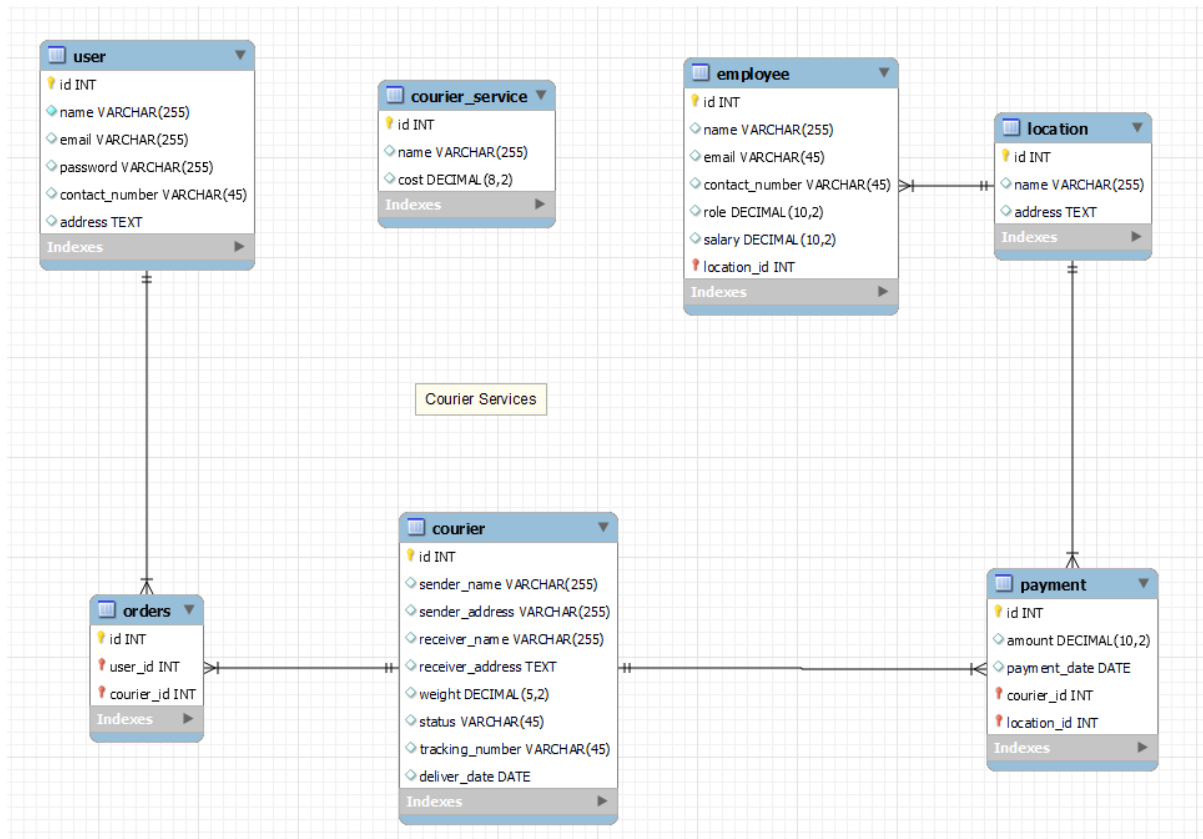


Courier Management



```
create database courierMang;
```

```
use courierMang;
```

```
show databases;
```

```
show tables;
```

```
CREATE TABLE Users (
    UserID INT PRIMARY KEY,
    name_ VARCHAR(255),
    Email VARCHAR(255) UNIQUE,
    Passkey VARCHAR(255),
    ContactNumber VARCHAR(20),
    Address varchar(255)
);
```

```
CREATE TABLE Couriers (  
    CourierID INT PRIMARY KEY,  
    SenderName VARCHAR(255),  
    SenderAddress varchar(255),  
    ReceiverName VARCHAR(255),  
    ReceiverAddress varchar(255),  
    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 Employees (  
    EmployeeID INT PRIMARY KEY,  
    name_ VARCHAR(255),  
    Email VARCHAR(255) UNIQUE,  
    ContactNumber VARCHAR(20),  
    Role_ VARCHAR(50),  
    Salary DECIMAL(10, 2)
```

);

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

```
CREATE TABLE Payments (  
    PaymentID INT PRIMARY KEY,  
    CourierID INT,  
    LocationID INT,  
    Amount DECIMAL(10, 2),  
    PaymentDate DATE,  
    FOREIGN KEY (CourierID) REFERENCES Couriers(CourierID),  
    FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)  
);
```

```
CREATE TABLE Orders (  
    OrderID INT PRIMARY KEY,  
    CustomerID INT,  
    OrderDate DATE,  
    CONSTRAINT FK_Orders_Customers FOREIGN KEY (CustomerID) REFERENCES  
Users(UserID)  
);
```

```

CREATE TABLE Parcels (
    ParcelID INT PRIMARY KEY,
    OrderID INT,
    CourierID INT,
    ServiceID INT,
    Weight DECIMAL(5, 2),
    Status_ VARCHAR(50),
    TrackingNumber VARCHAR(20) UNIQUE,
    DeliveryDate DATE,
    CONSTRAINT FK_Parcels_Orders FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    CONSTRAINT FK_Parcels_Couriers FOREIGN KEY (CourierID) REFERENCES
    Couriers(CourierID),
    CONSTRAINT FK_Parcels_Services FOREIGN KEY (ServiceID) REFERENCES
    CourierServices(ServiceID)
);

```

```

INSERT INTO Users (UserID, Name_, Email, Passkey, ContactNumber, Address)
VALUES(1, 'Rajesh Kumar', 'rajesh.kumar@email.com', 'password123', '9876543210', '12
Gandhi Nagar, Chennai'),
(2, 'Priya Sharma', 'priya.sharma@email.com', 'password456', '8765432109', '34 Kaveri
Street, Bangalore'),
(3, 'Amit Patel', 'amit.patel@email.com', 'password789', '7654321098', '56 Krishna Lane,
Hyderabad'),
(4, 'Ananya Singh', 'ananya.singh@email.com', 'passwordabc', '6543210987', '78 Vindhya
Nagar, Coimbatore');

```

```

INSERT INTO Couriers (CourierID, SenderName, SenderAddress, ReceiverName,
ReceiverAddress, Weight, Status_, TrackingNumber, DeliveryDate)
VALUES(1, 'Sender1', 'SenderAddress1', 'Receiver1', 'ReceiverAddress1', 2.5, 'In Transit',
'TN123456', '2024-03-01'),

```

```
(2, 'Sender2', 'SenderAddress2', 'Receiver2', 'ReceiverAddress2', 1.8, 'Delivered', 'TN789012', '2024-03-02'),
```

```
(3, 'Sender3', 'SenderAddress3', 'Receiver3', 'ReceiverAddress3', 3.0, 'In Transit', 'TN345678', '2024-03-03');
```

```
INSERT INTO CourierServices (ServiceID, ServiceName, Cost)
```

```
VALUES(1, 'Standard', 10.00),
```

```
        (2, 'Express', 15.00),
```

```
        (3, 'Same Day', 25.00),
```

```
        (4, 'Economy', 5.00),
```

```
        (5, 'International', 30.00),
```

```
        (6, 'Same Day', 20.00),
```

```
        (7, 'Bulk', 8.00);
```

```
INSERT INTO Employees (EmployeeID, Name_, Email, ContactNumber, Role_, Salary)
```

```
VALUES(1, 'Manager1', 'manager1@email.com', '1112223333', 'Manager', 50000.00),
```

```
(2, 'DeliveryPerson1', 'delivery1@email.com', '4445556666', 'Delivery Person', 30000.00),
```

```
(3, 'DeliveryPerson2', 'delivery2@email.com', '5556667777', 'Delivery Person', 30000.00),
```

```
(4, 'John cena', 'john.cena@email.com', '1234567890', 'Manager', 60000.00),
```

```
(5, 'Authur Johnson', 'authur.johnson@email.com', '9876543210', 'Clerk', 45000.00);
```

```
INSERT INTO Locations (LocationID, LocationName, Address)
```

```
VALUES(1, 'Warehouse1', '789 Storage St, Chennai'),
```

```
(2, 'Warehouse2', '456 Logistics Ave, Bangalore'),
```

```
(3, 'Warehouse3', '123 Distribution Rd, Hyderabad');
```

```
INSERT INTO Payments (PaymentID, CourierID, LocationID, Amount, PaymentDate)
```

```
VALUES(1, 1, 1, 10.00, '2024-03-03'),
```

```
(2, 2, 2, 15.00, '2024-03-04'),
(3, 3, 3, 12.50, '2024-03-05'),
(4, 3, 3, 50.50, '2024-03-05');

INSERT INTO Payments (PaymentID, CourierID, LocationID, Amount, PaymentDate)
VALUES(5, 1, 1, 1000.00, '2024-03-04');

INSERT INTO Payments (PaymentID, CourierID, LocationID, Amount, PaymentDate)
VALUES(6, 2, 2, 7000.00, '2024-03-05');
```

```
INSERT INTO Orders (OrderID, CustomerID, OrderDate)
VALUES(1, 1, '2024-03-01'),
(2, 2, '2024-03-02'),
(3, 3, '2024-03-03');
```

```
INSERT INTO Parcels (ParcelID, OrderID, CourierID, ServiceID, Weight, Status_,
TrackingNumber, DeliveryDate)
VALUES(1, 1, 1, 1, 2.5, 'In Transit', 'TN123456', '2024-03-01'),
(2, 2, 2, 2, 1.8, 'Delivered', 'TN789012', '2024-03-02'),
(3, 3, 3, 1, 3.0, 'In Transit', 'TN345678', '2024-03-03');
```

```
ALTER TABLE Parcels ADD EmployeeID int default 1;
ALTER TABLE Couriers ADD EmployeeID int default 1;
ALTER TABLE Couriers ADD LocationID int default 1;
ALTER TABLE Couriers ADD ServiceID int default 1;
ALTER TABLE Payments ADD EmployeeID int default 1;
```

-- Task 2--

-- 1. List all customers:

```
Select * from users;
```

-- 2. List all orders for a specific customer:

```
SELECT * FROM Orders WHERE CustomerID = 1;
```

-- 3. List all couriers:

```
Select * from couriers;
```

-- 4. List all packages for a specific order:

```
Select * from orders where orderid=3;
```

-- 5. List all deliveries for a specific courier:

```
SELECT * FROM Parcels WHERE CourierID = 1;
```

-- 6. List all undelivered packages:

```
Select * from parcels where not status_='delivered';
```

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

```
Select * from parcels where date(DeliveryDate)=date(now());
```

-- 8. List all packages with a specific status:

```
Select * from parcels where status_='delivered';
```

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

```
SELECT CourierID, COUNT(*) AS TotalPackages FROM Parcels GROUP BY CourierID;
```

-- doubt 10. Find the average delivery time for each courier

```
SELECT p.CourierID, AVG(DAY, o.OrderDate, p.DeliveryDate) AS AvgDeliveryTime  
FROM Parcels p JOIN Orders o ON p.OrderID = o.OrderID WHERE p.Status_ = 'Delivered'  
GROUP BY p.CourierID;
```

-- 11. List all packages with a specific weight range:

```
SELECT * FROM Parcels WHERE Weight BETWEEN 1.0 AND 2.0;
```

-- 12. Retrieve employees whose names contain 'John':

```
SELECT * FROM Employees WHERE Name_ LIKE '%John%';
```

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

```
SELECT * FROM Payments WHERE Amount > 50.00;
```

-- Task 3 --

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

```
SELECT e.EmployeeID, e.Name_, COUNT(p.CourierID) AS TotalCouriersHandled
FROM Employees e
LEFT JOIN Couriers p ON e.EmployeeID = p.EmployeeID
GROUP BY e.EmployeeID, e.Name_;
```

-- 15. Calculate the total revenue generated by each location

```
SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalRevenue
FROM Locations l
LEFT JOIN Payments p ON l.LocationID = p.LocationID
GROUP BY l.LocationID, l.LocationName;
```

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

```
Select l.locationid, l.address, sum(c.courierid) as total_couriers
from locations l left join couriers c on l.locationid=c.courierid
group by l.locationid, l.locationname;
```

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

```
SELECT p.courierid, MAX(avg(o.orderdate,p.deliverydate)) AS AverageDeliveryTime
```



```
FROM Packages p
JOIN CourierServices c ON p.CourierID = c.ServiceID
GROUP BY c.ServiceName
HAVING AverageDeliveryTime = MAX(AverageDeliveryTime);
```

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

```
select l.locationid, l.locationname, p.amount
from locations l left join payments p on l.locationid=p.locationid
where p.amount>10.00
group by l.locationid;
```

-- 19. Calculate Total Payments per Location

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

-- 20. Retrieve couriers who have received payments totaling more than \$1000 in a specific location (LocationID = X):

```
SELECT p.CourierID, c.SenderName, c.ReceiverName, p.LocationID, SUM(p.Amount) AS
TotalPayments
FROM Payments p
JOIN Couriers c ON p.CourierID = c.CourierID
WHERE p.LocationID = 1
GROUP BY p.CourierID, c.SenderName, c.ReceiverName, p.LocationID
HAVING SUM(p.Amount) > 1000;
```

-- 21. Retrieve couriers who have received payments totaling more than \$1000 after a certain date (PaymentDate > 'YYYY-MM-DD'):

```
SELECT p.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments
FROM Payments p
```

```
JOIN Couriers c ON p.CourierID = c.CourierID
WHERE p.PaymentDate > '2024-03-01'
GROUP BY p.CourierID, c.SenderName, c.ReceiverName
HAVING SUM(p.Amount) > 1000;
```

-- 22. Retrieve locations where the total amount received is more than \$5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

```
SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalAmountReceived
FROM Locations l
JOIN Payments p ON l.LocationID = p.LocationID
WHERE p.PaymentDate > '2024-03-01'
GROUP BY l.LocationID, l.LocationName
HAVING SUM(p.Amount) > 5000;
```

-- TASK 4 --

-- 23. Retrieve Payments with Courier Information

```
SELECT p.PaymentID, p.CourierID, p.LocationID, p.Amount, p.PaymentDate, c.SenderName,
c.SenderAddress, c.ReceiverName, c.ReceiverAddress, c.Weight, c.Status_,
c.TrackingNumber, c.DeliveryDate
FROM Payments p
inner JOIN Couriers c ON p.CourierID = c.CourierID;
```

-- 24. Retrieve Payments with Location Information

```
SELECT p.PaymentID, p.CourierID, p.LocationID, p.Amount, p.PaymentDate, l.LocationName,
l.Address
FROM Payments p
inner JOIN Locations l ON p.locationID =l.locationid ;
```

-- 25. Retrieve Payments with Courier and Location Information

```
select c.*,p.paymentid,p.amount,p.paymentdate,l.address
from couriers c join payments p on c.courierid=p.courierid
```

join locations l on l.locationid=p.locationid;

-- 26. List all payments with courier details

```
SELECT p.PaymentID, p.CourierID, p.LocationID, p.Amount, p.PaymentDate, c.SenderName,  
c.SenderAddress, c.ReceiverName, c.ReceiverAddress, c.Weight, c.Status_,  
c.TrackingNumber, c.DeliveryDate
```

```
FROM Payments p
```

```
inner JOIN couriers c ON p.courierid =c.courierid;
```

-- 27. Total payments received for each courier

```
SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments
```

```
FROM Couriers c
```

```
LEFT JOIN Payments p ON c.CourierID = p.CourierID
```

```
GROUP BY c.CourierID, c.SenderName, c.ReceiverName;
```

-- 28. List payments made on a specific date

```
SELECT *
```

```
FROM Payments
```

```
WHERE PaymentDate = '2024-03-05';
```

-- 29. Get Courier Information for Each Payment

```
SELECT p.*, c.*
```

```
FROM Payments p
```

```
left JOIN Couriers c ON p.CourierID = c.CourierID;
```

-- 30. Get Payment Details with Location

```
Select l.locationname,l.address,p.*
```

```
from payments p
```

```
inner join locations l on l.locationid=p.locationid;
```

-- 31. Calculating Total Payments for Each Courier

```
SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPayments
FROM Couriers c
LEFT JOIN Payments p ON c.CourierID = p.CourierID
GROUP BY c.CourierID, c.SenderName, c.ReceiverName;
```

-- 32. List Payments Within a Date Range

```
SELECT *
FROM Payments
WHERE PaymentDate BETWEEN '2024-03-01' AND '2024-03-03';
```

-- 33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

```
SELECT u.*, c.*
FROM Users u
FULL OUTER JOIN Couriers c ON u.name = c.SenderName;
```