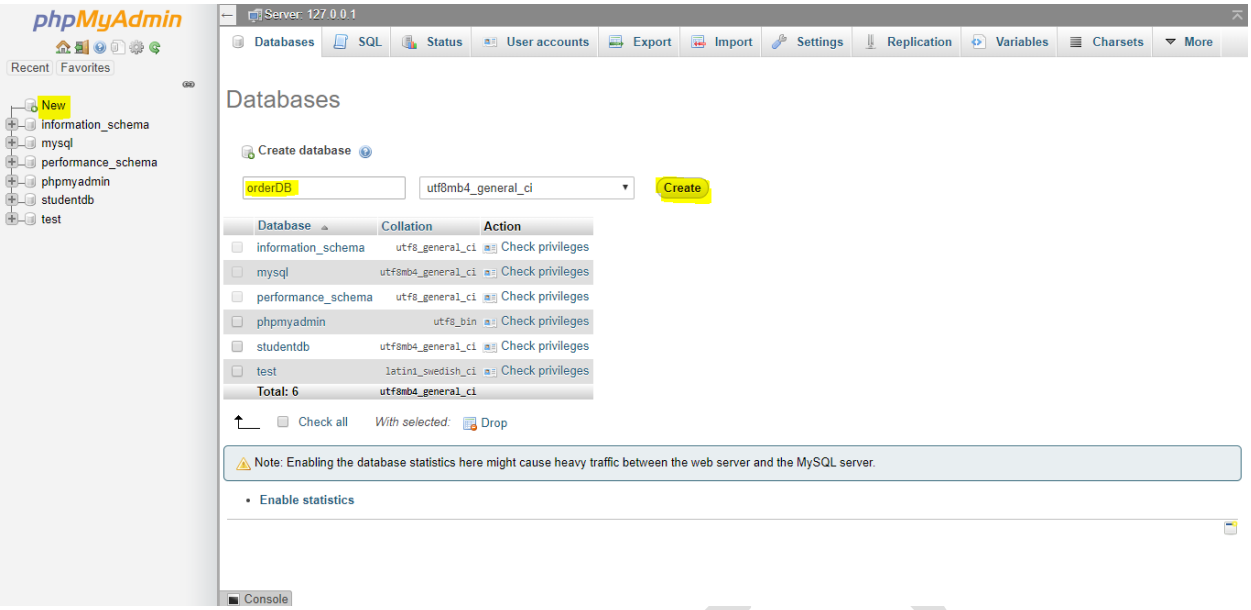
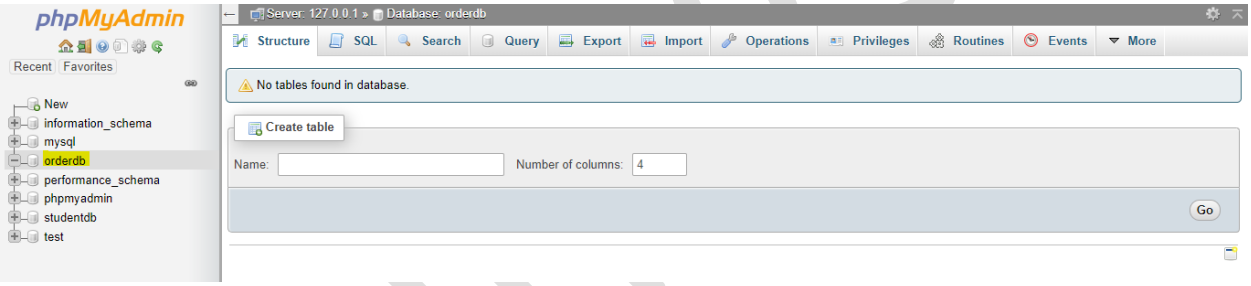
Create a new database named orderDB in PHP MYSQL.

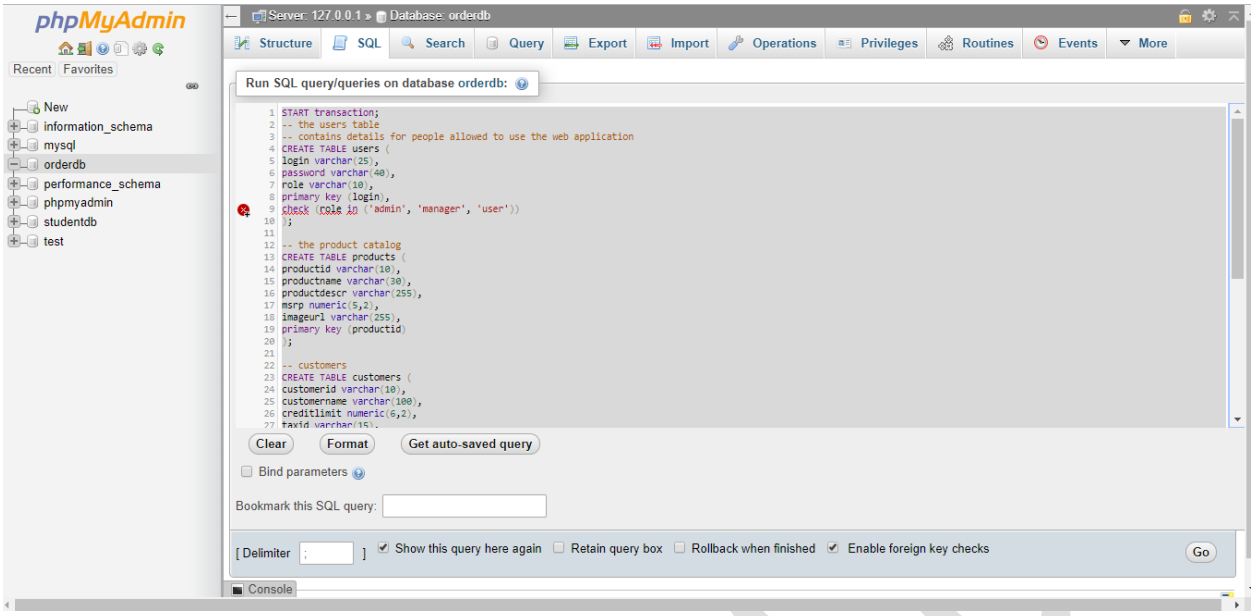


OrderDB has been created



Now Click on SQL and copy/paste the table creation (database schema) commands in the text-area.

In MYSQL we have used START instead of BEGIN transaction. I have also changed the foreign key statements in the CREATE table commands.



START transaction;

-- the users table

-- contains details for people allowed to use the web application

CREATE TABLE users (

login varchar(25),

password varchar(40),

role varchar(10),

primary key (login),

check (role in ('admin', 'manager', 'user'))

);

-- the product catalog

CREATE TABLE products (

productid varchar(10),

productname varchar(30),

productdescr varchar(255),

```
msrp numeric(5,2),
imageurl varchar(255),
primary key (productid)
);

-- customers
CREATE TABLE customers (
customerid varchar(10),
customername varchar(100),
creditlimit numeric(6,2),
taxid varchar(15),
primary key (customerid)
);

CREATE TABLE orders (
orderid varchar(10),
customerid varchar(10),
status varchar(10),
check (status in ('new', 'prepared', 'shipped', 'billed', 'paid')),
foreign key (customerid) references customers(customerid),
primary key (orderid)
);

CREATE TABLE lineitems (
orderid varchar(10),
```

WritingSol

productid varchar(10),

numprods numeric(3,0),

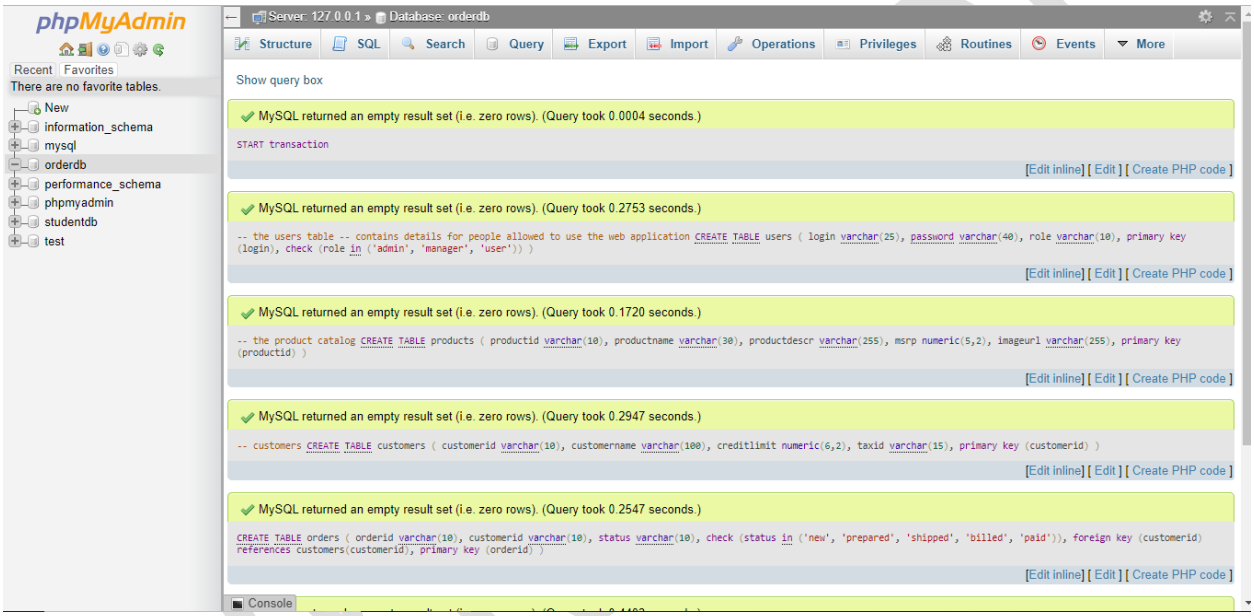
prodprice numeric(5,2),

foreign key (productid) references products(productid),

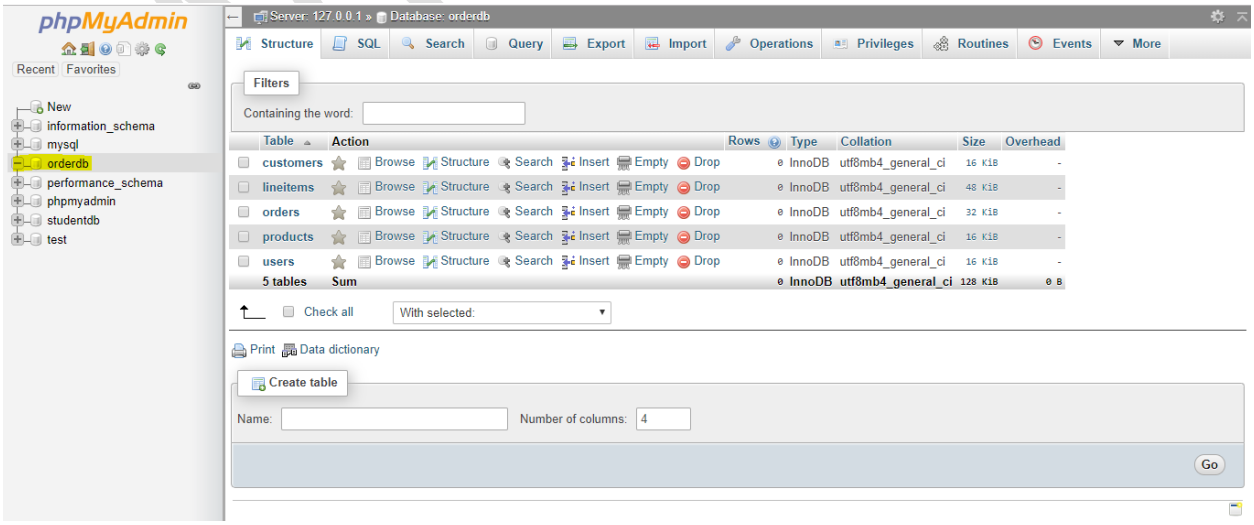
foreign key (orderid) references orders(orderid)

);

commit;



orderDB structure is here after creation commands executed successfully.



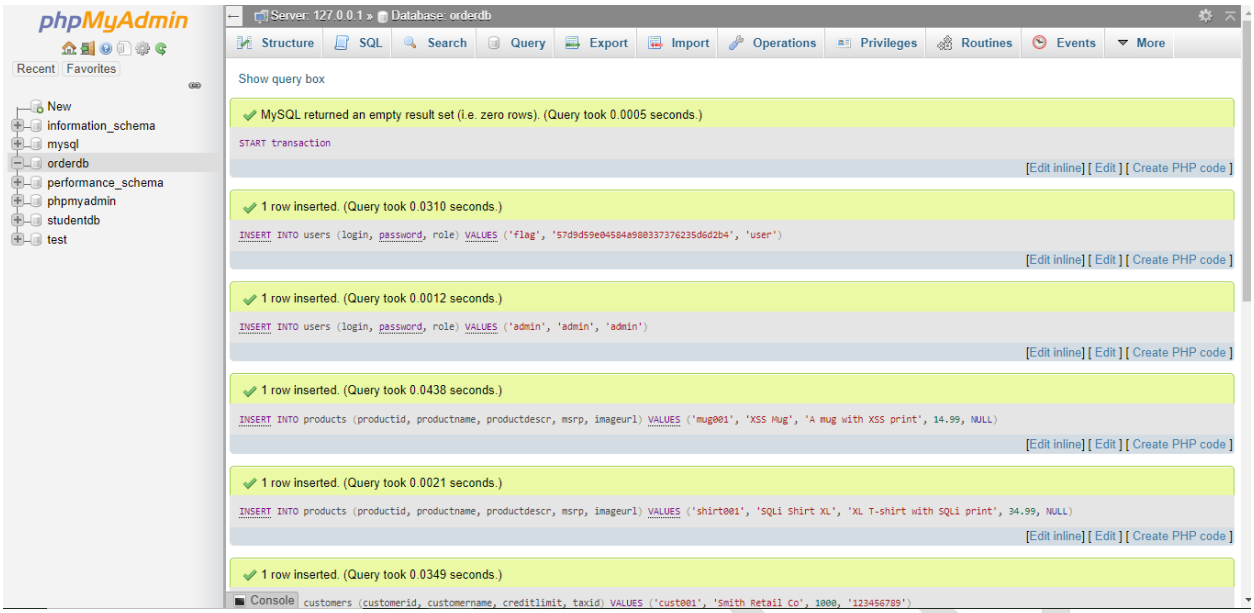
To execute the INSERT statement I have copy/paste the below commands:

```
START transaction;

INSERT INTO users

(login, password, role)

VALUES ('flag', '57d9d59e04584a980337376235d6d2b4', 'user');


INSERT INTO users

(login, password, role)

VALUES ('admin', 'admin', 'admin');


INSERT INTO products

(productid, productname, productdescr, msrp, imageurl)

VALUES

('mug001', 'XSS Mug', 'A mug with XSS print', 14.99, NULL);


INSERT INTO products

(productid, productname, productdescr, msrp, imageurl)

VALUES

('shirt001', 'SQLi Shirt XL', 'XL T-shirt with SQLi print', 34.99, NULL);


INSERT INTO customers

(customerid, customername, creditlimit, taxid)

VALUES

('cust001', 'Smith Retail Co', 1000, '123456789');
```

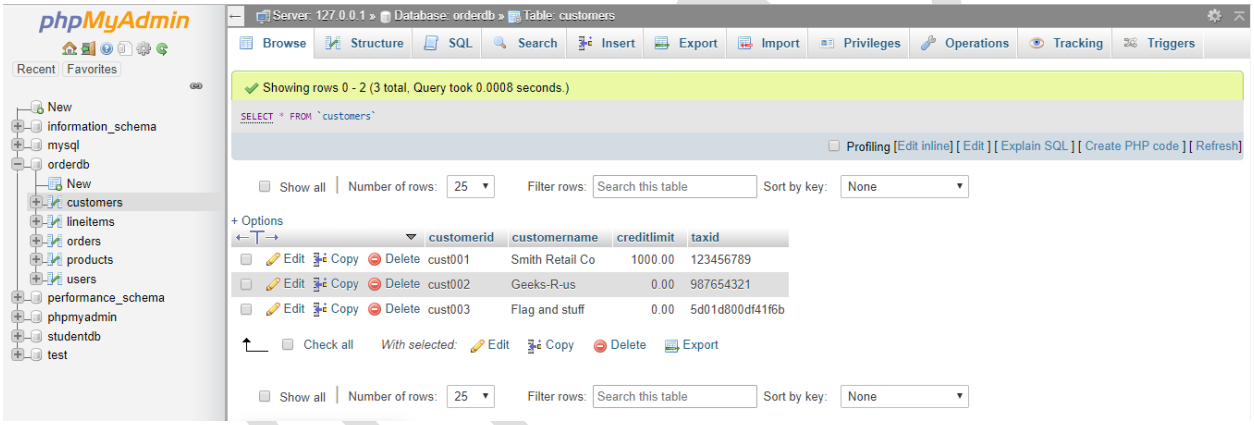
WritingSol

```
INSERT INTO customers
(customerid, customername, creditlimit, taxid)
VALUES
('cust002', 'Geeks-R-us', 0, '987654321');

INSERT INTO customers
(customerid, customername, creditlimit, taxid)
VALUES
('cust003', 'Flag and stuff', 0, '5d01d800df41f6b');

INSERT INTO orders
(orderid, customerid, status)
VALUES
('201700001', 'cust002', 'prepared');

INSERT INTO lineitems
(orderid, productid, numprods, prodprice)
VALUES
('201700001', 'mug001', 5, 12.50);

INSERT INTO lineitems
(orderid, productid, numprods, prodprice)
VALUES
('201700001', 'shirt001', 10, 24.95);
commit;
```

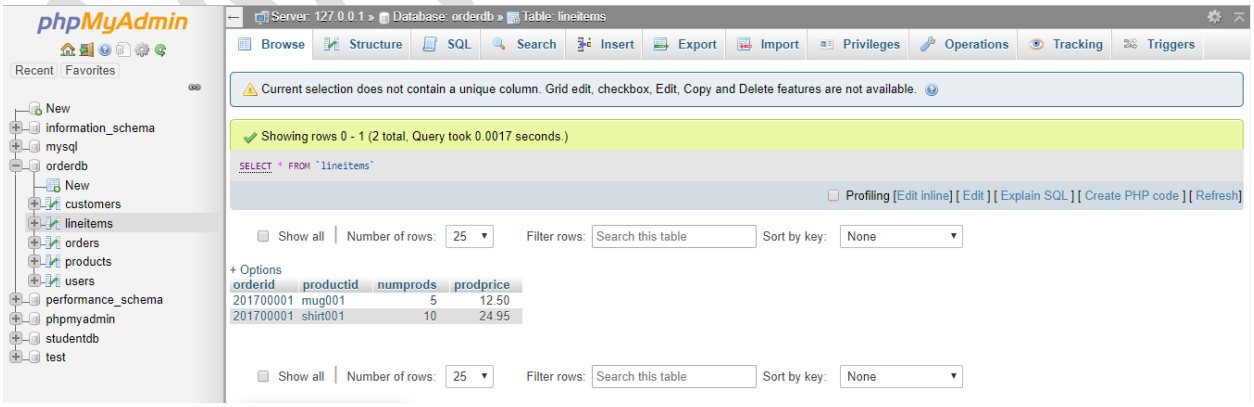
WritingSol



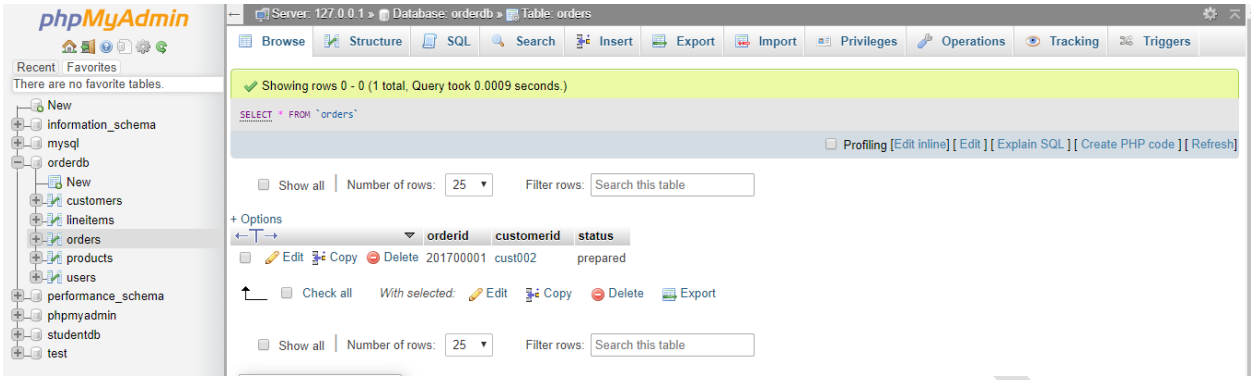
CUSTOMERS TABLE:



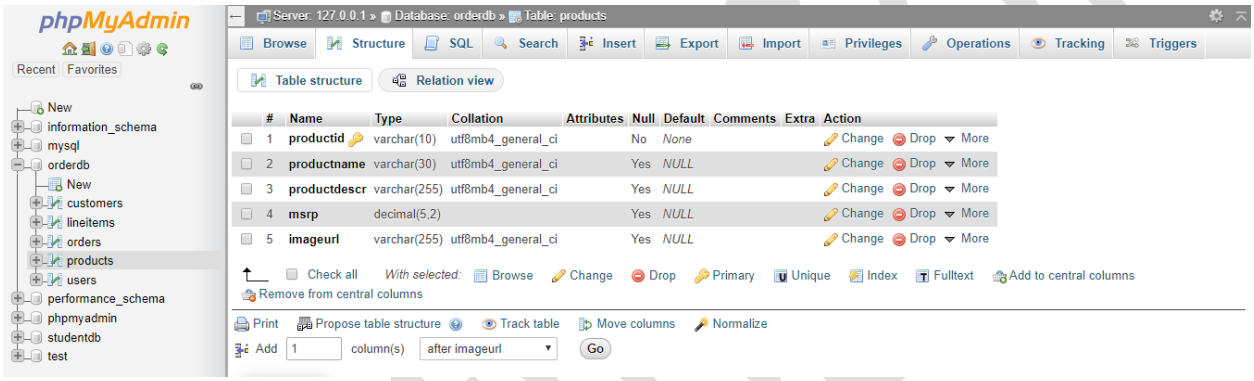
LINEITEMS TABLE:



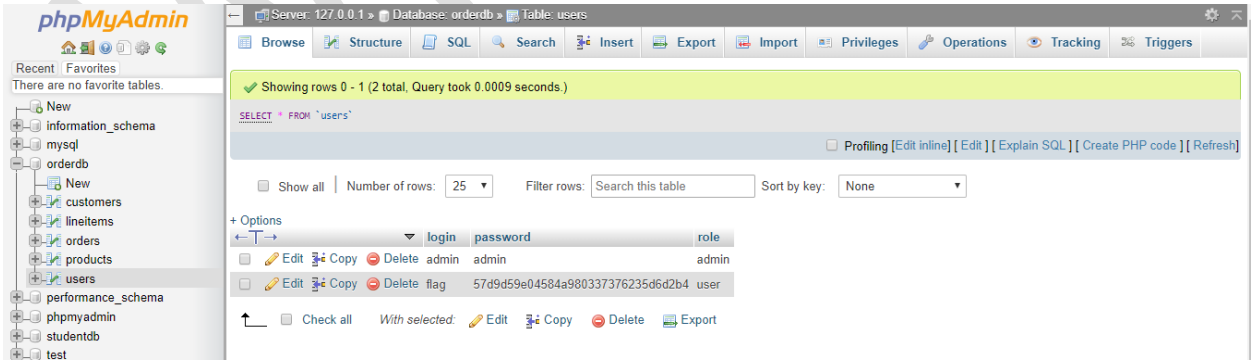
ORDERS TABLE:



PRODUCTS TABLE:



USERS TABLE:



1. Create a view that contains sufficient data to, given an order number, return the name of the customer, a list of the names, amounts, and price of each item. The view cannot return any other information that what is specified here.

CREATE A VIEW USING BELOW COMMAND:

CREATE VIEW ORDER_DETAILS AS

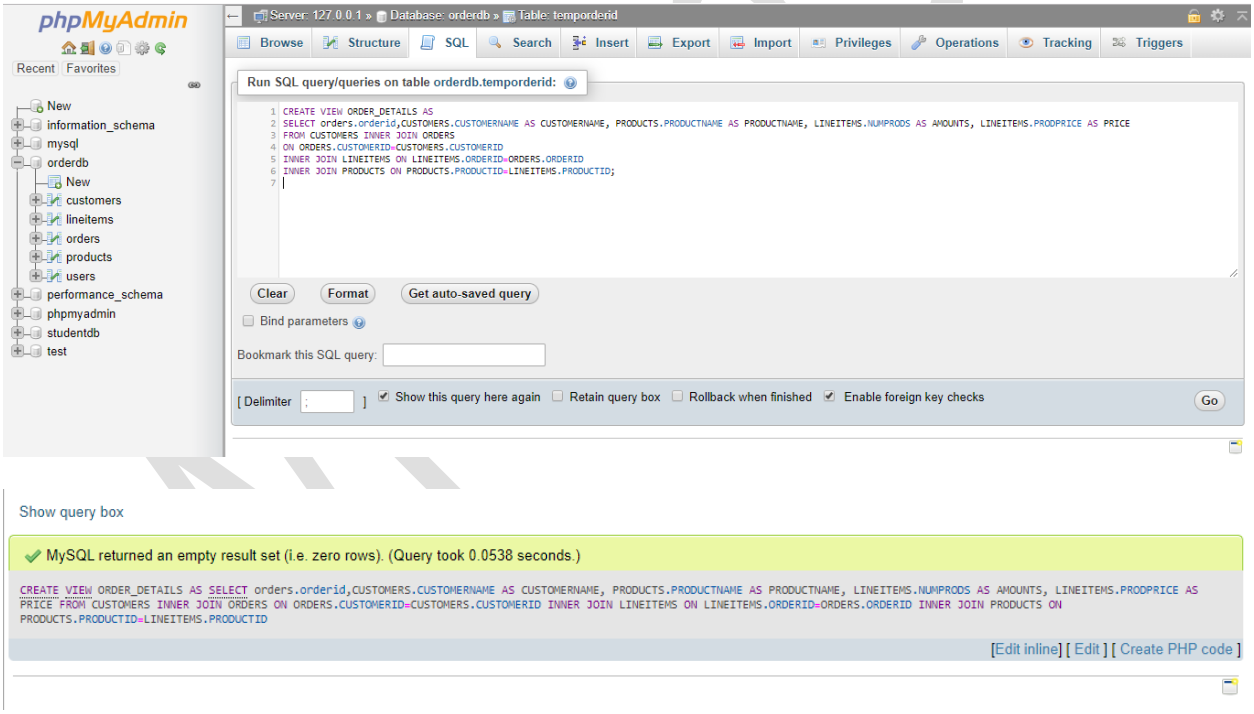
SELECT orders.orderid,CUSTOMERS.CUSTOMERNAME AS CUSTOMERNAME, PRODUCTS.PRODUCTNAME AS PRODUCTNAME, LINEITEMS.NUMPRODS AS AMOUNTS, LINEITEMS.PRODPRICE AS PRICE

FROM CUSTOMERS INNER JOIN ORDERS

ON ORDERS.CUSTOMERID=CUSTOMERS.CUSTOMERID

INNER JOIN LINEITEMS ON LINEITEMS.ORDERID=ORDERS.ORDERID

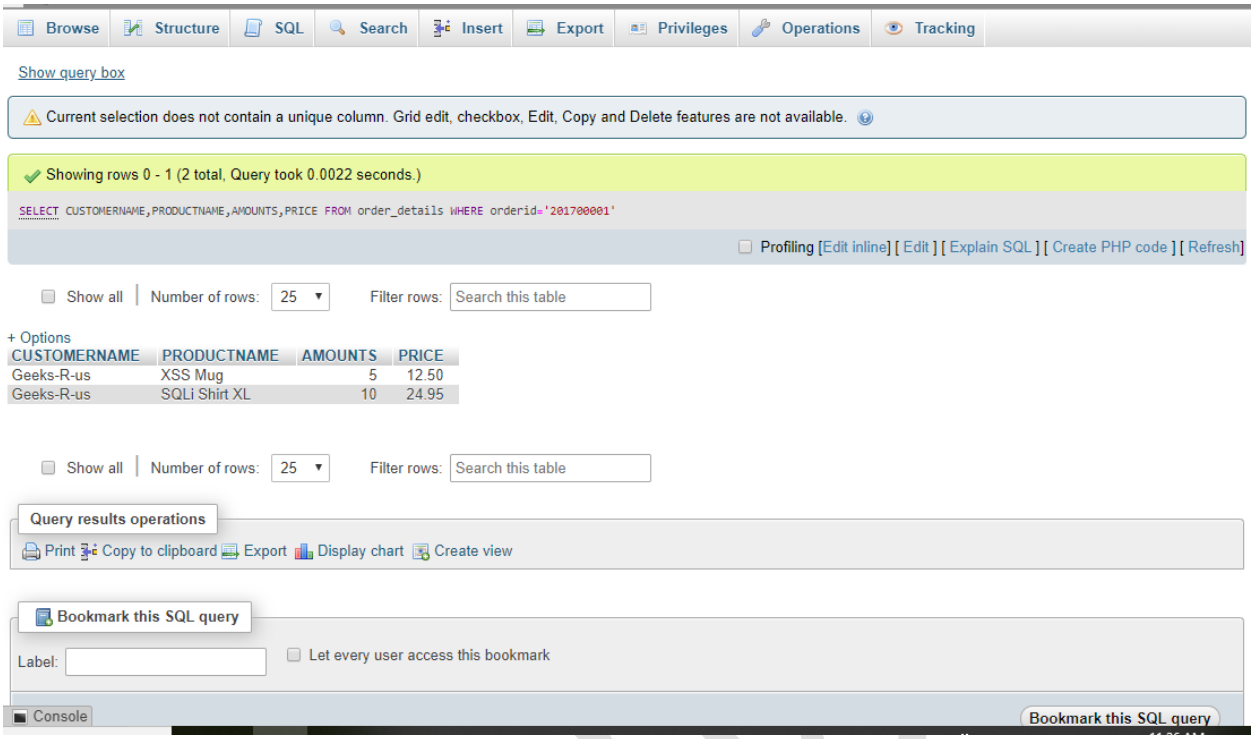
INNER JOIN PRODUCTS ON PRODUCTS.PRODUCTID=LINEITEMS.PRODUCTID;



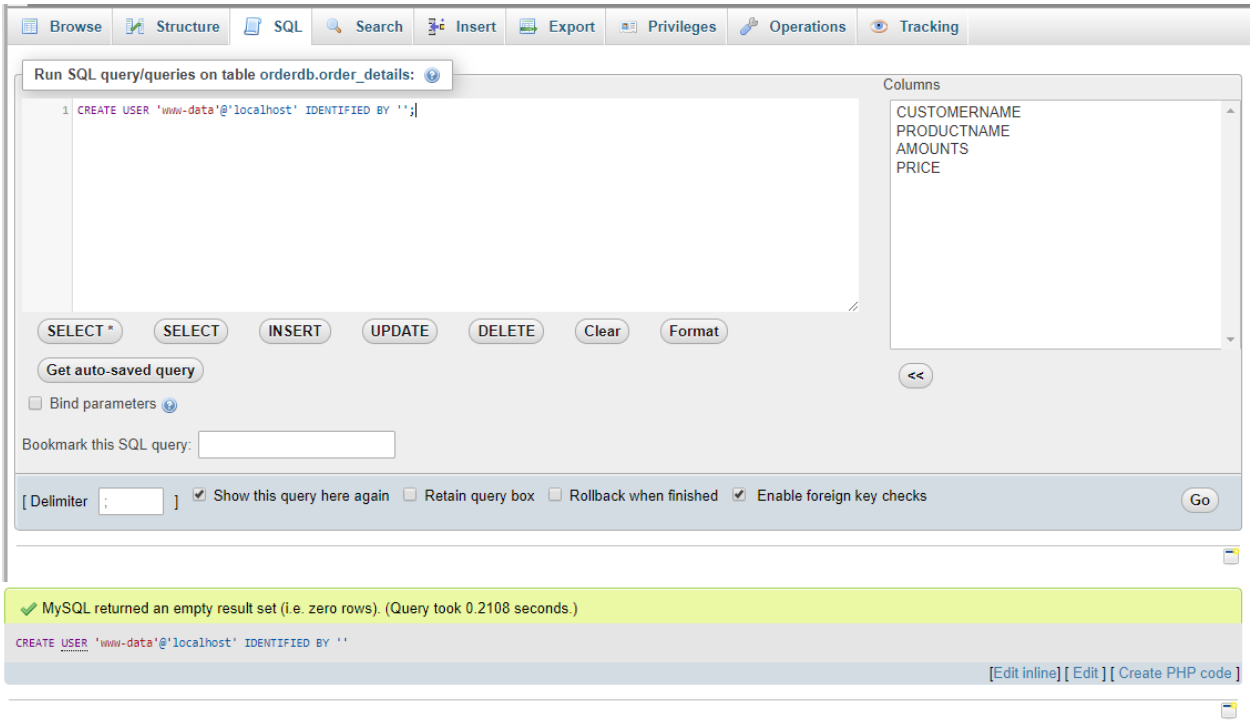
TEST the VIEW:

SELECT CUSTOMERNAME,PRODUCTNAME,AMOUNTS,PRICE FROM order_details

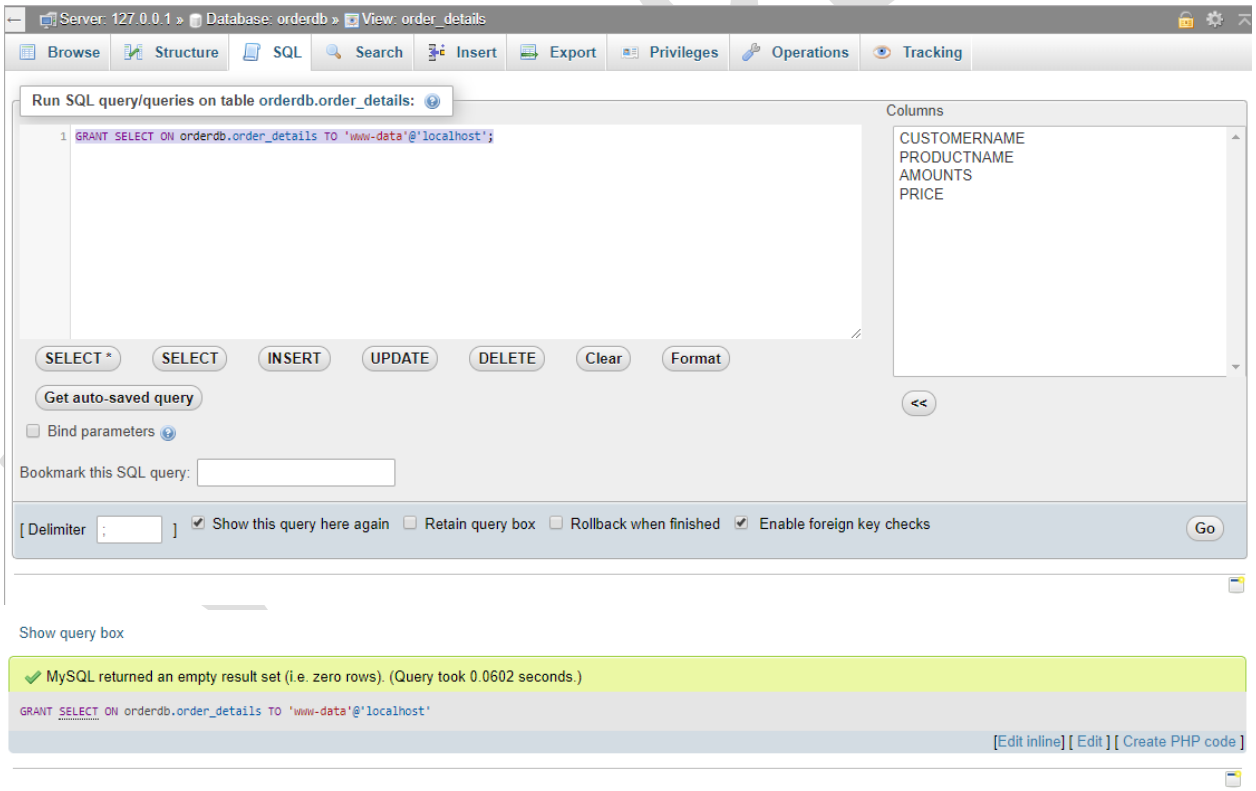
WHERE orderid='201700001';



2. Extend appropriate rights to the www-data user on the view you created so that it can retrieve data from the view, but that it can do nothing else.
   CREATE USER 'www-data'@'localhost' IDENTIFIED BY '';



GRANT SELECT ON orderdb.order_details TO 'www-data'@'localhost';



3. Create a web form that will prompt the user to enter an order number, and use that to execute a parameterized query. Present the results to the users as a single table.

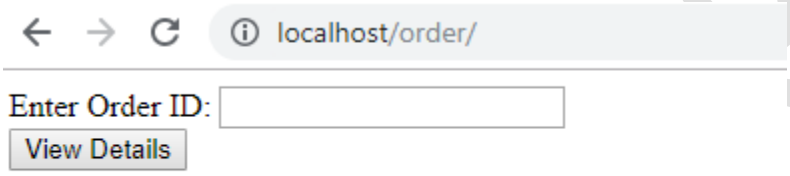
```
<?php

if(isset($_POST['submit']))

{

        $hostname = "localhost";//servername

        $user = 'www-data';//username

        $password = '';//password

        $database = "orderdb";//database name

        $mysqli = new mysqli($hostname, $user, $password,$database);

        if ($mysqli->connect_error) {//check either database connection is successful or not

                throw new Exception("Cannot connect to the databse.");//if not display error

        }

   $orderID = $_POST['txtOrderID'];//get the Order ID from the form

   $stmt = $mysqli->prepare("SELECT CUSTOMERNAME,PRODUCTNAME,AMOUNTS,PRICE FROM order_details WHERE orderid = ?");//using parameterized select statement on view

        $stmt->bind_param("s", $orderID);//pass the order id

        $stmt->execute();//execute the statement

        $result = $stmt->get_result();//query result to the result variable

        if($result->num_rows === 0) {echo "Order Id does not exists";}//if no record found the display error message

        else

        {

?>

<table border="1">
```

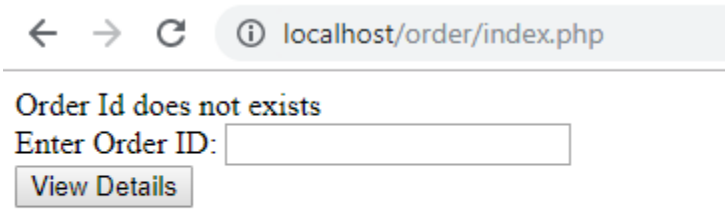
Writingsol

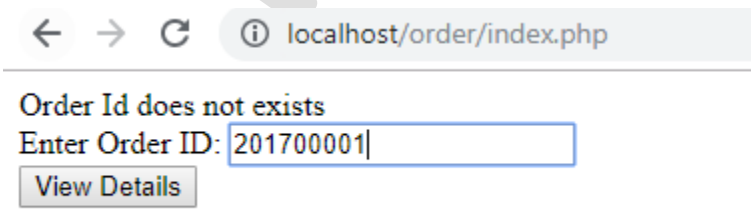
```
<tr><th>Customer Name</th><th>Product Name</th><th>Amounts</th><th>Price ($)</th></tr>


<?php
        while($row = $result->fetch_assoc()) {//fetch each row and display its records in each row of the table.
?>
<tr><td>
<?php
            echo $row['CUSTOMERNAME'];
?>
</td><td>
<?php
            echo  $row['PRODUCTNAME'];
?>
</td><td>
<?php
            echo $row['AMOUNTS'];
?>
</td><td>
<?php
            echo $row['PRICE'];
?>
</td></tr>
<?php
        }
```

WritingSol

```
?>
</table>
<?php
        }
        $stmt->close();
}
?>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
Enter Order ID: <input type="text" name="txtOrderID"><br>
<input type="submit" name="submit" value="View Details"><br>
</form>
```







Customer Name	Product Name	Amounts	Price ($)
Geeks-R-us	XSS Mug	5	12.50
Geeks-R-us	SQLi Shirt XL	10	24.95

Enter Order ID:

View Details

WritingSol