# BIBLIOTECA

## Introducción:

El proyecto de la biblioteca da acceso a compra y alquiler a clientes sobre diferentes tipos de libretos, ya sean mangas, comics o libros.

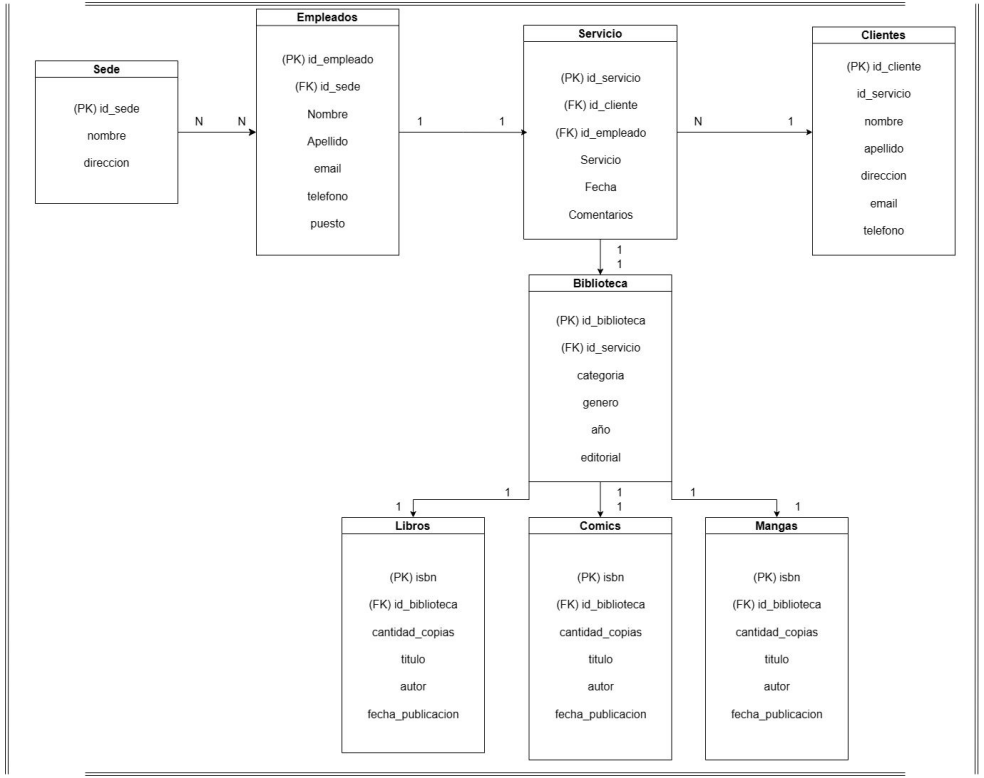
## Modelo de Negocio:

El modelo de negocio gestiona a los clientes, empleados y libretos por sección, organiza, vende y alquila. Ideado para poder administrar cada gusto de cliente, cada sección a la cual pertenece un empleado y cada sede de la biblioteca.

## Objetivo:

El objetivo a cumplir es que los clientes queden registrados en la biblioteca para que el dia de mañana si desean comprar o alquilar lo puedan hacer en base a las recomendaciones por lo que adquirieron previamente. Así mismo poder expandirse a otros sectores.

# Diagrama Entidad - Relación



# Tablas

## Biblioteca:

| | | |
|---|---|---|
| id_biblioteca | PK | varchar(15) |
| id_servicio | FK | varchar(15) |
| categoria | FK | text(20) |
| genero | | text(20) |
| anio | | int |
| editorial | | text(40) |

Clientes:

| | | |
|---|---|---|
| id_cliente | PK | int not null |
| id_servicio | | varchar(15) |
| nombre | | text(20) |
| apellido | | text(30) |
| direccion | | varchar(50) |
| email | | varchar (50) |
| telefono | | int |

Comics:

| | | |
|---|---|---|
| isbn | PK | int not null |
| id_biblioteca | FK | varchar(15) |
| cantidad_copias | | int |
| titulo | | varchar(50) |
| autor | | text(50) |
| fecha_publicacion | | date |

Empleados:

| | | |
|---|---|---|
| id_empleado | PK | int not null |
| id_sede | FK | int not null |
| nombre | | text(20) |
| apellido | | varchar(30) |
| email | | varchar(50) |
| telefono | | int |
| puesto | | text(30) |

Libros:

| | | |
|---|---|---|
| isbn | PK | int not null |
| id_biblioteca | FK | varchar(15) |
| cantidad_copias | | int |
| titulo | | varchar(50) |
| autor | | text(50) |
| fecha_publicacion | | date |

Mangas:

| | | |
|---|---|---|
| isbn | PK | int not null |
| id_biblioteca | FK | varchar(15) |
| cantidad_copias | | int |
| titulo | | varchar(50) |
| autor | | text(50) |
| fecha_publicacion | | date |

Sede:

| | | |
|---|---|---|
| id_sede | PK | int not null |
| nombre | | varchar(30) |
| direccion | | varchar(50) |

Servicio:

| | | |
|---|---|---|
| id_servicio | PK | varchar (15) |
| id_cliente | FK | int not null |
| id_empleado | FK | int not null |
| servicio | | text(20) |
| fecha | | date |
| comentarios | | varchar(60) |

# Funciones

## Función 1 (Averiguar el nombre del cómic mediante el ISBN):

```
delimiter $$
create function Function1 (comicisbn varchar(50))
returns varchar(50)
reads sql data
begin
      declare resultado varchar(50);
    select titulo
    into resultado
    from comics
    where isbn = comicisbn;
return resultado;
end $$
delimiter ;
```

Función 2 (Seleccionar nombre y apellido del cliente mediante el id_cliente):

```
delimiter $$
create function Function2 (clientid varchar(50))
returns varchar(50)
reads sql data
begin
      declare nombre_cliente varchar(30);
    declare apellido_cliente varchar(30);
    declare resultado varchar(50);

    select nombre, apellido
    into nombre_cliente, apellido_cliente
    from clientes
    where id_cliente = clientid;
    set resultado = concat (nombre_cliente, ' ', apellido_cliente);
return resultado;
end $$
delimiter ;
```

# Sentencias

```
select @@autocommit;

set autocommit = 0;
```

## Tabla 1:

```
select * from comics;
start transaction;
delete from comics where isbn = 112347;
-- insert into comics values (112347, 'C3', 6, 'Marvel Zombies 3', 'OvniPress', '1999-05-12');

-- rollback;

-- commit;
```

Tabla 2:

```
select * from clientes;

start transaction;

savepoint parte_0;

insert into clientes values (42123470, 'COM5', 'Philip', 'Morris', 'Campos 1242', 'pmorris@gmail.com', 1134567898);
insert into clientes values (42123471, 'COM6', 'Matthew', 'Perry', 'Campos 1243', 'mperry@gmail.com', 1134567899);
insert into clientes values (42123472, 'COM7', 'Courtney', 'Cox', 'Campos 1244', 'ccox@gmail.com', 1134567900);
insert into clientes values (42123473, 'COM8', 'Samantha', 'Carpenter', 'Campos 1245', 'scarpenter@gmail.com',
1134567901);
savepoint parte_1;

insert into clientes values (42123474, 'ALQ5', 'Michael', 'Myers', 'Campos 1246', 'mmyers@gmail.com', 1134567902);
insert into clientes values (42123475, 'ALQ6', 'Mike', 'Towers', 'Campos 1247', 'mtowers@gmail.com', 1134567903);
insert into clientes values (42123476, 'ALQ7', 'Michael', 'Jackson', 'Campos 1248', 'mjackson@gmail.com', 1134567904);
insert into clientes values (42123477, 'ALQ8', 'Megan', 'Fox', 'Campos 1249', 'mfox@gmail.com', 1134567905);
savepoint parte_2;

-- rollback to parte_1;
```

# Stored Procedures

## SP 1:

Este sp puede ordenar la columna que desees en ascendente o descendente de la tabla que elijas.

```
delimiter $$
CREATE PROCEDURE OrdenarLibros (
    IN NombreTabla VARCHAR(100),
    IN CampoOrden VARCHAR(100),
    IN OrdenTipo VARCHAR(10))
BEGIN
    SET @query = CONCAT('SELECT * FROM ', NombreTabla, ' ORDER BY ', CampoOrden, ' ', OrdenTipo);
    PREPARE stmt FROM @query;
    EXECUTE stmt;
    DEALLOCATE PREPARE stmt;
END $$
delimiter ;
```

## Llamar SP:

```
CALL OrdenarLibros('Biblioteca', 'id_biblioteca', 'ASC'); -- Orden ascendente
```

SP 2:

Procedimiento almacenado para insertar registros en la tabla estetica.tabla_ejemplo

```
DELIMITER $$
CREATE PROCEDURE InsertarComic(
    IN b_isbn INT,
    IN b_id_biblioteca VARCHAR(20),
    IN b_cantidad_copias INT,
    IN b_titulo varchar(50),
    IN b_autor varchar(30),
    IN b_fecha_publicacion date)
BEGIN
    DECLARE mensaje VARCHAR(100);
        INSERT INTO comics (isbn, id_biblioteca, cantidad_copias, titulo, autor, fecha_publicacion)
        VALUES (b_isbn, b_id_biblioteca, b_cantidad_copias, b_titulo, b_autor, b_fecha_publicacion);
    IF ROW_COUNT() > 0 THEN
        SET mensaje = 'Biblioteca cargada correctamente';
    ELSE
        SET mensaje = 'No se pudo insertar la biblioteca. Corroborar los datos ingresados.';
    END IF;
    SELECT mensaje AS Mensaje;
END $$
DELIMITER ;
```

Llamar SP:

```
call InsertarComic('112348', 'C4', '12', 'Secret Wars 1', 'Marvel', '2023-01-01');
```

# Triggers

## Trigger 1:

En la tabla empleados, se modifica el dia, la categoría, el género, el anio y la editorial quedando registro de quien lo hizo, en qué fecha y hora. Estas tablas son before.

| Empleados Insert | Empleados Update | Empleados Delete |
|---|---|---|
| create table empleados_log_i<br>(id_empleado int,<br>id_sede int,<br>nombre text(30),<br>apellido text(30),<br>email varchar(50),<br>telefono int,<br>puesto text(30),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); | create table empleados_log_u<br>(id_empleado int,<br>id_sede int,<br>nombre text(30),<br>apellido text(30),<br>email varchar(50),<br>telefono int,<br>puesto text(30),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); | create table empleados_log_d<br>(id_empleado int,<br>id_sede int,<br>nombre text(30),<br>apellido text(30),<br>email varchar(50),<br>telefono int,<br>puesto text(30),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); |

## Trigger 2:

En la tabla biblioteca, se modifica el id, la categoría, el género, el anio y la editorial quedando registro de quien lo hizo, en qué fecha y hora. Estas tablas son after.

| Biblioteca Insert | Biblioteca Update | Biblioteca Delete |
|---|---|---|
| create table biblioteca_log_i<br>(id_biblioteca varchar(15),<br>categoria text(30),<br>genero text(30),<br>anio int,<br>editorial varchar(50),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); | create table biblioteca_log_u<br>(id_biblioteca varchar(15),<br>categoria text(30),<br>genero text(30),<br>anio int,<br>editorial varchar(50),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); | create table biblioteca_log_d<br>(id_biblioteca varchar(15),<br>categoria text(30),<br>genero text(30),<br>anio int,<br>editorial varchar(50),<br>fechacambio date,<br>horacambio time,<br>usercambio varchar(30)); |

# Users

User 1:

```
use mysql;

create user 'UsuarioStgSelect'@'localhost';
-- El usuario creado tendra permisos solamente para lectura.
alter user 'UsuarioStgSelect'@'localhost' identified by 'stg';
-- Se crea la contraseña del usuario
grant select on biblioteca.* to 'UsuarioStgSelect'@'localhost';
-- Se le asigna permisos al usuario de solo lectura a la bd biblioteca.
```

User 2:

```
use mysql;

create user 'UsuarioStg'@'localhost';
-- El usuario creado tendra permisos solamente para lectura.
alter user 'UsuarioStg'@'localhost' identified by 'stg';
-- Se crea la contraseña del usuario
grant select, insert, update on biblioteca.* to 'UsuarioStg'@'localhost';
-- Se le asigna permisos al usuario de lectura, insercion y update a la bd biblioteca.
```

# Vistas

## Vista 1:

Que servicio eligió cada cliente.

```
create view View1 as select s.id_servicio, s.servicio, c.id_cliente, c.nombre, c.apellido from servicio s join clientes c on s.id_servicio = c.id_servicio;
```

## Vista 2:

En que sede está cada empleado.

```
create view View2 as select s.id_sede, s.nombre as NombreSede, e.id_empleado, e.nombre, e.apellido from sede s join empleados e on s.id_sede = e.id_sede;
```

## Vista 3:

Que libros de suspenso hay.

```
create view View3 as select l.titulo, b.id_biblioteca, b.genero from biblioteca b join libros l on l.id_biblioteca = b.id_biblioteca where b.genero LIKE '%suspenso%';
```

## Vista 4:

Que genero alquilaron los clientes.

```
create view View4 as select s.servicio, b.genero, c.id_cliente, c.nombre, c.apellido from servicio s join biblioteca b on s.id_servicio = b.id_servicio join clientes c on s.id_servicio = c.id_servicio where s.servicio LIKE '%ALQ%';
```

## Vista 5:

Qué clientes compraron comics.

```
create view View5 as select c.id_cliente, c.nombre, c.apellido, b.categoria, s.servicio from clientes c join servicio s on c.id_servicio = s.id_servicio join biblioteca b on s.id_servicio = b.id_servicio where s.servicio LIKE '%COM%';
```

Schema Complete Biblioteca

Import Data Biblioteca

Backup Biblioteca