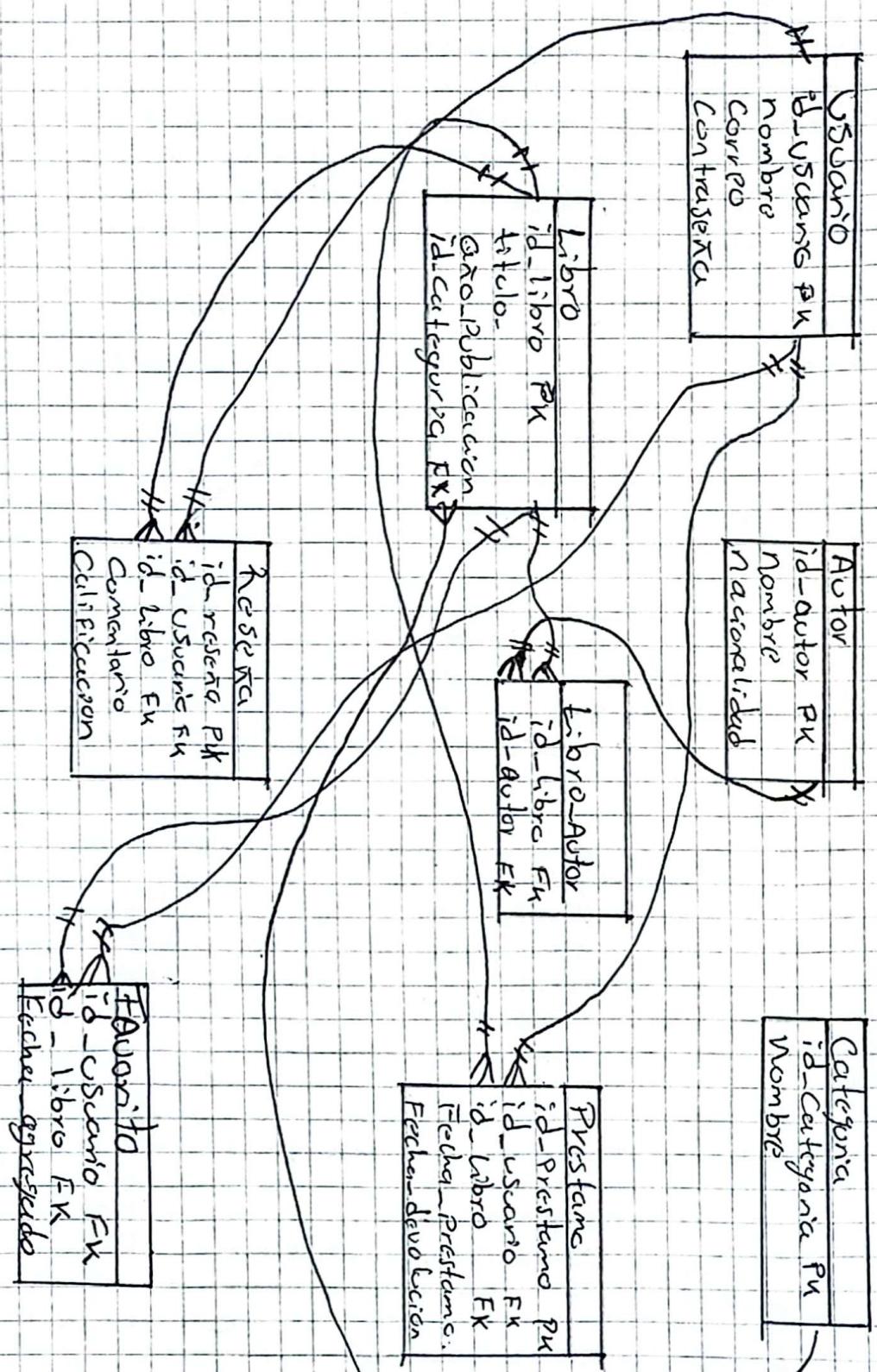


# MER 1



# Diccionario

USUARIO		
Campo	Tipo	Descripción
id_usuario	INT	identificador usuario
nombre	VARCHAR(400)	Nombre usuario
correo	VARCHAR(700)	Correo usuario
Contrasenia	VARCHAR(700)	Contraseña usuario

AUTOR		
Campo	Tipo	Descripción
id_autor	INT	identificador autor
nombre	VARCHAR(400)	nombre autor
Nacionalidad	VARCHAR(50)	nacionalidad autor

CATEGORÍA		
Campo	Tipo	Descripción
id_categoria	INT	identificador categoría
nombre	VARCHAR(50)	nombre del género del libro

LIBRO		
Campo	Tipo	Descripción
id_libro	INT	identificador libro
título	VARCHAR(100)	título libro
año_publicación	INT	año publicación
id_categoria	INT	FK a la categoría

LIBRO-AUTOR		
Campo	Tipo	Descripción
id_libro	INT	FK al libro
id_autor	INT	FK al autor

PRESTAMO		
Campo	Tipo	Descripción
id_Prestamo	INT	identificador préstamo
id_usuario	INT	FK usuario del préstamo
id_libr	INT	FK al libro prestado
Fecha_Prestamo	DATE	Fecha que se presta al lib.
Fecha_Devolucion	DATE	Fecha donde se entrega

## Reseña

Campo	Tipo	Descripción
id_resena	INT	Identificador resena
id_usuario	INT	FK cuando escribe resena
id_libro	INT	FK al libro resenado
Comentario	TEXT	Opinion escrita del usuario
Calificacion	INT	Puntuación numérica

## FAVORITO

Campo	Tipo	Descripción
id_usuario	INT	FK usuario que marca favorito
id_libro	INT	FK al libro
Fecha_creado	DATE	Fecha en que fue marcado

```
CREATE DATABASE Biblioteca;
USE Biblioteca;

CREATE TABLE Usuario (
    id_Usuario INT PRIMARY KEY,
    nombre VARCHAR(700),
    correo VARCHAR(700) UNIQUE,
    Contraseña VARCHAR(700)
);

CREATE TABLE Autor (
    id_Autor INT PRIMARY KEY,
    nombre VARCHAR(100),
    Nacionalidad (50)
);

CREATE TABLE Categoría (
    id_Categoría INT PRIMARY KEY,
    nombre VARCHAR(50)
);

CREATE TABLE Libro (
    id_Libro INT PRIMARY KEY,
    título VARCHAR(700),
    Año_Publicación INT,
    id_Categoría INT,
    FOREIGN KEY (id_Categoría) REFERENCES Categoría (id_Categoría)
);

CREATE TABLE Libro_Autor (
    id_Libro INT,
    id_Autor INT,
    PRIMARY KEY (id_Libro, id_Autor),
    FOREIGN KEY (id_Libro) REFERENCES Libro (id_Libro),
    FOREIGN KEY (id_Autor) REFERENCES Autor (id_Autor)
);

CREATE TABLE Prestamo (
    id_Prestamo INT PRIMARY KEY,
    id_Usuario INT,
    id_Libro INT,
    Fecha_Prestamo DATE,
    Fecha_Devolución DATE,
    FOREIGN KEY (id_Usuario) REFERENCES Usuario (id_Usuario),
    FOREIGN KEY (id_Libro) REFERENCES Libro (id_Libro)
);
```

```

CREATE TABLE Reseña (
    id_reseña INT PRIMARY KEY,
    id_usuario INT,
    id_libro INT,
    Comentario TEXT,
    Calificación INT,
    FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario),
    FOREIGN KEY (id_libro) REFERENCES Libro (id_libro)
);

```

```

CREATE TABLE Favorito (
    id_usuario INT,
    id_libro INT,
    Fecha_agregado DATE,
    PRIMARY KEY (id_usuario, id_libro),
    FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario),
    FOREIGN KEY (id_libro) REFERENCES Libro (id_libro)
);

```

```

INSERT INTO Usuario VALUES (1, 'Carlos Ruiz', 'carlos@gmail.com', 'clave723');
INSERT INTO Usuario VALUES (2, 'Ana Gómez', 'ana@gmail.com', 'clave456');
INSERT INTO Usuario VALUES (3, 'Luis Pérez', 'luis@gmail.com', 'clave789');
INSERT INTO Usuario VALUES (4, 'Mario Díaz', 'mario@gmail.com', 'clave777');
INSERT INTO Usuario VALUES (5, 'Sofía Torres', 'sofia@gmail.com', 'clave321');

```

```

INSERT INTO Autor VALUES (1, 'Gabriel García Márquez', 'Colombia');
INSERT INTO Autor VALUES (2, 'Stephen Hawking', 'Reino Unido');
INSERT INTO Autor VALUES (3, 'Isaac Asimov', 'Rusia');
INSERT INTO Autor VALUES (4, ' Umberto Eco', 'Italia');
INSERT INTO Autor VALUES (5, 'Carl Sagan', 'EEUU');

```

```

INSERT INTO Libro VALUES ('Cien años de Soledad', 1967, 1);
INSERT INTO Libro VALUES ('Breve historia del tiempo', 1988, 1);
INSERT INTO Libro VALUES ('Fundación', 1954, 1);
INSERT INTO Libro VALUES ('El nombre de la rosa', 1980, 1);
INSERT INTO Libro VALUES ('Cosmos', 1980, 1);

```

```

INSERT INTO Libro-Autor VALUES (1, 1);
INSERT INTO Libro-Autor VALUES (2, 2);
INSERT INTO Libro-Autor VALUES (3, 3);
INSERT INTO Libro-Autor VALUES (4, 4);
INSERT INTO Libro-Autor VALUES (5, 5);

```

```

INSERT INTO Prestamo VALUES (1, 1, 1, '2024-04-06', NULL);
INSERT INTO Prestamo VALUES (2, 2, '2024-02-06', '2024-02-14');
INSERT INTO Prestamo VALUES (3, 3, 3, '2024-03-06', NULL);
INSERT INTO Prestamo VALUES (4, 4, 4, '2024-04-06', '2024-04-14');
INSERT INTO Prestamo VALUES (5, 5, 5, '2024-05-06', NULL);

```

```
INSERT INTO Reseña VALUES(1,1,1,'Excelente libro',5);  
INSERT INTO Reseña VALUES(1,2,2,'Muy interesante',4);  
INSERT INTO Reseña VALUES(3,3,3,'Fascinante',5);  
INSERT INTO Reseña VALUES(4,4,4,'Un poco denso',3);  
INSERT INTO Reseña VALUES(5,5,5,'Inspirador',5);
```

```
INSERT INTO Favorito VALUES('2024-06-10');  
INSERT INTO Favorito VALUES('2024-06-11');  
INSERT INTO Favorito VALUES('2024-06-12');  
INSERT INTO Favorito VALUES('2024-06-13');  
INSERT INTO Favorito VALUES('2024-06-14');
```

```
SELECT L.título, C.nombre  
FROM Libro L  
JOIN Categoría C ON L.id_Categoría = C.id_Categoría;
```

```
SELECT U.nombre, C.nombre  
FROM Prestamo P  
JOIN Usuario U ON P.id_usuario = U.id_usuario  
JOIN Libro L ON P.id_libro = L.id_libro;
```

```
SELECT U.nombre, L.título  
FROM Favorito F  
JOIN Usuario U ON F.id_usuario = U.id_usuario  
JOIN Libro L ON F.id_libro;
```

```
SELECT L.título, A.nombre  
FROM Libro-Autor LA  
JOIN Libro L ON LA.id_libro = L.id_libro  
JOIN Autor A ON LA.id_Autor = A.id_Autor;
```

```
SELECT L.título, R.comentario  
FROM Reseña R  
JOIN Libro L ON R.id_libro = L.id_libro  
WHERE R.calificación > 4;
```

```
DELETE FROM Favorito  
WHERE id_usuario = 1 AND id_libro = 3;
```

```
DELETE FROM Reseña  
WHERE id_reseña = 2;
```

```
DELETE FROM Prestamo  
WHERE id_Prestamo = 4;
```

```
DELETE FROM Libro-Autor  
WHERE id_libro = 2 AND id_Autor = 3;
```

```
DELETE FROM Libro  
WHERE id_libro = 5;
```

```
CREATE FUNCTION totalLibrosPorCategorias (cat_id INT) RETURNS  
INT BEGIN  
DECLARE total INT;  
SELECT COUNT(*) INTO total FROM Libro WHERE id_categoria =  
Cat_id;  
RETURN total;  
END;
```

```
CREATE FUNCTION CalificacionPromedio (libro_id INT) RETURNS  
FLOAT BEGIN  
DECLARE Promedio FLOAT;  
SELECT AVG(Calificacion) INTO Promedio FROM Presta WHERE  
id_libro = libro_id;  
RETURN Promedio;  
END;
```

```
CREATE FUNCTION cantidadPrestamoUsuario (uid INT) RETURNS  
INT BEGIN  
DECLARE total INT;  
SELECT COUNT(*) INTO total FROM Prestamo WHERE id_usuario =  
uid;  
RETURN total;  
END;
```

```
CREATE FUNCTION cantidadFavoritosUsuario (uid INT) RETURNS  
INT BEGIN  
DECLARE total INT;  
SELECT COUNT(*) INTO total FROM Favorito WHERE id_usuario =  
uid;  
RETURN total;  
END;
```

```
CREATE FUNCTION existeLibroFavorito (uid INT, lid INT)  
RETURNS BOOLEAN BEGIN  
DECLARE existe INT;  
SELECT COUNT(*) INTO existe FROM Favorito WHERE id_usuario =  
uid AND id_libro = lid;  
RETURN existe > 0;  
END;
```

```
CREATE FUNCTION libroPorAutor (aid INT) RETURNS INT  
BEGIN  
DECLARE INT;  
SELECT COUNT(*) INTO total FROM Libro_Autor WHERE  
id_libro = aid;
```

```
    RETURN total;
END;
```

```
CREATE FUNCTION PrestamoPorLibro (lid INT) RETURNS INT
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total FROM Prestamo WHERE
    id_Libro = lid;
    RETURN total;
END;
```

```
CREATE FUNCTION totalAutores () RETURNS INT
BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total FROM Autor;
    RETURN total;
END;
```

```
CREATE FUNCTION totalUSuario () RETURNS INT BEGIN
    DECLARE total INT;
    SELECT COUNT(*) INTO total FROM USUARIO;
    RETURN total;
END;
```

```
CREATE FUNCTION librosConResenas () RETURNS INT BEGIN
    DECLARE tabla INT;
    SELECT COUNT(DISTINCT id_Libro) INTO tabla FROM Resena;
    RETURN tabla;
END;
```

```
CREATE PROCEDURE registrarPrestamo (
    IN Pid INT,
    IN PidU INT,
    IN PFecha DATE,
);

```

```
BEGIN
    INSERT INTO Prestamo (id_Prestamo, id_Usuario, id_Libro, Fecha_Prestamo, Fecha_Devolucion) VALUES (NULL, Pid_Usuario, Pid_Libro, P_Fecha, NULL);
END;
```

```
SELECT Titulo FROM Libro
WHERE id_Libro IN (
    SELECT id_Libro FROM Libro_Autor
    GROUP BY id_Libro
    HAVING COUNT(id_Autor) >= 2
);
```