Employee Management System - SQL Exercises

Database Schema

The Employee Management System database schema consists of the following tables:

1. Departments

Column	Data Type	Description
DepartmentID	INT (PK)	Unique department ID
DepartmentName	VARCHAR(100)	Name of the department

2. Employees

Column	Data Type	Description
EmployeeID	INT (PK)	Unique employee ID
FirstName	VARCHAR(50)	Employee's first name
LastName	VARCHAR(50)	Employee's last name
DepartmentID	INT (FK)	Linked to Departments
Salary	DECIMAL(10,2)	Monthly salary
JoinDate	DATE	Date of joining

Sample Data

Sample data for testing:

Departments:

DepartmentID	DepartmentName
1	HR
2	IT
3	Finance

Employees:

EmployeeID	FirstName	LastName	DepartmentID	Salary	JoinDate
1	John	Doe	1	5000.00	2020-01-15
2	Jane	Smith	2	6000.00	2019-03-22
3	Bob	Johnson	3	5500.00	2021-07-01

Exercises

Exercise 1: Create a Scalar Function

Goal: Create a scalar function to calculate the annual salary of an employee.

Steps:

1. Define a scalar function named `fn_CalculateAnnualSalary`.

2. The function should take `Salary` as input and return `Salary * 12`.

3. Test the function by selecting the annual salary for each employee.

Exercise 2: Create a Table-Valued Function

Goal: Create a table-valued function to return employees in a specific department.

Steps:

1. Define a table-valued function named `fn_GetEmployeesByDepartment`.

2. The function should take `DepartmentID` as input and return a table with employee details.

3. Test the function by selecting employees from the IT department.

Exercise 3: Create a User-Defined Function

Goal: Create a user-defined function to calculate the bonus for an employee.

Steps:

1. Define a user-defined function named `fn_CalculateBonus`.

2. The function should take `Salary` as input and return `Salary * 0.10`.

3. Test the function by selecting the bonus for each employee.

Exercise 4: Modify a User-Defined Function

Goal: Modify the `fn_CalculateBonus` function to return `Salary * 0.15`.

Steps:

1. Alter the `fn_CalculateBonus` function to return `Salary * 0.15`.

2. Test the modified function by selecting the bonus for each employee.

Exercise 5: Delete a User-Defined Function

Goal: Delete the `fn_CalculateBonus` function.

Steps:

1. Drop the `fn_CalculateBonus` function.

2. Verify that the function has been deleted.

Exercise 6: Execute a User-Defined Function

Goal: Execute the `fn_CalculateAnnualSalary` function.

Steps:

1. Use the `fn_CalculateAnnualSalary` function to calculate the annual salary for each employee.

2. Verify the results.

Exercise 7: Return Data from a Scalar Function

Goal: Return the annual salary for a specific employee using `fn_CalculateAnnualSalary`.

Steps:

1. Execute the `fn_CalculateAnnualSalary` function for an employee with `EmployeeID = 1`.

2. Verify the result.

Exercise 8: Return Data from a Table-Valued Function

Goal: Return employees from the Finance department using `fn_GetEmployeesByDepartment`.

Steps:

1. Execute the `fn_GetEmployeesByDepartment` function for `DepartmentID = 3`.

2. Verify the results.

Exercise 9: Create a Nested User-Defined Function

Goal: Create a nested user-defined function to calculate the total compensation for an employee.

Steps:

1. Define a user-defined function named `fn_CalculateTotalCompensation`.

2. The function should use `fn_CalculateAnnualSalary` and `fn_CalculateBonus` to calculate total compensation.

3. Test the function by selecting the total compensation for each employee.

Exercise 10: Modify a Nested User-Defined Function

Goal: Modify the `fn_CalculateTotalCompensation` function to include a new bonus calculation.

Steps:

1. Alter the `fn_CalculateTotalCompensation` function to use the modified `fn_CalculateBonus` function.

2. Test the modified function by selecting the total compensation for each employee.