



Manejo de Vistas y Funciones - SQL Server

Procesual **Hito 4**

Base de Datos I - 2021

ESTUDIANTE: ISRAEL ALEJANDRO ZAMBRANA
SEMESTRE: SEGUNDO SEMESTRE
UNIVERSIDAD: FRANZ TAMAYO

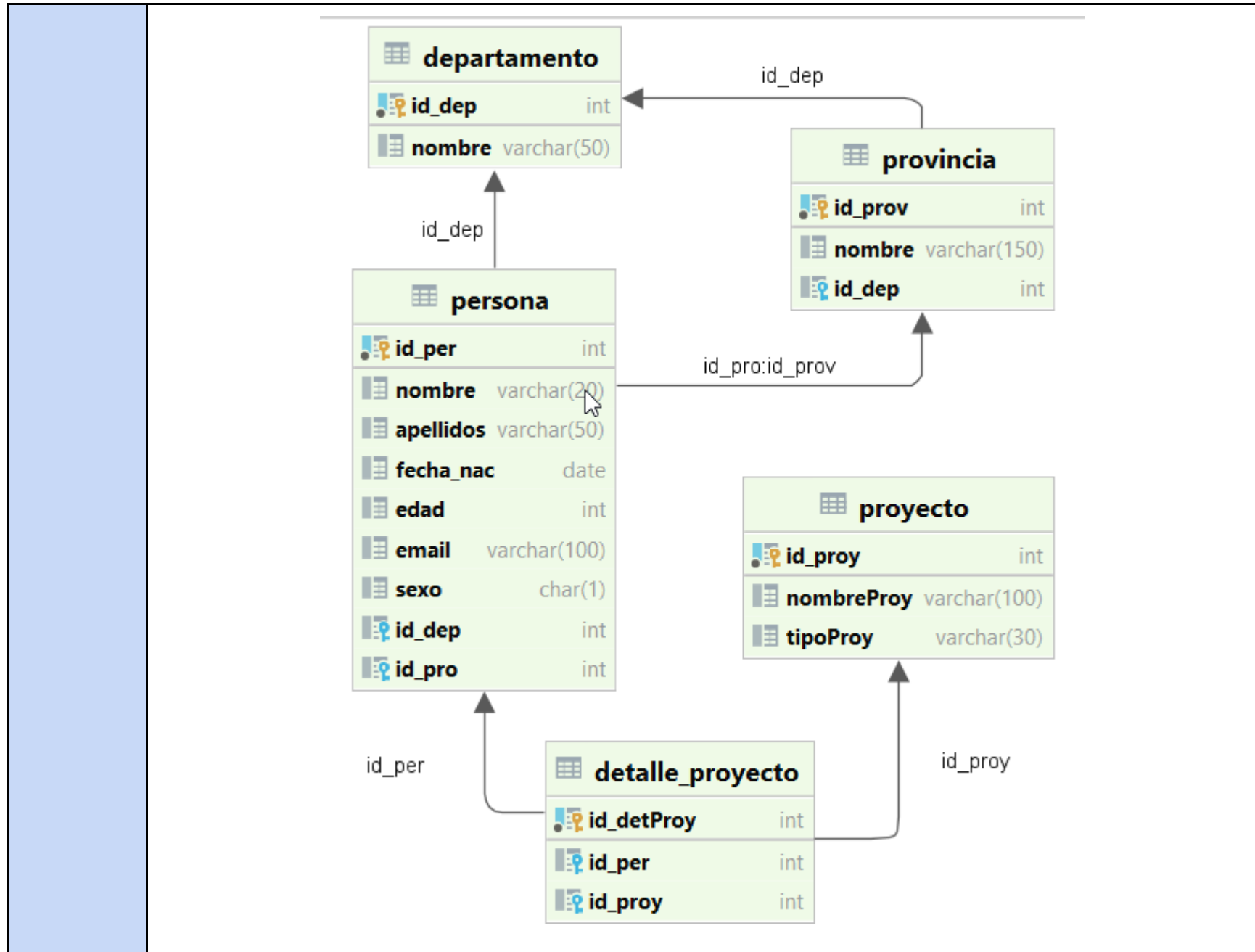
Consigna

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 idéntico 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 dónde 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**

CREAMOS EL DATA BASE ONG_H4 Y LA TABLA DEPARTAMENTO Y PROVINCIA

```
console_1 [SQLSERVER01] x console_1 [UNIFRANZITOS] x examenh4 [UNIFRANZITOS] x ONG_H4 [UNIFRANZITOS] x
Txc Auto v ✓ ↺ ■ 📄

CREATE DATABASE ONG_H4

USE ONG_H4

CREATE TABLE departamento
(
    id_dep INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombre varchar(50),
)

CREATE TABLE provincia
(
    id_prov INTEGER IDENTITY PRIMARY KEY NOT NULL,
    nombre varchar(150),
    id_dep int,
    FOREIGN KEY (id_dep) REFERENCES departamento(id_dep)
)
```

CREAMOS OTRA TABLA PARA PERSONA

```
console_1 [SQLSERVER01] × console_1 [UNIFRANZITOS] × examenh4 [UNIFRANZITOS] × ONG_H4 [UNIFRANZITOS]
Txc: Auto ✓ ↺
CREATE TABLE persona
(
    id_per INTEGER IDENTITY PRIMARY KEY NOT NULL,
    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),
)
```

Y POR ULTIMO CREAMOS OTRAS TABLAS PARA PROYECTO Y DETALLE-PROYECTO

```
5  
6  
7 CREATE TABLE proyecto  
8 (  
9     id_proy INTEGER IDENTITY PRIMARY KEY NOT NULL,  
10    nombreproy varchar(100),  
11    tipoproxy varchar(30),  
12 )  
13  
14 CREATE TABLE detalle_proyecto  
15 (  
16     id_detproxy INTEGER IDENTITY PRIMARY KEY NOT NULL,  
17     id_per      int,  
18     id_proy     int,  
19     FOREIGN KEY (id_per) REFERENCES persona (id_per),  
20     FOREIGN KEY (id_proy) REFERENCES proyecto (id_proy),  
21 )
```

AHORA COLOCAMOS LOS RESPECTIVOS INSERTS PARA DEPARTAMENTO, PROVINCIA Y PROYECTO

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

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

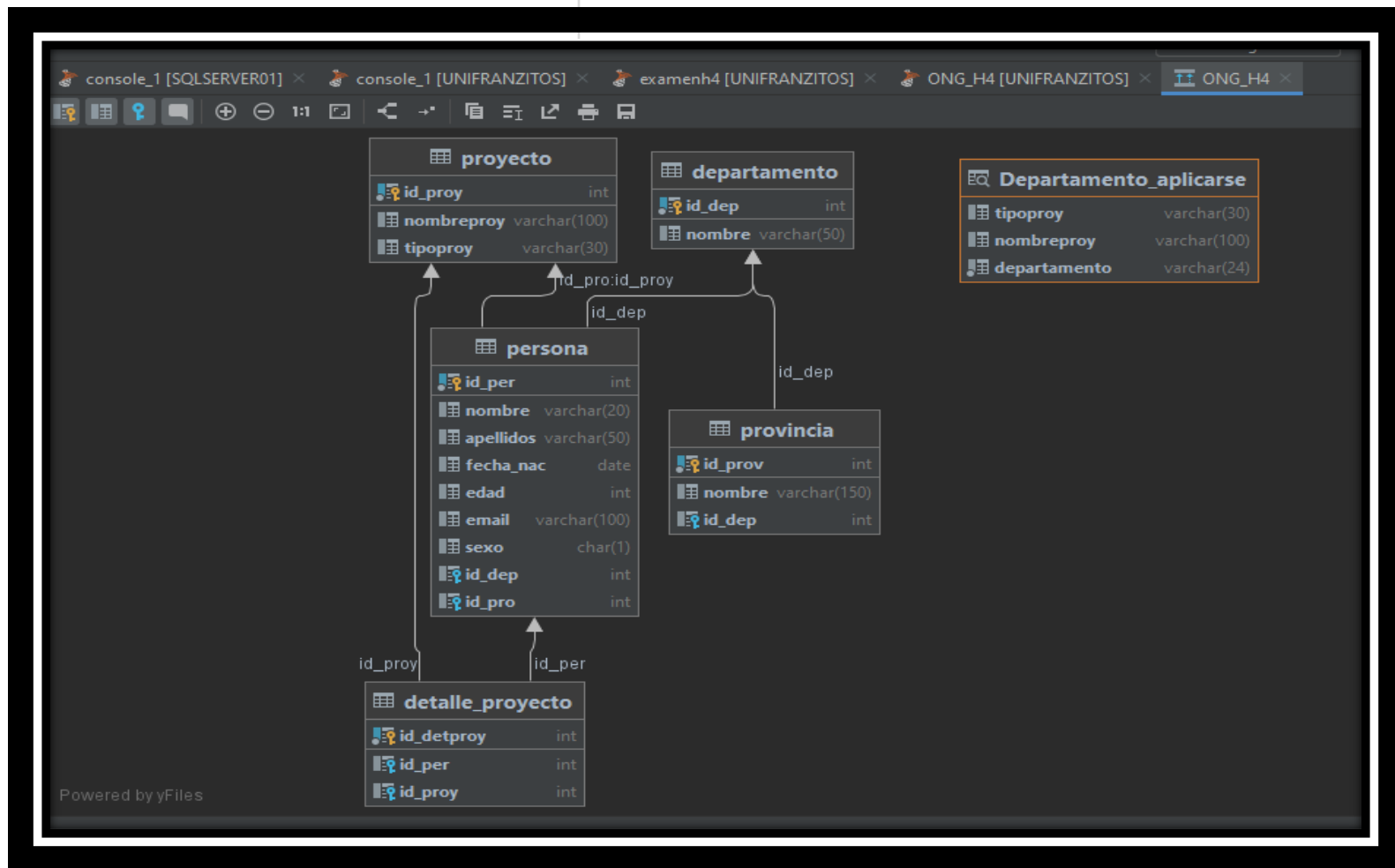
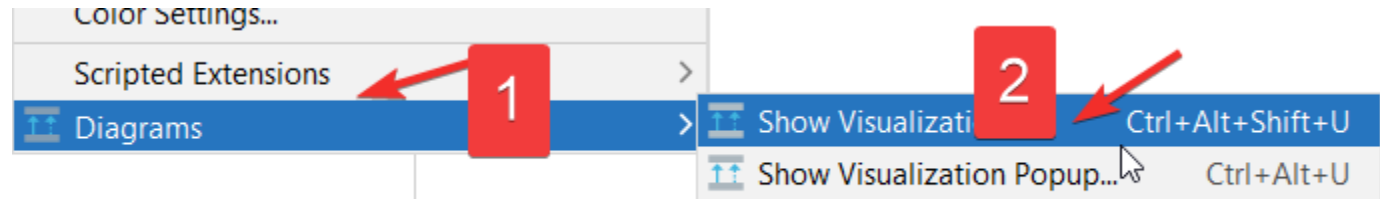
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Sembrado de Arboles', 'TIPO_A');
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Estudio de Semillas', 'TIPO_A');
INSERT INTO proyecto (nombreProy, tipoProy) VALUES ('Alfabetizacion', 'TIPO_B');
INSERT INTO proyecto (nombreProy, tipoProy) VALUES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

Y POR ULTIMO LOS INSERTS DE PERSONA Y DETALLE_PROYECTO

```
71
72 INSERT INTO persona (nombre, apellidos, fecha_nac, edad, email, sexo, id_dep, id_pro) VALUES
73     ('nombre1', 'apellidos1', '1990-10-30', 28, 'nombre1@gmail.com', 'm', 1, 1),
74     ('nombre2', 'apellidos2', '1992-10-30', 28, 'nombre2@gmail.com', 'm', 1, 2),
75     ('nombre3', 'apellidos3', '1994-10-30', 26, 'nombre3@gmail.com', 'm', 1, 3),
76     ('nombre4', 'apellidos4', '1996-10-30', 24, 'nombre4@gmail.com', 'm', 2, 4),
77     ('nombre5', 'apellidos5', '1992-10-30', 28, 'nombre5@gmail.com', 'm', 3, 5),
78     ('nombre6', 'apellidos6', '1999-10-30', 19, 'nombre6@gmail.com', 'f', 3, 5);
79
80 INSERT INTO detalle_proyecto (id_per, id_proy) VALUES
81     (1, 1),
82     (2, 1),
83     (3, 1),
84     (3, 2),
85     (4, 3),
86     (4, 4),
87     (5, 5),
88     (6, 5);
89
```


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**.
- 2.2. Mostrar la persona (**nombres y apellidos**) y el **nombre del proyecto** en donde trabajan.
- 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'
- 2.4. Crear una vista cualquiera que muestre 5 columna

```

90 CREATE OR ALTER VIEW Departamento_aplicarse AS
91     SELECT tipoproy, nombreproy, departamento=
92         CASE
93             WHEN tipoproy = 'TIPO_A' THEN 'COCHABAMBA'
94             WHEN tipoproy = 'TIPO_B' THEN 'LA PAZ'
95             WHEN tipoproy = 'TIPO_C' THEN 'SANTA CRUZ'
96             ELSE 'en proceso de asignacion'
97         END
98     FROM proyecto AS tipoproy;
99
100 ✓ SELECT bn.*
101 FROM Departamento_aplicarse AS bn;

```

Output x ONG_H4.dbo.Departamento_aplicarse x		
tipoproy	nombreproy	departamento
1 TIPO_A	Sembrado de Arboles	COCHABAMBA
2 TIPO_A	Estudio de Semillas	COCHABAMBA
3 TIPO_B	Alfabetizacion	LA PAZ
4 TIPO_B	Creacion de Escuelas	LA PAZ
5 TIPO_C	Apoyo al dibujo	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
- 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
- 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**.
- 3.4. Crear una **función** cualquiera.
 - La función debe de recibir **2 parámetros**.
 - Usar los parámetros en la lógica de la función.

Detalles sobre la entrega de la tarea

1. Crear un documento(archivo PPT) por cada pregunta adjuntar una respuesta(texto explicativo) y una captura de pantalla(Screenshot). Es decir debe crear una presentación power point.
2. El documento generado en el paso 1 convertir en un archivo PDF.
3. Crear un único vídeo explicando todos los pasos realizados para resolver esta actividad.
4. El documento PDF y el video tienen que estar en una carpeta de nombre **procesual** en github (**hito4/procesual**)
5. En la plataforma Moodle solo subir la carátula.
 - a. En la parte inferior agregar **URL de github**: {{mi_url_de_github}}