



PROCESUAL HITO 4

Nombre: Daniel Escobar Saravia
Código: SIS11070582

Creación de la Base de Datos y sus insert

```
CREATE DATABASE ONG_H4;
```

```
USE ONG_H4;
```

```
CREATE TABLE departamento
```

```
( id_dep INTEGER PRIMARY KEY NOT NULL,
```

```
nombre VARCHAR(50) NOT NULL);
```

```
INSERT INTO departamento (id_dep, nombre)
```

```
VALUES (111, 'La Paz'),
```

```
(222, 'Cochabamba'),
```

```
(333, 'Santa Cruz');
```

```
CREATE TABLE provincia
```

```
( id_prov INTEGER PRIMARY KEY NOT NULL,
```

```
nombre VARCHAR(150) NOT NULL,
```

```
id_dep INTEGER NOT NULL,
```

```
FOREIGN KEY (id_dep) REFERENCES  
departamento(id_dep));
```

```
INSERT INTO provincia(id_prov, nombre, id_dep)
```

```
VALUES(112, 'Aroma', 111),
```


```
(113, 'Nor Yungas', 111),
```

```
(212, 'Punata', 222),
```

```
(213, 'Quillacollo', 222),
```

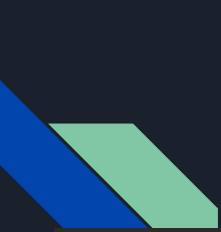
```
(312, 'Chiquitos', 333),
```

```
(313, 'Ichilo', 333);
```



```
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),
  sexo CHAR(1) NOT NULL,
  id_dep INTEGER NOT NULL,
  id_pro INTEGER NOT NULL,
  FOREIGN KEY (id_dep) REFERENCES departamento (id_dep),
  FOREIGN KEY (id_pro) REFERENCES provincia (id_prov));
```

```
INSERT INTO persona(id_per, nombre, apellidos, fecha_nac, edad,
email, sexo, id_dep, id_pro)
VALUES(1,'nombre1','apellido1','1995-06-05',26,'nombre1@gmail.com',
'M',111,112),
(2,'nombre2','apellido2','2002-06-07',19,'nombre2@gmail.com','F',1
11,112),
(3,'nombre3','apellido3','1990-02-15',31,'nombre3@gmail.com','F',1
11,113),
(4,'nombre4','apellido4','2000-01-15',21,'nombre4@gmail.com','M',1
11,113),
(5,'nombre5','apellido5','1999-05-24',22,'nombre5@gmial.com','M',2
22,212),
(6,'nombre6','apellido6','2001-08-10',20,'nombre6@gmial.com','F',2
22,212),
```



```
(7,'nombre7','apellido7','1998-01-26',23,'nombre7@gmail.com','F',222,213),
```

```
(8,'nombre8','apellido8','1995-07-14',26,'nombre8@gmail.com','F',222,213),
```

```
(9,'nombre9','apellido9','1996-04-12',27,'nombre9@gmail.com','M',333,312),
```

```
(10,'nombre10','apellido10','2002-05-16',19,'nombre10@gmail.com','M',333,312),
```

```
(11,'nombre11','apellido11','1997-06-14',24,'nombre11@gmail.com','F',333,313),
```

```
(12,'nombre12','apellido12','1998-06-25',23,'nombre12@gmail.com','F',333,313);
```

```
CREATE TABLE proyecto
```

```
(
```

```
    id_proy INTEGER PRIMARY KEY NOT NULL,
```

```
    nombreProy VARCHAR(100) NOT NULL,
```

```
    tipoProy VARCHAR(30) NOT NULL
```

```
);
```


```
INSERT INTO proyecto(id_proy, nombreProy, tipoProy)
```

```
VALUES (122,'Sendero Joven','Tipo A'),
```

```
        (123,'Por una sonrisa','Tipo B'),
```

```
        (124,'Ayuda al necesitado','Tipo B'),
```

```
        (125,'Por un buen futuro','Tipo C');
```



```
CREATE TABLE detalle_proyecto
```

```
( id_detProy INTEGER 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));
```

```
INSERT INTO detalle_proyecto(id_detProy, id_per,  
id_proy)
```

```
VALUES (01,1,122),
```

```
(02,2,123),
```

```
(03,3,122),
```

```
(04,4,123),
```

```
(05,5,124),
```

```
(06,6,125),
```

```
(07,7,124),
```

```
(08,8,124),
```

```
(09,9,125),
```

```
(010,10,122),
```

```
(011,11,123),
```

```
(012,12,124);
```

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

```
CREATE VIEW Cochabamba  
  
AS SELECT per.*, dep.nombre as  
departamento  
  
FROM persona AS per, departamento AS dep  
  
WHERE per.id_dep=222 and  
dep.nombre='Cochabamba';  
  
SELECT *  
  
FROM Cochabamba;
```

The screenshot shows a SQL IDE with a script editor and an output window. The script editor contains the following SQL code:

```
85 (012,12,124);  
86 CREATE VIEW Cochabamba  
87 AS SELECT per.*, dep.nombre as departamento  
88 FROM persona AS per, departamento AS dep  
89 WHERE per.id_dep=222 and dep.nombre='Cochabamba';  
90  
91 SELECT *  
92 FROM Cochabamba;  
93  
94 CREATE OR ALTER VIEW nombre_proyecto  
95 AS SELECT per.nombre, per.apellidos, pro.nombreProy AS nombre_proyecto  
96 FROM detalle_proyecto AS det  
97 INNER JOIN proyecto AS pro ON pro.id_proy = det.id_proy
```

The output window shows the results of the query executed in line 92. The results are displayed in a table with the following columns: id_per, nombre, apellidos, fecha_nac, edad, email, sexo, id_dep, id_pro, and departamento. The table contains 4 rows of data.

id_per	nombre	apellidos	fecha_nac	edad	email	sexo	id_dep	id_pro	departamento
1	5 nombre5	apellido5	1999-05-24	22	nombre5@gmail.com	M	222	212	Cochabamba
2	6 nombre6	apellido6	2001-08-10	20	nombre6@gmail.com	F	222	212	Cochabamba
3	7 nombre7	apellido7	1998-01-26	23	nombre7@gmail.com	F	222	213	Cochabamba
4	8 nombre8	apellido8	1995-07-14	26	nombre8@gmail.com	F	222	213	Cochabamba

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

```
CREATE OR ALTER VIEW nombre_proyecto  
  
AS SELECT per.nombre,per.apellidos,pro.nombreProy AS  
nombre_proyecto  
  
FROM detalle_proyecto AS det  
  
INNER JOIN proyecto AS pro ON pro.id_proy = det.id_proy  
  
INNER JOIN persona AS per ON per.id_per=det.id_per;  
  
SELECT *  
  
FROM nombre_proyecto ;
```

```
94 CREATE OR ALTER VIEW nombre_proyecto  
95 AS SELECT per.nombre,per.apellidos,pro.nombreProy AS nombre_proyecto  
96 FROM detalle_proyecto AS det  
97 INNER JOIN proyecto AS pro ON pro.id_proy = det.id_proy  
98 INNER JOIN persona AS per ON per.id_per=det.id_per;  
99  
100 SELECT *  
101 FROM nombre_proyecto;  
102
```

Output x ONG_H4.dbo.nombre_proyecto x

	nombre	apellidos	nombre_proyecto
1	nombre1	apellido1	Sendero Joven
2	nombre2	apellido2	Por una sonrisa
3	nombre3	apellido3	Sendero Joven
4	nombre4	apellido4	Por una sonrisa
5	nombre5	apellido5	Ayuda al necesitado
6	nombre6	apellido6	Por un buen futuro
7	nombre7	apellido7	Ayuda al necesitado
8	nombre8	apellido8	Ayuda al necesitado
9	nombre9	apellido9	Por un buen futuro
10	nombre10	apellido10	Sendero Joven
11	nombre11	apellido11	Por una sonrisa
12	nombre12	apellido12	Ayuda al necesitado

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

```
CREATE VIEW Location proyecto
AS SELECT pro.*, departameno_aplicarse =
case
when pro.tipoProy='Tipo A' then 'CBB'
when pro.tipoProy='Tipo B' then 'LPZ'
when pro.tipoProy='Tipo C' then 'SCZ'
else 'En proceso de análisis'
end
FROM proyecto AS pro
```

```
SELECT *
FROM Location_proyecto;
```

```
103 CREATE VIEW Location_proyecto
104 AS SELECT pro.*, departameno_aplicarse =
105 case
106 when pro.tipoProy='Tipo A' then 'CBB'
107 when pro.tipoProy='Tipo B' then 'LPZ'
108 when pro.tipoProy='Tipo C' then 'SCZ'
109 else 'En proceso de análisis'
110 end
111 FROM proyecto AS pro
112
113 SELECT *
114 FROM Location_proyecto;
115
```

EQ Location_proyecto

Output x ONG_H4.dbo.Location_proyecto x

< 4 rows > Tx: Auto DDL

	id_proy	nombreProy	tipoProy	departameno_aplicarse
1	122	Sendero Joven	Tipo A	CBB
2	123	Por una sonrisa	Tipo B	LPZ
3	124	Ayuda al necesitado	Tipo B	LPZ
4	125	Por un buen futuro	Tipo C	SCZ

Crear una vista cualquiera que muestre 5 columnas

```
CREATE OR ALTER VIEW Encargado_de_departamento
```

```
AS SELECT per.nombre, per.apellidos, dep.nombre AS departamento, prov.nombre AS provincia, per.id_pro
```

```
FROM provincia AS prov
```

```
INNER JOIN departamento AS dep ON dep.id_dep =  
prov.id_dep
```

```
INNER JOIN persona AS per ON per.id_pro=prov.id_prov;
```

```
SELECT *
```

```
FROM Encargado_de_departamento
```

```
115  
116 CREATE OR ALTER VIEW Encargado_de_departamento  
117 AS SELECT per.nombre, per.apellidos, dep.nombre AS departamento, prov.nombre AS provincia, per.id_pro  
118 FROM provincia AS prov  
119 INNER JOIN departamento AS dep ON dep.id_dep = prov.id_dep  
120 INNER JOIN persona AS per ON per.id_pro=prov.id_prov;  
121  
122 ✓ SELECT *  
123 FROM Encargado_de_departamento;
```

Output x ONG_H4.dbo.Encargado_de_departamento x

	nombre	apellidos	departamento	provincia	id_pro
1	nombre1	apellido1	La Paz	Aroma	112
2	nombre2	apellido2	La Paz	Aroma	112
3	nombre13	apellido13	La Paz	Aroma	112
4	nombre3	apellido3	La Paz	Nor Yungas	113
5	nombre4	apellido4	La Paz	Nor Yungas	113
6	nombre5	apellido5	Cochabamba	Punata	212
7	nombre6	apellido6	Cochabamba	Punata	212
8	nombre7	apellido7	Cochabamba	Quillacollo	213
9	nombre8	apellido8	Cochabamba	Quillacollo	213
10	nombre9	apellido9	Santa Cruz	Chiquitos	312
11	nombre10	apellido10	Santa Cruz	Chiquitos	312
12	nombre11	apellido11	Santa Cruz	Ichilo	313
13	nombre12	apellido12	Santa Cruz	Ichilo	313

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

```
CREATE OR ALTER FUNCTION numero_de_proyectos_D_E_F()  
  
    RETURNS INTEGER AS  
  
    BEGIN  
  
        DECLARE @contador INTEGER;  
  
        SELECT @contador = COUNT(*)  
  
        FROM proyecto AS pro  
  
        WHERE pro.tipoProy!='Tipo A' AND pro.tipoProy!='Tipo B'  
        AND pro.tipoProy!='Tipo C';  
  
        RETURN @contador;  
  
    END;  
  
select dbo.numero_de_proyectos_D_E_F();
```

```
125 CREATE OR ALTER FUNCTION numero_de_proyectos_D_E_F()  
126     RETURNS INTEGER AS  
127     BEGIN  
128         DECLARE @contador INTEGER;  
129         SELECT @contador = COUNT(*)  
130         FROM proyecto AS pro  
131         WHERE pro.tipoProy!='Tipo A' AND pro.tipoProy!='Tipo B' AND pro.tipoProy!='Tipo C';  
132         RETURN @contador;  
133     END;  
134  
135  
136 select dbo.numero_de_proyectos_D_E_F();  
137  
138 CREATE OR ALTER PROCEDURE añadir_registros @id_per INT ,@nombre VARCHAR(20),@apellido VARCHAR(50),@n  
139 AS  
    numero_de_proyectos_D_E_F()
```

Output x dbo.numero_de_proyectos_D_E_F()int x

< 1 row > | ↺ | ↻ | ↗

<anonymous> ▾

1	0
---	---

Crear una función que genere los primeros N números impares.

```
CREATE OR ALTER FUNCTION numeros_impares(@parametro1 INT)
RETURNS VARCHAR(100) AS
BEGIN
    DECLARE @respuesta VARCHAR(100) = '';
    DECLARE @contador INTEGER = 1;

    WHILE @contador <= @parametro1
    BEGIN
        SET @respuesta = CONCAT(@respuesta, @contador, ', ');
        SET @contador = @contador + 2;
    END;

    RETURN @respuesta;

select [dbo].[numeros_impares](11);
```

```
138 CREATE OR ALTER FUNCTION numeros_impares(@parametro1 INT)
139 RETURNS VARCHAR(100) AS
140 BEGIN
141     DECLARE @respuesta VARCHAR(100) = '';
142     DECLARE @contador INTEGER = 1;
143
144     WHILE @contador <= @parametro1
145     BEGIN
146         SET @respuesta = CONCAT(@respuesta, @contador, ', ');
147         SET @contador = @contador + 2;
148     END;
149     RETURN @respuesta;
150 END;
151 select [dbo].[numeros_impares]( @parametro1: 11);
152
153 CREATE OR ALTER PROCEDURE añadir_registros @id_per INT, @nombre VARCHAR(20), @apellido
154 AS
155 BEGIN
    numeros_impares()
```

Output × [dbo].[numeros_impares](11):varchar(100) ×

1 row

<anonymous>

1	1, 3, 5, 7, 9, 11,
---	--------------------

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

```
CREATE OR ALTER PROCEDURE añadir_registros @id_per INT ,@nombre VARCHAR(20),@apellido
VARCHAR(50) ,@nacimiento DATE,@edad INT, @email VARCHAR (100),@sexo CHAR(1),@id_dep INT,
@id_pro INT

AS

BEGIN

    INSERT INTO persona

    VALUES (@id_per,@nombre,@apellido,@nacimiento,@edad,@email,@sexo,@id_dep,@id_pro) ;

END;

EXEC añadir_registros
13,'nombre13','apellido13','1990-02-20',31,'nombre13@gmail.com','F',111,112;
```

Crear una función cualquiera

```
CREATE OR ALTER FUNCTION potencia(@base  
INT,@exponente INT) RETURNS INT AS BEGIN
```

```
DECLARE @respuesta INT=1
```

```
DECLARE @contador INTEGER = 1;
```

```
WHILE @contador <= @exponente
```

```
BEGIN SET @respuesta=@respuesta*@base ;
```

```
SET @contador = @contador + 1;
```

```
END;
```

```
RETURN @respuesta ;
```

```
END;
```

```
select [dbo].[potencia](11,2);
```

```
161 CREATE OR ALTER FUNCTION potencia(@base INT,@exponente INT)
162 RETURNS INT AS
163 BEGIN
164     DECLARE @respuesta INT=1
165     DECLARE @contador INTEGER = 1;
166
167     WHILE @contador <= @exponente
168     BEGIN
169         SET @respuesta=@respuesta*@base;
170         SET @contador = @contador + 1;
171     END;
172
173     RETURN @respuesta;
174
175 END;
176 select [dbo].[potencia]( @base: 11, @exponente: 2);
177
178
```

añadir_registros() potencia()

Output x [dbo].[potencia](11,2):int x

<< < 1 row > > > ↺ ⌵ ↗

<anonymous>

1	121
---	-----