



PROJECT ON DATABASE MANAGEMENT SYSTEM

“CAMPUS MART”



Submitted by:

PES1201700005: Bharani Ujjaini Kempaiah

PES1201700032: Bhavya Charan

PES1201700646: Tanvi P Karennavar

TABLE OF CONTENTS:

PROBLEM STATEMENT	2
THE MINI WORLD STATEMENT	2
REQUIREMENTS	2
ER DIAGRAM	3
SCHEMA DIAGRAM	4
CREATE TABLE STATEMENTS	5
UI SCREENS	7
COMPLEX QUERIES	10
EXCEPTION HANDLING	11
DISCUSSIONS AND CONCLUSIONS	11

PROBLEM STATEMENT:

Our database Campusmart has been designed by keeping the existing 'campusmart' in PES University in mind. We have developed a system which makes it easier for the shopkeeper to keep track of their sales.

In reality since, the shopkeeper takes no account of the buyer, we have replicated the same where only the details of the order are stored and details of the customer are not dealt with.

This database helps in reducing the time that a buyer spends in conversing with the shopkeeper to find out whether a certain product exists and whether there is sufficient stock.

The user can look at the stock before visiting the mart, decide whether they need to visit the store or not.

If they visit the store, there will be no delay in deciding the product and the shopkeeper notes the transaction and the whole system runs smoothly.

This database was designed to increase the latency of each customer's transaction and the throughput of the sales of the Campusmart as a whole.

THE MINI WORLD STATEMENT:

The database contains information about the campus mart. The campus mart is managed by the administrator. Each administrator has a password and a login id which uniquely identifies them. The administrator manages the stock of products. The stock is supplied by the supplier. The supplier has a name, id (identifies them), address, phone number, and the id of the product they supply. It has a set of products. The campus mart contains products. Every product belongs to a particular category. Category has a unique category id and a name. Each product has a unique product id, a name, description, price, category id, price. Orders created in the campus mart contain items which are chosen from products. Every order has a unique order number, an order date, total amount, items. An item has a quantity, product id (unique) and price.

REQUIREMENTS:

Admin options:

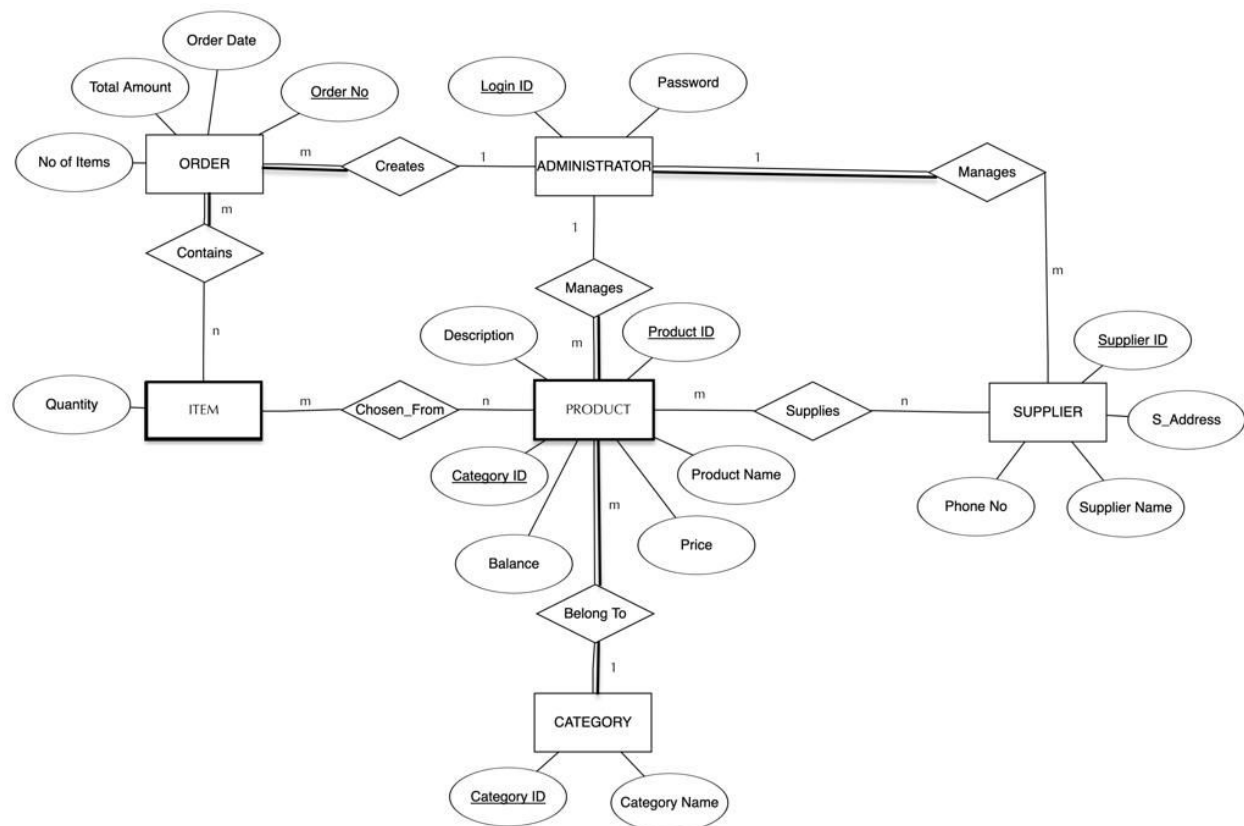
The website provides the admin of Campus Mart to create order for the new customers. If we want to add a new product to our store, that can be done through a form. Changing details of an existing product, Updating stock of an existing product can all be done through separate forms.

If we have got a new supplier for our products, he can also be added. The details of the existing supplier can also be modified. All these operations can only be used by the owner (admin) and he can access all these forms by logging in using the required user ID and passcode.

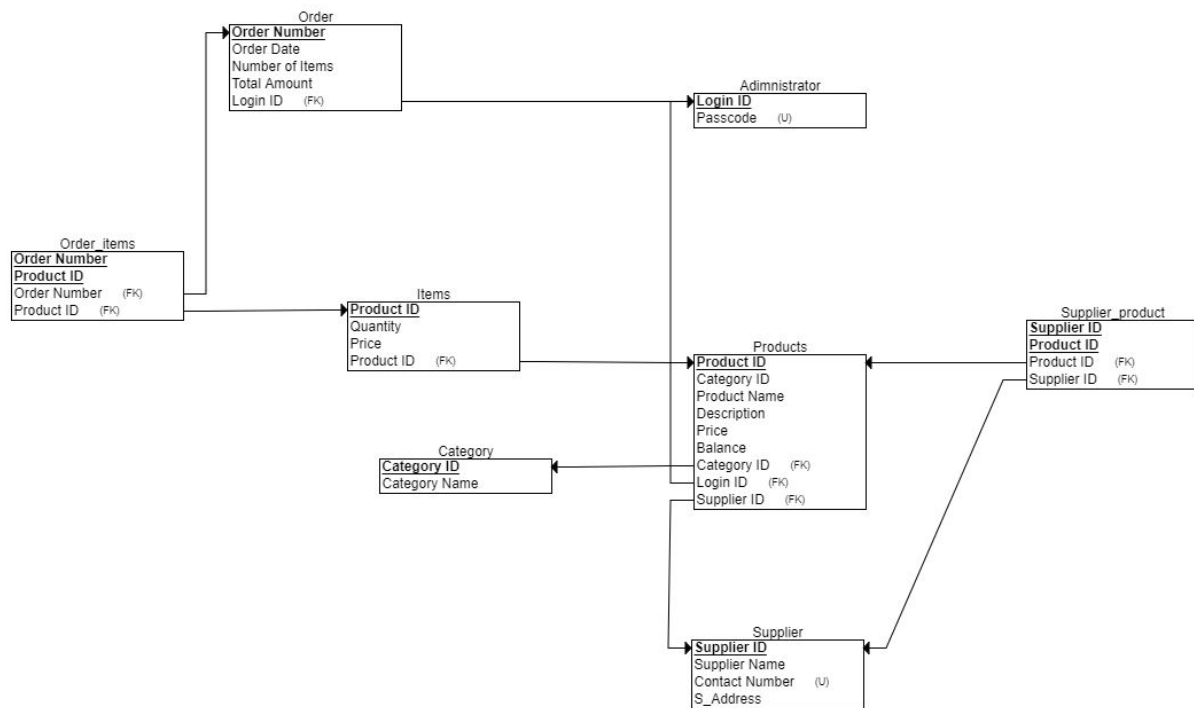
User options:

The website provides the user to view all the products segregated by 7 different categories. Each of the categories displays the product name, product image, price, stock and also the product description.

ER DIAGRAM:



SCHEMA DIAGRAM:



CREATE TABLE STATEMENTS:

```
CREATE TABLE PLACE_ORDER
(
    Order_Number INT NOT NULL,
    Order_Date DATE NOT NULL,
    Number_of_Items INT NOT NULL,
    Total_Amount FLOAT NOT NULL,
    PRIMARY KEY (Order_Number)
);
```

```
CREATE TABLE ADMINISTRATOR
(
    Login_ID VARCHAR(20) NOT NULL,
    Password VARCHAR(10) NOT NULL,
    PRIMARY KEY (Login_ID),
    UNIQUE (Password)
);
```

```
CREATE TABLE CATEGORY
(
    Category_ID INT NOT NULL,
    Category_Name VARCHAR(30) NOT NULL,
    PRIMARY KEY (Category_ID)
);
```

```
CREATE TABLE CAMPUS_MART
(
    Contact_Number NUMERIC(10) NOT NULL,
    Timings VARCHAR(20) NOT NULL,
    Address VARCHAR(50) NOT NULL,
    UNIQUE (Contact_Number)
);
```

```
CREATE TABLE PRODUCTS
(
    Product_ID INT NOT NULL,
    Category_ID INT NOT NULL,
    Product_Name VARCHAR(30) NOT NULL,
    Description VARCHAR(100) NOT NULL,
```

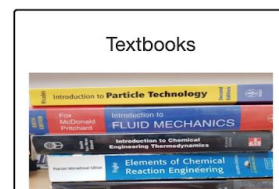
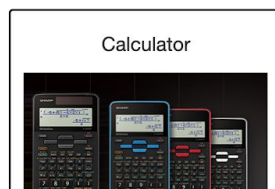
```
Price FLOAT NOT NULL,  
PRIMARY KEY (Product_ID),  
FOREIGN KEY (Category_ID) REFERENCES CATEGORY(Category_ID)  
);
```

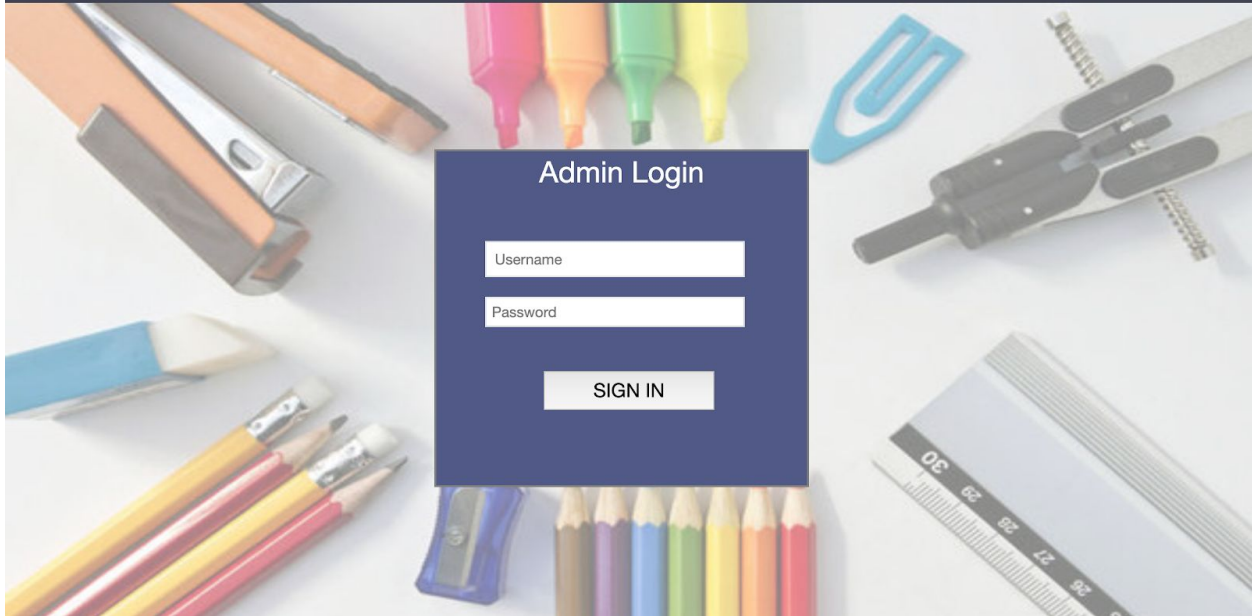
```
CREATE TABLE ITEMS  
(  
    Product_ID INT NOT NULL,  
    Quantity INT NOT NULL,  
    Price FLOAT NOT NULL,  
    PRIMARY KEY (Product_ID),  
    FOREIGN KEY (Product_ID) REFERENCES PRODUCTS(Product_ID),  
    UNIQUE (Product_ID)  
);
```

```
CREATE TABLE ORDER_ITEMS  
(  
    Order_Number INT NOT NULL,  
    Product_ID INT NOT NULL,  
    PRIMARY KEY (Order_Number, Product_ID),  
    FOREIGN KEY (Order_Number) REFERENCES PLACE_ORDER(Order_Number),  
    FOREIGN KEY (Product_ID) REFERENCES ITEMS(Product_ID)  
);
```

```
CREATE TABLE SUPPLIER  
(  
    supplier_name VARCHAR(40) NOT NULL,  
    supplier_id INT NOT NULL,  
    contact_number VARCHAR(10) NOT NULL,  
    s_address VARCHAR(40) NOT NULL,  
    category_id INT NOT NULL,  
    FOREIGN KEY (category_id) REFERENCES category(category_id)  
);
```

UI SCREENS:





Admin Login

[SIGN IN](#)

At Campus Mart you can

Got a new customer ?
Just create their order here!

[Create Order](#)

Want to add a new product
to your store ?
Enter the details here!

[Add a new Product](#)

Change the details of an
existing product ?
You can do that here

[Modify an existing
product](#)

Want to update the stock of
a product ?

[Update stock of existing
product](#)

Got a new supplier ?
Add them here

[Add a new Supplier](#)

Changing details of an
existing supplier ?
Do it here

[Modify details of existing
supplier](#)



Enter details of the items in the order

Enter product name

Enter product quantity

Enter product name

Enter product quantity

Enter product name

Enter product quantity

[Add more products](#)

[Create Order](#)

Get a quick reference to all products



Enter new stock

Product name :

New stock :

[Cancel](#)

[Update stock](#)

Get a quick reference to all products

ABOUT US

Campus Book Mart is a local bookstore serving the students, faculty and staff of PES University - Bangalore with pride. Our primary goal is to ensure students are able to obtain the course materials they need at reasonable prices. We also provide many other items to support your academic career and college spirit. Please browse our website or stop by our store anytime!

ADDRESS

PES University
100 Feet Ring Road, BSK III stage
Bangalore - 560085

CONTACT

Ph-no : +91 80 26721983
+91 80 26772108

COMPLEX QUERIES:

1. Which supplier supplies all products?

```
SELECT supplier_name, supplier_id
FROM supplier
WHERE NOT EXISTS
( SELECT product_id
FROM products EXCEPT
( SELECT product_id from supplier_product where
supplier.supplier_id = supplier_product.supplier_id));
```

2. How many orders exceeded a total amount of 500 on 25th March, 2019?

```
SELECT count(*)
FROM all_orders
WHERE total_amount >500 AND order_date = '2019-03-25';
```

3. What was the total amount of sales in the month of march?

```
SELECT sum(total_amount)
FROM all_orders
WHERE ( SELECT EXTRACT( MONTH FROM order_date))=3;
```

4. Select all products of category 3 which cost less than 170 rupees?

```
SELECT product_id, product_name, price
FROM products
WHERE category_id = 3 AND price<170;
```

5. How many suppliers are from BSK?

```
SELECT supplier_id ,supplier_name ,s_address ,count(*)
FROM supplier
WHERE s_address like '%BSK%';
```

6. What is the minimum and maximum bill amount amongst all orders till date?

```
SELECT min(total_amount) as min_amount ,max(total_amount) as
max_amount
FROM all_orders;
```

7. What all categories of products have been bought by order number 1?

```
SELECT category_name, category_id
FROM category as c, products as p, order_items as o
WHERE( o.order_number = 1 AND o.product_id = p.product_id
AND p.category_id = c.category_id);
```

EXCEPTION HANDLING:

1. Allow the modifications to the database only if the user ID and password matches that of the owner's ID and passcode. The modifications include create order ,Add a new product ,Modify an existing product ,Update stock of existing product ,Add a new supplier and modify the details of existing supplier.
2. The create order and all the other modifications can happen only if the product is present in the database (checks for the primary key product_id).

DISCUSSIONS AND CONCLUSIONS:

1. What all features of a product are important, which have to be stored in the database?

This question was one of the most important questions to be answered considering the fact that the database should facilitate the daily tasks of the admin and help in smooth functioning. For the conventional department store, certain attributes such as color etc are not that important and thus we decided on the given set of attributes name, category, description.

2. How do we store information about orders?

Sculpting the right structure for this took us some time because all information about an order has to be captured and there should be proper storage of this data such that the admin can make use of it to the benefit of the store.

We implemented 3 tables

- all_orders : which contains information about all orders i.e quantity price etc
- items : this is a temporary buffer table that facilitates the functioning of the BEGIN and COMMIT clauses by ensuring all the products that were bought in an order are update in the database at the same time.
- order_items : this table has been created due to the m:n relationship between order and products