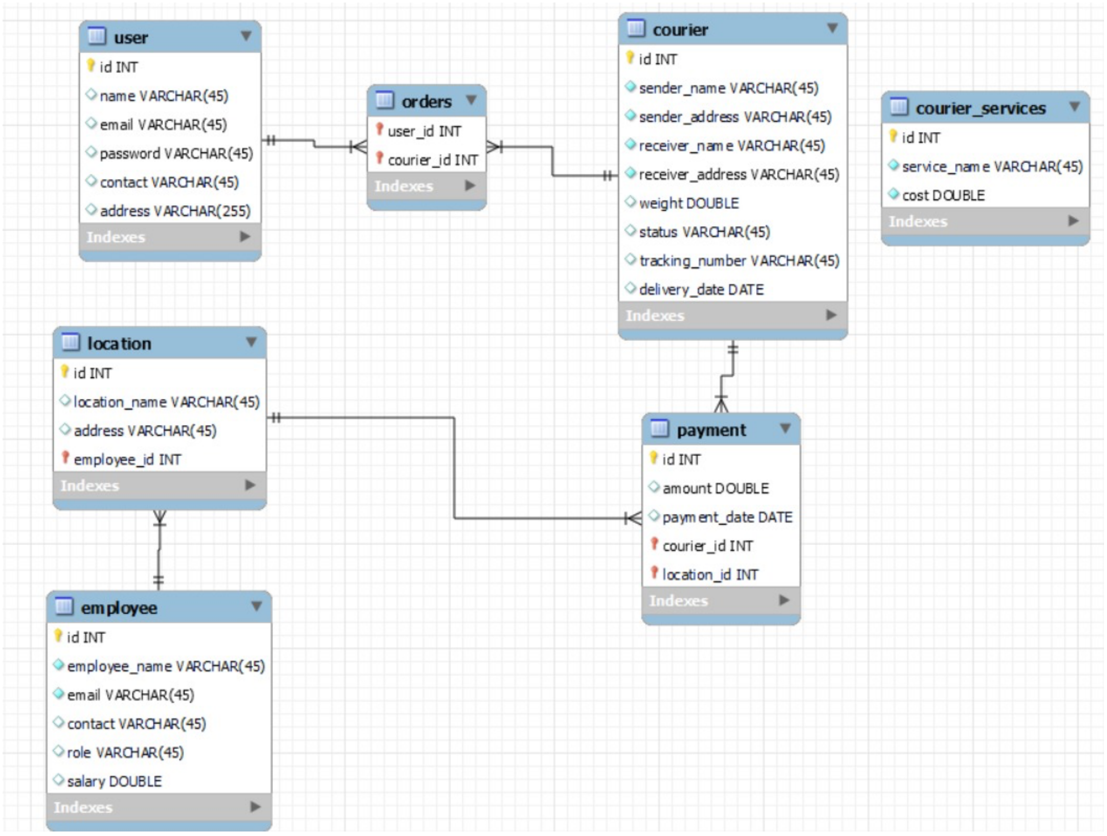
Project- Courier Management System

ER Diagram:



Code:

```
create database learnings4;
use learnings4;


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

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)
);


INSERT INTO User (UserID, Name, Email, Password, ContactNumber, Address)
VALUES
(1, 'Arjun Kumar', 'arjun@example.com', 'password123', '1234567890', 'Chennai, Tamil Nadu, India'),
(2, 'Divya Rao', 'divya@example.com', 'password456', '9876543210', 'Hyderabad, Telangana, India'),
(3, 'Nithya Nair', 'nithya@example.com', 'password789', '1231231234', 'Kochi, Kerala, India'),
(4, 'Rajesh Sharma', 'rajesh@example.com', 'passwordabc', '4564564567', 'Bangalore, Karnataka, India'),
(5, 'Priya Menon', 'priya@example.com', 'passworddef', '7897897890', 'Coimbatore, Tamil Nadu, India'),
(6, 'Aruna Krishnan', 'aruna@example.com', 'passwordghi', '0987654321', 'Thiruvananthapuram, Kerala,
India'),
(7, 'Sanjay Iyer', 'sanjay@example.com', 'passwordjkl', '9876543210', 'Mysore, Karnataka, India'),
(8, 'Deepika Reddy', 'deepika@example.com', 'passwordmno', '1234567890', 'Visakhapatnam, Andhra
```

```sql
Pradesh, India'),
(9, 'Manoj Kumar', 'manoj@example.com', 'passwordpqr', '4567890123', 'Vijayawada, Andhra Pradesh,
India'),
(10, 'Sangeeta Singh', 'sangeeta@example.com', 'passwordstu', '7890123456', 'Puducherry, India');


INSERT INTO Courier (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight,
Status, TrackingNumber, DeliveryDate)
VALUES
(1, 'Sender1', 'Address1, Chennai, Tamil Nadu, India', 'Receiver1', 'Address2, Hyderabad, Telangana, India',
2.5, 'Pending', '123456', '2024-03-10'),
(2, 'Sender2', 'Address3, Kochi, Kerala, India', 'Receiver2', 'Address4, Bangalore, Karnataka, India', 1.8, 'In
Transit', '789012', '2024-03-12'),
(3, 'Sender3', 'Address5, Coimbatore, Tamil Nadu, India', 'Receiver3', 'Address6, Thiruvananthapuram,
Kerala, India', 3.2, 'Delivered', '345678', '2024-03-14'),
(4, 'Sender4', 'Address7, Mysore, Karnataka, India', 'Receiver4', 'Address8, Visakhapatnam, Andhra
Pradesh, India', 1.5, 'Pending', '901234', '2024-03-16'),
(5, 'Sender5', 'Address9, Vijayawada, Andhra Pradesh, India', 'Receiver5', 'Address10, Puducherry, India',
2.0, 'In Transit', '234567', '2024-03-18'),
(6, 'Sender6', 'Address11, Mysore, Karnataka, India', 'Receiver6', 'Address12, Hyderabad, Telangana, India',
2.8, 'Delivered', '654321', '2024-03-20'),
(7, 'Sender7', 'Address13, Coimbatore, Tamil Nadu, India', 'Receiver7', 'Address14, Chennai, Tamil Nadu,
India', 1.2, 'Delivered', '876543', '2024-03-22'),
(8, 'Sender8', 'Address15, Bangalore, Karnataka, India', 'Receiver8', 'Address16, Thiruvananthapuram,
Kerala, India', 2.3, 'Pending', '432109', '2024-03-24'),
(9, 'Sender9', 'Address17, Kochi, Kerala, India', 'Receiver9', 'Address18, Visakhapatnam, Andhra Pradesh,
India', 3.5, 'In Transit', '765432', '2024-03-26'),
(10, 'Sender10', 'Address19, Vijayawada, Andhra Pradesh, India', 'Receiver10', 'Address20, Puducherry,
India', 1.9, 'Delivered', '210987', '2024-03-28');


INSERT INTO CourierServices (ServiceID, ServiceName, Cost)
VALUES
(1, 'Standard', 10.00),
(2, 'Express', 15.00),
(3, 'Priority', 20.00),
(4, 'Next Day Delivery', 25.00),
(5, 'Overnight', 30.00),
(6, 'Same Day Delivery', 35.00),
(7, 'International', 50.00),
(8, 'Domestic', 40.00),
(9, 'Special Handling', 45.00),
(10, 'Customized', 55.00);


INSERT INTO Employee (EmployeeID, Name, Email, ContactNumber, Role, Salary)
VALUES
(1, 'Employee1', 'emp1@example.com', '1111111111', 'Manager', 50000.00),
(2, 'Employee2', 'emp2@example.com', '2222222222', 'Delivery Staff', 30000.00),
(3, 'Employee3', 'emp3@example.com', '3333333333', 'Warehouse Supervisor', 45000.00),
(4, 'Employee4', 'emp4@example.com', '4444444444', 'Customer Service Representative', 35000.00),
(5, 'Employee5', 'emp5@example.com', '5555555555', 'Driver', 28000.00),
(6, 'Employee6', 'emp6@example.com', '6666666666', 'IT Support', 40000.00),
(7, 'Employee7', 'emp7@example.com', '7777777777', 'Accountant', 42000.00),
(8, 'Employee8', 'emp8@example.com', '8888888888', 'Human Resources', 38000.00),
(9, 'Employee9', 'emp9@example.com', '9999999999', 'Security', 32000.00),
(10, 'Employee10', 'emp10@example.com', '1010101010', 'Sales Representative', 37000.00);

INSERT INTO Location (LocationID, LocationName, Address)
VALUES
(1, 'Location1', 'Address1, Chennai, Tamil Nadu, India'),
(2, 'Location2', 'Address2, Hyderabad, Telangana, India'),
(3, 'Location3', 'Address3, Kochi, Kerala, India'),
(4, 'Location4', 'Address4, Bangalore, Karnataka, India'),
(5, 'Location5', 'Address5, Coimbatore, Tamil Nadu, India'),
(6, 'Location6', 'Address6, Thiruvananthapuram, Kerala, India'),
(7, 'Location7', 'Address7, Mysore, Karnataka, India'),
(8, 'Location8', 'Address8, Visakhapatnam, Andhra Pradesh, India'),
(9, 'Location9', 'Address9, Vijayawada, Andhra Pradesh, India'),
(10, 'Location10', 'Address10, Puducherry, India');

INSERT INTO Payment (PaymentID, CourierID, LocationID, Amount, PaymentDate)
VALUES
(1, 1, 1, 20.00, '2024-03-09'),
(2, 2, 2, 15.00, '2024-03-10'),
(3, 3, 3, 30.00, '2024-03-11'),
(4, 4, 4, 25.00, '2024-03-12'),
(5, 5, 5, 40.00, '2024-03-13'),
(6, 6, 6, 35.00, '2024-03-14'),
(7, 7, 7, 45.00, '2024-03-15'),
(8, 8, 8, 55.00, '2024-03-16'),
(9, 9, 9, 50.00, '2024-03-17'),
(10, 10, 10, 60.00, '2024-03-18');


--  List all customers
SELECT * FROM User;

-- List all orders for a specific customer
SELECT * FROM Courier WHERE SenderName = 'Specific Customer Name' OR ReceiverName = 'Specific Customer Name';

-- List all couriers
SELECT * FROM Courier;

--  List all packages for a specific order
SELECT * FROM Courier WHERE OrderID = 'Specific Order ID';

--  List all deliveries for a specific courier
SELECT * FROM Courier WHERE CourierID = 'Specific Courier ID';

-- List all undelivered packages
SELECT * FROM Courier WHERE Status = 'Pending' OR Status = 'In Transit';

-- List all packages that are scheduled for delivery today
SELECT * FROM Courier WHERE DeliveryDate = CURDATE();

-- List all packages with a specific status
```

```
SELECT * FROM Courier WHERE Status = 'Specific Status';

--  Calculate the total number of packages for each courier
SELECT CourierID, COUNT(*) AS TotalPackages FROM Courier GROUP BY CourierID;

--  Find the average delivery time for each courier
SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS AvgDeliveryTime FROM Courier GROUP BY CourierID;

--  List all packages with a specific weight range
SELECT * FROM Courier WHERE Weight BETWEEN 'Lower Weight' AND 'Upper Weight';

--  Retrieve employees whose names contain 'John'
SELECT * FROM Employee WHERE Name LIKE '%John%';

-- Retrieve all courier records with payments greater than $50
SELECT Courier.*, Payment.Amount FROM Courier INNER JOIN Payment ON Courier.CourierID = Payment.CourierID WHERE Payment.Amount > 50.00;

-- Task 3: Group By, Aggregate Functions, Having, Order By, Where
-- Find the total number of couriers handled by each employee.
SELECT EmployeeID, COUNT(*) AS TotalCouriersHandled
FROM Courier
GROUP BY EmployeeID;

-- Calculate the total revenue generated by each location
SELECT LocationID, SUM(Amount) AS TotalRevenue
FROM Payment
GROUP BY LocationID;

-- Find the total number of couriers delivered to each location.
SELECT LocationID, COUNT(*) AS TotalCouriersDelivered
FROM Courier
GROUP BY LocationID;

-- Find the courier with the highest average delivery time:
SELECT CourierID, AVG(DATEDIFF(DeliveryDate, OrderDate)) AS AvgDeliveryTime
FROM Courier
GROUP BY CourierID
ORDER BY AvgDeliveryTime DESC
LIMIT 1;

-- Find Locations with Total Payments Less Than a Certain Amount
SELECT LocationID, SUM(Amount) AS TotalPayment
FROM Payment
GROUP BY LocationID
HAVING TotalPayment < 'Certain Amount';

-- Calculate Total Payments per Location
SELECT LocationID, SUM(Amount) AS TotalPayment
FROM Payment
GROUP BY LocationID;

-- Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID
```

```sql
= X):
SELECT Courier.*
FROM Courier
INNER JOIN Payment ON Courier.CourierID = Payment.CourierID
WHERE Payment.LocationID = 'X'
GROUP BY Courier.CourierID
HAVING SUM(Payment.Amount) > 1000;

-- Retrieve couriers who have received payments totaling more than $1000 after a certain date (PaymentDate > 'YYYY-MM-DD'):
SELECT Courier.*
FROM Courier
INNER JOIN Payment ON Courier.CourierID = Payment.CourierID
WHERE Payment.PaymentDate > 'YYYY-MM-DD'
GROUP BY Courier.CourierID
HAVING SUM(Payment.Amount) > 1000;

-- Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')
SELECT Payment.LocationID
FROM Payment
WHERE Payment.PaymentDate > 'YYYY-MM-DD'
GROUP BY Payment.LocationID
HAVING SUM(Payment.Amount) > 5000;

-- Task 4: Inner Join, Full Outer Join, Cross Join, Left Outer Join, Right Outer Join
-- Retrieve Payments with Courier Information
SELECT *
FROM Payment
INNER JOIN Courier ON Payment.CourierID = Courier.CourierID;

-- Retrieve Payments with Location Information
SELECT *
FROM Payment
INNER JOIN Location ON Payment.LocationID = Location.LocationID;

-- Retrieve Payments with Courier and Location Information
SELECT *
FROM Payment
INNER JOIN Courier ON Payment.CourierID = Courier.CourierID
INNER JOIN Location ON Payment.LocationID = Location.LocationID;

-- List all payments with courier details
SELECT Payment.*, Courier.*
FROM Payment
LEFT JOIN Courier ON Payment.CourierID = Courier.CourierID;

-- Total payments received for each courier
SELECT Courier.CourierID, SUM(Payment.Amount) AS TotalPayments
FROM Courier
LEFT JOIN Payment ON Courier.CourierID = Payment.CourierID
GROUP BY Courier.CourierID;

-- List payments made on a specific date
```

```sql
SELECT *
FROM Payment
WHERE Payment.PaymentDate = 'Specific Date';

-- Get Courier Information for Each Payment
SELECT Payment.*, Courier.*
FROM Payment
RIGHT JOIN Courier ON Payment.CourierID = Courier.CourierID;

-- Get Payment Details with Location
SELECT Payment.*, Location.*
FROM Payment
LEFT JOIN Location ON Payment.LocationID = Location.LocationID;

-- Calculating Total Payments for Each Courier
SELECT Courier.CourierID, SUM(Payment.Amount) AS TotalPayments
FROM Courier
LEFT JOIN Payment ON Courier.CourierID = Payment.CourierID
GROUP BY Courier.CourierID;

-- List Payments Within a Date Range
SELECT *
FROM Payment
WHERE Payment.PaymentDate BETWEEN 'Start Date' AND 'End Date';

-- Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side
SELECT *
FROM User
FULL OUTER JOIN Courier ON User.UserID = Courier.UserID;

-- Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side
SELECT *
FROM Courier
FULL OUTER JOIN CourierServices ON Courier.ServiceID = CourierServices.ServiceID;

-- Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side
SELECT *
FROM Employee
FULL OUTER JOIN Payment ON Employee.EmployeeID = Payment.EmployeeID;

-- List all users and all courier services, showing all possible combinations.
SELECT *
FROM User
CROSS JOIN CourierServices;

-- List all employees and all locations, showing all possible combinations:
SELECT *
FROM Employee
CROSS JOIN Location;

-- Retrieve a list of couriers and their corresponding sender information (if available)
```

```sql
SELECT Courier.*, User.Name AS SenderName, User.Address AS SenderAddress
FROM Courier
LEFT JOIN User ON Courier.SenderName = User.Name;

-- Retrieve a list of couriers and their corresponding receiver information (if available):
SELECT Courier.*, User.Name AS ReceiverName, User.Address AS ReceiverAddress
FROM Courier
LEFT JOIN User ON Courier.ReceiverName = User.Name;

-- Retrieve a list of couriers along with the courier service details (if available):
SELECT Courier.*, CourierServices.*
FROM Courier
LEFT JOIN CourierServices ON Courier.ServiceID = CourierServices.ServiceID;
```