

## SQL CREACION DE TABLAS (BASE DE DATOS)

### //CREACIÓN BASE DE DATOS

```
CREATE DATABASE personaliud;
```

### //USAR BASE DE DATOS

```
USE personaliud;
```

### //CREAR TABLA UNIVERSIDAD

```
CREATE TABLE personaliud.universidad(  
id_universidad INT (10) PRIMARY KEY,  
universidad VARCHAR (100) NOT NULL  
);
```

### //CREAR TABLA IDENTIFICACIÓN

```
CREATE TABLE personaliud.tipo_identificacion(  
id_tidentificacion INT (10) PRIMARY KEY,  
tipo_identificacion VARCHAR (100) NOT NULL  
);
```

### //CREAR TABLA NIVEL DE ESTUDIO

```
CREATE TABLE personaliud.nivel_estudio(  
id_destudio INT (10) PRIMARY KEY,  
nivel_estudio VARCHAR (100) NOT NULL  
);
```

### //CREAR TABLA TÍTULO DE ESTUDIO

```
CREATE TABLE personaliud.titulo_estudio(  
id_testudio INT (10) PRIMARY KEY,  
titulo_estudio VARCHAR (100) NOT NULL  
);
```

### //CREAR TABLA ESTADO

```
CREATE TABLE personaliud.estado(  
id_estado INT (10) PRIMARY KEY,  
estado VARCHAR (100) NOT NULL  
);
```

### //CREAR TABLA SEXO

```
CREATE TABLE personaliud.sexo(  
id_sexo INT (10) PRIMARY KEY,  
sexo VARCHAR (100) NOT NULL  
);
```

### ///CREAR TABLA FUNCIONARIO (CON LLAVES FK Y RELACIONES)

```
CREATE TABLE personaliud.funcionario(  
id_funcionario INT (10) PRIMARY KEY,  
id_tidentificacion INT (10) NOT NULL,  
nombre VARCHAR (100) NOT NULL,  
apellido VARCHAR (100) NOT NULL,  
n_identificacion INT (10) NOT NULL,  
id_estado INT (10) NOT NULL,  
id_sexo INT (10) NOT NULL,  
direccion VARCHAR (100) NOT NULL,  
telefono VARCHAR (100) NOT NULL,  
fecha_nacimiento DATE NOT NULL,  
  
CONSTRAINT fk_tidentificacion FOREIGN KEY (id_tidentificacion) REFERENCES  
tipo_identificacion (id_tidentificacion),  
CONSTRAINT fk_estado FOREIGN KEY (id_estado) REFERENCES estado (id_estado),  
CONSTRAINT fk_sexo FOREIGN KEY (id_sexo) REFERENCES sexo (id_sexo)  
);
```

### ///CREAR TABLA GRUPO FAMILIAR (CON LLAVES FK Y RELACIONES)

```
CREATE TABLE personaliud.grupo_familiar(  
id_gfamiliar INT (10) PRIMARY KEY,  
id_funcionario INT (10) NOT NULL,  
nombre VARCHAR (100) NOT NULL,  
apellido VARCHAR (100) NOT NULL,  
rol VARCHAR (100) NOT NULL,  
  
CONSTRAINT fk_funcionario FOREIGN KEY (id_funcionario) REFERENCES funcionario  
(id_funcionario)  
);
```

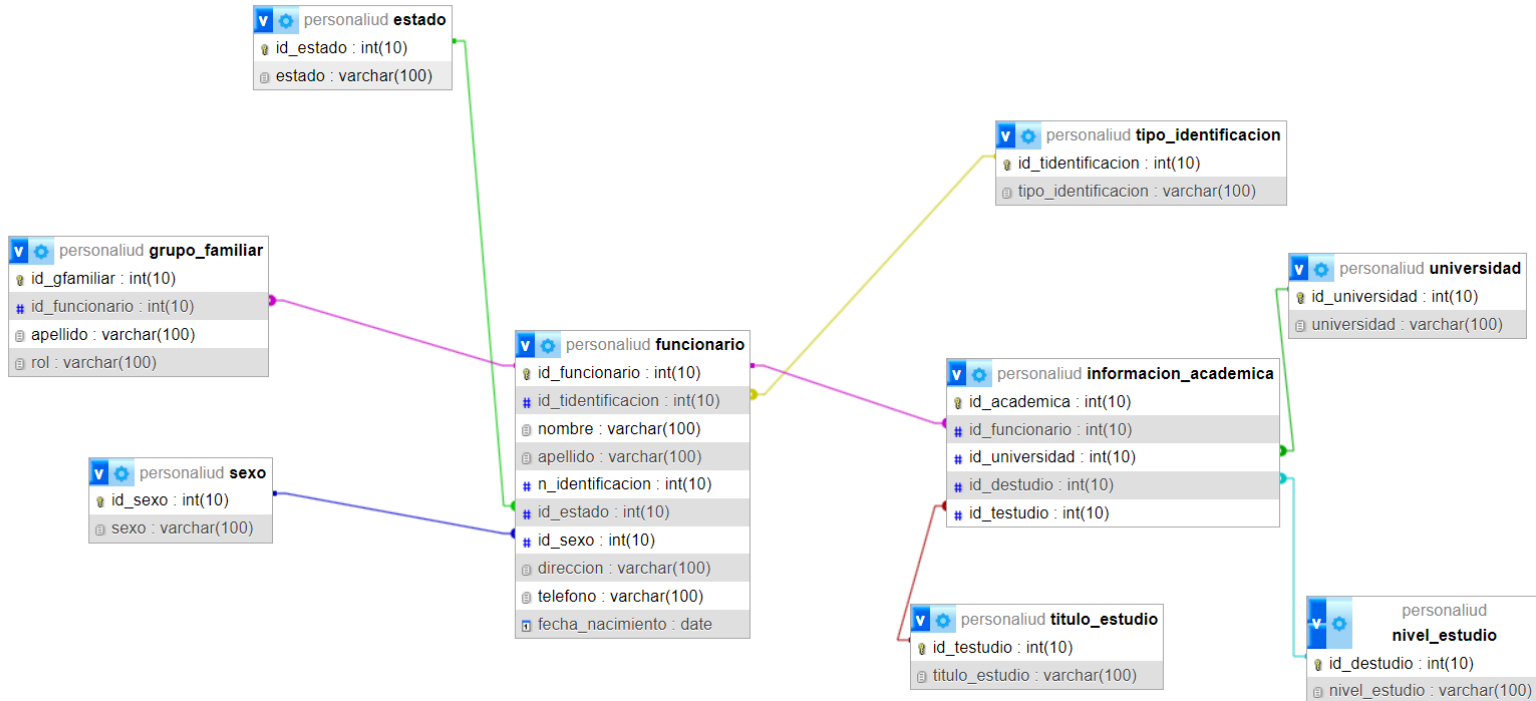
### ///CREAR TABLA INFORMACIÓN ACADÉMICA (CON LLAVES FK Y RELACIONES)

```
CREATE TABLE personaliud.informacion_academica(  
id_academica INT (10) PRIMARY KEY,  
id_funcionario INT (10) NOT NULL,  
id_universidad INT(10) NOT NULL,  
id_destudio INT (10) NOT NULL,  
id_testudio INT (10) NOT NULL,  
  
CONSTRAINT fk_id_funcionario FOREIGN KEY (id_funcionario) REFERENCES funcionario  
(id_funcionario),  
CONSTRAINT fk_universidad FOREIGN KEY (id_universidad) REFERENCES universidad  
(id_universidad),  
CONSTRAINT fk_destudio FOREIGN KEY (id_destudio) REFERENCES nivel_estudio  
(id_destudio),
```

```

CONSTRAINT fk_testudio FOREIGN KEY (id_testudio) REFERENCES titulo_estudio
(id_testudio)
);

```



poblado de las tablas:

//TABLA UNIVERSIDAD

```

INSERT INTO personaliud.universidad (id_universidad, universidad) VALUES (1,
'Universidad Nacional de Colombia');
INSERT INTO personaliud.universidad (id_universidad, universidad) VALUES (2,
'Universidad de los Andes');
INSERT INTO personaliud.universidad (id_universidad, universidad) VALUES (3,
'Universidad de Antioquia');

```

//TABLA DE TIPO DE IDENTIFICACIÓN.

```

insert into personaliud.tipo_identificacion (id_tidentificacion,tipo_identificacion)
values (1,'CC'),
      (2,'TI'),
      (3,'RS');

```

//TABLA DE TÍTULOS DE ESTUDIO.

```

insert into personaliud.titulo_estudio (id_testudio,titulo_estudio)
values (1,' desarrollo de software ');

```

```
(2,' análisis y desarrollo de sistemas'),  
(3,' científico de datos ');
```

#### //NIVEL DE ESTUDIOS.

```
insert into personaliud.nivel_estudio (id_destudio,nivel_estudio)  
values (1,'tecnico'),  
        (2,'tecnologo'),  
        (3,'profesional');
```

#### //ESTADO

```
insert into personaliud.estado (id_estado,estado)  
values (1,'casado'),  
        (2,' soltero');
```

#### //SEXO

```
insert into personaliud.sexo (id_sexo,sexo)  
values (1,'masculino'),  
        (2,'femenino');
```

#### //FUNCIONARIO

```
INSERT INTO personaliud.funcionario (id_funcionario, id_tidentificacion, nombre, apellido,  
n_identificacion, id_estado, id_sexo, direccion, telefono, fecha_nacimiento)  
VALUES (1, 1, 'Juan', 'Perez', 1001394747, 1, 1, 'Calle 10 # 20-30', '1234567890',  
'1990-01-01'),  
(2, 2, 'Maria', 'Gomez', 1001394748, 1, 2, 'Carrera 15 # 25-40', '0987654321', '1991-02-02'),  
(3, 3, 'Pedro', 'Lopez', 1001394749, 1, 1, 'Avenida 20 # 30-50', '0123456789', '1992-03-03');
```

#### //GRUPO FAMILIAR

```
INSERT INTO personaliud.grupo_familiar (id_gfamiliar, id_funcionario, nombre, apellido, rol)  
VALUES (1, 1, 'Ana', 'Perez', 'Esposa'),  
(2, 1, 'Juan', 'Perez', 'Hijo'),  
(3, 2, 'Carlos', 'Gomez', 'Esposo'),  
(4, 2, 'Maria', 'Gomez', 'Hija'),  
(5, 3, 'Luisa', 'Lopez', 'Esposa');
```

#### //INFORMACIÓN ACADÉMICA

```
INSERT INTO personaliud.informacion_academica (id_academica, id_funcionario,  
id_universidad, id_destudio, id_testudio)  
VALUES (1, 1, 1, 1, 1),  
(2,2,2,2,2),  
(3,3,3,3,3);
```

