T-SQL and Stored Procedures

1.Create a database with two tables

Persons (id (PK), first name, last name, SSN) and Accounts (id (PK), person id (FK), balance). Insert few records for testing.

Write a stored procedure that selects the full names of all persons.

2.Create a stored procedure

Your task is to create a stored procedure that accepts a number as a parameter and returns all persons who have more money in their accounts than the supplied number.

3.Create a function with parameters

Your task is to create a function that accepts as parameters – sum, yearly interest rate and number of months. It should calculate and return the new sum. Write a SELECT to test whether the function works as expected.

4.Create a stored procedure that uses the function from the previous example.

Your task is to create a stored procedure that uses the function from the previous example to give an interest to a person's account for one month. It should take the AccountId and the interest rate as parameters.

5.Add two more stored procedures WithdrawMoney and DepositMoney.

Add two more stored procedures WithdrawMoney (AccountId, money) and DepositMoney (AccountId, money) that operate in transactions.

6.Create table Logs.

Create another table – Logs (LogID, AccountID, OldSum, NewSum). Add a trigger to the Accounts table that enters a new entry into the Logs table every time the sum on an account changes.

7.Define function in the FMI database.

Define a function in the database FMI that returns all Employee's names (first or middle or last name) and all town's names that are comprised of given set of letters.

Example: 'oistmiahf' will return 'Sofia', 'Smith', but not 'Rob' and 'Guy'.

8.Using database cursor write a T-SQL

Using database cursor write a T-SQL script that scans all employees and their addresses and prints all pairs of employees that live in the same town.