

Courier Management System

Keerthika Nagarajan

Superset ID: 5370583

Saveetha Engineering College

Task1: Database Design

Creating Database

```
CREATE DATABASE courierdb;
```

```
USE courierdb;
```

Creating User table and inserting data

```
CREATE TABLE User (
```

```
    UserID INT PRIMARY KEY,
```

```
    Name VARCHAR(255) NOT NULL,
```

```
    Email VARCHAR(255) UNIQUE NOT NULL,
```

```
    Password VARCHAR(255) NOT NULL,
```

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

```
    Address TEXT
```

```
);
```

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

```
VALUES
```

```
(1011, 'Rajesh K', 'rajeshk@gmail.com', 'Rajesh@111', '9123456789', 'T Nagar, Chennai, Tamil Nadu'),
```

```
(1012, 'Sneha R', 'snehar@gmail.com', 'Sneha@222', '9234567890', 'Peelamedu, Coimbatore, Tamil Nadu'),
```

```
(1013, 'Aravind M', 'aravindm@gmail.com', 'Aravind@333', '9345678901', 'Sellur, Madurai, Tamil Nadu'),
```

```
(1014, 'Pooja S', 'poojas@gmail.com', 'Pooja@444', '9456789012', 'Perundurai, Erode, Tamil Nadu'),
```

```
(1015, 'Harish V', 'harishv@gmail.com', 'Harish@555', '9567890123', 'Woraiyur, Trichy, Tamil Nadu'),
```

```
(1016, 'Lakshmi N', 'lakshmin@gmail.com', 'Lakshmi@666', '9678901234', 'RS Puram, Coimbatore, Tamil Nadu'),
```

```
(1017, 'Gokul P', 'gokulp@gmail.com', 'Gokul@777', '9789012345', 'Villapuram, Madurai, Tamil Nadu'),
```

```
(1018, 'Deepa K', 'deepak@gmail.com', 'Deepa@888', '9890123456', 'Fairlands, Salem, Tamil Nadu'),
```

```
(1019, 'Vignesh R', 'vigneshr@gmail.com', 'Vignesh@999', '9901234567', 'Katpadi, Vellore, Tamil Nadu'),
```

```
(1020, 'Swetha T', 'swethat@gmail.com', 'Swetha@000', '9812345678', 'Srirangam, Trichy, Tamil Nadu');
```

```
SELECT * FROM User;
```


Creating Employee table and inserting data

```
CREATE TABLE Employee (  
    EmployeeID INT PRIMARY KEY,  
    Name VARCHAR(255),  
    Email VARCHAR(255) UNIQUE,  
    ContactNumber VARCHAR(20),  
    Role VARCHAR(50),  
    Salary DECIMAL(10,2)  
);  
  
INSERT INTO Employee (EmployeeID, Name, Email, ContactNumber, Role, Salary)  
VALUES  
(126, 'Arjun M', 'arjun.m@email.com', '9876054321', 'HR Manager', 95000.00),  
(127, 'Rahul V', 'rahul.v@email.com', '9768451234', 'Operations Head', 72000.00),  
(128, 'Divya S', 'divya.s@email.com', '9854123698', 'Logistics Coordinator', 48000.00),  
(129, 'Kavitha T', 'kavitha.t@email.com', '9745124789', 'Customer Service Rep', 38000.00),  
(130, 'Sandeep K', 'sandeep.k@email.com', '9632547891', 'Inventory Supervisor', 36500.00);  
  
SELECT * FROM Employee;
```

	EmployeeID	Name	Email	ContactNumber	Role	Salary
▶	126	Oliver Queen	oliverq@email.com	9876054321	HR Manager	95000.00
	127	Barry Allen	barrya@email.com	9768451234	Operations Head	72000.00
	128	Kara Danvers	karad@email.com	9854123698	Logistics Coordinator	48000.00
	129	Clark Kent	clarkk@email.com	9745124789	Customer Service Rep	38000.00
	130	Bruce Wayne	brucew@email.com	9632547891	Inventory Supervisor	36500.00
*	NULL	NULL	NULL	NULL	NULL	NULL

Creating Service table and inserting data

```
CREATE TABLE Service (  
    ServiceID VARCHAR(10) PRIMARY KEY,  
    ServiceName VARCHAR(100),  
    Cost DECIMAL(8,2)  
);  
  
INSERT INTO Service (ServiceID, ServiceName, Cost)  
VALUES  
('S16', 'Eco-Friendly Delivery', 180.00),  
('S17', 'Same-Day Express', 550.00),  
('S18', 'Global Priority Shipping', 3200.00);  
  
SELECT * FROM Service;
```

	ServiceID	ServiceName	Cost
▶	S16	Eco-Friendly Delivery	180.00
	S17	Same-Day Express	550.00
	S18	Global Priority Shipping	3200.00
•	NULL	NULL	NULL

Creating Location table and inserting data

CREATE TABLE Location (

LocationID INT PRIMARY KEY,

LocationName VARCHAR(100),

Address TEXT

);

INSERT INTO Location (LocationID, LocationName, Address)

VALUES

(11, 'Chengalpattu Logistics Hub', '12, GST Road, Chengalpattu, Tamil Nadu - 603001'),

(12, 'Thoothukudi Parcel Center', '25, Beach Road, Thoothukudi, Tamil Nadu - 628002'),

(13, 'Tiruvannamalai Distribution Hub', '36, Arunachala Street, Tiruvannamalai, Tamil Nadu - 606603');

SELECT * FROM Location;

	LocationID	LocationName	Address
▶	11	Springfield Logistics Hub	123 Main St, Springfield, IL
	12	Metropolis Parcel Center	456 Elm St, Metropolis, NY
	13	Gotham Distribution Hub	789 Oak St, Gotham, NJ
•	NULL	NULL	NULL

Creating Payment table and inserting data

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY,

CourierID INT,

LocationID INT,

UserID INT,

ServiceID VARCHAR(10),

Amount DECIMAL(10, 2),

PaymentDate DATE,

FOREIGN KEY (UserID) REFERENCES User(UserID),

FOREIGN KEY (ServiceID) REFERENCES Service(ServiceID),

FOREIGN KEY (CourierID) REFERENCES Courier(CourierID),

FOREIGN KEY (LocationID) REFERENCES Location(LocationID)

);

INSERT INTO Payment (PaymentID, CourierID, LocationID, UserID, ServiceID, Amount, PaymentDate)

VALUES

```
(501, 11, 11, 1011, 'S16', 150.00, '2025-01-12'),
(502, 12, 12, 1012, 'S17', 250.00, '2025-03-21'),
(503, 13, 13, 1013, 'S18', 500.00, '2025-02-25');
SELECT * FROM Payment;
```

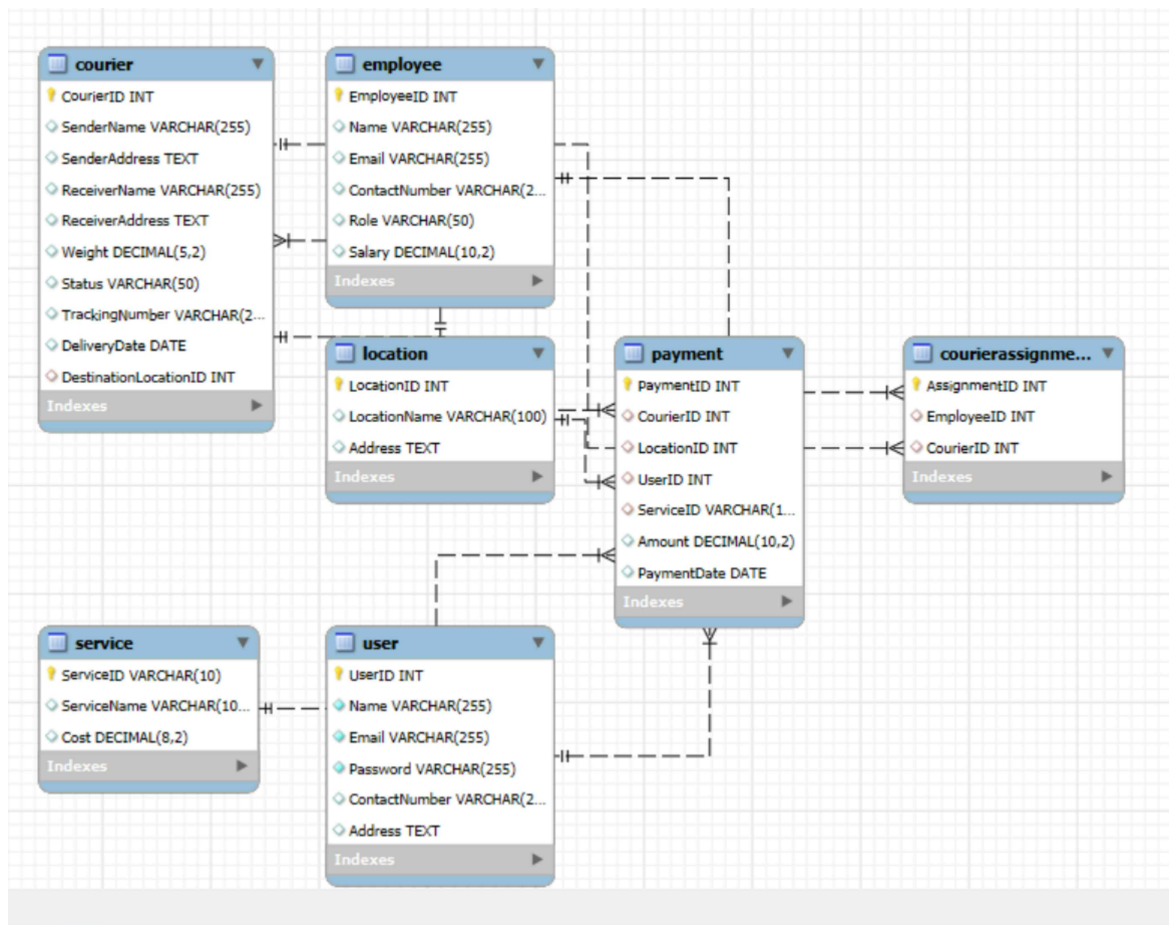
	PaymentID	CourierID	LocationID	UserID	ServiceID	Amount	PaymentDate
▶	501	11	11	1011	S16	150.00	2025-01-12
	502	12	12	1012	S17	250.00	2025-01-20
	503	13	13	1013	S18	500.00	2025-02-10
	504	12	11	1012	S17	1200.00	2025-03-25
	505	13	11	1013	S18	1500.00	2025-03-26
	506	14	12	1014	S16	6000.00	2025-02-15
	507	15	12	1015	S17	7000.00	2025-02-20
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Creating CourierAssignment table and inserting data

```
CREATE TABLE CourierAssignment (
    AssignmentID INT PRIMARY KEY AUTO_INCREMENT,
    EmployeeID INT,
    CourierID INT,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID) ON DELETE CASCADE,
    FOREIGN KEY (CourierID) REFERENCES Courier(CourierID) ON DELETE CASCADE
);
INSERT INTO CourierAssignment (EmployeeID, CourierID) VALUES
(126, 11),
(127, 12),
(128, 13),
(129, 14),
(130, 15);
```

	AssignmentID	EmployeeID	CourierID
▶	1	126	11
	2	127	12
	3	128	13
	4	129	14
	5	130	15
•	NULL	NULL	NULL

EER:



Task 2: Select, Where

1. List all customers from the User table.

SELECT * FROM User;

UserID	Name	Email	Password	ContactNumber	Address
1011	John Doe	johndoe@gmail.com	John@123	9123456789	123 Main St, Springfield, IL
1012	Jane Smith	janesmith@gmail.com	Jane@456	9234567890	456 Elm St, Metropolis, NY
1013	Alice Johnson	alicej@gmail.com	Alice@789	9345678901	789 Oak St, Gotham, NJ
1014	Bob Brown	bobbrown@gmail.com	Bob@101	9456789012	101 Pine St, Star City, CA
1015	Charlie Davis	charlied@gmail.com	Charlie@202	9567890123	202 Maple St, Central City, TX
1016	Diana Evans	dianae@gmail.com	Diana@303	9678901234	303 Birch St, Coast City, FL
1017	Eve Foster	evef@gmail.com	Eve@404	9789012345	404 Cedar St, National City, WA
1018	Frank Green	frankg@gmail.com	Frank@505	9890123456	505 Walnut St, Bludhaven, OR
1019	Grace Hall	graceh@gmail.com	Grace@606	9901234567	606 Cherry St, Fawcett City, AZ
1020	Henry Irving	henryi@gmail.com	Henry@707	9812345678	707 Spruce St, Ivy Town, CO
NULL	NULL	NULL	NULL	NULL	NULL

2. List all orders for a specific customer using WHERE.

SELECT c.*

FROM Courier c

JOIN User u ON c.SenderName = u.Name

WHERE u.UserID = 1011;

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	DestinationLocationID
11	John Doe	123 Main St, Springfield, IL	Jane Smith	456 Elm St, Metropolis, NY	2.90	In Transit	SPR223344556	2025-04-10	11

```
SELECT * FROM Courier;
```

4. List all packages for a specific order.

```
SELECT *
FROM Courier
WHERE CourierID = 11;
```

5. List all deliveries for a specific courier.

```
SELECT *
FROM Courier
WHERE CourierID = 12 AND Status = 'Delivered';
```

6. List all undelivered packages (Status != 'Delivered').

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

7. List all packages scheduled for delivery today (DeliveryDate = CURDATE()).

```
SELECT *
FROM Courier
WHERE DeliveryDate = CURDATE();
```

[illegible]

8. List all packages with a specific status (WHERE Status = 'In Transit').

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

[illegible]

9. Calculate the total number of packages for each courier using COUNT().

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

	SenderName	TotalPackages
▶	John Doe	1
	Alice Johnson	1
	Charlie Davis	1
	Eve Foster	1
	Grace Hall	1

10. Find the average delivery time for each courier using AVG().

```
SELECT c.CourierID,  
       AVG(DATEDIFF(c.DeliveryDate, p.PaymentDate)) AS AvgDeliveryTime  
FROM Courier c  
JOIN Payment p ON c.CourierID = p.CourierID  
GROUP BY c.CourierID;
```

	CourierID	AvgDeliveryTime
▶	11	88.0000
	12	5.0000
	13	5.0000
	14	35.0000
	15	77.0000

11. List all packages within a specific weight range (WHERE Weight BETWEEN x AND y).

```
SELECT *
FROM Courier
WHERE Weight BETWEEN 2.00 AND 5.00;
```

[illegible]

12. Retrieve employees whose names contain 'John' (WHERE Name LIKE '%John%').

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

	EmployeeID	Name	Email	ContactNumber	Role	Salary
*	NULL	NULL	NULL	NULL	NULL	NULL

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

```
SELECT c.*  
FROM Courier c  
JOIN Payment p ON c.CourierID = p.CourierID  
WHERE p.Amount > 50;
```

	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	DestinationLocationID
▶	11	John Doe	123 Main St, Springfield, IL	Jane Smith	456 Elm St, Metropolis, NY	2.90	In Transit	SPR223344556	2025-04-10	11
	12	Alice Johnson	789 Oak St, Gotham, NJ	Bob Brown	101 Pine St, Star City, CA	1.65	Delivered	GTH998877665	2025-01-25	12
	13	Charlie Davis	202 Maple St, Central City, TX	Diana Evans	303 Birch St, Coast City, FL	3.75	Shipped	CCT119988776	2025-02-15	13
	12	Alice Johnson	789 Oak St, Gotham, NJ	Bob Brown	101 Pine St, Star City, CA	1.65	Delivered	GTH998877665	2025-01-25	12
	13	Charlie Davis	202 Maple St, Central City, TX	Diana Evans	303 Birch St, Coast City, FL	3.75	Shipped	CCT119988776	2025-02-15	13
	14	Eve Foster	404 Cedar St, National City, WA	Frank Green	505 Walnut St, Bludhaven, OR	2.10	Pending	NAT227766554	2025-03-22	11
	15	Grace Hall	606 Cherry St, Fawcett City, AZ	Henry Irving	707 Spruce St, Ivy Town, CO	4.25	Dispatched	FAW337788665	2025-05-08	12

Task 3: GroupBy, Aggregate Functions, Having, Order By, where

14. Find the total number of couriers handled by each employee using COUNT() and GROUP BY.

```
SELECT e.EmployeeID, e.Name, COUNT(ca.CourierID) AS TotalCouriers  
FROM Employee e  
JOIN CourierAssignment ca ON e.EmployeeID = ca.EmployeeID  
GROUP BY e.EmployeeID, e.Name;
```

	EmployeeID	Name	TotalCouriers
▶	126	Oliver Queen	1
	127	Barry Allen	1
	128	Kara Danvers	1
	129	Clark Kent	1
	130	Bruce Wayne	1

15. Calculate the total revenue generated by each location using SUM() and GROUP BY.

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

	LocationID	LocationName	TotalRevenue
▶	11	Springfield Logistics Hub	2850.00
	12	Metropolis Parcel Center	13250.00
	13	Gotham Distribution Hub	500.00

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

```
SELECT I.LocationID, I.LocationName, COUNT(c.CourierID) AS TotalDelivered
FROM Location I
JOIN Courier c ON I.LocationID = c.DestinationLocationID
WHERE c.Status = 'Delivered'
GROUP BY I.LocationID, I.LocationName;
```

	LocationID	LocationName	TotalDelivered
▶	12	Metropolis Parcel Center	1

17. Find the courier with the highest average delivery time using AVG() and ORDER BY.

```
SELECT c.CourierID,
       AVG(DATEDIFF(c.DeliveryDate, p.PaymentDate)) AS AvgDeliveryTime
FROM Courier c
JOIN Payment p ON c.CourierID = p.CourierID
GROUP BY c.CourierID
ORDER BY AvgDeliveryTime DESC
LIMIT 1;
```

	CourierID	AvgDeliveryTime
▶	11	88.0000

18. Find locations with total payments less than a certain amount using HAVING.

```
SELECT I.LocationID, I.LocationName, SUM(p.Amount) AS TotalPayments
FROM Payment p
JOIN Location I ON p.LocationID = I.LocationID
GROUP BY I.LocationID, I.LocationName
HAVING TotalPayments < 3000;
```

	LocationID	LocationName	TotalPayments
▶	11	Springfield Logistics Hub	2850.00
	13	Gotham Distribution Hub	500.00

19. Calculate total payments per location using SUM() and GROUP BY.

```
SELECT I.LocationID, I.LocationName, SUM(p.Amount) AS TotalPayments
FROM Payment p
JOIN Location I ON p.LocationID = I.LocationID
GROUP BY I.LocationID, I.LocationName;
```

	LocationID	LocationName	TotalPayments
▶	11	Springfield Logistics Hub	2850.00
	12	Metropolis Parcel Center	13250.00
	13	Gotham Distribution Hub	500.00

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

```
SELECT p.CourierID, SUM(p.Amount) AS TotalPayment
FROM Payment p
WHERE p.LocationID = 11
GROUP BY p.CourierID
HAVING TotalPayment > 1000;
```

	CourierID	TotalPayment
▶	12	1200.00
	13	1500.00

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

```
SELECT p.CourierID, SUM(p.Amount) AS TotalPayment
FROM Payment p
WHERE p.PaymentDate > '2025-03-01'
GROUP BY p.CourierID
HAVING TotalPayment > 1000;
```

	CourierID	TotalPayment
	12	1200.00
▶	13	1500.00

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

```
SELECT l.LocationID, l.LocationName, SUM(p.Amount) AS TotalPayment
FROM Payment p
JOIN Location l ON p.LocationID = l.LocationID
WHERE p.PaymentDate < '2025-03-01'
GROUP BY l.LocationID, l.LocationName
HAVING TotalPayment > 5000;
```

	LocationID	LocationName	TotalPayment
▶	12	Metropolis Parcel Center	13250.00

Task 4: Inner Join, Full Outer Join, Cross Join, Left Outer Join, Right Outer Join

23. Retrieve Payments with Courier Information using INNER JOIN.

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

	PaymentID	Amount	CourierID	SenderName	ReceiverName
▶	501	150.00	11	John Doe	Jane Smith
	502	250.00	12	Alice Johnson	Bob Brown
	504	1200.00	12	Alice Johnson	Bob Brown
	503	500.00	13	Charlie Davis	Diana Evans
	505	1500.00	13	Charlie Davis	Diana Evans
	506	6000.00	14	Eve Foster	Frank Green
	507	7000.00	15	Grace Hall	Henry Irving

24. Retrieve Payments with Location Information using INNER JOIN.

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

	PaymentID	Amount	LocationID	LocationName
▶	501	150.00	11	Springfield Logistics Hub
	504	1200.00	11	Springfield Logistics Hub
	505	1500.00	11	Springfield Logistics Hub
	502	250.00	12	Metropolis Parcel Center
	506	6000.00	12	Metropolis Parcel Center
	507	7000.00	12	Metropolis Parcel Center
	503	500.00	13	Gotham Distribution Hub

25. Retrieve Payments with Courier and Location Information using INNER JOIN.

```
SELECT p.PaymentID, p.Amount, c.CourierID, c.SenderName, l.LocationName  
FROM Payment p  
JOIN Courier c ON p.CourierID = c.CourierID  
JOIN Location l ON p.LocationID = l.LocationID;
```

	PaymentID	Amount	CourierID	SenderName	LocationName
▶	501	150.00	11	John Doe	Springfield Logistics Hub
	504	1200.00	12	Alice Johnson	Springfield Logistics Hub
	505	1500.00	13	Charlie Davis	Springfield Logistics Hub
	502	250.00	12	Alice Johnson	Metropolis Parcel Center
	506	6000.00	14	Eve Foster	Metropolis Parcel Center
	507	7000.00	15	Grace Hall	Metropolis Parcel Center
	503	500.00	13	Charlie Davis	Gotham Distribution Hub

26. List all payments with courier details.

```
SELECT p.*, c.SenderName, c.ReceiverName  
FROM Payment p  
JOIN Courier c ON p.CourierID = c.CourierID;
```

	PaymentID	CourierID	LocationID	UserID	ServiceID	Amount	PaymentDate	SenderName	ReceiverName
▶	501	11	11	1011	S16	150.00	2025-01-12	John Doe	Jane Smith
	502	12	12	1012	S17	250.00	2025-01-20	Alice Johnson	Bob Brown
	504	12	11	1012	S17	1200.00	2025-03-25	Alice Johnson	Bob Brown
	503	13	13	1013	S18	500.00	2025-02-10	Charlie Davis	Diana Evans
	505	13	11	1013	S18	1500.00	2025-03-26	Charlie Davis	Diana Evans
	506	14	12	1014	S16	6000.00	2025-02-15	Eve Foster	Frank Green
	507	15	12	1015	S17	7000.00	2025-02-20	Grace Hall	Henry Irving

27. Total payments received for each courier using SUM() and GROUP BY.

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

	CourierID	TotalPayments
▶	11	150.00
	12	1450.00
	13	2000.00
	14	6000.00
	15	7000.00

28. List payments made on a specific date using WHERE PaymentDate = 'YYYY-MM-DD'.

```
SELECT *
FROM Payment
WHERE PaymentDate = '2025-01-12';
```

	PaymentID	CourierID	LocationID	UserID	ServiceID	Amount	PaymentDate
▶	501	11	11	1011	S16	150.00	2025-01-12
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

29. Get Courier Information for Each Payment using JOIN.

```
SELECT p.PaymentID, c.*
FROM Payment p
JOIN Courier c ON p.CourierID = c.CourierID;
```

	PaymentID	CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	DestinationLoc
▶	501	11	John Doe	123 Main St, Springfield, IL	Jane Smith	456 Elm St, Metropolis, NY	2.90	In Transit	SPR223344556	2025-04-10	11
	502	12	Alice Johnson	789 Oak St, Gotham, NJ	Bob Brown	101 Pine St, Star City, CA	1.65	Delivered	GTH998877665	2025-01-25	12
	504	12	Alice Johnson	789 Oak St, Gotham, NJ	Bob Brown	101 Pine St, Star City, CA	1.65	Delivered	GTH998877665	2025-01-25	12
	503	13	Charlie Davis	202 Maple St, Central City, TX	Diana Evans	303 Birch St, Coast City, FL	3.75	Shipped	CCT119988776	2025-02-15	13
	505	13	Charlie Davis	202 Maple St, Central City, TX	Diana Evans	303 Birch St, Coast City, FL	3.75	Shipped	CCT119988776	2025-02-15	13
	506	14	Eve Foster	404 Cedar St, National City, WA	Frank Green	505 Walnut St, Bludhaven, OR	2.10	Pending	NAT227766554	2025-03-22	11
	507	15	Grace Hall	606 Cherry St, Fawcett City, AZ	Henry Irving	707 Spruce St, Ivy Town, CO	4.25	Dispatched	FAW337788665	2025-05-08	12

30. Get Payment Details with Location using JOIN.

```
SELECT p.*, l.LocationName
FROM Payment p
JOIN Location l ON p.LocationID = l.LocationID;
```

	PaymentID	CourierID	LocationID	UserID	ServiceID	Amount	PaymentDate	LocationName
▶	501	11	11	1011	S16	150.00	2025-01-12	Springfield Logistics Hub
	504	12	11	1012	S17	1200.00	2025-03-25	Springfield Logistics Hub
	505	13	11	1013	S18	1500.00	2025-03-26	Springfield Logistics Hub
	502	12	12	1012	S17	250.00	2025-01-20	Metropolis Parcel Center
	506	14	12	1014	S16	6000.00	2025-02-15	Metropolis Parcel Center
	507	15	12	1015	S17	7000.00	2025-02-20	Metropolis Parcel Center
	503	13	13	1013	S18	500.00	2025-02-10	Gotham Distribution Hub

31. Calculate Total Payments for Each Courier using SUM() and GROUP BY.

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

	CourierID	TotalPayments
▶	11	150.00
	12	1450.00
	13	2000.00
	14	6000.00
	15	7000.00

32. List Payments Within a Date Range using WHERE PaymentDate BETWEEN 'YYYY-MM-DD' AND 'YYYY-MM-DD'.

```
SELECT *
FROM Payment
WHERE PaymentDate BETWEEN '2025-01-01' AND '2025-12-31';
```

	PaymentID	CourierID	LocationID	UserID	ServiceID	Amount	PaymentDate
▶	501	11	11	1011	S16	150.00	2025-01-12
	502	12	12	1012	S17	250.00	2025-01-20
	503	13	13	1013	S18	500.00	2025-02-10
	504	12	11	1012	S17	1200.00	2025-03-25
	505	13	11	1013	S18	1500.00	2025-03-26
	506	14	12	1014	S16	6000.00	2025-02-15
	507	15	12	1015	S17	7000.00	2025-02-20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

```
SELECT u.UserID, u.Name, c.CourierID, c.SenderName
FROM User u
LEFT JOIN Courier c ON u.Name = c.SenderName
UNION
SELECT u.UserID, u.Name, c.CourierID, c.SenderName
FROM User u
RIGHT JOIN Courier c ON u.Name = c.SenderName;
```

UserID	Name	CourierID	SenderName
1011	John Doe	11	John Doe
1012	Jane Smith	NULL	NULL
1013	Alice Johnson	12	Alice Johnson
1014	Bob Brown	NULL	NULL
1015	Charlie Davis	13	Charlie Davis
1016	Diana Evans	NULL	NULL
1017	Eve Foster	14	Eve Foster
1018	Frank Green	NULL	NULL
1019	Grace Hall	15	Grace Hall
1020	Henry Irving	NULL	NULL

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

```

SELECT c.CourierID, c.SenderName, s.ServiceID, s.ServiceName
FROM Courier c
LEFT JOIN Payment p ON c.CourierID = p.CourierID
LEFT JOIN Service s ON p.ServiceID = s.ServiceID
UNION
SELECT c.CourierID, c.SenderName, s.ServiceID, s.ServiceName
FROM Courier c
RIGHT JOIN Payment p ON c.CourierID = p.CourierID
RIGHT JOIN Service s ON p.ServiceID = s.ServiceID;

```

CourierID	SenderName	ServiceID	ServiceName
11	John Doe	S16	Eco-Friendly Delivery
12	Alice Johnson	S17	Same-Day Express
13	Charlie Davis	S18	Global Priority Shipping
14	Eve Foster	S16	Eco-Friendly Delivery
15	Grace Hall	S17	Same-Day Express

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

```

SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount
FROM Employee e
LEFT JOIN CourierAssignment ca ON e.EmployeeID = ca.EmployeeID
LEFT JOIN Payment p ON ca.CourierID = p.CourierID
UNION
SELECT e.EmployeeID, e.Name, p.PaymentID, p.Amount
FROM Employee e
RIGHT JOIN CourierAssignment ca ON e.EmployeeID = ca.EmployeeID
RIGHT JOIN Payment p ON ca.CourierID = p.CourierID;

```


EmployeeID	Name	PaymentID	Amount
126	Oliver Queen	501	150.00
127	Barry Allen	502	250.00
127	Barry Allen	504	1200.00
128	Kara Danvers	503	500.00
128	Kara Danvers	505	1500.00
129	Clark Kent	506	6000.00
130	Bruce Wayne	507	7000.00

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

```
SELECT u.UserID, u.Name, s.ServiceID, s.ServiceName
FROM User u
CROSS JOIN Service s;
```

UserID	Name	ServiceID	ServiceName
1011	John Doe	S18	Global Priority Shipping
1011	John Doe	S17	Same-Day Express
1011	John Doe	S16	Eco-Friendly Delivery
1012	Jane Smith	S18	Global Priority Shipping
1012	Jane Smith	S17	Same-Day Express
1012	Jane Smith	S16	Eco-Friendly Delivery
1013	Alice Johnson	S18	Global Priority Shipping
1013	Alice Johnson	S17	Same-Day Express
1013	Alice Johnson	S16	Eco-Friendly Delivery
1014	Bob Brown	S18	Global Priority Shipping
1014	Bob Brown	S17	Same-Day Express
1014	Bob Brown	S16	Eco-Friendly Delivery
1015	Charlie Davis	S18	Global Priority Shipping
1015	Charlie Davis	S17	Same-Day Express
1015	Charlie Davis	S16	Eco-Friendly Delivery
1016	Diana Evans	S18	Global Priority Shipping
1016	Diana Evans	S17	Same-Day Express
1016	Diana Evans	S16	Eco-Friendly Delivery
1017	Eve Foster	S18	Global Priority Shipping

37. List all employees and all locations, showing all possible combinations using CROSS JOIN.

```
SELECT e.EmployeeID, e.Name, l.LocationID, l.LocationName
FROM Employee e
CROSS JOIN Location l;
```

EmployeeID	Name	LocationID	LocationName
126	Oliver Queen	13	Gotham Distribution Hub
126	Oliver Queen	12	Metropolis Parcel Center
126	Oliver Queen	11	Springfield Logistics Hub
127	Barry Allen	13	Gotham Distribution Hub
127	Barry Allen	12	Metropolis Parcel Center
127	Barry Allen	11	Springfield Logistics Hub
128	Kara Danvers	13	Gotham Distribution Hub
128	Kara Danvers	12	Metropolis Parcel Center
128	Kara Danvers	11	Springfield Logistics Hub
129	Clark Kent	13	Gotham Distribution Hub
129	Clark Kent	12	Metropolis Parcel Center
129	Clark Kent	11	Springfield Logistics Hub
130	Bruce Wayne	13	Gotham Distribution Hub
130	Bruce Wayne	12	Metropolis Parcel Center
130	Bruce Wayne	11	Springfield Logistics Hub

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

```
SELECT CourierID, SenderName, SenderAddress  
FROM Courier;
```

	CourierID	SenderName	SenderAddress
▶	11	John Doe	123 Main St, Springfield, IL
	12	Alice Johnson	789 Oak St, Gotham, NJ
	13	Charlie Davis	202 Maple St, Central City, TX
	14	Eve Foster	404 Cedar St, National City, WA
	15	Grace Hall	606 Cherry St, Fawcett City, AZ
•	NULL	NULL	NULL

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

```
SELECT CourierID, ReceiverName, ReceiverAddress  
FROM Courier;
```

	CourierID	ReceiverName	ReceiverAddress
▶	11	Jane Smith	456 Elm St, Metropolis, NY
	12	Bob Brown	101 Pine St, Star City, CA
	13	Diana Evans	303 Birch St, Coast City, FL
	14	Frank Green	505 Walnut St, Bludhaven, OR
	15	Henry Irving	707 Spruce St, Ivy Town, CO
•	NULL	NULL	NULL

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

```
SELECT c.CourierID, c.SenderName, s.ServiceID, s.ServiceName  
FROM Courier c  
LEFT JOIN Payment p ON c.CourierID = p.CourierID  
LEFT JOIN Service s ON p.ServiceID = s.ServiceID;
```

	CourierID	SenderName	ServiceID	ServiceName
▶	11	John Doe	S16	Eco-Friendly Delivery
	12	Alice Johnson	S17	Same-Day Express
	12	Alice Johnson	S17	Same-Day Express
	13	Charlie Davis	S18	Global Priority Shipping
	13	Charlie Davis	S18	Global Priority Shipping
	14	Eve Foster	S16	Eco-Friendly Delivery
	15	Grace Hall	S17	Same-Day Express

41. Retrieve a list of employees and the number of couriers assigned to each employee using COUNT() and GROUP BY.

```
SELECT e.EmployeeID, e.Name, COUNT(ca.CourierID) AS NumberOfCouriers  
FROM Employee e  
LEFT JOIN CourierAssignment ca ON e.EmployeeID = ca.EmployeeID  
GROUP BY e.EmployeeID, e.Name;
```

EmployeeID	Name	NumberOfCouriers
126	Oliver Queen	1
127	Barry Allen	1
128	Kara Danvers	1
129	Clark Kent	1
130	Bruce Wayne	1

42. Retrieve a list of locations and the total payment amount received at each location using SUM() and GROUP BY.

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

LocationID	LocationName	TotalPayments
11	Springfield Logistics Hub	2850.00
12	Metropolis Parcel Center	13250.00
13	Gotham Distribution Hub	500.00

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

```
SELECT c1.CourierID, c1.SenderName, c1.SenderAddress
FROM Courier c1
JOIN Courier c2 ON c1.SenderName = c2.SenderName AND c1.CourierID <> c2.CourierID;
```

CourierID	SenderName	SenderAddress
-----------	------------	---------------

44. List all employees who share the same role using GROUP BY Role and HAVING COUNT(*) > 1.

```
SELECT e1.EmployeeID, e1.Name, e1.Role
FROM Employee e1
JOIN Employee e2 ON e1.Role = e2.Role AND e1.EmployeeID <> e2.EmployeeID;
```

EmployeeID	Name	Role
------------	------	------

45. Retrieve all payments made for couriers sent from the same location using JOIN and WHERE.

```
SELECT p1.PaymentID, p1.Amount, p1.PaymentDate, l.LocationName
FROM Payment p1
JOIN Payment p2 ON p1.LocationID = p2.LocationID AND p1.PaymentID <> p2.PaymentID
JOIN Location l ON p1.LocationID = l.LocationID;
```

	PaymentID	Amount	PaymentDate	LocationName
▶	501	150.00	2025-01-12	Springfield Logistics Hub
	501	150.00	2025-01-12	Springfield Logistics Hub
	504	1200.00	2025-03-25	Springfield Logistics Hub
	504	1200.00	2025-03-25	Springfield Logistics Hub
	505	1500.00	2025-03-26	Springfield Logistics Hub
	505	1500.00	2025-03-26	Springfield Logistics Hub
	502	250.00	2025-01-20	Metropolis Parcel Center
	502	250.00	2025-01-20	Metropolis Parcel Center
	506	6000.00	2025-02-15	Metropolis Parcel Center
	506	6000.00	2025-02-15	Metropolis Parcel Center
	507	7000.00	2025-02-20	Metropolis Parcel Center
	507	7000.00	2025-02-20	Metropolis Parcel Center

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

```
SELECT c1.CourierID, c1.SenderName, c1.SenderAddress
```

```
FROM Courier c1
```

```
JOIN Courier c2 ON c1.SenderAddress = c2.SenderAddress AND c1.CourierID <> c2.CourierID;
```

CourierID	SenderName	SenderAddress
-----------	------------	---------------

47. List employees and the number of couriers they have delivered using COUNT() and GROUP BY.

```
SELECT e.EmployeeID, e.Name, COUNT(c.CourierID) AS NumberOfDeliveries
```

```
FROM Employee e
```

```
LEFT JOIN CourierAssignment ca ON e.EmployeeID = ca.EmployeeID
```

```
LEFT JOIN Courier c ON ca.CourierID = c.CourierID AND c.Status = 'Delivered'
```

```
GROUP BY e.EmployeeID, e.Name;
```

	EmployeeID	Name	NumberOfDeliveries
▶	126	Oliver Queen	0
	127	Barry Allen	1
	128	Kara Danvers	0
	129	Clark Kent	0
	130	Bruce Wayne	0

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

```
SELECT c.CourierID, c.SenderName, p.Amount, s.ServiceName, s.Cost
```

```
FROM Courier c
```

```
JOIN Payment p ON c.CourierID = p.CourierID
```

```
JOIN Service s ON p.ServiceID = s.ServiceID
```

```
WHERE p.Amount > s.Cost;
```

	CourierID	SenderName	Amount	ServiceName	Cost
▶	14	Eve Foster	6000.00	Eco-Friendly Delivery	180.00
	12	Alice Johnson	1200.00	Same-Day Express	550.00
	15	Grace Hall	7000.00	Same-Day Express	550.00

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 using a subquery with AVG().

```
SELECT CourierID, SenderName, Weight
FROM Courier
WHERE Weight > (SELECT AVG(Weight) FROM Courier);
```

	CourierID	SenderName	Weight
▶	13	Charlie Davis	3.75
	15	Grace Hall	4.25
*	NULL	NULL	NULL

50. Find the names of all employees who have a salary greater than the average salary using a subquery with AVG().

```
SELECT EmployeeID, Name, Salary
FROM Employee
WHERE Salary > (SELECT AVG(Salary) FROM Employee);
```

	EmployeeID	Name	Salary
▶	126	Oliver Queen	95000.00
	127	Barry Allen	72000.00
*	NULL	NULL	NULL

51. Find the total cost of all courier services where the cost is less than the maximum cost using SUM() and a subquery with MAX().

```
SELECT SUM(Cost) AS TotalCost
FROM Service
WHERE Cost < (SELECT MAX(Cost) FROM Service);
```

	TotalCost
▶	730.00

52. Find all couriers that have been paid for using EXISTS or JOIN with the Payment table.

```
SELECT DISTINCT c.CourierID, c.SenderName
FROM Courier c
JOIN Payment p ON c.CourierID = p.CourierID;
```

	CourierID	SenderName
▶	11	John Doe
	12	Alice Johnson
	13	Charlie Davis
	14	Eve Foster
	15	Grace Hall

53. Find the locations where the maximum payment amount was made using MAX() and GROUP BY.

```
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);
```

	LocationID	LocationName	Amount
▶	12	Metropolis Parcel Center	7000.00

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

```
SELECT CourierID, SenderName, Weight
FROM Courier
WHERE Weight > ALL (
    SELECT Weight
    FROM Courier
    WHERE SenderName = 'John Doe'
);
```

	CourierID	SenderName	Weight
▶	13	Charlie Davis	3.75
	15	Grace Hall	4.25
*	NULL	NULL	NULL