DBMS Mini Project

E-Commerce
Database Management System

Submitted by :

Abhishek Patil

PES1UG20CS010

V Semester

Section A

Short Description and Scope of the Project

This project simulates an e-commerce database. Customers and suppliers can register by adding their details. Products are classified into categories. Suppliers can add products to multiple categories and Customers can order multiple products from different categories. Customers can also add multiple delivery address and choose one of them for each order that they make. Customers can also see their order history with price and count of each product from that order. Analytics is done with the help of various operators and functions and procedures. An UI is also developed as a part of this project which performs basic CRUD operations as required and also has an option to run queries on the database.

RDBMS – MySQL

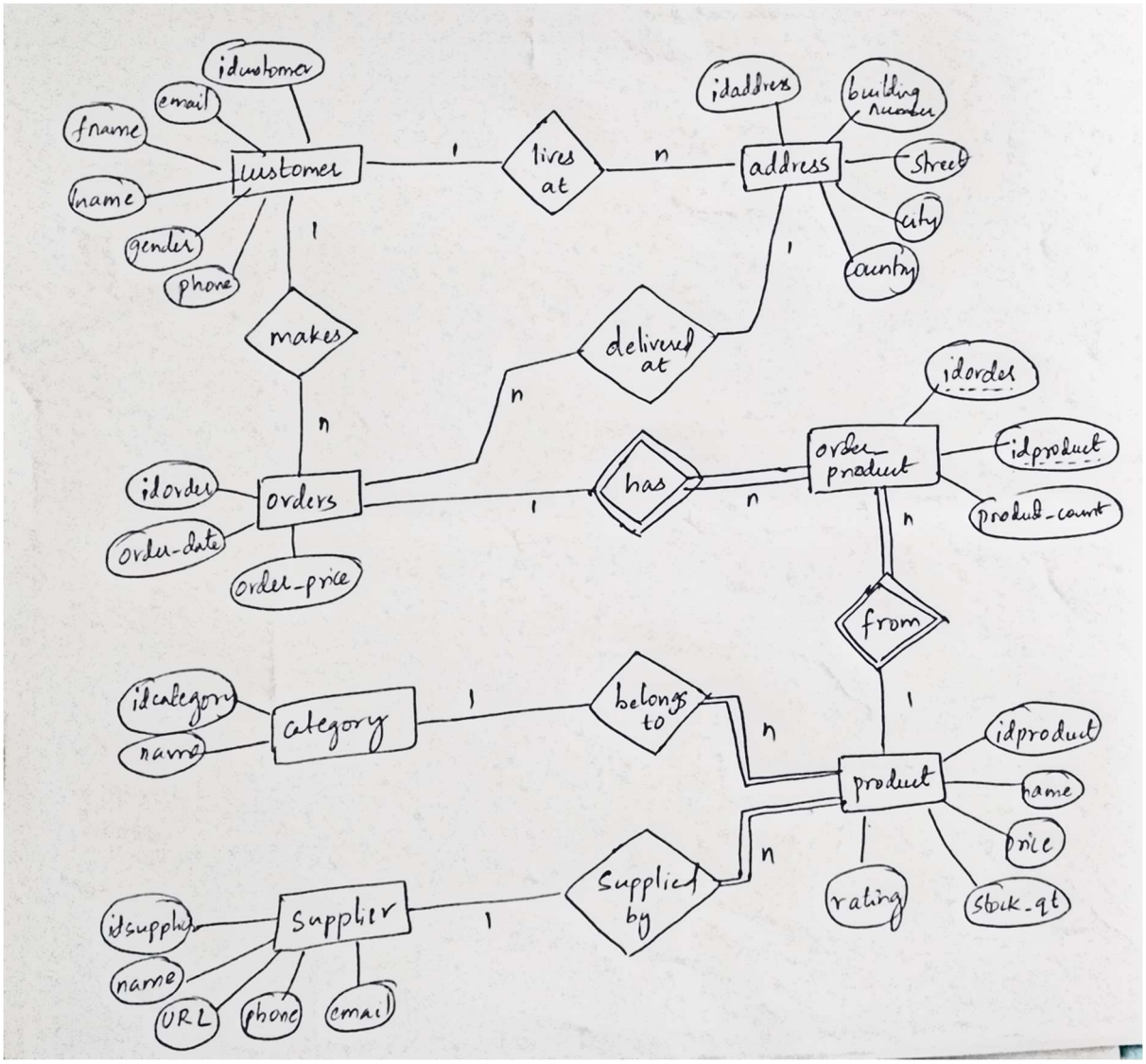
Query language - SQL

SQL editor – phpMyAdmin

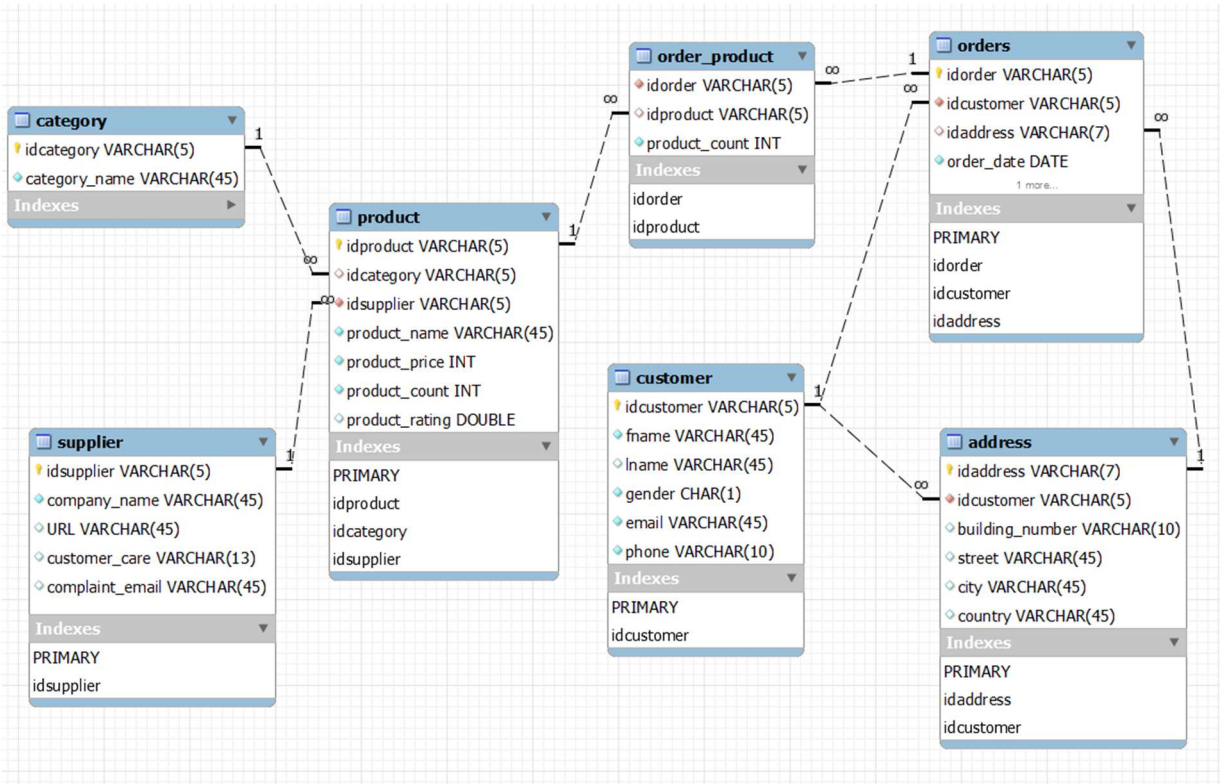
Relational Schema – MySQL Workbench

Frontend – python and Streamlit

ER diagram


idcustomer
email
fname
customer
lname
gender
phone
lives at
idaddress
building number
address
street
city
country
makes
delivered at
idorder
order product
idproduct
product_count
has
orders
idorder
order_date
order_price
from
idcategory
name
category
belongs to
idproduct
name
price
product
rating
stock_qt
Supplied by
idsupplier
name
URL
phone
email
Supplier

Relational Schema



DDL statements - Building the database

Run SQL query/queries on database e-commerce_database_010:

Console

```sql
/*customer*/
CREATE TABLE `e-commerce_database_010`.`customer` (
  `idcustomer` varchar(5) NOT NULL UNIQUE,
  `fname` varchar(45) NOT NULL,
  `lname` varchar(45) DEFAULT NULL,
  `gender` char(1) NOT NULL,
  `email` varchar(45) NOT NULL,
  `phone` varchar(10) NOT NULL,
  PRIMARY KEY (`idcustomer`)
);

/*supplier*/
CREATE TABLE `e-commerce_database_010`.`supplier` (
  `idsupplier` VARCHAR(5) NOT NULL UNIQUE,
  `company_name` VARCHAR(45) NOT NULL,
  `URL` VARCHAR(45) DEFAULT NULL,
  `customer_care` VARCHAR(13) DEFAULT NULL,
  `complaint_email` VARCHAR(45) DEFAULT NULL,
  PRIMARY KEY (`idsupplier`)
);

/*category*/
CREATE TABLE `e-commerce_database_010`.`category` (
  `idcategory` VARCHAR(5) NOT NULL UNIQUE,
  `category_name` VARCHAR(45) NOT NULL,
  `product_count` INT NULL DEFAULT 0,
  PRIMARY KEY (`idcategory`)
);

/*address*/
CREATE TABLE `e-commerce_database_010`.`address` (
  `idaddress` VARCHAR(7) NOT NULL UNIQUE,
  `idcustomer` VARCHAR(5) NOT NULL,
  `building_number` VARCHAR(10) DEFAULT NULL,
  `street` VARCHAR(45) DEFAULT NULL,
  `city` VARCHAR(45) DEFAULT NULL,
  `country` VARCHAR(45) DEFAULT "India",
  PRIMARY KEY (`idaddress`),
  FOREIGN KEY (`idcustomer`) REFERENCES `customer` (`idcustomer`)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

```sql

/*product*/
CREATE TABLE `e-commerce_database_010`.`product` (
  `idproduct` VARCHAR(5) NOT NULL UNIQUE,
  `idcategory` VARCHAR(5),
  `idsupplier` VARCHAR(5) NOT NULL,
  `product_name` VARCHAR(45) NOT NULL,
  `product_price` INT NOT NULL,
  `product_count` INT NOT NULL,
  `product_rating` DOUBLE DEFAULT 0,
  PRIMARY KEY (`idproduct`),
  FOREIGN KEY (`idcategory`) REFERENCES `category` (`idcategory`)
  ON DELETE SET NULL ON UPDATE SET NULL,
  FOREIGN KEY (`idsupplier`) REFERENCES `supplier` (`idsupplier`)
  ON DELETE CASCADE ON UPDATE CASCADE
);

/*order*/
CREATE TABLE `e-commerce_database_010`.`order` (
  `idorder` VARCHAR(5) NOT NULL UNIQUE,
  `idcustomer` VARCHAR(5) NOT NULL,
  `idaddress` VARCHAR(7),
  `order_date` DATE NOT NULL,
  `order_price` INT NOT NULL,
  PRIMARY KEY (`idorder`),
  FOREIGN KEY (`idcustomer`) REFERENCES `customer` (`idcustomer`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`idaddress`) REFERENCES `address` (`idaddress`)
  ON DELETE SET NULL ON UPDATE SET NULL
);

/*order_product*/
CREATE TABLE `e-commerce_database_010`.`order_product` (
  `idorder` VARCHAR(5) NOT NULL,
  `idproduct` VARCHAR(5),
  `product_count` INT NOT NULL,
  FOREIGN KEY (`idorder`) REFERENCES `order` (`idorder`)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (`idproduct`) REFERENCES `product` (`idproduct`)
  ON DELETE SET NULL ON UPDATE SET NULL
);
```

Server: localhost » Database: e-commerce_database_010

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | More

Show query box

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0251 seconds.)

```
/*customer*/ CREATE TABLE `e-commerce_database_010`.`customer` ( `idcustomer` varchar(5) NOT NULL UNIQUE, `fname` varchar(45) NOT NULL, `lname` varchar(45) DEFAULT NULL, `gender` char(1) NOT NULL, `email` varchar(45) NOT NULL, `phone` varchar(10) NOT NULL, PRIMARY KEY (`idcustomer`) );
```

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0315 seconds.)

```
/*supplier*/ CREATE TABLE `e-commerce_database_010`.`supplier` ( `idsupplier` VARCHAR(5) NOT NULL UNIQUE, `company_name` VARCHAR(45) NOT NULL, `URL` VARCHAR(45) DEFAULT NULL, `customer_care` VARCHAR(13) DEFAULT NULL, `complaint_email` VARCHAR(45) DEFAULT NULL, PRIMARY KEY (`idsupplier`) );
```

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0266 seconds.)

```
/*category*/ CREATE TABLE `e-commerce_database_010`.`category` ( `idcategory` VARCHAR(5) NOT NULL UNIQUE, `category_name` VARCHAR(45) NOT NULL, `product_count` INT NULL DEFAULT 0, PRIMARY KEY (`idcategory`) );
```

Console

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0307 seconds.)

```
/*address*/ CREATE TABLE `e-commerce_database_010`.`address` ( `idaddress` VARCHAR(7) NOT NULL UNIQUE, `idcustomer` VARCHAR(5) NOT NULL, `building_number` VARCHAR(10) DEFAULT NULL, `street` VARCHAR(45) DEFAULT NULL, `city` VARCHAR(45) DEFAULT NULL, `country` VARCHAR(45) DEFAULT "India", PRIMARY KEY (`idaddress`), FOREIGN KEY (`idcustomer`) REFERENCES `customer` (`idcustomer`) ON DELETE CASCADE ON UPDATE CASCADE );
```

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0344 seconds.)

```
/*product*/ CREATE TABLE `e-commerce_database_010`.`product` ( `idproduct` VARCHAR(5) NOT NULL UNIQUE, `idcategory` VARCHAR(5), `idsupplier` VARCHAR(5) NOT NULL, `product_name` VARCHAR(45) NOT NULL, `product_price` INT NOT NULL, `product_count` INT NOT NULL, `product_rating` DOUBLE DEFAULT 0, PRIMARY KEY (`idproduct`), FOREIGN KEY (`idcategory`) REFERENCES `category` (`idcategory`) ON DELETE SET NULL ON UPDATE SET NULL, FOREIGN KEY (`idsupplier`) REFERENCES `supplier` (`idsupplier`) ON DELETE CASCADE ON UPDATE CASCADE );
```

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0390 seconds.)

```
/*order*/ CREATE TABLE `e-commerce_database_010`.`order` ( `idorder` VARCHAR(5) NOT NULL UNIQUE, `idcustomer` VARCHAR(5) NOT NULL, `idaddress` VARCHAR(7), `order_date` DATE NOT NULL, `order_price` INT NOT NULL, PRIMARY KEY (`idorder`), FOREIGN KEY (`idcustomer`) REFERENCES `customer` (`idcustomer`) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (`idaddress`) REFERENCES `address` (`idaddress`) ON DELETE SET NULL ON UPDATE SET NULL );
```

[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0337 seconds.)

```
/*order_product*/ CREATE TABLE `e-commerce_database_010`.`order_product` ( `idorder` VARCHAR(5) NOT NULL, `idproduct` VARCHAR(5), `product_count` INT NOT NULL, FOREIGN KEY (`idorder`) REFERENCES `order` (`idorder`) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (`idproduct`) REFERENCES `product` (`idproduct`) ON DELETE SET NULL ON UPDATE SET NULL );
```

[Edit inline] [Edit] [Create PHP code

Populating the Database

a. Through INSERT command

Run SQL query/queries on database e-commerce_database_010:

```sql
/*insert customers*/
INSERT INTO customer VALUES
('C0001', 'Sherlock', 'Holmes', 'M', 'sherlock@gmail.com', '11111'),
('C0002', 'Sheldon', 'Cooper', 'M', 'sheldon@gmail.com', '22222'),
('C0003', 'Patrick', 'Jane', 'M', 'patrick@gmail.com', '33333'),
('C0004', 'Red', 'John', 'M', 'redjohn@gmail.com', '44444'),
('C0005', 'Rajesh', 'Koothrapalli', 'M', 'rajesh@gmail.com', '55555'),
('C0006', 'Teresa', 'Lisbon', 'F', 'lisbon@gmail.com', '66666'),
('C0007', 'Naruto', 'Uzumaki', 'M', 'narutouzumaki@gmail.com', '99999'),
('C0008', 'Kakashi', 'Hatake', 'M', 'kakashi00@gmail.com', '10101');
```

Console

```sql
/*insert suppliers*/
INSERT INTO supplier VALUES
('S0001', 'Lenovo', 'www.lenovo.com', '1800-123-4545', 'lenovo.care@gmail.com'),
('S0002', 'Dell', 'www.dell.com', '1800-123-4569', 'dell.care@gmail.com'),
('S0003', 'Adidas', 'www.adidas.com', '1800-123-4200', 'adidas.care@gmail.com'),
('S0004', 'Puma', 'www.puma.com', '1800-123-4848', 'puma.care@gmail.com'),
('S0005', 'Nike', 'www.nike.com', '1800-123-4343', 'nike.care@gmail.com'),
('S0006', 'Reebok', 'www.reebok.com', '1800-123-4646', 'reebok.care@gmail.com'),
('S0007', 'Apple', 'www.apple.com', '1800-123-4567', 'apple.care@gmail.com');

/*insert categories*/
INSERT INTO category VALUES
('C0001', 'Laptop'),
('C0002', 'Smartphone'),
('C0003', 'Shoes'),
('C0004', 'Clothing');

/*insert address*/
INSERT INTO address VALUES
('C0001-1', 'C0001', '#221B', 'Baker Street', 'London', 'UK'),
('C0002-1', 'C0002', '#215S', 'Madison Ave', 'Pasadena', 'USA'),
('C0003-1', 'C0003', '#1309', 'Cedars Street', 'Malibu', 'USA '),
('C0003-2', 'C0003', '#3600', 'Riverside Blvd', 'Sacramento', 'USA'),
('C0004-1', 'C0004', '#14', 'Street 1', 'Napa', 'USA'),
('C0005-1', 'C0005', '#15', 'Madison Ave', 'Pasadena', 'USA'),
('C0006-1', 'C0006', '#3600', 'Riverside Blvd', 'Sacramento', 'USA'),
('C0007-1', 'C0007', '#02', 'Hokage Residence', 'Konohagakure', 'Japan'),
('C0008-1', 'C0008', '#25', 'Hokage Residence', 'Konohagakure', 'Japan');
```

```sql
/*insert products*/
INSERT INTO product VALUES
('P0001', 'C0001', 'S0001', 'Lenovo IdeaPad 330', 35000, 5, 3.5),
('P0002', 'C0001', 'S0001', 'Lenovo ThinkPad E14', 75000, 10, 4.0),
('P0003', 'C0001', 'S0002', 'Dell Inspiron 15', 40000, 15, 3.8),
('P0004', 'C0001', 'S0002', 'Dell G15 Gaming', 90000, 10, 4.5),
('P0005', 'C0001', 'S0002', 'Dell Latitude 9430', 200000, 5, 4.2),
('P0006', 'C0002', 'S0007', 'iPhone 13', 70000, 25, 4.6),
('P0007', 'C0002', 'S0007', 'iPhone 14', 90000, 25, 4.4),
('P0008', 'C0001', 'S0007', 'MacBook Pro 2022', 150000, 10, 4.8),
('P0009', 'C0003', 'S0005', 'Nike Air Force 1 Mid QS', 13000, 3, 4.0),
('P0010', 'C0003', 'S0005', 'Nike Air Jordan 1 Retro ', 16000, 5, 4.0),
('P0011', 'C0004', 'S0005', 'Nike Pro Dri-FIT Tshirt', 2000, 15, 3.5),
('P0012', 'C0004', 'S0005', 'Nike Dri-FIT Tshirt', 2000, 15, 3.9),
('P0013', 'C0003', 'S0004', 'Puma Redon Move Shoes', 6000, 5, 4.2),
('P0014', 'C0003', 'S0004', 'Puma Electron E Pro Shoes', 4500, 12, 4.4),
('P0015', 'C0004', 'S0003', 'Adidas Gameday Hoodie', 4800, 13, 4.5),
('P0016', 'C0004', 'S0003', 'Adidas TraceRocker Jacket', 5000, 18, 4.6);

/*insert order*/
INSERT INTO orders VALUES
('O0001', 'C0002', 'C0002-1', '2022-08-11', 110000),
('O0002', 'C0005', 'C0005-1', '2022-07-12', 9800),
('O0003', 'C0002', 'C0002-1', '2020-05-14', 22000),
('O0004', 'C0003', 'C0003-1', '2021-06-15', 200000),
('O0005', 'C0003', 'C0003-2', '2022-07-16', 4000),
('O0006', 'C0005', 'C0005-1', '2020-08-17', 8000),
('O0007', 'C0004', 'C0004-1', '2020-04-09', 70000);

/*insert order details*/
INSERT INTO order_product VALUES
('O0001', 'P0001', 1),
('O0001', 'P0002', 1),
('O0002', 'P0015', 1),
('O0002', 'P0016', 1),
('O0003', 'P0010', 1),
('O0003', 'P0013', 1),
('O0004', 'P0005', 1),
('O0005', 'P0011', 2),
('O0006', 'P0012', 4),
('O0007', 'P0006', 1);

```

b. Through LOAD command

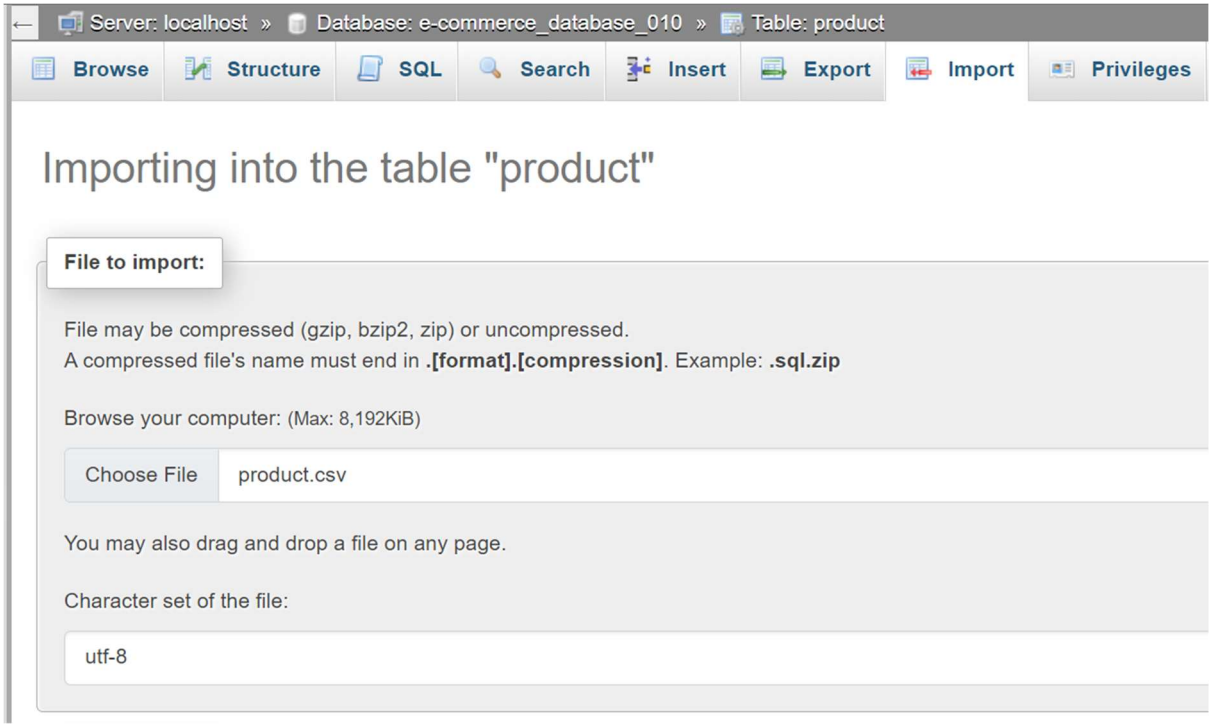
Run SQL query/queries on database e-commerce_database_010:

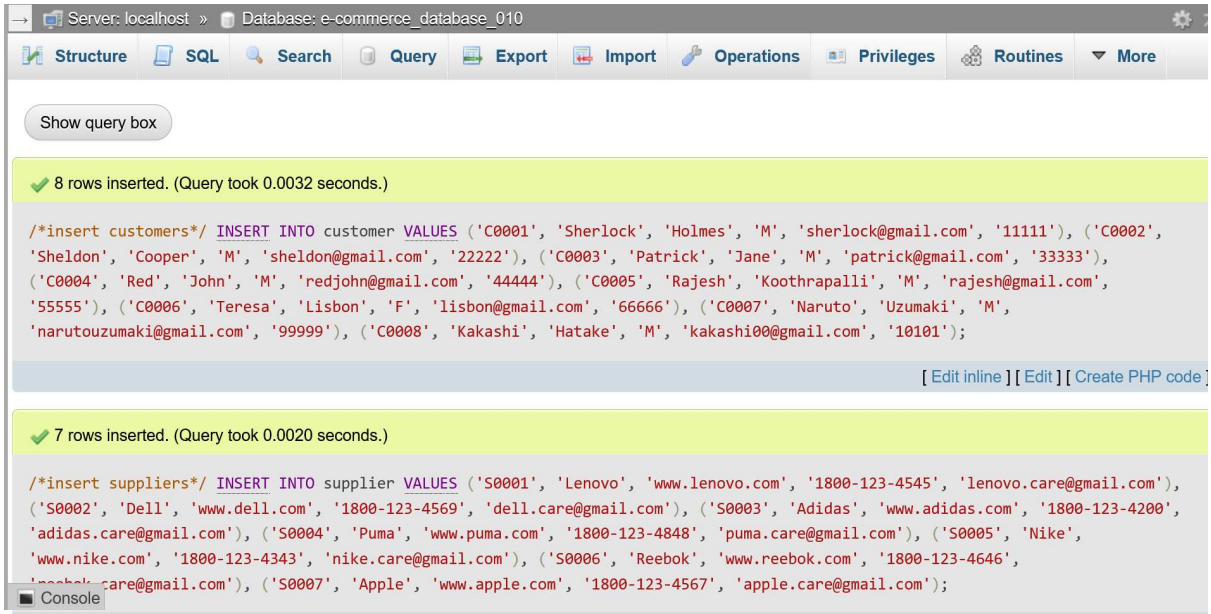
```
LOAD DATA INFILE 'C:\\Users\\Abhishek Patil\\Downloads\\orders.csv'
INTO TABLE orders
FIELDS ENCLOSED BY '"'
TERMINATED BY ';'
ESCAPED BY '"'
LINES TERMINATED BY '\n';

LOAD DATA INFILE 'C:\\Users\\Abhishek Patil\\Downloads\\order_product.csv'
INTO TABLE order_product
FIELDS ENCLOSED BY '"'
TERMINATED BY ';'
ESCAPED BY '"'
LINES TERMINATED BY '\n';

```

c. Through UI





4 rows inserted. (Query took 0.0025 seconds.)

```
/*insert categories*/ INSERT INTO category VALUES ('C0001', 'Laptop'), ('C0002', 'Smartphone'), ('C0003', 'Shoes'), ('C0004',
'Clothing');
```

[Edit inline] [Edit] [Create PHP code]

9 rows inserted. (Query took 0.0019 seconds.)

```
/*insert address*/ INSERT INTO address VALUES ('C0001-1', 'C0001', '#221B', 'Baker Street', 'London', 'UK'), ('C0002-1', 'C0002',
'#215S', 'Madison Ave', 'Pasadena', 'USA'), ('C0003-1', 'C0003', '#1309', 'Cedars Street', 'Malibu', 'USA '), ('C0003-2',
'C0003', '#3600', 'Riverside Blvd', 'Sacramento', 'USA'), ('C0004-1', 'C0004', '#14', 'Street 1', 'Napa', 'USA'), ('C0005-1',
'C0005', '#15', 'Madison Ave', 'Pasadena', 'USA'), ('C0006-1', 'C0006', '#3600', 'Riverside Blvd', 'Sacramento', 'USA'),
('C0007-1', 'C0007', '#02', 'Hokage Residence', 'Konohagakure', 'Japan'), ('C0008-1', 'C0008', '#25', 'Hokage Residence',
'Konohagakure', 'Japan');
```

[Edit inline] [Edit] [Create PHP code]

16 rows inserted. (Query took 0.0032 seconds.)

```
'C0001', 'S0001', 'Lenovo ThinkPad E14', 75000, 10, 4.0), ('P0003', 'C0001', 'S0002', 'Dell Inspiron 15', 40000, 15, 3.8),
('P0004', 'C0001', 'S0002', 'Dell G15 Gaming', 90000, 10, 4.5), ('P0005', 'C0001', 'S0002', 'Dell Latitude 9430', 200000, 5,
4.2), ('P0006', 'C0002', 'S0007', 'iPhone 13', 70000, 25, 4.6), ('P0007', 'C0002', 'S0007', 'iPhone 14', 90000, 25, 4.4),
('P0008', 'C0001', 'S0007', 'MacBook Pro 2022', 150000, 10, 4.8), ('P0009', 'C0003', 'S0005', 'Nike Air Force 1 Mid QS', 13000,
3, 4.0), ('P0010', 'C0003', 'S0005', 'Nike Air Jordan 1 Retro ', 16000, 5, 4.0), ('P0011', 'C0004', 'S0005', 'Nike Pro Dri-FIT
Tshirt', 2000, 15, 3.5), ('P0012', 'C0004', 'S0005', 'Nike Dri-FIT Tshirt', 2000, 15, 3.9), ('P0013', 'C0003', 'S0004', 'Puma
Redon Move Shoes', 6000, 5, 4.2), ('P0014', 'C0003', 'S0004', 'Puma Electron E Pro Shoes', 4500, 12, 4.4), ('P[...]
```

[Edit]

7 rows inserted. (Query took 0.0031 seconds.)

```
/*insert order*/ INSERT INTO orders VALUES ('O0001', 'C0002', 'C0002-1', '2022-08-11', 110000), ('O0002', 'C0005', 'C0005-1',
'2022-07-12', 9800), ('O0003', 'C0002', 'C0002-1', '2020-05-14', 22000), ('O0004', 'C0003', 'C0003-1', '2021-06-15', 200000),
('O0005', 'C0003', 'C0003-2', '2022-07-16', 4000), ('O0006', 'C0005', 'C0005-1', '2020-08-17', 8000), ('O0007', 'C0004',
'C0004-1', '2020-04-09', 70000);
```

[Edit inline] [Edit] [Create PHP code]

10 rows inserted. (Query took 0.0030 seconds.)

```
/*insert order details*/ INSERT INTO order_product VALUES ('O0001', 'P0001', 1), ('O0001', 'P0002', 1), ('O0002', 'P0015', 1),
('O0002', 'P0016', 1), ('O0003', 'P0010', 1), ('O0003', 'P0013', 1), ('O0004', 'P0005', 1), ('O0005', 'P0011', 2), ('O0006',
'P0012', 4), ('O0007', 'P0006', 1);
```

[Edit inline] [Edit] [Create PHP code]

JOIN Queries

a. Get a list of all customers and their address(es)

LEFT JOIN:

Run SQL query/queries on database e-commerce_database_010:

```
SELECT
    customer.fname,
    customer.lname,
    address.building_number,
    address.street,
    address.city,
    address.country
FROM
    customer
        LEFT JOIN
    address ON address.idcustomer = customer.idcustomer
```

Console

Showing rows 0 - 8 (9 total, Query took 0.0006 seconds.)

```
SELECT customer.fname, customer.lname, address.building_number, address.street, address.city,
address.country FROM customer LEFT JOIN address ON address.idcustomer = customer.idcustomer;
```

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

Show all | Number of rows: 25

Extra options

fname	lname	building_number	street	city	country
Sherlock	Holmes	#221B	Baker Street	London	UK
Sheldon	Cooper	#215S	Madison Ave	Pasadena	USA
Patrick	Jane	#1309	Cedars Street	Malibu	USA
Patrick	Jane	#3600	Riverside Blvd	Sacramento	USA
Red	John	#14	Street 1	Napa	USA
Rajesh	Koothrapalli	#15	Madison Ave	Pasadena	USA
Teresa	Lisbon	#3600	Riverside Blvd	Sacramento	USA
Naruto	Uzumaki	#02	Hokage Residence	Konohagakure	Japan
Kakashi	Hatake	#25	Hokage Residence	Konohagakure	Japan

b. GET a list of all laptops with their brand name, price, and rating.

RIGHT JOIN

Run SQL query/queries on database e-commerce_database_010:

```sql
SELECT
    supplier.company_name Company,
    laptops.product_name as model,
    laptops.product_price AS price,
    laptops.product_rating AS rating
FROM
    supplier
    RIGHT JOIN (
        SELECT
            category.category_name,
            product.idsupplier,
            product.product_name,
            product.product_price,
            product.product_rating
        FROM
            product
            RIGHT JOIN category ON category.idcategory = product.idcategory
        WHERE
            category.category_name = 'Laptop'
    ) AS laptops ON laptops.idsupplier = supplier.idsupplier
```

Console

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

```sql
SELECT supplier.company_name Company, laptops.product_name as model, laptops.product_price AS price, laptops.product_rating AS rating FROM supplier RIGHT JOIN ( SELECT category.category_name, product.idsupplier, product.product_name, product.product_price, product.product_rating FROM product RIGHT JOIN category ON category.idcategory = product.idcategory WHERE category.category_name = 'Laptop' ) AS laptops ON laptops.idsupplier = supplier.idsupplier;
```

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

Company	model	price	rating
Lenovo	Lenovo IdeaPad 330	35000	3.5
Lenovo	Lenovo ThinkPad E14	75000	4
Dell	Dell Inspiron 15	40000	3.8
Dell	Dell G15 Gaming	90000	4.5
Dell	Dell Latitude 9430	200000	4.2
Apple	MacBook Pro 2022	150000	4.8

c. GET a list of all products ordered by Patrick Jane

INNER JOIN

Run SQL query/queries on database e-commerce_database_010:

```
SELECT
    prod.order_date as `Order date`,
    prod.product_count as Quantity,
    product.product_name AS Product
from
    product
    INNER JOIN (
        SELECT
            order_product.idproduct,
            order_product.product_count,
            ord.order_date
        FROM
            order_product
            INNER JOIN (
                SELECT
                    orders.idorder,
                    orders.order_date
                FROM
                    orders
                    INNER JOIN customer ON customer.idcustomer = orders.idcustomer
                WHERE
                    customer.fname = 'Patrick'
                    AND customer.lname = 'Jane'
            ) AS ord ON ord.idorder = order_product.idorder
    ) AS prod ON prod.idproduct = product.idproduct
```

Console

Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

```
SELECT prod.order_date as `Order date`, prod.product_count as Quantity, product.product_name AS Product from product INNER JOIN ( SELECT order_product.idproduct, order_product.product_count, ord.order_date FROM order_product INNER JOIN ( SELECT orders.idorder, orders.order_date FROM orders INNER JOIN customer ON customer.idcustomer = orders.idcustomer WHERE customer.fname = 'Patrick' AND customer.lname = 'Jane' ) AS ord ON ord.idorder = order_product.idorder ) AS prod ON prod.idproduct = product.idproduct;
```

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

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

Extra options

Order date	Quantity	Product
2021-06-15	1	Dell Latitude 9430
2022-07-16	2	Nike Pro Dri-FIT Tshirt

Console

d. Get a list of all products with rating >= 4.5 with their supplier

JOIN

Run SQL query/queries on database e-commerce_database_010:

```
1  SELECT
2      product.product_name as Name,
3      product.product_price as Price,
4      product.product_rating as Rating,
5      supplier.company_name as Supplier
6  from
7      product
8      JOIN supplier ON supplier.idsupplier = product.idsupplier
9  WHERE
10     product.product_rating >= 4.5
```

Showing rows 0 - 4 (5 total, Query took 0.0008 seconds.)

```
SELECT product.product_name as Name, product.product_price as Price,
product.product_rating as Rating, supplier.company_name as Supplier from product
JOIN supplier ON supplier.idsupplier = product.idsupplier WHERE
product.product_rating >= 4.5;
```

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

Show all | Number of rows: All Filter rows: Search this table Sort by key:

Extra options

Name	Price	Rating	Supplier
Dell G15 Gaming	90000	4.5	Dell
iPhone 13	70000	4.6	Apple
MacBook Pro 2022	150000	4.8	Apple
Adidas Gameday Hoodie	4800	4.5	Adidas
Adidas TraceRocker Jacket	5000	4.6	Adidas

Aggregate Functions

a. Get total money spent by customers on the platform

SUM

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT
2     customer.fname,
3     customer.lname,
4     SUM(orders.order_price) as `Total Spent`
5 FROM
6     customer
7     INNER JOIN orders ON customer.idcustomer = orders.idcustomer
8 GROUP BY
9     orders.idcustomer
```

Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT customer.fname, customer.lname, SUM(orders.order_price) as `Total Spent`
FROM customer INNER JOIN orders ON customer.idcustomer = orders.idcustomer GROUP BY
orders.idcustomer;
```

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

Show all | Number of rows: All Filter rows: Search this table

Extra options

fname	lname	Total Spent
Sheldon	Cooper	132000
Rajesh	Koothrapalli	17800
Patrick	Jane	204000
Red	John	70000

b. GET Average price of all products in each category

AVG

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT
2     category.category_name,
3     AVG(product.product_price) as `Average Price`
4 from
5     product
6     JOIN category ON category.idcategory = product.idcategory
7 GROUP BY
8     product.idcategory
```

Showing rows 0 - 3 (4 total, Query took 0.0045 seconds.)

```
SELECT category.category_name, AVG(product.product_price) as `Average Price` from product JOIN category ON category.idcategory = product.idcategory GROUP BY product.idcategory;
```

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

Show all | Number of rows: All Filter rows: Search this table

Extra options

category_name	Average Price
Laptop	98333.3333
Smartphone	80000.0000
Shoes	9875.0000
Clothing	3450.0000

c. GET number of products supplied by each supplier

COUNT

Run SQL query/queries on database e-commerce_database_010:

```
1  SELECT
2      supplier.company_name as Supplier,
3      COUNT(product.idproduct) as `Number of products supplied`
4  from
5      product
6      JOIN supplier ON supplier.idsupplier = product.idsupplier
7  GROUP BY
8      supplier.idsupplier
9  ORDER BY
10     supplier.company_name
```

Showing rows 0 - 5 (6 total, Query took 0.0013 seconds.)

```
SELECT supplier.company_name as Supplier, COUNT(product.idproduct) as `Number of products supplied` from product JOIN supplier ON supplier.idsupplier = product.idsupplier GROUP BY supplier.idsupplier ORDER BY supplier.company_name;
```

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

Show all | Number of rows: All Filter rows: Search this table

Extra options

Supplier	Number of products supplied
Adidas	2
Apple	3
Dell	3
Lenovo	2
Nike	4
Puma	2

d. GET the highest priced product in each category

MAX

Run SQL query/queries on database e-commerce_database_010:

```
1  SELECT
2      product.product_name AS `Product Name`,
3      maxi.Category,
4      maxi.`Max Price`
5  FROM
6      product
7      INNER JOIN (
8          SELECT
9              category.category_name AS Category,
10             category.idcategory,
11             MAX(product.product_price) as `Max Price`
12         from
13             product
14             INNER JOIN category ON category.idcategory = product.idcategory
15         GROUP BY
16             product.idcategory
17         ORDER BY
18             category.category_name
19     ) AS maxi ON maxi.`Max Price` = product.product_price
20     AND maxi.idcategory = product.idcategory
```

Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT product.product_name AS `Product Name`, maxi.Category, maxi.`Max Price` FROM product
INNER JOIN ( SELECT category.category_name AS Category, category.idcategory,
MAX(product.product_price) as `Max Price` from product INNER JOIN category ON
category.idcategory = product.idcategory GROUP BY product.idcategory ORDER BY
category.category_name ) AS maxi ON maxi.`Max Price` = product.product_price AND
maxi.idcategory = product.idcategory;
```

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

Product Name	Category	Max Price
Dell Latitude 9430	Laptop	200000
iPhone 14	Smartphone	90000
Nike Air Jordan 1 Retro	Shoes	16000
Adidas TraceRocker Jacket	Clothing	5000

SET OPERATIONS

a. GET the list of users who haven't ordered products - EXCEPT

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT customer.idcustomer, customer.fname, customer.lname FROM customer NATURAL JOIN
2 ((SELECT customer.idcustomer FROM customer)
3 EXCEPT
4 (SELECT orders.idcustomer FROM orders)) AS not_ordered;
```

Showing rows 0 - 3 (4 total, Query took 0.0027 seconds.)

```
SELECT customer.idcustomer, customer.fname, customer.lname FROM customer NATURAL JOIN ((SELECT customer.idcustomer FROM customer) EXCEPT (SELECT
orders.idcustomer FROM orders)) AS not_ordered;
```

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

idcustomer	fname	lname
C0001	Sherlock	Holmes
C0006	Teresa	Lisbon
C0007	Naruto	Uzumaki
C0008	Kakashi	Hatake

b. Get the list of suppliers who supply both laptops and smartphones - INTERSECT

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT supplier.idsupplier, supplier.company_name, supplier.URL FROM supplier NATURAL JOIN
2 (
3     (SELECT product.idsupplier FROM product WHERE product.idcategory IN
4         (SELECT category.idcategory FROM category WHERE category.category_name = "Laptop"))
5 INTERSECT
6     (SELECT product.idsupplier FROM product WHERE product.idcategory IN
7         (SELECT category.idcategory FROM category WHERE category.category_name = "Smartphone"))
8 ) AS electronics;
```

Showing rows 0 - 0 (1 total, Query took 0.0015 seconds.)

```
SELECT supplier.idsupplier, supplier.company_name, supplier.URL FROM supplier NATURAL JOIN ( (SELECT product.idsupplier FROM product WHERE
product.idcategory IN (SELECT category.idcategory FROM category WHERE category.category_name = "Laptop")) INTERSECT (SELECT product.idsupplier
FROM product WHERE product.idcategory IN (SELECT category.idcategory FROM category WHERE category.category_name = "Smartphone")) ) AS
electronics;
```

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

idsupplier	company_name	URL
S0007	Apple	www.apple.com

c. GET the list of products that either weren't bought by customers or have a rating less than 4 – UNION

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT product.idproduct, product.product_name FROM product WHERE product.idproduct NOT IN
2     (SELECT order_product.idproduct FROM order_product)
3 UNION
4 SELECT product.idproduct, product.product_name FROM product WHERE product.product_rating < 4;
```

Showing rows 0 - 8 (9 total, Query took 0.0052 seconds.)

```
SELECT product.idproduct, product.product_name FROM product WHERE product.idproduct NOT IN (SELECT order_product.idproduct FROM order_product)
UNION SELECT product.idproduct, product.product_name FROM product WHERE product.product_rating < 4;
```

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

idproduct	product_name
P0003	Dell Inspiron 15
P0004	Dell G15 Gaming
P0007	iPhone 14
P0008	MacBook Pro 2022
P0009	Nike Air Force 1 Mid QS
P0014	Puma Electron E Pro Shoes
P0001	Lenovo IdeaPad 330
P0011	Nike Pro Dri-FIT Tshirt
P0012	Nike Dri-FIT Tshirt

d. Get the list of suppliers who are not supplying any products currently - EXCEPT

Run SQL query/queries on database e-commerce_database_010:

```
1 SELECT supplier.idsupplier, supplier.company_name, supplier.URL FROM supplier NATURAL JOIN
2 ((SELECT supplier.idsupplier FROM supplier)
3 EXCEPT
4 (SELECT product.idsupplier FROM product)) AS supplier;
```

Showing rows 0 - 0 (1 total, Query took 0.0050 seconds.)

```
SELECT supplier.idsupplier, supplier.company_name, supplier.URL FROM supplier NATURAL JOIN ((SELECT supplier.idsupplier
FROM supplier) EXCEPT (SELECT product.idsupplier FROM product)) AS supplier;
```

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

idsupplier	company_name	URL
S0006	Reebok	www.reebok.com

Functions and Procedures

Function

Objective: Supplier can input product ID and know the total sales of the product. This gives the supplier a good analysis of his product's performance on the e-commerce site.

Run SQL query/queries on database e-commerce_database_010:

```
DELIMITER $$
CREATE FUNCTION total_sales (ProductID VARCHAR(5))
RETURNS INT
READS SQL DATA
DETERMINISTIC
BEGIN
    DECLARE sale_amount INT;
    DECLARE total_sold INT;
    DECLARE price INT;
    SET total_sold  = (SELECT SUM(product_count) FROM order_product
                       WHERE idproduct = ProductID);
    SET price = (SELECT product_price FROM product
                 WHERE idproduct = ProductID);
    SET sale_amount = total_sold * price;
    RETURN sale_amount;
END; $$
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0127 seconds.)

```
CREATE FUNCTION total_sales (ProductID VARCHAR(5)) RETURNS INT READS SQL DATA DETERMINISTIC BEGIN DECLARE sale_amount INT;
DECLARE total_sold INT; DECLARE price INT; SET total_sold = (SELECT SUM(product_count) FROM order_product WHERE idproduct =
ProductID); SET price = (SELECT product_price FROM product WHERE idproduct = ProductID); SET sale_amount = total_sold *
price; RETURN sale_amount; END;;
```

[Edit inline] [Edit] [Create PHP code]

Sales of Nike – S0005 *(NULL indicates quantity_sold = 0)*

```
SELECT product_name, total_sales(idproduct) as `Total Sales`
FROM product WHERE product.idsupplier = 'S0005';
```

	product_name	Total Sales
Edit Copy Delete	Nike Air Force 1 Mid QS	*NULL*
Edit Copy Delete	Nike Air Jordan 1 Retro	16000
Edit Copy Delete	Nike Pro Dri-FIT Tshirt	4000
Edit Copy Delete	Nike Dri-FIT Tshirt	8000

Procedure

Objective: Customer can input Category name and get products with >4.5 rating from the catgeory along with their prices. Helps the customer in keeping tabs on good products on the ecommerce site.

```
DELIMITER $$
CREATE PROCEDURE display_good_products(
    IN categoryname VARCHAR(45),
    IN rating DOUBLE
)
BEGIN
    DECLARE categoryID VARCHAR(5);
    SET categoryID = (SELECT idcategory FROM category
                      WHERE category_name = categoryName);
    SELECT product_name, product_price, product_rating FROM product
    WHERE idcategory = categoryID AND product_rating >= rating;
END; $$
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0052 seconds.)

```
CREATE PROCEDURE display_good_products( IN categoryname VARCHAR(45), IN rating DOUBLE ) BEGIN DECLARE categoryID VARCHAR(5); SET categoryID =
(SELECT idcategory FROM category WHERE category_name = categoryName); SELECT product_name, product_price, product_rating FROM product WHERE
idcategory = categoryID AND product_rating >= rating; END;;
```

[Edit inline] [Edit] [Create PHP code]

Customer wants to see laptops with rating>=4.0

```
CALL display_good_products("Laptop", 4.0);
```

Showing rows 0 - 3 (4 total, Query took 0.0027 seconds.)

```
CALL display_good_products("Laptop", 4.0);
```

[Edit inline] [Edit] [Create PHP code]

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

Extra options

product_name	product_price	product_rating
Lenovo ThinkPad E14	75000	4
Dell G15 Gaming	90000	4.5
Dell Latitude 9430	200000	4.2
MacBook Pro 2022	150000	4.8

TRIGGER

Objective: A new product which hasn't had any sale yet cannot have >0 rating. Hence, this trigger shows an error when supplier tries to add product with >0 rating.

```
1  DELIMITER $$
2  CREATE TRIGGER update_product_rating
3  BEFORE INSERT
4  ON `product` FOR EACH ROW
5  BEGIN
6      DECLARE error_msg VARCHAR(100);
7      SET error_msg = ('New product must have 0 rating');
8      IF NEW.product_rating > 0 THEN
9          SIGNAL SQLSTATE '45000'
10         SET MESSAGE_TEXT = error_msg;
11     END IF;
12 END $$
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0137 seconds.)

```
CREATE TRIGGER update_product_rating BEFORE INSERT ON `product` FOR EACH ROW BEGIN DECLARE error_msg VARCHAR(100); SET error_msg = ('New product must have 0 rating'); IF NEW.product_rating > 0 THEN SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = error_msg; END IF; END;
```

[Edit inline] [Edit] [Create PHP code]

Reebok Tries to add shoes with rating 4.0

```
1 INSERT INTO product VALUES('P0017', 'C0003', 'S0006', 'Reebok Shoes', 4000, 10, 4.0);
```

Error

SQL query: Copy

```
INSERT INTO product VALUES('P0017', 'C0003', 'S0006', 'Reebok Shoes', 4000, 10, 4.0);
```

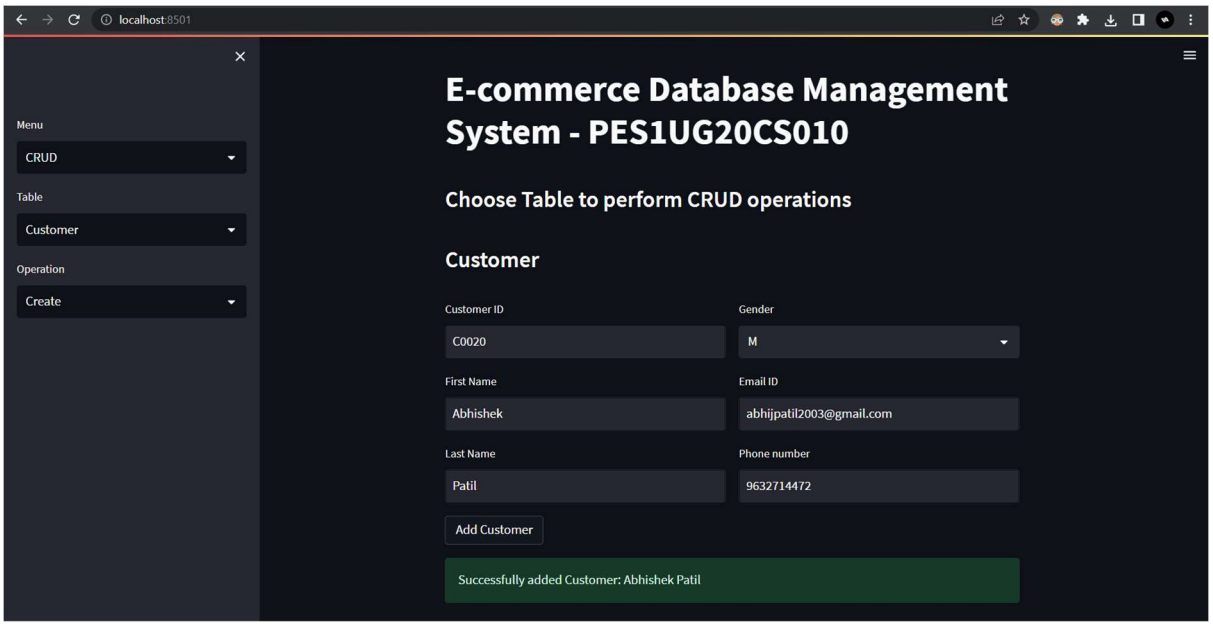
MySQL said:

```
#1644 - New product must have 0 rating
```

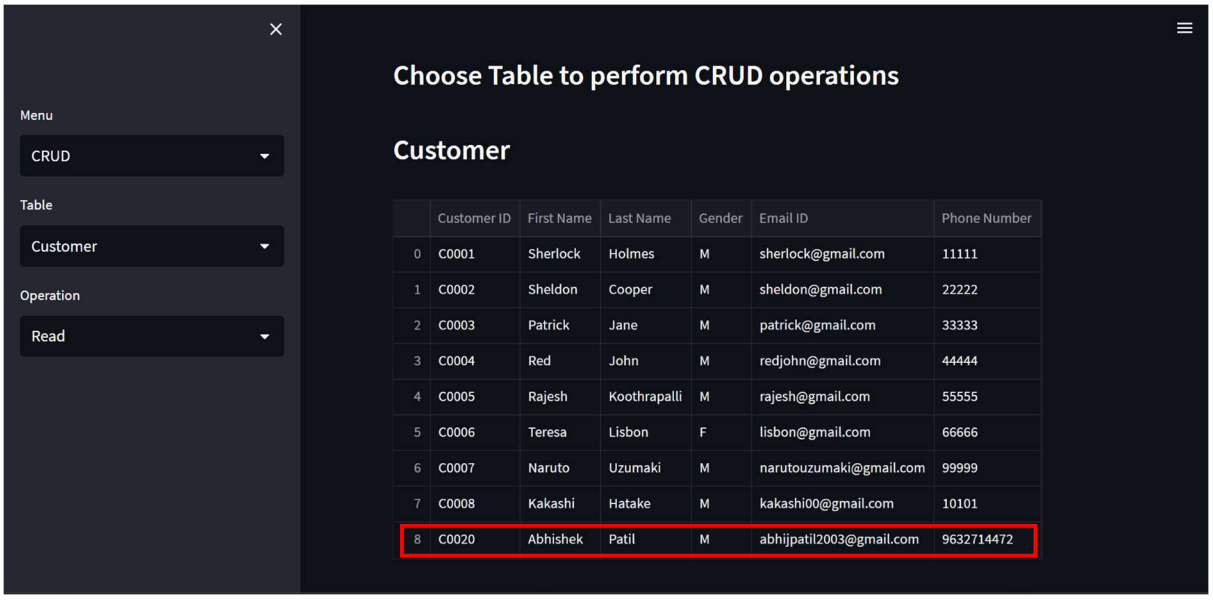
FRONTEND

CRUD operations :

Create/Add data



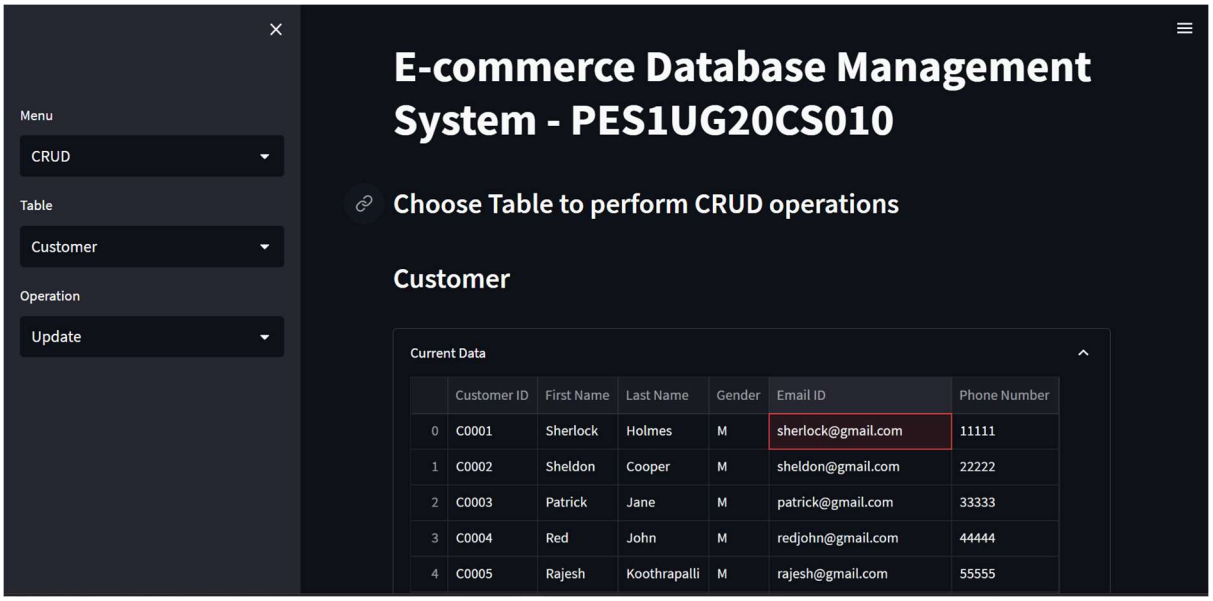
Read data

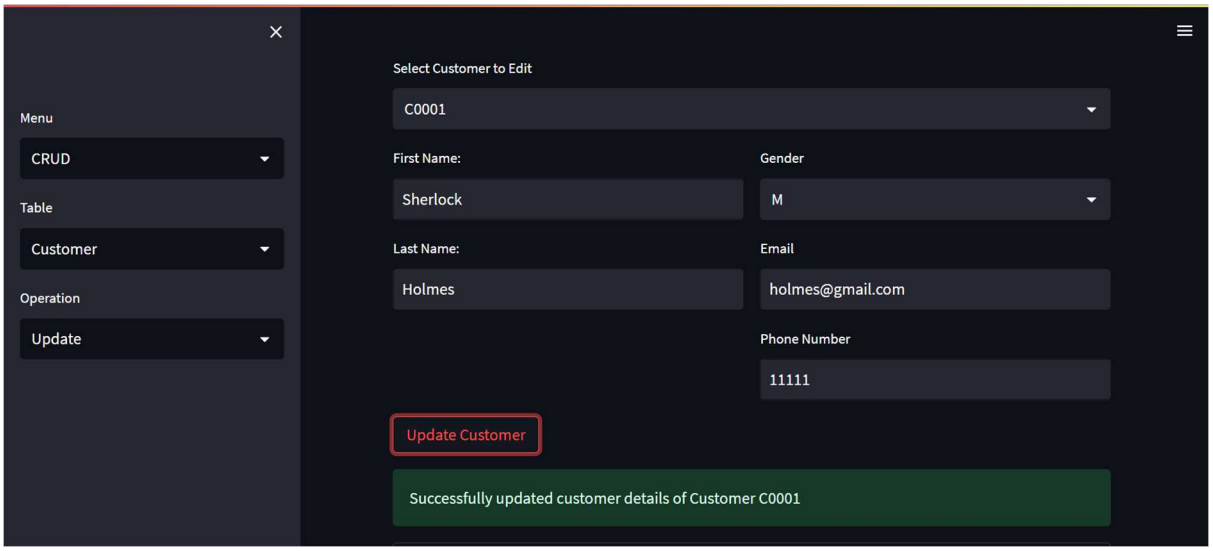


The highlighted record is the one which was added through the frontend.

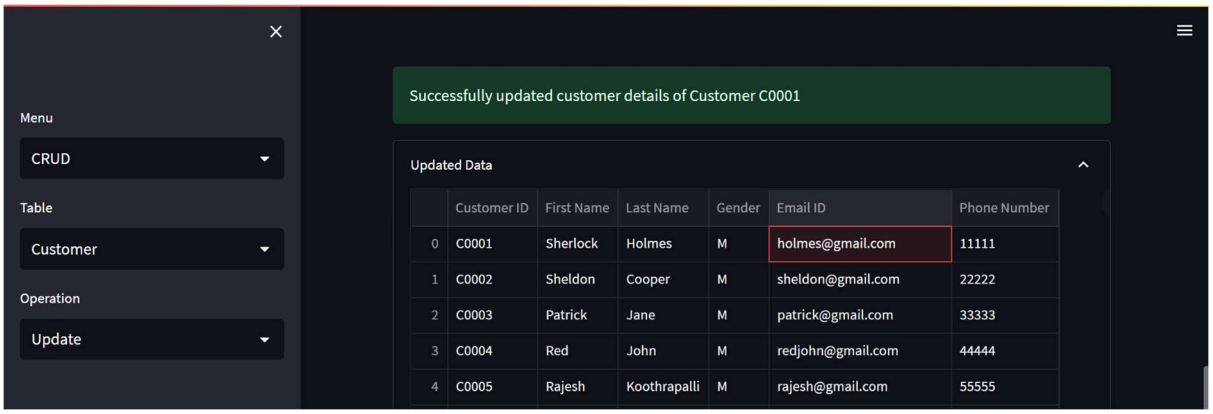
Update data

Before updating email of Sherlock Holmes



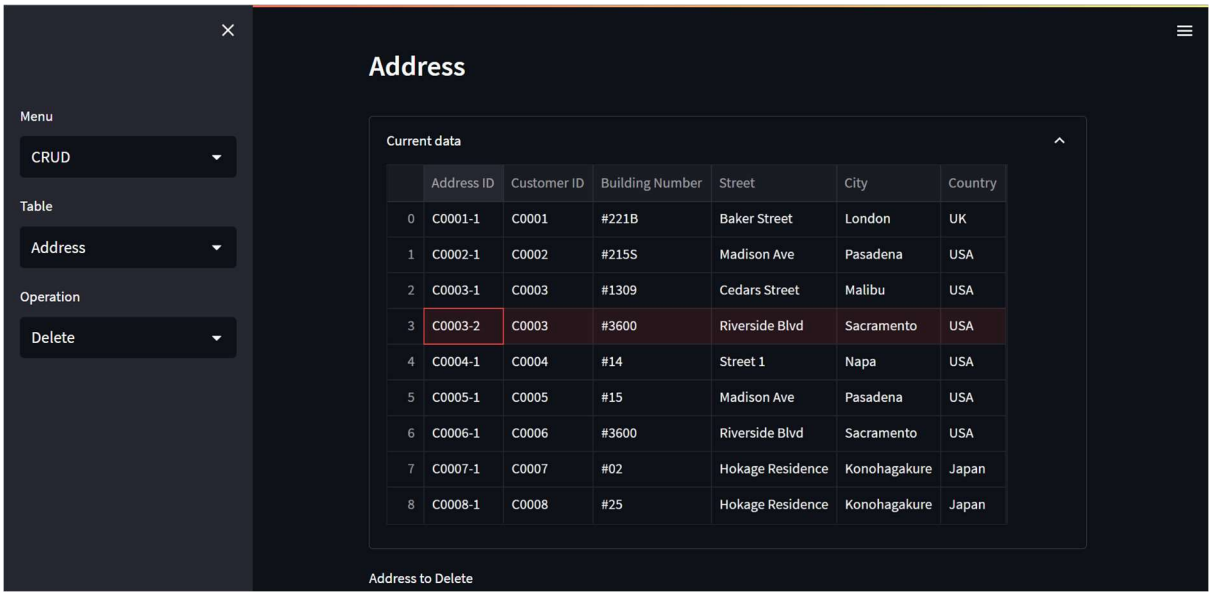


After updating email of Sherlock Holmes



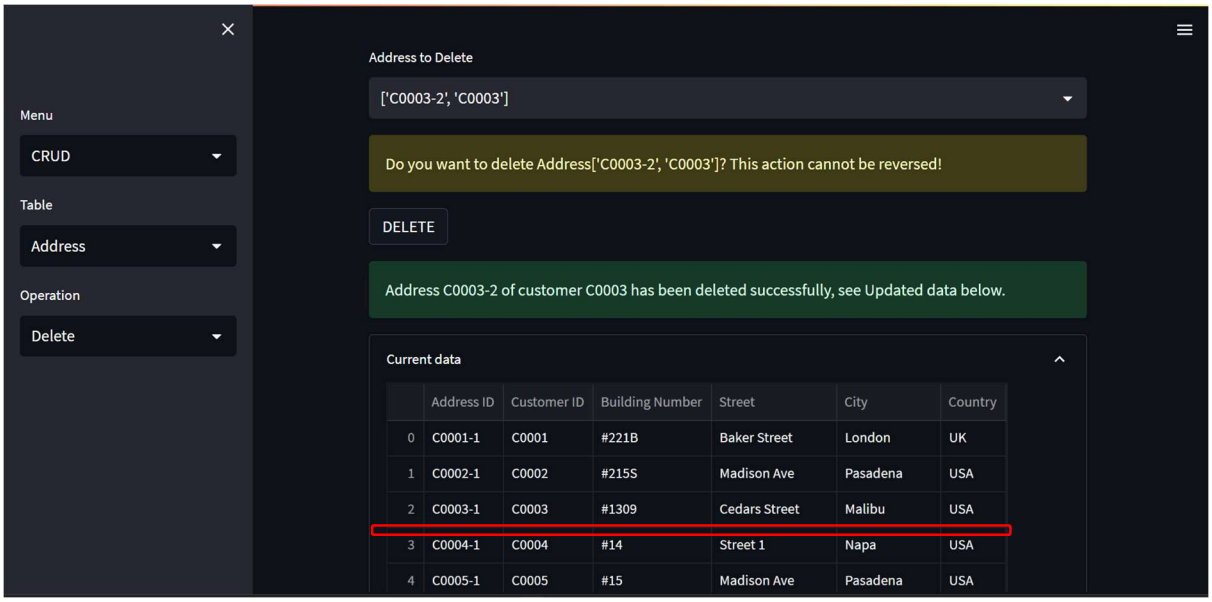
Delete data

Before deleting second address of customer C0003



	Address ID	Customer ID	Building Number	Street	City	Country
0	C0001-1	C0001	#221B	Baker Street	London	UK
1	C0002-1	C0002	#215S	Madison Ave	Pasadena	USA
2	C0003-1	C0003	#1309	Cedars Street	Malibu	USA
3	C0003-2	C0003	#3600	Riverside Blvd	Sacramento	USA
4	C0004-1	C0004	#14	Street 1	Napa	USA
5	C0005-1	C0005	#15	Madison Ave	Pasadena	USA
6	C0006-1	C0006	#3600	Riverside Blvd	Sacramento	USA
7	C0007-1	C0007	#02	Hokage Residence	Konohagakure	Japan
8	C0008-1	C0008	#25	Hokage Residence	Konohagakure	Japan

After deleting second address – C0003-2, only address C0003-1 remains for customer C0003



	Address ID	Customer ID	Building Number	Street	City	Country
0	C0001-1	C0001	#221B	Baker Street	London	UK
1	C0002-1	C0002	#215S	Madison Ave	Pasadena	USA
2	C0003-1	C0003	#1309	Cedars Street	Malibu	USA
3	C0004-1	C0004	#14	Street 1	Napa	USA
4	C0005-1	C0005	#15	Madison Ave	Pasadena	USA

Query Box

Calling function total_sales