

Primera Actividad

1. Elaborar el diagrama E-R del Ejercicio de la librería

Se tienen las siguientes entidades con sus respectivos atributos

Editorial: Nombre, Dirección (Compuesto).

Libro: ISBN, Título, Número de páginas.

Cliente: ID, Nombre, Teléfono (Multivalor).

Autor: Código de autor, Nacionalidad, Fecha de nacimiento, Edad (Derivado).

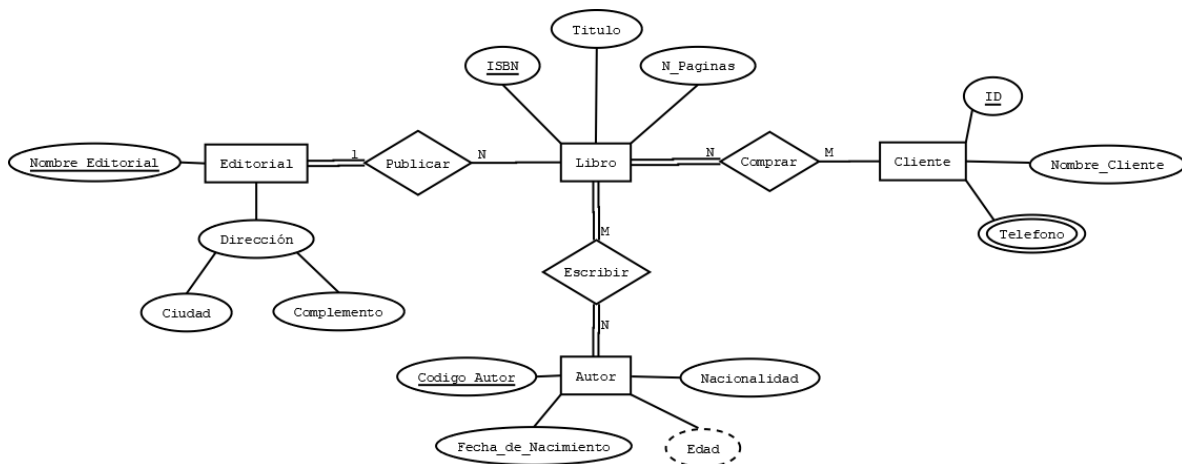
Relaciones

Publicar: Editorial 1:N Libro. Es decir, una editorial puede publicar muchos libros y un libro solo puede ser publicado por una editorial.

Comprar: Libros N:M Cliente. Es decir, un cliente puede comprar muchos libros y un libro puede ser comprado por muchos clientes.

Escribir: Libro N:M Autor. Es decir, un autor puede escribir muchos libros y un libro puede ser escrito por muchos autores.

A partir de esto se crea el siguiente diagrama:



2. Elaborar el modelo relación del punto 1.

Se transforman las entidades en tablas

Libro		Editorial		Cliente		Autor	
PK	<u>ISBN</u>	PK	<u>nombre</u>	PK	<u>id</u>	PK	<u>Codigo_autor</u>
	titulo		ciudad		nombre_cliente		fecha de nacimiento
	n_paginas		complemento		telefono		edad
							nacionalidad

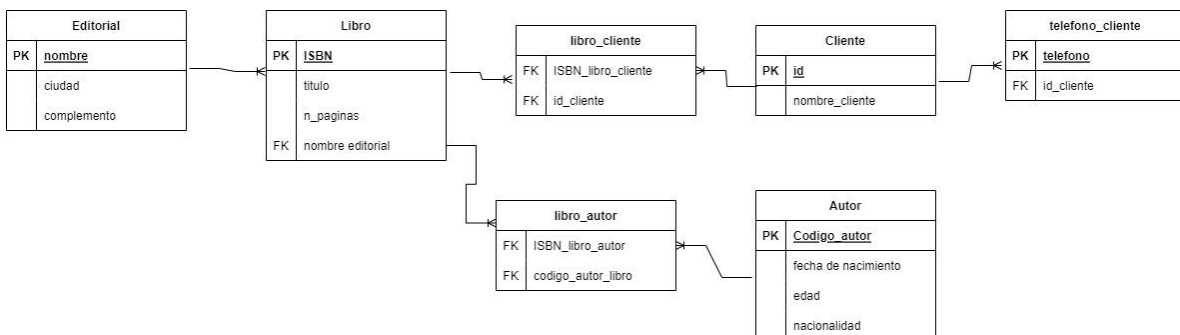
Se actualiza la tabla libro que tiene cardinalidad 1:N con editorial

Libro	
PK	<u>ISBN</u>
	titulo
	n_paginas
FK	nombre editorial

Se crean las tablas para las relaciones con cardinalidades muchos a muchos.

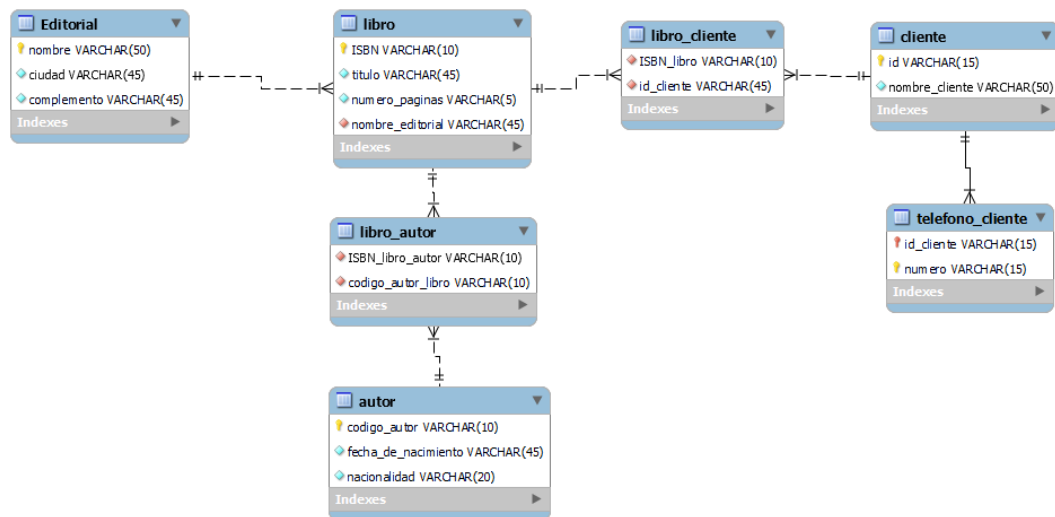
libro_cliente		libro_autor	
FK	ISBN_libro_cliente	FK	ISBN_libro_autor
FK	id_cliente	FK	codigo_autor_libro

Finalmente, el diagrama de modelo relacional queda de esta forma además se crea una tabla para el teléfono del cliente para cumplir con la primera forma normal



El diagrama estaría cumpliendo con las tres formas normales.

3. Diagramar en workbench el punto 2.



4. Escribir con sentencias de SQL la representación del modelo relacional del punto 2.

Usamos la sentencia CREATE DATABASE + el nombre de la base de datos (en este caso librería) para crear la base de datos. Adicional a eso, usamos la sentencia USE para seleccionar una base de datos específica. en este caso sería la base de datos librería.

```
1 • CREATE DATABASE libreria;  
2  
3 • USE libreria;
```

Para crear una tabla usamos la sentencia CREATE TABLE + el nombre de la tabla y dentro de paréntesis van a ir el nombre de los atributos con el tipo de dato, además que se le debe colocar PRIMARY KEY al dato que sea la llave primaria y NOT NULL a los datos que deban ser obligatoriamente llenados.

```
5 • CREATE TABLE editorial (  
6     nombre VARCHAR(50) NOT NULL PRIMARY KEY,  
7     ciudad VARCHAR(45) NOT NULL,  
8     complemento VARCHAR(45) NOT NULL  
9 );
```

Para las tablas que tengan relaciones de **uno a muchos**, en la tabla que esta del lado de **muchos** se agrega un nuevo atributo con el nombre de la llave primaria de la tabla que esta de lado de **uno** y se agrega la sentencia FOREIGN KEY con el nombre de la llave foránea y REFERENCES con el nombre de la tabla donde vienen esa llave y entre paréntesis el nombre de la llave principal.

```
30 CREATE TABLE libro (  
    ISBN VARCHAR(10) NOT NULL PRIMARY KEY,  
    titulo VARCHAR(45) NOT NULL,  
    numero_paginas VARCHAR(5) NOT NULL,  
    nombre_editorial VARCHAR(45) NOT NULL,  
    FOREIGN KEY (nombre_editorial) REFERENCES editorial (nombre)  
);
```

Para las tablas que tienen relaciones **muchos a muchos** se crea una nueva tabla con el nombre de las dos tablas, y se crean dos atributos con el nombre de las llaves primarias de cada tabla. además, se asignan las llaves foráneas referenciando las tablas que se relacionan y las llaves primarias de cada tabla.

```
24 • 31 CREATE TABLE libro_cliente (  
25     ISBN_libro VARCHAR(10) NOT NULL,  
26     id_cliente VARCHAR(45) NOT NULL,  
27     FOREIGN KEY (ISBN_libro) REFERENCES libro (ISBN),  
28     FOREIGN KEY (id_cliente) REFERENCES cliente (id)  
29 );
```

Para las tablas que tienen atributos multivalor (por ejemplo el teléfono del cliente) se crea una nueva tabla con dos atributos que serán llaves principales, uno que va ser la llave principal de la tabla cliente y otro va a ser el numero del teléfono. Y se coloca como llave foránea el atributo del id del cliente.

```
31 • 32 CREATE TABLE telefono_cliente (  
32     id_cliente VARCHAR(15) NOT NULL,  
33     numero VARCHAR(15) NOT NULL,  
34     PRIMARY KEY (id_cliente, numero),  
35     FOREIGN KEY (id_cliente) REFERENCES cliente (id)  
36 );
```

Finalmente, de esta forma quedaría todo el script:

```
1  ●   CREATE DATABASE libreria;
2
3  ●   USE libreria;
4
5  ●   ○ CREATE TABLE editorial (
6      nombre VARCHAR(50) NOT NULL PRIMARY KEY,
7      ciudad VARCHAR(45) NOT NULL,
8      complemento VARCHAR(45) NOT NULL
9  );
10
11  ●   ○ CREATE TABLE libro (
12      ISBN VARCHAR(10) NOT NULL PRIMARY KEY,
13      titulo VARCHAR(45) NOT NULL,
14      numero_paginas VARCHAR(5) NOT NULL,
15      nombre_editorial VARCHAR(45) NOT NULL,
16      FOREIGN KEY (nombre_editorial) REFERENCES editorial (nombre)
17  );
18
19  ●   ○ CREATE TABLE cliente (
20      id VARCHAR(15) NOT NULL PRIMARY KEY,
21      nombre_cliente VARCHAR(50) NOT NULL
22  );
23
24  ●   ○ CREATE TABLE libro_cliente (
25      ISBN_libro VARCHAR(10) NOT NULL,
26      id_cliente VARCHAR(45) NOT NULL,
27      FOREIGN KEY (ISBN_libro) REFERENCES libro (ISBN),
28      FOREIGN KEY (id_cliente) REFERENCES cliente (id)
29  );
30
31  ●   ○ CREATE TABLE telefono_cliente (
32      id_cliente VARCHAR(15) NOT NULL,
33      numero VARCHAR(15) NOT NULL,
34      PRIMARY KEY (id_cliente, numero),
35      FOREIGN KEY (id_cliente) REFERENCES cliente (id)
36  );
37
```

```

38 ● CREATE TABLE autor (
39     codigo_autor VARCHAR(10) NOT NULL PRIMARY KEY,
40     fecha_de_nacimiento VARCHAR(45) NOT NULL,
41     nacionalidad VARCHAR(20) NOT NULL
42 );
43
44 ● CREATE TABLE libro_autor (
45     ISBN_libro_autor VARCHAR(10) NOT NULL,
46     codigo_autor_libro VARCHAR(10) NOT NULL,
47     FOREIGN KEY (ISBN_libro_autor) REFERENCES libro (ISBN),
48     FOREIGN KEY (codigo_autor_libro) REFERENCES autor (codigo_autor)
49 );
50

```

El script estará disponible dentro del repositorio.

5. Poblar los datos usando instrucciones SQL

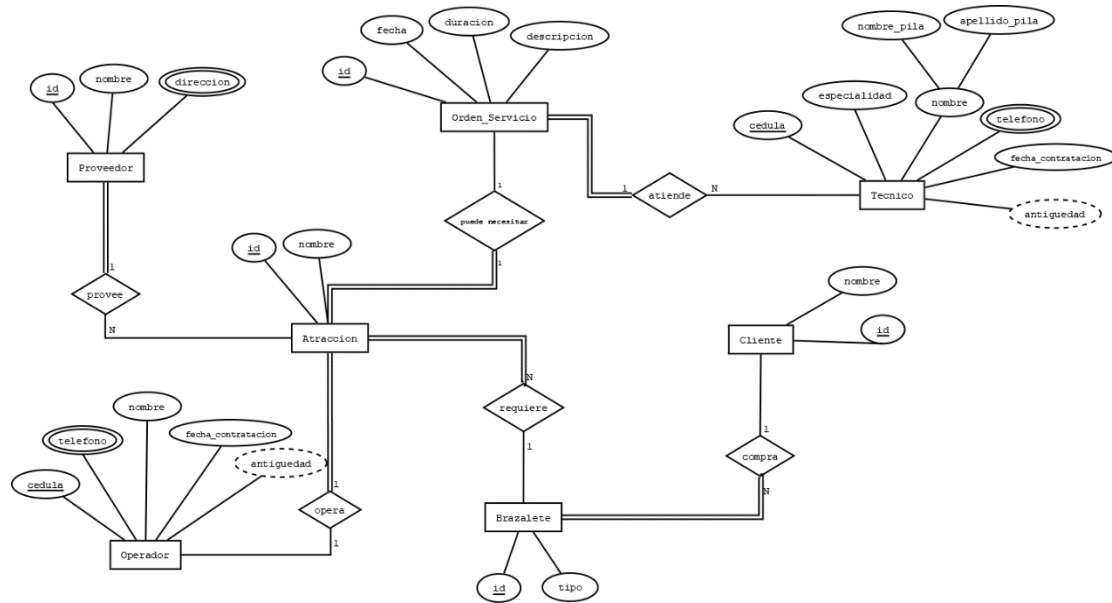
```

51 ● INSERT INTO editorial (nombre, ciudad, complemento)
52 VALUES ('Planeta', 'Bogotá', 'Cl. 73 #7-60'), ('Critica', 'Barcelona', 'Avenida Diagonal, 662');
53
54 ● SELECT *
55 FROM editorial;
56
57 ● INSERT INTO autor (codigo_autor, fecha_de_nacimiento, nacionalidad)
58 VALUES ('100', '9 de Nov 1934', 'estadounidense'), ('200', '8 de Ene 1942', 'Ingles');
59
60 ● SELECT *
61 FROM autor;
62
63 ● INSERT INTO libro (ISBN, titulo, numero_paginas, nombre_editorial)
64 VALUES ('1111', 'El mundo y sus demonios', '489', 'Planeta'), ('2222', 'Breve historia del tiempo', '258', 'Critica');
65
66 ● SELECT *
67 FROM libro;
68
69 ● INSERT INTO cliente (id, nombre_cliente)
70 VALUES ('1090494364', 'Jesús Miguel Molina Mendoza'), ('1092584352', 'Maria Fernanda Vera Omaña');
71
72 ● SELECT *
73 FROM cliente;
74
75 ● INSERT INTO telefono_cliente (id_cliente, numero)
76 VALUES ('1090494364', '3125848043'), ('1092584352', '3205482157');
77
78 ● SELECT *
79 FROM telefono_cliente;
80
81 ● INSERT INTO libro_cliente (ISBN_libro, id_cliente)
82 VALUES ('1111', '1090494364'), ('2222', '1092584352');
83
84 ● SELECT *
85 FROM libro_cliente;
86
87 ● INSERT INTO libro_autor (ISBN_libro_autor, codigo_autor_libro)
88 VALUES ('1111', '100'), ('2222', '200');
89
90 ● SELECT *
91 FROM libro_autor;

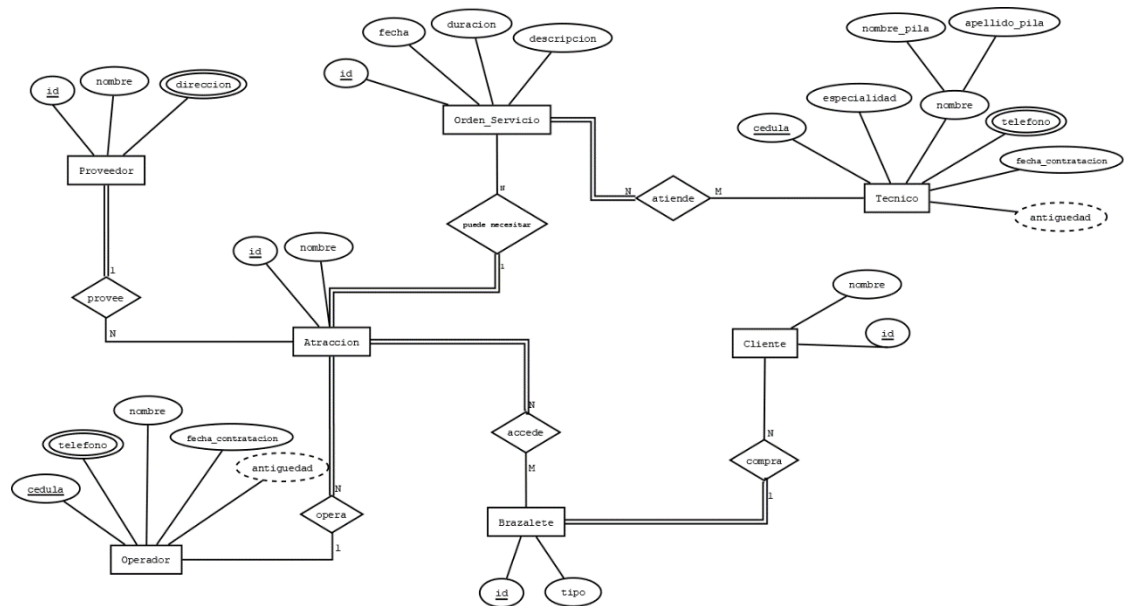
```

Segunda actividad

1. Se tiene el siguiente diagrama del compañero Juan David



2. Después de realizar algunas modificaciones a las cardinalidades queda de la siguiente forma.



Los cambios fueron:

- Atracción 1:1 Orden de servicio - Atracción N:1 Orden de servicio. Porque una atracción puede necesitar muchas ordenes de servicio
- Orden de servicio 1:N Técnico - Orden de servicio N:M Técnico. Porque una orden de servicio puede requerir varios técnicos y un técnico puede atender muchas ordenes de servicio
- Cliente 1:N Brazaletes - Cliente N:1 Brazaletes. Porque un brazalete puede ser llevado por muchos clientes, pero un cliente solo puede llevar un brazalete
- Atracción N:1 Brazaletes - Atracción N:M Brazaletes. Porque un brazalete puede acceder a muchas atracciones y una atracción puede recibir muchos brazaletes.

3. Diseñar nuevamente el Modelo relacional en caso de haya sufrido una modificación.

Se transforman las entidades en tablas

Proveedor	
PK	<u>id_proveedor</u>
	nombre_proveedor
	dirección

Atracción	
PK	<u>id_atraccion</u>
	nombre atracción

Orden_servicio	
PK	<u>id_orden</u>
	fecha
	duración
	descripción

Tecnico	
PK	<u>cedula técnico</u>
	nombre_tecnico
	especialidad
	fecha_contratación
	teléfono_técnico

Operador	
PK	<u>cedula_operador</u>
	nombre_operdor
	fecha_contratacion
	telefono

Brazaletes	
PK	<u>id_brazaletes</u>
	tipo

Cliente	
PK	<u>id_cliente</u>
	nombre

Se actualizan las tablas con cardinalidades 1:N

Proveedor	
PK	<u>id_proveedor</u>
	nombre_proveedor
	dirección

Atracción	
PK	<u>id_atraccion</u>
	nombre atracción

Orden_servicio	
PK	<u>id_orden</u>
	fecha
	duración
	descripción

Tecnico	
PK	<u>cedula técnico</u>
	nombre_tecnico
	especialidad
	fecha_contratación
	teléfono_técnico

Operador	
PK	<u>cedula_operador</u>
	nombre_operdor
	fecha_contratacion
	telefono

Brazalete	
PK	<u>id_brazalete</u>
	tipo

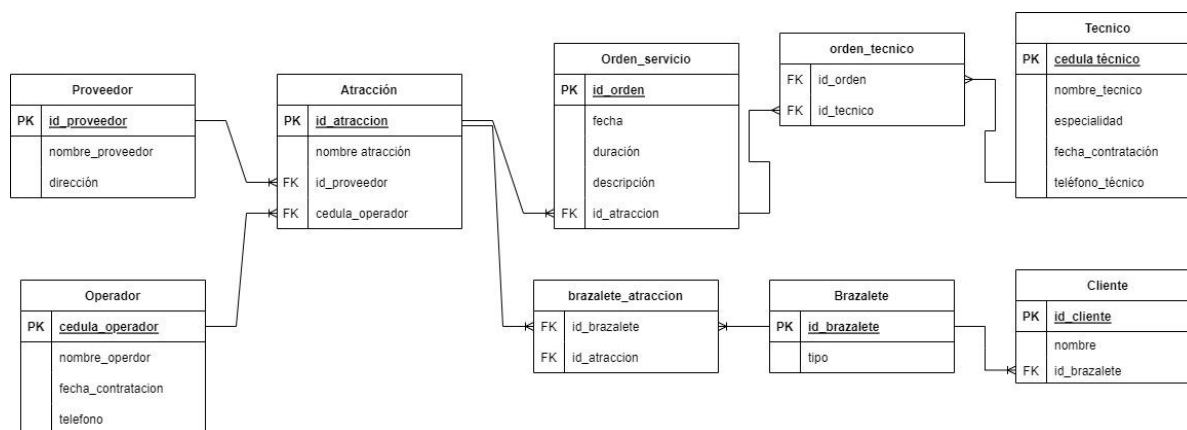
Cliente	
PK	<u>id_cliente</u>
	nombre

Se crean las tablas para las relaciones con cardinalidades muchos a muchos.

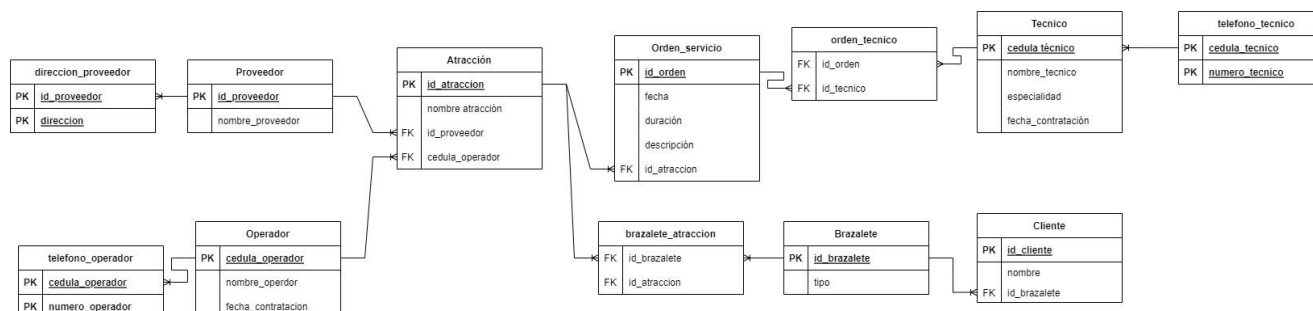
orden_tecnico	
FK	id_orden
FK	id_tecnico

brazalete_atraccion	
FK	id_brazalete
FK	id_atraccion

El diagrama quedaría de la siguiente forma.

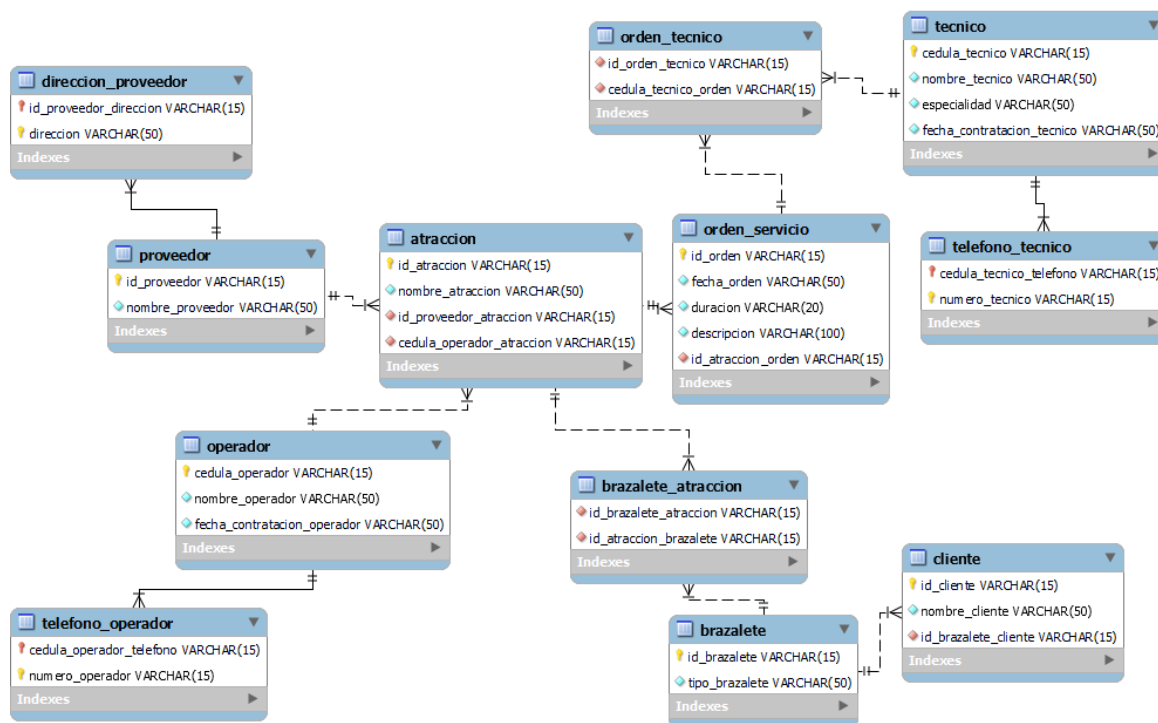


Después de normalizar el diagrama quedaría de la siguiente forma.



de esta forma la tabla cumple todas las formas normales, ya que dejaron de existir atributos multivalor, es decir ahora todos son atómicos, todas las tablas poseen una llave primaria única, cada tabla depende de su llave primaria y no existe una dependencia transitiva entre atributos que no son clave.

4. Diagramar en workbench el punto.



5. Escribir con sentencias de SQL la representación del modelo relacional

Usamos la sentencia CREATE DATABASE + el nombre de la base de datos (en este caso parqueNorte) para crear la base de datos. Adicional a eso, usamos la sentencia USE para seleccionar una base de datos específica. en este caso sería la base de datos parqueNorte.

```
1 • CREATE DATABASE parqueNorte;
2
3 • USE parqueNorte;
```

Para crear una tabla usamos la sentencia CREATE TABLE + el nombre de la tabla y dentro de paréntesis van a ir el nombre de los atributos con el tipo de dato, además que se le debe colocar PRIMARY KEY al dato que sea la llave primaria y NOT NULL a los datos que deban ser obligatoriamente llenados.

```
6 • CREATE TABLE tecnico (
7     cedula_tecnico VARCHAR(15) NOT NULL PRIMARY KEY,
8     nombre_tecnico VARCHAR(50) NOT NULL,
9     especialidad VARCHAR(50) NOT NULL,
10    fecha_contratacion_tecnico VARCHAR(50) NOT NULL
11 );
```

Para las tablas que tengan relaciones de **uno a muchos**, en la tabla que esta del lado de **muchos** se agrega un nuevo atributo con el nombre de la llave primaria de la tabla que esta de lado de **uno** y se agrega la sentencia FOREIGN KEY con el nombre de la llave foránea y REFERENCES con el nombre de la tabla donde vienen esa llave y entre paréntesis el nombre de la llave principal.

```
66 • CREATE TABLE cliente (  
67     id_cliente VARCHAR(15) NOT NULL PRIMARY KEY,  
68     nombre_cliente VARCHAR(50) NOT NULL,  
69     id_brazalete_cliente VARCHAR(15) NOT NULL,  
70     FOREIGN KEY (id_brazalete_cliente) REFERENCES brazalete (id_brazalete)  
71 );
```

Para las tablas que tienen relaciones **muchos a muchos** se crea una nueva tabla con el nombre de las dos tablas, y se crean dos atributos con el nombre de las llaves primarias de cada tabla. además, se asignan las llaves foráneas referenciando las tablas que se relacionan y las llaves primarias de cada tabla.

```
54 • CREATE TABLE brazalete_atraccion (  
55     id_brazalete_atraccion VARCHAR(15) NOT NULL,  
56     id_atraccion_brazalete VARCHAR(15) NOT NULL,  
57     FOREIGN KEY (id_brazalete_atraccion) REFERENCES brazalete (id_brazalete),  
58     FOREIGN KEY (id_atraccion_brazalete) REFERENCES ATRACCION (id_atraccion)  
59 );
```

Para las tablas que tienen atributos multivalor (por ejemplo el teléfono del tecnico) se crea una nueva tabla con dos atributos que serán llaves principales, uno que va ser la llave principal de la tabla técnico y otro va a ser el numero del técnico. Y se coloca como llave foránea el atributo del id del técnico.

```
13 • CREATE TABLE telefono_tecnico (  
14     cedula_tecnico_telefono VARCHAR(15) NOT NULL,  
15     numero_tecnico VARCHAR(15) NOT NULL,  
16     PRIMARY KEY (cedula_tecnico_telefono, numero_tecnico),  
17     FOREIGN KEY (cedula_tecnico_telefono) REFERENCES tecnico (cedula_tecnico)  
18 );
```

Finalmente, de esta forma quedaría todo el script:

```
1  ● CREATE DATABASE parqueNorte;
2
3  ● USE parqueNorte;
4
5
6  ● ○ CREATE TABLE tecnico (
7      cedula_tecnico VARCHAR(15) NOT NULL PRIMARY KEY,
8      nombre_tecnico VARCHAR(50) NOT NULL,
9      especialidad VARCHAR(50) NOT NULL,
10     fecha_contratacion_tecnico VARCHAR(50) NOT NULL
11 );
12
13 ● ○ CREATE TABLE telefono_tecnico (
14     cedula_tecnico_telefono VARCHAR(15) NOT NULL,
15     numero_tecnico VARCHAR(15) NOT NULL,
16     PRIMARY KEY (cedula_tecnico_telefono, numero_tecnico),
17     FOREIGN KEY (cedula_tecnico_telefono) REFERENCES tecnico (cedula_tecnico)
18 );
19
20 ● ○ CREATE TABLE operador (
21     cedula_operador VARCHAR(15) NOT NULL PRIMARY KEY,
22     nombre_operador VARCHAR(50) NOT NULL,
23     fecha_contratacion_operador VARCHAR(50) NOT NULL
24 );
25
26 ● ○ CREATE TABLE telefono_operador (
27     cedula_operador_telefono VARCHAR(15) NOT NULL,
28     numero_operador VARCHAR(15) NOT NULL,
29     PRIMARY KEY (cedula_operador_telefono, numero_operador),
30     FOREIGN KEY (cedula_operador_telefono) REFERENCES operador (cedula_operador)
31 );
32
33 ● ○ CREATE TABLE proveedor (
34     id_proveedor VARCHAR(15) NOT NULL PRIMARY KEY,
35     nombre_proveedor VARCHAR(50) NOT NULL
36 );
37
38 ● ○ CREATE TABLE direccion_proveedor (
39     id_proveedor_direccion VARCHAR(15) NOT NULL,
40     direccion VARCHAR(50) NOT NULL,
41     PRIMARY KEY (id_proveedor_direccion, direccion),
42     FOREIGN KEY (id_proveedor_direccion) REFERENCES proveedor (id_proveedor)
43 );
```

```

44
45 ● ○ CREATE TABLE atraccion (
46     id_atraccion VARCHAR(15) NOT NULL PRIMARY KEY,
47     nombre_atraccion VARCHAR(50) NOT NULL,
48     id_proveedor_atraccion VARCHAR(15) NOT NULL,
49     cedula_operador_atraccion VARCHAR(15) NOT NULL,
50     FOREIGN KEY (id_proveedor_atraccion) REFERENCES proveedor (id_proveedor),
51     FOREIGN KEY (cedula_operador_atraccion) REFERENCES operador (cedula_operador)
52 );
53
54 ● ○ CREATE TABLE brazalete_atraccion (
55     id_brazalete_atraccion VARCHAR(15) NOT NULL,
56     id_atraccion_brazalete VARCHAR(15) NOT NULL,
57     FOREIGN KEY (id_brazalete_atraccion) REFERENCES brazalete (id_brazalete),
58     FOREIGN KEY (id_atraccion_brazalete) REFERENCES ATRACCION (id_atraccion)
59 );
60
61 ● ○ CREATE TABLE brazalete (
62     id_brazalete VARCHAR(15) NOT NULL PRIMARY KEY,
63     tipo_brazalete VARCHAR(50) NOT NULL
64 );
65
66 ● ○ CREATE TABLE cliente (
67     id_cliente VARCHAR(15) NOT NULL PRIMARY KEY,
68     nombre_cliente VARCHAR(50) NOT NULL,
69     id_brazalete_cliente VARCHAR(15) NOT NULL,
70     FOREIGN KEY (id_brazalete_cliente) REFERENCES brazalete (id_brazalete)
71 );
72
73 ● ○ CREATE TABLE orden_servicio (
74     id_orden VARCHAR(15) NOT NULL PRIMARY KEY,
75     fecha_orden VARCHAR(50) NOT NULL,
76     duracion VARCHAR(20) NOT NULL,
77     descripcion VARCHAR(100) NOT NULL,
78     id_atraccion_orden VARCHAR(15) NOT NULL,
79     FOREIGN KEY (id_atraccion_orden) REFERENCES atraccion (id_atraccion)
80
81
82 ● ○ CREATE TABLE orden_tecnico (
83     id_orden_tecnico VARCHAR(15) NOT NULL,
84     cedula_tecnico_orden VARCHAR(15) NOT NULL,
85     FOREIGN KEY (id_orden_tecnico) REFERENCES orden_servicio (id_orden),
86     FOREIGN KEY (cedula_tecnico_orden) REFERENCES tecnico (cedula_tecnico)
87 );

```

El script estará disponible dentro del repositorio.

6. Poblar los datos usando instrucciones SQL

```
89 • INSERT INTO tecnico (cedula_tecnico, nombre_tecnico, especialidad, fecha_contratacion_tecnico)
90 VALUES ('1111111111', 'alberto mendoza', 'electrico', '02 de diciembre del 2016'), ('2222222222', 'diego garcia', 'mecanico', '22 de octubre del 2018').
91
92 • INSERT INTO telefono_tecnico (cedula_tecnico_telefono, numero_tecnico)
93 VALUES ('1111111111', '3201234567'), ('2222222222', '3151234567');
94
95 • INSERT INTO operador (cedula_operador, nombre_operador, fecha_contratacion_operador)
96 VALUES ('3333333333', 'pedro gonzales', '10 de enero del 2017'), ('4444444444', 'jairo parada', '15 de febrero del 2017');
97
98 • INSERT INTO telefono_operador (cedula_operador_telefono, numero_operador)
99 VALUES ('3333333333', '3181111111'), ('4444444444', '3112222222');
100
101 • INSERT INTO proveedor (id_proveedor, nombre_proveedor)
102 VALUES ('1111', 'Maquinas divertidas'), ('2222', 'Los mejores');
103
104 • INSERT INTO direccion_proveedor (id_proveedor_direccion, direccion)
105 VALUES ('1111', 'av 5 calle 2'), ('1111', 'av 3 calle 10'), ('2222', 'av 15 calle 20'), ('2222', 'av 30 calle 1');
106
107 • INSERT INTO atraccion (id_atraccion, nombre_atraccion, id_proveedor_atraccion, cedula_operador_atraccion)
108 VALUES ('000', 'invasión', '1111', '3333333333'), ('555', 'gas station', '2222', '4444444444');
109
110 • INSERT INTO brazalete (id_brazalete, tipo_brazalete)
111 VALUES ('123123', 'Extremo'), ('456456', 'Fusión'), ('789789', 'Aventura');
112
113 • INSERT INTO cliente (id_cliente, nombre_cliente, id_brazalete_cliente)
114 VALUES ('963', 'camilo sanchez', '123123'), ('741', 'oscar prada', '456456');
115
116 • INSERT INTO orden_servicio (id_orden, fecha_orden, duracion, descripcion, id_atraccion_orden)
117 VALUES ('999', '10 de abril', '2 dias', 'falla electrica', '000'), ('888', '22 de julio', '5 dias', 'falla mecanica', '555')
```