**DBMS - Mini Project**

Submitted By:

Name : Tanishka Garg

SRN : PES1UG20CS459

V Semester Section : H

**Table of Content**

1. Short description and Scope of project
2. ER Diagram
3. Relational Schema
4. DDL Statements
5. Populating the data
6. Join Queries
7. Aggregate Functions
8. Set Functions
9. Functions and Procedures
10. Triggers and Cursors
11. Frontend

**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**

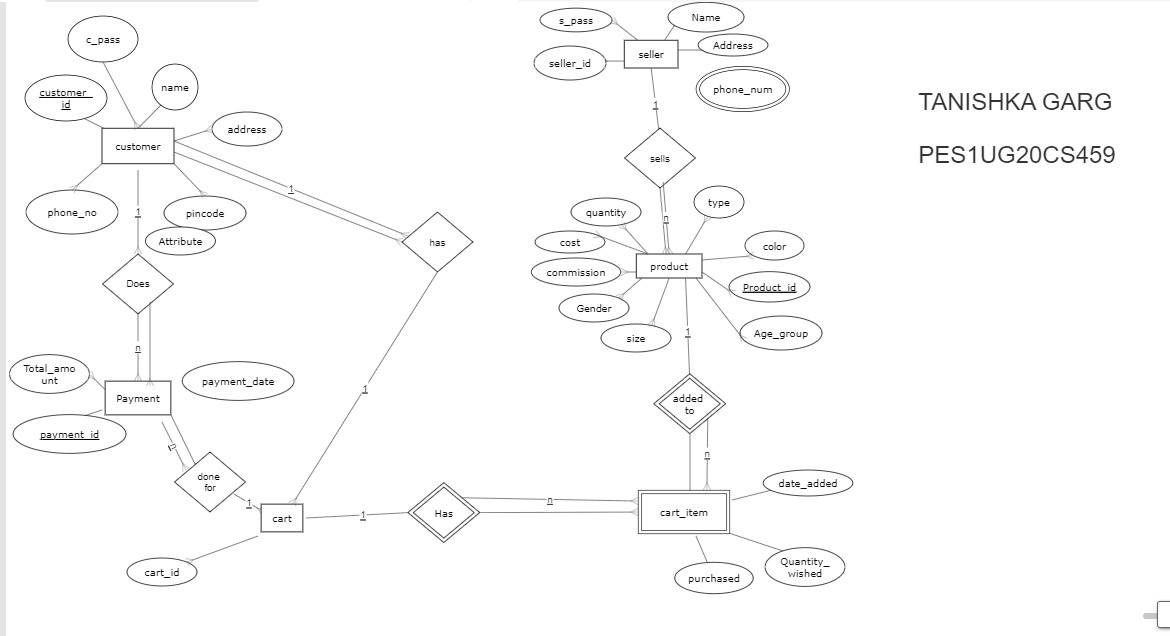


Fig1 – ER diagram

**Relational Schema**

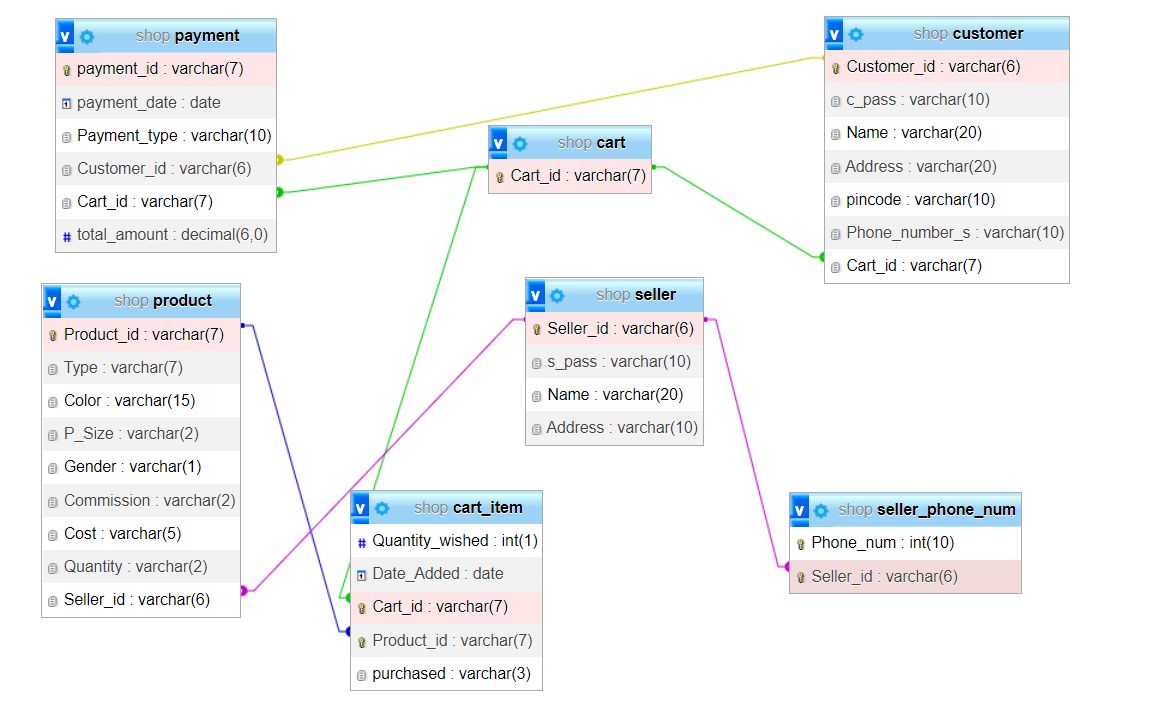
****

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**

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

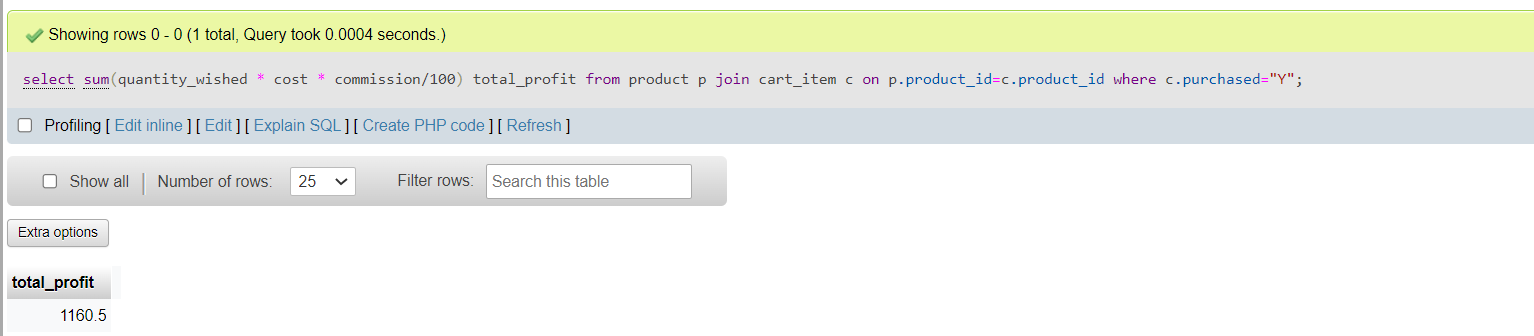
select sum(quantity\_wished \* cost \* commission/100) total\_profit from product p join cart\_item c on p.product\_id=c.product\_id where c.purchased="Y";

Fig3 - Join

1. 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");

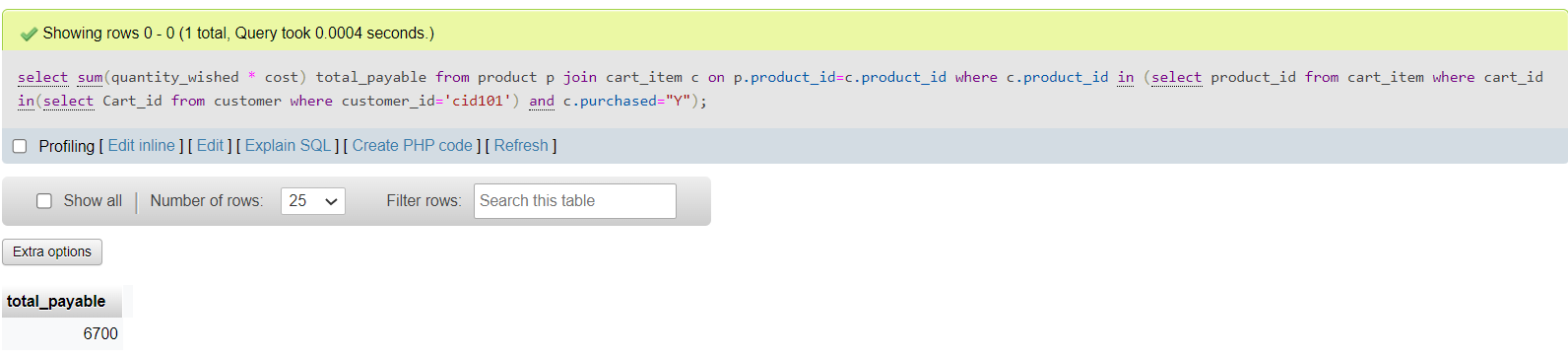
****

Fig4 – Join

1. 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;

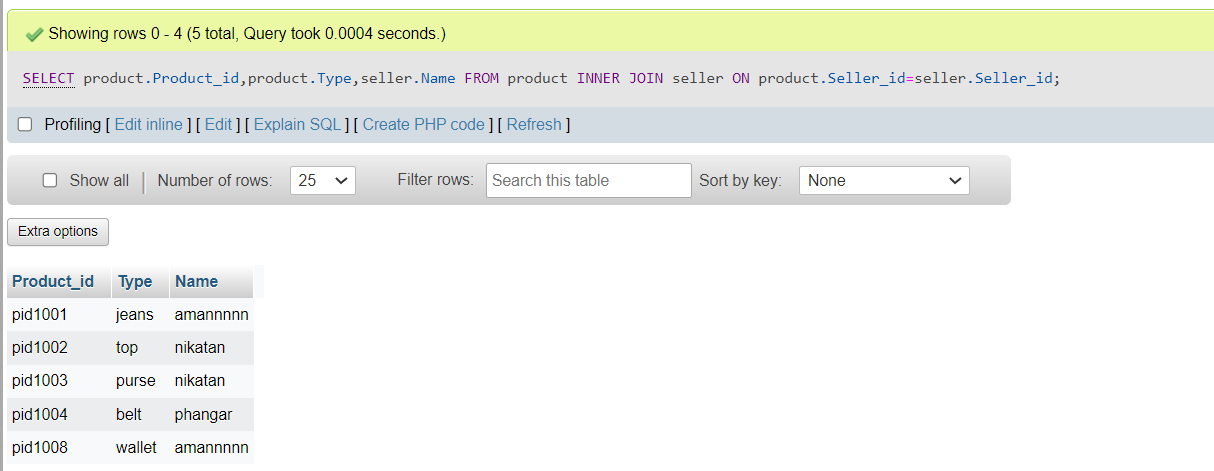
****

Fig5 - Join

1. Display product type which are present in store but not necessary in a cart

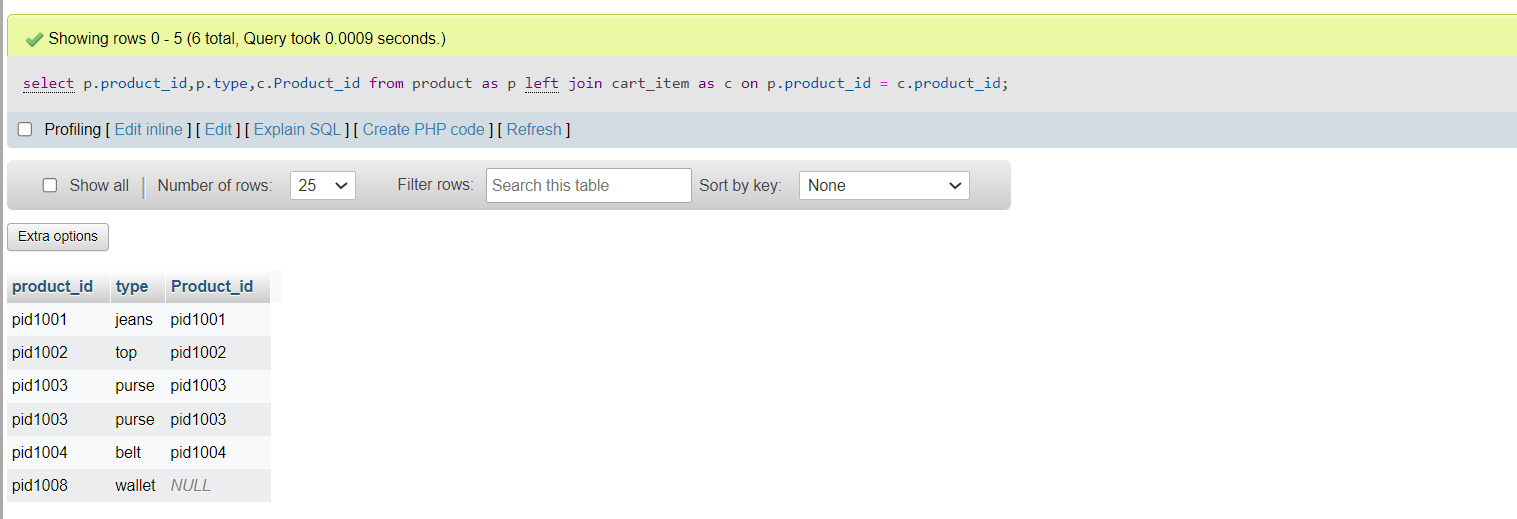
select p.product\_id,p.type,c.Product\_id from product as p left join cart\_item as c on p.product\_id = c.product\_id; 

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;

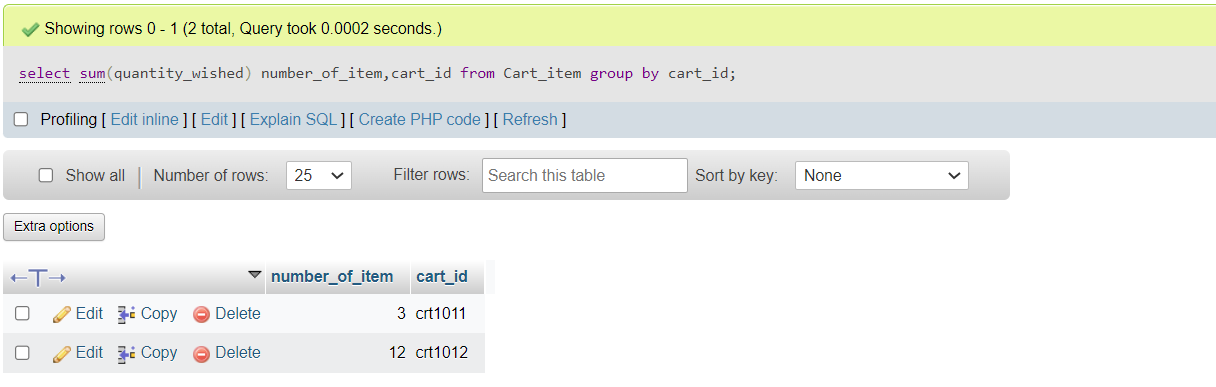


Fig7 - Aggregate

1. 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);

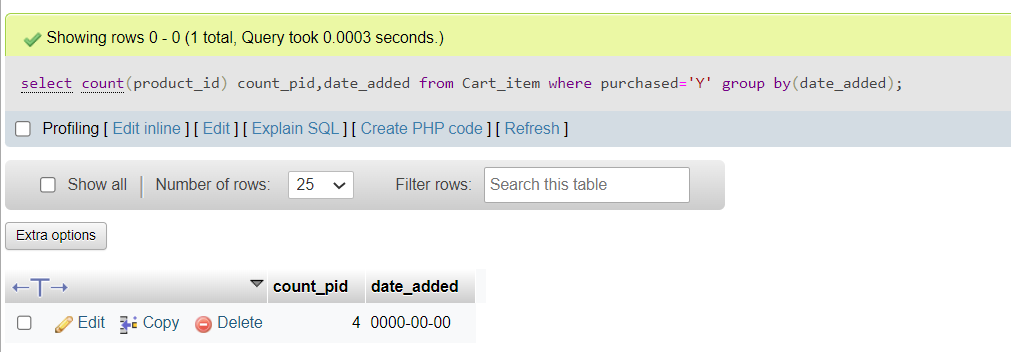


Fig8 - Aggregate

1. Highest priced item

select max(cost),type max\_cost from product;

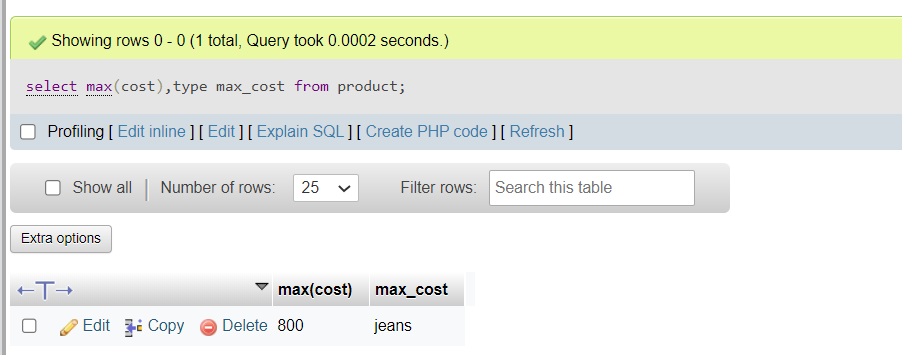


Fig9 - Aggregate

1. 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;

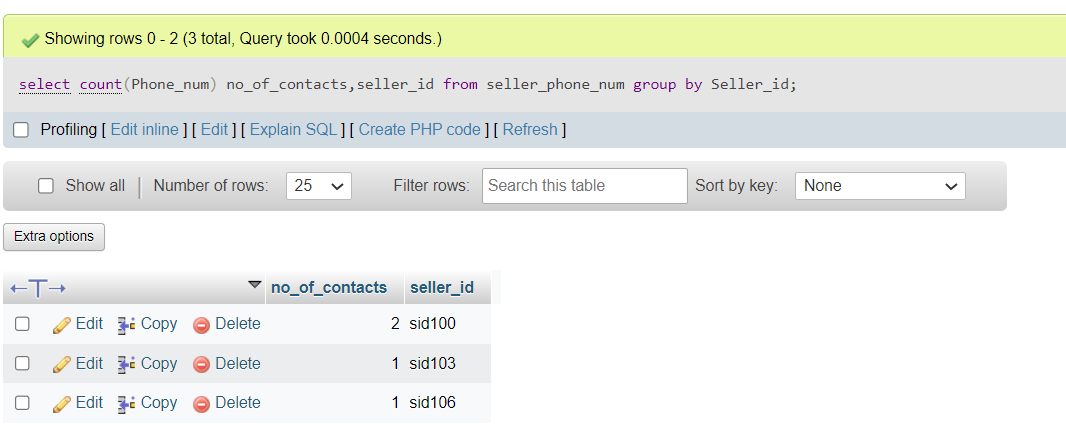


Fig10 - Aggregate

**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";

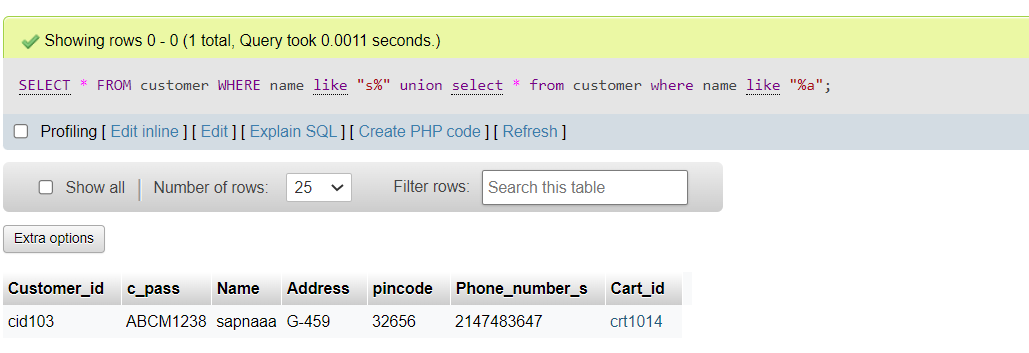


Fig11 - Set

1. Products which are added to cart to show in demand products

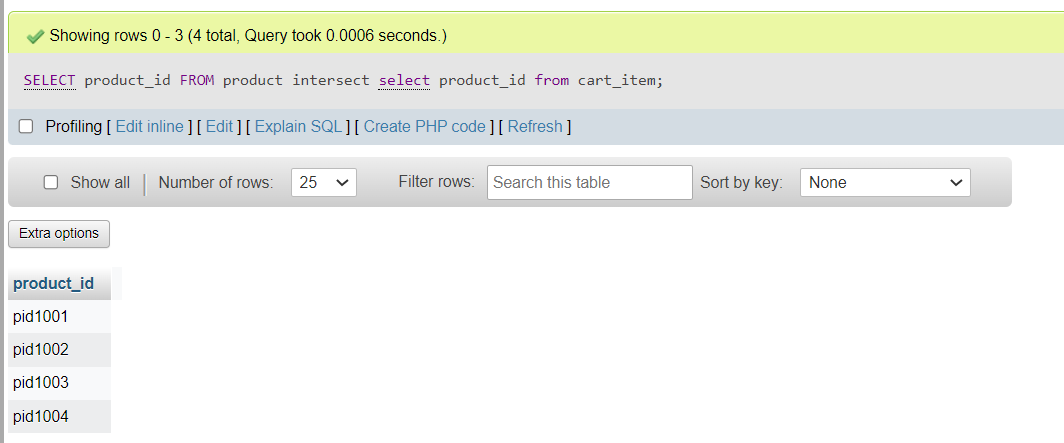
SELECT product\_id FROM product intersect select product\_id from cart\_item;

Fig12 - Set

1. Products which are never added to cart

SELECT product\_id FROM product except select product\_id from cart\_item;

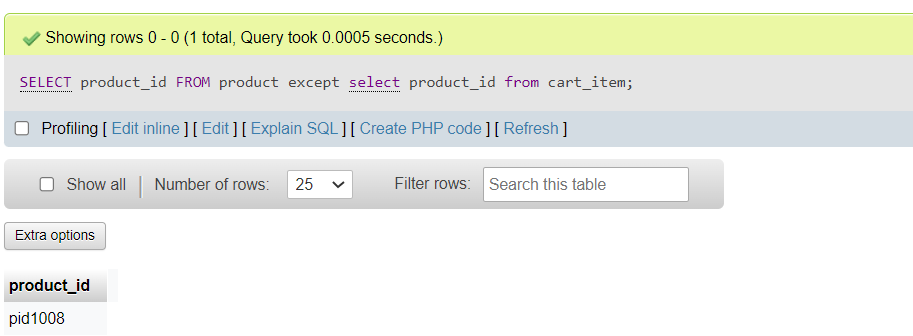


Fig13 - Set

1. Customers who are doing/done payments

SELECT Customer\_id FROM payment intersect select customer\_id from customer;

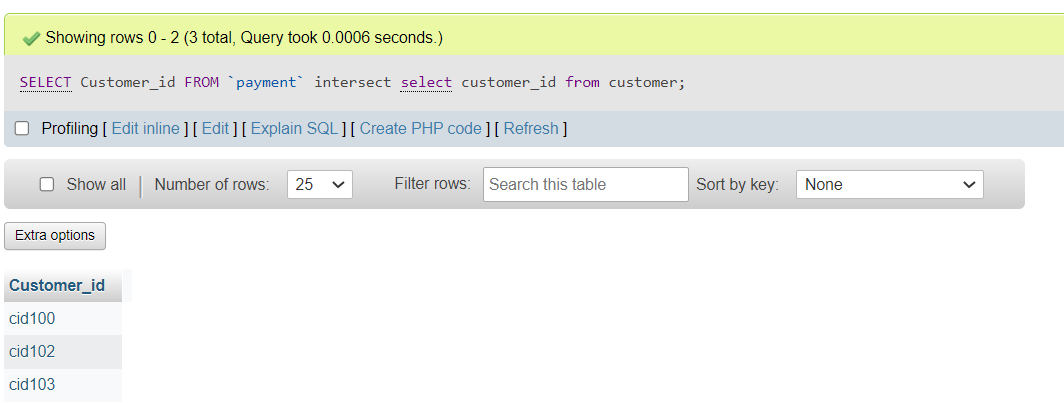


Fig14 - Set

**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;

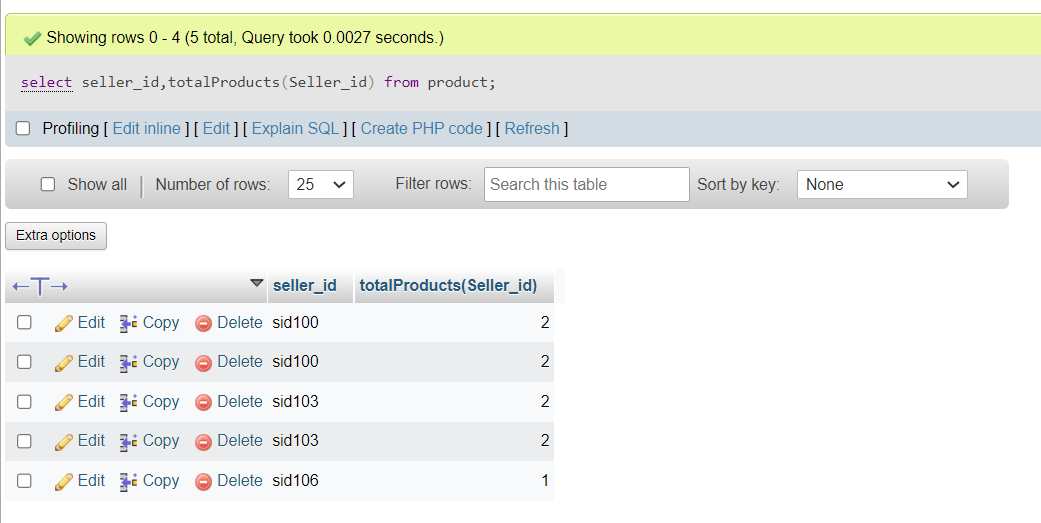


Fig15 - Function

1. 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');

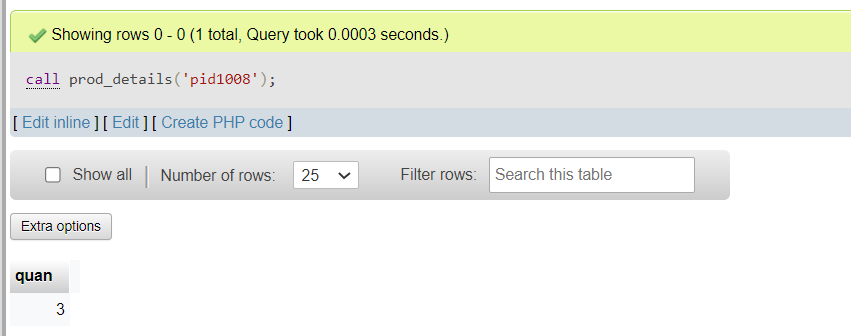


Fig16 - Procedure

**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;

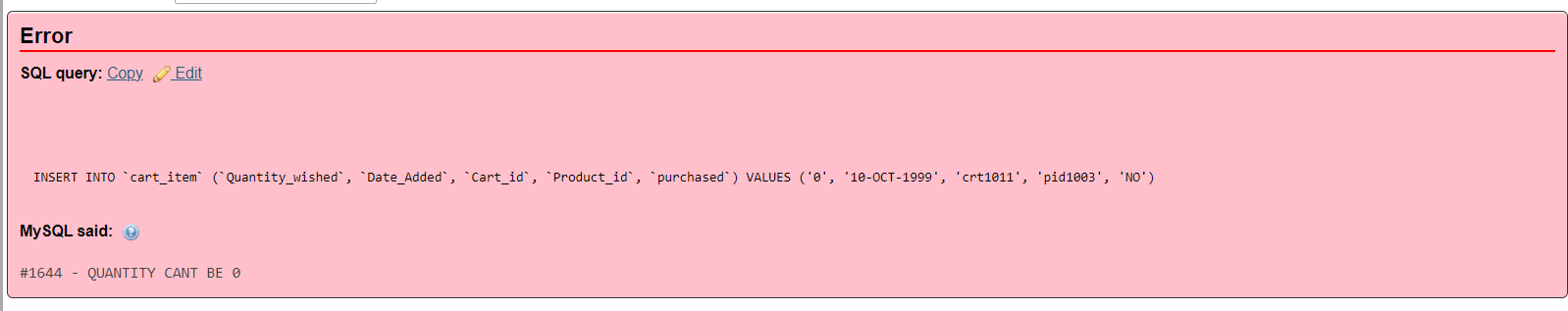


Fig17 - Trigger

1. Cursor has been used in frontend

c = mydb.cursor();

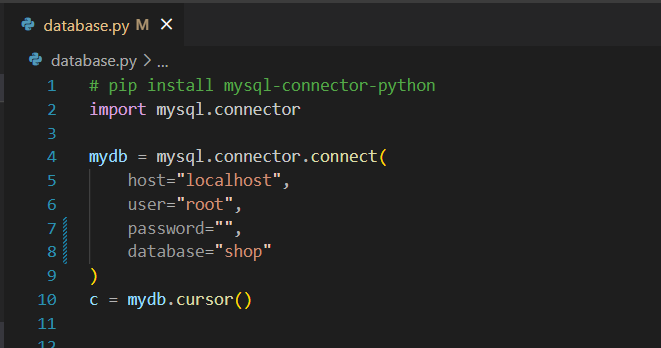


Fig18 - Cursor

**Frontend**

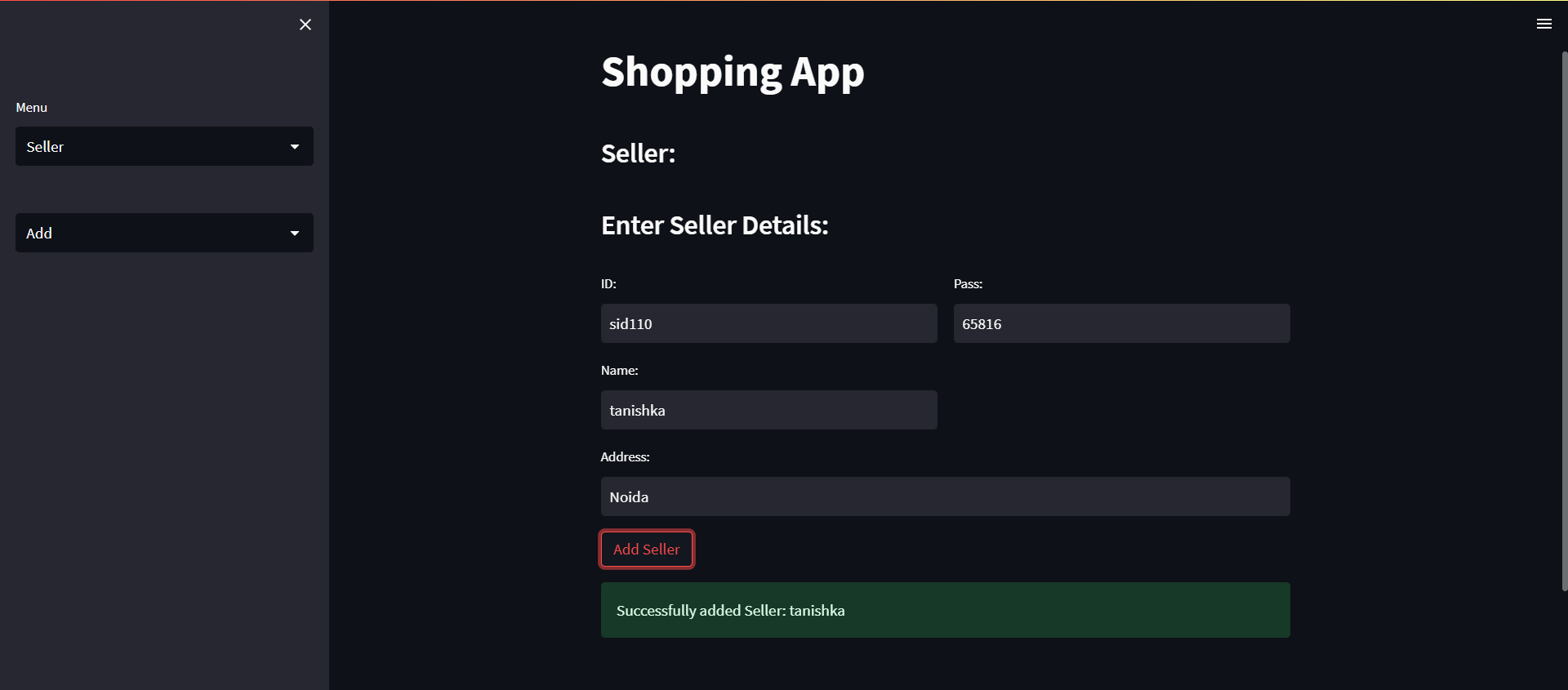
****

Fig19 - Add

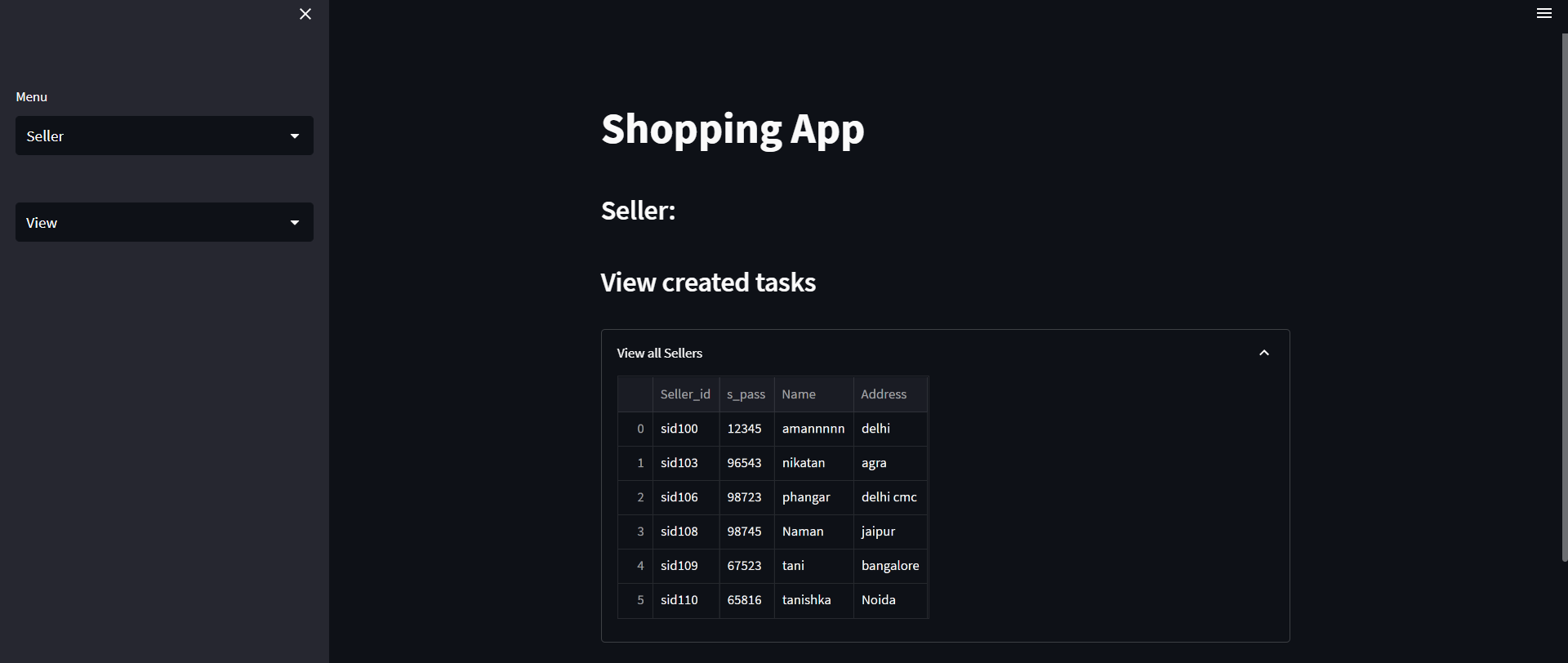


Fig20 - View

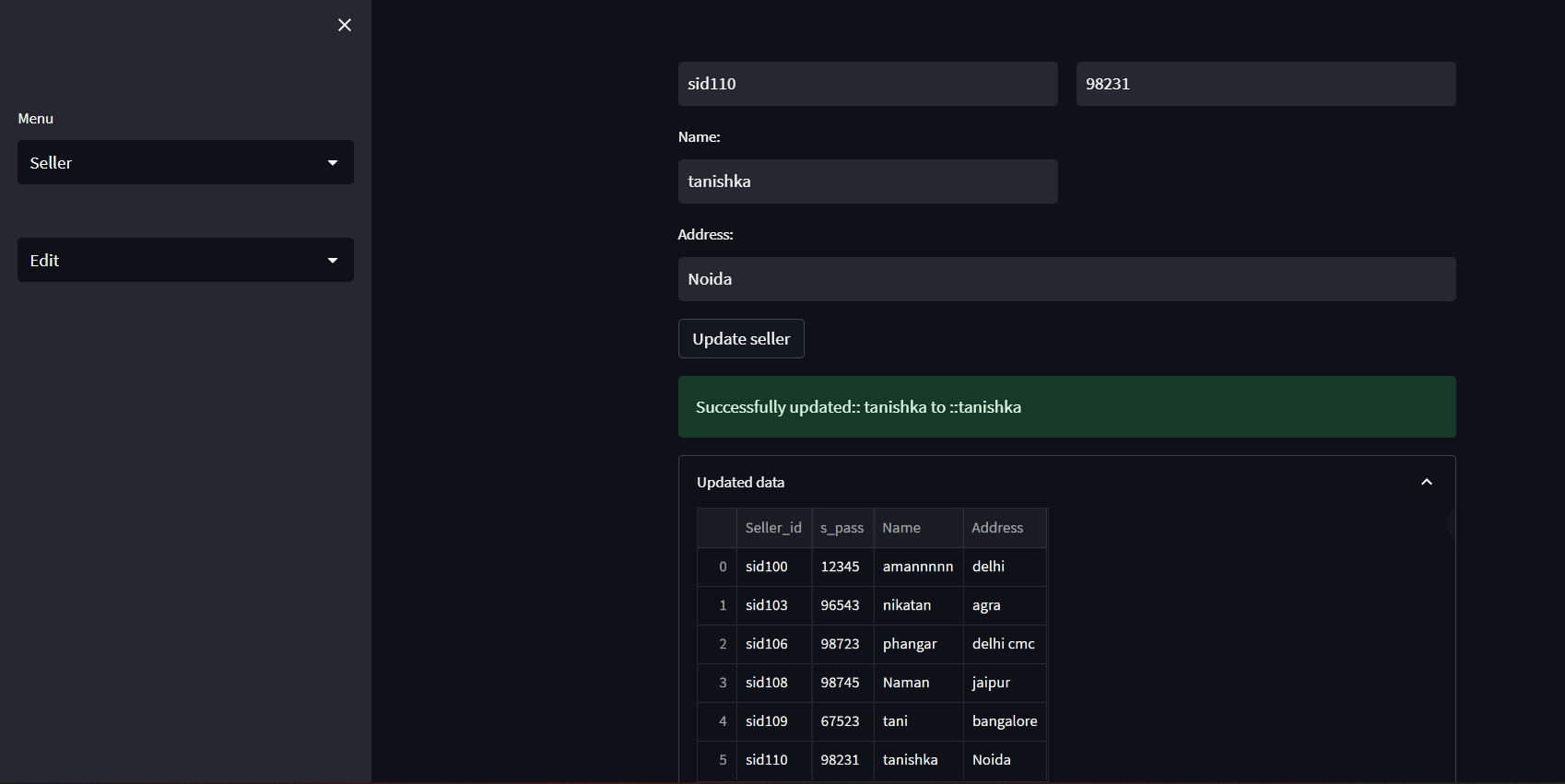


Fig21 - Edit

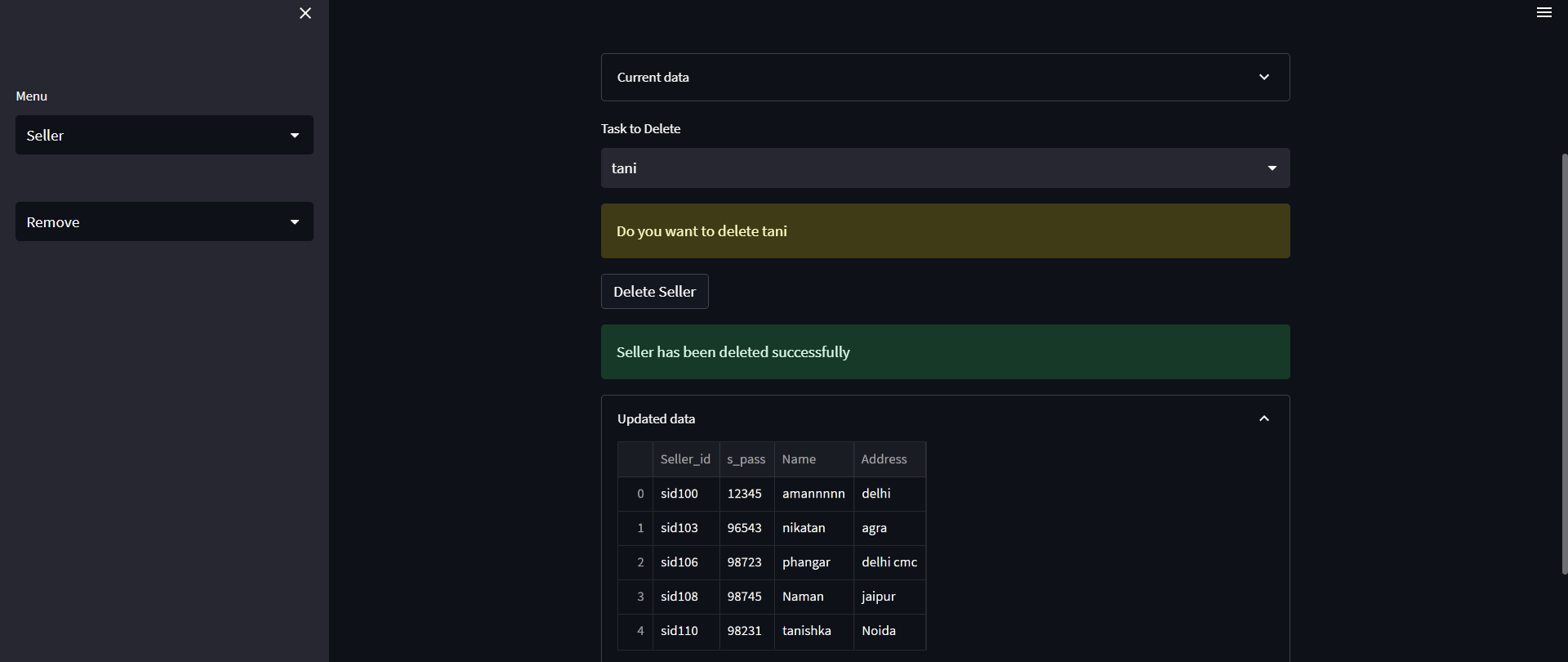


Fig 22 – Delete

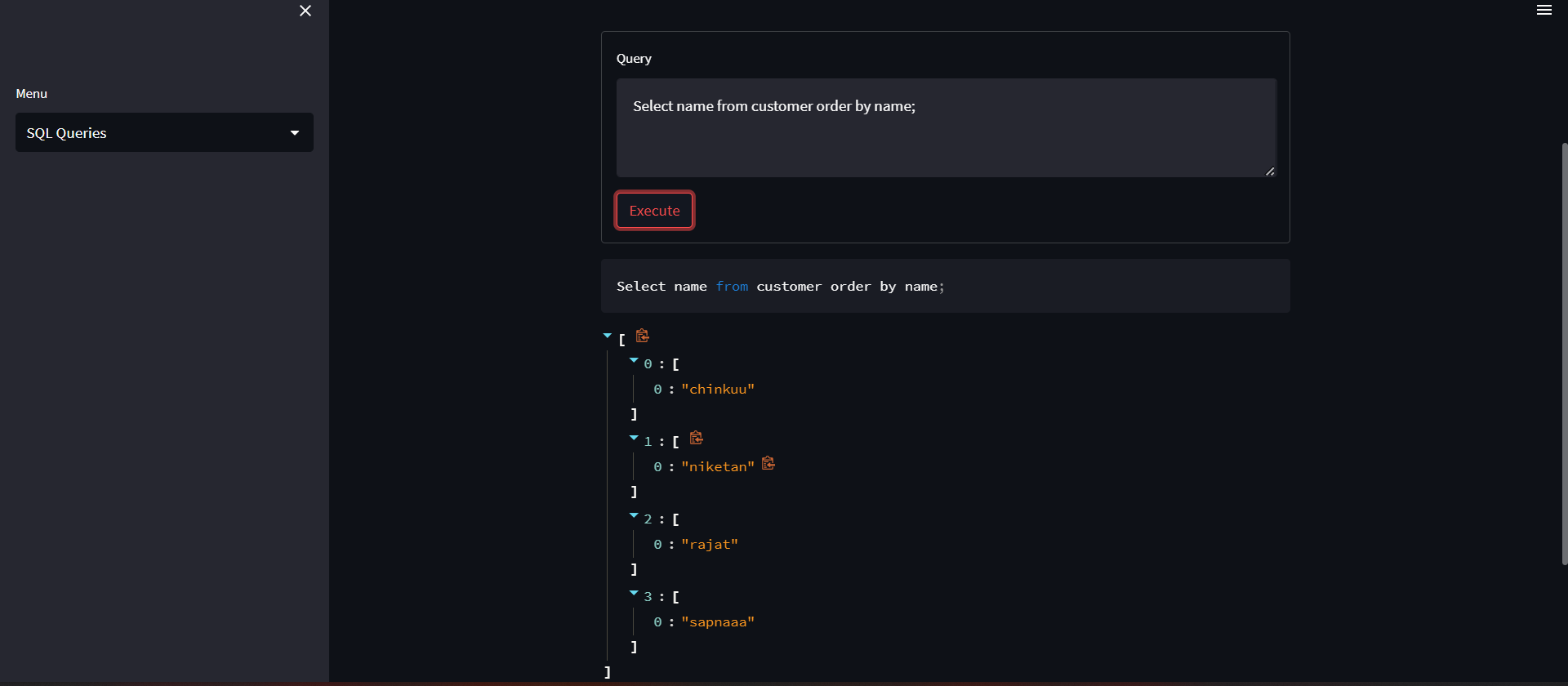


Fig23 – SQL Query