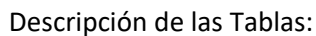


Prueba Practica .NET

Instrucciones: Realice la siguiente prueba generando los scripts / archivos que aplique para cada uno y súbalo a un **repositorio GIT** en cualquier servicio en la nube. El link deberá ser enviado al correo electrónico de donde recibió la prueba para poderlo descargar y revisar.

1. Escriba el código de una página ASPX que solicite el nombre del usuario, contraseña, y que valide que los campos no vayan vacíos antes de enviar la solicitud al servidor.
2. Escriba un documento XML que permita intercambiar información de tarjetas de crédito entre dos sistemas distintos: nombre del sistema, localización, número de tarjeta, ID único de cliente, etc.

```
<tarjetaCredito>
  <sistemaNombre>NombreSistema</sistemaNombre>
  <localizacion>LocalizacionSistema</localizacion>
  <numeroTarjeta>1234567890123456</numeroTarjeta>
  <idCliente>123</idCliente>
  <cliente>
    <nombre>NombreCliente</nombre>
    <apellido>ApellidoCliente</apellido>
    <direccion>DireccionCliente</direccion>
    <telefono>TelefonoCliente</telefono>
    <email>EmailCliente</email>
  </cliente>
  <detallesTarjeta>
    <fechaExpiracion>MM/AA</fechaExpiracion>
    <nombreTitular>NombreTitular</nombreTitular>
    <tipo>TipoTarjeta</tipo>
    <marcaEmisor>marcaEmisor</marcaEmisor>
  </detallesTarjeta>
</tarjetaCredito>
```



- Otras:

- **FORMATO_MENSAJE:** son los formatos de los mensajes existentes.
- **MENSAJE:** si el mensaje aplica a que proyecto y que producto

- A. Escriba la consulta en SQL que devuelva el nombre del proyecto y sus productos correspondientes del proyecto premia cuyo código es 1

```
SELECT P.NOMBRE AS NombreProyecto, PROD.DESCRIPCION AS NombreProducto
FROM PROYECTO P
INNER JOIN PRODUCTO_PROYECTO PP ON P.PROYECTO = PP.PROYECTO
INNER JOIN PRODUCTO PROD ON PP.PRODUCTO = PROD.PRODUCTO
WHERE P.PROYECTO = 1;
```

- B. Escriba una consulta SQL que devuelva los distintos mensajes que hay, indicando a qué proyecto y producto pertenecen.

```
SELECT M.COD_MENSAJE, P.NOMBRE AS NombreProyecto, PROD.DESCRIPCION AS
NombreProducto
FROM MENSAJE M
INNER JOIN PROYECTO P ON M.PROYECTO = P.PROYECTO
INNER JOIN PRODUCTO PROD ON M.PRODUCTO = PROD.PRODUCTO;
```

- C. Escriba una consulta SQL que devuelva los distintos mensajes que hay, indicando a qué proyecto y producto pertenecen. Pero si el mensaje está en todos los productos de un proyecto, en lugar de mostrar cada producto, debe mostrar el nombre del proyecto y un solo producto que diga "TODOS".

```
SELECT
    M.COD_MENSAJE,
    P.NOMBRE AS NombreProyecto,
    CASE
        WHEN PP.CantidadProductos = PM.CantidadMensajes THEN 'TODOS'
        ELSE PROD.DESCRIPCION
    END AS NombreProducto
FROM MENSAJE M
INNER JOIN PROYECTO P ON M.PROYECTO = P.PROYECTO
INNER JOIN PRODUCTO PROD ON M.PRODUCTO = PROD.PRODUCTO
INNER JOIN (SELECT PROYECTO, COUNT(*) AS CantidadProductos FROM
PRODUCTO_PROYECTO GROUP BY PROYECTO) PP ON P.PROYECTO = PP.PROYECTO
INNER JOIN (SELECT PROYECTO, PRODUCTO, COUNT(*) AS CantidadMensajes FROM MENSAJE
GROUP BY PROYECTO, PRODUCTO) PM ON P.PROYECTO = PM.PROYECTO AND
PROD.PRODUCTO = PM.PRODUCTO;
```

4. Escriba el código de una clase, en sintaxis C# que permita hacer conexiones a una base de datos cualquiera, y que devuelva en uno de sus métodos un Dataset al hacer un SELECT, y en otro de sus métodos el número de filas afectadas al ejecutar un INSERT, UPDATE o DELETE. También debe poder devolver el número de filas que devolvería un SELECT si se ejecutara.

```
using System;
using System.Data;
using System.Data.SqlClient;

public class DatosCRUD
{
    private string connectionString;

    public DatosCRUD(string connectionString)
    {
        this.connectionString = connectionString;
    }

    public DataSet EjecutarConsulta(string query)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                using (SqlDataAdapter adapter = new SqlDataAdapter(command))
                {
                    DataSet dataSet = new DataSet();
                    adapter.Fill(dataSet);
                    return dataSet;
                }
            }
        }
    }

    public int ContarFilasAfectadas(string query)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                return command.ContarFilasAfectadas();
            }
        }
    }
}
```

```
public int ContarFilasRespuesta(string query)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            connection.Open();
            using (SqlDataReader reader = command.ExecuteReader())
            {
                int count = 0;
                while (reader.Read())
                {
                    count++;
                }
                return count;
            }
        }
    }
}
```

5. Construya un modelo Entidad Relación que permita llevar el control de Canje de sus puntos acumulados por premios.
- El artículo debe tener código, nombre, tipo y subtipo.
 - Se debe manejar la existencia de artículos por sucursal. Ejemplo, está la sucursal Promerica Majadas y Promerica Columbus y en cada sucursal existen artículos como DVD de Disney, lapiceros, mouse, etc. Y cada sucursal debe saber qué cantidad posee en inventario.
 - Van a existir varias promociones para canjear y estas van a estar compuestas por varios artículos. Estas promociones poseen rangos de fecha de vigencia y en dichas promociones estará el costo de los puntos y el precio del artículo
 - El canje debe tener ID del artículo, Cantidad Canjeada, Precio del artículo, la promoción y observaciones.

