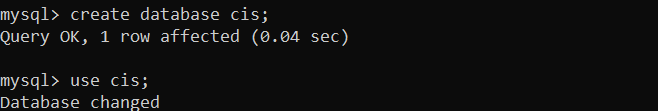
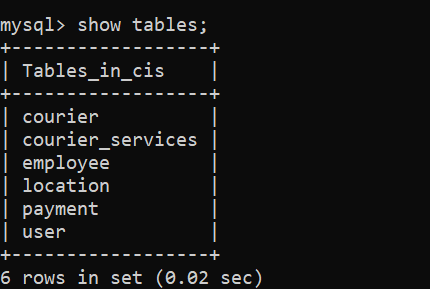
**ASSISGNMENT 4**

**Courier Management System**

**TASK 1**

1.DATABASE DESIGN





CREATE TABLE user(

userid INT PRIMARY KEY,

name VARCHAR(255),

email VARCHAR(255) UNIQUE,

password VARCHAR(255),

contact VARCHAR(20),

address TEXT);

INSERT INTO USER VALUES

(1,'Gowtham','gowtham@gmail.com','gg123','1234567890','ABC street'),

(2,'John','john@gmail.com','john','1234567980','BBB street'),

(3,'Nithya','nithya@gmail.com','nithya','1234568790','CCC street'),

(4,'Rohan','rohan@gmail.com','rohan','1234567890','DAC street'),

(5,'Vikram','vikram@gmail.com','vikram','1234567890','ECR street'),

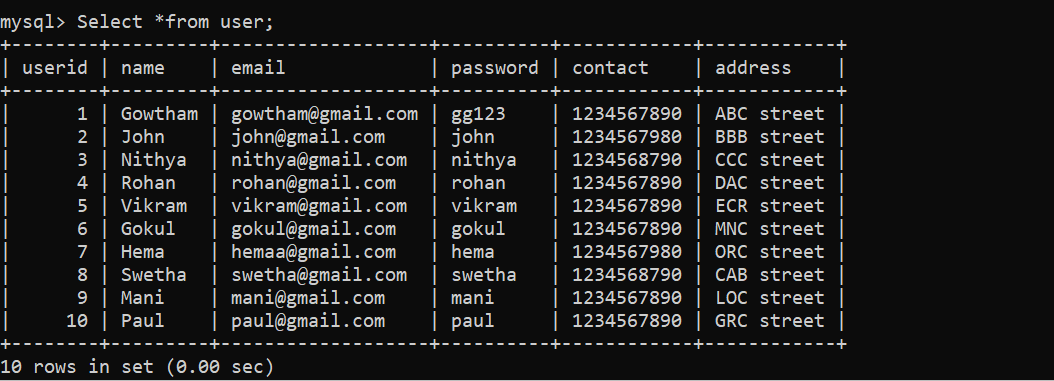
(6,'Gokul','gokul@gmail.com','gokul','1234567890','MNC street'),

(7,'Hema','hemaa@gmail.com','hema','1234567980','ORC street'),

(8,'Swetha','swetha@gmail.com','swetha','1234568790','CAB street'),

(9,'Mani','mani@gmail.com','mani','1234567890','LOC street'),

(10,'Paul','paul@gmail.com','paul','1234567890','GRC street');



CREATE TABLE courier(

Courier\_id INT PRIMARY KEY,

Sender\_name VARCHAR(255),

sender\_address TEXT,

receiver\_name VARCHAR(255),

receiver\_address TEXT,

weight DECIMAL(5,2),

status VARCHAR(50),

tracking\_no VARCHAR(20) UNIQUE,

delivery\_date DATE);

INSERT INTO COURIER VALUES

(101,'rohan','dac street','Gokul','mnc street',565,'undelivered','1234','2023-12-25'),

(102,'John','bbb street','nithya','ccc street',500,'delivered','4321','2023-12-01'),

(103,'paul','grc street','John','bbb street',200,'cancelled','5555','2023-11-19'),

(104,'Gokul','mnc street','swetha','cab street',750,'undelivered','5424','2023-12-25'),

('105', 'vikram', 'ecr street', 'mani', 'loc street', '900.00', 'delivered', '4657', '2023-10-05'),

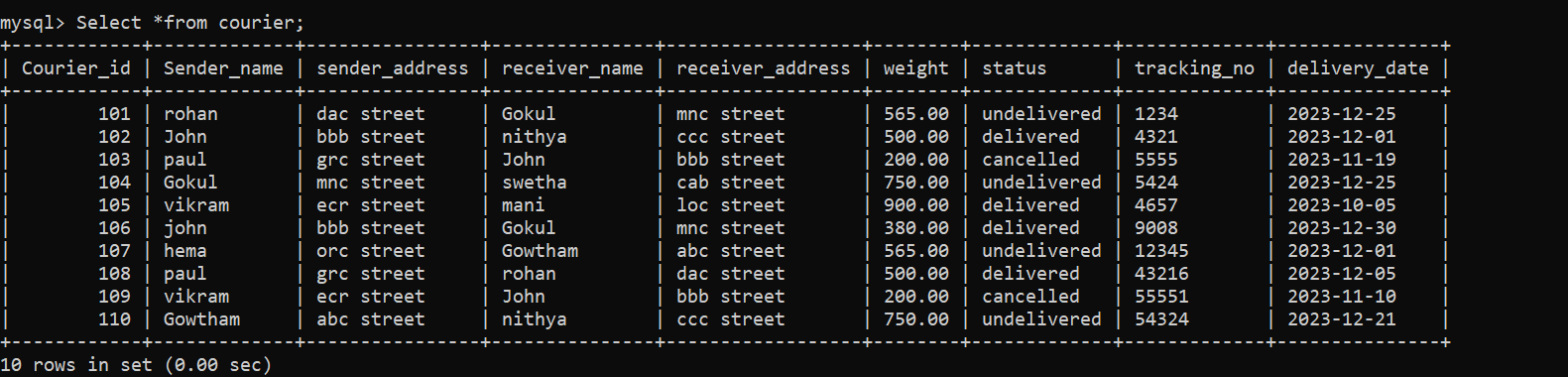
('106', 'john', 'bbb street', 'Gokul', 'mnc street', '380.00', 'delivered', '9008', '2023-12-30'),

(107,'hema','orc street','Gowtham','abc street',565,'undelivered','12345','2023-12-01'),

(108,'paul','grc street','rohan','dac street',500,'delivered','43216','2023-12-05'),

(109,'vikram','ecr street','John','bbb street',200,'cancelled','55551','2023-11-10'),

(110,'Gowtham','abc street','nithya','ccc street',750,'undelivered','54324','2023-12-21');



CREATE TABLE Courier\_services(

Service\_id INT PRIMARY KEY,

Service\_name VARCHAR(100),

cost DECIMAL(8,2));

INSERT INTO COURIER\_SERVICES VALUES

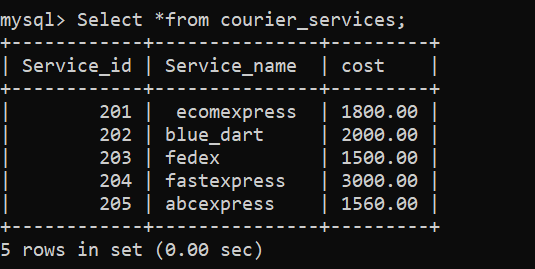
(201,' ecomexpress ',1800),

(202,'blue\_dart',2000),

(203,'fedex',1500),

(204, 'fastexpress ',3000),

(205, 'abcexpress',1560);



CREATE TABLE employee(

Emp\_id INT PRIMARY KEY,

Name VARCHAR(255),

Email VARCHAR(255) UNIQUE,

Contact VARCHAR(20),

Role VARCHAR(50),

Salary DECIMAL(10,2));

INSERT INTO EMPLOYEE VALUES

(301,'Doel','doel@gmail','1234557890','delivery',30000),

(302,'Michel','michel@gmail','1243557890','delivery',25000),

(303,'Tharun','tharun@gmail','1234557790','supervisor',40000),

(304,'Hari','hari@gmail','9234557890','delivery',25000),

(305,'Sathya','sathya@gmail','1234555890','Manager',95000),

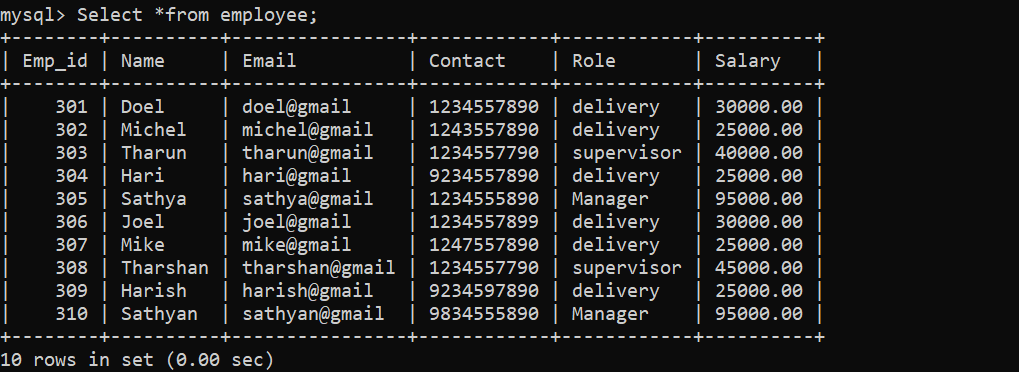
(306,'Joel','joel@gmail','1234557899','delivery',30000),

(307,'Mike','mike@gmail','1247557890','delivery',25000),

(308,'Tharshan','tharshan@gmail','1234557790','supervisor',45000),

(309,'Harish','harish@gmail','9234597890','delivery',25000),

(310,'Sathyan','sathyan@gmail','9834555890','Manager',95000);



CREATE TABLE location(

Location\_id INT PRIMARY KEY,

Location\_name VARCHAR(100),

address TEXT);

INSERT INTO LOCATION VALUES

(401,'A place','abc street'),

(402,'B place','bbb street'),

(403,'C place','ccc street'),

(404,'D place','dac street'),

(405,'E place','ecr street'),

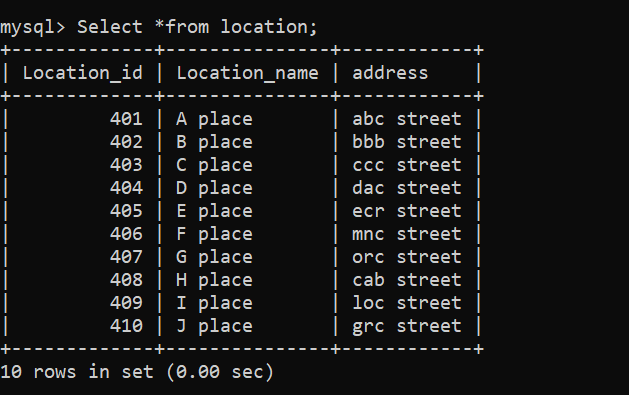
(406,'F place','mnc street'),

(407,'G place','orc street'),

(408,'H place','cab street'),

(409,'I place','loc street'),

(410,'J place','grc street');



CREATE TABLE payment(

Payment\_id INT PRIMARY KEY,

Courier\_id INT,

Location\_id INT,

amount DECIMAL(10,2),

payment\_date DATE,

FOREIGN KEY(courier\_id) REFERENCES courier(courier\_id),

FOREIGN KEY(location\_id) REFERENCES location(location\_id));

INSERT INTO PAYMENT VALUES

(501,101,401,5650,'2023-12-28'),

(502,102,404,5000,'2023-10-01'),

(503,103,402,2000,'2023-10-19'),

(504,104,403,7500,'2023-12-15'),

(505, 106, 405, 3800,'2023-12-03'),

(506,105,406,5750,'2023-12-01'),

(507,107,407,5500,'2023-12-11'),

(508,108,408,2100,'2023-11-29'),

(509,109,409,6500,'2023-12-05'),

(510, 110, 410, 3900,'2023-12-03');

INSERT INTO payment (`paymentid`, `courierid`, `locationid`, `amount`, `paymentdate`) VALUES ('505', '106', '401', '3800.00', '2023-12-30');

UPDATE payment SET `courierid` = '104', `locationid` = '403', `amount` = '7500.00', `paymentdate` = '2023-12-25' WHERE (`paymentid` = '503');

UPDATE payment SET `courierid` = '105', `locationid` = '405', `amount` = '9000.00', `paymentdate` = '2023-12-05' WHERE (`paymentid` = '504');

ALTER TABLE PAYMENT ADD COLUMN payment\_time time;

UPDATE `payment` SET `payment\_time` = '09:00:00' WHERE (`payment\_id` = '501');

UPDATE `payment` SET `payment\_time` = '10:00:00' WHERE (`payment\_id` = '502');

UPDATE `payment` SET `payment\_time` = '09:30:00' WHERE (`payment\_id` = '503');

UPDATE `payment` SET `payment\_time` = '11:00:00' WHERE (`payment\_id` = '504');

UPDATE `payment` SET `payment\_time` = '09:00:00' WHERE (`payment\_id` = '505'),

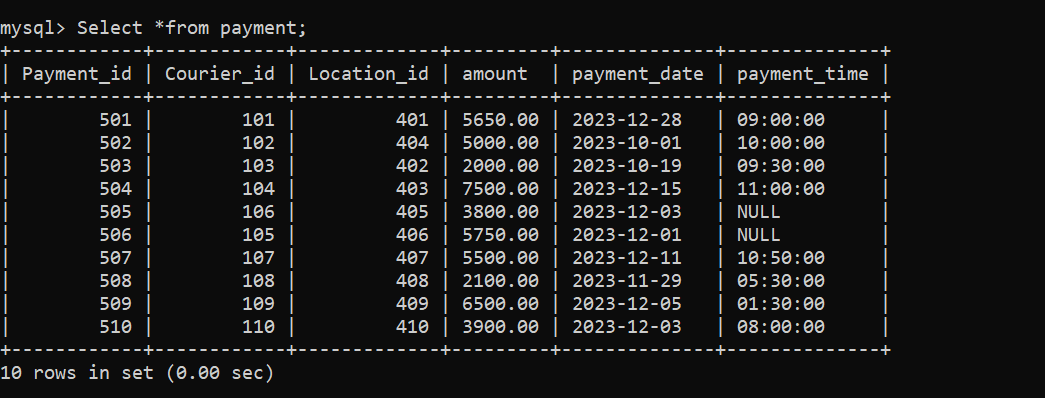
UPDATE `payment` SET `payment\_time` = '03:00:00' WHERE (`payment\_id` = '506');

UPDATE `payment` SET `payment\_time` = '10:50:00' WHERE (`payment\_id` = '507');

UPDATE `payment` SET `payment\_time` = '05:30:00' WHERE (`payment\_id` = '508');

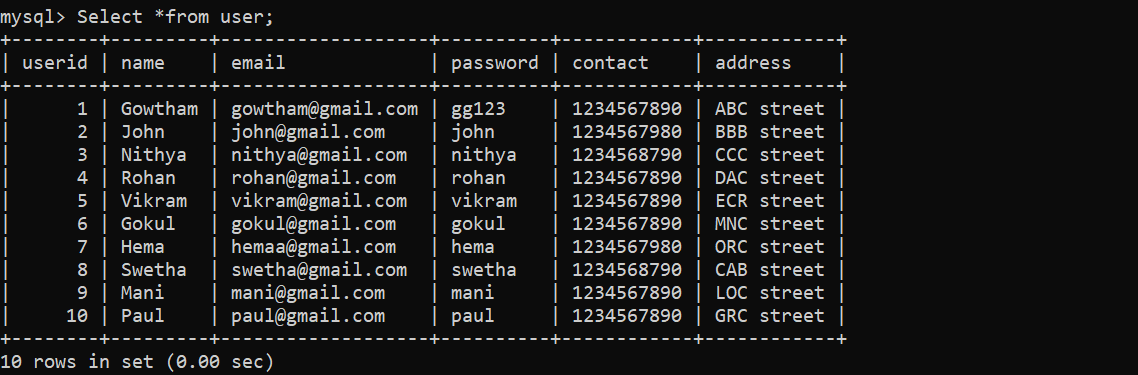
UPDATE `payment` SET `payment\_time` = '01:30:00' WHERE (`payment\_id` = '509');

UPDATE `payment` SET `payment\_time` = '08:00:00' WHERE (`payment\_id` = '510');

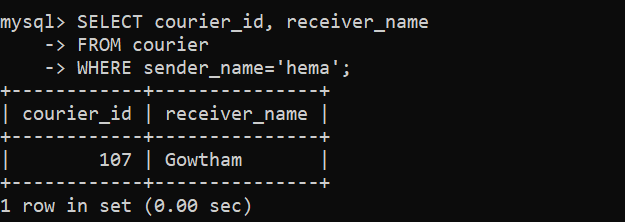


TASK 2

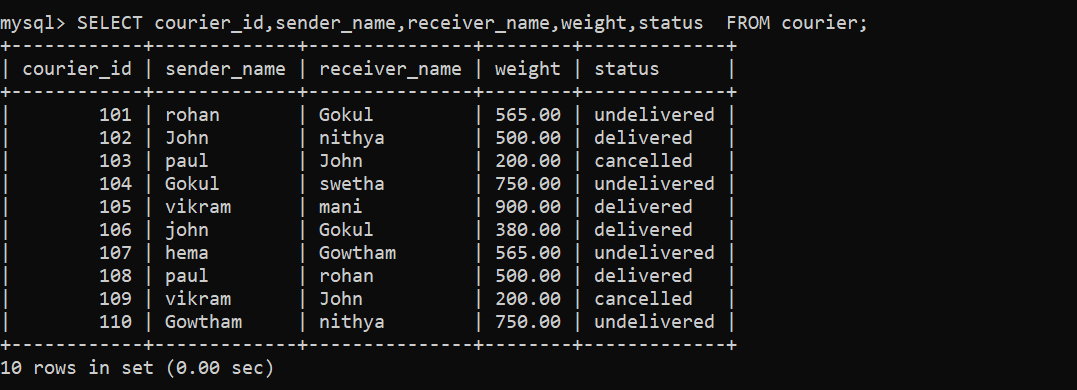
1. List all customers



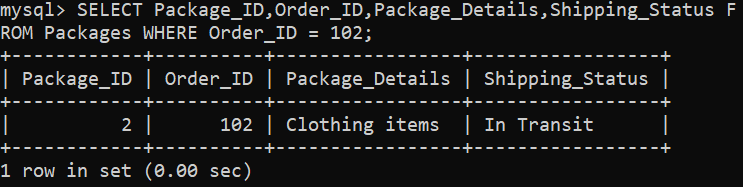
2. List all orders for a specific customer



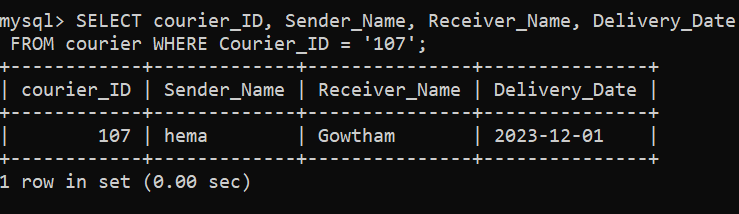
3. List all couriers



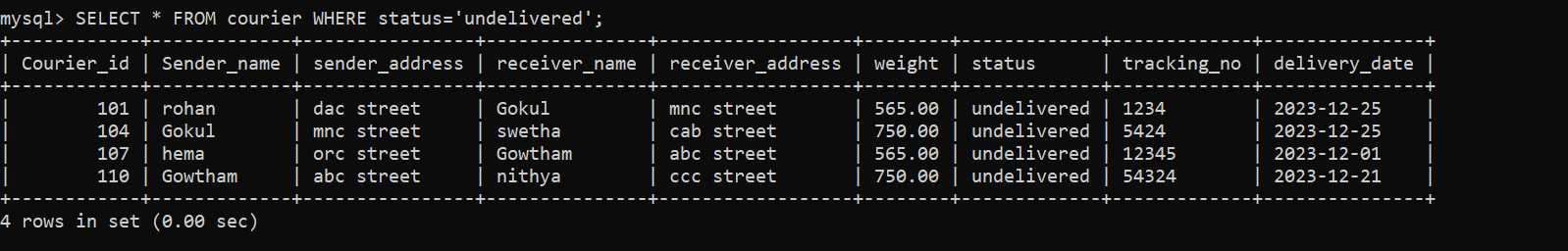
4. List all packages for a specific order



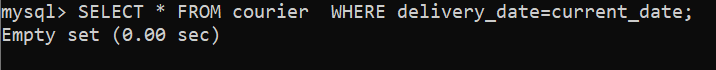
5. List all deliveries for a specific courier



6. List all undelivered packages



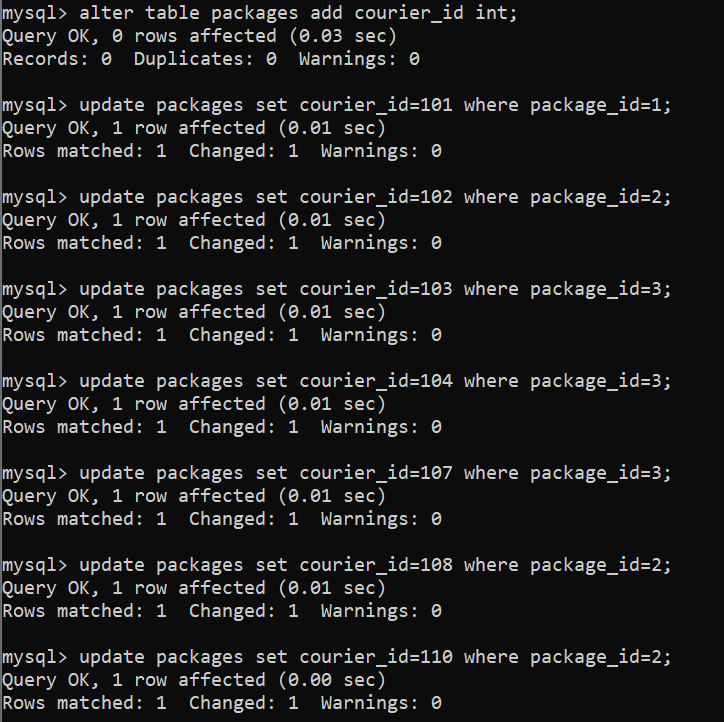
7. List all packages that are scheduled for delivery today

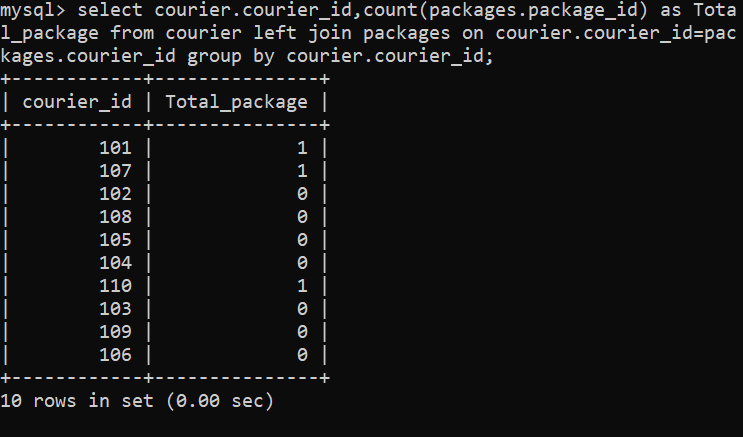


8. List all packages with a specific status

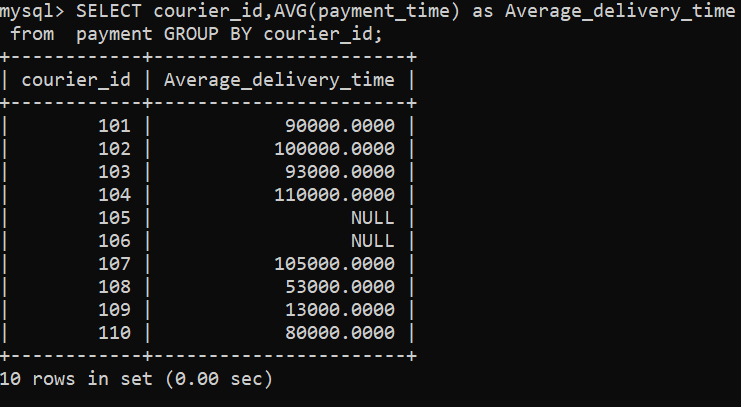


9. Calculate the total number of packages for each courier.

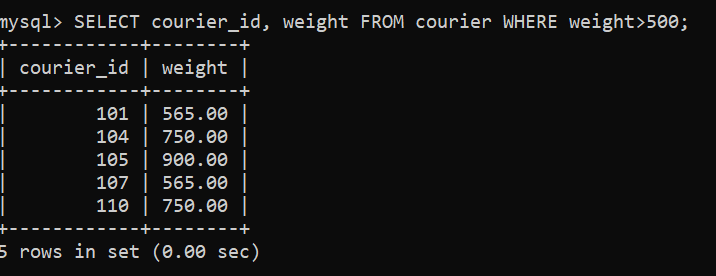




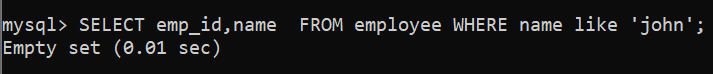
10. Find the average delivery time for each courier



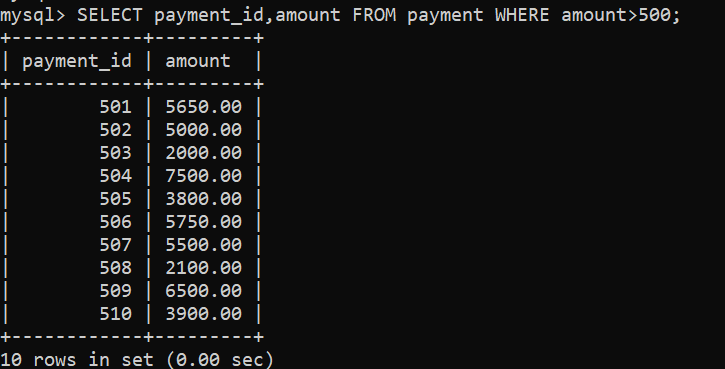
11. List all packages with a specific weight range



12. Retrieve employees whose names contain 'John'

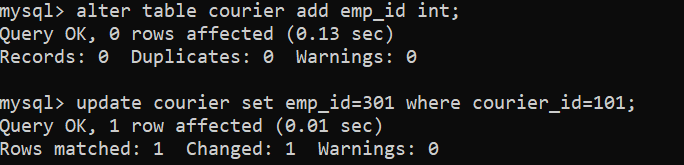


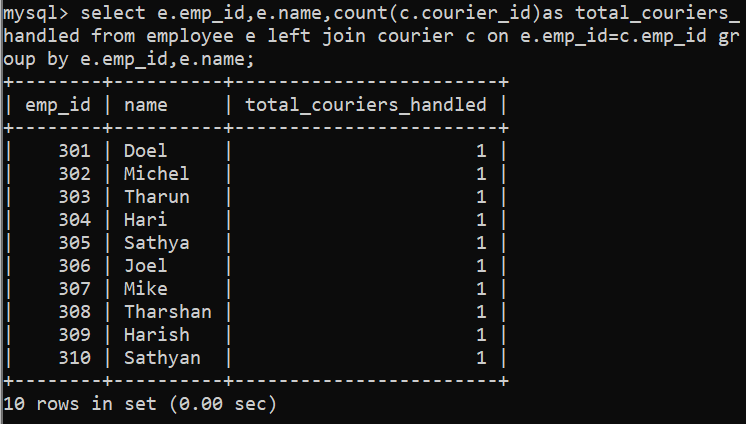
13. Retrieve all courier records with payments greater than $50.

****

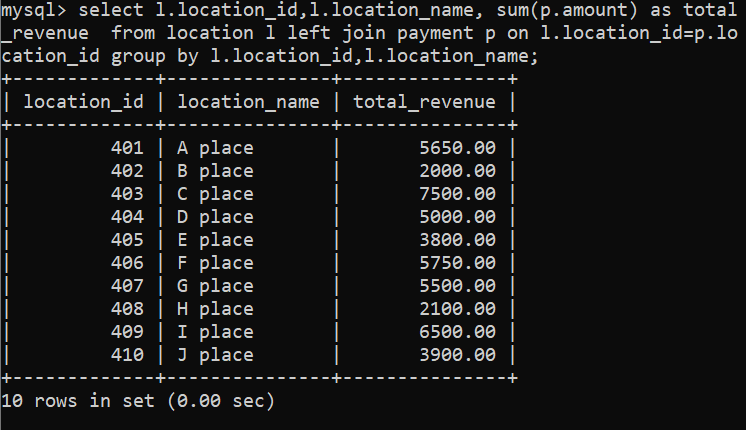
**Task 3**

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

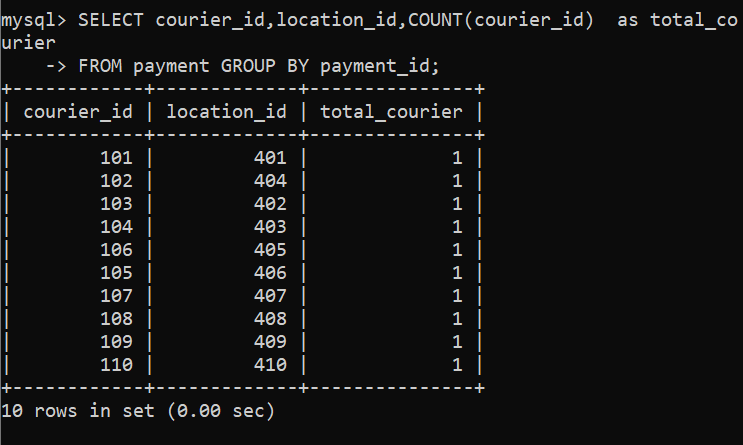




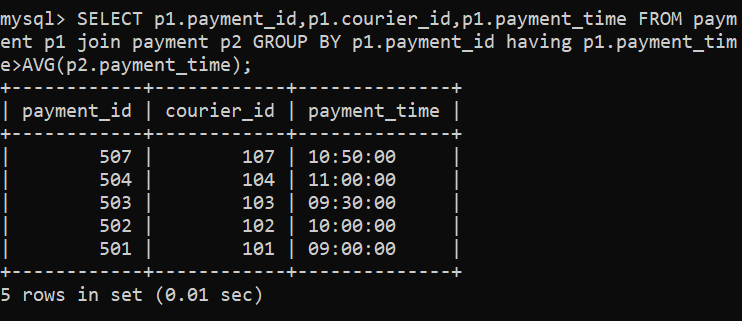
15. Calculate the total revenue generated by each location



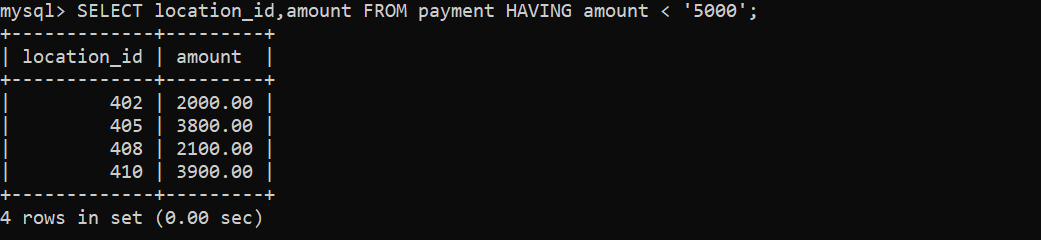
16. Find the total number of couriers delivered to each location.



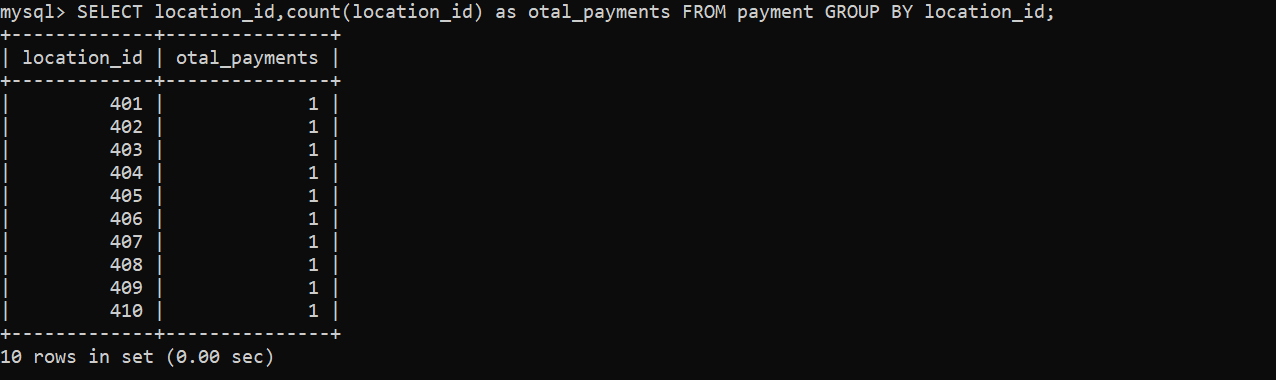
17. Find the courier with the highest average delivery time



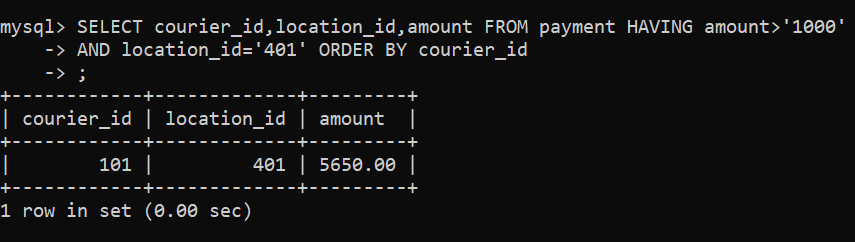
18. Find Locations with Total Payments Less Than a Certain Amount



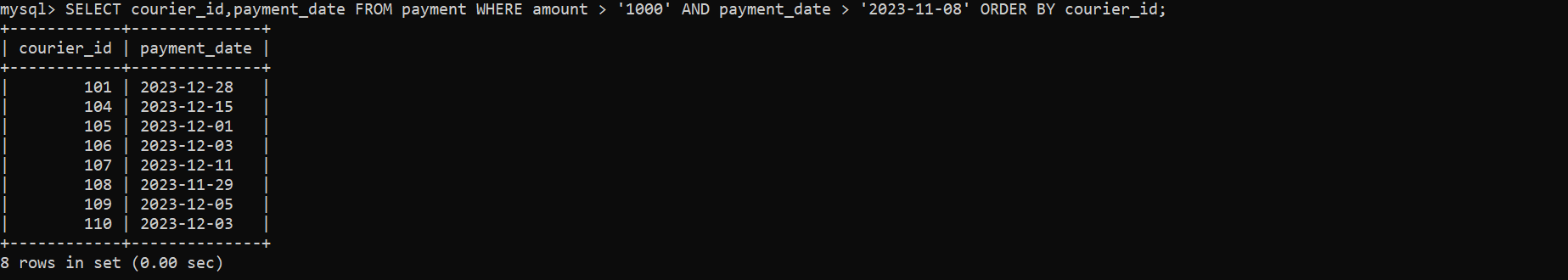
19. Calculate Total Payments per Location



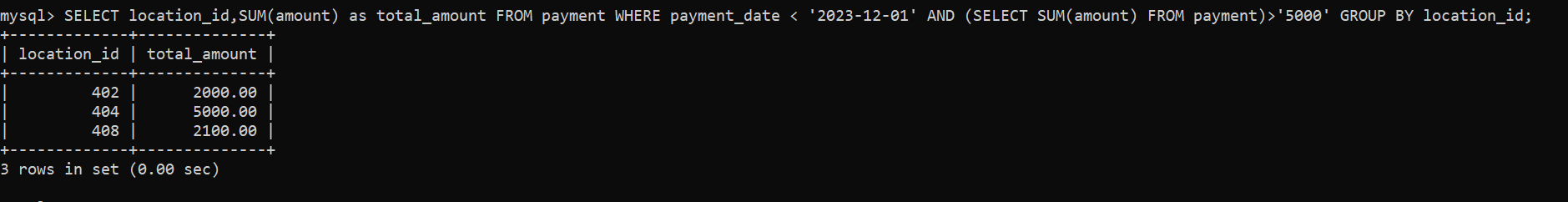
20. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = X)



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

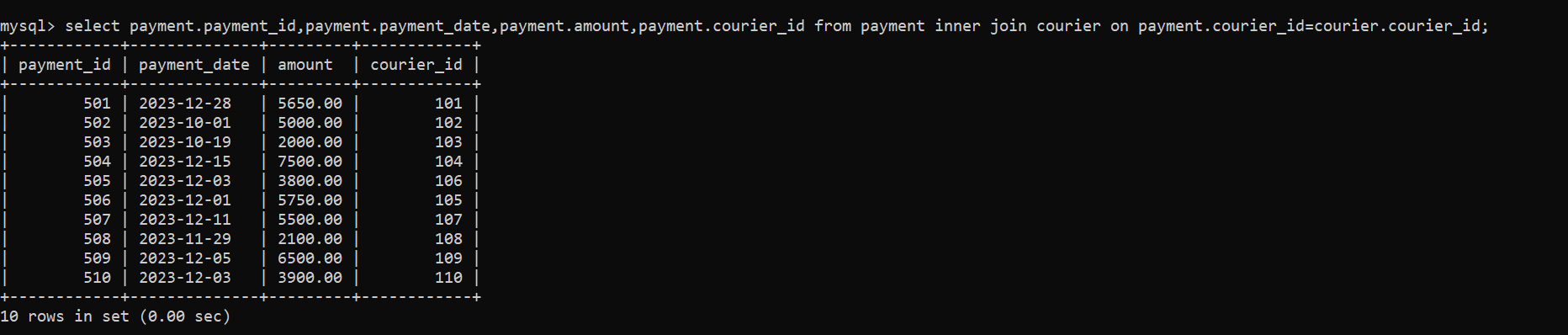


22. Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

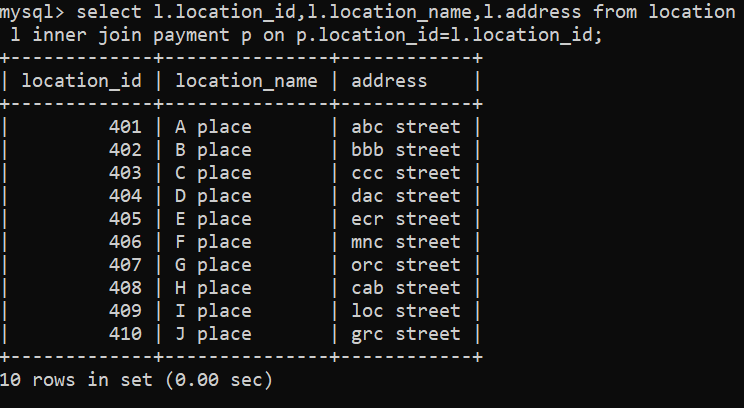


Task 4

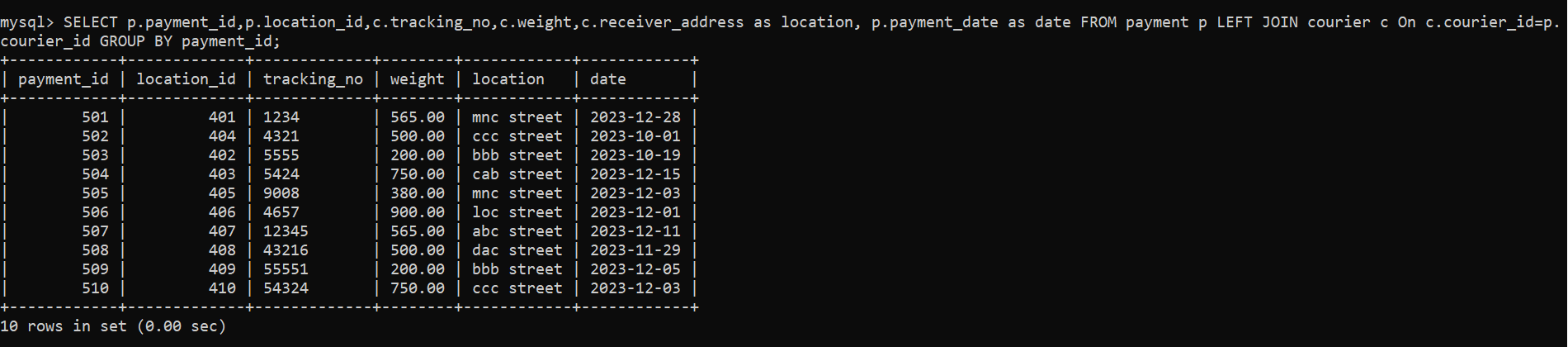
23. Retrieve Payments with Courier Information



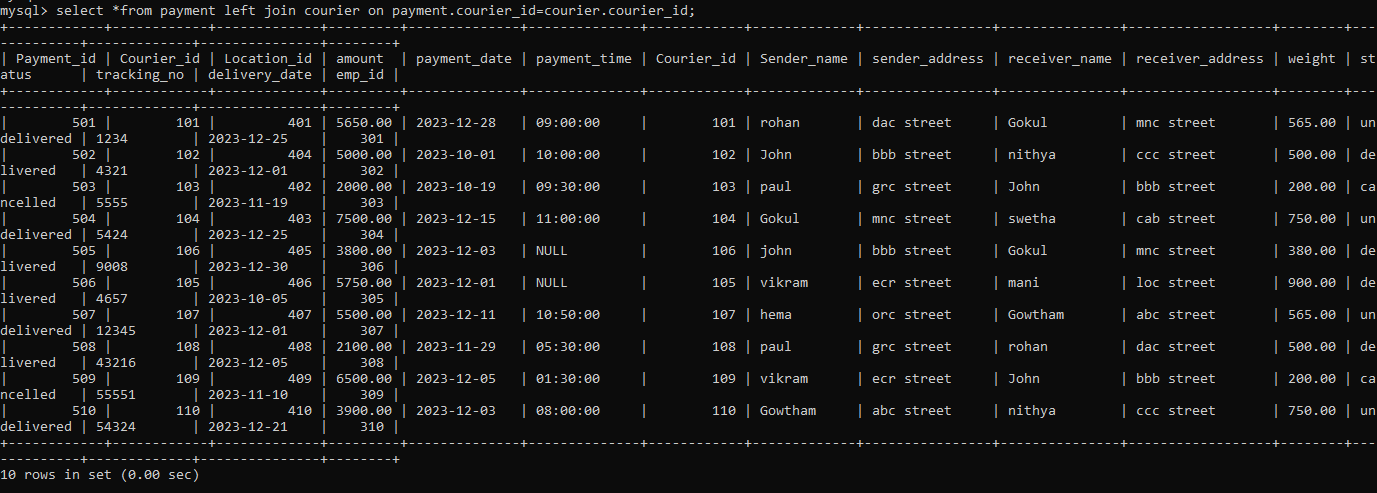
24. Retrieve Payments with Location Information



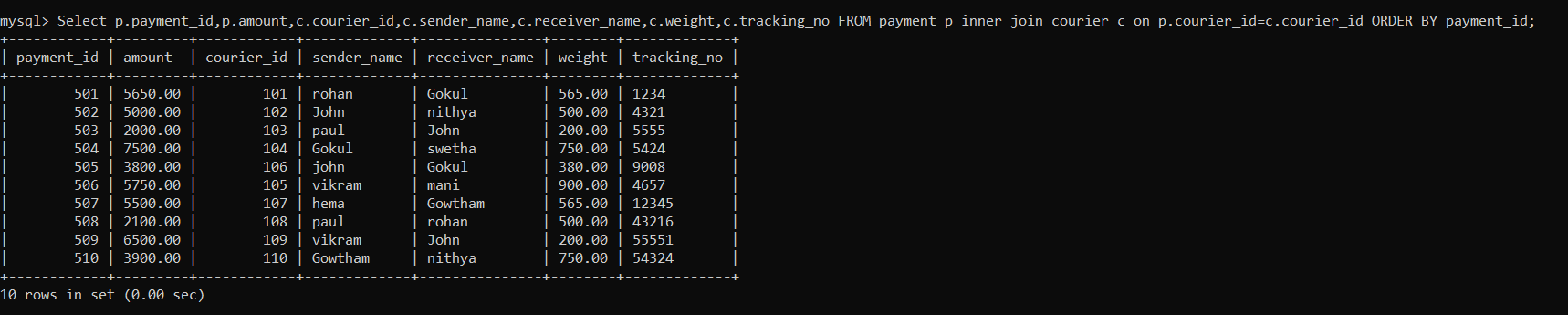
25. Retrieve Payments with Courier and Location Information



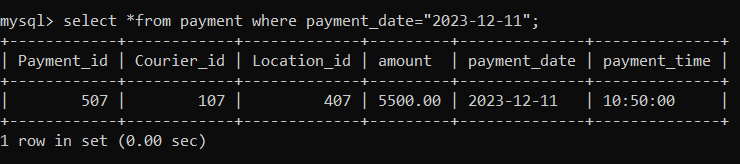
26. List all payments with courier details



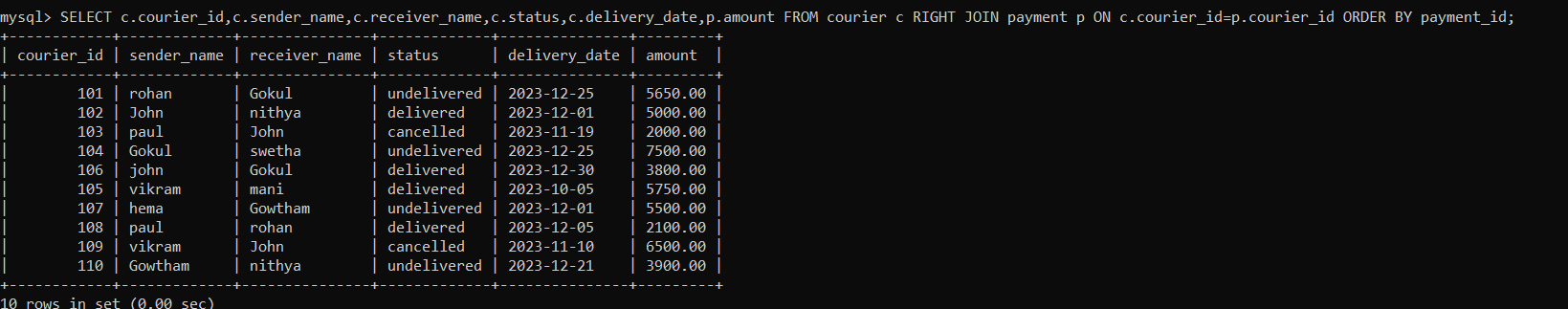
27. Total payments received for each courier



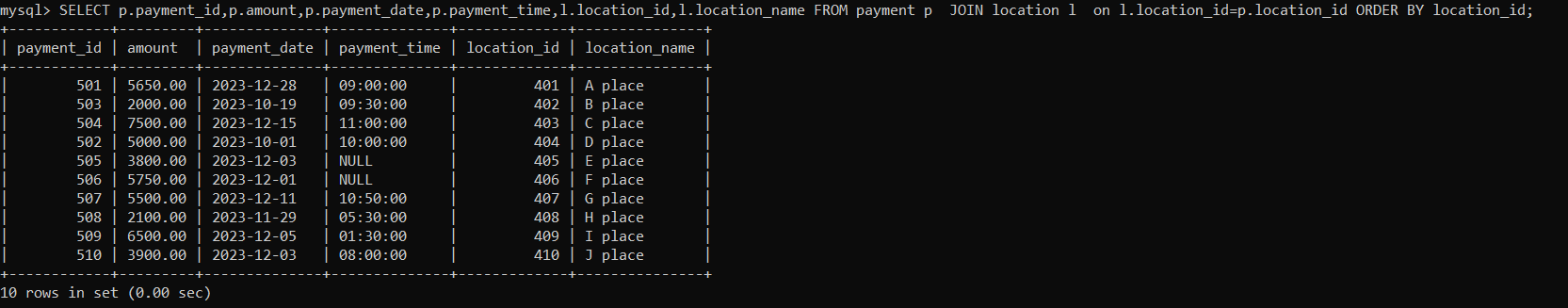
28. List payments made on a specific date



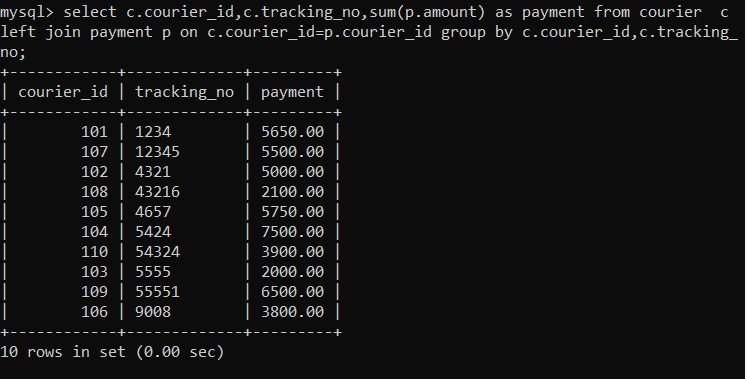
29. Get Courier Information for Each Payment



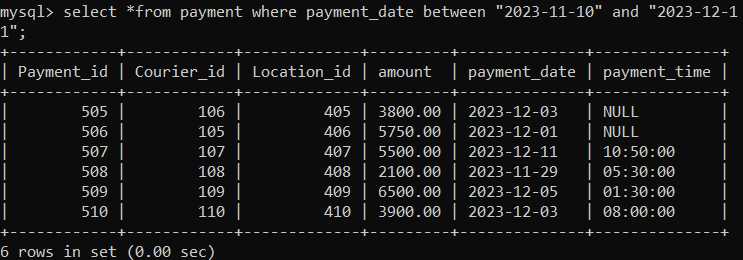
30. Get Payment Details with Location



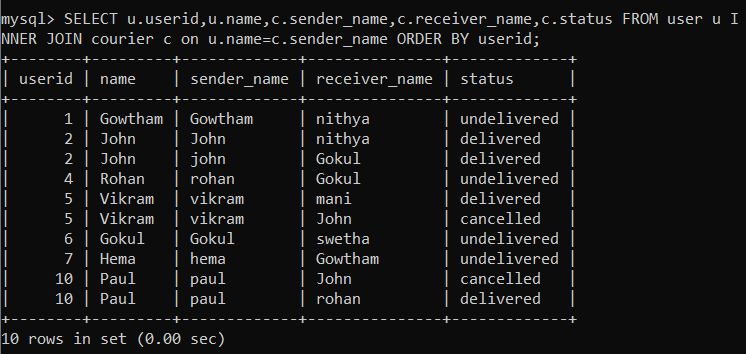
31. Calculating Total Payments for Each Courier



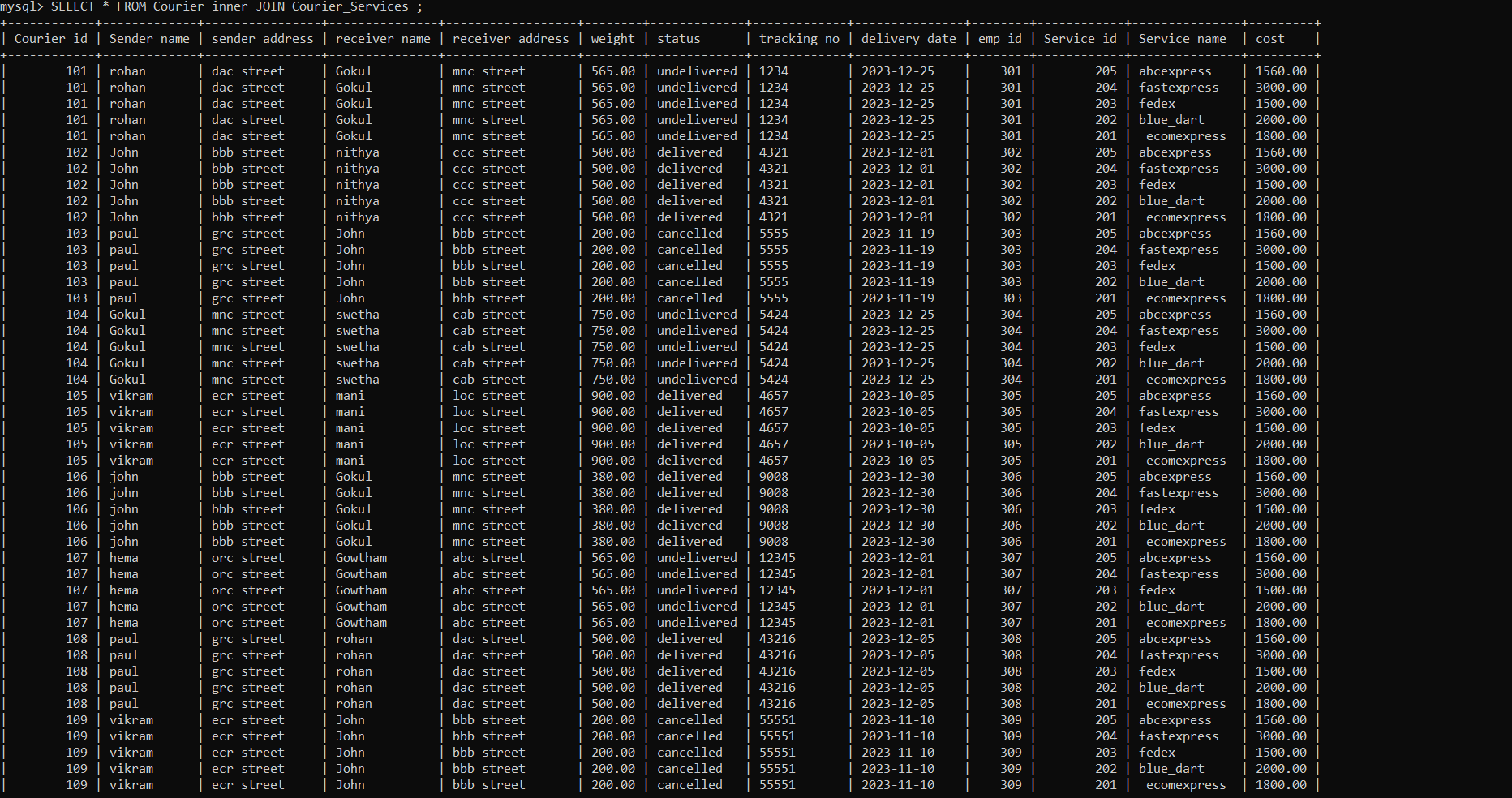
32. List Payments Within a Date Range



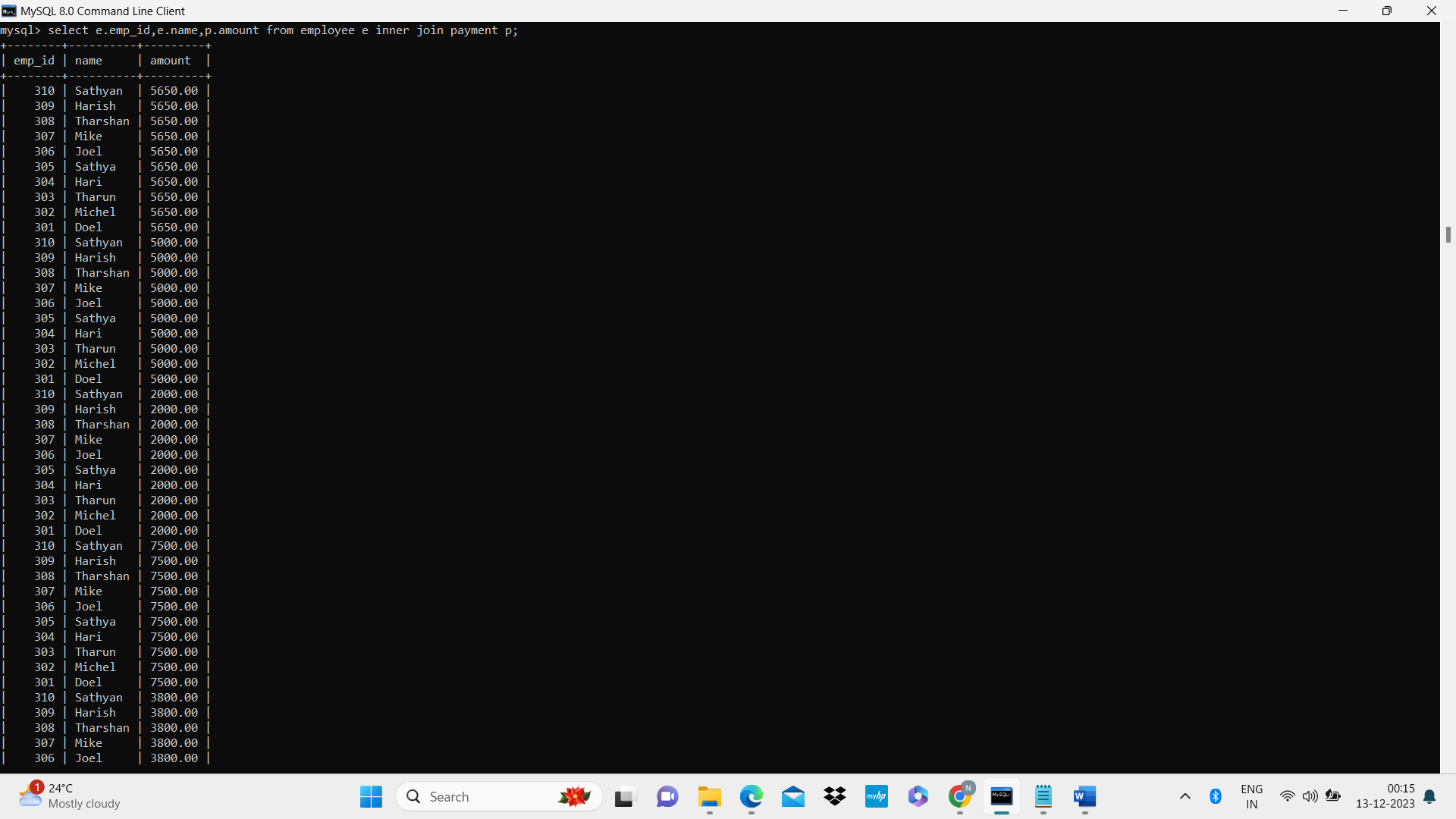
33. Retrieve a list of all users and their corresponding courier records



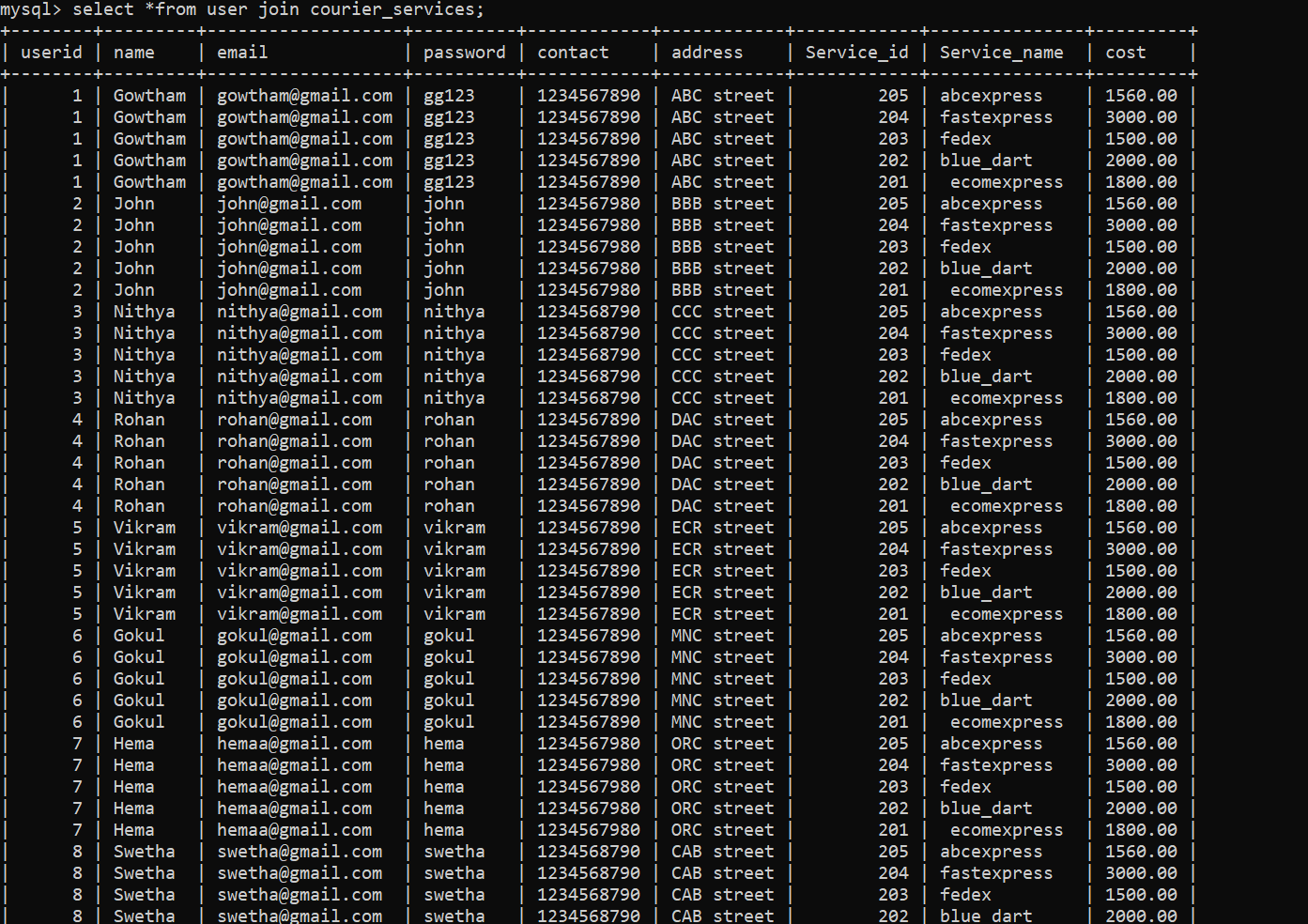
34. Retrieve a list of all couriers and their corresponding services

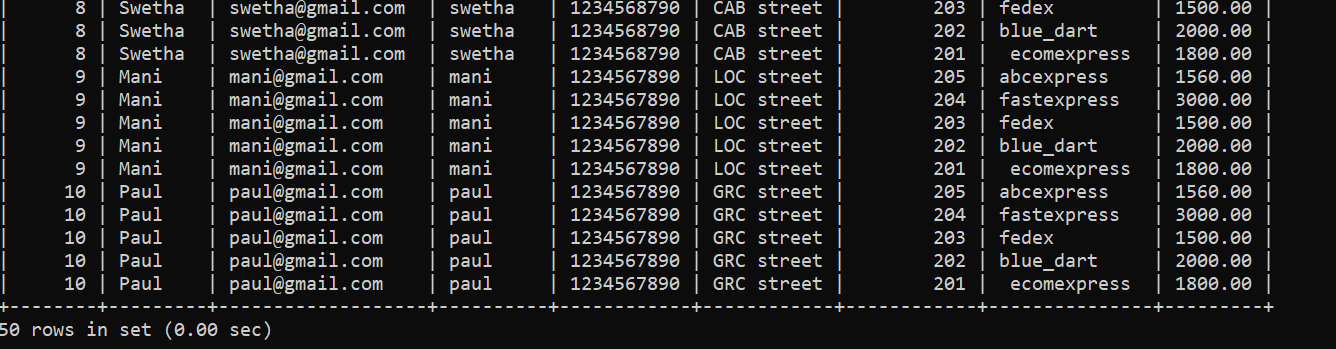


35. Retrieve a list of all employees and their corresponding payments

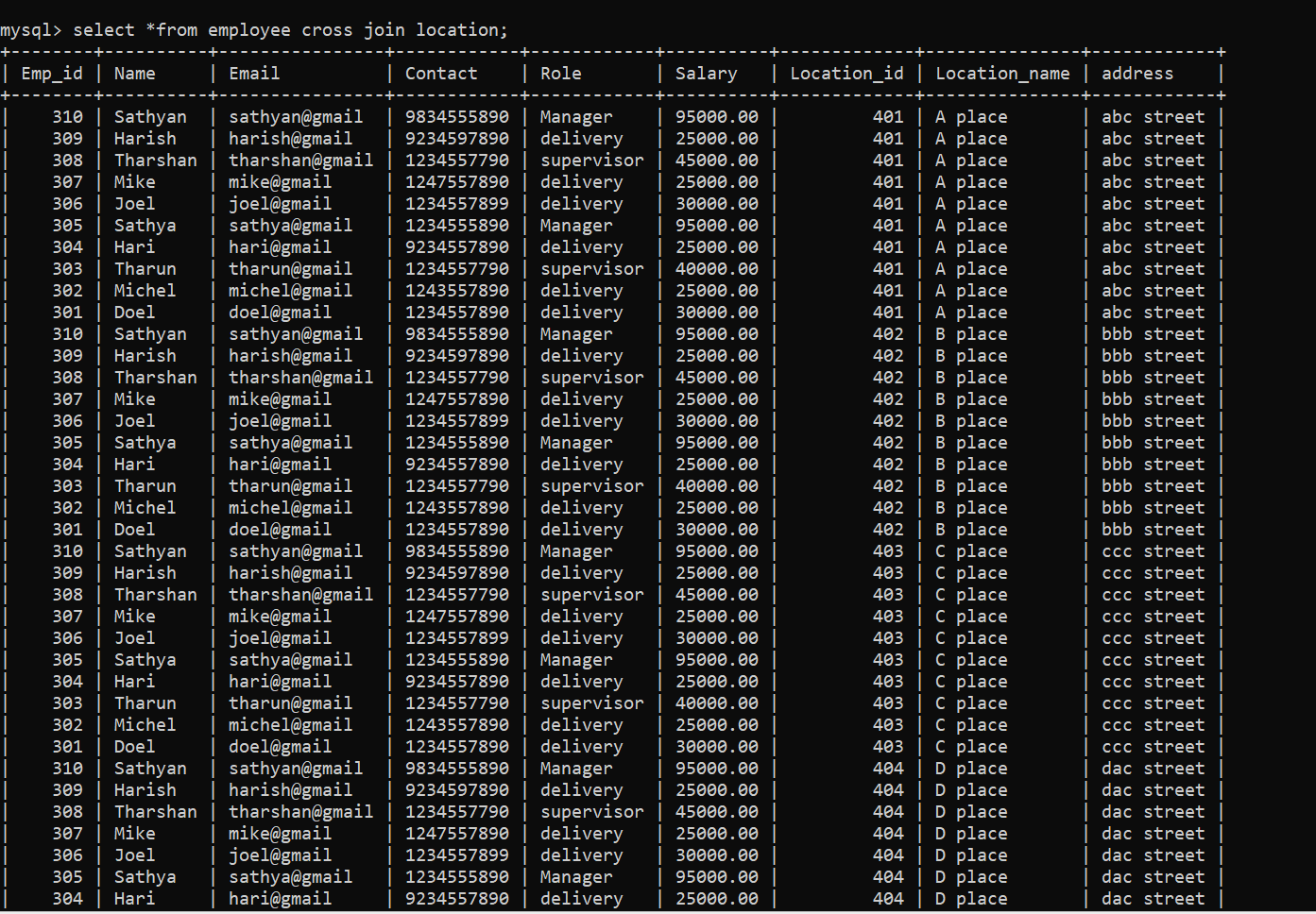


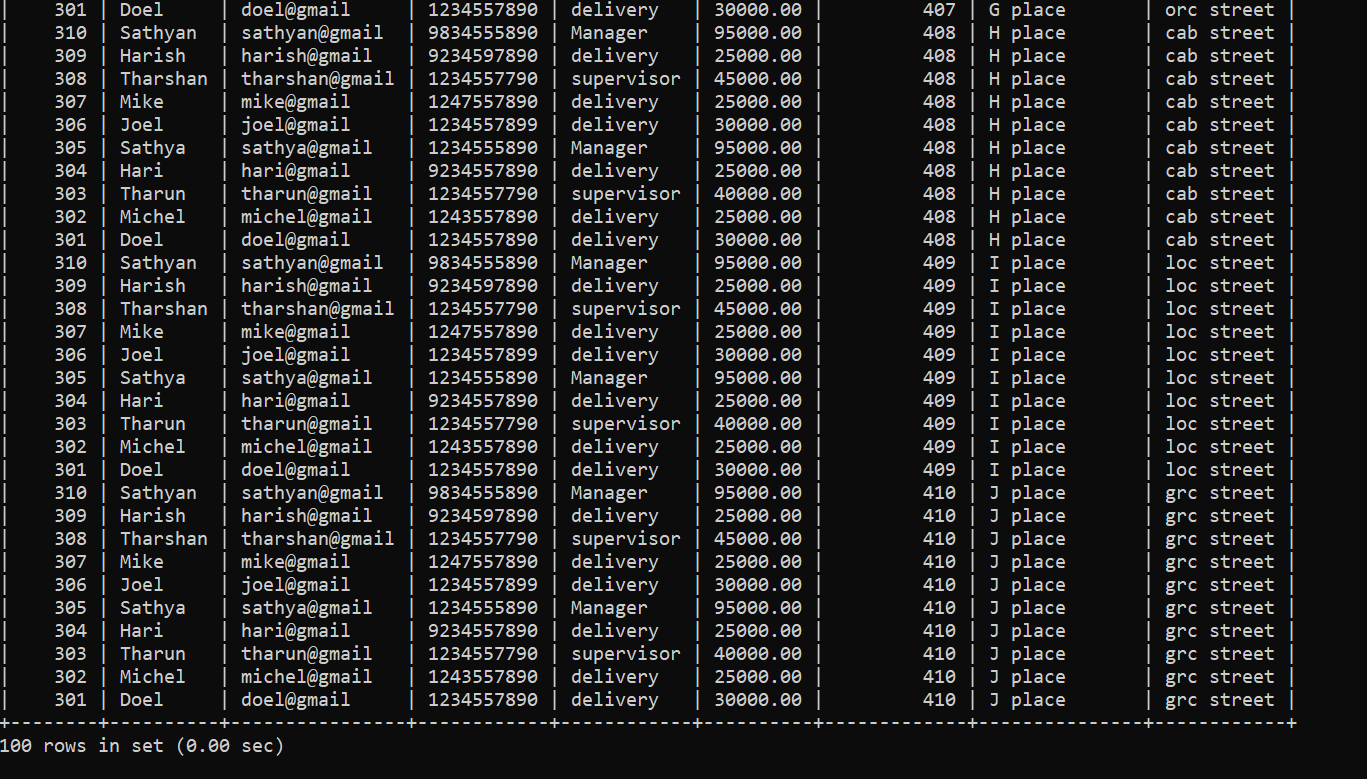
36. List all users and all courier services, showing all possible combinations.



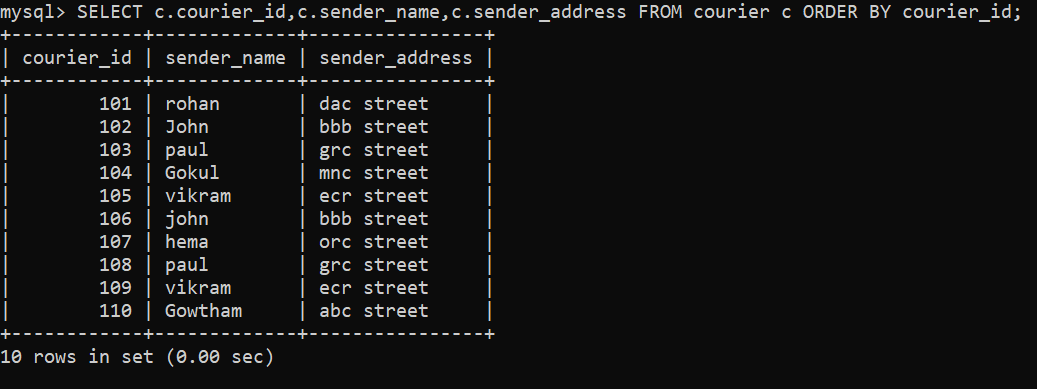


37. List all employees and all locations, showing all possible combinations





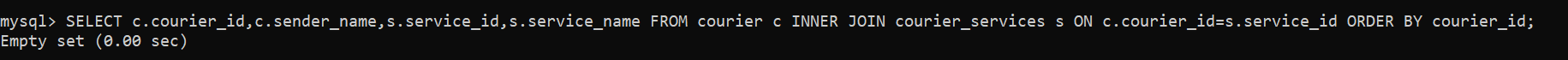
38. Retrieve a list of couriers and their corresponding sender information (if available)



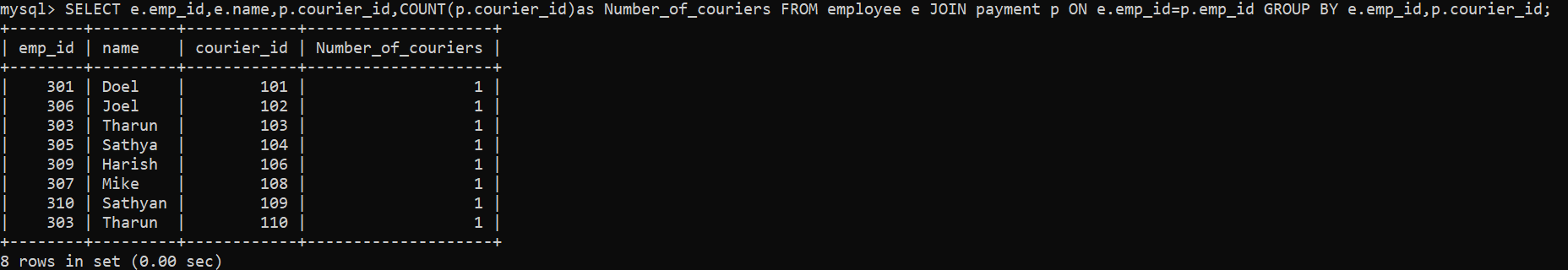
39. Retrieve a list of couriers and their corresponding receiver information (if available)



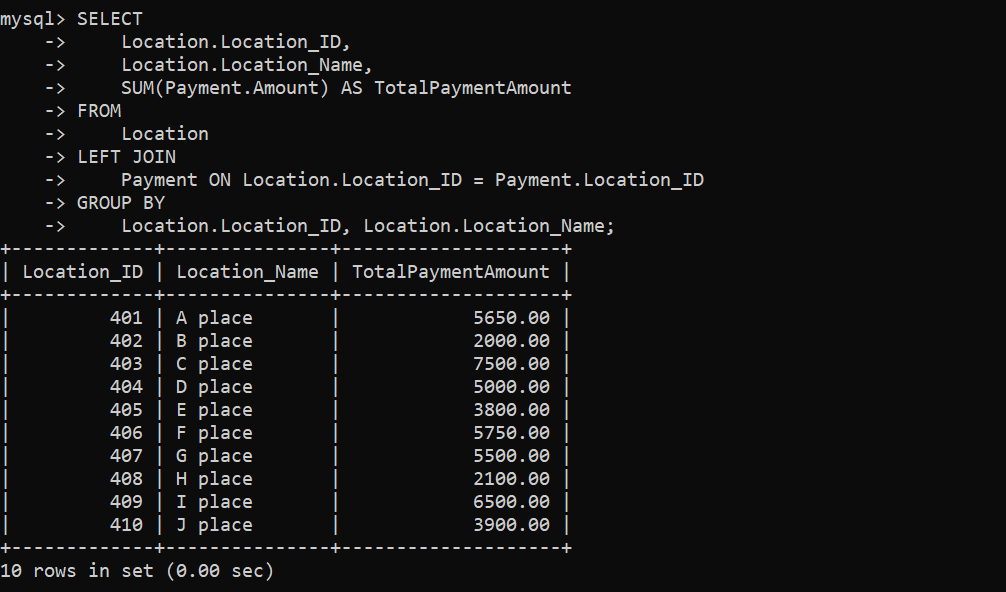
40. Retrieve a list of couriers along with the courier service details (if available)



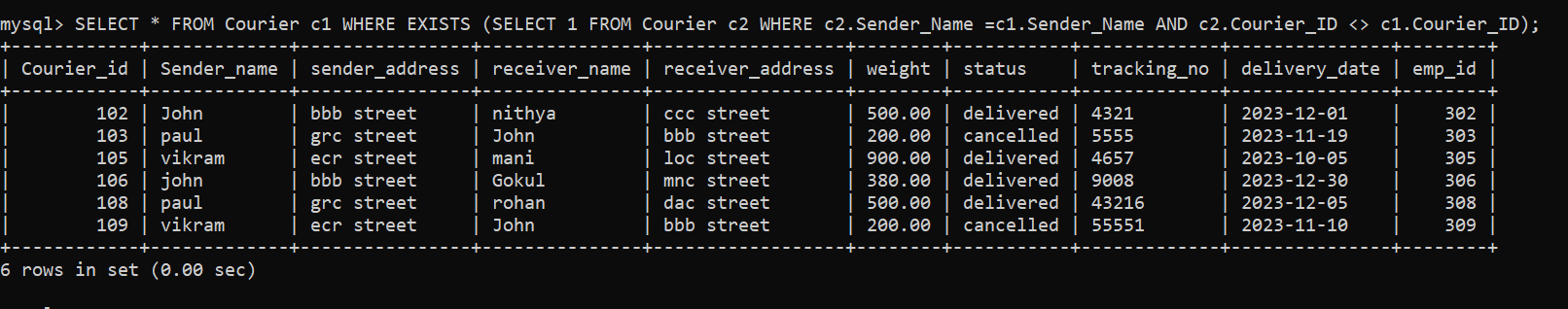
41. Retrieve a list of employees and the number of couriers assigned to each employee



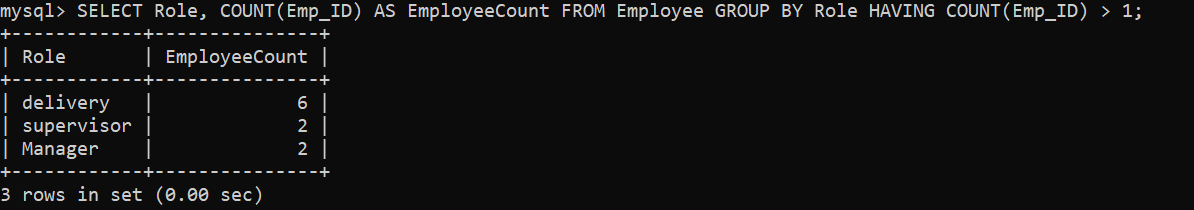
42. Retrieve a list of locations and the total payment amount received at each location



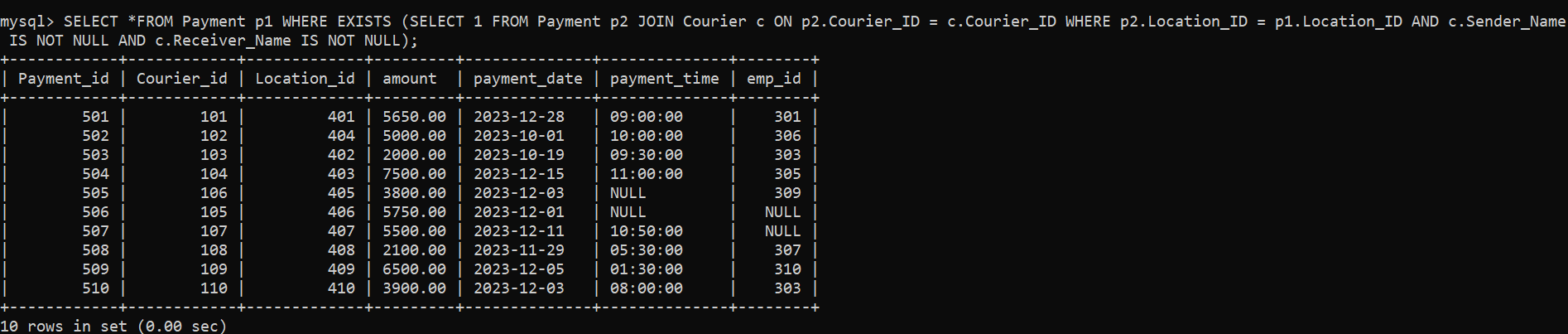
43. Retrieve all couriers sent by the same sender (based on SenderName).



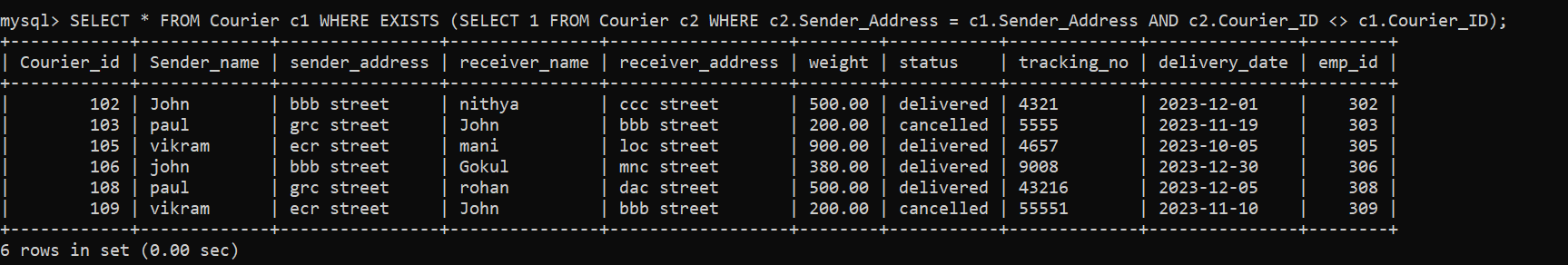
44. List all employees who share the same role.



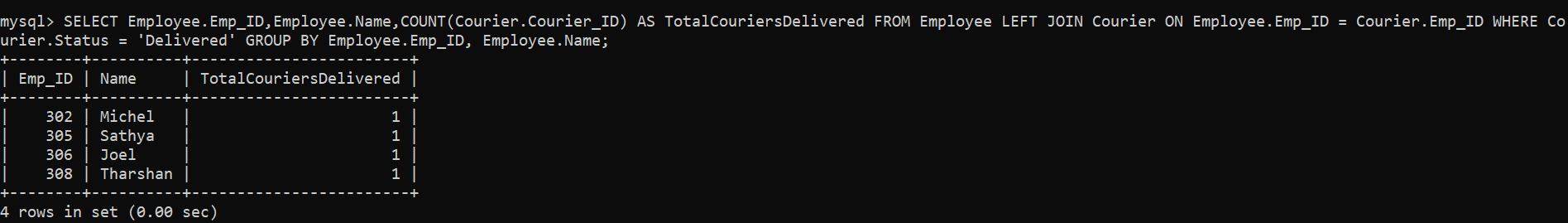
45. Retrieve all payments made for couriers sent from the same location.



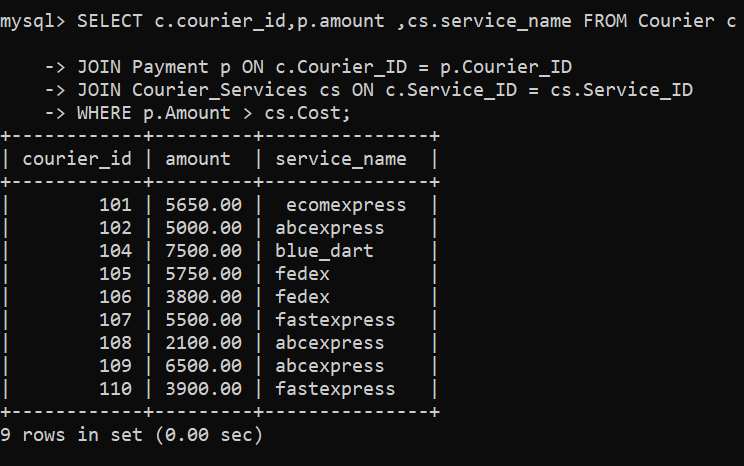
46. Retrieve all couriers sent from the same location (based on SenderAddress).



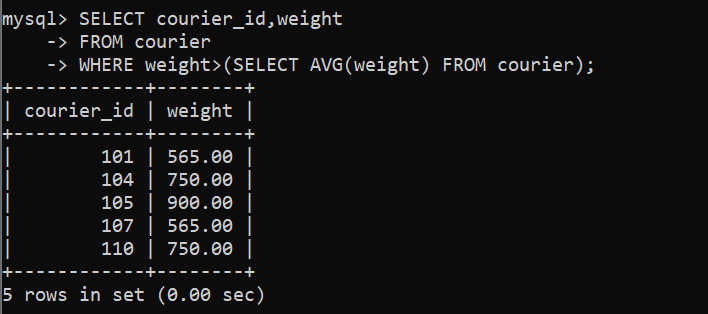
47. List employees and the number of couriers they have delivered



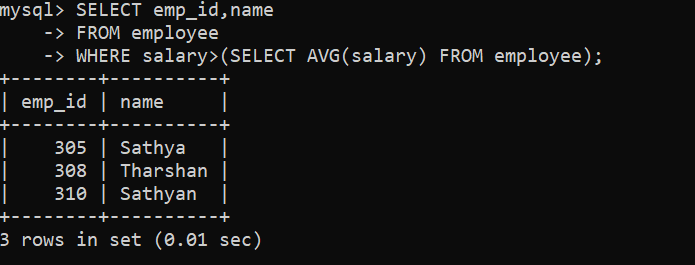
48. Find couriers that were paid an amount greater than the cost of their respective courier services Scope: Inner Queries, Non Equi Joins, Equi joins,Exist,Any,All



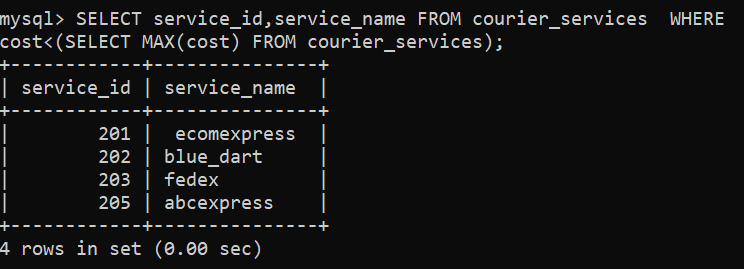
49. Find couriers that have a weight greater than the average weight of all couriers



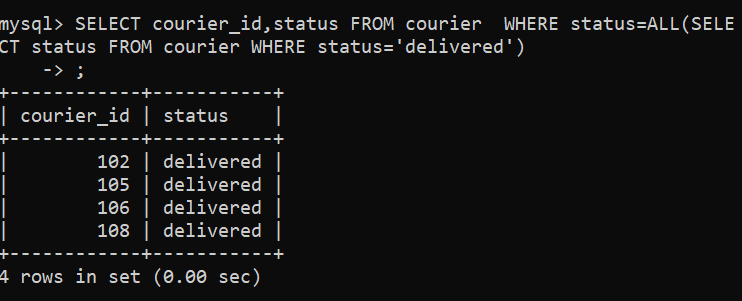
50. Find the names of all employees who have a salary greater than the average salary:



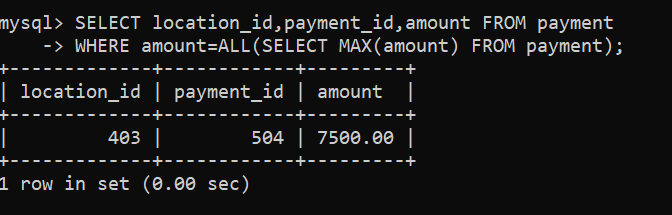
51. Find the total cost of all courier services where the cost is less than the maximum cost



52. Find all couriers that have been paid for



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



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

