```
CREATE DATABASE TechShop;
USE TechShop;
CREATE TABLE Customers (
   CustomerID INT PRIMARY KEY AUTO_INCREMENT,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Email VARCHAR(100),
   Phone VARCHAR(20),
   Address VARCHAR(255)
);
CREATE TABLE Products (
   ProductID INT PRIMARY KEY AUTO_INCREMENT,
   ProductName VARCHAR(100),
   Description TEXT,
   Price DECIMAL(10, 2)
);
CREATE TABLE Orders (
   OrderID INT PRIMARY KEY AUTO_INCREMENT,
   CustomerID INT,
   OrderDate DATE,
   TotalAmount DECIMAL(10, 2),
   FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
CREATE TABLE OrderDetails (
   OrderDetailID INT PRIMARY KEY AUTO_INCREMENT,
   OrderID INT,
   ProductID INT,
   Quantity INT,
   FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
   FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
```

```
);
CREATE TABLE Inventory (
   InventoryID INT PRIMARY KEY AUTO_INCREMENT,
   ProductID INT,
   QuantityInStock INT,
   LastStockUpdate DATE,
   FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address) VALUES
('John', 'Doe', 'john.doe@example.com', '1234567890', '123 Main St'),
('Jane', 'Smith', 'jane.smith@example.com', '2345678901', '456 Oak St'),
('Alice', 'Brown', 'alice.brown@example.com', '3456789012', '789 Pine St'),
('Bob', 'White', 'bob.white@example.com', '4567890123', '321 Cedar St'),
('Emma', 'Wilson', 'emma.wilson@example.com', '5678901234', '654 Maple St'),
('Liam', 'Taylor', 'liam.taylor@example.com', '6789012345', '987 Birch St'),
('Olivia', 'Anderson', 'olivia.anderson@example.com', '7890123456', '159 Walnut St'),
('Noah', 'Thomas', 'noah.thomas@example.com', '8901234567', '753 Cherry St'),
('Sophia', 'Martin', 'sophia.martin@example.com', '9012345678', '852 Willow St'),
('Mason', 'Lee', 'mason.lee@example.com', '0123456789', '951 Spruce St');
INSERT INTO Products (ProductName, Description, Price) VALUES
('Laptop', '14 inch, Intel i5, 8GB RAM', 74999),
('Smartphone', '6.5 inch, Android 12', 49999),
('Tablet', '10 inch display, 64GB storage', 29999),
('Headphones', 'Wireless, noise cancelling', 9999),
('Keyboard', 'Mechanical, RGB backlit', 5999),
('Mouse', 'Wireless ergonomic mouse', 3999),
('Monitor', '24 inch, 1080p Full HD', 14999),
('Webcam', '1080p USB webcam', 6999),
('Charger', '65W fast charger', 29.99),
('External HDD', '1TB USB 3.0', 89.99);
INSERT INTO Orders (CustomerID, OrderDate, TotalAmount) VALUES
```

```
(1, '2025-04-01', 849.98),
(2, '2025-04-02', 499.99),
(3, '2025-04-03', 159.98),
(4, '2025-04-04', 749.99),
(5, '2025-04-05', 329.98),
(6, '2025-04-06', 299.99),
(7, '2025-04-07', 189.98),
(8, '2025-04-08', 599.97),
(9, '2025-04-09', 459.97),
(10, '2025-04-10', 149.99);
INSERT INTO OrderDetails (OrderID, ProductID, Quantity) VALUES
(1, 1, 1),
(1, 4, 1),
(2, 2, 1),
(3, 5, 1),
(3, 6, 1),
(4, 1, 1),
(5, 3, 1),
(5, 9, 1),
(6, 3, 1),
(7, 4, 2),
(8, 1, 1),
(8, 7, 1),
(8, 10, 1),
(9, 2, 1),
(9, 6, 1),
(9, 8, 1),
(10, 7, 1);
INSERT INTO Inventory (ProductID, QuantityInStock, LastStockUpdate) VALUES
(1, 10, '2025-03-30'),
(2, 20, '2025-03-30'),
```

```
(3, 15, '2025-03-30'),
(4, 25, '2025-03-30'),
(5, 18, '2025-03-30'),
(6, 30, '2025-03-30'),
(7, 12, '2025-03-30'),
(8, 17, '2025-03-30'),
(9, 22, '2025-03-30'),
(10, 14, '2025-03-30');
Select FirstName,LastName,Email from Customers;
SELECT Orders.OrderID, Orders.OrderDate,Customers.FirstName,Customers.LastName
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID;
INSERT INTO Customers (FirstName, LastName, Email, Phone, Address)
VALUES ('Tom', 'Cruise', 'tom@example.com', '7894561230', '235 Main St');
UPDATE Products
SET Price = round( Price * 1.10,2)
WHERE ProductName IN ('Laptop', 'Smartphone', 'Tablet', 'Headphones');

DELETE FROM OrderDetails
WHERE OrderID = ?;
DELETE FROM Orders
WHERE OrderID = ?;

Drop database Courier;



-- Task 1

Create database CourierManag;
Use CourierManag;
-- user Table
```

```
Create table usertable(
userid int primary key AUTO_INCREMENT,
Name varchar(255) Not null,
Email varchar(255) Not null,
Pass varchar(255) Not null,
Contact varchar(255),
Address TEXT
);
-- Courier Table
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
);

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

-- Employee Table
CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
```

```
    Name VARCHAR(255),
    Email VARCHAR(255) UNIQUE,
    ContactNumber VARCHAR(20),
    Role VARCHAR(50),
    Salary DECIMAL(10, 2)
);

-- Location Table
CREATE TABLE Location (
    LocationID INT PRIMARY KEY,
    LocationName VARCHAR(100),
    Address TEXT
);
Drop table if exists Payment;
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT,
    OrderDate DATE NOT NULL,
    TotalAmount DECIMAL(10, 2),
    Status VARCHAR(50) DEFAULT 'Pending',
    FOREIGN KEY (UserID) REFERENCES usertable(userid)
);
CREATE TABLE Parcels (
    ParcelID INT PRIMARY KEY AUTO_INCREMENT,
    OrderID INT,
    Weight DECIMAL(5, 2), -- in kilograms
    Dimensions VARCHAR(100), -- e.g., "30x20x15 cm"
    ContentDescription TEXT,
    TrackingNumber VARCHAR(100) UNIQUE,
    DeliveryStatus VARCHAR(50) DEFAULT 'Not Shipped',
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)
```

```
);



-- Payment Table
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 usertable (Name, Email, Pass, Contact, Address) VALUES
('Alice Johnson', 'alice.j@example.com', 'alice123', '1234567890', '123 Maple Street'),
('Bob Smith', 'bob.smith@example.com', 'bobSecure!', '2345678901', '456 Oak Avenue'),
('Carol White', 'carol.w@example.com', 'carolPass$', '3456789012', '789 Pine Lane'),
('David Lee', 'david.lee@example.com', 'dlee!pass', '4567890123', '321 Cedar Road'),
('Eva Green', 'eva.green@example.com', 'greenPwd99', '5678901234', '654 Birch Blvd'),
('Frank Moore', 'frank.m@example.com', 'mooreSecure', '6789012345', '987 Elm Street'),
('Grace Kim', 'grace.kim@example.com', 'grace@123', '7890123456', '159 Willow Way'),
('Henry Clark', 'henry.clark@example.com', 'henry456!', '8901234567', '753 Cherry Court'),
('Isabel Young', 'isabel.y@example.com', 'iyoungPass!', '9012345678', '852 Spruce Terrace'),
('Jake Brown', 'jake.brown@example.com', 'jakeSecure#', '0123456789', '951 Redwood Drive');
INSERT INTO Courier (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress,
Weight, Status, TrackingNumber, DeliveryDate) VALUES
(1, 'John Doe', '123 Sender St, CityA', 'Alice Smith', '456 Receiver Ave, CityB', 2.50, 'In Transit',
'TRK100001', '2025-04-11'),
(2, 'Emily Clark', '789 Oak St, CityA', 'Liam Johnson', '101 Elm St, CityC', 1.20, 'Delivered',
'TRK100002', '2025-04-10'),
(3, 'Michael Brown', '456 Birch St, CityB', 'Sophia Green', '202 Pine St, CityD', 3.10, 'Pending',
'TRK100003', '2025-04-12'),
```

(4, 'Sarah Davis', '321 Spruce St, CityC', 'Noah Lee', '303 Cedar St, CityE', 0.90, 'In Transit', 'TRK100004', '2025-04-13'),

(5, 'James Wilson', '147 Willow St, CityD', 'Olivia Hall', '404 Cherry St, CityF', 5.00, 'Delivered', 'TRK100005', '2025-04-09'),

(6, 'Emma Turner', '258 Palm St, CityE', 'Lucas King', '505 Ash St, CityG', 2.70, 'Shipped', 'TRK100006', '2025-04-11'),

(7, 'William Scott', '369 Magnolia St, CityF', 'Ava Young', '606 Fir St, CityH', 4.00, 'Pending', 'TRK100007', '2025-04-12'),

(8, 'Isabella Adams', '741 Poplar St, CityG', 'Ethan Moore', '707 Redwood St, CityI', 1.50, 'In Transit', 'TRK100008', '2025-04-13'),

(9, 'Benjamin Cox', '852 Palm St, CityH', 'Mia Perez', '808 Sequoia St, CityJ', 3.60, 'Delivered', 'TRK100009', '2025-04-10'),

(10, 'Charlotte Evans', '963 Aspen St, CityI', 'Logan Brooks', '909 Beech St, CityA', 2.30, 'Shipped', 'TRK100010', '2025-04-11');

INSERT INTO CourierServices (ServiceID, ServiceName, Cost) VALUES

(1, 'Standard Delivery', 10.00),

(2, 'Express Delivery', 25.00),

(3, 'Overnight Delivery', 40.00),

(4, 'Same Day Delivery', 55.00),

(5, 'Weekend Delivery', 30.00),

(6, 'Holiday Delivery', 35.00),

(7, 'Return Pickup', 15.00),

(8, 'Fragile Item Handling', 20.00),

(9, 'International Delivery', 75.00),

(10, 'Eco-Friendly Shipping', 12.00);

INSERT INTO Employee (EmployeeID, Name, Email, ContactNumber, Role, Salary) VALUES

(1, 'Emily Carter', 'emily.carter@example.com', '1234567890', 'Manager', 55000.00),

(2, 'Tom Wilson', 'tom.wilson@example.com', '0987654321', 'Courier', 35000.00),

(3, 'Nina Patel', 'nina.patel@example.com', '2345678901', 'Dispatcher', 40000.00),

(4, 'Jake Miller', 'jake.miller@example.com', '3456789012', 'Courier', 32000.00),

(5, 'Grace Lee', 'grace.lee@example.com', '4567890123', 'Support', 37000.00),

(6, 'Sam Morgan', 'sam.morgan@example.com', '5678901234', 'Courier', 33000.00),

(7, 'Rachel Kim', 'rachel.kim@example.com', '6789012345', 'Manager', 58000.00),

(8, 'Chris Bell', 'chris.bell@example.com', '7890123456', 'Support', 36000.00),

(9, 'Sophie Reed', 'sophie.reed@example.com', '8901234567', 'Courier', 34000.00),

(10, 'Daniel West', 'daniel.west@example.com', '9012345678', 'Dispatcher', 41000.00);

INSERT INTO Orders (UserID, OrderDate, TotalAmount, Status) VALUES

(1, '2025-04-01', 45.00, 'Pending'),

(2, '2025-04-02', 60.50, 'Shipped'),

(3, '2025-04-03', 30.75, 'Delivered'),

(4, '2025-04-04', 85.20, 'In Transit'),

(5, '2025-04-05', 49.99, 'Delivered'),

(6, '2025-04-06', 100.00, 'Pending'),

(7, '2025-04-07', 75.30, 'Shipped'),

(8, '2025-04-08', 59.99, 'In Transit'),

(9, '2025-04-09', 39.80, 'Pending'),

(10, '2025-04-10', 89.90, 'Delivered');

INSERT INTO Parcels (OrderID, Weight, Dimensions, ContentDescription, TrackingNumber, DeliveryStatus) VALUES

(1, 2.5, '30x20x15 cm', 'Books', 'TRK001001', 'Not Shipped'),

(2, 1.2, '25x15x10 cm', 'Phone accessories', 'TRK001002', 'Shipped'),

(3, 0.8, '20x10x5 cm', 'Wristwatch', 'TRK001003', 'Delivered'),

(4, 3.1, '40x25x20 cm', 'Clothes', 'TRK001004', 'In Transit'),

(5, 1.5, '28x18x12 cm', 'Bluetooth speaker', 'TRK001005', 'Delivered'),

(6, 4.0, '45x30x20 cm', 'Shoes', 'TRK001006', 'Not Shipped'),

(7, 2.2, '35x25x15 cm', 'Gadgets', 'TRK001007', 'Shipped'),

(8, 1.8, '32x20x18 cm', 'Gaming mouse', 'TRK001008', 'In Transit'),

(9, 0.9, '22x12x8 cm', 'Notebook', 'TRK001009', 'Not Shipped'),

(10, 2.7, '33x22x17 cm', 'Tablet', 'TRK001010', 'Delivered');

INSERT INTO Location (LocationID, LocationName, Address) VALUES

(1, 'CityA Hub', '123 Logistics Rd, CityA'),

(2, 'CityB Hub', '456 Transport Blvd, CityB'),

(3, 'CityC Hub', '789 Delivery Ln, CityC'),

```
(4, 'CityD Hub', '321 Shipping Way, CityD'),
(5, 'CityE Hub', '147 Courier Rd, CityE'),
(6, 'CityF Hub', '258 Freight Ave, CityF'),
(7, 'CityG Hub', '369 Parcel St, CityG'),
(8, 'CityH Hub', '741 Express Blvd, CityH'),
(9, 'CityI Hub', '852 Ground St, CityI'),
(10, 'CityJ Hub', '963 Swift Rd, CityJ');
INSERT INTO Payment (PaymentID, CourierID, LocationID, Amount, PaymentDate) VALUES
(1, 1, 1, 25.00, '2025-04-11'),
(2, 2, 2, 40.00, '2025-04-10'),
(3, 3, 3, 15.00, '2025-04-12'),
(4, 4, 4, 20.00, '2025-04-13'),
(5, 5, 5, 30.00, '2025-04-09'),
(6, 6, 6, 25.00, '2025-04-11'),
(7, 7, 7, 35.00, '2025-04-12'),
(8, 8, 8, 18.00, '2025-04-13'),
(9, 9, 9, 40.00, '2025-04-10'),
(10, 10, 10, 22.00, '2025-04-11');
Update Payment set Amount='78.00' where PaymentID='3';

-- Task 2

-- 1.List all customers
Select * from usertable;

-- 2.List all orders for a specific customer
Select * from Orders where UserId=1 ;

-- 3.List all couriers
Select * from Courier;
```

```
-- 4.List all packages for a specific order
Select * from Parcels where orderid='8';

-- 5.List all deliveries for a specific courier
SELECT * FROM Courier WHERE CourierID = 1; -- Replace with CourierID

-- 6.List all undelivered packages
SELECT * FROM Courier WHERE Status != 'Delivered';

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

-- 8.List all packages with a specific status
SELECT * FROM Courier WHERE Status = 'In Transit';

-- 9.Calculate the total number of packages for each courier
Select OrderId,Count(*) As Total_Package from Parcels Group by OrderId;

-- 10.Find the average delivery time for each courier
Alter table Courier Add ShippingDate Date;
UPDATE Courier SET ShippingDate = '2025-04-01' WHERE CourierID = 1;
UPDATE Courier SET ShippingDate = '2025-04-02' WHERE CourierID = 2;
UPDATE Courier SET ShippingDate = '2025-04-03' WHERE CourierID = 3;
UPDATE Courier SET ShippingDate = '2025-04-04' WHERE CourierID = 4;
UPDATE Courier SET ShippingDate = '2025-04-05' WHERE CourierID = 5;
UPDATE Courier SET ShippingDate = '2025-04-06' WHERE CourierID = 6;
UPDATE Courier SET ShippingDate = '2025-04-07' WHERE CourierID = 7;
UPDATE Courier SET ShippingDate = '2025-04-08' WHERE CourierID = 8;
UPDATE Courier SET ShippingDate = '2025-04-09' WHERE CourierID = 9;
UPDATE Courier SET ShippingDate = '2025-04-10' WHERE CourierID = 10;
SELECT CourierID,
```

```
AVG(DATEDIFF(DeliveryDate, ShippingDate)) As Days
FROM Courier Group By CourierID;

-- 11.List all packages with a specific weight range
Select CourierID from Courier where Weight between '2.00' and '5.00';

-- 12.Retrieve employees whose names contain 'John'
Select Name from Employee where Name Like 'John%';

-- 13.Retrieve all courier records with payments greater than $50
Select * from Payment where Amount > 50  ;

-- Task 3

-- 14.Total number of couriers handled by each employee
Select EmployeeID,COunt(*) As No_of_Package
From Employee
group by EmployeeID;

-- 15.Total revenue generated by each location
Select LocationID,Sum(Amount) As Total_Revenue
from Payment
Group By LocationID;

-- 16.Total number of couriers delivered to each location
SELECT LocationID, COUNT(*) AS DeliveredCount
FROM Courier
WHERE Status = 'Delivered'
GROUP BY LocationID;

-- 17.Courier with highest average delivery time
```

```
Select CourierID,Avg(datediff(DeliveryDate,ShippingDate)) As No_C
from Courier
Group By CourierID
Order By No_C Desc
Limit 1;

-- 18. Locations with total payments < a certain amount (e.g., $1000)
Select LocationID,Sum(Amount) As Total_Pay
from Payment
group by LocationID
having Sum(Amount) < 1000;

-- 19.Calculate total payments per location
Select LocationID,Sum(Amount) As Total_Amount
from Payment
Group By LocationID;

-- 20.Couriers with total payments > $1000 in a specific location
Select CourierID,SUm(Amount) As T_A
from Payment
Where LocationID=5
Group By CourierID
HAving Sum(Amount)>10000;

-- 21.Couriers with payments > $1000 after a specific date
Select CourierID,SUm(Amount) As T_A
from Payment
Where PaymentDate>'2025/04/02'
Group By CourierID
HAving Sum(Amount)>10000;
```

```
-- 22.Locations where total amount received > $5000 before a date
Select CourierID,SUm(Amount) As T_A
from Payment
Where PaymentDate<'2025/04/02'
Group By CourierID
HAving Sum(Amount)>5000;

-- 23.Retrieve Payments with Courier Information
SELECT
p.PaymentID,p.CourierID,c.SenderName,c.ReceiverName,c.Weight,c.Status,p.Amount,p.PaymentDat
e
FROM Payment p
JOIN Courier c ON p.CourierID = c.CourierID;

-- 24.Retrieve Payments with Location Information
SELECT p.PaymentID,p.LocationID,l.LocationName,l.Address,p.Amount,p.PaymentDate
FROM Payment p
JOIN Location l ON p.LocationID = l.LocationID;

-- 25.Retrieve Payments with Courier and Location Information
SELECT Payment.PaymentID, Courier.CourierID, Location.LocationID
FROM Payment
JOIN Courier ON Payment.CourierID = Courier.CourierID
JOIN Location ON Payment.LocationID = Location.LocationID;

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

-- 27.Total payments received for each courier
SELECT Payment.CourierID, SUM(Payment.Amount) AS TotalPayments
FROM Payment
```

```
GROUP BY Payment.CourierID;

-- 28.List payments made on a specific date
SELECT * FROM Payment
WHERE PaymentDate = '2025-04-10';

-- 29.Get Courier Information for Each Payment
SELECT Payment.PaymentID,Payment.Amount, Courier.CourierID
FROM Payment JOIN Courier ON Payment.CourierID = Courier.CourierID;

-- 30.Get Payment Details with Location
SELECT Payment.PaymentID, Payment.Amount, Location.LocationID
FROM Payment JOIN Location ON Payment.LocationID = Location.LocationID;

-- 31.Calculating Total Payments for Each Courier
SELECT CourierID, SUM(Amount) AS TotalAmount
FROM Payment
GROUP BY CourierID;

-- 32.List Payments Within a Date Range
SELECT * FROM Payment
WHERE PaymentDate BETWEEN '2025-04-08' AND '2025-04-12';

-- 33. Retrieve a list of all users and their corresponding courier records, including cases where there are
SELECT u.Name, c.TrackingNumber, c.Status
FROM User u
LEFT JOIN Courier c ON u.Name = c.SenderName
WHERE u.Name IS NOT NULL OR c.SenderName IS NOT NULL;

-- 34.All couriers and their corresponding services
```

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

-- 35.All employees and their corresponding payments
SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount
FROM Employee e
LEFT JOIN Payment p ON e.EmployeeID = p.EmployeeID;

-- 36.List all users and all courier services, showing all possible combinations
SELECT u.Name AS UserName, cs.ServiceName
FROM User u, CourierServices cs;

-- 37.All employees and all locations
SELECT e.Name AS EmployeeName, l.LocationName
FROM Employee e, Location l;

-- 38.Couriers and their sender information
SELECT c.CourierID, c.TrackingNumber, u.Name AS SenderName, u.Address
FROM Courier c
JOIN User u ON c.SenderName = u.Name;

-- 39.Couriers and their receiver information
SELECT c.CourierID, c.TrackingNumber, c.ReceiverName
FROM Courier c;

-- 40.Couriers with Courier Service Details
SELECT c.*, cs.ServiceName, cs.Cost
FROM Courier c
LEFT JOIN CourierServices cs ON c.CourierID = cs.ServiceID;
```

```
-- 41.Employees and Number of Couriers Assigned
SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS CourierCount
FROM Employee e
LEFT JOIN Courier c ON e.EmployeeID = c.EmployeeID
GROUP BY e.EmployeeID, e.Name;

-- 42. Locations and Total Payments Received
Select Location.LocationID,Location.LocationName,Sum(Payment.Amount) As Pay
from Location
Inner join Payment on Location.LocationID=Payment.LocationID
Group by Location.LocationName,Location.LocationID;

-- 43.Couriers Sent by the Same Sender
SELECT *
FROM Courier
WHERE SenderName IN (
   SELECT SenderName
   FROM Courier
   GROUP BY SenderName
   HAVING COUNT(*) > 1
);

-- 44.Employees Who Share the Same Role
SELECT *
FROM Employee
WHERE Role IN (
   SELECT Role
   FROM Employee
   GROUP BY Role
   HAVING COUNT(*) > 1
);
```

```
-- 45.Payments for Couriers from the Same Location (based on SenderAddress)
SELECT p.*
FROM Payment p
JOIN Courier c ON p.CourierID = c.CourierID
WHERE c.SenderAddress IN (
   SELECT SenderAddress
   FROM Courier
   GROUP BY SenderAddress
   HAVING COUNT(*) > 1
);

-- 46.Couriers Sent from the Same Location (SenderAddress)
SELECT *
FROM Courier
WHERE SenderAddress IN (
   SELECT SenderAddress
   FROM Courier
   GROUP BY SenderAddress
   HAVING COUNT(*) > 1
);

-- 47.Employees and the Number of Couriers Delivered
Select Employee.EmployeeID,Employee.Name,COunt(Courier.CourierID)
from Employee
Join Courier  on  Employee.EmployeeID=Courier.CourierID
Group by Employee.EmployeeID,Employee.Name;

-- 48.Couriers Paid More Than Their Service Cost
SELECT c.CourierID, p.Amount AS PaymentAmount, cs.Cost AS ServiceCost
FROM Courier c
```

```
JOIN Payment p ON c.CourierID = p.CourierID
JOIN CourierServices cs ON c.CourierID = cs.ServiceID
WHERE p.Amount > cs.Cost;

-- Scope

-- 49.Couriers with weight greater than the average weight
Select *
from courier
where Weight>(Select avg(Weight) From Courier);

-- 50.Employees with salary greater than the average salary
SELECT Name, Salary
FROM Employee
WHERE Salary > (SELECT AVG(Salary) FROM Employee);

-- 51.Total cost of courier services with cost less than the maximum
SELECT SUM(Cost) AS TotalCost
FROM CourierServices
WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);

-- 52.All couriers that have been paid for
SELECT DISTINCT c.*
FROM Courier c
JOIN Payment p ON c.CourierID = p.CourierID;

-- 53.Locations where the maximum payment amount was made
SELECT l.LocationID, l.LocationName, p.Amount
FROM Payment p
JOIN Location l ON p.LocationID = l.LocationID
WHERE p.Amount = (SELECT MAX(Amount) FROM Payment);
```

```
-- 54.Couriers whose weight is greater than all couriers sent by a specific sender (e.g., 'John Doe')
SELECT *
FROM Courier
WHERE Weight > ALL (
   SELECT Weight
   FROM Courier
   WHERE SenderName = 'John Doe'
);
```