DATABASE SCEMA-

* Customer (customer\_id, name, email, phone\_number, address\_id )
* Product (product\_id,product\_name, stock ,category\_id,MRP,Brand)
* Orders (order\_id,shipping\_date , cart\_id , order\_amount ,order\_date )
* OrderItem (order\_item\_id,order\_id , product\_id, quantity , unit\_price , total\_price )
* payment ( customer\_id,order\_id , payment\_id ,payment\_mode ,payment\_date )
* payment\_history (payment\_history\_id ,payment\_id ,data ,status , transaction\_id )
* Admins ( admin\_id, username, password, email)
* cart ( cart\_id ,total\_item ,grand\_total, customer\_id )
* Supplier ( name, email ,phone\_number, address)
* review (review\_id,description,ratings ,product\_id , customer\_id )
* wishlist (wishlist\_id, customer\_id,product\_id, added\_at)
* shipment ( shipment\_id ,arrival\_date ,estimate\_date ,shipment\_date , order\_id )
* category (category\_id, category\_name, description )

To normalize the given database to 3rd Normal Form 93NF), we need to eliminate any transitive dependencies and ensure that each non-prime attribute is fully functionally dependent on the primary key.

First, lets identify the functional dependencies-

1. Address Table (to store customer addresses)
2. Customers Table (with a foreign key referencing the Addresses table):
3. Product Table (unchanged):
4. Orders Table (unchanged):
5. OrderItem Table (unchanged):
6. PaymentHistory Table (with a foreign key referencing the Payment table):
7. Admins Table (unchanged):
8. Cart Table (with a foreign key referencing the Customers table):
9. Supplier Table (unchanged):
10. Review Table (with foreign keys referencing the Product and Customers tables):
11. Wishlist Table (with foreign keys referencing the Product and Customers tables):
12. Shipment Table (with a foreign key referencing the Orders table):
13. Category Table (unchanged):

Now, let's normalize the database:

**First Normal Form (1NF):**

- Ensure that each attribute contains atomic values.

- There are no multivalued attributes in the given schema, so it's already in 1NF.

**Second Normal Form (2NF):**

- Ensure that there are no partial dependencies.

All the tables seem to have single attribute primary keys, so there are no partial dependencies. Therefore, it's already in 2NF.

**Third Normal Form (3NF):**

- Ensure that there are no transitive dependencies.

Tables after Normalization-

1. CREATE TABLE Addresses (

address\_id INT PRIMARY KEY,

street VARCHAR(255),

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(20),

country VARCHAR(100)

);

1. CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(255),

phone\_number VARCHAR(20),

address\_id INT,

FOREIGN KEY (address\_id) REFERENCES Addresses(address\_id)

);

1. CREATE TABLE Product (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(255),

stock INT,

category\_id INT,

MRP DECIMAL(10, 2),

Brand VARCHAR(100)

);

1. CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

shipping\_date DATE,

cart\_id INT,

order\_amount DECIMAL(10, 2),

order\_date DATE

);

1. CREATE TABLE OrderItem (

order\_item\_id INT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

unit\_price DECIMAL(10, 2),

total\_price DECIMAL(10, 2),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

1. Payment Table (with foreign keys referencing Customers and Orders tables):

CREATE TABLE Payment (

payment\_id INT PRIMARY KEY,

customer\_id INT,

order\_id INT,

payment\_mode VARCHAR(50),

payment\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

1. CREATE TABLE PaymentHistory (

payment\_history\_id INT PRIMARY KEY,

payment\_id INT,

data VARCHAR(255),

status VARCHAR(50),

transaction\_id VARCHAR(100),

FOREIGN KEY (payment\_id) REFERENCES Payment(payment\_id)

);

1. CREATE TABLE Admins (

admin\_id INT PRIMARY KEY,

username VARCHAR(50),

password VARCHAR(255),

email VARCHAR(255)

);

1. CREATE TABLE Cart (

cart\_id INT PRIMARY KEY,

total\_item INT,

grand\_total DECIMAL(10, 2),

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

1. CREATE TABLE Supplier (

supplier\_id INT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(255),

phone\_number VARCHAR(20),

address VARCHAR(255)

);

1. CREATE TABLE Review (

review\_id INT PRIMARY KEY,

description VARCHAR(255),

ratings DECIMAL(3, 2),

product\_id INT,

customer\_id INT,

FOREIGN KEY (product\_id) REFERENCES Product(product\_id),

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)

);

1. CREATE TABLE Wishlist (

wishlist\_id INT PRIMARY KEY,

customer\_id INT,

product\_id INT,

added\_at TIMESTAMP,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),

FOREIGN KEY (product\_id) REFERENCES Product(product\_id)

);

CREATE TABLE Shipment (

shipment\_id INT PRIMARY KEY,

arrival\_date DATE,

estimate\_date DATE,

shipment\_date DATE,

order\_id INT,

FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)

);

CREATE TABLE Category (

category\_id INT PRIMARY KEY,

category\_name VARCHAR(50),

description VARCHAR(255)

);