



Procesual Hito 4

Nombre completo: Kevin Michael Apaza Lara

Universidad: Universidad Franz Tamayo (Unifranz el alto)

Paralelo: 3

Semestre: II/2021

Materia: Base de Datos I

Consigna

Diseñe un sistema de Base de Datos Relacional utilizando el gestor de Base de Datos SQL Server teniendo como premisa el uso de buenas prácticas en diseño de la base de datos aplicados al siguiente escenario. Una organización sin fines de lucro ONG requiere de un sistema de información para poder gestionar proyectos manejados por ellos.

ONG la muralla verde	
Diseño	<p>En función al escenario se identificó que las posibles entidades son departamento, provincia, persona, proyecto y detalle proyecto, pues un cliente vive en cierta localidad y esta persona trabaja en un proyecto específico.</p> <p>El objetivo es poder tener un control de todos los proyectos, por ejemplo quiero saber donde vive una persona y en qué proyecto participa.</p> <p>En la primera fase solo debe resolver este requerimiento inicial manejando Vistas y Funciones.</p> <ul style="list-style-type: none">● Determinar donde vive una persona y en qué proyecto participa.

1. Diseño de base de datos.

1.1. Adjuntar el código SQL que genera la base de datos, tablas y los registros correspondientes.

■ La base de datos debe llamarse ONG_H4

```
create database ONG_H4;  
go  
use ONG_H4;  
go
```

Creamos y usamos la base de datos

Creamos la tabla departamento,
con sus respectivos campos

```
create table departamento
(
    id_dep integer identity primary key not
null,
    nombre varchar(50)
);
```

Creamos la tabla provincia,
con sus respectivos campos,
además de su llave foránea.

```
create table provincia
(
    id_prov integer identity primary key
not null,
    nombre varchar(150),
    id_dep integer not null,
    foreign key (id_dep) references
departamento(id_dep)
);
```

Creamos la tabla persona,
con sus respectivos campos,
además de su llave foránea.

```
create table persona
(
    id_per integer primary key not null,
    nombre varchar(20) not null ,
    apellidos varchar(50) not null ,
    fecha_nac date not null,
    edad integer not null ,
    email varchar(100)not null ,
    sexo char(1)not null,
    id_dep integer not null ,
    id_prov integer not null ,
    foreign key (id_dep) references departamento
(id_dep),
    foreign key (id_prov) references provincia
(id_prov)
);
```

Creamos la tabla proyecto,
con sus respectivos campos.

```
create table proyecto
(
    id_proy integer identity primary key not
null ,
    nombre_proy varchar(100)not null ,
    tipo_proy varchar(30)not null
);
```

Creamos la tabla detalle_proyecto,
con sus respectivos campos,
además de su llave foránea.

```
create table detalle_proyecto
(
    id_det_proy integer identity primary key not
null ,
    id_per integer not null ,
    id_proy integer not null ,
    foreign key (id_per) references
persona(id_per),
    foreign key (id_proy) references
proyecto(id_proy),
);
go
```

Insertamos datos en la tabla
departamento

```
insert into departamento(nombre) values  
('Pando'),  
('Beni'),  
('La Paz'),  
('Cochabamba'),  
('Santa Cruz'),  
('Oruro'),  
('Potosi'),  
('Chuquisaca'),  
('Tarija');  
go
```

Insertamos datos en la tabla
provincia

```
--pando
insert into provincia(nombre, id_dep) values
('Nicolas Suarez',1),
('Manuripi',1),
('Madre de Dios',1),
('Abuna',1),
('Gral. Federico Roman',1);
--beni
insert into provincia(nombre, id_dep) values
('Cercado',2),
('Vaca Diez',2),
('Gral. J. Ballivian',2),
('Yacuma',2),
('Moxos',2);
--la paz
insert into provincia(nombre, id_dep) values
('Murillo',3),
('Omasuyos',3),
('Pacajes',3),
('Camacho',3),
('Munecas',3);
```


Insertamos datos en la tabla
provincia

```
--cochabamba
insert into provincia(nombre, id_dep) values
('Cercado', 4),
('Campero', 4),
('Ayopaya', 4),
('Esteban Arce', 4),
('Arani', 4);
--Santa Cruz
insert into provincia(nombre, id_dep) values
('Andres Ibanez', 5),
('Warnes', 5),
('Velasco', 5),
('Ichilo', 5),
('Chiquitos', 5);
--Oruro
insert into provincia(nombre, id_dep) values
('Cercado', 6),
('Abaroa', 6),
('Carangas', 6),
('Sajama', 6),
('Litoral', 6);
```

Insertamos datos en la tabla
provincia

```
--Potosi
insert into provincia(nombre, id_dep) values
('Tomas Frias',7),
('Rafael Bustillo',7),
('Cornelio Saavedra',7),
('Chayanta',7),
('Charcas',7);
--Chuquisaca
insert into provincia(nombre, id_dep) values
('Oropeza',8),
('Azurduy',8),
('Zudanez',8),
('Tomina',8),
('Hernando',8);
--Tarija
insert into provincia(nombre, id_dep) values
('Cercado',9),
('Arce',9),
('Gran Chaco',9),
('Avilez',9),
('Mendez',9);
go
```

1.2. Después de generar el modelo lógico de la base de datos.

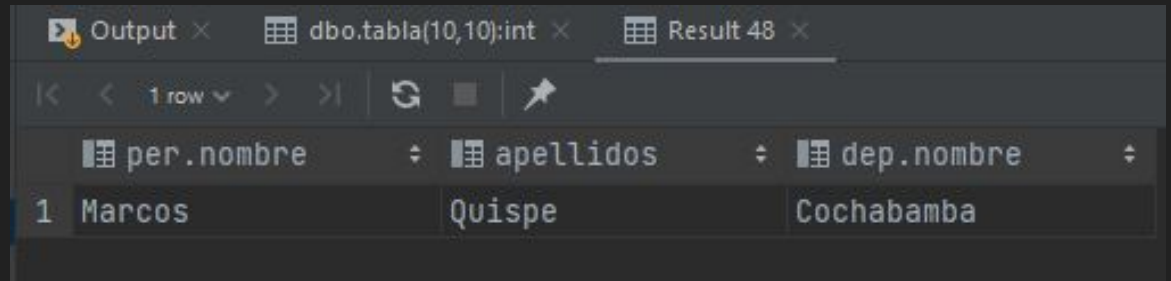
- Utilizar Datagrip para poder generar el diagrama



2. Manejo de Vistas

2.1. Mostrar a todas las personas que viven en el departamento de Cochabamba.

```
select per.nombre, per.apellidos, dep.nombre  
from persona as per  
inner join departamento dep on dep.id_dep = per.id_dep  
where dep.nombre='Cochabamba';
```



per.nombre	apellidos	dep.nombre
1 Marcos	Quispe	Cochabamba

```

select per.nombre, per.apellidos, proy.nombre_proy
from detalle_proyecto as dp
inner join persona per on per.id_per = dp.id_per
inner join proyecto proy on proy.id_proy = dp.id_proy;

```

2.2. Mostrar la persona (nombres y apellidos) y el nombre del proyecto en donde trabajan.

	🇬🇧 nombre	÷ 🇬🇧 apellidos	÷ 🇬🇧 nombre_proy	÷
1	Pedro	Ramirez	Plantando	
2	Maria	Perez	Plantando	
3	Jose	Cornel	Animal Libre	
4	Marcos	Quispe	Animal Libre	
5	Carmen	Fernandez	Ninos con derecho	
6	Rodrigo	Mercedes	Ninos con derecho	
7	Mario	Morales	Semillas del Futuro	
8	Josep	Perez	Semillas del Futuro	
9	Hugo	Quispe	Conservacion de animales	
10	Ronal	Casas	Conservacion de animales	

2.3. Asumir que tiene 3 tipos de proyectos (TIPO_A, TIPO_B, TIPO_C)

- El objetivo es crear una Vista con los mismos datos de la tabla proyectos
- Sin embargo generar una nueva columna en la vista de nombre departamento_aplicarse
- Si el tipo de proyecto es de TIPO_A asignar CBB
- Si el tipo de proyecto es de TIPO_B asignar LPZ
- Si el tipo de proyecto es de TIPO_C asignar SCZ
- Si el tipo de proyecto es otro tipo asignar 'En proceso de análisis'

Creamos la vista

```
create or alter view proyecto_datos as
select *,departamento_aplicarse=
    case
        when proy.tipo_proy = 'TIPO_A' then 'CBB'
        when proy.tipo_proy = 'TIPO_B' then 'LPZ'
        when proy.tipo_proy = 'TIPO_C' then 'SCZ'
        else 'OTRO DEPARTAMENTO'
    end
from proyecto as proy;
```

Hacemos la consulta

```
select * from proyecto_datos;
```

	id_proy	nombre_proy	tipo_proy	departamento_aplicarse
1	1	Plantando	TIPO_A	CBB
2	2	Animal Libre	TIPO_B	LPZ
3	3	Ninos con derecho	TIPO_C	SCZ
4	4	Semillas del Futuro	TIPO_D	OTRO DEPARTAMENTO
5	5	Conservacion de animales	TIPO_E	OTRO DEPARTAMENTO

Creamos la vista

```
create or alter view riesgo_en_ubicacion as
select per.nombre,per.apellidos,per.edad,per.id_prov,riesgo_derrumbe=
    case
        when per.id_prov<27 then 'EXISTE'
        else 'NO EXISTE'
    end
from persona as per;
```

2.4. Crear una vista cualquiera que muestre 5 columnas.

Hacemos la consulta

```
select * from riesgo_en_ubicacion;
```

Esta es una vista que muestra a aquellos que trabajan en una zona de riesgo de derrumbe

	nombre	apellidos	edad	id_prov	riesgo_d
1	Pedro	Ramirez	21	1	EXISTE
2	Maria	Perez	22	6	EXISTE
3	Jose	Cornel	23	11	EXISTE
4	Marcos	Quispe	24	16	EXISTE
5	Carmen	Fernandez	25	21	EXISTE
6	Rodrigo	Mercedes	26	26	EXISTE
7	Mario	Morales	27	31	NO EXISTE
8	Josep	Perez	28	36	NO EXISTE
9	Hugo	Quispe	29	41	NO EXISTE
10	Ronal	Casas	30	1	EXISTE

Creamos la función

3. Manejo de Funciones

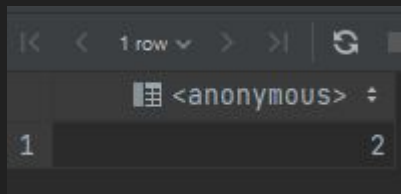
3.1. Crear una función que permita saber cuántos proyectos distintos del TIPO_A, TIPO_B y TIPO_C existen

- La función no recibe ningún parámetro

Llamamos a la función

```
create or alter function otros_proyectos()  
returns integer  
begin  
    declare @devolver integer;  
    set @devolver =(  
        select count(proy.tipo_proy) as proyectos_diferentes  
        from proyecto as proy  
        where proy.tipo_proy != 'TIPO A'  
            and proy.tipo_proy != 'TIPO B'  
            and proy.tipo_proy != 'TIPO C');  
  
    return @devolver;  
  
end;
```

```
select dbo.otros_proyectos();
```



The screenshot shows a SQL client interface with a toolbar at the top containing navigation and execution icons. Below the toolbar, a dropdown menu is set to "<anonymous>". The main area displays a single row of results with the value "2".

1	2
---	---

Creamos y llamamos a la función

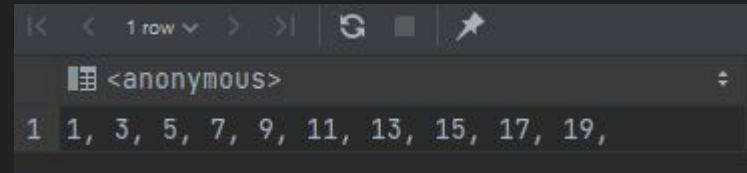
3.2. Crear una función que genere los primeros N números impares.

- La función recibe solo un parámetro (el valor N)
- Si n es 5 la salida debe ser: 1, 3, 5, 7, 9,
- Si n es 2 la salida debe ser: 1, 3,
- Si n es 4 la salida debe ser: 1, 3, 5, 7

```
create or alter function impares(@parametro int)
returns varchar(100)as
begin
    declare @respuesta varchar(100)='';
    declare @contador integer=1;

    while @contador<=@parametro* 2
    begin
        set @respuesta = concat(@respuesta,@contador,',',
    ');
        set @contador = @contador+ 2;
    end
    return @respuesta;
end;
go

select dbo.impares(10);
```



1	1, 3, 5, 7, 9, 11, 13, 15, 17, 19,
---	------------------------------------

3.3. Crear una función que permita insertar un registro a la tabla persona.

- La función recibe los datos a insertarse a la tabla persona
- La función retorna un mensaje indicando que se insertó satisfactoriamente el registro.

```
create or alter function insetar persona(@id perl
integer,@nombrel varchar(20),@apellidos1 varchar(50),
@fecha nac1 date,@edad1 integer,@email1 varchar(100),
@sexo1 char(1),@id_dep1 integer,
@id_prov1 integer)
returns varchar(100)as
begin
    insert into persona(id per, nombre, apellidos, fecha_nac,
edad, email, sexo, id dep, id prov)
    values (@id_per1, @nombrel, @apellidos1, @fecha_nac1,
@edad1, @email1, @sexo1, @id_dep1, @id_prov1);

    return 'Se inserto satisfactoriamente el registro'
end
go
```

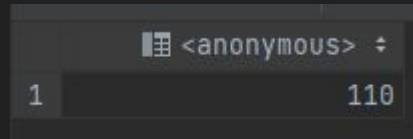
Error en la ejecución

Creamos y llamamos a la función

3.4. Crear una función cualquiera.

- La función debe de recibir 2 parámetros.
- Usar los parámetros en la lógica de la función.

```
create or alter function suma_de_numerosA(@numA integer,  
@numB integer)  
returns integer  
begin  
    declare @respuesta integer;  
  
    set @respuesta= @numA+@numB ;  
  
    return @respuesta;  
end  
  
select dbo.suma_de_numerosA(10,100)
```



<anonymous>	
1	110