Manejo de vistas y funciones -SQL Server

Procesual Hito 4

Nombre: Aaron Álvaro Huanca Salazar

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

```
id_dep int,
id_pro int,
FOREIGN KEY(id_dep)REFERENCES departamento(id_dep),
FOREIGN KEY(id_pro)REFERENCES provincia (id_prov)
 reate table detaile_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 (2,'La Paz');
INSERT INTO departamento (id_dep, nombre) VALUES (3,'Santa Cruz');
INSERT INTO departamento (id_dep, nombre) VALUES (4,'Beni');
 NSERT INTO provincia (id_prov,nombre, id_dep) VALUES (2, Sacaba*, 1);
INSERT INTO provincia (id_prov,nombre, id_dep) VALUES (3, Mizque, 1);
INSERT INTO provincia (id_prov,nombre, id_dep) VALUES (3, Mizque, 1);
INSERT INTO provincia (id_prov,nombre, id_dep) VALUES (5, Robore', 3);
INSERT INTO provincia (id_prov,nombre, id_dep) VALUES (5, Robore', 3);
INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (2, "setulation et artibutes," ProRESTACION), INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (3,"Alfabettzacion", "EDUCACION"); INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (4,"Alfabettzacion", "EDUCACION"); INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (5,"Apoyo al dibujo", "ARQUITECTURA"); INSERT INTO proyecto (id_proy,nombreProy, tipoProy) VALUES (5,"Apoyo al dibujo", "ARQUITECTURA");
```

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

O Utilizar Datagrip para poder generar el diagrama.



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

-- Código en SQL
CREATE VIEW Cochabambinos as
SELECT (nombre) from persona
Where id_dep = 1
WITH CHECK OPTION

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

```
--CODIGO EN SQL

CREATE VIEW Proyectos as

SELECT nombre as nombre_de_la_persona,apellidos as

apellido_de_la_persona,nombreProy as name_proyect from persona p INNER

JOIN proyecto d

ON p.id_pro = d.id_proy;

;
```

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

```
-- CODIGO EN SQL
CREATE VIEW TIPOS AS
SELECT * FROM proyecto
ALTER VIEW TIPOS AS
  SELECT p.id proy AS ID PROYECTO, p.nombreProy AS NOMBRE DEL PROYECTO, p.tipoProy AS TIPO DE PROYECTO, departamento aplicarse =
  CASE
      WHEN p.tipoProy = 'FORESTACION' THEN 'CBB'
      WHEN p.tipoProy = 'EDUCACION' THEN 'LPZ'
      WHEN p.tipoProy = 'ARQUITECTURA' THEN 'SCZ'
    ELSE 'En proceso de analisis'
      END
  FROM proyecto AS p;
```

2.4. Crear una vista cualquiera que muestre 5 columnas.

--CODIGO EN SQL

CREATE OR ALTER VIEW CINCO_COLUMNAS AS

SELECT id_per as numero_de_persona,nombre as nombre_de_la_persona, apellidos as apell_de_la_persona, fecha_nac as nacimiento, edad as Edad FROM persona

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

```
CREATE OR ALTER FUNCTION proyectasos (@dato VARCHAR(20))
RETURNS INT
AS
  BEGIN
  DECLARE @total int
    IF @dato = 'FORESTACION'
     BEGIN
       SET @total = 2;
     END
   ELSE IF @dato = 'EDUCACION'
     BEGIN
       SET @total = 2;
     END
   ELSE IF @dato = 'ARQUITECTURA'
     BEGIN
       SET @total = 1;
     END
   RETURN @total;
  END:
SELECT dbo.proyectasos ('FORESTACION')
```

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

```
CREATE OR ALTER FUNCTION numeros impares (@parametro1 INT)
 RETURNS VARCHAR(100) AS
 BEGIN
  -- utilizando WHILE
   DECLARE @respuesta VARCHAR(100) = ";
   DECLARE @contador INTEGER = 0;
   WHILE @contador < @parametro1
   BEGIN
     IF @contador % 2 <> 0
     BEGIN
       SET @respuesta = CONCAT(@respuesta, @contador, ', ');
     END;
     SET @contador = @contador + 1;
   END;
   RETURN @respuesta;
 END;
```

3.3. Crear una función cualquiera

```
-- CODIGO DE SQL
CREATE FUNCTION EnMayusculas
@Nombre Varchar(50),
@Apellido Varchar(50)
RETURNS Varchar(100)
AS
BEGIN
RETURN (UPPER(@Apellido) + ', ' + UPPER(@Nombre))
END
SELECT dbo. En Mayusculas ('Geynen', 'Montenegro')
```

GRACIAS

Hice lo que pude no se si este bien pero lo importante es aprender poco a poco como usted dice ,no explique mucho los ejercicios por espacio pero en el video estará mi explicación.