

## Actividad 5: Stored Procedures

NOMBRE DEL ALUMNO: GARCÍA QUIROZ GUSTAVO IVAN

GRUPO: 3CV2

BASE DE DATOS

29/11/2023

# Actividad 5: Stored Procedures

## Instrucciones

Realizar los ejercicios que vienen en el script, colocar:

- Portada
- Enunciado del ejercicio
- Código de Stored Procedure
- Ejecución (Capturas de pantalla)

--\* Instrucciones, si en la base de datos no existen valores que devolver, insertar registro que cumplan con la condición, para ver la funcionalidad del SP.

--1. Obtener todos los empleados que se dieron de alta entre una determinada fecha inicial y  
--fecha final y que pertenecen a un determinada aerolinea usando un store  
procedure.

```
CREATE PROCEDURE ObtenerEmpleadosPorFechaYAerolinea
    @FechaInicio datetime,
    @FechaFinal datetime,
    @AerolineaId int
AS
BEGIN
    SELECT *
    FROM Empleado
    WHERE fechaIngreso BETWEEN @FechaInicio AND @FechaFinal
    AND AerolineaId = @AerolineaId
END
```

EXEC ObtenerEmpleadosPorFechaYAerolinea '2000-01-01', '2022-12-31', 3

	idEmpleado	idPersona	idTipoEmpleado	Aerolineaid	fechaIngreso	sueldo	idJefe
1	1	4	1	3	2010-07-30 00:00:00.000	47500.00	5
2	5	13	5	3	2008-09-07 00:00:00.000	45700.00	NULL
3	9	2	3	3	2010-02-03 00:00:00.000	17000.00	5
4	10	6	3	3	2010-02-03 00:00:00.000	14700.00	5

--2. Crear un procedimiento que inserte un empleado y valide su previa existencia, si existe no lo inserte.

```
CREATE PROCEDURE InsertarEmpleado
    @idPersona int,
    @idTipoEmpleado int,
    @AerolineaId int,
    @fechaIngreso datetime,
    @sueldo money,
    @idJefe int
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Empleado WHERE idPersona = @idPersona)
```

```

BEGIN
    INSERT INTO Empleado (idPersona, idTipoEmpleado, AerolineaId, fechaIngreso,
sueldo, idJefe)
    VALUES (@idPersona, @idTipoEmpleado, @AerolineaId, @fechaIngreso, @sueldo,
@idJefe)
END
END

```

```
EXEC InsertarEmpleado 10, 3, 3, '2022-01-01', 50000, 5
```

	idEmpleado	idPersona	idTipoEmpleado	AerolineaId	fechaIngreso	sueldo	idJefe
1	1	4	1	3	2010-07-30 00:00:00.000	47500.00	5
2	2	17	1	4	2011-12-10 00:00:00.000	39050.00	6
3	3	19	1	5	2010-01-01 00:00:00.000	40000.00	7
4	5	13	5	3	2008-09-07 00:00:00.000	45700.00	NULL
5	6	14	5	4	2008-01-01 00:00:00.000	49200.00	NULL
6	7	20	5	5	2005-11-15 00:00:00.000	45900.00	NULL
7	9	2	3	3	2010-02-03 00:00:00.000	17000.00	5
8	10	6	3	3	2010-02-03 00:00:00.000	14700.00	5
9	11	10	3	3	2022-01-01 00:00:00.000	50000.00	5

--3. Crear un procedimiento que recupere el nombre de aerolínea, cantidad de empleados y

--idEmpleado, id TipoEmpleado a partir del idAerolinea.

```

CREATE PROCEDURE ObtenerInfoAerolinea
    @AerolineaId int
AS
BEGIN
    SELECT a.Nombre, COUNT(e.idEmpleado) as CantidadEmpleados, e.idEmpleado,
e.idTipoEmpleado
    FROM Aerolinea a
    JOIN Empleado e ON a.AerolineaId = e.AerolineaId
    WHERE a.AerolineaId = @AerolineaId
    GROUP BY a.Nombre, e.idEmpleado, e.idTipoEmpleado
END
EXEC ObtenerInfoAerolinea 3

```

	Nombre	CantidadEmpleados	idEmpleado	idTipoEmpleado
1	Volaris	1	1	1
2	Volaris	1	5	5
3	Volaris	1	9	3
4	Volaris	1	10	3
5	Volaris	1	11	3

--4. Crear un procedimiento igual que el anterior, pero que recupera también las personas que

--trabajan en dicha aerolínea, pasandole como parámetro el nombre y apellido.

```

CREATE PROCEDURE ObtenerInfoAerolineaYPersona
    @nombre varchar(50),
    @apellido varchar(50)
AS
BEGIN
    SELECT a.Nombre, COUNT(e.idEmpleado) as CantidadEmpleados, e.idEmpleado,
e.idTipoEmpleado, p.nombre, p.paterno
    FROM Aerolinea a
    JOIN Empleado e ON a.AerolineaId = e.AerolineaId

```

```

JOIN Persona p ON e.idPersona = p.IdPersona
WHERE p.nombre = @nombre AND p.paterno = @apellido
GROUP BY a.Nombre, e.idEmpleado, e.idTipoEmpleado, p.nombre, p.paterno
END

```

EXEC ObtenerInfoAerolineaYPersona 'Eduardo', 'Ortiz'

	Nombre	CantidadEmpleados	idEmpleado	idTipoEmpleado	nombre	paterno
1	Volaris	1	1	1	Eduardo	Ortiz

--5. Crear un procedimiento para devolver sueldo, tipo de empleado y jefe pasándole el apellido.

```

CREATE PROCEDURE ObtenerInfoEmpleadoPorApellido
    @apellido varchar(50)
AS
BEGIN
    SELECT e.sueldo, te.tipoEmpleado, j.nombre + ' ' + j.paterno as NombreJefe
    FROM Empleado e
    JOIN Persona p ON e.idPersona = p.IdPersona
    JOIN TipoEmpleado te ON e.idTipoEmpleado = te.idTipoEmpleado
    LEFT JOIN Empleado ej ON e.idJefe = ej.idEmpleado
    LEFT JOIN Persona j ON ej.idPersona = j.IdPersona
    WHERE p.paterno = @apellido
END

```

EXEC ObtenerInfoEmpleadoPorApellido 'Ortiz'

	sueldo	tipoEmpleado	NombreJefe
1	47500.00	Piloto	Carlos Tadeo

--6. Crear un procedimiento igual al anterior, pero si no le pasamos ningún valor, --mostrará los datos de todos los empleados, incluyendo su nombre, dirección, teléfono, etc.

```

CREATE PROCEDURE ObtenerInfoEmpleadoOpcional
    @apellido varchar(50) = NULL
AS
BEGIN
    SELECT p.nombre, p.paterno, p.materno, p.direccionId, p.telefono, e.sueldo,
    te.tipoEmpleado, j.nombre + ' ' + j.paterno as NombreJefe
    FROM Empleado e
    JOIN Persona p ON e.idPersona = p.IdPersona
    JOIN TipoEmpleado te ON e.idTipoEmpleado = te.idTipoEmpleado
    JOIN Direccion d ON p.direccionId = d.direccionId
    LEFT JOIN Empleado ej ON e.idJefe = ej.idEmpleado
    LEFT JOIN Persona j ON ej.idPersona = j.IdPersona
    WHERE (@apellido IS NULL OR p.paterno = @apellido)
END

```

EXEC ObtenerInfoEmpleadoOpcional 'Ortiz'

	nombre	paterno	materno	direccionId	telefono	sueldo	tipoEmpleado	NombreJefe
1	Eduardo	Ortiz	NULL	8	53123212231	47500.00	Piloto	Carlos Tadeo

--7. Crear un procedimiento para mostrar el sueldo, tipo de empleado y nombre de la aerolínea

--de todos los empleados que contengan en su apellido el valor que le pasemos como --parámetro.

```

CREATE PROCEDURE ObtenerInfoEmpleadoPorApellidoX
    @valor varchar(50)

```

```

AS
BEGIN
    SELECT e.sueldo, te.tipoEmpleado, a.Nombre
    FROM Empleado e
    JOIN Persona p ON e.idPersona = p.IdPersona
    JOIN TipoEmpleado te ON e.idTipoEmpleado = te.idTipoEmpleado
    JOIN Aerolinea a ON e.AerolineaId = a.AerolineaId
    WHERE p.paterno LIKE '%' + @valor + '%'
END

```

```

EXEC ObtenerInfoEmpleadoPorApellido 'Ortiz'

```

	sueldo	tipoEmpleado	NombreJefe
1	47500.00	Piloto	Carlos Tadeo

--8. Crear un procedimiento para mostrar las ventas de un rango de fechas,  
--indicando fecha compra, vendedor, aerolinea, idvuelo, número de boletos, total de compra , pasando  
--como parámetro el rango de fechas.

```

CREATE PROCEDURE ObtenerVentasPorRangoFechaX
    @fechaInicio datetime,
    @fechaFin datetime
AS
BEGIN
    SELECT v.fechaCompra, ve.personaId, a.Nombre, v.idVuelo, v.numAsientos,
    v.numAsientos * vu.precioPorAsiento as totalCompra
    FROM Venta v
    JOIN Vendedor ve ON v.idVendedor = ve.vendedorId
    JOIN Vuelo vu ON v.idVuelo = vu.idVuelo
    JOIN Avion av ON vu.matriculaAvion = av.matricula
    JOIN Aerolinea a ON av.aerolineaId = a.AerolineaId
    WHERE v.fechaCompra BETWEEN @fechaInicio AND @fechaFin
END

```

```

EXEC ObtenerVentasPorRangoFechaX '2014-01-01', '2022-12-31'

```

	fechaCompra	personalId	Nombre	idVuelo	numAsientos	totalCompra
1	2014-07-01 00:00:00.000	1	Volaris	2	2	6000.00
2	2014-08-07 00:00:00.000	3	Volaris	7	1	3670.00
3	2014-05-07 00:00:00.000	3	Volaris	9	4	24000.00
4	2014-03-24 00:00:00.000	11	Volaris	2	2	6000.00
5	2014-02-26 00:00:00.000	11	Volaris	7	1	3670.00
6	2014-02-26 00:00:00.000	11	Volaris	10	2	5000.00
7	2014-01-01 00:00:00.000	3	Volaris	8	4	10360.00
8	2014-02-20 00:00:00.000	1	Volaris	9	2	12000.00
9	2014-12-23 00:00:00.000	1	Volaris	12	1	3000.00
10	2014-08-30 00:00:00.000	1	InterJet	13	4	20000.00
11	2014-02-10 00:00:00.000	3	InterJet	15	2	8064.00
12	2014-03-02 00:00:00.000	1	InterJet	15	2	8064.00
13	2014-05-09 00:00:00.000	3	InterJet	15	4	16128.00
14	2014-08-30 00:00:00.000	3	InterJet	15	4	16128.00
15	2014-07-23 00:00:00.000	3	Volaris	12	4	12000.00
16	2014-01-01 00:00:00.000	1	Volaris	8	4	10360.00
17	2014-09-02 00:00:00.000	11	American Airlines	20	3	3972.00
18	2014-09-02 00:00:00.000	11	American Airlines	19	3	19020.00
19	2014-11-25 00:00:00.000	1	Volaris	8	4	10360.00
20	2014-12-31 00:00:00.000	1	Volaris	8	3	7770.00

--9. Crear un procedimiento para mostrar los datos completos de los 3 clientes que más han comprado en un determinado año,

```
--pasando como parámetro el tipo de cliente y el año.*
CREATE PROCEDURE ObtenerTopClientesPorAño
    @tipoCliente varchar(50),
    @año int
AS
BEGIN
    SELECT TOP 3 p.nombre, p.paterno, p.materno, p.telefono, SUM(v.numAsientos) as
totalCompras
    FROM Cliente c
    JOIN Persona p ON c.personaId = p.IdPersona
    JOIN TipoCliente tc ON c.idTipoCliente = tc.idTipoCliente
    JOIN Venta v ON c.clienteId = v.idCliente
    WHERE tc.tipoCliente = @tipoCliente AND YEAR(v.fechaCompra) = @año
    GROUP BY p.nombre, p.paterno, p.materno, p.telefono
    ORDER BY totalCompras DESC
END
EXEC ObtenerTopClientesPorAño 'Nuevo', 2014
```

	nombre	paterno	materno	telefono	totalCompras
1	Efrain	Bautista	Beltran	55231234	7
2	Nubia	García	NULL	53679754	3
3	Adrian	Lara	Parez	56372823	2

--10. Crear un procedimiento para mostrar el total de comisión a cobrar por un empleado, de las ventas realizadas en un determinado mes,  
--pasando como parámetro su IdEmpleado, el mes y año la de venta.\*

```
CREATE PROCEDURE ObtenerComisionPorMes
    @idEmpleado int,
    @mes int,
    @año int
AS
BEGIN
    SELECT SUM(v.numAsientos * ve.porcentajeComision / 100) as totalComision
    FROM Venta v
    JOIN Vendedor ve ON v.idVendedor = ve.vendedorId
    JOIN Empleado e ON ve.personaId = e.idPersona
    WHERE e.idEmpleado = @idEmpleado AND MONTH(v.fechaCompra) = @mes AND
YEAR(v.fechaCompra) = @año
END
EXEC ObtenerComisionPorMes 4, 3, 2014
```

	totalComision
1	NULL

--11. Crear un procedimiento para mostrar, cuántos vuelos a realizado el "personal de vuelo" de una aerolinea en especifico, mostrar Aerolínea,nombre completo del --personal, TipoEmpleado, número de vuelos. Pasando como parámetro el nombre de la Aerolínea.

```
CREATE PROCEDURE ObtenerVuelosPorAerolineaX
    @nombreAerolinea varchar(50)
AS
BEGIN
    SELECT a.Nombre, p.nombre, p.paterno, p.materno, te.tipoEmpleado,
COUNT(ve.idVuelo) as numeroVuelos
    FROM Empleado e
    JOIN Persona p ON e.idPersona = p.IdPersona
    JOIN TipoEmpleado te ON e.idTipoEmpleado = te.idTipoEmpleado
```

```

JOIN VueloEmpleado ve ON e.idEmpleado = ve.idEmpleado
JOIN Aerolinea a ON e.AerolineaId = a.AerolineaId
WHERE a.Nombre = @nombreAerolinea AND te.esDeVuelo = 1
GROUP BY a.Nombre, p.nombre, p.paterno, p.materno, te.tipoEmpleado
END
EXEC ObtenerVuelosPorAerolineaX 'Volaris'

```

	Nombre	nombre	paterno	materno	tipoEmpleado	numeroVuelos
1	Volaris	Beatriz	Pedraza	López	Aeromosa	1
2	Volaris	Edith	Parez	Granados	Aeromosa	1
3	Volaris	Eduardo	Ortiz	NULL	Piloto	2