Assignment – Courier Management System

Task 1: Database Design

Design a SQL schema for a Courier Management System with tables for Customers, Couriers, Orders, and Parcels. Define the relationships between these tables using appropriate foreign keys.

**Requirements:**

• Define the Database Schema • Create SQL tables for entities such as User, Courier, Employee, Location, Payment.

• Define relationships between these tables (one-to-many, many-to-many, etc.).

• Populate Sample Data

• Insert sample data into the tables to simulate real-world scenarios.

Solution:

Database Schema

CREATE TABLE User (UserID INT PRIMARY KEY, Name VARCHAR(255), Email VARCHAR(255) UNIQUE, Password VARCHAR(255), ContactNumber VARCHAR(20), Address TEXT);

CREATE TABLE Courier (CourierID INT PRIMARY KEY, SenderName VARCHAR(255), SenderAddress TEXT, ReceiverName VARCHAR(255), ReceiverAddress TEXT, Weight DECIMAL(5, 2), Status VARCHAR(50), TrackingNumber VARCHAR(20) UNIQUE, DeliveryDate DATE);

CREATE TABLE CourierServices (ServiceID INT PRIMARY KEY, ServiceName VARCHAR(100), Cost DECIMAL(8, 2));

CREATE TABLE Employee (EmployeeID INT PRIMARY KEY, Name VARCHAR(255), Email VARCHAR(255) UNIQUE, ContactNumber VARCHAR(20), Role VARCHAR(50), Salary DECIMAL(10, 2));

CREATE TABLE Location (LocationID INT PRIMARY KEY, LocationName VARCHAR(100), Address TEXT);

CREATE TABLE Payment (PaymentID INT PRIMARY KEY, CourierID INT, LocationID INT, Amount DECIMAL(10, 2), PaymentDate DATE, FOREIGN KEY (CourierID) REFERENCES Courier(CourierID), FOREIGN KEY (LocationID) REFERENCES Location(LocationID));



Populating Sample Data:

INSERT INTO User (UserID, Name, Email, Password, ContactNumber, Address) VALUES (1, 'Rajesh Kumar', 'rajesh.kumar@example.com', 'Pass@123', '9876543210', '12, MG Road, Bengaluru'), (2, 'Aditi Sharma', 'aditi.sharma@example.com', 'Secure@456', '8765432109', '45, Nehru Nagar, New Delhi'), (3, 'Vikram Reddy', 'vikram.reddy@example.com', 'Strong@789', '7654321098', '88, Jubilee Hills, Hyderabad'), (4, 'Nisha Patel', 'nisha.patel@example.com', 'Safe@101', '6543210987', '23, SG Highway, Ahmedabad'), (5, 'Manoj Singh', 'manoj.singh@example.com', 'Protect@202', '5432109876', '67, MI Road, Jaipur');

INSERT INTO Courier (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate) VALUES (1, 'Rajesh Kumar', '12, MG Road, Bengaluru', 'Nisha Patel', '23, SG Highway, Ahmedabad', 2.50, 'Shipped', 'INTRK001234', '2024-10-01'), (2, 'Aditi Sharma', '45, Nehru Nagar, New Delhi', 'Vikram Reddy', '88, Jubilee Hills, Hyderabad', 1.75, 'Delivered', 'INTRK001235', '2024-09-25'), (3, 'Vikram Reddy', '88, Jubilee Hills, Hyderabad', 'Manoj Singh', '67, MI Road, Jaipur', 3.20, 'In Transit', 'INTRK001236', '2024-10-03'), (4, 'Nisha Patel', '23, SG Highway, Ahmedabad', 'Rajesh Kumar', '12, MG Road, Bengaluru', 4.10, 'Shipped', 'INTRK001237', '2024-10-02'), (5, 'Manoj Singh', '67, MI Road, Jaipur', 'Aditi Sharma', '45, Nehru Nagar, New Delhi', 5.00, 'Pending', 'INTRK001238', '2024-10-05');

INSERT INTO CourierServices (ServiceID, ServiceName, Cost) VALUES (1, 'Standard Delivery', 100.00), (2, 'Express Delivery', 200.00), (3, 'Same Day Delivery', 300.00), (4, 'Overnight Delivery', 250.00), (5, 'International Delivery', 500.00);

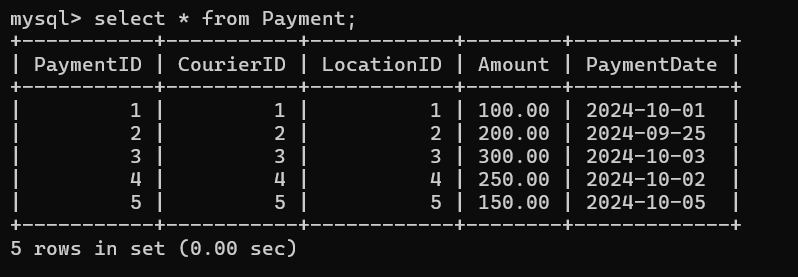
INSERT INTO Employee (EmployeeID, Name, Email, ContactNumber, Role, Salary) VALUES (1, 'Suresh Bhatia', 'suresh.bhatia@example.com', '9988776655', 'Courier Handler', 25000.00), (2, 'Priya Iyer', 'priya.iyer@example.com', '9877665544', 'Delivery Agent', 22000.00), (3, 'Anil Deshmukh', 'anil.deshmukh@example.com', '9766554433', 'Logistics Manager', 35000.00), (4, 'Neha Kapoor', 'neha.kapoor@example.com', '9655443322', 'Warehouse Supervisor', 27000.00), (5, 'Vivek Jain', 'vivek.jain@example.com', '9544332211', 'Delivery Coordinator', 24000.00);

INSERT INTO Location (LocationID, LocationName, Address) VALUES (1, 'Bengaluru Warehouse', 'Electronic City, Bengaluru'), (2, 'New Delhi Hub', 'Connaught Place, New Delhi'), (3, 'Hyderabad Sorting Center', 'Begumpet, Hyderabad'), (4, 'Ahmedabad Warehouse', 'SG Highway, Ahmedabad'), (5, 'Jaipur Delivery Center', 'MI Road, Jaipur');

INSERT INTO Payment (PaymentID, CourierID, LocationID, Amount, PaymentDate) VALUES (1, 1, 1, 100.00, '2024-10-01'), (2, 2, 2, 200.00, '2024-09-25'), (3, 3, 3, 300.00, '2024-10-03'), (4, 4, 4, 250.00, '2024-10-02'), (5, 5, 5, 150.00, '2024-10-05');







Task 2: Select, Where

Solve the following queries in the Schema that you have created above

1. List all customers:

2. List all orders for a specific customer:

3. List all couriers:

4. List all packages for a specific order:

5. List all deliveries for a specific courier:

6. List all undelivered packages:

7. List all packages that are scheduled for delivery today:

8. List all packages with a specific status:

9. Calculate the total number of packages for each courier.

10. Find the average delivery time for each courier

11. List all packages with a specific weight range:

12. Retrieve employees whose names contain 'John'

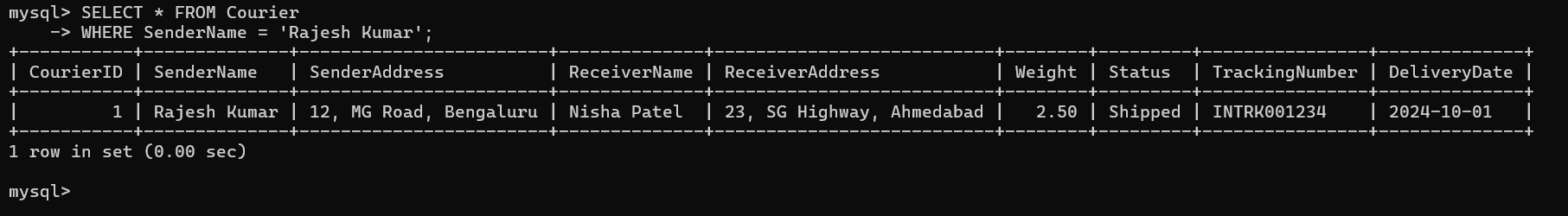
13. Retrieve all courier records with payments greater than $50.

Solution

SELECT \* FROM User;

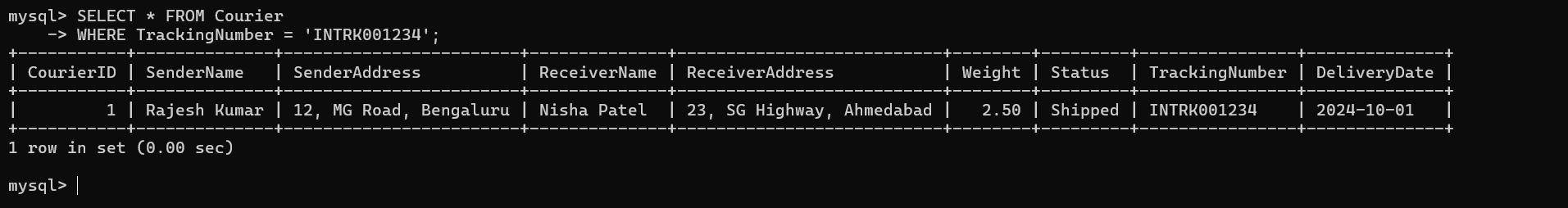


SELECT \* FROM Courier WHERE SenderName = 'Rajesh Kumar';

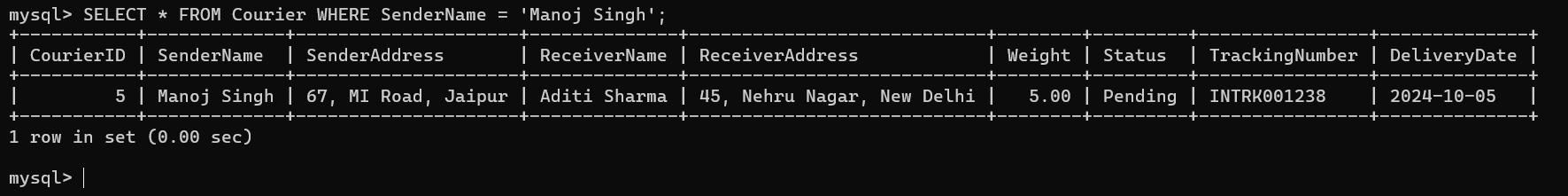


SELECT \* FROM Courier; 

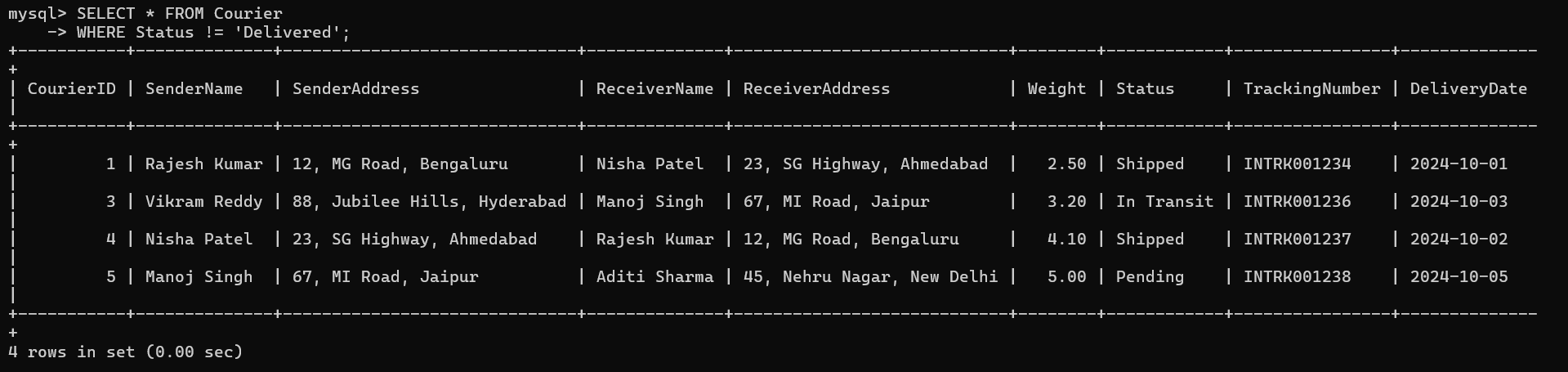
SELECT \* FROM Courier WHERE TrackingNumber = 'INTRK001234';



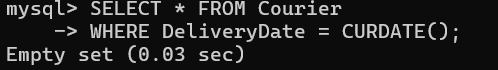
SELECT \* FROM Courier WHERE SenderName = 'Suresh Bhatia';



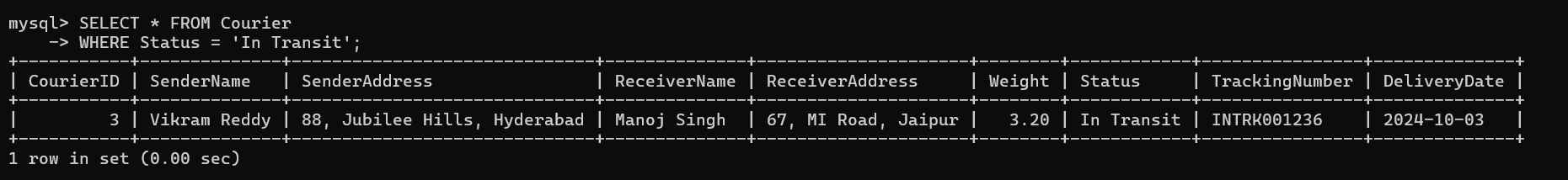
SELECT \* FROM Courier WHERE Status != 'Delivered';



SELECT \* FROM Courier WHERE DeliveryDate = CURDATE();



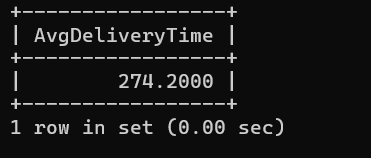
SELECT \* FROM Courier WHERE Status = 'In Transit';



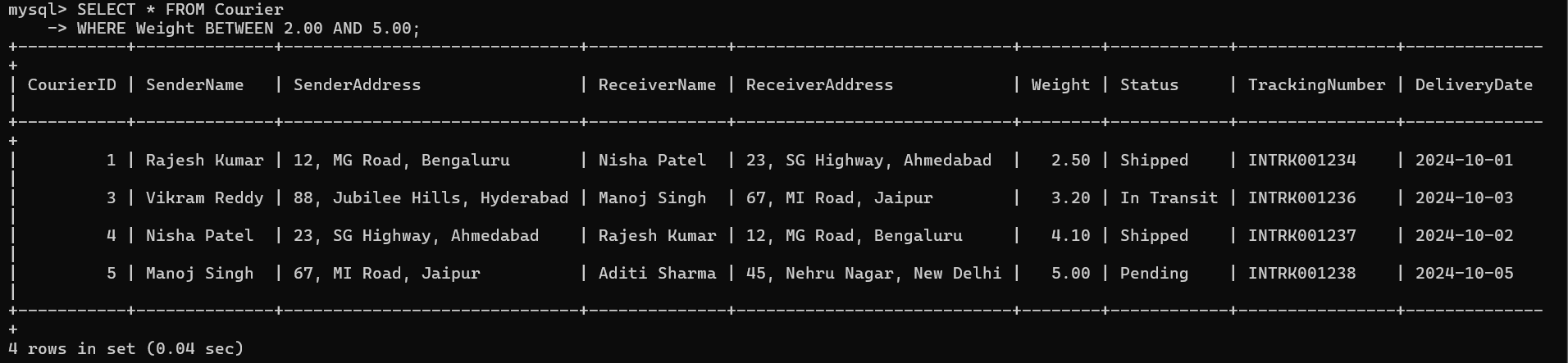
SELECT COUNT(CourierID) AS TotalPackages FROM Courier;



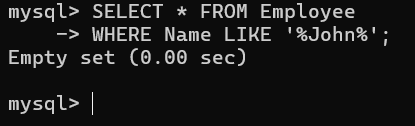
SELECT AVG(DATEDIFF(DeliveryDate, '2024-01-01')) AS AvgDeliveryTime FROM Courier WHERE DeliveryDate IS NOT NULL;



SELECT \* FROM Courier WHERE Weight BETWEEN 2.00 AND 5.00;



SELECT \* FROM Employee WHERE Name LIKE '%John%';



SELECT \* FROM Courier WHERE CourierID IN (SELECT CourierID

FROM Payment

WHERE Amount > 50

);



Task 3: Group By, Aggregate Functions, Having, Order By, where

14. Find the total number of couriers handled by each employee.

15. Calculate the total revenue generated by each location

16. Find the total number of couriers delivered to each location.

17. Find the courier with the highest average delivery time:

18. Find Locations with Total Payments Less Than a Certain Amount

19. Calculate Total Payments per Location

20. Retrieve couriers who have received payments totaling more than $1000 in a specific location

(LocationID = X):

21. Retrieve couriers who have received payments totaling more than $1000 after a certain date

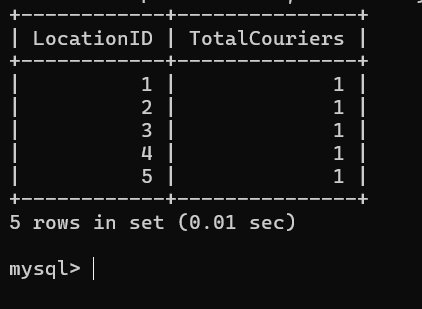
(PaymentDate > 'YYYY-MM-DD'):

22. Retrieve locations where the total amount received is more than $5000 before a certain date

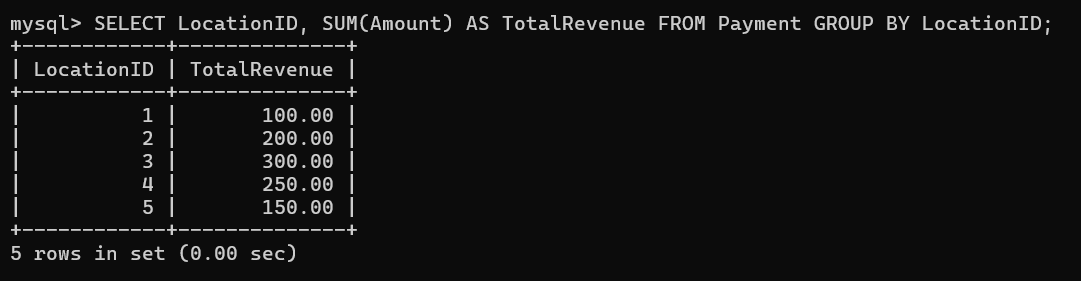
(PaymentDate > 'YYYY-MM-DD')

Solution

SELECT p.LocationID, COUNT(DISTINCT c.CourierID) AS TotalCouriers FROM Payment p JOIN Courier c ON p.CourierID = c.CourierID GROUP BY p.LocationID;

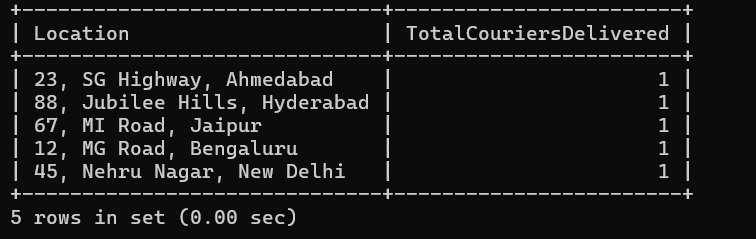


SELECT LocationID, SUM(Amount) AS TotalRevenue FROM Payment GROUP BY LocationID;



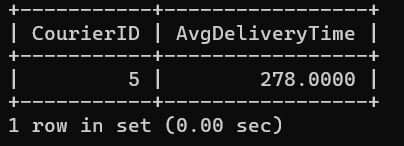
SELECT c.ReceiverAddress AS Location, COUNT(c.CourierID) AS TotalCouriersDelivered FROM Courier c

GROUP BY c.ReceiverAddress;

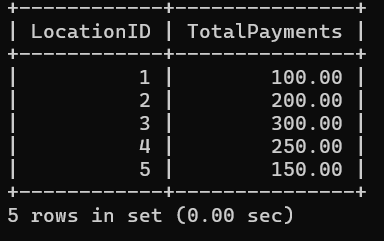


SELECT CourierID, AVG(DATEDIFF(DeliveryDate, '2024-01-01')) AS AvgDeliveryTime FROM Courier

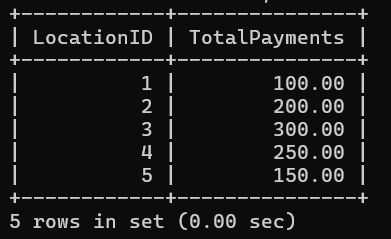
GROUP BY CourierID ORDER BY AvgDeliveryTime DESC;



SELECT LocationID, SUM(Amount) AS TotalPayments FROM Payment GROUP BY LocationID HAVING TotalPayments < 500;



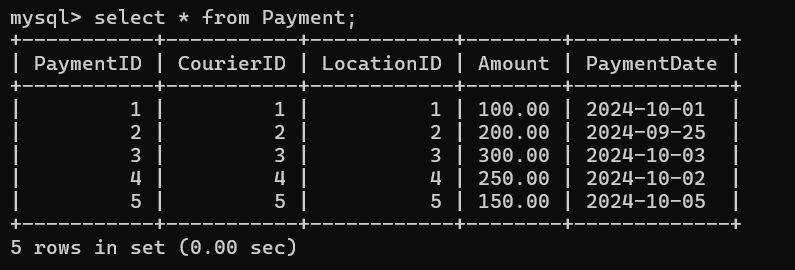
SELECT LocationID, SUM(Amount) AS TotalPayments FROM Payment GROUP BY LocationID;

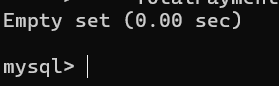


SELECT c.CourierID, SUM(p.Amount) AS TotalPayments, l.LocationName FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID JOIN Location l ON p.LocationID = l.LocationID WHERE l.LocationID = 1 GROUP BY c.CourierID, l.LocationName HAVING TotalPayments > 1000;



SELECT c.CourierID, SUM(p.Amount) AS TotalPayments FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID WHERE p.PaymentDate > '2024-10-02' WHERE GROUP BY c.CourierID HAVING TotalPayments > 1000;



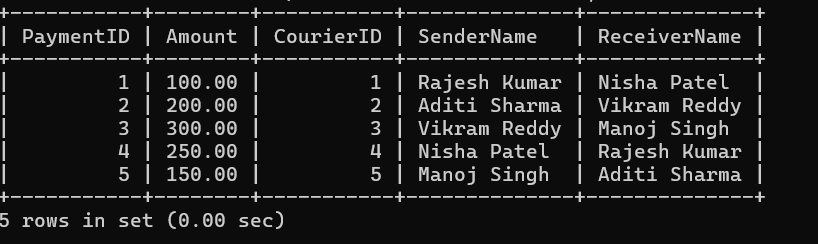


SELECT p.LocationID, SUM(p.Amount) AS TotalReceived FROM Payment p WHERE p.PaymentDate < '2024-01-01' GROUP BY p.LocationID HAVING TotalReceived > 5000;

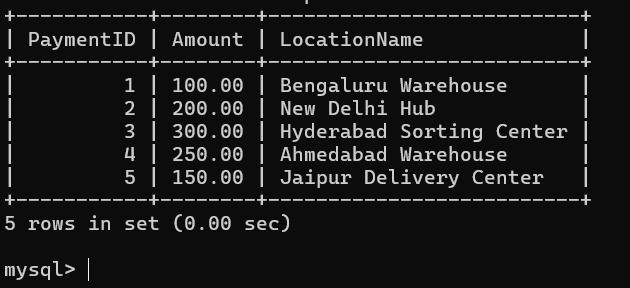


Task 4: Inner Join, Full Outer Join, Cross Join, Left Outer Join, Right Outer Join

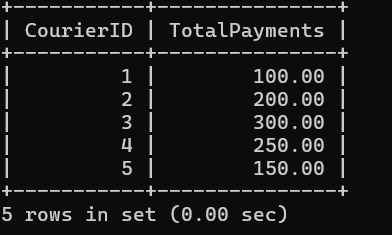
SELECT p.PaymentID, p.Amount, c.CourierID, c.SenderName, c.ReceiverName FROM Payment p LEFT JOIN Courier c ON p.CourierID = c.CourierID;



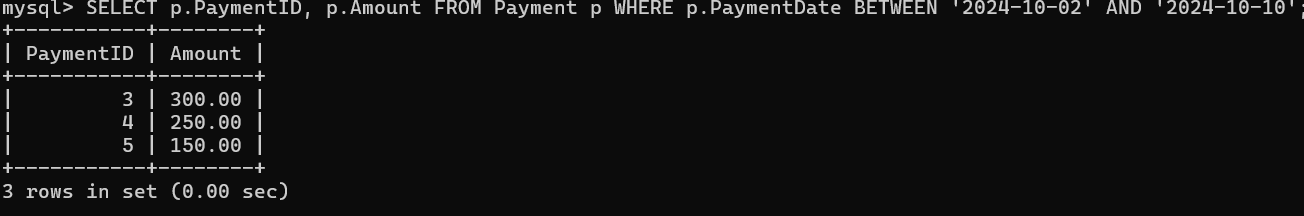
SELECT p.PaymentID, p.Amount, l.LocationName FROM Payment p LEFT JOIN Location l ON p.LocationID = l.LocationID;



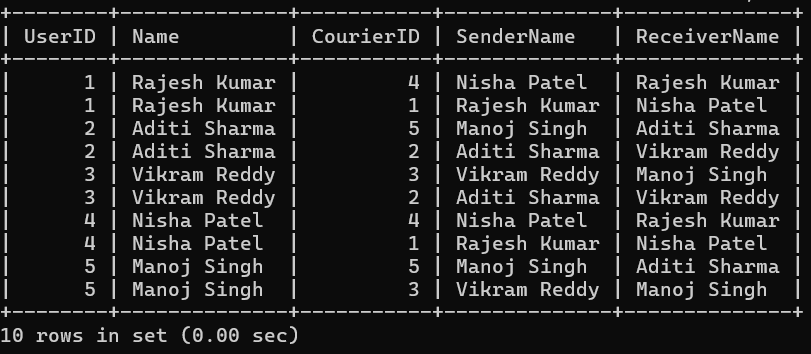
SELECT c.CourierID, SUM(p.Amount) AS TotalPayments FROM Payment p INNER JOIN Courier c ON p.CourierID = c.CourierID GROUP BY c.CourierID;



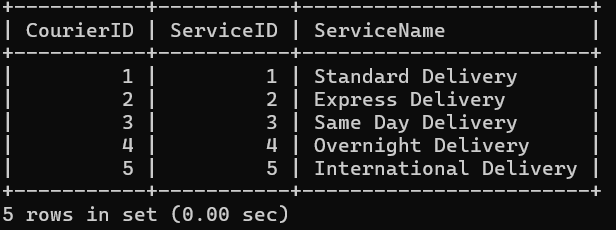
SELECT p.PaymentID, p.Amount FROM Payment p WHERE p.PaymentDate BETWEEN '2024-10-02' AND '2024-10-10';



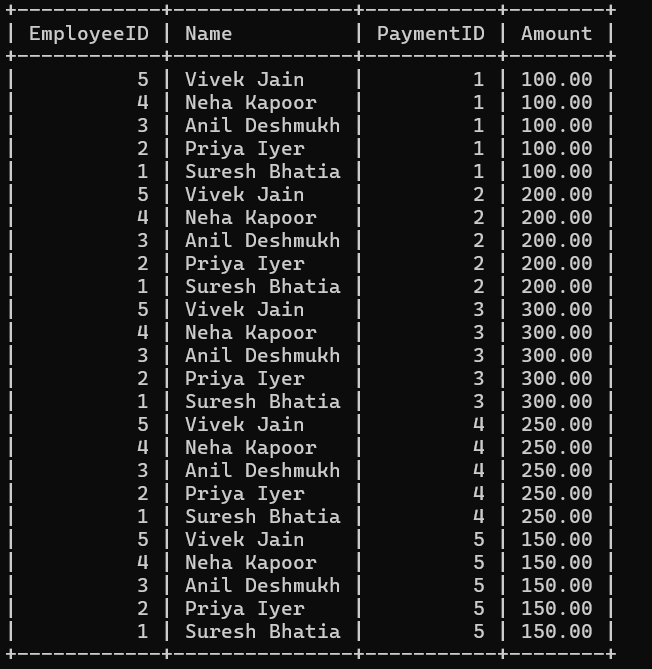
SELECT u.UserID, u.Name, c.CourierID, c.SenderName, c.ReceiverName FROM User u LEFT JOIN Courier c ON u.Name = c.SenderName OR u.Name = c.ReceiverName;



SELECT c.CourierID, cs.ServiceID, cs.ServiceName FROM Courier c LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID;



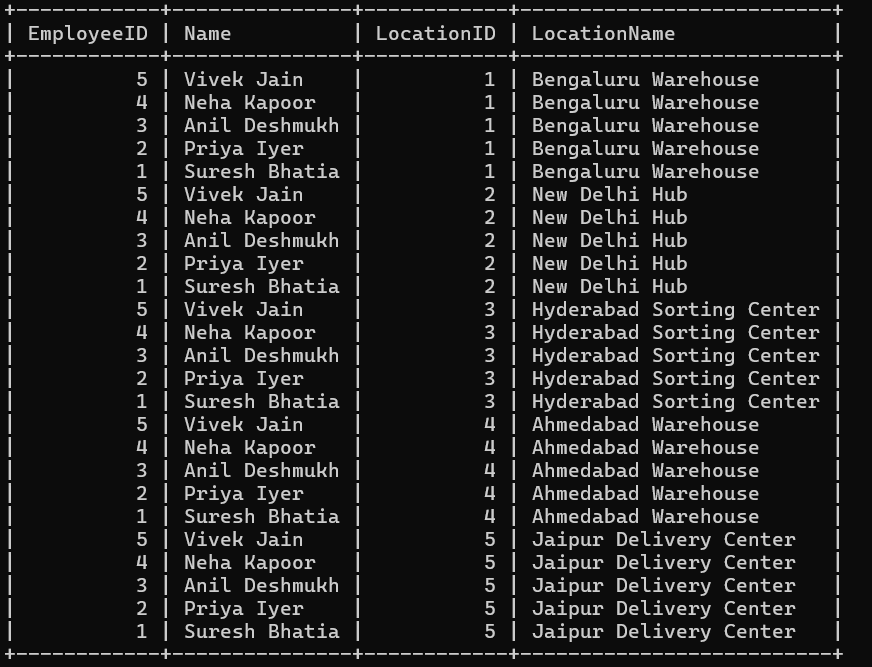
SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount FROM Employee e CROSS JOIN Payment p;



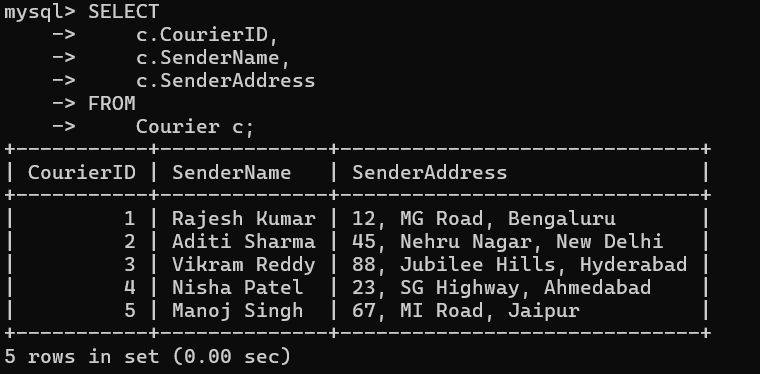
SELECT u.UserID, u.Name, cs.ServiceID, cs.ServiceName FROM User u CROSS JOIN CourierServices cs;



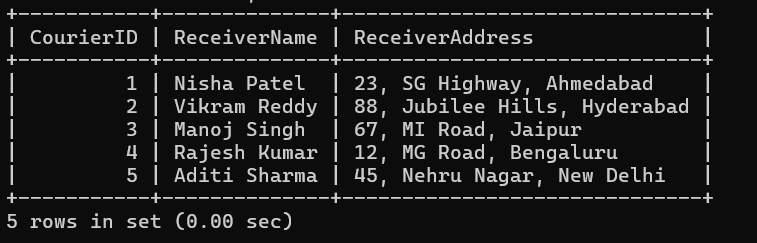
SELECT e.EmployeeID, e.Name, l.LocationID, l.LocationName FROM Employee e CROSS JOIN Location l;



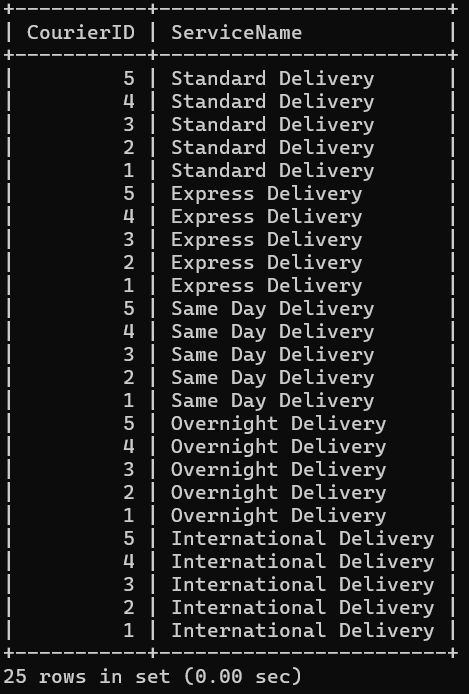
SELECT c.CourierID, c.SenderName, c.SenderAddress FROM Courier c;



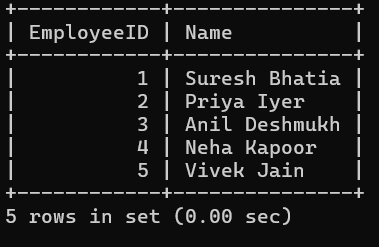
SELECT c.CourierID, c.ReceiverName, c.ReceiverAddress FROM Courier c;



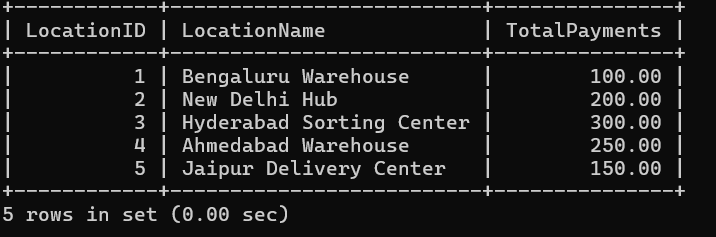
SELECT c.CourierID, cs.ServiceName FROM Courier c CROSS JOIN CourierServices cs;



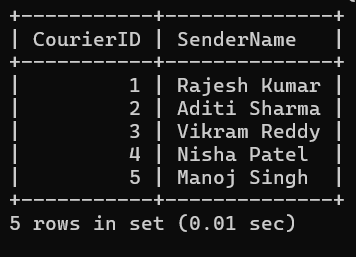
SELECT e.EmployeeID, e.Name FROM Employee e;

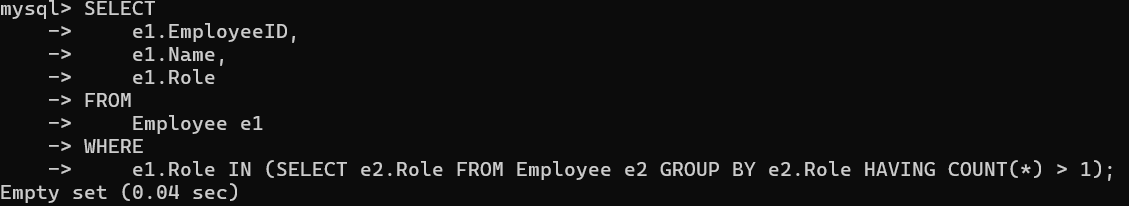


SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayments FROM Location l LEFT JOIN Payment p ON l.LocationID = p.LocationID GROUP BY l.LocationID;

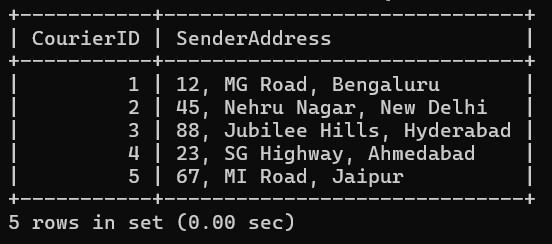


SELECT c1.CourierID, c1.SenderName FROM Courier c1 WHERE c1.SenderName IN (SELECT SenderName FROM Courier c2 WHERE c1.SenderName = c2.SenderName);

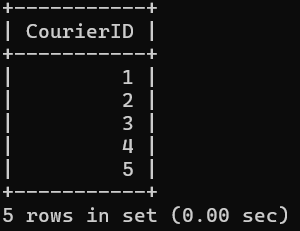


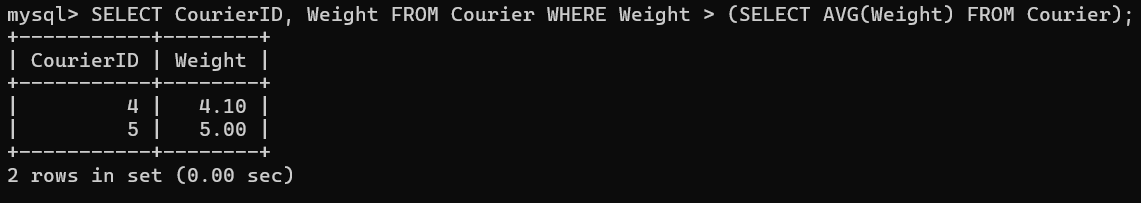


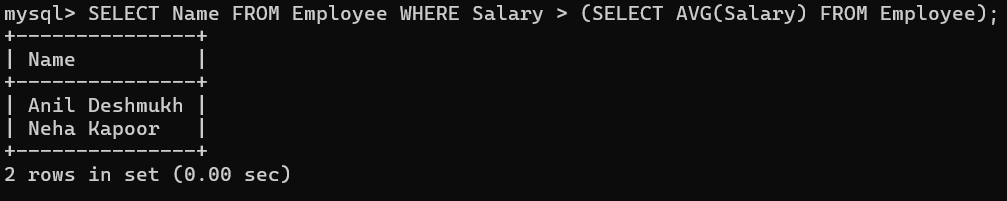
SELECT c1.CourierID, c1.SenderAddress FROM Courier c1 WHERE c1.SenderAddress IN (SELECT SenderAddress FROM Courier c2 WHERE c1. c2.SenderAddress);



SELECT c.CourierID FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID WHERE p.Amount > 50;



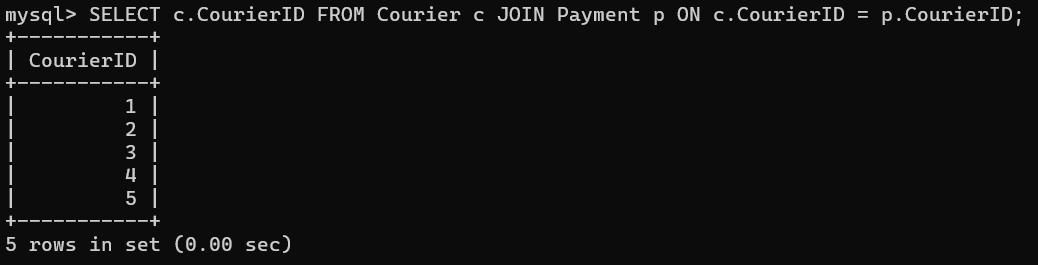




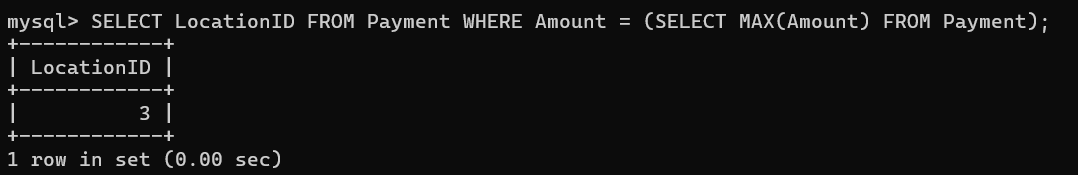
SELECT SUM(Cost) AS TotalCost FROM CourierServices WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);



SELECT c.CourierID FROM Courier c JOIN Payment p ON c.CourierID = p.CourierID;



SELECT LocationID FROM Payment WHERE Amount = (SELECT MAX(Amount) FROM Payment);



SELECT CourierID, Weight FROM Courier WHERE Weight > (SELECT MAX(Weight) FROM Courier WHERE SenderName = 'SenderName');



ER Diagram

