**Content**

**Page No:**

1. Introduction ----------------------------------------------------------------- 2
2. Project Proposal ------------------------------------------------------------------ 2
3. Class Diagram ------------------------------------------------------------------- 3
4. Use Case Diagram ----------------------------------------------------------------- 4
5. Activity Diagram ------------------------------------------------------------------ 5-7
6. User Interface ------------------------------------------------------------------ 8-9
7. Scenario Description -------------------------------------------------------------- 10
8. ER Diagram ------------------------------------------------------------------ 10
9. Normalization ------------------------------------------------------------------ 11
10. Schema Diagram ------------------------------------------------------------------ 12
11. Table Creation ------------------------------------------------------------------ 12-16
12. Data Insertion ------------------------------------------------------------------ 17-20
13. Query Writing(SQL) ---------------------------------------------------------------- 20-25
14. Relational Algebra ---------------------------------------------------------------- 26
15. Conclusion ------------------------------------------------------------------ 26

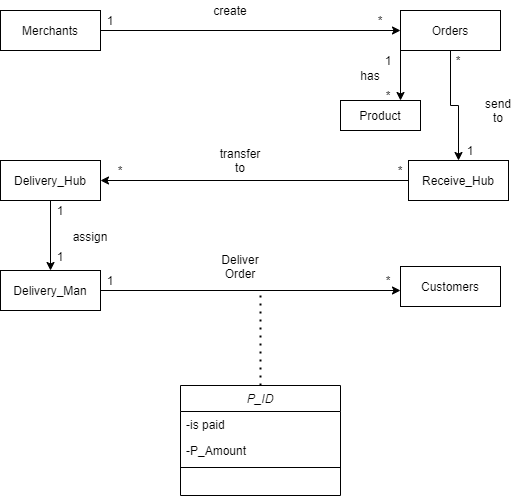
**Introduction:**

In today's fast-paced world, efficient delivery of products is essential for businesses to grow. Our system aims to provide a seamless experience for everyone involved in the courier process. The primary objective of our Courier Management System is to facilitate the smooth movement of goods from the merchant to the customer, ensuring timely and secure deliveries. With our system in place, merchants can easily place orders for their products, which will then be processed and dispatched through a well-structured workflow.

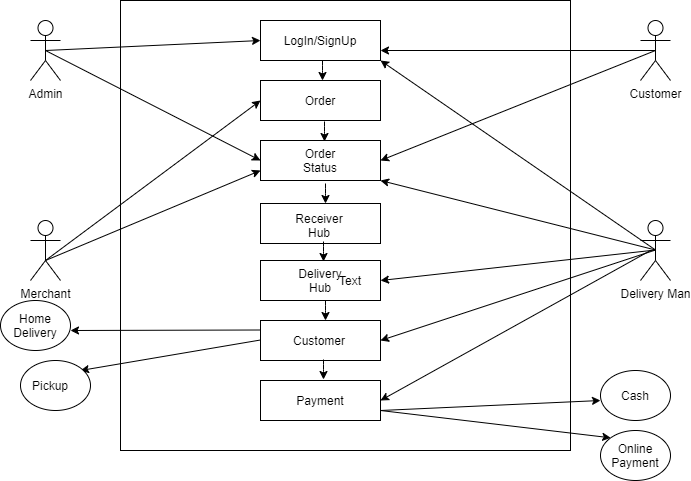
**Project Proposal:**

Our system will allow merchants to effortlessly place their products, which are then routed to a receiver hub. From there, a dedicated delivery hub takes charge, assigning a delivery person to ensure efficient order fulfillment. With our system, merchants can effortlessly send products to customers, creating a smooth and reliable delivery experience for everyone involved. The journey of a product begins when a merchant initiates an order. The system captures essential information such as Merchant Name, and Merchant Contact. This data allows for effective communication and coordination throughout the delivery process. Once an order is placed, it is assigned a unique Order ID and associated with the order location. The system tracks the order as it moves from the merchant to the receiver hub, where it awaits further processing. At the receiver hub, crucial details such as the receiver's name, Receiver Location, and expected delivery time are recorded. After the order has reached the receiver hub, a dedicated delivery man is assigned to handle the order. The delivery man is responsible for transporting the order from the receiver hub to the customer's location. The delivery location and estimated delivery time are captured to ensure accurate and efficient delivery.

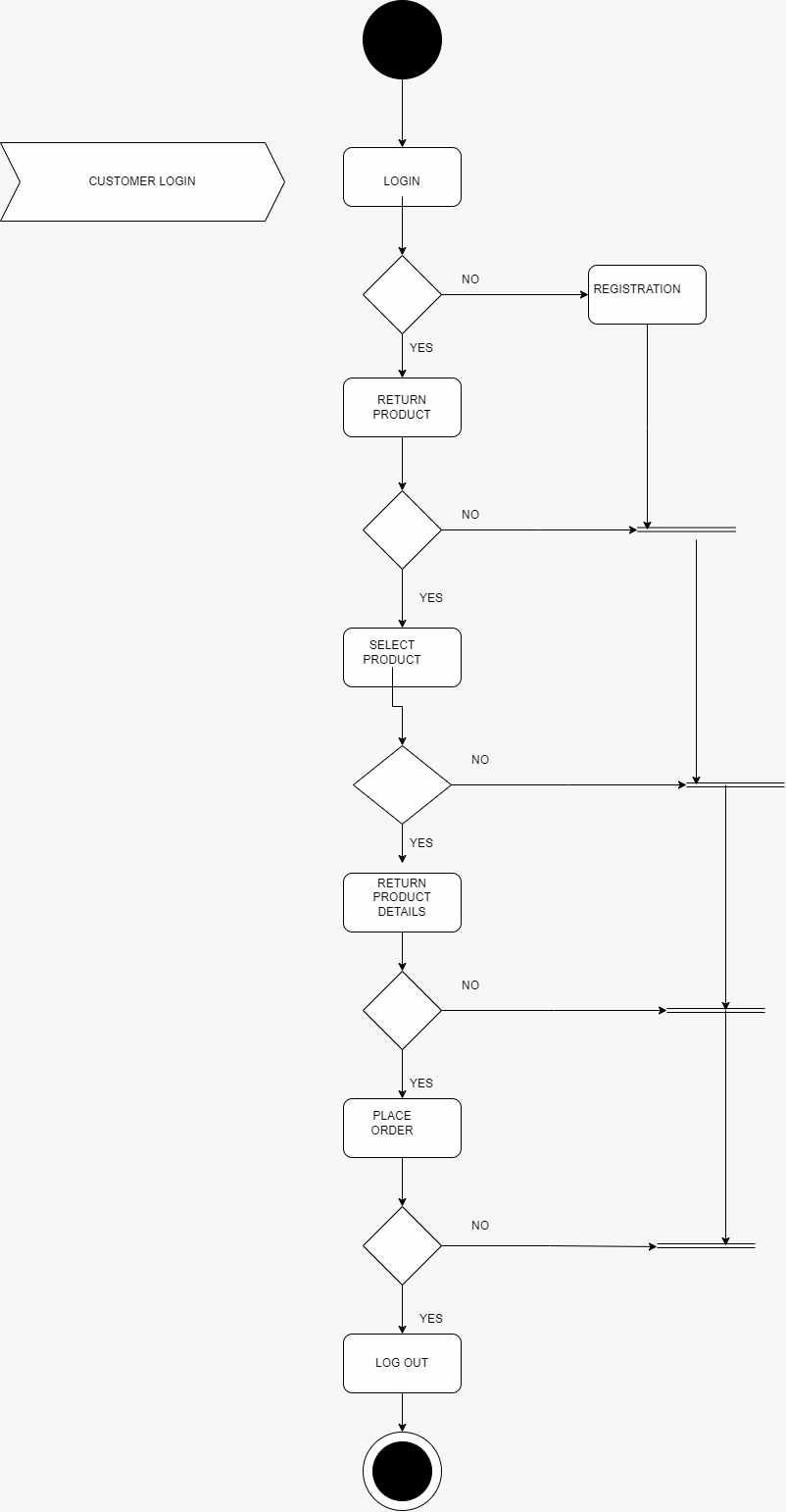
**Class Diagram:**

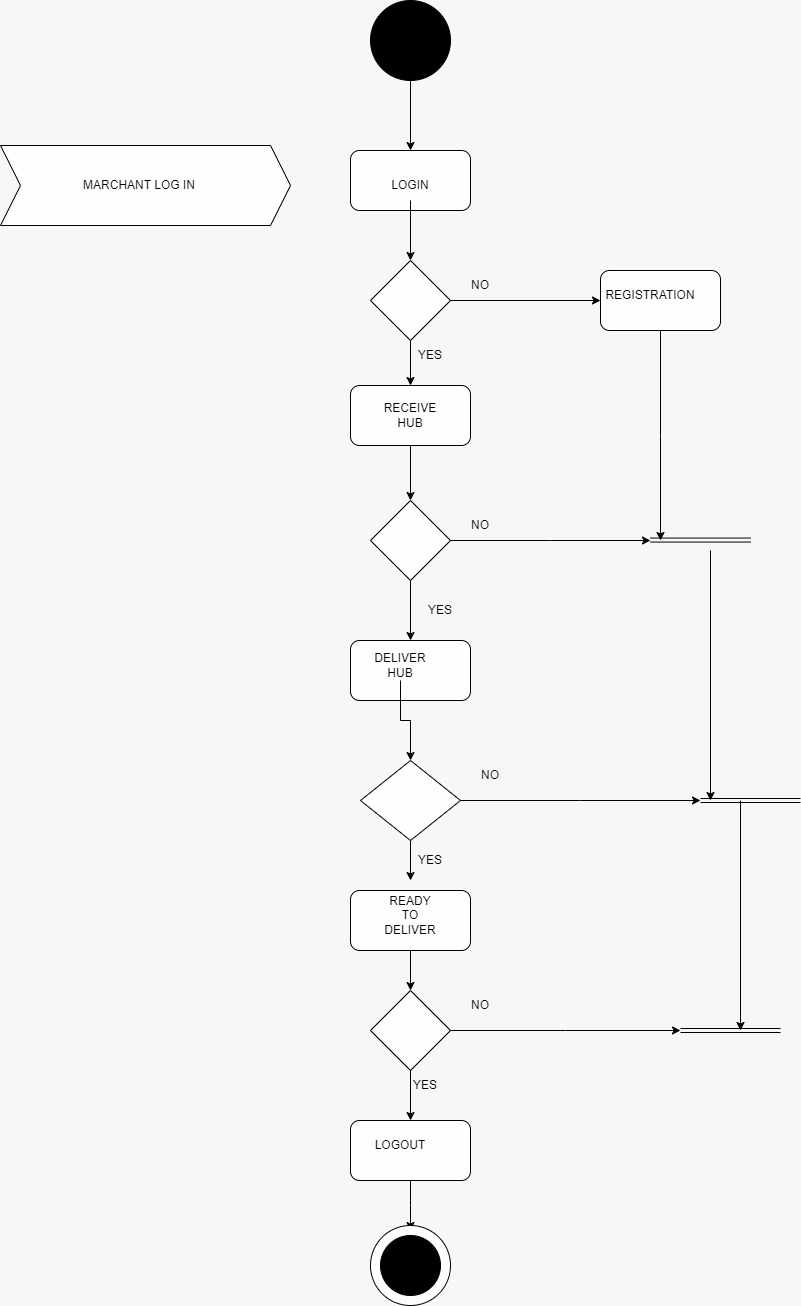
****

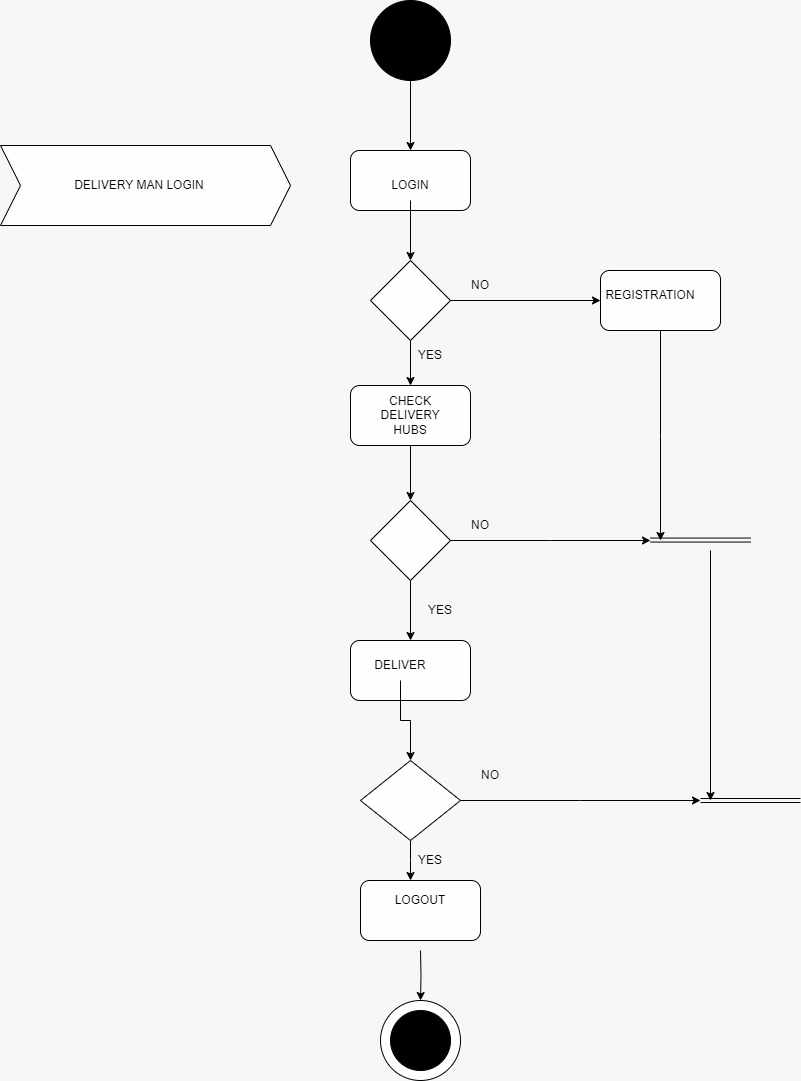
**Use Case Diagram:**

****

**Activity Diagram:**







**User-Interface:**

**Login Page**

**A screenshot of a login screen

Description automatically generated****SignUP Page**A screenshot of a login form

Description automatically generated

**Order Place Page**

A screenshot of a computer

Description automatically generated

**Track Product Page**

**A screenshot of a computer

Description automatically generated**

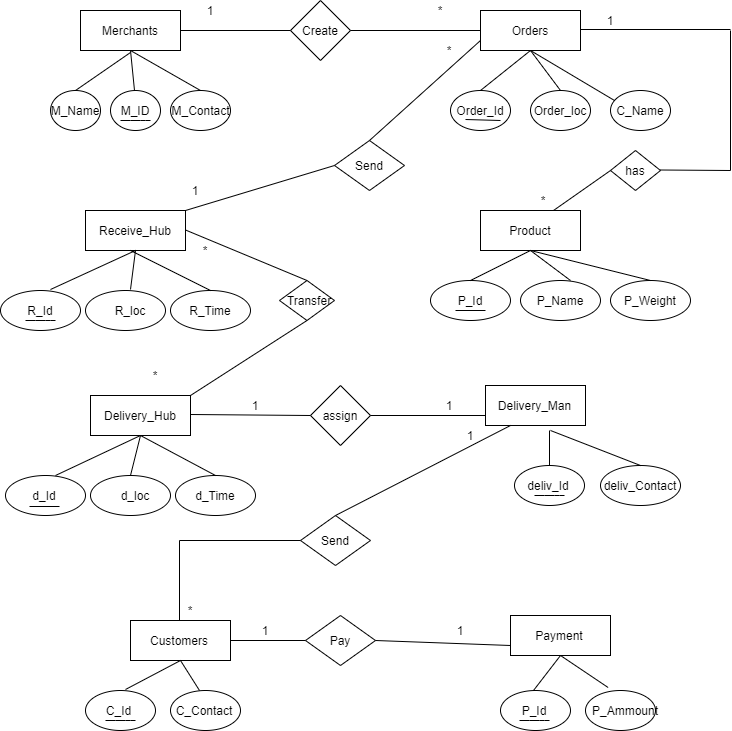
**Order Status Page** A screenshot of a computer

Description automatically generated

**Scenario Description:**

This is a courier service management system. In the data base customer’s details Customer id, customer contact will be stored. Customer will place an order and each of the order should have order id order location, order name. Customer will make payment and the payment id and payment amount will be stored on the data base. Then merchant will create an order receive hub. Marchant also have a unique id, name, contact. Receive hub id, location, time also be stored on the data base. Then transfer that to the delivery hub. Delivery hub have a unique delivery id, location and time. And it will come to deliveryman. Delivery man has a unique id, and contact number.

**ER Diagram:**



**Normalization:**

**NF**  
**UNF**  
M\_ID, M\_Name, M\_Contact, Order\_ID , Order\_Loc, C\_Name, P\_ID , P\_Name , P\_Weight, C\_Name, R\_ID , R\_Loc, R\_time, D\_ID , D\_Loc, D\_Time, Deliv\_ID , Deliv\_Contact

**Merchants**

1NF:

1: M\_ID, M\_Name, M\_Contact

2: Order\_ID , Order\_Loc, C\_Name

2NF:  
1: M\_ID, M\_Name, M\_Contact

2: Order\_ID , Order\_Loc, C\_Name, M\_ID

3NF:  
1: M\_ID, M\_Name, M\_Contact

2: Order\_ID , Order\_Loc, C\_Name, M\_ID

**Orders**

1NF:

1: Order\_ID , Order\_Loc, C\_Name

2: P\_ID , P\_Name , P\_Weight

2NF:  
1: Order\_ID , Order\_Loc, C\_Name

2: P\_ID , P\_Name , P\_Weight, Order\_ID

3NF:

1: Order\_ID , Order\_Loc, C\_Name

2: P\_ID , P\_Name , P\_Weight, Order\_ID

**Receive\_HUB**  
1NF:

1: Order\_ID , Order\_Loc, C\_Name

2: R\_ID , R\_Loc, R\_time

2NF:  
1: Order\_ID , Order\_Loc, C\_Name , R\_ID

2: R\_ID , R\_Loc, R\_time

3NF:  
1: Order\_ID , Order\_Loc, C\_Name , R\_ID

2: R\_ID , R\_Loc, R\_time

**Delivery\_Hub**

1NF:

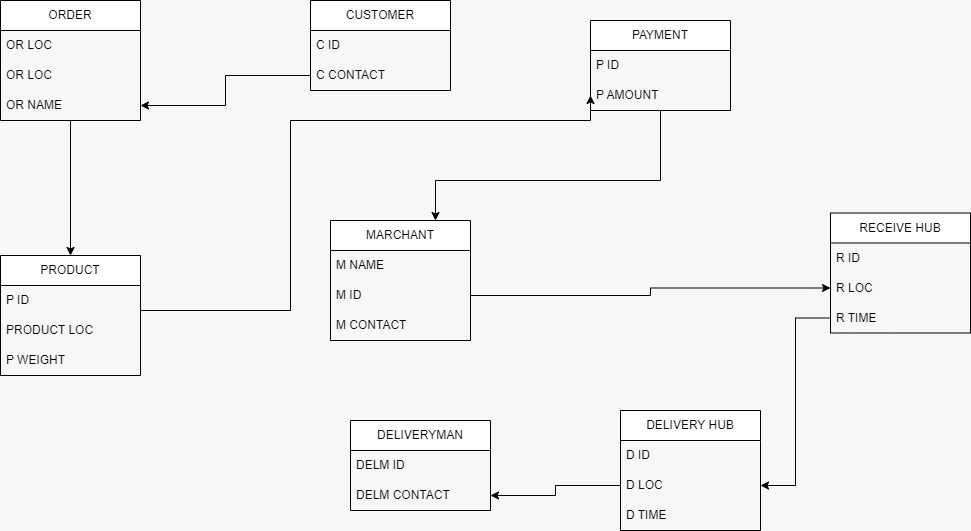
1: R\_ID , R\_Loc, R\_time

2: D\_ID , D\_Loc, D\_Time

2NF:  
1: R\_ID , R\_Loc, R\_time, D\_ID

2: D\_ID , D\_Loc, D\_Time

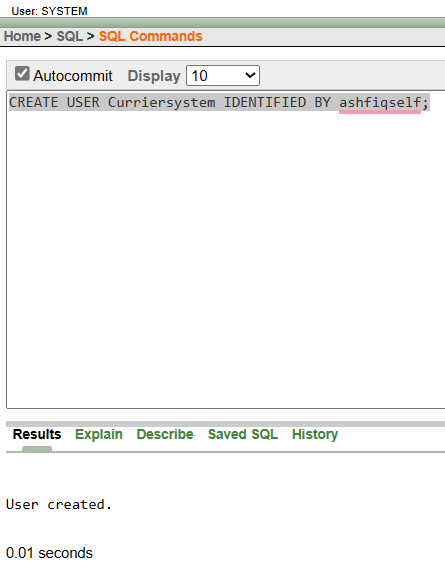
**Schema Diagram:**



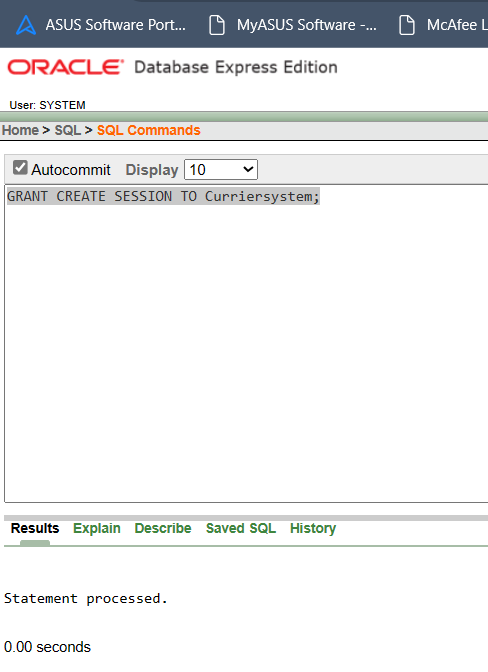
**Table Creation:**

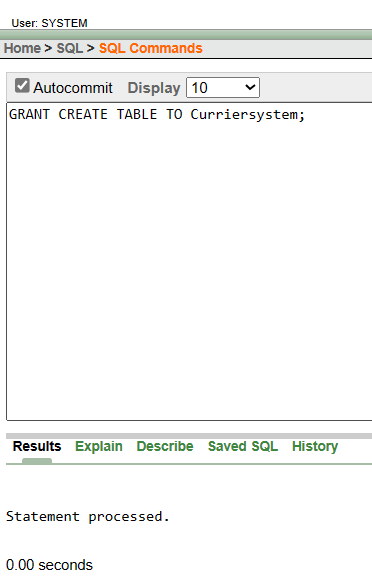
**User Create**

CREATE USER Curriersystem IDENTIFIED BY ashfiqself;

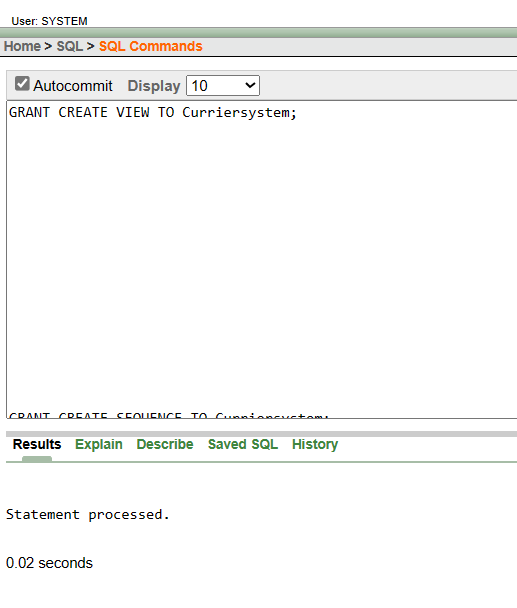


GRANT CREATE SESSION TO Curriersystem;

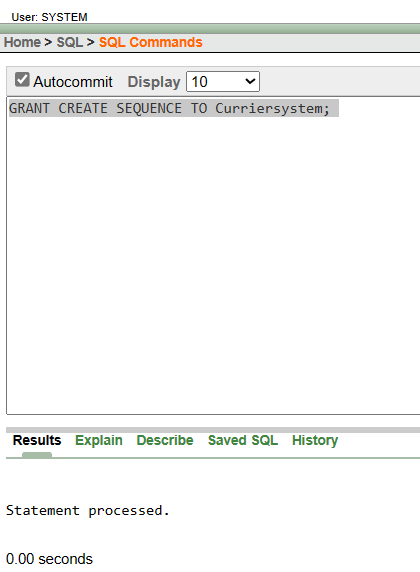


GRANT CREATE TABLE TO Curriersystem;  


GRANT CREATE VIEW TO Curriersystem;



GRANT CREATE SEQUENCE TO Curriersystem;



COMMIT;

A screenshot of a computer

Description automatically generated

**Create Marchant’s table**

CREATE TABLE Marchants (

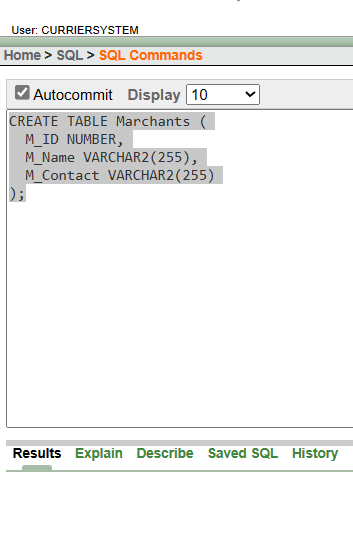
M\_ID NUMBER,

M\_Name VARCHAR2(255),

M\_Contact VARCHAR2(255)

);

DESCRIBE MARCHANTS;



A screenshot of a computer

Description automatically generated

**Order Table Creation**  
CREATE TABLE Orders (

Order\_ID NUMBER,

Order\_Loc VARCHAR2(255),

C\_Name VARCHAR2(255),

M\_ID NUMBER

);DESCRIBE ORDERS;  
A screenshot of a computer

Description automatically generated

**Product Table Creation**

CREATE TABLE Product (

P\_ID NUMBER,

P\_Name VARCHAR2(255),

P\_Weight NUMBER,

Order\_ID NUMBER);

DESCRIBE PRODUCTS;

A screenshot of a computer

Description automatically generated

**Receive\_Hub Table Creation**

CREATE TABLE Recieve\_Hub (

R\_ID NUMBER,

R\_Loc VARCHAR2(255),

R\_time TIMESTAMP,

D\_ID NUMBER);  
DESCRIBE RECEIVE\_HUB;

A screenshot of a computer

Description automatically generated

**Delivery\_HUB Table Creation**

DELIVARY\_HUB TABLE:

CREATE TABLE Delivery\_Hub (

D\_ID NUMBER,

D\_Loc VARCHAR2(255),

D\_Time TIMESTAMP,

Deliv\_ID NUMBER);

A screenshot of a computer

Description automatically generated

**Delivery\_Man Table:**

CREATE TABLE Delivery\_Man (

Deliv\_ID NUMBER,

Deliv\_Contact VARCHAR2(255));

A screenshot of a computer

Description automatically generated

**Customer Table**

CREATE TABLE Customers (

C\_ID INT PRIMARY KEY,

C\_Contact VARCHAR(50));

A screenshot of a computer

Description automatically generated

**Payment Table**

CREATE TABLE Paymentsys (

P\_ID INT PRIMARY KEY,

P\_Amount DECIMAL(10, 2),

C\_ID INT,

FOREIGN KEY (C\_ID) REFERENCES Customers (C\_ID)); A screenshot of a computer

Description automatically generated

**Data Insertion**

**Data Insertion in Merchants:**  
INSERT INTO Marchants (M\_ID, M\_Name, M\_Contact) VALUES (1, 'Ashfi', '01767333701');

INSERT INTO Marchants (M\_ID, M\_Name, M\_Contact) VALUES (2, 'Muhammed', '01767333733');

INSERT INTO Marchants (M\_ID, M\_Name, M\_Contact) VALUES (3, 'Raiyan', '01867333701');

INSERT INTO Marchants (M\_ID, M\_Name, M\_Contact) VALUES (4, 'Wahid', '01767236789');

INSERT INTO Marchants (M\_ID, M\_Name, M\_Contact) VALUES (5, 'Sajin', '01459861751');

A screenshot of a computer

Description automatically generated

**Data Insert in Orders Table:**  
INSERT INTO Orders (Order\_ID, Order\_Loc, C\_Name, M\_ID) VALUES (1, '11/A, Ranoir Kuti, Cumilla', 'Sabuj', 101);

INSERT INTO Orders (Order\_ID, Order\_Loc, C\_Name, M\_ID) VALUES (1002, 'F/A, Kuratuli, Dhaka', 'Rafin', 116);

INSERT INTO Orders (Order\_ID, Order\_Loc, C\_Name, M\_ID) VALUES (1015, '#12/D, Bashundhara, Dhaka', 'Rafsan', 204);

INSERT INTO Orders (Order\_ID, Order\_Loc, C\_Name, M\_ID) VALUES (1111, '702, Asf Taower, 4D,Banani, Dhaka', 'Ahmed', 305);

INSERT INTO Orders (Order\_ID, Order\_Loc, C\_Name, M\_ID) VALUES (2221, '#345, 6F, Jakir Tower, Madhya Badda, Dhaka' , 'Faruk', 410);

select \* from Orders;

A screenshot of a computer

Description automatically generated

**Product Table Data Insertion:**  
INSERT INTO Product (P\_ID, P\_Name, P\_Weight, Order\_ID) VALUES (1, 'Trimer', '1', 1);

INSERT INTO Product (P\_ID, P\_Name, P\_Weight, Order\_ID) VALUES (2, 'Makeup Box', '2', 1002);

INSERT INTO Product (P\_ID, P\_Name, P\_Weight, Order\_ID) VALUES (3, 'T-Shirt', '0.5', 1015);

INSERT INTO Product (P\_ID, P\_Name, P\_Weight, Order\_ID) VALUES (4, 'Mango', '5', 1111);

INSERT INTO Product (P\_ID, P\_Name, P\_Weight, Order\_ID) VALUES (5, 'Tool', '10', 2221);

select \* from product;

A screenshot of a computer

Description automatically generated

**Receive\_Hub Data Insertion:**  
INSERT INTO Recieve\_Hub (R\_ID, R\_Loc, R\_time, D\_ID) VALUES (101, 'Chittagong', TIMESTAMP '2023-07-13 10:00:00', 1);

INSERT INTO Recieve\_Hub (R\_ID, R\_Loc, R\_time, D\_ID) VALUES (102, 'Khulna', TIMESTAMP '2023-07-14 12:00:00', 2);

INSERT INTO Recieve\_Hub (R\_ID, R\_Loc, R\_time, D\_ID) VALUES (103, 'Rajshahi', TIMESTAMP '2023-07-15 14:00:00', 3);

INSERT INTO Recieve\_Hub (R\_ID, R\_Loc, R\_time, D\_ID) VALUES (104, 'Jessore', TIMESTAMP '2023-07-16 11:00:00', 4);

INSERT INTO Recieve\_Hub (R\_ID, R\_Loc, R\_time, D\_ID) VALUES (105, 'Rangpur', TIMESTAMP '2023-07-17 1:00:00', 5);

Select \* from Recieve\_Hub ;

A screenshot of a computer

Description automatically generated

**Delivery\_Hub Data Insertion:**  
INSERT INTO Delivery\_Hub (D\_ID, D\_Loc, D\_Time, Deliv\_ID) VALUES (201, 'Cumilla', TIMESTAMP '2023-07-20 08:00:00', 1);

INSERT INTO Delivery\_Hub (D\_ID, D\_Loc, D\_Time, Deliv\_ID) VALUES (202, 'Kuratoli, Dhaka', TIMESTAMP '2023-07-21 09:00:00', 2);

INSERT INTO Delivery\_Hub (D\_ID, D\_Loc, D\_Time, Deliv\_ID) VALUES (203, 'Bashundhara, Dhaka', TIMESTAMP '2023-07-22 10:00:00', 3);

INSERT INTO Delivery\_Hub (D\_ID, D\_Loc, D\_Time, Deliv\_ID) VALUES (204, 'Banani, Dhaka', TIMESTAMP '2023-07-23 09:00:00', 4);

INSERT INTO Delivery\_Hub (D\_ID, D\_Loc, D\_Time, Deliv\_ID) VALUES (205, 'Badda, Dhaka', TIMESTAMP '2023-07-24 09:00:00', 5);

Select \* from Delivery\_Hub;

A screenshot of a computer

Description automatically generated

**Delivery\_man Data Insertion:**

INSERT INTO Delivery\_man (Deliv\_ID, Deliv\_Contact) VALUES (10, '1561514515');

INSERT INTO Delivery\_man (Deliv\_ID, Deliv\_Contact) VALUES (20, '4565145455');

INSERT INTO Delivery\_man (Deliv\_ID, Deliv\_Contact) VALUES (30, '2151515154');

INSERT INTO Delivery\_man (Deliv\_ID, Deliv\_Contact) VALUES (40, '8851154596');

INSERT INTO Delivery\_man (Deliv\_ID, Deliv\_Contact) VALUES (50, '4448777774');

select \* from Delivery\_man ;

A screenshot of a computer

Description automatically generated

**Customer Data Insertion**

INSERT INTO Customers (C\_ID, C\_Contact)

VALUES (1,'123-456-7890');

INSERT INTO Customers (C\_ID, C\_Contact)

VALUES (2,'5551545599');

INSERT INTO Customers (C\_ID, C\_Contact)

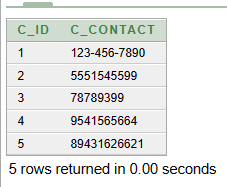
VALUES (3,'78789399');

INSERT INTO Customers (C\_ID, C\_Contact)

VALUES (4,'9541565664');

INSERT INTO Customers (C\_ID, C\_Contact)

VALUES (5,'89431626621');



**Payments Data Insertion**

INSERT INTO Paymentsys (P\_ID, P\_Amount, C\_ID) VALUES (1, 100.00, 1);

INSERT INTO Paymentsys (P\_ID, P\_Amount, C\_ID) VALUES (2, 100.00, 2);

INSERT INTO Paymentsys (P\_ID, P\_Amount, C\_ID) VALUES (3, 100.00, 3);

INSERT INTO Paymentsys (P\_ID, P\_Amount, C\_ID) VALUES (4, 100.00, 4);

INSERT INTO Paymentsys (P\_ID, P\_Amount, C\_ID) VALUES (5, 100.00, 5);

A screenshot of a calculator

Description automatically generated

**Query Writing**

**SQL**

**-3 single-row function**

**Question 1:** Retrieve the length of the M\_Name column for all rows in the Marchants table.

**Answer:**

SELECT M\_Name, LENGTH(M\_Name) AS Name\_Length

FROM Marchants;

A screenshot of a computer

Description automatically generated

**Question 2:** Retrieve the first three characters of the C\_Name for all rows in the Orders table.

**Answer:**

SELECT C\_Name, SUBSTR(C\_Name, 1, 3) AS First\_Three\_Characters

FROM Orders;

A screenshot of a computer

Description automatically generated

**Question 3:** Retrieve the current date and time.

**Answer:**

SELECT SYSDATE AS Current\_Date\_Time

FROM DUAL;

A screenshot of a computer

Description automatically generated

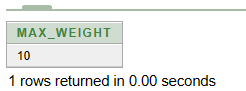
**-3 group function**

**Question 1:** Calculate the maximum P\_Weight in the Product table.

**Answer:**

SELECT MAX(P\_Weight) AS Max\_Weight

FROM Product;



**Question 2:** Count the number of distinct locations in the "Recieve Hub" table.

**Answer:**

SELECT COUNT(DISTINCT R\_Loc) AS Distinct\_Locations

FROM Receive\_Hub;

A screenshot of a computer

Description automatically generated

**Question: 3:** Determine the number of orders placed by each merchant in the Marchants table

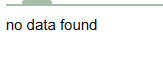
**Answer:**

SELECT M\_Name, COUNT(\*) AS Order\_Count

FROM Marchants M

JOIN "Order" O ON M.M\_ID = O.M\_ID

GROUP BY M\_Name;



**-3 subquery**

**Question 1:** Retrieve the C\_Name and Order\_Loc for all orders where the M\_ID is not present in the Marchants table.

**Answer:**

SELECT O.C\_Name, O.Order\_Loc

FROM Orders O

WHERE O.M\_ID NOT IN (SELECT M\_ID FROM Marchants);

A screenshot of a computer

Description automatically generated

**Question: 2:** Retrieve the M\_ID, M\_Name, and M\_Contact from the 'Marchants' table for the merchant with the highest M\_ID."

**Answer:**

SELECT M\_ID, M\_Name, M\_Contact

FROM Marchants

WHERE M\_ID = (SELECT MAX(M\_ID) FROM Marchants);

A screenshot of a computer

Description automatically generated

**Question: 3:** Retrieve the Order\_ID and Order\_Loc from the 'Orders' table for orders that have an Order\_Loc value present in the 'Customers' table.

**Answer:**

SELECT Order\_ID, Order\_Loc

FROM Orders

WHERE Order\_Loc IN (SELECT DISTINCT Order\_Loc FROM Customers);

A screenshot of a computer

Description automatically generated

**-3 joining:  
Inner Joining:**1. Inner Join between Orders and Product:

Answer:

SELECT O.Order\_ID, O.Order\_Loc, P.P\_ID, P.P\_Name

FROM Orders O

INNER JOIN Product P ON O.Order\_ID = P.Order\_ID;

A screenshot of a computer

Description automatically generated

**Left Joining:**

SELECT C.C\_ID, C.C\_Contact, P.P\_ID, P.P\_Amount

FROM Customers C

LEFT JOIN Paymentsys P ON C.C\_ID = P.C\_ID;

A screenshot of a computer

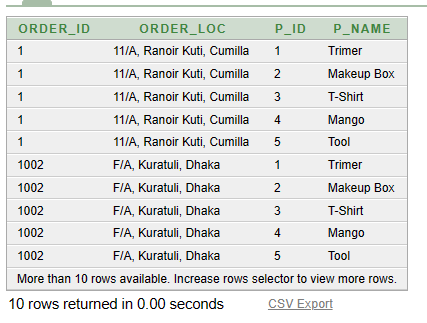
Description automatically generated

**Cross Joining:**

SELECT O.Order\_ID, O.Order\_Loc, P.P\_ID, P.P\_Name

FROM Orders O

CROSS JOIN Product P;



**-3 view**

**Question:1:Create View for Marchant Orders**

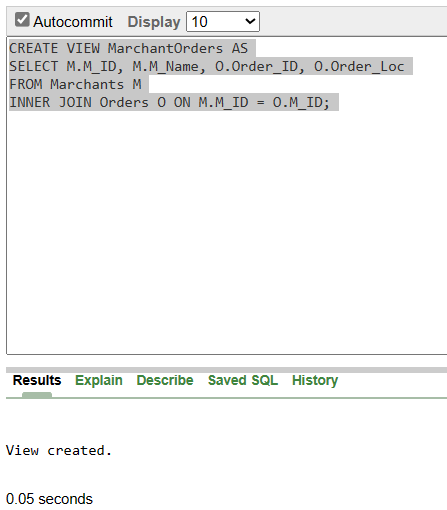
**Answer:**

CREATE VIEW MarchantOrders AS

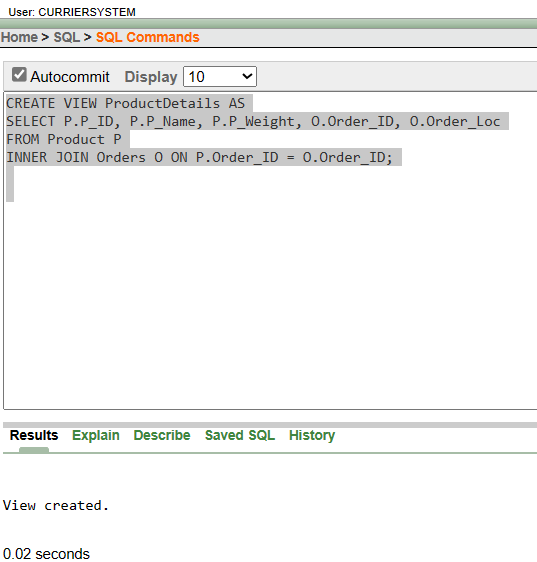
SELECT M.M\_ID, M.M\_Name, O.Order\_ID, O.Order\_Loc

FROM Marchants M

INNER JOIN Orders O ON M.M\_ID = O.M\_ID;

****

**Question:2: Create view for Product Details:**

****

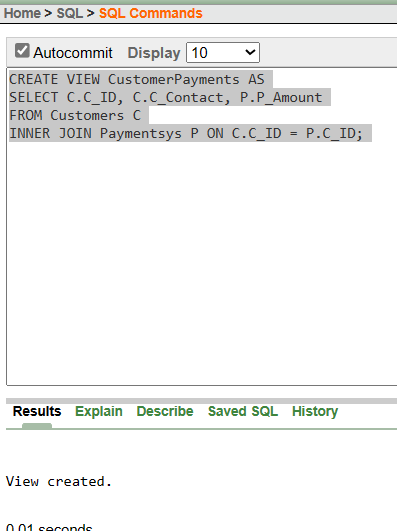
**Question: 3: Create view for Customer Payments:**

CREATE VIEW CustomerPayments AS

SELECT C.C\_ID, C.C\_Contact, P.P\_Amount

FROM Customers C

INNER JOIN Paymentsys P ON C.C\_ID = P.C\_ID;

****

**-3 synonym**

GRANT CREATE SYNONYM TO Curriersystem;

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Question: 1: Create a synonym for the "Marchants" table:**

**Answer:**

CREATE SYNONYM Marchants\_Synonym FOR Marchants;

**Question: 2: Create a synonym for the "Orders" table:**

**Answer:**

CREATE SYNONYM Orders\_Synonym FOR Orders;

**Question: 3: Create a synonym for the "Customers" table:**

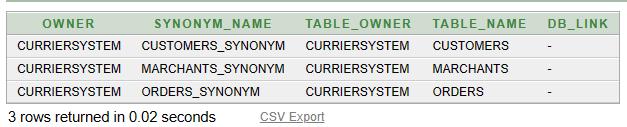
**Answer:**

CREATE SYNONYM Customers\_Synonym FOR Customers;

SELECT \*

FROM ALL\_SYNONYMS

WHERE OWNER = 'CURRIERSYSTEM';



**Relational Algebra:**

**Selection**: Retrieve all orders from the Orders table where the order location is in Dhaka. σ(Order\_Loc = 'Dhaka', Orders)

**Projection**: Retrieve the order ID and customer name from the Orders table.

π(Order\_ID, C\_Name)(Orders)

**Union**: Combine the orders from the Orders table and the receive hubs from the Receive\_Hub table.

Orders ⋃ Receive\_Hub

**Cartesian Product**: Generate all possible combinations of orders and delivery hubs.

Orders × Delivery\_Hub

**Rename**: Change the attribute name "D\_Time" to "Delivery\_Time" in the Delivery\_Hub table. ρ(Delivery\_Hub, (D\_Time AS Delivery\_Time))(Delivery\_Hub)

**Conclusion:**

In final term, We plan to add customer to merchant communication system. Customer to delivery man communication system.