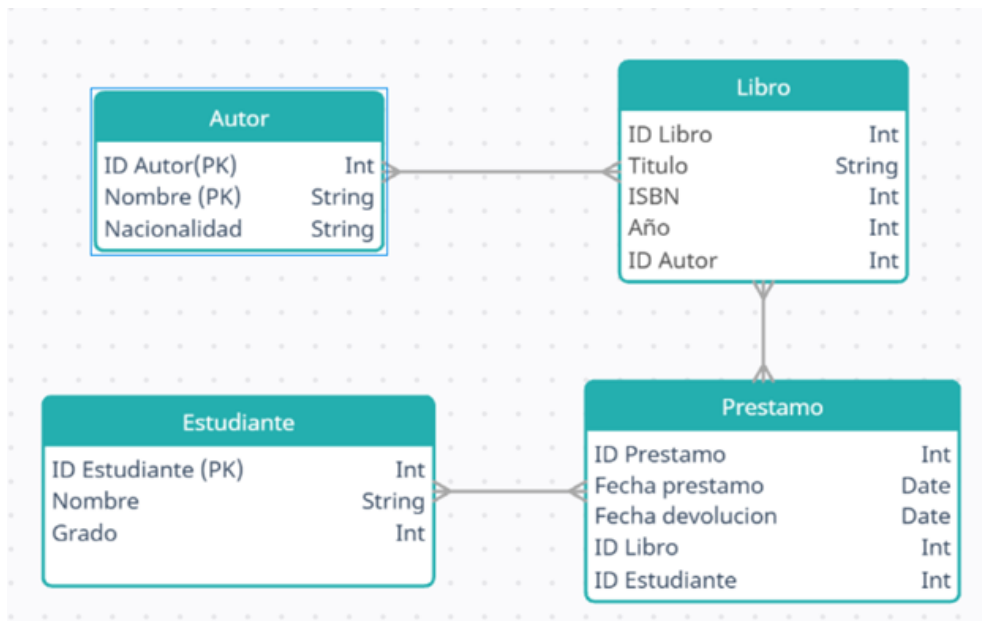


Fase 2: Normalización del Sistema de Biblioteca

En este documento se presenta la explicación de que significa cada nivel de normalización en un modelo relacional, incluyendo los problemas que se evitan al aplicar cada uno de estos. Así mismo se hará la conversión del modelo ER de la fase 1 a un modelo relacional normalizado hasta la Tercera Forma Normal (3FN). El objetivo es eliminar redundancias, garantizar la integridad de datos y optimizar el rendimiento de la base de datos.

Diagrama ER diseñado en la fase anterior.



Modelo relacional resultante

Atributos de Autor	Restricciones
ID_Autor	INT PRIMARY KEY
Nombre	NOT NULL
Nacionalidad	NOT NULL

Atributos de Libro	Restricciones
ID_Libro	INT PRIMARY KEY
Título	NOT NULL
ISBN	UNIQUE
Año	INT
ID_Autor	INT
FOREIGN KEY (ID_Autor) REFERENCES Autor(ID_Autor)	

Atributos de Estudiante	Restricciones
ID_Estudiante	INT PRIMARY KEY
Nombre	NOT NULL
Grado	INT

Atributos de Prestamo	Restricciones
ID_Prestamo	INT PRIMARY KEY
Fecha_prestamo	DATE NOT NULL
Fecha_devolucion	DATE
ID_Libro	INT
ID_Estudiante	INT
FOREIGN KEY (ID_Libro) REFERENCES Libro(ID_Libro),	
FOREIGN KEY (ID_Estudiante) REFERENCES Estudiante(ID_Estudiante)	

Normalización

1. **Primera Forma Normal (1FN):** Se cumple cuando todos los atributos son atómicos, es decir se guarda un solo valor en cada celda. Por ejemplo, un libro tiene un solo ISBN, un solo título, etc. Esto evita problemas como almacenar varios autores en un solo campo.
2. **Segunda Forma Normal (2FN):** Se cumple cuando la tabla está en 1FN y ninguna tiene claves primarias compuestas que generen dependencias parciales. Cada tabla tiene una clave primaria simple, por ejemplo: id_libro en Libro o id_autor en Autor. Y en el caso del préstamo, en sus atributos están los id de Libro y Autor, pero su clave es primaria es solamente id_prestamo.
3. **Tercera Forma Normal (3FN):** Se cumple cuando la tabla está en 2FN y no existen dependencias transitivas. Por ejemplo, en la tabla Libro no se repite el nombre del autor, solo se almacena su id_autor. Esto evita redundancia, ya que los datos del autor están únicamente en la tabla Autor.

Ejemplo de desnormalización evitada

Si en la tabla Libro se almacenara el nombre del autor en lugar de su ID, cada vez que un autor publicara un libro nuevo se repetiría su nombre y nacionalidad, generando redundancia de la información. Al normalizar y cumplir hasta la tercera forma, esos datos se almacenan una sola vez en la tabla Autor y no se repiten innecesariamente.