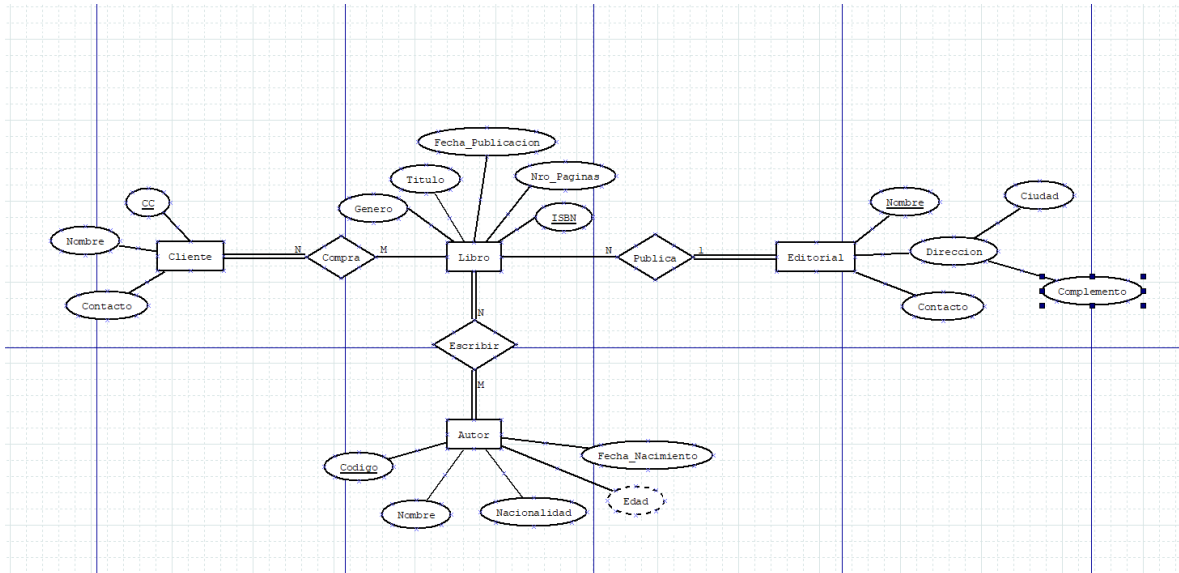
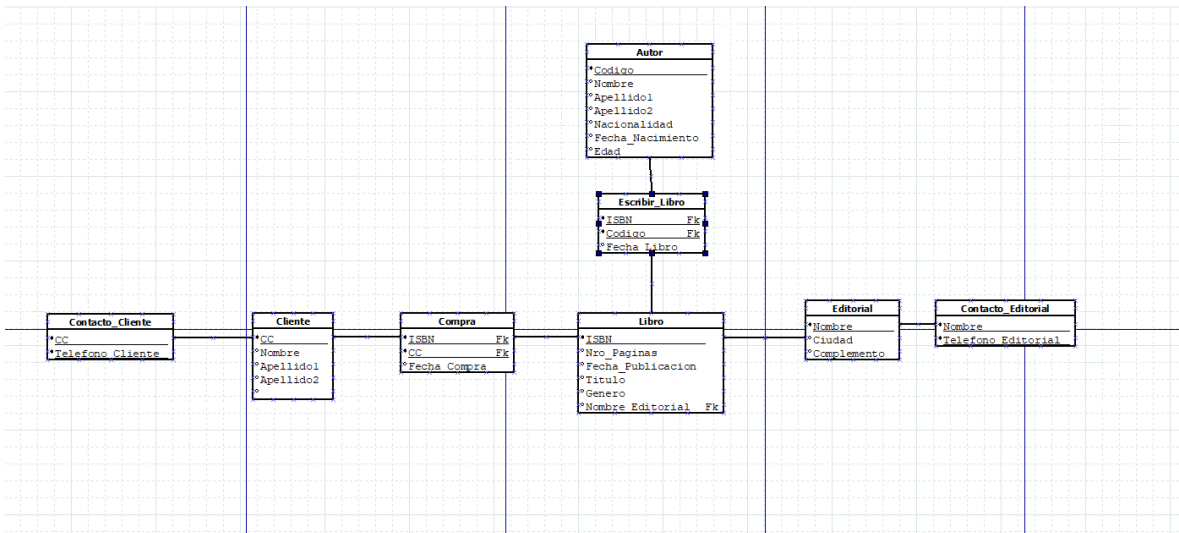


1) Modelo Entidad-Relación (Actividad libro)



2) Transformación a modelo relacional



Transformación:

Relación N:M:

- Se crea la tabla "Compra" ya que la relación entre el cliente y libro es de M:N, esta tabla contiene la llave primaria tanto de cliente como de libro y un atributo nuevo que nos permite registrar la fecha de la compra.
- Se crea la tabla "Escribir_Libro" por la relación de M:N entre la tabla autor y la tabla libro, esta con tiene las llaves primarias de las tablas mencionadas y un atributo nuevo que permite guardar la fecha en que el autor escribió el libro.

Relación 1:N:

- Libro y editorial al tener una relación de N:1, solo debemos guardar la llave primaria de editorial en la tabla de libros, esta es guardada como una llave foránea.

Atributos multivaluados:

- Se crean la tablas de Contacto_Cliente y Contacto_Editorial para almacenar los números de contacto de sus respectivas tablas, ya que estas entidades pueden tener mas de un numero de contacto.

Atributos compuestos:

- Para tratar estos atributos solo se generaron mas columnas en las tablas que contenían un atributo de este tipo, como por ejemplo en la tabla Editorial, se crearon dos atributos nuevos (ciudad y complemento) que reemplazan a dirección.

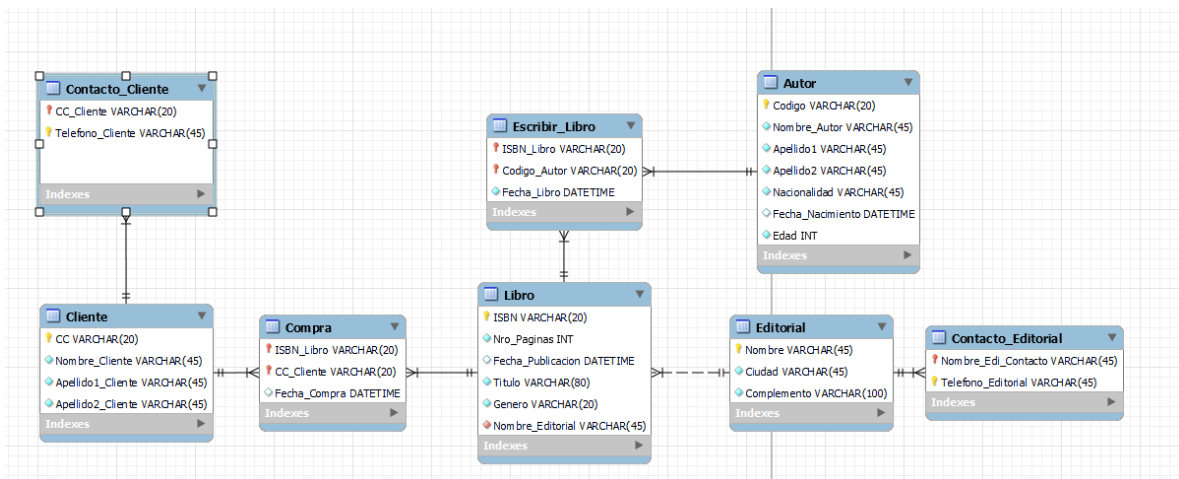
Normalización:

Primera forma: No se encontró redundancia en el modelo, cada tabla tiene una llave principal y se crearon las tablas necesarias para que cada tabla guarde solo datos que sean de su área, por ende, se llega a la consideración de que el modelo ya esta en primera forma normal.

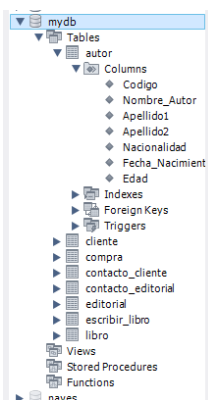
Segunda forma: Todas las tablas están conectadas y cada tabla tiene una llave que almacena la información que le corresponde sin depende de otra tabla, por ende, se llega a la consideración que el modelo ya se encuentra en segunda forma normal

Tercera forma: No fue necesario eliminar ningún atributo, ninguno de los atributos que se consideraron incumplen las condiciones para pertenecer a su tabla, por ende, se llega a la conclusión de que el modelo ya esta en tercera forma normal.

3) Diagrama en Workbench SQL



Schema creado a través del diagrama:

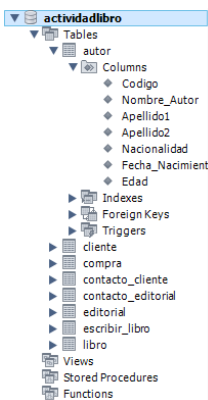


4) Código

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

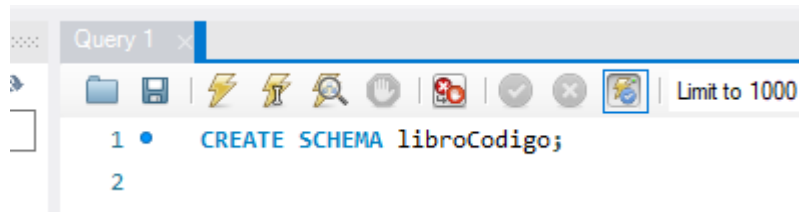
```
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Editorial` (  
  `Nombre` VARCHAR(45) NOT NULL,  
  `Ciudad` VARCHAR(45) NOT NULL,  
  `Complemento` VARCHAR(100) NOT NULL,  
  PRIMARY KEY (`Nombre`))  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Autor` (  
  `Codigo` VARCHAR(20) NOT NULL,  
  `Nombre_Autor` VARCHAR(45) NOT NULL,  
  `Apellido1` VARCHAR(45) NOT NULL,  
  `Apellido2` VARCHAR(45) NOT NULL,  
  `Nacionalidad` VARCHAR(45) NOT NULL,  
  `Fecha_Nacimiento` DATETIME NULL,  
  `Edad` INT NOT NULL,  
  PRIMARY KEY (`Codigo`))  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Escribir_Libro` (  
  `ISBN_Libro` VARCHAR(20) NOT NULL,  
  `Codigo_Autor` VARCHAR(20) NOT NULL,  
  `Fecha_Libro` DATETIME NOT NULL,  
  PRIMARY KEY (`ISBN_Libro`, `Codigo_Autor`),  
  INDEX `Codigo_Autor_idx` (`Codigo_Autor` ASC) VISIBLE,  
  CONSTRAINT `ISBN_Libro`  
    FOREIGN KEY (`ISBN_Libro`) REFERENCES `ActividadLibro`.`Libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `Codigo_Autor`  
    FOREIGN KEY (`Codigo_Autor`) REFERENCES `ActividadLibro`.`Autor` (`Codigo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Libro` (  
  `ISBN` VARCHAR(20) NOT NULL,  
  `Nro_Paginas` INT NOT NULL,  
  `Fecha_Publicacion` DATETIME NULL,  
  `Titulo` VARCHAR(40) NOT NULL,  
  `Genero` VARCHAR(20) NOT NULL,  
  `Nombre_Editorial` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`ISBN`),  
  UNIQUE INDEX `ISBN_UNIQUE` (`ISBN` ASC) VISIBLE,  
  INDEX `Nombre_Editorial_idx` (`Nombre_Editorial` ASC) VISIBLE,  
  CONSTRAINT `Nombre_Editorial`  
    FOREIGN KEY (`Nombre_Editorial`) REFERENCES `ActividadLibro`.`Editorial` (`Nombre`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Cliente` (  
  `CC` VARCHAR(20) NOT NULL,  
  `Nombre_Cliente` VARCHAR(45) NOT NULL,  
  `Apellido1_Cliente` VARCHAR(45) NOT NULL,  
  `Apellido2_Cliente` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`CC`))  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Contacto_Cliente` (  
  `CC_Cliente` VARCHAR(20) NOT NULL,  
  `Telefono_Cliente` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`CC_Cliente`, `Telefono_Cliente`),  
  CONSTRAINT `CC_Cliente`  
    FOREIGN KEY (`CC_Cliente`) REFERENCES `ActividadLibro`.`Cliente` (`CC`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Contacto_Editorial` (  
  `Nombre_Edi_Contacto` VARCHAR(45) NOT NULL,  
  `Telefono_Editorial` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Nombre_Edi_Contacto`, `Telefono_Editorial`),  
  CONSTRAINT `Nombre_Edi_Contacto`  
    FOREIGN KEY (`Nombre_Edi_Contacto`) REFERENCES `ActividadLibro`.`Editorial` (`Nombre`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
CREATE TABLE IF NOT EXISTS `ActividadLibro`.`Compra` (  
  `ISBN_Libro` VARCHAR(20) NOT NULL,  
  `CC_Cliente` VARCHAR(20) NOT NULL,  
  `Fecha_Compra` DATETIME NULL,  
  PRIMARY KEY (`ISBN_Libro`, `CC_Cliente`),  
  INDEX `CC_Cliente_idx` (`CC_Cliente` ASC) VISIBLE,  
  CONSTRAINT `ISBN_Libro_Compra`  
    FOREIGN KEY (`ISBN_Libro`) REFERENCES `ActividadLibro`.`Libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `CC_Cliente_Compra`  
    FOREIGN KEY (`CC_Cliente`) REFERENCES `ActividadLibro`.`Cliente` (`CC`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

Resultado:



Código manual y registros:

Empiezo creando el schema mediante el código:



Con el schema creado procedo a generar las tablas y sus registros en el siguiente orden, para no tener complicaciones con las llaves foráneas:

Tabla de Editorial y sus contactos:

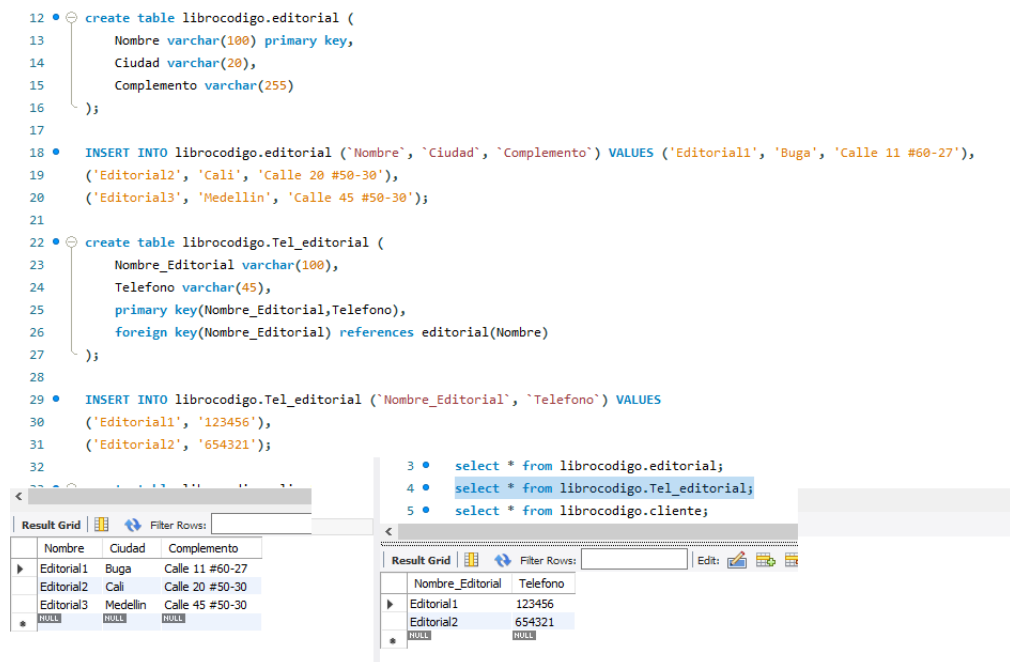


Tabla cliente y sus contactos:

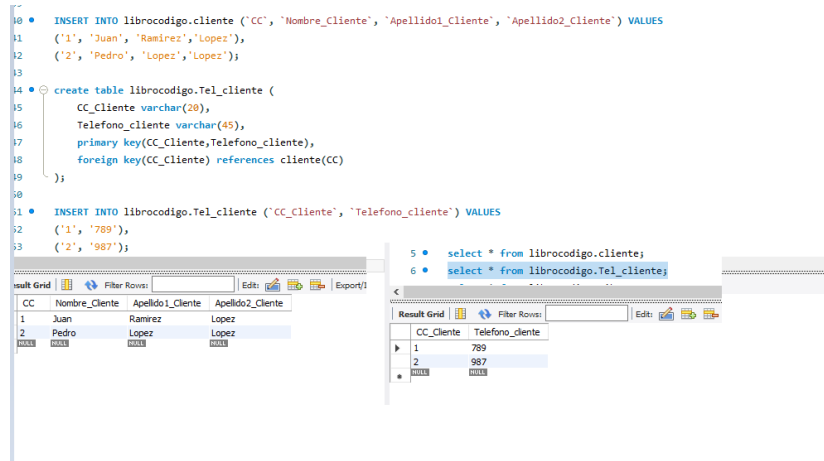


Tabla libro y compra

```
55 • create table librocodigo.Libro (  
56     ISBN varchar(20) primary key,  
57     Nro_Paginas int,  
58     Fecha_publicacion varchar(40),  
59     Titulo varchar(40),  
60     Genero varchar(40),  
61     Nombre_edit varchar(200),  
62     foreign key(Nombre_edit) references editorial(Nombre)  
63 );  
64  
65 • INSERT INTO librocodigo.Libro ('ISBN', 'Nro_Paginas', 'Fecha_publicacion', 'Titulo', 'Genero', 'Nombre_edit') VALUES  
66 ('9695', 20, '12/01/96','Programacion','Academico','Editorial1'),  
67 ('9696', 200, '01/06/99','Cocina','Academico','Editorial2');  
68  
69 • create table librocodigo.Compra (  
70     ISBN_compra varchar(20),  
71     CC_compra varchar(20),  
72     Fecha_compra varchar(40),  
73     primary key(ISBN_compra,CC_compra),  
74     foreign key(ISBN_compra) references Libro(ISBN),  
75     foreign key(CC_compra) references cliente(CC)  
76 );  
77  
78 • INSERT INTO librocodigo.Compra ('ISBN_compra', 'CC_compra', 'Fecha_compra') VALUES  
79 ('9695', '1', '10/02/23'),  
80 ('9696', '2', '10/01/23');
```

Result Grid for librocodigo.Libro:

ISBN	Nro_Paginas	Fecha_publicacion	Titulo	Genero	Nombre_edit
9695	20	12/01/96	Programacion	Academico	Editorial1
9696	200	01/06/99	Cocina	Academico	Editorial2

Result Grid for librocodigo.Compra:

ISBN_compra	CC_compra	Fecha_compra
9695	1	10/02/23
9696	2	10/01/23

Al haber creado la clase libro, y ya tener la clase cliente, me permitió generar la clase compro y registrar dos de estas, en un orden diferente no habría servido ya que compra tiene dos llaves foráneas que apuntan a libro y cliente.

Por ultimo las tablas de Autor y EscribirLibro:

```
82 • create table librocodigo.Autor (  
83    Codigo varchar(20) primary key,  
84     Nombre_autor varchar(40),  
85     Apellido1_autor varchar(40),  
86     Apellido2_autor varchar(40),  
87     Nacionalidad varchar(40),  
88     Fecha_autor_nacimiento varchar(50),  
89     Edad int  
90 );  
91  
92 • INSERT INTO librocodigo.Autor ('Codigo', 'Nombre_autor', 'Apellido1_autor', 'Apellido2_autor', 'Nacionalidad', 'Fecha_autor_nacimiento', 'Edad') VALUES  
93 ('2020', 'Juan', 'Lopez', 'Vanegas', 'Colombiano', '05/09/90', 33),  
94 ('2121', 'Mario', 'Ramirez', 'Vanegas', 'Mexicano', '16/11/85', 38);  
95  
96 • create table librocodigo.EscribirLibro (  
97     ISBN_escribir varchar(20),  
98     Codigo_autor varchar(20),  
99     Fecha_libro varchar(40),  
100     primary key(ISBN_escribir,Codigo_autor),  
101     foreign key(ISBN_escribir) references Libro(ISBN),  
102     foreign key(Codigo_autor) references autor(Codigo)  
103 );  
104  
105 • INSERT INTO librocodigo.EscribirLibro ('ISBN_escribir', 'Codigo_autor', 'Fecha_libro') VALUES  
106 ('9695', '2020', '10/02/21'),
```

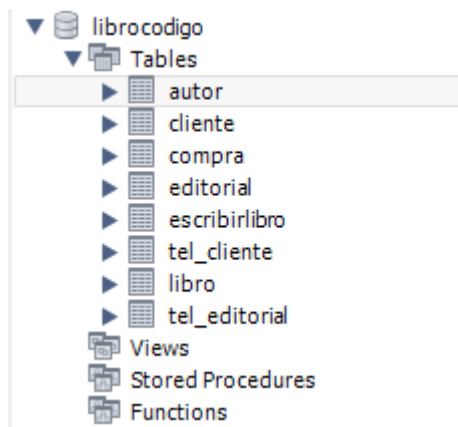
Result Grid for librocodigo.Autor:

Codigo	Nombre_autor	Apellido1_autor	Apellido2_autor	Nacionalidad	Fecha_autor_nacimiento	Edad
2020	Juan	Lopez	Vanegas	Colombiano	05/09/90	33
2121	Mario	Ramirez	Vanegas	Mexicano	16/11/85	38

Result Grid for librocodigo.EscribirLibro:

ISBN_escribir	Codigo_autor	Fecha_libro
9695	2020	10/02/21
9696	2121	10/01/22

En EscribirLibro pasa lo mismo que con la tabla comprar, esta entidad depende de otras dos (Autor y Libro), por eso debía crearlas antes.



Se puede observar que el resultado es exactamente igual que al haberlo creado con el diagrama.