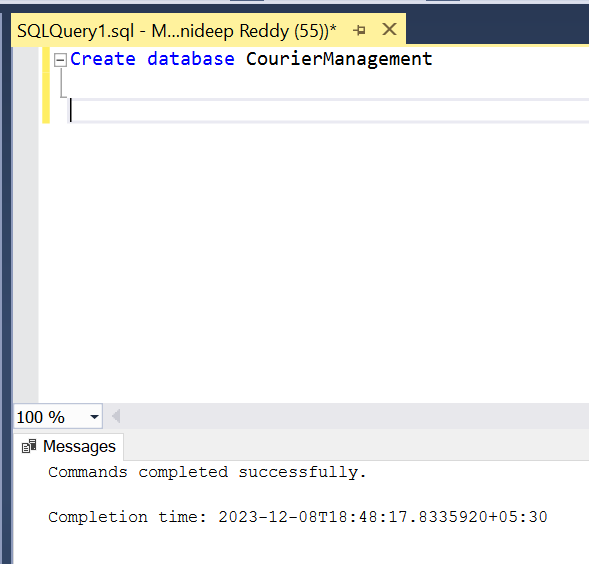
**ASSINGMENT 4:**

**TASK 1: Database Creation**

****

CREATING TABLES:

-- Create User table

CREATE TABLE Users

(

UserID INT PRIMARY KEY,

Name VARCHAR(255),

Email VARCHAR(255) UNIQUE,

Password VARCHAR(255),

ContactNumber VARCHAR(20),

Address TEXT

);

-- Create Courier table

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 CourierServices table

CREATE TABLE CourierServices (

ServiceID INT PRIMARY KEY,

ServiceName VARCHAR(100),

Cost DECIMAL(8, 2)

);

-- Create Employee table

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 Location table

CREATE TABLE Location (

LocationID INT PRIMARY KEY,

LocationName VARCHAR(100),

Address TEXT

);

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY,

CourierID INT,

LocationID INT,

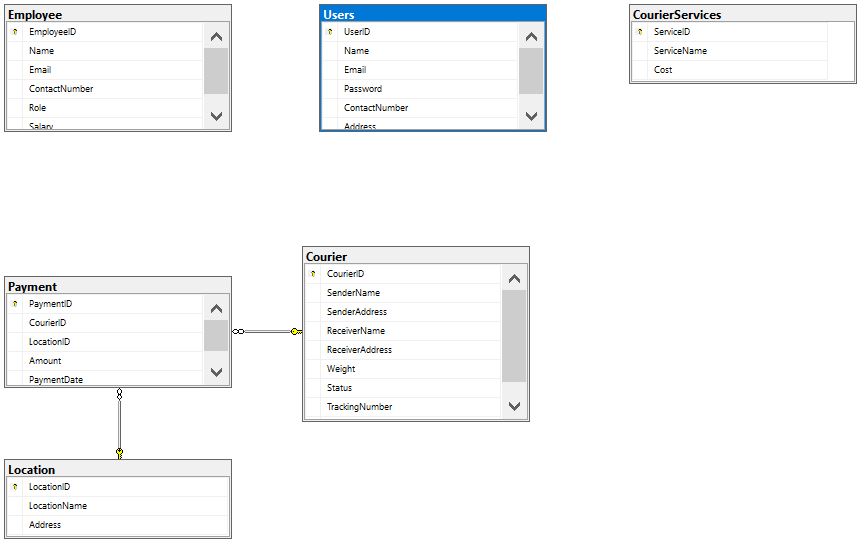
Amount DECIMAL(10, 2),

PaymentDate DATE,

FOREIGN KEY (CourierID) REFERENCES Courier(CourierID),

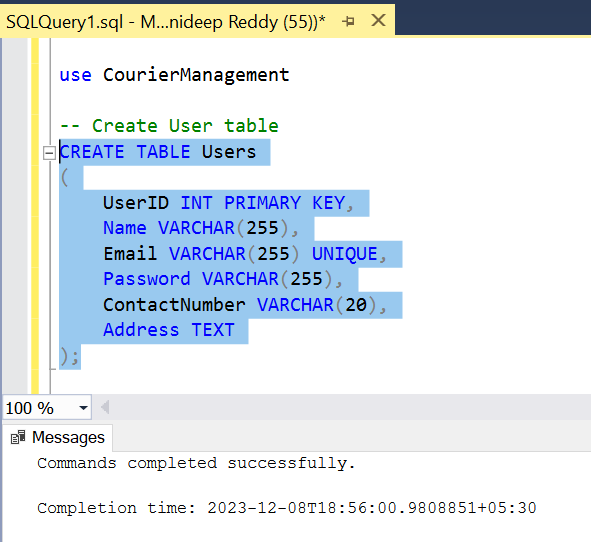
FOREIGN KEY (LocationID) REFERENCES Location(LocationID)

ENTITY REALTION DIAGRAM:

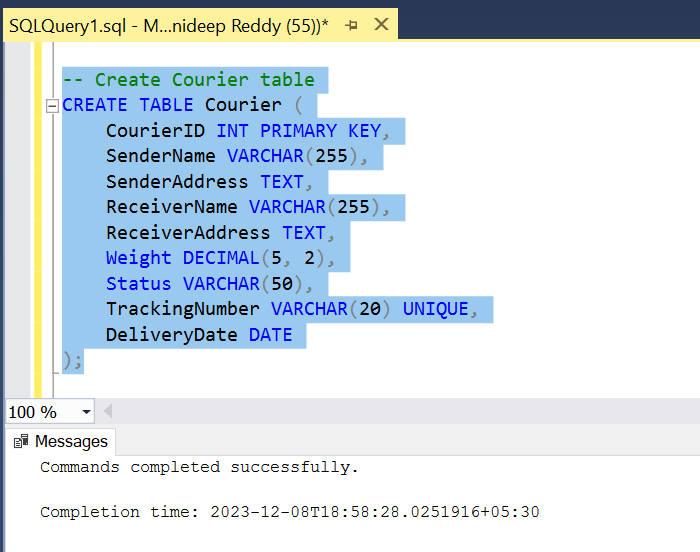


INSERTING SAMPLE RECORDS IN TO TABLES:

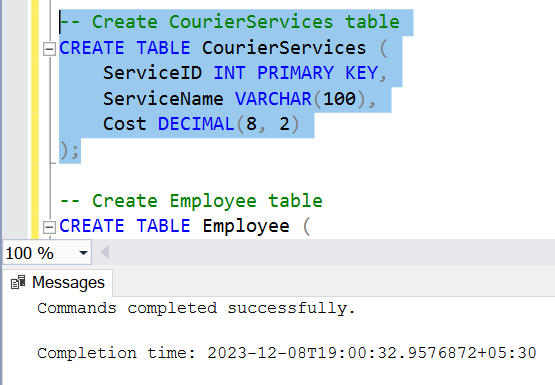
1. Users Table:

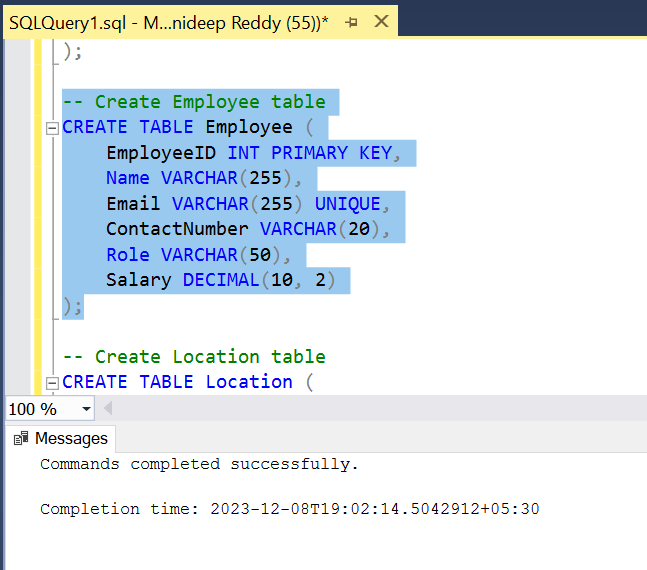


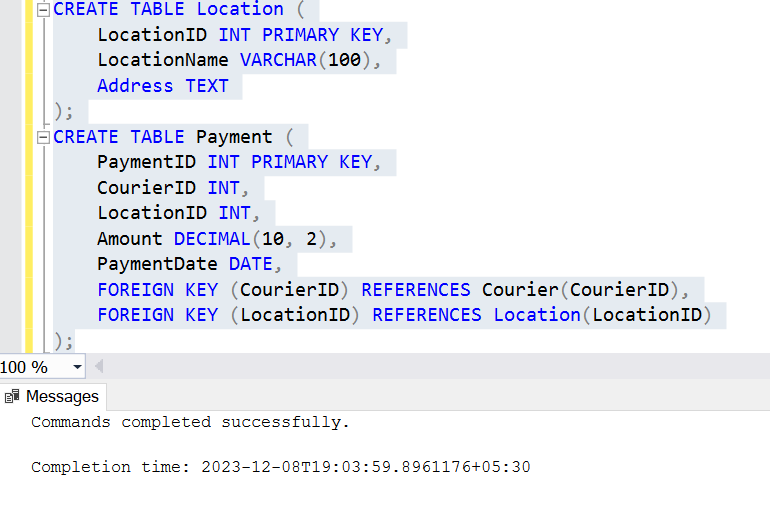
1. Courier Table:



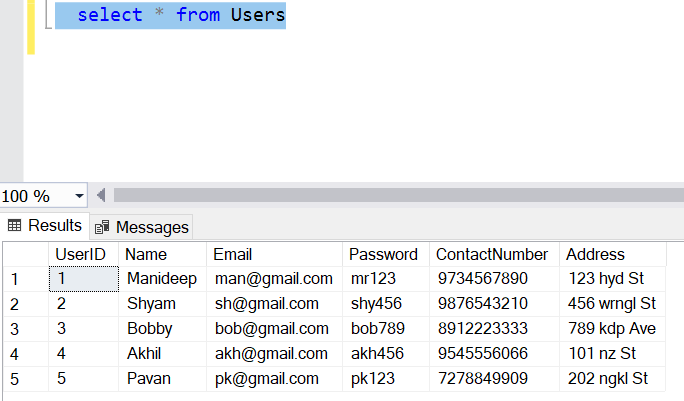
1. Courier services Table:

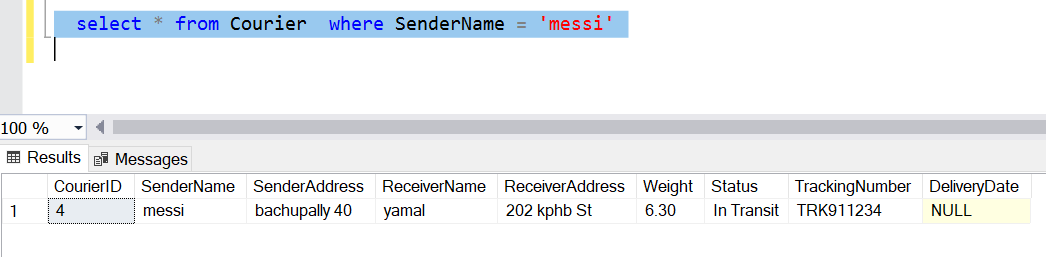


1. Employee Table:
2. Location Table and Payment Table:

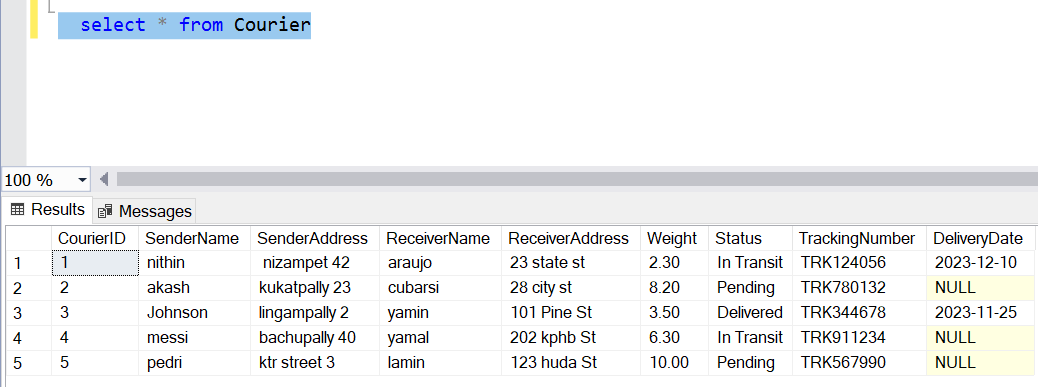


**TASK 2**: **Select, Where**

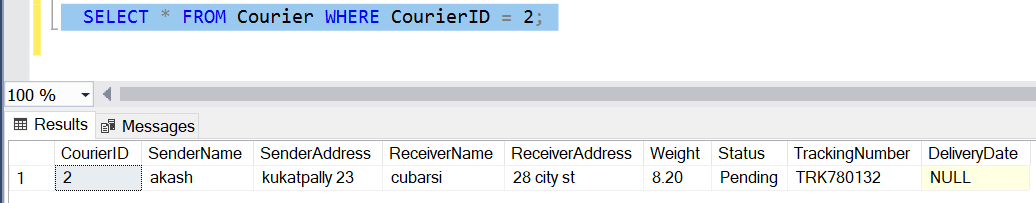
1. List all customers:
2. List all orders for a specific customer:



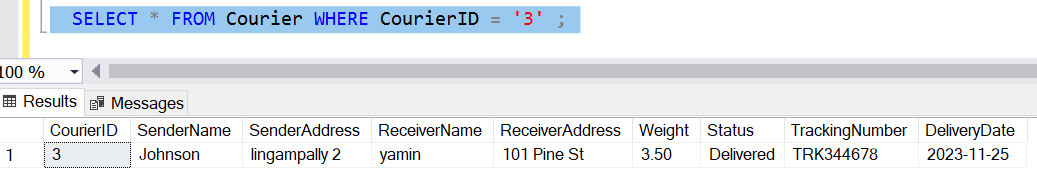
1. List all courier:



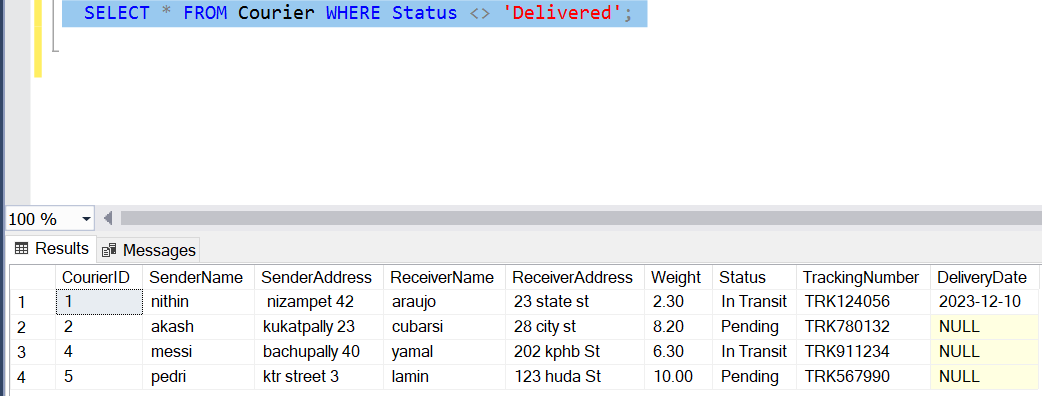
1. List all packages for a specific order:



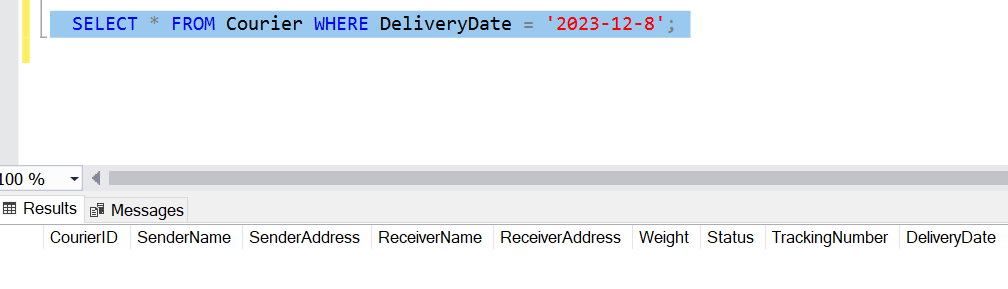
1. List all deliveries of specific courier:



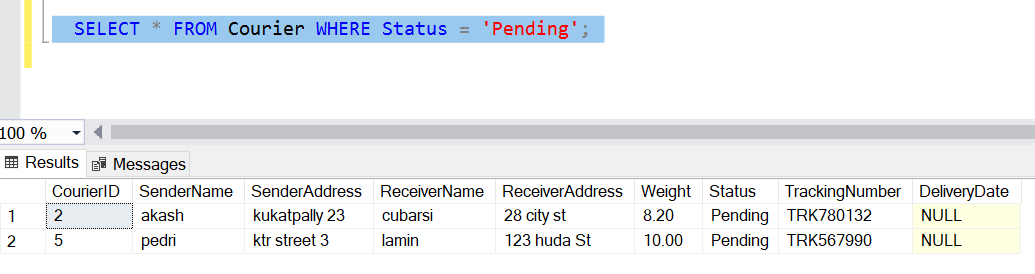
1. List all undelivered packages:



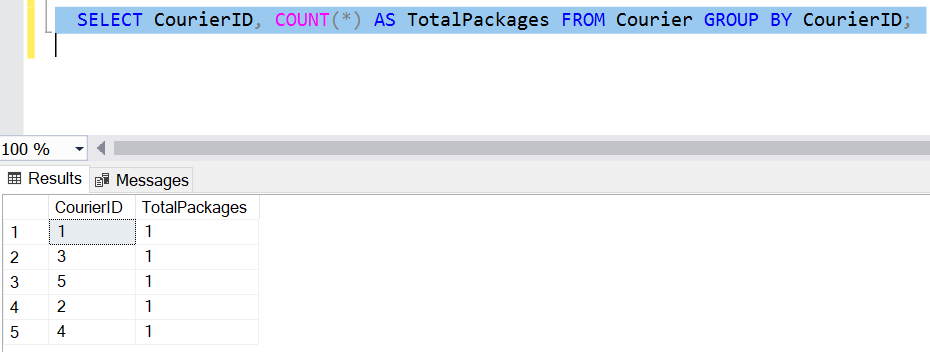
1. List all packages that are scheduled for delivery today:



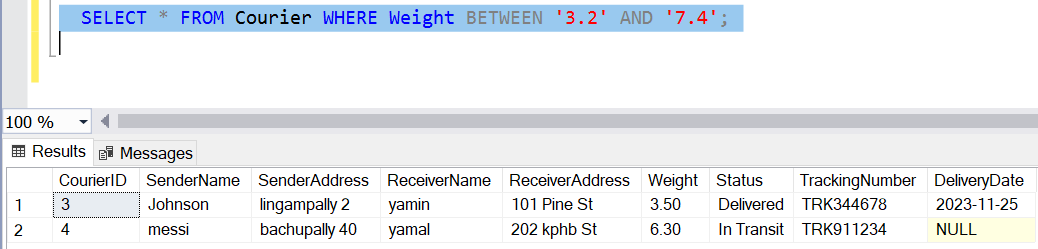
1. List all packages with a specific status:



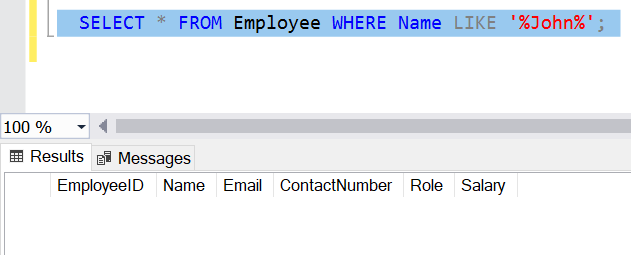
1. Calculate total number of packages for each other:



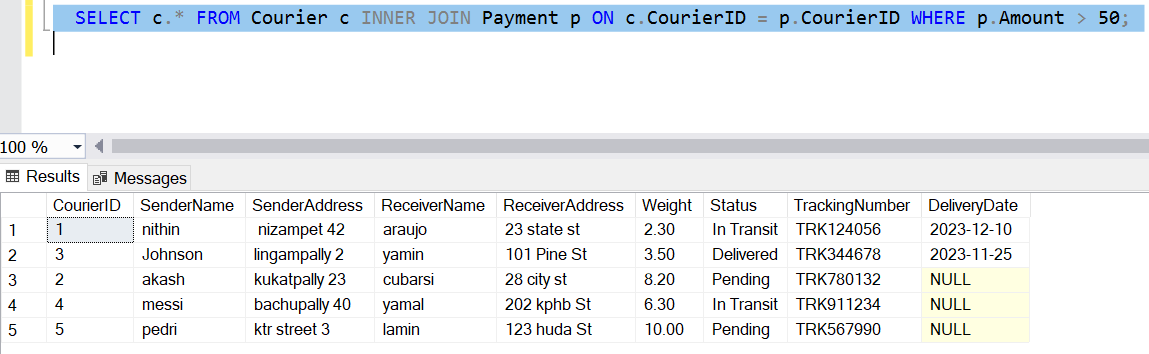
1. List all packages with a specific weight range:



1. Retrieve employees whose names contain ‘John’:

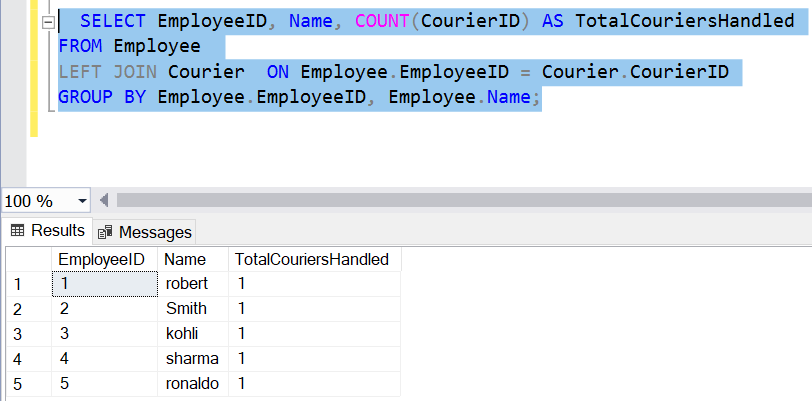


1. Retrieve all courier records with payments greater that 50$:

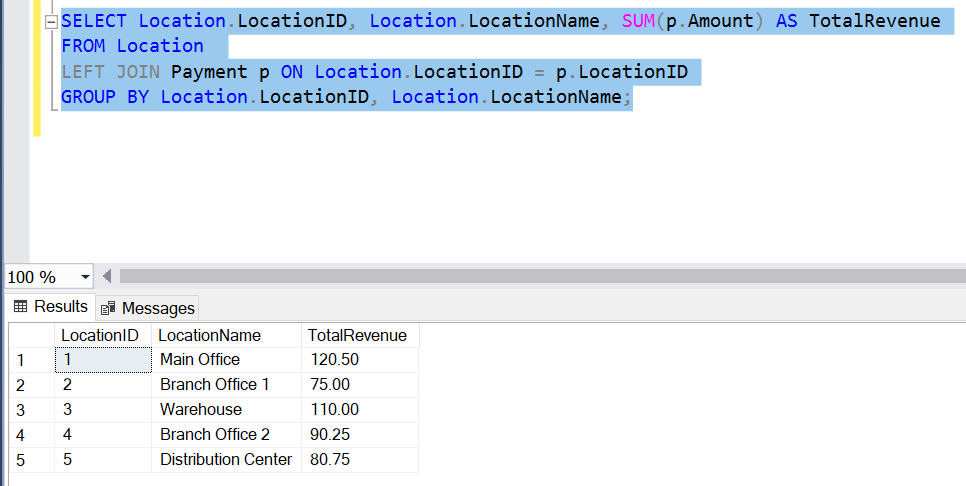


**TASK 3**: **Group by, Aggregate, Having, Order by, Where**

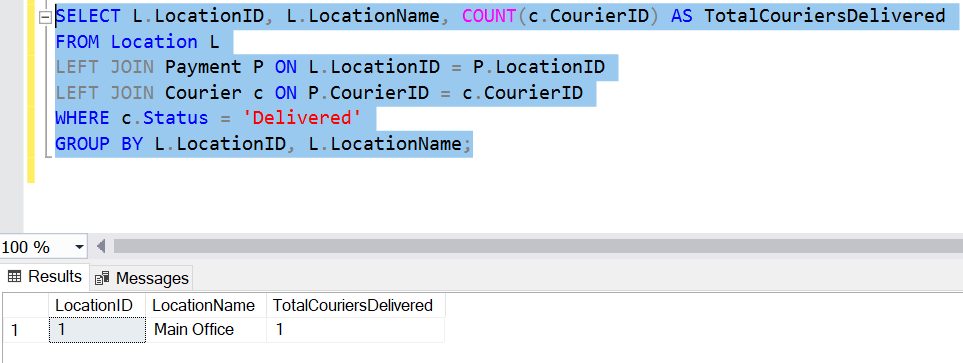
1. Find total number of couriers handled by each employee.



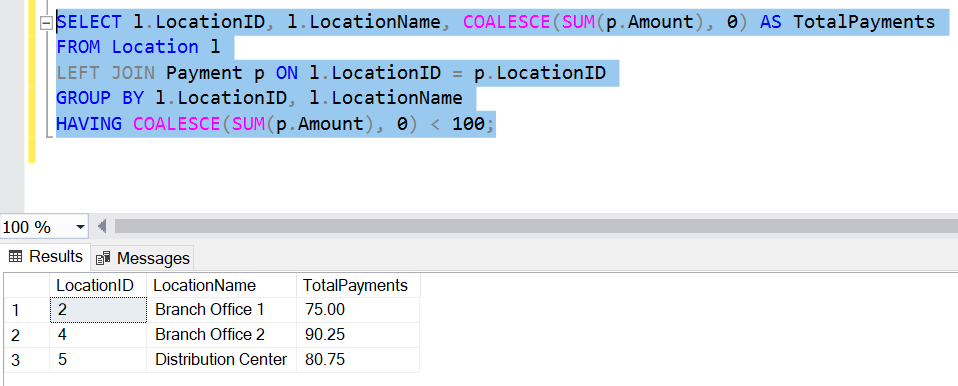
1. Calculate total revenue generated by each location.



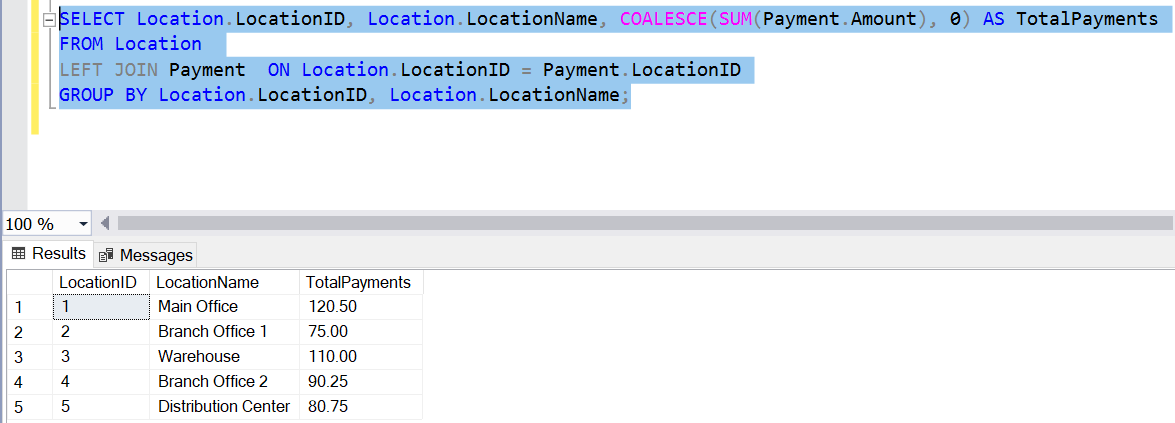
1. Find total number of couriers delivered to each location.



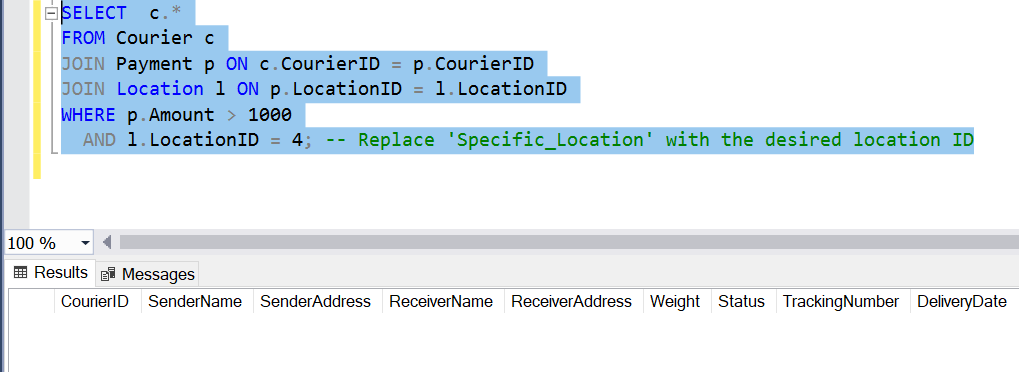
1. Find locations with total payments less than certain amount.



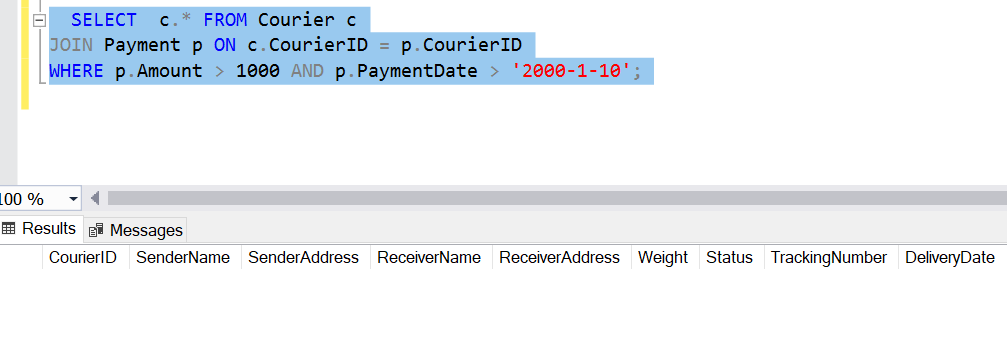
1. Calculate total payments per location.



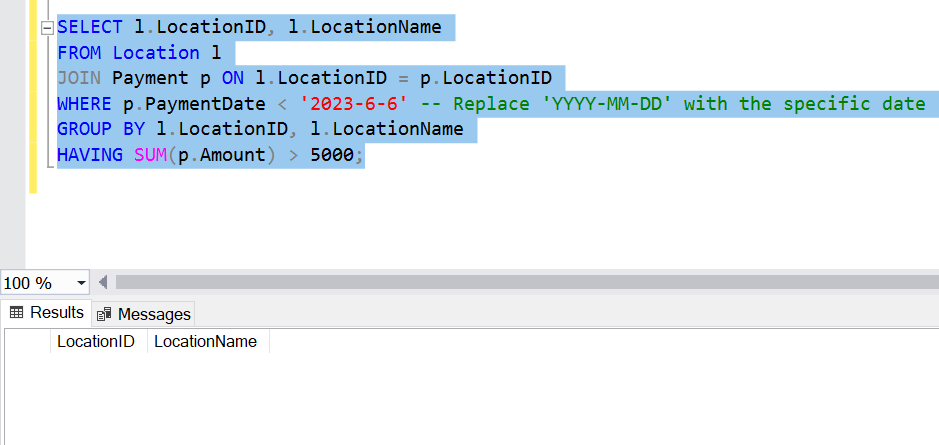
1. Retrieve courier who have received payments totaling more than 1000$ in a specific location.



1. Retrieve courier who have received payments totaling more than 1000$ after a certain date.

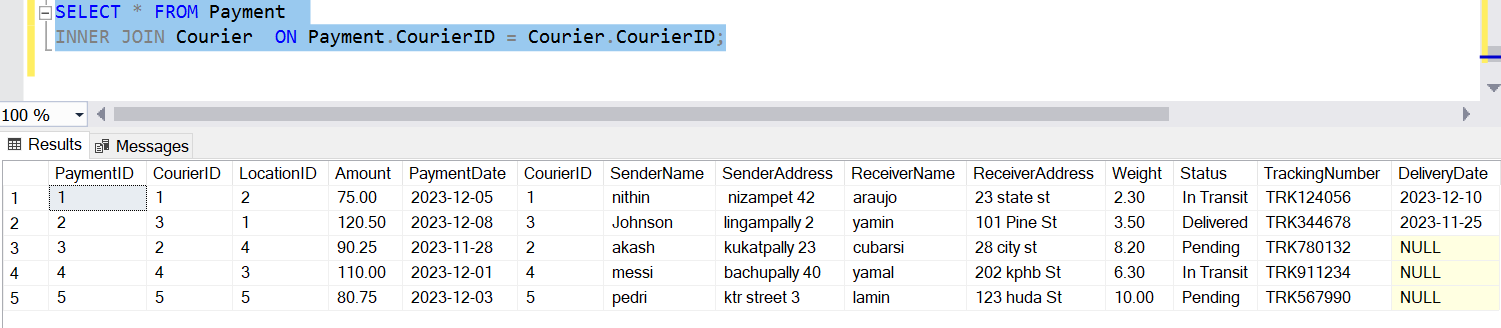


1. Retrieve locations where the total amount received is more than 5000$ before a certain date.

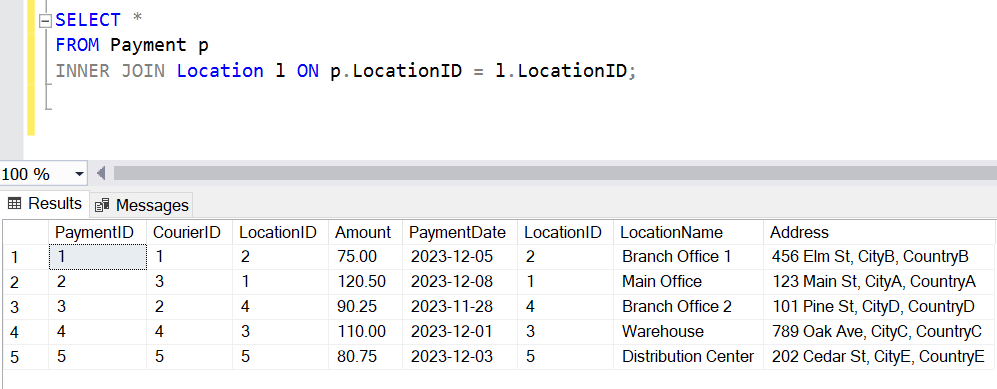


**TASK 4: Inner join, Full outer join, Cross join, Left outer join, Right outer join**

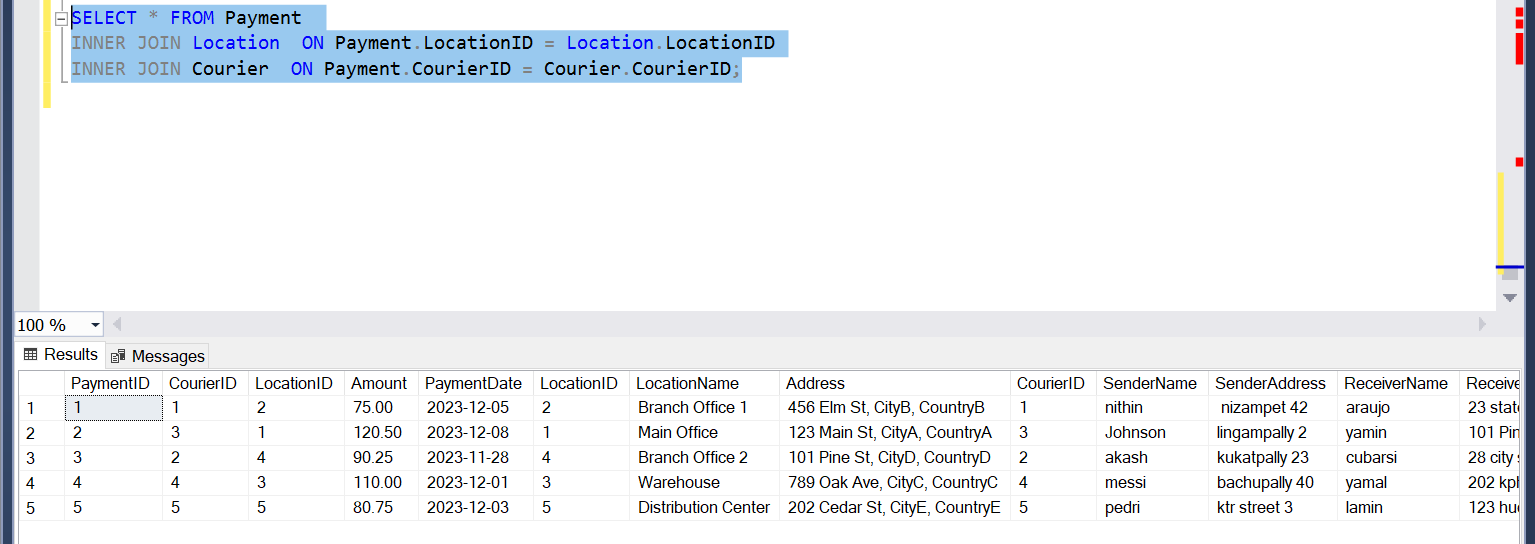
1. Retrieve Payments with Courier Information.



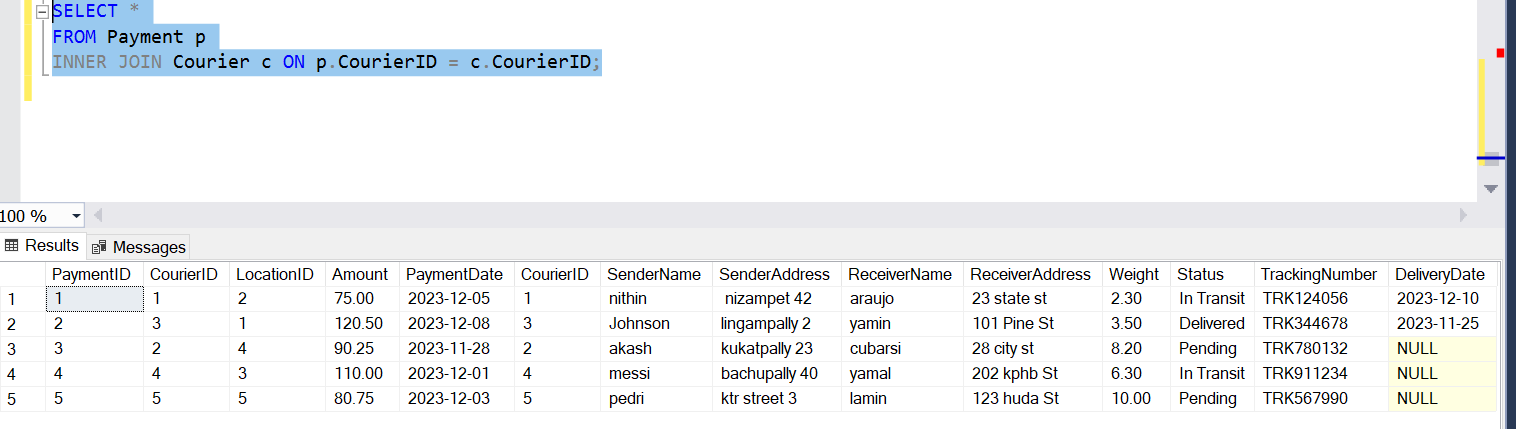
1. Retrieve payments with location information



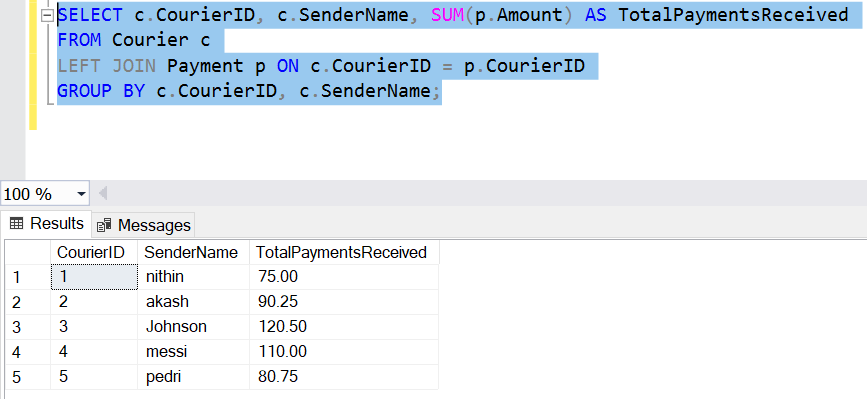
1. Retrieve payments with courier and location information.



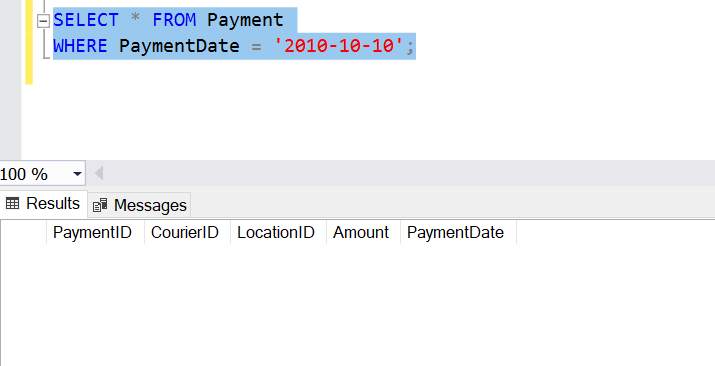
1. List all payments with courier details.



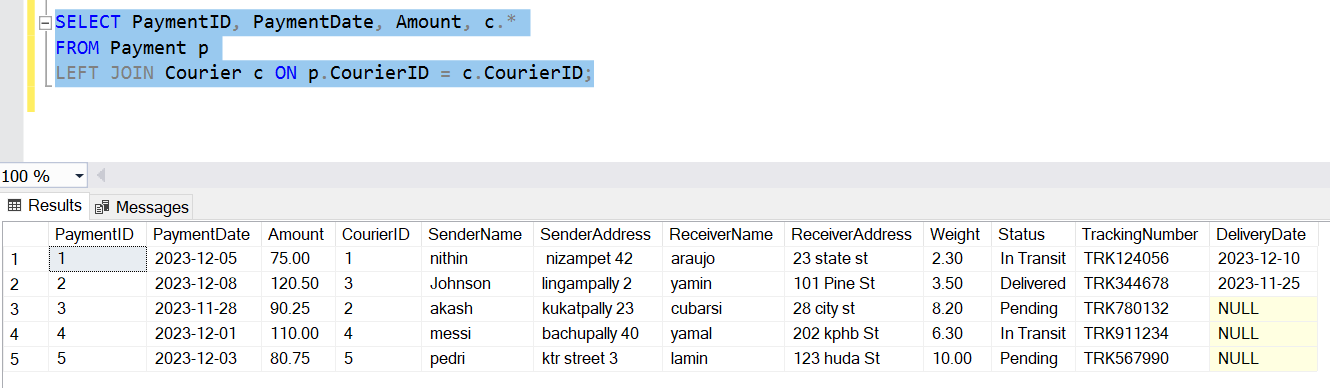
1. Total payments received for each courier



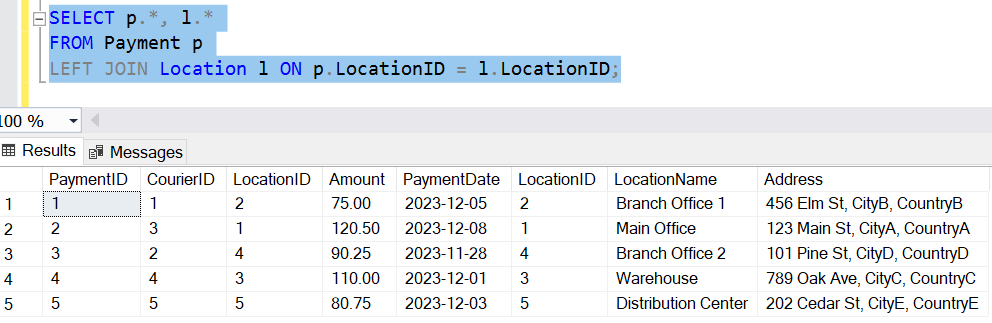
1. List payments made on specific date.



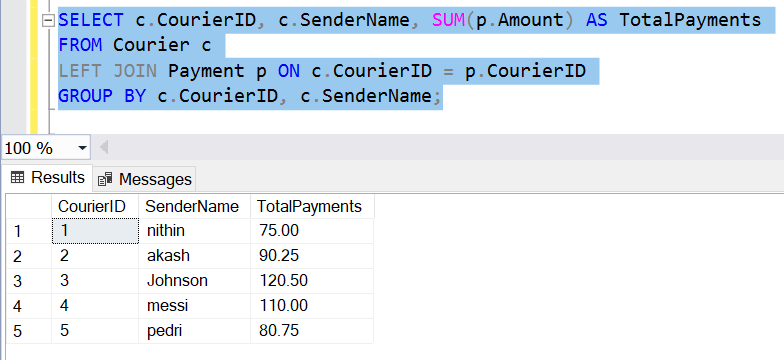
1. Get courier information for each payment.



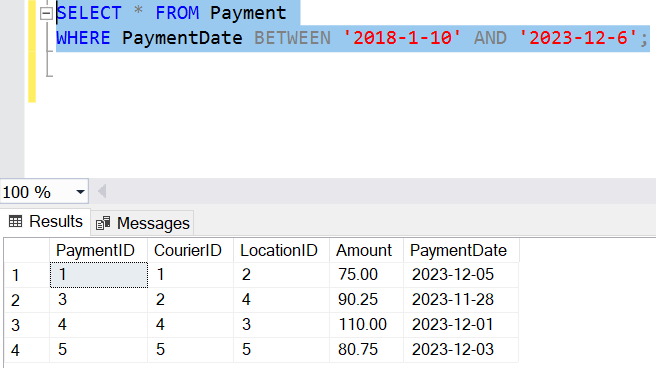
1. Get payment details with location.



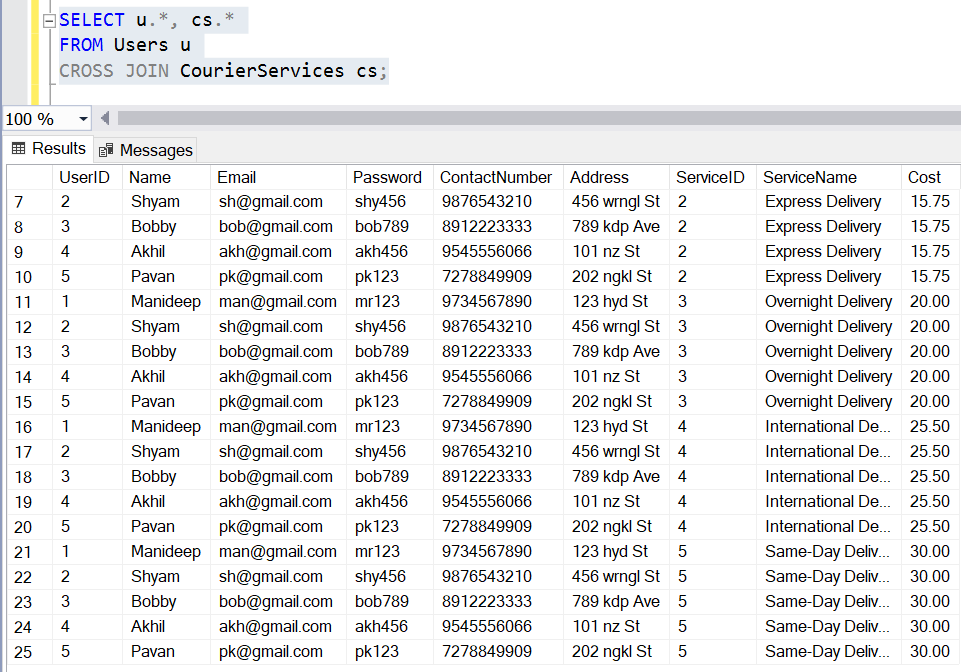
1. Calculating total payment for each courier.



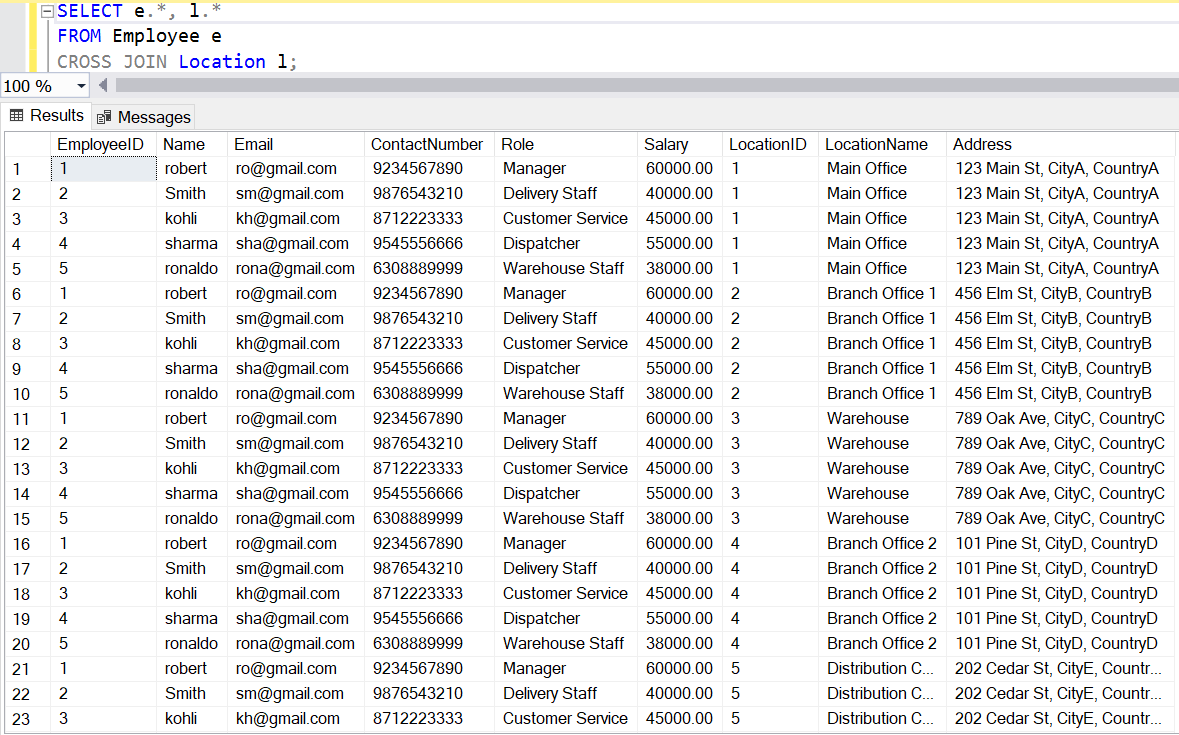
1. List payments within a date range.



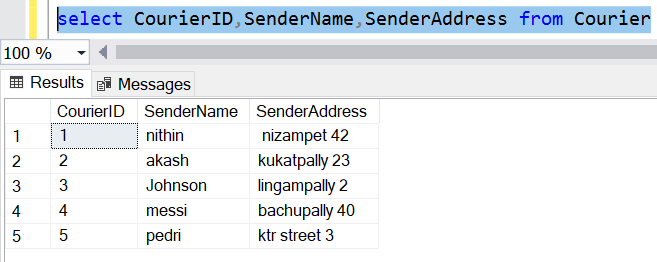
1. List all users and all courier services, showing all possible combinations.



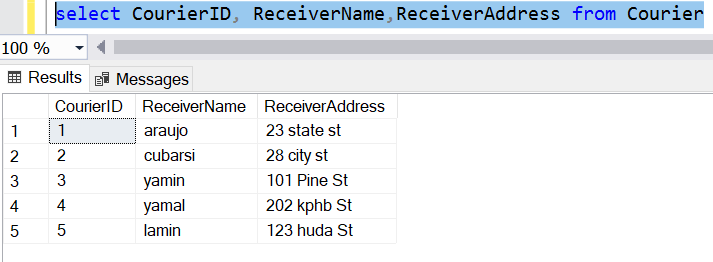
1. List all employees and all locations, showing all possible combinations.



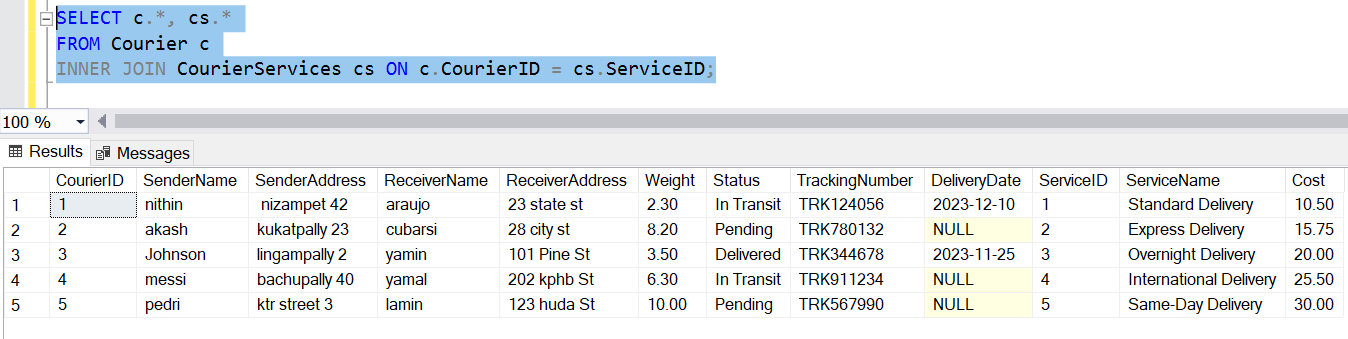
1. Retrieve a list of couriers and their corresponding sender information (if available)



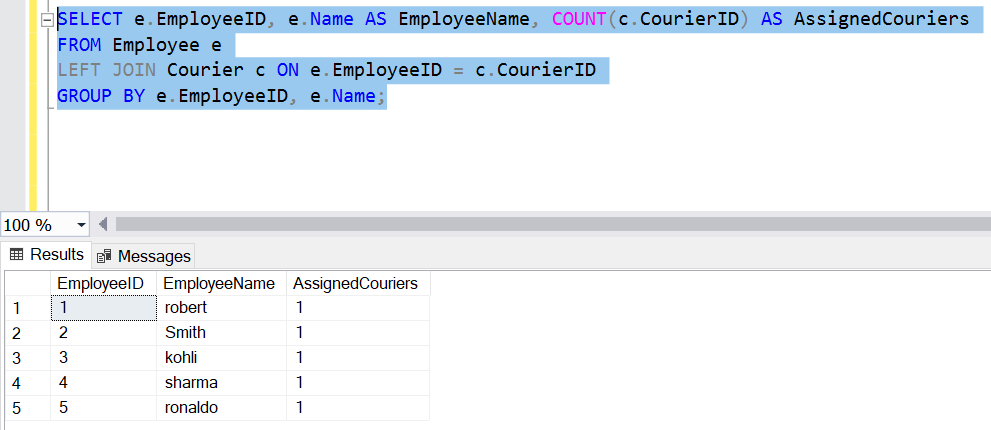
1. Retrieve a list of couriers and their corresponding receiver information (if available)



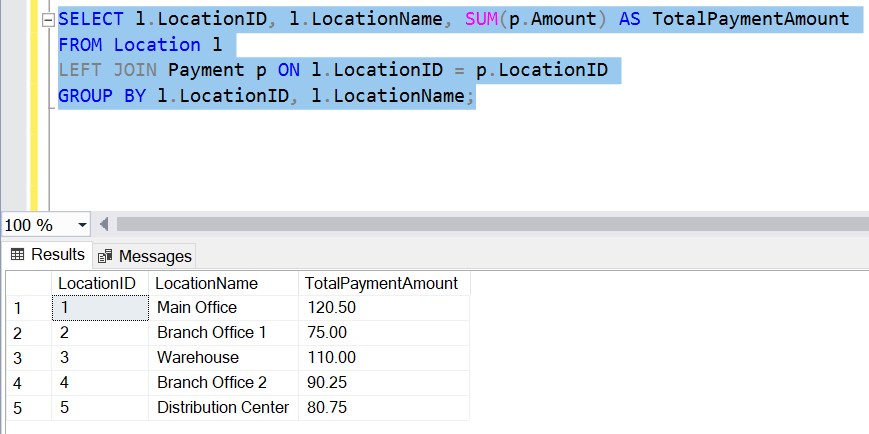
1. Retrieve a list of couriers along with the courier service details.



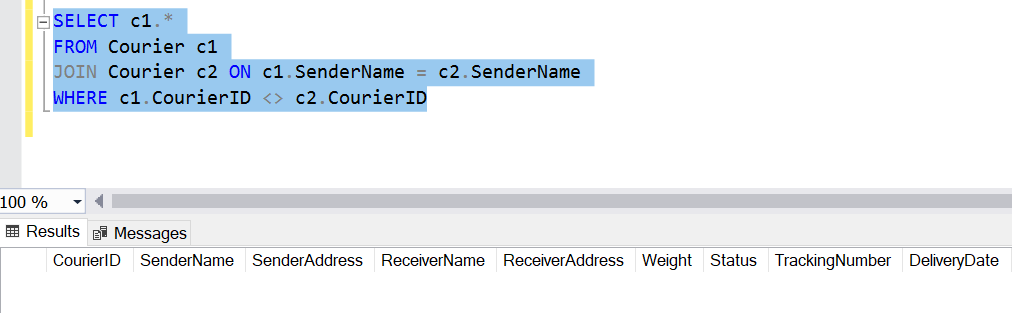
1. Retrieve a list of employees and the number of couriers assigned to each employee:



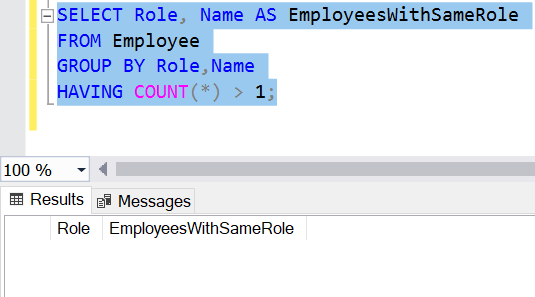
1. Retrieve a list of locations and the total payment amount received at each location:



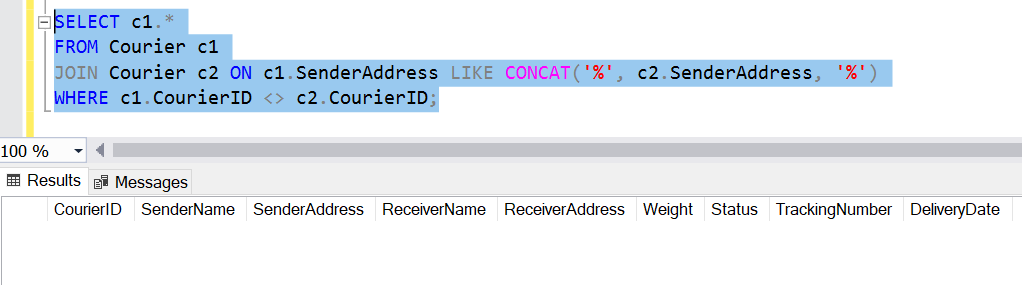
1. Retrieve all couriers sent by the same sender.



1. List all employees who share the same role.

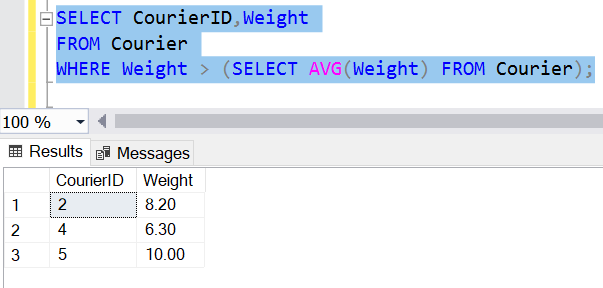


1. Retrieve all couriers sent from the same location (based on SenderAddress).

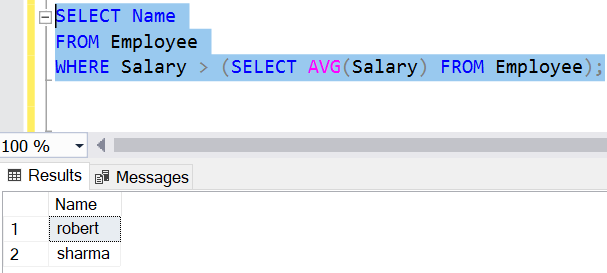


**SCOPE: Inner Queries, Non Equi Joins, Exist, Any, All**

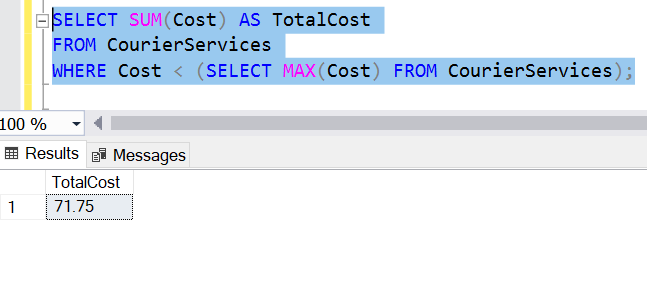
1. Find couriers that have a weight greater than the average weight of all couriers.



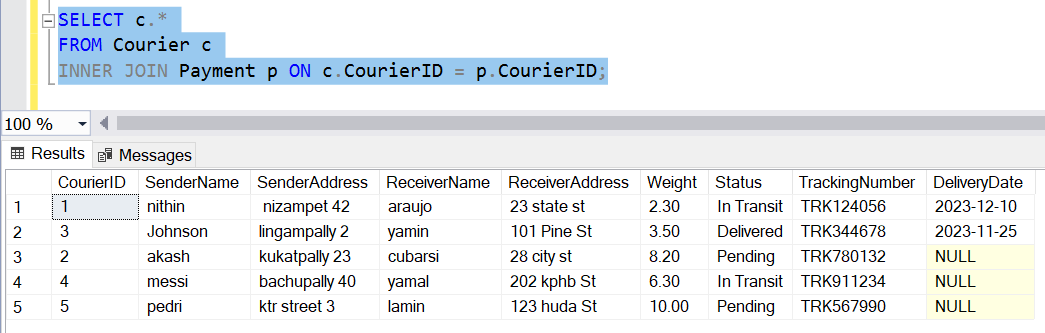
1. Find the names of all employees who have a salary greater than the average salary:



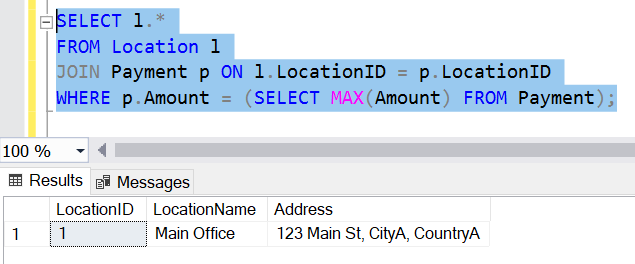
1. Find the total cost of all courier services where the cost is less than the maximum cost.



1. Find all couriers that have been paid for.



1. 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'):

