Task1 Database Design

```
CREATE TABLE User (
    UserID INT PRIMARY KEY,
    Name VARCHAR(255),
    Email VARCHAR(255) UNIQUE,
    Password VARCHAR(255),
    ContactNumber VARCHAR(20),
    Address TEXT
);


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
);

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

CREATE TABLE Employee (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(255),
    Email VARCHAR(255) UNIQUE,
```

```
    ContactNumber VARCHAR(20),
    Role VARCHAR(50),
    Salary DECIMAL(10, 2)
);

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

CREATE TABLE Payment (
    PaymentID INT PRIMARY KEY,
    CourierID INT,
    LocationID INT,
    Amount DECIMAL(10, 2),
    PaymentDate DATE,
    FOREIGN KEY (CourierID) REFERENCES Courier(CourierID),
    FOREIGN KEY (LocationID) REFERENCES Location(LocationID)
);



INSERT INTO User (UserID, Name, Email, Password, ContactNumber, Address)
VALUES
(1, 'John Doe', 'john@example.com', 'password123', '1234567890', '123 Main St'),
(2, 'Jane Smith', 'jane@example.com', 'password456', '0987654321', '456 Elm St'),
(3, 'Alice Johnson', 'alice@example.com', 'password789', '9876543210', '789 Oak St'),
(4, 'Bob Brown', 'bob@example.com', 'passwordabc', '4561237890', '321 Maple St'),
(5, 'Eve Wilson', 'eve@example.com', 'passwordefg', '7894561230', '654 Pine St');



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

(1, 'John Doe', '123 Main St', 'Jane Smith', '456 Elm St', 3.5, 'In transit', 'ABC123', '2024-04-20'),

(2, 'Alice Johnson', '789 Oak St', 'Bob Brown', '321 Maple St', 2.0, 'Delivered', 'DEF456', '2024-04-18'),

(3, 'Eve Wilson', '654 Pine St', 'John Doe', '123 Main St', 1.8, 'Pending', 'GHI789', NULL),

(4, 'Bob Brown', '321 Maple St', 'Alice Johnson', '789 Oak St', 4.2, 'In transit', 'JKL012', '2024-04-22'),

(5, 'Jane Smith', '456 Elm St', 'Eve Wilson', '654 Pine St', 5.0, 'Pending', 'MNO345', NULL);

INSERT INTO CourierServices (ServiceID, ServiceName, Cost)

VALUES

(1, 'Standard', 10.00),

(2, 'Express', 20.00),

(3, 'Same Day', 30.00),

(4, 'Overnight', 15.00),

(5, 'International', 50.00);

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

VALUES

(1, 'Mark Johnson', 'mark@example.com', '1239876540', 'Manager', 50000.00),

(2, 'Emily Brown', 'emily@example.com', '9876543210', 'Clerk', 30000.00),

(3, 'Michael Smith', 'michael@example.com', '4567890123', 'Driver', 35000.00),

(4, 'Jessica Wilson', 'jessica@example.com', '3216549870', 'Administrator', 40000.00),

(5, 'David Davis', 'david@example.com', '6541237890', 'Courier', 32000.00);

INSERT INTO Location (LocationID, LocationName, Address)

VALUES

(1, 'Office 1', '123 Main St'),

(2, 'Office 2', '456 Elm St'),

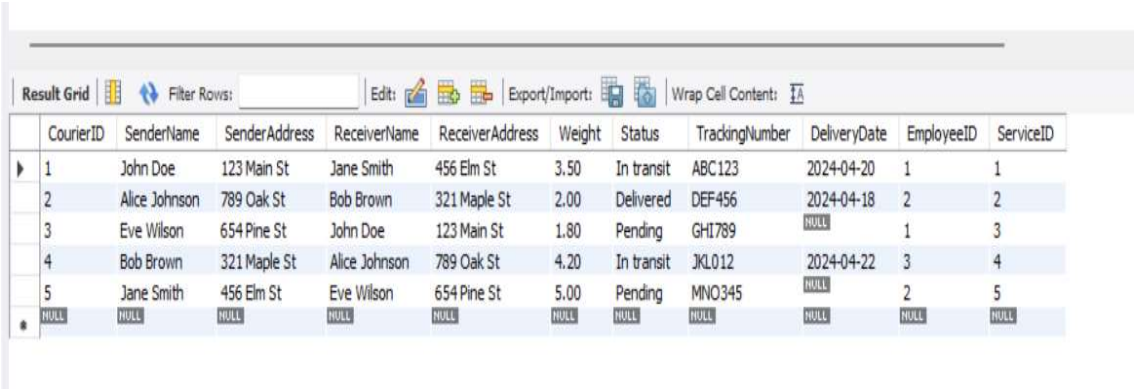
(3, 'Office 3', '789 Oak St'),

(4, 'Office 4', '321 Maple St'),

(5, 'Office 5', '654 Pine St');

```
INSERT INTO Payment (PaymentID, CourierID, LocationID, Amount, PaymentDate)
VALUES
(1, 1, 2, 15.00, '2024-04-18'),
(2, 2, 3, 20.00, '2024-04-19'),
(3, 3, 1, 25.00, '2024-04-20'),
(4, 4, 2, 18.00, '2024-04-21'),
(5, 5, 3, 22.00, '2024-04-22');



ALTER TABLE courier
ADD COLUMN ServiceID INT,
ADD CONSTRAINT ServiceID FOREIGN KEY (ServiceID) REFERENCES
CourierServices(ServiceID);


UPDATE Courier
SET ServiceID =
  CASE
    WHEN CourierID = 1 THEN 1
    WHEN CourierID = 2 THEN 2
    WHEN CourierID = 3 THEN 3
    WHEN CourierID = 4 THEN 4
    WHEN CourierID = 5 THEN 5
    ELSE NULL
  END
WHERE CourierID > 0;
```

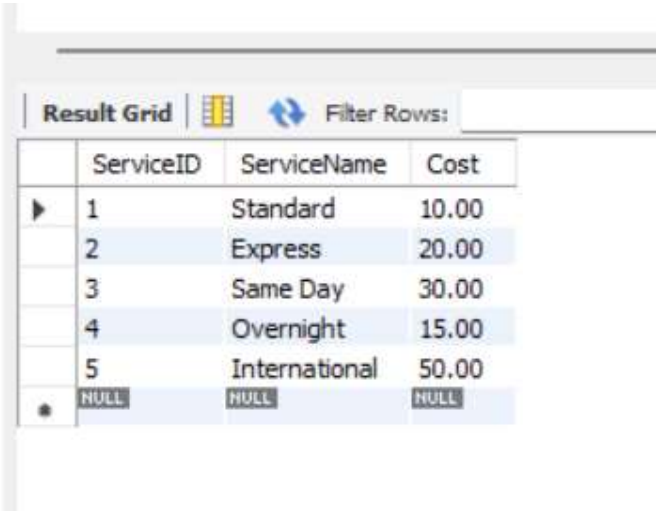
1) **Courier Table**



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	EmployeeID	ServiceID
1	John Doe	123 Main St	Jane Smith	456 Elm St	3.50	In transit	ABC123	2024-04-20	1	1
2	Alice Johnson	789 Oak St	Bob Brown	321 Maple St	2.00	Delivered	DEF456	2024-04-18	2	2
3	Eve Wilson	654 Pine St	John Doe	123 Main St	1.80	Pending	GHI789	NULL	1	3
4	Bob Brown	321 Maple St	Alice Johnson	789 Oak St	4.20	In transit	JKL012	2024-04-22	3	4
5	Jane Smith	456 Elm St	Eve Wilson	654 Pine St	5.00	Pending	MNO345	NULL	2	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

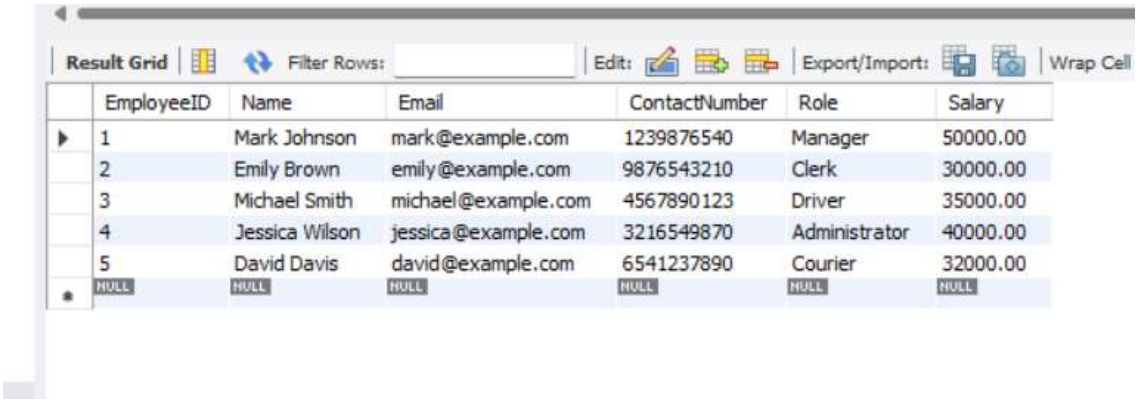
2) **Courier Services Table**



Result Grid | Filter Rows:

ServiceID	ServiceName	Cost
1	Standard	10.00
2	Express	20.00
3	Same Day	30.00
4	Overnight	15.00
5	International	50.00
NULL	NULL	NULL

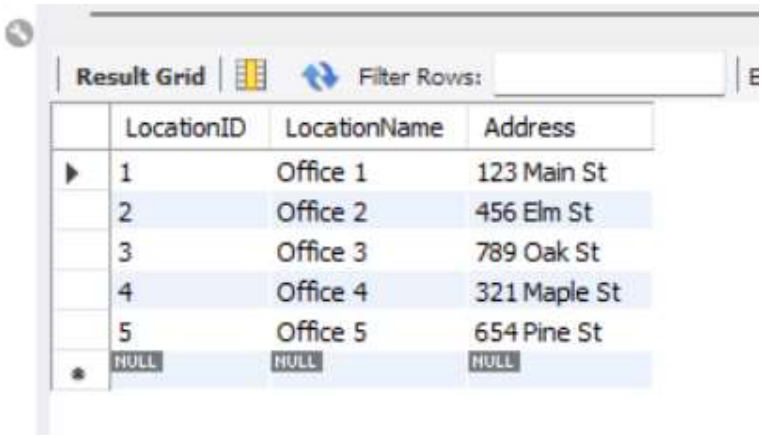
3) **Employee Table**



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

EmployeeID	Name	Email	ContactNumber	Role	Salary
1	Mark Johnson	mark@example.com	1239876540	Manager	50000.00
2	Emily Brown	emily@example.com	9876543210	Clerk	30000.00
3	Michael Smith	michael@example.com	4567890123	Driver	35000.00
4	Jessica Wilson	jessica@example.com	3216549870	Administrator	40000.00
5	David Davis	david@example.com	6541237890	Courier	32000.00
NULL	NULL	NULL	NULL	NULL	NULL

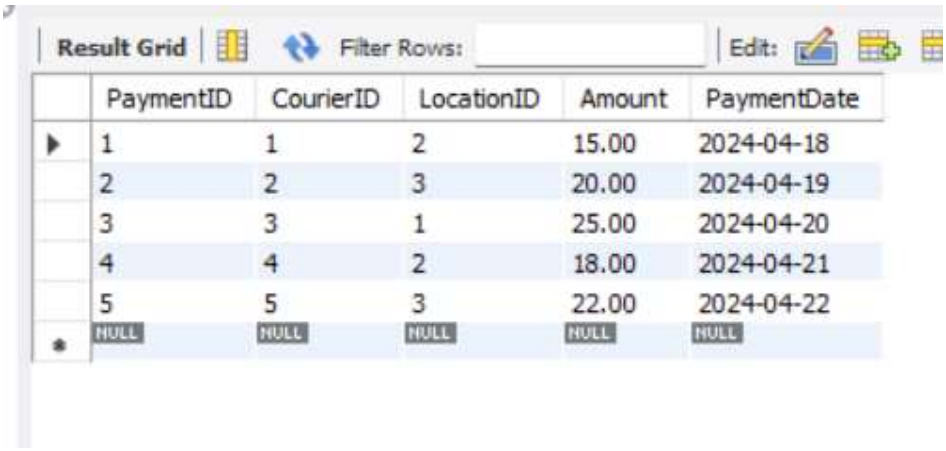
4) **Location Table**



Result Grid | Filter Rows:

LocationID	LocationName	Address
1	Office 1	123 Main St
2	Office 2	456 Elm St
3	Office 3	789 Oak St
4	Office 4	321 Maple St
5	Office 5	654 Pine St
NULL	NULL	NULL

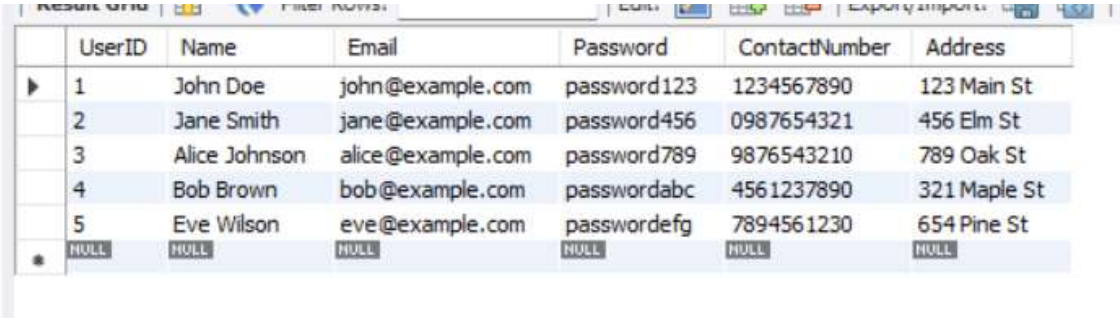
5) **Payment Table**



Result Grid | Filter Rows: | Edit:

PaymentID	CourierID	LocationID	Amount	PaymentDate
1	1	2	15.00	2024-04-18
2	2	3	20.00	2024-04-19
3	3	1	25.00	2024-04-20
4	4	2	18.00	2024-04-21
5	5	3	22.00	2024-04-22
NULL	NULL	NULL	NULL	NULL

6) **User Table**

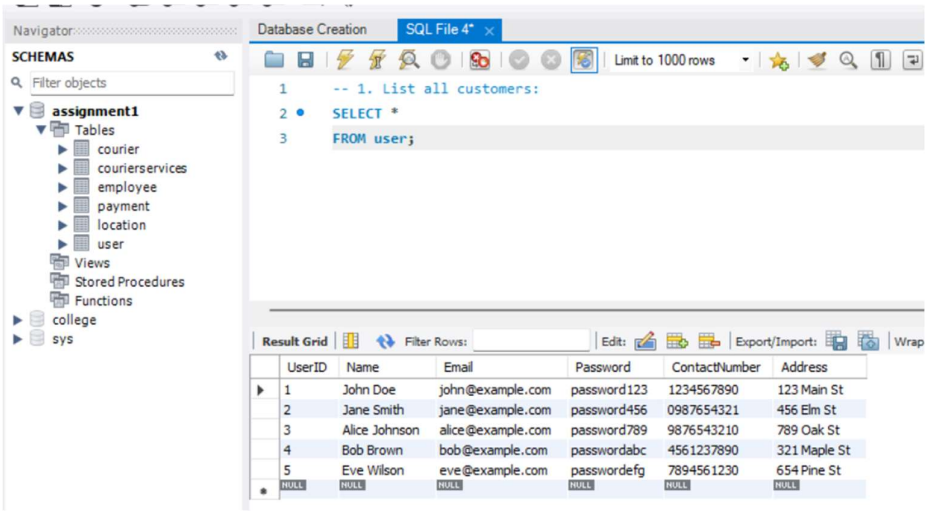


UserID	Name	Email	Password	ContactNumber	Address
1	John Doe	john@example.com	password123	1234567890	123 Main St
2	Jane Smith	jane@example.com	password456	0987654321	456 Elm St
3	Alice Johnson	alice@example.com	password789	9876543210	789 Oak St
4	Bob Brown	bob@example.com	passwordabc	4561237890	321 Maple St
5	Eve Wilson	eve@example.com	passwordefg	7894561230	654 Pine St
NULL	NULL	NULL	NULL	NULL	NULL

-- 1. List all customers:

SELECT *

FROM user;

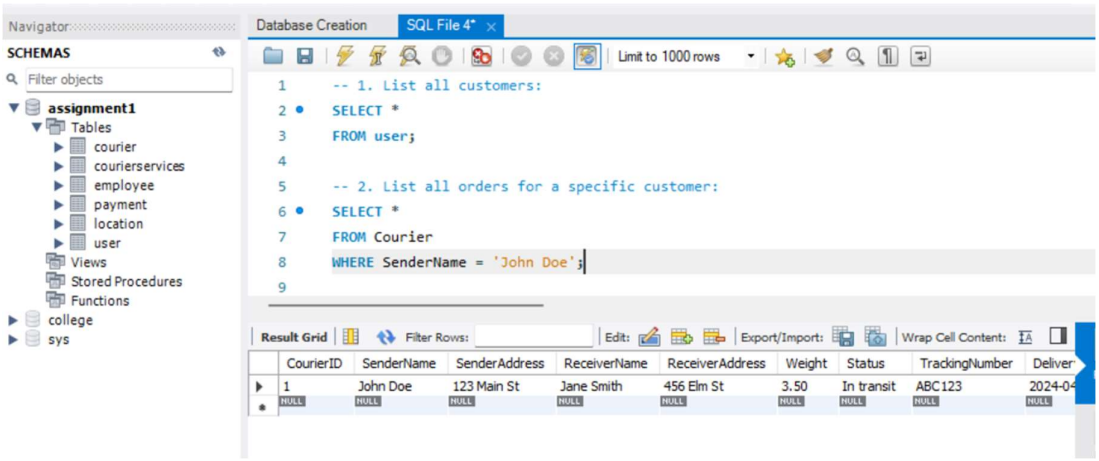


-- 2. List all orders for a specific customer:

SELECT *

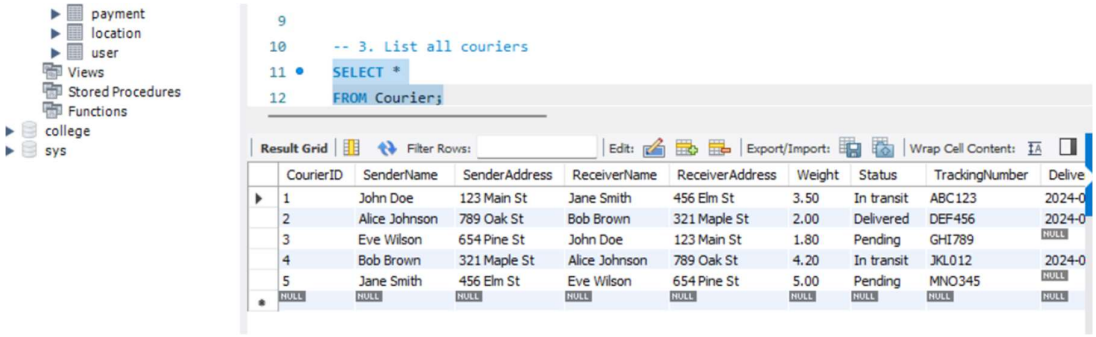
FROM Courier

WHERE SenderName = 'John Doe';



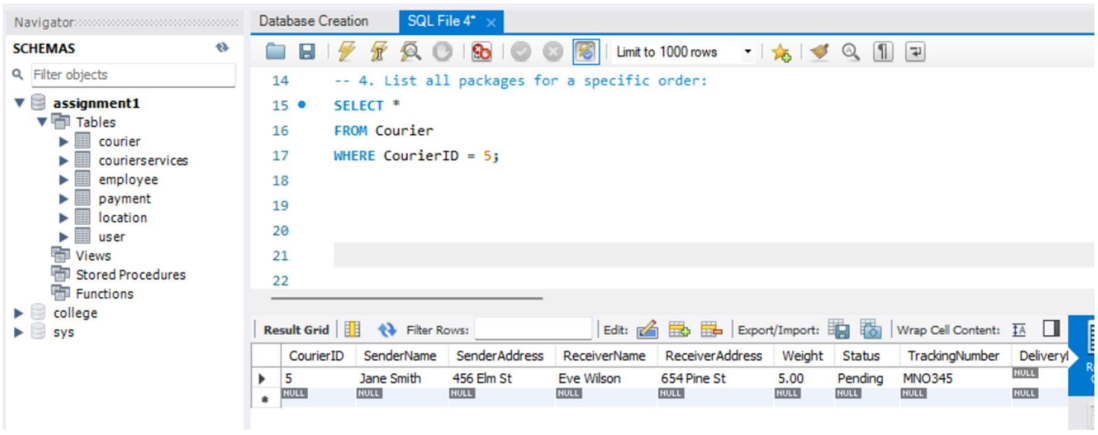
-- 3. List all couriers

```
SELECT *
FROM Courier;
```



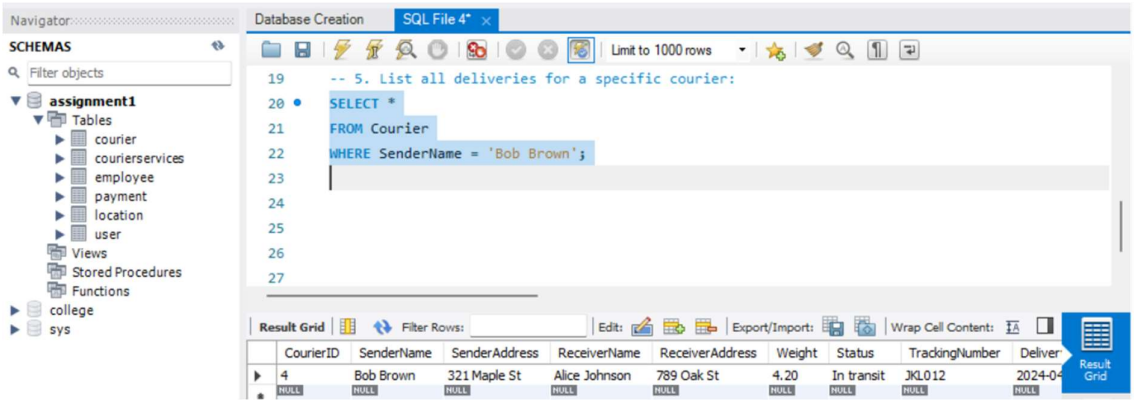
-- 4. List all packages for a specific order:

```
SELECT *
FROM Courier
WHERE CourierID = 5;
```



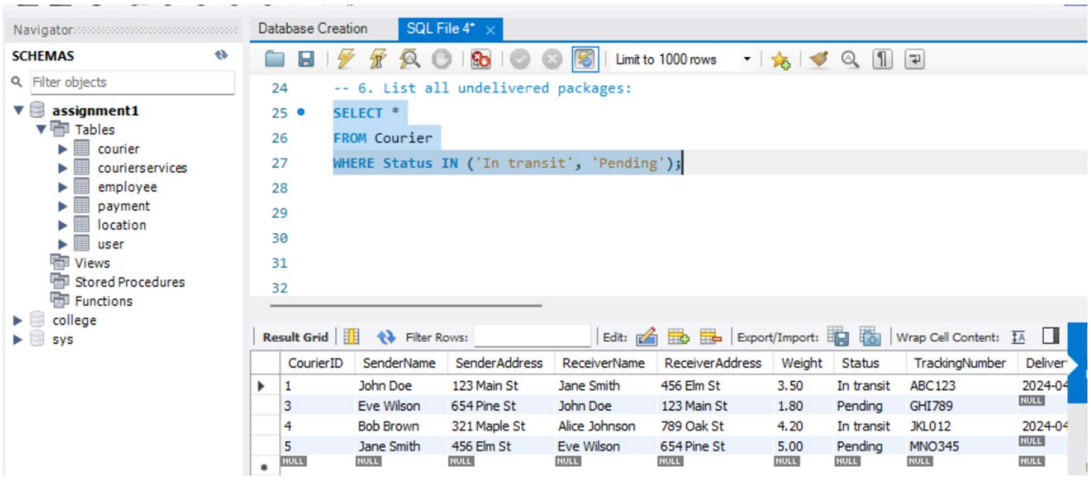
-- 5. List all deliveries for a specific courier:

```
SELECT *
FROM Courier
WHERE SenderName = 'Bob Brown';
```



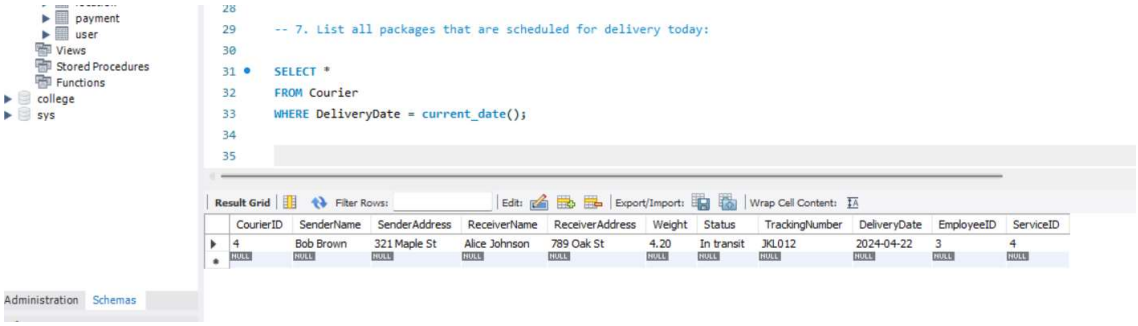
-- 6. List all undelivered packages:

```
SELECT *
FROM Courier
WHERE Status IN ('In transit', 'Pending');
```



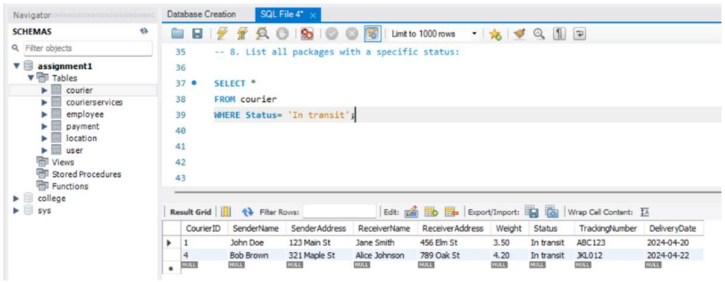
-- 7. List all packages that are scheduled for delivery today:

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



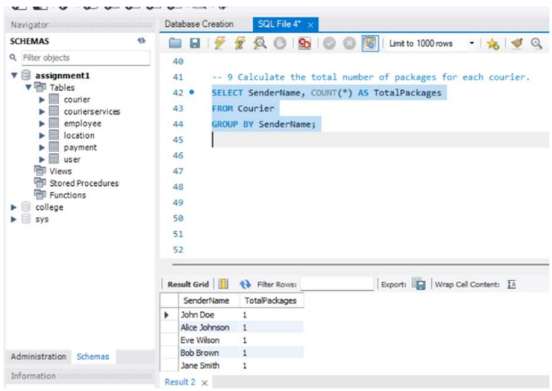
-- 8. List all packages with a specific status:

```
SELECT *
FROM courier
WHERE Status= 'In transit';
```



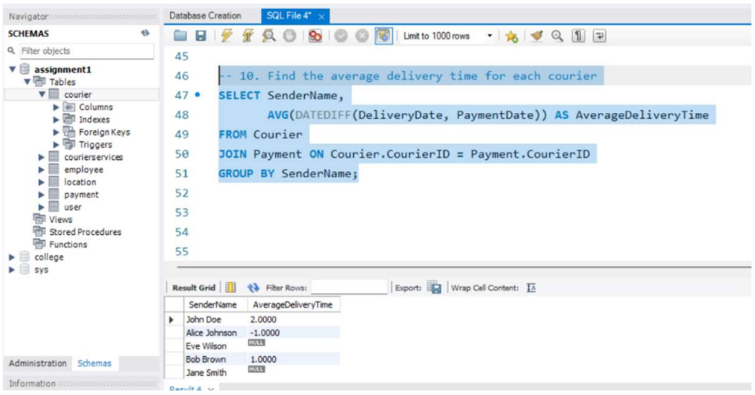
-- 9 Calculate the total number of packages for each courier.

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



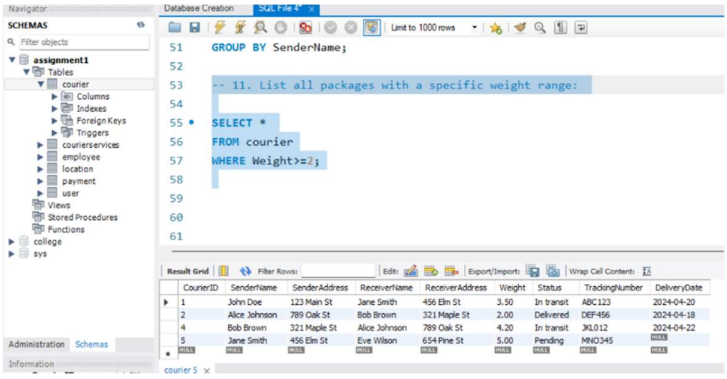
-- 10. Find the average delivery time for each courier

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



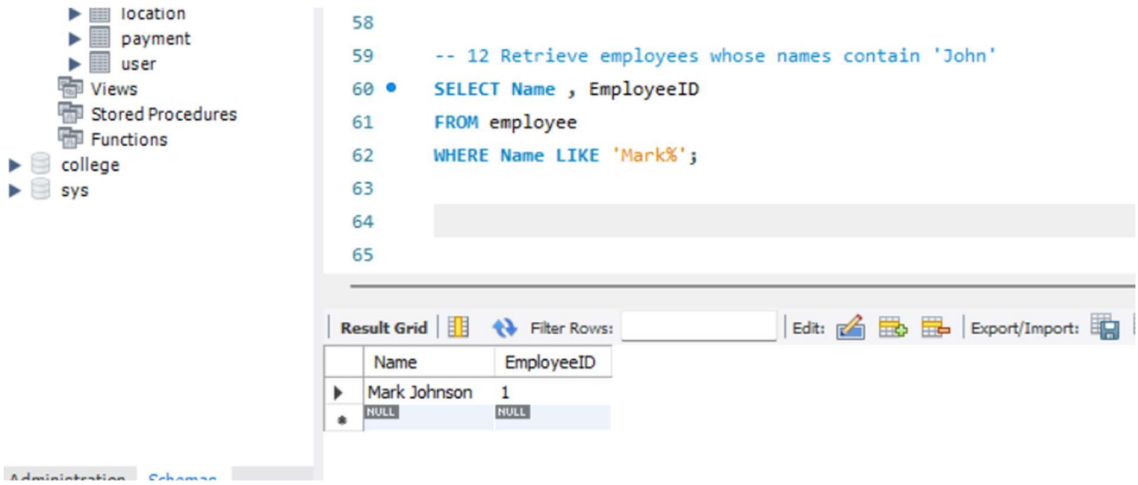
-- 11. List all packages with a specific weight range:

```
SELECT *
FROM courier
WHERE Weight>=2;
```



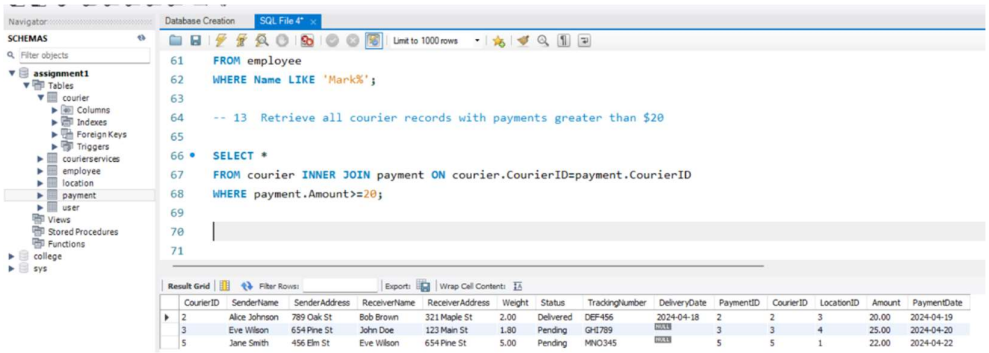
-- 12 Retrieve employees whose names contain 'John'

```
SELECT Name , EmployeeID
FROM employee
WHERE Name LIKE 'Mark%';
```



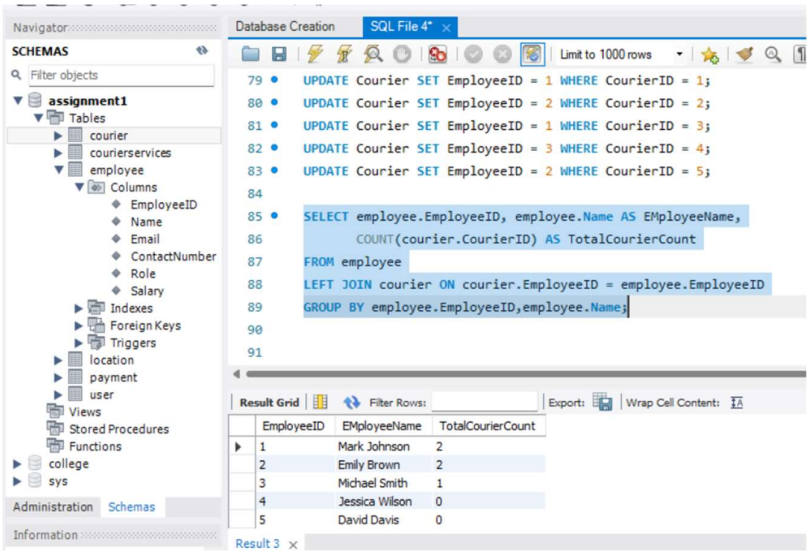
-- 13 Retrieve all courier records with payments greater than $20

```
SELECT *
FROM courier INNER JOIN payment ON courier.CourierID=payment.CourierID
WHERE payment.Amount>=20;
```



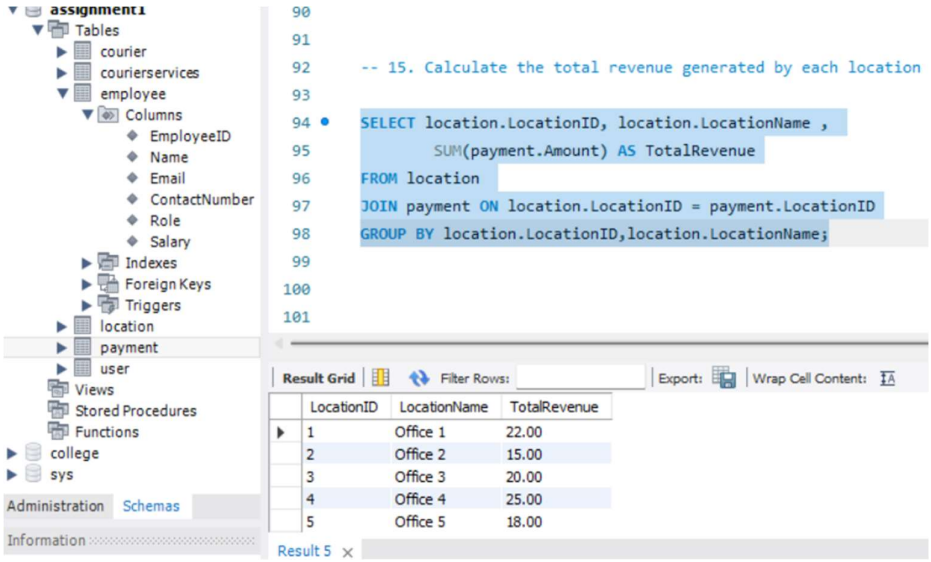
-- **14. Find the total number of couriers handled by each employee**

```
ALTER TABLE courier
ADD COLUMN EmployeeID INT;

ALTER TABLE courier
ADD CONSTRAINT EmployeeID FOREIGN KEY (EmployeeID) REFERENCES employee
(EmployeeID);

UPDATE Courier SET EmployeeID = 1 WHERE CourierID = 1;
UPDATE Courier SET EmployeeID = 2 WHERE CourierID = 2;
UPDATE Courier SET EmployeeID = 1 WHERE CourierID = 3;
UPDATE Courier SET EmployeeID = 3 WHERE CourierID = 4;
UPDATE Courier SET EmployeeID = 2 WHERE CourierID = 5;

SELECT employee.EmployeeID, employee.Name AS EMployeeName,
        COUNT(courier.CourierID) AS TotalCourierCount
FROM employee
LEFT JOIN courier ON courier.EmployeeID = employee.EmployeeID
GROUP BY employee.EmployeeID,employee.Name;
```



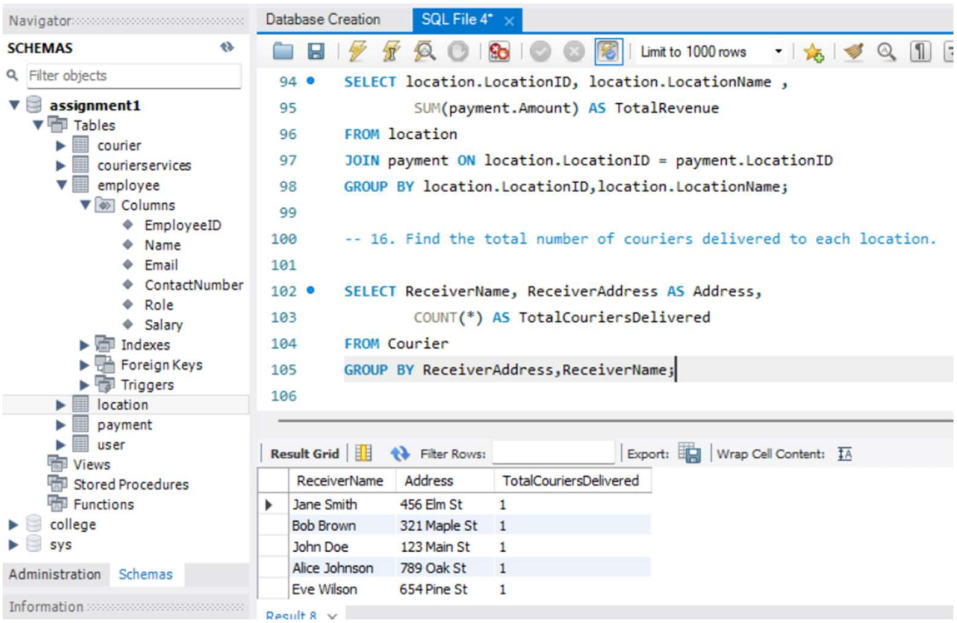
-- 15. Calculate the total revenue generated by each location

```
SELECT location.LocationID, location.LocationName ,
                SUM(payment.Amount) AS TotalRevenue
FROM location
JOIN payment ON location.LocationID = payment.LocationID
GROUP BY location.LocationID,location.LocationName;
```



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

```
SELECT ReceiverName, ReceiverAddress AS Address,
                COUNT(*) AS TotalCouriersDelivered
FROM Courier
GROUP BY ReceiverAddress,ReceiverName;
```



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

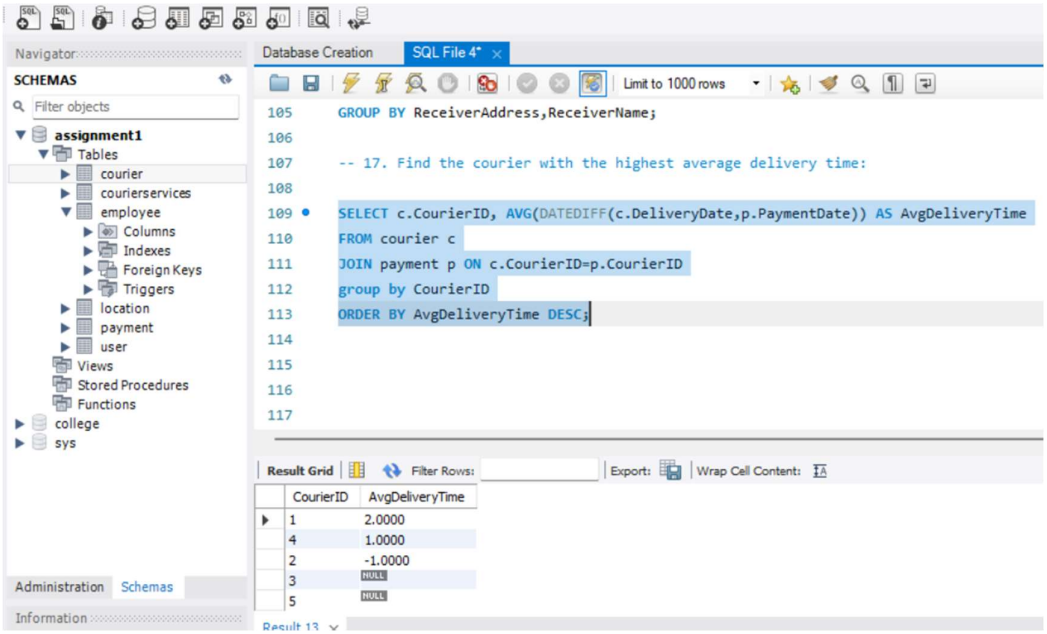
SELECT c.CourierID, AVG(DATEDIFF(c.DeliveryDate,p.PaymentDate)) AS AvgDeliveryTime

FROM courier c

JOIN payment p ON c.CourierID=p.CourierID

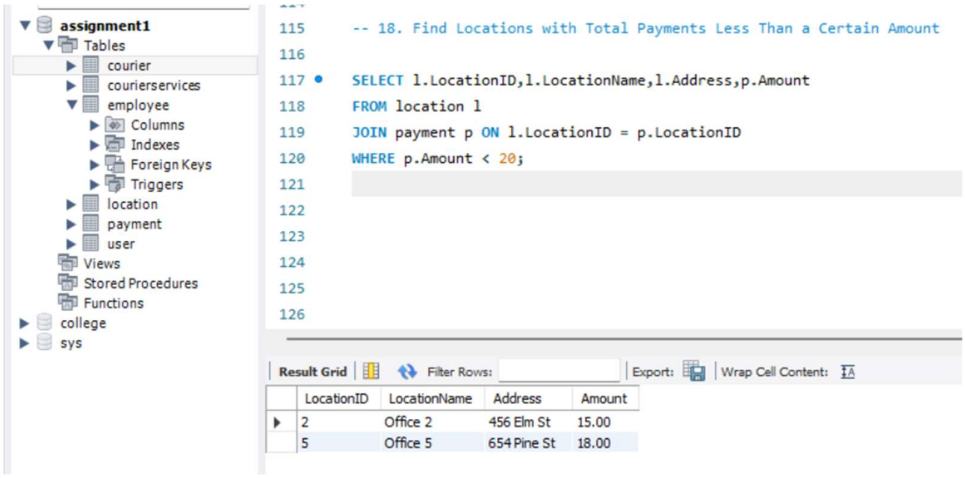
group by CourierID

ORDER BY AvgDeliveryTime DESC;



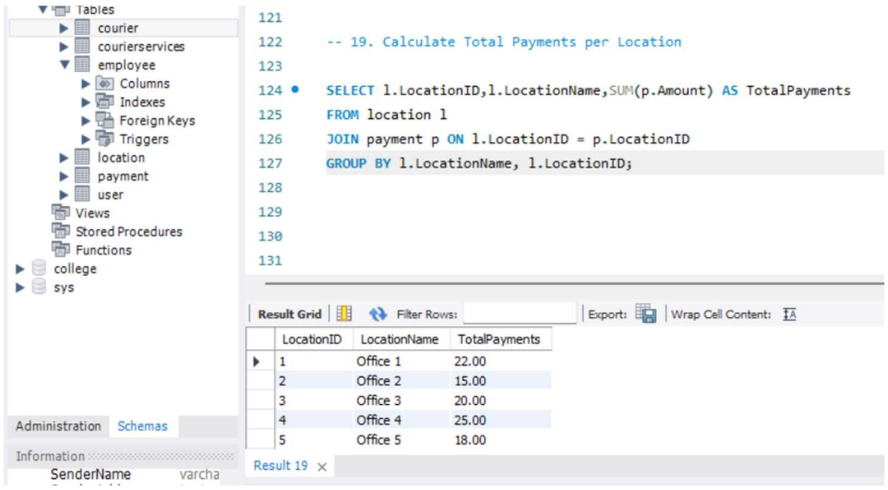
-- 18. Find Locations with Total Payments Less Than a Certain Amount

```
SELECT l.LocationID,l.LocationName,l.Address,p.Amount
FROM location l
JOIN payment p ON l.LocationID = p.LocationID
WHERE p.Amount < 20;
```



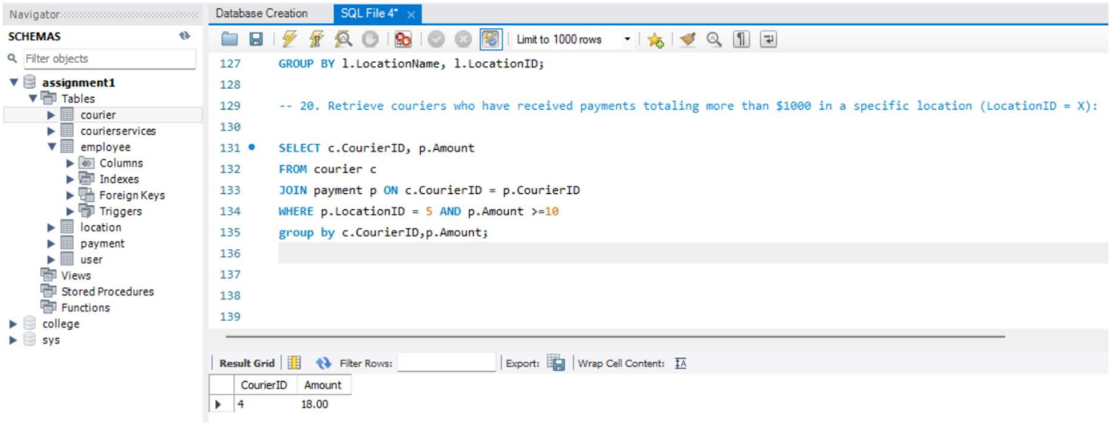
-- 19. Calculate Total Payments per Location

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



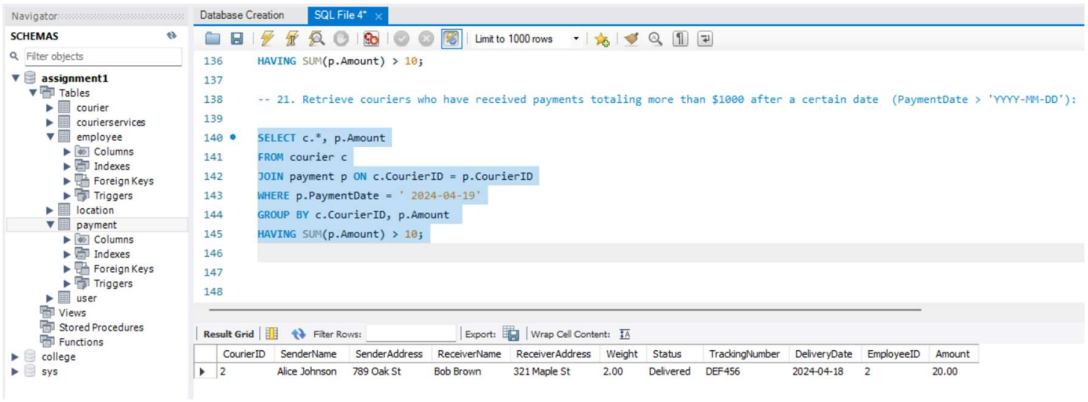
-- 20. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = X):

```
SELECT c.*, p.Amount
FROM courier c
JOIN payment p ON c.CourierID = p.CourierID
WHERE p.LocationID = 2
GROUP BY c.CourierID, p.Amount
HAVING SUM(p.Amount) > 10;
```



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

```
SELECT c.*, p.Amount
FROM courier c
JOIN payment p ON c.CourierID = p.CourierID
WHERE p.PaymentDate > ' 2024-04-19'
GROUP BY c.CourierID, p.Amount
HAVING SUM(p.Amount) > 10;
```



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

SELECT l.LocationName, SUM(p.Amount) AS TotalAmountReceived

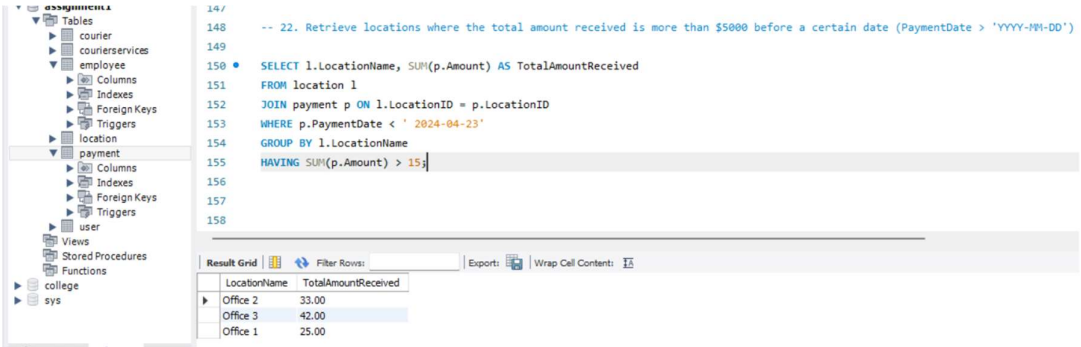
FROM location l

JOIN payment p ON l.LocationID = p.LocationID

WHERE p.PaymentDate < ' 2024-04-23'

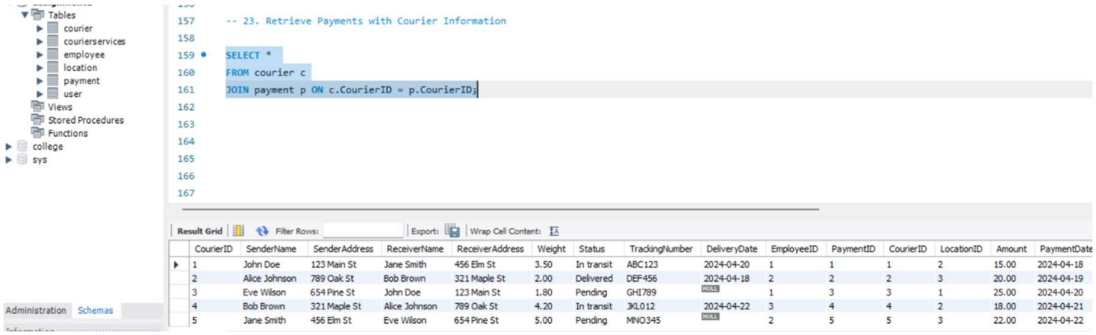
GROUP BY l.LocationName

HAVING SUM(p.Amount) > 15;



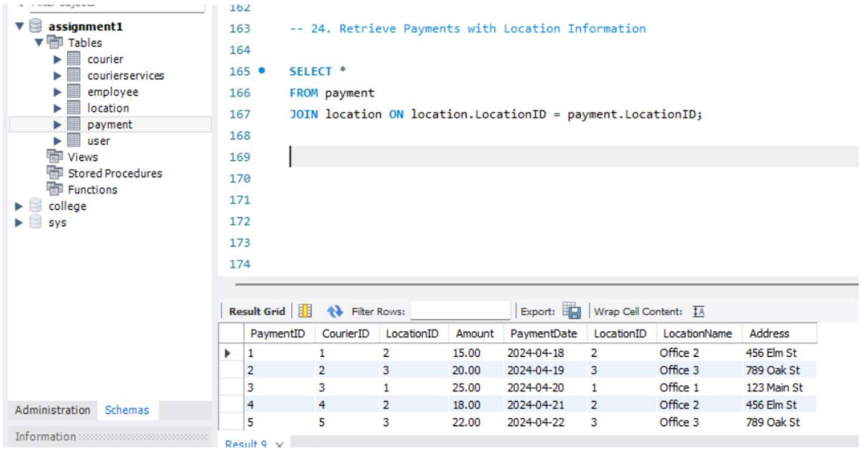
--- 23. Retrieve Payments with Courier Information

```
SELECT *
FROM courier
JOIN payment  ON courier.CourierID = payment.CourierID;
```



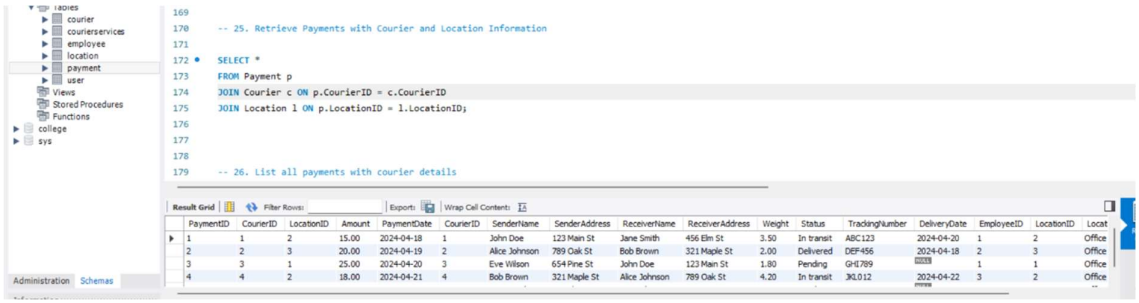
-- 24. Retrieve Payments with Location Information

```
SELECT *
FROM payment
JOIN location ON location.LocationID = payment.LocationID;
```



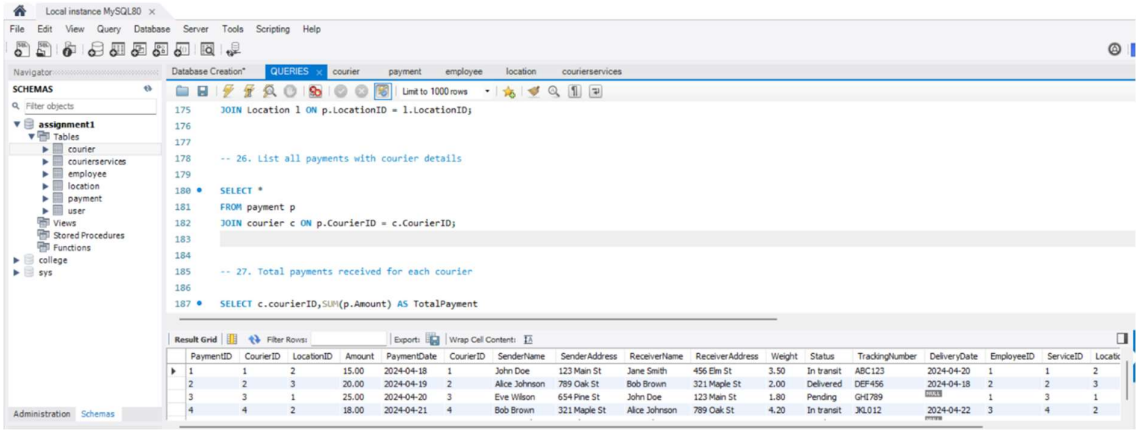
-- 25. Retrieve Payments with Courier and Location Information

```
SELECT *
FROM Payment p
JOIN Courier c ON p.CourierID = c.CourierID
JOIN Location l ON p.LocationID = l. LocationID;
```



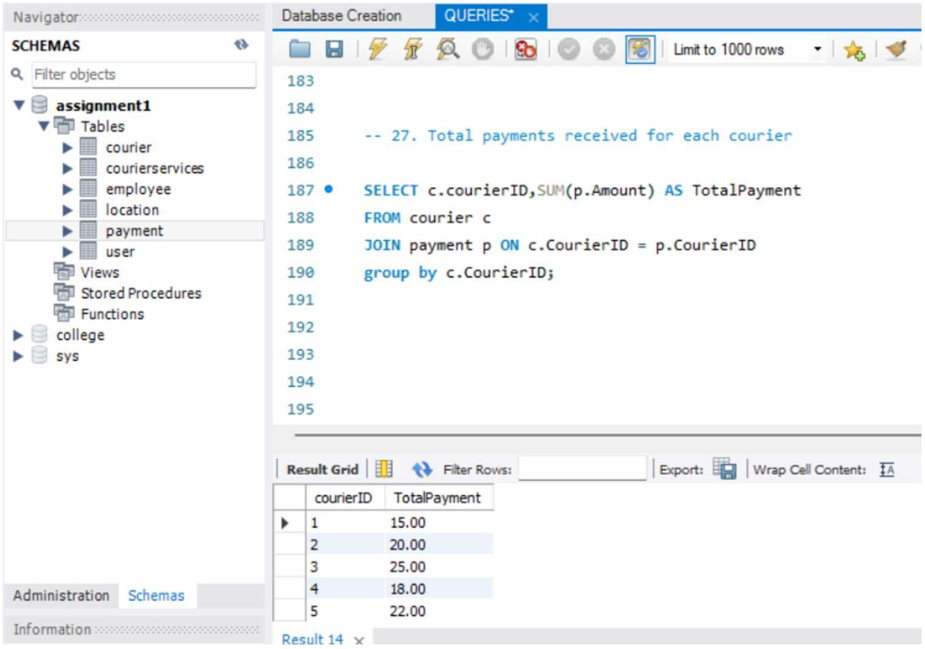
-- 26. List all payments with courier details

```
SELECT *
FROM payment p
JOIN courier c ON p.CourierID = c.CourierID;
```



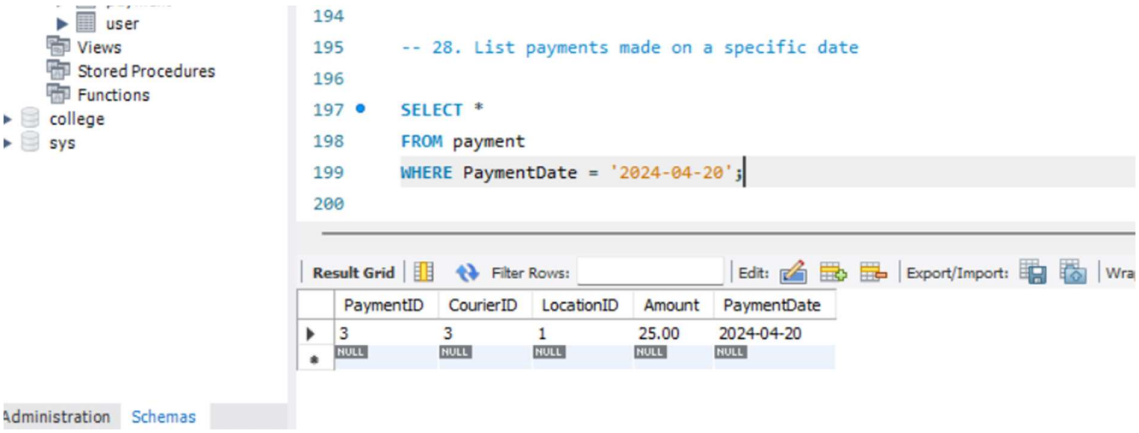
-- 27. Total payments received for each courier

```
SELECT c.courierID,SUM(p.Amount) AS TotalPayment
FROM courier c
JOIN payment p ON c.CourierID = p.CourierID
group by c.CourierID;
```



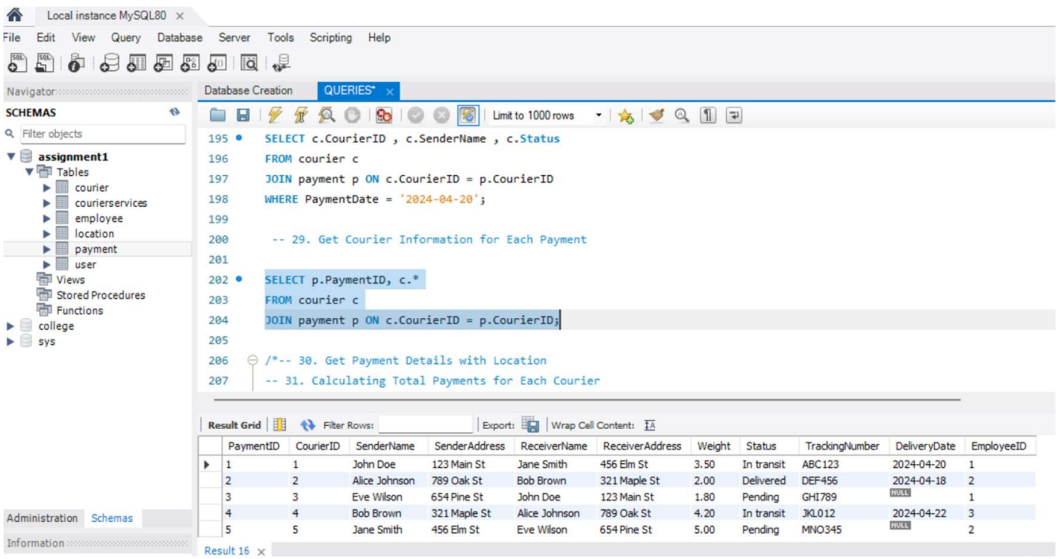
-- 28. List payments made on a specific date

```
SELECT *
FROM payment
WHERE PaymentDate = '2024-04-20';
```



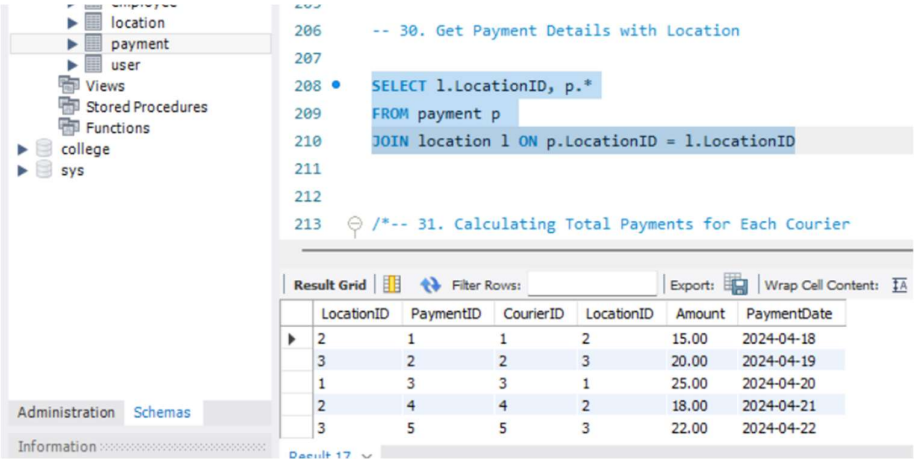
-- 29. Get Courier Information for Each Payment

```
SELECT p.PaymentID, c.*
FROM courier c
JOIN payment p ON c.CourierID = p.CourierID;
```



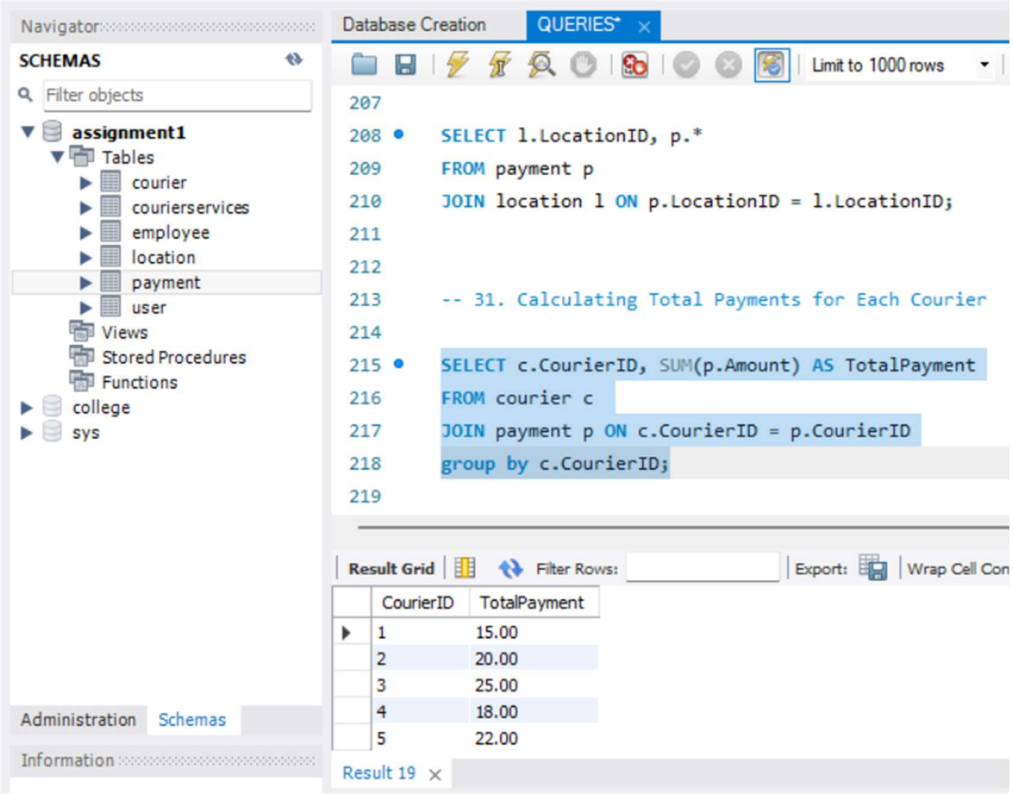
-- 30. Get Payment Details with Location

```
SELECT l.LocationID, p.*
FROM payment p
JOIN location l ON p.LocationID = l.LocationID;
```



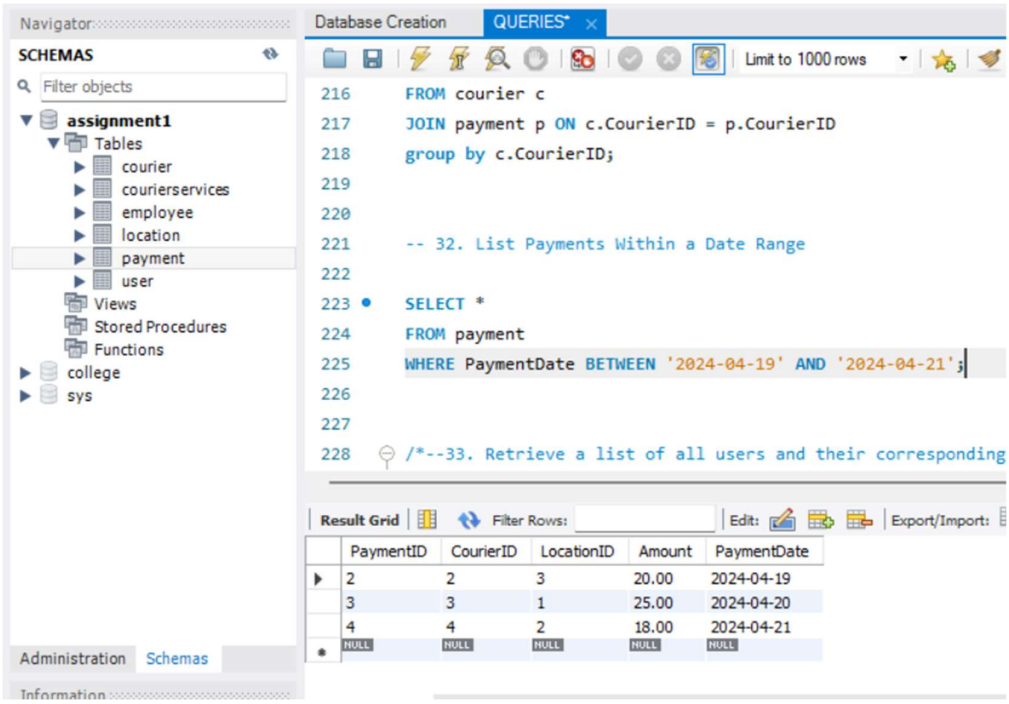
-- 31. Calculating Total Payments for Each Courier

```
SELECT c.CourierID, SUM(p.Amount) AS TotalPayment
FROM courier c
JOIN payment p ON c.CourierID = p.CourierID
group by c.CourierID;
```



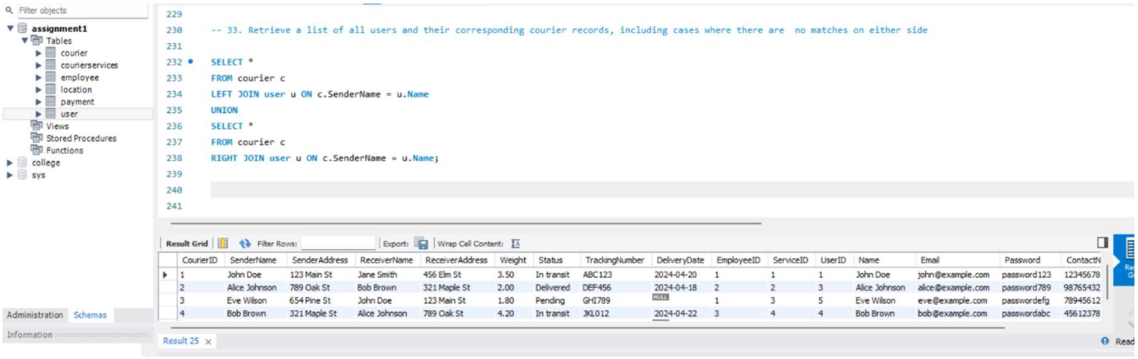
-- 32. List Payments Within a Date Range

```
SELECT *
FROM payment
WHERE PaymentDate BETWEEN '2024-04-19' AND '2024-04-21';
```



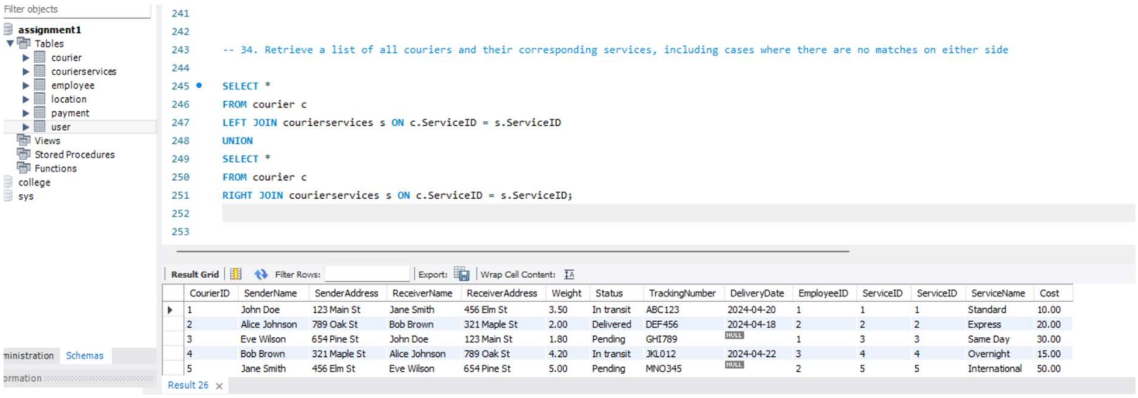
-- 33. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side

```
SELECT *
FROM courier c
LEFT JOIN user u ON c.SenderName = u.Name
UNION
SELECT *
FROM courier c
RIGHT JOIN user u ON c.SenderName = u.Name;
```



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

```
SELECT *
FROM courier c
LEFT JOIN courierservices s ON c.ServiceID = s.ServiceID
UNION
SELECT *
FROM courier c
RIGHT JOIN courierservices s ON c.ServiceID = s.ServiceID;
```



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

```
SELECT e.*, p.*
FROM employee e
LEFT JOIN courier c ON c.EmployeeID = e.EmployeeID
LEFT JOIN payment p ON c.CourierID = p.CourierID
UNION
SELECT e.*, p.*
FROM employee e
RIGHT JOIN courier c ON c.EmployeeID = e.EmployeeID
RIGHT JOIN payment p ON c.CourierID = p.CourierID;
```

```
256
257     -- 35. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side
258
259 •   SELECT e.*, p.*
260     FROM employee e
261     LEFT JOIN courier c ON c.EmployeeID = e.EmployeeID
262     LEFT JOIN payment p ON c.CourierID = p.CourierID
263     UNION
264     SELECT e.*, p.*
265     FROM employee e
266     RIGHT JOIN courier c ON c.EmployeeID = e.EmployeeID
267     RIGHT JOIN payment p ON c.CourierID = p.CourierID;
268
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EmployeeID	Name	Email	ContactNumber	Role	Salary	PaymentID	CourierID	LocationID	Amount	PaymentDate
1	Mark Johnson	mark@example.com	1239876540	Manager	50000.00	1	1	2	15.00	2024-04-18
1	Mark Johnson	mark@example.com	1239876540	Manager	50000.00	3	3	1	25.00	2024-04-20
2	Emily Brown	emily@example.com	9876543210	Clerk	30000.00	2	2	3	20.00	2024-04-19
2	Emily Brown	emily@example.com	9876543210	Clerk	30000.00	5	5	3	22.00	2024-04-22
3	Michael Smith	michael@example.com	4567890123	Driver	35000.00	4	4	2	18.00	2024-04-21

Result 27 ×

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

SELECT *

FROM courier c

CROSS JOIN courierservices s ON c.ServiceID = s.ServiceID; SELECT *

FROM courier c

CROSS JOIN courierservices s ON c.ServiceID = s.ServiceID;

```
272
273     --  36. List all users and all courier services, showing all possible combinations.
274
275 •       SELECT *
276         FROM courier c
277         CROSS JOIN courierservices s ON c.ServiceID = s.ServiceID;
278
279
280
281
282
283
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CourierID	SenderName	SenderAddress	ReceiverName	ReceiverAddress	Weight	Status	TrackingNumber	DeliveryDate	EmployeeID	ServiceID	ServiceID	ServiceName	Cost
1	John Doe	123 Main St	Jane Smith	456 Elm St	3.50	In transit	ABC123	2024-04-20	1	1	1	Standard	10.00
2	Alice Johnson	789 Oak St	Bob Brown	321 Maple St	2.00	Delivered	DEF456	2024-04-18	2	2	2	Express	20.00
3	Eve Wilson	654 Pine St	John Doe	123 Main St	1.80	Pending	GHI789	NULL	1	3	3	Same Day	30.00
4	Bob Brown	321 Maple St	Alice Johnson	789 Oak St	4.20	In transit	JKL012	2024-04-22	3	4	4	Overnight	15.00
5	Jane Smith	456 Elm St	Eve Wilson	654 Pine St	5.00	Pending	MNO345	NULL	2	5	5	International	50.00

Result 28 ×

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

```
SELECT e.*,l.*
FROM location l
CROSS JOIN payment p ON l.LocationID = p.LocationID
CROSS JOIN courier c ON p.CourierID = c.CourierID
CROSS JOIN employee e ON c.EmployeeID = e.EmployeeID;
```

```
284
285     -- 37. List all employees and all locations, showing all possible combinations:
286
287 •   SELECT e.*,l.*
288     FROM location l
289     CROSS JOIN payment p ON l.LocationID = p.LocationID
290     CROSS JOIN courier c ON p.CourierID = c.CourierID
291     CROSS JOIN employee e ON c.EmployeeID = e.EmployeeID;
292
293
294
295
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CourierID	UserID	Name	Email	Password	ContactNumber	Address	Status	TrackingNumber	DeliveryDate
1	1	John Doe	john@example.com	password123	1234567890	123 Main St	In transit	ABC123	2024-04-20
2	3	Alice Johnson	alice@example.com	password789	9876543210	789 Oak St	Delivered	DEF456	2024-04-18
3	5	Eve Wilson	eve@example.com	passwordefg	7894561230	654 Pine St	Pending	GHI789	NULL
4	4	Bob Brown	bob@example.com	passwordabc	4561237890	321 Maple St	In transit	JKL012	2024-04-22
5	2	Jane Smith	jane@example.com	password456	0987654321	456 Elm St	Pending	MNO345	NULL

Result 30

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

```
SELECT c.CourierID, u.*, c.Status, c.TrackingNumber,c.DeliveryDate
FROM courier c
LEFT JOIN user u ON c.SenderName = u.Name;
```

```
297
298     -- 38. Retrieve a list of couriers and their corresponding sender information (if available)
299
300 •   SELECT c.CourierID, u.*, c.Status, c.TrackingNumber,c.DeliveryDate
301     FROM courier c
302     LEFT JOIN user u ON c.SenderName = u.Name;
303
304
305
306
307
```

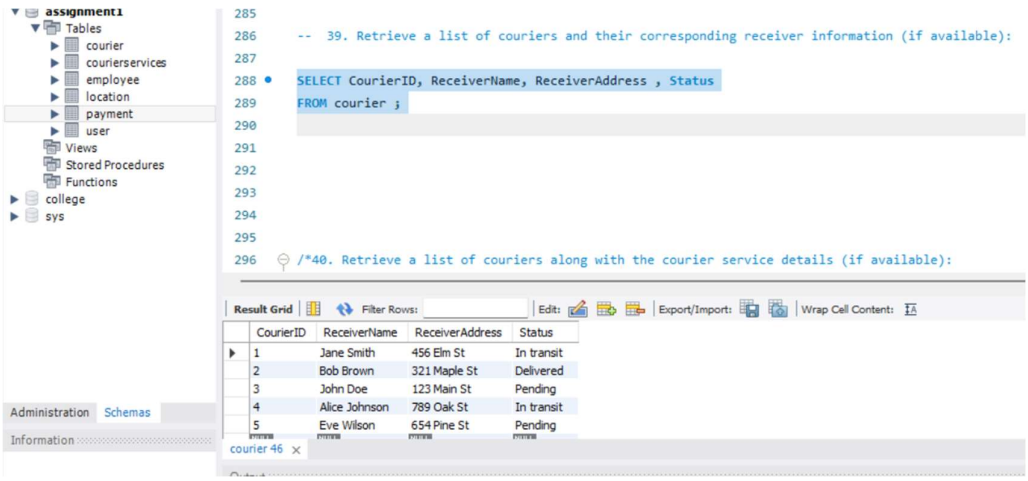
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CourierID	UserID	Name	Email	Password	ContactNumber	Address	Status	TrackingNumber	DeliveryDate
1	1	John Doe	john@example.com	password123	1234567890	123 Main St	In transit	ABC123	2024-04-20
2	3	Alice Johnson	alice@example.com	password789	9876543210	789 Oak St	Delivered	DEF456	2024-04-18
3	5	Eve Wilson	eve@example.com	passwordefg	7894561230	654 Pine St	Pending	GHI789	NULL
4	4	Bob Brown	bob@example.com	passwordabc	4561237890	321 Maple St	In transit	JKL012	2024-04-22
5	2	Jane Smith	jane@example.com	password456	0987654321	456 Elm St	Pending	MNO345	NULL

Result 30

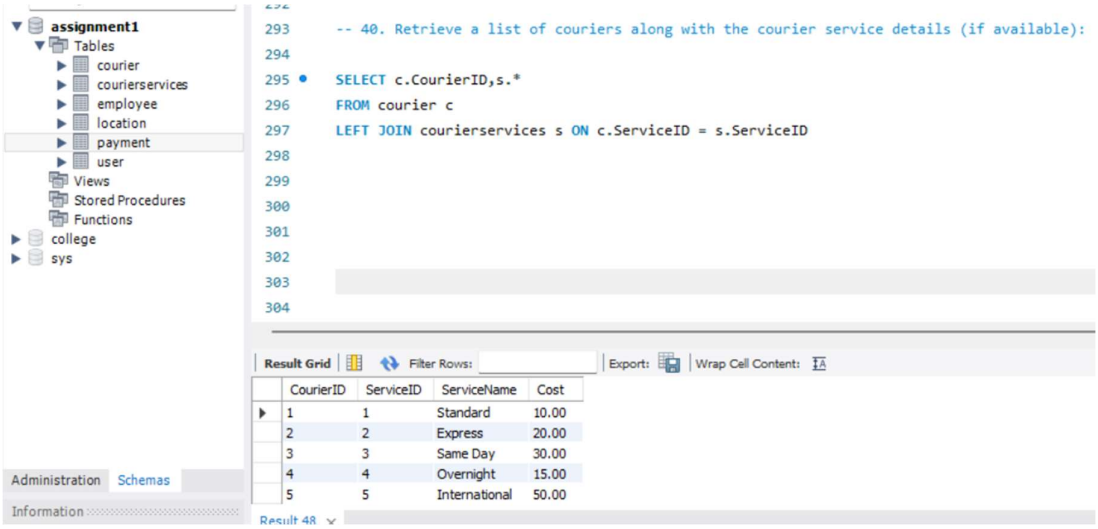
-- 39. Retrieve a list of couriers and their corresponding receiver information (if available):

```
SELECT CourierID, ReceiverName, ReceiverAddress , Status
FROM courier ;
```



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

```
SELECT c.CourierID,s.*
FROM courier c
LEFT JOIN courierservices s ON c.ServiceID = s.ServiceID;
```



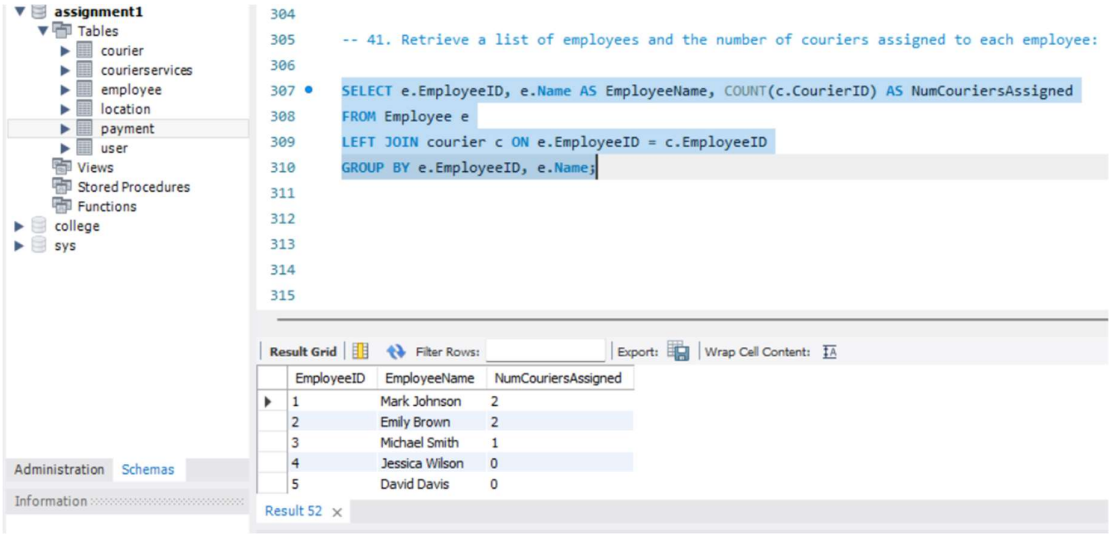
-- 41. Retrieve a list of employees and the number of couriers assigned to each employee:

SELECT e.EmployeeID, e.Name AS EmployeeName, COUNT(c.CourierID) AS NumCouriersAssigned

FROM Employee e

LEFT JOIN courier c ON e.EmployeeID = c.EmployeeID

GROUP BY e.EmployeeID, e.Name;



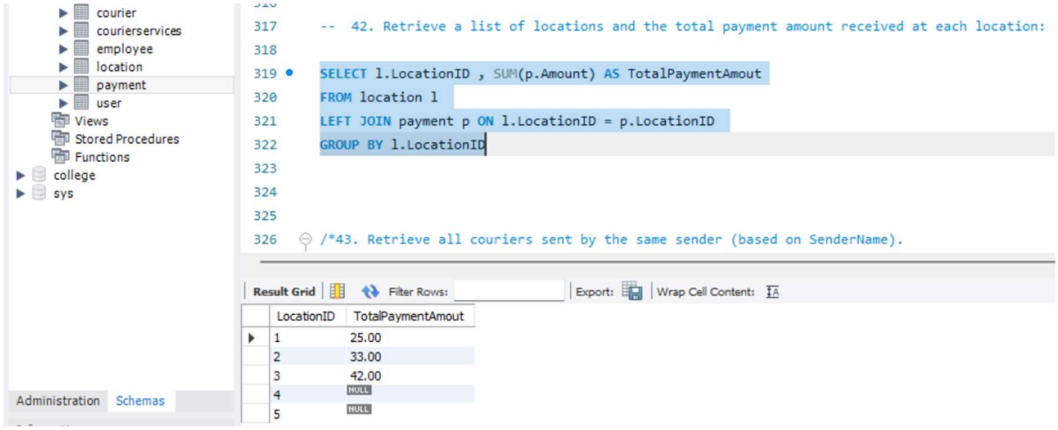
-- 42. Retrieve a list of locations and the total payment amount received at each location:

SELECT l.LocationID , SUM(p.Amount) AS TotalPaymentAmout

FROM location l

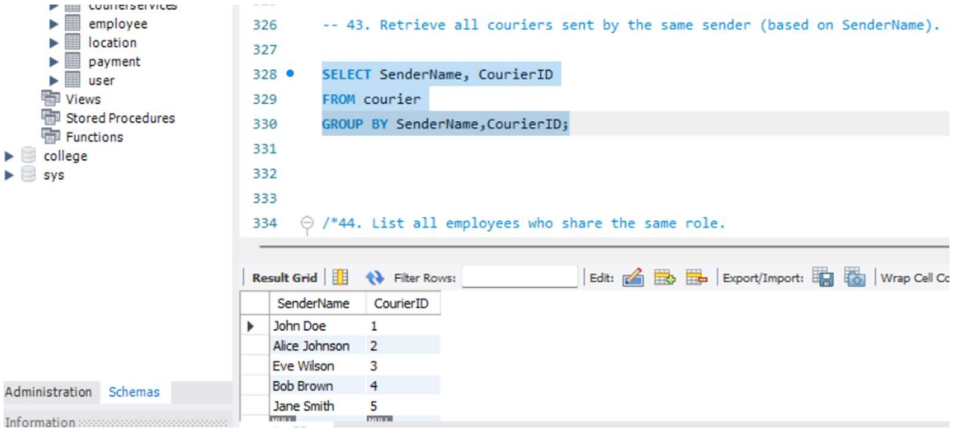
LEFT JOIN payment p ON l.LocationID = p.LocationID

GROUP BY l.LocationID;



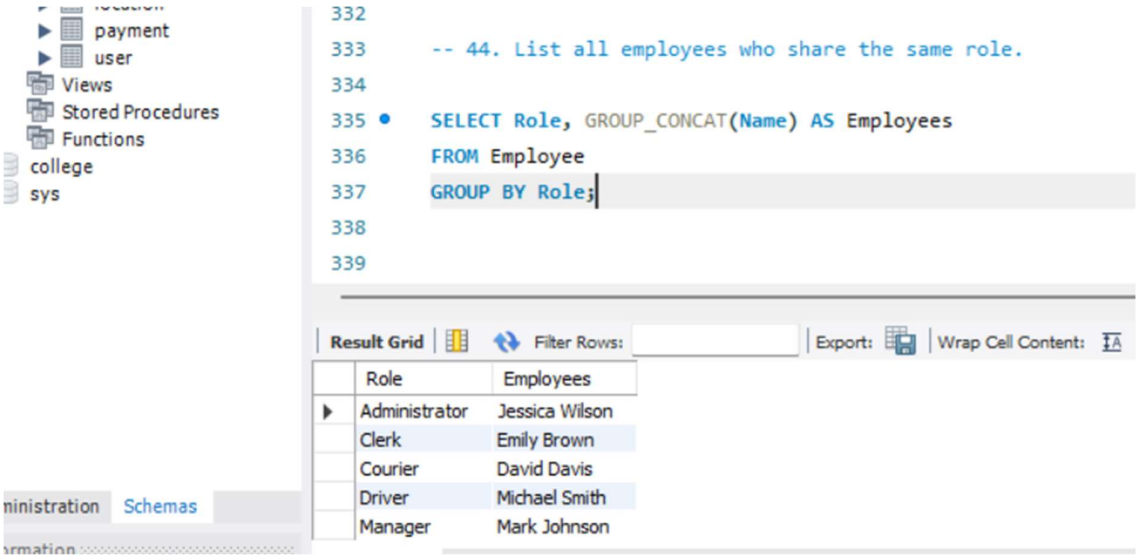
-- 43. Retrieve all couriers sent by the same sender (based on SenderName).

```
SELECT SenderName, CourierID
FROM courier
GROUP BY SenderName,CourierID;
```



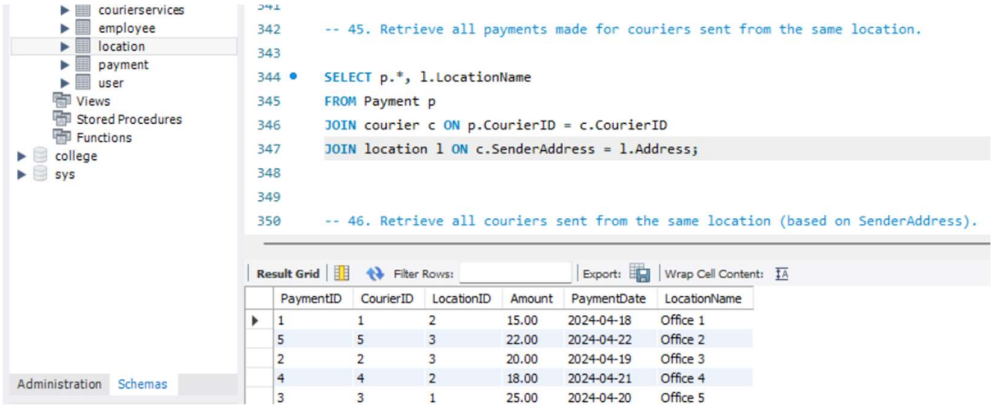
-- 44. List all employees who share the same role.

```
SELECT Role, GROUP_CONCAT(Name) AS Employees
FROM Employee
GROUP BY Role;
```



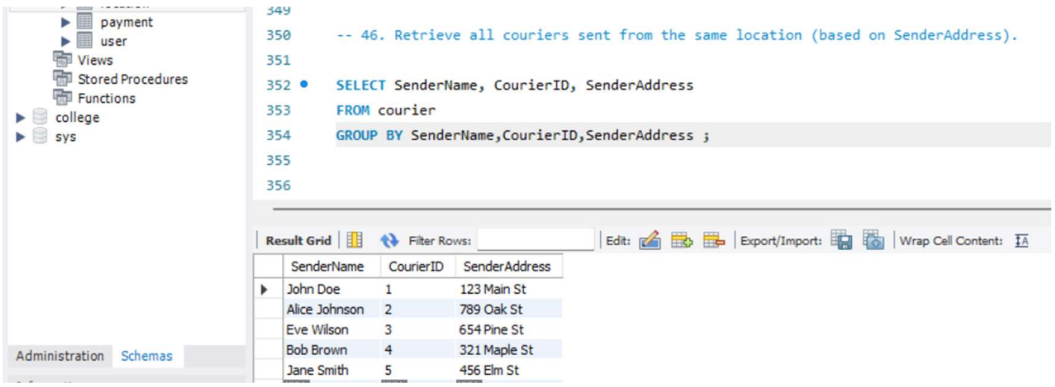
-- 45. Retrieve all payments made for couriers sent from the same location.

```
SELECT p.*, l.LocationName
FROM Payment p
JOIN courier c ON p.CourierID = c.CourierID
JOIN location l ON c.SenderAddress = l.Address;
```



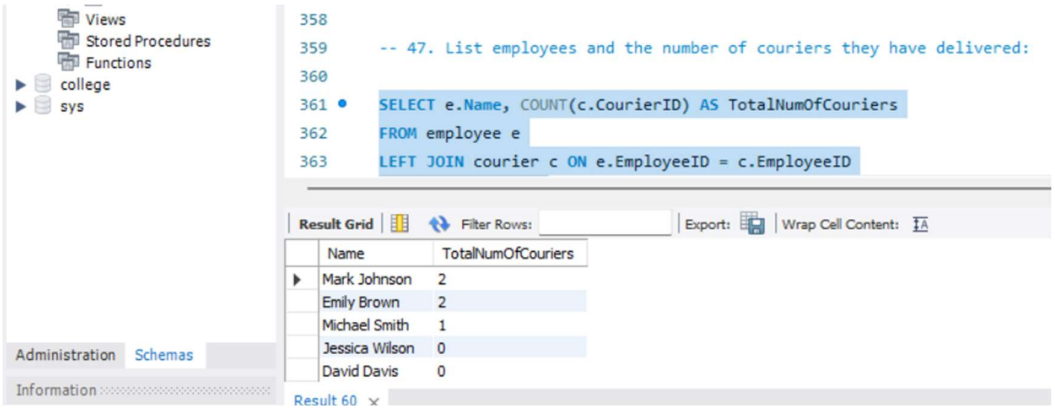
-- 46. Retrieve all couriers sent from the same location (based on SenderAddress).

```
SELECT SenderName, CourierID, SenderAddress
FROM courier
GROUP BY SenderName,CourierID,SenderAddress ;
```



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

```
SELECT e.Name, COUNT(c.CourierID) AS TotalNumOfCouriers
FROM employee e
LEFT JOIN courier c ON e.EmployeeID = c.EmployeeID
 GROUP BY e.Name ;
```



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

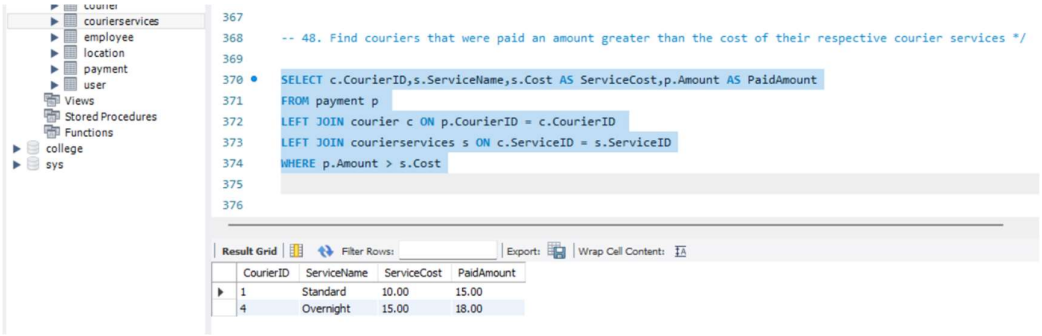
SELECT c.CourierID,s.ServiceName,s.Cost AS ServiceCost,p.Amount AS PaidAmount

FROM payment p

LEFT JOIN courier c ON p.CourierID = c.CourierID

LEFT JOIN courierservices s ON c.ServiceID = s.ServiceID

WHERE p.Amount > s.Cost;

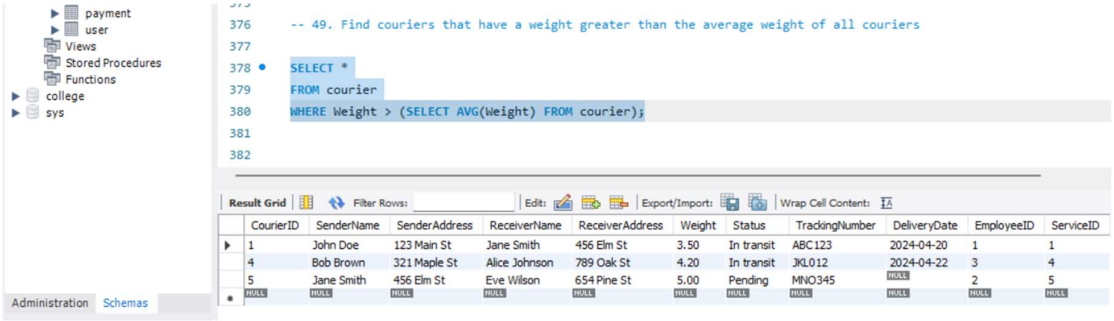


-- 49. Find couriers that have a weight greater than the average weight of all couriers

SELECT *

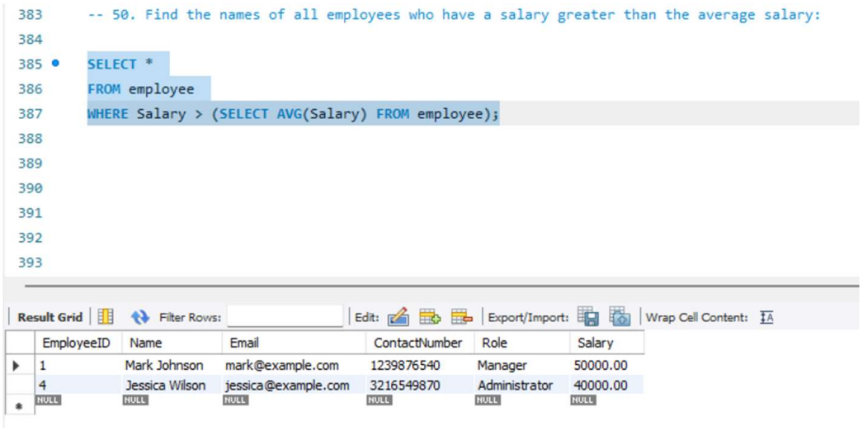
FROM courier

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



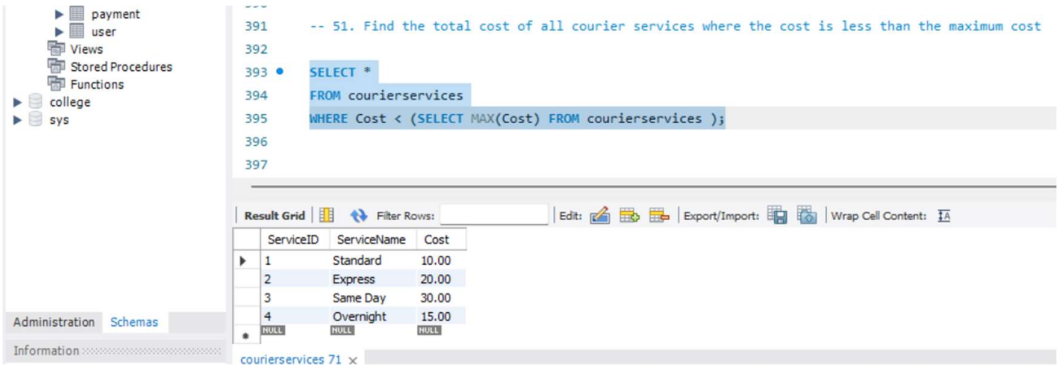
-- 50. Find the names of all employees who have a salary greater than the average salary:

```
SELECT *
FROM employee
WHERE Salary > (SELECT AVG(Salary) FROM employee);
```



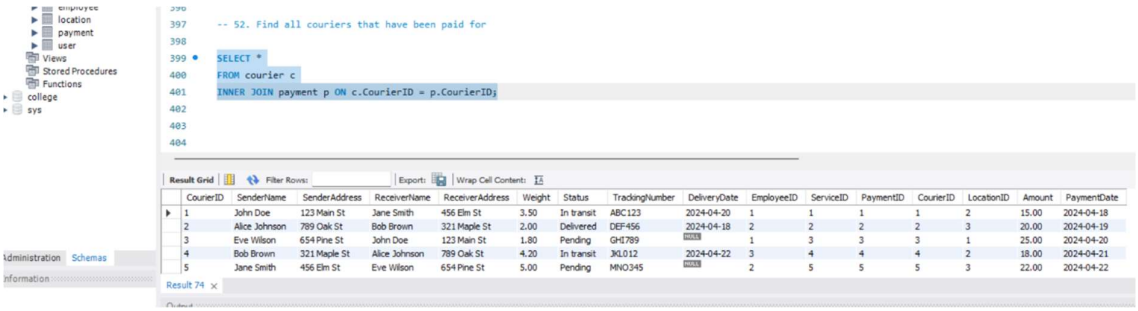
-- 51. Find the total cost of all courier services where the cost is less than the maximum cost

```
SELECT *
FROM courierservices
WHERE Cost < (SELECT MAX(Cost) FROM courierservices );
```



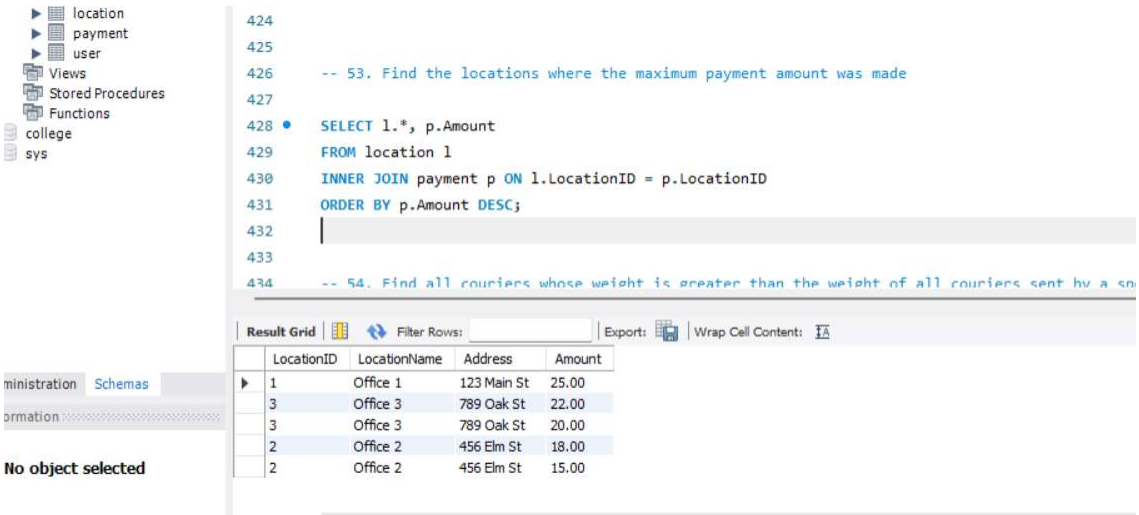
-- 52. Find all couriers that have been paid for

```
SELECT *
FROM courier c
INNER JOIN payment p ON c.CourierID = p.CourierID;
```



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

```
SELECT l.*, p.Amount
FROM location l
INNER JOIN payment p ON l.LocationID = p.LocationID
ORDER BY p.Amount DESC;
```



-- 54. 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 > (SELECT SUM(Weight)

FROM courier

WHERE SenderName = 'Eve Wilson')