

```

use FUH_COMPANY

--Sử dụng biến @model_salary, @model_salary và gán giá trị cho biến @model_salary = 90000
DECLARE @model_salary As INT;
SET @model_salary = 90000;
SELECT empSSN, empName, empName, empSalary
FROM
    tblEmployee
WHERE
    empSalary < @model_salary
ORDER BY empSSN;
Go

---Tổng lương của những người có lương lớn hơn 90000
Declare @total int
Select @total=Sum(empSalary)
From tblEmployee
Where empSalary >= 90000
SELECT
    @total AS "Lương"
Go

---Cập nhật tên Trịnh Hạnh bằng tên của nhân viên có empSSN=30121050015
DECLARE @name VARCHAR(MAX)
SELECT @name=empName
FROM    tblEmployee
WHERE   empSSN=30121050015

UPDATE tblEmployee
SET     empName=@name
WHERE   empName=N'Trịnh Hạnh'

---Sử dụng lệnh để in thông tin
DECLARE
@Diem_toan decimal(3,1) = 9.5,
@Diem_ly decimal(3,1) = 7.5,
@Diem_hoa decimal(3,1) = 8.0
DECLARE @diem_trung_binh decimal(3,1)
SET @Diem_trung_binh = (@Diem_toan + @Diem_ly + @Diem_hoa) / 3
PRINT @Diem_trung_binh

--Sử dụng print và select để in và nối chuỗi
DECLARE @Ho nvarchar(30), @Ten nvarchar(20)
SET @Ho = N'Hà Thị Bạch'
SET @Ten = N'Lan'
SELECT @Ho = N'Hà Thị Bạch', @Ten = N'Lan'
--PRINT N'Họ: ' + @Ho
--PRINT N'Tên: ' + @Ten
--OR
SELECT @Ho AS 'Họ', @Ten AS 'Tên'
Go

---Chuyển đổi kiểu dữ liệu
SELECT CONVERT(int, '25') + 30;
go
DECLARE @empName NVARCHAR(20), @empSalary DECIMAL
SET @empName=N'Phan Lệ Thuỳên'

```

```

SET @empSalary=90000
PRINT @empName + ' with salary is ' + CAST(@empSalary AS VARCHAR)
PRINT @empName + ' with salary is ' + CONVERT(VARCHAR, @empSalary)
Go

--Sử dụng cấu trúc điều kiện
DECLARE @nhanvien_salary INT, @ten NVarchar(200), @bonus int, @total int
SELECT @nhanvien_salary=empSalary, @ten = empName
FROM tblEmployee
WHERE empSSN=30121050004
IF @nhanvien_salary < 900000
    begin
        set @bonus=10000
        set @total= @bonus + @nhanvien_salary
        PRINT N'Lương bạn ' + @ten + ' là: ' + CAST(@total AS NVARCHAR)
    end
else
    begin
        set @bonus=20
        set @total= @bonus + @nhanvien_salary
        PRINT N'Lương bạn' + @ten + N' là: ' + convert(Nvarchar,@total)
    end
End

-- sử dụng case
DECLARE @emp_sex char(1), @str NVARCHAR(50)
SELECT @emp_sex=empSex
FROM tblEmployee
where empSSN=30121050004
SET @str= CASE @emp_sex
            WHEN 'F' THEN N'giới tính nữ'
            WHEN 'M' THEN N'Giới tính nam'
            ELSE N'Chưa đăng ký giới tính'
        END
PRINT @str

--Sử dụng vòng lặp while
DECLARE @counter INT = 1;
WHILE @counter <= 5
BEGIN
    PRINT 'Counter Value: ' + CAST(@counter AS NVARCHAR(10));
    SET @counter = @counter + 1;
END

--try catch
BEGIN TRY
DECLARE @num int;
SELECT @num=217/0;
END TRY
BEGIN CATCH
PRINT 'Error occurred, unable to devide by 0'
END CATCH;

---SỬ dụng cấu trúc If-statement nested
DECLARE @Number INT;
SET @Number = 50;
IF @Number > 100
    PRINT 'The number is large.';
ELSE

```

```

BEGIN
    IF @Number < 10
        PRINT 'The number is small.';
    ELSE
        PRINT 'The number is medium.';
    END ;

---Tạo proce để sử dụng lâu dài
CREATE PROCEDURE SelectAllEmployees
AS
SELECT *
FROM tblEmployee
GO;
--Chạy (biên dịch 1 lần), và sau này chỉ việc gọi Proce bằng lệnh exec
EXEC SelectAllEmployees

--Tìm kiếm theo địa chỉ cụ thể
CREATE PROCEDURE Selectemployess
@addr nvarchar(30)
AS
SELECT *
FROM tblEmployee
WHERE empAddress = @addr
GO;
exec Selectemployess @addr = N'TP. Hồ Chí Minh';

---Tìm theo nhiều điều kiện
CREATE PROCEDURE Search_employee
@addr nvarchar(30),
@code_emp decimal(18, 0)
AS
SELECT *
FROM tblEmployee
WHERE empAddress = @addr AND empSSN = @code_emp
GO;
exec Search_employee @addr =N'TP. Hà Nội', @code_emp =30121050254;

go

CREATE PROCEDURE ins_Emp
@ID decimal(18, 0),
@Name nvarchar(50)
AS
INSERT INTO tblEmployee(empSSN,empName)
VALUES (@ID,@Name)
go

EXEC ins_Emp @ID=45789635,@Name = N'Phan Thị Lệ Thuỳên'

--Tạo function

CREATE FUNCTION Multiply(@a INT,@b INT)
RETURNS INT AS
BEGIN
    RETURN @a * @b;
END;

```

```

GO
-- Gõi function
SELECT dbo.Multiply(25, 500) AS product_result;

GO

CREATE FUNCTION thuong(@Salary INT)
RETURNS nvarchar(50)
AS
BEGIN
DECLARE @tour NVARCHAR(50);
IF @Salary < 60000
SET @tour = N'Được tham gia du lịch';
ELSE
SET @tour = N'Tham gia chạy deadline dự án của công ty';
RETURN @tour;
END;
SELECT dbo.thuong(60000) AS Tour;

--Hàm trả về dạng bảng
CREATE FUNCTION salary_80 (@salary_max INT)
RETURNS TABLE
AS
RETURN (SELECT empSSN, empName, empAddress, empSalary
FROM tblEmployee
WHERE empSalary >= @salary_max);

go
SELECT *
FROM salary_80 (80000);

---Hàm trả về nhiều giá trị cho bảng
CREATE FUNCTION table_emp()
RETURNS @Employee TABLE
(
Id_emp decimal(18, 0),
e_Name nvarchar(50),
sex char(1),
salary decimal(18, 0)) AS
BEGIN
INSERT INTO @Employee
SELECT E.empSSN, E.empName, E.empSex, E.empSalary
FROM tblEmployee E;
UPDATE @Employee SET e_Name = N'Võ Việt Anh'
WHERE Id_emp = 30121050184;
RETURN
END
go
SELECT *
FROM table_emp();

---ví dụ khác---

go
CREATE FUNCTION SalaryRange
(
@minSalary DECIMAL(10, 2),
@maxSalary DECIMAL(10, 2)

```

```

)
RETURNS @EmployeeTable TABLE
(
    empID decimal(18, 0),
    empName NVARCHAR(50),
    empSalary DECIMAL(10, 2)
)
AS
BEGIN
    INSERT INTO @EmployeeTable
    SELECT empSSN, empName, empSalary
    FROM tblEmployee
    WHERE empSalary >= @minSalary AND empSalary <= @maxSalary;
    RETURN;
END;
go
SELECT *
FROM dbo.SalaryRange(40000, 100000);

--Sử dụng Cursor
DECLARE @Name nvarchar(50)
DECLARE Employee_Cursor CURSOR FOR
SELECT empName
FROM tblEmployee
OPEN Employee_Cursor
--Use the FETCH statement to get each row of data from the record set
--get data put into variable
FETCH NEXT FROM Employee_Cursor INTO @Name
--WHILE by checking the global variable @@FETCH_STATUS (=0 means success)
WHILE @@FETCH_STATUS = 0
    BEGIN
        PRINT 'Employee:' + @Name
        FETCH NEXT FROM Employee_Cursor INTO @Name
    END
CLOSE Employee_Cursor
DEALLOCATE Employee_Cursor

```