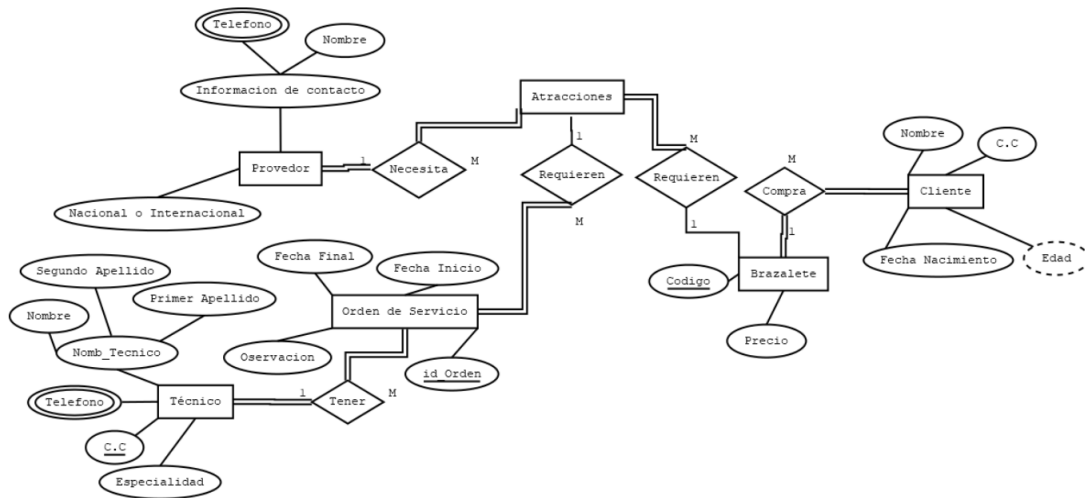


Modelo Entidad-Relacion de Nervardo



Revisando el modelo entidad de mi compañero veo que la estructura está bien, solo cambiaria unos verbos de las relaciones, algunas cardinalidades, agregaría algunos atributos y entidades.

Entidades y atributos:

Operador: Agrego la entidad Operador con sus respectivos atributos y la relacionaría con atracciones, ya que el operador es el que maneja una atracción.

Proveedor: Agrego el Id al proveedor para que tenga llave primaria.

Atracciones: Le agrego los atributos IdAtraccion y nombre.

Técnico: Especialidad la vuelvo un atributo multivaluado.

Relaciones:

Creo la relación entre operador y atracción llamada "Manipula".

Cambio la relación entre proveedor y atracción de "Necesita" a "Vende".

Cambio la relación entre técnico y orden de servicio de "Tener" a "Lleva a cabo".

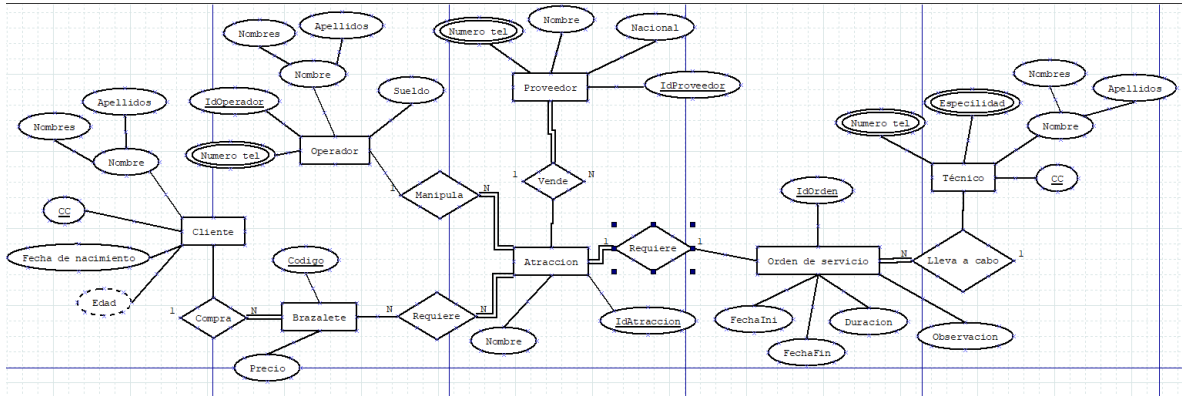
Cardinalidades y participaciones:

Arreglo la cardinalidad de clientes con brazaletes de N:1 a 1:N, ya que considero que es más lógico que un cliente pueda comprar varios brazaletes a que un brazalete sea comprado por varios clientes.

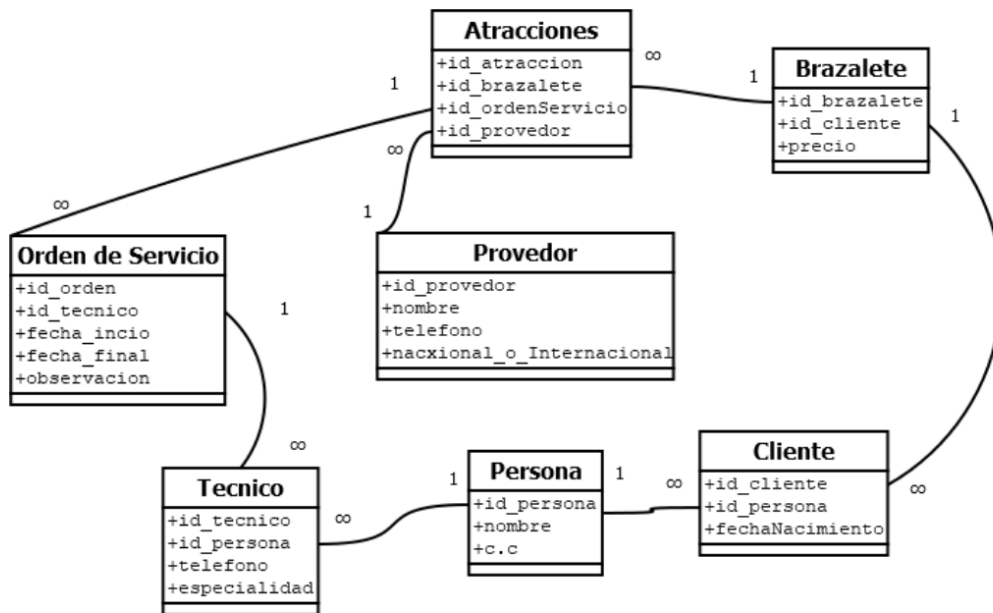
Arreglo la cardinalidad de brazalete con atracciones de N:M, ya que considero que un brazalete sirve para entrar a varias atracciones y a una atracción se puede entrar con distintos brazaletes.

Agrego la cardinalidad de proveedor con atracción 1:N, un proveedor puede proveer uno o varias atracciones, pero una atracción solo la puede proveer un proveedor.

Resultado al aplicar las correcciones:



Modelo Relacional de Nevardo:



El modelo relacional del compañero no se fija en sus Cardinalidades, por ende, no aplicar las reglas para relacionar las tablas, es decir, no crea llaves foráneas que conecten las tablas. También crea una nueva entidad “Persona” en la que trata de hacer una herencia, lo cual no es recomendable.

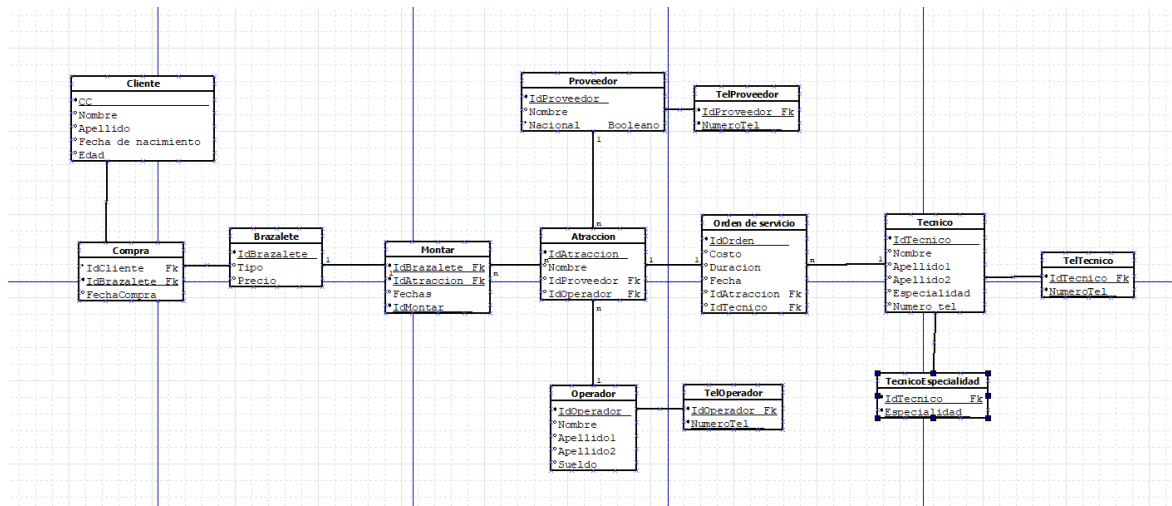
Para corregir el modelo relacional me voy a basar en el modelo entidad relación después de las correcciones que creí necesarias.

Se genera una nueva tabla “Montar” que resulta de la relación de N:M entre brazaletes y atracción.

Se genera una nueva tabla “Comprar” que resulta de la relación 0-1:1-N entre cliente y brazaletes.

Se generan nuevas tablas que provienen de los atributos multivaluados como los teléfonos y especialidades de los técnicos para evitar redundancia.

Resultado al aplicar las correcciones:



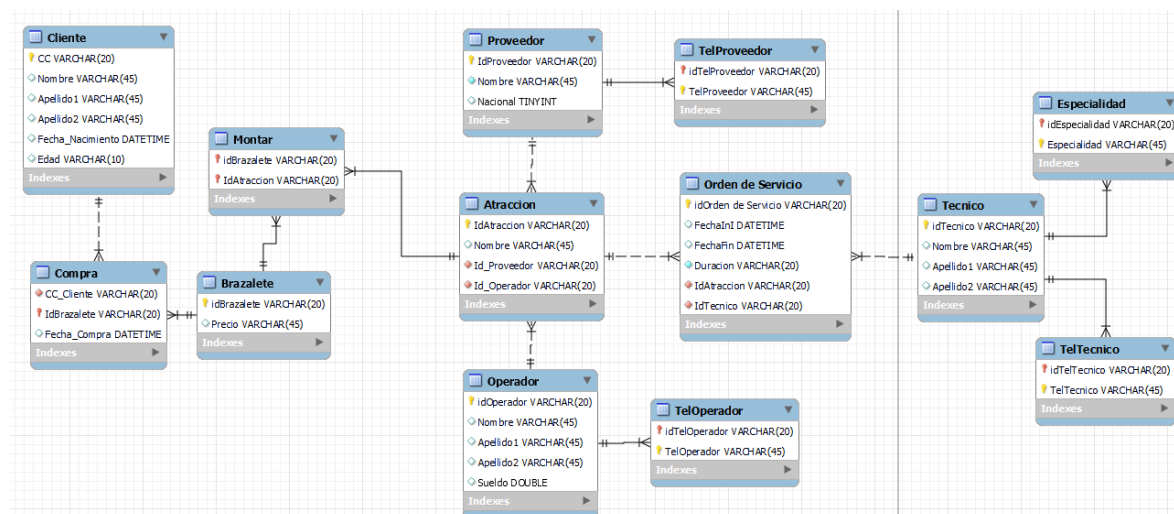
Normalización:

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 está 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 depender 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 está en tercera forma normal.

Diagrama creado en workbench:



Código generado:

```

1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7  --
8  -- Schema mydb
9  --
10
11  --
12  -- Schema mydb
13  --
14  CREATE SCHEMA IF NOT EXISTS 'mydb' DEFAULT CHARACTER SET utf8 ;
15  USE 'mydb' ;
16
17  --
18  -- Table 'mydb`.`Proveedor`
19  --
20  CREATE TABLE IF NOT EXISTS 'mydb`.`Proveedor` (
21    'IdProveedor' VARCHAR(20) NOT NULL,
22    'Nombre' VARCHAR(45) NOT NULL,
23    'Nacional' TINYINT NULL,
24    PRIMARY KEY ('IdProveedor'))
25  ENGINE = InnoDB;
26
27
28  --
29  -- Table 'mydb`.`Operador`
30  --
31  CREATE TABLE IF NOT EXISTS 'mydb`.`Operador` (
32    'IdOperador' VARCHAR(20) NOT NULL,
  
```

Schema generado:



Crear modelo de forma manual:

Creación del schema:

32

33 • `CREATE SCHEMA `parque` ;`

34

Creación de tablas y registros:

Proveedor y su tabla de contactos.

```
12 • create table parqueCodigo.Proveedor (  
13     IdProveedor varchar(20) primary key,  
14     Nombre varchar(20),  
15     Nacional boolean  
16 );  
17  
18 • INSERT INTO parqueCodigo.Proveedor ('IdProveedor', 'Nombre', 'Nacional') VALUES  
19     ('01', 'MontañasAS', 0),  
20     ('02', 'ParquesAS', 1),  
21     ('03', 'ExtremoSAS', 1);  
22  
23 • create table parqueCodigo.tel_Proveedor (  
24     IdProveedor varchar(20),  
25     Telefono varchar(45),  
26     primary key(IdProveedor, Telefono),  
27     foreign key(IdProveedor) references Proveedor(IdProveedor)  
28 );  
29  
30 • INSERT INTO parqueCodigo.tel_Proveedor ('IdProveedor', 'Telefono') VALUES  
31     ('01', '123456'),  
32     ('02', '654321');  
33  
34 • create table parqueCodigo.Tecnico (  
35     IdTecnico varchar(20) primary key,  
36     Nombre varchar(40),  
37     Ane111d1n1 varchar(40);  
38 );  
39  
40 • INSERT INTO parqueCodigo.Tecnico ('IdTecnico', 'Nombre', 'Ane111d1n1') VALUES  
41     ('01', 'Tecnico1', 'Ane111d1n1'),  
42     ('02', 'Tecnico2', 'Ane111d1n2'),  
43     ('03', 'Tecnico3', 'Ane111d1n3');  
44  
45 • select * from parqueCodigo.Proveedor;  
46  
47 • select * from parqueCodigo.tel_Proveedor;  
48  
49 • select * from parqueCodigo.Tecnico;
```

IdProveedor	Nombre	Nacional
01	MontañasAS	0
02	ParquesAS	1
03	ExtremoSAS	1

IdProveedor	Telefono
01	123456
02	654321

Técnico con sus respectivas tablas de contacto y especialidad

The screenshot displays the SQL Server Enterprise Manager interface with a query window open. The query contains SQL statements to create three tables: `parqueCodigo.Tecnico`, `parqueCodigo.tel_Tecnico`, and `parqueCodigo.especialidad_Tecnico`. Each table has a primary key and a foreign key relationship with the `Tecnico` table. The `Tecnico` table has columns `IdTecnico`, `Nombre`, `Apellido1`, and `Apellido2`. The `tel_Tecnico` table has columns `IdTecnico` and `Telefono`. The `especialidad_Tecnico` table has columns `IdTecnico` and `Especialidad`. The query also includes `INSERT INTO` statements for each table. Below the query window, three result grids are shown, each displaying the data inserted into the respective table.

```
34 • create table parqueCodigo.Tecnico (
35     IdTecnico varchar(20) primary key,
36     Nombre varchar(40),
37     Apellido1 varchar(40),
38     Apellido2 varchar(40)
39 );
40
41 • INSERT INTO parqueCodigo.Tecnico ('IdTecnico', 'Nombre', 'Apellido1','Apellido2') VALUES
42 ('11', 'Fabio','Lopez','Lopez'),
43 ('12', 'Carlos','Ramirez','');
44
45 • create table parqueCodigo.tel_Tecnico (
46     IdTecnico varchar(20),
47     Telefono varchar(45),
48     primary key(IdTecnico,Telefono),
49     foreign key(IdTecnico) references Tecnico(IdTecnico)
50 );
51
52 • INSERT INTO parqueCodigo.tel_Tecnico ('IdTecnico', 'Telefono') VALUES
53 ('11', '123456'),
54 ('12', '654321');
55
56 • create table parqueCodigo.especialidad_Tecnico (
57     IdTecnico varchar(20),
58     Especialidad varchar(45),
59     primary key(IdTecnico,Especialidad),
60     foreign key(IdTecnico) references Tecnico(IdTecnico)
61 );
62
63 • INSERT INTO parqueCodigo.especialidad_Tecnico ('IdTecnico', 'Especialidad') VALUES
64 ('11', 'Software'),
65 ('12', 'Programar'),
66 ('11', 'Maquinas');
```

Result Grid 1 (parqueCodigo.Tecnico):

IdTecnico	Nombre	Apellido1	Apellido2
11	Fabio	Lopez	Lopez
12	Carlos	Ramirez	

Result Grid 2 (parqueCodigo.tel_Tecnico):

IdTecnico	Telefono
11	123456
12	654321

Result Grid 3 (parqueCodigo.especialidad_Tecnico):

IdTecnico	Especialidad
11	Maquinas
11	Software
12	Programar

Creación de la clase cliente, brazalete y compra:

The screenshot displays the SQL Server Enterprise Manager interface with a query window open. The query contains SQL statements to create three tables: `parqueCodigo.cliente`, `parqueCodigo.Brazalete`, and `parqueCodigo.Compra`. Each table has a primary key and a foreign key relationship with the `cliente` table. The `cliente` table has columns `CC`, `Nombre_Cliente`, `Apellido1_Cliente`, `Apellido2_Cliente`, `Fecha_Nacimiento`, and `Edad`. The `Brazalete` table has columns `Id_brazalete` and `Precio`. The `Compra` table has columns `CC_Cliente` and `Id_brazalete`. The query also includes `INSERT INTO` statements for each table. Below the query window, three result grids are shown, each displaying the data inserted into the respective table.

```
68 • create table parqueCodigo.cliente (
69     CC varchar(20) primary key,
70     Nombre_Cliente varchar(50),
71     Apellido1_Cliente varchar(50),
72     Apellido2_Cliente varchar(50),
73     Fecha_Nacimiento varchar(50),
74     Edad int
75 );
76
77 • INSERT INTO parqueCodigo.cliente ('CC', 'Nombre_Cliente', 'Apellido1_Cliente', 'Apellido2_Cliente', 'Fecha_Nacimiento', 'Edad') VALUES
78 ('1', 'Juan', 'Ramirez','Lopez','01/01/96',26),
79 ('2', 'Pedro', 'Lopez','Lopez','01/01/93',30);
80
81 • create table parqueCodigo.Brazalete (
82     Id_brazalete varchar(20) primary key,
83     Precio double
84 );
85
86 • INSERT INTO parqueCodigo.Brazalete ('Id_brazalete', 'Precio') VALUES
87 ('21', 15000),
88 ('22', 30000);
89
90 • create table parqueCodigo.Compra (
91     CC_Cliente varchar(20),
92     Id_brazalete varchar(45),
93     primary key(CC_Cliente,Id_brazalete),
94     foreign key(CC_Cliente) references cliente(CC)
95 );
96
97 • INSERT INTO parqueCodigo.Compra ('CC_Cliente', 'Id_brazalete') VALUES
98 ('1', '21'),
99 ('2', '22');
```

Result Grid 1 (parqueCodigo.cliente):

CC	Nombre_Cliente	Apellido1_Cliente	Apellido2_Cliente	Fecha_Nacimiento	Edad
1	Juan	Ramirez	Lopez	01/01/96	26
2	Pedro	Lopez	Lopez	01/01/93	30

Result Grid 2 (parqueCodigo.Brazalete):

Id_brazalete	Precio
21	15000
22	30000

Result Grid 3 (parqueCodigo.Compra):

CC_Cliente	Id_brazalete
1	21
2	22

Tabla operador con su tabla de contacto

```
102
103 • create table parqueCodigo.Operador (
104     IdOperador varchar(20) primary key,
105     Nombre_Operador varchar(50),
106     Apellido1_Operador varchar(50),
107     Apellido2_Operador varchar(50),
108     Sueldo double
109 );
110
111 • INSERT INTO parqueCodigo.Operador (`IdOperador`, `Nombre_Operador`, `Apellido1_Operador`, `Apellido2_Operador`, `Sueldo`) VALUES
112     ('31', 'Juan', 'Ramirez', 'Lopez', 600000),
113     ('32', 'Pedro', 'Lopez', 'Lopez', 650000);
114
115 • create table parqueCodigo.tel_Operador (
116     IdOperador varchar(20),
117     Telefono varchar(45),
118     primary key(IdOperador, Telefono),
119     foreign key(IdOperador) references Operador(IdOperador)
120 );
121
122 • INSERT INTO parqueCodigo.tel_Operador (`IdOperador`, `Telefono`) VALUES
123     ('31', '123456'),
124     ('32', '654321');
```

Result Grid

IdOperador	Nombre_Operador	Apellido1_Operador	Apellido2_Operador	Sueldo
31	Juan	Ramirez	Lopez	600000
32	Pedro	Lopez	Lopez	650000
NULL	NULL	NULL	NULL	NULL

13 • select * from parqueCodigo.tel_Operador;

Result Grid

IdOperador	Telefono
31	123456
32	654321
NULL	NULL

Tabla atracción y tabla requiere

```
126
127 • create table parqueCodigo.Atraccion (
128     IdAtraccion varchar(20) primary key,
129     Nombre varchar(40),
130     Id_Proveedor varchar(20),
131     Id_Operador varchar(20),
132     foreign key(Id_Proveedor) references Proveedor(Id_Proveedor),
133     foreign key(Id_Operador) references Operador(Id_Operador)
134 );
135
136 • INSERT INTO parqueCodigo.Atraccion (`IdAtraccion`, `Nombre`, `Id_Proveedor`, `Id_Operador`) VALUES
137     ('41', 'Carritos', '01', '31'),
138     ('42', 'Carrusel', '02', '32');
139
140 • create table parqueCodigo.Requiere (
141     IdBrazalete varchar(20),
142     IdAtraccion varchar(20),
143     primary key(IdBrazalete, IdAtraccion),
144     foreign key(IdBrazalete) references Brazalete(Id_brazalete),
145     foreign key(IdAtraccion) references Atraccion(IdAtraccion)
146 );
147
148 • INSERT INTO parqueCodigo.Requiere (`IdBrazalete`, `IdAtraccion`) VALUES
149     ('21', '41'),
150     ('22', '42');
```

Result Grid

IdAtraccion	Nombre	Id_Proveedor	Id_Operador
41	Carritos	01	31
42	Carrusel	02	32
NULL	NULL	NULL	NULL

9 • select * from parqueCodigo.Atraccion;

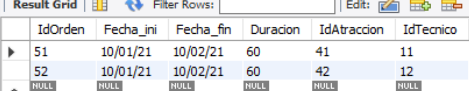
10 • select * from parqueCodigo.Requiere;

Result Grid

IdBrazalete	IdAtraccion
21	41
22	42
NULL	NULL

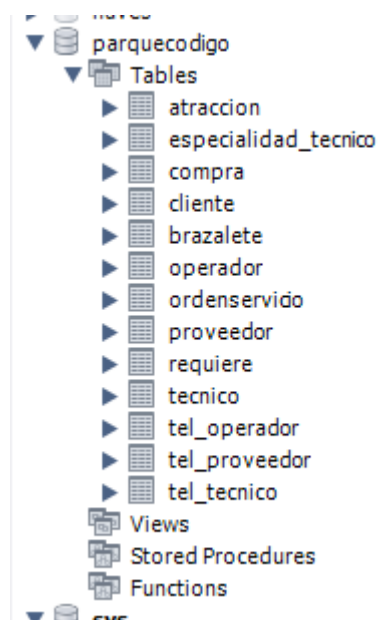
Tabla orden de servicio

```
153
154 • create table parqueCodigo.OrdenServicio (
155     IdOrden varchar(20) primary key,
156     Fecha_ini varchar(40),
157     Fecha_fin varchar(40),
158     Duracion varchar(40),
159     IdAtraccion varchar(20),
160     IdTecnico varchar(20),
161     foreign key(IdAtraccion) references Atraccion(IdAtraccion),
162     foreign key(IdTecnico) references Tecnico(IdTecnico)
163 );
164
165 • INSERT INTO parqueCodigo.OrdenServicio ('IdOrden', 'Fecha_ini', 'Fecha_fin', 'Duracion', 'IdAtraccion', 'IdTecnico') VALUES
166     ('51', '10/01/21', '10/02/21', '60', '41', '11'),
167     ('52', '10/01/21', '10/02/21', '60', '42', '12');
```



	IdOrden	Fecha_ini	Fecha_fin	Duracion	IdAtraccion	IdTecnico
▶	51	10/01/21	10/02/21	60	41	11
▶	52	10/01/21	10/02/21	60	42	12
•	NULL	NULL	NULL	NULL	NULL	NULL

Schema final



Como se puede observar la conexión entre las tablas se hizo de manera correcta, ya que no hubo ningún problema a la hora de crear los registros.