**FinTech Dev - Assessment Test**

Version 2.1

***Duration****: 60 minutes*

***Instructions****:*

1. *This is open book test.*
2. *There are three* ***mandatory*** *sections in the test:*

[*Section 1*](#Section1)*: T-SQL*

[*Section 2*](#Section2)*: C#/.NET/Algorithm*

[*Section 3*](#Section3)*: JavaScript*

*Each section has one question.*

1. *There is a* ***bonus*** [*Section 4*](#Section4)*with a supplementary T-SQL question.*

*Please attempt the bonus question only if you are done with the mandatory sections.*

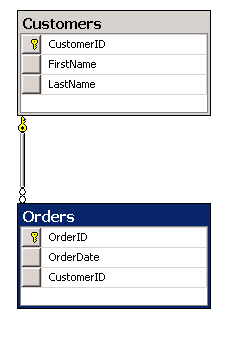
1. *You may want to take help of fiddles to test run your code.*

*Fiddles for C#, JavaScript and T-SQL can be found at* [*link1*](https://dotnetfiddle.net/)*,* [*link2*](https://jsfiddle.net/) *and* [*link3*](http://rextester.com/l/sql_server_online_compiler) *respectively.*

*[To be able to see output from console.log() in JSFiddle, you mayconsider using the solution mentioned* [*here*](https://stackoverflow.com/questions/17382200/print-var-in-jsfiddle)*]*

**Section 1: T-SQL**

Consider a database that contains following tables.



Write a T-SQL query for a report. The query must meet the following requirements:

* Use the first initial of the table as an alias.
* Return the most recent order date for each customer.
* Retrieve *FirstName* of the person (who placed the order) as *CustomerName*.
* Return the Order date in a column named *MostRecentOrderDate* that appears as the last column in the report.
* Return the most recent orders first.

Your answer here:

*SELECT C.FirstName AS CustomerName, O.OrderId, O.OrderDate as MostRecentOrderDate*

*FROM Customers C*

*inner join Orders O*

*on C.CustomerID = O.CustomerID*

*and OrderID = (*

*SELECT TOP 1 SO.OrderID*

*FROM Orders SO*

*WHERE SO.CustomerID = O.CustomerID*

*ORDER BY SO.OrderDate DESC*

*)*

*Order By MostRecentOrderDate Desc*

**Section 2: C#/.NET/Algorithm**

Write a function that, when passed a list and a target sum, prints combinations of all numbers, whose sum is equal to the target sum. If there are no two numbers, the function should print **“*no pair found*”**.

For example,

FindTwoSum(new List<int>() { 3, 1, 5, 7, 5, 9 }, 10)

Should print in the console

{3, 7}

{1, 9}

{5, 5}

Your answer here:

*public static void FindTwoSum(List<int> list, int targetSum)*

*{*

*List<string> resultList = new List<string>();*

*foreach (int n in list)*

*{*

*foreach (int i in list)*

*{*

*if (n + i == targetSum)*

*{*

*if (!resultList.Contains(string.Format("{{{0},{1}}}", i, n)))*

*resultList.Add(string.Format("{{{0},{1}}}", n, i));*

*}*

*}*

*}*

*if (resultList.Count != 0)*

*{*

*resultList.Distinct().ToList().ForEach(s => Console.WriteLine(s+"\n"));*

*}*

*else*

*{*

*Console.WriteLine("No pair found.");*

*}*

*}*

**Section 3: JavaScript**

Write a function called *getClone* that takes an object and creates an object copy of it but does not copy deep property of the input object.

Example:

var obj = {foo : 'Bar'};

var cloneObj = getClone(obj); // getClone is the function which you have to write

console.log(cloneObj === getClone(obj)); // this should return false

console.log(cloneObj == getClone(obj)); // this should return true

Your answer here:

There seems to be a problem with the question as I could not achieve the following even after trying certain ways of cloning/copying/assigning the object. I used Visual Studio Code instead of the provided link for JSFiddle.

console.log(cloneObj === getClone(obj)); // this should return false

console.log(cloneObj == getClone(obj)); // this should return true

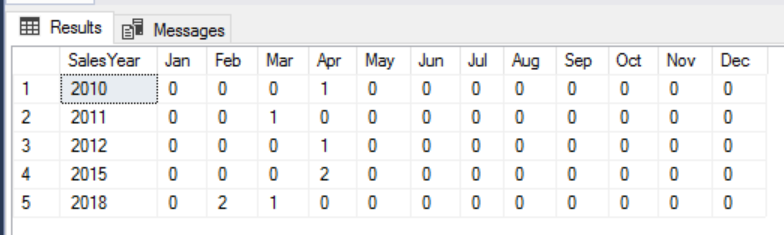
**Section 4: Bonus T-SQL**

Consider the same two tables from previous T-SQL question.

Write a T-SQL query for a report. The query must meet the following requirements:

* Use the first initial of the table as an alias.
* Return year and month wise total sales count.
* First column should be years in ascending order. The remaining columns should be twelve months.

An example query output:



Your answer here: