Lab53:

Functions in SQL server:

Scalar function:

A scalar function returns a single value based on the input parameters.

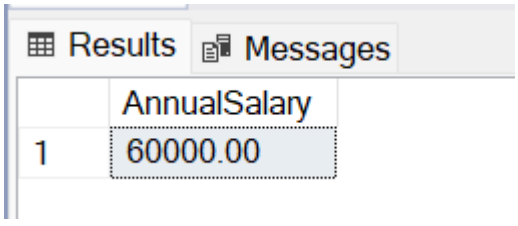
Creating a scalar function to calculate the annual salary of an employee.

```
-- Creating a scalar function to calculate the annual salary of an employee
CREATE FUNCTION dbo.CalculateAnnualSalary
(
    @MonthlySalary DECIMAL(18, 2)
)
RETURNS DECIMAL(18, 2)
AS
BEGIN
    -- Calculate the annual salary by multiplying the monthly salary by 12
    RETURN @MonthlySalary * 12;
END;
```

his script creates a scalar function named CalculateAnnualSalary.

The function accepts one input parameter @MonthlySalary of type DECIMAL(18, 2). It calculates the annual salary by multiplying the monthly salary by 12 and returns the result.

```
-- Using the CalculateAnnualSalary function to get the annual salary
SELECT dbo.CalculateAnnualSalary(5000) AS AnnualSalary;
```



Inline Table-Valued Function:

An inline table-valued function returns a table data type and is defined using a single SELECT statement.

e.g.

Creating an inline table-valued function to get employees with a salary above a certain threshold.

```
CREATE FUNCTION dbo.GetHighSalaryEmployees
(
    @SalaryThreshold DECIMAL(18, 2)
)
RETURNS TABLE
AS
RETURN
(
    -- Returning a table of employees with a salary greater than the provided threshold
    SELECT EmployeeID, EmpName, EmpSalary, EmpGender
    FROM dbo.Employee
    WHERE EmpSalary > @SalaryThreshold
);
```

- This script creates an inline table-valued function named GetHighSalaryEmployees.
- The function accepts one input parameter @SalaryThreshold of type DECIMAL(18, 2).
- It returns a table of employees with a salary greater than the provided threshold.

Using the GetHighSalaryEmployees function to get employees with a salary above 50000

```
SELECT * FROM dbo.GetHighSalaryEmployees(50000);
```

Results | Messages

	EmployeeID	EmpName	EmpSalary	EmpGender
1	2	Jane Smith	60000.00	F
2	3	Alice Johnson	55000.00	F
3	5	Alice	75000.00	F

Multi-Statement Table-Valued Function:

A multi-statement table-valued function returns a table data type and allows for multiple statements to populate the table.

e.g.

Creating a multi-statement table-valued function to get employees and their annual salaries.

```
CREATE FUNCTION dbo.GetEmployeesWithAnnualSalary
()
RETURNS @EmployeeAnnualSalary TABLE
(
    EmployeeID INT,
    EmpName NVARCHAR(100),
    EmpSalary DECIMAL(18, 2),
    AnnualSalary DECIMAL(18, 2)
)
AS
BEGIN
    -- Insert data into the return table with calculated annual salary
    INSERT INTO @EmployeeAnnualSalary (EmployeeID, EmpName, EmpSalary, AnnualSalary)
    SELECT EmployeeID, EmpName, EmpSalary, EmpSalary * 12 AS AnnualSalary
    FROM dbo.Employee;

    -- Return the result set
    RETURN;
END;
```

- This script creates a multi-statement table-valued function named GetEmployeesWithAnnualSalary.
- It does not accept any input parameters.
- It returns a table @EmployeeAnnualSalary with columns EmployeeID, EmpName, EmpSalary, and AnnualSalary.
- The function populates the return table by calculating the annual salary for each employee.

Using the GetEmployeesWithAnnualSalary function to get employees and their annual salaries

```
SELECT * FROM dbo.GetEmployeesWithAnnualSalary();
```

Results Messages

	EmployeeID	EmpName	EmpSalary	AnnualSalary
1	1	John Doe	50000.00	600000.00
2	2	Jane Smith	60000.00	720000.00
3	3	Alice Johnson	55000.00	660000.00
4	4	Bob Brown	45000.00	540000.00
5	5	Alice	75000.00	900000.00