



## Lab 22 Stored Procedures

▼ Class	Data Bases
🕒 Created	@May 30, 2020 3:59 PM
🔗 Materials	<a href="https://ejuares.bitbucket.io/daw/laboratorios/lab14_store_procedures.html">https://ejuares.bitbucket.io/daw/laboratorios/lab14_store_procedures.html</a>
☑ Reviewed	<input type="checkbox"/>
▼ Type	Homework

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

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

### En tu reporte de este laboratorio, contesta las siguientes preguntas:

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

Verifica que exista un procedimiento que se llama creaMaterial en los objetos del sistema, y de ser así, lo elimina.

#### ¿Para qué sirve la instrucción GO?

Determina que una instrucción ya se terminó y es el equivalente a un ; en un lenguaje como C o C++.

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

Recibe la clave de un material, su descripción, el costo y el impuesto que se le carga, y opera de manera que inserta un registro a la tabla materiales con los datos que se reciben.

## Crear procedimientos para manipular datos

### 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.

```
DROP PROCEDURE IF EXISTS modificaMaterial;
CREATE PROCEDURE modificaMaterial( uclave NUMERIC(5,0), descripcion VARCHAR(50), ucosto NUMERIC (8,2), uimpuesto NUMERIC(6,2))
BEGIN
    UPDATE Materiales SET Descripcion = descripcion, Costo = ucosto, PorcentajeImpuesto = uimpuesto WHERE Clave = uclave;
END;
call modificaMaterial(5015, 'Clavos 1/2 Inch', 20,2);
```

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

```
DROP PROCEDURE IF EXISTS eliminaMaterial;
CREATE PROCEDURE eliminaMaterial( uclave NUMERIC(5,0))
BEGIN
```

```
DELETE FROM Materiales WHERE Clave = uclave;
END;

CALL eliminaMaterial(5015);
```

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

- PROYECTOS

```
DROP PROCEDURE IF EXISTS creaProyecto;
CREATE PROCEDURE creaProyecto( unumero NUMERIC(5,0), udenominacion VARCHAR(50))
BEGIN
    INSERT INTO Proyectos VALUES(unumero, udenominacion);
END;
call creaProyecto(3000, 'Proyect1');

DROP PROCEDURE IF EXISTS modificaProyecto;
CREATE PROCEDURE modificaProyecto( unumero NUMERIC(5,0), udenominacion VARCHAR(50))
BEGIN
    UPDATE Proyectos SET Denominacion = udenominacion WHERE Numero = unumero;
END;
call modificaProyecto(3000, 'Carretera MX-QR0');

DROP PROCEDURE IF EXISTS eliminaProyecto;
CREATE PROCEDURE eliminaProyecto( unumero NUMERIC(5,0))
BEGIN
    DELETE FROM Proyectos WHERE Numero = unumero;
END;

call eliminaProyecto(3000);
```

- PROVEEDORES

```
DROP PROCEDURE IF EXISTS creaProveedor;
CREATE PROCEDURE creaProveedor( urfc CHAR(13), urazonsocial VARCHAR(50))
BEGIN
    INSERT INTO Proveedores VALUES(urfc, urazonsocial);
END;
call creaProveedor('PEMEXCORP23121', 'PEMEX');

DROP PROCEDURE IF EXISTS modificaProveedor;
CREATE PROCEDURE modificaProveedor( urfc CHAR(13), urazonsocial VARCHAR(50))
BEGIN
    UPDATE Proveedores SET RazonSocial = urazonsocial WHERE RFC = urfc;
END;
call modificaProveedor('PEMEXCORP23121', 'PEMEX');

DROP PROCEDURE IF EXISTS eliminaProveedor;
CREATE PROCEDURE eliminaProveedor( urfc CHAR(13))
BEGIN
    DELETE FROM Proveedores WHERE RFC = urfc;
END;
call eliminaProveedor('PEMEXCORP23121');
```

- ENTREGAS

```
DROP PROCEDURE IF EXISTS creaEntrega;
CREATE PROCEDURE creaEntrega(uclave NUMERIC(5,0), urfc CHAR(13), unumero NUMERIC(5,0), ufecha DATETIME, ucantidad NUMERIC(8,2))
BEGIN
    INSERT INTO Entregan VALUES(uclave, urfc, unumero, ufecha, ucantidad);
END;
call creaEntrega(1000, 'AAAA123438', 5015, '2020-03-03 00:00:00',100);

DROP PROCEDURE IF EXISTS modificaEntrega;
CREATE PROCEDURE modificaEntrega(uclave NUMERIC(5,0), urfc CHAR(13), unumero NUMERIC(5,0), ufecha DATETIME, ucantidad NUMERIC(8,2))
BEGIN
    UPDATE Entregan SET Cantidad = ucantidad WHERE Clave = uclave AND RFC = urfc AND Numero = unumero AND Fecha = ufecha;
END;

call modificaEntrega(1000, 'AAAA123438', 5015, '2020-03-03 00:00:00',150);

DROP PROCEDURE IF EXISTS eliminaEntrega;
DELIMITER //
CREATE PROCEDURE eliminaEntrega(uclave NUMERIC(5,0), urfc CHAR(13), unumero NUMERIC(5,0), ufecha DATETIME)
BEGIN
```

```
DELETE FROM Entregan WHERE Clave = uclave AND RFC = urfc AND Numero = unumero AND Fecha = ufecha;
END;
//
DELIMITER ;

call eliminaEntrega(1000, 'AAAA123438', 5015, '2020-03-03 00:00:00');
```

## Crear procedimientos para realizar consultas con parámetros

Define el siguiente stored procedure en tu base de datos:

```
IF EXISTS (SELECT name FROM sysobjects
WHERE name = 'queryMaterial' AND type = 'P')
DROP PROCEDURE queryMaterial
GO
CREATE PROCEDURE queryMaterial
@descripcion VARCHAR(50),
@ucosto NUMERIC(8,2)
AS
SELECT * FROM Materiales WHERE descripcion
LIKE '%' + @descripcion + '%' AND costo > @ucosto
GO
```

Ejecútalo con la siguiente instrucción:

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

Explica en tu reporte qué recibe como parámetro este procedimiento y qué hace.

Recibe una descripcion y costo y regresa los materiales que tienen esos aspectos.

## 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](#)

.



En PHPMyAdmin no se pueden implementar stored procedures, pero fueron usados en el examen parcial 2

Contesta en tu reporte las siguientes preguntas de reflexión:

**¿Qué ventajas tienen el utilizar Stored Procedures en una aplicación cliente-servidor?**

Permite hacer varias interacciones y operaciones en la base de datos en una sola acción, lo cual permite que haya muchas menos conexiones cuando hay que hacerlas más de 3 veces, o muy seguido.

**¿Qué ventajas tiene utilizar SP en un proyecto?**

La ventaja que yo puedo observar es que añade seguridad en cuanto a la estructura de datos, porque al solo usar CALL a una función, las operaciones de inserción se hacen por el propio DBMS y no por los desarrolladores

**Para reforzar el entendimiento de los SP, crea un SP que sea de utilidad en tu proyecto (en caso de usar otra tecnología, se deberá asumir). Incorpora dicho SP en tu reporte.**

```
DROP PROCEDURE IF EXISTS insertarPuesto;
CREATE PROCEDURE insertarPuesto(@unombre VARCHAR(30))
BEGIN
INSERT INTO puesto VALUES nombre=(@unombre);
END
```

