DBMS PROJECT REPORT

Team Members:
ASHWIN SHEORAN, 2020288
HARSH GOYAL, 2020562
HARSHIT GARG, 2020301
MEGHNA, 2020080

GROUP NUMBER: 64

ONLINE RETAIL STORE SYSTEM

APPLICATION AND ITS SCOPE

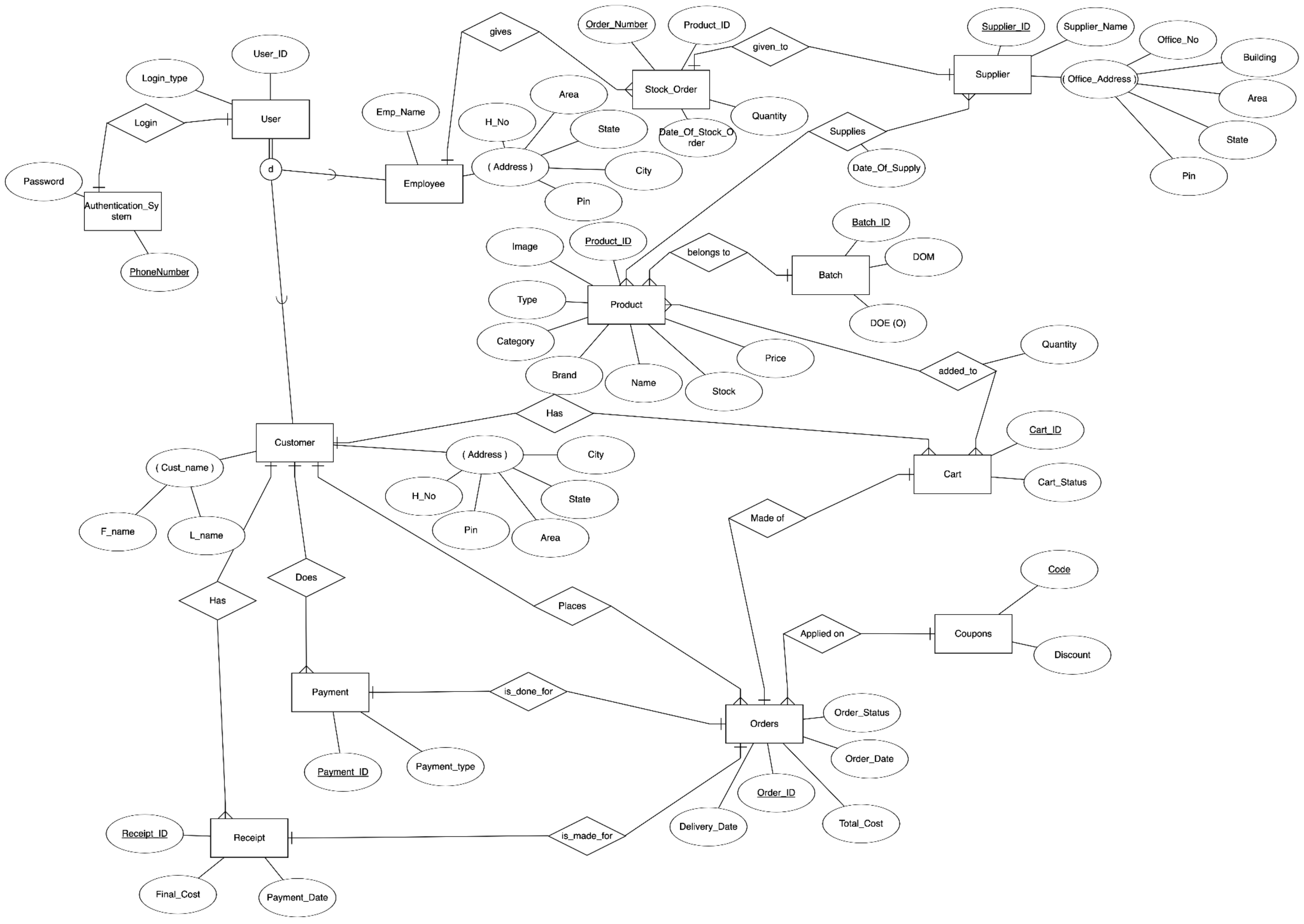
We have designed an online retail store system like Big bazaar. Where Customers can view and purchase products and the Employees can order stock from the Suppliers. The transactions can be done through Net-Banking, E-wallets, Credit/Debit-Cards or even pay through Cash on Delivery.

STAKEHOLDERS

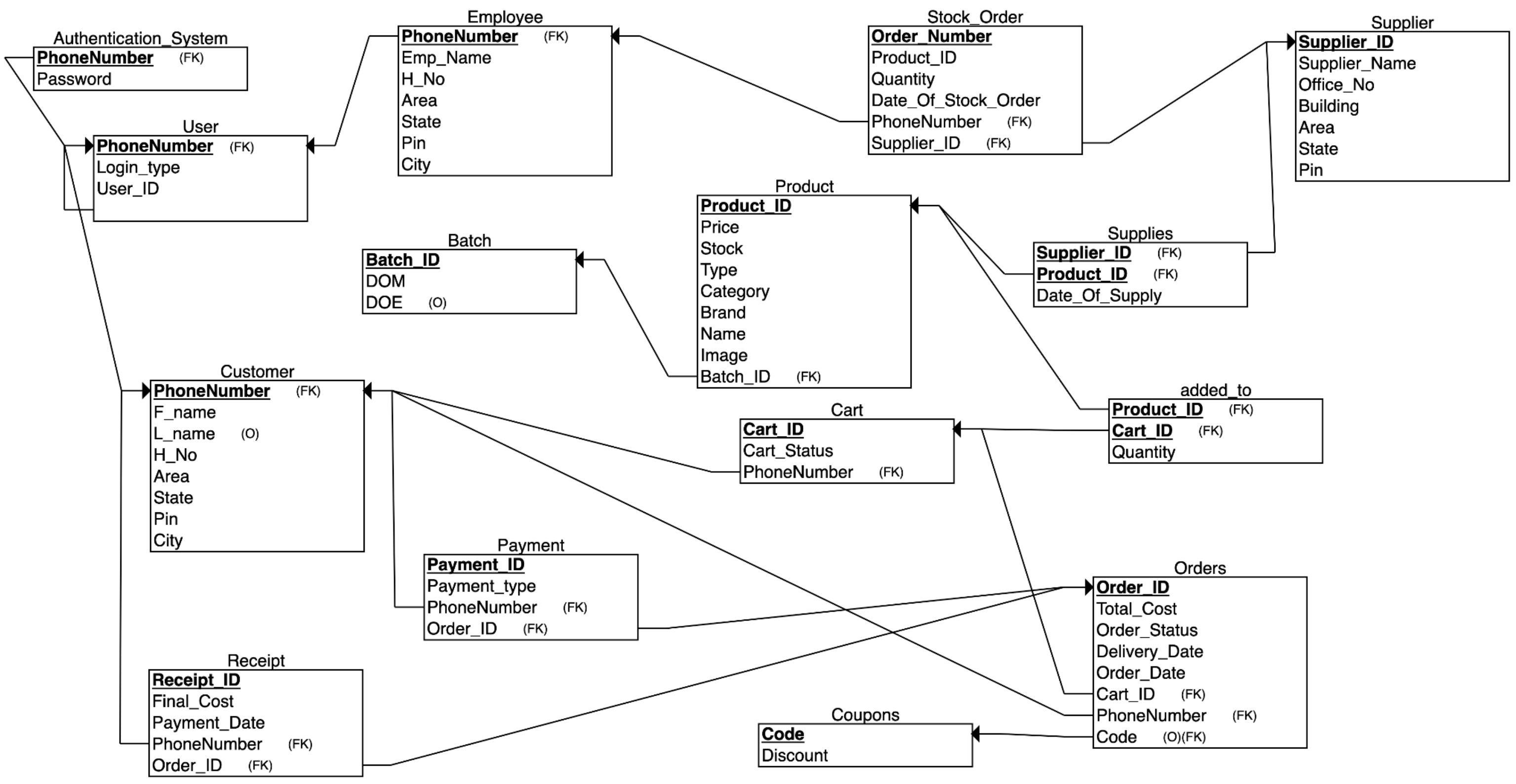
The Stakeholder for our application would be

- Customers (View and Purchase Products)
- Employees (Manage the Data in the Tables, Products and Discounts)
- Database Admin (Manage the Database, And the Tables)
- Suppliers (Take stock orders from Employees and supply the products)

ER MODEL


gives
Order_Number
Product_ID
given_to
Supplier_ID
Supplier_Name
Office_No
Building
User_ID
Login_type
User
Login
Stock_Order
Supplier
(Office_Address)
Area
Emp_Name
H_No
Area
State
Quantity
Date_Of_Stock_Order
Supplies
State
d
(Address)
City
Date_Of_Supply
Pin
Password
Employee
Pin
Authentication_System
PhoneNumber
Batch_ID
Image
Product_ID
belongs to
DOM
Batch
Type
Product
DOE (O)
Category
Quantity
Price
added_to
Brand
Name
Stock
Has
Cart_ID
Customer
(Address)
City
Cart
Cart_Status
(Cust_name)
H_No
State
Made of
F_name
L_name
Pin
Area
Code
Does
Has
Places
Applied on
Coupons
Discount
Payment
is_done_for
Order_Status
Orders
Order_Date
Payment_ID
Payment_type
Order_ID
Total_Cost
Delivery_Date
Receipt_ID
Receipt
is_made_for
Final_Cost
Payment_Date

DATABASE SCHEMA



DATABASE SCHEMA Table

TABLES	FIELDS	DATA TYPES	KEYS
User	PhoneNumberNOT NULL)	Varchar(30)	Primary Key, Foreign Key Phone_Number references from Authentication_System(PhoneNumber)
	User_IDNOT NULL)	Int	
	Login_TypeNOT	Varchar(30)	

	NULL)		

Authentication_System	Phone_Number(NOT NULL)
	
	Password(NOT NULL)
		VarChar(30)
		
		VarChar(200)
			Primary Key

Employee	PhoneNumber(NOT NULL)
	
	Emp_Name (NOT NULL)
	
	H_No(House Number) (NOT NULL)
	
	Area (NOT NULL)
	
	City (NOT NULL)
	
	State (NOT NULL)
	
	Pin (NOT NULL)
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(100)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
			Primary Key
			Foriegn Key(PhoneNumber) references from User(PhoneNumber)

Customer	PhoneNumber (NOT NULL)
	
	F_name(First Name) (NOT NULL)
	
	L_name(Last Name) (NOT NULL)
	
	H_No (NOT NULL)
	
	Area (NOT NULL)
	
	City (NOT NULL)
	
	State (NOT NULL)
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(100)
		
		VarChar(100)
		
		VarChar(50)
			Primary Key
			Foriegn Key(PhoneNumber) references from User(PhoneNumber)

	Pin (NOT NULL)	VarChar(30)	

Product	Product_ID (NOT NULL)
	
	Name (NOT NULL)
	
	Price (NOT NULL)
	
	Quantity (NOT NULL)
	
	Type (NOT NULL)
	
	Category (NOT NULL)
	
	Brand (NOT NULL)
	
	Image(NOT NULL)
	
	Batch_ID (NOT NULL)
		int
		
		VarChar(100)
		
		Float
		
		Int
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		Varchar(200)
		
		int
			Primary Key
			
			Foriegn Key(Batch_ID) references from Batch(Batch_ID)

Stock_Order	Order_Number (NOT NULL) Product_ID (NOT NULL) Quantity (NOT NULL) Date_Of_Stock_Order (NOT NULL) PhoneNumber (NOT NULL) Supplier_ID (NOT NULL)	int VarChar(30) INT DATE VarChar(30) int	Primary Key Foriegn Key(PhoneNumber) references from Employee(PhoneNumber) Foriegn

			Key(Supplier_ID) references from Supplier(Supplier_ID
Supplies	Supplier_ID (NOT NULL) Product_ID (NOT NULL) Date_Of_Supply (NOT NULL) Order_Number (NOT NULL)	int VarChar(30) DATE int	Primary Key Foriegn Key(Supplier_ID) references from Supplier(Supplier_ID) Primary Key Foriegn Key(Product_ID) references from Product(Product_ID) Foriegn Key(Order_Number) references from Stock_Order(Order_Number)

Supplier	Supplier_ID (NOT NULL)
	
	Supplier_Name (NOT NULL)
	
	Office_No (NOT NULL)
	
	Building (NOT NULL)
	
	Area (NOT NULL)
	
	State (NOT NULL)
	
	Pin (NOT NULL)
		int
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(30)
		
		VarChar(1000)
		
		VarChar(30)
			Primary Key

Batch	Batch_ID (NOT NULL)
	
	DOM(DATE Of Manufacturing) (NOT NULL)
	
	DOE(DATE of Expiry)
		int
		
		DATE
		
		DATE
			Primary Key

Orders	Order_ID (NOT NULL) Total_Cost (NOT NULL) Order_Status (NOT NULL) Delivery_Date (NOT NULL) DOO(DATE of Order) (NOT NULL) Cart_ID (NOT NULL) PhoneNumber (NOT NULL) Code	int FLOAT VARCHAR(30) DATE DATE int VarChar(30) VarChar(30)	Primary Key Foriegn Key(Cart_ID) references from Cart(Cart_ID) Primary Key Foriegn Key(PhoneNumber) references from User(PhoneNumber) Foriegn Key(Code) references from Coupons(Code)

Cart	Cart_ID (NOT NULL)
	
	PhoneNumber (NOT NULL)
	
	Cart_Status (NOT NULL)
		int
		
		VarChar(30)
		
		VARCHAR(30)
			Primary Key
			
			Primary Key
			Foriegn Key(PhoneNumber) references from User(PhoneNumber)

Coupons	Code (NOT NULL) Discount (NOT NULL)	VarChar(30) Float	Primary Key
added_to	Cart_ID (NOT NULL) Product_ID (NOT NULL) Quantity (NOT NULL)	int int Int	Primary Key Foriegn Key(Cart_ID) references from Cart(Cart_ID) Primary Key Foriegn Key(Product_ID) references from Product(Product_ID)
Payment	Payment_ID (NOT NULL) Payment_type (NOT NULL) PhoneNumber (NOT NULL) Order_ID (NOT NULL)	Int VarChar(30) VarChar(30) int	Primary Key Foriegn Key(PhoneNumber) references from Customer(PhoneNumber) Foriegn Key(Order_ID) references from Orders(Order_ID)
Receipt	Receipt_ID (NOT NULL) Payment_Date(NOT NULL) Final_Cost (NOT NULL) PhoneNumber (NOT NULL) Order_ID (NOT NULL)	int DATE Float VarChar(30) int	Primary Key Primary Key Foriegn Key(PhoneNumber) references from Customer(PhoneNumber)

			Primary Key Foriegn Key(Order_ID) references from Orders(Order_ID)

DATABASE CREATION QUERIES

```
CREATE TABLE Batch
(
  Batch_ID INT NOT NULL,
  DOM DATE NOT NULL,
  DOE DATE,
  PRIMARY KEY (Batch_ID)
);

CREATE TABLE Supplier
(
  Supplier_ID INT NOT NULL,
  Supplier_Name VARCHAR(50) NOT NULL,
  Office_No VARCHAR(50) NOT NULL,
  Building VARCHAR(100) NOT NULL,
  Area VARCHAR(100) NOT NULL,
  State VARCHAR(100) NOT NULL,
  Pin VARCHAR(20) NOT NULL,
  PRIMARY KEY (Supplier_ID)
);

CREATE TABLE Coupons
(
  Code VARCHAR(30) NOT NULL,
  Discount FLOAT NOT NULL,
  PRIMARY KEY (Code)
);

CREATE TABLE Product
(
  Product_ID INT NOT NULL,
  Price FLOAT NOT NULL,
  Stock INT NOT NULL,
  Type VARCHAR(30) NOT NULL,
  Category VARCHAR(30) NOT NULL,
  Brand VARCHAR(30) NOT NULL,
  Name VARCHAR(30) NOT NULL,
  Batch_ID INT NOT NULL,
  PRIMARY KEY (Product_ID),
  FOREIGN KEY (Batch_ID) REFERENCES Batch(Batch_ID)
);

CREATE TABLE Supplies
(
  Date_Of_Supply DATE NOT NULL,
  Supplier_ID INT NOT NULL,
  Product_ID INT NOT NULL,
  Order_ID INT NOT NULL,
  PRIMARY KEY (Supplier_ID, Product_ID),
  FOREIGN KEY (Supplier_ID) REFERENCES Supplier(Supplier_ID),
  FOREIGN KEY (Order_ID) REFERENCES Stock_Order(Order_ID),
  FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID)
);

CREATE TABLE Authentication_System
(
  Password VARCHAR(255) NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  PRIMARY KEY (PhoneNumber)
);

CREATE TABLE Payment
(
  Payment_ID INT NOT NULL,
  Payment_type VARCHAR(30) NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  Order_ID INT NOT NULL,
  PRIMARY KEY (Payment_ID),
  FOREIGN KEY (PhoneNumber) REFERENCES Customer(PhoneNumber),
  FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID)
);

CREATE TABLE Receipt
(
  Receipt_ID INT NOT NULL,
  Final_Cost FLOAT NOT NULL,
  Payment_Date DATE NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  Order_ID INT NOT NULL,
  PRIMARY KEY (Receipt_ID),
  FOREIGN KEY (PhoneNumber) REFERENCES Customer(PhoneNumber),
  FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID)
);

CREATE TABLE Orders
(
  Order_ID INT NOT NULL,
  Total_Cost FLOAT NOT NULL,
  Order_Status VARCHAR(30) NOT NULL,
  Delivery_Date DATE NOT NULL,
  Order_Date DATE NOT NULL,
  Cart_ID INT NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  Code VARCHAR(30),
  PRIMARY KEY (Order_ID),
  FOREIGN KEY (Cart_ID) REFERENCES Cart(Cart_ID),
  FOREIGN KEY (PhoneNumber) REFERENCES Customer(PhoneNumber),
  FOREIGN KEY (Code) REFERENCES Coupons(Code)
);

CREATE TABLE Cart
(
  Cart_ID INT NOT NULL,
  Cart_Status VARCHAR(30) NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  PRIMARY KEY (Cart_ID),
  FOREIGN KEY (PhoneNumber) REFERENCES Customer(PhoneNumber)
);
```

```
CREATE TABLE Stock_Order
(
  Order_Number INT NOT NULL,
  Product_ID INT NOT NULL,
  Quantity INT NOT NULL,
  Date_Of_Stock_Order DATE NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  Supplier_ID INT NOT NULL,
  PRIMARY KEY (Order_Number),
  FOREIGN KEY (PhoneNumber) REFERENCES Employee(PhoneNumber),
  FOREIGN KEY (Supplier_ID) REFERENCES Supplier(Supplier_ID)
);

CREATE TABLE User
(
  Login_type VARCHAR(30) NOT NULL,
  User_ID INT NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  PRIMARY KEY (PhoneNumber),
  FOREIGN KEY (PhoneNumber) REFERENCES Authentication_System(PhoneNumber)
);

CREATE TABLE Employee
(
  Emp_Name VARCHAR(30) NOT NULL,
  H_No VARCHAR(30) NOT NULL,
  Area VARCHAR(30) NOT NULL,
  State VARCHAR(30) NOT NULL,
  Pin VARCHAR(30) NOT NULL,
  City VARCHAR(30) NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  PRIMARY KEY (PhoneNumber),
  FOREIGN KEY (PhoneNumber) REFERENCES User(PhoneNumber)
);

CREATE TABLE Customer
(
  F_name VARCHAR(30) NOT NULL,
  L_name VARCHAR(30),
  H_No VARCHAR(30) NOT NULL,
  Area VARCHAR(100) NOT NULL,
  State VARCHAR(50) NOT NULL,
  Pin VARCHAR(20) NOT NULL,
  City VARCHAR(50) NOT NULL,
  PhoneNumber VARCHAR(30) NOT NULL,
  PRIMARY KEY (PhoneNumber),
  FOREIGN KEY (PhoneNumber) REFERENCES User(PhoneNumber)
);

CREATE TABLE added_to
(
  Quantity INT NOT NULL,
  Product_ID INT NOT NULL,
  Cart_ID INT NOT NULL,
  PRIMARY KEY (Product_ID, Cart_ID),
  FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID),
  FOREIGN KEY (Cart_ID) REFERENCES Cart(Cart_ID)
);
```

VIEWS

```
CREATE VIEW VWPAB
AS
SELECT *
FROM PRODUCT  natural JOIN BATCH ;


CREATE VIEW  VWATP
AS
SELECT *
from ADDED_TO NATURAL JOIN PRODUCT;

CREATE VIEW  VWRECEIPT
AS
SELECT *
from
Product natural join added_to natural joinorders natural join coupons;
```

GRANTS

Implemented Grants based on Type of User Logged In
Created 2 Database Users ‘cust1’ and ‘emp1’ for give appropriate permissions to customer and employee respectively, created the ‘tempu’ user (Temporary user) to facilitate the login process.

```
use logintest;
grant select on product to 'cust1'@'localhost';
grant update on product to 'cust1'@'localhost';
grant select on authentication_system to 'cust1'@'localhost';
grant insert on authentication_system to 'cust1'@'localhost';
grant insert , select on user to 'cust1'@'localhost';
grant insert,select on Customer to 'cust1'@'localhost';
grant select on batch to 'cust1'@'localhost';
grant insert,select , update on Cart to 'cust1'@'localhost';
grant insert,select , update  on added_to to 'cust1'@'localhost';
grant insert,select , update  on orders to 'cust1'@'localhost';
grant select on coupons to 'cust1'@'localhost';
grant insert,select   on payment to 'cust1'@'localhost';
grant insert,select   on receipt to 'cust1'@'localhost';
grant insert,select , update   on vwpab to 'cust1'@'localhost';
grant insert,select , update   on vwatp to 'cust1'@'localhost';
```

show grants for 'cust1'@'localhost';

grant insert , select on authentication_system to 'emp1'@'localhost';
grant insert , select on user to 'emp1'@'localhost';
grant insert , select , update on Customer to 'emp1'@'localhost';
grant insert , select on employee to 'emp1'@'localhost';
grant insert , select , update , delete on product to 'emp1'@'localhost';
grant insert , select , update , delete on batch to 'emp1'@'localhost';
grant select on cart to 'emp1'@'localhost';
grant select on added_to to 'emp1'@'localhost';
grant select on orders to 'emp1'@'localhost';
grant insert , select , update , delete on coupons to 'emp1'@'localhost';
grant select on payment to 'emp1'@'localhost';
grant select on receipt to 'emp1'@'localhost';
grant insert , select , update , delete on stock_order to 'emp1'@'localhost';
grant insert , select , update , delete on supplier to 'emp1'@'localhost';
grant insert , select , update , delete on supplies to 'emp1'@'localhost';
grant insert , select , update , delete on vwpab to 'emp1'@'localhost';
grant insert , select , update , delete on vwatp to 'emp1'@'localhost';

show grants for 'emp1'@'localhost';

grant insert , select on authentication_system to 'tempu'@'localhost';
grant insert , select on user to 'tempu'@'localhost';
grant insert , select on Customer to 'tempu'@'localhost';

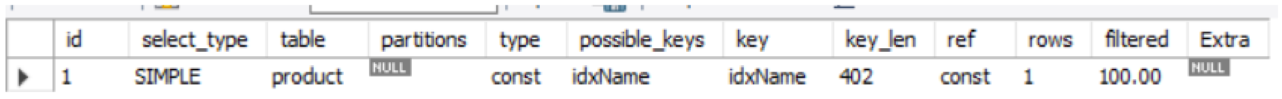
show grants for 'tempu'@'localhost';

SQL QUERIES (OPTIMIZED)

use logintest;

#1.Select the Supplier who send the product
Select Supplier_Name from Supplier Natural Join (Select Supplier_ID from Supplies where Product_Id = 3) AS S;

#2. Finding the stock of Product when searched using name (using indexing)
CREATE UNIQUE INDEX idxName ON Product (Name);
select Stock from product where name = 'Red Local Carrot- 1 kg' ;
explain select Stock from product where name = 'Red Local Carrot- 1 kg' ; #searches only one row



id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	product	NULL	const	idxName	idxName	402	const	1	100.00	NULL

Only one row is checked so the query is optimised

#3 . Item in most demand (item in most carts) most famous
Select name , count(Product_ID) as counter from vwatp group by product_ID order by counter desc Limit 1;

#4.
#All the customers who ordered on a particular date
Select F_name , L_name , PhoneNumber from Customer Natural Join (Select PhoneNumber from Receipt where Payment_Date = '2022-04-27') AS R;

#5. Supplier who Supplied the biggest collection of products at a time
Select Supplier_Name from Supplier Where Supplier_ID = (Select Supplier_ID from Supplies Where Order_Number = (Select Order_Number from Stock_Order where Quantity = (Select MAX(Quantity) from Stock_Order)   ));

#6. Customers who bought a particular product
SELECT F_name , L_name , PhoneNumber from Customer where PhoneNumber IN (Select PhoneNumber From cart natural join (Select cart_id from added_to where product_id = 58) AS C) ;

#7. FIND NAME AND QUANTITY OF EXPIRED PORDUCTS
Select Name , STOCK , DOE from vwpab where DOE< CURDATE() ;

#8. Most used Payment Type
SELECT payment_Type , Count(Payment_type) as counter from payment group by Payment_Type order by Counter desc ;

#9. Average sale of Last 7 days
SELECT AVG(FINAL_COST) as Average_Sale from receipt where Payment_date <= curdate() And Payment_date > Curdate() - 7;

#10. name of User with maximum purchase in a month
Select F_name , L_name, PhoneNumber from Customer where PhoneNumber in (Select PhoneNumber from receipt where Final_Cost = (Select Max(Final_Cost) from receipt)) ;

INDEXING

Appropriate attributes for indexing identified

1. Primary Key of All Tables (Clustering Index - Primary key) (For example - Product_ID of Product)
2. Name of the product - Secondary Index (Non Clustering) (Used in SQL queries also)
3. Order_Date of Orders-   Clustering Index (Non Primary key)
4. Payment_Date of Receipt - Clustering Index (Non Primary key)
5. DOE of Batch - Clustering Index (Non Primary key)
6. Category of product - Clustering Index (Non Primary key)
7. Brand of product - Clustering Index (Non Primary key)
8. Type of product - Clustering Index (Non Primary key)
9. State of Customer(address) - Clustering Index (Non Primary key)
10. Area of Supplier(address) - Clustering Index (Non Primary key)

TRIGGERS

IMPLEMENTED TRIGGERS ON RECEIPT AFTER_INSERT

```
CREATE DEFINER=`root`@`localhost` TRIGGER `receipt_AFTER_INSERT` AFTER INSERT ON
`receipt` FOR EACH ROW BEGIN
UPDATE orders
set order_status = 'paid'
where order_id = NEW.order_id;

UPDATE cart
set cart_status = 'ordered'
where PhoneNumber = NEW.PhoneNumber and ( SELECT MAX(cart_id) ) ;

 insert into cart(PhoneNumber, cart_status) values ( NEW.PhoneNumber , 'current');

END
```

FUTURE SCALABILITY

- Add Delivery charges
- Add Premium membership, (free, fast delivery)
- Recommendations for Users based on their search
- Reviews on Products
- Saving Credit card / Net-banking details for future payments
- Address Change option
- Forgot Password option

ASSUMPTIONS

- All products are delivered on the same day
- All Products of a cart are delivered together
- An Employee can only restock the old products not add new products.
- An Employee cannot order products from his work phone number. (Office Number)

GENERAL FLOW OF THE WEBSITE

First the Customer has to create an account on the website using his Phone Number and has to give details like his address. Then he has to Login into the website, Then he can browse and add products to his cart. On the Cart page the Customer can order items present in his cart and apply

Coupon code to his order. The Customer will then has to select his payment method (Net-Banking/COD/Card/E-Wallet) then the customer order will be confirmed and he will get the invoice for his order. And then he can select other products for the next order or he can select logout from navigation bar to end his sessions and logout.

For the employee part , the employee has to first login using his employee ID and password , he can view sale of the day , order new stock of the products , check product details and stock , check the list of suppliers and run the SQL Queries.

RESPONSIBILITIES OF EACH MEMBER

ASHWIN SHEORAN, 2020288

a. ER-Diagram
   i. Made the various entities of ER Diagram like Users, Customer, Employee, Cart, Products, Employees, Suppliers, Orders, Receipt
   ii. Attributes and relationships between the above entities
   iii. Creating database and Tables in SQL workspace

b. Relational Schema
   i. Made the relational Schema of the above database

c. Views
   i. Created the required Views for the working of the Programs

d. Database Connection and Grants
   I. Establishing database connection between PHP and MySQL server
   ii. Created the Different database users for providing grants

e. SQL Queries and their Optimization
   i. Written all the 10 submitted SQL queries
   ii. Optimization of the 10 SQL queries

f. Embedded SQL queries (PL/SQL)
   i. Written and implemented the (PL/SQL) queries used in the working of the Program

g. Indexing
   i. Identify the attribute(s) to create Index tables required for

Queries.

h. Triggers
    i. Written and Implemented the Triggers in SQL

i. Backend
    i. Created the Backend in PHP for the Main page (Product Page)
    ii.Login (Using Session Storage) and <u>Password Decryption</u>
    iii. Sign Up and <u>Password Hashing</u>
    Iv. Logout (Using Session Storage)
    v. Cart
    v. Orders
    vi. Receipt
    vii. Invoice
    viii. Employee Login
    ix. Employee front Page
    x. Check Product Stock
    xi. Product List
    xii. Add Stock
    xiii. Supplier List
    xiv. Pages of all 10 SQL queries
    xv. Navigation Bar for Customer
    xvi. Navigation Bar for Employee

j. Front End
    i. Created the Front End in PHP for Main page (Product Page)
    ii.Login Page
    iii. Sign Up Page
    iv. Cart Page
    v. Orders page
    vi. Receipt Page
    vii. Invoice Page
    viii. Paytm Page
    ix. Card page
    x. Net-Banking Page
    xi. Cash on Delivery page
    xii. Employee Login Page
    xiii. Employee FrontPage
    xiv. Check Product Stock Page

xv. Product List Page
xvi. Add Stock Page
xvii. Supplier List page
xviii. Pages of all 10 SQL queries
xix. Navigation Bar for Customer
xx. Navigation Bar for Employee

HARSH GOYAL, 2020562

a. ER-Diagram
    i. Made the payments part of ER diagram
b. Views
    i. created views for the working of the program
c. Embedded SQL
    i. Created the embedded SQL query for receipt page
d. Indexing
    i. Identified attributes for Indexing
e. Backend
    i. Created Backend for the receipt page (Invoice) opening of page in new tab.
f. Frontend
    i. Created the frontend of invoice

HARSHIT GARG, 2020301

a. Relational Schema
    i. Identification of data types and relationships
b. ER Diagram
    i. Creation of Relationships in ER Diagram
c. Embedded SQL Query
d. Data Population
    i. Data Population of Batch, Coupons

MEGHNA, 2020080

a. Data Population
    i. Data Population of Suppliers information, Supplies, Stock orders.
    ii. Provide the assets for the products on the website.
    iii. Create the queries for products and their assets.