

Proyecto: Sistema de Gestion de Datos - Avance 3

Integrantes del equipo:

Esteban Alvarez Garcia

Juan Pablo Lopera Arrazola

Profesor : Edwin Nelson Montoya Munera

Universidad Eafit.

2025

Diseño Lógico y Diseño Físico de la Base de Datos

En el diseño de bases de datos se distinguen principalmente dos etapas: el diseño lógico y el diseño físico. Este documento muestra ambos modelos aplicados al sistema de gestión de usuarios, publicaciones, libros, reseñas, ventas, intercambios y clubes de lectura.

1. Diseño Lógico

El diseño lógico corresponde al modelo relacional. Define las tablas, atributos, claves primarias y claves foráneas, sin detallar los tipos de datos específicos del motor.

- Usuario(ID_usuario PK, nombre, correo, telefono, password, ciudad)
- Catalogo(ID_catalogo PK, nombre, descripcion, fecha_creacion)
- Club_lectura(ID_club PK, nombre, descripcion, estado, fecha_creacion)
- Libro(ID_libro PK1, ISBN PK2, titulo, autor, año_publicacion, genero, modalidad, resumen, propietario, ID_usuario_recibe FK→Usuario)
- Publicacion(ID_publicacion PK, modalidad, estado, fecha_publicacion, precio, ID_usuario FK→Usuario, ID_catalogo FK→Catalogo, (ID_libro,ISBN) FK→Libro)
- Reseña(ID_reseña PK, calificacion, contenido, fecha_reseña, ID_usuario_autor FK→Usuario, ID_usuario_comenta FK→Usuario, (ID_libro,ISBN) FK→Libro)
- Intercambio(ID_intercambio PK, fecha, estado, ID_usuario FK→Usuario, (ID_libro,ISBN) FK→Libro, UNIQUE(ID_libro,ISBN))
- Venta(ID_venta PK, fecha_venta, estado, precio, ID_usuario FK→Usuario, (ID_libro,ISBN) FK→Libro, UNIQUE(ID_libro,ISBN))
- usuario_club(ID_usuario PK, ID_club PK, fecha_union, FK Usuario, FK Club_lectura)
- club_lectura_libro(ID_club PK, ID_libro PK, ISBN PK, FK Club_lectura, FK Libro)

2. Diseño Físico

El diseño físico es la implementación concreta en el motor de base de datos (MySQL). Incluye tipos de datos, restricciones y la sintaxis exacta de las claves primarias y foráneas. A continuación, se presenta el código SQL completo:

```
CREATE TABLE `Usuario` (  
  `ID_usuario` INT PRIMARY KEY AUTO_INCREMENT,  
  `nombre` VARCHAR(100) NOT NULL,  
  `correo` VARCHAR(120) UNIQUE,  
  `telefono` VARCHAR(30),  
  `password` VARCHAR(255) NOT NULL,  
  `ciudad` VARCHAR(100)
```

);

```
CREATE TABLE `Catalogo` (  
  `ID_catalogo` INT PRIMARY KEY AUTO_INCREMENT,  
  `nombre` VARCHAR(120) NOT NULL,  
  `descripcion` TEXT,  
  `fecha_creacion` DATE NOT NULL  
);
```

```
CREATE TABLE `Club_lectura` (  
  `ID_club` INT PRIMARY KEY AUTO_INCREMENT,  
  `nombre` VARCHAR(120) NOT NULL,  
  `descripcion` TEXT,  
  `estado` VARCHAR(30),  
  `fecha_creacion` DATE NOT NULL  
);
```

```
CREATE TABLE `Libro` (  
  `ID_libro` INT NOT NULL,  
  `ISBN` VARCHAR(20) NOT NULL,  
  `titulo` VARCHAR(200) NOT NULL,  
  `autor` VARCHAR(150) NOT NULL,  
  `año_publicacion` INT,  
  `genero` VARCHAR(80),  
  `modalidad` VARCHAR(40),  
  `resumen` TEXT,  
  `propietario` VARCHAR(120),  
  `ID_usuario_recibe` INT,  
  CONSTRAINT `pk_libro` PRIMARY KEY (`ID_libro`, `ISBN`),  
  CONSTRAINT `fk_libro_usuario_recibe`  
    FOREIGN KEY (`ID_usuario_recibe`) REFERENCES `Usuario` (`ID_usuario`)  
    ON UPDATE CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE `Publicacion` (  
  `ID_publicacion` INT PRIMARY KEY AUTO_INCREMENT,  
  `modalidad` VARCHAR(40),  
  `estado` VARCHAR(30),  
  `fecha_publicacion` DATE,  
  `precio` DECIMAL(10,2) CHECK (`precio` >= 0),  
  `ID_usuario` INT NOT NULL,  
  `ID_catalogo` INT NOT NULL,  
  `ID_libro` INT NOT NULL,
```

```

`ISBN`          VARCHAR(20) NOT NULL,
CONSTRAINT `fk_publicacion_usuario`
  FOREIGN KEY (`ID_usuario`) REFERENCES `Usuario`(`ID_usuario`)
  ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT `fk_publicacion_catalogo`
  FOREIGN KEY (`ID_catalogo`) REFERENCES `Catalogo`(`ID_catalogo`)
  ON UPDATE CASCADE ON DELETE RESTRICT,
CONSTRAINT `fk_publicacion_libro`
  FOREIGN KEY (`ID_libro`, `ISBN`) REFERENCES `Libro`(`ID_libro`, `ISBN`)
  ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE `Reseña` (
  `ID_resena` INT PRIMARY KEY AUTO_INCREMENT,
  `calificacion` TINYINT CHECK (`calificacion` BETWEEN 1 AND 5),
  `contenido` TEXT,
  `fecha_resena` DATE,
  `ID_usuario_autor` INT NOT NULL,
  `ID_usuario_comenta` INT,
  `ID_libro` INT NOT NULL,
  `ISBN` VARCHAR(20) NOT NULL,
  CONSTRAINT `fk_resena_usuario_autor`
    FOREIGN KEY (`ID_usuario_autor`) REFERENCES `Usuario`(`ID_usuario`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT `fk_resena_usuario_comenta`
    FOREIGN KEY (`ID_usuario_comenta`) REFERENCES `Usuario`(`ID_usuario`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT `fk_resena_libro`
    FOREIGN KEY (`ID_libro`, `ISBN`) REFERENCES `Libro`(`ID_libro`, `ISBN`)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE `Intercambio` (
  `ID_intercambio` INT PRIMARY KEY AUTO_INCREMENT,
  `fecha` DATE,
  `estado` VARCHAR(30),
  `ID_usuario` INT NOT NULL,
  `ID_libro` INT NOT NULL,
  `ISBN` VARCHAR(20) NOT NULL,
  CONSTRAINT `fk_intercambio_usuario`
    FOREIGN KEY (`ID_usuario`) REFERENCES `Usuario`(`ID_usuario`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
  CONSTRAINT `fk_intercambio_libro`

```

```

    FOREIGN KEY (`ID_libro`, `ISBN`) REFERENCES `Libro`(`ID_libro`, `ISBN`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `uq_intercambio_libro` UNIQUE (`ID_libro`, `ISBN`)
);

```

```

CREATE TABLE `Venta` (
    `ID_venta` INT PRIMARY KEY AUTO_INCREMENT,
    `fecha_venta` DATE,
    `estado` VARCHAR(30),
    `precio` DECIMAL(10,2) CHECK (`precio` >= 0),
    `ID_usuario` INT NOT NULL,
    `ID_libro` INT NOT NULL,
    `ISBN` VARCHAR(20) NOT NULL,
    CONSTRAINT `fk_venta_usuario`
    FOREIGN KEY (`ID_usuario`) REFERENCES `Usuario`(`ID_usuario`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_venta_libro`
    FOREIGN KEY (`ID_libro`, `ISBN`) REFERENCES `Libro`(`ID_libro`, `ISBN`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `uq_venta_libro` UNIQUE (`ID_libro`, `ISBN`)
);

```

```

CREATE TABLE `usuario_club` (
    `ID_usuario` INT NOT NULL,
    `ID_club` INT NOT NULL,
    `fecha_union` DATE NOT NULL,
    CONSTRAINT `pk_usuario_club` PRIMARY KEY (`ID_usuario`, `ID_club`),
    CONSTRAINT `fk_uc_usuario`
    FOREIGN KEY (`ID_usuario`) REFERENCES `Usuario`(`ID_usuario`)
    ON UPDATE CASCADE ON DELETE RESTRICT,
    CONSTRAINT `fk_uc_club`
    FOREIGN KEY (`ID_club`) REFERENCES `Club_lectura`(`ID_club`)
    ON UPDATE CASCADE ON DELETE RESTRICT
);

```

```

CREATE TABLE `club_lectura_libro` (
    `ID_club` INT NOT NULL,
    `ID_libro` INT NOT NULL,
    `ISBN` VARCHAR(20) NOT NULL,
    CONSTRAINT `pk_club_libro` PRIMARY KEY (`ID_club`, `ID_libro`, `ISBN`),
    CONSTRAINT `fk_club_libro_club`
    FOREIGN KEY (`ID_club`) REFERENCES `Club_lectura`(`ID_club`)
    ON UPDATE CASCADE ON DELETE RESTRICT,

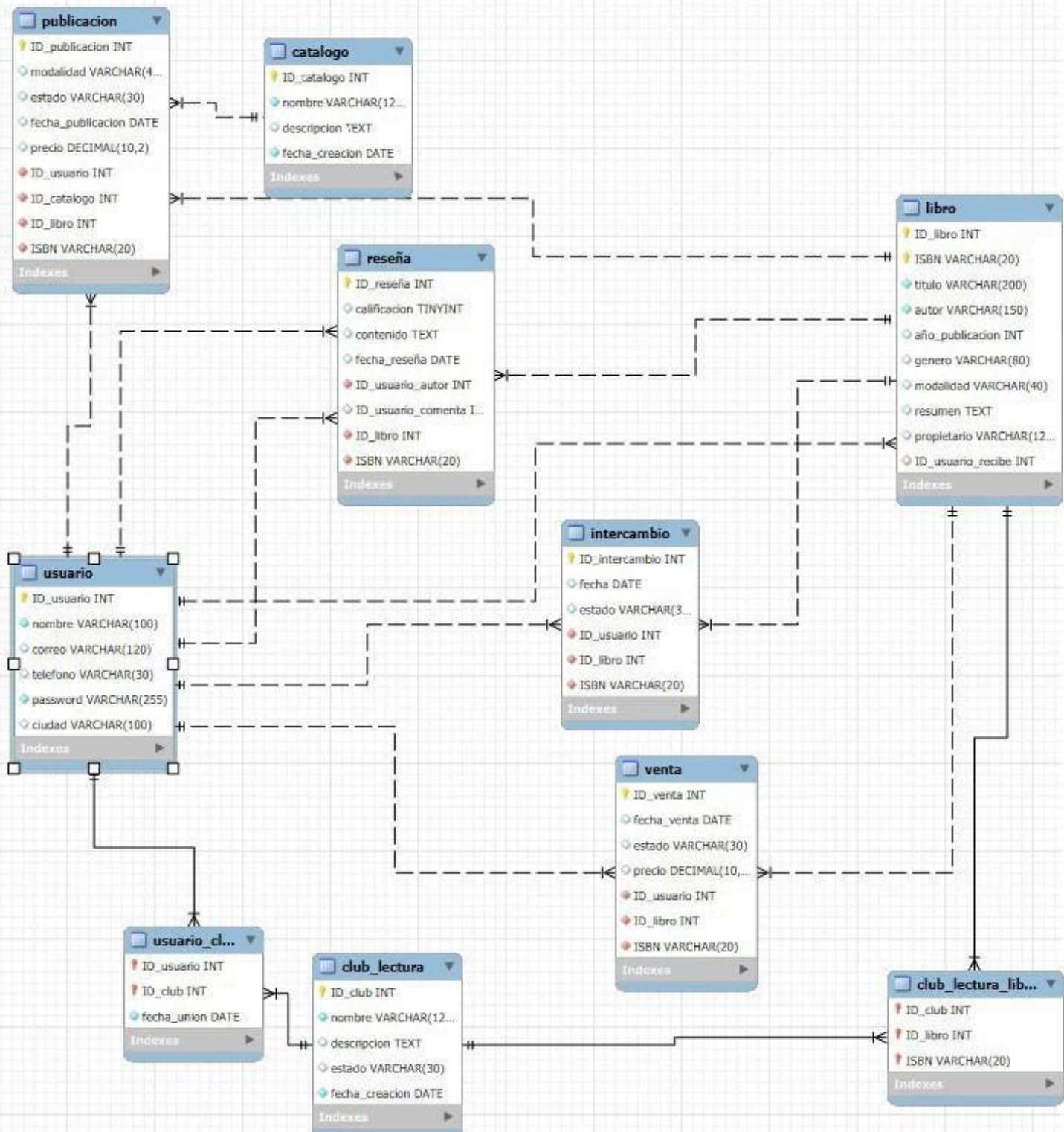
```

```

CONSTRAINT `fk_club_libro_libro`
FOREIGN KEY (`ID_libro`,`ISBN`) REFERENCES `Libro`(`ID_libro`,`ISBN`)
ON UPDATE CASCADE ON DELETE RESTRICT
);

```

3. Diseño de SQL



4. CONSULTAS SQL

CONSULTAS SQL

Las consultas SQL corresponden a la etapa de interacción directa con la base de datos, en la cual se formulan instrucciones que permiten obtener información relevante a partir de las tablas del modelo.

Estas sentencias permiten analizar, organizar y presentar los datos de manera estructurada dentro del sistema, facilitando así la verificación del funcionamiento del modelo lógico y físico del proyecto “Club de Lectura”.

A continuación, se presentan las 15 consultas SQL principales, diseñadas para responder a los requerimientos de información del sistema y demostrar las relaciones entre los diferentes módulos (Usuarios, Libros, Publicaciones, Reseñas, Ventas, Intercambios y Clubes de Lectura):

1. Listar todos los libros publicados en el catálogo ordenados por título.
2. Listar todos los libros disponibles para intercambio en una ciudad específica.
3. Buscar libros por título, autor o resumen.
4. Buscar reseñas dentro de un libro.
5. Listar todas las reseñas de un libro dado con su calificación promedio.
6. Listar los clubes de lectura activos y sus miembros.
7. Listar las propuestas de intercambio pendientes que ha recibido un usuario.
8. Listar las órdenes de compra de un usuario indicando su estado.
9. Listar los usuarios que han hecho más reseñas en el catálogo.
10. Listar los libros más intercambiados en un rango de fechas.
11. Listar los libros en venta dentro de un rango de precios.

12. Listar los libros mejor calificados con un mínimo de reseñas.
13. Listar los clubes de lectura y el total de miembros de cada club.
14. Listar el número de libros vendidos por mes.
15. Listar los autores con el mayor número de libros publicados.