# Certura Internship Program(Database Handling)

## Task No: 01

**Submitted by :**

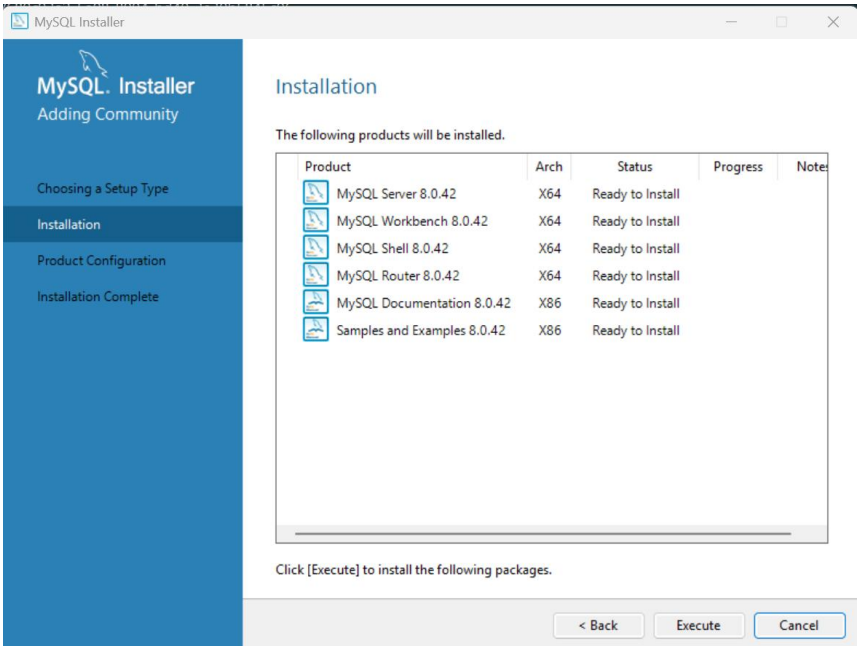
Huzaifa Waqar.

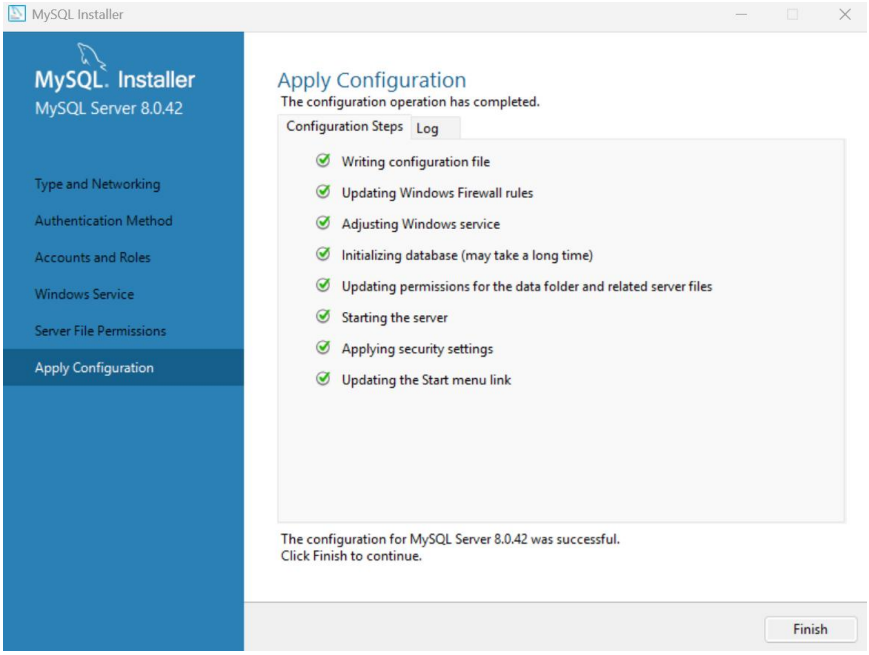
**Submitted to:**

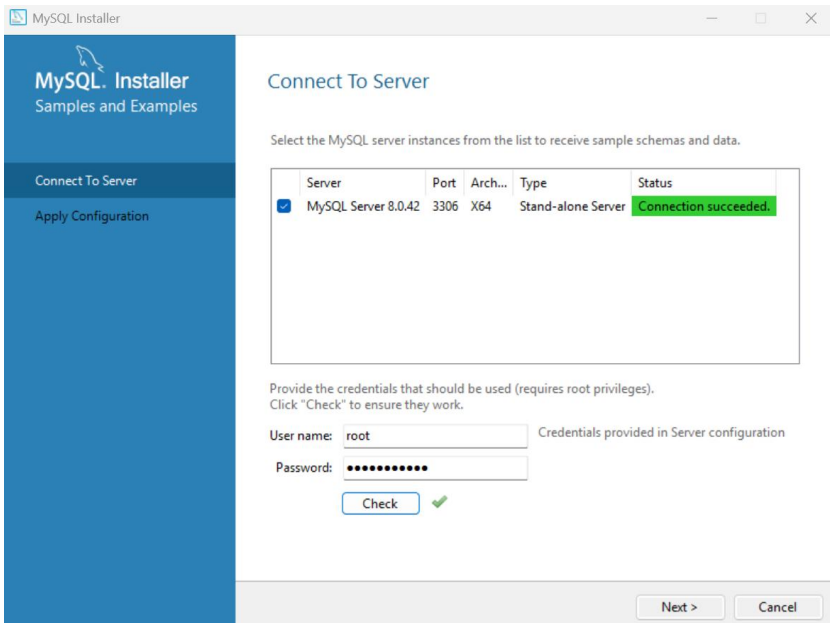
Certura.

## ✅ TOOLS USED

1. **MySQL Server**
   - This is the database engine.
2. **MySQL Workbench**
   - This is the GUI tool where you write queries and manage your database.






---

## □ TASK: 1

Got it — you've installed MySQL and set up the connection, great! Now you're asking for **detailed steps** on how to **solve the e-commerce database task using MySQL Workbench**, right?
I'll guide you in a very detailed way step-by-step as if you're doing it practically. Let's go:

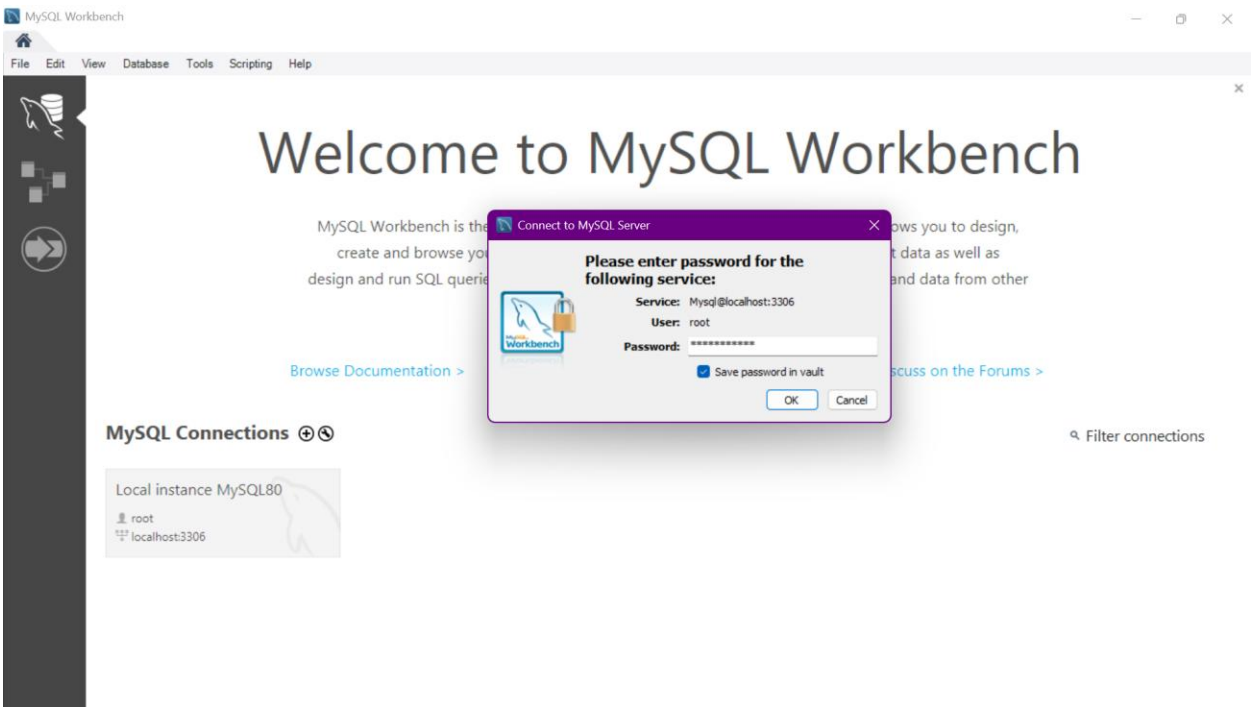
---

## 🔑 How to Solve the Task (Detailed Steps)

---

### Step 1: Open MySQL Workbench and Connect

- Open **MySQL Workbench**.
- Click on your **saved connection** (e.g., `Local instance MySQL80` or whatever name you gave).
- You'll enter the **SQL Editor** window.



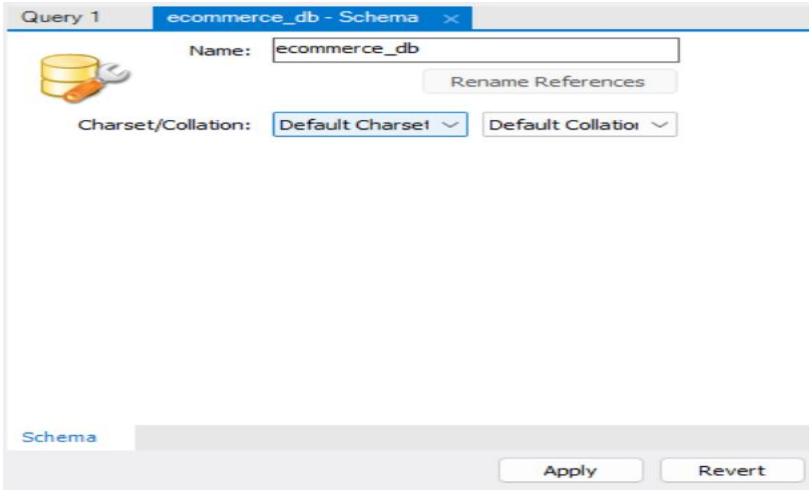

---

## Step 2: Create a New Schema (Database)

### In Workbench:

- Click the **"Schemas" tab** on the left side.
- Right-click and choose **"Create Schema"**.
- Name it something like: `ecommerce_db`.
- Click **Apply → Apply → Finish**.

✅ Your database (`ecommerce_db`) is created.

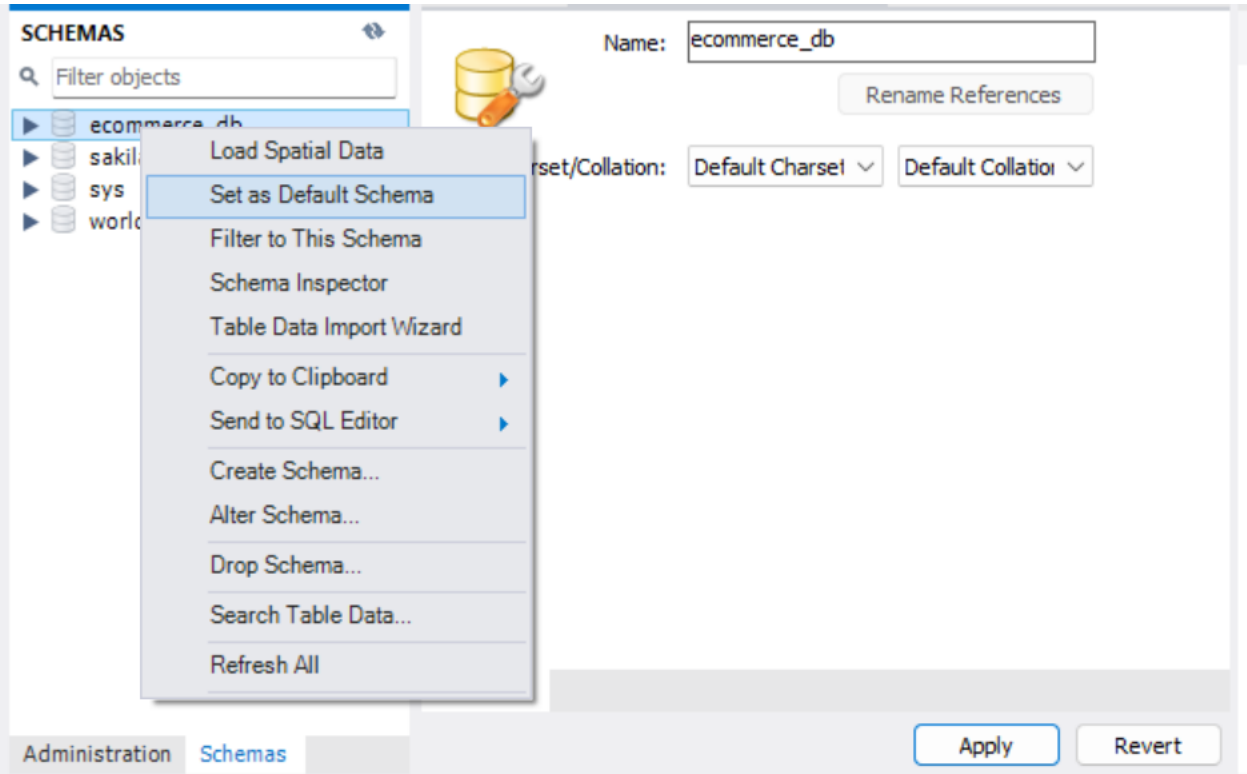


---

## Step 3: Write SQL Commands to Create Tables

☞ Now in the query editor, **choose your schema** (`ecommerce_db`) by right-clicking it and choosing "**Set as Default Schema**".

Now design the tables:




---

## Step 4: Identify Tables

From the task, basic tables you need:

1. **Users**
2. **Products**
3. **Orders**
4. **Order_Details** (optional but better — because one order can have multiple products)

---

## Step 5: Create Tables with Relationships

Here's the SQL for each:

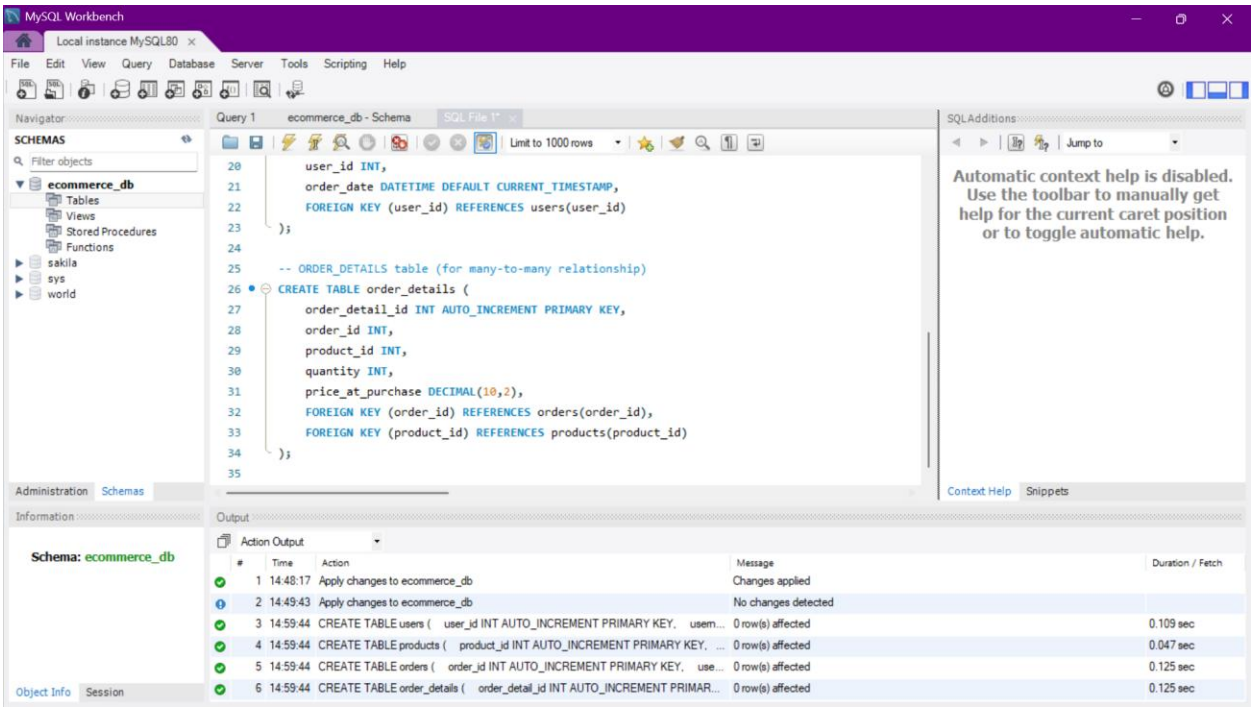
```
-- USERS table
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(100) NOT NULL
);

-- PRODUCTS table
CREATE TABLE products (
    product_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    stock_quantity INT NOT NULL
);

-- ORDERS table
CREATE TABLE orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- ORDER_DETAILS table (for many-to-many relationship)
CREATE TABLE order_details (
    order_detail_id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT,
    product_id INT,
    quantity INT,
    price_at_purchase DECIMAL(10,2),
    FOREIGN KEY (order_id) REFERENCES orders(order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id)
);
```

---

## Step 6: Execute the SQL

- Paste the above code into the **SQL Query window**.
- Click the ⚡ **Execute (lightning bolt)** button to run.

✅ Tables and relationships will be created!



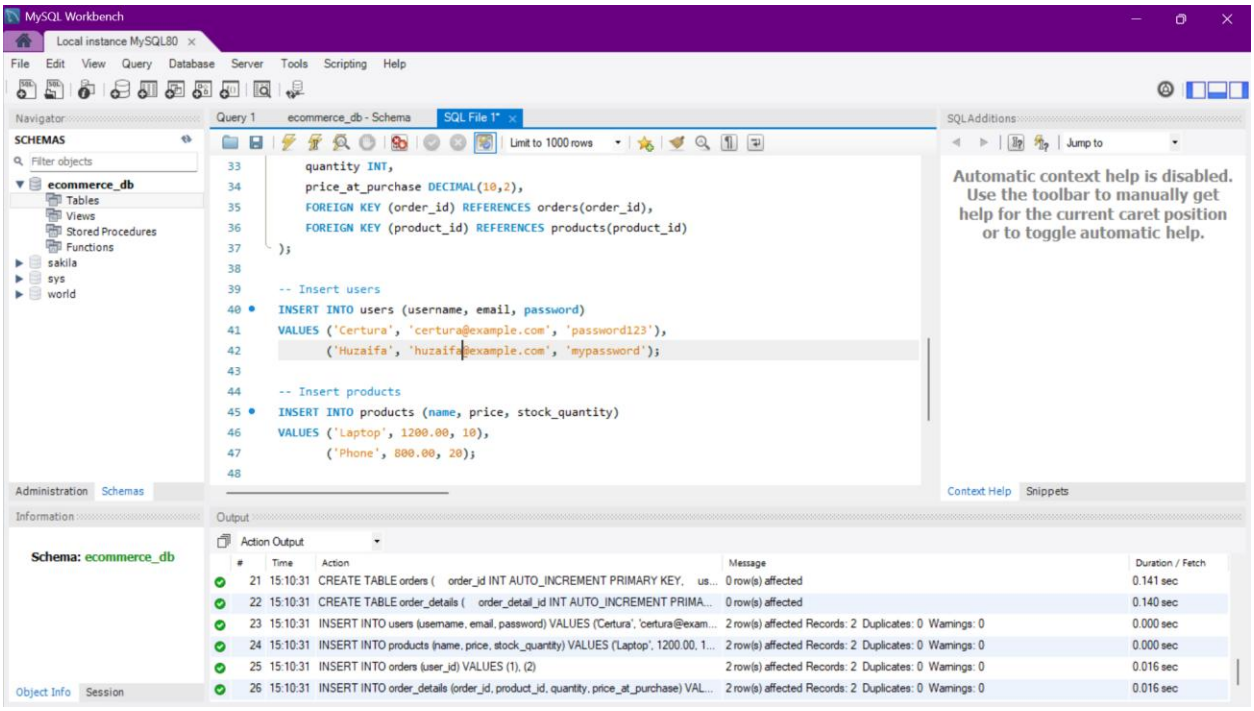

---

## Step 7: Insert Some Sample Data

To **test integrity**, insert a few sample rows:

```
-- Insert users
INSERT INTO users (username, email, password)
VALUES ('certura', 'certura@example.com', 'password123'),
       ('huzaifa', 'huzaifa@example.com', 'password123');

-- Insert products
INSERT INTO products (name, price, stock_quantity)
VALUES ('Laptop', 1200.00, 10),
       ('Phone', 800.00, 20);

-- Insert orders
INSERT INTO orders (user_id)
VALUES (1), (2);

-- Insert order details
INSERT INTO order_details (order_id, product_id, quantity, price_at_purchase)
VALUES (1, 1, 1, 1200.00),
       (2, 2, 2, 800.00);
```

⚡ Execute this too.




---

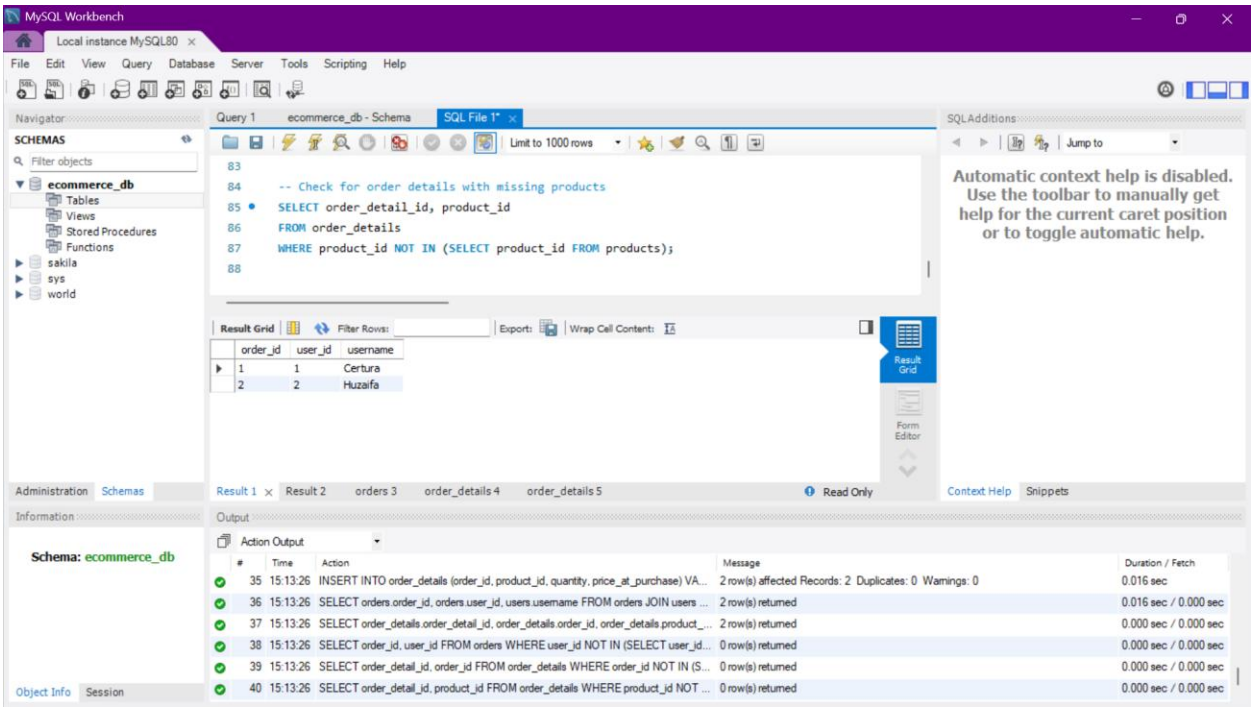
## Step 8: Test the Data

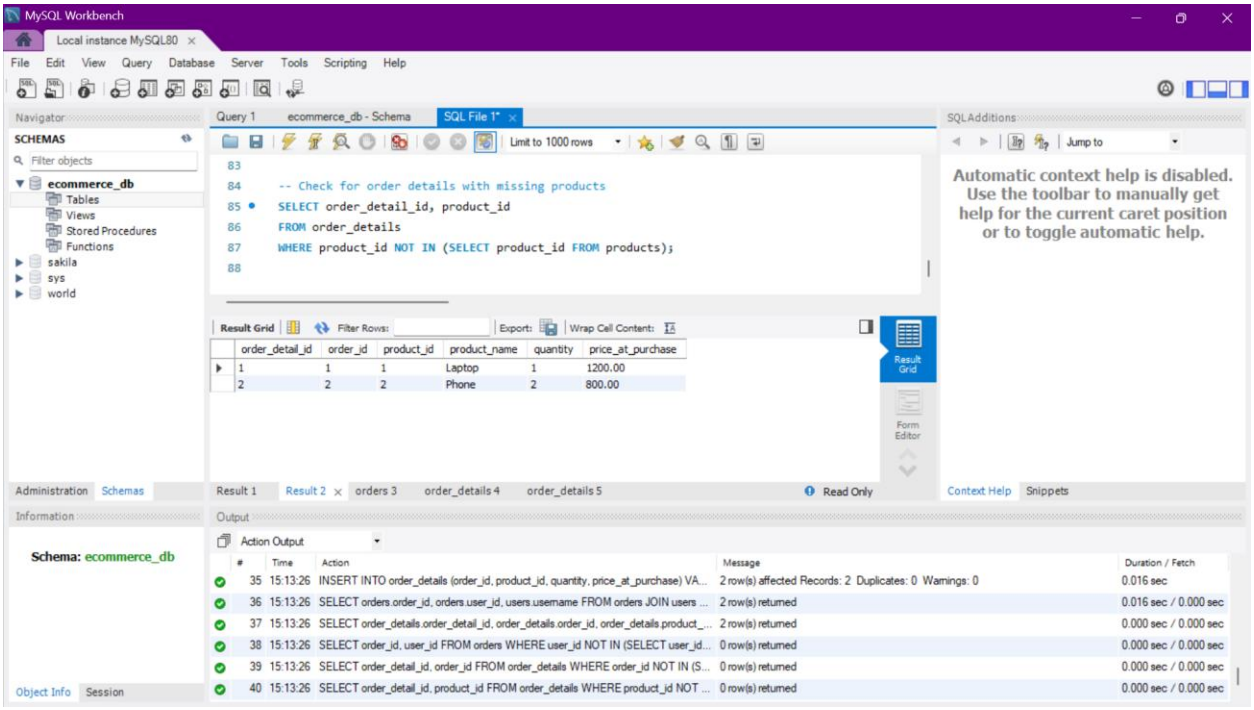
Use **SELECT queries** to check:

```
SELECT * FROM users;
SELECT * FROM products;
SELECT * FROM orders;
SELECT * FROM order_details;
```

✅ Confirm that:

- Orders are linked to users.
- Order details correctly link orders and products.
- Data integrity is maintained (no missing foreign keys).


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ecommerce_db
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Query 1
ecommerce_db - Schema
SQL File 1*
Limit to 1000 rows
83
84 -- Check for order details with missing products
85 SELECT order_detail_id, product_id
86 FROM order_details
87 WHERE product_id NOT IN (SELECT product_id FROM products);
88
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
order_id user_id username
1 1 Certura
2 2 Huzaifa
Result Grid
Form Editor
Administration Schemas
Result 1
Result 2
orders 3
order_details 4
order_details 5
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help
Snippets
Information
Schema: ecommerce_db
Object Info
Session
Output
Action Output
# Time Action Message Duration / Fetch
35 15:13:26 INSERT INTO order_details (order_id, product_id, quantity, price_at_purchase) VA... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.016 sec
36 15:13:26 SELECT orders.order_id, orders.user_id, users.username FROM orders JOIN users ... 2 row(s) returned 0.016 sec / 0.000 sec
37 15:13:26 SELECT order_details.order_detail_id, order_details.order_id, order_details.product_... 2 row(s) returned 0.000 sec / 0.000 sec
38 15:13:26 SELECT order_id, user_id FROM orders WHERE user_id NOT IN (SELECT user_id... 0 row(s) returned 0.000 sec / 0.000 sec
39 15:13:26 SELECT order_detail_id, order_id FROM order_details WHERE order_id NOT IN (S... 0 row(s) returned 0.000 sec / 0.000 sec
40 15:13:26 SELECT order_detail_id, product_id FROM order_details WHERE product_id NOT ... 0 row(s) returned 0.000 sec / 0.000 sec


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
ecommerce_db
Tables
Views
Stored Procedures
Functions
sakila
sys
world
Query 1
ecommerce_db - Schema
SQL File 1*
Limit to 1000 rows
83
84 -- Check for order details with missing products
85 SELECT order_detail_id, product_id
86 FROM order_details
87 WHERE product_id NOT IN (SELECT product_id FROM products);
88
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
order_detail_id order_id product_id product_name quantity price_at_purchase
1 1 1 Laptop 1 1200.00
2 2 2 Phone 2 800.00
Result Grid
Form Editor
Administration Schemas
Result 1
Result 2
orders 3
order_details 4
order_details 5
Read Only
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help
Snippets
Information
Schema: ecommerce_db
Object Info
Session
Output
Action Output
# Time Action Message Duration / Fetch
35 15:13:26 INSERT INTO order_details (order_id, product_id, quantity, price_at_purchase) VA... 2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0 0.016 sec
36 15:13:26 SELECT orders.order_id, orders.user_id, users.username FROM orders JOIN users ... 2 row(s) returned 0.016 sec / 0.000 sec
37 15:13:26 SELECT order_details.order_detail_id, order_details.order_id, order_details.product_... 2 row(s) returned 0.000 sec / 0.000 sec
38 15:13:26 SELECT order_id, user_id FROM orders WHERE user_id NOT IN (SELECT user_id... 0 row(s) returned 0.000 sec / 0.000 sec
39 15:13:26 SELECT order_detail_id, order_id FROM order_details WHERE order_id NOT IN (S... 0 row(s) returned 0.000 sec / 0.000 sec
40 15:13:26 SELECT order_detail_id, product_id FROM order_details WHERE product_id NOT ... 0 row(s) returned 0.000 sec / 0.000 sec

## Final Summary (In Short)

| Step | Task |
|---|---|
| 1 | Open Workbench and connect |
| 2 | Create a schema |
| 3 | Identify tables |
| 4 | Write SQL to create tables with FOREIGN KEYS |
| 5 | Insert sample data |
| 6 | Test using SELECT queries |

---