Quiz 1 - Written

**1. SQL**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Person**   |  |  | | --- | --- | | **Name** | **City** | | Adam | Kuala Lumpur | | John | Singapore | | Henry | Singapore | | Dominic | Bangkok | | **Log**   |  |  |  | | --- | --- | --- | | **Name** | **City** | **ModifiedDate** | | Adam | Singapore | 2015-01-01 | |

**1.1 Query**

Write a query using **sub-query** and **aggregation** to get the list of person which city appeared more than once.

Answer:

SELECT ROW\_NUMBER() OVER(ORDER BY [Name]) [NO], [Name] FROM

(

SELECT A.Name FROM [Person] A

WHERE 1 = 1

AND A.City IN (

SELECT [City]

FROM [dbo].[Person]

GROUP BY [City]

HAVING COUNT([City])> 1

)

) A

**1.2 CRUD – output (atomic query)**

Write a single query that update the **Person** table and simultaneously insert into **Log** table for the deleted information with the modified date.

Answer:

DECLARE @NAME\_PARAM VARCHAR(50)='Adam', @CITY\_PARAM VARCHAR(50)='Singapore'

BEGIN

SET NOCOUNT ON;

BEGIN TRANSACTION

BEGIN TRY

INSERT INTO [Log](

[Name],

[City],

ModifiedDate

)

SELECT [Name],

[City],

GETDATE()

FROM [Person]

WHERE [Name] = @NAME\_PARAM;

UPDATE [Person] SET

[City] = @CITY\_PARAM

WHERE [Name] = @NAME\_PARAM;

COMMIT TRANSACTION

END TRY

BEGIN CATCH

ROLLBACK TRANSACTION [ARCH\_LOG]

DECLARE @error\_msg\_catch VARCHAR(MAX)

SET @error\_msg\_catch = ERROR\_MESSAGE();

PRINT @error\_msg\_catch

END CATCH

END

**2. JavaScript**

**2.1 Simple JS & Logic question**

Write 2 ways to get multiplication by 8 for the variable **x** without using multiplication operator.

var x = 2;

console.log(y1); // output : 16

console.log(y2); // output : 16

Answer:

var x = 2;

var y1 = 0,

  y2 = 0;

var a = 8;

for (var i = 0; i < a; i++) {

  y1 += x;

}

for (var j = 0; j < x; j++) {

  for (var i = 0; i < a; i++) {

    y2 += 1;

  }

}

console.log(y1);

console.log(y2);

**2.2 Prototype in JS**

Create native methods for string data type to repeat *N* times.

console.log(‘hello’.repeat(3)); //output: hellohellohello

Answer:

var repeat = 3;

var word = "hello";

var result = "";

for (let i = 0; i < repeat; i++) {

  result += word;

}

console.log(result);

**3. C#**

**3.1 Simple OOP and syntax**

Create a class for employee who has **name** and **salary** properties with **setter** and **getter** methods.

Answer:

public class Employee

{

public string Name { get; set; }

public decimal Salary { get; set; }

}

**3.2** **Pointer reference in C#**

What is the output from b.Name?

Student a;

Student b = new Student();

a = b;

a.Name = “Michael”;

Response.Write(b.Name);

Answer:

Michael