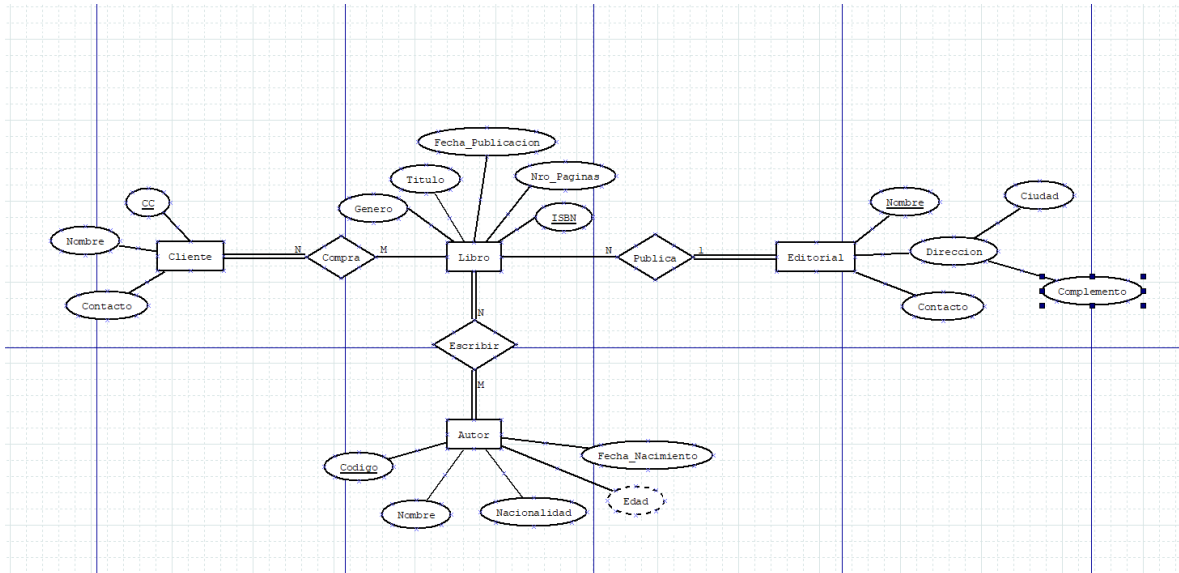
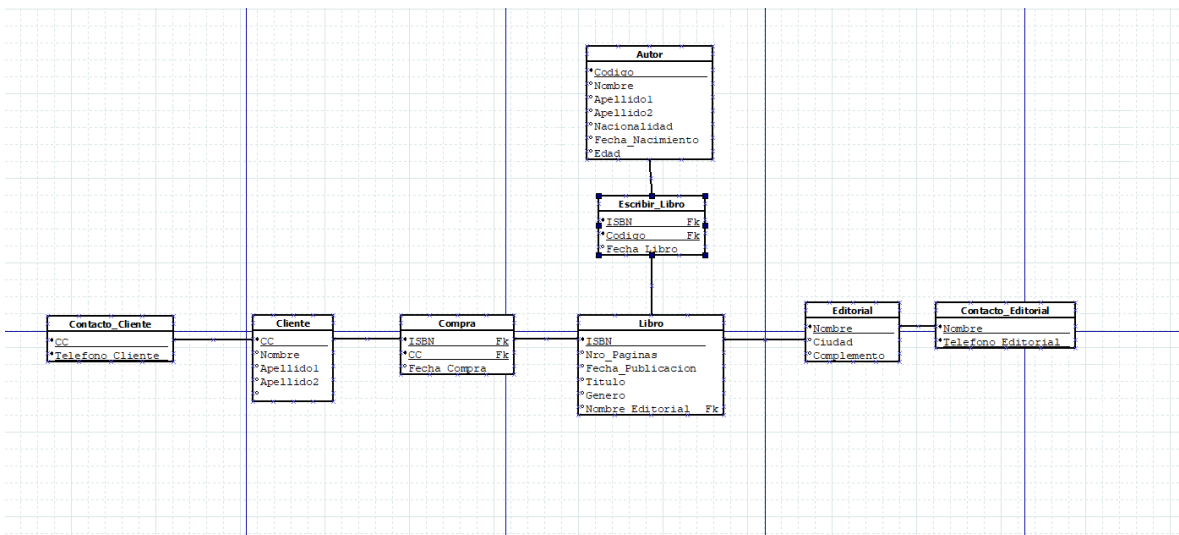


1) Modelo Entidad-Relación (Actividad libro)



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.

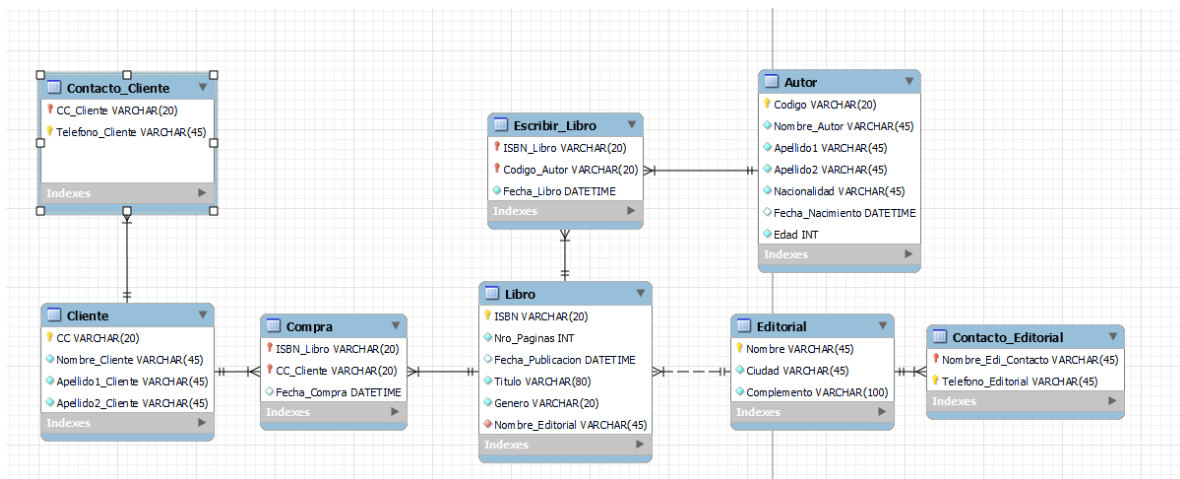
Atributos multivaluados:

- 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.

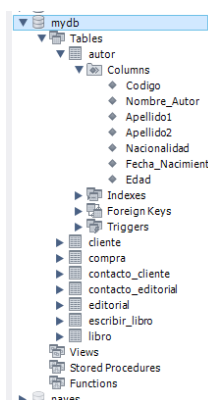
Atributos compuestos:

- 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.

3) Diagrama en Workbench SQL



Schema creado a través del diagrama:

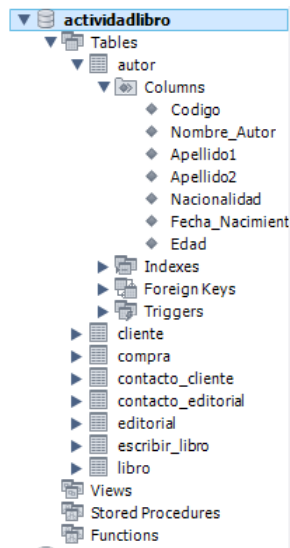


4) Código

```
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(80) 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:



```
Query 1 x orden de servicio autor - Table
Limit to 1000 rows

1 CREATE TABLE `libro1`.`autor` (
2     `Codigo` VARCHAR(20) NOT NULL,
3     `Nombre` VARCHAR(45) NULL,
4     `Apellido1` VARCHAR(45) NULL,
5     `Apellido2` VARCHAR(45) NULL,
6     `Nacionalidad` VARCHAR(45) NULL,
7     `Fecha_Nacimiento` DATETIME NULL,
8     `Edad` INT NULL,
9     PRIMARY KEY (`Codigo`));
10
11 CREATE TABLE `libro1`.`libro` (
12     `ISBN` VARCHAR(20) NOT NULL,
13     `Nro_Paginas` VARCHAR(45) NULL,
14     `Titulo` VARCHAR(45) NULL,
15     `Genero` VARCHAR(45) NULL,
16     `Fecha_Publicacion` DATETIME NULL,
17     `Edad` INT NULL,
18     PRIMARY KEY (`ISBN`));
19
20 CREATE TABLE `libro1`.`cliente` (
21     `CC` VARCHAR(20) NOT NULL,
22     `Nombre` VARCHAR(45) NULL,
23     `Apellido1` VARCHAR(45) NULL,
24     `Apellido2` VARCHAR(45) NULL,
25     PRIMARY KEY (`CC`));
26
27 CREATE TABLE `libro1`.`editorial` (
28     `Nombre` VARCHAR(20) NOT NULL,
29     `Ciudad` VARCHAR(45) NULL,
30     `Complemento` VARCHAR(100) NULL,
31     PRIMARY KEY (`Nombre`));
32
33
```

De esta manera se crearon todas las tablas, y las relaciones entre ellas se hicieron mediante el programa como se explico en clase.

