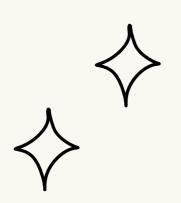
INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

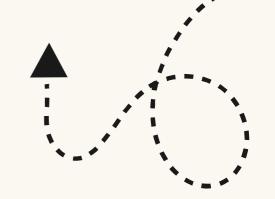
ACTIVIDAD 6: SCRIPT STORED PROCEDURES

DATA BASE SELECTED TOPICS 3CV14

Martinez Cruz José Antonio 2019630583

Bocanegra Heziquio Yestlanezi 2019340090



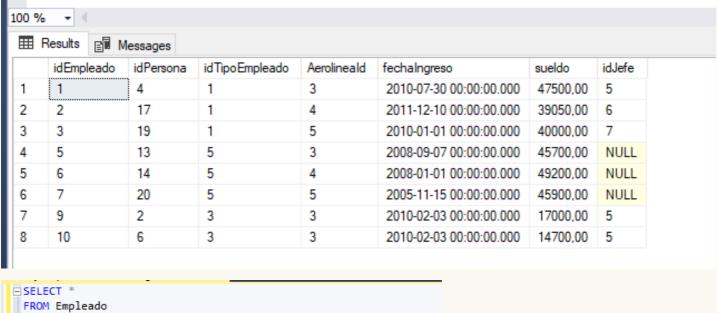


Profesora: Galeana Chávez Ing. María del Rosario



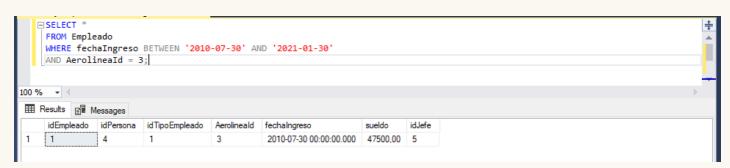
Obtener todos los empleados que se dieron de alta entre una determinada fecha inicial y fecha final y que pertenecen a un determinada aerolinea.

SELECT * FROM EMPLEADO;



```
WHERE fechaIngreso BETWEEN 'fechaInicial' AND 'fechaFinal'
AND AerolineaId = 'AerolineaId';
```

REEMPLAZAMOS LAS FECHAS POR LA QUE QUEREMOS OBTENER, SELECCIONANDO POR EL NÚMERO DE AEROLÍNEA QUE DESEAMOS TENER.





UTILIZAMOS EL SIGUIENTE CÓDIGO PARA QUE SE INSERTEN LOS DATOS DE EMPLEADO.

```
□CREATE PROCEDURE insertar_empleado
      @nombre varchar(50)
      Opaterno varchar(50).
      @materno varchar(50).
      @calle varchar(50).
      @numero varchar(10)
      @colonia varchar(50),
      @codigopostal varchar(10),
      @telefono varchar(20),
      @nacimiento date.
      @sexo bit,
      @tipoempleado varchar(50),
      @aerolinea varchar(50),
      @fechaingreso date
      @sueldo decimal(10,2),
      @idjefe int
 AS
BEGIN
     -- Verificar si el empleado ya existe
IF EXISTS (SELECT * FROM Empleado e
                  JOIN Persona p ON e.Idpersona = p.Idpersona
                  WHERE p.nombre = @nombre
                    AND p.paterno = @paterno
                    AND p.materno = @materno)
      BEGIN
          PRINT 'El empleado ya existe.'
          RETURN
```

```
÷
        -- Obtener el ID de la aerolínea
        DECLARE @aerolineaId int
        SELECT @aerolineaId = AerolineaID FROM Aerolinea WHERE nombre = @aerolinea
        -- Obtener el ID del tipo de empleado
        DECLARE @tipoEmpleadoId int
        SELECT @tipoEmpleadoId = IdTipoEmpleado FROM TipoEmpleado WHERE TipoEmpleado = @tipoempleado
        -- Obtener el ID de la dirección o insertarla si no existe
        DECLARE @direccionID int
        IF EXISTS (SELECT * FROM Direccion WHERE calle = @calle AND numero = @numero AND colonia = @colonia AND codigopostal = @codi
        BEGIN
            SELECT @direccionID = direccionID FROM Direccion WHERE calle = @calle AND numero = @numero AND colonia = @colonia AND co
        FND
        ELSE
        BEGIN
            INSERT INTO Direccion (calle, numero, colonia, codigopostal)
            VALUES (@calle, @numero, @colonia, @codigopostal)
            SET @direccionID = SCOPE_IDENTITY()
        END
        -- Insertar la persona
        INSERT INTO Persona (nombre, paterno, materno, direccionID, telefono, nacimiento, sexo)
        VALUES (@nombre, @paterno, @materno, @direccionID, @telefono, @nacimiento, @sexo)
        -- Obtener el ID de la persona insertada
        DECLARE @personald int
        SELECT @personald = SCOPE_IDENTITY()
00 %

    Messages

    Query executed successfully

                                                             5CG3222WQ7\MSSQLSERVER01 (1... | 5CG3222WQ7\Yestlanezi ... | Aerolineas | 00:00:00 | 0 rows
```

```
SQLQuery1.sql - 5C...ezi Bocanegra (82))* 💠 🗶
        DECLARE @tipoEmpleadoId int
        SELECT @tipoEmpleadoId = IdTipoEmpleado FROM TipoEmpleado WHERE TipoEmpleado = @tipoempleado
        -- Obtener el ID de la dirección o insertarla si no existe
        DECLARE @direccionID int
        IF EXISTS (SELECT * FROM Direccion WHERE calle = @calle AND numero = @numero AND colonia = @colonia AND codigopostal = @codi
            SELECT @direccionID = direccionID FROM Direccion WHERE calle = @calle AND numero = @numero AND colonia = @colonia AND co
        END
        ELSE
        BEGIN
            INSERT INTO Direccion (calle, numero, colonia, codigopostal)
            VALUES (@calle, @numero, @colonia, @codigopostal)
            SET @direccionID = SCOPE IDENTITY()
        END
        -- Insertar la persona
        INSERT INTO Persona (nombre, paterno, materno, direccionID, telefono, nacimiento, sexo)
        VALUES (@nombre, @paterno, @materno, @direccionID, @telefono, @nacimiento, @sexo)
         -- Obtener el ID de la persona insertada
        DECLARE @personald int
        SELECT @personaId = SCOPE IDENTITY()
        -- Insertar el empleado
        INSERT INTO Empleado (Idpersona, IdTipoEmpleado, AerolineaID, fechaingreso, sueldo, idjefe)
        VALUES (@personaId, @tipoEmpleadoId, @aerolineaId, @fechaingreso, @sueldo, @idjefe)
    FND
```

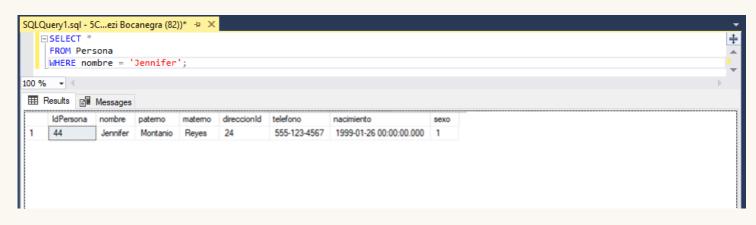
UNA VEZ QUE INSERTAMOS LOS DATOS DEL EMPLEADO, AL INTENTARLO NUEVAMENTE NO LO PERMITE, OBTENIENDO UN MENSAJE DE "EL EMPLEADO YA EXISTE"

```
EXEC insertar_empleado
         @nombre = 'Jennifer'
         @paterno = 'Montanio',
         @materno = 'Reyes',
         @calle = 'Jacarandas',
         @numero = '555',
         @colonia = 'Del bosque',
         @codigopostal = '07210',
         @telefono = '555-123-4567',
         @nacimiento = '1999-01-26',
         @sexo = '1',
         @tipoempleado = 'Piloto',
         @aerolinea = 'Volaris',
         @fechaingreso = '2023-05-14',
         @sueldo = 25000.00,
         @idjefe = NULL;
100 % -

    Messages

   El empleado ya existe.
   Completion time: 2023-05-14T23:40:52.6087298-06:00
```

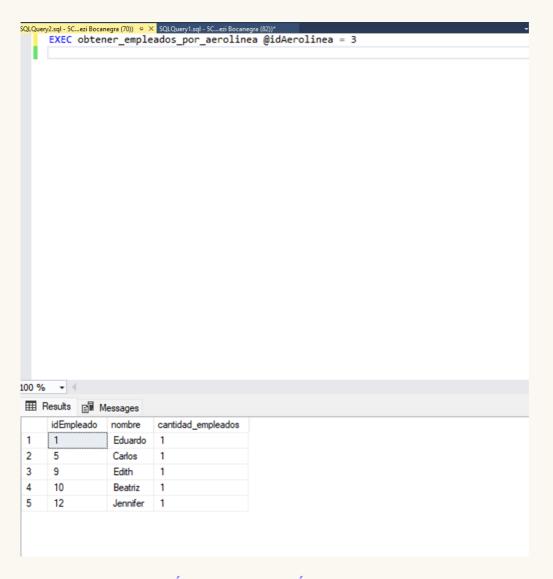
COLOCAMOS LA NUEVA CONSULTA, OBTENEMOS LOS DATOS DEL EMPLEADO QUE HEMOS REGISTRADO.





Crear un procemimiento que recupere el nombre de aerolínea, cantidad de empleados y idEmpleado, id TipoEmpleado a partir del idAerolinea.

```
-- -----
SET ANSI NULLS ON
SET QUOTED IDENTIFIER ON
<Author,,Name>
-- Author:
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- -----
CREATE PROCEDURE obtener_empleados_por_aerolinea
   @idAerolinea int
BEGIN
   SELECT e.idEmpleado, p.nombre, COUNT(*) as cantidad_empleados
   FROM Empleado e
   JOIN Persona p ON e.Idpersona = p.Idpersona
   WHERE e.AerolineaID = @idAerolinea
   GROUP BY e.idEmpleado, p.nombre
END
```

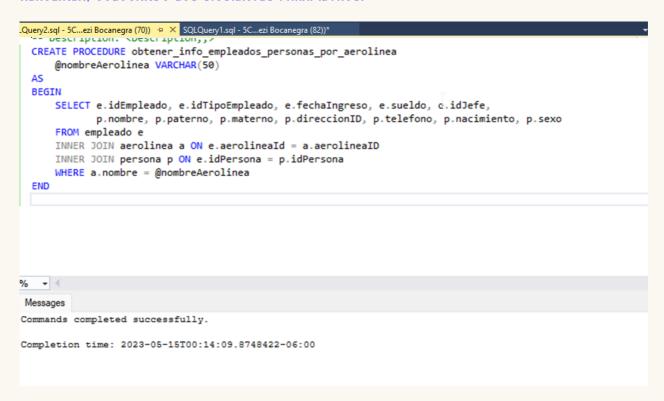


SE PUEDE CAMBIAR EL NÚMERO DE LA AEROLÍNEA POR LA QUE DESEAMOS CONSULTAR LOS DATOS.



Crear un procedimiento igual que el anterior, pero que recupera también las personas que trabajan en dicha aerolinea, pasandole como parámetro el nombre y apellido.

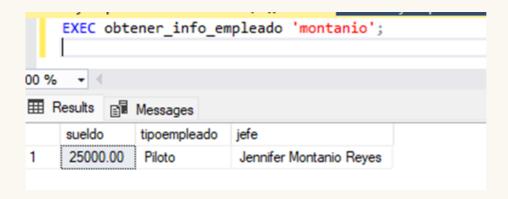
DADAS LAS ESPECIFICACIONES, RECUPERAMOS LA INFORMACIÓN DE LOS TRABAJADORES DE LA AEROLÍNEA, COLOCANDO LOS SIGUIENTES PARÁMETROS.



OBTENEMOS LOS DATOS EJECUTANDO CON EL SIGUIENTE COMANDO.



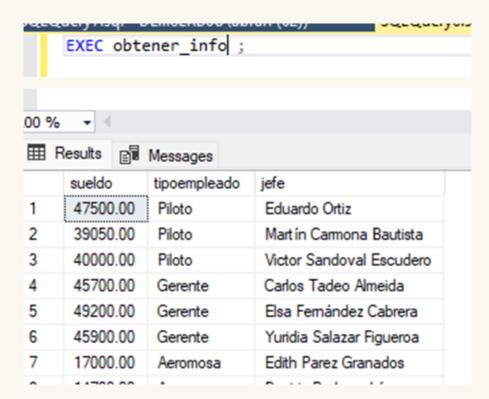
5





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 obtener info (
   @apellido VARCHAR(50) = NULL
AS
BEGIN
    SELECT
        e.sueldo,
        te.tipoempleado,
        CONCAT(p.nombre, ' ', p.paterno, ' ', p.materno) AS jefe
    FROM
        Empleado e
        INNER JOIN tipoempleado te ON e.IdTipoEmpleado = te.IdTipoEmpleado
        LEFT JOIN Empleado j ON e.idjefe = j.IdEmpleado
        INNER JOIN Persona p ON e.Idpersona = p.Idpersona
   WHERE
        @apellido IS NULL OR p.paterno = @apellido
END
```



AL EJECUTAR LOS DATOS DE LA TABLA OBTENEMOS LOS DATOS DE TODOS LOS EMPLEADOS TRAS NO COLOCAR LOS PARÁMETROS CORRESPONDIENTES.



Crear un procedimiento para mostrar el sueldo, tipo de empleado y nombre de la aerolinea de todos los empleados que contengan en su apellido el valor que le pasemos como parámetro

NO SE ENCUENTRAN LOS DATOS DEL NOMBRE DEL DEPARTAMENTO, POR ENDE, SOLO OBTENEMOS EL NOMBRE DE LA AFROLÍNFA.

```
ALTER PROCEDURE mostrar_empleados_por_apellido(@apellido VARCHAR(50) = NULL)

AS

BEGIN

SELECT e.sueldo, te.tipoempleado, a.nombre AS nombre_departamento

FROM Empleado e

INNER JOIN tipoempleado te ON e.IdTipoEmpleado = te.IdTipoEmpleado

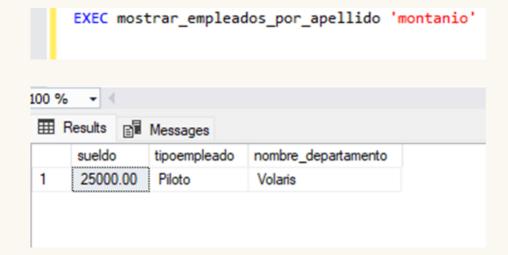
INNER JOIN aerolinea a ON e.AerolineaID = a.AerolineaID

INNER JOIN Direccion d ON a.direccionID = d.direccionID

INNER JOIN Persona p ON e.Idpersona = p.Idpersona

WHERE p.paterno LIKE '%'+@apellido+'%'

END
```



DE MODO QUE OBTENEMOS LOS SIGUIENTES DATOS. COMO NOMBRE DE DEPARTAMENTO.