PRUEBA TÉCNICA

Nombre del aspirante: Daniel Alejandro Moreno

Cargo al que aspira: Ingeniero de desarrollo Jr

Celular: 3174371246 email: daniel\_9127@hotmail.com

Fecha: 17-03-23

SQL

1. Escribe el código sql para realizar las siguientes acciones:
   1. Que es un procedimiento almacenado y para que sirve
   2. Crear una tabla de nombre “producto” que tenga un campo “nombre”, “precio” y “fecha”.
   3. Insertar el producto leche con fecha del 15 de dciembre del 2010 y precio de $12.
   4. Inserte por lo menos 6 registros en la tabla “Productos” con diferentes fechas y precios.
   5. Cambiar el nombre del producto “Leche” a “Leche Entera”.
   6. Mostrar el nombre de todos los productos que cuesten más de $5.
   7. Borrar todos los productos.
   8. Que es una función
   9. Crear un Procedimiento que grabe en la tabla “Producto” y como parametros reciba (Nombre,Precio,fecha)
   10. Que es y para que sirve un Trigger
   11. Que diferencia hay entre una función tipo tabla y una funcion escalar
   12. Que es join y para que sirve
   13. Para que utilizo un left join y de un ejemplo
   14. Cree una vista que me retorne todos los productos de la tabla Producto
   15. Cree una funcion que retorne el precio de un producto como parametro recibirá el nombre del producto

PROGRAMACIÓN

Entidades básicas para el ejercicio.

* Técnico (nombre, código, sueldo base)
* Sucursal (código, nombre)
* Elementos (código, nombre)

1. Debe crear una aplicación web en C# que permita almacenar la información básica referente a unos técnicos los cuales pueden pertenecer una sucursal y pueden tener uno o varios elementos asignados, esta aplicación debe cumplir los siguientes requerimientos:

* Se debe permitir visualizar, crear, editar y eliminar el técnico.
* La sucursal debe venir de un maestro desde la base de datos, no requiere administración.
* Los elementos deben venir de un maestro desde la base de datos, no requieren administración.
* Cada empleado debe tener minio 1 elementos asignados, debe validar que no se guarde el técnico si aún no se le asignan elementos.
* Por cada elemento asignado al técnico debe agregar una cantidad de ese elemento que le asignaron (esta cantidad es independiente para cada técnico) , mínimo pueden asignarle 1 cantidad y máximo 10 del mismo elemento.
* Un elemento solo puede asignarse una vez, no puede agregar 2 veces el mismo elemento al técnico, para esto esta la cantidad, debe validarse esto.
* Todos los datos base del Técnico (nombre, código y sueldo base) y la sucursal son obligatorios.
* El dato de código del técnico solo debe permitir caracteres numéricos y letras, no puede llevar espacios ni caracteres especiales, debe validarse.
* El usuario puede pertenecer únicamente a una sucursal.
* Para la visualización de los técnicos creados debe presentarse una tabla al iniciar la pantalla con la información básica del técnico, la sucursal asignada y la cantidad de elementos que tiene asignados.
* La creación, edición y eliminación del técnico debe realizarse en una ventana emergente que se despliegue desde un botón en la pantalla de visualización y al guardar la información.
* Si puede bloquear la pantalla mientras realiza la petición asíncrona por favor impleméntelo.

Debe crear en la base de datos las tablas necesarias para representar las entidades y las relaciones entre estas con el fin de cumplir con los requerimientos establecidos, puede agregar tantas propiedades y tablas como crea necesario.

Para el desarrollo de la aplicación se requiere:

* Usar una base de datos en SQL Server, por favor indicar que versión fue utilizada.
* La aplicación debe desarrollarse en C#, .Net Framework 4.6 o posteriores.
* Si conoce la arquitectura de N-Capas por favor usarla. (No requerido pero deseable).
* Para la aplicación puede usar MVC y API REST.
* Para la conexión a base de datos se prefiere que use Clases LINQ to SQL pero Entity Framework también es aceptado.
* La consulta en la base de datos puede usar procedimientos almacenados o LINQ.
* Puede usar JQUERY donde lo requiera.

**ENVIÓ DE LA INFORMACIÓN**

Para el envió de la información puede comprimir el proyecto en un .zip y la base de datos realizarle un backup y adjuntarlo, por favor indicar la versión de visual studio utilizada y la versión de Sql server usada en la base de datos.

**Respuestas**

a. un procedimiento almacenado es un conjunto de instrucciones ya sean parámetros para recibir valores de salida, o ejecutar instrucciones programadas que se ejecutan directamente en la base de datos. Por lo general esto se usa para generar mejor rendimiento

b. CREATE TABLE producto ( nombre VARCHAR(50) PRIMARY KEY , precio INT , fecha DATE);

c. INSERT INTO producto (nombre, precio, fecha) VALUES (‘Leche’,12,’2010-12-15’);

d. NSERT INTO producto (nombre, precio, fecha) VALUES (‘Leche’,12,’2010-12-15’),(‘Pan’,5,’2010-12-10’),(‘Huevos’,2,’2010-12-5’),(‘Arepa’,8,’2010-12-11’),(‘Yoghurt’,14,’2010-12-18’),(‘Café’,9,’2010-12-19’) ;

e. UPDATE producto SET nombre=‘Leche’ WHERE nombre=‘Leche Entera’ ;

f. SELECT nombre FROM producto WHERE (precio>5);

g. DELETE FROM producto;

h. Una función es una rutina de instrucciones diseñadas para realizar alguna tarea en especifico, la cual recibe parámetros y devuelve bien sea un valor o una tabla.

1. CREATE TABLE producto ( nombre VARCHAR(50) , precio INT , fecha DATE);

INSERT INTO producto (nombre, precio, fecha) VALUES (‘producto1’,20,’2020-08-25’);

j. Los trigger son objetos de la base de datos que pueden ejecutar diferentes acciones dependiendo de las diferentes acciones o eventos que queramos modificar.

k. La diferencia entre una función tipo tabla y una función escalar radica en lo que retorna cada una, la primera retorna una tabla con sus respectivos valores mientras que la escalar un único valor ya sea una cadena de texto, o un valor.

l. Join se utiliza para enlazar los datos de dos tablas relacionadas entre si por un campo en común que por lo general es una llave foránea y así mezclar los datos provenientes de las dos tablas.

m. Left join es utilizada para devolver todos los datos de la tabla que esta a la izquierda y los datos que coinciden con la tabla de la derecha en caso de que existan.

Un ejemplo sencillo puede ser tener dos tablas, la primera con un listado de clientes con su respectivo id y la segunda con con un listado de pedidos con id de pedido, el total del pedido y el id del cliente que lo hizo; En este caso el left Join lo que haría es que nos traería una tabla con clientes, el id del pedido y del cliente y el valor del pedido, si un usuario no ha realizado pedido nos enviaría los datos en null.

n. CREATE VIEW vistaProductos AS

SELECT \* FROM producto

SELECT \* FROM vistaProductos

o. CREATE FUNCTION obtenerPrecio (nombreProducto VARCHAR(50))

RETURNS INT

AS

BEGIN

DECLARE precio INT;

SELECT precio INTO precio

FROM producto

WHERE nombre=nombreProducto;

RETURN precio;

END;