TP N°2 BASE DE DATOS

Consignas:

Para la segunda entrega del trabajo práctico se deberán realizar los stored procedure necesarios para realizar altas, bajas y modificaciones en cada una de las tablas.

Tener en cuenta que la base guarda la información histórica.

Se deben realizar tres procedures por cada tabla, y cada procedure tiene que hacer todas las operaciones necesarias para mantener el histórico. Las operaciones pueden ser realizadas directamente
en el procedure o se pueden complementar con el uso de triggers.

```
--EMPLEADOS

use empleados

GO


CREATE PROC nuevo_empleado
@id_emp int,
@fecha_nacimiento date,
@nombre varchar(14),
@apellido varchar(16),
@genero varchar(2)
AS
    BEGIN TRY

        /*para ingresar un empleado debro verificar que sea mayor de edad .*/

        declare @edad int;

        select  @edad=datediff(year, @fecha_nacimiento, getDate());
        if @edad>=18
            BEGIN

                insert into empleados (id_emp, fecha_nacimiento, nombre, apellido, genero,
fecha_alta)
                VALUES (@id_emp, @fecha_nacimiento, @nombre, @apellido, @genero, getDate())

                PRINT 'empleado ingresado';
            END
        else
            BEGIN
                PRINT 'No se puede ingresar el empleado';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH
```

```
GO

CREATE PROC actualizar_empleado
@id_emp int,
@fecha_nacimiento date,
@nombre varchar(14),
@apellido varchar(16),
@genero varchar(2)
AS
    BEGIN TRY

        declare @edad int;

        select  @edad=datediff(year, @fecha_nacimiento, getDate());
        --verificar que sea mayor de edad y la fecha de alta igual o mayor a la actual
        if @edad>=18
            BEGIN

                update empleados set  nombre=@nombre, apellido=@apellido,
fecha_nacimiento=@fecha_nacimiento, genero=@genero
                where id_emp=@id_emp

                PRINT 'empleado actualizado';
            END
        else
            BEGIN
                PRINT 'No se puede ingresar el empleado';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH


GO


CREATE PROC baja_empleado
@id_emp int,
@fecha_desde date
AS
    BEGIN TRY


    /*
    para eliminar un empleado tengo que darlo de baja del departamento, de su puesto,
sueldo, responsabilidad sobre un departamento.
    */

        update puestos set fecha_hasta = @fecha_desde
            where fecha_hasta='99990101' and id_emp=@id_emp;

        update sueldos set fecha_hasta = @fecha_desde
            where fecha_hasta='99990101' and id_emp=@id_emp;

        update dept_emp set fecha_hasta=@fecha_desde
            where id_emp=@id_emp and fecha_hasta='99990101';
```

```
        update dept_respo set fecha_hasta=@fecha_desde
            where id_emp=@id_emp and fecha_hasta='99990101';

        PRINT 'empleado dado de baja';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH


GO

-----------------------------------------------------------------------
```

```
--DEPARTAMENTO

USE Empleados;

GO

/*elimino , en caso de existir, los stored procedure que voy a utilizar*/
drop PROCEDURE sp_eliminar_departamento;
drop PROCEDURE nuevo_departamento;
drop PROCEDURE actualizar_nombre_departamento

GO

CREATE PROC sp_nuevo_departamento
@id_dept as char(4),
@nombre_dept as VARCHAR(40)
AS
    BEGIN TRY

        /*para ingresar un departamento valido que el nombre o codigo no esten ingresados.
          si ya existe no hago nada, sino lo ingreso.
        */
        declare @bandera int;
        set @bandera=0;

        select @bandera=1 from departamentos where LOWER(nombre_dept)=LOWER(@nombre_dept) or
id_dept=@id_dept;

        if @bandera=0
            BEGIN

                insert into departamentos (id_dept,nombre_dept) values (@id_dept,
@nombre_dept);
                PRINT 'departamento ingresado';
            END

        else
            BEGIN

                PRINT 'El departamento no se ingreso, ya existe';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
```

```
    END CATCH


GO


CREATE PROC sp_actualizar_nombre_departamento
@id_dept as char(4),
@nombre_dept as VARCHAR(40)
AS
    BEGIN TRY

        /*para actualizar un departamento valido que el nombre no este ingresado.
          si ya existe no hago nada, sino lo actualizo.
        */
        declare @bandera int;
        set @bandera=0;

        select @bandera=1 from departamentos where LOWER(nombre_dept)=LOWER(@nombre_dept);

        if @bandera=0
            BEGIN

                update departamentos set nombre_dept= @nombre_dept
                    where id_dept = @id_dept;

                PRINT 'departamento actualizado';
            END
        else
            BEGIN

                PRINT 'El departamento no se actualizó, ya existe';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH

GO

CREATE PROC sp_eliminar_departamento
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

    /*para eliminar un departamento tengo que dar de baja a todos los empleados que formen
    parte de este, y por lo tanto que eliminar sus puestos, sueldos, responsabilidad sobre
un departamento.

    */

        update puestos set fecha_hasta = @fecha_desde
            where fecha_hasta='99990101' and id_emp in
                (select de.id_emp
                from dept_emp de
                where de.id_dept = @id_dept and de.fecha_hasta='99990101');
```

```
        update sueldos set fecha_hasta =@fecha_desde
            where fecha_hasta='99990101' and id_emp in
                (select de.id_emp
                from dept_emp de
                where de.id_dept = @id_dept and de.fecha_hasta='99990101');


        update dept_emp set fecha_hasta=@fecha_desde
            where id_dept = @id_dept and fecha_hasta='99990101';

        update dept_respo set fecha_hasta=@fecha_desde
            where id_dept = @id_dept and fecha_hasta='99990101';

        PRINT 'departamento dado de baja';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH

-----------------------------------------------------------------------
```

```
--DEPARTAMENTO_RESPONSABLE

use Empleados;
GO
/*
modifique la clave primaria para no tener problemas al momento de ingresar registros
la tabla dept_respo
*/

alter table dept_respo drop CONSTRAINT PK_dept_respo;

alter table dept_respo add constraint PK_dept_respo PRIMARY KEY
(id_emp,id_dept,fecha_desde)

GO
/*elimino , en caso de existir, los stored procedure que voy a utilizar*/
drop procedure sp_asignar_respo_dept

drop procedure sp_baja_respo_dept

drop procedure sp_baja_emp_respo_dept

GO

CREATE PROC sp_asignar_respo_dept
@id_emp int,
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

        /*
          debo validar para saber si el empleado ya es responsable de ese departamento en
particular actualmente,
          y si lo es no cambiar nada.
```

```
        */
          declare @emp int;
          declare @dept int;

          set @emp=0;
          set @dept=0;

          select @emp=1 from dept_respo where id_emp=@id_emp and id_dept=@id_dept and
fecha_hasta='99990101'

        if @emp=1
            BEGIN

                PRINT 'la operacion no se realizo, el empleado ya es responsable de ese
departamento';
            END

        else
            BEGIN
                /*
                Tambien tengo que comprobar que el empleado trabaje en ese departamento.
                */

                select @dept=1 from dept_emp where id_emp=@id_emp and id_dept=@id_dept and
fecha_hasta='99990101'

                /*
                si trabaja en el departamento, debo eliminar al responsable del departamento
que paso por parametro.

                */

                if @dept=1
                    BEGIN

                        EXEC sp_baja_respo_dept @id_dept,@fecha_desde;
                    END

                else
                    BEGIN
                        /*
                         si no trabaja en el departamento  debo cambiarlo a ese
departamento,
                         y si fuera responsable de algun departamento debo eliminarlo como
tal.

                        Cuando esté en el departamento correspondiente, debo eliminar al
responsable del departamento que paso por parametro
                        y debo hacer que el empleado no sea responsable de ningun
departamento.


                        */



                        EXEC sp_alta_emp_dept @id_emp,@id_dept,@fecha_desde;

                        EXEC sp_baja_emp_respo_dept @id_emp,@fecha_desde;

                        EXEC sp_baja_respo_dept @id_dept,@fecha_desde;
```

```
                    END

                /*
                 Por ultimo ingreso al empleado como responsable del departamento.
                */

                INSERT INTO dept_respo(id_emp, id_dept, fecha_desde, fecha_hasta)
                VALUES (@id_emp, @id_dept, @fecha_desde, '99990101');

                PRINT 'se asigno al empleado como responsable de departamento';
            END


    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH

GO


--baja por departamento
CREATE PROC sp_baja_respo_dept
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

    /*
    doy de baja al responsable del departamento
    */

        update dept_respo set fecha_hasta=@fecha_desde
            where id_dept=@id_dept and fecha_hasta='99990101';

        PRINT 'El departamento ya no tiene responsable';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH

GO

--baja por empleado
CREATE PROC sp_baja_emp_respo_dept
@id_emp int,
@fecha_desde date
AS
    BEGIN TRY

        /*
        doy de baja al empleado como responsable de departamento,
        pero no lo doy de dept_emp porque puede seguir siendo empleado del departamento.
        */

        update dept_respo set fecha_hasta=@fecha_desde
            where id_emp=@id_emp and fecha_hasta='99990101';
```

```
        PRINT 'El empleado ya no es responsable de departamento';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



GO



CREATE PROC sp_actualizar_emp_respo_dept
@id_emp int,
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

        /*
        doy de baja al empleado como responsable de departamento,
        pero no lo doy de dept_emp porque puede seguir siendo empleado del departamento.
        */

        EXEC sp_baja_emp_respo_dept @id_emp,@fecha_desde;

        EXEC sp_baja_respo_dept @id_dept,@fecha_desde;


        INSERT INTO dept_respo(id_emp, id_dept, fecha_desde, fecha_hasta)
                VALUES (@id_emp, @id_dept, @fecha_desde, '99990101');


        PRINT 'se asigno al empleado como responsable de departamento';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



--------------------------------------------------------------------------------
```

```
--DEPARTAMENTO_EMPLEADOS

use Empleados;

GO

/*elimino , en caso de existir, los stored procedure que voy a utilizar*/

drop procedure sp_alta_emp_dept;
drop procedure sp_baja_emp_dept;
drop procedure sp_actualizar_emp_dept;

GO

CREATE PROC sp_alta_emp_dept
@id_emp int,
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

        /*
```

```
          debo validar para saber si el empleado ya forma parte de ese departamento en
particular actualmente,
          y si forma parte no cambiar nada.
        */

        declare @bandera int=0;

        select @bandera=1 from dept_emp where id_emp=@id_emp and id_dept=@id_dept and
fecha_hasta='99990101'

        if @bandera=1
            BEGIN

                PRINT 'la operacion no se realizo, el empleado ya trabaja en el
departamento';
            END

        else
            BEGIN
               /*
                si no trabaja en el departamento  debo cambiarlo a ese departamento, y si
fuera responsable de algun departamento
                debo eliminarlo como tal.
                */

                 EXEC sp_actualizar_emp_dept @id_emp,@id_dept,@fecha_desde;
            END


    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



GO

CREATE PROC sp_baja_emp_dept
@id_emp int,
@fecha_desde date
AS
    BEGIN TRY

        --para dar de baja a un empleado de un departamento debo eliminarlo como responsable
de departamento,
        --y del departamento donde trabaja.

        /*update dept_respo set fecha_hasta=@fecha_desde
            where id_emp=@id_emp and fecha_hasta='99990101';*/

        EXEC sp_baja_emp_respo_dept @id_emp,@fecha_desde


        update dept_emp set fecha_hasta=@fecha_desde
            where id_emp=@id_emp  and fecha_hasta='99990101';


        PRINT 'El empleado ya no trabaja en el departamento';
```

```
    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH




GO


CREATE PROCEDURE sp_actualizar_emp_dept
@id_emp int,
@id_dept char(4),
@fecha_desde date
AS
    BEGIN TRY

        EXEC sp_baja_emp_dept @id_emp,@fecha_desde;

        insert into dept_emp(id_emp,id_dept,fecha_desde,fecha_hasta)
            values (@id_emp,@id_dept,@fecha_desde,'99990101');

        PRINT 'El empleado se asignó al departamento';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



-------------------------------------------------------------------------------------------------




GO

--PUESTO_DESCRIPCION

CREATE PROC crear_puesto
@puesto_descr_es varchar(100),
@puesto_descr_en varchar(100)
AS
    BEGIN TRY

        declare @bandera int;
        set @bandera=0;

        select @bandera=1
        from puesto_descripcion
        where LOWER(puesto_descr_es)=LOWER(@puesto_descr_es) or
LOWER(puesto_descr_en)=LOWER(@puesto_descr_en);

        if @bandera=0
```

```
            BEGIN

                insert into puesto_descripcion(puesto_descr_es, puesto_descr_en)
                values (@puesto_descr_es, @puesto_descr_en);

                PRINT 'puesto ingresado';
            END

        else
            BEGIN

                PRINT 'El puesto no se ingreso, ya existe';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH


GO

CREATE PROC actualizar_puesto
@id smallint,
@puesto_descr_es varchar(100),
@puesto_descr_en varchar(100)
AS
    BEGIN TRY

        declare @bandera int;
        set @bandera=0;

        select @bandera=1
        from puesto_descripcion
        where LOWER(puesto_descr_es)=LOWER(@puesto_descr_es) or
LOWER(puesto_descr_en)=LOWER(@puesto_descr_en);

        if @bandera=0
            BEGIN

                update puesto_descripcion set puesto_descr_es=@puesto_descr_es,
puesto_descr_en=@puesto_descr_en
                    where id=@id;

                PRINT 'puesto actualizado';
            END

        else
            BEGIN

                PRINT 'Los valores ingresados ya existen';
            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
```

```
    END CATCH



GO

CREATE PROC baja_puesto
@id_puesto int
AS
    BEGIN TRY

        update puestos set fecha_hasta=getDate()
            where puesto=@id_puesto and fecha_alta='99990101';

        PRINT 'operacion realizada con exito';


    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();

    END CATCH
```

```
--PUESTOS
use empleados;


GO

--drop PROCEDURE sp_asignar_puesto_emp;

CREATE PROC sp_asignar_puesto_emp
```

```
@id_emp int,
@id_puesto int,
@fecha_desde date
AS
    BEGIN TRY

       /*
        debo validar que el puesto ingresado no lo tenga actualmente.
        si lo tiene no se realiza la operacion.
        */
        declare @emp int;
        set @emp=0;

        select @emp=1
        from puestos
        where id_emp=@id_emp and puesto=@id_puesto and fecha_hasta='99990101'; -- me fijo si
el empleado ya tiene asignado ese puesto en paricular;


        if @emp=1
            BEGIN

                PRINT 'no se realizo la operacion, el empleado ya esta registrado
actualmente con el puesto que esta tratando de asignar';
            END
        else
            BEGIN

                EXEC sp_actualizar_puesto_emp @id_emp,@id_puesto,@fecha_desde;

            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH




GO
-- drop procedure sp_baja_puesto_emp
CREATE PROC sp_baja_puesto_emp
@id_emp int,
@fecha_desde date
AS
    BEGIN TRY

        update puestos set fecha_hasta=@fecha_desde
            where id_emp=@id_emp  and fecha_hasta='99990101'

        PRINT 'El empleado ya no tiene el puesto';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH
```

```sql
GO


--drop procedure sp_actualizar_puesto_emp

CREATE PROC sp_actualizar_puesto_emp
@id_emp int,
@id_puesto int,
@fecha_desde date
AS
    BEGIN TRY

        /*
        para actualizar puesto del empleado debo dar
        de baja su puesto actual y ingresar el nuevo puesto
        */

        EXEC sp_baja_puesto_emp @id_emp,@fecha_desde

        insert into puestos (id_emp,puesto,fecha_desde,fecha_hasta)
            values (@id_emp,@id_puesto,@fecha_desde,'99990101');

        PRINT 'Se realizo la operacion, se asigno el puesto al empleado';


    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH









--SUELDOS
```

```
use Empleados

--DROP PROCEDURE sp_baja_sueldo_emp

--drop procedure sp_actualizar_sueldo_emp

--drop procedure sp_asignar_sueldo_emp


GO


CREATE PROC sp_asignar_sueldo_emp
@id_emp int,
@sueldo int,
@fecha_desde date
AS
    BEGIN TRY

        /*
        debo validar que el monto ingresado no lo tenga actualmente.
        si lo tiene no se realiza la operacion.
        */

        declare @emp int;
        set @emp=0;

        select @emp=1 from sueldos where id_emp=@id_emp and sueldo=@sueldo and
fecha_hasta='99990101';

        if @emp=1
            BEGIN

                PRINT 'No se realizo la operacion, el empleado ya tiene ese monto asignado
actualmente';
            END

        else
            BEGIN
                /*
                si no lo tiene procedo a actualizar la fecha hasta del sueldo actual,
                y ingreso al empleado con el sueldo nuevo.
                */

                EXEC sp_actualizar_sueldo_emp @id_emp, @sueldo,@fecha_desde;


            END

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



GO

CREATE PROC sp_baja_sueldo_emp
@id_emp int,
```

```
@fecha_desde date
AS
    BEGIN TRY

        update sueldos set fecha_hasta=@fecha_desde
            where id_emp=@id_emp and fecha_hasta='99990101';

        PRINT 'Se elimino el sueldo del empleado';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH



GO

CREATE PROC sp_actualizar_sueldo_emp
@id_emp int,
@sueldo int,
@fecha_desde date
AS
    BEGIN TRY

        /*
        para actualizar sueldo del empleado debo dar
        de baja su sueldo actual y ingresar el nuevo sueldo.
        */

        EXEC sp_baja_sueldo_emp @id_emp,@fecha_desde

        insert into sueldos(id_emp,sueldo,fecha_desde,fecha_hasta)
            values(@id_emp,@sueldo,@fecha_desde,'99990101');



        PRINT 'Se asigno el nuevo sueldo del empleado';

    END TRY
    BEGIN CATCH

        PRINT 'Se produjo un error';
        PRINT 'Linea de Error : ' + CAST(ERROR_LINE() AS VARCHAR(10));
        PRINT 'Mensaje de Error : ' + ERROR_MESSAGE();
    END CATCH

-----------------------------------------------------------------------------------------------
```