



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



BASE DE DATOS

PROFESOR:

Ing. Yadira Franco R

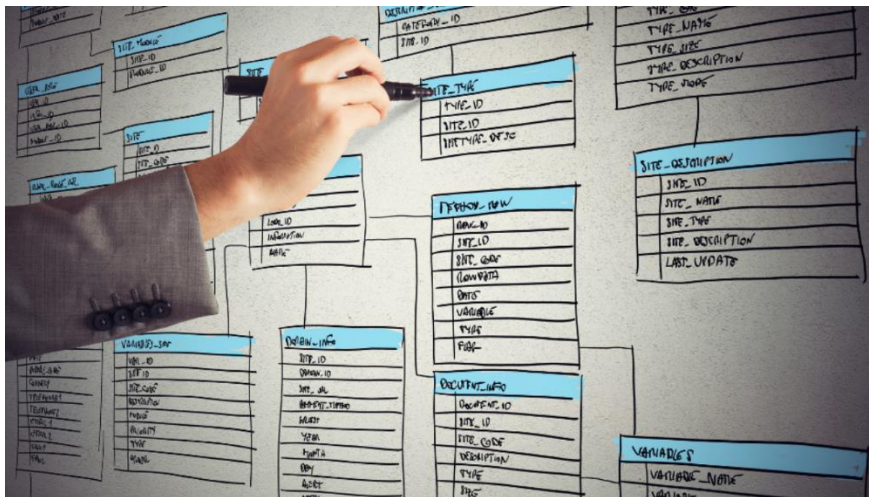
PERÍODO ACADÉMICO:

2024-B

TAREA

TÍTULO:

INVESTIGACIÓN Y PRACTICA



Estudiante

David Muela

2024-B

INVESTIGAR QUE SON Procedimientos Almacenados en Bases de Datos

- **Entender qué son los procedimientos almacenados y cómo funcionan.**

Es un conjunto de instrucciones que se pueden llamar desde otras consultas o desde otros procedimientos almacenados

Introducción a los Procedimientos Almacenados **Sql Server**

1. Concepto y Beneficios de los Procedimientos Almacenados

- **Explicación:** Los procedimientos almacenados son conjuntos de instrucciones SQL que se guardan y ejecutan en el servidor de base de datos. Permiten ejecutar operaciones complejas, con seguridad, rendimiento optimizado y reutilización de código.
- **Beneficios:**
 - Reutilización de código.
 - Mejora en la seguridad (al evitar inyecciones SQL).
 - Optimización en el rendimiento de consultas frecuentes.
 - Consistencia en las operaciones realizadas.

2. ESPECIFICAR LA Sintaxis Básica de un Procedimiento Almacenado

- **Explicación:** El delimitador se cambia temporalmente para permitir el uso de ; dentro del procedimiento.

1. Creación de la base de datos

```
CREATE DATABASE DBProcedimiento;  
use DBProcedimiento;
```

2. Creación de la tabla clientes

```
CREATE TABLE cliente(  
    ClienteID INT IDENTITY(1,1) PRIMARY KEY,  
    Nombre NVARCHAR(100),  
    Estatura DECIMAL(5,2),  
    FechaNacimiento DATE,  
    Sueldo DECIMAL(10,2),  
    Edad INT  
);
```

3. Insertar datos clientes

```

INSERT INTO cliente (Nombre, Estatura, FechaNacimiento, Sueldo,Edad) VALUES
('David Muela', 1.73, '1990-05-14', 4500.50,20),
('Laura García', 1.62, '1985-08-22', 3700.00,15),
('Carlos Pérez', 1.80, '1992-11-10', 5200.75,60),
('Ana López', 1.55, '2000-02-18', 2800.40,39),
('Luis Martínez', 1.67, '1988-09-05', 3100.20,41);

```

3. Ejercicio 1: Crear un procedimiento simple que seleccione datos de la tabla cliente

```

CREATE PROCEDURE MostrarInformacion
AS
BEGIN
    SELECT * FROM cliente;
END;

EXEC MostrarInformacion;

```

Explicación: Para inicializar un procedimiento en SQL Server se necesita **CREATE PROCEDURE (Nombre variable) AS** para inicia con **BEGIN** dentro de begin aplicamos las consultas o lógica a realizar finalizando con eso terminamos con **END;** para todos los casos

Para indicar el procedimiento realizamos un **EXEC NombreVariable**

Inserción, Actualización y Eliminación de Datos

1. Procedimiento de Inserción (INSERT)

- Iniciamos creando nuestro procedimiento y para agregar un nuevo registro toca agregar cada columna que deseamos agregar, agregamos todos para evitar valores nulos

```

CREATE PROCEDURE InsertarDatos
    @Nombre NVARCHAR(100),
    @Estatura DECIMAL(5,2),
    @FechaNacimiento DATE,
    @Sueldo DECIMAL(10,2),
    @Edad INT
AS
BEGIN
    INSERT INTO cliente (Nombre, Estatura, FechaNacimiento, Sueldo,Edad)
    VALUES (@Nombre,@Estatura,@FechaNacimiento,@Sueldo,@Edad);
END;

```

- Insertamos los datos

```

EXEC InsertarDatos
@Nombre = 'Pepe Perez',
@Estatura = 1.75,
@FechaNacimiento = '1985-03-20',
@Sueldo = 4000.50,
@Edad = 24;

```

Resultado:

	ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo	Edad
1	1	David Muela	1.73	1990-05-14	4500.50	20
2	2	Laura García	1.62	1985-08-22	3700.00	15
3	3	Carlos Pérez	1.80	1992-11-10	5200.75	60
4	4	Ana López	1.55	2000-02-18	2800.40	39
5	5	Luis Martínez	1.67	1988-09-05	3100.20	41
6	6	Pepe Perez	1.75	1985-03-20	4000.50	24

2. Procedimiento de Actualización (UPDATE)

Actualizar la edad de un cliente específico:

- Inicializamos creando nuestro procedimiento la cual vamos a necesitar la columna ClienteID para identificar el cliente y la edad para actualizar la edad a nuevo valor

```

CREATE PROCEDURE ActualizarDatos
    @ClienteID INT,
    @Edad INT
AS
BEGIN
    UPDATE cliente SET Edad = @Edad WHERE ClienteID = @ClienteID;
END;

```

- Actualizamos la Edad mediante el ID del cliente

```

EXEC ActualizarDatos
@ClienteID = 1,
@Edad = 35;

```

Resultado:

	CienteID	Nombre	Estatura	FechaNacimiento	Sueldo	Edad
1	1	David Muela	1.73	1990-05-14	4500.50	35
2	2	Laura García	1.62	1985-08-22	3700.00	15

3. Procedimiento de Eliminación (DELETE)

Eliminar un cliente de la base de datos usando su CienteID:

- Creamos nuestro procedimiento para eliminar la cual solo requerimos el ID del cliente para proceder con la eliminación del Cliente mediante su ID

```

CREATE PROCEDURE EliminarCliente
    @ClienteID INT
AS
BEGIN
    DELETE FROM cliente
    WHERE CienteID = @ClienteID;
END;

EXEC EliminarCliente
    @ClienteID = 1;

```

- Indicamos la ID del usuario a eliminar

```

EXEC EliminarCliente
    @ClienteID = 1;

```

Resultado:

	CienteID	Nombre	Estatura	FechaNacimiento	Sueldo	Edad
1	2	Laura García	1.62	1985-08-22	3700.00	15
2	3	Carlos Pérez	1.80	1992-11-10	5200.75	60
3	4	Ana López	1.55	2000-02-18	2800.40	39

Introducción a Condiciones en Procedimientos Almacenados

Uso de Condicionales (IF)

El uso de condicionales dentro de los procedimientos es fundamental para tomar decisiones basadas en los datos.

Verifica si la edad de un cliente es mayor o igual a 22:

- Inicializamos nuestro procedimiento aplicando lógica donde la edad se verifica que sea mayor o igual a 22 y se mostrara todo los registros de clientes que tenga la edad mayor a 22 caso contrario enviara un mensaje de que la edad mínima deber ser mayor o igual a 22 a buscar

```

CREATE PROCEDURE VerificarEdad
    @EdadMinima INT
AS
BEGIN
    IF @EdadMinima >= 22
    BEGIN
        SELECT *
        FROM cliente
        WHERE Edad >= @EdadMinima;
    END
    ELSE
    BEGIN
        PRINT 'La edad minima debe ser mayor o igual a 22.';
    END
END;

```

- Indicamos la edad

```

EXEC VerificarEdad
    @EdadMinima = 22;

```

Resultado:

	ClienteID	Nombre	Estatura	FechaNacimiento	Sueldo	Edad
1	3	Carlos Pérez	1.80	1992-11-10	5200.75	60
2	4	Ana López	1.55	2000-02-18	2800.40	39
3	5	Luis Martínez	1.67	1988-09-05	3100.20	41
4	6	Pepe Perez	1.75	1985-03-20	4000.50	24

Creación de la Tabla de Órdenes CON RELACIÓN CON EL CLIENTE - FORANEA

Para almacenar las órdenes de los clientes, se debe crear la tabla **ordenes**:

```
-- CREAR TABLA ORDENES
CREATE TABLE ordenes (
    OrdenID INT IDENTITY(1,1) PRIMARY KEY,
    ClienteID INT,
    Detalle NVARCHAR(255),
    Fecha DATE,
    FOREIGN KEY (ClienteID) REFERENCES cliente(ClienteID)
);
```

- Procedimientos de Órdenes -Insertar Orden
 - Inicializamos nuestro procedimiento, agregamos las columnas para poder insertar nuevos valores, luego usamos EXEC para llamar el procedimiento y agregar el nuevo registro

```
-- INSERTAR UNA ORDEN CON PROCEDIMIENTO

CREATE PROCEDURE InsertarOrdenes
    @ClienteID INT,
    @Detalle NVARCHAR(255),
    @Fecha DATE
AS
BEGIN
    INSERT INTO ordenes (ClienteID, Detalle, Fecha)
    VALUES (@ClienteID, @Detalle, @Fecha);
END;

EXEC InsertarOrdenes
    @ClienteID = 2,
    @Detalle = 'Compra de electrodomésticos',
    @Fecha = '2024-12-17';

SELECT * FROM ordenes;
```

20 %

Results Messages

	OrdenID	ClienteID	Detalle	Fecha
1	1	2	Compra de electrodomésticos	2024-12-17

- Procedimientos Actualizar Orden
 - Inicializamos nuestro procedimiento, actualizar la columna mediante el id de la orden y detalle de la orden

```
-- ACTUALIZAR UNA ORDEN CON PROCEDIMIENTO

CREATE PROCEDURE ActualizarOrden
    @OrdenID INT,
    @NuevoDetalle NVARCHAR(255)
AS
BEGIN
    UPDATE ordenes SET Detalle = @NuevoDetalle WHERE OrdenID = @OrdenID;
END;

EXEC ActualizarOrden
    @OrdenID = 1,
    @NuevoDetalle = 'Compra de muebles';
```

Results Messages

	OrdenID	CienteID	Detalle	Fecha
1	1	2	Compra de muebles	2024-12-17

- Procedimientos Eliminar Orden
 - Inicializamos nuestro procedimiento, para eliminar una columna mediante el id para poder eliminar la orden correctamente

```
-- ELIMINAR UNA ORDEN CON PROCEDIMIENTO

CREATE PROCEDURE EliminarOrden
    @OrdenID INT
AS
BEGIN
    DELETE FROM ordenes
    WHERE OrdenID = @OrdenID;
END;

EXEC EliminarOrden
    @OrdenID = 1;

SELECT * FROM ordenes;
```