarh','2022-12-25'),
(9,'Mohit Jain','mohit@gmail.com','9876543218','Indore','2023-06-30'),
(10,'Pooja Nair','pooja@gmaul.com','9876543219','Kochi','2021-09-12');


INSERT INTO Orders VALUES
(1,1,'2024-01-10',15500,'Delivered'),
(2,2,'2024-02-05',55000,'Shipped'),
(3,3,'2023-12-15',300,'Cancelled'),
(4,4,'2024-03-10',1200,'Pending'),
(5,5,'2024-01-20',2500,'Delivered'),
(6,6,'2024-02-25',900,'Delivered'),
(7,7,'2024-03-01',15000,'Pending'),
(8,8,'2023-11-15',500,'Delivered'),
(9,9,'2024-01-30',300,'Cancelled'),
(10,10,'2024-02-10',55000,'Delivered'),
(12, 1, '2024-03-22', 3000, 'Delivered'),
(13, 1, '2024-03-23', 4500, 'Delivered'),
(14, 1, '2024-03-24', 2000, 'Pending');


INSERT INTO Order_Items VALUES
(1,1,1,1,15000),
(2,1,5,1,500),
(3,2,2,1,55000),
(4,3,5,1,300),
(5,4,4,1,1200),
(6,5,6,1,2500),
(7,6,7,1,900),
(8,7,1,1,15000),
(9,8,3,1,500),
(10,10,2,1,55000);
```

```sql
INSERT INTO Payments VALUES
(1,1,'2024-01-10','UPI','Paid'),
(2,2,'2024-02-05','Credit Card','Paid'),
(3,3,'2023-12-15','UPI','Failed'),
(4,4,'2024-03-10','PayPal','Pending'),
(5,5,'2024-01-20','Credit Card','Paid'),
(6,6,'2024-02-25','UPI','Paid'),
(7,7,'2024-03-01','UPI','Pending'),
(8,8,'2023-11-15','Credit Card','Paid'),
(9,9,'2024-01-30','PayPal','Failed'),
(10,10,'2024-02-10','Credit Card','Paid');
```

```sql
INSERT INTO Shipping VALUES
(1,1,'2024-01-11','2024-01-15','Delivered'),
(2,2,'2024-02-06',NULL,'In Transit'),
(3,4,'2024-03-11',NULL,'Dispatched'),
(4,5,'2024-01-21','2024-01-25','Delivered'),
(5,6,'2024-02-26','2024-03-01','Delivered');

SELECT * FROM Categories;

SELECT * FROM Products;

SELECT * FROM Customers;

SELECT * FROM Orders;

SELECT * FROM Order_Items;

SELECT * FROM Payments;

SELECT * FROM Shipping;
```

```sql
-- CRUD
-- INSERT new product
INSERT INTO Products VALUES
(9, 'Headphones', 1, 2000, 25, '2024-03-15');
SELECT * FROM Products;
```

product_id	name	category_id	price	stock_quantity	added_date
1	Mobile	1	15000.00	49	2024-01-10
2	Laptop	1	55000.00	20	2024-02-05
3	T-Shirt	2	500.00	100	2024-01-15
4	Jeans	2	1200.00	60	2024-02-20
5	Novel	3	300.00	80	2024-03-01
6	Cookware Set	4	2500.00	30	2024-01-25
7	Toy bike	5	900.00	40	2024-02-10
8	BhagvatGeeta	3	1800.00	60	2024-02-20
9	Headphones	1	2000.00	25	2024-03-15
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- INSERT new customer
INSERT INTO Customers VALUES
(11, 'Ritu Rao', 'ritu@gmail.com', '9876500000', 'Jaipur', '2024-03-20');
SELECT * FROM Customers;

-- INSERT new order
INSERT INTO Orders VALUES
(11, 11, '2024-03-21', 2000, 'Pending');
SELECT * FROM Orders;
```

```sql
-- UPDATE stock when order is placed
UPDATE Products
SET stock_quantity = stock_quantity - 1
WHERE product_id = 1;
SELECT * FROM Products;
```

product_id	name	category_id	price	stock_quantity	added_date
1	Mobile	1	15000.00	48	2024-01-10
2	Laptop	1	55000.00	20	2024-02-05
3	T-Shirt	2	500.00	100	2024-01-15
4	Jeans	2	1200.00	60	2024-02-20
5	Novel	3	300.00	80	2024-03-01
6	Cookware Set	4	2500.00	30	2024-01-25
7	Toy bike	5	900.00	40	2024-02-10
8	BhagvatGeeta	3	1800.00	60	2024-02-20
9	Headphones	1	2000.00	25	2024-03-15
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- DELETE cancelled orders older than 30 days
DELETE FROM Orders
WHERE status = 'Cancelled'
AND order_date < CURDATE() - INTERVAL 30 DAY;
SELECT * FROM Orders;
```

order_id	customer_id	order_date	total_amount	status
1	1	2024-01-10	15500.00	Delivered
2	2	2024-02-05	55000.00	Shipped
3	3	2023-12-15	300.00	Cancelled
4	4	2024-03-10	1200.00	Pending
5	5	2024-01-20	2500.00	Delivered
6	6	2024-02-25	900.00	Delivered
7	7	2024-03-01	15000.00	Pending
8	8	2023-11-15	500.00	Delivered
9	9	2024-01-30	300.00	Cancelled
10	10	2024-02-10	55000.00	Delivered
NULL	NULL	NULL	NULL	NULL

```sql
-- USE SQL CLAUSES (WHERE, HAVING, LIMIT)
-- Orders placed in last 6 months
SELECT * FROM Orders
WHERE order_date >= CURDATE() - INTERVAL 6 MONTH;
```

order_id	customer_id	order_date	total_amount	status
NULL	NULL	NULL	NULL	NULL

```sql
-- Top 5 highest-priced products
SELECT * FROM Products
ORDER BY price DESC
LIMIT 5;
```

product_id	name	category_id	price	stock_quantity	added_date
2	Laptop	1	55000.00	20	2024-02-05
1	Mobile	1	15000.00	48	2024-01-10
6	Cookware Set	4	2500.00	30	2024-01-25
9	Headphones	1	2000.00	25	2024-03-15
8	BhagvatGeeta	3	1800.00	60	2024-02-20
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- Customers who placed more than 3 orders
SELECT c.customer_id, c.name, COUNT(o.order_id) AS total_orders
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
HAVING COUNT(o.order_id) > 3;
```

customer_id	name	total_orders
1	Rahul Sharma	4

```sql
-- APPLY SQL OPERATORS (AND, OR, NOT)
-- Orders where status = 'Pending' AND payment_status = 'Paid'
SELECT o.order_id, o.status, p.payment_status
FROM Orders o
JOIN Payments p ON o.order_id = p.order_id
WHERE o.status = 'Pending'
AND p.payment_status = 'Paid';
```

order_id	status	payment_status

```sql
-- Products that are NOT out of stock
SELECT * FROM Products
WHERE NOT stock_quantity = 0;
```

product_id	name	category_id	price	stock_quantity	added_date
1	Mobile	1	15000.00	48	2024-01-10
2	Laptop	1	55000.00	20	2024-02-05
3	T-Shirt	2	500.00	100	2024-01-15
4	Jeans	2	1200.00	60	2024-02-20
5	Novel	3	300.00	80	2024-03-01
6	Cookware Set	4	2500.00	30	2024-01-25
7	Toy bike	5	900.00	40	2024-02-10
8	BhagvatGeeta	3	1800.00	60	2024-02-20
9	Headphones	1	2000.00	25	2024-03-15
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- Customers registered after 2022 OR purchased above 10000
SELECT DISTINCT c.customer_id, c.name
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
WHERE c.registration_date > '2022-12-31'
OR o.total_amount > 10000;
```

customer_id	name
1	Rahul Sharma
2	Priya Patel
4	Neha Singh
5	Karan Mehta
7	Arjun Shah
9	Mohit Jain
10	Pooja Nair
11	Ritu Rao

```sql
-- SORTING & GROUPING (ORDER BY, GROUP BY)
-- Products sorted by price (descending)
SELECT * FROM Products
ORDER BY price DESC;
```

product_id	name	category_id	price	stock_quantity	added_date
2	Laptop	1	55000.00	20	2024-02-05
1	Mobile	1	15000.00	48	2024-01-10
6	Cookware Set	4	2500.00	30	2024-01-25
9	Headphones	1	2000.00	25	2024-03-15
8	BhagvatGeeta	3	1800.00	60	2024-02-20
4	Jeans	2	1200.00	60	2024-02-20
7	Toy bike	5	900.00	40	2024-02-10
3	T-Shirt	2	500.00	100	2024-01-15
5	Novel	3	300.00	80	2024-03-01
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- Number of orders placed by each customer
SELECT c.customer_id, c.name, COUNT(o.order_id) AS total_orders
FROM Customers c
LEFT JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;
```

customer_id	name	total_orders
1	Rahul Sharma	4
2	Priya Patel	1
3	Amit Verma	1
4	Neha Singh	1
5	Karan Mehta	1
6	Riya Desai	1
7	Arjun Shah	1
8	Simran Kaur	1
9	Mohit Jain	1
10	Pooja Nair	1

```sql
-- Total revenue generated per category
SELECT cat.category_name, SUM(oi.subtotal) AS total_revenue
FROM Order_Items oi
JOIN Products p ON oi.product_id = p.product_id
JOIN Categories cat ON p.category_id = cat.category_id
GROUP BY cat.category_name;
```

category_name	total_revenue
Electronics	140000.00
Clothing	1700.00
Books	800.00
Home	2500.00
Toys	900.00

```sql
-- AGGREGATE FUNCTIONS (SUM, AVG, MAX, MIN, COUNT)
-- Total revenue generated by store
SELECT SUM(total_amount) AS total_revenue
FROM Orders
WHERE status = 'Delivered';
```

total_revenue
81900.00

```sql
-- Most purchased product
SELECT p.name, SUM(oi.quantity) AS total_quantity
FROM Order_Items oi
JOIN Products p ON oi.product_id = p.product_id
GROUP BY p.name
ORDER BY total_quantity DESC
LIMIT 1;
```

name	total_quantity
Mobile	2

```sql
-- Average order value
SELECT AVG(total_amount) AS average_order_value
FROM Orders;
```

average_order_value
11264.285714

```sql
-- JOINS
-- INNER JOIN: Products with category names
SELECT p.product_id, p.name, c.category_name, p.price
FROM Products p
INNER JOIN Categories c
ON p.category_id = c.category_id;
```

product_id	name	category_name	price
1	Mobile	Electronics	15000.00
2	Laptop	Electronics	55000.00
9	Headphones	Electronics	2000.00
3	T-Shirt	Clothing	500.00
4	Jeans	Clothing	1200.00
5	Novel	Books	300.00
8	BhagvatGeeta	Books	1800.00
6	Cookware Set	Home	2500.00
7	Toy bike	Toys	900.00

```sql
-- LEFT JOIN: All orders with customer details
SELECT o.order_id, o.order_date, o.total_amount, o.status, c.customer_id, c.name, c.email
FROM Orders o
LEFT JOIN Customers c
ON o.customer_id = c.customer_id;
```

order_id	order_date	total_amount	status	customer_id	name	email
1	2024-01-10	15500.00	Delivered	1	Rahul Sharma	rahul@gmail.com
2	2024-02-05	55000.00	Shipped	2	Priya Patel	priya@gmail.com
3	2023-12-15	300.00	Cancelled	3	Amit Verma	amit@gmail.com
4	2024-03-10	1200.00	Pending	4	Neha Singh	neha@gmail.com
5	2024-01-20	2500.00	Delivered	5	Karan Mehta	karan@gmail.com
6	2024-02-25	900.00	Delivered	6	Riya Desai	riya@gmail.com
7	2024-03-01	15000.00	Pending	7	Arjun Shah	arjun@gmail.com
8	2023-11-15	500.00	Delivered	8	Simran Kaur	simran@gmail.com
9	2024-01-30	300.00	Cancelled	9	Mohit Jain	mohit@gmail.com
10	2024-02-10	55000.00	Delivered	10	Pooja Nair	pooja@gmaul.com

```sql
-- RIGHT JOIN: Orders that haven't been shipped
SELECT o.order_id, s.shipping_status
FROM Orders o
RIGHT JOIN Shipping s
ON o.order_id = s.order_id
WHERE s.shipping_status IS NULL;
```

	order_id	shipping_status

```sql
-- FULL OUTER JOIN:
SELECT c.customer_id, c.name, o.order_id
FROM Customers c
LEFT JOIN Orders o
ON c.customer_id = o.customer_id
UNION
SELECT c.customer_id, c.name, o.order_id
FROM Customers c
RIGHT JOIN Orders o
ON c.customer_id = o.customer_id;
```

customer_id	name	order_id
1	Rahul Sharma	1
1	Rahul Sharma	12
1	Rahul Sharma	13
1	Rahul Sharma	14
2	Priya Patel	2
3	Amit Verma	3
4	Neha Singh	4
5	Karan Mehta	5
6	Riya Desai	6
7	Arjun Shah	7
8	Simran Kaur	8
9	Mohit Jain	9
10	Pooja Nair	10
11	Ritu Rao	11

```sql
-- Customers who never placed an order
SELECT c.customer_id, c.name
FROM Customers c
LEFT JOIN Orders o
ON c.customer_id = o.customer_id
WHERE o.order_id IS NULL;
```

	customer_id	name

```sql
-- SUBQUERIES
-- Orders placed by customers who registered after 2022
SELECT * FROM Orders
WHERE customer_id IN (
    SELECT customer_id
    FROM Customers
    WHERE registration_date > '2022-12-31'
);
```

order_id	customer_id	order_date	total_amount	status
2	2	2024-02-05	55000.00	Shipped
4	4	2024-03-10	1200.00	Pending
5	5	2024-01-20	2500.00	Delivered
7	7	2024-03-01	15000.00	Pending
9	9	2024-01-30	300.00	Cancelled
11	11	2024-03-21	2000.00	Pending
NULL	NULL	NULL	NULL	NULL

```sql
-- Customer who has spent the most
SELECT customer_id, SUM(total_amount) AS total_spent
FROM Orders
GROUP BY customer_id
ORDER BY total_spent DESC
LIMIT 1;
```

customer_id	total_spent
2	55000.00

```sql
-- Products that have never been ordered
SELECT * FROM Products
WHERE product_id NOT IN (
    SELECT product_id FROM Order_Items
);
```

product_id	name	category_id	price	stock_quantity	added_date
8	BhagvatGeeta	3	1800.00	60	2024-02-20
9	Headphones	1	2000.00	25	2024-03-15
NULL	NULL	NULL	NULL	NULL	NULL

```sql
-- DATE & TIME FUNCTIONS
-- Count orders per month
SELECT MONTH(order_date) AS month, COUNT(*) AS total_orders
FROM Orders
GROUP BY MONTH(order_date);
```

month	total_orders
1	3
2	3
12	1
3	6
11	1

```sql
-- Calculate delivery time (difference in days)
SELECT order_id, DATEDIFF(delivery_date, shipping_date) AS delivery_days
FROM Shipping
WHERE delivery_date IS NOT NULL;
```

order_id	delivery_days
1	4
5	4
6	4

```sql
-- Format order_date as DD-MM-YYYY
SELECT order_id, DATE_FORMAT(order_date, '%d-%m-%Y') AS formatted_date
FROM Orders;
```

order_id	formatted_date
1	10-01-2024
2	05-02-2024
3	15-12-2023
4	10-03-2024
5	20-01-2024
6	25-02-2024
7	01-03-2024
8	15-11-2023
9	30-01-2024
10	10-02-2024

```sql
--  STRING FUNCTIONS
-- Convert product names to uppercase
SELECT UPPER(name) AS product_name
FROM Products;
```

product_name
MOBILE
LAPTOP
T-SHIRT
JEANS
NOVEL
COOKWARE SET
TOY BIKE
BHAGVATGEETA
HEADPHONES

```sql
-- Trim whitespace from customer names
SELECT TRIM(name) AS clean_name
FROM Customers;
```

clean_name
Rahul Sharma
Priya Patel
Amit Verma
Neha Singh
Karan Mehta
Riya Desai
Arjun Shah
Simran Kaur
Mohit Jain
Pooja Nair

```sql
-- Replace missing email with "Not Provided"
SELECT name,COALESCE(email, 'Not Provided') AS email_status
FROM Customers;
```

name	email_status
Rahul Sharma	rahul@gmail.com
Priya Patel	priya@gmail.com
Amit Verma	amit@gmail.com
Neha Singh	neha@gmail.com
Karan Mehta	karan@gmail.com
Riya Desai	riya@gmail.com
Arjun Shah	arjun@gmail.com
Simran Kaur	simran@gmail.com
Mohit Jain	mohit@gmail.com
Pooja Nair	pooja@gmaul.com

```sql
-- WINDOW FUNCTIONS
-- Rank customers based on total spending
SELECT customer_id, SUM(total_amount) AS total_spent,
RANK()
OVER (ORDER BY SUM(total_amount) DESC) AS rank_position
FROM Orders
GROUP BY customer_id;
```

customer_id	total_spent	rank_position
2	55000.00	1
10	55000.00	1
1	25000.00	3
7	15000.00	4
5	2500.00	5
11	2000.00	6
4	1200.00	7
6	900.00	8
8	500.00	9
3	300.00	10
9	300.00	10

```sql
-- Cumulative revenue per month
SELECT MONTH(order_date) AS month, SUM(total_amount) AS monthly_revenue,
SUM(SUM(total_amount)) OVER (ORDER BY MONTH(order_date)) AS cumulative_revenue
FROM Orders
GROUP BY MONTH(order_date);
```

month	monthly_revenue	cumulative_revenue
1	18300.00	18300.00
2	110900.00	129200.00
3	27700.00	156900.00
11	500.00	157400.00
12	300.00	157700.00

```sql
-- Running total of orders
SELECT order_id, order_date, SUM(total_amount)
OVER (ORDER BY order_date) AS running_total
FROM Orders;
```

order_id	order_date	running_total
8	2023-11-15	500.00
3	2023-12-15	800.00
1	2024-01-10	16300.00
5	2024-01-20	18800.00
9	2024-01-30	19100.00
2	2024-02-05	74100.00
10	2024-02-10	129100.00
6	2024-02-25	130000.00
7	2024-03-01	145000.00
4	2024-03-10	146200.00
11	2024-03-21	148200.00
12	2024-03-22	151200.00
13	2024-03-23	155700.00
14	2024-03-24	157700.00

```sql
-- Assign Loyalty_Status to Customers
SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_spent,
       CASE
           WHEN SUM(o.total_amount) > 50000 THEN 'Gold'
           WHEN SUM(o.total_amount) BETWEEN 20000 AND 50000 THEN 'Silver'
           ELSE 'Bronze'
       END AS Loyalty_Status
FROM Customers c
LEFT JOIN Orders o
ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name;
```

customer_id	name	total_spent	Loyalty_Status
1	Rahul Sharma	25000.00	Silver
2	Priya Patel	55000.00	Gold
3	Amit Verma	300.00	Bronze
4	Neha Singh	1200.00	Bronze
5	Karan Mehta	2500.00	Bronze
6	Riya Desai	900.00	Bronze
7	Arjun Shah	15000.00	Bronze
8	Simran Kaur	500.00	Bronze
9	Mohit Jain	300.00	Bronze
10	Pooja Nair	55000.00	Gold

```sql
-- Categorize Products Based on Units Sold
SELECT p.product_id,p.name, COALESCE(SUM(oi.quantity),0) AS total_units_sold,
       CASE
           WHEN SUM(oi.quantity) > 500 THEN 'Best Seller'
           WHEN SUM(oi.quantity) BETWEEN 200 AND 500 THEN 'Popular'
           ELSE 'Regular'
       END AS product_category
FROM Products p
LEFT JOIN Order_Items oi
ON p.product_id = oi.product_id
GROUP BY p.product_id, p.name;
```

product_id	name	total_units_sold	product_category
1	Mobile	2	Regular
2	Laptop	2	Regular
3	T-Shirt	1	Regular
4	Jeans	1	Regular
5	Novel	2	Regular
6	Cookware Set	1	Regular
7	Toy bike	1	Regular
8	BhagvatGeeta	0	Regular
9	Headphones	0	Regular