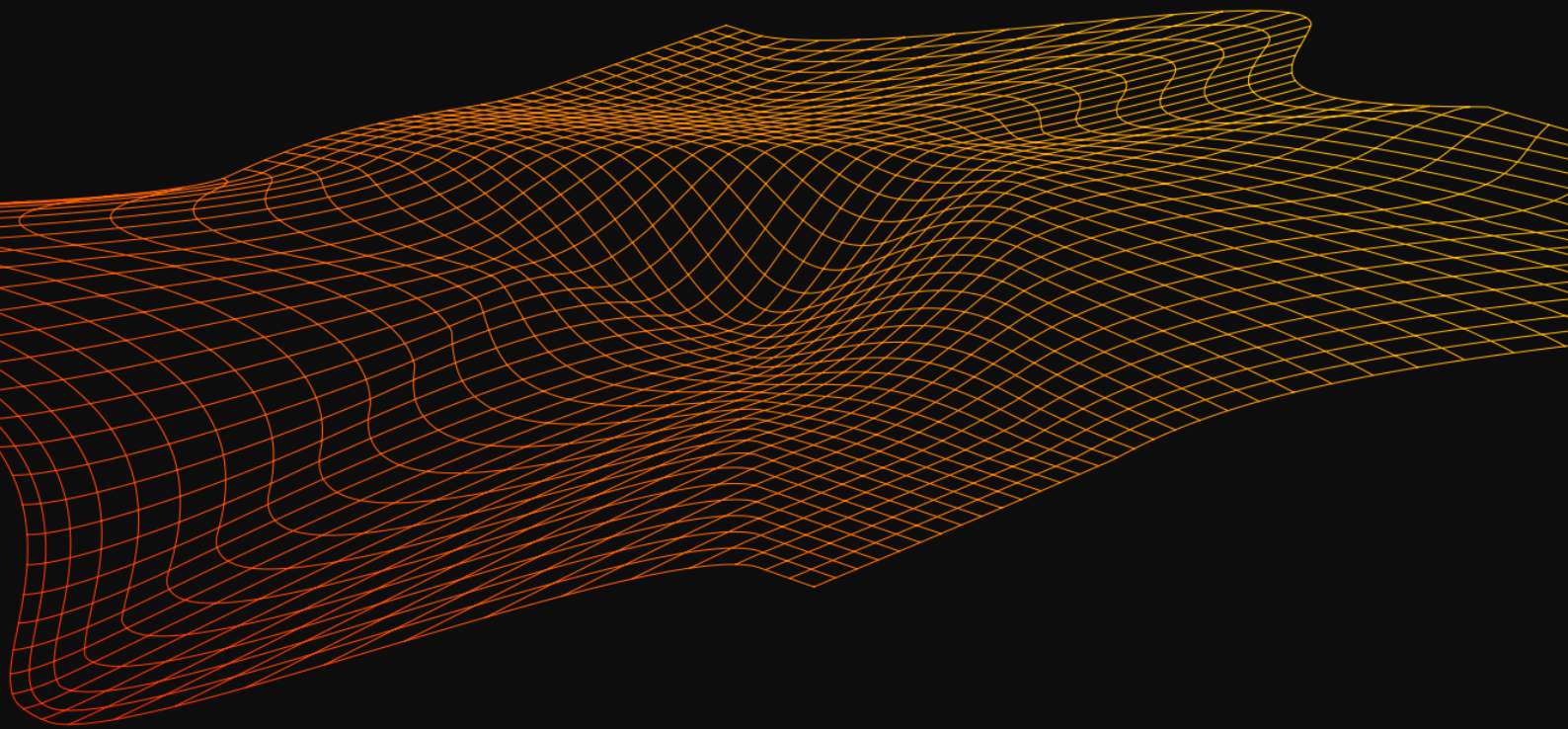

Base de Datos

Biblioteca

Jesús García Cerón

Ingeniería de Software y Sistemas Computacionales



Base de Datos

Alejandro Díaz Ruiz

Universidad La Salle Oaxaca

Santa Cruz Xoxocotlán Oaxaca de Juárez, México

Introducción

El propósito de este proyecto es diseñar y desarrollar una base de datos eficiente que permita gestionar una biblioteca de manera estructurada y funcional. El sistema proporcionará un marco que facilite la administración de autores, editoriales, libros, ejemplares, usuarios y préstamos, optimizando el acceso y control de la información. A través de esta base de datos, se busca resolver las problemáticas comunes en la gestión bibliotecaria, como el seguimiento de préstamos, la organización de los ejemplares disponibles y la relación entre autores y sus obras.

La implementación incluye la creación de tablas y relaciones que reflejan las entidades y sus interacciones dentro de una biblioteca. Se empleará un modelo entidad-relación (E/R) como base para estructurar el diseño lógico, el cual posteriormente será traducido a un modelo relacional para su implementación práctica en SQL. Este modelo garantiza la integridad referencial mediante restricciones y claves primarias y foráneas. Los aspectos considerados incluyen la resolución de problemas específicos, el procedimiento de desarrollo, los fragmentos de código de creación, las capturas de pantalla para validar las tablas creadas, y la generación de modelos relacionales mediante la aplicación correspondiente.

En resumen, el sistema propuesto no solo será un repositorio de datos, sino también una herramienta efectiva para la gestión de las actividades fundamentales de una biblioteca, ofreciendo escalabilidad y robustez en su diseño.

INDICE

| | |
|--|-----------|
| Propuesta de resolución de problema | 1 |
| Procedimiento de desarrollo | 2 |
| Inserción de 500,000 registros | 4 |
| Tablas | 6 |
| Modelo E/R | 9 |
| Modelo Relacional | 9 |
| Modelo Relacional generado | 10 |
| Fragmentos de código de creación | 11 |
| Conclusiones | 14 |

Propuesta de resolución de problema

1. Identificación del Problema: Hoy en día las bibliotecas enfrentan dificultades comunes para organizar la información de manera eficiente, como el seguimiento de libros, autores, editoriales, ejemplares y préstamos a los usuarios. Esto puede generar retrasos y confusión tanto para el personal como para los usuarios. Este proyecto busca crear un sistema de base de datos que facilite estas tareas y permita una administración ordenada y eficaz.

2. Etapas del Desarrollo:

- **Análisis del Problema:** Comprender las necesidades de una biblioteca, identificando los elementos clave que se deben gestionar, como autores, libros, usuarios y préstamos.
- **Diseño del Modelo Conceptual (E/R):** Representar las entidades (como libros, usuarios, autores) y sus relaciones mediante un diagrama Entidad-Relación, que permita visualizar cómo interactúan entre sí.
- **Diseño Relacional:** Convertir el modelo conceptual en un modelo relacional detallado, donde cada entidad y relación se traduzca en tablas, incluyendo claves primarias y foráneas para mantener la integridad de los datos.
- **Creación de Tablas en SQL:** Escribir código SQL para crear cada tabla con las restricciones necesarias, como CHECK, FOREIGN KEY, y UNIQUE, garantizando la calidad y consistencia de los datos.
- **Validación:** Comprobar que las tablas y relaciones funcionan correctamente, ejecutando consultas simples como `SELECT * FROM nombre_tabla` para verificar su contenido.
- **Documentación Visual:** Utilizando la herramienta MySQL Workbench para generar diagramas visuales del modelo relacional y validar que refleje el diseño propuesto.
- **Pruebas Funcionales:** Probar el sistema simulando escenarios reales, como registrar nuevos libros, gestionar préstamos y realizar búsquedas de información.

3. Resultados Esperados:

- Un sistema funcional de base de datos que almacene y gestione toda la información clave de una biblioteca.
- Diagramas claros que representen las entidades, relaciones y restricciones del modelo.
- Fragmentos de código SQL documentados, fáciles de entender y reutilizar.
- Un sistema flexible y escalable que pueda adaptarse a las necesidades futuras de la biblioteca.

4. Entregables Clave:

- **Modelo Entidad-Relación (E/R):** Mapa visual de las entidades y cómo están relacionadas.
- **Modelo Relacional:** Representación lógica implementada en SQL.
- **Código SQL:** Instrucciones paso a paso para crear y verificar las tablas.
- **Capturas de Pantalla:** Resultados de consultas SQL ejecutadas para cada tabla creada.

5. Beneficio del Proyecto: Este proyecto no solo proporcionará una solución técnica para organizar la biblioteca, sino que también servirá como ejemplo práctico para comprender el uso de bases de datos en el mundo real. Al utilizar un diseño lógico y estructurado, cualquier persona podrá entender y ampliar el sistema en caso de necesitarlo en el futuro.

Procedimiento de Desarrollo

Identificación de Requisitos

Analizar las necesidades de la biblioteca: ¿Qué datos necesitan ser gestionados? Por ejemplo, libros, autores, usuarios, préstamos, etc.

Definir las entidades clave y sus relaciones.

Plasmar los objetivos generales del sistema, como la organización y el acceso rápido a la información.

Diseño del Modelo Conceptual (Entidad-Relación - E/R)

Crear un diagrama a partir de la aplicación Draw.io

Asegurarse de incluir las entidades principales (Autor, Editorial, Libro, Usuario, Ejemplar, Préstamo) y sus relaciones. Por ejemplo:

Un autor puede escribir muchos libros (relación "muchos a muchos").

Un usuario puede pedir prestados varios ejemplares (relación "uno a muchos").

Diseño del Modelo Relacional

Convertir el diagrama E/R en tablas relacionales.

Asignar claves primarias para identificar de manera única a cada registro.

Establecer claves foráneas para crear relaciones entre tablas.

Definir restricciones necesarias (como integridad referencial, restricciones CHECK, etc.).

Escritura del Código SQL

- Escribir el código para la creación de las tablas usando MySQL Workbench
- Incluir restricciones como:
- PRIMARY KEY para identificar registros únicos.
- FOREIGN KEY para establecer relaciones entre tablas.
- Restricciones como CHECK para validar información (por ejemplo, que la fecha de préstamo sea anterior a la fecha de devolución).
- He de asegurar que el script siga la estructura lógica

1. Pruebas y Validación

- Probar diferentes escenarios, como registrar nuevos libros, agregar autores, realizar préstamos y verificar las devoluciones, próximamente implementar el medio millón de datos.
- Asegurarse de que las relaciones y restricciones funcionan según lo planeado. Por ejemplo, que no sea posible eliminar un libro que tiene ejemplares asociados sin actualizar los datos relacionados.

2. Documentación

- Capturar imágenes de las tablas creadas y de los datos insertados usando sentencias como `SELECT * FROM nombre_tabla`.
- Generar un modelo relacional gráfico usando la herramienta MySQL Workbench.
- Guardar y organizar los fragmentos de código SQL utilizados, para referencia futura o mejoras.

3. Entrega y Retroalimentación

- Presentar el proyecto con diagramas, capturas de pantalla, y una explicación clara de cómo funciona el sistema.

Inserción de 500,000 registros

1. Generación de 500,000 datos aleatorios con Python en formato .csv
2. Con los datos generados empezar la inserción de datos por tabla en MySQL Workbench
3. *Script de Inserción – ejemplo*

-- Cargar autores

LOAD DATA INFILE

'C:/Users/Jesus/Downloads/datos_biblioteca/autores.csv' **Ruta**

INTO TABLE autor **Tabla de Insercion**

FIELDS TERMINATED BY ',' **Engloba el formato y separación por atributo**

ENCLOSED BY '"' **Engloba el formato y separación por atributo**

LINES TERMINATED BY '\n' **Engloba el formato y separación por atributo**

IGNORE 1 LINES **Ignora la primera linea que nos dice el formato de los atributos**

(codigo_autor, nombre, apellidos);

Hacer lo mismo con las otras tablas

Notas adicionales de posibles errores;

Para aquellas tablas que están relacionadas entre tablas foráneas, es necesario verificar que los datos , en este caso *ids* sean los mismos en ambos documentos, para evitar errores de integridad e inconsistencia, en este caso errores como lo es el 1452 o 1062 de duplicado

Error por query rows

Establecer como límite 500,000 rows, en la sección de Edit< Preferences< SQL Execution , Esto será necesario ya que si no lo realizamos es posible que se presente errores como lo es el 2013 que nos indica que el tiempo de ejecución ha pasado. Esto significa que existen mas datos de los que el servidor de query puede soportar

Error por errores de sintaxis en los archivos de formato .csv generado por el script de Python

Es lo más común ya que no se están respetando los atributos manejados, faltan las comillas, o inclusive la cantidad supera los 500,000, para solucionar esto podemos verificar nuestros limites y cambiarlos o incluso dentro del archivo csv filtrar la cantidad y modificarla a nuestro favor. Otra alternativa es tener en cuenta la cantidad, esto podemos verificarlo inclusive en *powershell* a través del siguiente comando.

```
Cd 'C: Ruta...'
(Get-Content "C:\ruta\al\archivo.csv").Count
```

Esto nos contara las líneas y podemos verificar si sobrepasa los 500,000 datos o lo contrario.

4. Utilizar SELECT*FROM para verificar y comprobar de que los archivos se hayan generado en sus respectivas tablas

25 21:28:06 Select*from libro LIMIT 0, 500000

500000 row(s) returned

Adicional

En caso de sobrepasarse o la mala implementación de los datos podemos utilizar la linea TRUNCATE TABLE para eliminar los datos de la tabla

En caso del error 1701 [Forzar a la eliminación]

Descripción: Cannot truncate a table referenced in a foreign key constraint

Causa: Quisiste vaciar (TRUNCATE) la tabla usuario que está referenciada por la tabla préstamo.

Solución: Borrar con DELETE FROM usuario o deshabilitar temporalmente los checks de FK (SET FOREIGN_KEY_CHECKS = 0;).

Ejemplo de inserción **Manual** de datos

--Insertar editoriales

INSERT INTO editorial (nombre) VALUES

('Editorial Anagrama'),

('Alfaguara'),

('Planeta'),

('Penguin Random House'),

('Tusquets Editores');

-- Insertar autores

INSERT INTO autor (nombre, apellidos) VALUES

('Gabriel', 'García Márquez'),

('Isabel', 'Allende'),

('Jorge Luis', 'Borges'),

('Julio', 'Cortázar'),

('Mario', 'Vargas Llosa');

Tablas

21 21:24:18 Select*from autor LIMIT 0, 500000

500000 row(s) returned

Tabla Autor – SELECT*FROM autor

| | codigo_autor | nombre | apellidos |
|---|--------------|------------|-----------|
| ▶ | 2 | Mauro | Garibay |
| | 3 | José María | Mora |
| | 4 | Abraham | López |
| | 5 | Felix | Esquivel |
| | 6 | Yuridia | Mojica |
| | 7 | Víctor | Cornejo |
| | 8 | Bruno | Romo |
| | 9 | Emilio | Marín |
| | 10 | Ofelia | Mateo |
| | 11 | Sofía | del Valle |
| | 12 | Samuel | Apodaca |
| | 13 | Flov | Alcántar |

Tabla Editorial – SELECT*FROM editorial

✓ 23 21:26:49 Select*from editorial LIMIT 0, 500000

500000 row(s) returned

| | codigo_editorial | nombre |
|---|------------------|-----------------------------------|
| ▶ | 1 | nombre |
| | 2 | Casas-Cotto S.A. de C.V. |
| | 3 | Duarte S.C. |
| | 4 | Urías-Acuña S.A. de C.V. |
| | 5 | Mascareñas-Ornelas e Hijos |
| | 6 | Arellano-Garibay |
| | 7 | Torres S. R.L. de C.V. |
| | 8 | Laboratorios Zayas, Rivas y Amaya |
| | 9 | Berrios S.A. de C.V. |
| | 10 | Vargas-Tijerina e Hijos |
| | 11 | Casas-Cotto S.A. de C.V. |
| | 12 | Duarte S.C. |

Tabla Libro – SELECT*FROM libro

25 21:28:06 Select*from libro LIMIT 0, 500000

500000 row(s) returned

| | codigo_libro | titulo | isbn | codigo_editorial | paginas |
|---|--------------|-------------------------|-------------------|------------------|---------|
| ▶ | 000GHES2EZ | La sombra de silencio | 978-1-111-35897-6 | 337700 | 803 |
| | 000GVTHLBQ | Ecos de verdad | 978-1-4318-4864-5 | 75899 | 378 |
| | 000J26WVVV | Crónicas de memoria | 978-0-7942-2439-4 | 90631 | 781 |
| | 000QFJCO05 | La historia de silencio | 978-0-326-24802-7 | 328323 | 241 |
| | 000R9C3REW | Crónicas de silencio | 978-0-13-009738-5 | 131625 | 749 |
| | 0010G5ASQG | El misterio de deseo | 978-0-609-18337-3 | 495650 | 209 |
| | 0016ODL9QR | Ecos de oscuridad | 978-1-948116-17-6 | 75015 | 378 |
| | 0017ZHMJ1Q | El misterio de silencio | 978-1-64997-582-9 | 1916 | 291 |
| | 001ELOQPM0 | La sombra de memoria | 978-0-11-032134-9 | 184671 | 856 |
| | 001HBPNYY3 | A través del tiempo | 978-1-320-77835-0 | 275057 | 107 |
| | 001KYMOV3W | El misterio de memoria | 978-0-555-89550-4 | 195499 | 890 |
| | 001R71GCM0 | El legado de tiempo | 978-1-80795-810-7 | 270609 | 123 |

Tabla Ejemplar – SELECT*FROM ejemplar

18 21:20:07 Select*from ejemplar LIMIT 0, 500000

500000 row(s) returned

| | id_ejemplar | localizacion | codigo_libro |
|---|-------------|--------------|--------------|
| ▶ | 2 | Sala-26-A-4 | K5NY4KQBAX |
| | 3 | Sala-5-C-13 | 6AVXLEDA3M |
| | 4 | Sala-14-B-6 | SO4A2I2QT5 |
| | 5 | Sala-18-A-18 | 9OCYAJZSUM |
| | 6 | Sala-6-A-17 | KF7TIJ7GRG |
| | 7 | Sala-3-C-15 | LCTV99VWGX |
| | 8 | Sala-29-B-18 | 8P49ZPE1QD |
| | 9 | Sala-15-B-16 | QYOCZ8AZO8 |
| | 10 | Sala-13-B-5 | 1Z4CRZMNL3 |
| | 11 | Sala-27-A-12 | XC1E5HADDW |
| | 12 | Sala-19-B-1 | X4OQ3BJVGP |
| | 13 | Sala-14-B-15 | C6T687IIM2G |

ejemplar 51 x

Tabla Autor-Libro – SELECT*FROM autor_libro

4 11:18:20 Select*from autor_libro LIMIT 0, 500005 500005 row(s) returned

| | codigo_autor | codigo_libro |
|---|--------------|----------------|
| ▶ | 2 | TYENPWX9K9 |
| | 4 | HKR8LC57U2 |
| | 4 | SVNVQIPS3P |
| | 6 | LRR3ZW1YVJ |
| | 6 | LRR3ZW1YVJ... |
| | 6 | PT9B8RAT2C |
| | 6 | PT9B8RAT2C... |
| | 7 | ZZZ3J5YFP8 |
| | 10 | N84EDJ9C0E |
| | 10 | NEVZ1P669T |
| | 10 | NEVZ1P669T"... |
| | 11 | FTI ICTRI 07W |

autor libro 2

Tabla Usuario – SELECT*FROM usuario

5 11:19:41 Select*from usuario LIMIT 0, 500005 500000 row(s) returned

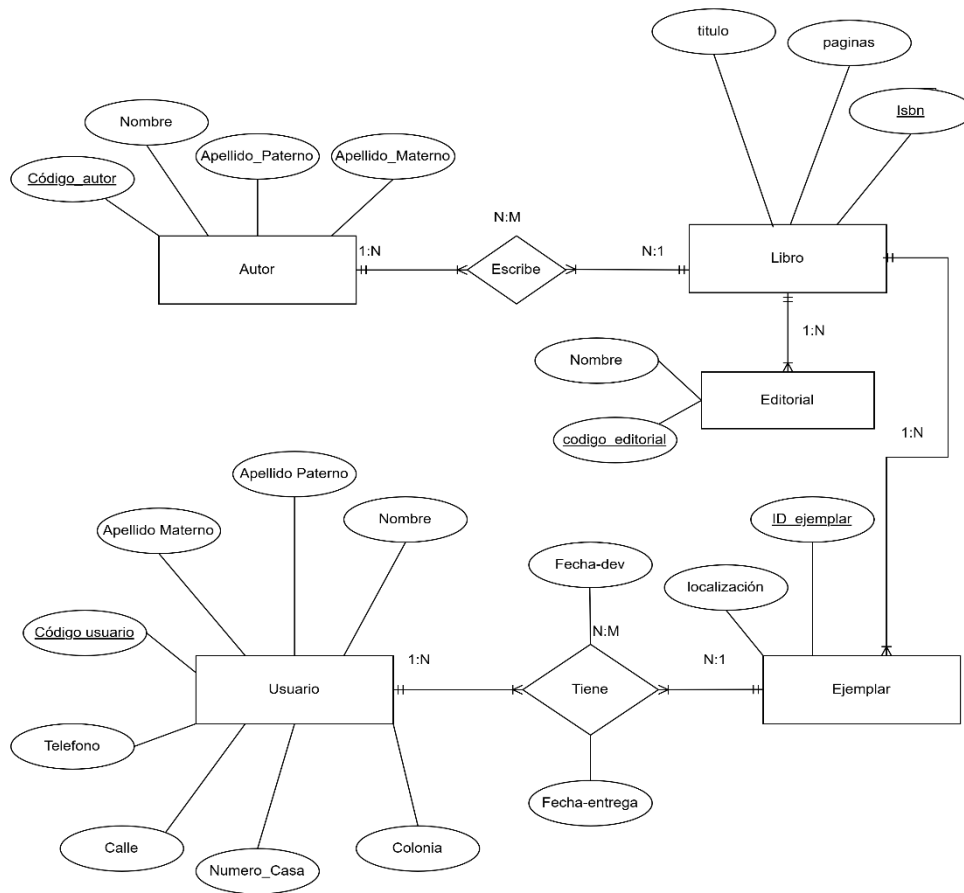
| | codigo_usuario | nombre | apellidos | telefono | direccion |
|---|----------------|---------|------------------|----------------------|--|
| ▶ | 1 | Claudia | Cabrera Tórrez | +02(3)4268609166 | Periférico Pacheco 282 Edif. 650 , Depto. 623, ... |
| | 2 | Rubén | Abrego Zaragoza | (142)513-1410 | Circunvalación Kirguistán 492 Interior 950, San ... |
| | 3 | Alvaro | Soria Gómez | 450-281-0795 | Peatonal Sur Gracia 352 Edif. 432 , Depto. 289,... |
| | 4 | Alberto | Chávez Mares | (774)922-4190 | Periférico Norte Nava 582 003, Nueva Serbia, G... |
| | 5 | Adela | Adame Armenta | 040.868.0932x3366 | Retorno Maldivas 848 Edif. 850 , Depto. 540, S... |
| | 6 | Homero | Guillen Carmona | 1-535-209-9996 | Calle Jalisco 265 Interior 018, San Esparta de la... |
| | 7 | Javier | Muñiz Huerta | 1-894-577-1523x52932 | Ampliación Turquía 041 961, Vieja Namibia, Q. R... |
| | 8 | Adela | Navarrete Solís | 1-808-291-0393 | Privada Colima 397 Interior 846, San Cynthia lo... |
| | 9 | Miguel | Esquibel Paz | 1-415-029-2715x821 | Viaducto Colima 887 839, San Soledad de la Mo... |
| | 10 | José | León Rael | 1-654-472-1426x978 | Corredor Sur Quintanilla 151 Edif. 292 , Depto. ... |
| | 11 | Antonia | Paz Melgar | 068.167.6442x873 | Viaducto Emiratos Árabes Unidos 577 442, San ... |
| | 12 | Indira | Guevara Tijerina | 558 175 1051 | Callejón Crespo 487 Interior 025, Vieja República |

Tabla Préstamo- SELECT*FROM préstamo

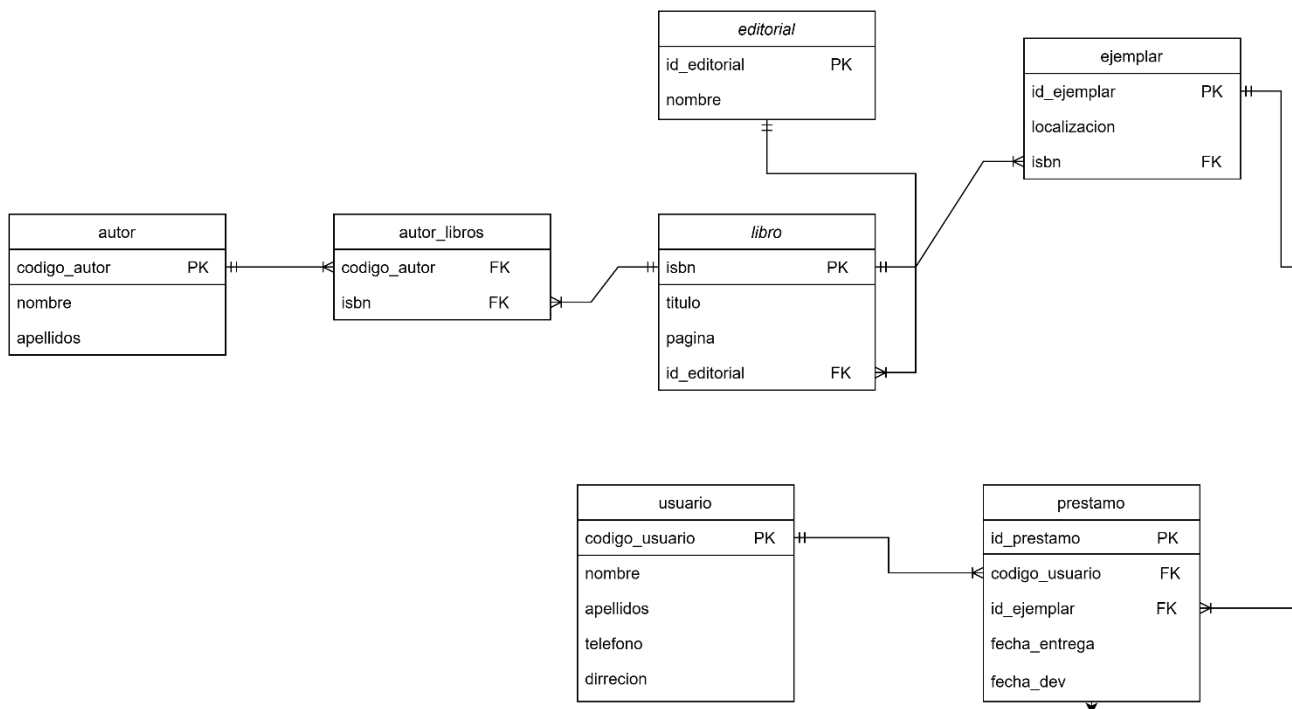
6 11:20:37 Select*from prestamo LIMIT 0, 500005 500000 row(s) returned

| | id_prestamo | codigo_usuario | id_ejemplar | fecha_prestamo | fecha_dev |
|--|-------------|----------------|-------------|----------------|------------|
| | 41 | 41 | 304584 | 0000-00-00 | 2023-01-04 |
| | 42 | 42 | 49658 | 0000-00-00 | 2023-07-02 |
| | 43 | 43 | 346415 | 0000-00-00 | 2021-09-09 |
| | 44 | 44 | 15377 | 0000-00-00 | 2021-03-10 |
| | 45 | 45 | 7461 | 0000-00-00 | 2022-02-17 |
| | 46 | 46 | 237457 | 0000-00-00 | 2022-11-23 |
| | 47 | 47 | 338944 | 0000-00-00 | 2024-09-07 |
| | 48 | 48 | 99140 | 0000-00-00 | 2024-02-11 |
| | 49 | 49 | 353458 | 0000-00-00 | 2020-10-08 |
| | 50 | 50 | 273666 | 0000-00-00 | 2024-03-01 |
| | 51 | 51 | 21112 | 0000-00-00 | 2024-05-03 |
| | 52 | 52 | 258527 | 0000-00-00 | 2022-01-14 |

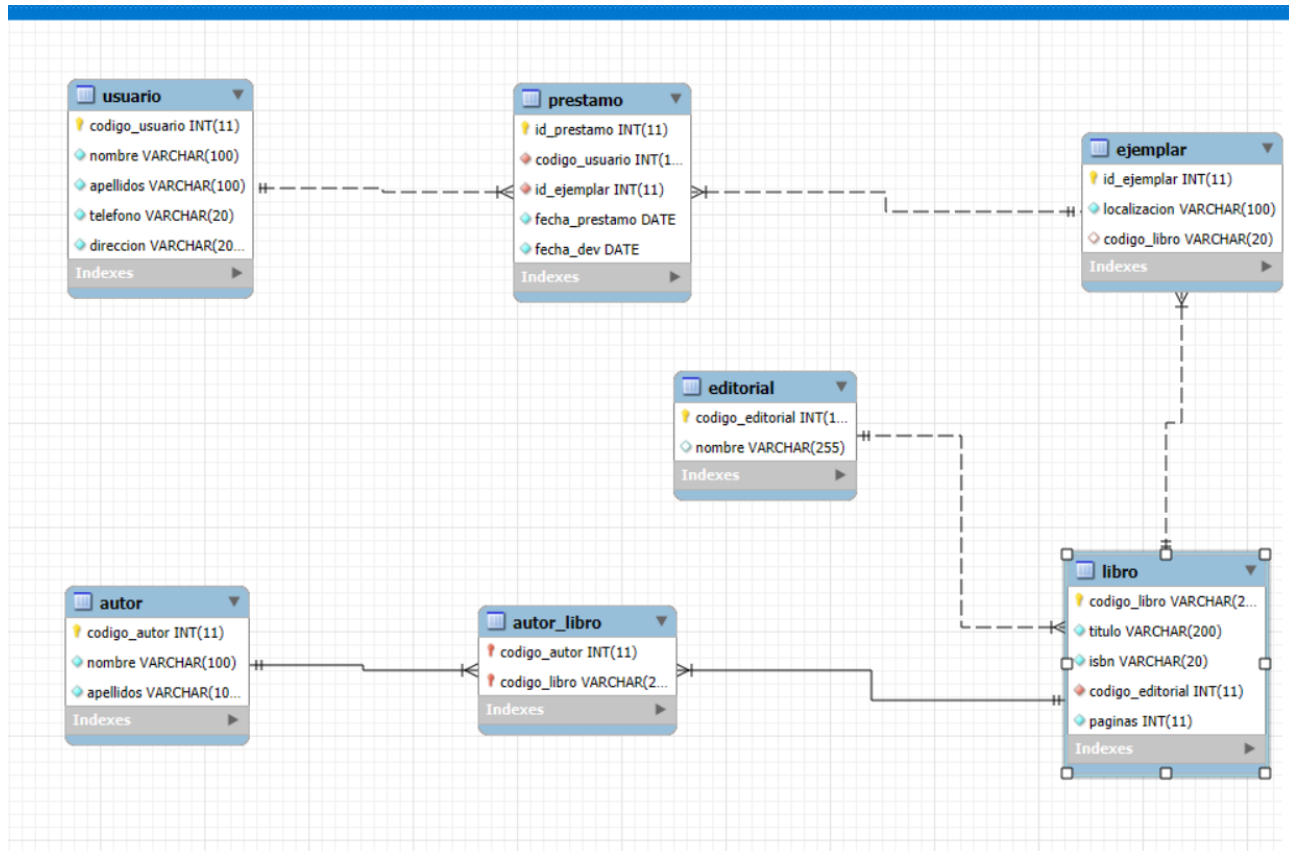
Modelo Entidad-Relación



Modelo Relacional



Modelo Relacional Generado por MySQL Workbench



Fragmentos de código de creación

DDL

```
DROP DATABASE IF EXISTS biblioteca;
CREATE DATABASE biblioteca;
USE biblioteca;

-- Tabla autor
DROP TABLE IF EXISTS autor;
CREATE TABLE autor (
    codigo_autor INT AUTO_INCREMENT
    PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT NULL
);

-- Tabla editorial
DROP TABLE IF EXISTS editorial;
CREATE TABLE editorial (
    codigo_editorial INT
    AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL
);

-- Tabla libro
DROP TABLE IF EXISTS libro;
CREATE TABLE libro (
    codigo_libro VARCHAR(20)
    PRIMARY KEY,
    titulo VARCHAR(200) NOT NULL,
    isbn VARCHAR(20) NOT NULL
    UNIQUE,
    codigo_editorial INT NOT NULL,
    paginas INT NOT NULL,
    CHECK (paginas > 0),
    FOREIGN KEY (codigo_editorial)
    REFERENCES
    editorial(codigo_editorial)
    ON DELETE CASCADE ON UPDATE
    CASCADE
);

-- Tabla ejemplar
CREATE TABLE ejemplar (
    id_ejemplar INT AUTO_INCREMENT
    PRIMARY KEY,
    localizacion VARCHAR(100) NOT
    NULL,
    codigo_libro VARCHAR(20),
    FOREIGN KEY (codigo_libro)
    REFERENCES libro(codigo_libro)
    ON DELETE CASCADE ON UPDATE
    CASCADE
);

-- Tabla usuario
CREATE TABLE usuario (
    codigo_usuario INT
    AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(100) NOT NULL,
    apellidos VARCHAR(100) NOT
    NULL,
    telefono VARCHAR(20) NOT NULL,
    direccion VARCHAR(200) NOT NULL
);

-- Tabla de relación autor-libro
(many-to-many)
DROP TABLE IF EXISTS autor_libro;
CREATE TABLE autor_libro (
    codigo_autor INT,
    codigo_libro VARCHAR(20),
    PRIMARY KEY (codigo_autor,
    codigo_libro),
    FOREIGN KEY (codigo_autor)
    REFERENCES autor(codigo_autor),
    FOREIGN KEY (codigo_libro)
    REFERENCES libro(codigo_libro)
);
```

```
-- Tabla de relación usuario-
ejemplar (préstamos)
DROP TABLE IF EXISTS prestamo;
CREATE TABLE prestamo (
    id_prestamo INT AUTO_INCREMENT
PRIMARY KEY,
    codigo_usuario INT NOT NULL,
    id_ejemplar INT NOT NULL,
    fecha_prestamo DATE NOT NULL,
    fecha_dev DATE NOT NULL,
    UNIQUE (codigo_usuario,
id_ejemplar, fecha_prestamo),
```

```
FOREIGN KEY (codigo_usuario)
REFERENCES usuario(codigo_usuario)
ON DELETE CASCADE ON UPDATE
CASCADE,
FOREIGN KEY (id_ejemplar)
REFERENCES ejemplar(id_ejemplar)
ON DELETE CASCADE ON UPDATE
CASCADE,
CHECK (fecha_prestamo <
fecha_dev)
);
```

DML

```
-- Insertar editoriales
INSERT INTO editorial (nombre)
VALUES
('Editorial Anagrama'),
('Alfaguara'),
('Planeta'),
('Penguin Random House'),
('Tusquets Editores');
```

```
-- Insertar autores
INSERT INTO autor (nombre,
apellidos) VALUES
('Gabriel', 'García Márquez'),
('Isabel', 'Allende'),
('Jorge Luis', 'Borges'),
('Julio', 'Cortázar'),
('Mario', 'Vargas Llosa');
```

```
-- Insertar libros
INSERT INTO libro (codigo_libro,
titulo, isbn, codigo_editorial,
paginas) VALUES
('L001', 'Cien años de soledad',
'9788420471839', 1, 432),
('L002', 'La casa de los
espíritus', '9788401352898', 2,
368),
('L003', 'Ficciones',
'9788426405159', 3, 218),
('L004', 'Rayuela',
'9788437604572', 4, 635),
('L005', 'La ciudad y los perros',
'9788432211928', 5, 382),
('L006', 'El amor en los tiempos
```

```
del cólera', '9788497592451', 1,
490),
('L007', 'El Aleph',
'9788499089515', 3, 210);
```

```
-- Relacionar autores con libros
INSERT INTO autor_libro
(codigo_autor, codigo_libro)
VALUES
(1, 'L001'), -- García Márquez -
Cien años de soledad
(1, 'L006'), -- García Márquez - El
amor en los tiempos del cólera
(2, 'L002'), -- Allende - La casa
de los espíritus
(3, 'L003'), -- Borges - Ficciones
(3, 'L007'), -- Borges - El Aleph
(4, 'L004'), -- Cortázar - Rayuela
(5, 'L005'); -- Vargas Llosa - La
ciudad y los perros
```

```
-- Insertar ejemplares
INSERT INTO ejemplar (localizacion,
codigo_libro) VALUES
('Estante A1-01', 'L001'),
('Estante A1-02', 'L001'),
('Estante A2-01', 'L002'),
('Estante B1-01', 'L003'),
('Estante B2-01', 'L004'),
('Estante C1-01', 'L005'),
('Estante A1-03', 'L006'),
('Estante B1-02', 'L007'),
('Estante A2-02', 'L002'),
('Estante C1-02', 'L005');
```

-- Insertar usuarios

```
INSERT INTO usuario (nombre,
apellidos, telefono, direccion)
VALUES
('Ana', 'Martínez López',
'612345678', 'Calle Mayor 15,
Madrid'),
('Carlos', 'González Ruiz',
'623456789', 'Avenida Libertad 23,
Barcelona'),
('Elena', 'Sánchez Pérez',
'634567890', 'Plaza España 7,
Sevilla'),
('Miguel', 'Rodríguez Torres',
'645678901', 'Calle Real 34,
Valencia'),
('Laura', 'Fernández Gómez',
'656789012', 'Calle Nueva 12,
Bilbao');
```

-- Insertar préstamos

```
INSERT INTO prestamo
(codigo_usuario, id_ejemplar,
fecha_prestamo, fecha_dev) VALUES
(1, 1, '2025-03-01', '2025-03-15'),
(1, 4, '2025-03-10', '2025-03-24'),
(2, 2, '2025-03-05', '2025-03-19'),
(3, 3, '2025-03-07', '2025-03-21'),
(4, 5, '2025-03-12', '2025-03-26'),
(5, 6, '2025-03-15', '2025-03-29'),
(2, 7, '2025-03-20', '2025-04-03'),
(3, 8, '2025-03-22', '2025-04-05');
```

INSERT INTO usuario VALUES

```
(1, 'Ana', 'Gómez Torres', '555-
1234', 'Calle Olmo 10, CDMX'),
(2, 'Luis', 'Martínez Ruiz', '555-
5678', 'Av. Reforma 100, CDMX'),
(3, 'Clara', 'Hernández Soto',
'555-8765', 'Calle Cedro 21, GDL'),
(4, 'Pedro', 'López Díaz', '555-
3456', 'Av. Juárez 50, MTY'),
(5, 'Sofía', 'Pérez Vargas', '555-
6789', 'Calle Roble 30, QRO');
```

INSERT INTO ejemplar VALUES

```
(100, 'El Quijote', 'Anagrama',
2005),
(101, 'Cien Años de Soledad',
'Sudamericana', 1999),
(102, 'La Sombra del Viento',
'Planeta', 2007),
(103, 'Rayuela', 'Alfaguara',
1985),
(104, 'Pedro Páramo', 'FCE', 1974);
```

INSERT INTO prestamo VALUES

```
(1, 1, 100, '2024-01-05', '2024-01-
15'),
(2, 2, 101, '2024-02-10', '2024-02-
20'),
(3, 3, 102, '2024-03-01', '2024-03-
15'),
(4, 4, 103, '2024-04-05', '2024-04-
25'),
(5, 5, 104, '2024-05-10', '2024-05-
20');
```


Conclusiones

A lo largo del desarrollo de esta base de datos para el sistema de gestión de biblioteca, se logró comprender y aplicar de manera práctica los fundamentos del diseño relacional. Se identificaron entidades clave como libros, autores, usuarios, ejemplares, préstamos, y editoriales, y se establecieron relaciones entre ellas a través de claves primarias y foráneas, respetando las reglas de integridad referencial.

Además, se experimentó con la carga masiva de datos mediante archivos CSV utilizando `LOAD DATA INFILE`, lo cual demostró ser una herramienta eficiente para poblar grandes volúmenes de información, siempre y cuando se validen previamente los datos para evitar errores como claves duplicadas o referencias inválidas.

También se aprendió a manejar errores comunes (como los de integridad referencial y datos truncados), a limpiar datos desde scripts de Python, y a verificar su consistencia antes de insertarlos en la base. Estas tareas, aunque desafiantes, reforzaron el conocimiento sobre la importancia de un buen diseño y una correcta preparación de los datos.

En resumen, este proyecto no solo ayudó a desarrollar habilidades técnicas relacionadas con SQL y Python, sino que también promovió una visión más completa sobre cómo estructurar, alimentar y consultar bases de datos reales de forma eficiente, escalable y ordenada.