

Laboratorio 22: Manipulación de datos usando Stored Procedures

Crear procedimientos para manipular datos

```
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'creaMaterial' AND type = 'P')
DROP PROCEDURE creaMaterial
GO

CREATE PROCEDURE creaMaterial
    @uclave NUMERIC(5,0),
    @udescpcion VARCHAR(50),
    @ucosto NUMERIC(8,2),
    @uimpuesto NUMERIC(6,2)
AS
    INSERT INTO Materiales
    VALUES(@uclave, @udescpcion, @ucosto, @uimpuesto)
GO
```

¿Qué hace el primer bloque del código (bloque del IF)?

Busca en la base de datos y si el procedimiento denominado "creaMaterial" ya existe, lo elimina para que no haya ningún conflicto.

¿Para qué sirve la instrucción GO?

Es un comando que indica la finalización de un lote de sentencias, puede ser usado como un reemplazo del (;) pues ayuda a ejecutar varios procesos dentro de una sola consulta.

En este caso manda a ejecutar las líneas previas a este (bloque del IF) antes de que continúe con la ejecución.

¿Explica que recibe como parámetro este Procedimiento y qué tabla modifica?




Recibe los valores a insertar en la tabla para la creación de un material nuevo, es decir, recibe la clave, descripción, costo e impuesto del mismo. Claramente, este proceso modifica la tabla Materiales.

Ejecuta el stored procedure

```
EXECUTE creaMaterial 5000, 'Martillos Acme', 250, 15
```

✓ 1 fila insertada. (La consulta tardó 0.0108 segundos.)

Consulta la tabla "Materiales" para verificar el efecto de esta ejecución.

	Clave	Descripcion	Costo	PorcentajeImpuesto
<input type="checkbox"/>  Editar  Copiar  Borrar	5000	Martillos Acme	250	15.00

Crea procedimientos almacenados con los siguientes objetivos:

- **modificaMaterial** que permite modificar un material que reciba como parámetros las columnas de la tabla materiales y actualice las columnas correspondientes con los valores recibidos, para el registro cuya llave sea la clave que se recibe como parámetro.

```
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'modificaMaterial' AND type = 'P')
DROP PROCEDURE modificaMaterial
GO

CREATE PROCEDURE modificaMaterial
    @uclave NUMERIC(5,0),
    @udescpcion VARCHAR(50),
    @ucosto NUMERIC(8,2),
    @uimpuesto NUMERIC(6,2)
AS
    UPDATE Materiales
    SET descripcion = @udescpcion,
        costo = @ucosto,
        impuesto = @uimpuesto
    WHERE clave = @uclave
GO
```

- **eliminaMaterial** que elimina el registro de la tabla materiales cuya llave sea la clave que se recibe como parámetro.

```

IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'eliminaMaterial' AND type = 'P')
DROP PROCEDURE eliminaMaterial
GO

CREATE PROCEDURE eliminaMaterial
    @uclave NUMERIC(5,0),
AS
    DELETE FROM Materiales
    WHERE clave = @uclave
GO

```

- **Desarrollar los procedimientos (almacenados):** creaProyecto , modificaproyecto y eliminaproyecto, hacer lo mismo para las tablas proveedores y entregan.

```

/* CREA PROYECTO */
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'creaProyecto' AND type = 'P')
DROP PROCEDURE creaProyecto
GO

CREATE PROCEDURE creaProyecto
    @unumero NUMERIC(5,0),
    @udenominacion VARCHAR(50),
AS
    INSERT INTO Proyectos
    VALUES(@unumero, @udenominacion)
GO

/* MODIFICA PROYECTO */
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'modificaProyecto' AND type = 'P')
DROP PROCEDURE modificaProyecto
GO

CREATE PROCEDURE modificaProyecto
    @unumero NUMERIC(5,0),
    @udenominacion VARCHAR(50),
AS
    UPDATE Proyectos
    SET denominacion = @udenominacion,
    WHERE numero = @unumero
GO

/* ELIMINA PROYECTO */

```

```

IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'eliminaProyecto' AND type = 'P')
DROP PROCEDURE eliminaProyecto
GO

CREATE PROCEDURE eliminaProyecto
    @unumero NUMERIC(5,0),
AS
    DELETE FROM Proyectos
    WHERE numero = @unumero
GO

/* CREA PROVEEDOR */
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'creaProveedor' AND type = 'P')
DROP PROCEDURE creaProveedor
GO

CREATE PROCEDURE creaProveedor
    @uRFC CHAR(13),
    @uRazonSocial VARCHAR(50),
AS
    INSERT INTO Proveedores
    VALUES(@uRFC, @uRazonSocial)
GO

/* MODIFICA PROVEEDOR*/
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'modificaProveedor' AND type = 'P')
DROP PROCEDURE modificaProveedor
GO

CREATE PROCEDURE modificaProveedor
    @uRFC CHAR(13),
    @uRazonSocial VARCHAR(50),
AS
    UPDATE Proveedores
    SET RazonSocial = @uRazonSocial,
    WHERE RFC = @uRFC
GO

/* ELIMINA PROVEEDOR */
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'eliminaProveedor' AND type = 'P')
DROP PROCEDURE eliminaProveedor
GO

```

```

CREATE PROCEDURE eliminaProveedor
    @uRFC CHAR(13),
AS
    DELETE FROM Proveedores
    WHERE RFC = @uRFC
GO

/*CREA ENTREGA*/
IF EXISTS (SELECT name FROM sysobjects
            WHERE name = 'creaEntrega' AND type = 'P')
DROP PROCEDURE creaEntrega
GO

CREATE PROCEDURE creaEntrega
    @uclave NUMERIC(5,0),
    @uRFC CHAR(13),
    @unumero NUMERIC(5,0),
    @ufecha DATETIME,
    @ucantidad NUMERIC(6,2)
AS
    INSERT INTO Entregan
    VALUES(@uclave, @uRFC, @unumero, @ufecha, @ucantidad)
GO

/* MODIFICA ENTREGA */
IF EXISTS (SELECT name FROM sysobjects
            WHERE name = 'modificaEntrega' AND type = 'P')
DROP PROCEDURE modificaEntrega
GO

CREATE PROCEDURE modificaEntrega
    @uclave NUMERIC(5,0),
    @uRFC CHAR(13),
    @unumero NUMERIC(5,0),
    @ufecha DATETIME,
    @ucantidad NUMERIC(6,2)
AS
    UPDATE Entregan
    VALUES
        clave = @uclave,
        RFC = @uRFC,
        numero = @unumero,
        cantidad = @ucantidad
    WHERE RFC = @uRFC AND clave = @uclave AND numero = @unumero AND fecha =
@ufecha

```

GO

```
/* ELIMINA ENTREGA */
```

```
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'eliminaEntrega' AND type = 'P')
```

```
DROP PROCEDURE eliminaEntrega
```

GO

```
CREATE PROCEDURE eliminaEntrega
```

```
    @uclave NUMERIC(5,0),
```

```
    @uRFC CHAR(13),
```

```
    @unumero NUMERIC(5,0),
```

```
    @ufecha DATETIME,
```

```
AS
```

```
    DELETE FROM Proveedores
```

```
    WHERE RFC = @uRFC AND clave = @uclave AND numero = @unumero AND fecha =  
    @ufecha
```

GO

Crear procedimientos para realizar consultas con parámetros

```
IF EXISTS (SELECT name FROM sysobjects
           WHERE name = 'queryMaterial' AND type = 'P')
```

```
DROP PROCEDURE queryMaterial
```

GO

```
CREATE PROCEDURE queryMaterial
```

```
    @udescpcion VARCHAR(50),
```

```
    @ucosto NUMERIC(8,2)
```







```
AS
```

```
    SELECT * FROM Materiales
```

```
    WHERE descripcion LIKE '%' + @udescpcion + '%' AND costo > @ucosto
```

GO

```
EXECUTE queryMaterial 'Lad',20
```

				Clave	Descripcion	Costo	PorcentajeImpuesto	
<input type="checkbox"/>		Editar	 Copiar	 Borrar	1080	Ladrillos rojos	50	21.60
<input type="checkbox"/>		Editar	 Copiar	 Borrar	1090	Ladrillos grises	35	21.80

¿Qué recibe como parámetro este procedimiento y qué hace?

Recibe la descripción del material y el costo del mismo, despliega todos los valores de la tabla Materiales para el material descrito donde el costo pasado como parámetro sea menor que el costo registrado en la tabla.

Ejecutar Stored Procedures desde la aplicación cliente

Para uno de tus laboratorios anteriores o proyecto, crea un stored procedure en MySQL de manera similar a como lo acabas de hacer en SQL Server. Posteriormente agrega a tu librería de la capa del modelo, una función de php para ejecutar tu stored procedure y prueba su funcionamiento desde la aplicación. Toma como referencia la guía de [procedimientos almacenados de php](#).

Prueba al menos **3** de los **Stored Procedures** que definiste anteriormente, desde una **aplicación cliente en el lenguaje de tu elección**, queda a tu elección los stored procedures que desees evaluar. Puedes incluirlos en la implementación de algún requisito funcional de tu proyecto.

Conclusión

¿Qué ventajas tienen el utilizar Stored Procedures en una aplicación cliente-servidor? ¿Qué ventajas tiene utilizar SP en un proyecto?

VENTAJA	DESCRIPCION
Velocidad de procesamiento en servidor	Ya que los datos no viajan por la red, solamente los resultados finales lo hacen, la ganancia en la velocidad de respuesta es mucho más eficiente. Por otro lado, generalmente la computadora donde se encuentra el servidor es más poderosa que las computadoras que usan los clientes y por lo tanto procesará los datos más rápidamente.
Seguridad	En un <i>stored procedure</i> solamente puedes usar tablas y columnas que existen en la base de datos, es decir, los errores serán encontrados en tiempo de compilación y no en tiempo de ejecución. Encontrar errores en tiempo de ejecución puede ser muy malo si es un usuario quien los encuentra.
Independencia	Dado que los SP se guardan dentro de la base de datos, estos están disponibles para ser usados por cualquier programa que se conecte a la misma. Dicho programa puede estar escrito en cualquier lenguaje de programación en consecuencia no dependes de un lenguaje de programación. Esto también significa una gran ayuda a la hora de programar en equipo pues cualquier integrante puede utilizar el SP que ya existe inclusive si el lenguaje no es el mismo.
Encapsulamiento	Después de compilar un SP se le pueden dar derechos de ejecución sobre el mismo a algún usuario o a un rol o a un programa, quienes nunca verán el código fuente de dicho SP, simplemente lo usarán, logrando así un encapsulamiento.

Crea un SP que sea de utilidad en tu proyecto, incorpóralo en el reporte.