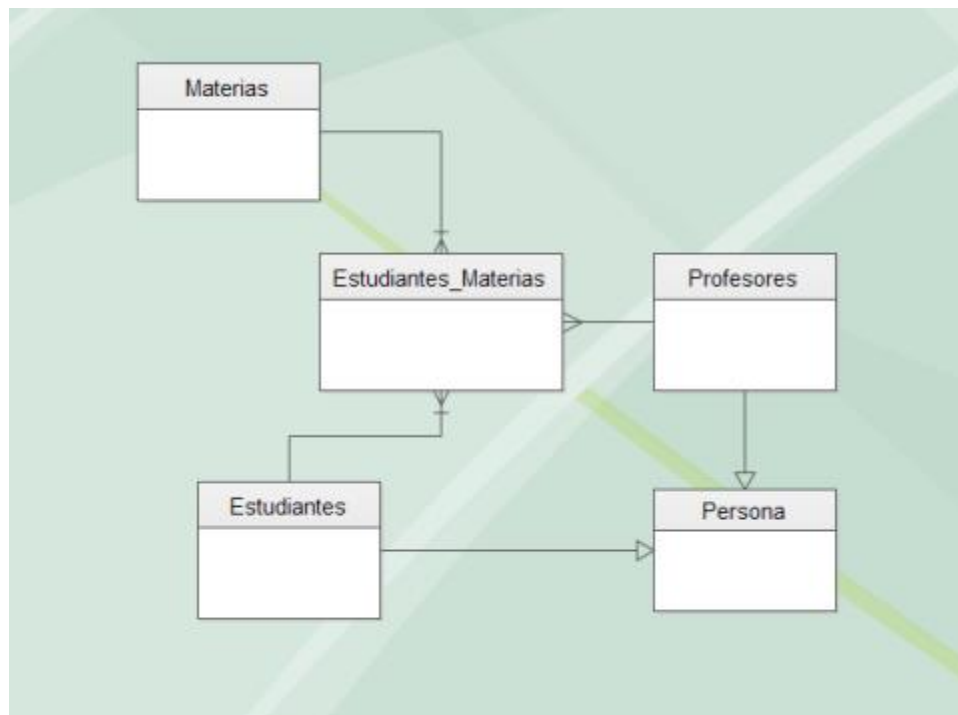


## Ejemplo de la creación de una base de datos usando arrays, herencia, tipos compuestos, otros.

En esta nueva entrada mostraremos la creación de una base de datos utilizando los elementos de Array, tipos compuestos, tipos enumerados particionamiento, herencia e índices.

La base de datos a implementar consiste en un pequeño sistema para un liceo que almacena la información de los estudiantes, profesores y las materias que cursan. La calificación de las materias se realiza de manera cualitativa [Excelente, Muy bien, Bien y mejorable]. Adicionalmente, se registra de que año pertenece la materia. En esta academia se dictan los cursos desde 7mo grado o primer año a 5to año del ciclo diversificado, además se exige que cada estudiante tenga un representante. Primeramente, debemos crear un nuevo usuario en nuestra base de datos. Nuestro usuario se denomina **“admin”** y nuestra base de datos se denomina **“tuto2”**.

### Modelo Relacional



## Características de la base de datos.

### Array

PostgreSQL permite que los campos de una tabla, esta se define con longitud variable. Los arrays pueden ser unidimensionales (vectores) o multidimensionales (Matrices). **En este tutorial usaremos un array unidimensional en la tabla “Materias” para representar con vectores en el campo de turno (Mañana, tarde, noche).** Las matrices de cualquier integrada o definida por el usuario tipo base, tipo de enumeración o tipo de compuesto puede ser creado. Las matrices de dominios no están todavía soportados.

```
psql=# create table materias (  
codigo decimal(4),  
nombre varchar(40),  
turno text[],  
prof_cod decimal(4)  
);  
CREATE TABLE
```

Insert a la tabla de materias.

```
psql=# select cedula, nombre, titulo from profesores;  
 cedula | nombre | titulo  
-----+-----+-----  
 921456 | Juan Perez | Licenciado en matematica y fisica  
10987125 | Douglas Vivas | Licenciado en quimica y biologia  
 6478951 | Martha Calzadilla | Licenciado en artes plasticas  
(3 rows)  
  
psql=# select * from materias;  
codigo | nombre | turno | id | ced_prof  
-----+-----+-----+-----+-----  
(0 rows)  
  
psql=# insert into materias values(1001,'Quimica',{'Manana','Tarde'},1,10987125);  
INSERT 0 1  
psql=#  
psql=# insert into materias values(1002,'Matematica',{'Tarde'},2,921456);  
INSERT 0 1  
psql=#  
psql=# insert into materias values(1003,'Artistica',{'Manana','Tarde'},3,6478951);  
INSERT 0 1  
psql=#
```

## Herencia.

La herencia se refiere a que existen dos posibles tablas que posean la misma información solo que infiere por ciertos datos una a la otra. El ejemplo mas practico es el siguiente: Todas las personas poseen un nombre, apellido, cédula, entre otros datos. Un estudiante forma parte de una persona, pero adicionalmente posee representante y año de ingreso a esta institución. Un profesor también es una persona, pero este posee también fecha de contratación y titulo obtenido. Para estos casos se usa la herencia entre la tabla personas con la tabla profesor o con la tabla estudiantes.

```
plsql=# create table personas (  
cedula decimal(8),  
nombre varchar(40),  
fecha_nacimiento date  
);  
CREATE TABLE
```

```
plsql=# create table profesores (  
titulo varchar(40),  
ano_contrato decimal(4)  
)inherits(personas);  
CREATE TABLE
```

Insert a la tabla de profesores.

```
cedula | nombre | fecha_nacimiento | titulo | ano_contrato  
-----  
(0 rows)  
  
plsql=# insert into profesores values(921456,'Juan Perez','26/10/1987','Licencia  
do en matematica y fisica',2006);  
INSERT 0 1  
plsql=# insert into profesores values(10987125,'Douglas Vivas','26/10/1950','Lic  
enciado en quimica y biologia',2002);  
INSERT 0 1  
plsql=# insert into profesores values(6478951,'Martha Calzadilla','17/08/1974','  
Licenciado en artes plasticas',2002);  
INSERT 0 1
```

## Los tipos Enumerados.

Los tipos enumerados son utilizadas en un campo de una base de datos para generar un listado de opciones que puede contener dicha variable, pero solo se tomará una de las opciones. Se podría comparar con los check constraints, pero este comando es bastante útil cuando el campo posee varias opciones para escoger y el check constraint sería una opción un tanto tediosa para programar.

```
plsql=# create type calificacion as enum('excelente','muy bien','bien','mejorable');
CREATE TYPE
```

```
plsql=# create table Estudiantes_materias (
grado varchar(40),
cedula decimal(8),
cod_materia decimal(4),
calificacion calificacion
);
CREATE TABLE
```

Insert a la tabla de Estudiantes\_materias.

```
plsql=# select * from estudiantes_materias;
grado | cedula | cod_materia | calificacion
-----+-----+-----+-----
(0 rows)

plsql=# insert into estudiantes_materias values('1 ano','29745368',1001,'excelente');
INSERT 0 1
plsql=# insert into estudiantes_materias values('1 ano','26478951',1001,'muy bien');
INSERT 0 1
plsql=# insert into estudiantes_materias values('1 ano','27139712',1002,'mejorable');
INSERT 0 1
plsql=# insert into estudiantes_materias values('1 ano','27139712',1003,'bien');
INSERT 0 1
```

## *Tipos de compuestos*

Una de las columnas posee una estructura con diferentes tipos de datos, las personas que poseen conocimientos medios de programación lo puede asociar con las struct existentes en Lenguaje C o C++. Para una mejor asociación es como un grupo de variables dentro de uno de los atributos de la tabla. En este tutorial se empleará para especificar en la tabla estudiantes el representante con los datos nombre, teléfono correo electrónico y dirección. Esta información fácilmente puede almacenarse en una nueva tabla, pero queremos mostrar la utilización de este tipo.

```
plsql=# create type representante as (  
plsql(# nombre varchar(40),  
plsql(# ocupacion varchar(40),  
plsql(# telefono varchar(15)  
plsql(# );  
CREATE TYPE
```

```
plsql=# create table estudiantes (  
representante representante,  
plsql(# ano_ingreso decimal(4)  
plsql(# )inherits(personas);  
CREATE TABLE
```

Insert a la tabla de estudiantes.

```
plsql=# select * from estudiantes;  
cedula | nombre | fecha_nacimiento | representante | ano_ingreso  
-----+-----+-----+-----+-----  
(0 rows)  
  
plsql=# insert into estudiantes values(26478951,'Pedro Lopez','03/04/2000',('Mariana Carrero','Secretaria','0426-3784561'),2009);  
INSERT 0 1  
plsql=# insert into estudiantes values(29167943,'Camila Morantes','17/06/1990',('Carlos Morantes','Contador','0414-0647812'),2004);  
INSERT 0 1  
plsql=# insert into estudiantes values(27139712,'Paul Quiroz','07/09/1999',('Angel Quiroz','Ingeniero','0414-0641469'),2005);  
INSERT 0 1  
plsql=# insert into estudiantes values(27139712,'Mariana Ochoa','28/11/2002',('Luis Ochoa','Ingeniero','0412-7164923'),2006);  
INSERT 0 1  
plsql=# insert into estudiantes values(27156412,'Luis Villamizar','17/05/2000',('Carmen Salazar','Chef','No suministro'),2004);  
INSERT 0 1  
plsql=# insert into estudiantes values(24975351,'Javier Colmenares','06/07/1998',('Teresa Parra','Politico','0426-1456987'),2004);  
INSERT 0 1  
plsql=# insert into estudiantes values(29745368,'Ana Castro','06/07/1998',('Manuel Castro','Medico','0426-8941029'),2003);  
INSERT 0 1
```

## Inserciones adicionales.

### *Claves primarias, claves foráneas y claves únicas.*

Modificación de las claves primarias de las tablas personas y estudiantes\_materias.

```
psql=# alter table personas add constraint pk_personas primary key (cedula);
ALTER TABLE
psql=# alter table Estudiantes_materias add constraint pk_estmat primary key (c
edula,cod_materia);
ALTER TABLE
```

Se le agrega una columna a la tabla materias [Esta será convertida en la clave primaria y para la foránea usaremos una clave única UK]

```
psql=# alter table materias add column ID decimal(4);
ALTER TABLE
```

```
psql=# select * from materias;
codigo | nombre | turno | prof_cod | id
-----+-----+-----+-----+---
(0 rows)

psql=# alter table materias add constraint pk_materias primary key (id);
ALTER TABLE
```

```
psql=# alter table materias add constraint uk_materias unique (codigo);
ALTER TABLE
```

Generación de las claves foráneas.

```
psql=# alter table materias add constraint fk_materias foreign key (prof_cod) r
eferences personas (cedula);
ALTER TABLE
```

```
psql=# alter table estudiantes_materias add constraint fk_estmat_1 foreign key
(cod_materia) references materias (codigo);
ALTER TABLE
```

```
psql=# alter table estudiantes_materias add constraint fk_estmat foreign key (c
edula) references personas (cedula);
ALTER TABLE
```