Online Shopping Management System

DBMS - Mini Project

Submitted By:

Name : Tanishka Garg

SRN : PES1UG20CS459

V Semester Section : H

Online Shopping Management System

Table of Content

Short Description and Scope of the Project

This project is about the admin view of the database of ecommerce website, where the admin can manage the data of orders, products purchased by the customers, products sold by sellers, along with the list of items available etc.

The application's frontend is build using python library streamlit and database is Mysql.

This database can be used by the admin of ecommerce website to view, add, delete, insert information about customers, sellers and products and also view the payment details. Customer details like customerid, Name, phone number is stored, various sellers details across the region like seller id, Name, phone number along with the product which they are selling is also stored and linked to each other using the relational database. Also the details of the Cart items, items orders is maintained in the relational database tables. Payments are also maintained in separate tables and are linked to various other tables using the foreign key.

Therefore it helps in maintaining and updating the data related to the ecommerce store.

ER Diagram

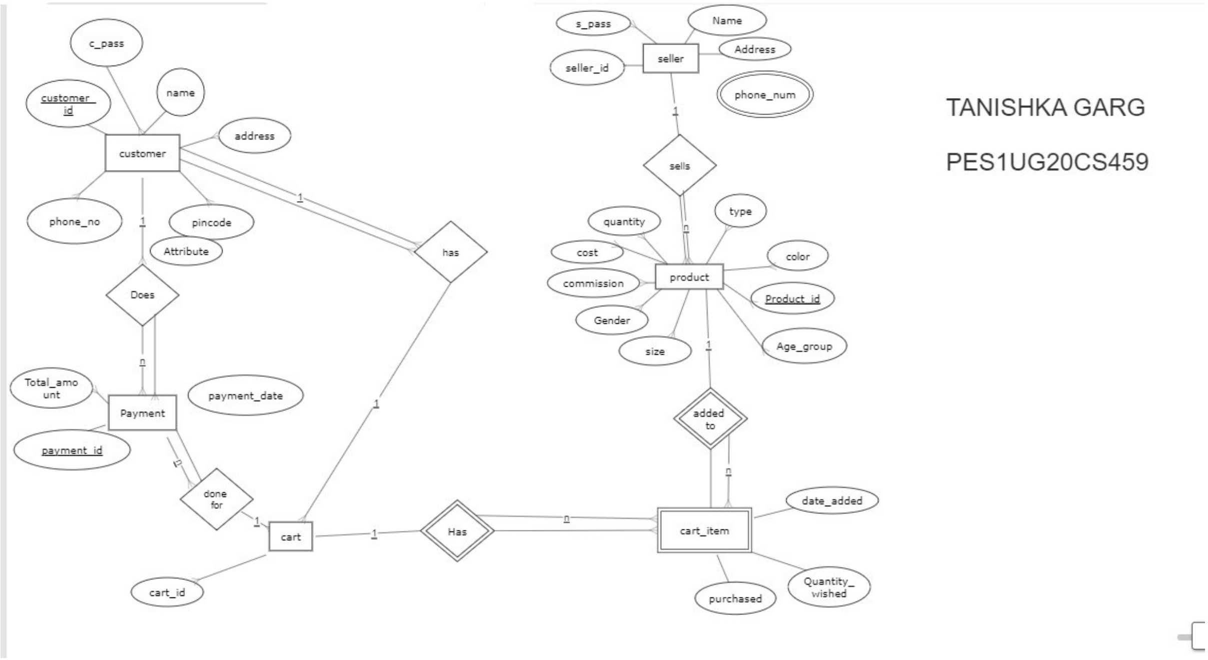


Fig1 – ER diagram

Relational Schema

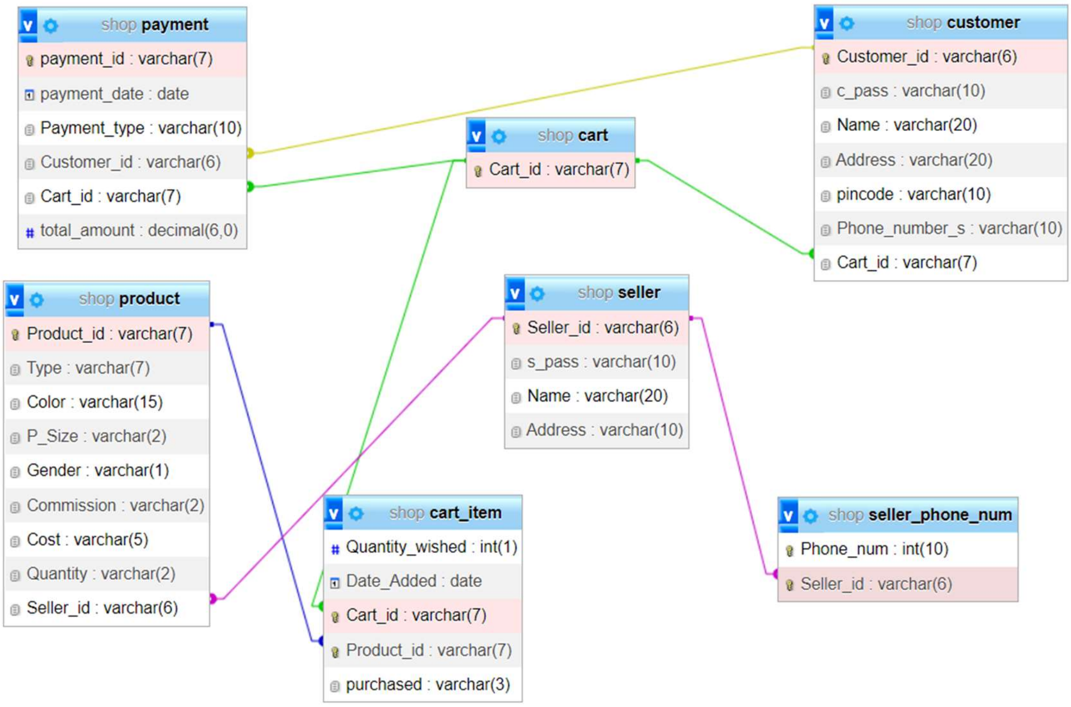


Fig2 – Relational Schema

DDL Statements

```
CREATE TABLE `cart` (
  `Cart_id` varchar(7) NOT NULL PRIMARY KEY
);
CREATE TABLE `cart_item` (
  `Quantity_wished` int(1) NOT NULL,
  `Date_Added` date NOT NULL,
  `Cart_id` varchar(7) NOT NULL,
  `Product_id` varchar(7) NOT NULL,
  `purchased` varchar(3) DEFAULT 'NO'
);
CREATE TABLE `customer` (
  `Customer_id` varchar(6) NOT NULL PRIMARY KEY,
  `c_pass` varchar(10) NOT NULL,
  `Name` varchar(20) NOT NULL,
  `Address` varchar(20) NOT NULL,
  `pincode` varchar(10) DEFAULT NULL,
  `Phone_number_s` varchar(10) NOT NULL,
  `Cart_id` varchar(7) NOT NULL
);
CREATE TABLE `payment` (
  `payment_id` varchar(7) NOT NULL PRIMARY KEY,
  `payment_date` date NOT NULL,
  `Payment_type` varchar(10) NOT NULL,
  `Customer_id` varchar(6) NOT NULL,
  `Cart_id` varchar(7) NOT NULL,
  `total_amount` decimal(6,0) DEFAULT NULL
);
CREATE TABLE `product` (
  `Product_id` varchar(7) NOT NULL PRIMARY KEY,
  `Type` varchar(7) NOT NULL,
  `Color` varchar(15) NOT NULL,
  `P_Size` varchar(2) NOT NULL,
  `Gender` varchar(1) NOT NULL,
  `Commission` varchar(2) NOT NULL,
  `Cost` varchar(5) NOT NULL,
  `Quantity` varchar(2) NOT NULL,
  `Seller_id` varchar(6) DEFAULT NULL
);
CREATE TABLE `seller` (
  `Seller_id` varchar(6) NOT NULL PRIMARY KEY,
  `s_pass` varchar(10) NOT NULL,
  `Name` varchar(20) NOT NULL,
  `Address` varchar(10) NOT NULL
);
CREATE TABLE `seller_phone_num` (
  `Phone_num` int(10) NOT NULL,
  `Seller_id` varchar(6) NOT NULL
);
ALTER TABLE `customer`ADD PRIMARY KEY (`Customer_id`),ADD KEY `Cart_id`
(`Cart_id`);
ALTER TABLE `seller_phone_num`ADD PRIMARY KEY (`Phone_num`,`Seller_id`),ADD
KEY `Seller_id` (`Seller_id`);
```

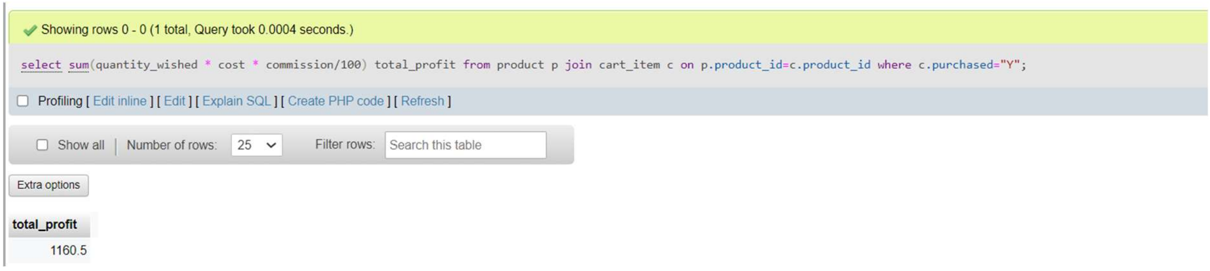
Populating data

```sql
INSERT INTO `cart` (`Cart_id`) VALUES
('crt1011'),
('crt1012'),
('crt1013'),
('crt1014'),
('crt1015');
INSERT INTO `cart_item` (`Quantity_wished`, `Date_Added`, `Cart_id`, `Product_id`,
`purchased`) VALUES
(1, '0000-00-00', 'crt1011', 'pid1001', 'Y'),
(3, '0000-00-00', 'crt1012', 'pid1004', 'NO');
INSERT INTO `customer` (`Customer_id`, `c_pass`, `Name`, `Address`, `pincode`,
`Phone_number_s`, `Cart_id`) VALUES
('cid100', 'ABCM1235', 'rajat', 'G-432', '632014', '2147483647', 'crt1011'),
('cid101', 'ABCM1236', 'niketan', 'G-454', '55786', '2147483647', 'crt1012'),
('cid102', 'ABCM1237', 'chinkuu', 'G-456', '65379', '2147483647', 'crt1013'),
('cid103', 'ABCM1238', 'sapnaaa', 'G-459', '32656', '2147483647', 'crt1014');
INSERT INTO `payment` (`payment_id`, `payment_date`, `Payment_type`, `Customer_id`,
`Cart_id`, `total_amount`) VALUES
('pmt1001', '0000-00-00', 'online', 'cid100', 'crt1011', NULL),
('pmt1002', '0000-00-00', 'online', 'cid100', 'crt1012', NULL),
('pmt1003', '0000-00-00', 'cash', 'cid102', 'crt1013', NULL),
('pmt1004', '0000-00-00', 'online', 'cid103', 'crt1014', NULL);
INSERT INTO `product` (`Product_id`, `Type`, `Color`, `P_Size`, `Gender`, `Commission`,
`Cost`, `Quantity`, `Seller_id`) VALUES
('pid1001', 'jeans', 'red', '32', 'M', '10', '10005', '0', 'sid100'),
('pid1002', 'top', 'red', '30', 'F', '12', '500', '0', 'sid103'),
('pid1003', 'purse', 'purple', '32', 'F', '10', '800', '0', 'sid103'),
('pid1004', 'belt', 'brown', '30', 'M', '11', '300', '0', 'sid106'),
('pid1008', 'wallet', 'brown', '10', 'M', '10', '600.0', '3.', 'sid100');
INSERT INTO `seller` (`Seller_id`, `s_pass`, `Name`, `Address`) VALUES
('sid100', '12345', 'amannnnn', 'delhi  '),
('sid103', '96543', 'nikatan', 'agra'),
('sid106', '98723', 'phangar', 'delhi cmc'),
('sid108', '98745', 'Naman', 'jaipur'),
('sid109', '67523', 'tani', 'bangalore');
INSERT INTO `seller_phone_num` (`Phone_num`, `Seller_id`) VALUES
(906416370, 'sid100'),
(906486537, 'sid103'),
(990016870, 'sid100');
```

Join Queries

1) Find total profit of website from sales

select sum(quantity_wished * cost * commission/100) total_profit from product p join cart_item c on p.product_id=c.product_id where c.purchased="Y";



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

```
select sum(quantity_wished * cost * commission/100) total_profit from product p join cart_item c on p.product_id=c.product_id where c.purchased="Y";
```

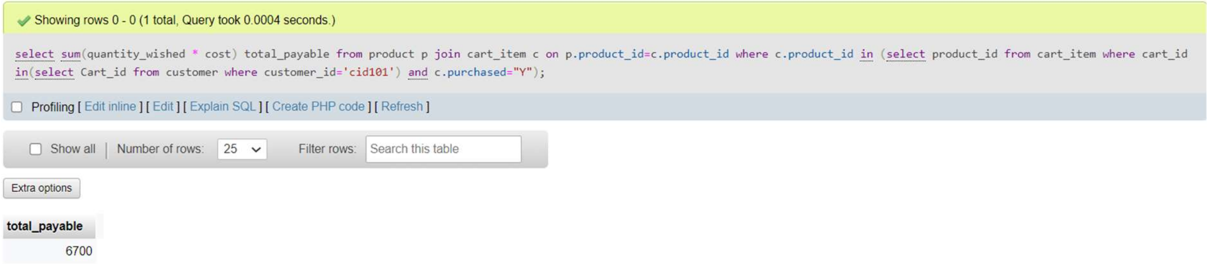
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

total_profit
1160.5

Fig3 - Join

2) If a customer want to know total price present in cart

select sum(quantity_wished * cost) total_payable from product p join cart_item c on p.product_id=c.product_id where c.product_id in (select product_id from cart_item where cart_id in(select Cart_id from customer where customer_id='cid101') and c.purchased="Y");



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

```
select sum(quantity_wished * cost) total_payable from product p join cart_item c on p.product_id=c.product_id where c.product_id in (select product_id from cart_item where cart_id
in(select Cart_id from customer where customer_id='cid101') and c.purchased="Y");
```

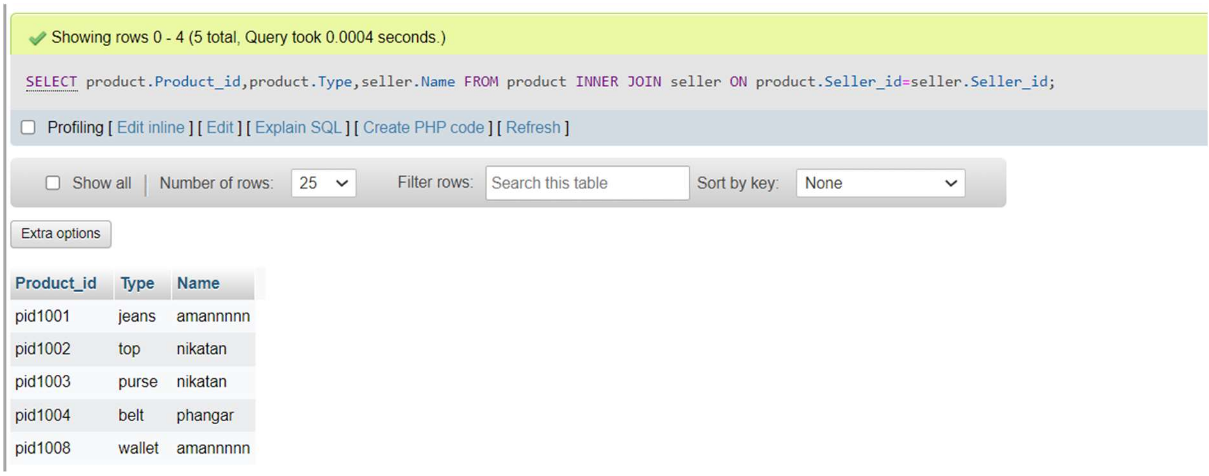
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

total_payable
6700

Fig4 – Join

3) Display the name of the sellers along with the items they are selling

SELECT product.Product_id,product.Type,seller.Name FROM product INNER JOIN seller ON product.Seller_id=seller.Seller_id;



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

```
SELECT product.Product_id,product.Type,seller.Name FROM product INNER JOIN seller ON product.Seller_id=seller.Seller_id;
```

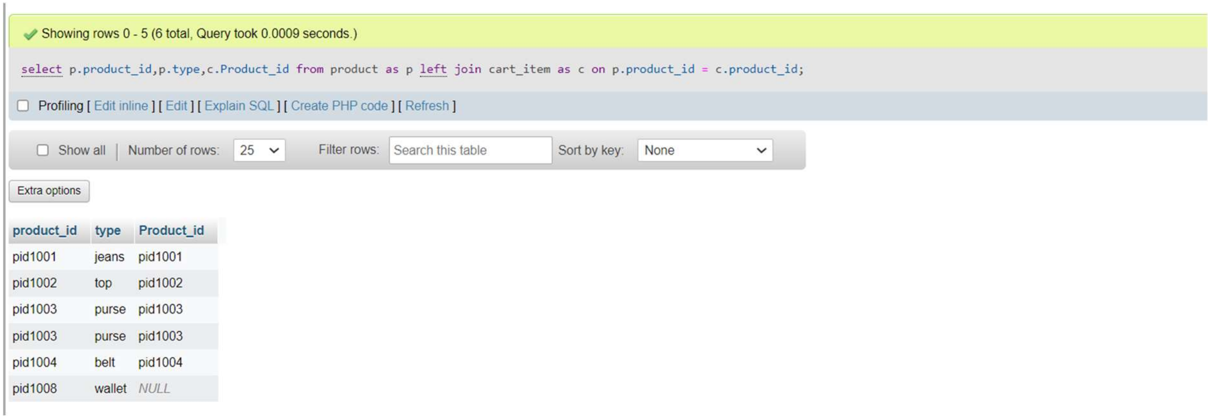
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

Product_id	Type	Name
pid1001	jeans	amannnnn
pid1002	top	nikatan
pid1003	purse	nikatan
pid1004	belt	phangar
pid1008	wallet	amannnnn

Fig5 - Join

4) Display product type which are present in store but not necessary in a cart

select p.product_id,p.type,c.Product_id from product as p left join cart_item as c on p.product_id = c.product_id;



✓ Showing rows 0 - 5 (6 total, Query took 0.0009 seconds.)

```
select p.product_id,p.type,c.Product_id from product as p left join cart_item as c on p.product_id = c.product_id;
```

□ 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

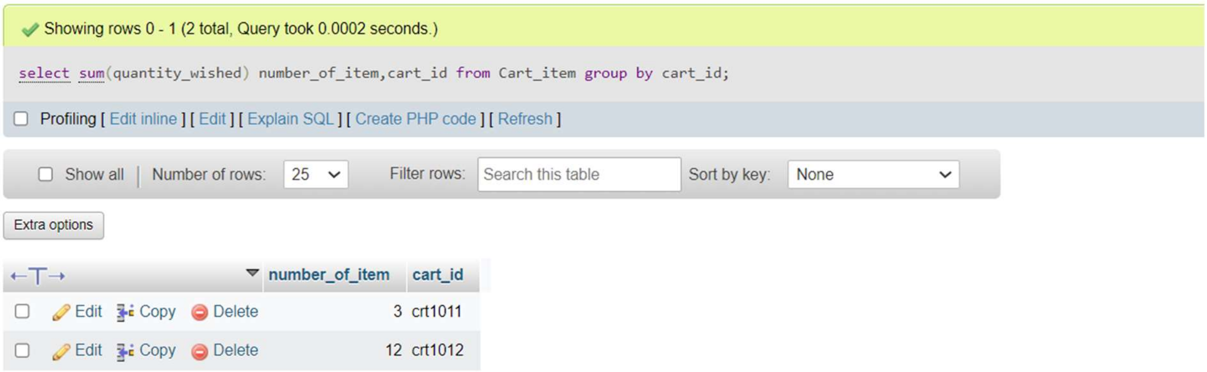
product_id	type	Product_id
pid1001	jeans	pid1001
pid1002	top	pid1002
pid1003	purse	pid1003
pid1003	purse	pid1003
pid1004	belt	pid1004
pid1008	wallet	NULL

Fig6 - Join

Aggregate Functions

1) Total number of items added in a cart

select sum(quantity_wished) number_of_item,cart_id from Cart_item group by cart_id;



✓ Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
select sum(quantity_wished) number_of_item,cart_id from Cart_item group by cart_id;
```

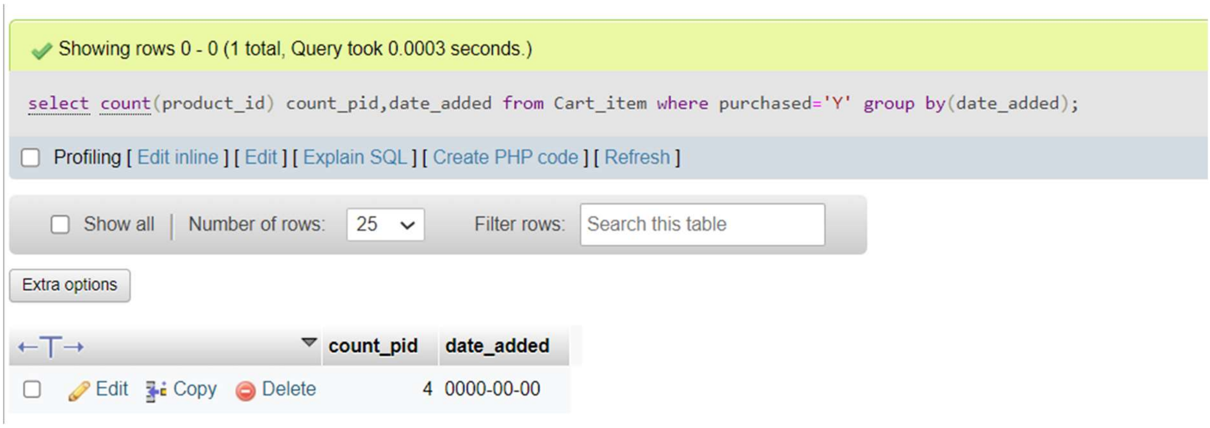
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
←T→

	number_of_item	cart_id
□ Edit Copy Delete	3	crt1011
□ Edit Copy Delete	12	crt1012

Fig7 - Aggregate

2) Number of products ordered on a particular date

select count(product_id) count_pid,date_added from Cart_item where purchased='Y' group by(date_added);



✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
select count(product_id) count_pid,date_added from Cart_item where purchased='Y' group by(date_added);
```

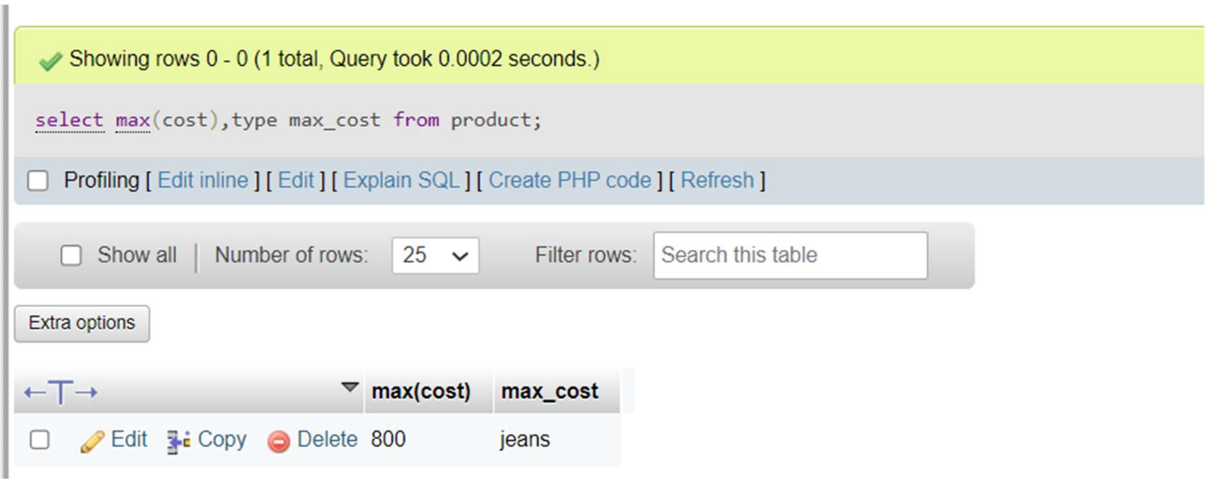
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25
Filter rows: Search this table
Extra options
←T→

	count_pid	date_added
□ Edit Copy Delete	4	0000-00-00

Fig8 - Aggregate

3) Highest priced item

select max(cost),type max_cost from product;



✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

```
select max(cost),type max_cost from product;
```

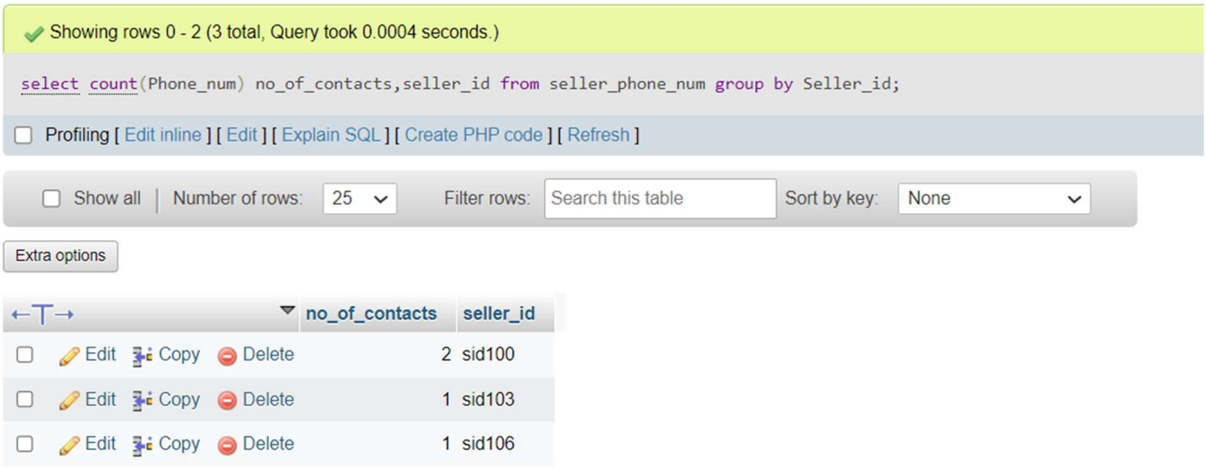
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

max(cost)	max_cost
800	jeans

Fig9 - Aggregate

4) Number of phone numbers that each seller has

select count(Phone_num) no_of_contacts,seller_id from seller_phone_num group by Seller_id;



✓ Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
select count(Phone_num) no_of_contacts,seller_id from seller_phone_num group by Seller_id;
```

□ 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

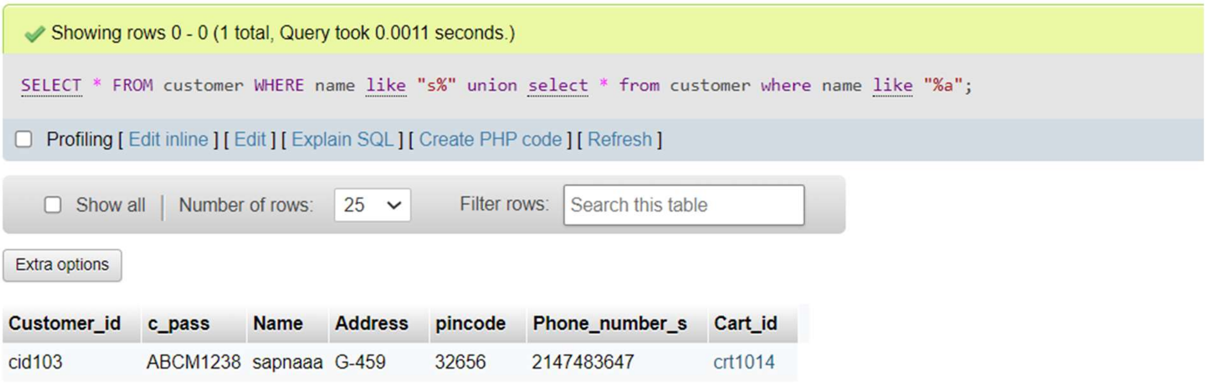
no_of_contacts	seller_id
2	sid100
1	sid103
1	sid106

Fig10 - Aggregate

Set Functions

1) Customer names which either start with ‘s’ or end with ‘a’

SELECT * FROM customer WHERE name like "s%" union select * from customer where name like "%a";



✓ Showing rows 0 - 0 (1 total, Query took 0.0011 seconds.)

```
SELECT * FROM customer WHERE name like "s%" union select * from customer where name like "%a";
```

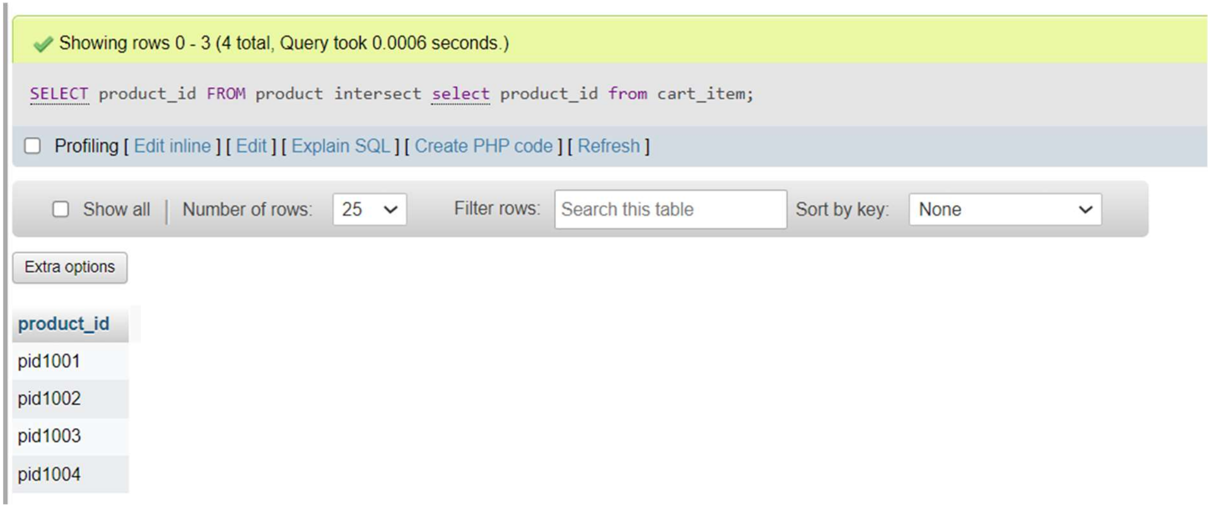
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

Customer_id	c_pass	Name	Address	pincode	Phone_number_s	Cart_id
cid103	ABCM1238	sapnaaa	G-459	32656	2147483647	crt1014

Fig11 - Set

2) Products which are added to cart to show in demand products

SELECT product_id FROM product intersect select product_id from cart_item;



✓ Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT product_id FROM product intersect select product_id from cart_item;
```

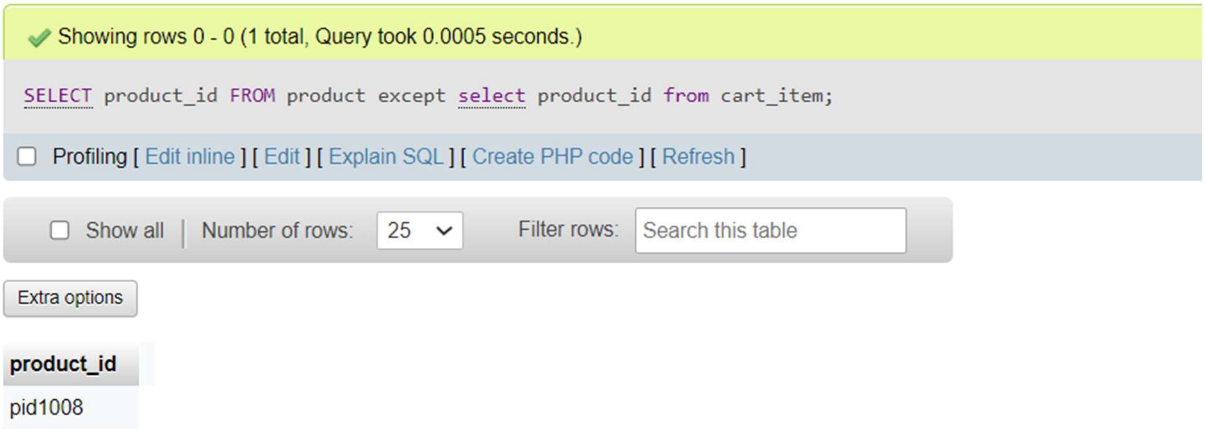
□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Extra options

product_id
pid1001
pid1002
pid1003
pid1004

Fig12 - Set

3) Products which are never added to cart

SELECT product_id FROM product except select product_id from cart_item;



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

```
SELECT product_id FROM product except select product_id from cart_item;
```

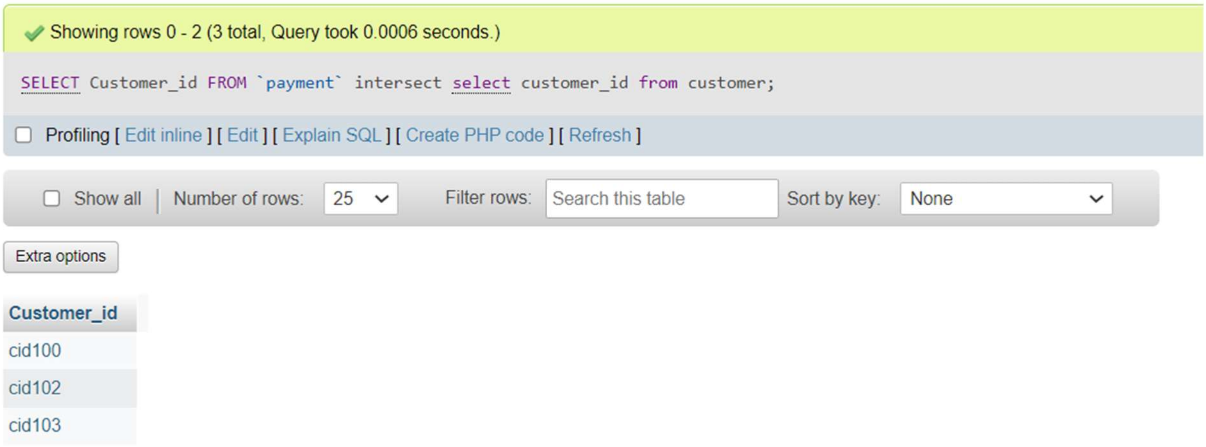
☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
☐ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

product_id
pid1008

Fig13 - Set

4) Customers who are doing/done payments

SELECT Customer_id FROM payment intersect select customer_id from customer;



✓ Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.)

```
SELECT Customer_id FROM `payment` intersect select customer_id from customer;
```

☐ 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

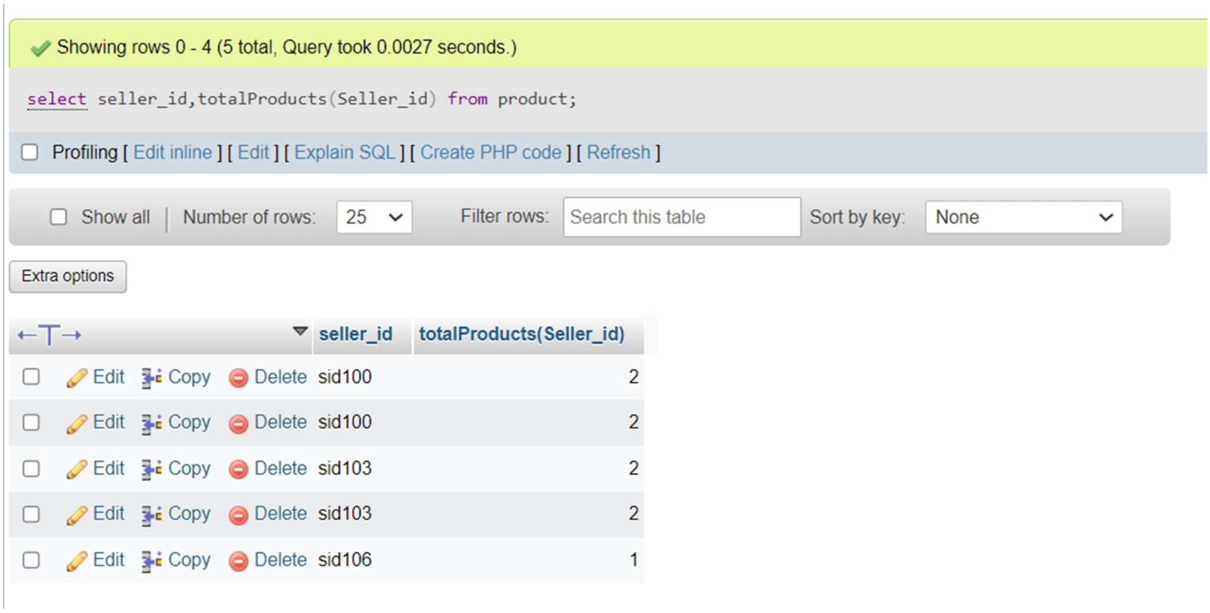
Customer_id
cid100
cid102
cid103

Fig14 - Set

Function and Procedures

1) Total number of products which a particular seller sells

```
CREATE FUNCTION totalProducts(sId varchar(6))
RETURNS int(3)
DETERMINISTIC
BEGIN
declare total int default 0;
select count(*) into total
from product
where seller_id=sId;
return total;
end
select seller_id,totalProducts(Seller_id) from product;
```



✓ Showing rows 0 - 4 (5 total, Query took 0.0027 seconds.)

```
select seller_id,totalProducts(Seller_id) from product;
```

□ 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

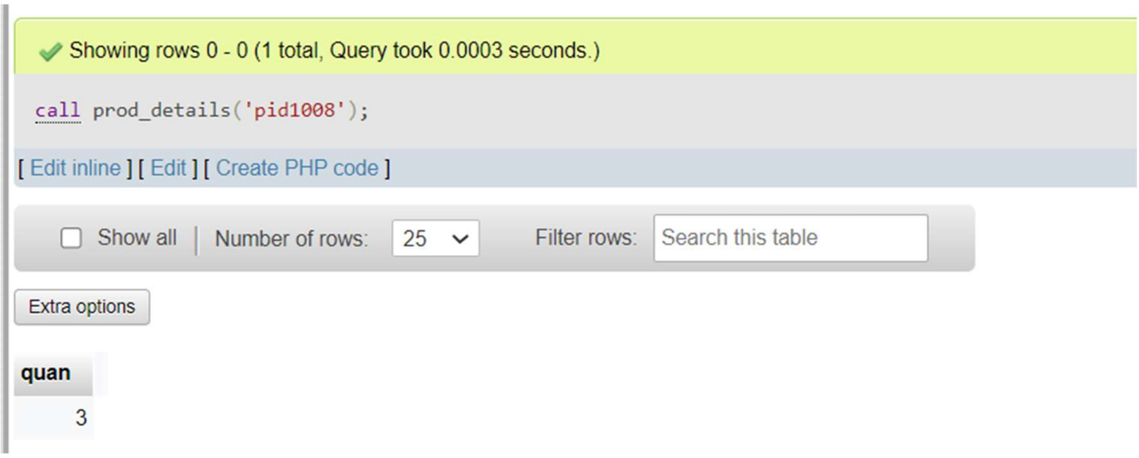
				seller_id	totalProducts(Seller_id)
□	Edit	Copy	Delete	sid100	2
□	Edit	Copy	Delete	sid100	2
□	Edit	Copy	Delete	sid103	2
□	Edit	Copy	Delete	sid103	2
□	Edit	Copy	Delete	sid106	1

Fig15 - Function

2) Quantity of product with given ID

```
DELIMITER $$
CREATE procedure prod_details(p_id varchar(10))
BEGIN
declare quan int(2) default 0;
   select quantity into quan from product where product_id=p_id;
```

```
END $$
DELIMITER ;
call prod_details('pid1008');
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0003 seconds.)

```
call prod_details('pid1008');
```

[Edit inline] [Edit] [Create PHP code]
□ Show all | Number of rows: 25
Filter rows: Search this table
Extra options

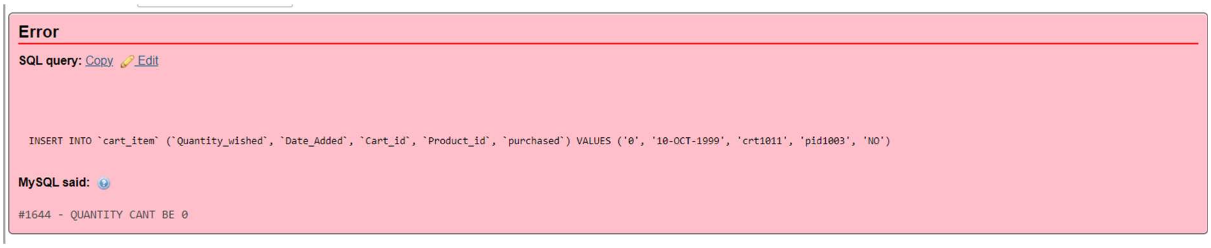
quan
3

Fig16 - Procedure

Trigger and Cursor

1) If quantity entered is 0 trigger is invoked displaying error.

```
DELIMITER //
CREATE TRIGGER quan_check BEFORE INSERT ON `cart_item`
 FOR EACH ROW BEGIN
    IF new.quantity_wished = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'QUANTITY CANT BE 0';
    END IF;
    end;
```



Error

SQL query:

```
INSERT INTO `cart_item` (`Quantity_wished`, `Date_Added`, `Cart_id`, `Product_id`, `purchased`) VALUES ('0', '10-OCT-1999', 'crt1011', 'pid1003', 'NO')
```

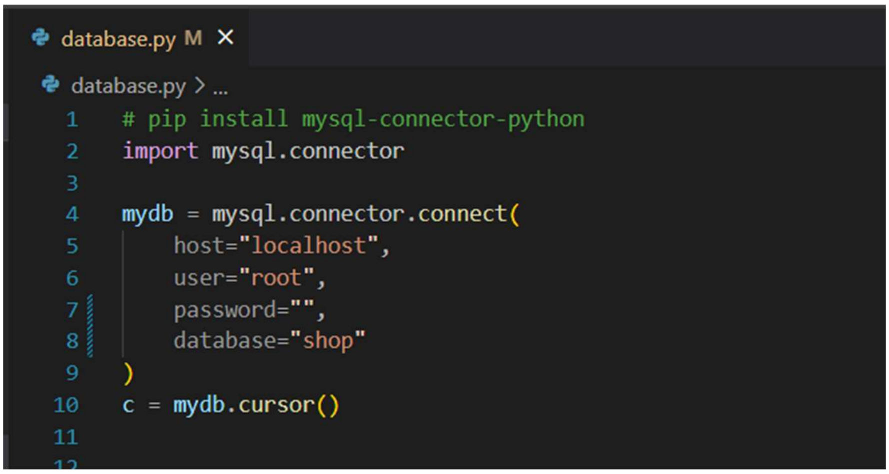
MySQL said:

```
#1644 - QUANTITY CANT BE 0
```

Fig17 - Trigger

2) Cursor has been used in frontend

c = mydb.cursor();



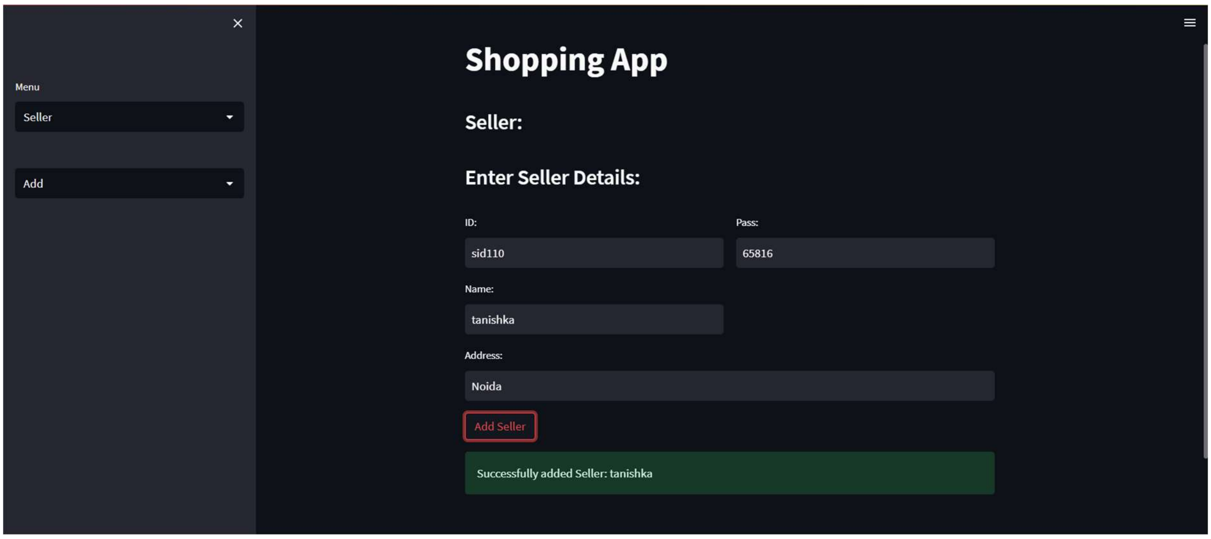
```python
# pip install mysql-connector-python
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="shop"
)
c = mydb.cursor()

```

Fig18 - Cursor

Frontend



Shopping App

Seller:

Enter Seller Details:

ID: sid110

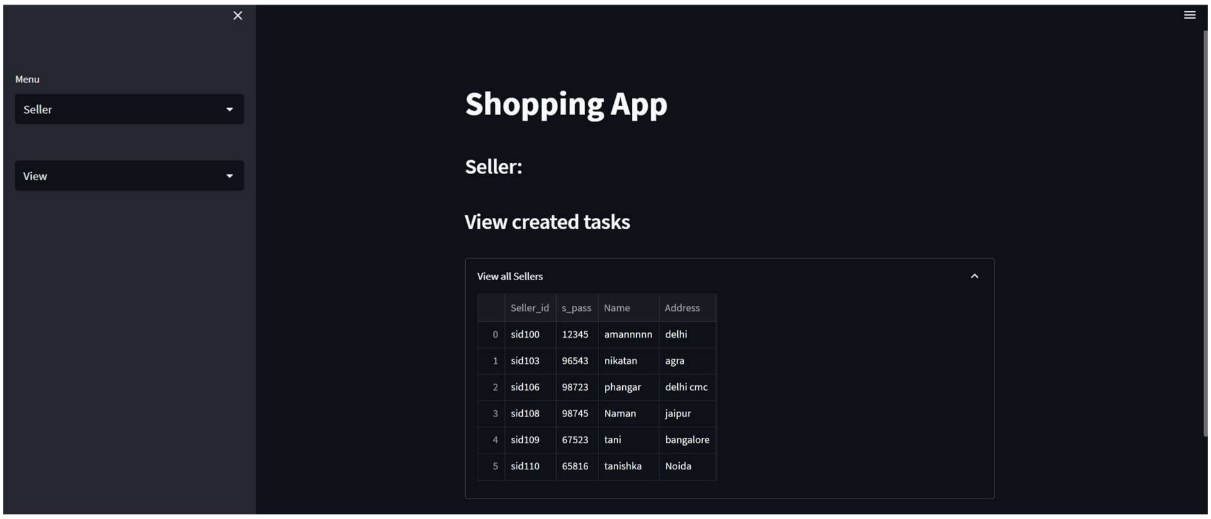
Pass: 65816

Name: tanishka

Address: Noida

Successfully added Seller: tanishka

Fig19 - Add



Shopping App

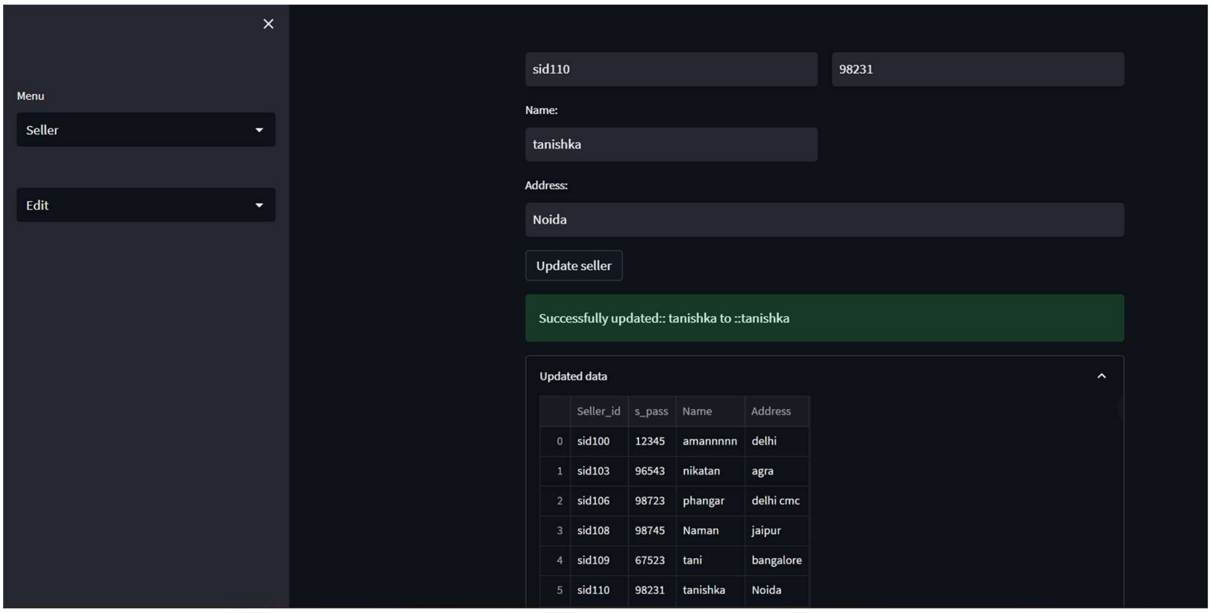
Seller:

View created tasks

View all Sellers

	Seller_id	s_pass	Name	Address
0	sid100	12345	amannnnn	delhi
1	sid103	96543	nikatan	agra
2	sid106	98723	phangar	delhi cmc
3	sid108	98745	Naman	jaipur
4	sid109	67523	tani	bangalore
5	sid110	65816	tanishka	Noida

Fig20 - View



sid110

98231

Name:

tanishka

Address:

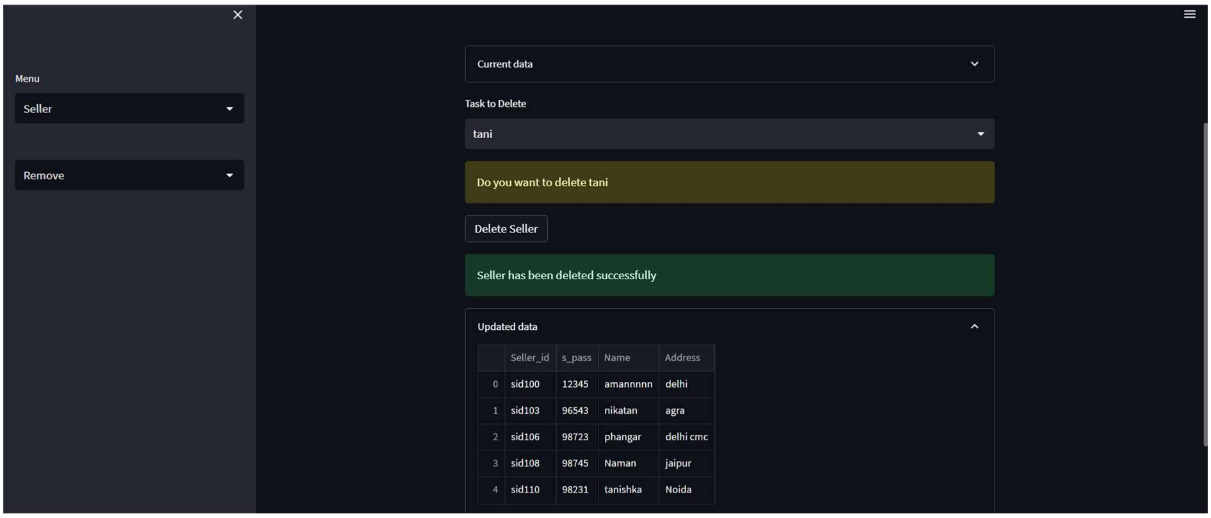
Noida

Successfully updated:: tanishka to ::tanishka

Updated data

	Seller_id	s_pass	Name	Address
0	sid100	12345	amannnnn	delhi
1	sid103	96543	nikatan	agra
2	sid106	98723	phangar	delhi cmc
3	sid108	98745	Naman	jaipur
4	sid109	67523	tani	bangalore
5	sid110	98231	tanishka	Noida

Fig21 - Edit



Task to Delete

tani

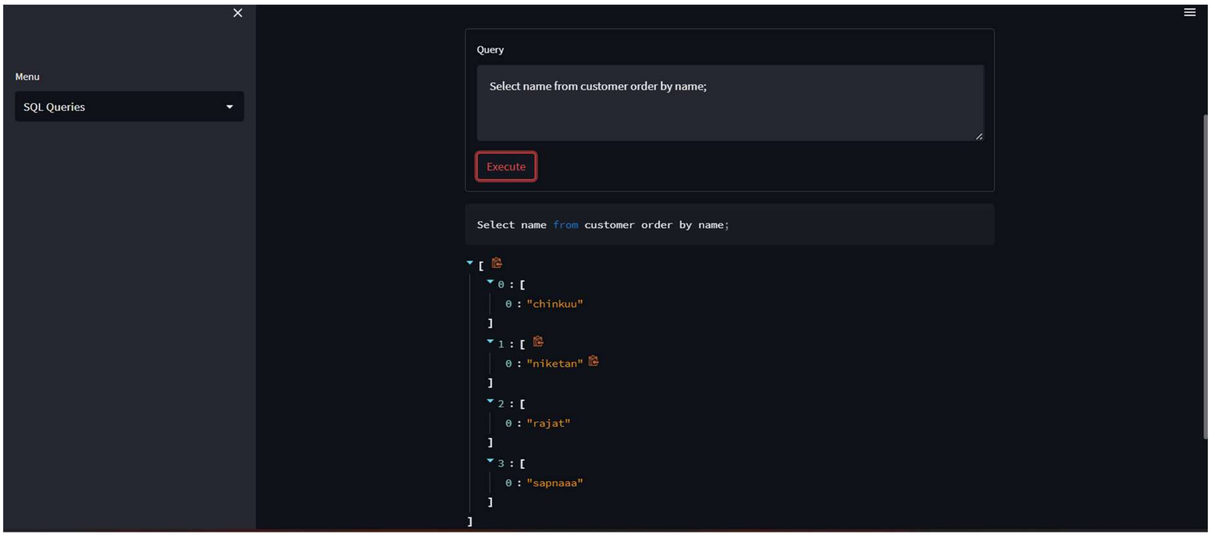
Do you want to delete tani

Seller has been deleted successfully

Updated data

	Seller_id	s_pass	Name	Address
0	sid100	12345	amannnnn	delhi
1	sid103	96543	nikatan	agra
2	sid106	98723	phangar	delhi cmc
3	sid108	98745	Naman	jaipur
4	sid110	98231	tanishka	Noida

Fig 22 – Delete



Query

Select name from customer order by name;

```
Select name from customer order by name;
```

```
[
  0 : [
    0 : "chinkuu"
  ]
  1 : [
    0 : "niketan"
  ]
  2 : [
    0 : "rajat"
  ]
  3 : [
    0 : "sapnaaa"
  ]
]
```

Fig23 – SQL Query