# Exercises: User Defined Functions

You can check your solutions here: <https://judge.softuni.org/Contests/3140/User-Defined-Functions>.

# Queries for SoftUni Database

## Salary Level Function

Write a function **ufn\_GetSalaryLevel(@salary DECIMAL(18,4))** that receives **salary of an employee** and returns the **level of the salary**.

* If salary is **< 30000** return **"Low"**
* If salary is **between 30000 and 50000 (inclusive)** return **"Average"**
* If salary is **> 50000** return **"High"**

#### Example

|  |  |
| --- | --- |
| **Salary** | **Salary Level** |
| 13500.00 | Low |
| 43300.00 | Average |
| 125500.00 | High |

## Define Function

Define a function **ufn\_IsWordComprised(@setOfLetters, @word)** that returns **true** or **false** depending on that if the word is a comprised of the given set of letters.

#### Example

|  |  |  |
| --- | --- | --- |
| **SetOfLetters** | **Word** | **Result** |
| oistmiahf | Sofia | 1 |
| oistmiahf | halves | 0 |
| bobr | Rob | 1 |
| pppp | Guy | 0 |

# Queries for Bank Database

## Future Value Function

Your task is to create a function **ufn\_CalculateFutureValue** that accepts as parameters – **sum (decimal)**, **yearly interest rate (float)** and **number of years(int)**. It should calculate and return the future value of the initial sum rounded to the **fourth digit** after the decimal delimiter. Using the following formula:

* **I** – Initial sum
* **R** – Yearly interest rate
* **T** – Number of years

#### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| **Initial sum:** 1000  **Yearly Interest rate:** 10%  **years:** 5  ufn\_CalculateFutureValue(1000, 0.1, 5) | 1610.5100 |

# Queries for Diablo Database

You are given a **database "Diablo"** holding users, games, items, characters and statistics available as SQL script. Your task is to write some stored procedures, views and other server-side database objects and write some SQL queries for displaying data from the database.

**Important:** start with a **clean copy of the "Diablo" database** **on each problem**. Just execute the SQL script again.

## \*Scalar Function: Cash in User Games Odd Rows

Create a **function** **ufn\_CashInUsersGames** that **sums the cash of odd rows**. Rows must be ordered by cash in descending order. The function should take a **game name** as a **parameter** and **return the result as table**. Submit **only your function** **in**.

Execute the function over the following game names, ordered exactly like: "**Love in a mist**".

#### Output

|  |
| --- |
| **SumCash** |
| 8585.00 |

#### Hint

Use **ROW\_NUMBER** to get the rankings of all rows based on order criteria.