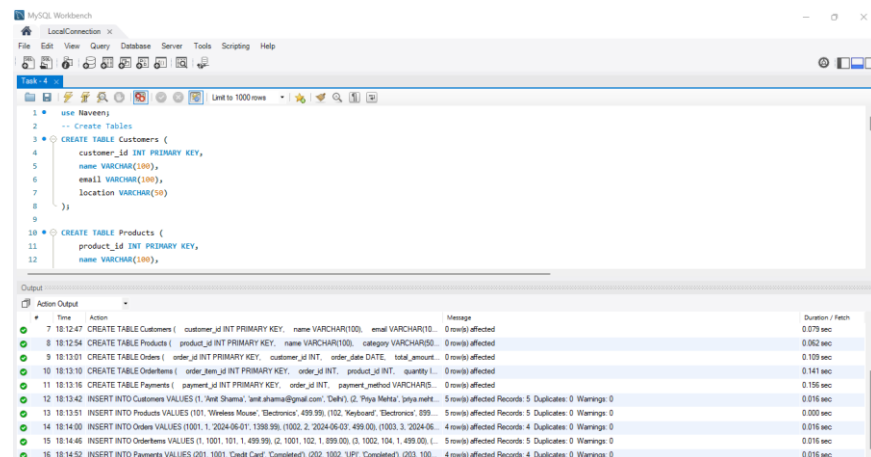
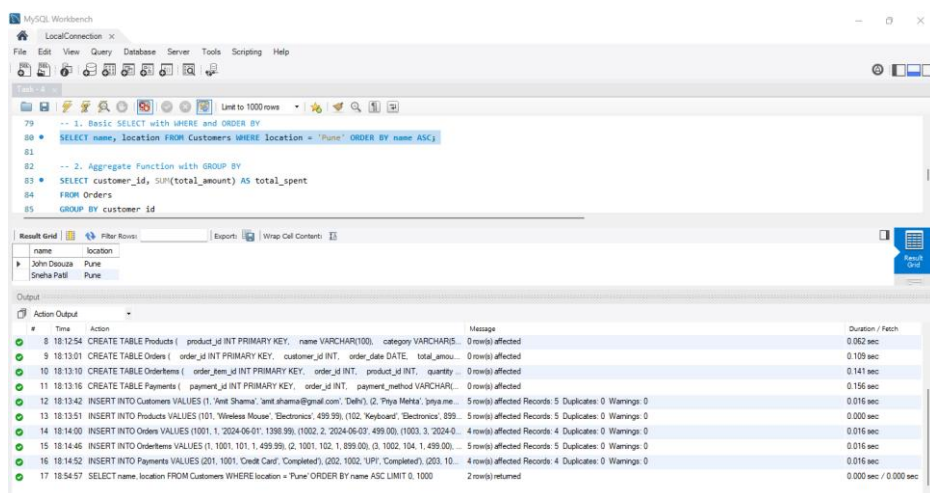


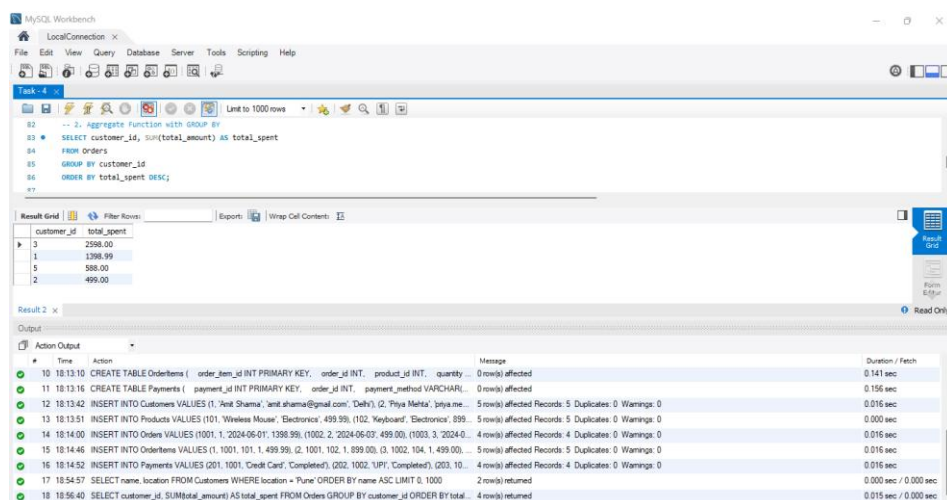
## ## Screenshots



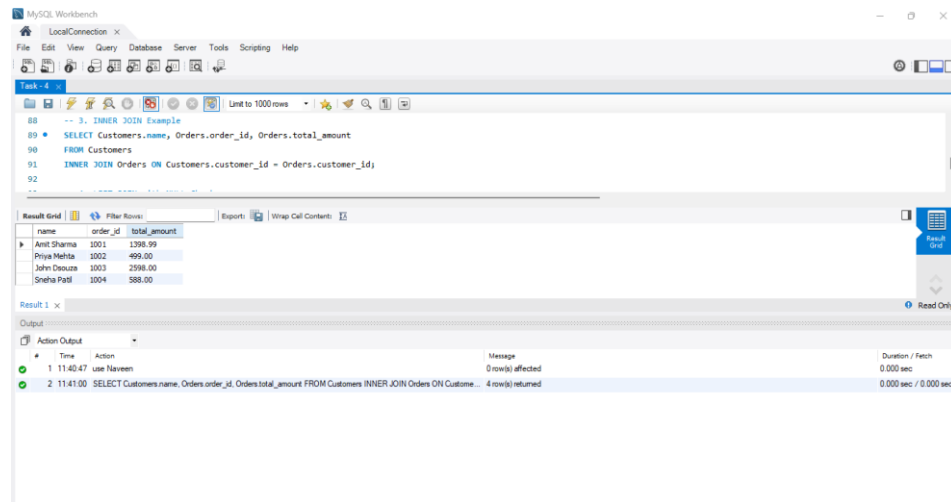
## ## 1. Basic Query – Electronics Category Products



## ### 2. Grouped Products by Category



### ## 3. Customer Orders using INNER JOIN



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
-- 3. INNER JOIN Example
88
89 SELECT Customers.name, Orders.order_id, Orders.total_amount
90 FROM Customers
91 INNER JOIN Orders ON Customers.customer_id = Orders.customer_id;
92
```

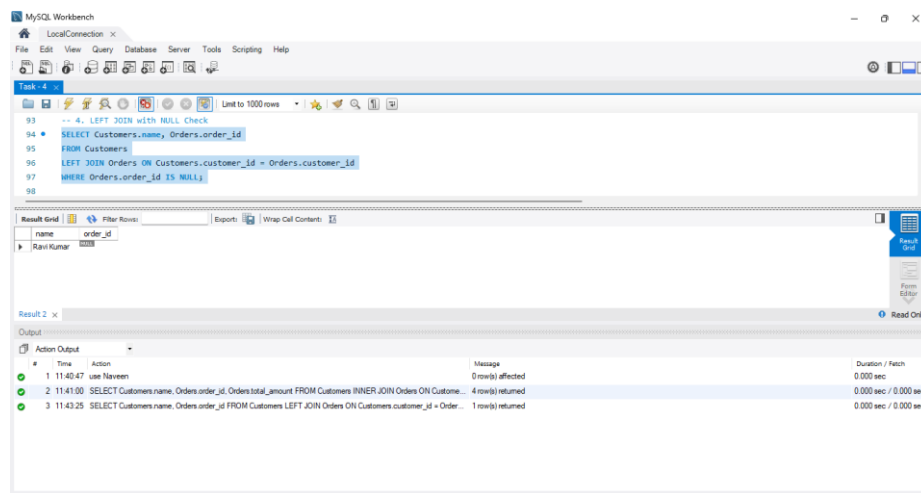
The Result Grid displays the following data:

name	order_id	total_amount
Amit Sharma	1001	1298.99
Priya Mehra	1002	499.00
John Doe	1003	2998.00
Sneha Patel	1004	588.00

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navem	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id, Orders.total_amount FROM Customers INNER JOIN Orders ON Customers.customer_id = Orders.customer_id	4 row(s) returned	0.000 sec / 0.000 sec

### ## 4. Products Above Average Price (Subquery)



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
-- 4. LEFT JOIN with NULL Check
93
94 SELECT Customers.name, Orders.order_id
95 FROM Customers
96 LEFT JOIN Orders ON Customers.customer_id = Orders.customer_id
97 WHERE Orders.order_id IS NULL;
98
```

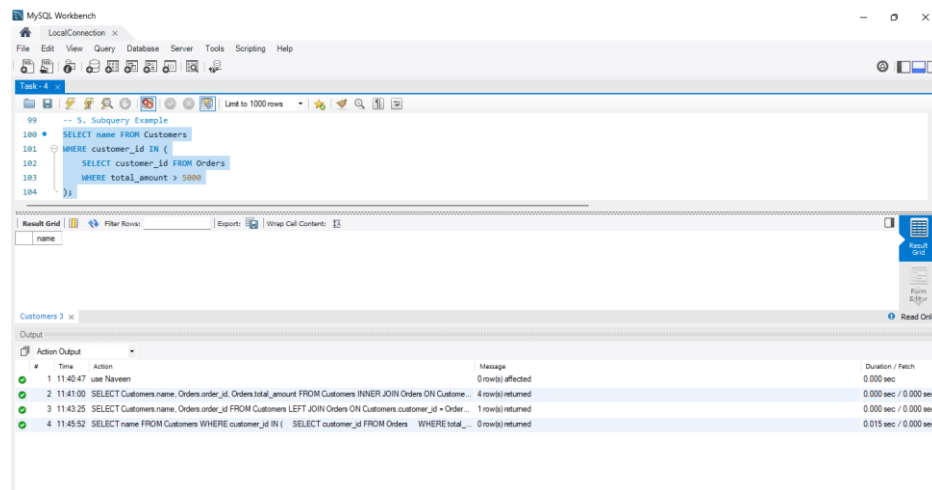
The Result Grid displays the following data:

name	order_id
Ravi Kumar	

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navem	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id FROM Customers INNER JOIN Orders ON Customers.customer_id = Orders.customer_id	4 row(s) returned	0.000 sec / 0.000 sec
3	11:43:25	SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Orders.customer_id WHERE Orders.order_id IS NULL	1 row(s) returned	0.000 sec / 0.000 sec

### ## 5. Revenue by Category (Aggregate Function)



The screenshot shows the MySQL Workbench interface. The query editor contains the following SQL code:

```
-- 5. Subquery Example
99
100 SELECT name FROM Customers
101 WHERE customer_id IN (
102 SELECT customer_id FROM Orders
103 WHERE total_amount > 5000
104 );
```

The Result Grid displays the following data:

name
------

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navem	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id, Orders.total_amount FROM Customers INNER JOIN Orders ON Customers.customer_id = Orders.customer_id	4 row(s) returned	0.000 sec / 0.000 sec
3	11:43:25	SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Orders.customer_id	1 row(s) returned	0.000 sec / 0.000 sec
4	11:45:52	SELECT name FROM Customers WHERE customer_id IN ( SELECT customer_id FROM Orders WHERE total_amount > 5000 )	0 row(s) returned	0.015 sec / 0.000 sec

## ## 6. Customer Order Summary View

The screenshot shows a MySQL Workbench window with a query editor and a result grid. The query is as follows:

```
-- 6. Aggregate with JOIN
186
187 * SELECT Products.category, SUM(OrderItems.quantity * OrderItems.price) AS total_sales
188 FROM OrderItems
189 JOIN Products ON OrderItems.product_id = Products.product_id
190 GROUP BY Products.category;
```

The result grid displays the following data:

category	total_sales
Electronics	3996.59
Clothing	499.00
Stationery	588.00

The output pane shows the execution of the query, with the following messages:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navern	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id, Orders.total_amount FROM Customers INNER JOIN Orders ON Customer...	4 row(s) returned	0.000 sec / 0.000 sec
3	11:43:25	SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Order...	1 row(s) returned	0.000 sec / 0.000 sec
4	11:45:52	SELECT name FROM Customers WHERE customer_id IN ( SELECT customer_id FROM Orders WHERE total_...	0 row(s) returned	0.015 sec / 0.000 sec
5	11:47:51	SELECT Products.category, SUM(OrderItems.quantity * OrderItems.price) AS total_sales FROM OrderItems JOIN Pro...	3 row(s) returned	0.031 sec / 0.000 sec

## ## 7. Index Created on Orders

The screenshot shows a MySQL Workbench window with a query editor. The query is as follows:

```
-- 7. Create a View for High Value Customers
112
113 * CREATE VIEW high_value_customers AS
114 SELECT customer_id, SUM(total_amount) AS total_spent
115 FROM Orders
116 GROUP BY customer_id
117 HAVING total_spent > 10000;
```

The output pane shows the execution of the query, with the following messages:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navern	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id, Orders.total_amount FROM Customers INNER JOIN Orders ON Customer...	4 row(s) returned	0.000 sec / 0.000 sec
3	11:43:25	SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Order...	1 row(s) returned	0.000 sec / 0.000 sec
4	11:45:52	SELECT name FROM Customers WHERE customer_id IN ( SELECT customer_id FROM Orders WHERE total_...	0 row(s) returned	0.015 sec / 0.000 sec
5	11:47:51	SELECT Products.category, SUM(OrderItems.quantity * OrderItems.price) AS total_sales FROM OrderItems JOIN Pro...	3 row(s) returned	0.031 sec / 0.000 sec
6	11:48:46	CREATE VIEW high_value_customers AS SELECT customer_id, SUM(total_amount) AS total_spent FROM Orders ...	0 row(s) affected	0.187 sec

The screenshot shows a MySQL Workbench window with a query editor. The query is as follows:

```
-- 8. Create Index for Optimization
120 * CREATE INDEX idx_customer_id ON Orders(customer_id);
121
```

The output pane shows the execution of the query, with the following messages:

#	Time	Action	Message	Duration / Fetch
1	11:40:47	use Navern	0 row(s) affected	0.000 sec
2	11:41:00	SELECT Customers.name, Orders.order_id, Orders.total_amount FROM Customers INNER JOIN Orders ON Customer...	4 row(s) returned	0.000 sec / 0.000 sec
3	11:43:25	SELECT Customers.name, Orders.order_id FROM Customers LEFT JOIN Orders ON Customers.customer_id = Order...	1 row(s) returned	0.000 sec / 0.000 sec
4	11:45:52	SELECT name FROM Customers WHERE customer_id IN ( SELECT customer_id FROM Orders WHERE total_...	0 row(s) returned	0.015 sec / 0.000 sec
5	11:47:51	SELECT Products.category, SUM(OrderItems.quantity * OrderItems.price) AS total_sales FROM OrderItems JOIN Pro...	3 row(s) returned	0.031 sec / 0.000 sec
6	11:48:46	CREATE VIEW high_value_customers AS SELECT customer_id, SUM(total_amount) AS total_spent FROM Orders ...	0 row(s) affected	0.187 sec
7	11:50:53	CREATE INDEX idx_customer_id ON Orders(customer_id)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.484 sec