**DAY 2 (Exercises and Hands on)**

**YAZHINI S (DE130)**

**Date: 05-11-2024**

**-- Creating the Employee Table with Data Integrity Constraints**

CREATE TABLE Employee (

EmployeeID INT PRIMARY KEY,

FirstName NVARCHAR(50) NOT NULL,

LastName NVARCHAR(50) NOT NULL,

Age INT CHECK (Age >= 18),

Department NVARCHAR(50) DEFAULT 'IT',

Salary DECIMAL(10, 2) CHECK (Salary >= 0),

DateOfBirth DATE

);

**-- Insert Sample Data**

INSERT INTO Employee (EmployeeID, FirstName, LastName, Age, Department, Salary, DateOfBirth)

VALUES

(1, 'Priya', 'Mehta', 30, 'IT', 60000, '1994-01-15'),

(2, 'Raj', 'Tamizhavan', 28, 'Marketing', 50000, '1996-05-22'),

(3, 'Ananya', 'Sharma', 35, 'HR', 55000, '1989-08-11'),

(4, 'Kiran', 'Kumar', 40, 'IT', 70000, '1984-03-03');

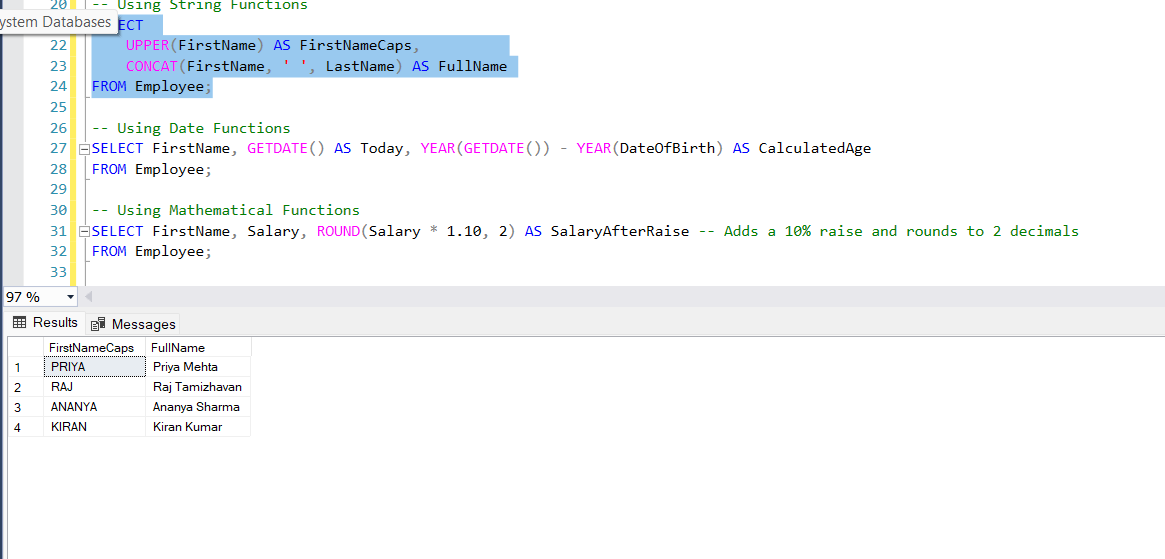
**-- Using String Functions**

SELECT

UPPER(FirstName) AS FirstNameCaps,

CONCAT(FirstName, ' ', LastName) AS FullName

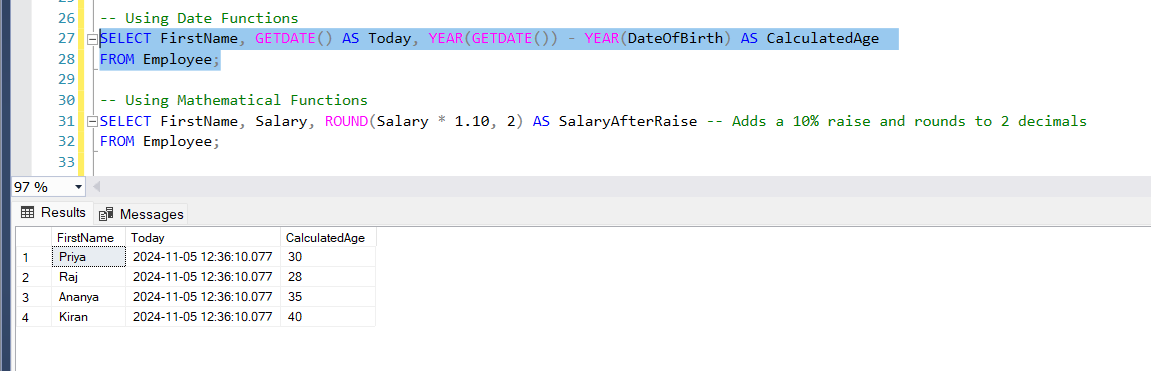
FROM Employee;



**-- Using Date Functions**

SELECT FirstName, GETDATE() AS Today, YEAR(GETDATE()) - YEAR(DateOfBirth) AS CalculatedAge

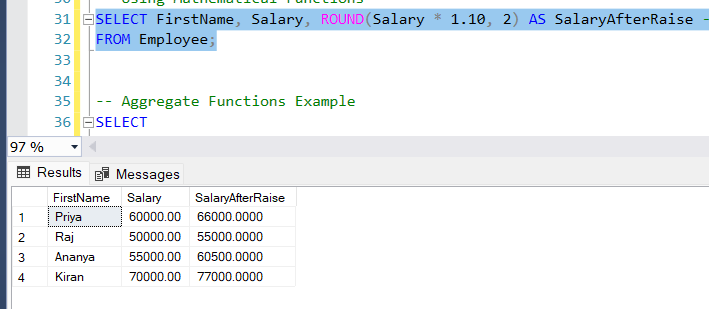
FROM Employee;



**-- Using Mathematical Functions**

SELECT FirstName, Salary, ROUND(Salary \* 1.10, 2) AS SalaryAfterRaise -- Adds a 10% raise and rounds to 2 decimals

FROM Employee;



**HANDS ON:**

* **Hands on Exercise: Filtering Data using SQL Queries**
* **Hands on Exercise: Total Aggregations using SQL Queries**
* **Hands on Exercise: Group By Aggregations using SQL Queries**
* **Hands on Exercise: Order of Execution of SQL Queries**
* **Hands on Exercise: Rules and Restrictions to Group and Filter Data in SQL queries**
* **Hands on Exercise: Filter Data based on Aggregated Results using Group By and Having**

**-- Aggregate Functions Example**

SELECT

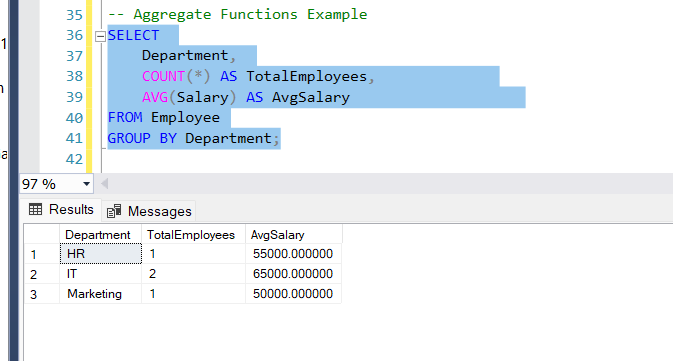
Department,

COUNT(\*) AS TotalEmployees,

AVG(Salary) AS AvgSalary

FROM Employee

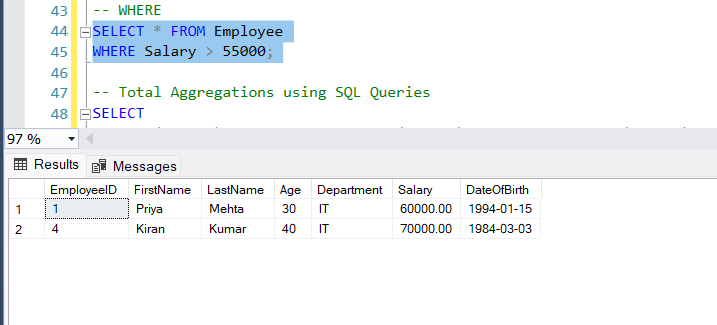
GROUP BY Department;



**-- WHERE**

SELECT \* FROM Employee

WHERE Salary > 55000;

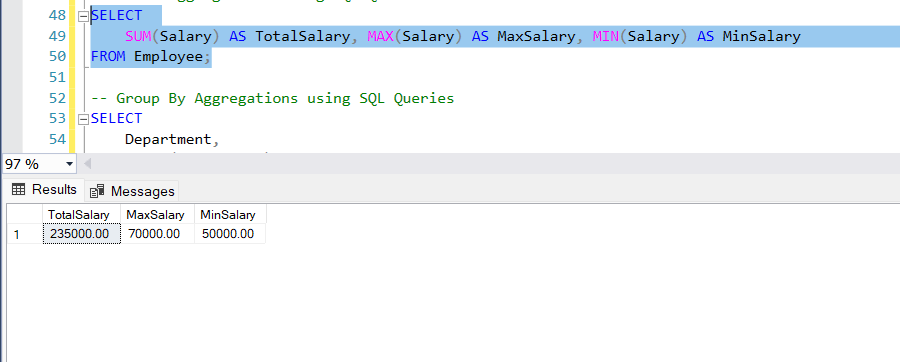


**-- Total Aggregations using SQL Queries**

SELECT

SUM(Salary) AS TotalSalary, MAX(Salary) AS MaxSalary, MIN(Salary) AS MinSalary

FROM Employee;



**-- Group By Aggregations using SQL Queries**

SELECT

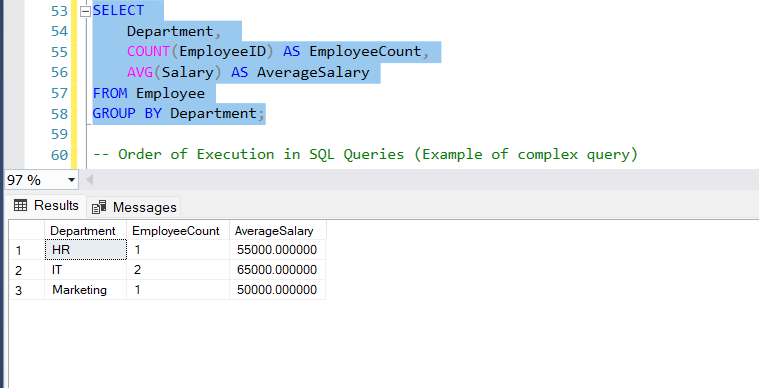
Department,

COUNT(EmployeeID) AS EmployeeCount,

AVG(Salary) AS AverageSalary

FROM Employee

GROUP BY Department;



**-- Order of Execution in SQL Queries, Group By and Having**

SELECT

Department,

AVG(Salary) AS AvgSalary

FROM Employee

WHERE Age > 25

GROUP BY Department

HAVING AVG(Salary) > 55000

ORDER BY AvgSalary DESC;

