Instructions:

- This assignment is designed to test your knowledge of MySQL database concepts in the context of a stock management system for an e-commerce platform.
- Write SQL queries for each task and provide explanations where necessary.
- Submit your assignment as a single SQL script file.

Task 1: Create a Database Create a new MySQL database named `ecommerce_stock_db`.

Ans:-

```
create database ecommerce_stock_db
```

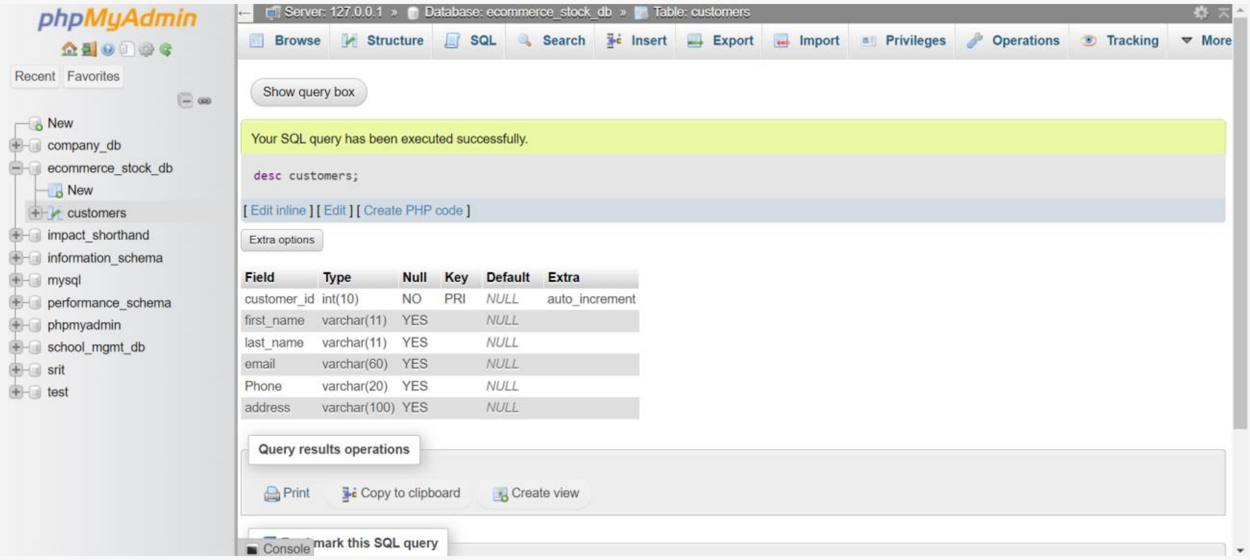
Task 2: Create Tables Inside the `ecommerce_stock_db`, create the following tables:

Ans:-

- `customers` table with the following columns:
  - `customer_id` (Primary Key, Auto Increment)
  - `first_name`
  - `last_name`
  - `email`
  - `Phone`
  - `Address`

Creating customers table :-

```
create table customers (

 customer_id int(10) not null auto_increment,

   first_name varchar(11) default null,

   last_name varchar(11) default null,

   email varchar(60) default null,

   Phone varchar(20) default null,
    address varchar(100) default null,

    primary key (customer_id)

);



desc customers;
```



Your SQL query has been executed successfully.

```
desc customers;
```

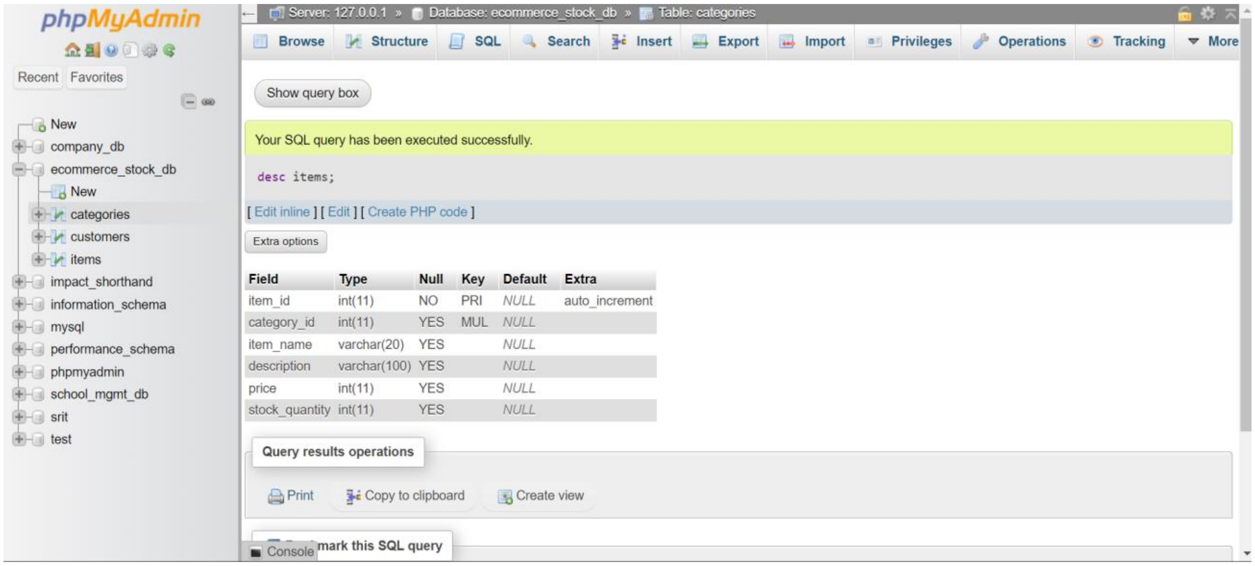
Field	Type	Null	Key	Default	Extra
customer_id	int(10)	NO	PRI	*NULL*	auto_increment
first_name	varchar(11)	YES		*NULL*	
last_name	varchar(11)	YES		*NULL*	
email	varchar(60)	YES		*NULL*	
Phone	varchar(20)	YES		*NULL*	
address	varchar(100)	YES		*NULL*	

- `items` table with the following columns:
    - `item_id` (Primary Key, Auto Increment)
    - `item_name`
    - `description`
    - `price`
    - `stock_quantity`
    - `category_id` (Foreign Key referencing `category_id` in the `categories` table)

creating items table:-

```
create table items ( item_id int(11) not null auto_increment, category_id int(11), foreign key(category_id) references categories(category_id), item_name varchar(20) default null, description varchar(100) default null, price int(11) default null, stock_quantity int(11) default null, primary key (item_id) );
```

```
desc items;
```



Your SQL query has been executed successfully.

```
desc items;
```

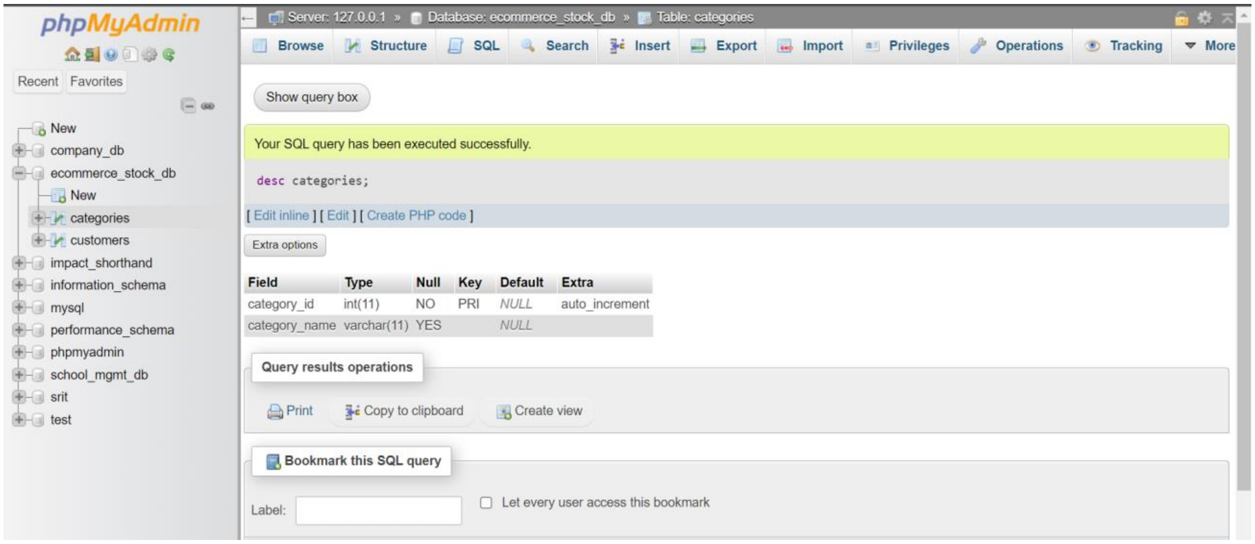
Field	Type	Null	Key	Default	Extra
item_id	int(11)	NO	PRI	NULL	auto_increment
category_id	int(11)	YES	MUL	NULL	
item_name	varchar(20)	YES		NULL	
description	varchar(100)	YES		NULL	
price	int(11)	YES		NULL	
stock_quantity	int(11)	YES		NULL	

- `categories` table with the following columns:
  - `category_id` (Primary Key, Auto Increment)
  - `category_name`

Creating category table:-

```
create table categories ( category_id int(11) not null auto_increment, catego
ry_name varchar(11) default null, primary key (category_id) );
```

```
desc categories;
```



Your SQL query has been executed successfully.

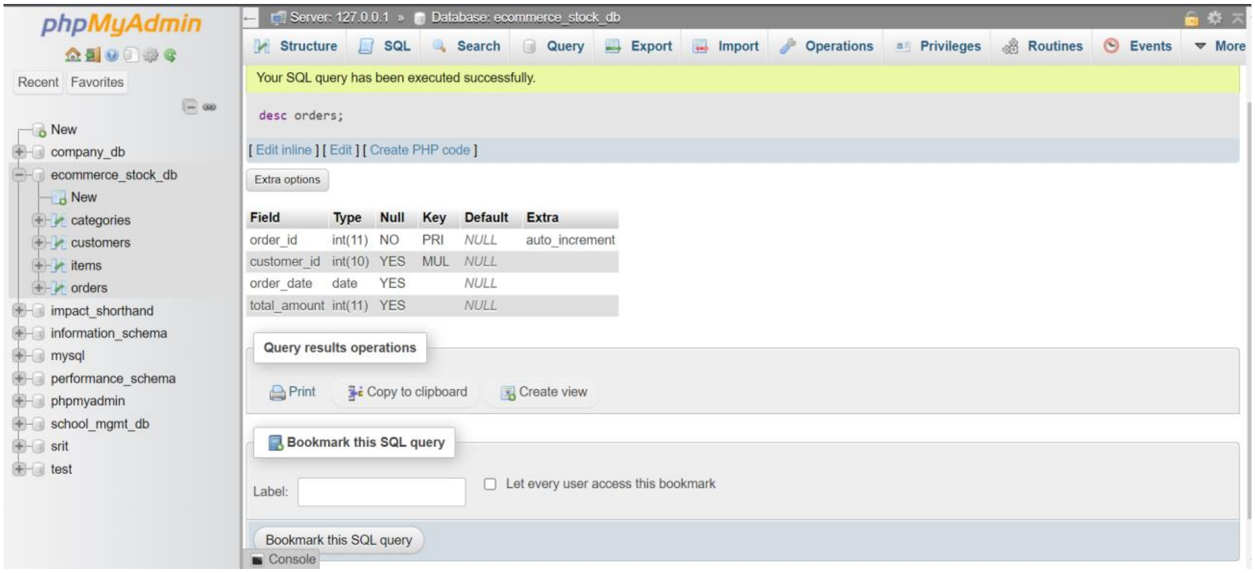
```
desc categories;
```

Field	Type	Null	Key	Default	Extra
category_id	int(11)	NO	PRI	NULL	auto_increment
category_name	varchar(11)	YES		NULL	

- `orders` table with the following columns:
    - `order_id` (Primary Key, Auto Increment)
    - `customer_id` (Foreign Key referencing `customer_id` in the `customers` table)
    - `order_date`
    - `total_amount`

creating orders table:-

```
create table orders ( order_id int(11) not null auto_increment, customer_id int(10), foreign key (customer_id) references customers (customer_id), order_date date default null, total_amount int(11) default null, primary key(order_id) );
```



Your SQL query has been executed successfully.

```
desc orders;
```

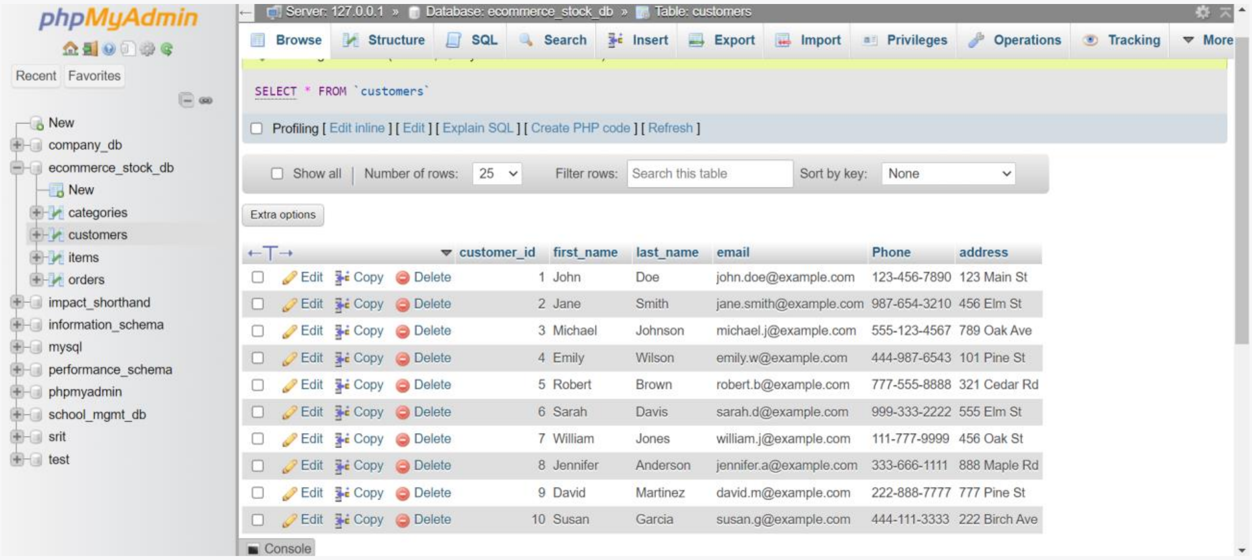
Field	Type	Null	Key	Default	Extra
order_id	int(11)	NO	PRI	NULL	auto_increment
customer_id	int(10)	YES	MUL	NULL	
order_date	date	YES		NULL	
total_amount	int(11)	YES		NULL	

Task 3: Insert Data Insert at least 10 records into the `customers` table, 20 records into the `items` table, 5 records into the `categories` table, and 30 records into the `orders` table. Ensure that items are assigned to categories and orders are associated with customers and items.

Ans:-

Inserting data into customers table :-

```
insert into customers (first_name, last_name, email, Phone, address)

values

    ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '123 Main St'),

    ('Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '456 Elm St'),

    ('Michael', 'Johnson', 'michael.j@example.com', '555-123-4567', '789 Oak Ave'),

    ('Emily', 'Wilson', 'emily.w@example.com', '444-987-6543', '101 Pine St'),

    ('Robert', 'Brown', 'robert.b@example.com', '777-555-8888', '321 Cedar Rd'),

    ('Sarah', 'Davis', 'sarah.d@example.com', '999-333-2222', '555 Elm St'),

    ('William', 'Jones', 'william.j@example.com', '111-777-9999', '456 Oak St'),

    ('Jennifer', 'Anderson', 'jennifer.a@example.com', '333-666-1111', '888 Maple Rd'),

    ('David', 'Martinez', 'david.m@example.com', '222-888-7777', '777 Pine St'),

    ('Susan', 'Garcia', 'susan.g@example.com', '444-111-3333', '222 Birch Ave');
```

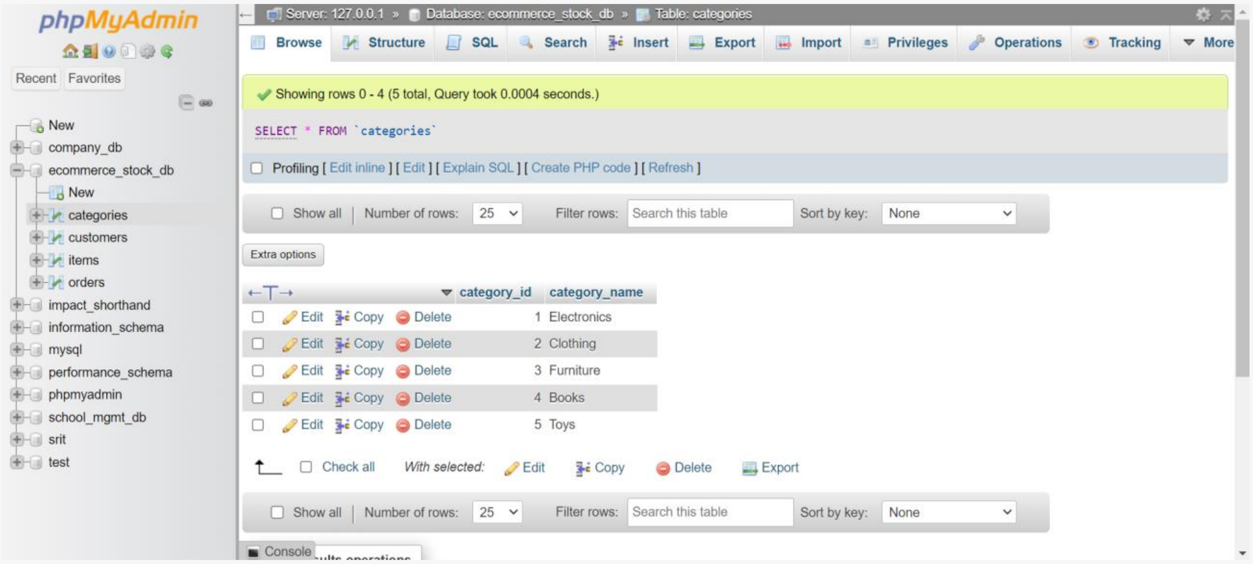


```
SELECT * FROM `customers`
```

customer_id	first_name	last_name	email	Phone	address
1	John	Doe	john.doe@example.com	123-456-7890	123 Main St
2	Jane	Smith	jane.smith@example.com	987-654-3210	456 Elm St
3	Michael	Johnson	michael.j@example.com	555-123-4567	789 Oak Ave
4	Emily	Wilson	emily.w@example.com	444-987-6543	101 Pine St
5	Robert	Brown	robert.b@example.com	777-555-8888	321 Cedar Rd
6	Sarah	Davis	sarah.d@example.com	999-333-2222	555 Elm St
7	William	Jones	william.j@example.com	111-777-9999	456 Oak St
8	Jennifer	Anderson	jennifer.a@example.com	333-666-1111	888 Maple Rd
9	David	Martinez	david.m@example.com	222-888-7777	777 Pine St
10	Susan	Garcia	susan.g@example.com	444-111-3333	222 Birch Ave

Then I had to insert data into categories table where no foreign key is used

```
insert into categories (category_name) values ('Electronics'), ('Clothing'),
('Furniture'), ('Books'), ('Toys');
```



✓ Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `categories`
```

category_id	category_name
1	Electronics
2	Clothing
3	Furniture
4	Books
5	Toys

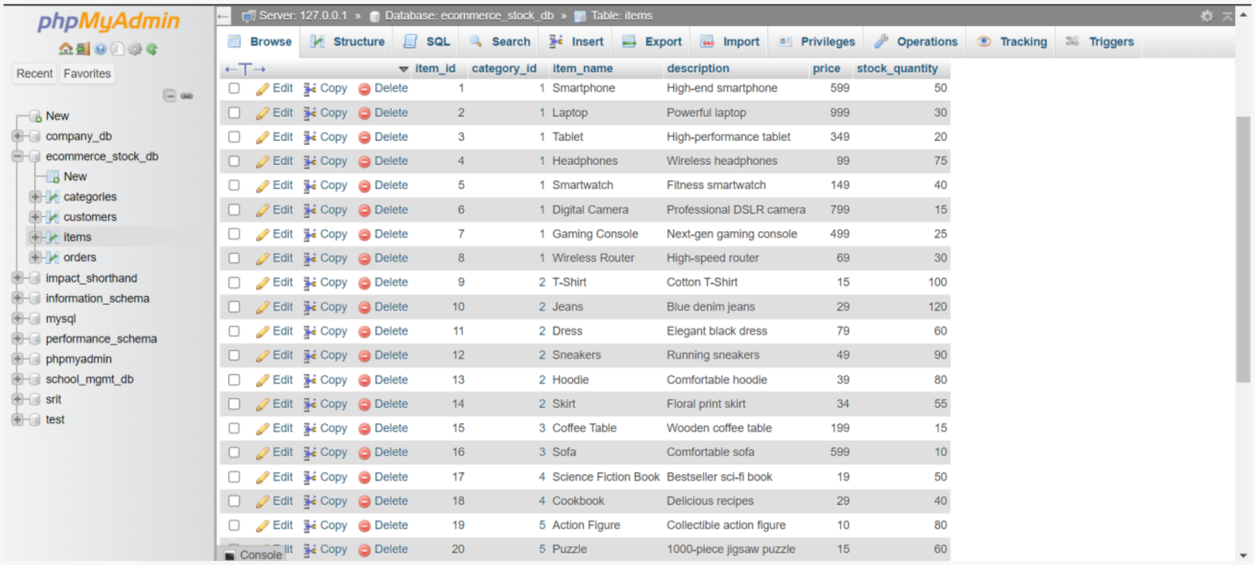
Then I inserted data into items table:-

insert into items (category_id, item_name, description, price, stock_quantity)

values

(1, 'Smartphone', 'High-end smartphone', 599, 50),

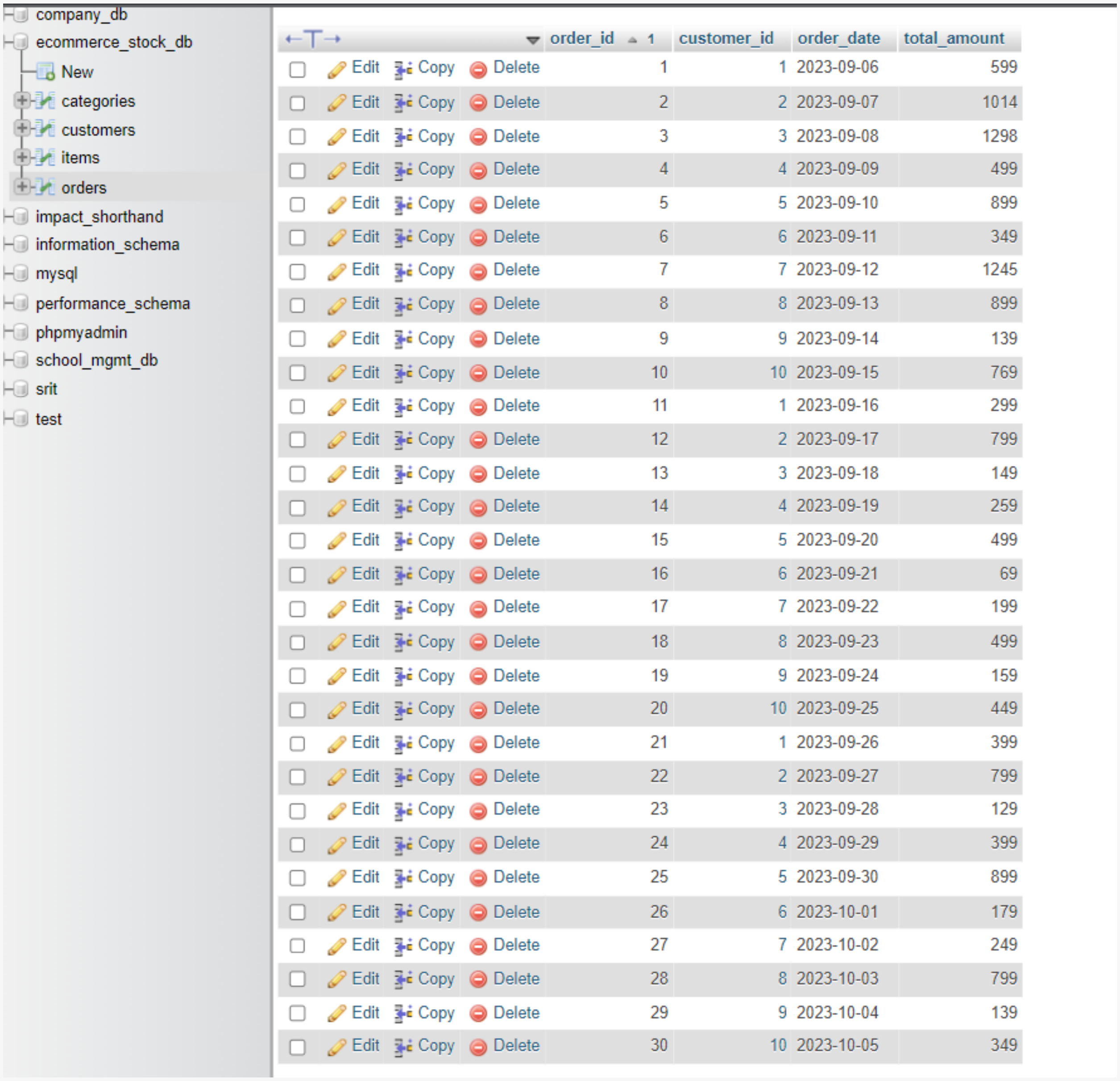
```
    (1, 'Laptop', 'Powerful laptop', 999, 30),
    (1, 'Tablet', 'High-performance tablet', 349, 20),
    (1, 'Headphones', 'Wireless headphones', 99, 75),
    (1, 'Smartwatch', 'Fitness smartwatch', 149, 40),
    (1, 'Digital Camera', 'Professional DSLR camera', 799, 15),
    (1, 'Gaming Console', 'Next-gen gaming console', 499, 25),
    (1, 'Wireless Router', 'High-speed router', 69, 30),
    (2, 'T-Shirt', 'Cotton T-Shirt', 15, 100),
    (2, 'Jeans', 'Blue denim jeans', 29, 120),
    (2, 'Dress', 'Elegant black dress', 79, 60),
    (2, 'Sneakers', 'Running sneakers', 49, 90),
    (2, 'Hoodie', 'Comfortable hoodie', 39, 80),
    (2, 'Skirt', 'Floral print skirt', 34, 55),
    (3, 'Coffee Table', 'Wooden coffee table', 199, 15),
    (3, 'Sofa', 'Comfortable sofa', 599, 10),
    (4, 'Science Fiction Book', 'Bestseller sci-fi book', 19, 50),
    (4, 'Cookbook', 'Delicious recipes', 29, 40),
    (5, 'Action Figure', 'Collectible action figure', 10, 80),
    (5, 'Puzzle', '1000-piece jigsaw puzzle', 15, 60)
;
```



item_id	category_id	item_name	description	price	stock_quantity
1	1	Smartphone	High-end smartphone	599	50
2	1	Laptop	Powerful laptop	999	30
3	1	Tablet	High-performance tablet	349	20
4	1	Headphones	Wireless headphones	99	75
5	1	Smartwatch	Fitness smartwatch	149	40
6	1	Digital Camera	Professional DSLR camera	799	15
7	1	Gaming Console	Next-gen gaming console	499	25
8	1	Wireless Router	High-speed router	69	30
9	2	T-Shirt	Cotton T-Shirt	15	100
10	2	Jeans	Blue denim jeans	29	120
11	2	Dress	Elegant black dress	79	60
12	2	Sneakers	Running sneakers	49	90
13	2	Hoodie	Comfortable hoodie	39	80
14	2	Skirt	Floral print skirt	34	55
15	3	Coffee Table	Wooden coffee table	199	15
16	3	Sofa	Comfortable sofa	599	10
17	4	Science Fiction Book	Bestseller sci-fi book	19	50
18	4	Cookbook	Delicious recipes	29	40
19	5	Action Figure	Collectible action figure	10	80
20	5	Puzzle	1000-piece jigsaw puzzle	15	60

Now inserting data into orders table :-

```
insert into orders (customer_id, order_date, total_amount)

values

    (1, '2023-09-06', 599),

    (2, '2023-09-07', 1014),

    (3, '2023-09-08', 1298),

    (4, '2023-09-09', 499),

    (5, '2023-09-10', 899),

    (6, '2023-09-11', 349),

    (7, '2023-09-12', 1245),

    (8, '2023-09-13', 899),

    (9, '2023-09-14', 139),

    (10, '2023-09-15', 769),

    (1, '2023-09-16', 299),
(2, '2023-09-17', 799),
(3, '2023-09-18', 149),
(4, '2023-09-19', 259),
(5, '2023-09-20', 499),
(6, '2023-09-21', 69),
(7, '2023-09-22', 199),
(8, '2023-09-23', 499),
(9, '2023-09-24', 159),
(10, '2023-09-25', 449),
(1, '2023-09-26', 399),
(2, '2023-09-27', 799),
(3, '2023-09-28', 129),
(4, '2023-09-29', 399),
(5, '2023-09-30', 899),
(6, '2023-10-01', 179),
(7, '2023-10-02', 249),
(8, '2023-10-03', 799),
(9, '2023-10-04', 139),
(10, '2023-10-05', 349);
```

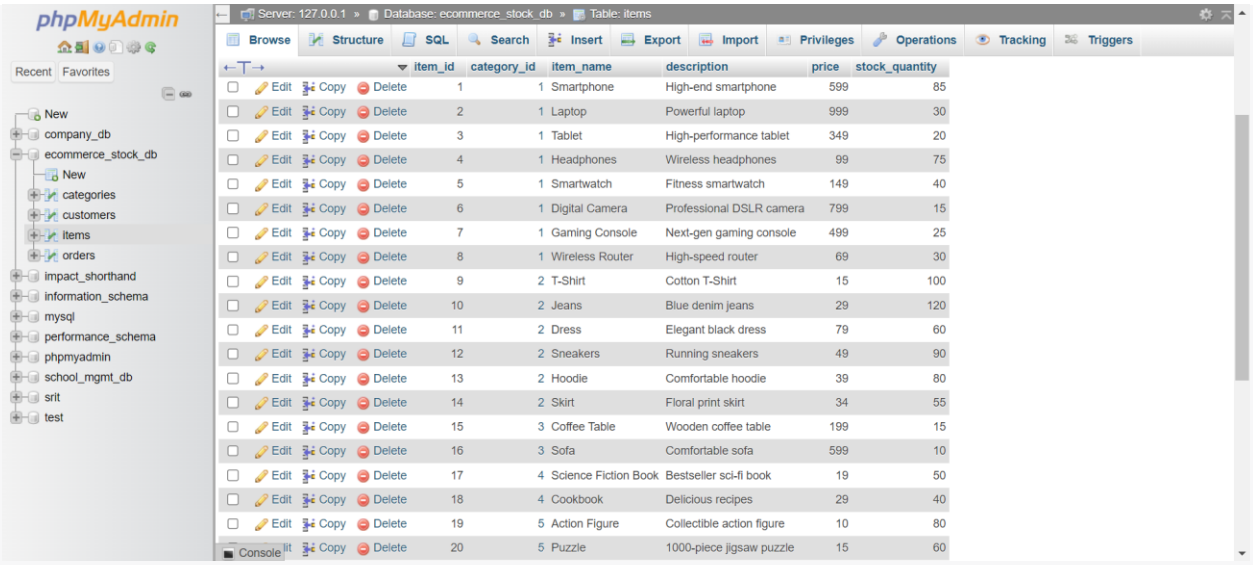


order_id	customer_id	order_date	total_amount
1	1	2023-09-06	599
2	2	2023-09-07	1014
3	3	2023-09-08	1298
4	4	2023-09-09	499
5	5	2023-09-10	899
6	6	2023-09-11	349
7	7	2023-09-12	1245
8	8	2023-09-13	899
9	9	2023-09-14	139
10	10	2023-09-15	769
11	1	2023-09-16	299
12	2	2023-09-17	799
13	3	2023-09-18	149
14	4	2023-09-19	259
15	5	2023-09-20	499
16	6	2023-09-21	69
17	7	2023-09-22	199
18	8	2023-09-23	499
19	9	2023-09-24	159
20	10	2023-09-25	449
21	1	2023-09-26	399
22	2	2023-09-27	799
23	3	2023-09-28	129
24	4	2023-09-29	399
25	5	2023-09-30	899
26	6	2023-10-01	179
27	7	2023-10-02	249
28	8	2023-10-03	799
29	9	2023-10-04	139
30	10	2023-10-05	349

Task 4: Update Data Write a SQL query to update the stock quantity of a specific item in the `items` table.

Ans:-

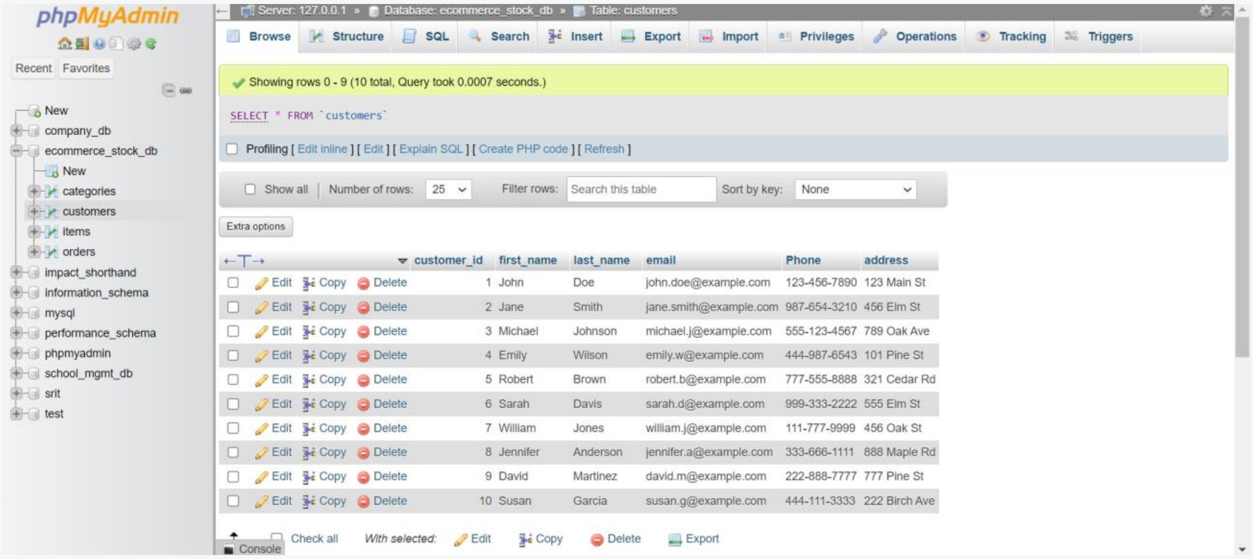
```
update items set stock_quantity = 85 where item_id=1;
```



item_id	category_id	item_name	description	price	stock_quantity
1	1	Smartphone	High-end smartphone	599	85
2	1	Laptop	Powerful laptop	999	30
3	1	Tablet	High-performance tablet	349	20
4	1	Headphones	Wireless headphones	99	75
5	1	Smartwatch	Fitness smartwatch	149	40
6	1	Digital Camera	Professional DSLR camera	799	15
7	1	Gaming Console	Next-gen gaming console	499	25
8	1	Wireless Router	High-speed router	69	30
9	2	T-Shirt	Cotton T-Shirt	15	100
10	2	Jeans	Blue denim jeans	29	120
11	2	Dress	Elegant black dress	79	60
12	2	Sneakers	Running sneakers	49	90
13	2	Hoodie	Comfortable hoodie	39	80
14	2	Skirt	Floral print skirt	34	55
15	3	Coffee Table	Wooden coffee table	199	15
16	3	Sofa	Comfortable sofa	599	10
17	4	Science Fiction Book	Bestseller sci-fi book	19	50
18	4	Cookbook	Delicious recipes	29	40
19	5	Action Figure	Collectible action figure	10	80
20	5	Puzzle	1000-piece jigsaw puzzle	15	60

Task 5: Delete Data Write a SQL query to delete a customer from the `customers` table and all related orders (if any) in the database.

Before deleting data from customers table:-



✓ Showing rows 0 - 9 (10 total, Query took 0.0007 seconds.)

```
SELECT * FROM `customers`
```

customer_id	first_name	last_name	email	Phone	address
1	John	Doe	john.doe@example.com	123-456-7890	123 Main St
2	Jane	Smith	jane.smith@example.com	987-654-3210	456 Elm St
3	Michael	Johnson	michael.j@example.com	555-123-4567	789 Oak Ave
4	Emily	Wilson	emily.w@example.com	444-987-6543	101 Pine St
5	Robert	Brown	robert.b@example.com	777-555-8888	321 Cedar Rd
6	Sarah	Davis	sarah.d@example.com	999-333-2222	555 Elm St
7	William	Jones	william.j@example.com	111-777-9999	456 Oak St
8	Jennifer	Anderson	jennifer.a@example.com	333-666-1111	888 Maple Rd
9	David	Martinez	david.m@example.com	222-888-7777	777 Pine St
10	Susan	Garcia	susan.g@example.com	444-111-3333	222 Birch Ave

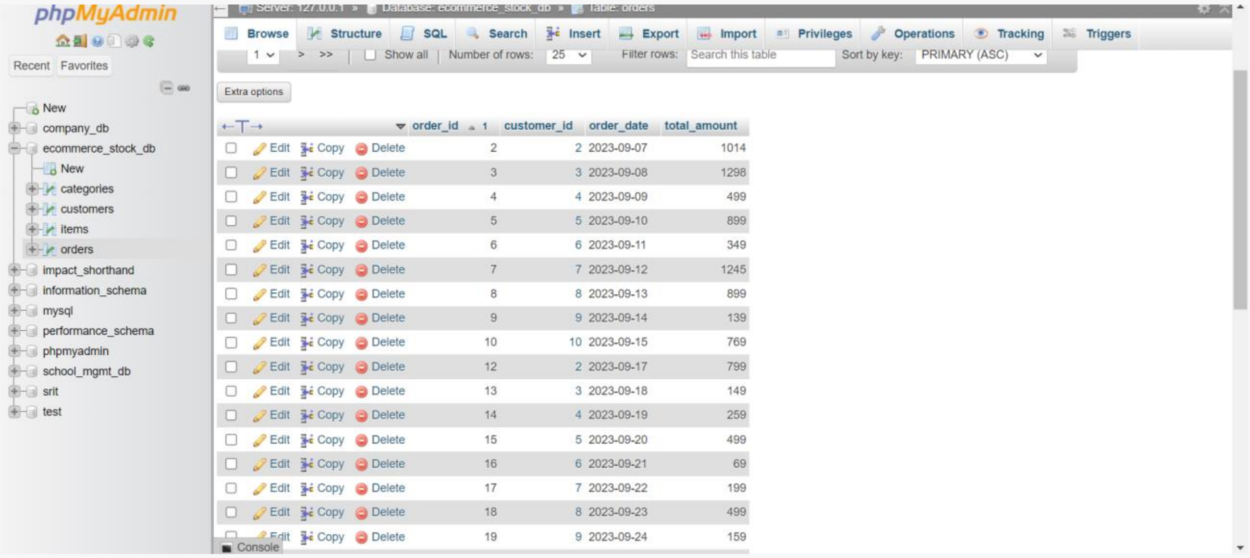
Note:-
This query is not working because of fkey constraints(delete customers, orders from customers left join orders on customers.customer_id=orders.customer_id where customers.customer_id=1;)

now I used
delete from customers where customer_id=1;

yet it will not work because there is data related to it in orders table so first of all I will delete from orders table.

```
delete from orders where customer_id=1;
```

```
(it worked)
now we will delete from customers table and all records for customer_id 1 is
deleted.
```

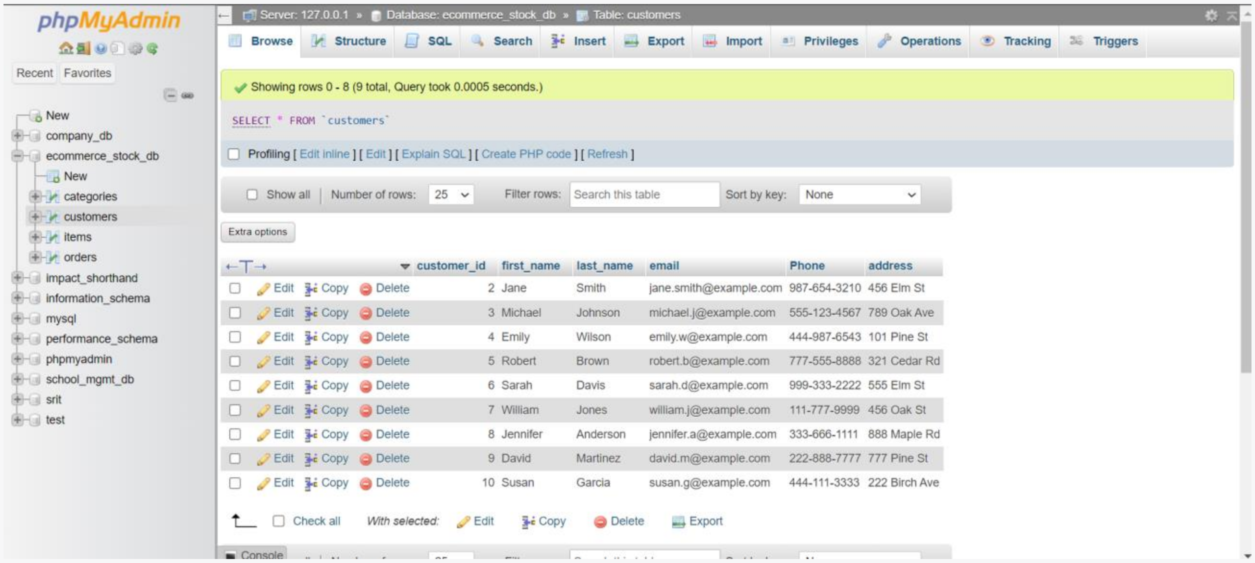


Server: 127.0.0.1 » Database: ecommerce_stock_db » Table: orders

order_id	customer_id	order_date	total_amount
2	2	2023-09-07	1014
3	3	2023-09-08	1298
4	4	2023-09-09	499
5	5	2023-09-10	899
6	6	2023-09-11	349
7	7	2023-09-12	1245
8	8	2023-09-13	899
9	9	2023-09-14	139
10	10	2023-09-15	769
12	2	2023-09-17	799
13	3	2023-09-18	149
14	4	2023-09-19	259
15	5	2023-09-20	499
16	6	2023-09-21	69
17	7	2023-09-22	199
18	8	2023-09-23	499
19	9	2023-09-24	159

```
now I will delete the data where customer_id =1
```

delete from customers where customer_id=1;



Server: 127.0.0.1 » Database: ecommerce_stock_db » Table: customers

Showing rows 0 - 8 (9 total, Query took 0.0005 seconds.)

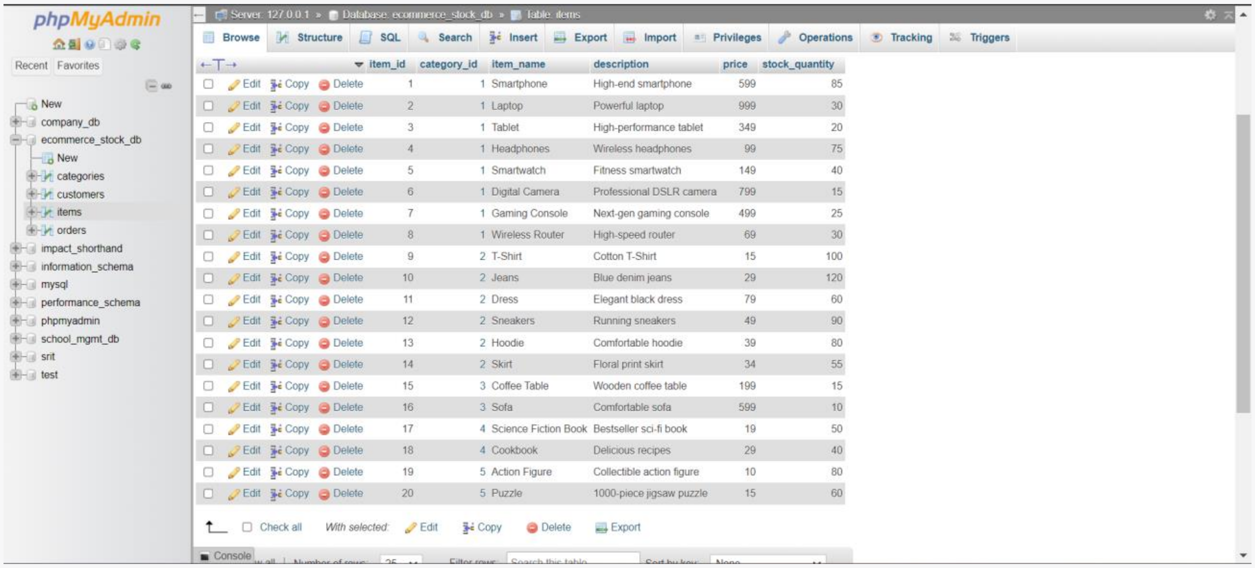
```
SELECT * FROM `customers`
```

customer_id	first_name	last_name	email	Phone	address
2	Jane	Smith	jane.smith@example.com	987-654-3210	456 Elm St
3	Michael	Johnson	michael.j@example.com	555-123-4567	789 Oak Ave
4	Emily	Wilson	emily.w@example.com	444-987-6543	101 Pine St
5	Robert	Brown	robert.b@example.com	777-555-8888	321 Cedar Rd
6	Sarah	Davis	sarah.d@example.com	999-333-2222	555 Elm St
7	William	Jones	william.j@example.com	111-777-9999	456 Oak St
8	Jennifer	Anderson	jennifer.a@example.com	333-666-1111	888 Maple Rd
9	David	Martinez	david.m@example.com	222-888-7777	777 Pine St
10	Susan	Garcia	susan.g@example.com	444-111-3333	222 Birch Ave

Task 6: Retrieve Item Information Write an SQL query to retrieve information about an item, including its name, description, price, and category name, for a specific item.

Ans:-

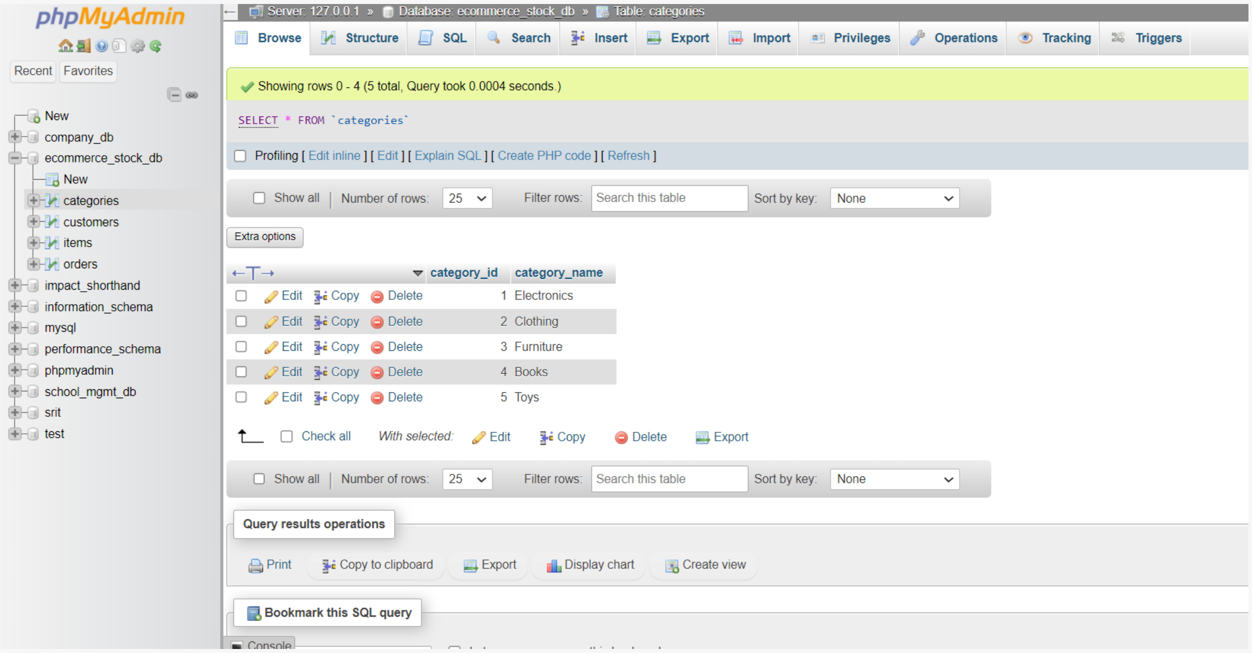
Items table :-



Server: 127.0.0.1 » Database: ecommerce_stock_db » Table: items

item_id	category_id	item_name	description	price	stock_quantity
1	1	Smartphone	High-end smartphone	599	85
2	1	Laptop	Powerful laptop	999	30
3	1	Tablet	High-performance tablet	349	20
4	1	Headphones	Wireless headphones	99	75
5	1	Smartwatch	Fitness smartwatch	149	40
6	1	Digital Camera	Professional DSLR camera	799	15
7	1	Gaming Console	Next-gen gaming console	499	25
8	1	Wireless Router	High-speed router	69	30
9	2	T-Shirt	Cotton T-Shirt	15	100
10	2	Jeans	Blue denim jeans	29	120
11	2	Dress	Elegant black dress	79	60
12	2	Sneakers	Running sneakers	49	90
13	2	Hoodie	Comfortable hoodie	39	80
14	2	Skirt	Floral print skirt	34	55
15	3	Coffee Table	Wooden coffee table	199	15
16	3	Sofa	Comfortable sofa	599	10
17	4	Science Fiction Book	Bestseller sci-fi book	19	50
18	4	Cookbook	Delicious recipes	29	40
19	5	Action Figure	Collectible action figure	10	80
20	5	Puzzle	1000-piece jigsaw puzzle	15	60

Categories table:-


phpMyAdmin
Recent Favorites
New
company_db
ecommerce_stock_db
New
categories
customers
items
orders
impact_shorthand
information_schema
mysql
performance_schema
phpmyadmin
school_mgmt_db
srit
test
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Extra options
Check all With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Console

Server: 127.0.0.1 » Database: ecommerce_stock_db » Table: categories

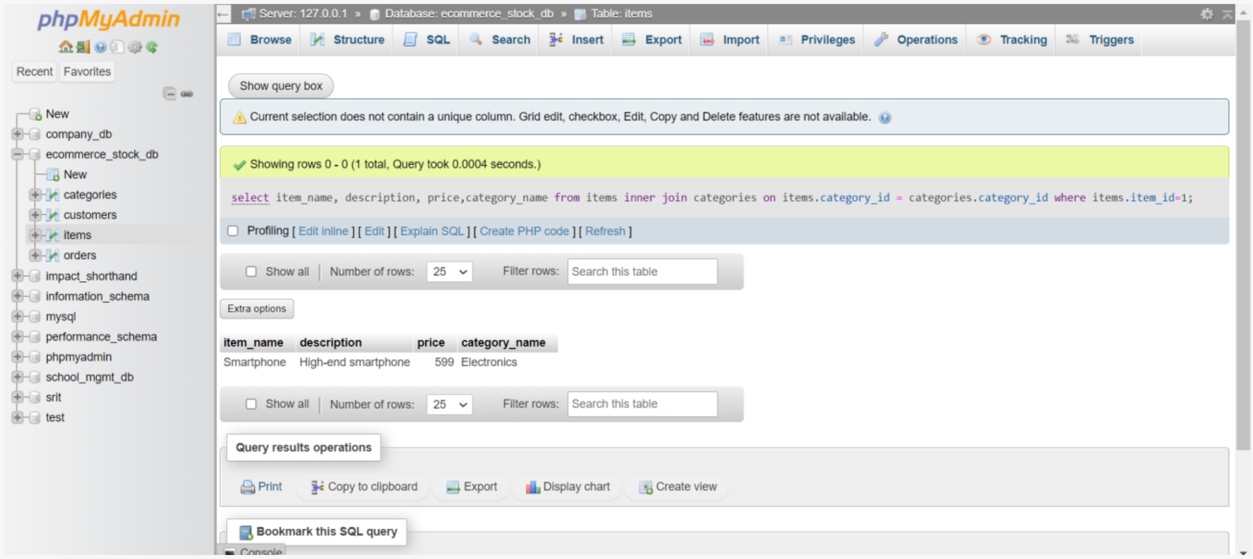
Showing rows 0 - 4 (5 total, Query took 0.0004 seconds.)

```
SELECT * FROM `categories`
```

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

category_id	category_name
1	Electronics
2	Clothing
3	Furniture
4	Books
5	Toys

```
select item_name, description, price,category_name from items inner join cate
gories on items.category_id = categories.category_id where items.item_id=1;
```



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

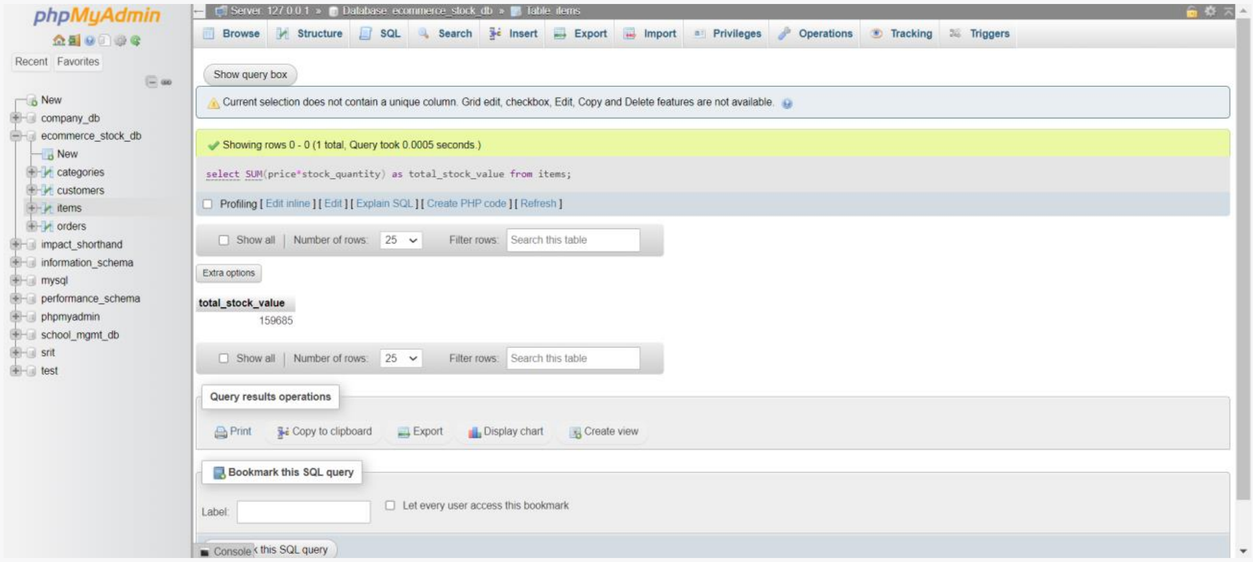
```
select item_name, description, price,category_name from items inner join categories on items.category_id = categories.category_id where items.item_id=1;
```

item_name	description	price	category_name
Smartphone	High-end smartphone	599	Electronics

Task 7: Calculate Total Stock Value Write an SQL query to calculate the total value of all items in stock, considering the price and stock quantity.

Ans:-

```
select sum(price*stock_quantity) as total_stock_value from items;
```



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
select SUM(price*stock_quantity) as total_stock_value from items;
```

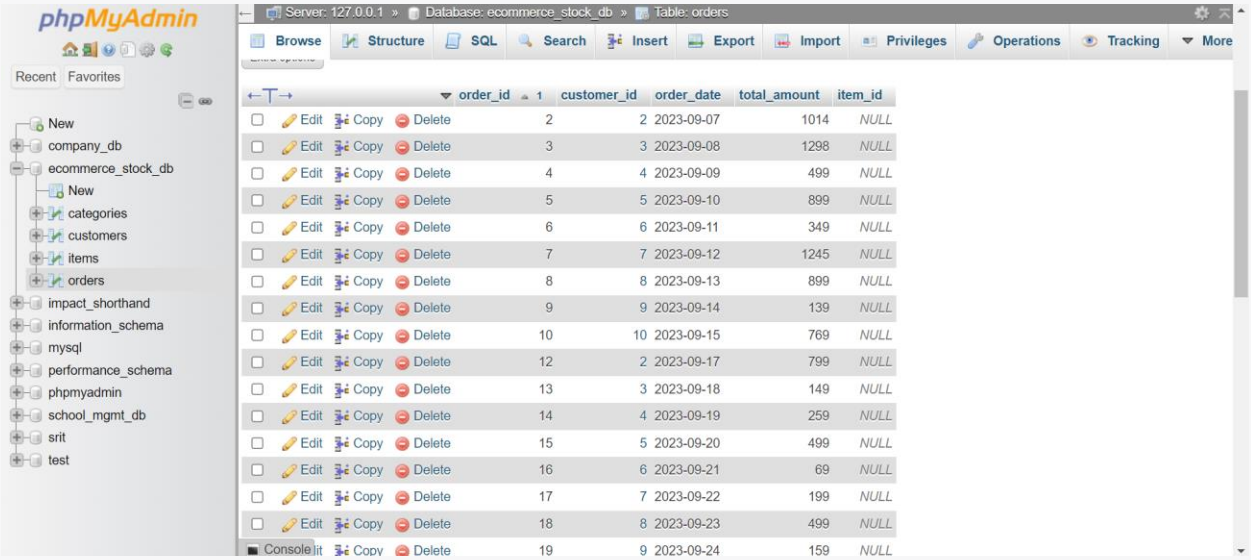
total_stock_value
159685

Task 8: Retrieve Customer Order History Write an SQL query to retrieve the order history for a specific customer, including order dates, item names, quantities, and total amounts spent.

Ans:- To achive this I must establish relationship among tables to fetch data using fkey, so I am going to add item_id column in orders table with foreign key constraints.

Adding item_id column in orders table:-

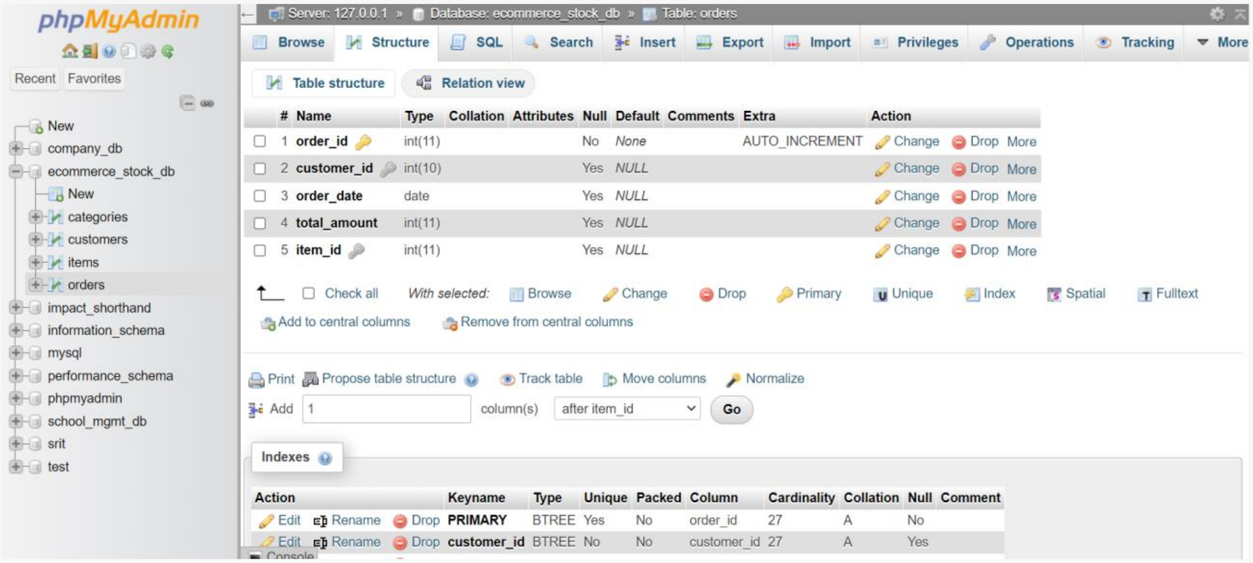
alter table orders add column item_id int(11);



order_id	customer_id	order_date	total_amount	item_id
2	2	2023-09-07	1014	NULL
3	3	2023-09-08	1298	NULL
4	4	2023-09-09	499	NULL
5	5	2023-09-10	899	NULL
6	6	2023-09-11	349	NULL
7	7	2023-09-12	1245	NULL
8	8	2023-09-13	899	NULL
9	9	2023-09-14	139	NULL
10	10	2023-09-15	769	NULL
12	2	2023-09-17	799	NULL
13	3	2023-09-18	149	NULL
14	4	2023-09-19	259	NULL
15	5	2023-09-20	499	NULL
16	6	2023-09-21	69	NULL
17	7	2023-09-22	199	NULL
18	8	2023-09-23	499	NULL
19	9	2023-09-24	159	NULL

Now making it foreign key refrencing from items table:-
Note:- (there is already another fkey available as customer_id referencing from customers table)

Alter table orders add constraint item_id foreign key (item_id) references items (item_id);



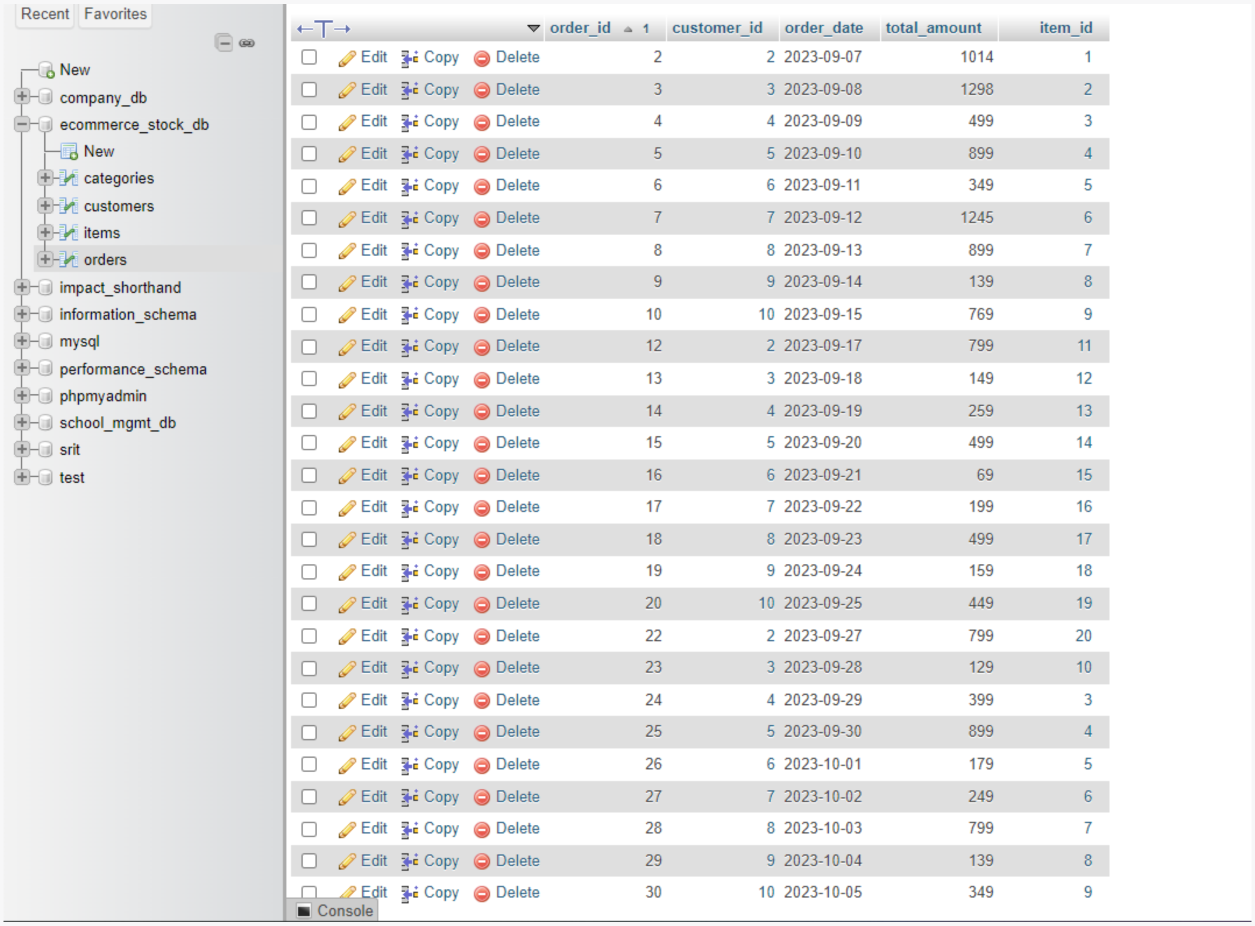
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	order_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	customer_id	int(10)			Yes	NULL			Change Drop More
3	order_date	date			Yes	NULL			Change Drop More
4	total_amount	int(11)			Yes	NULL			Change Drop More
5	item_id	int(11)			Yes	NULL			Change Drop More

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Rename Drop	PRIMARY	BTREE	Yes	No	order_id	27	A	No	
Edit Rename Drop	customer_id	BTREE	No	No	customer_id	27	A	Yes	

Now there are 2 fkeys in orders table 1st is customer_id and 2nd is items_id.

Now I will be adding or setting data into item_id associated with items table in item_id referencing from items table.

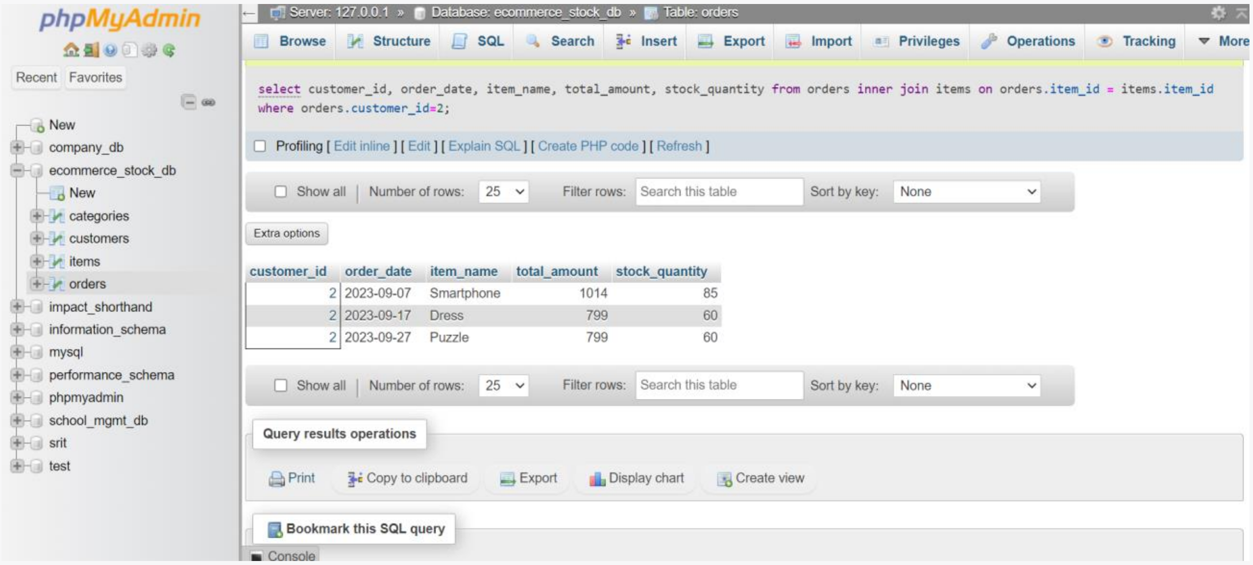
```
update orders set item_id=1 where orders.order_id=2;

update orders set item_id=2 where orders.order_id=3

update orders set item_id=3 where orders.order_id=4;

update orders set item_id=4 where orders.order_id=5;

update orders set item_id=5 where orders.order_id=6;

update orders set item_id=6 where orders.order_id=7;

update orders set item_id=7 where orders.order_id=8;

update orders set item_id=8 where orders.order_id=9;

update orders set item_id=9 where orders.order_id=10;

update orders set item_id=11 where orders.order_id=12;

update orders set item_id=12 where orders.order_id=13;

update orders set item_id=13 where orders.order_id=14;

update orders set item_id=14 where orders.order_id=15;

update orders set item_id=15 where orders.order_id=16;

update orders set item_id=16 where orders.order_id=17;

update orders set item_id=17 where orders.order_id=18;

update orders set item_id=18 where orders.order_id=19;

update orders set item_id=19 where orders.order_id=20;

update orders set item_id=20 where orders.order_id=22;
update orders set item_id=10 where orders.order_id=23;

update orders set item_id=3 where orders.order_id=24;

update orders set item_id=4 where orders.order_id=25;

update orders set item_id=5 where orders.order_id=26;

update orders set item_id=6 where orders.order_id=27;

update orders set item_id=7 where orders.order_id=28;

update orders set item_id=8 where orders.order_id=29;

update orders set item_id=9 where orders.order_id=30;
```



order_id	customer_id	order_date	total_amount	item_id
2	2	2023-09-07	1014	1
3	3	2023-09-08	1298	2
4	4	2023-09-09	499	3
5	5	2023-09-10	899	4
6	6	2023-09-11	349	5
7	7	2023-09-12	1245	6
8	8	2023-09-13	899	7
9	9	2023-09-14	139	8
10	10	2023-09-15	769	9
12	2	2023-09-17	799	11
13	3	2023-09-18	149	12
14	4	2023-09-19	259	13
15	5	2023-09-20	499	14
16	6	2023-09-21	69	15
17	7	2023-09-22	199	16
18	8	2023-09-23	499	17
19	9	2023-09-24	159	18
20	10	2023-09-25	449	19
22	2	2023-09-27	799	20
23	3	2023-09-28	129	10
24	4	2023-09-29	399	3
25	5	2023-09-30	899	4
26	6	2023-10-01	179	5
27	7	2023-10-02	249	6
28	8	2023-10-03	799	7
29	9	2023-10-04	139	8
30	10	2023-10-05	349	9

retrieving the order history for a specific customer, including order dates, item names, quantities, and total amounts spent.

```sql
select customer_id, order_date, item_name, total_amount, stock_quantity from orders inner join items on orders.item_id = items.item_id where orders.customer_id=2;
```



```sql
select customer_id, order_date, item_name, total_amount, stock_quantity from orders inner join items on orders.item_id = items.item_id
where orders.customer_id=2;
```

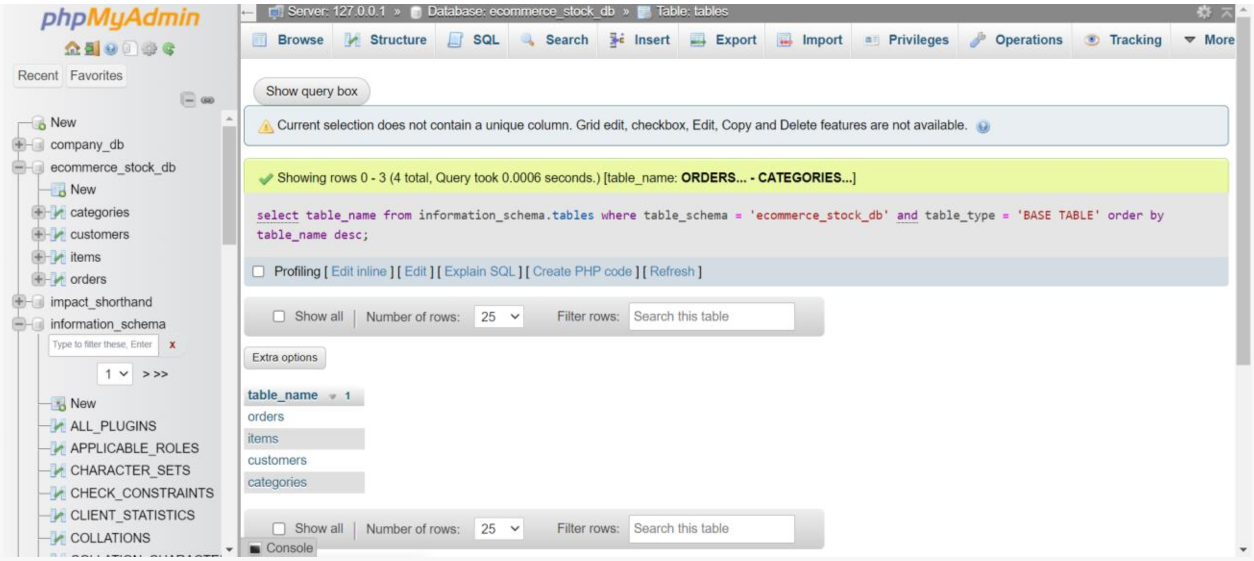
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

customer_id	order_date	item_name	total_amount	stock_quantity
2	2023-09-07	Smartphone	1014	85
2	2023-09-17	Dress	799	60
2	2023-09-27	Puzzle	799	60

Task 9: Implement ORDER BY DESC Write an SQL query to retrieve a list of all tables, ordering them in descending order by the `primary_key or id` column.

Ans:-

```sql
select table_name from information_schema.tables where table_schema = 'ecommerce_stock_db' and table_type = 'BASE TABLE' order by table_name desc;
```



⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.) [table_name: **ORDERS... - CATEGORIES...**]

```
select table_name from information_schema.tables where table_schema = 'ecommerce_stock_db' and table_type = 'BASE TABLE' order by
table_name desc;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

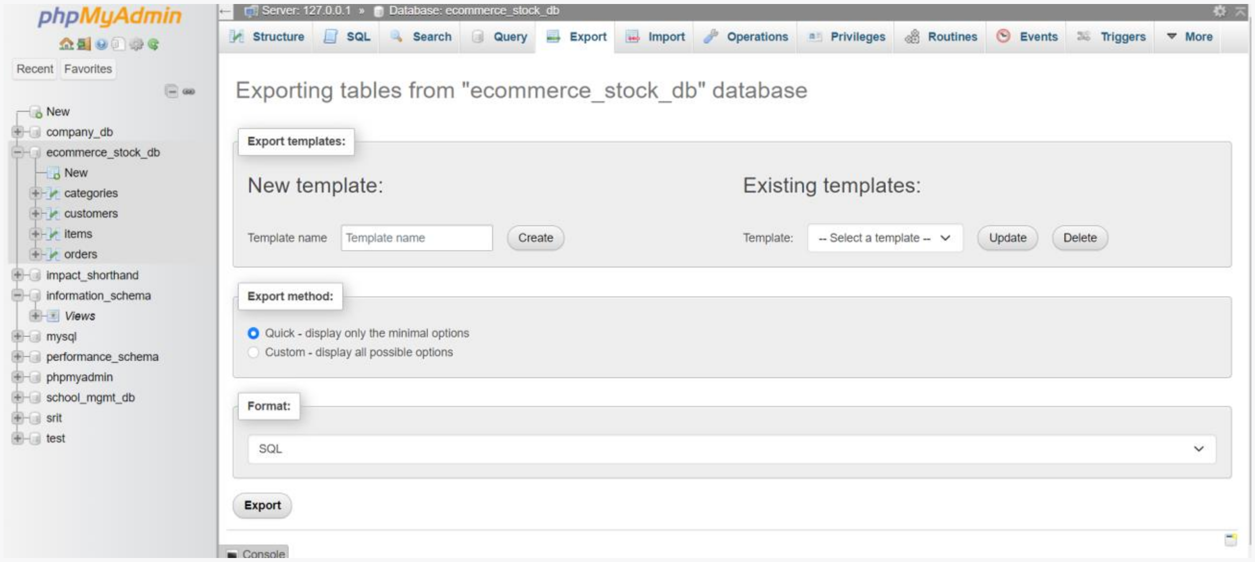
□ Show all | Number of rows: 25 Filter rows: Search this table

table_name ▼ 1
orders
items
customers
categories

□ Show all | Number of rows: 25 Filter rows: Search this table

Task 10: Backup and Restore Export a backup of the `ecommerce_stock_db` database to a SQL file. Then, write SQL statements to restore the database from the backup file. Submission:

Ans:-



Exporting tables from "ecommerce_stock_db" database

Export templates:

New template: Template name: Template name [Create]

Existing templates: Template: -- Select a template -- [Update] [Delete]

Export method:

● Quick - display only the minimal options
○ Custom - display all possible options

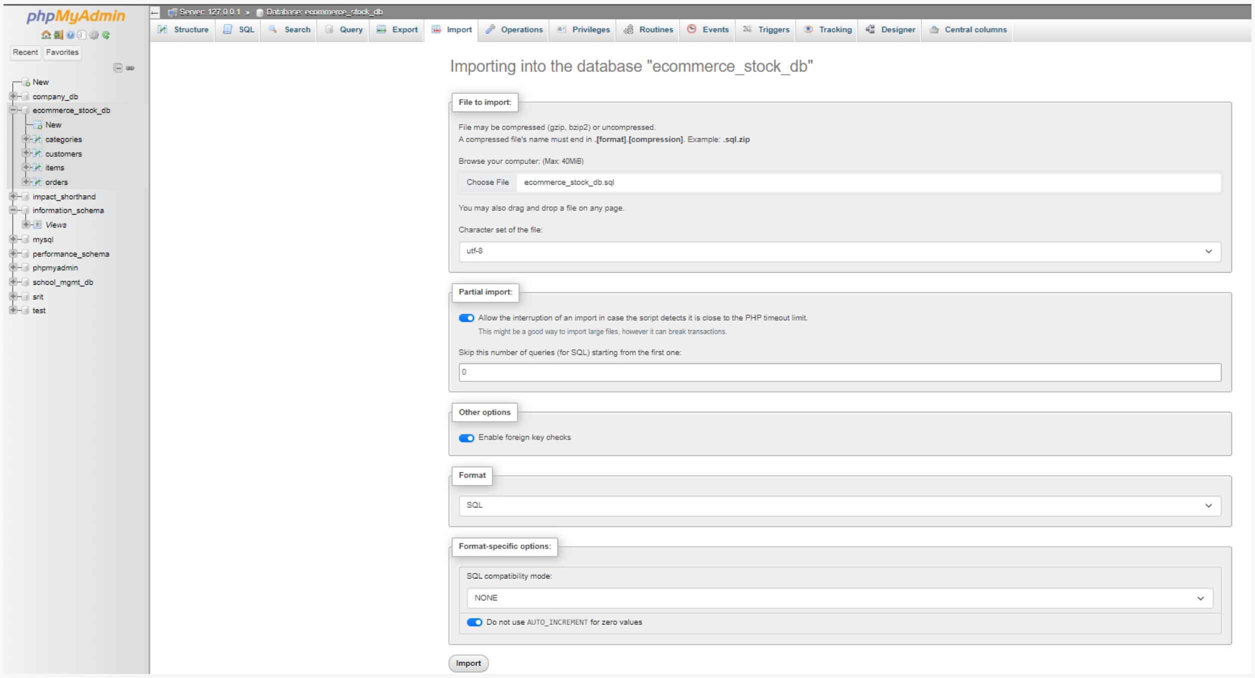
Format: SQL

[Export]

Export tab selected > Export method > quick -display only the minimal options and pressed Export button at bottom of the page which started downloading db file named ecommerce_stock_db.

File saved as ecommerce_stock_db.

Now to Import it
Import tab selected > in file import block choosed file ecommerce_stock_db > pressed import button to import db.



Importing into the database "ecommerce_stock_db"

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

Choose File ecommerce_stock_db.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

Partial import:

● Allow the interruption of an import in case the script detects it is close to the PHP timeout limit.
This might be a good way to import large files, however it can break transactions.

Skip this number of queries (for SQL) starting from the first one:

0

Other options

● Enable foreign key checks

Format

SQL

Format-specific options:

SQL compatibility mode:

NONE

● Do not use `AUTO_INCREMENT` for zero values

Import

- Save your SQL script with a meaningful filename (e.g., `ecommerce_assignment.sql`).

  :-Done, Please check above.
- Submit the SQL script by the due date specified.

  :-Done.