

--1. Hacer un procedimiento almacenado que permita dar de alta un paciente si el mismo no existe y o modificar su datos si ya existe, además al realizar una modificación se debe verificar que el paciente no tenga intervenciones previas.

```
create or alter procedure alta_paciente @pac_nombre varchar(50),
@pac_apellido varchar(50), @pac_FchNacim datetime, @pac_Mail varchar(50),
@pac_Tel varchar(50) AS
begin
    declare @pacID int, @cantIntervenciones int
    set @pacID = 0
    set @cantIntervenciones = 0
    select @pacID = pac_ID from Paciente where pac_nombre = @pac_nombre
and pac_apellido = @pac_apellido and pac_FchNacim = @pac_FchNacim
    if @pacID > 0
        begin
            select @cantIntervenciones = count(1) from
Intervenciones where pac_ID=@pacID
            if @cantIntervenciones = 0
                begin
                    UPDATE Paciente
                    set pac_nombre=@pac_nombre, pac_apellido=
@pac_apellido, pac_FchNacim=@pac_FchNacim, pac_Mail = @pac_Mail, pac_Tel
= @pac_Tel
                    where pac_ID=@pacID
                end
            else
                begin
                    print 'El paciente tiene intervenciones
previas, no se puede modificar'
                end
            end
        else
            begin
                insert into Paciente
(pac_nombre,pac_apellido,pac_FchNacim,pac_Mail,pac_Tel) values
(@pac_nombre, @pac_apellido, @pac_FchNacim, @pac_Mail, @pac_Tel)
            end
        end
    go
```

--2. Hacer una función que dado un paciente y un año retorne la o las intervenciones que dicho paciente se realizó en el año ingresado.

```
create function intervencionesXPaciente(@pacId int, @anio int) returns
table as
return
    (select *
    from Intervenciones
    where pac_ID = @pacID
    and year(int_fechaHora) = @anio)
go
```

--3. Hacer un disparador que no permita ingresar funcionario si el mismo ya existe (func_nombre, func_apellido y func_tipo) y que al insertar un nuevo funcionario cargue la fecha de ingreso con la fecha del día.

```
create or alter trigger trg_control_funcionarios
on Funcionarios
after insert as
begin
```

```

declare @funcionario int, @funcId int, @func_tipo varchar(50), @funNombre
varchar(50), @funcApellido varchar(50), @funcTipo varchar(20)
    select @funcId = func_Id, @funNombre = func_nombre, @funcApellido =
func_apellido, @funcTipo = func_tipo from inserted
    select @funcionario = count(1) from Funcionarios where @funNombre =
func_nombre and @funcApellido = func_apellido and @funcTipo = func_tipo
    if @funcionario = 1 --si da mayor a 1 registro, es porque se acaba
de insertar un registro identico por lo que realizo el rollback
        begin
            update Funcionarios set func_FchIngreso = getdate()
where func_Id = @funcId
        end
    else
        begin
            print 'No se puede ingresar el funcionario porque ya
existe en el sistema'
            rollback
        end
end
go

```

--4. Hacer un disparador que no permita ingresar una intervención de tipo 'Operacion' si el tipo de funcionario no es 'Cirujano' y el tipo de sala no es un 'Quirofano'.

```

create or alter trigger trg_control_intervencion
on Intervenciones
after insert as
begin
declare @funcID int, @funcTipo varchar(20), @salaID int, @salaTipo
varchar(20), @intTipo varchar(20)
    select @funcID=func_id, @salaID=sala_ID, @intTipo=int_tipo from
inserted
    select @funcTipo = func_tipo from Funcionarios where func_id =
@funcID
    select @salaTipo = sala_tipo from Salas where sala_ID = @salaID
    if @intTipo='Operacion' and (@funcTipo != 'Cirujano' or @salaTipo
!= 'Quirofano')
        begin
            rollback
            print 'No se permita ingresar una intervención de tipo
'Operacion' si el tipo de funcionario no es 'Cirujano' o el tipo de sala
no es un 'Quirofano'
        end
end
go

```

--5. Hacer un disparador que ante la modificación de la sala de una intervención deje un registro en la tabla de auditoría cargando todos los campos de dicha tabla.

```

create or alter trigger trg_control_sala_intervencion
on Intervenciones
after update as
begin
    if update(sala_ID)
        begin
            insert into Auditoria select
getdate(), 'Intervenciones', 'sala_ID', deleted.sala_ID, inserted.sala_ID,
'Update campo sala_id'

```

```
        from inserted,deleted
        where inserted.pac_ID=deleted.pac_ID
        and inserted.func_ID=deleted.func_ID
        and inserted.int_tipo=deleted.int_tipo
        and inserted.int_fechaHora=deleted.int_fechaHora
    print 'Se deja registro de auditoria'
end
end
go
```