DBMS Project

Ecommerce delivery system

Main objective of the project:

Objective of the project is to manage details of sellers, buyers, delivery options, payments. It manages all the information about products, payment, the purpose to built this project is to fullfillthe needs of customer, to make process easy between sellers and buyers.

An online selling and delivery database is a structured collection of data that stores information related to products, customers, orders, and deliveries for an e-commerce platform. It typically includes the following tables:

1. *Product Table*: This table contains information about the products available for sale. It may include fields such as product ID, name, description, price, quantity in stock, and category.

2. *Customer Table*: This table stores details about customers who use the platform. Fields may include customer ID, name, contact information, shipping address, and payment details.

3. *Order Table*: The order table tracks the orders placed by customers. It includes fields like order ID, customer ID, order date, and order status (e.g., processing, shipped, delivered).

4. *Order Items Table*: This table is linked to the order table and lists the specific items included in each order. It typically includes fields like order item ID, order ID, product ID, quantity, and total price.

5. *Delivery Table*: Information about the delivery process is stored here. Fields may include delivery ID, order ID (to link to specific orders), delivery status (e.g., in transit, delivered), delivery date, and delivery address.

6. *Payment Table*: This table records payment transactions. It includes fields such as payment ID, order ID (to link to specific orders), payment date, payment method, and transaction amount.

7. *User Table*: If there are user accounts for the platform, this table stores user credentials (username, password) and their roles (e.g., customer, admin).

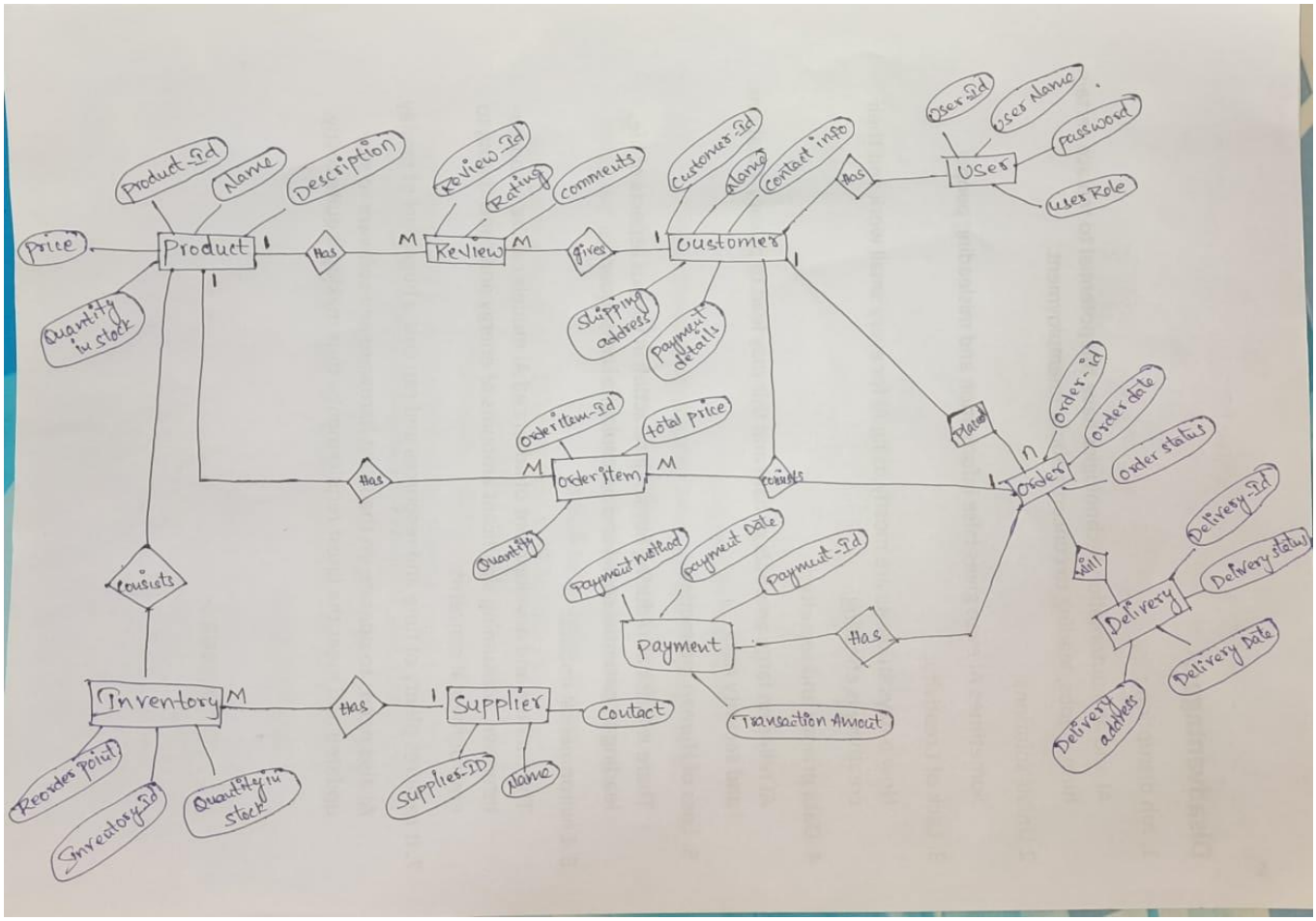
8. *Reviews and Ratings Table*: If customers can leave reviews and ratings, this table stores that data. Fields can include review ID, product ID, customer ID, rating, and comments.

9. *Inventory Table*: To manage product stock levels, this table includes information about the

quantity of each product in stock, reorder points, and supplier details.

10. *Supplier Table*: If you work with multiple suppliers, this table can store information about them, including supplier ID, name, contact details, and product catalog they provide.

These tables are interconnected through relationships, typically using unique identifiers (like IDs) to link records together. The database allows for efficient storage, retrieval, and management of data, supporting the functionality of an online selling and delivery platform

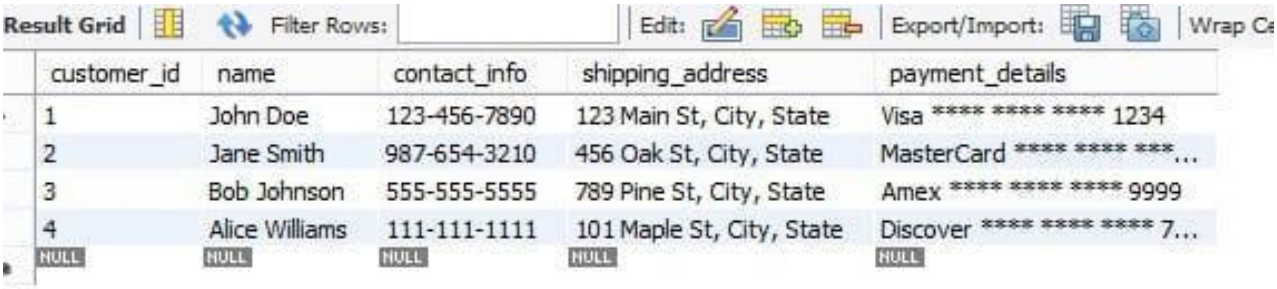


```
CREATE DATABASE Ecommerce_delivery_db;

USE Ecommerce_delivery_db;

CREATE TABLE Product (
  product_id INT PRIMARY KEY,
  name VARCHAR(255),
  description TEXT,
  price DECIMAL(10,2),
  quantity INT,
  category VARCHAR(255) );

INSERT INTO Product (product_id, name, description, price, quantity, category)
VALUES
  (1, 'Product 1', 'Description for Product 1', 10.00, 100, 'Category A'),
  (2, 'Product 2', 'Description for Product 2', 15.50, 50, 'Category B'),
  (3, 'Product 3', 'Description for Product 3', 8.99, 200, 'Category A'),
  (4, 'Product 4', 'Description for Product 4', 12.25, 75, 'Category B'),
  (5, 'Product 5', 'Description for Product 5', 9.75, 150, 'Category A'),
  (6, 'Product 6', 'Description for Product 6', 19.99, 80, 'Category C');
```

product_id	name	description	price	quantity	category
1	Product 1	Description for Product 1	10.00	100	Category A
2	Product 2	Description for Product 2	15.50	50	Category B
3	Product 3	Description for Product 3	8.99	200	Category A
4	Product 4	Description for Product 4	12.25	75	Category B
5	Product 5	Description for Product 5	9.75	150	Category A
6	Product 6	Description for Product 6	19.99	80	Category C
NULL	NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Customer (
  customer_id INT PRIMARY KEY,
  name VARCHAR(255),
```

```
  contact_info VARCHAR(255),
  shipping_address TEXT,
  payment_details VARCHAR(255) );
INSERT INTO Customer (customer_id, name, contact_info, shipping_address,
payment_details)
VALUES
  (1, 'John Doe', '123-456-7890', '123 Main St, City, State', 'Visa **** **** **** 1234'),
  (2, 'Jane Smith', '987-654-3210', '456 Oak St, City, State', 'MasterCard **** **** ****
5678'),
  (3, 'Bob Johnson', '555-555-5555', '789 Pine St, City, State', 'Amex **** **** **** 9999'),
  (4, 'Alice Williams', '111-111-1111', '101 Maple St, City, State', 'Discover **** **** ****
7777');
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

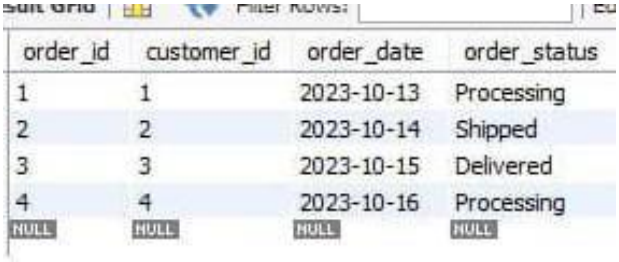
customer_id	name	contact_info	shipping_address	payment_details
1	John Doe	123-456-7890	123 Main St, City, State	Visa **** **** **** 1234
2	Jane Smith	987-654-3210	456 Oak St, City, State	MasterCard **** **** ***...
3	Bob Johnson	555-555-5555	789 Pine St, City, State	Amex **** **** **** 9999
4	Alice Williams	111-111-1111	101 Maple St, City, State	Discover **** **** **** 7...
NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE `Order` (
  order_id INT PRIMARY KEY,
  customer_id INT,
  order_date DATE,
  order_status VARCHAR(50),
  FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) );
INSERT INTO `Order` (order_id, customer_id, order_date, order_status)
VALUES
  (1, 1, '2023-10-13', 'Processing'),
```

(2, 2, '2023-10-14', 'Shipped'),

(3, 3, '2023-10-15', 'Delivered'),

(4, 4, '2023-10-16', 'Processing');



order_id	customer_id	order_date	order_status
1	1	2023-10-13	Processing
2	2	2023-10-14	Shipped
3	3	2023-10-15	Delivered
4	4	2023-10-16	Processing
NULL	NULL	NULL	NULL

```
CREATE TABLE OrderItems (
  order_item_id INT PRIMARY KEY,
  order_id INT,
  product_id INT,
  quantity INT,
  total_price DECIMAL(10,2),
  FOREIGN KEY (order_id) REFERENCES `Order`(order_id),
  FOREIGN KEY (product_id) REFERENCES Product(product_id) );
INSERT INTO OrderItems (order_item_id, order_id, product_id, quantity, total_price)
VALUES
  (1, 1, 1, 2, 20.00),
  (2, 1, 2, 1, 15.50),
  (3, 2, 3, 5, 44.95),
  (4, 3, 4, 3, 36.75),
  (5, 4, 5, 4, 39.00),
  (6, 4, 6, 2, 39.98);
```

order_item_id	order_id	product_id	quantity	total_price
1	1	1	2	20.00
2	1	2	1	15.50
3	2	3	5	44.95
4	3	4	3	36.75
5	4	5	4	39.00
6	4	6	2	39.98
NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Delivery (
  delivery_id INT PRIMARY KEY,
  order_id INT,
  delivery_status VARCHAR(50),
  delivery_date DATE,
  delivery_address TEXT,
  FOREIGN KEY (order_id) REFERENCES `Order`(order_id));
INSERT INTO Delivery (delivery_id, order_id, delivery_status, delivery_date,
delivery_address)
VALUES
  (1, 1, 'In Transit', '2023-10-14', '123 Main St, City, State'),
  (2, 2, 'Delivered', '2023-10-15', '456 Oak St, City, State'),
  (3, 3, 'Delivered', '2023-10-16', '789 Pine St, City, State'),
  (4, 4, 'Processing', '2023-10-17', '101 Maple St, City, State');
```

Result Grid | Filter Rows: | Edit: | Export/Import:

delivery_id	order_id	delivery_status	delivery_date	delivery_address
1	1	In Transit	2023-10-14	123 Main St, City, State
2	2	Delivered	2023-10-15	456 Oak St, City, State
3	3	Delivered	2023-10-16	789 Pine St, City, State
4	4	Processing	2023-10-17	101 Maple St, City, State
NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Payment (
  payment_id INT PRIMARY KEY,
```

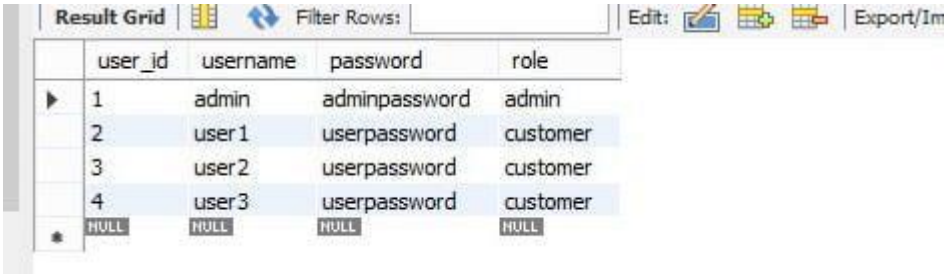
```
  order_id INT,
  payment_date DATE,
  payment_method VARCHAR(50),
  transaction_amount DECIMAL(10,2),
  FOREIGN KEY (order_id) REFERENCES `Order`(order_id));
INSERT INTO Payment (payment_id, order_id, payment_date, payment_method,
transaction_amount)
VALUES
  (1, 1, '2023-10-14', 'Visa', 35.50),
  (2, 2, '2023-10-15', 'MasterCard', 44.95),
  (3, 3, '2023-10-16', 'Amex', 36.75),
  (4, 4, '2023-10-17', 'Discover', 39.00);
```

payment_id	order_id	payment_date	payment_method	transaction_amount
1	1	2023-10-14	Visa	35.50
2	2	2023-10-15	MasterCard	44.95
3	3	2023-10-16	Amex	36.75
4	4	2023-10-17	Discover	39.00
NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE User (
  user_id INT PRIMARY KEY,
  username VARCHAR(50),
  password VARCHAR(255),
  role VARCHAR(50));
INSERT INTO User (user_id, username, password, role)
VALUES
  (1, 'admin', 'adminpassword', 'admin'),
  (2, 'user1', 'userpassword', 'customer'),
```

(3, 'user2', 'userpassword', 'customer'),

(4, 'user3', 'userpassword', 'customer');



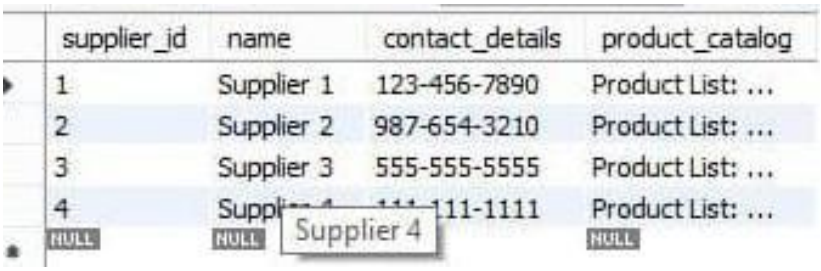
user_id	username	password	role
1	admin	adminpassword	admin
2	user1	userpassword	customer
3	user2	userpassword	customer
4	user3	userpassword	customer
NULL	NULL	NULL	NULL

```
CREATE TABLE ReviewsRatings (
  review_id INT PRIMARY KEY,
  product_id INT,
  customer_id INT,
  rating INT,
  comments TEXT,
  FOREIGN KEY (product_id) REFERENCES Product(product_id),
  FOREIGN KEY (customer_id) REFERENCES Customer(customer_id) );
INSERT INTO ReviewsRatings (review_id, product_id, customer_id, rating, comments)
VALUES
  (1, 1, 1, 4, 'Great product!'),
  (2, 2, 2, 5, 'Excellent quality.'),
  (3, 3, 3, 3, 'Good value for money.'),
  (4, 4, 1, 4, 'Impressed with the product.');
```

review_id	product_id	customer_id	rating	comments
1	1	1	4	Great product!
2	2	2	5	Excellent quality.
3	3	3	3	Good value for money.
4	4	1	4	Impressed with the product.
NULL	NULL	NULL	NULL	NULL

```
CREATE TABLE Supplier (
```

```
  supplier_id INT PRIMARY KEY,
  name VARCHAR(255),
  contact_details VARCHAR(255),
  product_catalog TEXT );
INSERT INTO Supplier (supplier_id, name, contact_details, product_catalog)
VALUES
  (1, 'Supplier 1', '123-456-7890', 'Product List: ...'),
  (2, 'Supplier 2', '987-654-3210', 'Product List: ...'),
  (3, 'Supplier 3', '555-555-5555', 'Product List: ...'),
  (4, 'Supplier 4', '111-111-1111', 'Product List: ...');
```



supplier_id	name	contact_details	product_catalog
1	Supplier 1	123-456-7890	Product List: ...
2	Supplier 2	987-654-3210	Product List: ...
3	Supplier 3	555-555-5555	Product List: ...
4	Supplier 4	111-111-1111	Product List: ...
NULL	NULL	NULL	NULL

```
CREATE TABLE Inventory (
  product_id INT PRIMARY KEY,
  quantity INT,
  reorder_point INT,
  supplier_id INT,
  FOREIGN KEY (product_id) REFERENCES Product(product_id),
  FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id));
INSERT INTO Inventory (product_id, quantity, reorder_point, supplier_id)
VALUES
  (1, 100, 10, 1),
  (2, 50, 5, 2),
```

(3, 200, 20, 1),

(4, 75, 10, 2),

(5, 150, 15, 1),

(6, 80, 8, 2);

product_id	quantity	reorder_point	supplier_id
1	100	10	1
2	50	5	2
3	200	20	1
4	75	10	2
5	150	15	1
6	80	8	2
NULL	NULL	NULL	NULL

Team Members:

Name: Kavyanjani

Scholar no: 2211201346

Roll no: 46

Name: Padmasri

Scholar no: 2211201364

Roll no:64

Name: Sai Sarayu

Scholar no: 2211201326

Roll no: 26