

Bases de Datos- SQL Server

Procesual Hito 4

Estudiante: Karla Belen Diaz Flores
Ingeniero: William Roddy Barra Paredes
Semestre: Segundo
Gestión:

2021



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_1;
USE ong_h4_1;
CREATE TABLE proyecto
(
    id_proy INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombreProy VARCHAR(100),
    tipoProy VARCHAR(30)
);

CREATE TABLE detalle_proyecto
(
    id_detProy INTEGER IDENTITY PRIMARY KEY NOT NULL,
    id_per INTEGER,
    id_proy INTEGER,
    FOREIGN KEY (id_per) REFERENCES persona (id_per),
    FOREIGN KEY (id_proy) REFERENCES proyecto (id_proy)
);
```

```
CREATE TABLE persona
(
    id_per INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombre VARCHAR(20),
    apellidos VARCHAR(50),
    fecha_nac DATE,
    edad INTEGER NOT NULL,
    email VARCHAR(100),
    sexo CHAR(1),
    id_dep INTEGER,
    id_pro INTEGER,
    FOREIGN KEY (id_dep) REFERENCES departamento (id_dep),
    FOREIGN KEY (id_pro) REFERENCES provincia (id_prov)
);

CREATE TABLE provincia
(
    id_prov INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombreProv VARCHAR(150),
    id_dep INTEGER,
    FOREIGN KEY (id_dep) REFERENCES departamento (id_dep)
);

CREATE TABLE departamento
(
    id_dep INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombreDepa VARCHAR(50),
);
```

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

```
INSERT INTO departamento (nombreDepa) VALUES  
( 'Cochabamba');  
INSERT INTO departamento (nombreDepa) VALUES ('La Paz');  
INSERT INTO departamento (nombreDepa) VALUES ('Santa Cruz');  
INSERT INTO departamento (nombreDepa) VALUES ('Beni');  
INSERT INTO departamento (nombreDepa) VALUES ('Pando');
```

```
INSERT INTO provincia (nombreProv, id_dep) VALUES ('Quillacollo',  
1);  
INSERT INTO provincia (nombreProv, id_dep) VALUES ('Sacaba', 1);  
INSERT INTO provincia (nombreProv, id_dep) VALUES ('Mizque', 1);  
INSERT INTO provincia (nombreProv, id_dep) VALUES ('Murillo', 2);  
INSERT INTO provincia (nombreProv, id_dep) VALUES ('Robore', 3);
```

```
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Sembrado  
de Arboles', 'TIPO_A');  
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Estudio  
de Semillas', 'TIPO_B');  
INSERT INTO proyecto (nombreProy, tipoProy) VALUES  
( 'Alfabetizacion', 'TIPO_C');  
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Creacion  
de Escuelas', 'TIPO_D');  
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Apoyo al  
dibujo', 'TIPO_E');
```

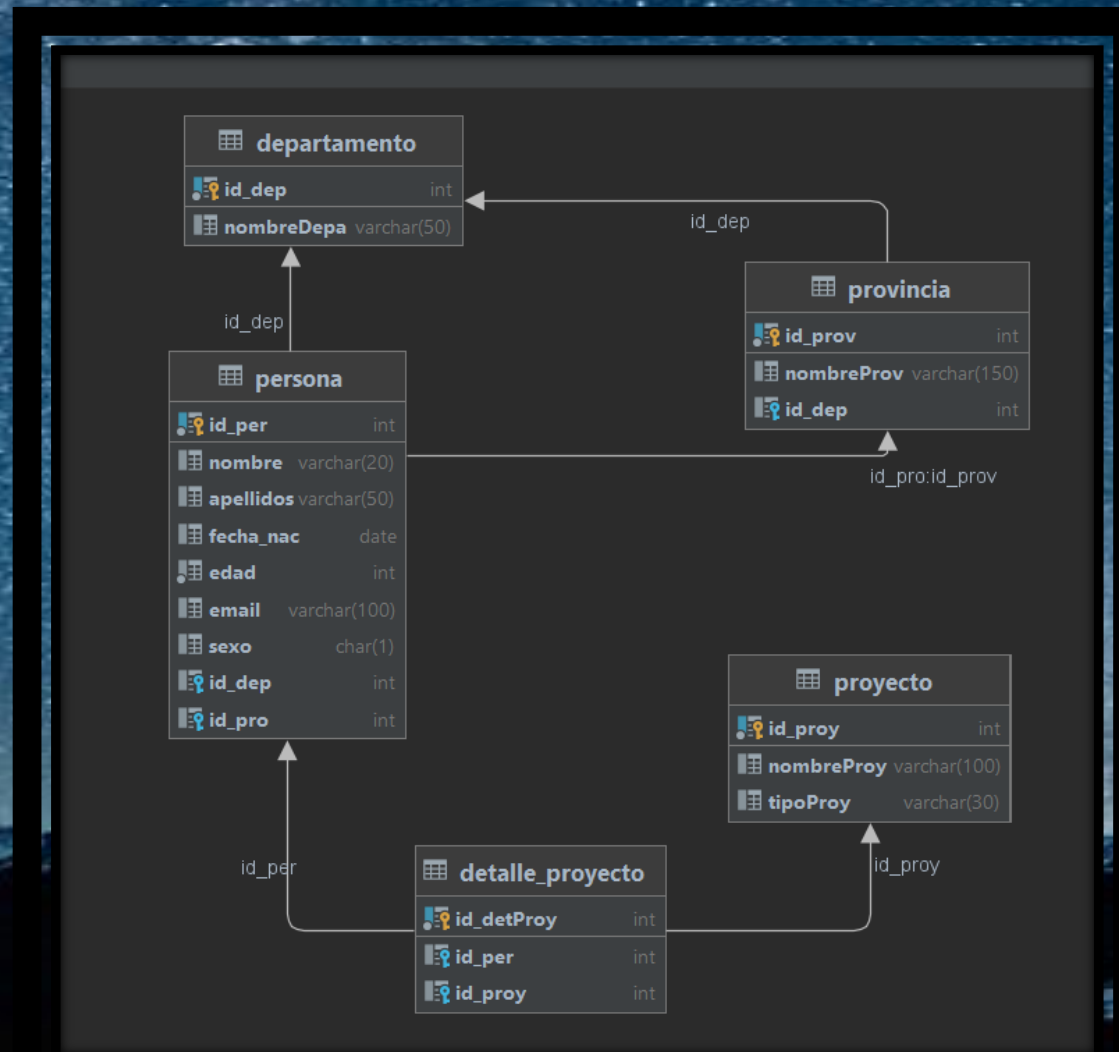
```
INSERT INTO persona (nombre, apellidos, fecha_nac, edad, email, sexo,  
id_dep, id_pro) VALUES  
( 'JUAN', 'TRUJILLO VASQUEZ', '1990-10-30', 28,  
'juan@gmail.com', 'm', 1, 1),  
( 'JOSE', 'GUTIERREZ COPA', '1992-10-30', 28,  
'jose@gmail.com', 'm', 1, 2),  
( 'LIDIA', 'SALINAS VELARDE', '1994-10-30', 26,  
'lidia3@gmail.com', 'f', 1, 3),  
( 'JOEL', 'CASTILLO FLORES', '1996-10-30', 24,  
'joel4@gmail.com', 'm', 2, 4),  
( 'LUNA', 'CAMPOS CACERES', '1992-10-30', 28,  
'luna@gmail.com', 'f', 3, 5),  
( 'CAROLA', 'SALAS SANCHEZ', '1999-10-30', 19,  
'carola@gmail.com', 'f', 3, 5);
```

```
INSERT INTO detalle_proyecto (id_per, id_proy) VALUES  
(1, 1),  
(2, 1),  
(3, 1),  
(3, 2),  
(4, 3),  
(4, 4),  
(5, 5),  
(6, 5);
```


Diseño de base de datos.

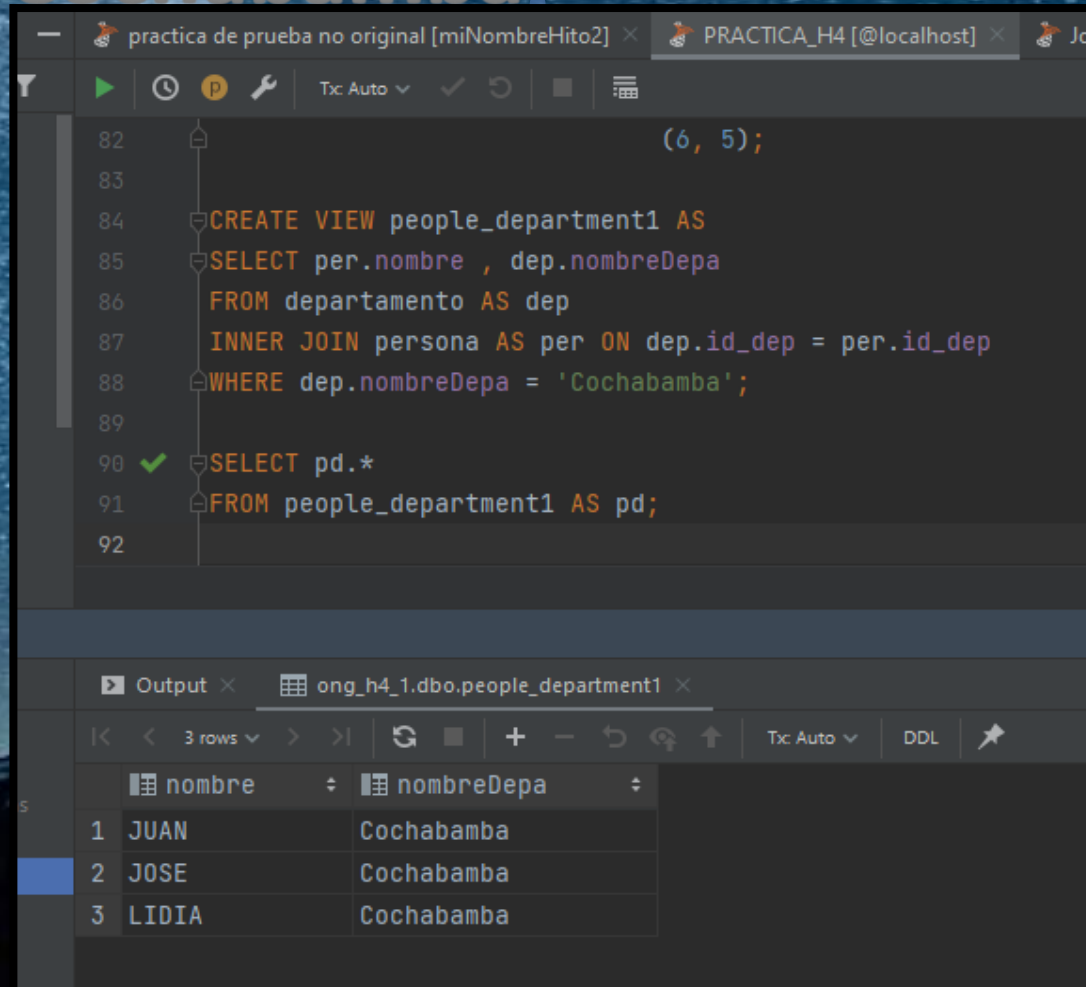
1.2. Después de generar el modelo lógico de la base de datos.

- Utilizar Datagrip para poder generar el diagrama



Manejo de Vistas

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



The screenshot shows a SQL IDE with two tabs: 'practica de prueba no original [miNombreHito2]' and 'PRACTICA_H4 [@localhost]'. The active tab displays SQL code for creating and querying a view. The code is as follows:

```
82      (6, 5);
83
84  CREATE VIEW people_department1 AS
85  SELECT per.nombre , dep.nombreDepa
86  FROM departamento AS dep
87  INNER JOIN persona AS per ON dep.id_dep = per.id_dep
88  WHERE dep.nombreDepa = 'Cochabamba';
89
90  SELECT pd.*
91  FROM people_department1 AS pd;
92
```

Below the code editor, the 'Output' tab is active, showing the results of the query. The output is a table with two columns: 'nombre' and 'nombreDepa'. It contains three rows of data:

	nombre	nombreDepa
1	JUAN	Cochabamba
2	JOSE	Cochabamba
3	LIDIA	Cochabamba

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

```
90 SELECT pu.*
91 FROM people_department1 AS pd;
92
93 //Mostrar la persona (nombres y apellidos) y el nombre del proyecto en donde trabajan.
94
95 CREATE OR ALTER VIEW view_project_name AS
96 SELECT nombre AS name, apellidos AS last_name, nombreProy AS projet_name
97 FROM persona AS per
98 INNER JOIN proyecto AS det ON per.id_pro = det.id_proy;
99
100 ✓ SELECT vpn.*
101 FROM view_project_name AS vpn;
102
```

Output x ong_h4_1.dbo.view_project_name x

	name	last_name	projet_name
1	JUAN	TRUJILLO VASQUEZ	Sembrado de Arboles
2	JOSE	GUTIERREZ COPA	Estudio de Semillas
3	LIDIA	SALINAS VELARDE	Alfabetizacion
4	JOEL	CASTILLO FLORES	Creacion de Escuelas
5	LUNA	CAMPOS CACERES	Apoyo al dibujo
6	CAROLA	SALAS SANCHEZ	Apoyo al dibujo

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'

```
109
110 CREATE OR ALTER VIEW view_project_type AS
111     SELECT proy.id_proy, proy.nombreProy, proy.tipoProy, departamento_aplicarse =
112         CASE
113             WHEN proy.tipoProy = 'TIPO_A' THEN 'LA PAZ'
114             WHEN proy.tipoProy = 'TIPO_B' THEN 'ORURO'
115             WHEN proy.tipoProy = 'TIPO_C' THEN 'SUCRE'
116             ELSE 'En proceso de análisis'
117         END
118     FROM proyecto AS proy;
119
120 ✓ SELECT vpt.*
121 FROM view_project_type AS vpt;
122 |
123 //Crear una vista cualquiera que muestre 5 columnas
124
```

Output x ong_h4_1.dbo.view_project_type x				
5 rows				
	id_proy	nombreProy	tipoProy	departamento_aplicarse
1	1	Sembrado de Arboles	TIPO_A	LA PAZ
2	2	Estudio de Semillas	TIPO_B	ORURO
3	3	Alfabetizacion	TIPO_C	SUCRE
4	4	Creacion de Escuelas	TIPO_D	En proceso de análisis
5	5	Apoyo al dibujo	TIPO_E	En proceso de análisis

2.4. Crear una vista cualquiera que muestre 5 columnas..

```
122
123  //Crear una vista cualquiera que muestre 5 columnas
124
125  CREATE OR ALTER VIEW view_columns AS
126  SELECT per.id_per, per.nombre, per.apellidos, per.fecha_nac, per.edad
127  FROM persona AS per;
128
129  ✓ SELECT vc.*
130  FROM view_columns AS vc;
131  |
```

Output × ong_h4_1.dbo.view_columns ×

	id_per	nombre	apellidos	fecha_nac	edad
1	1	JUAN	TRUJILLO VASQUEZ	1990-10-30	28
2	2	JOSE	GUTIERREZ COPA	1992-10-30	28
3	3	LIDIA	SALINAS VELARDE	1994-10-30	26
4	4	JOEL	CASTILLO FLORES	1996-10-30	24
5	5	LUNA	CAMPOS CACERES	1992-10-30	28
6	6	CAROLA	SALAS SANCHEZ	1999-10-30	19

Manejo de Funciones

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

```
134  // Crear una función que genere los primeros N números impares
135  CREATE OR ALTER FUNCTION odd_numbers (@parametro1 INT)
136      RETURNS VARCHAR(80) AS
137      BEGIN
138
139          DECLARE @respuesta VARCHAR(80) = '';
140          DECLARE @contador INTEGER = 0;
141
142          WHILE @contador <= @parametro1
143          BEGIN
144
145              IF @contador % 2 <> 0
146              BEGIN
147                  SET @respuesta = CONCAT(@respuesta, @contador, ', ');
148              END;
149
150              SET @contador = @contador + 1;
151          END;
152
153          RETURN @respuesta;
154      END;
155
156
157  SELECT dbo.odd_numbers ( @parametro1: 5);
```

Output x dbo.odd_numbers (5):varchar(80) x

1 row

<anonymous>

1	1, 3, 5,
---	----------

. 3.4. Crear una función cualquiera.

```
158
159  //Crear una función cualquiera.
160
161  CREATE OR ALTER FUNCTION multiplicacion_numeros(@parametro1 INT, @parametro2 INT)
162  RETURNS INTEGER
163  AS
164  BEGIN
165      DECLARE @respuesta INTEGER = 0;
166
167      SET @respuesta = @parametro1 * @parametro2 ;
168
169      RETURN @respuesta;
170  END;
171
172  ✓ SELECT dbo.multiplicacion_numeros( @parametro1: 20, @parametro2: 4);
173
174
```

Output × dbo.multiplicacion_numeros(20,4):int ×

1 row ▾ ↺ ⏏ ⚡

<anonymous> ▾

1	80
---	----

A cartoon illustration of Naruto Uzumaki from the anime Naruto. He is shown from the chest up, wearing his signature orange jumpsuit under a dark grey hoodie. He has his characteristic spiky blonde hair and a grey forehead protector with the Konoha symbol. He is giving a thumbs-up gesture with his right hand. The background is a blurred forest scene with green foliage and brown tree trunks.

¡GRACIAS POR SU ATENCIÓN!

https://drive.google.com/file/d/1W90sMm9ggQezlbq_zt-Pe9IsoTRNG_Ep/view?usp=sharing