

EJERCICIO PROPUESTO:

La librería Busca-libre desea mantener información de los libros que vende a sus clientes, la editorial y la información de su autor.

El identificador único de un libro es el ISBN (código universal) y un libro tiene una editorial. El nombre de la editorial es el identificador único de la editorial. Una editorial puede haber publicado muchos libros que la librería tiene en su existencia; sin embargo, la librería también desea mantener información de editoriales que no tienen libros en su inventario. Un libro tiene título, número de páginas y debe tener por lo menos un autor, pero puede tener muchos autores. Un autor es de interés para la librería si ha escrito por lo menos un libro y posiblemente muchos libros que tenga en su existencia; se desea almacenar la nacionalidad y fecha de nacimiento del autor, para identificar al autor se le asignará un código. Un libro de la librería puede haber sido comprado por muchos clientes, o que aún no haya sido comprado. Para que un cliente sea de interés de la librería debe haber comprado por lo menos un libro y posiblemente muchos.

SOLUCION:

Para dar solución a este ejercicio se crearon las siguientes entidades:

Entidades:

- Editoriales: Editoriales cuenta con una llave primaria (NombreEditorial) y un atributo compuesto correspondiente a la Dirección que se compone de Ciudad y Complemento.
- Libros: Libros cuenta con una llave primaria (ISBN), un atributo Título y uno Número de páginas.
- Clientes: Clientes cuenta con una llave primaria (ID), un atributo Nombre del cliente y uno Multivariado para los teléfonos del cliente.
- Autores: Autores cuenta con una llave primaria (Código de autor), un atributo Nombre del autor un atributo Nacionalidad, un atributo fecha de nacimiento y edad.

Una vez definidas las entidades y sus respectivos atributos se definieron las relaciones, dando este resultado:

Relaciones:

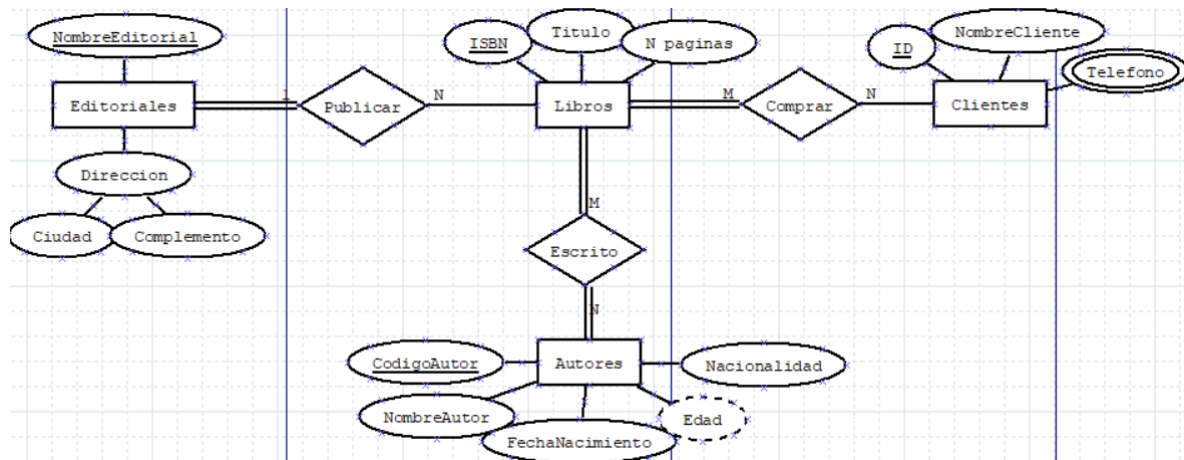
- Una editorial puede publicar uno o muchos libros, por otra parte, un libro solo puede ser publicado por una editorial. (1, N)
- Un libro puede ser escrito por uno o varios autores, y un autor puede escribir uno o muchos libros. (M, N)
- Un libro puede ser comprado por uno o varios clientes, y un cliente puede comprar uno o muchos libros. (M, N)

Ya que se definieron las relaciones, el siguiente paso es definir las participaciones, en este caso se identificaron algunas participaciones totales que se mencionan a continuación, las no mencionadas son parciales:

Participaciones:

- Editoriales cuenta con participación Total en su relación con Libros
- Libros cuenta con participación Total en su relación con Clientes y con Autores.
- Autores cuenta con participación total en su relación con Libros.

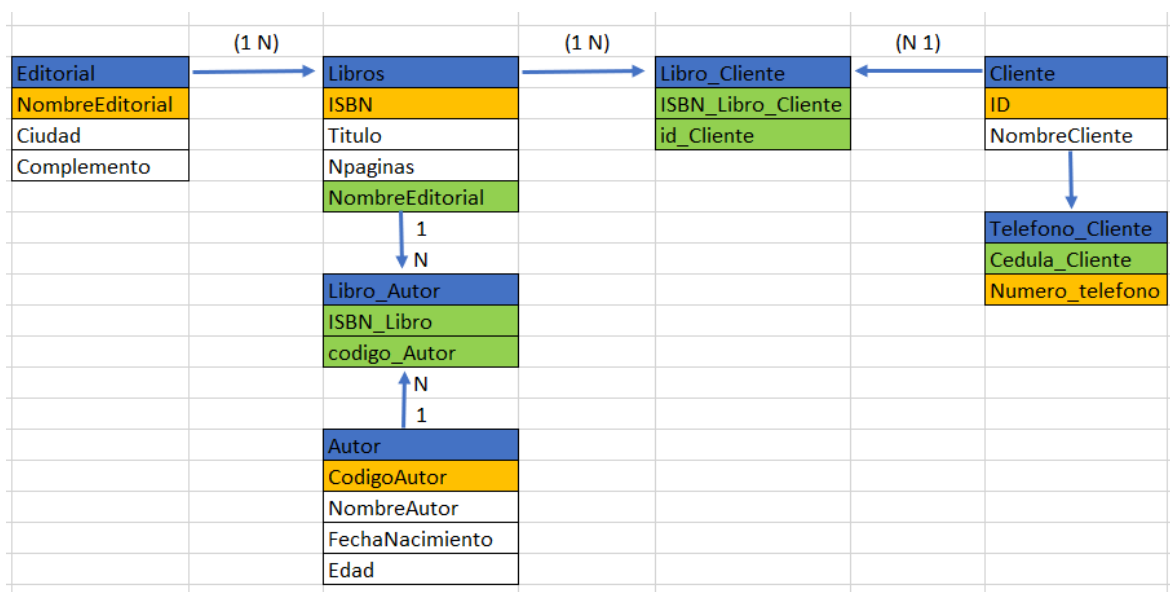
Con todo lo anteriormente mencionado este es el Diagrama ER de la base de datos Librería:



Una vez realizado el diagrama Entidad-Relación se procedió a continuar con la creación del Modelo relacional, a continuación, los pasos:

- Se concreto la relación de uno a muchos entre Editorial y Libros agregando una llave foránea en Libros correspondiente al nombre de la editorial.
- Para concretar la relación de muchos a muchos entre Libros y autor se creo una tabla que los relacionara en la cual se almacenan las respectivas llaves primarias.
- Se realizo el mismo procedimiento anteriormente mencionado para la relación entre Libros y Cliente
- Se creo una tabla mas para el atributo multivaluado con el cual contaba cliente, permitiendo guardar los números de teléfono de cada cliente.

Realizada la transformación este fue el resultado:



Para completar el proceso se realiza la respectiva normalización:

Normalización:

N1:

Normas	Estado
Todos los atributos tienen valores atómicos	Cumple
No hay atributos multivaluados	Cumple
No existen registros duplicados	Cumple
Se eliminaron todas las columnas repetidas	Cumple
Definir clave principal	Cumple

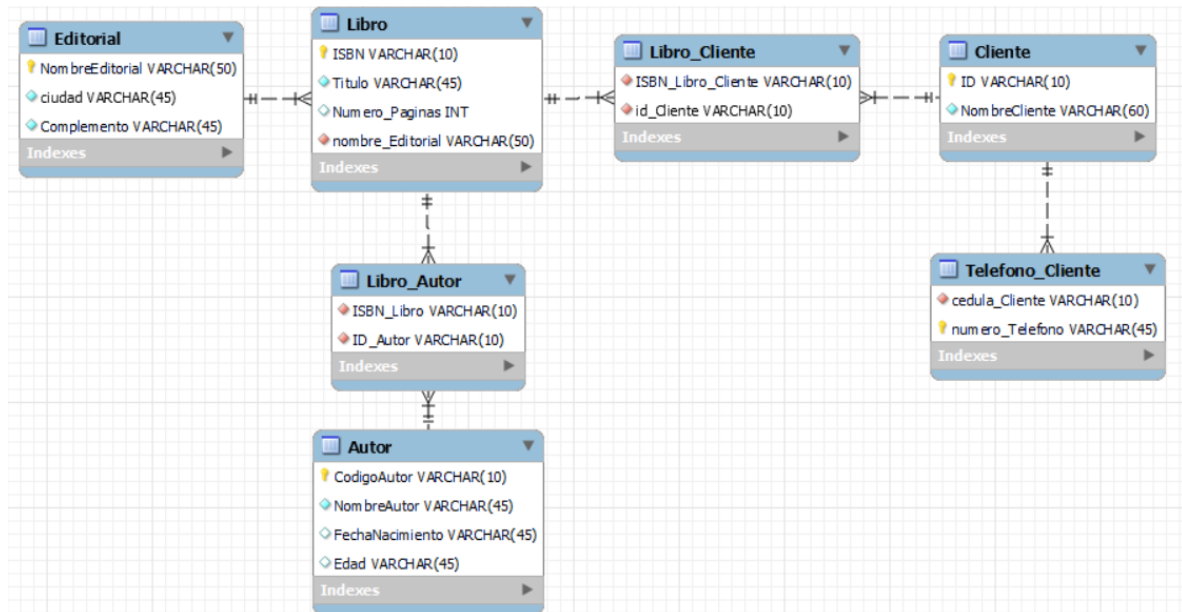
N2:

Normas	Estado
Cumple con la norma 1	Cumple
Los valores de las columnas dependen solo de la llave primaria	Cumple
Las tablas tienen una única llave primaria que las identifique	Cumple

N3:

Normas	Estado
Cumple la norma 2	Cumple
Los atributos no incluidos en la clave primaria no dependen transitivamente la clave primaria	Cumple

Una vez completado el modelo relacional se procede a realizar el Diagrama en Workbench, que luego será usado para crear la base de datos:



Siguiendo con los procesos se creó la base de datos Librería usando sentencias SQL, a continuación las sentencias y un ejemplo de los registros:

Creación de la base de datos, y la tabla editorial:

```

CREATE SCHEMA IF NOT EXISTS `Libreria` DEFAULT CHARACTER SET utf8 ;
USE `Libreria` ;

CREATE TABLE IF NOT EXISTS `Libreria`.`Editorial` (
  `NombreEditorial` VARCHAR(50) NOT NULL,
  `ciudad` VARCHAR(45) NOT NULL,
  `Complemento` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`NombreEditorial`));
  
```

Ejemplo registros tabla editorial:

```

3 • INSERT INTO `libreria`.`editorial` (`NombreEditorial`, `ciudad`, `Complemento`) VALUES ('Libros y Libros', 'Bogota', 'Norte');
4 • INSERT INTO `libreria`.`editorial` (`NombreEditorial`, `ciudad`, `Complemento`) VALUES ('Libro al viento', 'Medellin', 'El poblado');
  
```

NombreEditorial	ciudad	Complemento
Libro al viento	Medellin	El poblado
Libros y Libros	Bogota	Norte

Creación tabla Libro:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Libro` (  
  `ISBN` VARCHAR(10) NOT NULL,  
  `Titulo` VARCHAR(45) NOT NULL,  
  `Numero_Paginas` INT NULL,  
  `nombre_Editorial` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`ISBN`),  
  INDEX `nombre_Editorial_idx` (`nombre_Editorial` ASC) VISIBLE,  
  CONSTRAINT `nombre_Editorial`  
    FOREIGN KEY (`nombre_Editorial`)  
      REFERENCES `Libreria`.`Editorial` (`NombreEditorial`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros tabla libro:

```
3 • INSERT INTO `libreria`.`libro` (`ISBN`, `Titulo`, `Numero_Paginas`, `nombre_Editorial`) VALUES ('12L', 'Arroz', '37', 'Libros y Libros')  
4 • INSERT INTO `libreria`.`libro` (`ISBN`, `Titulo`, `Numero_Paginas`, `nombre_Editorial`) VALUES ('13I', 'Cielo', '50', 'Libro al viento')  
5
```

Result Grid			
ISBN	Titulo	Numero_Paginas	nombre_Editorial
12L	Arroz	37	Libros y Libros
13I	Cielo	50	Libro al viento

Creación tabla cliente:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Cliente` (  
  `ID` VARCHAR(10) NOT NULL,  
  `NombreCliente` VARCHAR(60) NOT NULL,  
  PRIMARY KEY (`ID`));
```

Ejemplo registros cliente:

```
4 • INSERT INTO `libreria`.`cliente` (`ID`, `NombreCliente`) VALUES ('123456789', 'Carlos Sanchez');  
5 • INSERT INTO `libreria`.`cliente` (`ID`, `NombreCliente`) VALUES ('987654312', 'Andrea ramirez');
```

Result Grid	
ID	NombreCliente
123456789	Carlos Sanchez
▶ 987654312	Andrea ramirez

Creación de la tabla autor:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Autor` (  
  `CodigoAutor` VARCHAR(10) NOT NULL,  
  `NombreAutor` VARCHAR(45) NOT NULL,  
  `FechaNacimiento` VARCHAR(45) NULL,  
  `Edad` VARCHAR(45) NULL,  
  PRIMARY KEY (`CodigoAutor`));
```

Ejemplo registros tabla autor:

```
4 • INSERT INTO `libreria`.`autor` (`CodigoAutor`, `NombreAutor`, `FechaNacimiento`, `Edad`) VALUES ('89', 'David Lopes', '1/1/2000', '23');  
5 • INSERT INTO `libreria`.`autor` (`CodigoAutor`, `NombreAutor`, `FechaNacimiento`, `Edad`) VALUES ('90', 'Camila Yepez', '2/1/2000', '24');
```

Result Grid				
Filter Rows:				
Edit: Export/Import: Wrap Cell Content:				
CodigoAutor	NombreAutor	FechaNacimiento	Edad	
89	David Lopes	1/1/2000	23	
90	Camila Yepez	2/1/2000	24	

Creación de la tabla libro-cliente:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Libro_Cliente` (  
  `ISBN_Libro_Cliente` VARCHAR(10) NOT NULL,  
  `id_Cliente` VARCHAR(10) NOT NULL,  
  INDEX `ISBN_Libro_Cliente_idx` (`ISBN_Libro_Cliente` ASC) VISIBLE,  
  INDEX `id_Cliente_idx` (`id_Cliente` ASC) VISIBLE,  
  PRIMARY KEY (`ISBN_Libro_Cliente`, `id_Cliente`),  
  CONSTRAINT `ISBN_Libro_Cliente`  
    FOREIGN KEY (`ISBN_Libro_Cliente`)  
      REFERENCES `Libreria`.`Libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `id_Cliente`  
    FOREIGN KEY (`id_Cliente`)  
      REFERENCES `Libreria`.`Cliente` (`ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros libro-cliente:

```
5 • INSERT INTO `libreria`.`libro_cliente` (`ISBN_Libro_Cliente`, `id_Cliente`) VALUES ('12L', '123456789');
6 • INSERT INTO `libreria`.`libro_cliente` (`ISBN_Libro_Cliente`, `id_Cliente`) VALUES ('13L', '987654312');
7
```

ISBN_Libro_Cliente	id_Cliente
12L	123456789
13L	987654312

Creación tabla libro-autor:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Libro_Autor` (  
  `ISBN_Libro` VARCHAR(10) NOT NULL,  
  `ID_Autor` VARCHAR(10) NOT NULL,  
  INDEX `ISBN_Libro_idx` (`ISBN_Libro` ASC) VISIBLE,  
  INDEX `ID_Autor_idx` (`ID_Autor` ASC) VISIBLE,  
  PRIMARY KEY (`ISBN_Libro`, `ID_Autor`),  
  CONSTRAINT `ISBN_Libro`  
    FOREIGN KEY (`ISBN_Libro`)  
      REFERENCES `Libreria`.`Libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `ID_Autor`  
    FOREIGN KEY (`ID_Autor`)  
      REFERENCES `Libreria`.`Autor` (`CodigoAutor`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros libro-autor:

```
4 • INSERT INTO `libreria`.`libro_autor` (`ISBN_Libro`, `ID_Autor`) VALUES ('12L', '89');
5 • INSERT INTO `libreria`.`libro_autor` (`ISBN_Libro`, `ID_Autor`) VALUES ('13L', '90');
```

ISBN_Libro	ID_Autor
12L	89
13L	90

Creación Teléfono-cliente:

```
CREATE TABLE IF NOT EXISTS `Libreria`.`Telefono_Cliente` (  
  `cedula_Cliente` VARCHAR(10) NOT NULL,  
  `numero_Telefono` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`numero_Telefono`, `cedula_Cliente`),  
  INDEX `cedula_Cliente_idx` (`cedula_Cliente` ASC) VISIBLE,  
  CONSTRAINT `cedula_Cliente`  
    FOREIGN KEY (`cedula_Cliente`)  
    REFERENCES `Libreria`.`Cliente` (`ID`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros Teléfono-cliente:

```
5 • INSERT INTO `libreria`.`telefono_cliente` (`cedula_Cliente`, `numero_Telefono`) VALUES ('123456789', '7500500');  
6 • INSERT INTO `libreria`.`telefono_cliente` (`cedula_Cliente`, `numero_Telefono`) VALUES ('123456789', '3203305');
```

sult Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
cedula_Cliente	numero_Telefono				
123456789	3203305				
123456789	7500500				