# Homework: Transact-SQL

This document defines the homework assignments from the ["Databases" Course @ Software University](https://softuni.bg/trainings/20/Databases-Feb-2015). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

## 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.

|  |
| --- |
| **Full name** |
| Susan |
| Kim |
| Jimmy |

## 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.

## 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.

## 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.

## Add two more stored procedures WithdrawMoney and DepositMoney.

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

## 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.

## Define function in the SoftUni database.

Define a function in the database **SoftUni** 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'**.

## 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.

|  |
| --- |
| Wood: John Wood Redmond John  Hill: John Wood Redmond Annette  Feng: John Wood Redmond Hanying  Sousa: John Wood Redmond Anibal  Glimp: John Wood Redmond Diane  Pournasseh: John Wood Redmond Houman  Kane: John Wood Redmond Lori  … |

## Define a .NET aggregate function

Define a .NET aggregate function **StrConcat** that takes as input a sequence of strings and return a single string that consists of the input strings separated by ','. For example the following SQL statement should return a single string:

|  |
| --- |
| **SELECT StrConcat (FirstName + ' ' + LastName)**  **FROM Employees** |

## \*Write a T-SQL script

Write a T-SQL script that shows for each town a list of all employees that live in it. Sample output:

|  |
| --- |
| **Sofia -> Svetlin Nakov, Martin Kulov, Vladimir Georgiev**  **Ottawa -> Jose Saraiva,**  **…** |