

MANEJO DE VISTAS Y FUNCIONES- SQL SERVER

LUDWING ANTONI VARGAS IBARRA

DISEÑO DE BASE DE DATOS.



ADJUNTAR EL CÓDIGO SQL QUE GENERA LA BASE DE DATOS, TABLAS Y LOS REGISTROS CORRESPONDIENTES

```
CREATE DATABASE ONG_H4;
USE ONG_H4;

CREATE TABLE departamento(
    id_dep int PRIMARY KEY,
    nombre VARCHAR(50)
);

INSERT INTO departamento (id_dep, nombre) VALUES (1, 'Cochabamba');
INSERT INTO departamento (id_dep, nombre) VALUES (2, 'La Paz');
INSERT INTO departamento (id_dep, nombre) VALUES (3, 'Santa Cruz');
INSERT INTO departamento (id_dep, nombre) VALUES (4, 'Beni');
INSERT INTO departamento (id_dep, nombre) VALUES (5, 'Pando');

SELECT id_dep, nombre
FROM departamento;
```

```
CREATE TABLE provincia(
    id_prov int PRIMARY KEY,
    nombre VARCHAR(150),
    id_dep int,
    FOREIGN KEY(id_dep) REFERENCES departamento(id_dep)
);

INSERT INTO provincia (id_prov, nombre, id_dep) VALUES (1, 'Quillacollo', 1);
INSERT INTO provincia (id_prov, nombre, id_dep) VALUES (2, 'Trinidad', 4);
INSERT INTO provincia (id_prov, nombre, id_dep) VALUES (3, 'Warnes', 3);
INSERT INTO provincia (id_prov, nombre, id_dep) VALUES (4, 'Murillo', 2);
INSERT INTO provincia (id_prov, nombre, id_dep) VALUES (5, 'Manuripi', 5);

SELECT id_prov, nombre, id_dep
FROM provincia;
```

```
CREATE TABLE persona(
    id_per int PRIMARY KEY,
    nombre VARCHAR(20),
    apellidos VARCHAR(50),
    fecha_nac DATE,
    edad int,
    email VARCHAR(100),
    sexo VARCHAR(1),
    id_dep int,
    id_prov int,
    FOREIGN KEY (id_dep) REFERENCES departamento(id_dep),
    FOREIGN KEY (id_prov) REFERENCES provincia(id_prov)
);

INSERT INTO persona (id_per, nombre, apellidos, fecha_nac, edad, email, sexo, id_dep, id_prov) VALUES
(1, 'nombre1', 'apellidos1', '1990-10-30', 28, 'nombre1@gmail.com', 'm', 1, 1),
(2, 'nombre2', 'apellidos2', '1992-10-30', 28, 'nombre2@gmail.com', 'm', 1, 2),
(3, 'nombre3', 'apellidos3', '1994-10-30', 26, 'nombre3@gmail.com', 'm', 1, 3),
(4, 'nombre4', 'apellidos4', '1996-10-30', 24, 'nombre4@gmail.com', 'm', 2, 4),
(5, 'nombre5', 'apellidos5', '1992-10-30', 28, 'nombre5@gmail.com', 'm', 3, 5),
(6, 'nombre6', 'apellidos6', '1999-10-30', 19, 'nombre6@gmail.com', 'f', 3, 5);

Data Source: miNombreHito2
Database: ONG_H4
```

```
FOREIGN KEY (id_prov) REFERENCES provincia(id_prov)
);

INSERT INTO persona (id_per, nombre, apellidos, fecha_nac, edad, email, sexo, id_dep, id_prov) VALUES
(1, 'nombre1', 'apellidos1', '1990-10-30', 28, 'nombre1@gmail.com', 'm', 1, 1),
(2, 'nombre2', 'apellidos2', '1992-10-30', 28, 'nombre2@gmail.com', 'm', 1, 2),
(3, 'nombre3', 'apellidos3', '1994-10-30', 26, 'nombre3@gmail.com', 'm', 1, 3),
(4, 'nombre4', 'apellidos4', '1996-10-30', 24, 'nombre4@gmail.com', 'm', 2, 4),
(5, 'nombre5', 'apellidos5', '1992-10-30', 28, 'nombre5@gmail.com', 'm', 3, 5),
(6, 'nombre6', 'apellidos6', '1999-10-30', 19, 'nombre6@gmail.com', 'f', 3, 5);

SELECT id_per, nombre, apellidos, fecha_nac, edad, email, sexo, id_dep, id_prov
FROM persona;

CREATE TABLE proyecto(
    id_proy int PRIMARY KEY,
    nombreProy VARCHAR(100),
    tipoProy VARCHAR(30)
);
```

```
tipoProy VARCHAR(30)
);

INSERT INTO proyecto (id_proy, nombreProy, tipoProy) VALUES (1, 'Sembrado de Arboles', 'FORESTACION');
INSERT INTO proyecto (id_proy, nombreProy, tipoProy) VALUES (1, 'Estudio de Semillas', 'FORESTACION');
INSERT INTO proyecto (id_proy, nombreProy, tipoProy) VALUES (2, 'Alfabetizacion', 'EDUCACION');
INSERT INTO proyecto (id_proy, nombreProy, tipoProy) VALUES (2, 'Creacion de Escuelas', 'EDUCACION');
INSERT INTO proyecto (id_proy, nombreProy, tipoProy) VALUES (3, 'Apoyo al dibujo', 'ARQUITECTURA');

SELECT nombreProy, tipoProy
FROM proyecto

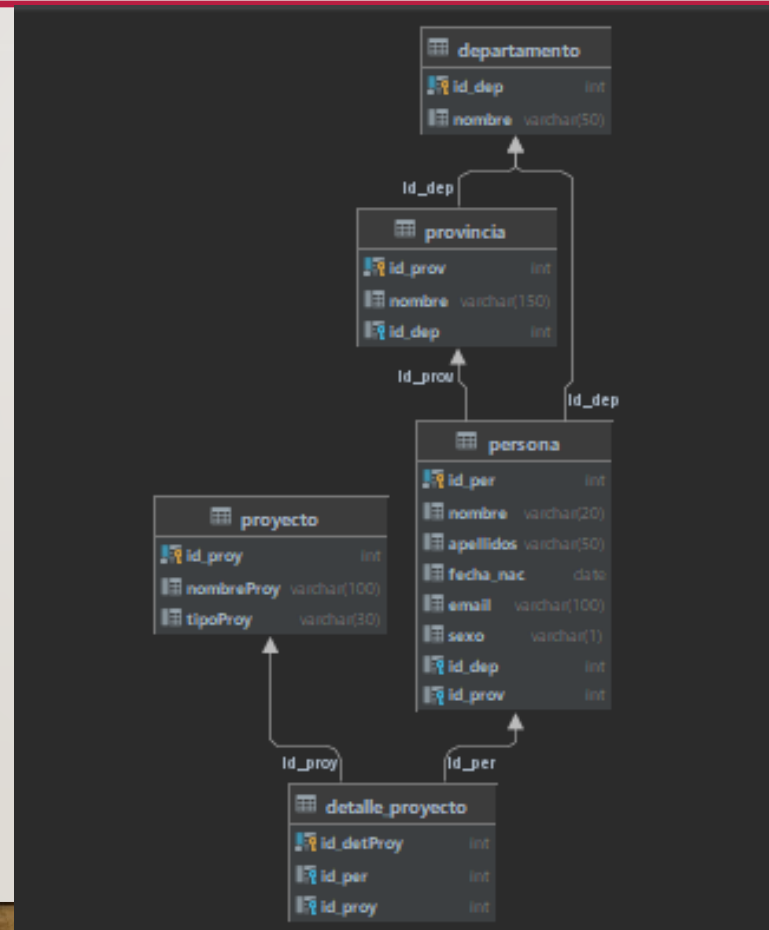
CREATE TABLE detalle_proyecto(
    id_detProy int PRIMARY KEY,
    id_per int,
    id_proy int,
    FOREIGN KEY (id_per) REFERENCES persona (id_per),
    FOREIGN KEY (id_proy) REFERENCES proyecto(id_proy)
);
```

```
id_per int,
id_proy int,
FOREIGN KEY (id_per) REFERENCES persona (id_per),
FOREIGN KEY (id_proy) REFERENCES proyecto(id_proy)
);

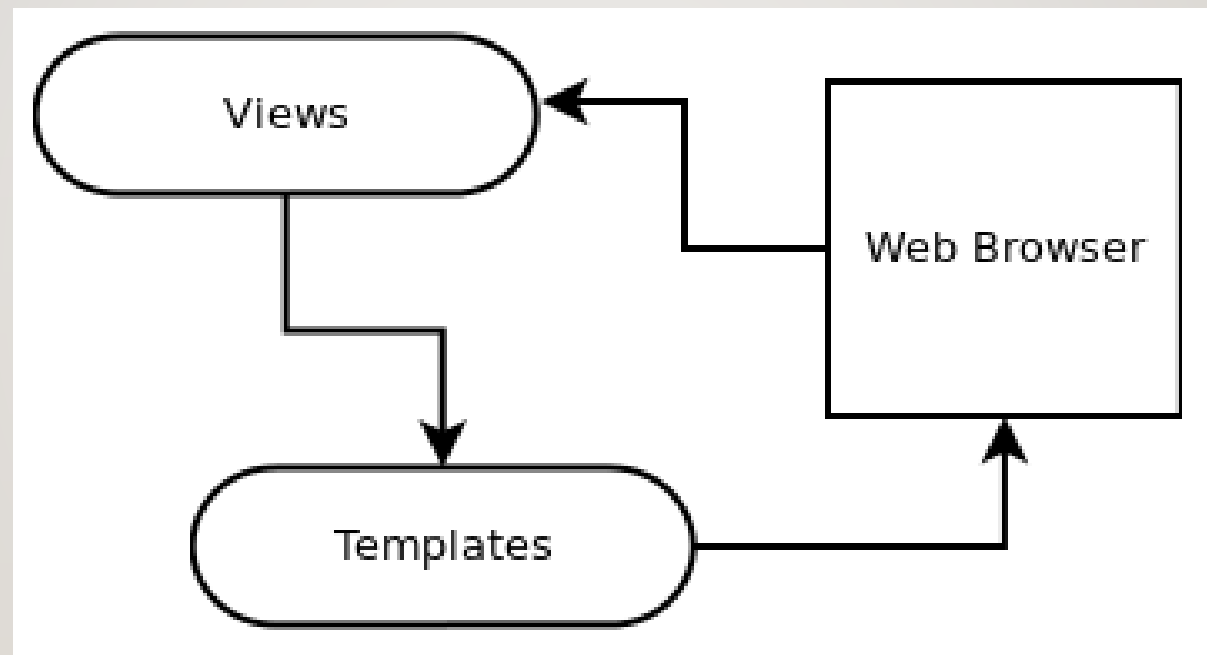
INSERT INTO detalle_proyecto (id_per, id_proy) VALUES
(1, 1),
(2, 1),
(3, 1),
(3, 2),
(4, 3),
(4, 4),
(5, 5),
(6, 5);

SELECT id_per, id_proy
FROM detalle_proyecto;
```

DESPUÉS DE GENERAR EL MODELO LÓGICO DE LA BASE DE DATOS



MANEJO DE VISTAS



MOSTRAR A TODAS LAS PERSONAS QUE VIVEN EN EL DEPARTAMENTO DE COCHABAMBA

```
✓ SELECT dep.nombre, per.id_dep  
FROM departamento AS dep, persona AS per  
WHERE dep.nombre='Cochabamba' and per.id_dep=1;
```

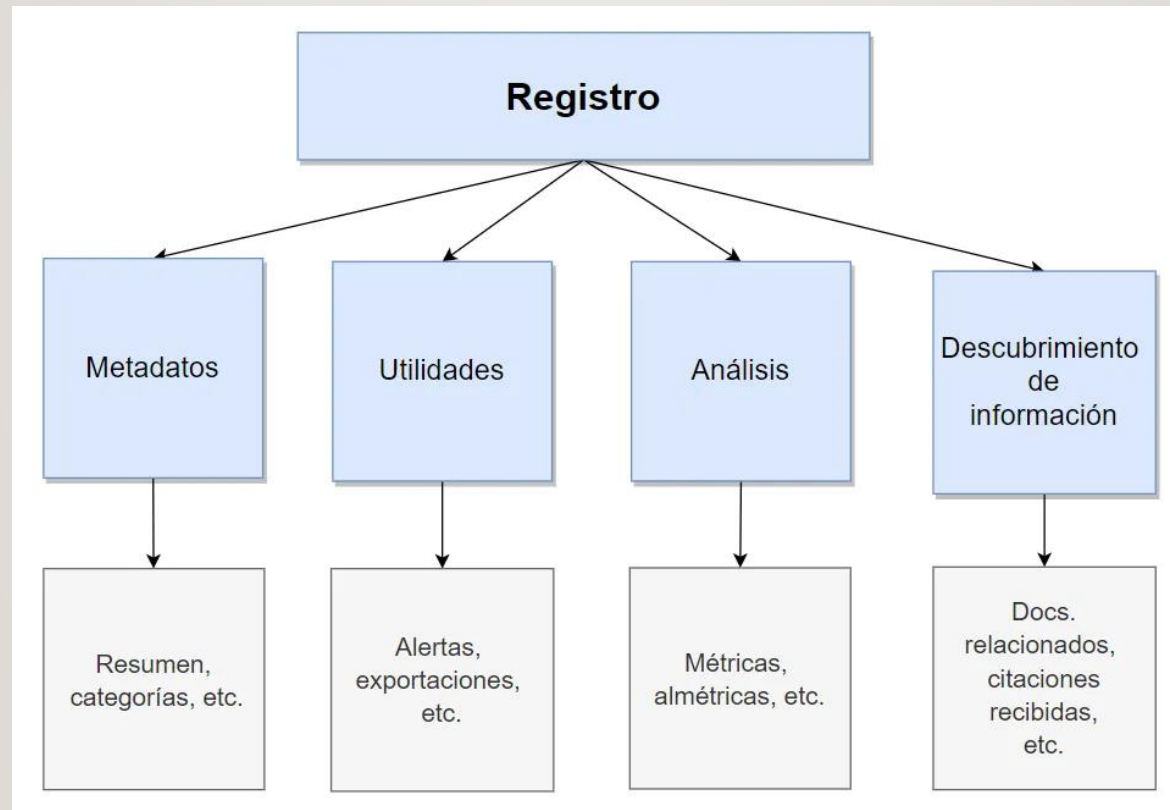
MOSTRAR LA PERSONA (NOMBRES Y APELLIDOS) Y EL NOMBRE DEL PROYECTO EN DONDE TRABAJAN

```
SELECT per.nombre, per.apellidos, proy.nombreProy, dp.id_proy, proy.id_proy, per.id_per, dp.id_per  
FROM proyecto as proy, persona as per, detalle_proyecto as dp  
WHERE proy.nombreProy=' ' and per.nombre=' ' and per.apellidos=' ';
```

ASUMIR QUE TIENE 3 TIPOS DE PROYECTOS (TIPO_A, TIPO_B, TIPO_C)

```
CREATE OR ALTER VIEW departamento_aplicarse AS
SELECT proy.tipoProy, departamento =
    CASE
        WHEN proy.tipoProy = 'FORESTACION' THEN 'COCHABAMBA'
        WHEN proy.tipoProy = 'EDUCACION' THEN 'LA PAZ'
        WHEN proy.tipoProy = 'ARQUITECTURA' THEN 'TARIJA'
        ELSE 'En proceso de asignacion'
    END
FROM proyecto AS proy;
```


MANEJO DE FUNCIONES



CREAR UNA FUNCIÓN QUE PERMITA SABER CUÁNTOS PROYECTOS DISTINTOS DEL TIPO_A, TIPO_B Y TIPO_C EXISTEN

```
SELECT COUNT(proy.tipoProy)
FROM proyecto AS proy
WHERE proy.tipoProy='FORESTACION' and proy.tipoProy='EDUCACION' and proy.tipoProy='ARQUITECTURA'
```

CREAR UNA FUNCIÓN QUE GENERE LOS PRIMEROS N NÚMEROS IMPARES.

```
CREATE OR ALTER FUNCTION numeros_impares(numero int)
RETURNS int AS
BEGIN
    DECLARE numero int=' ';
    DECLARE respuesta int;
    IF numero %2 = 1
    BEGIN
        SET respuesta= numero+1
    end
    ELSE
        RETURN respuesta;
    end;
```

CREAR UNA FUNCIÓN QUE PERMITA INSERTAR UN REGISTRO A LA TABLA PERSONA

```
SELECT nombre, apellidos, sexo, DATENAME(YEAR, fecha_nac)
FROM persona

CREATE OR ALTER FUNCTION registro_personas(nombre VARCHAR(20), apellidos VARCHAR(20), sexo VARCHAR(1), fecha_nac DATE)
RETURNS VARCHAR(100)
AS
BEGIN
    DECLARE registro VARCHAR(100) = '';
    IF LEN(nombre) >= 20
    BEGIN
        SET registro = 'Registro correcto!';
    END
    ELSE
    BEGIN
        SET registro = 'Registro incorrecto!';
    END
END
```

```
SET registro = 'Registro incorrecto!';
END
IF LEN(apellidos) >= 20
BEGIN
    SET registro = 'Registro correcto!';
END
ELSE
BEGIN
    SET registro = 'Registro incorrecto!';
END
IF LEN(sexo) >= 1
BEGIN
    SET registro = 'Registro correcto!';
END
ELSE
BEGIN
    SET registro = 'Registro incorrecto!';
END
```

```
ELSE
BEGIN
    SET registro = 'Registro incorrecto!';
END
IF LEN(sexo) >= 1
BEGIN
    SET registro = 'Registro correcto!';
END
ELSE
BEGIN
    SET registro = 'Registro incorrecto!';
END
RETURN registro;
END;
```


CREAR UNA FUNCIÓN CUALQUIERA.

```
SELECT nombre, DATENAME(DW, fecha_nac)+' ', '+DATENAME(DAY, fecha_nac)+' ', '+DATENAME(MONTH, fecha_nac)
        +', '+DATENAME(YEAR, fecha_nac)
FROM persona

SELECT nombre, apellidos, sexo, DATENAME(YEAR , fecha_nac)
FROM persona
```