

TASK 1

-- Create Database

```
CREATE DATABASE IF NOT EXISTS CourierManagementSystem;
```

```
USE CourierManagementSystem;
```

-- Create User Table

```
CREATE TABLE IF NOT EXISTS User (
```

```
    UserID INT PRIMARY KEY,
```

```
    Name VARCHAR(255),
```

```
    Email VARCHAR(255) UNIQUE,
```

```
    Password VARCHAR(255),
```

```
    ContactNumber VARCHAR(20),
```

```
    Address TEXT
```

```
);
```

-- Create Courier Table

```
CREATE TABLE IF NOT EXISTS 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 CourierServices Table

```
CREATE TABLE IF NOT EXISTS CourierServices (  
    ServiceID INT PRIMARY KEY,  
    ServiceName VARCHAR(100),  
    Cost DECIMAL(8, 2)  
);
```

-- Create Employee Table

```
CREATE TABLE IF NOT EXISTS Employee (  
    EmployeeID INT PRIMARY KEY,  
    Name VARCHAR(255),  
    Email VARCHAR(255) UNIQUE,  
    ContactNumber VARCHAR(20),  
    Role VARCHAR(50),  
    Salary DECIMAL(10, 2)  
);
```

-- Create Location Table

```
CREATE TABLE IF NOT EXISTS Location (  
    LocationID INT PRIMARY KEY,  
    LocationName VARCHAR(100),  
    Address TEXT  
);
```

-- Create Payment Table

```
CREATE TABLE IF NOT EXISTS 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 table

INSERT INTO User (UserID, Name, Email, Password, ContactNumber, Address)

VALUES

(1, 'John Doe', 'john.doe@example.com', 'password123', '1234567890', '123 Main St, Mumbai'),

(2, 'Jane Smith', 'jane.smith@example.com', 'password456', '9876543210', '456 Street Ave, Delhi'),

(3, 'Amit Kumar', 'amit.kumar@example.com', 'password789', '8765432109', '789 Lane Rd, Bangalore'),

(4, 'Preeti Sharma', 'preeti.sharma@example.com', 'passwordabc', '7654321098', '321 Garden St, Kolkata'),

(5, 'Raj Patel', 'raj.patel@example.com', 'passworddef', '6543210987', '456 Hill View, Chennai'),

(6, 'Sneha Verma', 'sneha.verma@example.com', 'passwordghi', '5432109876', '789 Lake View, Hyderabad'),

(7, 'Vijay Singh', 'vijay.singh@example.com', 'passwordjkl', '4321098765', '987 Skyline Ave, Pune'),

(8, 'Anita Das', 'anita.das@example.com', 'passwordmno', '3210987654', '654 Beach Rd, Jaipur'),

(9, 'Rohit Mehta', 'rohit.mehta@example.com', 'passwordpqr', '2109876543', '987 Forest Lane, Ahmedabad'),

(10, 'Priya Kapoor', 'priya.kapoor@example.com', 'passwordstu', '1098765432', '123 Park View, Chandigarh');

-- Insert into Courier table

INSERT INTO Courier (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate)

VALUES

(1, 'Sender A', '456 Sender St, Kolkata', 'Receiver B', '789 Receiver St, Bangalore', 2.5, 'In Transit', 'TRK123', '2023-01-15'),

(2, 'Sender C', '123 Sender St, Chennai', 'Receiver D', '987 Receiver St, Mumbai', 1.8, 'Delivered', 'TRK456', '2023-01-16'),

(3, 'Sender E', '789 Sender St, Delhi', 'Receiver F', '654 Receiver St, Hyderabad', 3.2, 'Pending', 'TRK789', '2023-01-17'),

(4, 'Sender G', '321 Sender St, Pune', 'Receiver H', '876 Receiver St, Jaipur', 1.5, 'In Transit', 'TRK101', '2023-01-18'),

(5, 'Sender I', '987 Sender St, Ahmedabad', 'Receiver J', '543 Receiver St, Chandigarh', 2.0, 'Delivered', 'TRK111', '2023-01-19'),

(6, 'Sender K', '654 Sender St, Mumbai', 'Receiver L', '210 Receiver St, Kolkata', 2.8, 'Pending', 'TRK121', '2023-01-20'),

(7, 'Sender M', '210 Sender St, Delhi', 'Receiver N', '321 Receiver St, Bangalore', 1.3, 'Delivered', 'TRK131', '2023-01-21'),

(8, 'Sender O', '876 Sender St, Chennai', 'Receiver P', '432 Receiver St, Pune', 2.1, 'In Transit', 'TRK141', '2023-01-22'),

(9, 'Sender Q', '543 Sender St, Hyderabad', 'Receiver R', '109 Receiver St, Jaipur', 1.7, 'Delivered', 'TRK151', '2023-01-23'),

(10, 'Sender S', '432 Sender St, Ahmedabad', 'Receiver T', '765 Receiver St, Chandigarh', 2.4, 'Pending', 'TRK161', '2023-01-24');

-- Insert into CourierServices table

INSERT INTO CourierServices (ServiceID, ServiceName, Cost)

VALUES

(1, 'Express', 15.99),

(2, 'Standard', 10.99),

(3, 'Economy', 8.99),

(4, 'Priority', 18.99),

(5, 'Same Day', 25.99),

(6, 'Next Day', 12.99),

(7, 'Two-Day', 14.99),

(8, 'Overnight', 30.99),

(9, 'Regular', 9.99),

(10, 'Super Saver', 5.99);

-- Insert into Employee table

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

VALUES

(1, 'Employee 1', 'employee1@example.com', '9876543210', 'Courier Handler', 3000.00),

(2, 'Employee 2', 'employee2@example.com', '8765432109', 'Manager', 5000.00),

(3, 'Employee 3', 'employee3@example.com', '7654321098', 'Driver', 4000.00),

```
(4, 'Employee 4', 'employee4@example.com', '6543210987', 'Warehouse Staff', 3500.00),
(5, 'Employee 5', 'employee5@example.com', '5432109876', 'Customer Service', 4500.00),
(6, 'Employee 6', 'employee6@example.com', '4321098765', 'Supervisor', 5500.00),
(7, 'Employee 7', 'employee7@example.com', '3210987654', 'IT Support', 4800.00),
(8, 'Employee 8', 'employee8@example.com', '2109876543', 'Accountant', 5200.00),
(9, 'Employee 9', 'employee9@example.com', '1098765432', 'Security', 3800.00),
(10, 'Employee 10', 'employee10@example.com', '9876543210', 'HR Manager', 6000.00);
```

-- Insert into Location table

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

VALUES

```
(1, 'Warehouse A', '789 Warehouse St, Mumbai'),
(2, 'Warehouse B', '456 Warehouse St, Delhi'),
(3, 'Warehouse C', '123 Warehouse St, Bangalore'),
(4, 'Warehouse D', '987 Warehouse St, Kolkata'),
(5, 'Warehouse E', '654 Warehouse St, Chennai'),
(6, 'Warehouse F', '210 Warehouse St, Hyderabad'),
(7, 'Warehouse G', '876 Warehouse St, Pune'),
(8, 'Warehouse H', '543 Warehouse St, Jaipur'),
(9, 'Warehouse I', '432 Warehouse St, Ahmedabad'),
(10, 'Warehouse J', '321 Warehouse St, Chandigarh');
```

-- Insert into Payment table

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

VALUES

```
(1, 1, 1, 20.50, '2023-01-20'),
(2, 2, 2, 15.75, '2023-01-22'),
(3, 3, 3, 12.00, '2023-01-25'),
(4, 4, 4, 18.25, '2023-01-28'),
(5, 5, 5, 22.80, '2023-01-30'),
(6, 6, 6, 14.50, '2023-02-02'),
```

(7, 7, 7, 17.30, '2023-02-05'),
(8, 8, 8, 21.00, '2023-02-08'),
(9, 9, 9, 10.75, '2023-02-11'),
(10, 10, 10, 25.20, '2023-02-14');

Task 2:

-- 1. List all customers:

SELECT * FROM User;

1	John Doe	john.doe@example.com	password123	1234567890	123 Main St, Mumbai
2	Jane Smith	jane.smith@example.com	password456	9876543210	456 Street Ave, Delhi
3	Amit Kumar	amit.kumar@example.com	password789	8765432109	789 Lane Rd, Bangalore
4	Preeti Sharma	preeti.sharma@example.com	passwordabc	7654321098	321 Garden St, Kolkata
5	Raj Patel	raj.patel@example.com	passworddef	6543210987	456 Hill View, Chennai
6	Sneha Verma	sneha.verma@example.com	passwordghi	5432109876	789 Lake View, Hyderabad
7	Vijay Singh	vijay.singh@example.com	passwordjkl	4321098765	987 Skyline Ave, Pune
8	Anita Das	anita.das@example.com	passwordmno	3210987654	654 Beach Rd, Jaipur
9	Rohit Mehta	rohit.mehta@example.com	passwordpqr	2109876543	987 Forest Lane, Ahmedabad
10	Priya Kapoor	priya.kapoor@example.com	passwordstu	1098765432	123 Park View, Chandigarh

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

SELECT * FROM Courier WHERE SenderName = 'CustomerName';

1	Sender A	456 Sender St, Kolkata	Receiver B	789 Receiver St, Bangalore	2.50	In Transit	TRK123	2023- 01-15
---	-------------	---------------------------	---------------	-------------------------------	------	---------------	--------	----------------

-- 3. List all couriers:

SELECT * FROM Courier;

1	Sender A	456 Sender St, Kolkata	Receiver B	789 Receiver St, Bangalore	2.50 In Transit TRK123	2023-01-15
2	Sender C	123 Sender St, Chennai	Receiver D	987 Receiver St, Mumbai	1.80 Delivered TRK456	2023-01-16
3	Sender E	789 Sender St, Delhi	Receiver F	654 Receiver St, Hyderabad	3.20 Pending TRK789	2023-01-17
4	Sender G	321 Sender St, Pune	Receiver H	876 Receiver St, Jaipur	1.50 In Transit TRK101	2023-01-18
5	Sender I	987 Sender St, Ahmedabad	Receiver J	543 Receiver St, Chandigarh	2.00 Delivered TRK111	2023-01-19
6	Sender K	654 Sender St, Mumbai	Receiver L	210 Receiver St, Kolkata	2.80 Pending TRK121	2023-01-20
7	Sender M	210 Sender St, Delhi	Receiver N	321 Receiver St, Bangalore	1.30 Delivered TRK131	2023-01-21
8	Sender O	876 Sender St, Chennai	Receiver P	432 Receiver St, Pune	2.10 In Transit TRK141	2023-01-22
9	Sender Q	543 Sender St, Hyderabad	Receiver R	109 Receiver St, Jaipur	1.70 Delivered TRK151	2023-01-23
10	Sender S	432 Sender St, Ahmedabad	Receiver T	765 Receiver St, Chandigarh	2.40 Pending TRK161	2023-01-24

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

SELECT * FROM Courier WHERE CourierID IN (SELECT CourierID FROM Courier WHERE CourierID = '2');

2	Sender C	123 Sender St, Chennai	Receiver D	987 Receiver St, Mumbai	1.80 Delivered TRK456	2023-01-16
---	-------------	---------------------------	---------------	----------------------------	-----------------------	------------

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

SELECT * FROM Courier WHERE CourierID = 'CourierID' AND Status = 'Delivered';

5	Sender I	987 Sender St, Ahmedabad	Receiver J	543 Receiver St, Chandigarh	2.00 Delivered TRK111	2023-01-19
---	-------------	-----------------------------	---------------	--------------------------------	-----------------------	------------

-- 6. List all undelivered packages:

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

1	Sender A	456 Sender St, Kolkata	Receiver B	789 Receiver St, Bangalore	2.50	In Transit	TRK123	2023-01-15
3	Sender E	789 Sender St, Delhi	Receiver F	654 Receiver St, Hyderabad	3.20	Pending	TRK789	2023-01-17
4	Sender G	321 Sender St, Pune	Receiver H	876 Receiver St, Jaipur	1.50	In Transit	TRK101	2023-01-18
6	Sender K	654 Sender St, Mumbai	Receiver L	210 Receiver St, Kolkata	2.80	Pending	TRK121	2023-01-20
8	Sender O	876 Sender St, Chennai	Receiver P	432 Receiver St, Pune	2.10	In Transit	TRK141	2023-01-22
10	Sender S	432 Sender St, Ahmedabad	Receiver T	765 Receiver St, Chandigarh	2.40	Pending	TRK161	2023-01-24

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

```
SELECT * FROM Courier WHERE DeliveryDate = CURRENT_DATE;
```

No output

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

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

3	Sender E	789 Sender St, Delhi	Receiver F	654 Receiver St, Hyderabad	3.20	Pending	TRK789	2023-01-17
6	Sender K	654 Sender St, Mumbai	Receiver L	210 Receiver St, Kolkata	2.80	Pending	TRK121	2023-01-20
10	Sender S	432 Sender St, Ahmedabad	Receiver T	765 Receiver St, Chandigarh	2.40	Pending	TRK161	2023-01-24

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

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

1	1
2	1
3	1
4	1

5 1
6 1
7 1
8 1
9 1
10 1

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

```
SELECT CourierID, AVG(DeliveryTime) AS AvgDeliveryTime FROM Courier GROUP BY CourierID;
```

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

```
SELECT * FROM Parcel WHERE Weight BETWEEN MinWeight AND MaxWeight;
```

1	Sender A	456 Sender St, Kolkata	Receiver B	789 Receiver St, Bangalore	2.50 In Transit	TRK123	2023-01-15
2	Sender C	123 Sender St, Chennai	Receiver D	987 Receiver St, Mumbai	1.80 Delivered	TRK456	2023-01-16
4	Sender G	321 Sender St, Pune	Receiver H	876 Receiver St, Jaipur	1.50 In Transit	TRK101	2023-01-18
5	Sender I	987 Sender St, Ahmedabad	Receiver J	543 Receiver St, Chandigarh	2.00 Delivered	TRK111	2023-01-19
6	Sender K	654 Sender St, Mumbai	Receiver L	210 Receiver St, Kolkata	2.80 Pending	TRK121	2023-01-20
7	Sender M	210 Sender St, Delhi	Receiver N	321 Receiver St, Bangalore	1.30 Delivered	TRK131	2023-01-21
8	Sender O	876 Sender St, Chennai	Receiver P	432 Receiver St, Pune	2.10 In Transit	TRK141	2023-01-22
9	Sender Q	543 Sender St, Hyderabad	Receiver R	109 Receiver St, Jaipur	1.70 Delivered	TRK151	2023-01-23
10	Sender S	432 Sender St, Ahmedabad	Receiver T	765 Receiver St, Chandigarh	2.40 Pending	TRK161	2023-01-24

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

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

Empty table

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

```
SELECT * FROM Payment WHERE Amount > 50;
```

Empty table

Task 3

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

```
SELECT EmployeeID, COUNT(*) AS TotalCouriers FROM Courier GROUP BY EmployeeID;
```

-- 2. Calculate the total revenue generated by each location:

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

```
1  20.50
2  15.75
3  12.00
4  18.25
5  22.80
6  14.50
7  17.30
8  21.00
9  10.75
10 25.20
```

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

```
SELECT LocationID, COUNT(*) AS TotalCouriersDelivered FROM Courier WHERE Status = 'Delivered'
GROUP BY LocationID;
```

-- 4. Find the courier with the highest average delivery time:

```
SELECT CourierID, AVG(DeliveryTime) AS AvgDeliveryTime FROM Courier GROUP BY CourierID ORDER
BY AvgDeliveryTime DESC LIMIT 1;
```

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

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

1	20.50
2	15.75
3	12.00
4	18.25
5	22.80
6	14.50
7	17.30
8	21.00
9	10.75
10	25.20

-- 6. Calculate Total Payments per Location:

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

1	20.50
2	15.75
3	12.00
4	18.25
5	22.80
6	14.50
7	17.30
8	21.00
9	10.75
10	25.20

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

```
SELECT CourierID FROM Payment WHERE LocationID = 4 GROUP BY CourierID HAVING SUM(Amount)  
> 10;
```

4

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

```
SELECT CourierID FROM Payment WHERE PaymentDate > 'YYYY-MM-DD' GROUP BY CourierID  
HAVING SUM(Amount) > 1000;
```

4
5
6
7
8
9
10

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

```
SELECT LocationID FROM Payment WHERE PaymentDate > 'YYYY-MM-DD' GROUP BY LocationID  
HAVING SUM(Amount) > 5000;
```

4
5
6
7
8
9
10

Task 4

-- 1. Retrieve Payments with Courier Information:

```
SELECT * FROM Payment
```

```
INNER JOIN Courier ON Payment.CourierID = Courier.CourierID;
```

-- 2. Retrieve Payments with Location Information:

```
SELECT * FROM Payment
```

```
INNER JOIN Location ON Payment.LocationID = Location.LocationID;
```

1	1	1	20.50	2023-01-20	1	Warehouse A 789 Warehouse St, Mumbai
2	2	2	15.75	2023-01-22	2	Warehouse B 456 Warehouse St, Delhi
3	3	3	12.00	2023-01-25	3	Warehouse C 123 Warehouse St, Bangalore
4	4	4	18.25	2023-01-28	4	Warehouse D 987 Warehouse St, Kolkata
5	5	5	22.80	2023-01-30	5	Warehouse E 654 Warehouse St, Chennai
6	6	6	14.50	2023-02-02	6	Warehouse F 210 Warehouse St, Hyderabad
7	7	7	17.30	2023-02-05	7	Warehouse G 876 Warehouse St, Pune
8	8	8	21.00	2023-02-08	8	Warehouse H 543 Warehouse St, Jaipur
9	9	9	10.75	2023-02-11	9	Warehouse I 432 Warehouse St, Ahmedabad
10	10	10	25.20	2023-02-14	10	Warehouse J 321 Warehouse St, Chandigarh

-- 3. 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;
```

-- 4. List all payments with courier details:

```
SELECT * FROM Payment
```

```
LEFT JOIN Courier ON Payment.CourierID = Courier.CourierID;
```

-- 5. Total payments received for each courier:

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

1	20.50
2	15.75
3	12.00
4	18.25
5	22.80
6	14.50
7	17.30
8	21.00
9	10.75

10 25.20

-- 6. List payments made on a specific date:

```
SELECT * FROM Payment WHERE PaymentDate = '2023-01-20';
```

1 1 1 20.50 2023-01-20

-- 7. Get Courier Information for Each Payment:

```
SELECT * FROM Payment
```

```
LEFT JOIN Courier ON Payment.CourierID = Courier.CourierID;
```

-- 8. Get Payment Details with Location:

```
SELECT * FROM Payment
```

```
LEFT JOIN Location ON Payment.LocationID = Location.LocationID;
```

1	1	1	20.50	2023-01-20	1	Warehouse A 789 Warehouse St, Mumbai
2	2	2	15.75	2023-01-22	2	Warehouse B 456 Warehouse St, Delhi
3	3	3	12.00	2023-01-25	3	Warehouse C 123 Warehouse St, Bangalore
4	4	4	18.25	2023-01-28	4	Warehouse D 987 Warehouse St, Kolkata
5	5	5	22.80	2023-01-30	5	Warehouse E 654 Warehouse St, Chennai
6	6	6	14.50	2023-02-02	6	Warehouse F 210 Warehouse St, Hyderabad
7	7	7	17.30	2023-02-05	7	Warehouse G 876 Warehouse St, Pune
8	8	8	21.00	2023-02-08	8	Warehouse H 543 Warehouse St, Jaipur
9	9	9	10.75	2023-02-11	9	Warehouse I 432 Warehouse St, Ahmedabad
10	10	10	25.20	2023-02-14	10	Warehouse J 321 Warehouse St, Chandigarh

-- 9. Calculating Total Payments for Each Courier:

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

1	20.50
2	15.75
3	12.00
4	18.25
5	22.80
6	14.50
7	17.30

8 21.00
9 10.75
10 25.20

-- 10. List Payments Within a Date Range:

SELECT * FROM Payment WHERE PaymentDate BETWEEN '2023-01-20' AND '2023-01-23';

1 1 1 20.50 2023-01-20
2 2 2 15.75 2023-01-22

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

SELECT * FROM User

LEFT JOIN Courier ON User.UserID = Courier.SenderID OR User.UserID = Courier.ReceiverID;

-- 12. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side:

SELECT * FROM Courier

LEFT JOIN CourierServices ON Courier.ServiceID = CourierServices.ServiceID;

-- 13. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side:

SELECT * FROM Employee

LEFT JOIN Payment ON Employee.EmployeeID = Payment.EmployeeID;

-- 14. List all users and all courier services, showing all possible combinations:

SELECT * FROM User

CROSS JOIN CourierServices;

-- 15. List all employees and all locations, showing all possible combinations:

SELECT * FROM Employee

CROSS JOIN Location;

-- 16. Retrieve a list of couriers and their corresponding sender information (if available):

```
SELECT * FROM Courier  
LEFT JOIN User ON Courier.SenderID = User.UserID;
```

-- 17. Retrieve a list of couriers and their corresponding receiver information (if available):

```
SELECT * FROM Courier  
LEFT JOIN User ON Courier.ReceiverID = User.UserID;
```

-- 18. Retrieve a list of couriers along with the courier service details (if available):

```
SELECT * FROM Courier  
LEFT JOIN CourierServices ON Courier.ServiceID = CourierServices.ServiceID;
```

-- 19. Retrieve a list of employees and the number of couriers assigned to each employee:

```
SELECT EmployeeID, COUNT(*) AS TotalCouriers FROM Courier GROUP BY EmployeeID;
```

-- 20. Retrieve a list of locations and the total payment amount received at each location:

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

-- 21. Retrieve all couriers sent by the same sender (based on SenderName):

```
SELECT * FROM Courier  
  
INNER JOIN Courier AS SenderCourier ON Courier.SenderName = SenderCourier.SenderName AND  
Courier.CourierID <> SenderCourier.CourierID;
```

-- 22. List all employees who share the same role:

```
SELECT * FROM Employee  
  
INNER JOIN Employee AS SameRoleEmployee ON Employee.Role = SameRoleEmployee.Role AND  
Employee.EmployeeID <> SameRoleEmployee.EmployeeID;
```

-- 23. Retrieve all payments made for couriers sent from the same location:

```
SELECT * FROM Payment  
  
INNER JOIN Courier ON Payment.CourierID = Courier.CourierID
```


INNER JOIN Location ON Courier.LocationID = Location.LocationID;

-- 24. Retrieve all couriers sent from the same location (based on SenderAddress):

SELECT * FROM Courier

INNER JOIN Courier AS SenderCourier ON Courier.SenderAddress = SenderCourier.SenderAddress
AND Courier.CourierID <> SenderCourier.CourierID;

-- 25. List employees and the number of couriers they have delivered:

SELECT EmployeeID, COUNT(*) AS TotalDeliveries FROM Courier WHERE Status = 'Delivered' GROUP
BY EmployeeID;

-- 26. Find couriers that were paid an amount greater than the cost of their respective courier services:

SELECT * FROM Courier

INNER JOIN Payment ON Courier.CourierID = Payment.CourierID

INNER JOIN CourierServices ON Courier.ServiceID = CourierServices.ServiceID

WHERE Payment.Amount > CourierServices.Cost;

-- 27. Find couriers that have a weight greater than the average weight of all couriers:

SELECT * FROM Courier

WHERE Weight > (SELECT AVG(Weight) FROM Courier);

-- 28. Find the names of all employees who have a salary greater than the average salary:

SELECT Name FROM Employee

WHERE Salary > (SELECT AVG(Salary) FROM Employee);

-- 29. Find the total cost of all courier services where the cost is less than the maximum cost:

SELECT SUM(Cost) AS TotalCost FROM CourierServices

WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);

-- 30. Find all couriers that have been paid for:

```
SELECT * FROM Courier  
WHERE CourierID IN (SELECT CourierID FROM Payment);
```

-- 31. Find the locations where the maximum payment amount was made:

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

-- 32. Find all couriers whose weight is greater than the weight of all couriers sent by a specific sender (e.g., 'SenderName'):

```
SELECT * FROM Courier  
WHERE Weight > ALL (SELECT Weight FROM Courier WHERE SenderName = 'SenderName');
```