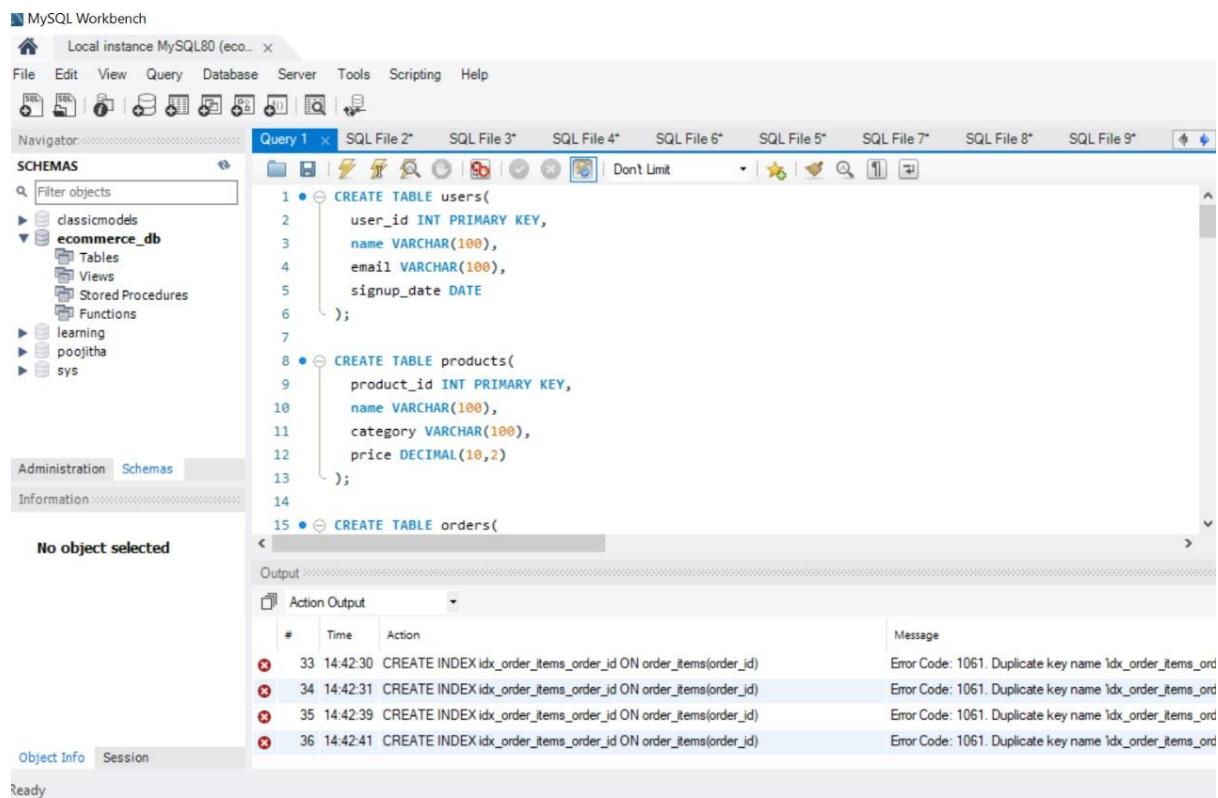


TASK 3

CREATING A TABLE AND INSERTING THE VALUES



The screenshot shows the MySQL Workbench interface with the following details:

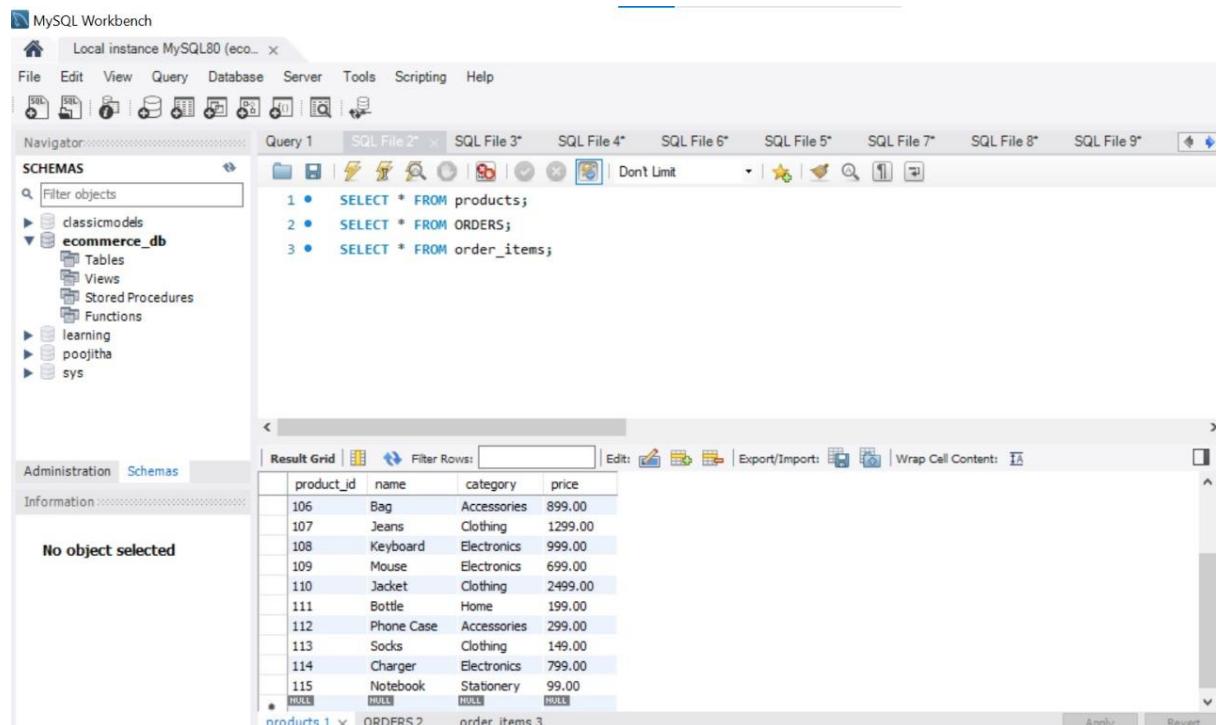
- Schemas:** The current schema is `ecommerce_db`, which contains `Tables`, `Views`, `Stored Procedures`, and `Functions`.
- Query Editor:** The `Query 1` tab displays the SQL code for creating three tables:

```
1 • CREATE TABLE users(
2     user_id INT PRIMARY KEY,
3     name VARCHAR(100),
4     email VARCHAR(100),
5     signup_date DATE
6 );
7
8 • CREATE TABLE products(
9     product_id INT PRIMARY KEY,
10    name VARCHAR(100),
11    category VARCHAR(100),
12    price DECIMAL(10,2)
13 );
14
15 • CREATE TABLE orders(

```
- Action Output:** The log shows four errors related to index creation:

#	Time	Action	Message
33	14:42:30	CREATE INDEX idx_order_items_order_id ON order_items(order_id)	Error Code: 1061. Duplicate key name `idx_order_items_ord
34	14:42:31	CREATE INDEX idx_order_items_order_id ON order_items(order_id)	Error Code: 1061. Duplicate key name `idx_order_items_ord
35	14:42:39	CREATE INDEX idx_order_items_order_id ON order_items(order_id)	Error Code: 1061. Duplicate key name `idx_order_items_ord
36	14:42:41	CREATE INDEX idx_order_items_order_id ON order_items(order_id)	Error Code: 1061. Duplicate key name `idx_order_items_ord

sql query select



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is `ecommerce_db`, which contains `Tables`, `Views`, `Stored Procedures`, and `Functions`.
- Query Editor:** The `Query 1` tab displays the following SQL queries:

```
1 • SELECT * FROM products;
2 • SELECT * FROM ORDERS;
3 • SELECT * FROM order_items;
```
- Result Grid:** The results of the third query are displayed in a grid:

product_id	name	category	price
106	Bag	Accessories	899.00
107	Jeans	Clothing	1299.00
108	Keyboard	Electronics	999.00
109	Mouse	Electronics	699.00
110	Jacket	Clothing	2499.00
111	Bottle	Home	199.00
112	Phone Case	Accessories	299.00
113	Socks	Clothing	149.00
114	Charger	Electronics	799.00
115	Notebook	Stationery	99.00
HULL	HULL	HULL	HULL

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 1 SQL File 2* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9*

```
1 • SELECT product_id, name, category
2 FROM products;
```

Administration Schemas

No object selected

Result Grid

product_id	name	category
101	Laptop	Electronics
102	Headphones	Electronics
103	T-shirt	Clothing
104	Shoes	Footwear
105	Smartwatch	Electronics
106	Bag	Accessories
107	Jean	Clothing
108	Keyboard	Electronics
109	Mouse	Electronics
110	Jacket	Clothing
111	Bottle	Home

products 1 x Apply Revert

SQL QUERY WHERE

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 1 SQL File 2* SQL File 3* SQL File 4* SQL File 5* SQL File 6* SQL File 7* SQL File 8* SQL File 9*

```
1 • SELECT *
2 FROM products
3 WHERE category = 'Electronics';
```

Administration Schemas

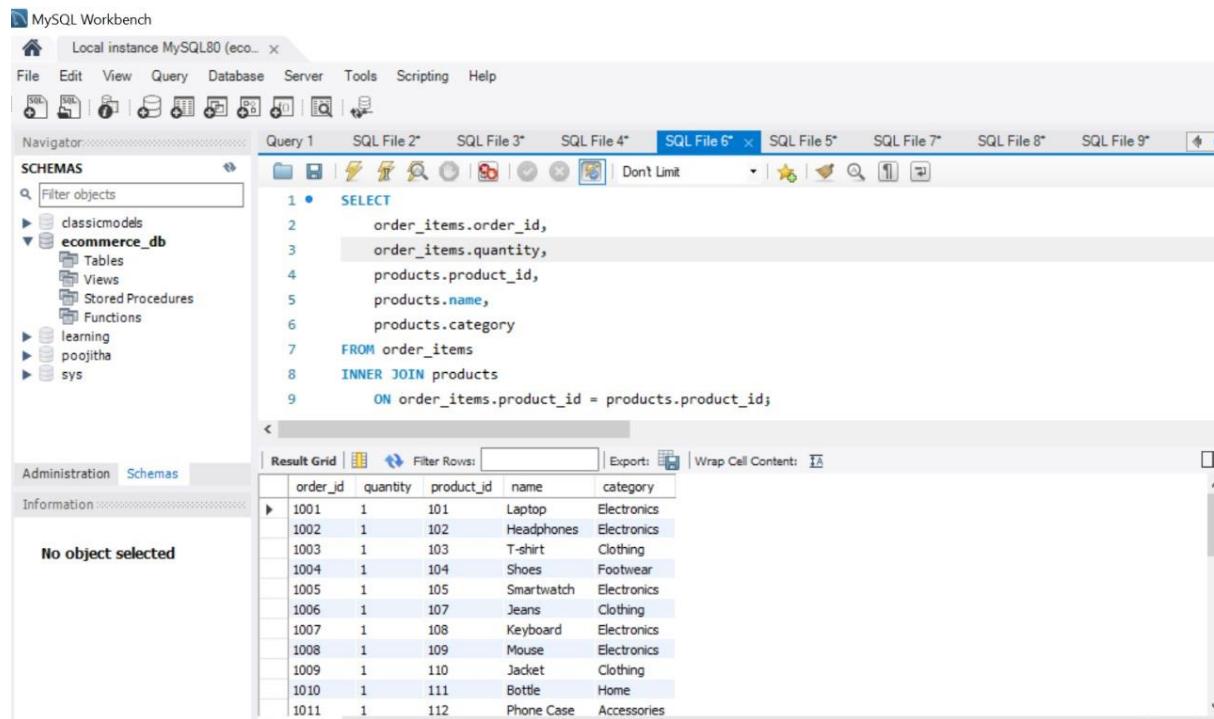
No object selected

Result Grid

product_id	name	category	price
101	Laptop	Electronics	55000.00
102	Headphones	Electronics	2500.00
105	Smartwatch	Electronics	3999.00
108	Keyboard	Electronics	999.00
109	Mouse	Electronics	699.00
114	Charger	Electronics	799.00
*	NULL	NULL	NULL

products 1 x Apply Revert

SQL QUERY INNER JOIN



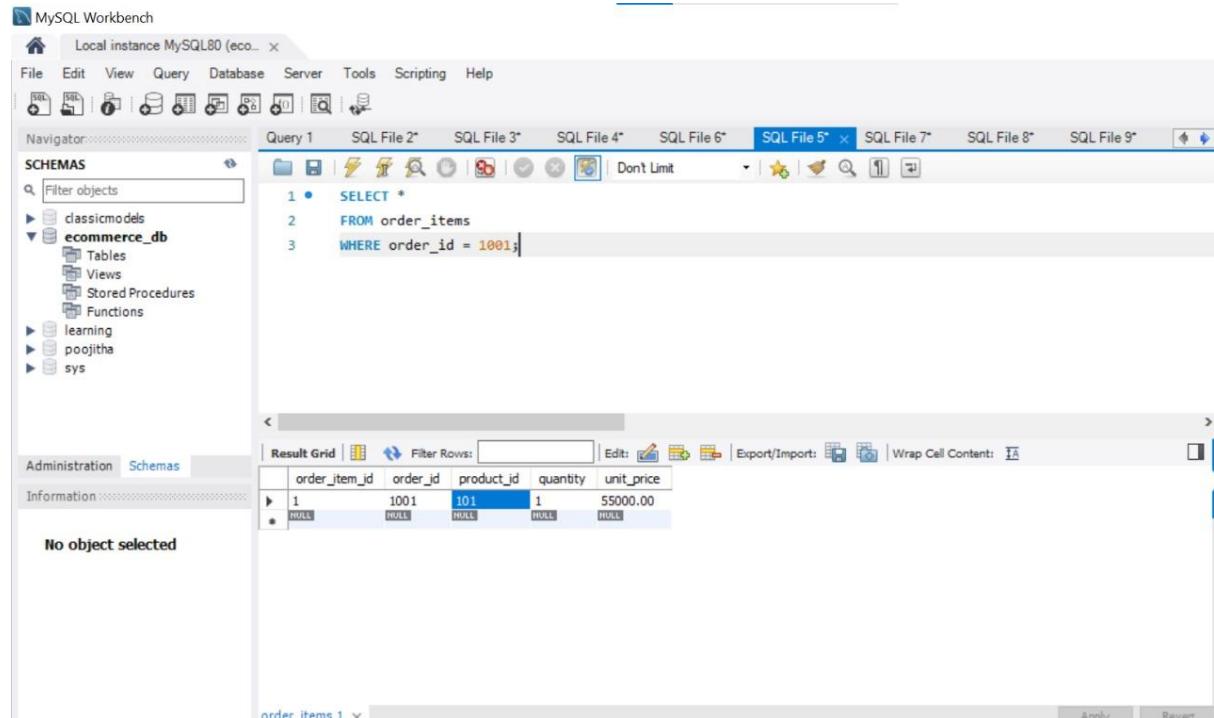
The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** ecommerce_db is selected.
- Query Editor:** Contains the following SQL code:

```
1 • SELECT
2     order_items.order_id,
3         order_items.quantity,
4         products.product_id,
5         products.name,
6         products.category
7     FROM order_items
8     INNER JOIN products
9         ON order_items.product_id = products.product_id;
```
- Result Grid:** Displays the results of the query, showing 11 rows of data from the order_items and products tables.

order_id	quantity	product_id	name	category
1001	1	101	Laptop	Electronics
1002	1	102	Headphones	Electronics
1003	1	103	T-shirt	Clothing
1004	1	104	Shoes	Footwear
1005	1	105	Smartwatch	Electronics
1006	1	107	Jeans	Clothing
1007	1	108	Keyboard	Electronics
1008	1	109	Mouse	Electronics
1009	1	110	Jacket	Clothing
1010	1	111	Bottle	Home
1011	1	112	Phone Case	Accessories

SQL QUERY LEFT JOIN



The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** ecommerce_db is selected.
- Query Editor:** Contains the following SQL code:

```
1 • SELECT *
2     FROM order_items
3     WHERE order_id = 1001;
```
- Result Grid:** Displays the results of the query, showing 1 row of data from the order_items table.

order_item_id	order_id	product_id	quantity	unit_price
1	1001	101	1	55000.00

SQL QUERY RIGHT JOIN

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
1 • SELECT
2     products.product_id,
3     products.name,
4     order_items.order_id,
5     order_items.quantity
6 FROM products
7 LEFT JOIN order_items
8     ON products.product_id = order_items.product_id;
```

The results grid displays the following data:

product_id	name	order_id	quantity
101	Laptop	1014	1
101	Laptop	1001	1
102	Headphones	1015	1
102	Headphones	1002	1
103	T-shirt	1021	1
103	T-shirt	1013	1
103	T-shirt	1003	1
104	Shoes	1025	1
104	Shoes	1004	1
105	Smartwatch	1029	1
105	Smartwatch	1005	1

SQL FOR SUBQUERY

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
1 • SELECT
2     order_items.order_id,
3     order_items.quantity,
4     products.product_id,
5     products.name
6 FROM order_items
7 RIGHT JOIN products
8     ON order_items.product_id = products.product_id;
```

The results grid displays the following data:

order_id	quantity	product_id	name
1014	1	101	Laptop
1001	1	101	Laptop
1015	1	102	Headphones
1002	1	102	Headphones
1021	1	103	T-shirt
1013	1	103	T-shirt
1003	1	103	T-shirt
1025	1	104	Shoes
1004	1	104	Shoes
1029	1	105	Smartwatch
1005	1	105	Smartwatch

SQL QUERY FOR SUM

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 1 SQL File 2* SQL File 3* SQL File 4* SQL File 6* SQL File 5* SQL File 7* SQL File 8* SQL File 9* SQL File 10*

```

1 •   SELECT *
2     FROM products
3     WHERE product_id IN (
4         SELECT product_id
5           FROM order_items
6     );

```

Administration Schemas

No object selected

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	product_id	name	category	price
▶	101	Laptop	Electronics	55000.00
	102	Headphones	Electronics	2500.00
	103	T-shirt	Clothing	799.00
	104	Shoes	Footwear	1999.00
	105	Smartwatch	Electronics	3999.00
	107	Jeans	Clothing	1299.00
	108	Keyboard	Electronics	999.00
	109	Mouse	Electronics	699.00
	110	Jacket	Clothing	2499.00
	111	Bottle	Home	199.00
	112	Phone Case	Accessories	299.00

products 1 × Apply Revert

SQL QUERY FOR AVG

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Query 6* SQL File 5* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13*

```

1 •   SELECT
2       order_id,
3       (SELECT SUM(quantity)
4        FROM order_items oi
5        WHERE oi.order_id = o.order_id) AS total_items
6   FROM orders o;

```

Administration Schemas

No object selected

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

	order_id	total_items
▶	1001	1
	1002	1
	1003	1
	1004	1
	1005	1
	1006	1
	1007	1
	1008	1
	1009	1
	1010	1
	1011	1

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Schemas

classicmodels
ecommerce_db
Tables
Views
Stored Procedures
Functions
learning
poojitha
sys

ile 6* SQL File 5* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 11* SQL File 12* SQL File 13*

```
1 • SELECT *
2   FROM products
3   WHERE price > (
4     SELECT AVG(price)
5       FROM products
6   );
```

Administration Schemas Information

No object selected

Result Grid | Filter Rows: [] Edit: [] Export/Import: [] Wrap Cell Content: []

	product_id	name	category	price
▶	101	Laptop	Electronics	55000.00
*	NULL	NULL	NULL	NULL

products 1 × Apply Revert

The screenshot shows the MySQL Workbench interface. In the top navigation bar, the database 'Local instance MySQL80 (ecommerce_db)' is selected. The 'Query' tab is active. The 'Navigator' pane on the left lists databases: classicmodels, ecommerce_db (selected), learning, poojitha, and sys. Under ecommerce_db, it shows Tables, Views, Stored Procedures, and Functions. The main area displays a query editor with the following SQL code:

```
1 • SELECT *
2   FROM products
3   WHERE price > (
4     SELECT AVG(price)
5       FROM products
6   );
```

Below the query editor is a results grid titled 'Result Grid'. It has columns: product_id, name, category, and price. One row is visible, showing product_id 101, name Laptop, category Electronics, and price 55000.00. The bottom status bar shows the table name 'products 1' and buttons for 'Apply' and 'Revert'.