

<b>Módulo:</b>	<b>Bases de Datos</b>
<b>UT07:</b>	<b>Base de Datos Objeto-Relacional</b>
<b>Actividad de desarrollo: Base de datos completa</b>	

## 1. Objetivo general

Comprender los elementos característicos de una base de datos objeto-relacional utilizando el sistema gestor: PostgreSQL

## 2. Metodología y material.

PostgreSQL instalado y la herramienta Pgadmin. Consultar apuntes de la asignatura en el campus.

## 3. Descripción:

Usando el software Pgadmin genera scripts que realicen los siguientes puntos:

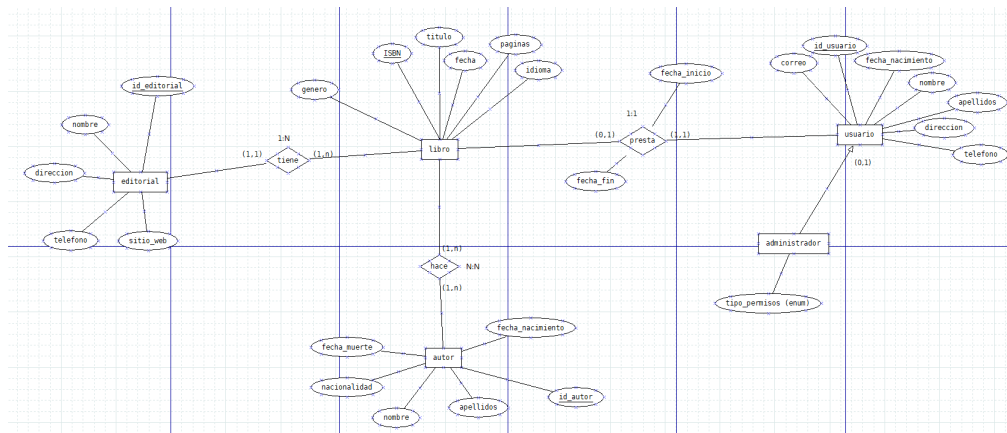
### *1. Plantea un contexto para una aplicación. Redactalo adecuadamente:*

Se Realizará una base de datos de una biblioteca, esta contará con las entidades editorial, libro, usuario, administrador, socio y autor. A cada usuario se le podrá dar en préstamo un solo libro a la vez. Un usuario solo podrá ser administrador o socio y nunca los dos a la vez.

Los libros podrán estar formados por uno o mas autores

Cada libro solo podrá pertenecer a una editorial

### *2. Elabora el esquema E-R correspondiente:*



**Modelo relacional:**

**editoriales(id\_editorial-,nombre,direccion,telefono,sitio\_web)**

**PK id\_editorial**

**libros(isbn-,genero,titulo,fecha,paginas,idioma,id\_editorial\*)**

**PK (isbn) FK id\_editorial references editoriales (id\_editorial)**

**autores(id\_autor-,nombre,apellidos,nacionalidad, fecha\_nacimiento,fecha\_muerte)**

**PK id\_autor**

**creaciones(isbn-\*, id\_autor-\*)**

**PK (isbn, id\_autor) FK isbn references libros (isbn) FK id\_autor refereces autores (id\_autor)**

**usuarios(id\_usuario,nombre,apellidos,direccion,telefono,correo,fecha\_nacimiento)**

**PK id\_usuario**

**administrador(herencia de usuario + tipo\_permisos)**

**prestamos(isbn-\*,id\_usuario-\*,fecha\_inicio,fecha\_fin)**

PK (id\_usuario,isbn,fecha\_inicio) FK isbn references libros (isbn) id\_autor references autores(id\_autor)

### 3. Implementa dicha base de datos (DDL):

**La opción de creación de base de datos fue:**

**CREATE DATABASE biblioteca TEMPLATE template1;**

- a. Establece al menos una opción en la creación de la base de datos.
- b. Debe tener al menos 5 tablas.
- c. La implementación debe contar con al menos una herencia.
- d. Alguna tabla debe contener una secuencia.
- e. Se deben crear dos tipos complejos (Haré dirección y tipo\_permisos)
- f. Alguna tabla debe tener un atributo que sea un vector(es)(teléfonos)

**Script de creación de la base de datos**

```
-- create database biblioteca template template1
```

```
-- creacion de tipos
```

```
create type permisos_type as(  
    ceder_libro boolean,  
    admitir_usuario boolean,  
    eliminar_usuario boolean  
);
```

```
create type direccion_type as(  
    direccion varchar(100),  
    telefono varchar(20)
```

```
calle varchar(50),  
numero integer,  
localidad varchar(50),  
municipio varchar(50)  
);
```

```
--creacion de secuencia
```

```
create sequence secuencia_usuario start with 2 increment by 2 maxvalue 999998  
minvalue 2;
```

```
-- Tabla Editoriales
```

```
CREATE TABLE editoriales (  
id_editorial SERIAL PRIMARY KEY,  
nombre VARCHAR(255) NOT NULL,  
direccion direccion_type,  
telefono VARCHAR(9)[],  
sitio_web VARCHAR(255)  
);
```

```
-- Tabla Libros
```

```
CREATE TABLE libros (  
isbn VARCHAR(50) PRIMARY KEY,  
titulo VARCHAR(255) NOT NULL,  
genero VARCHAR(20) NOT NULL,  
fecha_publicacion DATE NOT NULL,  
idioma VARCHAR(30) NOT NULL,  
numero_paginas INTEGER NOT NULL,
```

```
editorial integer NOT NULL  
);
```

-- Tabla Autores

```
CREATE TABLE autores (  
    id_autor SERIAL PRIMARY KEY,  
    nombre VARCHAR(40) NOT NULL,  
    apellidos VARCHAR(100) NOT NULL,  
    nacionalidad VARCHAR(20) NOT NULL,  
    fecha_nacimiento DATE,  
    fecha_muerte DATE  
);
```

-- Tabla creaciones (liro con autor)

```
CREATE TABLE creaciones(  
    libro_isbn varchar(50),  
    id_autor int,  
    primary key (libro_isbn,id_autor)  
);
```

-- Tabla Usuarios

```
CREATE TABLE usuarios (  
    id_usuario integer PRIMARY KEY default nextval('secuencia_usuario'),  
    nombre VARCHAR(40) NOT NULL,  
    apellidos VARCHAR(100) NOT NULL,  
    direccion direccion_type,  
    telefono VARCHAR(9)[],  
    correo_electronico VARCHAR(255),
```

```
    fecha_nacimiento date
);
```

```
CREATE TABLE administradores(
    tipo_permiso permisos_type
)inherits(usuarios);
```

-- Tabla Préstamos

```
CREATE TABLE prestamos (
    isbn VARCHAR(50),
    id_usuario INTEGER,
    fecha_prestamo DATE NOT NULL,
    fecha_devolucion DATE,
    primary key (isbn,id_usuario,fecha_prestamo)
);
```

-- Foreign keys

-- libros

```
alter table libros add foreign key (editorial) references editoriales(id_editorial);
```

-- creaciones

```
alter table creaciones add foreign key (libro_isbn) references libros(isbn);
```

```
alter table creaciones add foreign key (id_autor) references autores(id_autor);
```

-- prestamos

```
alter table prestamos add foreign key (isbn) references libros(isbn);
```

```
alter table prestamos add foreign key (id_usuario) references
usuarios(id_usuario);
```

```
-- administrador
```

```
alter table administradores add primary key (id_usuario);
```

#### 4. Realiza inserciones (en todas las tablas), actualizaciones y eliminaciones sobre dicha BD. \*\* Crea al menos una transacción

#### INSERCIONES, ACTUALIZACIONES Y DELETES + TRANSACCIÓN

```
select * from editoriales;  
select * from autores;  
select * from libros;  
select * from usuarios;  
select * from creaciones;  
select * from prestamos;  
select * from administradores;
```

```
-- INSERCIONES
```

```
-- editoriales
```

```
insert into editoriales values (default,'jazmin',  
                                row('Calle keke',3,'Telde',  
                                    'Telde'),'123123123,321321321','www.jazmin.com');  
insert into editoriales values (default,'rosa',  
                                row('Calle puentin',2,'Las Remudas',  
                                    'Telde'),'987987987,789789789','www.jazmin.com');  
insert into editoriales values (default,'margarita',  
                                row('Calle sarajevo',1,'Arucas',  
                                    'Arucas'),'123123123,321321321','www.jazmin.com');  
insert into editoriales values (default,'tulipan',  
                                row('Calle rumbiente',5,'Telde',  
                                    'Telde'),'123123123,321321321','www.jazmin.com');
```

```
-- autores
```

```
insert into autores values(default,'Myke','Wasowsky', 'Americano', '1950-01-01', '2024-01-01');  
insert into autores values(default,'Jonhy','Jones', 'Italiano', '1970-05-20', '2024-01-01');  
insert into autores values(default,'Daniel','Felipez', 'Español', '1996-12-15', '2024-02-11');
```

```
insert into autores values(default,'Fofó','Furlen', 'Puertorriqueño', '1990-06-04',
'2022-01-01');
```

-- libros

```
insert into libros values('020202', 'El señorito glotón','terror','2020-02-
05','español',230,1);
insert into libros values('123456', 'Pepe el hambriento','comedia','2023-06-
025','español',130,2);
insert into libros values('987654', 'Teletienda fantastica','accion','2023-12-
14','español',50,3);
insert into libros values('321543', 'Harry Potter punto','fantasia','2000-08-
01','español',500,4);
```

-- usuarios

```
insert into usuarios values(default, 'Lin','Yi',
row('Calle arucas',5,'popo',
'Arucas'), '{123450987,777777777}', 'yilin123451@gmail.com', '2000-08-12');
insert into usuarios values(default, 'Eliecer', 'Del Rosario',
row('Calle jojo',6,'Las Palamas',
'Palmas'), '{555555555,111122223}', 'Eliecerin@gmail.com', '2001-05-30');
insert into usuarios values(default, 'Jaza', 'Daikol',
row('Al lado del insti',1,'Almatriche', 'Las
Palmas'), '{55555444,098789043}', 'DaikolJaza@gmail.com', '2003-02-10');
insert into usuarios values(default, 'Victor', 'Kuzki',
row('Calle Rusia',34,'Skvtin',
'Skvtin'), '{884466220,113355779}', 'VicSpasiva1@gmail.com', '1995-02-26');
insert into usuarios values(default, 'Kevin', 'Cosner',
row('Calle Angeles',10,'Los Angeles',
'California'), '{555999000,111888444}', 'SupermanSteel@gmail.com', '1955-07-
20');
```

-- creaciones

```
insert into creaciones values('020202',1);
insert into creaciones values('123456',2);
insert into creaciones values('987654',3);
insert into creaciones values('321543',4);
```

-- préstamos

```
insert into prestamos values('020202',2, '2024-05-24','2024-05-27');
insert into prestamos values('123456',4, '2024-05-23','2024-05-26');
insert into prestamos values('987654',6, '2024-05-27','2024-05-30');
insert into prestamos values('321543',8, '2024-05-28','2024-05-31');
insert into prestamos values('987654',8, '2024-06-02','2024-06-10');
insert into prestamos values('987654',8, current_date,current_date + 3);
```



```

-- administradores
insert into administradores values(default, 'El Admin','Master',
    row('Calle Edificio',21,'Las Remudas',
'Telde'), '{776655443,987234267}','administratorator@gmail.com','1995-02-
26',row(true,true,true));

-- ACTUALIZACIONES
--editoriales
update editoriales set direccion=ROW((direccion).calle, 30,
(direccion).localidad, (direccion).municipio) where nombre='jazmin';
-- autores
update autores set nombre = 'Randall', apellidos = 'Boggs' where id_autor = 2;
-- libros
update libros set idioma = 'Ingles' where titulo ='Harry Potter punto';
-- usuarios
update usuarios set nombre = 'Administrador', apellidos = 'Definitivo' where
nombre='El Admin';

-- DELETES
-- prestamos
delete from prestamos where id_usuario = 2 and isbn = '020202';
-- usuarios
delete from usuarios where id_usuario = 2;
-- creaciones
delete from creaciones where libro_isbn = '020202';
-- libros
delete from libros where isbn = '020202';

-- TRANSACCION
start transaction;
insert into administradores values(default, 'Vice','Capi',
    row('Calle Comandante Franco',45,'Las Rubieras',
'Telde'), '{543216789,828908070}','SegundoAlMando@gmail.com','1999-07-
11',row(true,true,false));
delete from prestamos where id_usuario = 10;
commit;

```

## 5. Plantea 6 consultas:

a. Al menos dos inner join

b. Al menos dos subconsultas

### Consultas

```
-- 1º muestra los libros de x autor (subconsulta)
select * from libros where isbn = (select libro_isbn from creaciones where id_autor =

    (select id_autor from autores where nombre = 'Myke'));
-- 2º muestra los libros de x editorial (subconsulta)
select * from libros where editorial in (select id_editorial from editoriales where
nombre = 'jazmin');
-- 3º muestra el usuario y libro con prestamo de fecha x (join)
select usuarios.nombre, libros.titulo from usuarios join prestamos on
usuarios.id_usuario = prestamos.id_usuario
    join libros on prestamos.isbn = libros.isbn where prestamos.fecha_prestamo =
'2024-05-23';
-- 4º muestra editorial y autor de x libro (join)
select editoriales.nombre, autores.nombre from editoriales join libros on
editoriales.id_editorial = libros.editorial
    join creaciones on libros.isbn = creaciones.libro_isbn join autores on
autores.id_autor = creaciones.id_autor
    where libros.titulo = 'Teletienda fantastica';
-- 5º muestra las editoriales que tengan libro en prestamo
select editoriales.nombre from editoriales where id_editorial in
    (select id_editorial from libros where isbn in(select isbn from prestamos));
-- 6º muestra usuarios que tengan libro en prestamo
select usuarios.nombre from usuarios where id_usuario in (select id_usuario from
prestamos);
```

## 6. Crea dos funciones y dos trigger.

### 2º TRIGGER, CADA UNO CON SU FUNCIÓN

-- 1º Funcion y 1º trigger  
-- Cuando se elimine un registro de la tabla creaciones (autor asociado a su libro),  
-- se procederá a eliminar todos los prestamos de ese libro y  
-- posteriormente se borrará el libro de la base de datos.  
-- Todo esto atendiendo a que si un libro no ha sido creado, no puede existir.

```
create or replace function delete_libro() returns trigger as $$  
begin  
    delete from prestamos where isbn = old.libro_isbn;  
    delete from libros where isbn = old.libro_isbn;  
return new;  
END $$ language plpgsql;
```

```
create or replace trigger creaciones_AD_trigger after delete on creaciones for each  
row  
    execute procedure delete_libro();
```

```
-- BORRADO DE PRUEBA  
delete from creaciones where libro_isbn ='123456';
```

-- 2º Funcion y 2º trigger  
-- Cuando se añada un usuario nuevo, con motivo de una oferta,  
-- se le hará un préstamo de 3 días del último libro publicado en la tabla libros.

```
create or replace function libro_oferta() returns trigger as $$  
    declare isbnAux varchar(50) default (select isbn from libros order by  
fecha_publicacion desc limit 1);  
    declare dateAux date default (current_date + 3);  
begin  
    insert into prestamos values(isbnAux, new.id_usuario, current_date, dateAux);  
    return new;  
END $$ language plpgsql;
```

```
create or replace trigger usuarios_AI_trigger after insert on usuarios for each row
```

```
execute procedure libro_oferta();
```

```
-- INSERCIÓN DE PRUEBA
```

```
insert into usuarios values(default, 'Cynthia','Alfonso',  
                             row('Calle Aruquitas',10,'Arbusto',  
                                'Arucas'),{666333222,019283650},'Profeloqui@gmail.com','1995-11-23');
```