***Database Functions Assignment***

**Introduction;**

This report demonstrates the creation and usage of SQL functions to retrieve data from the AdventureWorks database. The task involved creating:

1. A **Table-Valued Function** to find employees by job title and arrange them by vacation hours.
2. A **Scalar-Valued Function** to retrieve an employee's salary rate based on their Person ID.

Both functions were implemented, tested, and the results are documented below.

**Table-Valued Function:**

**Objective:**  
This function retrieves employee details (FirstName, LastName, VacationHours) from the Employee and Person tables filtered by JobTitle and outputs the data unsorted. Sorting is applied during function execution.

**SQL Code**

CREATE FUNCTION dbo.GetEmployeesByJobTitle

(

@JobTitle NVARCHAR(50) -- Input parameter for the job title

)

RETURNS TABLE

AS

RETURN

(

SELECT

p.FirstName,

p.LastName,

e.VacationHours

FROM

HumanResources.Employee e

INNER JOIN

Person.Person p

ON

e.BusinessEntityID = p.BusinessEntityID

WHERE

e.JobTitle = @JobTitle -- Filters by the specified job title

);A computer screen with a white box

Description automatically generated

**Execution Example:**

SELECT \*

FROM dbo.GetEmployeesByJobTitle('Engineering Manager')

ORDER BY VacationHours ASC; -- Sorting applied here

**Screenshot of Results:**  
A computer screen with a white screen

Description automatically generated

**Scalar-Valued Function:**

**Objective:**  
This function retrieves the salary rate (Rate) of an employee from the EmployeePayHistory table based on their PersonID.

**SQL Code**

CREATE FUNCTION dbo.GetEmployeeSalaryRate

(

@PersonID INT -- Input parameter for the person's ID

)

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @SalaryRate DECIMAL(10, 2); -- Variable to store the salary rate

SELECT

@SalaryRate = eph.Rate

FROM

HumanResources.EmployeePayHistory eph

INNER JOIN

HumanResources.Employee e

ON

eph.BusinessEntityID = e.BusinessEntityID

WHERE

e.BusinessEntityID = @PersonID;

RETURN @SalaryRate;

END;

**Execution Example**

SELECT dbo.GetEmployeeSalaryRate(1) AS SalaryRate;

**Screenshot of Results:**  
A computer screen with a white screen

Description automatically generated

**Conclusion:**

Both functions were successfully created and tested. The following were accomplished:

1. **Table-Valued Function:** Retrieves employee details filtered by JobTitle, and sorting is applied during execution.
2. **Scalar-Valued Function:** Retrieves the salary rate of an employee based on their PersonID.

The screenshots of the test results are attached for reference.