



Base de datos II

UNIVERSIDAD PRIVADA FRANZ TAMAYO

DEFENSA HITO 2

Nombre Completo: Diego Emiliano Rivera Tapia

Asignatura: BASE DE DATOS II

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

Docente: Lic. William R. Barra Paredes

Fecha: 16/09/2019

Índice

Diseñar una Base de Datos Relacional para el siguiente escenario.

Una ONG (*organización no gubernamental*) desea implementar un proyecto de educación, para lo cual específicamente requiere saber qué personas nacieron en una fecha determinada en qué ciudad y que provincia.

Sea el siguiente ejemplo.

fecha: 1993/10/10 Departamento: Cochabamba Provincia: Tiquipaya

Personas que nacieron en esa fecha son:

- | | |
|---------------------------|-----------------------------|
| 1. Jaime vargas | Cochabamba Tiquipaya |
| 2. Marlene uriarte | Cochabamba Tiquipaya |
| 3. Eiza Duarte | Cochabamba Tiquipaya |

Dado el ejemplo anterior se puede entender lo siguiente.

que en la fecha **1993/10/10** nacieron 3 personas que son de Cochabamba y de la provincia Tiquipaya.

Nota.- Se sugiere crear mínimamente las siguientes tablas (Persona, Departamento, Provincia).

Manejo de consultas.

1. Crear la consulta SQL que pueda resolver el problema Dado.
2. Crear una consulta SQL manejando DDL.
3. Crear una consulta SQL manejando DML.
4. Crear una consulta SQL manejando 2 tablas (Persona - Departamento). Utilizando JOINS

Manejo de Conceptos.

1. Que es DDL.
2. Que es DML.
3. Que son Bases de Datos Relacionales y No relacionales.
4. Qué es SQL.
5. Cuando debería de usarse una base de datos relacional y no relacional.
6. Que es MariaDB.

Manejo de funciones.

- Crear una función sobre la tabla estudiantes que busca un nombre. (si el nombre existe debe mostrar el nombre y apellidos).
 - La función devuelve un boolean.
 - La función debe buscar un nombre.
 - Resultado.
- Mostrar todos los registros de la tabla estudiantes (nombres y apellidos concatenado) y la edad en otra columna, si la cantidad de estudiantes femeninos sea impar.
 - Utilice las funciones previamente creados.
 - La función parametrizable para saber masculinos o femeninos.
 - La función que concatena nombres y apellidos.
 -
- Mostrar nombre del estudiante y nombre de la materia de los estudiantes inscritos en la gestión 2017.
 - Solo debe generar la consulta SQL (Utilizar JOINS para relacionar las tablas).
- Mostrar el nombre, apellidos y el semestre de todos los estudiantes que estén inscritos, siempre y cuando la suma de las edades del sexo femenino sea par y mayores a 22 años.
 - Debe de crear una función que sume las edades (recibir como parámetro el sexo, y la edad).
 - Nota. Esta función recibe 2 parámetros.
 - La función creada anteriormente debe utilizar en la consulta SQL. (Cláusula WHERE)
- Crear una función sobre la tabla estudiantes que busca un nombre y apellidos. (si existe este nombre y apellido mostrar todos los datos del estudiante).
 - La función devuelve un boolean.
 - La función debe buscar un nombre y sus apellidos.

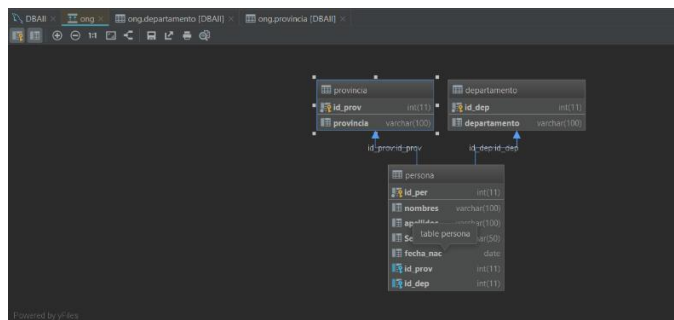
Creamos la base de datos organización no gubernamental para proceder a resolver los ejercicios:

```
create database ONG;  
use ONG;
```

Empezamos a crear las tablas de persona, departamento y provincia:

```
CREATE TABLE provincia  
(  
    id_prov INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
    provincia VARCHAR(100)  
);  
CREATE TABLE departamento  
(  
    id_dep INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
    departamento VARCHAR(100)  
);  
create table persona  
(  
    id_per INTEGER AUTO_INCREMENT PRIMARY KEY NOT NULL,  
    nombres VARCHAR(100),  
    apellidos VARCHAR(100),  
    Sexo varchar(50),  
    fecha_nac DATE,  
    id_dep INT not null,  
    id_prov INT NOT NULL,  
    FOREIGN KEY (id_dep) REFERENCES departamento (id_dep),  
  
    FOREIGN KEY (id_prov) REFERENCES provincia (id_prov)  
);
```

Las tablas de datos estarán relacionadas de tal forma:



Llenamos las tablas con datos de forma manual:

Personas

	id_per	nombres	apellidos	Sexo	fecha_nac	id_dep	id_prov
1	1	Diego	Rivera	Masculino	1993-10-10	1	1
2	2	Adan	Tudela	Masculino	2000-10-10	3	1
3	3	Daniela	Landa	Femenino	2000-09-15	4	4
4	4	Adriana	Contreras	Femenino	1999-11-10	9	2

Departamento

	id_dep	departamento
1	1	Cochabamba
2	2	Beni
3	3	La Paz
4	4	Oruro
5	5	Pando
6	6	Potosi
7	7	Tarija
8	8	Santa Cruz
9	9	Chuquisaca

Provincia

	id_prov	provincia
1	1	Tiquipaya
2	2	Chiquitos
3	3	Sacaba
4	4	Mamore

1.-Manejo de consultas:

1. Crear la consulta SQL que pueda resolver el problema Dado.

```
CREATE FUNCTION datos_personas(nombresPerso varchar(50),apellidosPerso
VARCHAR(50),fechPerso date)RETURNS varchar(100)
BEGIN
DECLARE datosPerso varchar(100);

SET datosPerso =CONCAT(nombresPerso,' ','',apellidosPerso,' ','',fechPerso);
RETURN datosPerso;

end;

SELECT datos_personas(per.nombres,per.apellidos,per.fecha_nac) AS
proyecto_de_educacion
FROM persona AS per;
```

2.-Crear una consulta SQL manejando DDL.

```
CREATE FUNCTION verifecha_nac(fech_persona DATE, fecha_comprueba DATE)
RETURNS BOOLEAN
BEGIN
DECLARE veri BOOLEAN;

IF fech_persona = fecha_comprueba
THEN
SET veri=1;
ELSE
SET veri=0;
END IF;
RETURN veri;
end;
```

3.-Crear una consulta SQL manejando DML.

```
SELECT nombres,apellidos,fecha_nac,dep.departamento
FROM persona
INNER JOIN departamento as dep on persona.id_dep = dep.id_dep
WHERE verifecha_nac(persona.fecha_nac,'2000-09-15')=1;
```

4.-Crear una consulta SQL manejando 2 tablas (Persona - Departamento). Utilizando JOINS

```
SELECT nombres,apellidos,fecha_nac, dep.departamento,prov.provincia
FROM
((persona INNER JOIN departamento as dep on persona.id_dep = dep.id_dep) INNER
JOIN provincia as prov on persona.id_prov = prov.id_prov)
WHERE verifecha_nac(persona.fecha_nac,'1999-11-10')=1;
```

2.- Manejo de Conceptos:

1. Que es DDL.

Data Definition Language trata de esquemas y descripciones de la base de datos, de cómo deben residir los datos en la base de datos.

Usando comandos como CREATE para crear la base de datos o tablas. Otro ejemplo puede ser DROP la cual se utiliza para eliminar tablas o base de datos.

2. Que es DML.

3. Data Manipulation Language es la complementación del DDL la cual se ocupa de la manipulación de datos, con comandos como SELECT, INSERT, UPDATE, DELETE, etc.

4. Que son Bases de Datos Relacionales y No relacionales.

La base de datos de tipo relacional: son las bases de datos las que permiten establecer relaciones entre los datos que están guardados en una tabla y los datos de otra tabla diferentes. (Ejemplo la base de datos de un banco).

La no relacional a diferencia de la relacional intenta solventar esas limitaciones de no poder manipular entornos de almacenamientos masivos de datos.
(Ejemplo Instagram)

5. Qué es SQL.

Es un lenguaje específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

6. Cuando debería de usarse una base de datos relacional y no relacional.

Esa decisión depende a que entorno estamos creando la base de datos, si vamos a manejar cantidad de datos masivos debemos usar la no relacional y en el caso de que tengamos que manipular los datos más específicamente deberíamos usar la relacional.

7. Que es MariaDB.

Es un sistema de gestión de base datos de carácter libre y de código abierto. Aparte es derivado de MySQL con licencia GPL.