

Actividad procesual Hito 4

MANEJO DE VISTAS Y FUNCIONES

Martin Josue Mamani Pilco

Diseño de base de datos

Código de la base de datos, las tablas y los registros

```
create database ONG_H4  
go
```

```
use ONG_H4  
go
```

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

```
create table provincia (  
    id_prov int primary key identity not null,  
    nombre varchar(100) not null,  
    id_dep int not null,  
    foreign key (id_dep) references departamento (id_dep)  
);
```

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

```
create table proyecto (
    id_proy int primary key identity not null ,
    nombreProy varchar(100) not null ,
    tipoProy varchar(30) not null
);
```

```
create table detalle_proyecto (
    id_detProy int primary key identity not null ,
    id_per int not null ,
    id_proy int not null,
    foreign key (id_per) references persona (id_per),
    foreign key (id_proy) references proyecto (id_proy)
);
```

```
insert into proyecto (nombreProy, tipoProy)
values ('PROYECTO_A', 'TIPO_A'),
       ('PROYECTO_B', 'TIPO_B'),
       ('PROYECTO_C', 'TIPO_C'),
       ('PROYECTO_D', 'TIPO_D'),
       ('PROYECTO_E', 'TIPO_E'),
       ('PROYECTO_F', 'TIPO_F'),
       ('PROYECTO_G', 'TIPO_G'),
       ('PROYECTO_H', 'TIPO_H');
```

```
go
```

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

go

```
insert into provincia (nombre, id_dep)
values ('Muñecas', 1),
       ('Quilacollo', 2),
       ('Carrasco', 3),
       ('Porvenir', 4),
       ('Cercado', 5),
       ('Chayanta', 6),
       ('Florida', 7),
       ('Murillo', 8);
```

go

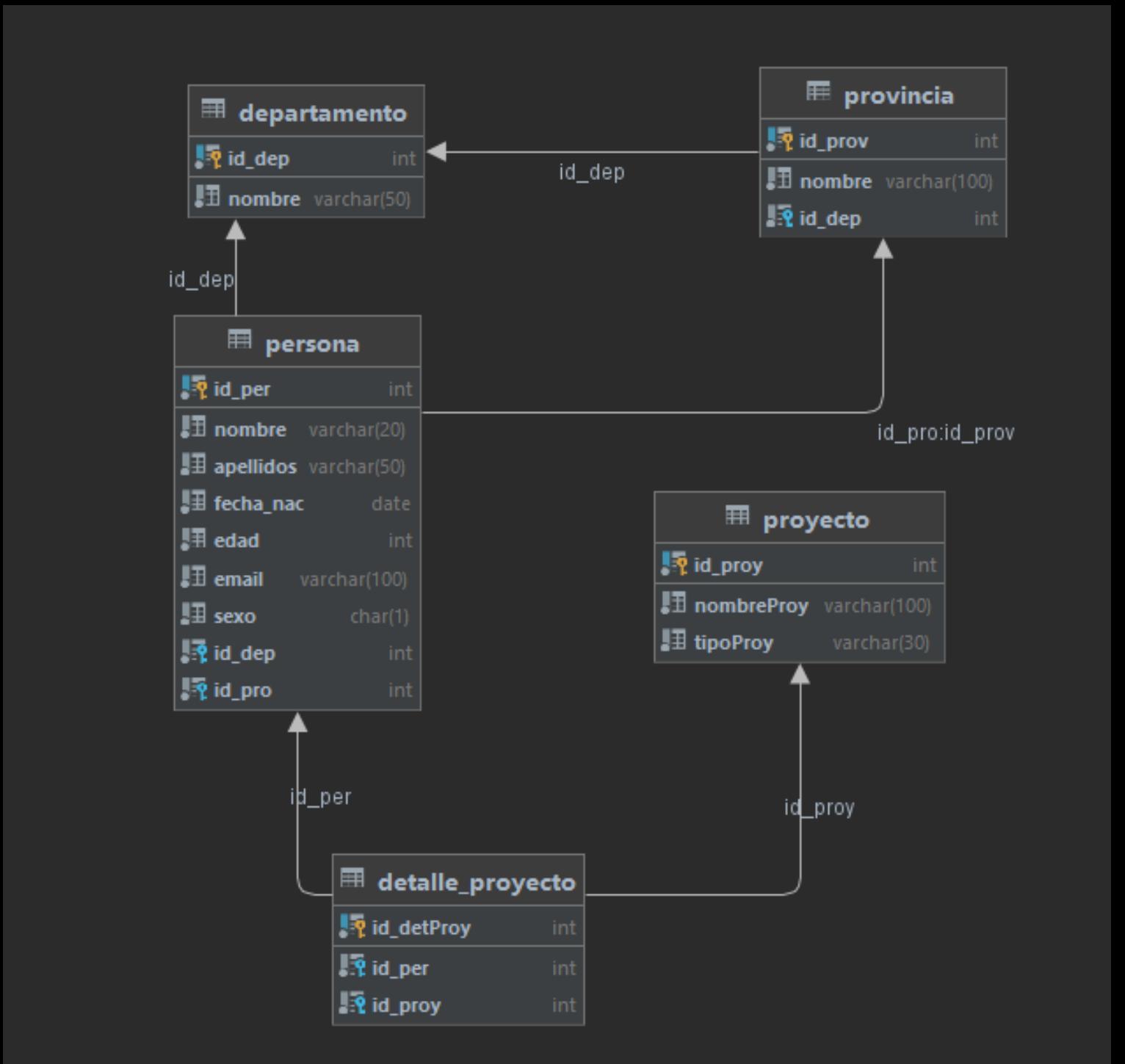
```
insert into persona (nombre, apellidos, fecha_nac, edad, email, sexo, id_dep,  
id_pro)  
values ('Martin', 'Mamani Pilco', '12-09-2003', 18, 'martinmn@gmail.com', 'M', 1, 1),  
('Alex', 'Callisaya Torrez', '5-04-2001', 20, 'alex45@gmail.com', 'M', 2, 2),  
('Lucero', 'Quispe Ticona', '9-06-2002', 19, 'susanqt@gmail.com', 'F', 3, 3),  
('Joel', 'Aranda Paz', '13-05-1998', 23, 'joel90a@gmail.com', 'M', 4, 4),  
('Henry', 'Apaza Hidalgo', '21-03-1991', 30, 'henapaza10@gmail.com', 'M', 5, 5),  
('Carla', 'Fernandez Paredes', '26-11-1997', 24, 'carlaFP@gmail.com', 'F', 6, 6),  
('Pedro', 'Mujica Cruz', '16-10-2000', 21, 'pedromcruz@gmail.com', 'M', 7, 7),  
('Anahi', 'Mamani Gonzales', '25-07-2003', 18, 'anaMG@gmail.com', 'F', 8, 8);
```

go

```
insert into detalle_proyecto (id_per, id_proy)  
values (1,1),  
(2,2),  
(3,3),  
(4,4),  
(5,5),  
(6,6),  
(7,7),  
(8,8);
```

go

MODELO LOGICO



MANEJO DE VISTAS

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

```
create view nombre_departamento as
    select per.*
        from persona as per
        inner join departamento as dep on per.id_dep = dep.id_dep
        where dep.nombre = 'Cochabamba'
```

```
select nomd.*
        from nombre_departamento as nomd
```

	id_per	nombre	apellidos	fecha_nac	edad	email	sexo	id_dep	id_pro		
1	2	Alex	Callisaya Torrez	5-04-2001	20	alex45@gmail.com	M	2	2		
2	3	Lucero	Quispe Ticona	9-06-2002	19	susanqt@gmail.com	F	3	3		

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

```
create view  
nombre_apellido_nomProy as  
    select per.nombre,  
per.apellidos, pro.nombreProy  
        from persona as per  
        inner join detalle_proyecto as  
dep on per.id_per = dep.id_per  
        inner join proyecto as pro on  
dep.id_proy = pro.id_proy
```

```
select nom_ap_pro.*  
from nombre_apellido_nomProy as  
nom_ap_pro
```

	nombre	apellidos	nombreProy
1	Martin	Mamani Pilco	PROYECTO_A
2	Alex	Callisaya Torrez	PROYECTO_B
3	Lucero	Quispe Ticona	PROYECTO_C
4	Joel	Aranda Paz	PROYECTO_D
5	Henry	Apaza Hidalgo	PROYECTO_E
6	Carla	Fernandez Paredes	PROYECTO_F
7	Pedro	Mujica Cruz	PROYECTO_G
8	Anahi	Mamani Gonzales	PROYECTO_H

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'

```
create view departamento_aplicarse as
  select detalle_aplicarse =
    case
      when proy.tipoProy = 'TIPO_A' then 'CBB'
      when proy.tipoProy = 'TIPO_B' then 'LPZ'
      when proy.tipoProy = 'TIPO_C' then 'SCZ'
      else 'En proceso de analisis'
    end
  from proyecto as proy
select dap.*
from departamento_aplicarse as dap
```

	detalle_aplicarse
1	CBB
2	LPZ
3	SCZ
4	En proceso de analisis
5	En proceso de analisis
6	En proceso de analisis
7	En proceso de analisis
8	En proceso de analisis

Crear una vista cualquiera que muestre 5 columnas.

```
create view datos_basicos as
    select per.nombre, per.apellidos,
per.edad, proy.nombreProy, proy.tipoProy
        from proyecto as proy
        inner join detalle_proyecto as depro on
proy.id_proy = depro.id_proy
        inner join persona as per on depro.id_per
= per.id_per
```

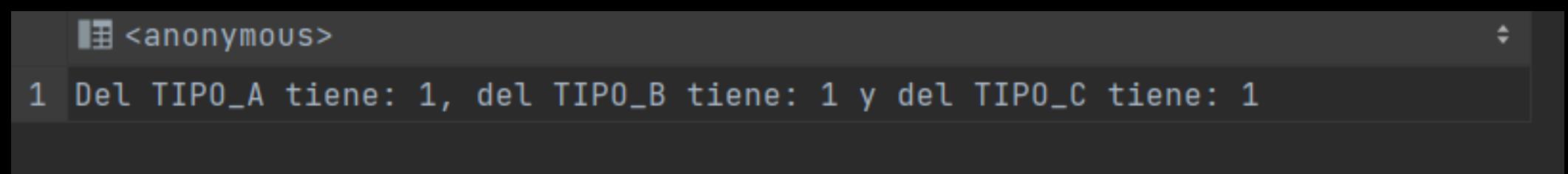
```
select dab.*
from datos_basicos as dab
```

	nombre	apellidos	edad	nombreProy	tipoProy
1	Martin	Mamani Pilco	18	PROYECTO_A	TIPO_A
2	Alex	Callisaya Torrez	20	PROYECTO_B	TIPO_B
3	Lucero	Quispe Ticona	19	PROYECTO_C	TIPO_C
4	Joel	Aranda Paz	23	PROYECTO_D	TIPO_D
5	Henry	Apaza Hidalgo	30	PROYECTO_E	TIPO_E
6	Carla	Fernandez Paredes	24	PROYECTO_F	TIPO_F
7	Pedro	Mujica Cruz	21	PROYECTO_G	TIPO_G
8	Anahi	Mamani Gonzales	18	PROYECTO_H	TIPO_H

Manejo de funciones

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

```
create or alter function tipos_proyectos ()  
returns varchar (100) as  
begin  
    declare @tipo_a int = 0  
    set @tipo_a = (  
        select count(pro.tipoProy)  
        from proyecto as pro  
        where pro.tipoProy = 'TIPO_A'  
    )  
    declare @tipo_b int = 0  
    set @tipo_b = (  
        select count(pro.tipoProy)  
        from proyecto as pro  
        where pro.tipoProy = 'TIPO_B'  
    )  
    declare @tipo_c int = 0  
    set @tipo_c = (  
        select count(pro.tipoProy)  
        from proyecto as pro  
        where pro.tipoProy = 'TIPO_C'  
    )  
    return concat('Del TIPO_A tiene: ', @tipo_a, ', del TIPO_B tiene: ',@tipo_b, ' y del TIPO_C  
tiene: ',@tipo_c)  
end
```

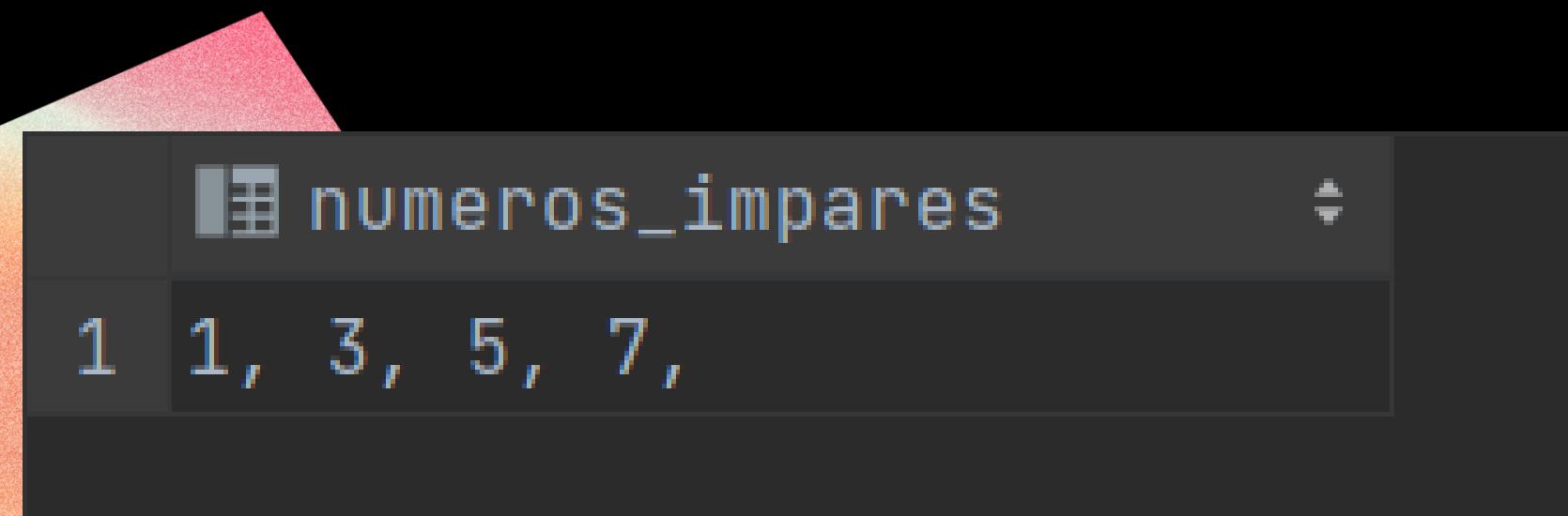


The screenshot shows a dark-themed database interface with a results grid. The header of the grid is labeled '<anonymous>'. The first and only row of data contains the number '1' followed by the text 'Del TIPO_A tiene: 1, del TIPO_B tiene: 1 y del TIPO_C tiene: 1'.

<anonymous>
1 Del TIPO_A tiene: 1, del TIPO_B tiene: 1 y del TIPO_C tiene: 1

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

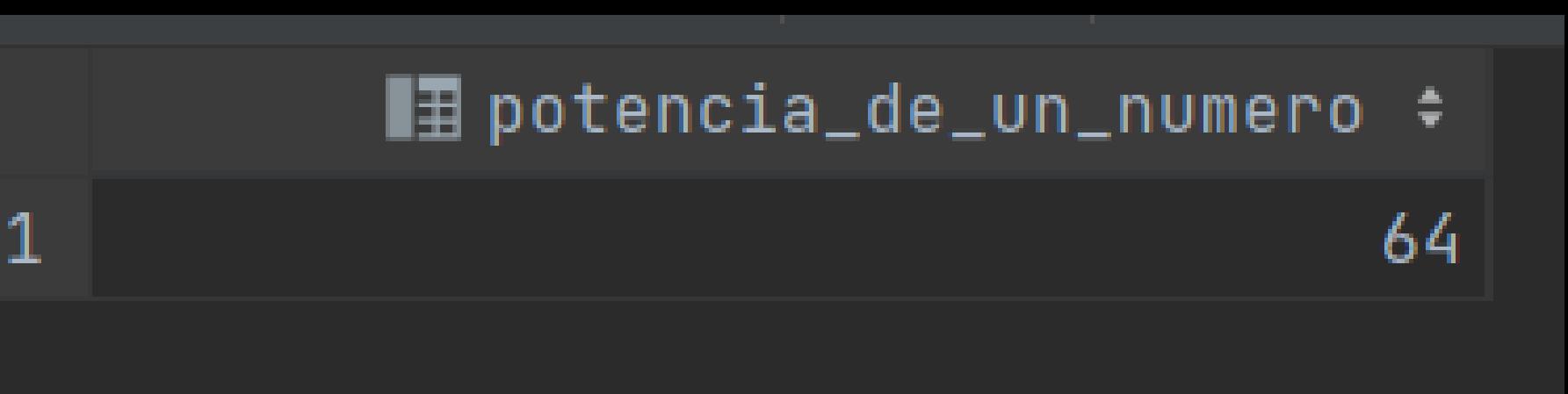


```
numeros_impares (@valorN int)
returns varchar (100) as
```

```
begin
    declare @contador int = 1
    declare @contador2 int = 1
    declare @concatenar varchar (100) =
    while @contador <= @valorN
        begin
            set @concatenar = concat(@concatenar, @contador2, ', ')
            set @contador2 = @contador2 + 2
            set @contador = @contador + 1
        end;
    return @concatenar
end
```

Crear una función cualquiera

```
create or alter function potencia_de_un_numero (@numero integer,  
@indice integer)  
returns integer as  
begin  
declare @cont int = 1;  
declare @aux int = 1;  
set @aux = @numero  
while @cont < @indice  
begin  
--set @aux = @aux * @numero  
set @aux = @aux * @numero  
set @cont = @cont + 1  
end  
return @aux  
end  
select dbo.potencia_de_un_numero(2, 6) as potencia_de_un_numero
```



The screenshot shows a database interface with a dark theme. At the top, there is a menu bar with several icons. Below the menu, the title of the current window is "potencia_de_un_numero". In the main area, there are two input fields: the first contains the value "1" and the second contains "64". To the right of these fields, the result of the query is displayed as "64".