

## Práctico de repaso para 2° parcial

### Tema1

Se tiene el diagrama de una base de datos de una clínica veterinaria que necesita registrar los datos de las consultas que realizan dueños con sus mascotas a los veterinarios.

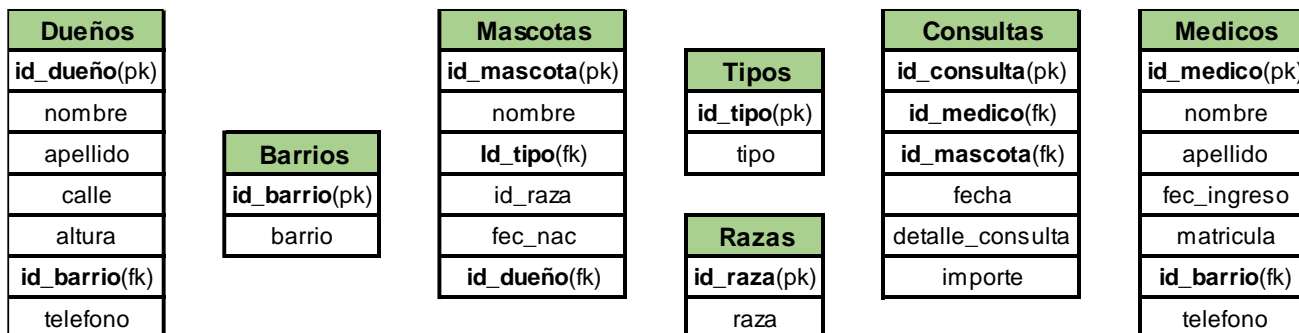


Ilustración 1: Elaboración propia

Escribir la sentencia SQL que permita realizar las consultas que se piden a continuación:

1. Crear una función que devuelva el nombre completo del dueño con el apellido todo en mayúsculas, coma, espacio y el nombre con la 1er. letra en mayúsculas y el resto en minúsculas

```
create function obt_nombre_dueñobis(@id_dueño int)
returns varchar(100)
as
begin
    declare @dueño varchar(100)
    set @dueño=(select UPPER(d.apellido)+', '+UPPER(LEFT(d.nombre,1))+
                LOWER(substring(d.nombre,2,LEN(d.nombre))) 'Nombre completo'
                from dueños d
                where id_dueño=@id_dueño)
    return @dueño
end
```

```
select dbo.obt_nombre_dueñobis(1)
```

2. Crear una vista que muestre el listado de mascotas (nombre, tipo, raza, edad) con sus dueños (nombre completo utilizando la función del punto 1, dirección completa, teléfono)

```
create view vista_masc2
as
select m.nombre , t.tipo , r.raza, DATEDIFF(YEAR,m.fec_nac,GETDATE()) edad,
       dbo.obt_nombre_dueñobis(m.id_dueño) dueño, d.calle+' '+cast(d.altura as varchar(10))
       direccion, d.telefono, m.id_mascota
from mascotas m,tipos t, dueños d, razas r
where m.id_tipo = t.id_tipo and d.id_dueño = m.id_dueño and m.id_raza = r.id_raza

select * from vista_masc2
```

3. Consultar la vista anterior mostrando nombre y raza de perros con más de 5 años, de dueños con teléfono conocido (mostrar nombre de dueño y teléfono), que vinieron a consulta este año

```
select nombre, raza, dueño, telefono
from vista_masc2 m2
where id_mascota in (select id_mascota
                    from consultas
                    where year(fecha)= YEAR(GETDATE()))
and edad > 5 and tipo = 'perro'
```

4. Mostrar los importes totales cobrados mensualmente por cada médico entre los años que se ingresarán por parámetro

```
create proc obtener_mensuales
@año_desde int, @año_hasta int
as
begin
select sum(c.importe) total, c.id_medico medico, MONTH(c.fecha) mes, year(c.fecha) año
from consultas c
where year(c.fecha) between @año_desde and @año_hasta
group by c.id_medico, MONTH(c.fecha), year(c.fecha)
order by 4, 1
end
```

```
exec obtener_mensuales 2021,2022
```

5. Crear un trigger que impida que se modifique el importe de las consultas.

```
create TRIGGER no_actualiza_consulta
ON consultas
AFTER UPDATE
AS
IF update(importe)
begin
    RAISERROR ('No actualizar importe', 16, 10); --si lo hace, rollback de la transaction y
                                                -- error
    rollback transaction
end
```

## Práctico de repaso para 2° parcial

### Tema 2

Se tiene el diagrama de una base de datos de una clínica veterinaria que necesita registrar los datos de las consultas que realizan dueños con sus mascotas a los veterinarios.

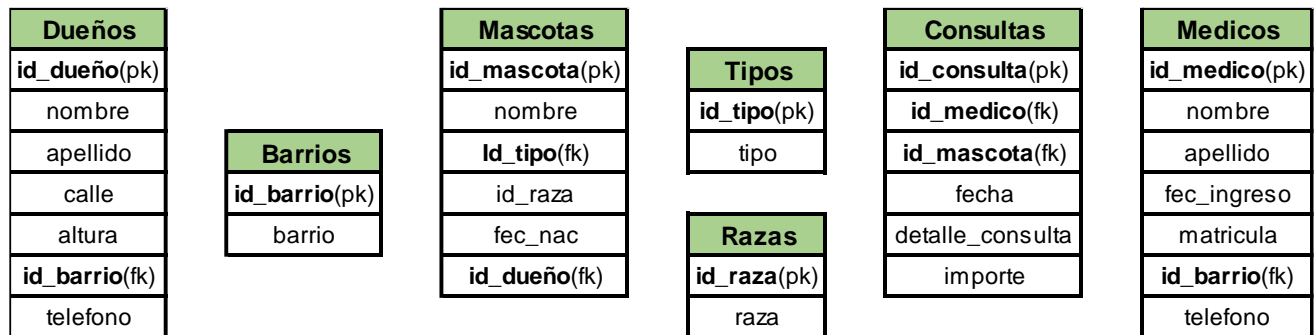


Ilustración 2: Elaboración propia

Escribir la sentencia SQL que permita realizar las consultas que se piden a continuación:

1. Crear un procedimiento almacenado para insertar un nuevo médico. (antes de completar este punto, agregue lo que se solicita en el punto 2 y 3)

```

procedure insertar_medico
  @id_medico int,
  @nombre varchar(30),
  @apellido varchar(30),
  @fec_ingreso datetime,
  @matricula int,
  @id_barrio int,
  @telefono varchar(30)
as
begin
  if (@id_medico is not null and @matricula is not null)
    insert into medicos(id_medico,nombre,apellido,fec_ingreso,matricula,id_barrio,telefono)
      values(@id_medico,@nombre,@apellido,@fec_ingreso,@matricula,@id_barrio,@telefono)
  else
    begin
      insert into #Errores_db (ErrorNumber, Error_Severity, Error_State, Error_Procedure,
        Error_Line , Error_Message )
        values(1,15,1,'insertar_medico',1,'No se puedo ingresar el medico nuevo')
    end
    raiserror('No se puedo ingresar el medico nuevo', 15, 1)
end

exec insertar_medico 6, 'pepe', N'pepe', N'1/1/1', 1, 1, N'1111111';
exec insertar_medico null, 'pepe', N'pepe', N'1/1/1', 1, 1, N'1111111';
select * from medicos
select * from #Errores_db
  
```

2. Modificar el procedimiento anterior para que en caso de que la matricula o el apellido sean nulos no permita hacer el insert y de un error por excepción

3. Cree una tabla temporal para guardar los errores del punto 2

```
CREATE TABLE #Errores_db
(ErrorNumber INT,
Error_Severity INT,
Error_State INT,
Error_Procedure varchar(126),
Error_Line INT,
Error_Message nvarchar(2048)
)
```

4. Crear una vista que liste la cantidad de consultas, importe total y promedio de importe, mayor y menor importe por dueño por año

```
create view vista_estadistica
as
select d.apellido apellido, d.nombre nombre, year(c.fecha) año,
count(c.id_consulta) cantidad_consul, sum(c.importe) importe_total,
avg(c.importe) prom_importe_total, max(c.importe) maximo, min(c.importe) minimo
from consultas c, mascotas m, dueños d
where c.id_mascota = m.id_mascota and m.id_dueño = d.id_dueño
group by d.apellido , d.nombre, year(c.fecha)

select * from vista_estadistica
```

5. Consultar la vista anterior mostrando el dueño, importe total y cantidad de consultas realizadas el año pasado cuyo importe promedio sea mayor al importe promedio de todas las consultas de este año

```
select apellido, nombre, año, importe_total, cantidad_consul
from vista_estadistica vm
where año = year(GETDATE())-1
and vm.prom_importe_total > (select avg(importe)
from consultas
where year(fecha) = year(GETDATE()) )
```