

Assignment Sql-4 : 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.

```
mysql> create database courier_management_system;
Query OK, 1 row affected (0.01 sec)

mysql> use courier_management_system;
Database changed
```

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

user

```
mysql> CREATE TABLE User (
  ->   UserID INT PRIMARY KEY,
  ->   Name VARCHAR(255),
  ->   Email VARCHAR(255) UNIQUE,
  ->   Password VARCHAR(255),
  ->   ContactNumber VARCHAR(20),
  ->   Address TEXT
  -> );
Query OK, 0 rows affected (0.09 sec)

mysql> desc user;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| UserID     | int           | NO   | PRI  | NULL    |       |
| Name       | varchar(255)  | YES  |      | NULL    |       |
| Email      | varchar(255)  | YES  | UNI  | NULL    |       |
| Password   | varchar(255)  | YES  |      | NULL    |       |
| ContactNumber | varchar(20)  | YES  |      | NULL    |       |
| Address    | text          | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)
```

Courier

```
mysql> 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
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc courier;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CourierID       | int           | NO   | PRI | NULL    |       |
| SenderName      | varchar(255)  | YES  |     | NULL    |       |
| SenderAddress   | text          | YES  |     | NULL    |       |
| ReceiverName    | varchar(255)  | YES  |     | NULL    |       |
| ReceiverAddress | text          | YES  |     | NULL    |       |
| Weight          | decimal(5,2)  | YES  |     | NULL    |       |
| Status          | varchar(50)   | YES  |     | NULL    |       |
| TrackingNumber  | varchar(20)   | YES  | UNI | NULL    |       |
| DeliveryDate    | date          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

Courservices

```
mysql> CREATE TABLE CourierServices (
->     ServiceID INT PRIMARY KEY,
->     ServiceName VARCHAR(100),
->     Cost DECIMAL(8, 2)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> desc courierservices;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ServiceID      | int           | NO   | PRI | NULL    |       |
| ServiceName     | varchar(100)  | YES  |     | NULL    |       |
| Cost           | decimal(8,2)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Employee

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

Query OK, 0 rows affected (0.06 sec)

```
mysql> desc employee;
```

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	
Name	varchar(255)	YES		NULL	
Email	varchar(255)	YES	UNI	NULL	
ContactNumber	varchar(20)	YES		NULL	
Role	varchar(50)	YES		NULL	
Salary	decimal(10,2)	YES		NULL	

6 rows in set (0.00 sec)

Location

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

Query OK, 0 rows affected (0.04 sec)

```
mysql> desc location;
```

Field	Type	Null	Key	Default	Extra
LocationID	int	NO	PRI	NULL	
LocationName	varchar(100)	YES		NULL	
Address	text	YES		NULL	

3 rows in set (0.00 sec)

Payment

```
mysql> 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)
-> );
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> desc payment;
```

Field	Type	Null	Key	Default	Extra
PaymentID	int	NO	PRI	NULL	
CourierID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
Amount	decimal(10,2)	YES		NULL	
PaymentDate	date	YES		NULL	

```
5 rows in set (0.00 sec)
```

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

```
mysql> INSERT INTO User (UserID, Name, Email, Password, ContactNumber, Address)
-> VALUES
-> (1, 'John Doe', 'john@example.com', 'password123', '123-456-7890', '123 Main St'),
-> (2, 'Jane Smith', 'jane@example.com', 'securepass', '987-654-3210', '456 Elm St'),
-> (3, 'Alice Johnson', 'alice@example.com', 'alicepass', '555-555-5555', '789 Oak St'),
-> (4, 'Bob Brown', 'bob@example.com', 'bobpass', '111-222-3333', '567 Pine St'),
-> (5, 'Eva Martinez', 'eva@example.com', 'evapass', '333-444-5555', '890 Maple Ave');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Courier (CourierID, SenderName, SenderAddress, ReceiverName, ReceiverAddress, Weight, Status, TrackingNumber, DeliveryDate)
-> VALUES
-> (1, 'John Doe', '123 Main St', 'Jane Smith', '456 Elm St', 2.5, 'In transit', 'TRACK123', '2023-12-15'),
-> (2, 'Alice Johnson', '789 Oak St', 'Bob Brown', '567 Pine St', 1.8, 'Delivered', 'TRACK456', '2023-12-10'),
-> (3, 'Eva Martinez', '890 Maple Ave', 'John Doe', '123 Main St', 3.2, 'Scheduled', 'TRACK789', '2023-12-18'),
-> (4, 'Jane Smith', '456 Elm St', 'Alice Johnson', '789 Oak St', 1.0, 'Pending', 'TRACKABC', NULL),
-> (5, 'Bob Brown', '567 Pine St', 'Eva Martinez', '890 Maple Ave', 2.0, 'In transit', 'TRACKDEF', '2023-12-12');
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO CourierServices (ServiceID, ServiceName, Cost)
-> VALUES
-> (1, 'Standard Delivery', 10.00),
-> (2, 'Express Delivery', 20.00),
-> (3, 'Same-Day Delivery', 25.00),
-> (4, 'International Delivery', 30.00),
-> (5, 'Overnight Delivery', 15.00);
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Employee (EmployeeID, Name, Email, ContactNumber, Role, Salary)
-> VALUES
-> (1, 'Emily Davis', 'emily@example.com', '111-222-3333', 'Manager', 50000.00),
-> (2, 'Michael Johnson', 'michael@example.com', '444-555-6666', 'Delivery Personnel', 30000.00),
-> (3, 'Sophia Lee', 'sophia@example.com', '777-888-9999', 'Supervisor', 45000.00),
-> (4, 'Daniel Clark', 'daniel@example.com', '333-666-9999', 'Customer Support', 35000.00),
-> (5, 'Olivia Rodriguez', 'olivia@example.com', '222-777-4444', 'Dispatcher', 40000.00);
Query OK, 5 rows affected (0.02 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Location (LocationID, LocationName, Address)
-> VALUES
-> (1, 'Warehouse A', '789 Oak St'),
-> (2, 'Warehouse B', '567 Pine St'),
-> (3, 'Main Office', '123 Main St'),
-> (4, 'Regional Hub', '890 Maple Ave'),
-> (5, 'Distribution Center', '456 Elm St');
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Payment (PaymentID, CourierID, LocationID, Amount, PaymentDate)
-> VALUES
-> (1, 1, 3, 10.00, '2023-12-15'),
-> (2, 2, 1, 20.00, '2023-12-10'),
-> (3, 3, 4, 25.00, '2023-12-18'),
-> (4, 4, 2, 30.00, NULL),
-> (5, 5, 5, 15.00, '2023-12-12');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

Task- 2 Select,Where

1. List all customers

```
mysql> select * from user;
```

UserID	Name	Email	Password	ContactNumber	Address
1	John Doe	john@example.com	password123	123-456-7890	123 Main St
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave

```
5 rows in set (0.00 sec)
```

2. List all orders for a specific customer

```
mysql> SELECT c.*
-> FROM Courier c
-> INNER JOIN User u ON c.SenderName = u.Name OR c.ReceiverName = u.Name
-> WHERE u.UserID = 1;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18

```
2 rows in set (0.00 sec)
```

3. List all couriers

```
mysql> select * from courier;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12

```
5 rows in set (0.00 sec)
```

4. List all packages for a specific order

```
mysql> SELECT *
-> FROM Courier
-> WHERE CourierID = 1;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15

```
1 row in set (0.00 sec)
```

5. List all deliveries for a specific courier

```
mysql> select * from courier where courierid=2;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10

```
1 row in set (0.00 sec)
```

6. List all undelivered packages

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

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12

4 rows in set (0.04 sec)

7. List all packages that are scheduled for delivery today

```
mysql> SELECT *
-> FROM Courier
-> WHERE DeliveryDate = CURDATE() AND Status = 'Scheduled';
Empty set (0.01 sec)
```

8. List all packages with a specific status:

```
mysql> SELECT *
-> FROM Courier
-> WHERE Status = 'In transit';
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12

2 rows in set (0.00 sec)

9. Calculate the total number of packages for each courier

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

CourierID	TotalPackages
1	1
2	1
3	1
4	1
5	1

5 rows in set (0.01 sec)

10. Find the average delivery time for each courier

```
mysql> SELECT CourierID, AVG(DATEDIFF(CURRENT_DATE, DeliveryDate)) AS AverageDeliveryTime
-> FROM Courier
-> GROUP BY CourierID;
```

CourierID	AverageDeliveryTime
1	-3.0000
2	2.0000
3	-6.0000
4	NULL
5	0.0000

5 rows in set (0.02 sec)

11. List all packages with a specific weight range

```
mysql> SELECT *
-> FROM Courier
-> WHERE Weight BETWEEN 1.5 AND 3.0;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12

3 rows in set (0.00 sec)

12. Retrieve employees whose names contain 'John'

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

EmployeeID	Name	Email	ContactNumber	Role	Salary
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00

1 row in set (0.03 sec)

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

```
mysql> SELECT c.*
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> WHERE p.Amount > 50.00;
```

Empty set (0.01 sec)

Task-3

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


```
mysql> SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS TotalCouriersHandled
-> FROM Employee e
-> LEFT JOIN Courier c ON e.Name = c.SenderName OR e.Name = c.ReceiverName
-> GROUP BY e.EmployeeID, e.Name;
```

EmployeeID	Name	TotalCouriersHandled
1	Emily Davis	0
2	Michael Johnson	0
3	Sophia Lee	0
4	Daniel Clark	0
5	Olivia Rodriguez	0

5 rows in set (0.03 sec)

15. Calculate the total revenue generated by each location

```
mysql> SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalRevenue
-> FROM Location l
-> LEFT JOIN Payment p ON l.LocationID = p.LocationID
-> GROUP BY l.LocationID, l.LocationName;
```

LocationID	LocationName	TotalRevenue
1	Warehouse A	20.00
2	Warehouse B	30.00
3	Main Office	10.00
4	Regional Hub	25.00
5	Distribution Center	15.00
6	Warehouse A	75.00
7	Warehouse B	NULL
8	Main Office	110.00

8 rows in set (0.01 sec)

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

```
mysql> SELECT l.LocationID, l.LocationName, COUNT(c.CourierID) AS TotalCouriersDelivered
-> FROM Location l
-> LEFT JOIN Courier c ON l.Address = c.ReceiverAddress
-> WHERE c.Status = 'Delivered'
-> GROUP BY l.LocationID, l.LocationName;
```

LocationID	LocationName	TotalCouriersDelivered
2	Warehouse B	2
7	Warehouse B	2

2 rows in set (0.00 sec)

17. Find the courier with the highest average delivery time

```
mysql> SELECT c.CourierID, AVG(DATEDIFF(p.PaymentDate, c.DeliveryDate)) AS avg_delivery_time
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> WHERE c.DeliveryDate IS NOT NULL
-> GROUP BY c.CourierID
-> ORDER BY avg_delivery_time DESC
-> LIMIT 1;
```

CourierID	avg_delivery_time
1	0.0000

1 row in set (0.02 sec)

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

```
mysql> SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS total_payments
-> FROM Location l
-> LEFT JOIN Payment p ON l.LocationID = p.LocationID
-> GROUP BY l.LocationID
-> HAVING COALESCE(SUM(p.Amount), 0) <75 ;
```

LocationID	LocationName	total_payments
1	Warehouse A	20.00
2	Warehouse B	30.00
3	Main Office	10.00
4	Regional Hub	25.00
5	Distribution Center	15.00
7	Warehouse B	NULL

6 rows in set (0.01 sec)

19. Calculate Total Payments per Location

```
mysql> SELECT l.LocationID, l.LocationName, COALESCE(SUM(p.Amount), 0) AS total_payments
-> FROM Location l
-> LEFT JOIN Payment p ON l.LocationID = p.LocationID
-> GROUP BY l.LocationID, l.LocationName;
```

LocationID	LocationName	total_payments
1	Warehouse A	20.00
2	Warehouse B	30.00
3	Main Office	10.00
4	Regional Hub	25.00
5	Distribution Center	15.00
6	Warehouse A	75.00
7	Warehouse B	0.00
8	Main Office	110.00

8 rows in set (0.00 sec)

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

```
mysql> SELECT c.CourierID, c.SenderName, c.ReceiverName, l.LocationID, l.LocationName, SUM(p.Amount) AS total_payments
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> JOIN Location l ON p.LocationID = l.LocationID
-> WHERE l.LocationID = 2
-> GROUP BY c.CourierID, c.SenderName, c.ReceiverName, l.LocationID, l.LocationName
-> HAVING total_payments > 1000;
Empty set (0.01 sec)
```

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

```
mysql> SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS total_payments
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> WHERE p.PaymentDate > '2023-12-15'
-> GROUP BY c.CourierID, c.SenderName, c.ReceiverName
-> HAVING total_payments > 1000;
Empty set (0.04 sec)
```

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

```
mysql> SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS total_payments
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> WHERE p.PaymentDate > '2023-12-15'
-> GROUP BY c.CourierID, c.SenderName, c.ReceiverName
-> HAVING total_payments > 5000;
Empty set (0.00 sec)
```

Task-4

23. Retrieve Payments with Courier Information

```
mysql> SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderName, c.ReceiverName
-> FROM Payment p
-> JOIN Courier c ON p.CourierID = c.CourierID;
```

PaymentID	Amount	PaymentDate	CourierID	SenderName	ReceiverName
1	10.00	2023-12-15	1	John Doe	Jane Smith
2	20.00	2023-12-10	2	Alice Johnson	Bob Brown
3	25.00	2023-12-18	3	Eva Martinez	John Doe
4	30.00	NULL	4	Jane Smith	Alice Johnson
5	15.00	2023-12-12	5	Bob Brown	Eva Martinez
6	50.00	2023-12-15	6	John Doe	Jane Smith
7	75.00	2023-12-10	7	Alice Johnson	Bob Brown
8	60.00	NULL	8	Eva Martinez	John Doe

8 rows in set (0.00 sec)

24. Retrieve Payments with Location Information

```
mysql> SELECT p.PaymentID, p.Amount, p.PaymentDate, l.LocationID, l.LocationName
-> FROM Payment p
-> JOIN Location l ON p.LocationID = l.LocationID;
```

PaymentID	Amount	PaymentDate	LocationID	LocationName
1	10.00	2023-12-15	3	Main Office
2	20.00	2023-12-10	1	Warehouse A
3	25.00	2023-12-18	4	Regional Hub
4	30.00	NULL	2	Warehouse B
5	15.00	2023-12-12	5	Distribution Center
6	50.00	2023-12-15	8	Main Office
7	75.00	2023-12-10	6	Warehouse A
8	60.00	NULL	8	Main Office

8 rows in set (0.00 sec)

25. Retrieve Payments with Courier and Location Information

```
mysql> SELECT py.PaymentID, py.Amount, py.PaymentDate, c.CourierID, c.SenderName, c.ReceiverName, l.LocationID, l.LocationName
-> FROM Payment py
-> JOIN Courier c ON py.CourierID = c.CourierID
-> JOIN Location l ON py.LocationID = l.LocationID;
```

PaymentID	Amount	PaymentDate	CourierID	SenderName	ReceiverName	LocationID	LocationName
1	10.00	2023-12-15	1	John Doe	Jane Smith	3	Main Office
2	20.00	2023-12-10	2	Alice Johnson	Bob Brown	1	Warehouse A
3	25.00	2023-12-18	3	Eva Martinez	John Doe	4	Regional Hub
4	30.00	NULL	4	Jane Smith	Alice Johnson	2	Warehouse B
5	15.00	2023-12-12	5	Bob Brown	Eva Martinez	5	Distribution Center
6	50.00	2023-12-15	6	John Doe	Jane Smith	8	Main Office
7	75.00	2023-12-10	7	Alice Johnson	Bob Brown	6	Warehouse A
8	60.00	NULL	8	Eva Martinez	John Doe	8	Main Office

8 rows in set (0.01 sec)

26. List all payments with courier details

```
mysql> SELECT p.PaymentID, p.Amount, p.PaymentDate, c.CourierID, c.SenderName, c.SenderAddress, c.ReceiverName, c.ReceiverAddress
-> FROM Payment p
-> JOIN Courier c ON p.CourierID = c.CourierID;
```

PaymentID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress
1	10.00	2023-12-15	1	John Doe	123 Main St	Jane Smith	456 Elm St
2	20.00	2023-12-10	2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St
3	25.00	2023-12-18	3	Eva Martinez	890 Maple Ave	John Doe	123 Main St
4	30.00	NULL	4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St
5	15.00	2023-12-12	5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave
6	50.00	2023-12-15	6	John Doe	123 Main St	Jane Smith	456 Elm St
7	75.00	2023-12-10	7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St
8	60.00	NULL	8	Eva Martinez	890 Maple Ave	John Doe	123 Main St

8 rows in set (0.04 sec)

27. Total payments received for each courier

```
mysql> SELECT c.CourierID, c.SenderName, c.ReceiverName, SUM(p.Amount) AS TotalPaymentsReceived
-> FROM Courier c
-> LEFT JOIN Payment p ON c.CourierID = p.CourierID
-> GROUP BY c.CourierID, c.SenderName, c.ReceiverName;
```

CourierID	SenderName	ReceiverName	TotalPaymentsReceived
1	John Doe	Jane Smith	10.00
2	Alice Johnson	Bob Brown	20.00
3	Eva Martinez	John Doe	25.00
4	Jane Smith	Alice Johnson	30.00
5	Bob Brown	Eva Martinez	15.00
6	John Doe	Jane Smith	50.00
7	Alice Johnson	Bob Brown	75.00
8	Eva Martinez	John Doe	60.00

8 rows in set (0.00 sec)

28. List payments made on a specific date

```
mysql> SELECT *
-> FROM Payment
-> WHERE PaymentDate = '2023-12-15';
```

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	3	10.00	2023-12-15
6	6	8	50.00	2023-12-15

2 rows in set (0.00 sec)

29. Get Courier Information for Each Payment

```
mysql> SELECT p.*, c.*
-> FROM Payment p
-> JOIN Courier c ON p.CourierID = c.CourierID;
```

PaymentID	CourierID	LocationID	Amount	PaymentDate	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	1	3	10.00	2023-12-15	1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
2	2	1	20.00	2023-12-10	2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
3	3	4	25.00	2023-12-18	3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18
4	4	2	30.00	NULL	4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL
5	5	5	15.00	2023-12-12	5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12
6	6	8	50.00	2023-12-15	6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15
7	7	6	75.00	2023-12-10	7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10
8	8	8	60.00	NULL	8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18

8 rows in set (0.00 sec)

30. Get Payment Details with Location

```
mysql> SELECT p.*, l.*
-> FROM Payment p
-> JOIN Location l ON p.LocationID = l.LocationID;
```

PaymentID	CourierID	LocationID	Amount	PaymentDate	LocationID	LocationName	Address
1	1	3	10.00	2023-12-15	3	Main Office	123 Main St
2	2	1	20.00	2023-12-10	1	Warehouse A	789 Oak St
3	3	4	25.00	2023-12-18	4	Regional Hub	890 Maple Ave
4	4	2	30.00	NULL	2	Warehouse B	567 Pine St
5	5	5	15.00	2023-12-12	5	Distribution Center	456 Elm St
6	6	8	50.00	2023-12-15	8	Main Office	123 Main St
7	7	6	75.00	2023-12-10	6	Warehouse A	789 Oak St
8	8	8	60.00	NULL	8	Main Office	123 Main St

8 rows in set (0.00 sec)

31. Calculating Total Payments for Each Courier

```
mysql> SELECT c.*, SUM(p.Amount) AS TotalPayments
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> GROUP BY c.CourierID;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	TotalPayments
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	10.00
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10	20.00
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	25.00
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL	30.00
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12	15.00
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	50.00
7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10	75.00
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	60.00

8 rows in set (0.00 sec)

32. List Payments Within a Date Range

```
mysql> SELECT *
-> FROM Payment
-> WHERE PaymentDate BETWEEN '2022-12-15' AND '2023-12-15';
```

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	3	10.00	2023-12-15
2	2	1	20.00	2023-12-10
5	5	5	15.00	2023-12-12
6	6	8	50.00	2023-12-15
7	7	6	75.00	2023-12-10

5 rows in set (0.00 sec)

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

```
mysql> SELECT u.*, c.*
-> FROM User u
-> LEFT JOIN Courier c ON u.Name = c.SenderName OR u.Name = c.ReceiverName
-> UNION
-> SELECT u.*, c.*
-> FROM Courier c
-> LEFT JOIN User u ON u.Name = c.SenderName OR u.Name = c.ReceiverName
-> WHERE u.UserID IS NULL;
```

UserID	Name	Email	Password	ContactNumber	Address	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18

16 rows in set (0.06 sec)

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

```
mysql> SELECT c.*, cs.*
-> FROM Courier c
-> LEFT JOIN CourierServices cs ON c.ServiceID = cs.ServiceID;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	ServiceID	ServiceName	Cost
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1	1	Standard Delivery	10.00
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10	NULL	NULL	NULL	NULL
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL	NULL	NULL	NULL
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL	NULL	NULL	NULL	NULL
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12	NULL	NULL	NULL	NULL
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL	NULL	NULL	NULL
7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10	NULL	NULL	NULL	NULL
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL	NULL	NULL	NULL

8 rows in set (0.00 sec)

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

```
mysql> SELECT * FROM Employee e LEFT JOIN Payment p ON e.EmployeeID = p.EmployeeID
-> UNION SELECT * FROM Employee e RIGHT JOIN Payment p ON e.EmployeeID = p.EmployeeID
-> WHERE e.EmployeeID IS NULL;
```

EmployeeID	Name	Email	ContactNumber	Role	Salary	PaymentID	CourierID	LocationID	Amount	PaymentDate	EmployeeID
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	1	1	3	10.00	2023-12-15	1
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	NUL	NUL	NUL	NUL	NUL	NUL
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	NUL	NUL	NUL	NUL	NUL	NUL
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	NUL	NUL	NUL	NUL	NUL	NUL
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	NUL	NUL	NUL	NUL	NUL	NUL
NUL	NUL	NUL	NUL	NUL	NUL	2	2	1	20.00	2023-12-10	NUL
NUL	NUL	NUL	NUL	NUL	NUL	3	3	4	25.00	2023-12-18	NUL
NUL	NUL	NUL	NUL	NUL	NUL	4	4	2	30.00	NUL	NUL
NUL	NUL	NUL	NUL	NUL	NUL	5	5	5	15.00	2023-12-12	NUL
NUL	NUL	NUL	NUL	NUL	NUL	6	6	8	50.00	2023-12-15	NUL
NUL	NUL	NUL	NUL	NUL	NUL	7	7	6	75.00	2023-12-10	NUL
NUL	NUL	NUL	NUL	NUL	NUL	8	8	8	60.00	NUL	NUL

12 rows in set (0.00 sec)

36. List all users and all courier services, showing all possible combinations.

```
mysql> SELECT u.*, cs.*
-> FROM User u
-> CROSS JOIN CourierServices cs;
```

UserID	Name	Email	Password	ContactNumber	Address	ServiceID	ServiceName	Cost
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	1	Standard Delivery	10.00
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	1	Standard Delivery	10.00
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	1	Standard Delivery	10.00
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	1	Standard Delivery	10.00
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	1	Standard Delivery	10.00
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	2	Express Delivery	20.00
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	2	Express Delivery	20.00
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	2	Express Delivery	20.00
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	2	Express Delivery	20.00
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	2	Express Delivery	20.00
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	3	Same-Day Delivery	25.00
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	3	Same-Day Delivery	25.00
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	3	Same-Day Delivery	25.00
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	3	Same-Day Delivery	25.00
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	3	Same-Day Delivery	25.00
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	4	International Delivery	30.00
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	4	International Delivery	30.00
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	4	International Delivery	30.00
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	4	International Delivery	30.00
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	4	International Delivery	30.00
5	Eva Martinez	eva@example.com	evapass	333-444-5555	890 Maple Ave	5	Overnight Delivery	15.00
4	Bob Brown	bob@example.com	bobpass	111-222-3333	567 Pine St	5	Overnight Delivery	15.00
3	Alice Johnson	alice@example.com	alicepass	555-555-5555	789 Oak St	5	Overnight Delivery	15.00
2	Jane Smith	jane@example.com	securepass	987-654-3210	456 Elm St	5	Overnight Delivery	15.00
1	John Doe	john@example.com	password123	123-456-7890	123 Main St	5	Overnight Delivery	15.00

25 rows in set (0.00 sec)

37. List all employees and all locations, showing all possible combinations:


```
mysql> SELECT e.*, l.*
-> FROM Employee e
-> CROSS JOIN Location l;
```

EmployeeID	Name	Email	ContactNumber	Role	Salary	LocationID	LocationName	Address
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	1	Warehouse A	789 Oak St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	1	Warehouse A	789 Oak St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	1	Warehouse A	789 Oak St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	1	Warehouse A	789 Oak St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	1	Warehouse A	789 Oak St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	2	Warehouse B	567 Pine St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	2	Warehouse B	567 Pine St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	2	Warehouse B	567 Pine St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	2	Warehouse B	567 Pine St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	2	Warehouse B	567 Pine St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	3	Main Office	123 Main St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	3	Main Office	123 Main St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	3	Main Office	123 Main St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	3	Main Office	123 Main St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	3	Main Office	123 Main St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	4	Regional Hub	890 Maple Ave
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	4	Regional Hub	890 Maple Ave
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	4	Regional Hub	890 Maple Ave
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	4	Regional Hub	890 Maple Ave
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	4	Regional Hub	890 Maple Ave
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	5	Distribution Center	456 Elm St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	5	Distribution Center	456 Elm St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	5	Distribution Center	456 Elm St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	5	Distribution Center	456 Elm St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	5	Distribution Center	456 Elm St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	6	Warehouse A	789 Oak St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	6	Warehouse A	789 Oak St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	6	Warehouse A	789 Oak St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	6	Warehouse A	789 Oak St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	6	Warehouse A	789 Oak St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	7	Warehouse B	567 Pine St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	7	Warehouse B	567 Pine St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	7	Warehouse B	567 Pine St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	7	Warehouse B	567 Pine St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	7	Warehouse B	567 Pine St
5	Olivia Rodriguez	olivia@example.com	222-777-4444	Dispatcher	40000.00	8	Main Office	123 Main St
4	Daniel Clark	daniel@example.com	333-666-9999	Customer Support	35000.00	8	Main Office	123 Main St
3	Sophia Lee	sophia@example.com	777-888-9999	Supervisor	45000.00	8	Main Office	123 Main St
2	Michael Johnson	michael@example.com	444-555-6666	Delivery Personnel	30000.00	8	Main Office	123 Main St
1	Emily Davis	emily@example.com	111-222-3333	Manager	50000.00	8	Main Office	123 Main St

10 rows in set (0.00 sec)

38. Retrieve a list of couriers and their corresponding sender information (if available)

```
mysql> SELECT c.CourierID, c.SenderName, c.SenderAddress
-> FROM Courier c;
```

CourierID	SenderName	SenderAddress
1	John Doe	123 Main St
2	Alice Johnson	789 Oak St
3	Eva Martinez	890 Maple Ave
4	Jane Smith	456 Elm St
5	Bob Brown	567 Pine St
6	John Doe	123 Main St
7	Alice Johnson	789 Oak St
8	Eva Martinez	890 Maple Ave

8 rows in set (0.00 sec)

39. Retrieve a list of couriers and their corresponding receiver information (if available)

```
mysql> SELECT CourierID, ReceiverName, ReceiverAddress
-> FROM Courier;
```

CourierID	ReceiverName	ReceiverAddress
1	Jane Smith	456 Elm St
2	Bob Brown	567 Pine St
3	John Doe	123 Main St
4	Alice Johnson	789 Oak St
5	Eva Martinez	890 Maple Ave
6	Jane Smith	456 Elm St
7	Bob Brown	567 Pine St
8	John Doe	123 Main St

8 rows in set (0.00 sec)

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

```
mysql> SELECT c.CourierID, c.SenderName, c.SenderAddress, c.ReceiverName, c.ReceiverAddress, c.Weight, c.Status, c.TrackingNumber, c.DeliveryDate,
-> cs.ServiceID, cs.ServiceName, cs.Cost
-> FROM Courier c
-> LEFT JOIN CourierServices cs ON c.ServiceID = cs.ServiceID;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	ServiceName	Cost
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1	Standard Delivery	10.00
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10	NULL	NULL	NULL
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL	NULL	NULL
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL	NULL	NULL	NULL
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12	NULL	NULL	NULL
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL	NULL	NULL
7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10	NULL	NULL	NULL
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL	NULL	NULL

8 rows in set (0.00 sec)

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

```
mysql> SELECT e.EmployeeID, e.Name AS EmployeeName,
-> COUNT(CASE WHEN e.Name = c.SenderName OR e.Name = c.ReceiverName THEN c.CourierID END) AS NumCouriersAssigned
-> FROM Employee e
-> LEFT JOIN Courier c ON e.Name = c.SenderName OR e.Name = c.ReceiverName
-> GROUP BY e.EmployeeID, e.Name;
```

EmployeeID	EmployeeName	NumCouriersAssigned
1	Emily Davis	0
2	Michael Johnson	0
3	Sophia Lee	0
4	Daniel Clark	0
5	Olivia Rodriguez	0

5 rows in set (0.01 sec)

42. Retrieve a list of locations and the total payment amount received at each location

```
mysql> SELECT l.LocationID, l.LocationName, COALESCE(SUM(p.Amount), 0) AS TotalPaymentAmount
-> FROM Location l
-> LEFT JOIN Payment p ON l.LocationID = p.LocationID
-> GROUP BY l.LocationID, l.LocationName;
```

LocationID	LocationName	TotalPaymentAmount
1	Warehouse A	20.00
2	Warehouse B	30.00
3	Main Office	10.00
4	Regional Hub	25.00
5	Distribution Center	15.00
6	Warehouse A	75.00
7	Warehouse B	0.00
8	Main Office	110.00

8 rows in set (0.00 sec)

43. Retrieve all couriers sent by the same sender (based on SenderName).

```
mysql> SELECT c1.*
-> FROM Courier c1
-> JOIN Courier c2 ON c1.SenderName = c2.SenderName AND c1.CourierID <> c2.CourierID
-> WHERE c2.SenderName = 'John Doe';
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1

2 rows in set (0.00 sec)

44. List all employees who share the same role.

```
mysql> SELECT Role, GROUP_CONCAT(Name) AS Employees
-> FROM Employee
-> GROUP BY Role
-> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

45. Retrieve all payments made for couriers sent from the same location.

```
mysql> SELECT p.*
-> FROM Payment p
-> JOIN Courier c1 ON p.CourierID = c1.CourierID
-> JOIN Courier c2 ON c1.SenderAddress = c2.SenderAddress AND c1.CourierID <> c2.CourierID
-> ;
```

PaymentID	CourierID	LocationID	Amount	PaymentDate	EmployeeID
6	6	8	50.00	2023-12-15	NULL
7	7	6	75.00	2023-12-10	NULL
8	8	8	60.00	NULL	NULL
1	1	3	10.00	2023-12-15	1
2	2	1	20.00	2023-12-10	NULL
3	3	4	25.00	2023-12-18	NULL

6 rows in set (0.00 sec)

46. Retrieve all couriers sent from the same location (based on SenderAddress).

```
mysql> SELECT c1.*
-> FROM Courier c1
-> JOIN Courier c2 ON c1.Senderaddress like c2.Senderaddress where c1.CourierID <> c2.CourierID
-> ;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL
7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10	NULL
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10	NULL
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL

6 rows in set (0.00 sec)

```
mysql> select * from courier;
```

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

```
mysql> ^C
mysql> SELECT e.EmployeeID, e.Name AS EmployeeName, COUNT(c.CourierID) AS NumCouriersDelivered
-> FROM Employee e
-> LEFT JOIN Courier c ON e.EmployeeID = c.EmployeeID
-> GROUP BY e.EmployeeID, e.Name;
```

EmployeeID	EmployeeName	NumCouriersDelivered
1	Emily Davis	0
2	Michael Johnson	2
3	Sophia Lee	0
4	Daniel Clark	0
5	Olivia Rodriguez	0

5 rows in set (0.00 sec)

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

```
mysql> SELECT c.*, p.Amount AS PaymentAmount, cs.Cost AS ServiceCost
-> FROM Courier c
-> JOIN Payment p ON c.CourierID = p.CourierID
-> JOIN CourierServices cs ON c.ServiceID = cs.ServiceID
-> WHERE p.Amount > cs.Cost;
Empty set (0.01 sec)
```

Scope: Inner Queries, Non Equi Joins, Equi joins,Exist,Any,All

49. Find couriers that have a weight greater than the average weight of all couriers

```
mysql> SELECT *
-> FROM Courier
-> WHERE Weight > (SELECT AVG(Weight) FROM Courier);
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	EmployeeID
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1	NULL
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL	NULL
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL	NULL
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL	NULL

4 rows in set (0.00 sec)

50. Find the names of all employees who have a salary greater than the average salary

```
mysql> SELECT Name
-> FROM Employee
-> WHERE Salary > (SELECT AVG(Salary) FROM Employee);
```

Name
Emily Davis
Sophia Lee

2 rows in set (0.00 sec)

51. Find the total cost of all courier services where the cost is less than the maximum cost

```
mysql> SELECT SUM(Cost) AS TotalCost
-> FROM CourierServices
-> WHERE Cost < (SELECT MAX(Cost) FROM CourierServices);
```

TotalCost
70.00

1 row in set (0.00 sec)

52. Find all couriers that have been paid for

```
mysql> SELECT DISTINCT c.*
-> FROM Courier c
-> INNER JOIN Payment p ON c.CourierID = p.CourierID;
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	EmployeeID
1	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK123	2023-12-15	1	NULL
2	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK456	2023-12-10	NULL	2
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL	NULL
4	Jane Smith	456 Elm St	Alice Johnson	789 Oak St	1.00	Pending	TRACKABC	NULL	NULL	NULL
5	Bob Brown	567 Pine St	Eva Martinez	890 Maple Ave	2.00	In transit	TRACKDEF	2023-12-12	NULL	NULL
6	John Doe	123 Main St	Jane Smith	456 Elm St	2.50	In transit	TRACK124	2023-12-15	NULL	NULL
7	Alice Johnson	789 Oak St	Bob Brown	567 Pine St	1.80	Delivered	TRACK457	2023-12-10	NULL	2
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL	NULL

8 rows in set (0.00 sec)

53. Find the locations where the maximum payment amount was made

```
mysql> SELECT LocationID, MAX(Amount) AS MaximumPaymentAmount
-> FROM Payment
-> GROUP BY LocationID
-> ORDER BY MaximumPaymentAmount DESC
-> LIMIT 1;
```

LocationID	MaximumPaymentAmount
6	75.00

1 row in set (0.00 sec)

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

```
mysql> SELECT *
-> FROM Courier
-> WHERE Weight > (
-> SELECT MAX(Weight)
-> FROM Courier
-> WHERE SenderName = 'John Doe'
-> );
```

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	ServiceID	EmployeeID
3	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK789	2023-12-18	NULL	NULL
8	Eva Martinez	890 Maple Ave	John Doe	123 Main St	3.20	Scheduled	TRACK790	2023-12-18	NULL	NULL

2 rows in set (0.00 sec)