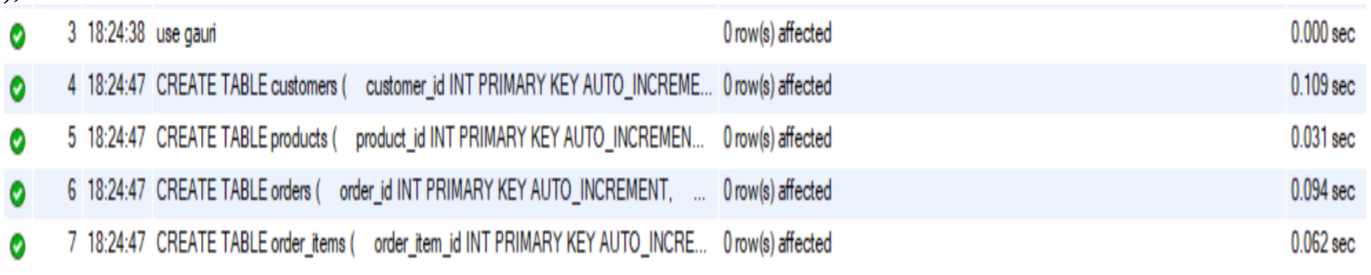
TASK 3 SQL for Data Analysis

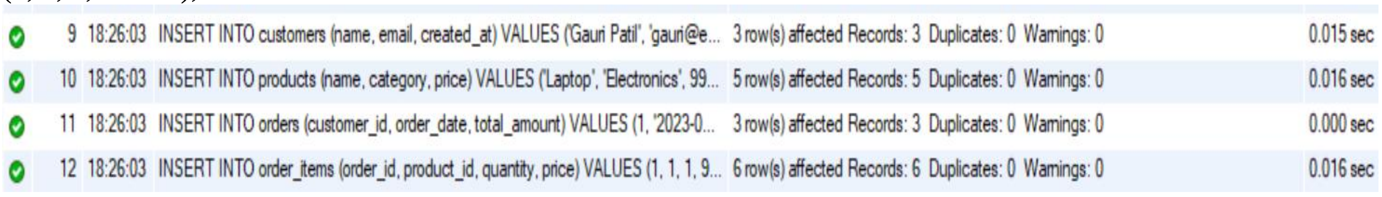
```
-- Customers table
use gauri;
CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    email VARCHAR(100),
    created_at DATE
);

-- Products table
CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    category VARCHAR(50),
    price DECIMAL(10, 2)
);

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

-- Order Items table
CREATE TABLE order_items (
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,
    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)
);
```



```
✓ 3 18:24:38 use gauri                                                          0 row(s) affected  0.000 sec
✓ 4 18:24:47 CREATE TABLE customers (   customer_id INT PRIMARY KEY AUTO_INCREME...  0 row(s) affected  0.109 sec
✓ 5 18:24:47 CREATE TABLE products (   product_id INT PRIMARY KEY AUTO_INCREMEN...   0 row(s) affected  0.031 sec
✓ 6 18:24:47 CREATE TABLE orders (   order_id INT PRIMARY KEY AUTO_INCREMENT,   ...  0 row(s) affected  0.094 sec
✓ 7 18:24:47 CREATE TABLE order_items (   order_item_id INT PRIMARY KEY AUTO_INCRE...  0 row(s) affected  0.062 sec
```

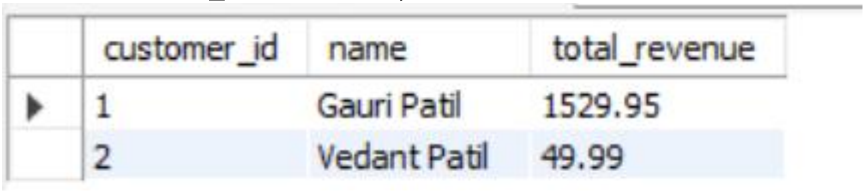
```
-- Insert into customers
INSERT INTO Customers (name, email, created_at) VALUES
```

```
('Gauri Patil', 'gauri@example.com', '2023-01-15'),
('Vedant Patil', 'vp@example.com', '2023-02-10'),
('Adinath Patil', 'adi@example.com', '2023-03-05');

-- Insert into products
INSERT INTO products (name, category, price) VALUES
('Laptop', 'Electronics', 999.99),
('Headphones', 'Electronics', 199.99),
('Keyboard', 'Accessories', 49.99),
('Mouse', 'Accessories', 29.99),
('Monitor', 'Electronics', 299.99);

-- Insert into orders
INSERT INTO orders (customer_id, order_date, total_amount) VALUES
(1, '2023-04-01', 1299.97),
(2, '2023-04-03', 49.99),
(1, '2023-04-10', 229.98);

-- Insert into order_items
INSERT INTO order_items (order_id, product_id, quantity, price) VALUES
(1, 1, 1, 999.99),
(1, 2, 1, 199.99),
(1, 3, 2, 49.99),
(2, 3, 1, 49.99),
(3, 4, 2, 29.99),
(3, 2, 1, 199.99);
```



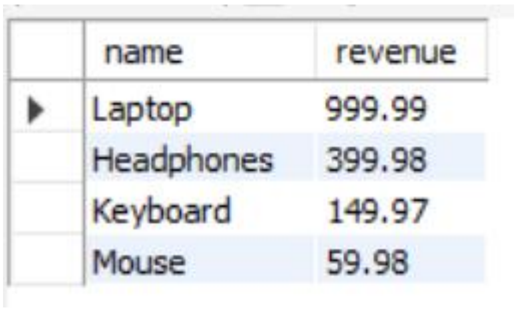
```
✓  9   18:26:03  INSERT INTO customers (name, email, created_at) VALUES ('Gauri Patil', 'gauri@e...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.015 sec
✓  10  18:26:03  INSERT INTO products (name, category, price) VALUES ('Laptop', 'Electronics', 99...   5 row(s) affected Records: 5  Duplicates: 0  Warnings: 0   0.016 sec
✓  11  18:26:03  INSERT INTO orders (customer_id, order_date, total_amount) VALUES (1, '2023-0...   3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0   0.000 sec
✓  12  18:26:03  INSERT INTO order_items (order_id, product_id, quantity, price) VALUES (1, 1, 1, 9...   6 row(s) affected Records: 6  Duplicates: 0  Warnings: 0   0.016 sec
```

```
-- 1. Total Revenue Per Customer
SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_revenue
FROM Customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
ORDER BY total_revenue DESC;
```



customer_id	name	total_revenue
1	Gauri Patil	1529.95
2	Vedant Patil	49.99

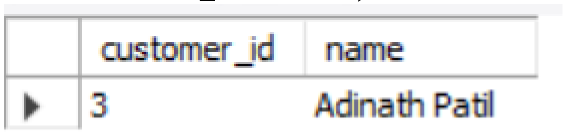
```
-- 2. Top 5 Products by Revenue
SELECT p.name, SUM(oi.quantity * oi.price) AS revenue
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY p.name
ORDER BY revenue DESC
LIMIT 5;
```



	name	revenue
▶	Laptop	999.99
	Headphones	399.98
	Keyboard	149.97
	Mouse	59.98

-- 3. Customers With No Orders

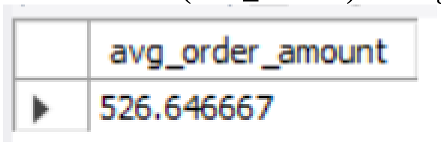
```
SELECT c.customer_id, c.name
FROM Customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id
WHERE o.order_id IS NULL;
```



	customer_id	name
▶	3	Adinath Patil

-- 4. Average Order Amount

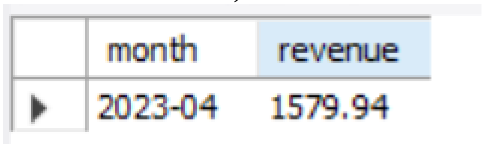
```
SELECT AVG(total_amount) AS avg_order_amount FROM orders;
```



	avg_order_amount
▶	526.646667

-- 5. Monthly Revenue Trend

```
SELECT DATE_FORMAT(order_date, '%Y-%m') AS month, SUM(total_amount) AS revenue
FROM orders
GROUP BY month;
```



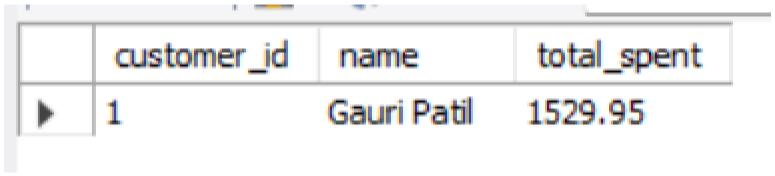
	month	revenue
▶	2023-04	1579.94

-- 6. Create a View for Revenue per Product

```
CREATE VIEW product_revenue AS
SELECT p.product_id, p.name, SUM(oi.quantity * oi.price) AS total_revenue
FROM products p
JOIN order_items oi ON p.product_id = oi.product_id
GROUP BY p.product_id, p.name;
```

-- 7. Customers Who Spent More Than $500

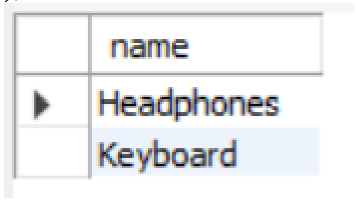
```
SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_spent
FROM Customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
HAVING total_spent > 500;
```



	customer_id	name	total_spent
▶	1	Gauri Patil	1529.95

-- 8. Subquery: Products Ordered More Than Once

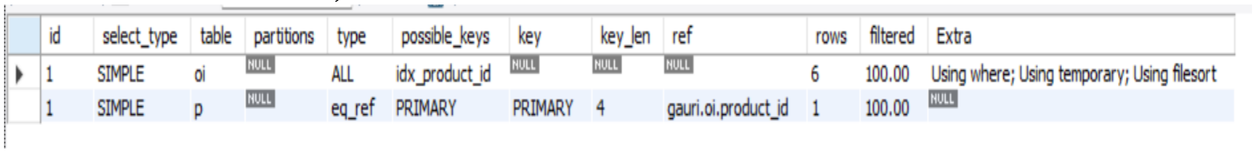
```
SELECT name FROM products
WHERE product_id IN (
   SELECT product_id
   FROM order_items
   GROUP BY product_id
   HAVING COUNT(*) > 1
);
```



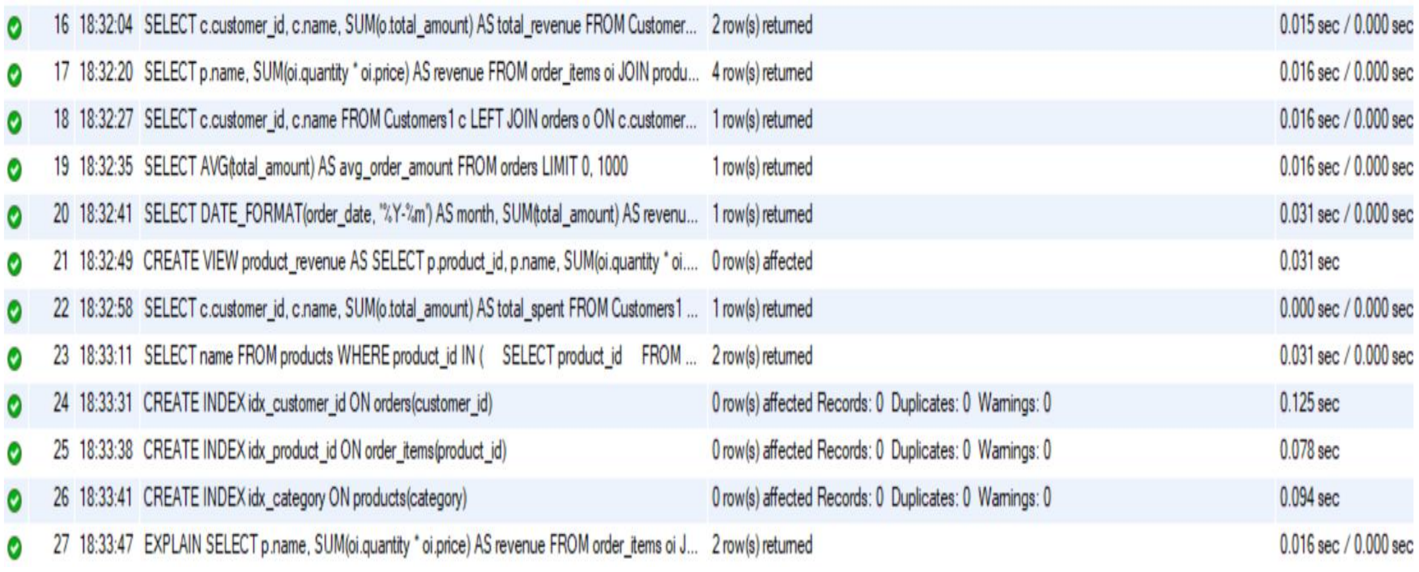
	name
▶	Headphones
	Keyboard

-- Use EXPLAIN on a complex query

```
EXPLAIN
SELECT p.name, SUM(oi.quantity * oi.price) AS revenue
FROM order_items oi
JOIN products p ON oi.product_id = p.product_id
GROUP BY p.name
ORDER BY revenue DESC;
```



	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	oi	NULL	ALL	idx_product_id	NULL	NULL	NULL	6	100.00	Using where; Using temporary; Using filesort
	1	SIMPLE	p	NULL	eq_ref	PRIMARY	PRIMARY	4	gauri.oi.product_id	1	100.00	NULL



```
✓ 16 18:32:04 SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_revenue FROM Customer...  2 row(s) returned                                      0.015 sec / 0.000 sec
✓ 17 18:32:20 SELECT p.name, SUM(oi.quantity * oi.price) AS revenue FROM order_items oi JOIN produ... 4 row(s) returned                                    0.016 sec / 0.000 sec
✓ 18 18:32:27 SELECT c.customer_id, c.name FROM Customers1 c LEFT JOIN orders o ON c.customer...      1 row(s) returned                                    0.016 sec / 0.000 sec
✓ 19 18:32:35 SELECT AVG(total_amount) AS avg_order_amount FROM orders LIMIT 0, 1000                 1 row(s) returned                                    0.016 sec / 0.000 sec
✓ 20 18:32:41 SELECT DATE_FORMAT(order_date, '%Y-%m') AS month, SUM(total_amount) AS revenu...        1 row(s) returned                                    0.031 sec / 0.000 sec
✓ 21 18:32:49 CREATE VIEW product_revenue AS SELECT p.product_id, p.name, SUM(oi.quantity * oi....   0 row(s) affected                                    0.031 sec
✓ 22 18:32:58 SELECT c.customer_id, c.name, SUM(o.total_amount) AS total_spent FROM Customers1 ...   1 row(s) returned                                    0.000 sec / 0.000 sec
✓ 23 18:33:11 SELECT name FROM products WHERE product_id IN (    SELECT product_id    FROM ...       2 row(s) returned                                    0.031 sec / 0.000 sec
✓ 24 18:33:31 CREATE INDEX idx_customer_id ON orders(customer_id)                                    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.125 sec
✓ 25 18:33:38 CREATE INDEX idx_product_id ON order_items(product_id)                                 0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.078 sec
✓ 26 18:33:41 CREATE INDEX idx_category ON products(category)                                        0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.094 sec
✓ 27 18:33:47 EXPLAIN SELECT p.name, SUM(oi.quantity * oi.price) AS revenue FROM order_items oi J... 2 row(s) returned                                    0.016 sec / 0.000 sec
```