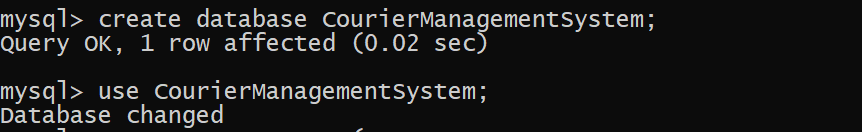
**Task1 Database Design**

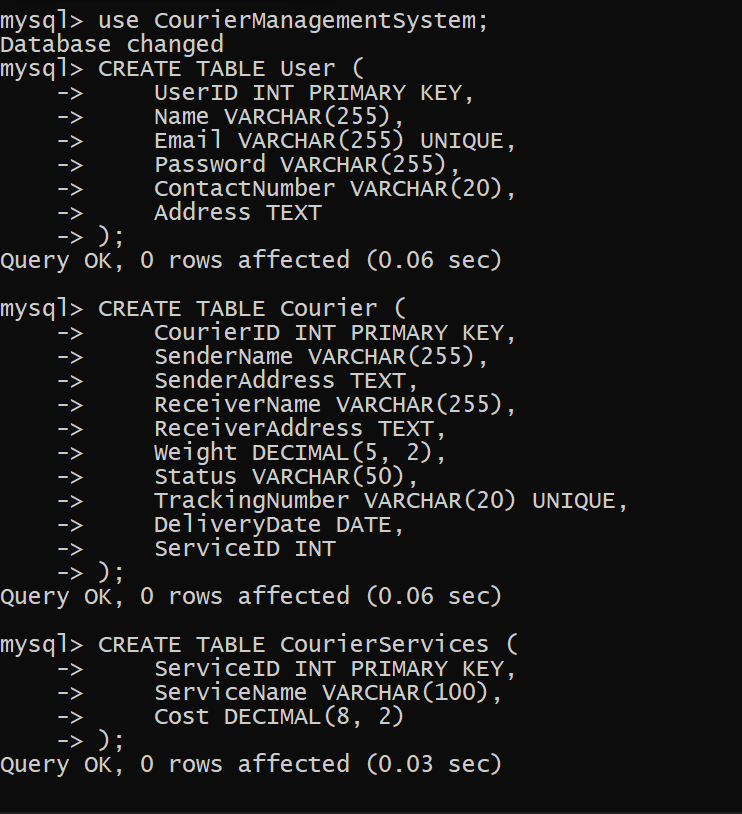
Design a SQL schema for a Courier Management System with tables for Customers, Couriers, Orders, and Parcels. Define the relationships between these tables using appropriate foreign keys.

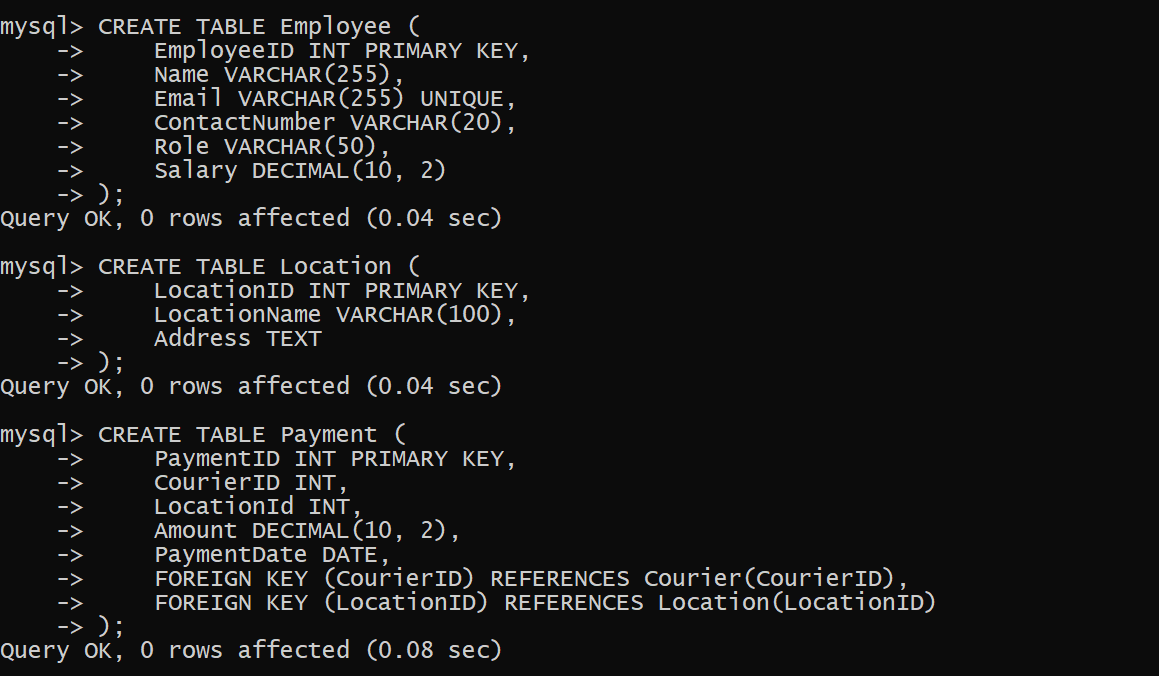


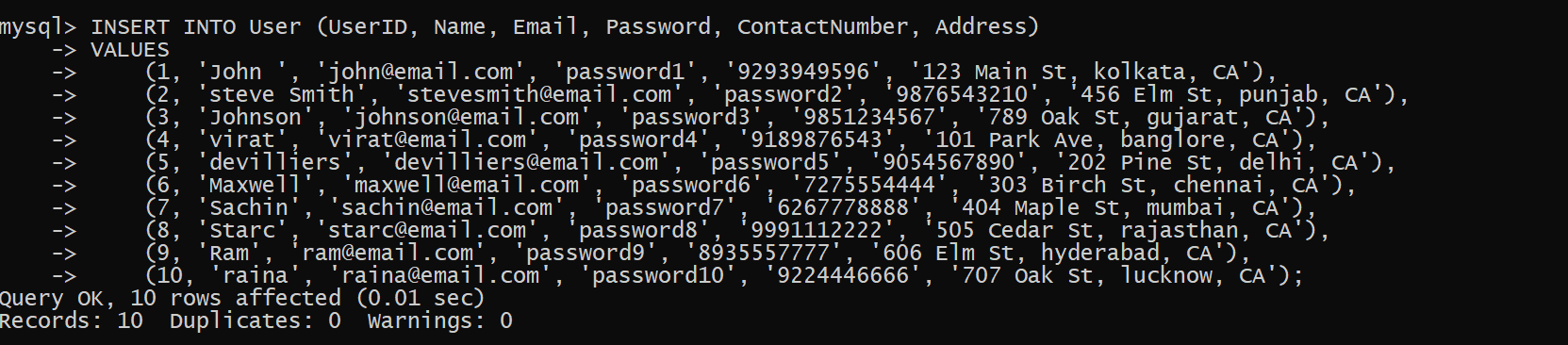
**Requirements:**

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

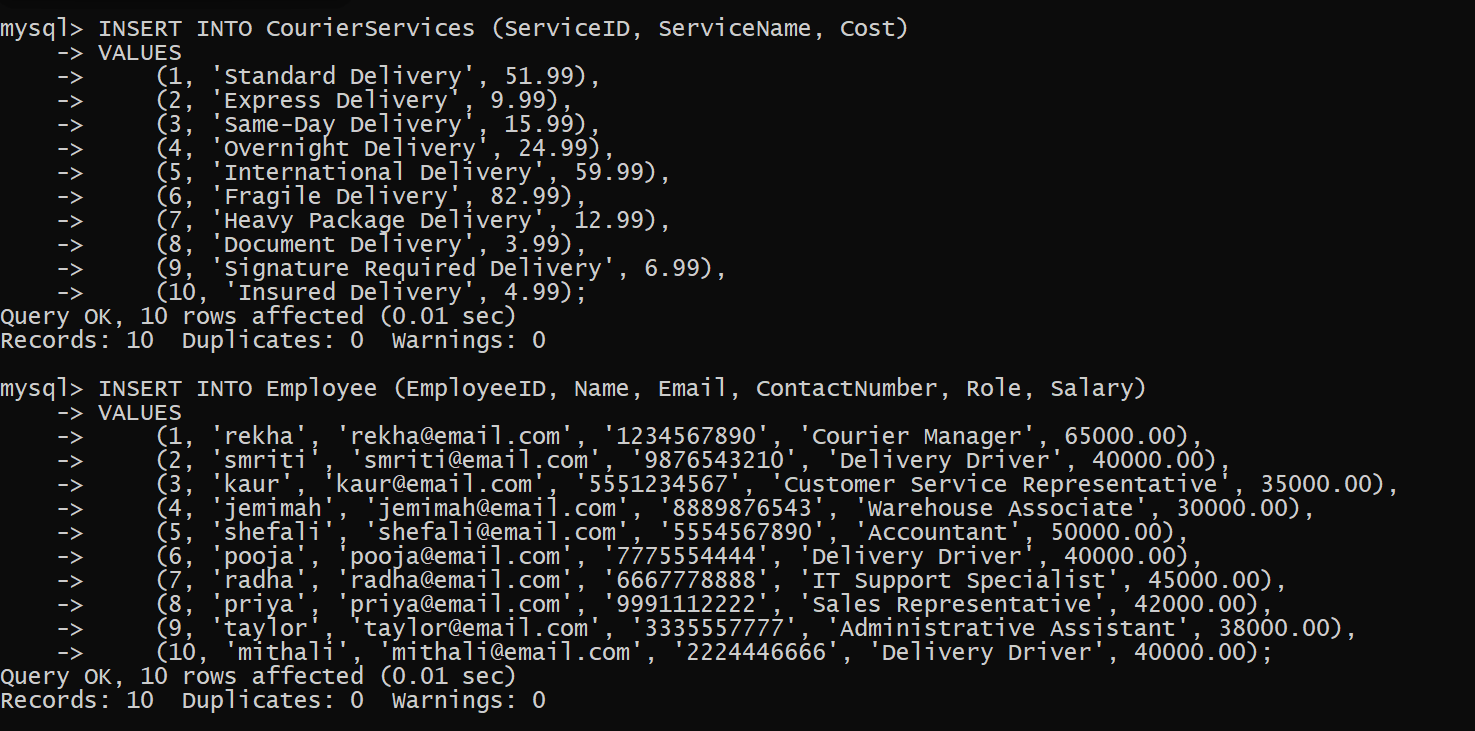
• Define relationships between these tables (one-to-many, many-to-many, etc.).

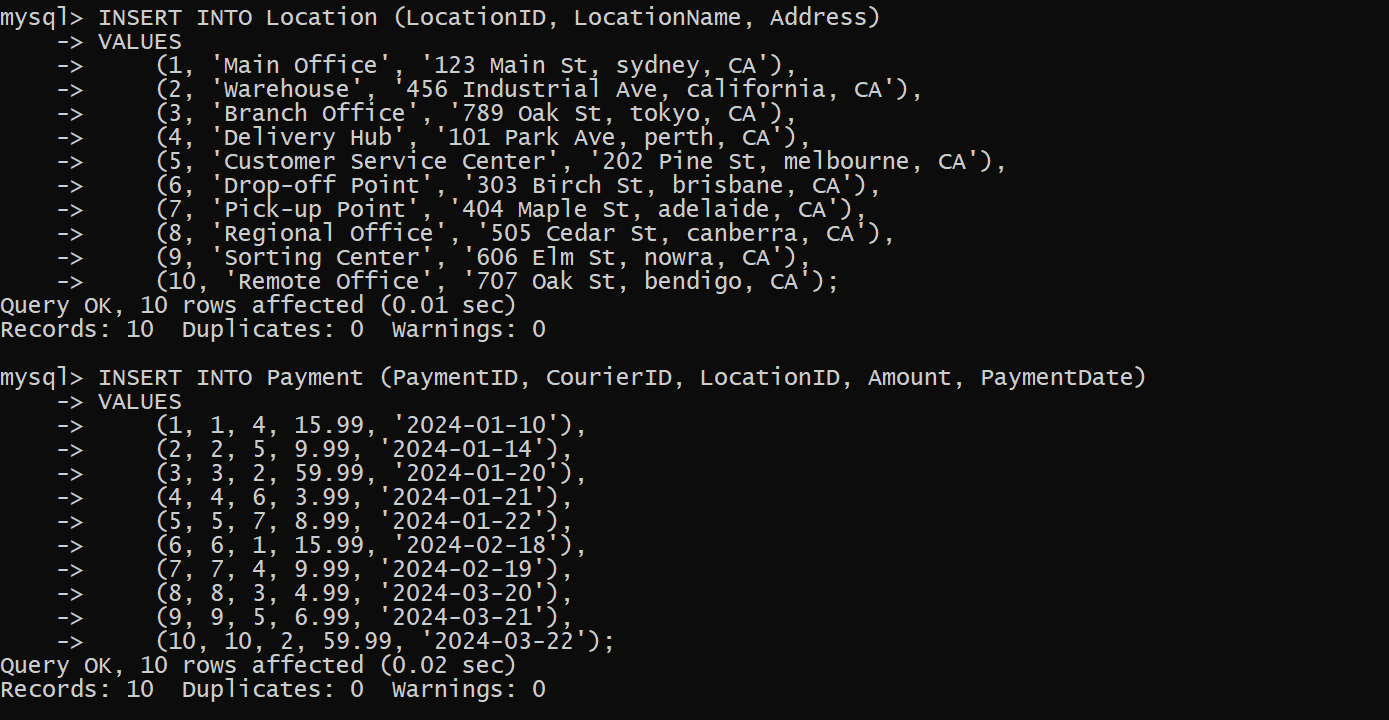








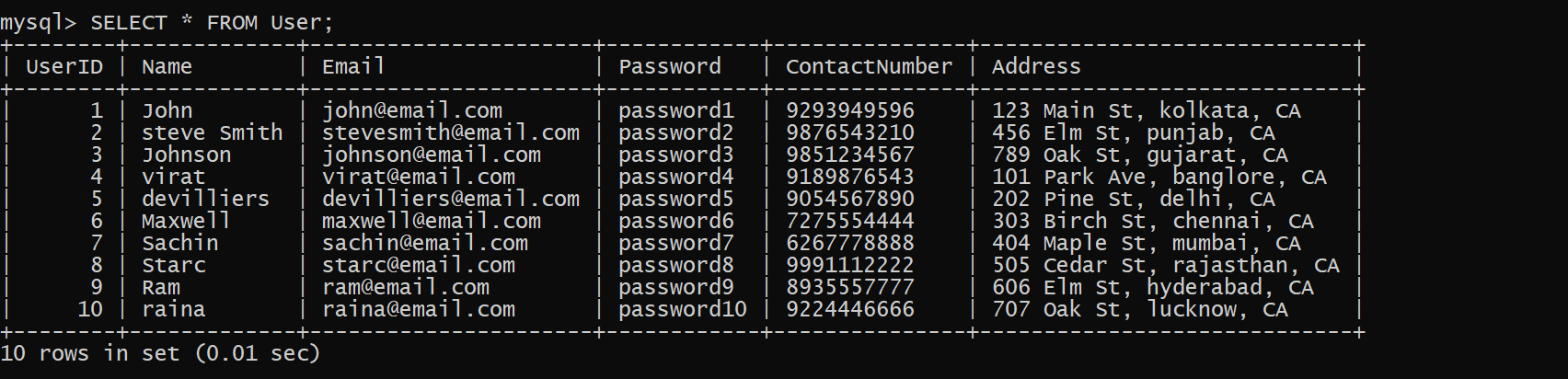




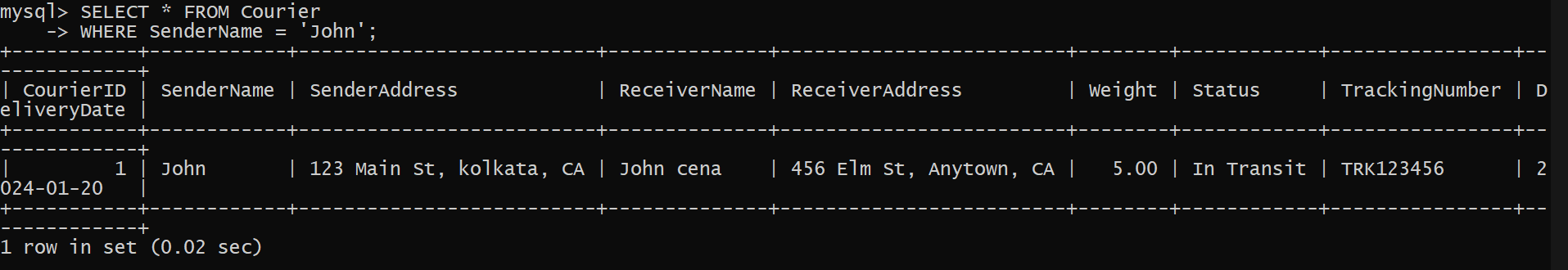
**Task 2: Select,Where**

Solve the following queries in the Schema that you have created above

1. List all customers:



2. List all orders for a specific customer:



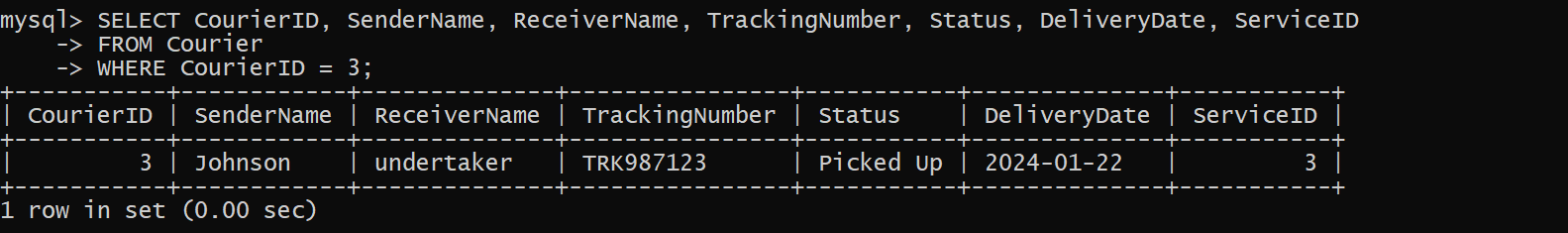
3.List all couriers:



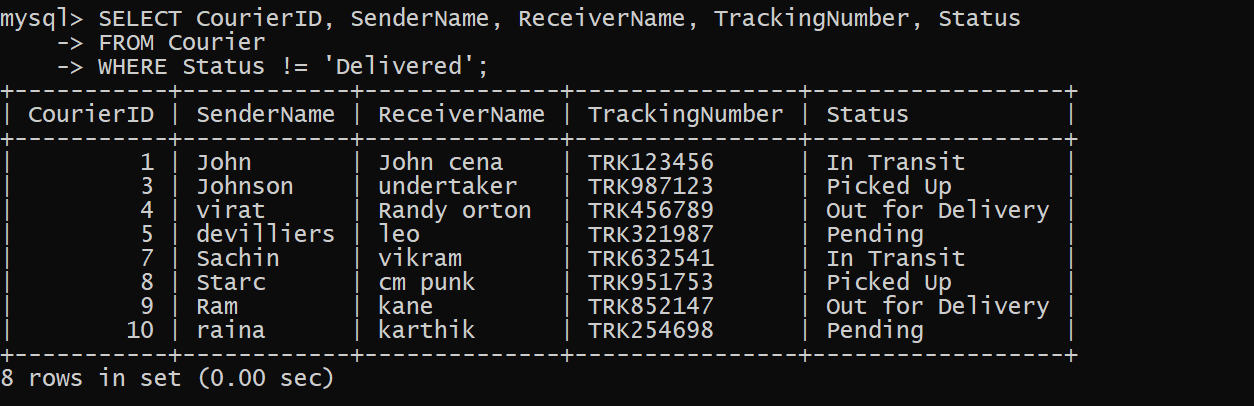
1. List all packages for a specific order:



1. List all deliveries for a specific courier:



1. List all undelivered packages:



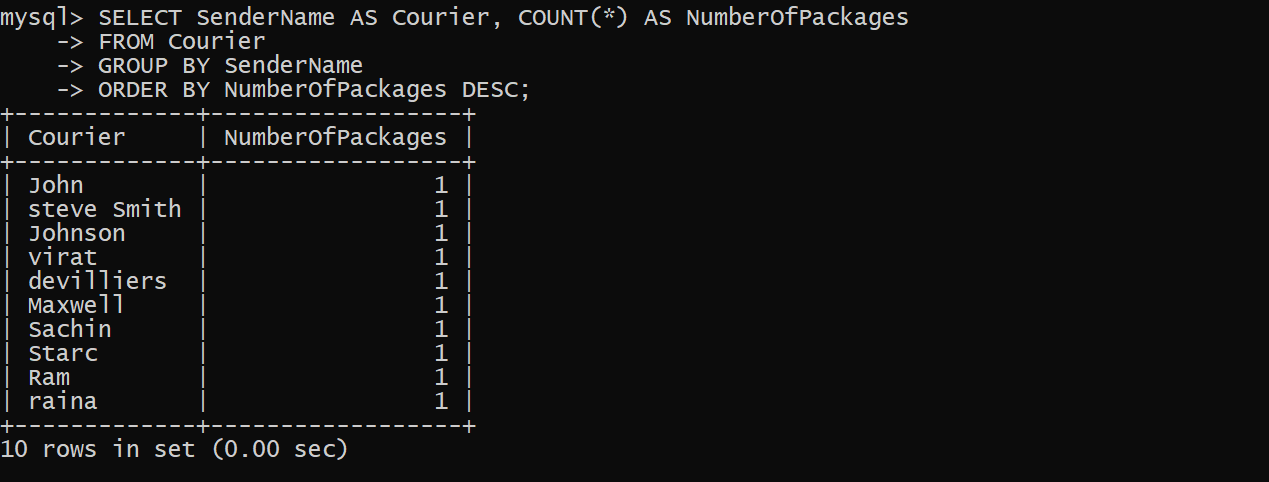
1. List all packages that are scheduled for delivery today:



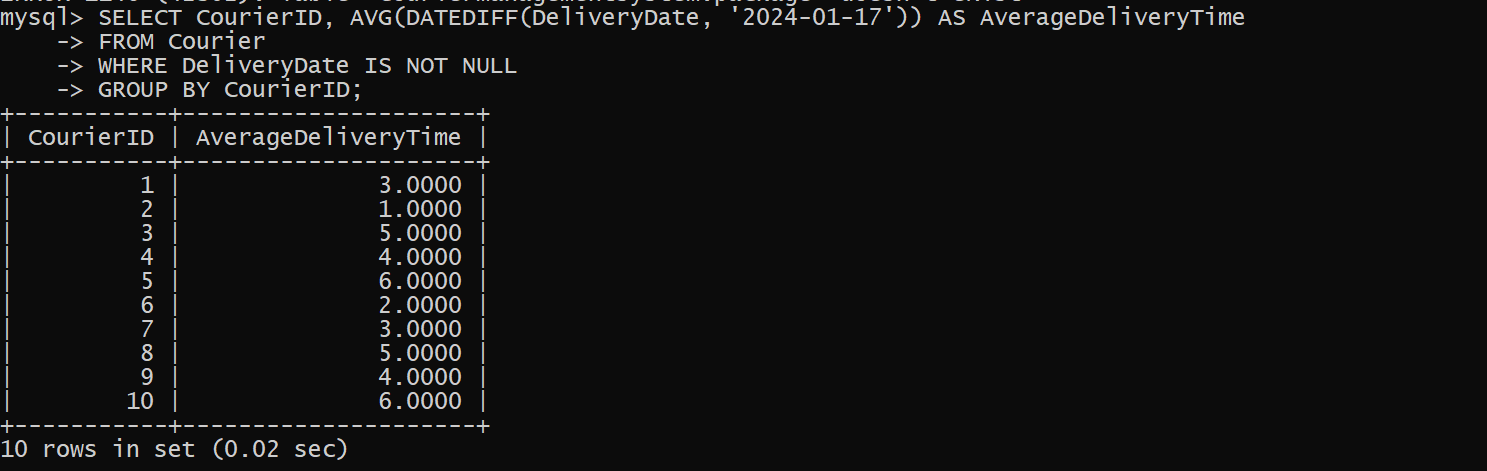
1. List all packages with a specific status:



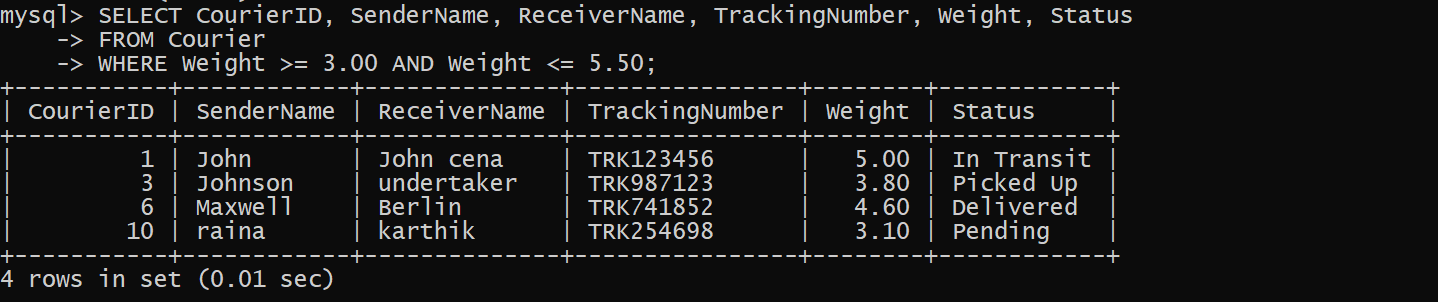
1. Calculate the total number of packages for each courier.



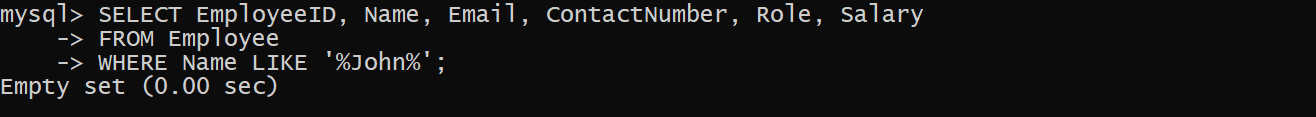
1. Find the average delivery time for each courier



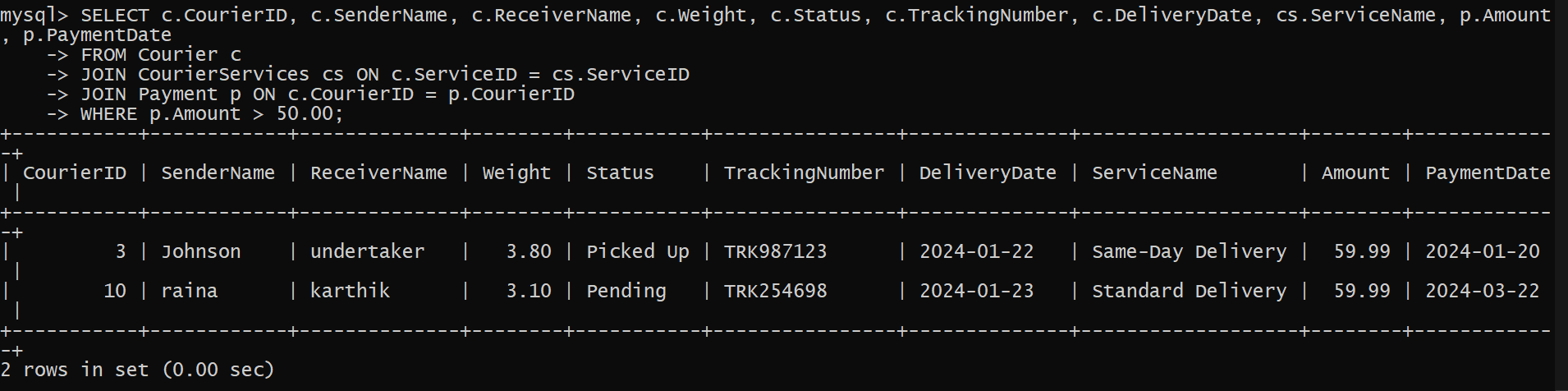
1. List all packages with a specific weight range:



1. Retrieve employees whose names contain 'John'

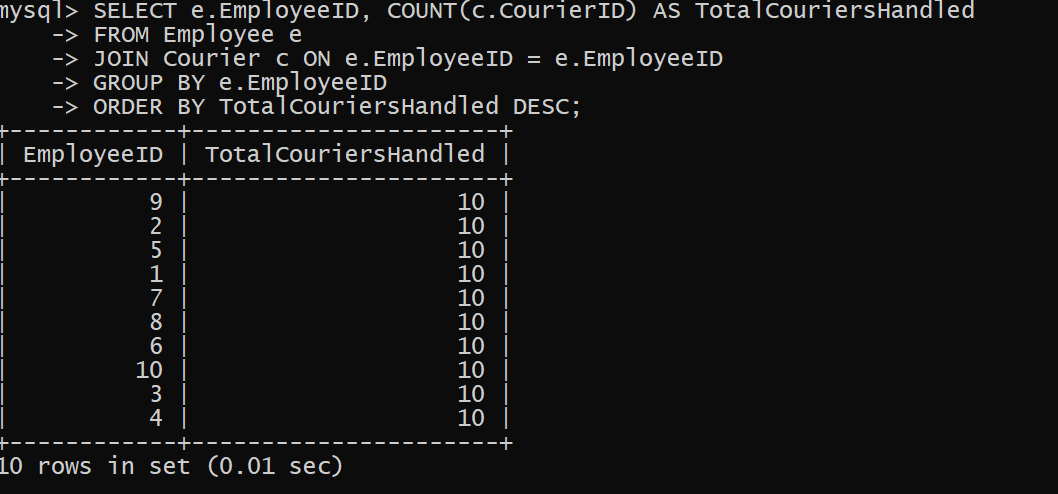


1. Retrieve all courier records with payments greater than $50.

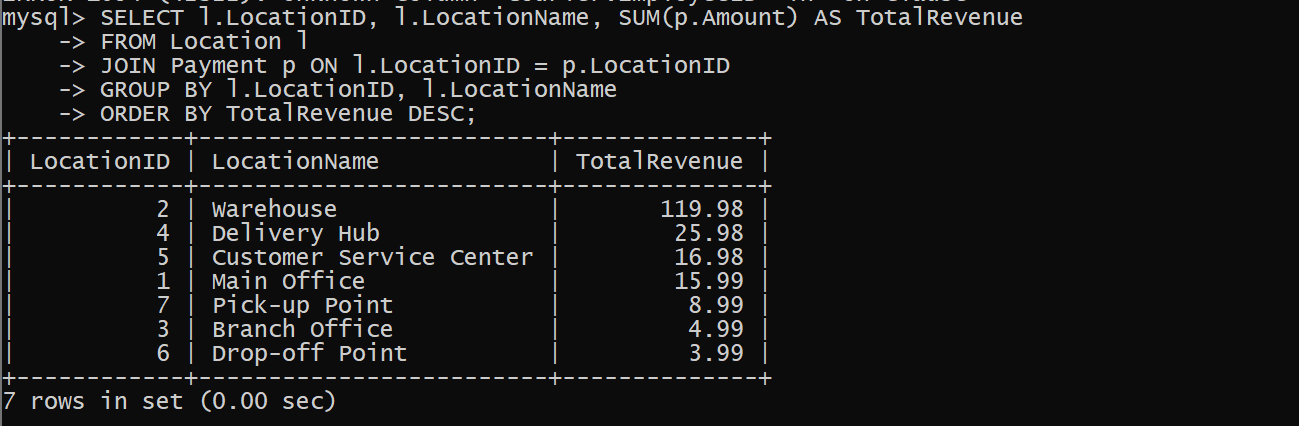


**Task 3: GroupBy, Aggregate Functions, Having, Order By, where**

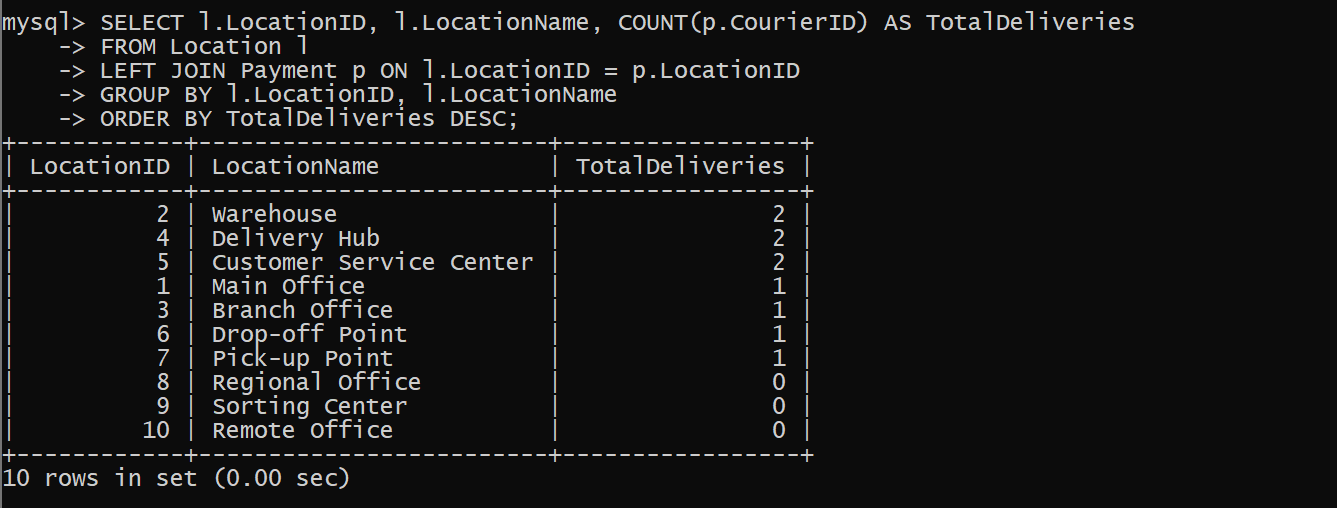
1. Find the total number of couriers handled by each employee.



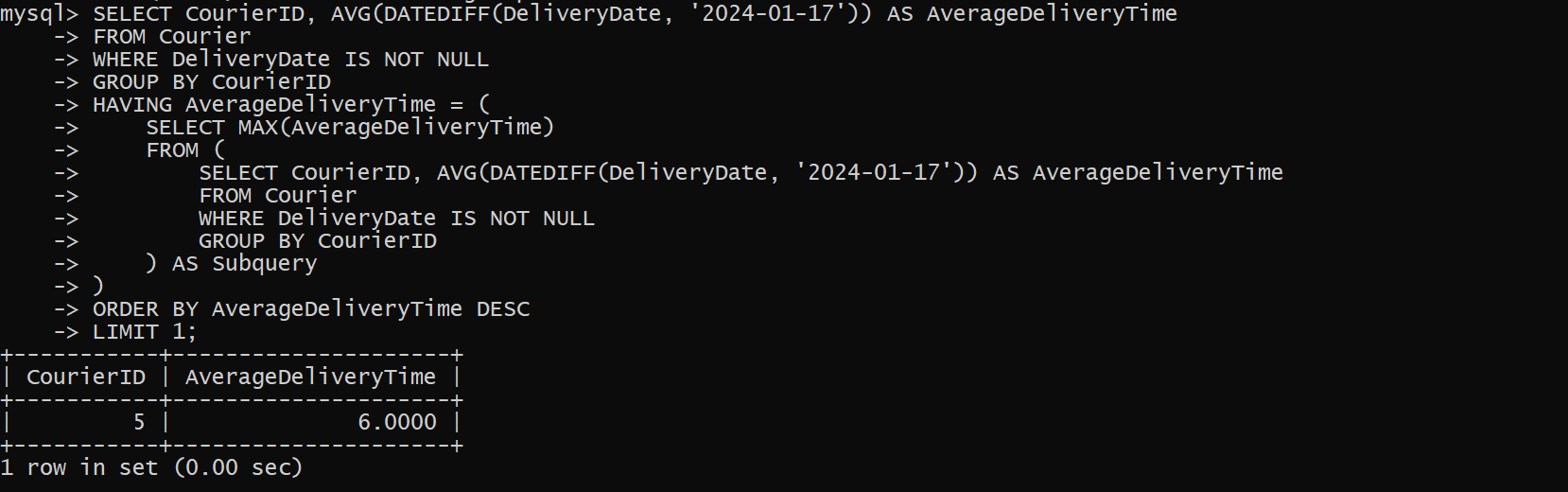
1. Calculate the total revenue generated by each location



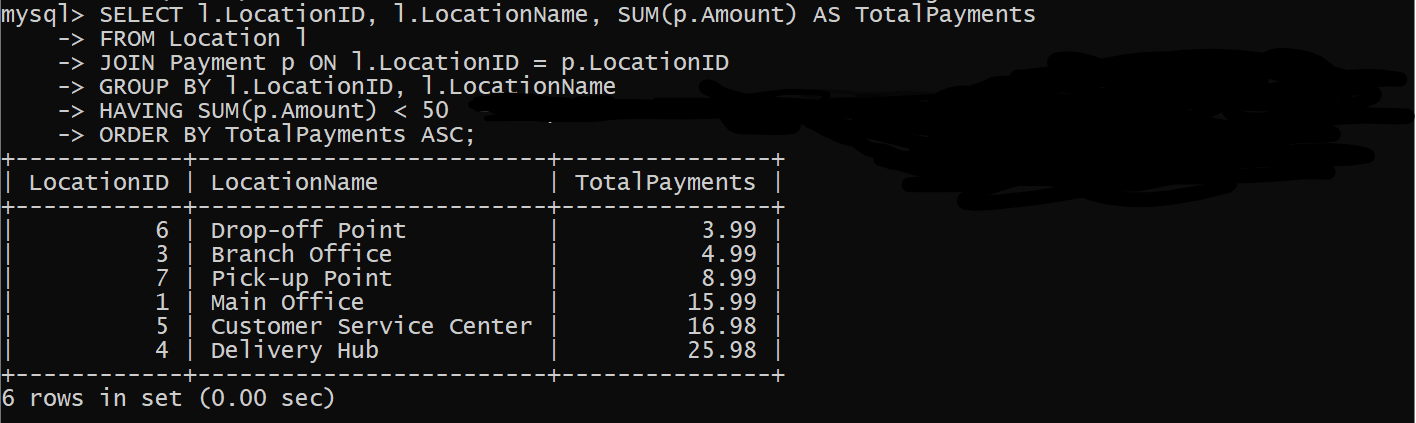
1. Find the total number of couriers delivered to each location.



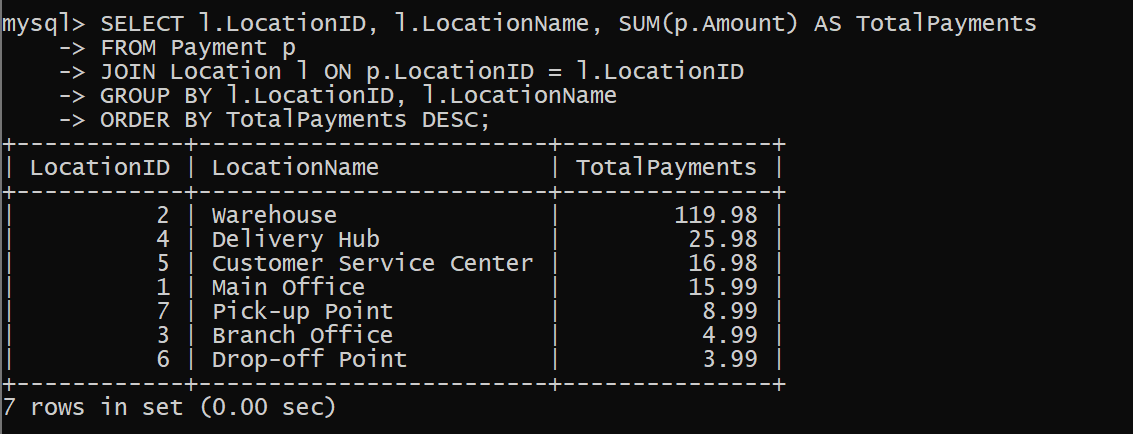
1. Find the courier with the highest average delivery time:



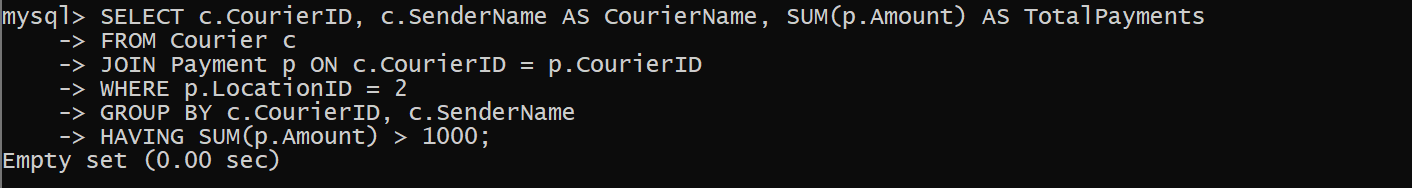
1. Find Locations with Total Payments Less Than a Certain Amount



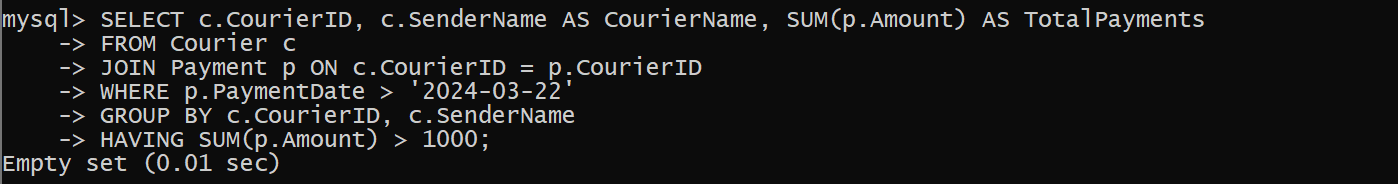
1. Calculate Total Payments per Location



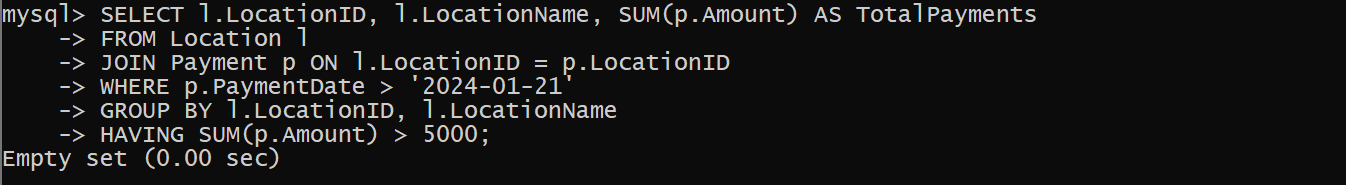
1. Retrieve couriers who have received payments totaling more than $1000 in a specific location (LocationID = 2):



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

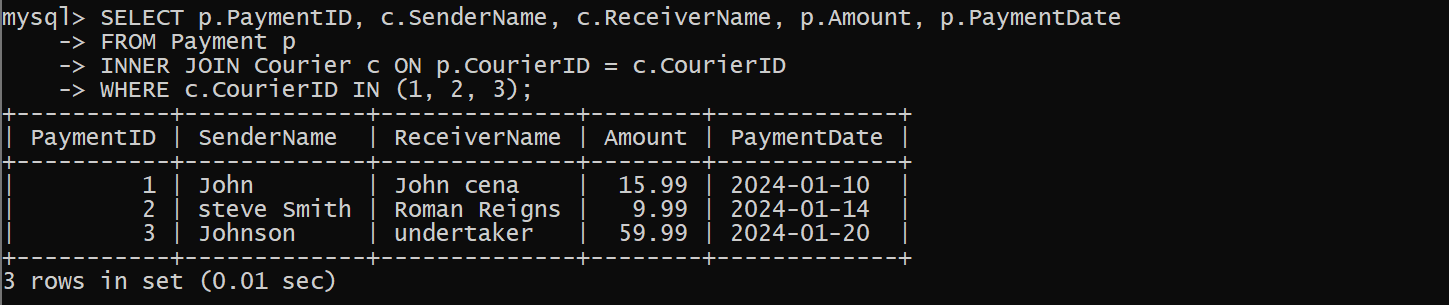


1. Retrieve locations where the total amount received is more than $5000 before a certain date (PaymentDate > 'YYYY-MM-DD')

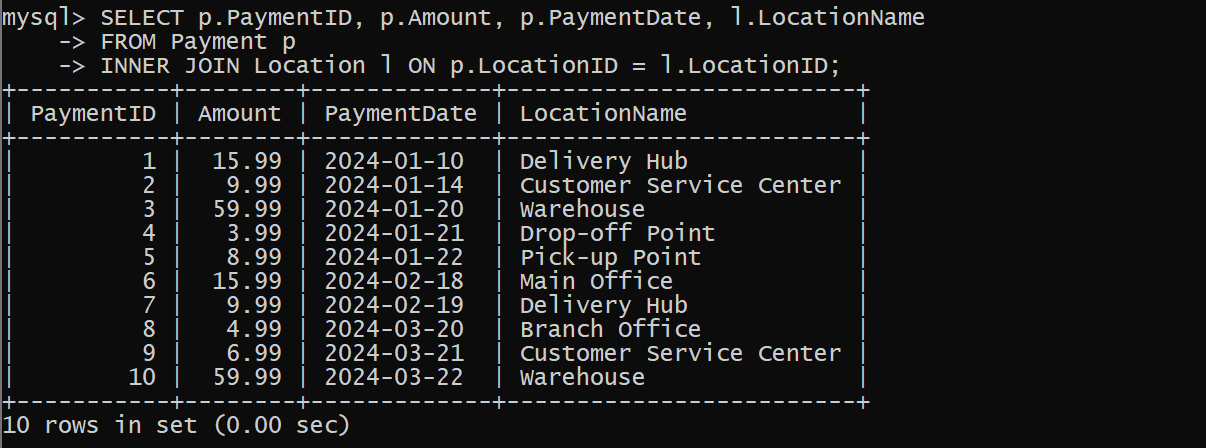


**Task 4: Inner Join,Full Outer Join, Cross Join, Left Outer Join,Right Outer Join**

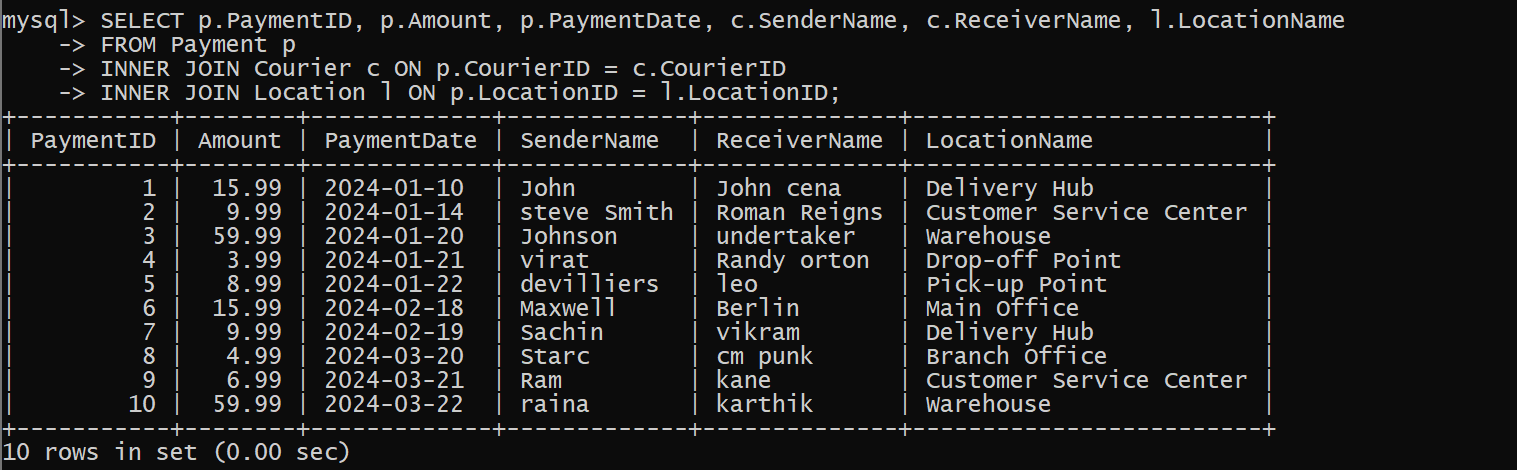
23.Retrieve Payments with Courier Information



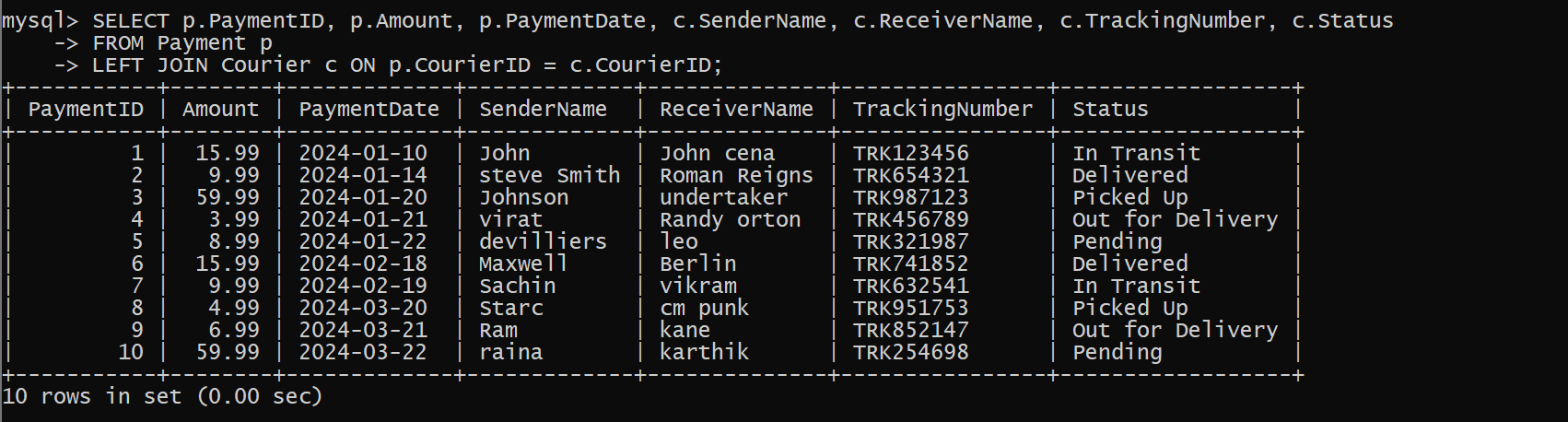
24.Retrieve Payments with Location Information



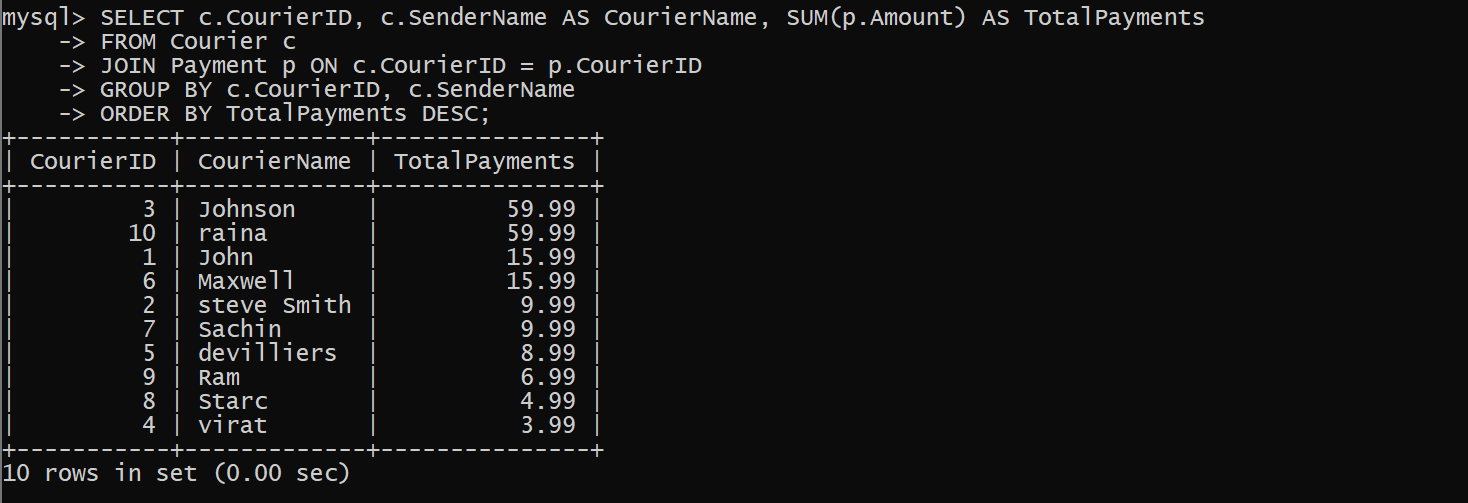
25.Retrieve Payments with Courier and Location Information



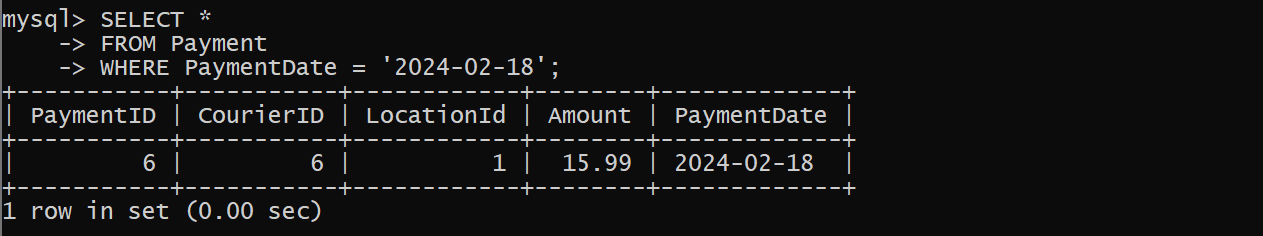
26.List all payments with courier details



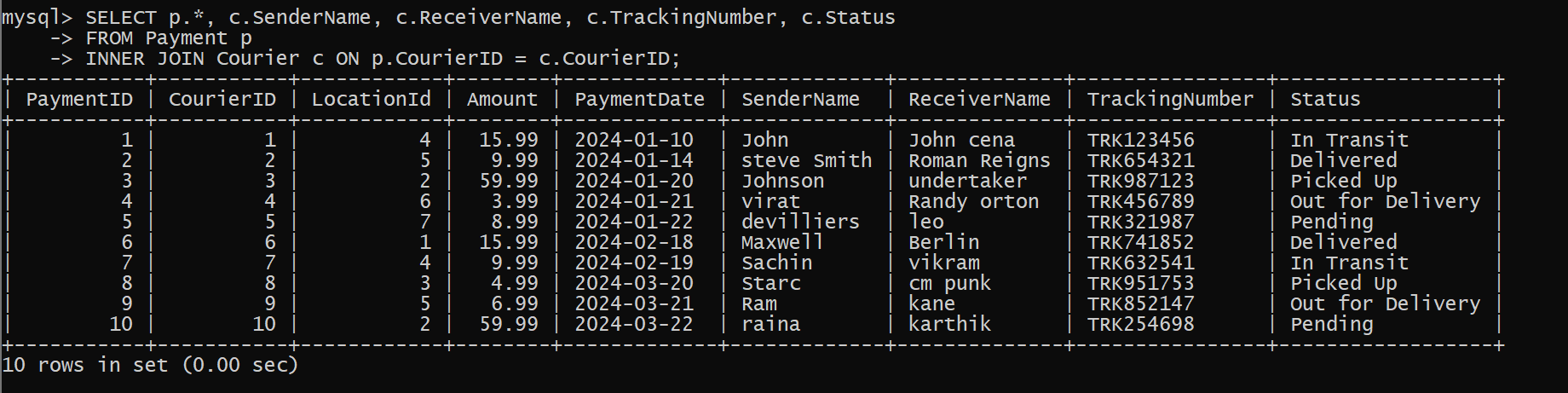
27.Total payments received for each courier



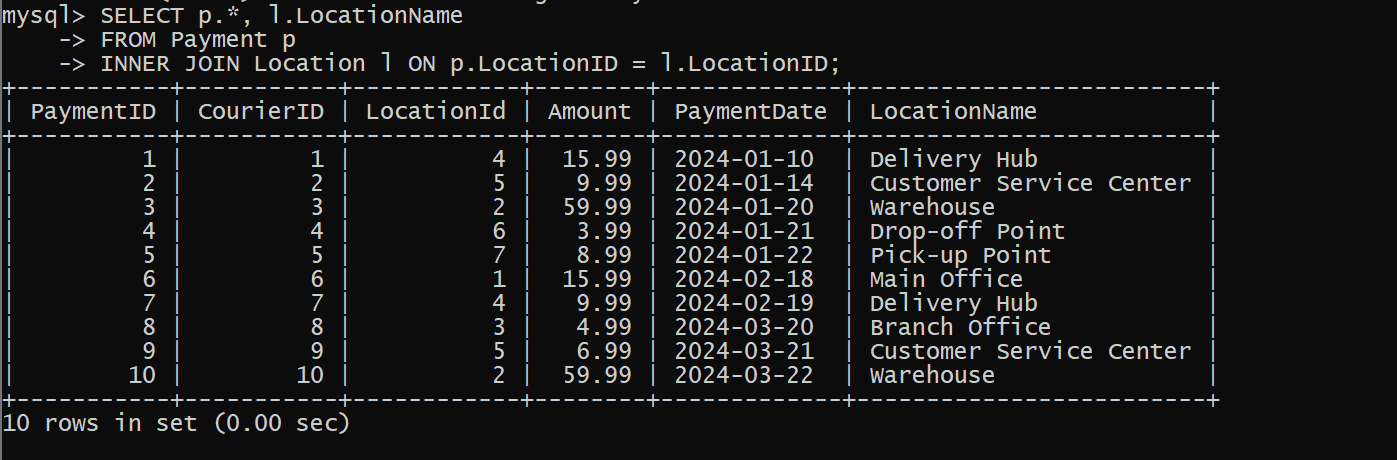
28.List payments made on a specific date



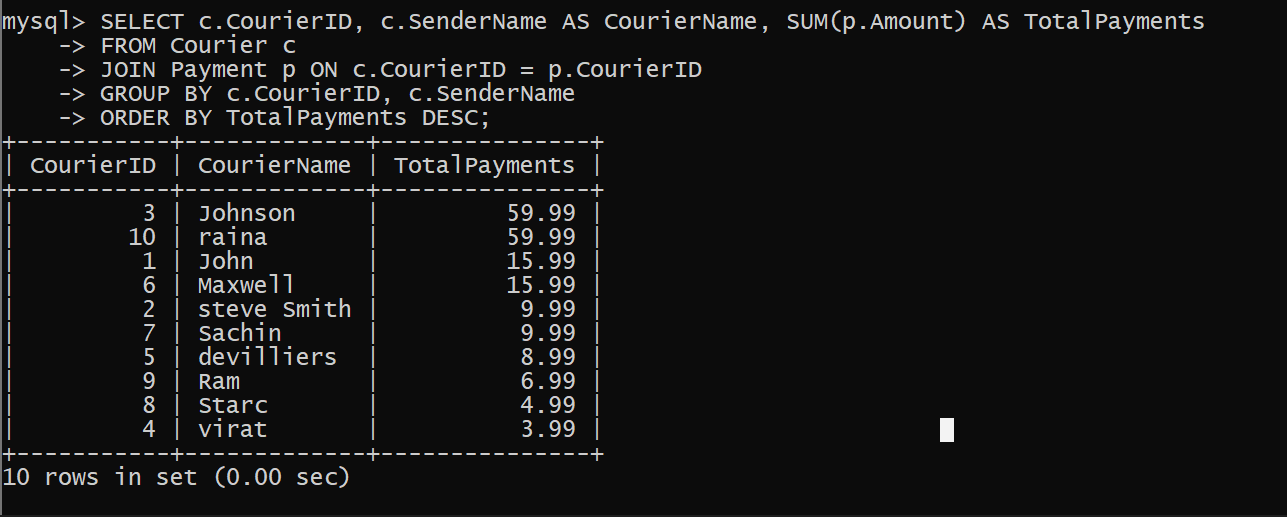
1. Get Courier Information for Each Payment



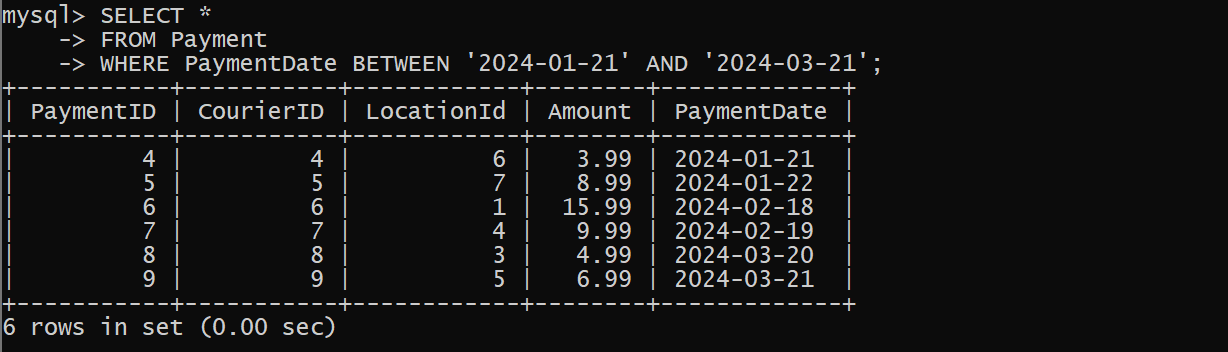
1. Get Payment Details with Location



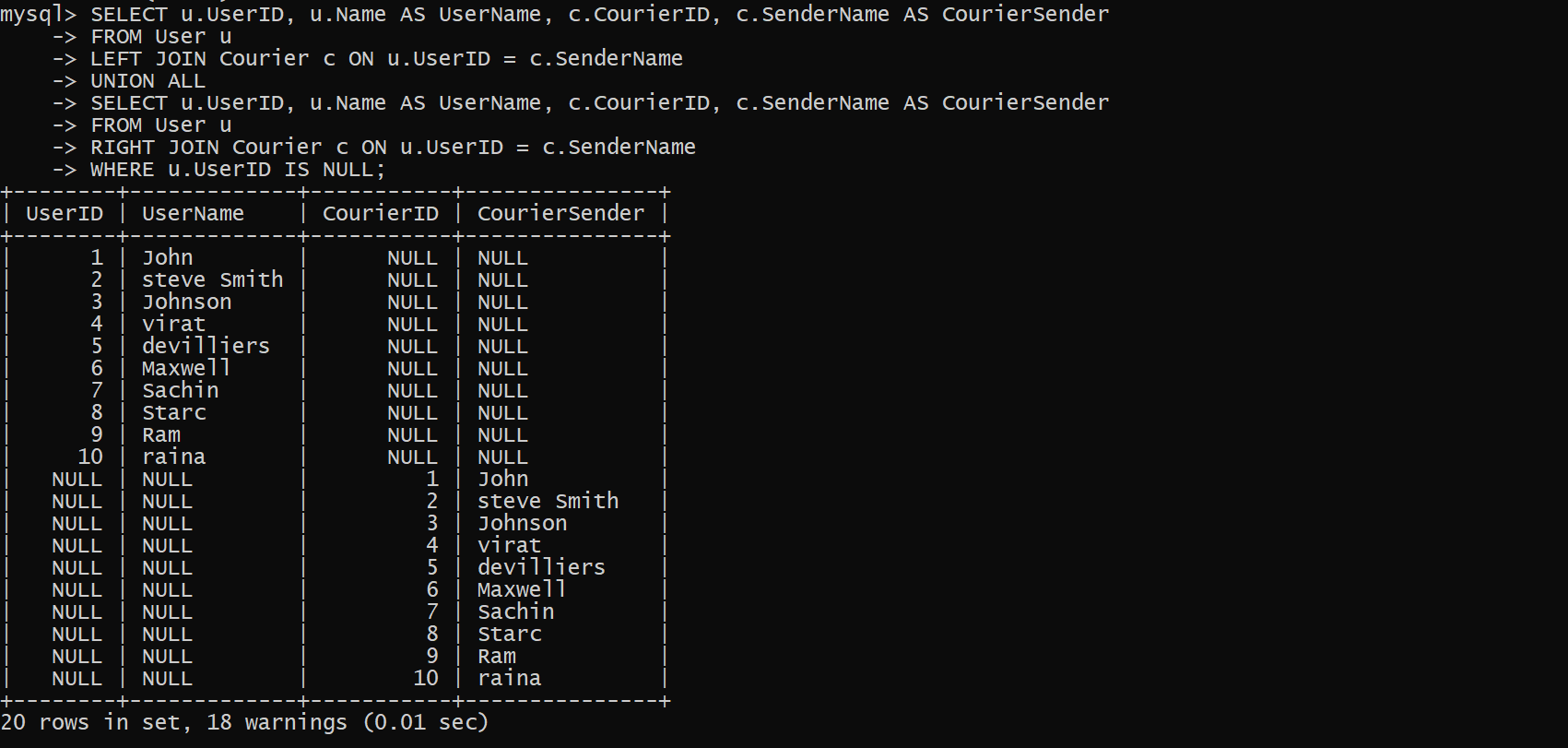
1. Calculating Total Payments for Each Courier



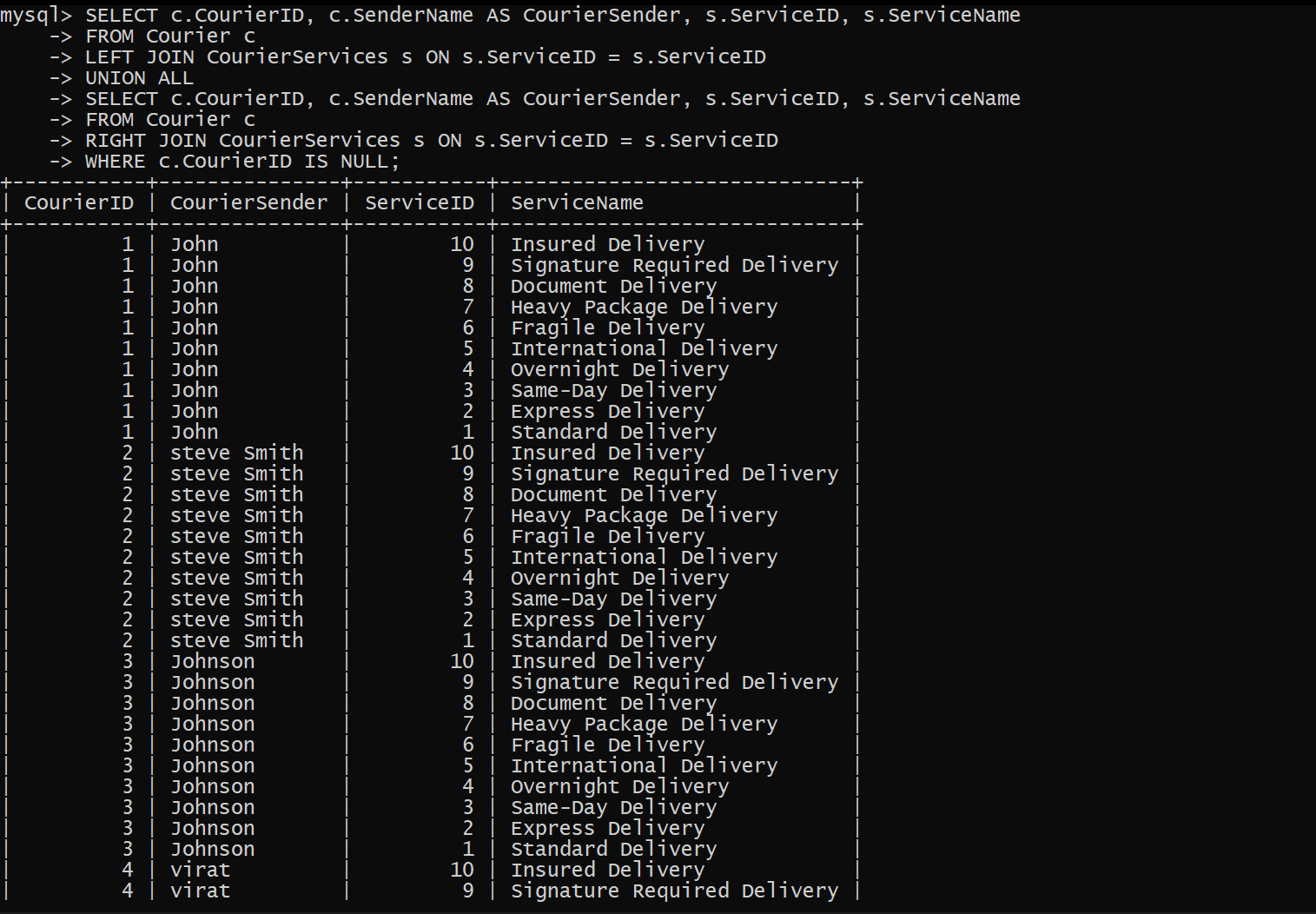
1. List Payments Within a Date Range

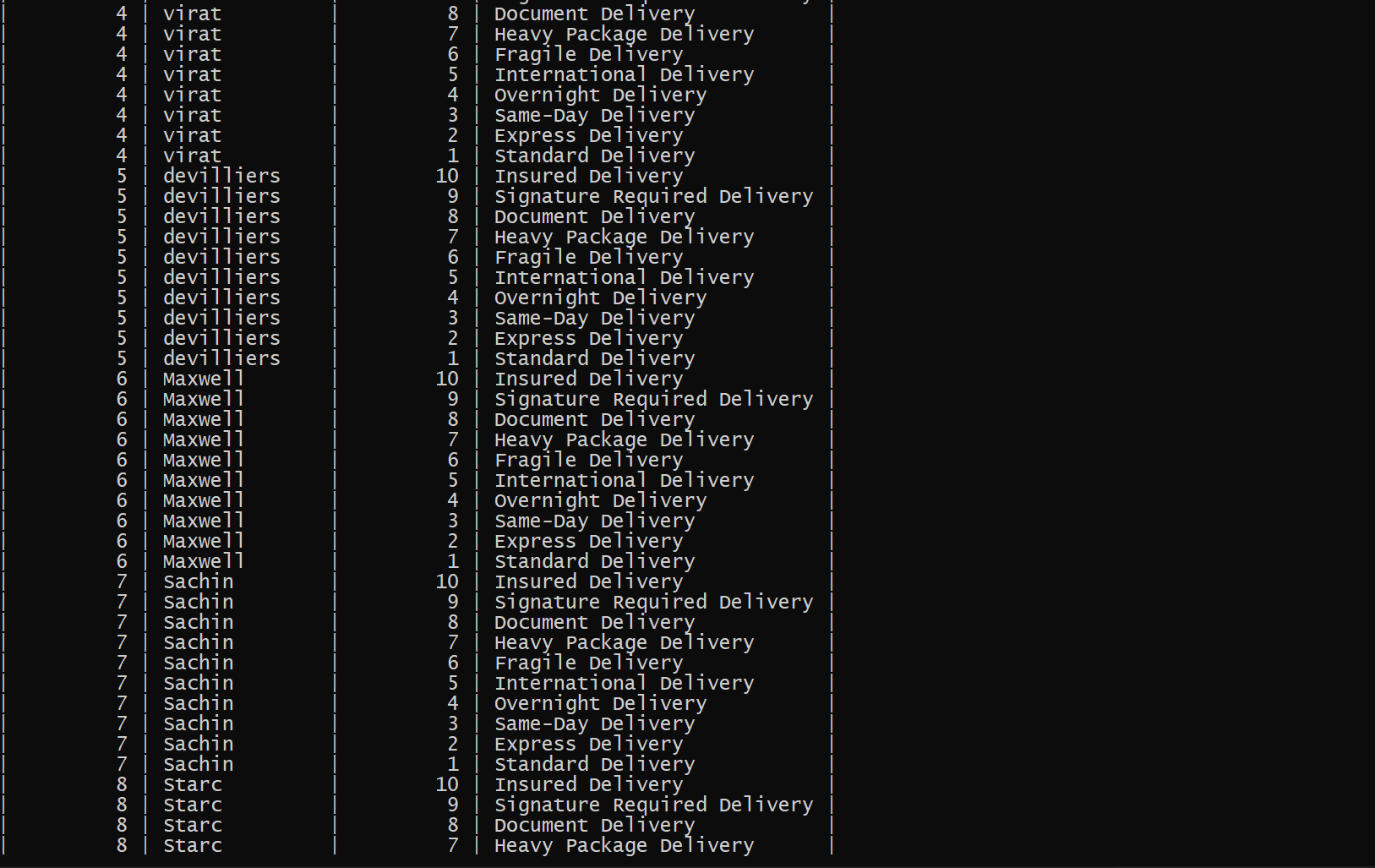


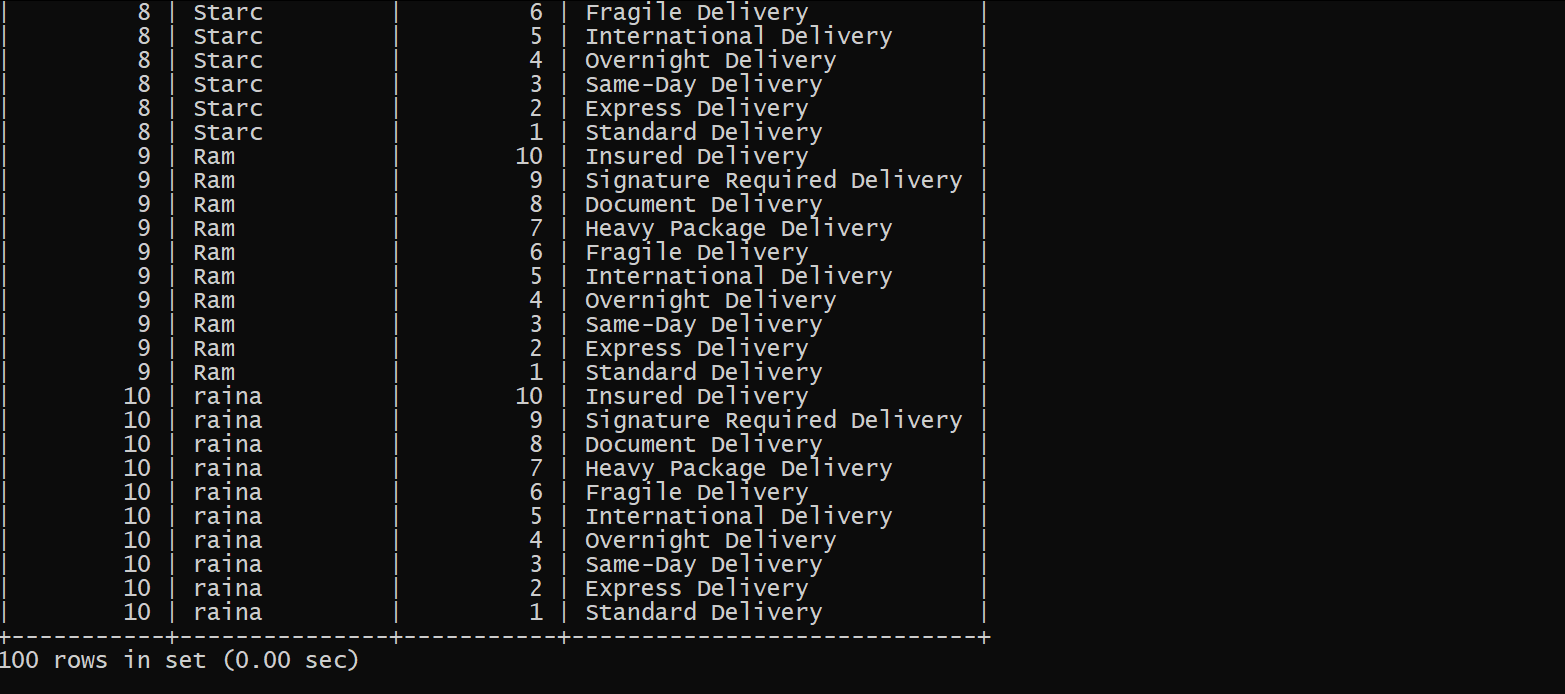
1. Retrieve a list of all users and their corresponding courier records, including cases where there are no matches on either side



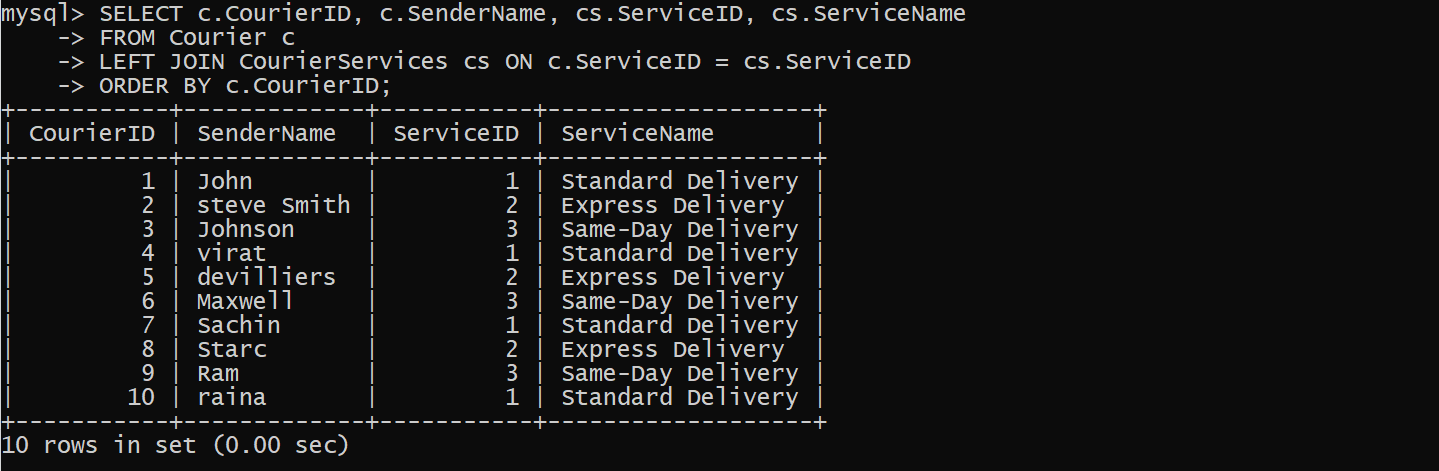
1. Retrieve a list of all couriers and their corresponding services, including cases where there are no matches on either side



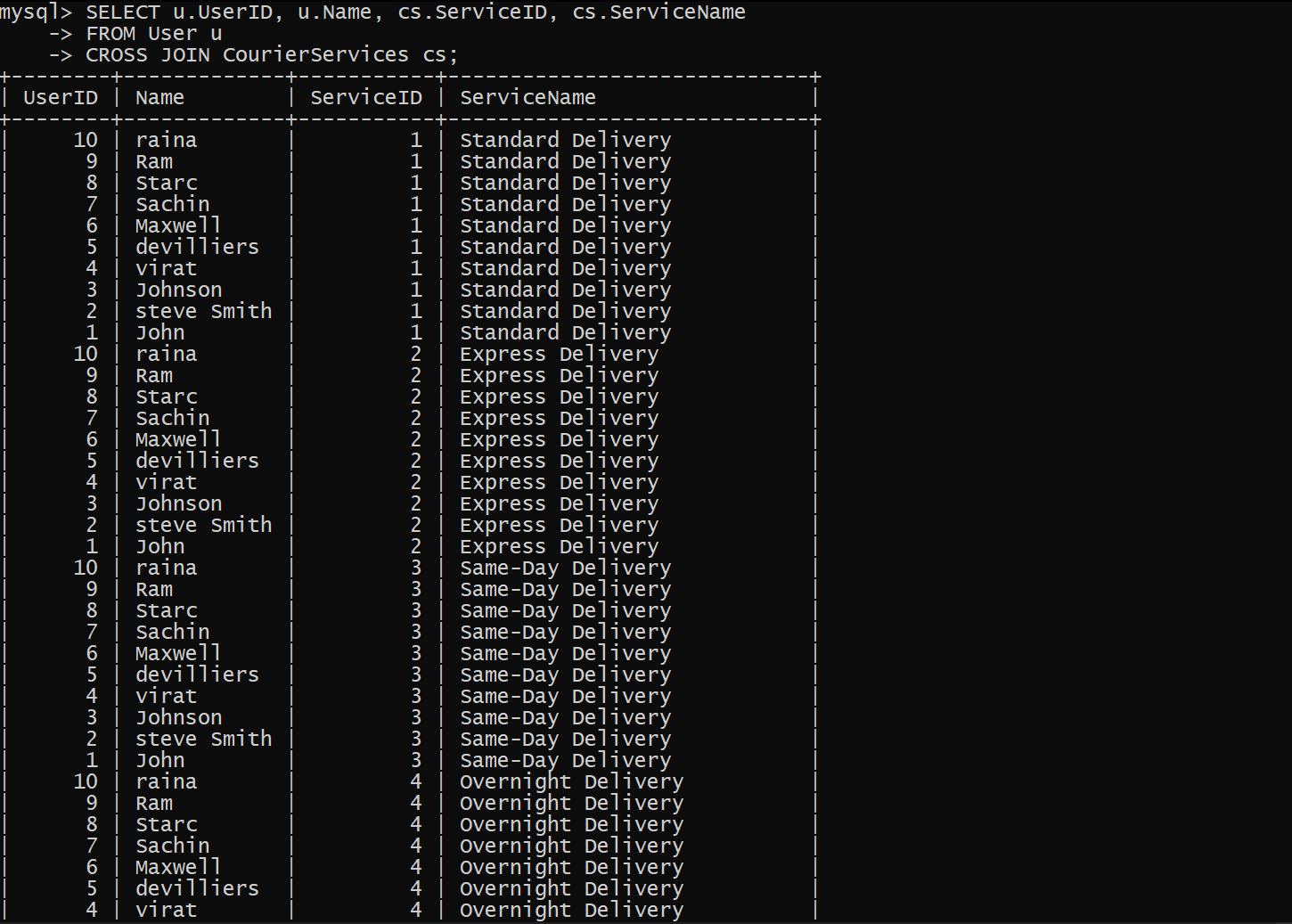


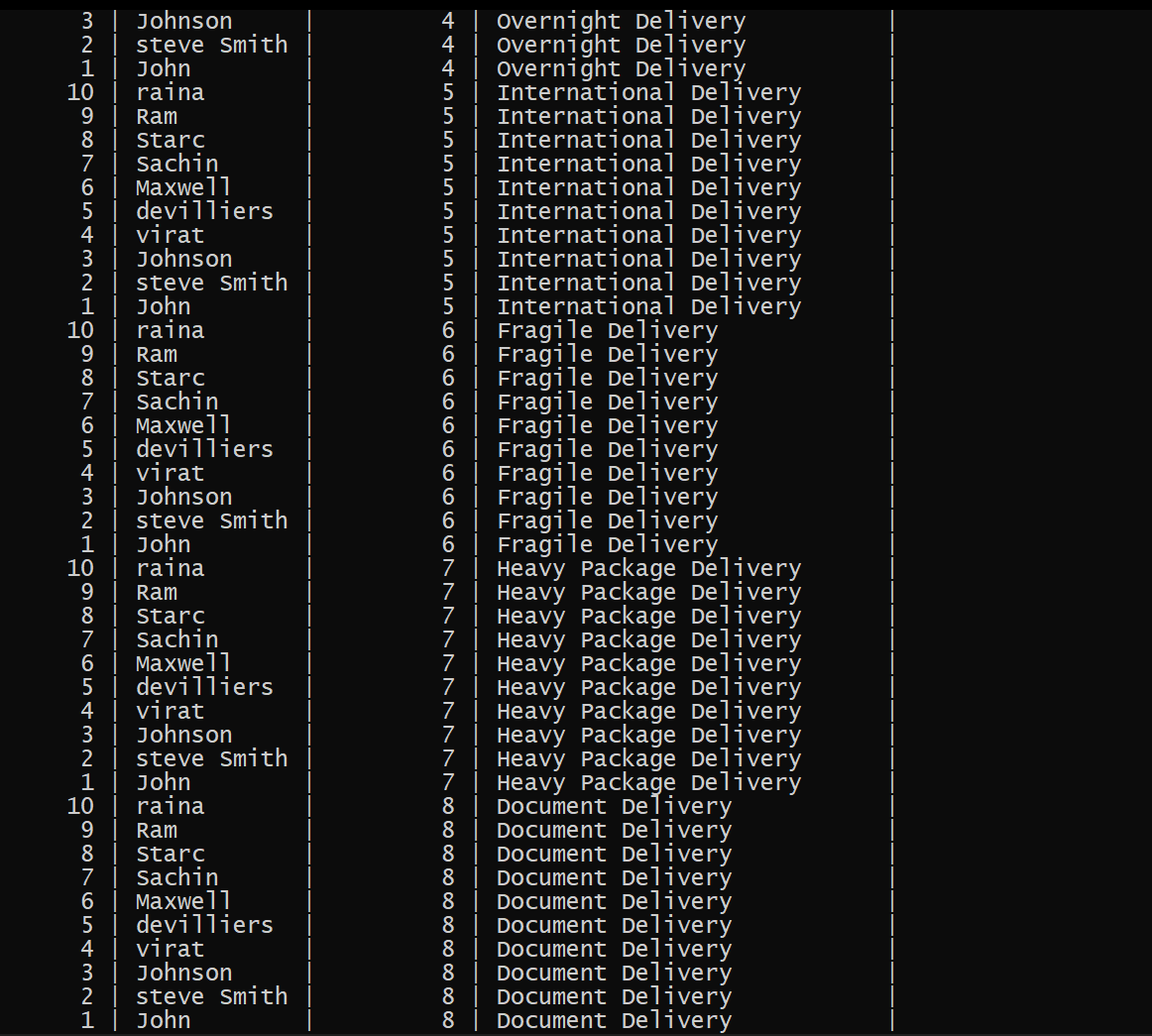


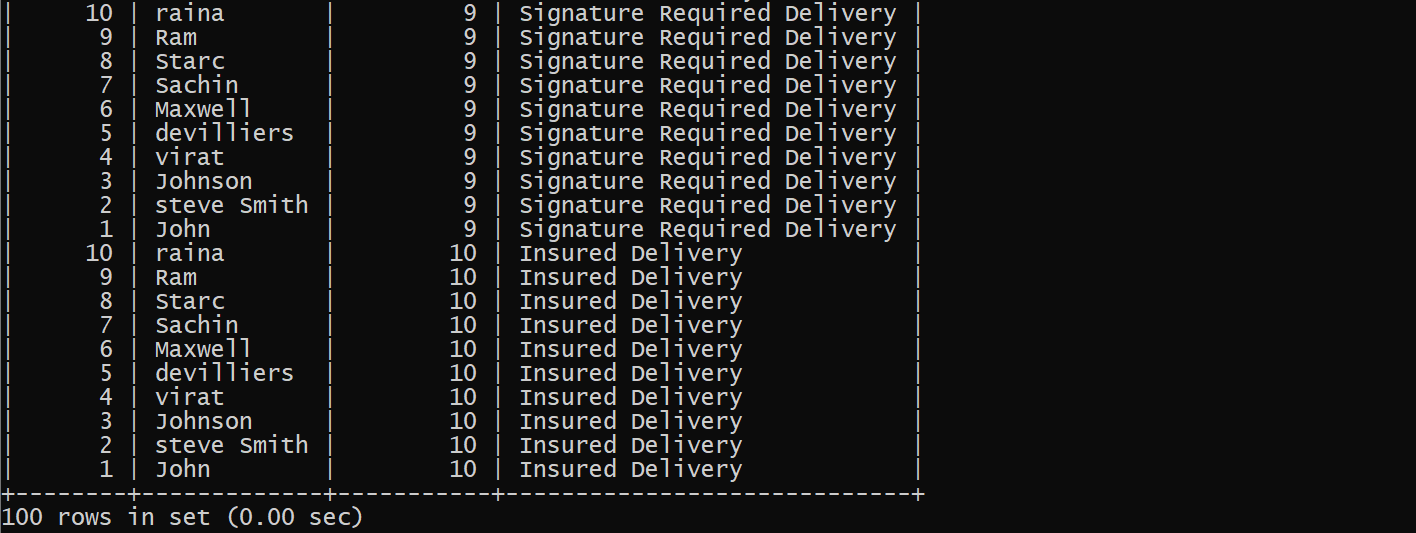
1. Retrieve a list of all employees and their corresponding payments, including cases where there are no matches on either side



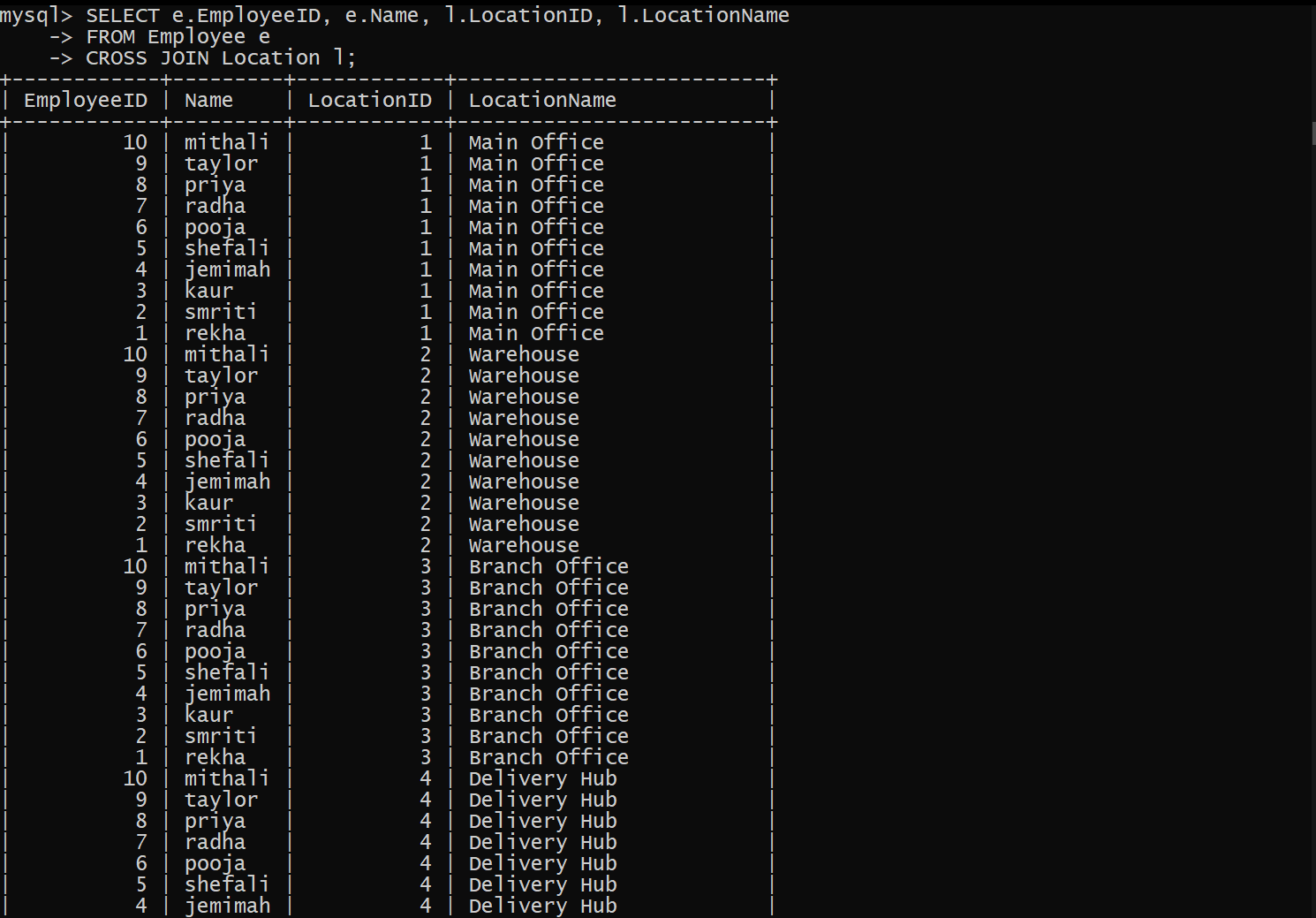
1. List all users and all courier services, showing all possible combinations.

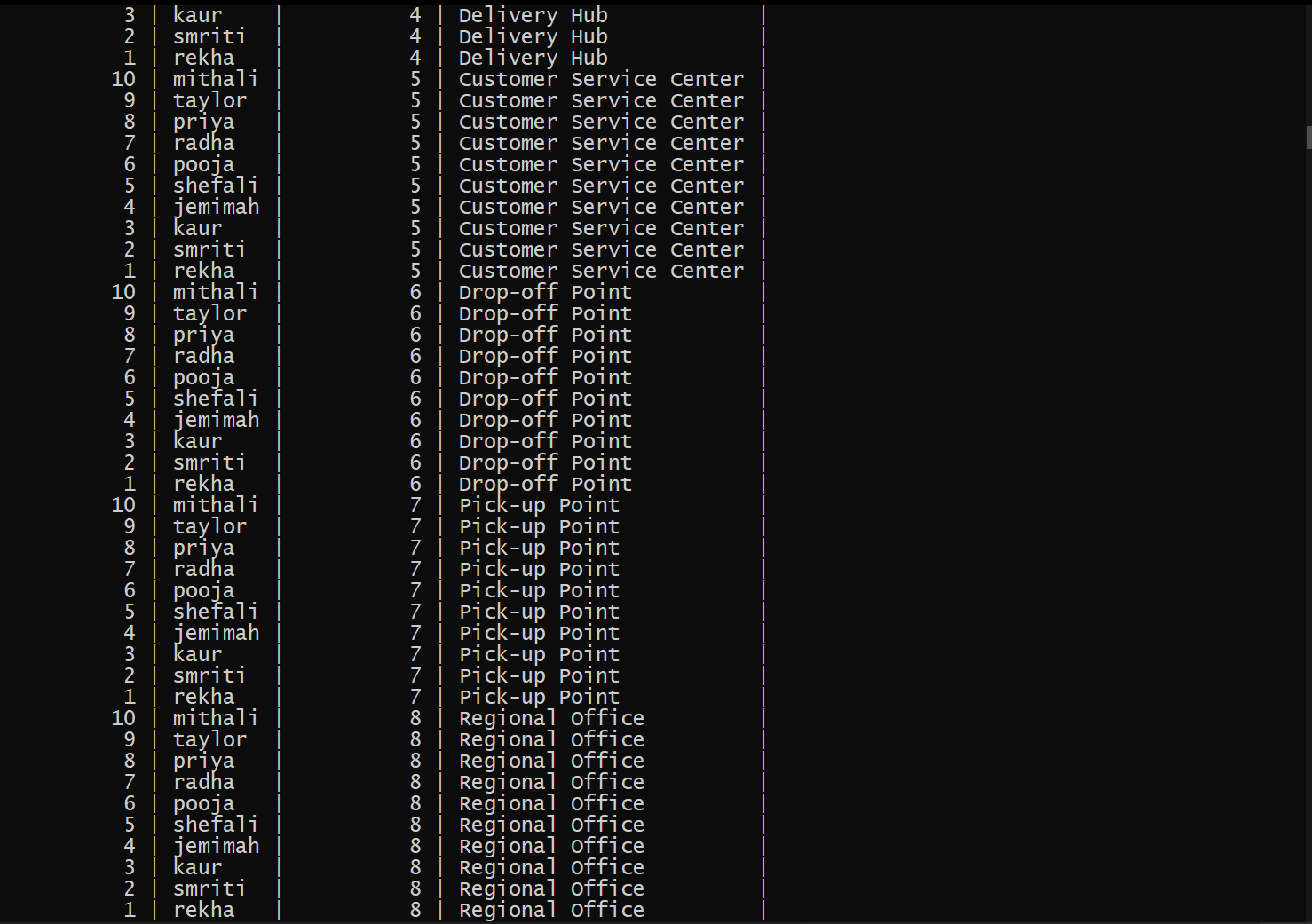


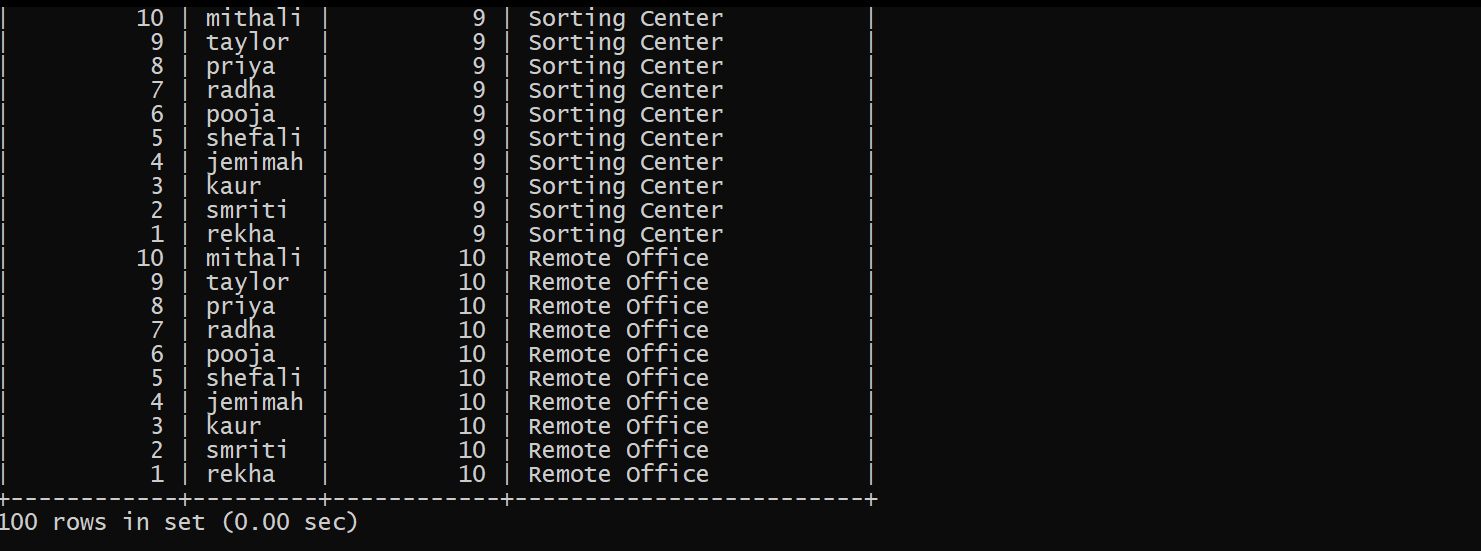




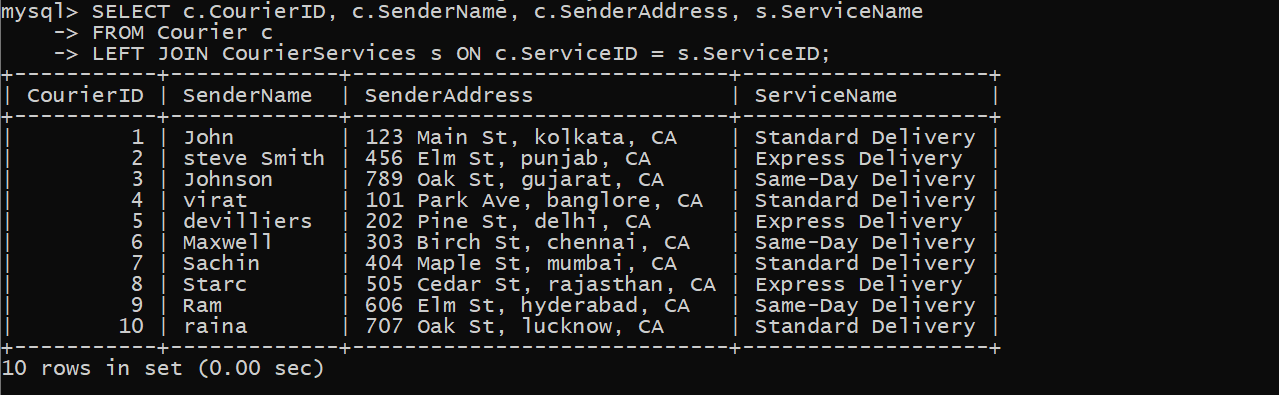
1. List all employees and all locations, showing all possible combinations:



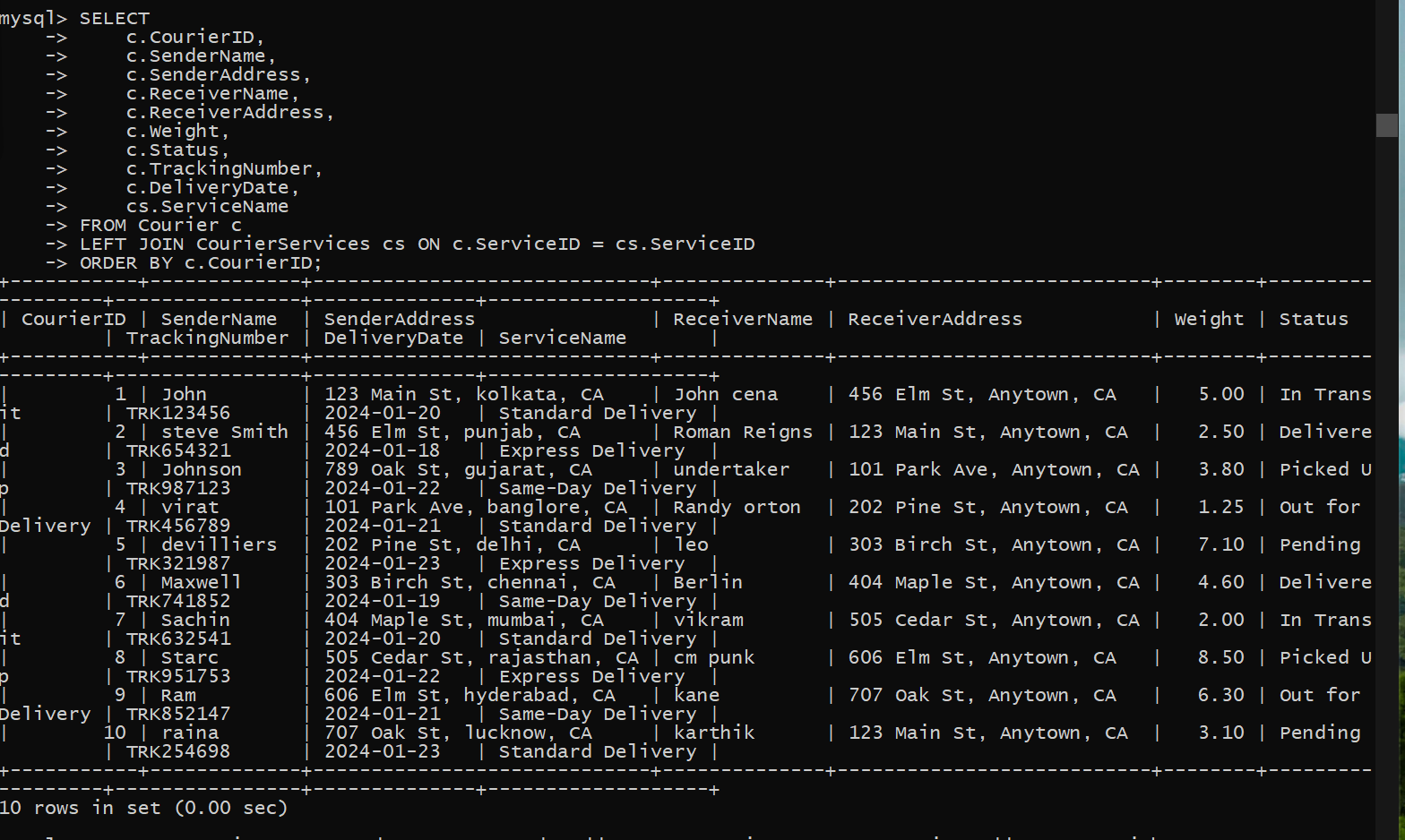




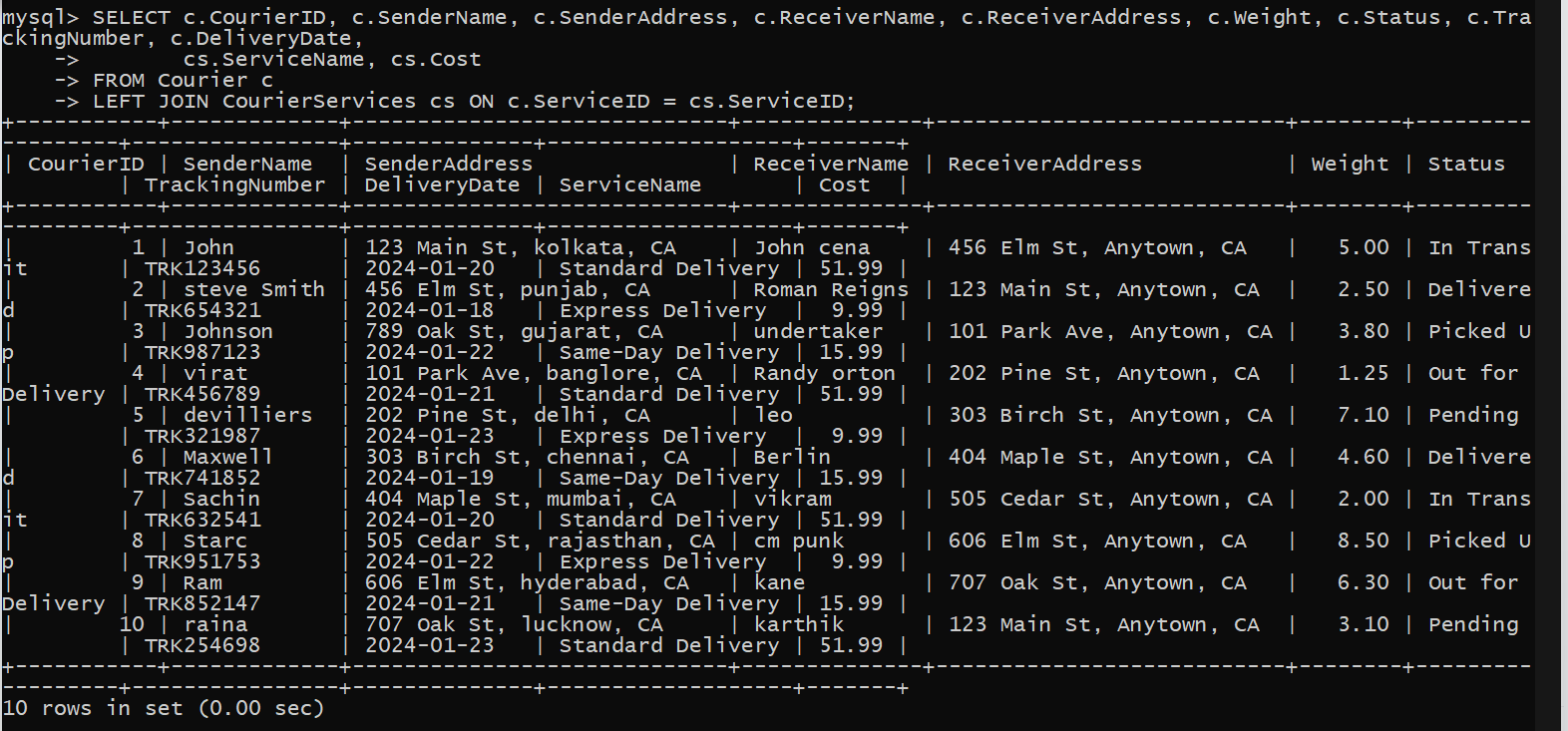
1. Retrieve a list of couriers and their corresponding sender information (if available)



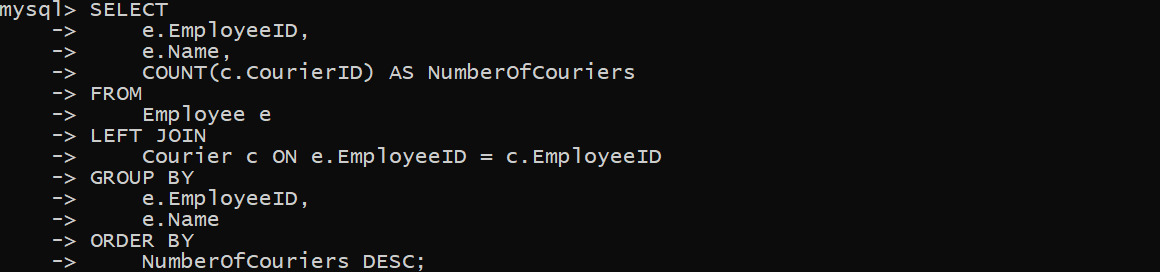
1. Retrieve a list of couriers and their corresponding receiver information (if available):



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

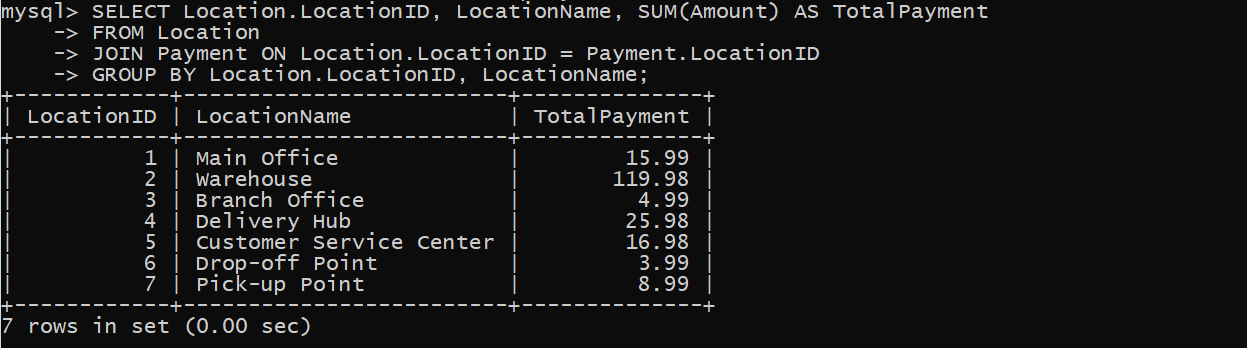


1. Retrieve a list of employees and the number of couriers assigned to each employee:

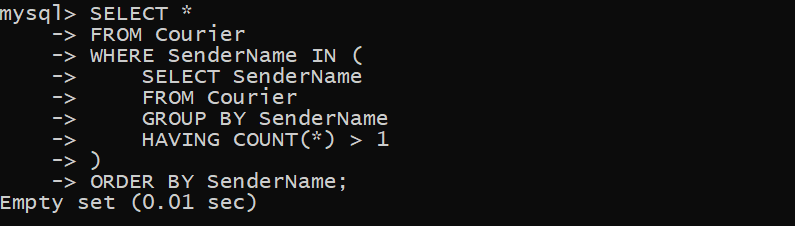




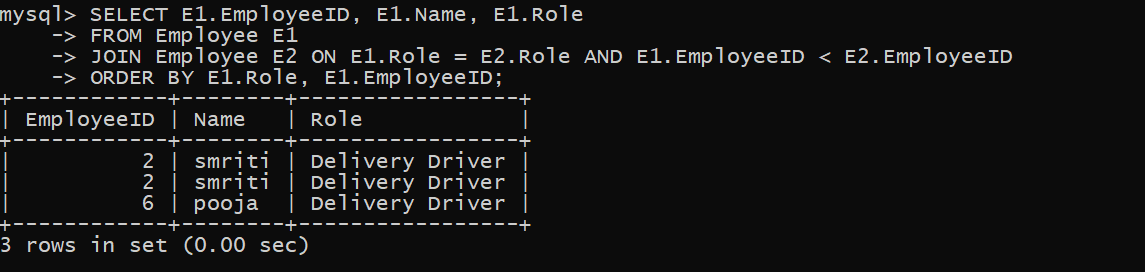
1. Retrieve a list of locations and the total payment amount received at each location:



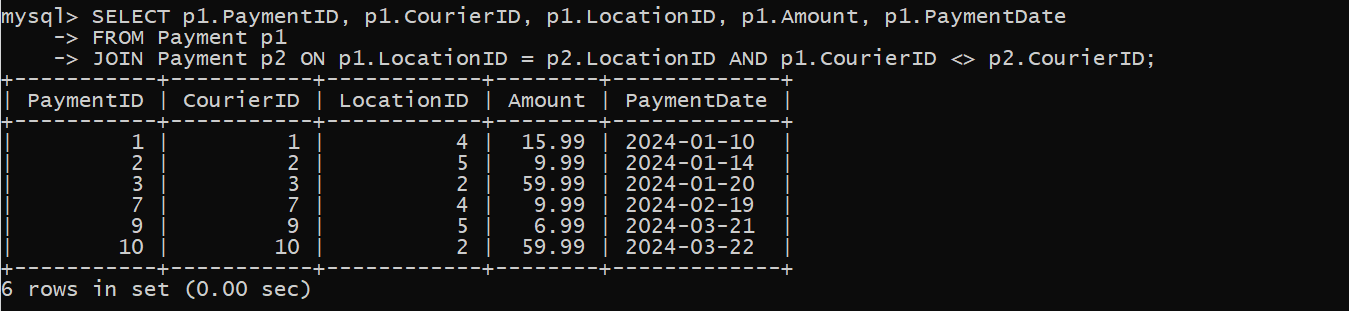
1. Retrieve all couriers sent by the same sender (based on SenderName).



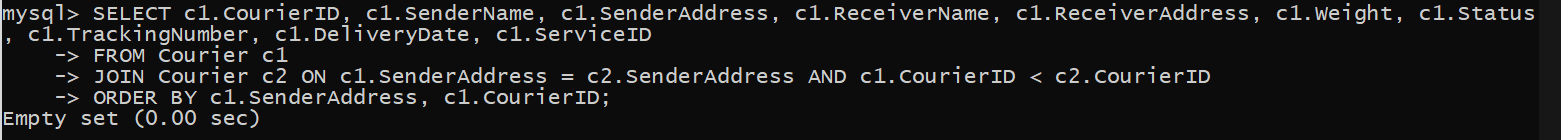
1. List all employees who share the same role.



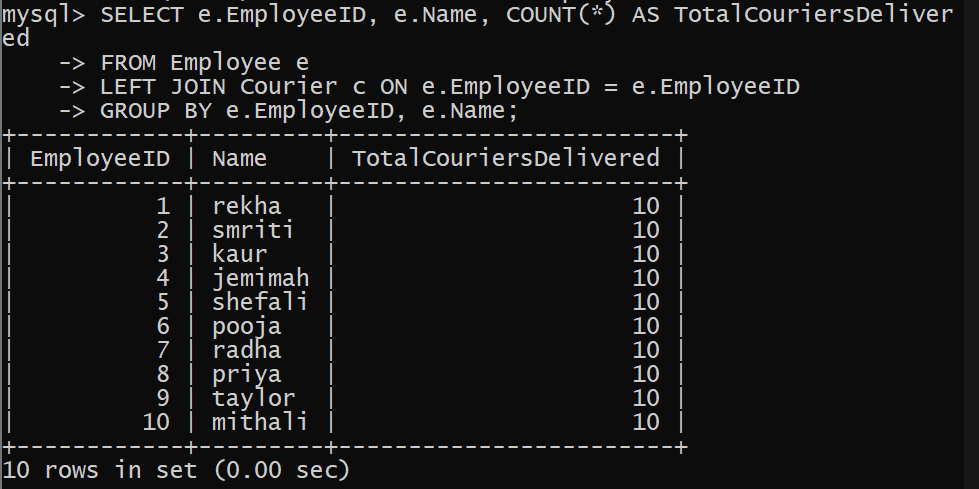
1. Retrieve all payments made for couriers sent from the same location.



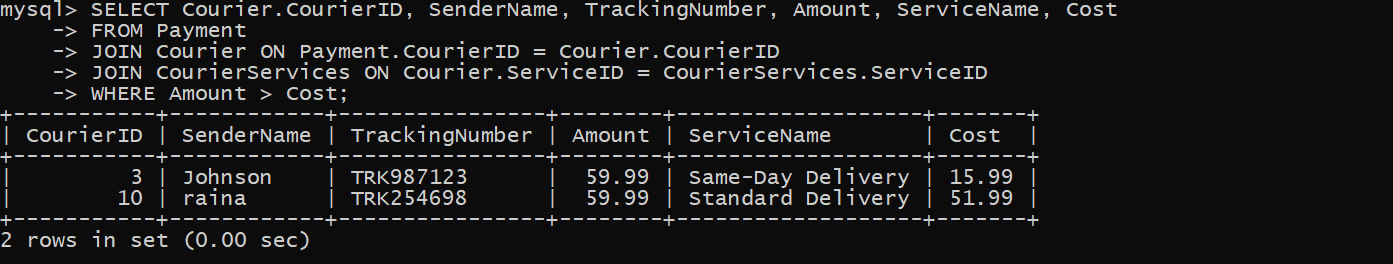
1. Retrieve all couriers sent from the same location (based on SenderAddress).



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



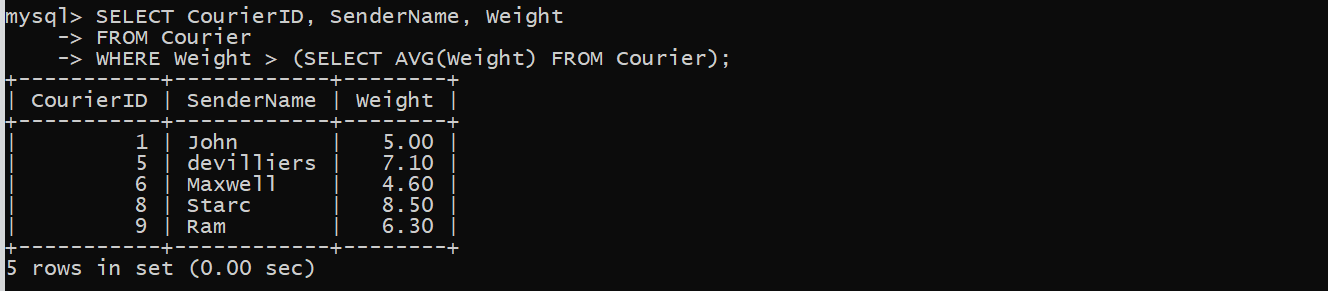
1. Find couriers that were paid an amount greater than the cost of their respective courier services



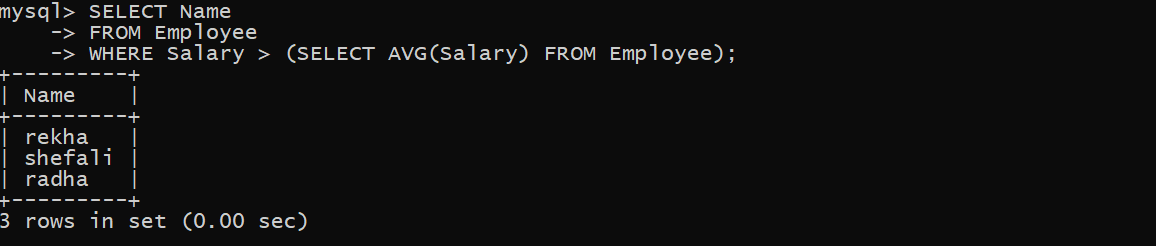
**Scope: Inner Queries**,

Non Equi Joins, Equi joins,Exist,Any,All

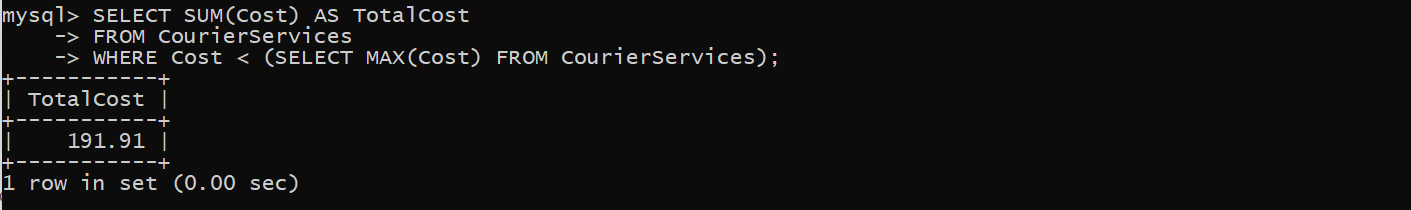
1. Find couriers that have a weight greater than the average weight of all couriers:



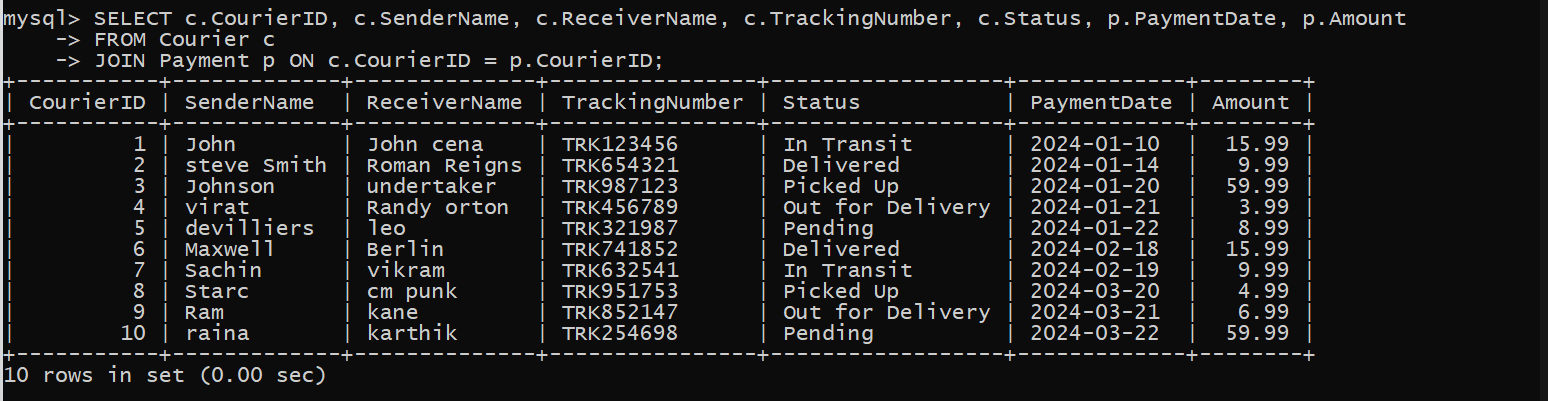
1. Find the names of all employees who have a salary greater than the average salary:



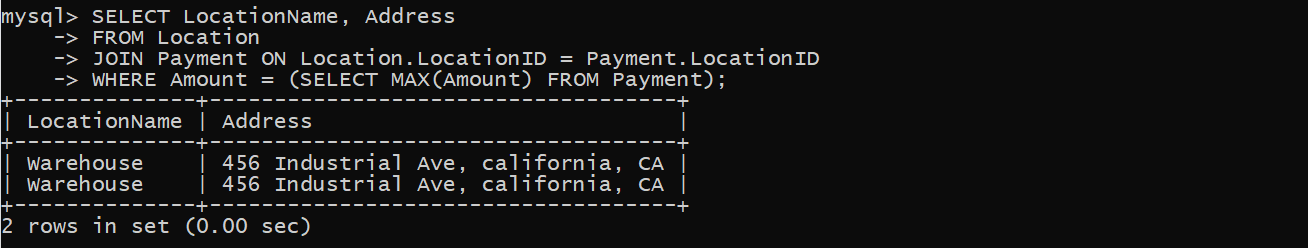
1. Find the total cost of all courier services where the cost is less than the maximum cost



1. Find all couriers that have been paid for



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



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

