



PROCESUAL HITO 4

BASE DE DATOS I

ESTUDIANTE: BRITTANY IBLING MARINO
QUISPE

UNIVERSIDAD: FRANZ TAMAYO

CARRERA: INGENIERIA DE SISTEMAS

CODIGO: SIS13181271

1.- DISEÑO DE BASE DE DATOS

1.1 DADO EL DETALLE EXPLICADO EN LA PARTE INICIAL DE ESTE DOCUMENTO DEBERIA GENERAR UNA BASE DE DATOS SIMILAR AL SIGUIENTE.

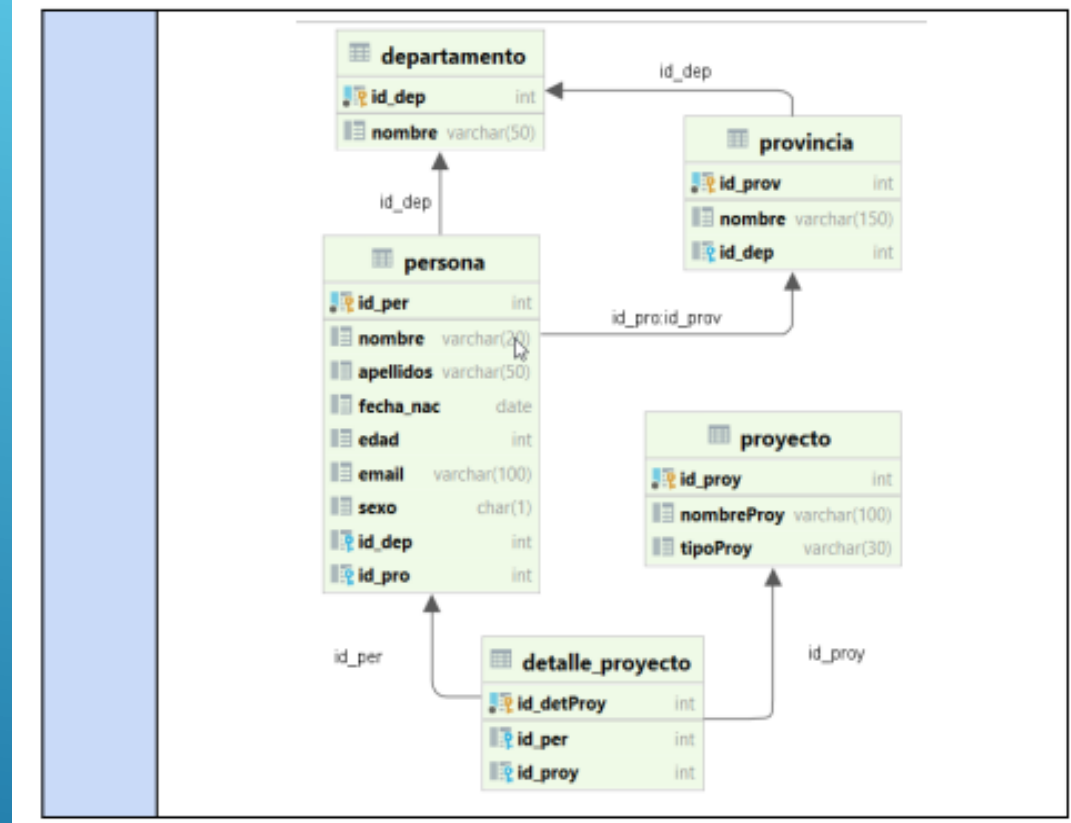
Consigna

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.

Detalles puntuales y análisis del problema

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.



Ejecute el comando CREATE DATABASE ONG_H4

Posicionarnos en la base que queremos trabajar. Ejecute el comando: USE ONG_H4

```
1 CREATE DATABASE ONG_H4
2 USE ONG_H4
3
4 CREATE TABLE departamento
5 (
6     id_dep int PRIMARY KEY,
7     nombre varchar(50),
8
9 );
10
11 CREATE TABLE provincia
12 (
13     id_prov int PRIMARY KEY,
14     nombre varchar(150),
15     id_dep int,
16     FOREIGN KEY (id_dep) REFERENCES departamento(id_dep)
17 );
18
19 CREATE TABLE proyecto
20 (
21     id_proy int PRIMARY KEY,
```

```
CREATE TABLE proyecto
(
    id_proy int PRIMARY KEY,
    nombreproy varchar(100),
    tipoproj varchar(30),
);

CREATE TABLE persona
(
    id_per int PRIMARY KEY,
    nombre varchar(20),
    apellidos varchar(50),
    fecha_nac date,
    edad int,
    email varchar (100),
    sexo char(1),
    id_dep int,
    id_pro int,
    FOREIGN KEY (id_dep) REFERENCES departamento(id_dep),
    FOREIGN KEY (id_pro) REFERENCES proyecto(id_proy)
```

Para crear una tabla me apoye en la clausula: CREATE TABLE nombre_tabla(...);
Como agregar registros a una tabla. Me apoye en la clausula INSERT INTO nombre_tabla

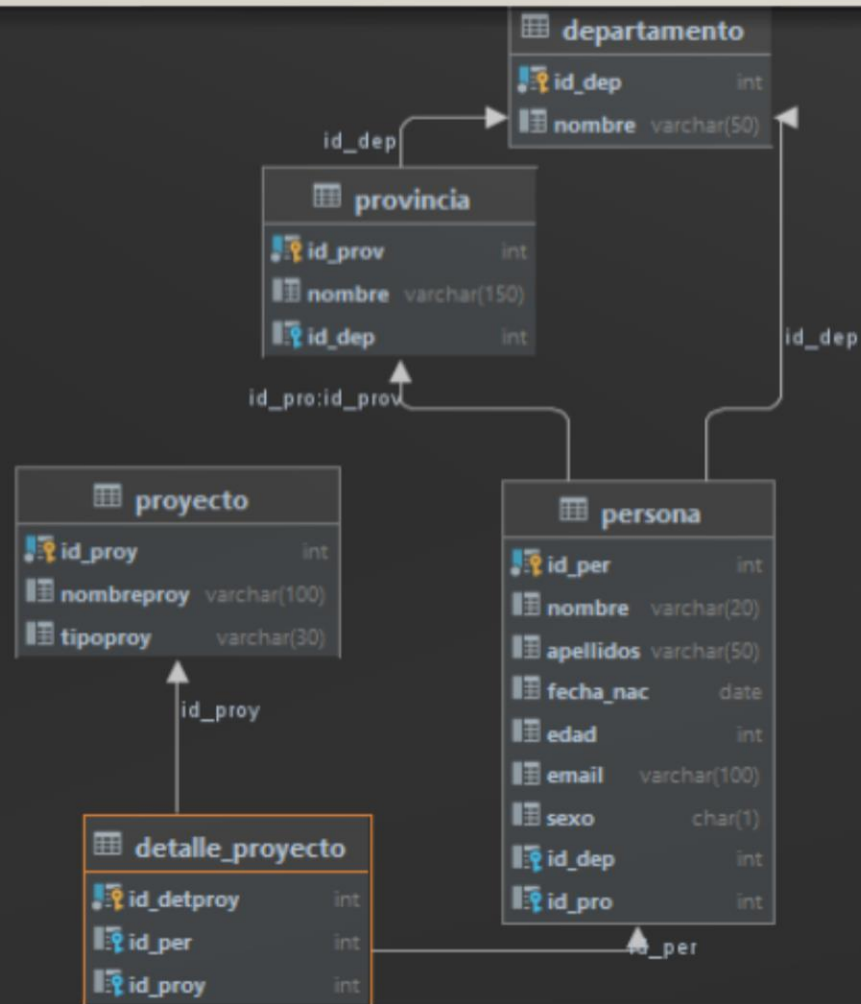
```
id_pro int,  
FOREIGN KEY (id_dep) REFERENCES departamento(id_dep),  
FOREIGN KEY (id_pro) REFERENCES provincia (id_prov),  
);  
  
CREATE TABLE detalle_proyecto  
(  
    id_detproy int PRIMARY KEY,  
    id_per      int,  
    id_proy     int,  
    FOREIGN KEY (id_per) REFERENCES persona (id_per),  
    FOREIGN KEY (id_proy) REFERENCES proyecto (id_proy),  
);  
  
INSERT INTO departamento(id_dep,nombre)  
VALUES ('35','La paz');  
INSERT INTO departamento(id_dep,nombre)  
VALUES ('36','Cochabamba');  
INSERT INTO departamento(id_dep,nombre)  
VALUES ('37','Santa Cruz');
```

```
INSERT INTO provincia (id_prov,nombre,id_dep)  
VALUES ('53','Caranavi','35');  
INSERT INTO provincia (id_prov,nombre,id_dep)  
VALUES ('63','Chapare','36');  
INSERT INTO provincia (id_prov,nombre,id_dep)  
VALUES ('73','Chiquitos','37');  
  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('14','Robot','Tipo_A');  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('15','Brazo robotico','Tipo_A');  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('16','germinar semillas en caja de cd','Tipo_B');  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('17','Espectroscopio casero','Tipo_B');  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('18','Crear mundos virtuales','Tipo_C');  
INSERT INTO proyecto (id_proy,nombreproy,tipoproj)  
VALUES ('19','Base de datos de una tienda','Tipo_C');
```

```
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1318','Carla','Flores','2002-05-15','19','carla@gmail.com','f','35','53');
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1419','Elias','Jarro','2002-05-04','19','Elias@gmail.com','m','35','53');
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1520','Ibling','Fernandez','2000-05-22','21','Ibling@gmail.com','f','36','63');
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1621','Jhonatan','Limachi','2001-05-27','20','Jhonatan@gmail.com','m','36','63');
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1722','Carlos','Vargas','2002-01-17','19','carlos@gmail.com','m','37','73');
INSERT INTO persona (id_per,nombre,apellidos,fecha_nac,edad,email,sexo,id_dep,id_pro)
VALUES ('1823','Iban','Gonzales','2001-08-29','20','Iban@gmail.com','m','37','73');
```

```
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('71','1318','14');
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('72','1419','15');
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('73','1520','16');
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('74','1621','17');
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('75','1722','18');
INSERT INTO detalle_proyecto (id_detproy,id_per,id_proy)
VALUES ('76','1823','19');
```

DIAGRAMA



2. Manejo de Vistas

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

```
108 ✓ SELECT per.nombre, per.apellidos, dep.nombre
109      FROM departamento AS dep
110      INNER JOIN persona AS per ON dep.id_dep = per.id_dep
111      WHERE per.id_dep = 36;
```

Output ×

Result 1 ×

2 rows ▾

↺ ↻ ⚡

	per.nombre	apellidos	dep.nombre
1	Ibling	Fernandez	Cochabamba
2	Jhonatan	Limachi	Cochabamba

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

```
114 ✓ SELECT pro.nombreproy, per.nombre, per.apellidos
115 FROM detalle_proyecto AS det
116     INNER JOIN proyecto AS pro ON det.id_proy = pro.id_proy
117     INNER JOIN persona AS per ON det.id_per = per.id_per
118 WHERE per.id_pro < 80;
119
```

Output × Result 2 ×

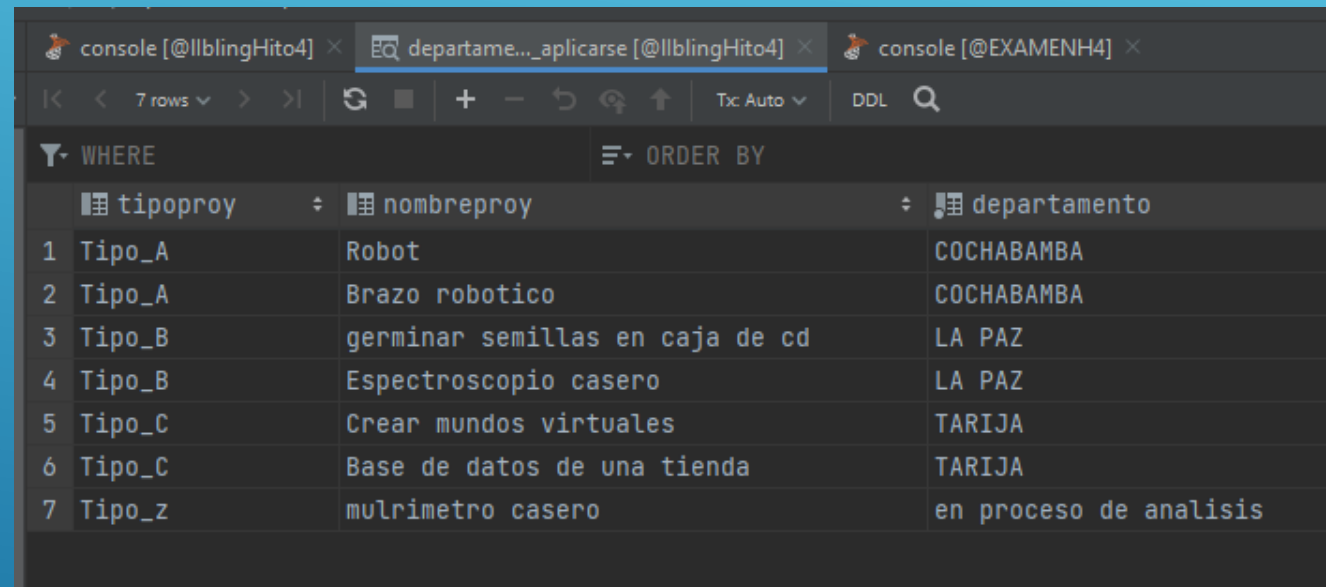
6 rows

	nombreproy	nombre	apellidos
1	Robot	Carla	Flores
2	Brazo robotico	Elias	Jarro
3	germinar semillas en caja de cd	Ibling	Fernandez
4	Espectroscopio casero	Jhonatan	Limachi
5	Crear mundos virtuales	Carlos	Vargas
6	Base de datos de una tienda	Iban	Gonzales

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'

```
123
124 CREATE OR ALTER VIEW departamento_aplicarse AS
125     SELECT tipoproy, nombreproy, departamento=
126         CASE
127             WHEN tipoproy = 'TIPO_A' THEN 'COCHABAMBA'
128             WHEN tipoproy= 'TIPO_B' THEN 'LA PAZ'
129             WHEN tipoproy= 'TIPO_C' THEN 'TARIJA'
130             ELSE 'en proceso de analisis'
131         END
132     FROM proyecto AS nombreproy;
133 ✓ SELECT bn.*
134 FROM departamento_aplicarse AS bn;
135
```



The screenshot shows a database console with three tabs: 'console [@@lblingHito4]', 'EQ departame..._aplicarse [@@lblingHito4]', and 'console [@@EXAMENH4]'. The active tab displays the results of a query. The table has three columns: 'tipoproy', 'nombreproy', and 'departamento'. There are 7 rows of data.

	tipoproy	nombreproy	departamento
1	Tipo_A	Robot	COCHABAMBA
2	Tipo_A	Brazo robotico	COCHABAMBA
3	Tipo_B	germinar semillas en caja de cd	LA PAZ
4	Tipo_B	Espectroscopio casero	LA PAZ
5	Tipo_C	Crear mundos virtuales	TARIJA
6	Tipo_C	Base de datos de una tienda	TARIJA
7	Tipo_z	multrimetro casero	en proceso de analisis

2.4. Crear una vista cualquiera que muestre 5 columnas

```
138 CREATE VIEW persona_and_departamento AS
139     SELECT per.id_per, per.apellidos, per.sexo, per.edad, dep.nombre
140     FROM persona AS per
141     INNER JOIN departamento AS dep ON per.id_dep = dep.id_dep;
142 ✓ SELECT pd.*
143 FROM persona_and_departamento AS pd;
144
```

WHERE		ORDER BY				
	id_per	apellidos	sexo	edad	nombre	
1	1318	Flores	f	19	La paz	
2	1419	Jarro	m	19	La paz	
3	1520	Fernandez	f	21	Cochabamba	
4	1621	Limachi	m	20	Cochabamba	
5	1722	Vargas	m	19	Santa Cruz	
6	1823	Gonzales	m	20	Santa Cruz	

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

```
147 ✓ CREATE OR ALTER VIEW PROYECTOS_DISTINTOS AS
148     SELECT pro.nombreproy,pro.id_proy, proyecto =
149         CASE
150             WHEN nombreproy = 'robot' THEN 'Tipo_A'
151             WHEN nombreproy= 'Brazo robotico' THEN 'Tipo_A'
152             WHEN nombreproy= 'germinar semillas en caja de cd' THEN 'Tipo_B'
153             WHEN nombreproy = 'Espectroscopio casero' THEN 'Tipo_B'
154             WHEN nombreproy= 'Crear mundos virtuales' THEN 'Tipo_C'
155             WHEN nombreproy= 'Base de datos de una tienda' THEN 'Tipo_C'
156             ELSE 'Otro proyecto'
157         END
158     FROM proyecto AS pro;
```

WHERE				ORDER BY			
	nombreproy			id_proy		proyecto	
1	Robot			14		Tipo_A	
2	Brazo robotico			15		Tipo_A	
3	germinar semillas en caja de cd			16		Tipo_B	
4	Espectroscopio casero			17		Tipo_B	
5	Crear mundos virtuales			18		Tipo_C	
6	Base de datos de una tienda			19		Tipo_C	
7	mulrimetro casero			20		Otro proyecto	