JSS
SCIENCE AND
TECHNOLOGY
UNIVERSITY
MYSURU

REPORT

SUPERMARKET DATABASE

Team 5, C Section, 5th Sem, CSE

Anagha P (01JST18CS011)
Fiza N Shaikh (01JST18CS035)
Rachana P Bennur (01JST18CS097)
Samruddhi C Shetty (01JST18CS187)

TABLE OF CONTENTS:

1. Problem Statement
2. Overview of Project
3. ER Diagram and Schema
4. Tools and Languages used
5. SQL Commands
6. Front-end Design
7. Conclusion

PROBLEM STATEMENT:

To create an e-commerce supermarket database and develop a simple application of the same.

OVERVIEW OF THE PROJECT:

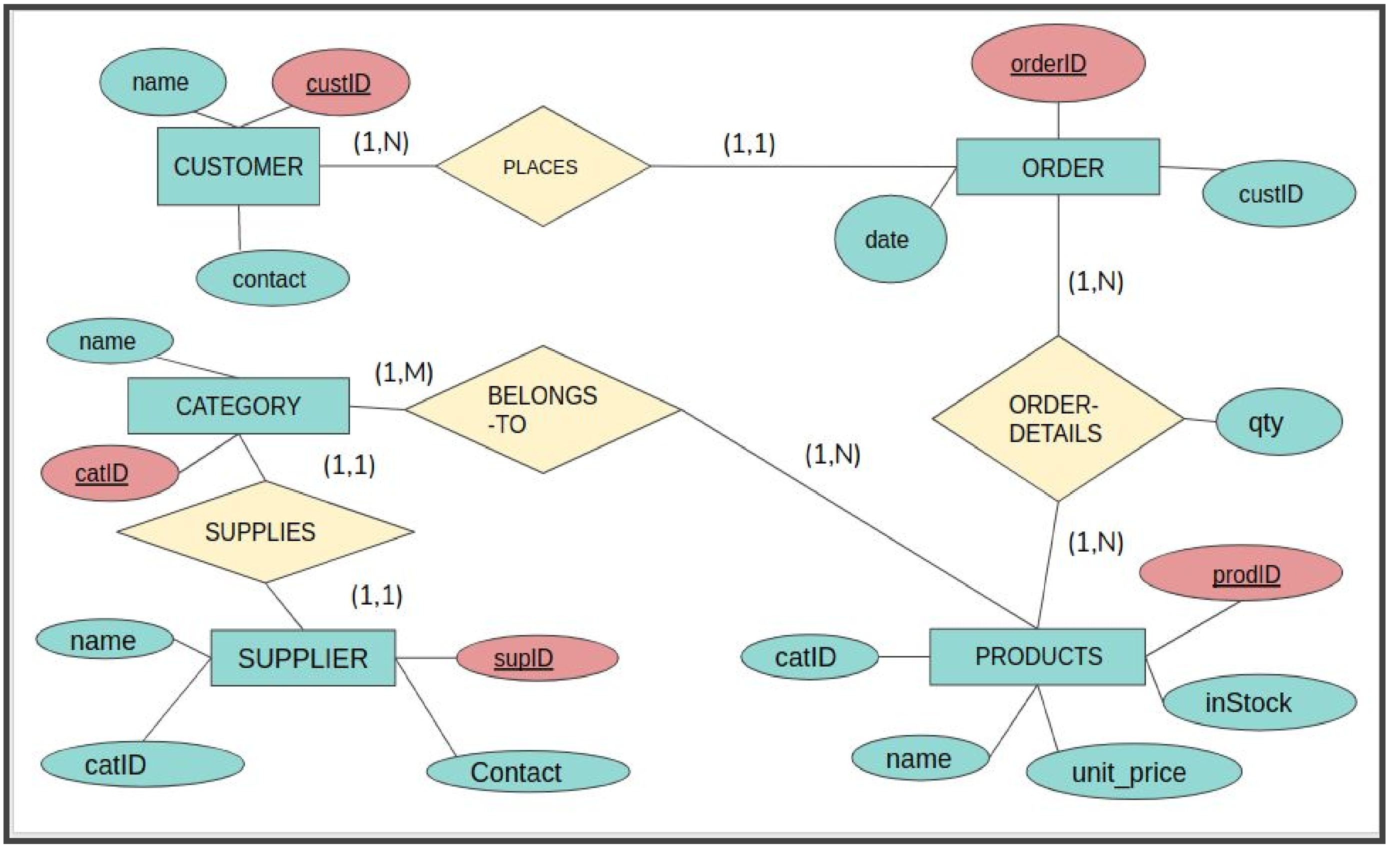
The objective of this project is to demonstrate the supermarket database and create a simple application to illustrate how e-commerce websites and applications work. It also makes the supermarket management system reliable, and more efficient. The manual system of storing and accessing records has following disadvantages:

1. It is more error-prone
2. The processing of data is less efficient
3. It consumes a lot of physical space of storage and time

To overcome these limitations, we have created a "SQLite database" that stores information about various components of the supermarket such as customer data, product data, order details and so on. The database stored is then accessed using a simple front-end application, that allows customers to order items online. To achieve the goals of our project, we have used a number of different tools.

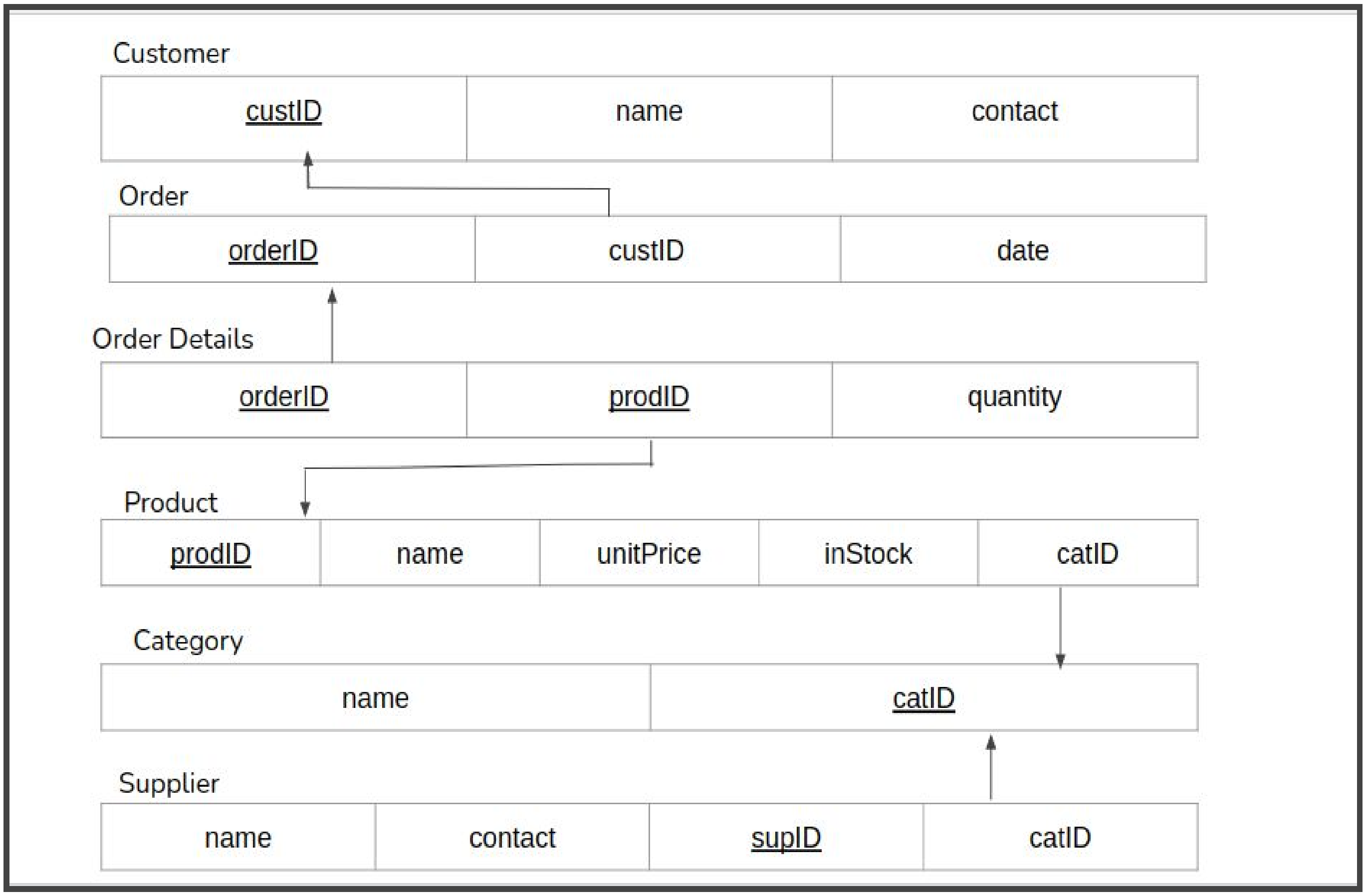
ER DIAGRAM AND SCHEMA

The ER diagram constructed during the designing process of the database has been depicted here.



SCHEMA

The schema diagram for the database design has been shown below.



Customer

custID	name	contact

Order

orderID	custID	date

Order Details

orderID	prodID	quantity

Product

prodID	name	unitPrice	inStock	catID

Category

name	catID

Supplier

name	contact	supID	catID

TOOLS AND LANGUAGES USED

The following tools and languages are used in the project:

1. DB Browser:
   DB is a light-weight, high quality, open source tool for creating, designing and editing database files that are compatible with SQLite. The reason for choosing DB Browser comes with the following benefits provided by the tool:

    1. The installation and running of the tool is simple and no pre-requisite knowledge for configuration of the tool is required.
    2. It allows users to write simple and short SQL commands to retrieve, store, and edit databases.
    3. SQLite is portable with all 32-bit and 64-bit OS. Also, it can be used with any programming language.

2. Django:

Django is a high level python, free and open source web framework that encourages rapid development along with clean and pragmatic design. It supports a number of database types such as MySQL, Oracle, SQLite, MariaDB and PostGreSQL. It has a collection of Python libraries that allows users to create web applications, both backend and frontend.

For our project, we have used scripting language, stylesheet and programming language as well.
Following are the languages used:

1. SQL - Structured Query Language or SQL is used to perform operations on databases we have created. It is a standard for ANSI and also there are various other versions of SQL. SQL can perform several operations such as

creating tables, retrieving data from database, storing in database, editing the database, deleting the database and so on.

2. HTML - HTML is used to build the layout of the web application for the supermarket database.
3. CSS - In conjunction to HTML, CSS is used to add styles and make the frontend more presentable.
4. Javascript - JS is used to add functionality to the static web application built with HTML and CSS.

SQL COMMANDS:

Building a database consists of several commands which can be categorised into 5 main categories. Among them, the prime commands are Data Definition Language(DDL), Data Manipulation Language(DML) and Data Query Language(DQL). The commands used in the project are as below:

DDL commands:

CUSTOMER :

```
CREATE TABLE "Customer" (
        "custID"      INTEGER NOT NULL,
        "name"        TEXT,
        "contact"     TEXT,
        PRIMARY KEY("custID")
);
```

ORDER :

```
CREATE TABLE "Order" (
        "orderID"     INTEGER NOT NULL,
        "custID"      INTEGER DEFAULT -1,
        "date"         DATE,
        PRIMARY KEY("orderID"),
        FOREIGN KEY("custID") REFERENCES "Customer"("custID") ON
UPDATE CASCADE ON DELETE SET DEFAULT
);
```

ORDER-DETAILS :

```
CREATE TABLE "Order-Details" (
        "custID"      INTEGER DEFAULT -1,
        "orderID"     INTEGER DEFAULT -1,
        "qty"  INTEGER,
        FOREIGN KEY("prodID") REFERENCES "Product"("prodID") ON
UPDATE CASCADE ON DELETE SET DEFAULT,
        FOREIGN KEY("custID") REFERENCES "Customer"("custID") ON
UPDATE CASCADE ON DELETE SET DEFAULT
);
```

PRODUCT :

```
CREATE TABLE "Product" (
        "prodID"      INTEGER,
        "name"        TEXT,
        "unitPrice"   INTEGER,
        "inStock"     INTEGER,
        "catID"       INTEGER NOT NULL DEFAULT -1,
        FOREIGN KEY("catID") REFERENCES "Category"("catID") ON
UPDATE CASCADE ON DELETE SET DEFAULT,
        PRIMARY KEY("prodID")
);
```

CATEGORY :

```
CREATE TABLE "Category" (
        "catID"       INTEGER NOT NULL,
        "name"        TEXT,
        PRIMARY KEY("catID")
);
```

SUPPLIER :

```
CREATE TABLE "Supplier" (
        "supID"       INTEGER NOT NULL,
        "catID"       INTEGER DEFAULT -1,
        "name"        TEXT,
        "contact"     STRING,
        FOREIGN KEY("catID") REFERENCES "Product"("prodID") ON
UPDATE CASCADE ON DELETE SET DEFAULT,
        PRIMARY KEY("supID")
);
```

DML Commands:

Category

```
INSERT INTO Category(catID, name) VALUES (101, "Stationery");
INSERT INTO Category(catID, name) VALUES (102, "Health and Wellness");
INSERT INTO Category(catID, name) VALUES (103, "Snacks");
INSERT INTO Category(catID, name) VALUES (104, "Dairy");
INSERT INTO Category(catID, name) VALUES (105, "Essentials");
INSERT INTO Category(catID, name) VALUES (106, "Fruits and Vegetables");
```

Customer

```
INSERT INTO Customer(custID,name,contact)
 VALUES (1,"Anagha","7204869010");

INSERT INTO Customer(custID,name,contact)
VALUES (2,"Rachana","9482153960");

INSERT INTO Customer(custID,name,contact)
VALUES (3,"Fiza","9380135810");

INSERT INTO Customer(custID,name,contact)
VALUES (4,"Samruddhi","9108241920");

INSERT INTO Customer(custID,name,contact)
VALUES (5,"Chris","0123456789");

INSERT INTO Customer(custID,name,contact)
VALUES (6,"Hemsworth","9876543210");

INSERT INTO Customer(custID,name,contact)
 VALUES (7,"Evans","1234509876");

INSERT INTO Customer(custID,name,contact)
VALUES (8,"Pratt","9886969696");

INSERT INTO Customer(custID,name,contact)
VALUES (9,"Steve","4206969696");

INSERT INTO Customer(custID,name,contact)
VALUES (10,"Bree","9612345678");
```

Supplier

```
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2000,"Unity
Traders","120120120",101);
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2001,"Birla
Suppliers","1112223331",102);
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2002,"Cooperative
Trading Company","5673451230",103);
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2003,"Gupta and
Son's Traders","9808765432",104);
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2004,"The
Provisional Traders Limited","1234523456",105);
INSERT INTO Supplier(supID,name,contact,catID) VALUES (2005,"The Twin's
Trading Society","980760540",106);
```

Order

```
INSERT INTO Order(orderID,custID,date) VALUES (1000,1,2020-12-08);
INSERT INTO Order(orderID,custID,date) VALUES (1001,2,2020-12-09);
INSERT INTO Order(orderID,custID,date) VALUES (1002,2,2020-12-09);
INSERT INTO Order(orderID,custID,date) VALUES (1003,3,2020-12-09);
INSERT INTO Order(orderID,custID,date) VALUES (1004,4,2020-12-10);
INSERT INTO Order(orderID,custID,date) VALUES (1005,5,2020-12-10);
INSERT INTO Order(orderID,custID,date) VALUES (1006,6,2020-12-10);
INSERT INTO Order(orderID,custID,date) VALUES (1007,7,2020-12-11);
INSERT INTO Order(orderID,custID,date) VALUES (1008,7,2020-12-11);
INSERT INTO Order(orderID,custID,date) VALUES (1009,8,2020-12-12);
```

Order-Details

```
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (201,2,1000);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (203,10,1000);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (206,1,1001);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (211,2,1002);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (207,1,1002);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (214,3,1002);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (209,2,1003);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (218,12,1003);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (204,5,1004);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (220,2,1004);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (216,2,1005);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (225,6,1005);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (202,7,1006);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (219,2,1006);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (205,1,1007);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (213,10,1007);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (216,2,1008);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (222,1,1008);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (211,5,1009);
INSERT INTO Order-Details(prodID,qty,orderID) VALUES (224,3,1009);
```

Product

```
INSERT INTO Product VALUES (223,"Apple",60,50,106);
INSERT INTO Product VALUES (222,"Banana",20,30,106);
INSERT INTO Product VALUES (211,"Bingo",20,140,103);
INSERT INTO Product VALUES (215,"Butter",20,50,104);
INSERT INTO Product VALUES (225,"Capsicum",70,40,106);
INSERT INTO Product VALUES (216,"Cheese",30,80,   104);
INSERT INTO Product VALUES (219,"Dal",60,70,105);
INSERT INTO Product VALUES (207,"Facewash",300,40,102);
INSERT INTO Product VALUES (213,"Kurkure",10,170,103);
INSERT INTO Product VALUES (212,"Lays",10,150,103);
INSERT INTO Product VALUES (208,"Lip-balm",170,45,102);
INSERT INTO Product VALUES (214,"Milk",12,45,104);
INSERT INTO Product VALUES (206,"Moisturizer",150,50,102);
INSERT INTO Product VALUES (203,"Natraja Pencil",5,102,101);
INSERT INTO Product VALUES (204,"Non-Dust Eraser",10,90,101);
INSERT INTO Product VALUES (201,"Notebook",35,50,101);
INSERT INTO Product VALUES (224,"Onion",100,30,106);
INSERT INTO Product VALUES (210,"PikniK",10,200,103);
INSERT INTO Product VALUES (202,"Reynolds Pen",15,100,101);
INSERT INTO Product VALUES (218,"Rice",30,60,105);
INSERT INTO Product VALUES (221,"Salt",15,20,105);
INSERT INTO Product VALUES (209,"Sanitary Napkins",365,200,102);
INSERT INTO Product VALUES (205,"Sharpener",3,80,101);
INSERT INTO Product VALUES (226,"Tomato",90,30,106);
INSERT INTO Product VALUES (220,"Vegetable Oil",120,40,105);
INSERT INTO Product VALUES (217,"Yogurt",12,35,104);
```

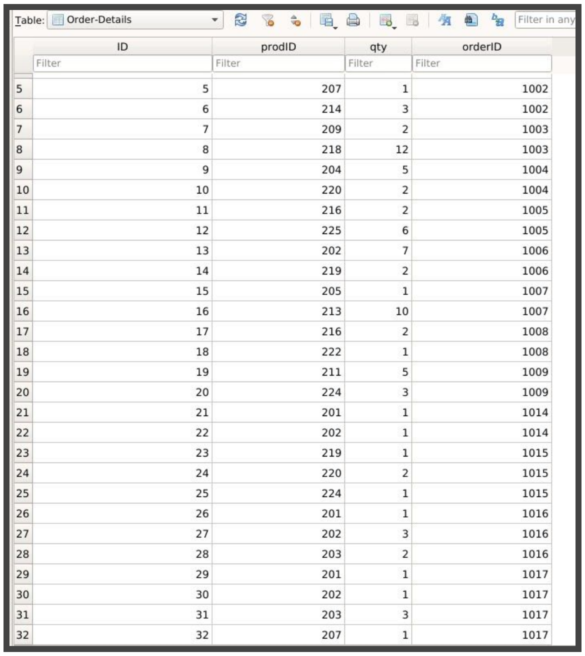


Table: Order-Details

	ID	prodID	qty	orderID
5	5	207	1	1002
6	6	214	3	1002
7	7	209	2	1003
8	8	218	12	1003
9	9	204	5	1004
10	10	220	2	1004
11	11	216	2	1005
12	12	225	6	1005
13	13	202	7	1006
14	14	219	2	1006
15	15	205	1	1007
16	16	213	10	1007
17	17	216	2	1008
18	18	222	1	1008
19	19	211	5	1009
20	20	224	3	1009
21	21	201	1	1014
22	22	202	1	1014
23	23	219	1	1015
24	24	220	2	1015
25	25	224	1	1015
26	26	201	1	1016
27	27	202	3	1016
28	28	203	2	1016
29	29	201	1	1017
30	30	202	1	1017
31	31	203	3	1017
32	32	207	1	1017

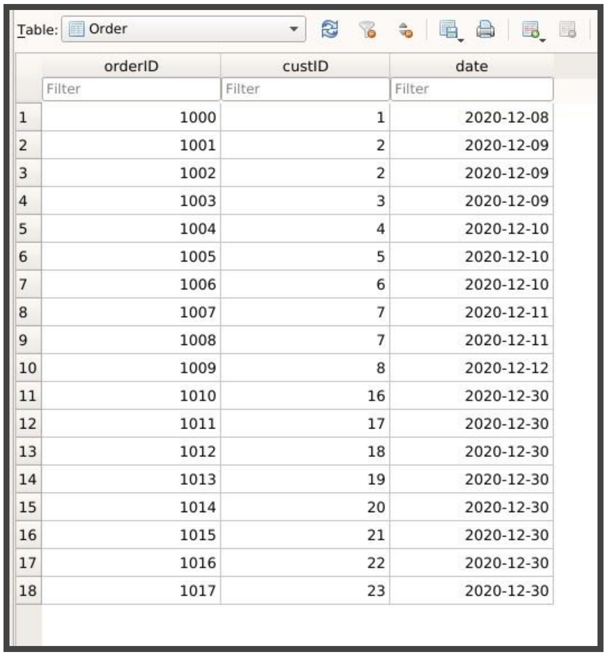
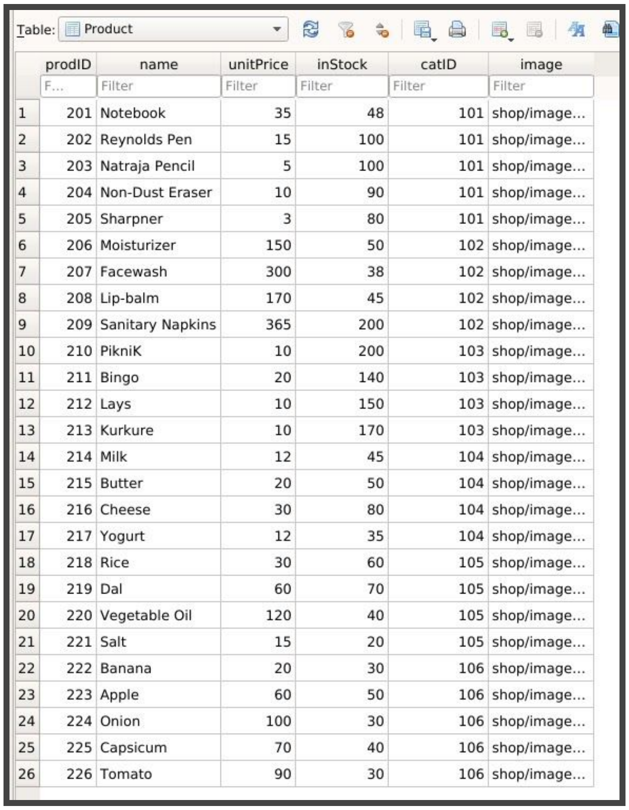
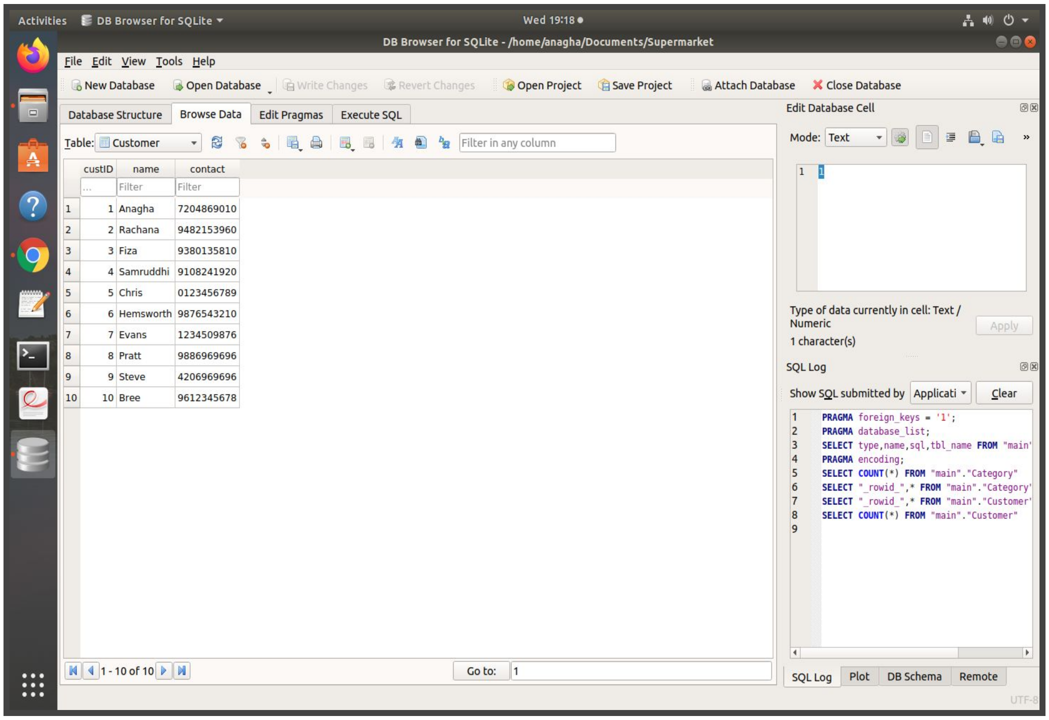


Table: Order

	orderID	custID	date
1	1000	1	2020-12-08
2	1001	2	2020-12-09
3	1002	2	2020-12-09
4	1003	3	2020-12-09
5	1004	4	2020-12-10
6	1005	5	2020-12-10
7	1006	6	2020-12-10
8	1007	7	2020-12-11
9	1008	7	2020-12-11
10	1009	8	2020-12-12
11	1010	16	2020-12-30
12	1011	17	2020-12-30
13	1012	18	2020-12-30
14	1013	19	2020-12-30
15	1014	20	2020-12-30
16	1015	21	2020-12-30
17	1016	22	2020-12-30
18	1017	23	2020-12-30



	prodID	name	unitPrice	inStock	catID	image
	F...	Filter	Filter	Filter	Filter	Filter
1	201	Notebook	35	48	101	shop/image...
2	202	Reynolds Pen	15	100	101	shop/image...
3	203	Natraja Pencil	5	100	101	shop/image...
4	204	Non-Dust Eraser	10	90	101	shop/image...
5	205	Sharpner	3	80	101	shop/image...
6	206	Moisturizer	150	50	102	shop/image...
7	207	Facewash	300	38	102	shop/image...
8	208	Lip-balm	170	45	102	shop/image...
9	209	Sanitary Napkins	365	200	102	shop/image...
10	210	PikniK	10	200	103	shop/image...
11	211	Bingo	20	140	103	shop/image...
12	212	Lays	10	150	103	shop/image...
13	213	Kurkure	10	170	103	shop/image...
14	214	Milk	12	45	104	shop/image...
15	215	Butter	20	50	104	shop/image...
16	216	Cheese	30	80	104	shop/image...
17	217	Yogurt	12	35	104	shop/image...
18	218	Rice	30	60	105	shop/image...
19	219	Dal	60	70	105	shop/image...
20	220	Vegetable Oil	120	40	105	shop/image...
21	221	Salt	15	20	105	shop/image...
22	222	Banana	20	30	106	shop/image...
23	223	Apple	60	50	106	shop/image...
24	224	Onion	100	30	106	shop/image...
25	225	Capsicum	70	40	106	shop/image...
26	226	Tomato	90	30	106	shop/image...



	custID	name	contact
	...	Filter	Filter
1	1	Anagha	7204869010
2	2	Rachana	9482153960
3	3	Fiza	9380135810
4	4	Samruddhi	9108241920
5	5	Chris	0123456789
6	6	Hemsworth	9876543210
7	7	Evans	1234509876
8	8	Pratt	9886969696
9	9	Steve	4206969696
10	10	Bree	9612345678

Type of data currently in cell: Text / Numeric
1 character(s)

```
PRAGMA foreign_keys = '1';
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main'
PRAGMA encoding;
SELECT COUNT(*) FROM "main"."Category"
SELECT "_rowid_",* FROM "main"."Category'
SELECT "_rowid_",* FROM "main"."Customer'
SELECT COUNT(*) FROM "main"."Customer"
```

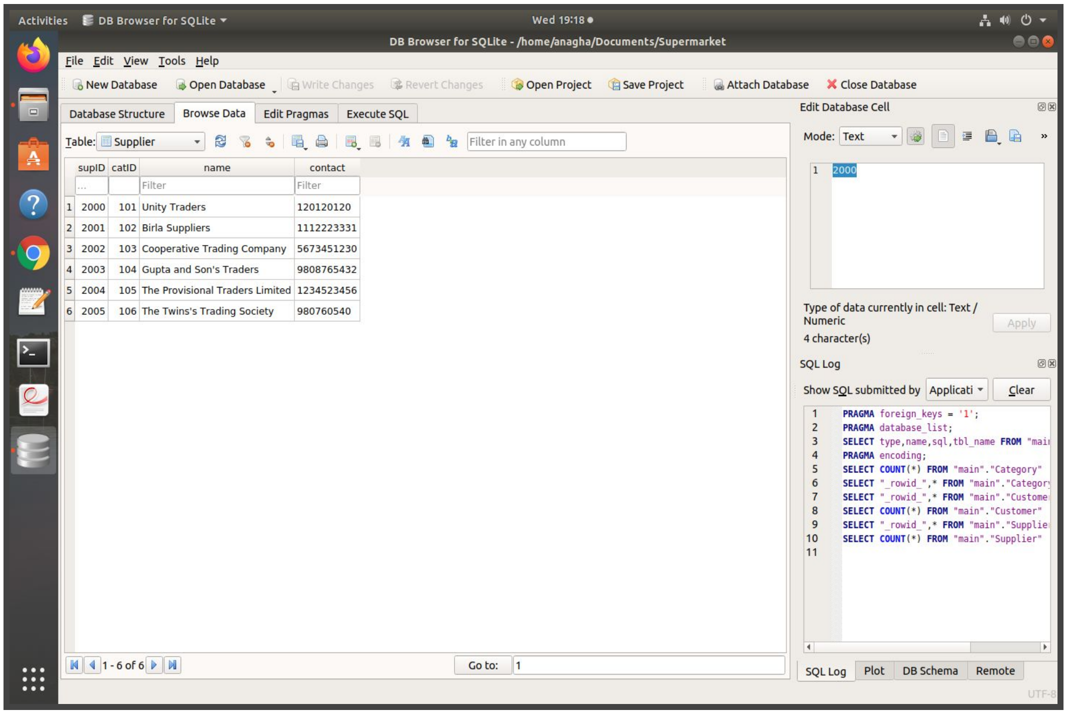


Table: Supplier

	supID	catID	name	contact
1	2000	101	Unity Traders	120120120
2	2001	102	Birla Suppliers	1112223331
3	2002	103	Cooperative Trading Company	5673451230
4	2003	104	Gupta and Son's Traders	9808765432
5	2004	105	The Provisional Traders Limited	1234523456
6	2005	106	The Twins's Trading Society	980760540

Edit Database Cell: 2000

Type of data currently in cell: Text / Numeric

4 character(s)

Show SQL submitted by Applicati

```
PRAGMA foreign_keys = '1';
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "mai
PRAGMA encoding;
SELECT COUNT(*) FROM "main"."Category"
SELECT "_rowid_",* FROM "main"."Category
SELECT "_rowid_",* FROM "main"."Custome
SELECT COUNT(*) FROM "main"."Customer"
SELECT "_rowid_",* FROM "main"."Supplie
SELECT COUNT(*) FROM "main"."Supplier"
```

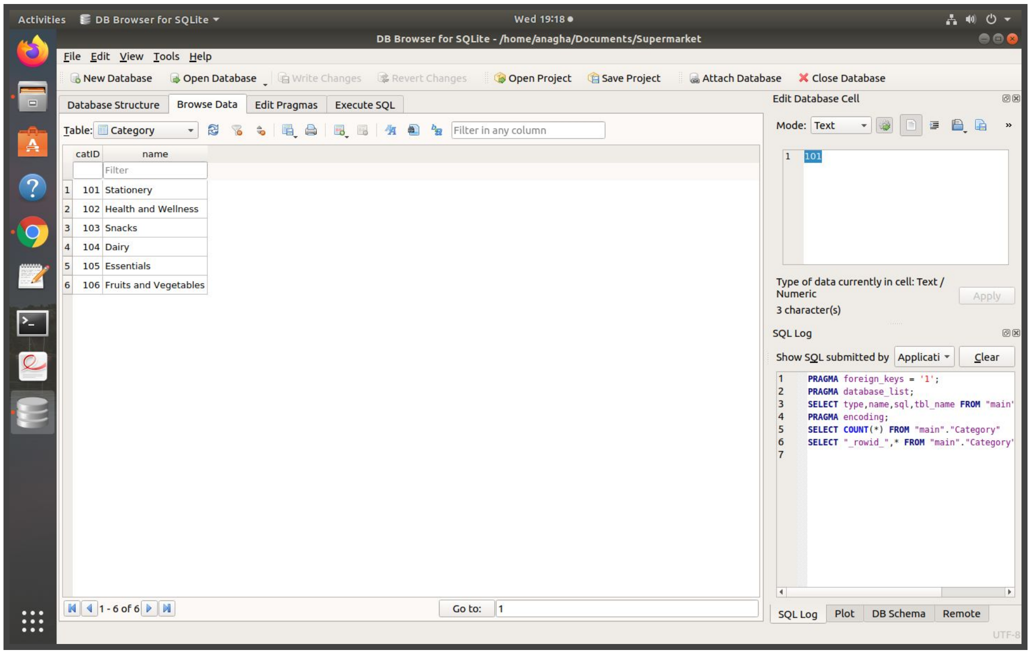


Table: Category

	catID	name
1	101	Stationery
2	102	Health and Wellness
3	103	Snacks
4	104	Dairy
5	105	Essentials
6	106	Fruits and Vegetables

Edit Database Cell: 101

Type of data currently in cell: Text / Numeric

3 character(s)

Show SQL submitted by Applicati

```
PRAGMA foreign_keys = '1';
PRAGMA database_list;
SELECT type,name,sql,tbl_name FROM "main"
PRAGMA encoding;
SELECT COUNT(*) FROM "main"."Category"
SELECT "_rowid_",* FROM "main"."Category"
```

DQL Commands:

Some DQL queries along with what they do has been listed below.

```
/*To find the list of all supplier names, id and product names supplied by them,
where product id is greater than 215.*/
1. SELECT supID as "Supplier ID", Supplier.name as "Supplier Name",
Product.name as "Product Name" FROM Supplier, Product WHERE
Product.prodID > 215 AND Product.catID = Supplier.catID;
```

```
/*To find the list of customer names and their respective IDs who have ordered on
10 Dec 2020.*/
2. SELECT Customer.custID as "Customer ID",name as "Customer Name" FROM
Customer, "Order" WHERE "Order".custID = Customer.custID AND "Order".date
= "2020-12-10";
```

```
/*To find the list of all products and their respective IDs which have been ordered
in quantity greater than 2 and are presented in the decreasing order of quantity*/
3. SELECT Product.prodID as "Product ID",name as "Product Name", qty as
"Quantity greater than 2" FROM Product, "Order-Details" WHERE
Product.prodID = "Order-Details".prodID AND qty>2 ORDER BY qty DESC;
```

```
/*To find the names of all customers whose names either starts with ‘S’ or ends
with ‘S’*/
4. SELECT * FROM Customer WHERE name LIKE 'S%' UNION SELECT *
FROM Customer WHERE name LIKE '%S';
/*To display the list of all customers in ascending order who have placed an
order*/
5. SELECT name FROM Customer WHERE EXISTS(SELECT custID FROM
"Order" WHERE Customer.custID = "Order".custID) GROUP BY custID ORDER
BY name ASC;
```

```
/*To display the list of each category, respective category ID and number of items
in each category.*/
```

```
6.SELECT count(Product.catID) as "Number of Items", Category.catID as
"Category ID", Category.name as "Category Name" FROM Product, Category
WHERE Product.catID = Category.catID GROUP BY Product.catID;

/*To find the product name and product ID which has the maximum unit price.*/
7. SELECT max(unitPrice), name, prodID FROM Product;

/*To find the product name and product ID which has the minimum number of
quantity in stock.*/
8. SELECT min(inStock), name, prodID FROM Product;


/*All customers names, ID who have ordered and who have not ordered*/
9. SELECT Customer.name, Customer.custID FROM Customer NATURAL JOIN
"Order";

/*Product names and quantities which have not been ordered.*/
10. SELECT name, Product.name, qty FROM Product LEFT OUTER JOIN
"Order-Details" ON Product.prodID = "Order-Details".prodID;

11. /*All the products supplied by "Unity Traders" and "The Provisional Traders
Limited"*/
SELECT P.prodID, P.name, S.name FROM Product as P, Supplier as S WHERE
P.catID = S.catID AND S.name="Unity Traders" UNION SELECT P.prodID,
P.name, S.name FROM Product as P, Supplier as S WHERE P.catID = S.catID
AND S.name="The Provisional Traders Limited";

12. /*Details of all the orders in one table*/
SELECT C.custID, O.orderID, O.date, P.name, "OR".qty FROM Customer as C,
"Order" as O, Product as P , "Order-Details" as "OR" WHERE C.custID =
O.custID AND O.orderID = "OR".orderID AND "OR".prodID= P.prodID;

13. /*Using Natural Join*/
SELECT Customer.name, "Order".orderID FROM (Customer NATURAL JOIN
"Order");
```

14. /*Group all the ordered products and count the quantity required from each one of them*/

```
SELECT prodID, sum(qty) FROM "Order-details" GROUP BY prodID;
```

15. /*Customers who placed an order at least once*/

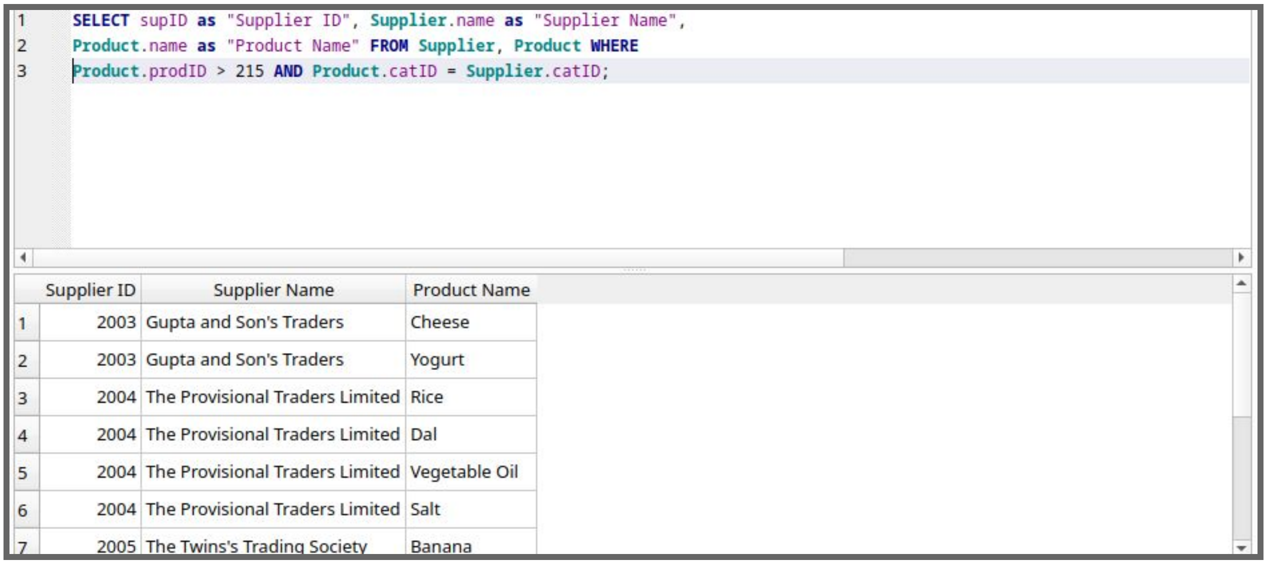
```
SELECT DISTINCT Customer.name FROM Customer, "Order" WHERE
Customer.custID = "Order".custID AND orderID IN (SELECT "Order".orderID
FROM (Customer  LEFT OUTER JOIN "Order" ON "Order".custID =
Customer.custID));
```

16. /*Customers who did not order anything*/

```
SELECT Customer.name FROM Customer WHERE NOT EXISTS (SELECT *
FROM "Order" WHERE Customer.custID = "Order".custID);
```

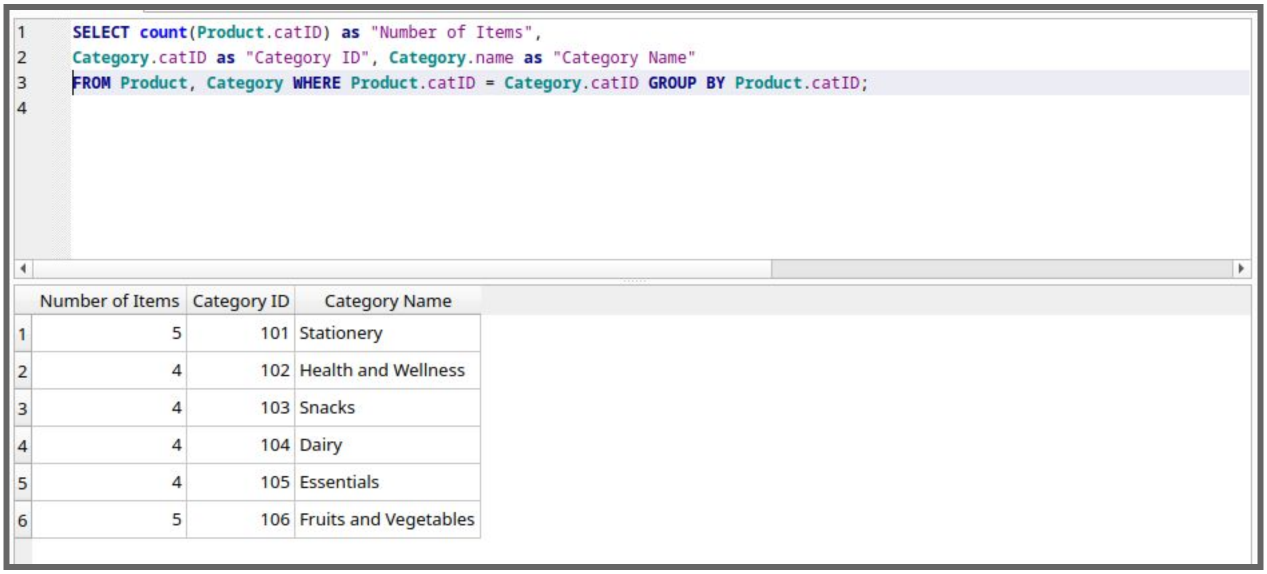
17. /*Quantity of each product Ordered*/

```
SELECT name AS Product_Name, unitPrice as Unit_price, sum(qty) AS
Total_Product_Ordered FROM (SELECT product.name, product.unitPrice, OD.qty
FROM Product JOIN "Order-Details" AS OD ON Product.prodID=OD.prodID
Order BY name)GROUP BY name, unitPrice;
```



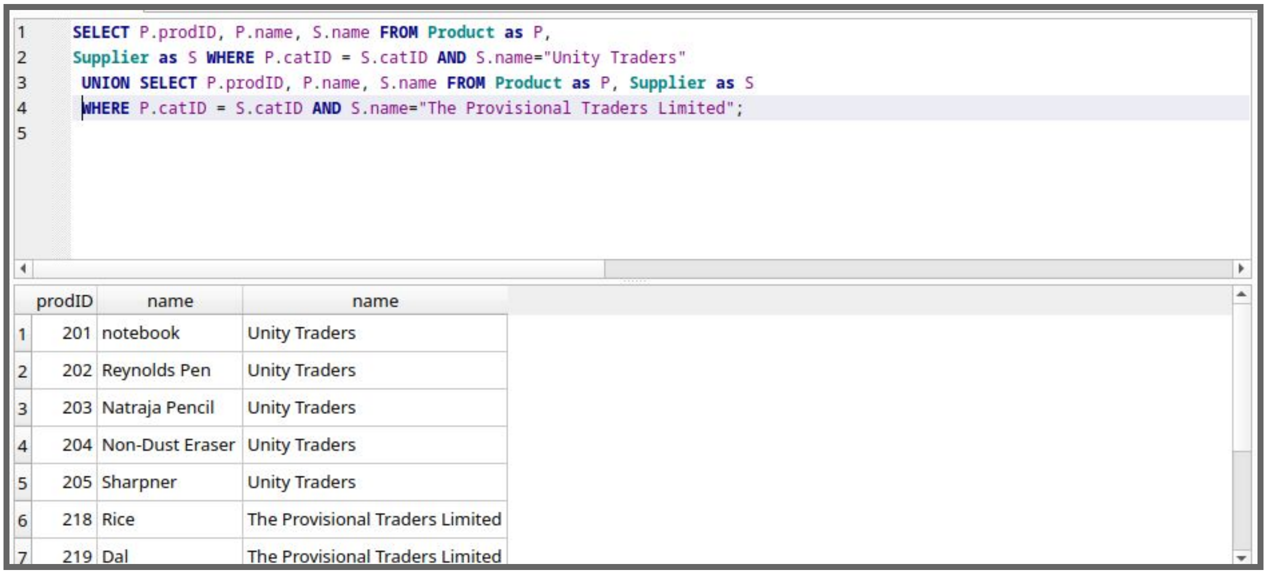
```sql
SELECT supID as "Supplier ID", Supplier.name as "Supplier Name",
Product.name as "Product Name" FROM Supplier, Product WHERE
Product.prodID > 215 AND Product.catID = Supplier.catID;
```

	Supplier ID	Supplier Name	Product Name
1	2003	Gupta and Son's Traders	Cheese
2	2003	Gupta and Son's Traders	Yogurt
3	2004	The Provisional Traders Limited	Rice
4	2004	The Provisional Traders Limited	Dal
5	2004	The Provisional Traders Limited	Vegetable Oil
6	2004	The Provisional Traders Limited	Salt
7	2005	The Twins's Trading Society	Banana



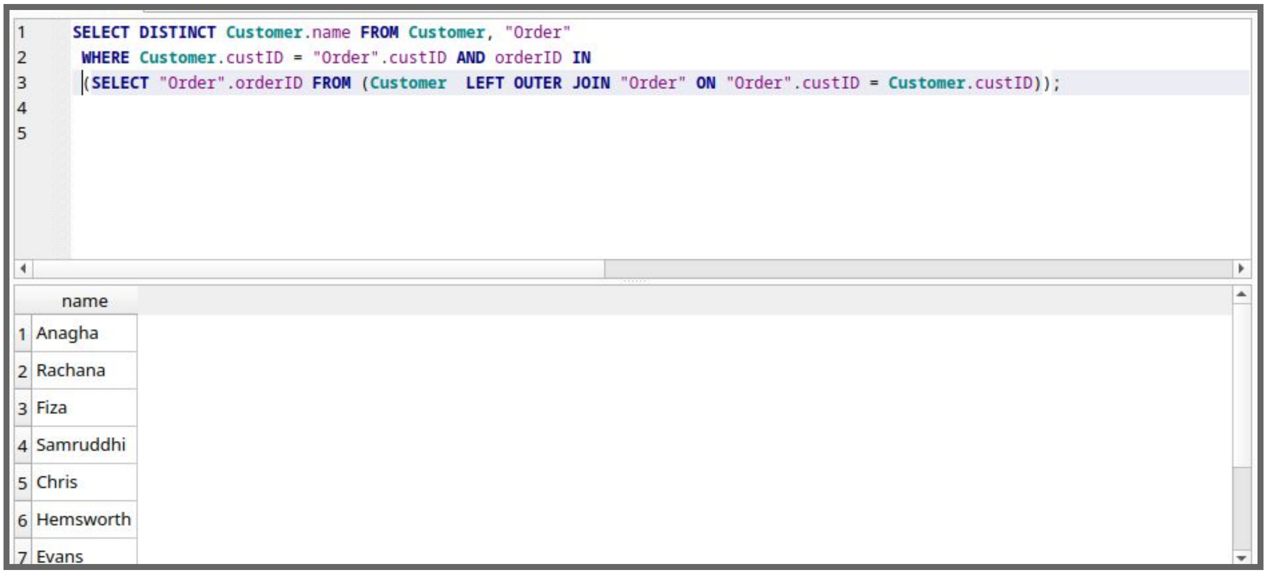
```sql
SELECT count(Product.catID) as "Number of Items",
Category.catID as "Category ID", Category.name as "Category Name"
FROM Product, Category WHERE Product.catID = Category.catID GROUP BY Product.catID;
```

	Number of Items	Category ID	Category Name
1	5	101	Stationery
2	4	102	Health and Wellness
3	4	103	Snacks
4	4	104	Dairy
5	4	105	Essentials
6	5	106	Fruits and Vegetables



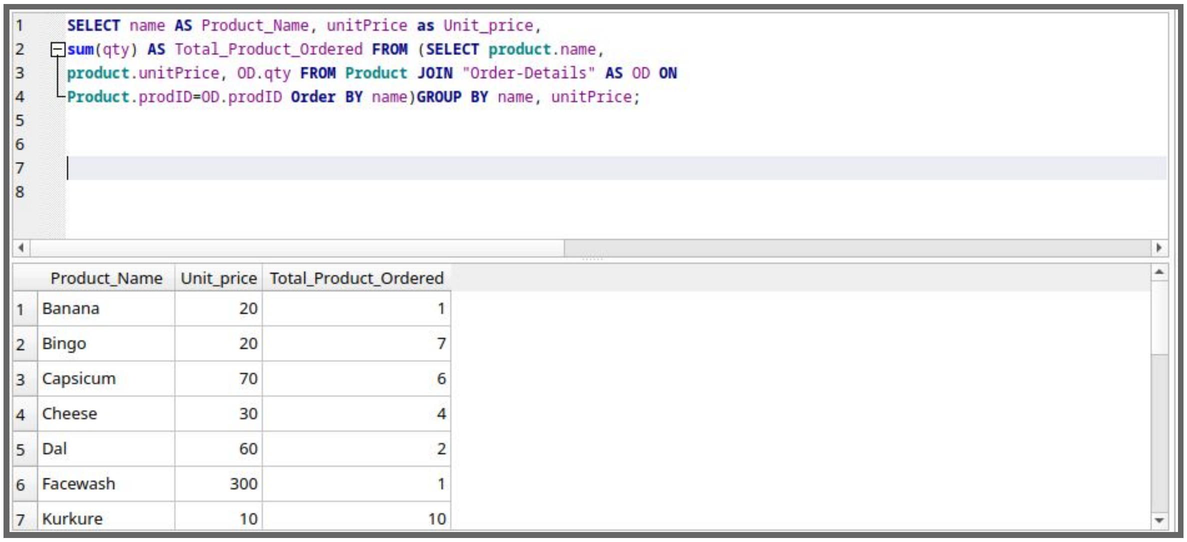
```sql
SELECT P.prodID, P.name, S.name FROM Product as P,
Supplier as S WHERE P.catID = S.catID AND S.name="Unity Traders"
 UNION SELECT P.prodID, P.name, S.name FROM Product as P, Supplier as S
 WHERE P.catID = S.catID AND S.name="The Provisional Traders Limited";
```

	prodID	name	name
1	201	notebook	Unity Traders
2	202	Reynolds Pen	Unity Traders
3	203	Natraja Pencil	Unity Traders
4	204	Non-Dust Eraser	Unity Traders
5	205	Sharpner	Unity Traders
6	218	Rice	The Provisional Traders Limited
7	219	Dal	The Provisional Traders Limited



```sql
SELECT DISTINCT Customer.name FROM Customer, "Order"
 WHERE Customer.custID = "Order".custID AND orderID IN
 (SELECT "Order".orderID FROM (Customer  LEFT OUTER JOIN "Order" ON "Order".custID = Customer.custID));
```

	name
1	Anagha
2	Rachana
3	Fiza
4	Samruddhi
5	Chris
6	Hemsworth
7	Evans

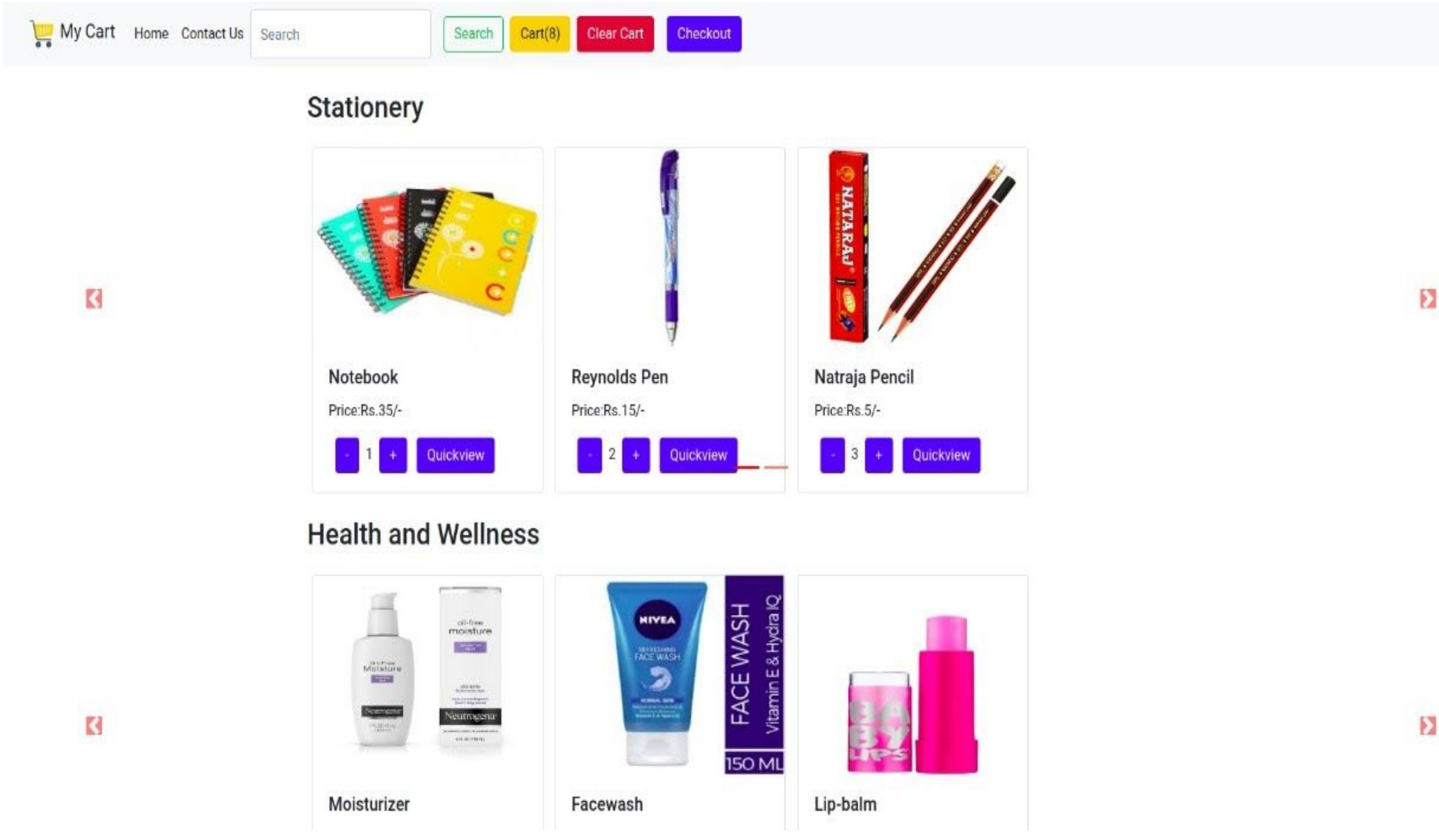


```sql
SELECT name AS Product_Name, unitPrice as Unit_price,
sum(qty) AS Total_Product_Ordered FROM (SELECT product.name,
product.unitPrice, OD.qty FROM Product JOIN "Order-Details" AS OD ON
Product.prodID=OD.prodID Order BY name)GROUP BY name, unitPrice;
```

	Product_Name	Unit_price	Total_Product_Ordered
1	Banana	20	1
2	Bingo	20	7
3	Capsicum	70	6
4	Cheese	30	4
5	Dal	60	2
6	Facewash	300	1
7	Kurkure	10	10

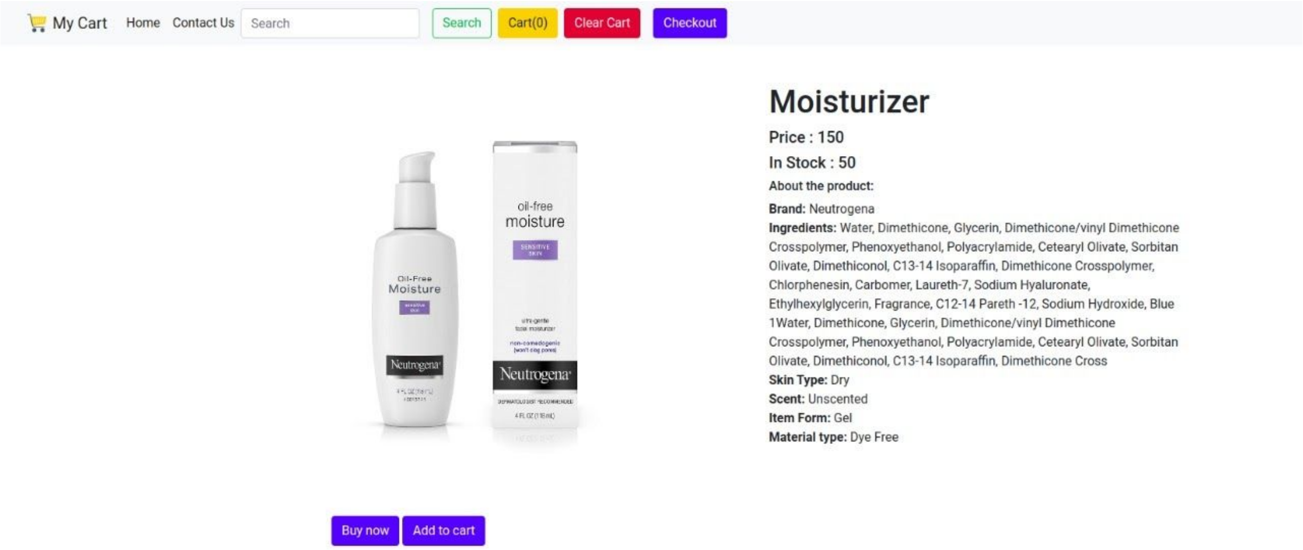
Front-end Design

The front-end has been designed using HTML5, CSS and JavaScript. It has been linked to the database using Django. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.



The home-page of our website consists of all the products listed under their respective categories. The product's prices have been listed and the end-user can choose the required products and the quantity they want directly by selecting the '+' and '-' buttons. The value on the cart icon(the yellow button at the top) will change accordingly. The end-user can also click the 'QuickView' button to read each product's description.

At the top, in the Navbar, We've been given the option 'Clear Cart' which will remove all the items present in the cart. We also have the 'Checkout' option. A search bar has been provided to search for items. A ‘Home’ button and a ‘Contact Us’ option has also been provided.



Moisturizer

Price : 150

In Stock : 50

About the product:

Brand: Neutrogena

Ingredients: Water, Dimethicone, Glycerin, Dimethicone/vinyl Dimethicone Crosspolymer, Phenoxyethanol, Polyacrylamide, Cetearyl Olivate, Sorbitan Olivate, Dimethiconol, C13-14 Isoparaffin, Dimethicone Crosspolymer, Chlorphenesin, Carbomer, Laureth-7, Sodium Hyaluronate, Ethylhexylglycerin, Fragrance, C12-14 Pareth -12, Sodium Hydroxide, Blue 1Water, Dimethicone, Glycerin, Dimethicone/vinyl Dimethicone Crosspolymer, Phenoxyethanol, Polyacrylamide, Cetearyl Olivate, Sorbitan Olivate, Dimethiconol, C13-14 Isoparaffin, Dimethicone Cross

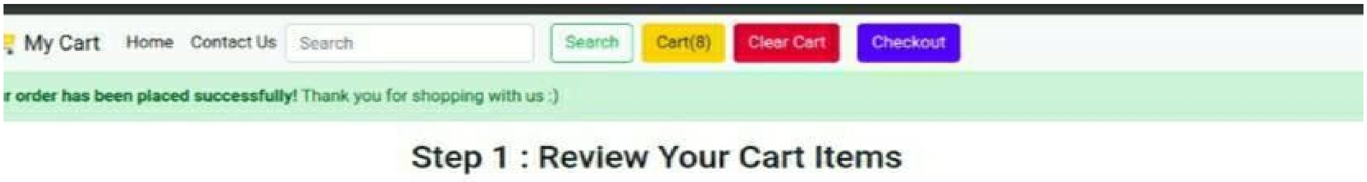
Skin Type: Dry

Scent: Unscented

Item Form: Gel

Material type: Dye Free

When we click on a product. It’s description will be displayed as shown above.



r order has been placed successfully! Thank you for shopping with us :)

Step 1 : Review Your Cart Items

Once an order has been placed successfully, the above message will be displayed to the end-user. Else, an error message will be given.

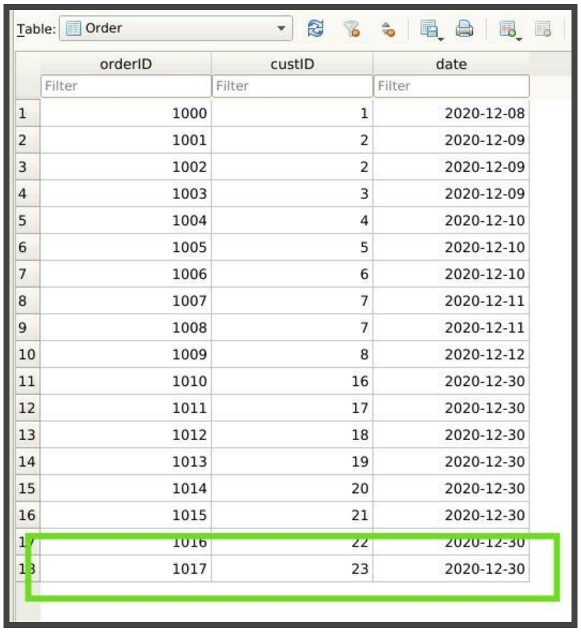


Table: Order

	orderID	custID	date
	Filter	Filter	Filter
1	1000	1	2020-12-08
2	1001	2	2020-12-09
3	1002	2	2020-12-09
4	1003	3	2020-12-09
5	1004	4	2020-12-10
6	1005	5	2020-12-10
7	1006	6	2020-12-10
8	1007	7	2020-12-11
9	1008	7	2020-12-11
10	1009	8	2020-12-12
11	1010	16	2020-12-30
12	1011	17	2020-12-30
13	1012	18	2020-12-30
14	1013	19	2020-12-30
15	1014	20	2020-12-30
16	1015	21	2020-12-30
17	1016	22	2020-12-30
18	1017	23	2020-12-30

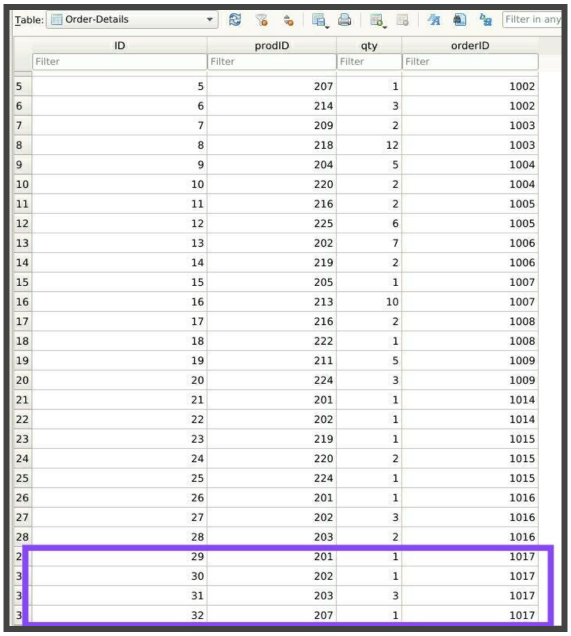


Table: Order-Details

Filter in any

	ID	prodID	qty	orderID
	Filter	Filter	Filter	Filter
5	5	207	1	1002
6	6	214	3	1002
7	7	209	2	1003
8	8	218	12	1003
9	9	204	5	1004
10	10	220	2	1004
11	11	216	2	1005
12	12	225	6	1005
13	13	202	7	1006
14	14	219	2	1006
15	15	205	1	1007
16	16	213	10	1007
17	17	216	2	1008
18	18	222	1	1008
19	19	211	5	1009
20	20	224	3	1009
21	21	201	1	1014
22	22	202	1	1014
23	23	219	1	1015
24	24	220	2	1015
25	25	224	1	1015
26	26	201	1	1016
27	27	202	3	1016
28	28	203	2	1016
2	29	201	1	1017
3	30	202	1	1017
3	31	203	3	1017
3	32	207	1	1017

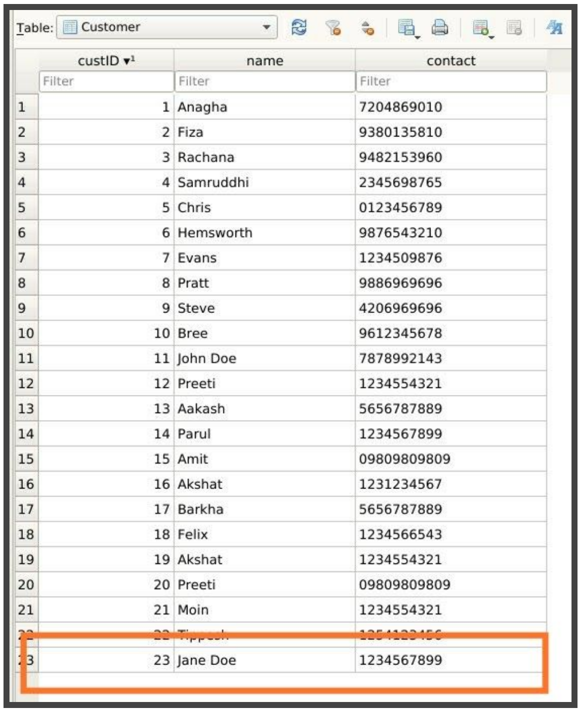
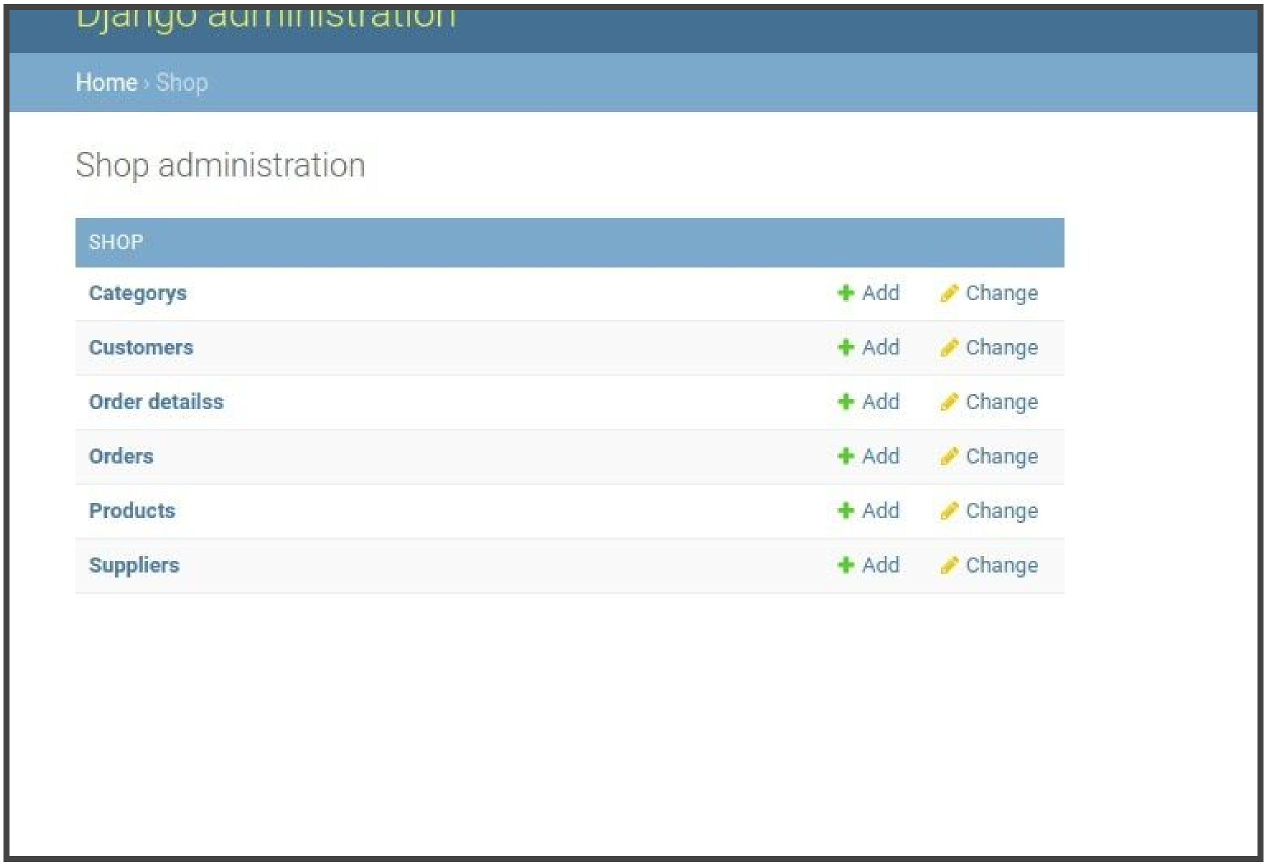


Table: Customer

	custID ▾1	name	contact
	Filter	Filter	Filter
1	1	Anagha	7204869010
2	2	Fiza	9380135810
3	3	Rachana	9482153960
4	4	Samruddhi	2345698765
5	5	Chris	0123456789
6	6	Hemsworth	9876543210
7	7	Evans	1234509876
8	8	Pratt	9886969696
9	9	Steve	4206969696
10	10	Bree	9612345678
11	11	John Doe	7878992143
12	12	Preeti	1234554321
13	13	Aakash	5656787889
14	14	Parul	1234567899
15	15	Amit	09809809809
16	16	Akshat	1231234567
17	17	Barkha	5656787889
18	18	Felix	1234566543
19	19	Akshat	1234554321
20	20	Preeti	09809809809
21	21	Moin	1234554321
22	22	Tippesh	1254123456
23	23	Jane Doe	1234567899

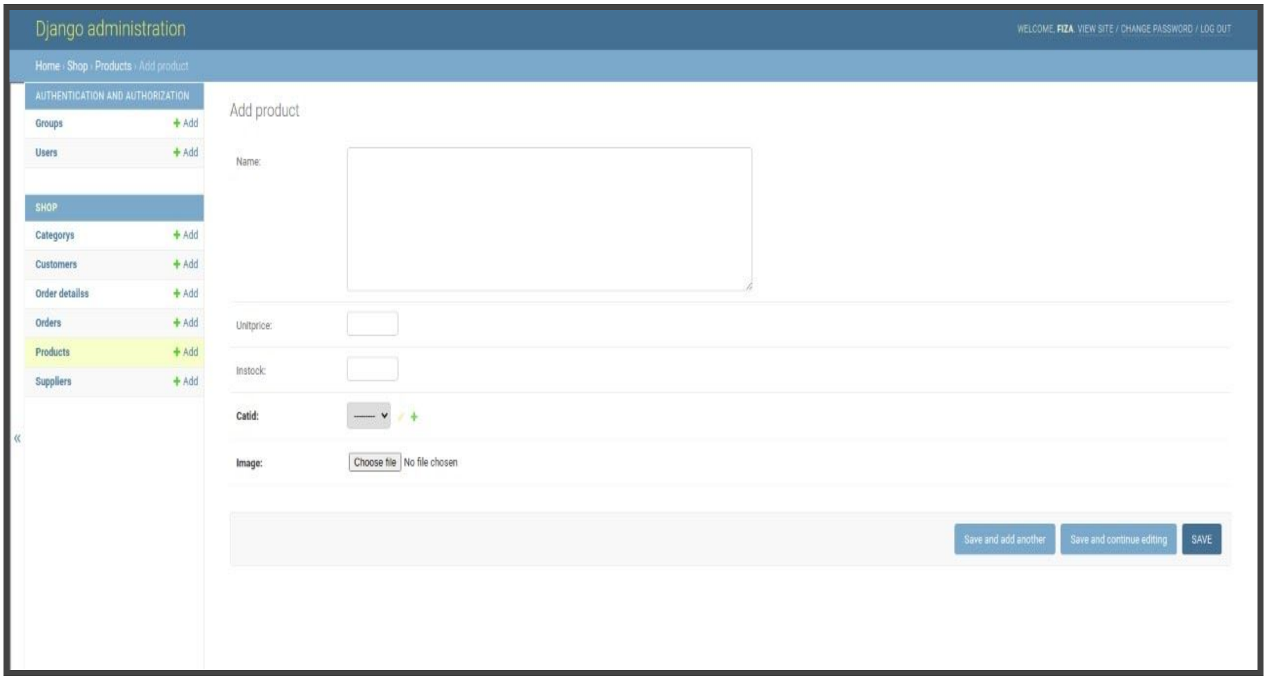
In the following images, you can see the database being updated in real-time as a customer checks out her products from the store.

The following images show the Django admin page that was made in order to connect the front-end and the database.



Shop administration

SHOP		
Categorys	+ Add	Change
Customers	+ Add	Change
Order detailss	+ Add	Change
Orders	+ Add	Change
Products	+ Add	Change
Suppliers	+ Add	Change



Add product

Name:

Unitprice:

Instock:

Catid: ——— +

Image: Choose file No file chosen

CONCLUSION

We have successfully built a website and linked it to a database using various tools. We now have a clear understanding about all the basic tools one requires to build an Ecommerce website. We now have a clear understanding of designing a database in an efficient way so as to avoid redundancy. We have learnt to use Django and Python as back-end tools to link the database. We have also learnt to use scripting languages like HTML5 and CSS along with Javascript to make our front-end design.

We would like to thank Dr. Manimala ma'am and Ashrita ma'am for having been a guiding light for us throughout the project.