

Taller comandos SQL (DDL DML)

Contenido

Primera actividad	2
Modelo Entidad relación para una librería.....	2
Diagramar en workbench	3
Sentencias de SQL la representación del modelo relacional.	3
Segunda actividad	6
Enunciado para el planteamiento del modelo entidad relación.....	6
Modelo entidad relación propuesta por un compañero.....	6
Comentarios y correcciones del MER propuesto	6
Propuesta de MER con respecto a comentarios y correcciones	8
Modelo entidad relación elaborado en MYSQL workbench.	9
Sentencias SQL para base de datos parquenorte.....	9

Primera actividad

Modelo Entidad relación para una librería.

Se definieron las entidades: editorial, libro, cliente y autor, relacionadas de la siguiente manera:
Editorial y libro: una editorial puede publicar muchos libros y un libro le pertenece a una editorial.
Libro y Autor: un autor escribir uno o más libros y un libro le pertenece a uno o más autores.
Cliente y libro: un cliente puede comprar uno o más libros y un libro le pertenece a un cliente.

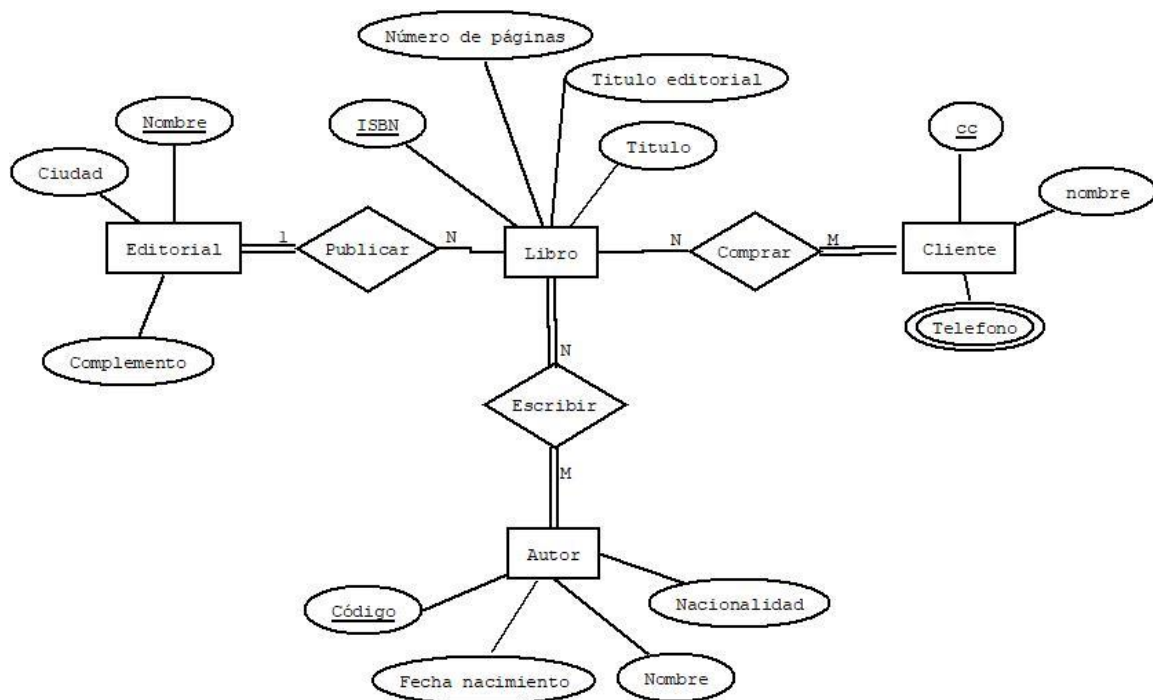


Figura 1: diagrama E R de librería.

Diagramar en workbench

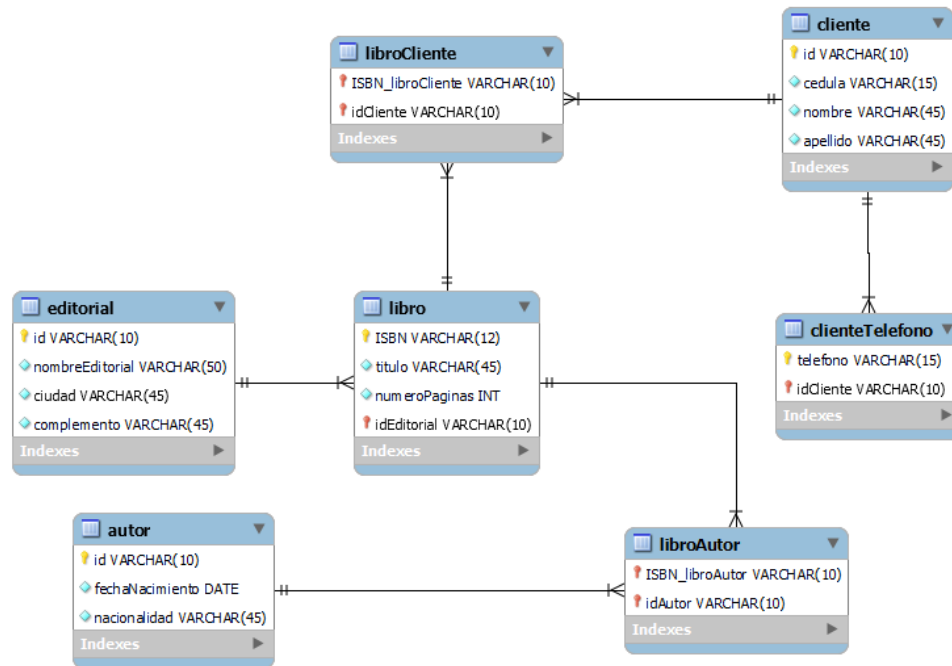


Figura 2: Diagrama MER librería.

Sentencias de SQL la representación del modelo relacional.

```
-- Table libreria.editorial
-----
CREATE TABLE IF NOT EXISTS libreria.editorial(
id VARCHAR (12),
nombreEditorial VARCHAR(50),
ciudad VARCHAR(45),
complemento VARCHAR(45),
PRIMARY KEY (id)
);

-- Table libreria.libro
-----
CREATE TABLE IF NOT EXISTS libreria.libro(
ISBN VARCHAR(10),
titulo VARCHAR(45),
numeroPaginas INT,
idEditorial VARCHAR(50),
PRIMARY KEY (ISBN),
FOREIGN KEY (idEditorial) REFERENCES editorial(id)
);
```

```
-- Table libreria.cliente
```

```
-----  
  
CREATE TABLE IF NOT EXISTS libreria.cliente(  
id VARCHAR(10),  
cedula VARCHAR(15),  
nombre VARCHAR(45),  
apellido VARCHAR (45),  
PRIMARY KEY (id)  
);
```

```
-- Table libreria.telefonoCliente
```

```
-----  
  
CREATE TABLE IF NOT EXISTS libreria.telefonoCliente(  
telefono VARCHAR(15),  
idCliente VARCHAR(10),  
    FOREIGN KEY (idCliente) REFERENCES cliente(id)  
);
```

```
-- Table libreria.libroCliente
```

```
-----  
  
CREATE TABLE IF NOT EXISTS libreria.libroCliente(  
ISBN_libroCliente VARCHAR(10),  
idCliente VARCHAR (10),  
    FOREIGN KEY (ISBN_libroCliente) REFERENCES libro(ISBN),  
    FOREIGN KEY (idCliente) REFERENCES cliente(id)  
);
```

```
-- Table libreria.autor
```

```
-----  
  
CREATE TABLE IF NOT EXISTS libreria.autor(  
    id VARCHAR(10),  
    fechaNacimiento DATE,  
    nacionalidad VARCHAR(45),  
    PRIMARY KEY (id)  
);
```

```
-- Table libreria.libroAutor
```

```
-----  
  
CREATE TABLE IF NOT EXISTS libreria.libroAutor(  
    ISBN_libroAutor VARCHAR (10),  
    idAutor VARCHAR(10),  
    FOREIGN KEY (ISBN_libroAutor) REFERENCES libro(ISBN),  
    FOREIGN KEY (idAutor) REFERENCES autor(id)
```

);

Scripts de bases de datos librería con datos.

SELECT * FROM libreria.autor;

	id	fechaNacimiento	nacionalidad	nombre	apellido
	1	2023-02-11	Colombiano	Juan	Bonilla
▶	2	2023-02-11	Colombiano	Marina	Cumbal
*	NULL	NULL	NULL	NULL	NULL

SELECT * FROM libreria.cliente;

	id	cedula	nombre	apellido
▶	1	2345	Yolima	Guadir
	2	45678	Alejandra	Diaz
*	NULL	NULL	NULL	NULL

SELECT * FROM libreria.editorial;

	id	nombreEditorial	ciudad	complemento
▶	1	Editores Yoli	Cumbal	Los mejores editores
	2	Editores Juan	Ibague	Editores de Ibague
*	NULL	NULL	NULL	NULL

SELECT * FROM libreria.libro;

	ISBN	titulo	numeroPaginas	idEditorial
▶	1234B	El psicoanalista	495	1
	5678B	The psicoanalista	495	1
*	NULL	NULL	NULL	NULL

SELECT * FROM libreria.libroautor;

	ISBN_libroAutor	idAutor
▶	1234B	1
	5678B	2

SELECT * FROM libreria.telefonocliente;

	telefono	idCliente
▶	123345	1
	1266348	1
	999999	2

SELECT * FROM libreria.librocliente;

	ISBN_libroCliente	idCliente
	5678B	1
	5678B	1
	5678B	1
▶	1234B	2

Segunda actividad

Enunciado para el planteamiento del modelo entidad relación

1. Las atracciones que están en el parque fueron adquiridas a un proveedor (nacional o internacional) el cual debe tener una información de contacto y son manipuladas por un empleado (operador).
2. Las atracciones que hay en el parque pueden necesitar un mantenimiento que será realizado por un técnico (nombre, especialidad y demás información que considere relevante) por medio de una orden de servicio (fecha, duración).
3. Para poder ingresar a las atracciones es necesario comprar un brazalete el cual es adquirido por los clientes.

Modelo entidad relación propuesta por un compañero.

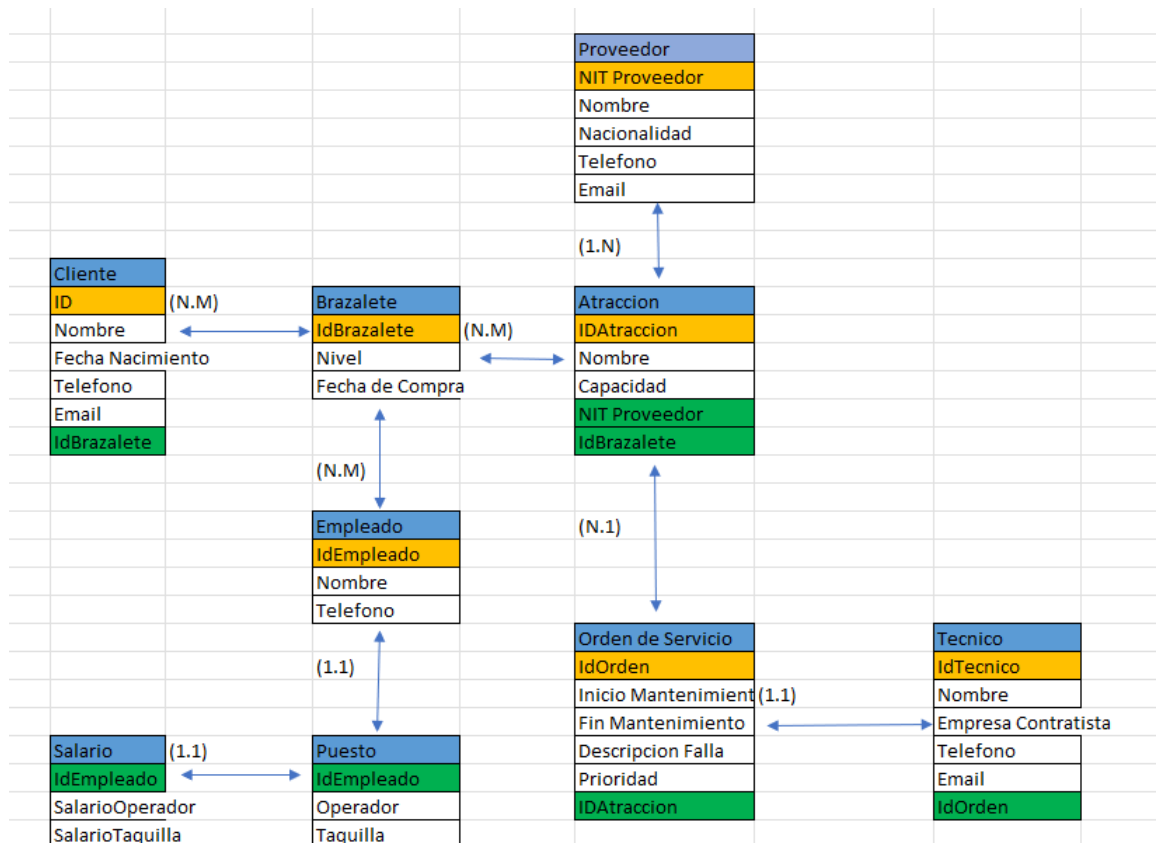


Figura 3: Modelo entidad relación

Los siguientes comentarios se realizan con respecto a las tres premisas asignadas en el taller, con respecto a esto se procede a realizar las siguientes correcciones.

Comentarios y correcciones del MER propuesto

El MER plantea no establece una relación entre atracción y un empleado con cargo de operador, por lo tanto, se agrega una tabla operador de tal manera que se cumpla la premisa de que las atracciones deben ser manipuladas por un operador. Aunque en el diagrama se vincula el operador como un atributo de la entidad puesto, la cual está relacionada con la entidad empleado, y a su vez con la entidad brazalete, lo cual permite interpretar que los empleados están relacionados con los brazaletes, sin embargo el enunciado inicial plantea una lógica donde el empleado con cargo de operador está relacionado con una tracción.

La entidad salario no se describe en el planteamiento inicial, por este motivo se procedió a eliminar la entidad.

Es necesario una tabla intermedia para registrar los brazaletes asignados por los clientes, por este motivo se agregó la tabla cliente brazalete con los atributos id cliente, id_brazalete y fecha de compra del brazalete tal como se indicaba en la entidad brazalete descrito en el diagrama establecido.

En la relación atracción, orden de servicio y técnico, se requiere vincular una tabla intermedia que relacione la orden del servicio con un técnico.

Con respecto a los anteriores comentarios se determinó las siguientes tablas con los respectivos atributos.

1. **Proveedor:** NIT_Proveedor, nombre, nacionalidad, teléfono, email.
2. **Atracción:** id_atraccion, nombre, proveedor_id, operador_id.
3. **Cliente:** id, nombre, fecha nacimiento, teléfono, email
4. **Operador:** id operador, cedula, nombre, apellido.
5. **Técnico:** d, nombre, empresa contratista, teléfono, email.
6. **Brazalete:** id: nivel, precio.

Además, se definió tablas intermedias para relacionar las siguientes tablas.

7. **Cliente_Brazalete:** id cliente, id brazalete, fecha de compra.
8. **OrdenServicio_Tecnico:** id orden de servicio, id técnico, inicio mantenimiento, fin mantenimiento.

Cardinalidad:

1. Cliente y cliente brazalete: Un cliente puede obtener muchos brazaletes, mientras que un brazalete le pertenece a un cliente.
2. Brazalete y atracción: brazalete permite la entrada a una atracción y una atracción permite el acceso de uno más brazaletes.
3. Proveedor y atracción: una atracción pertenece a un proveedor, mientras que un proveedor provee una o más atracciones.
4. Operador y atracción: un operador manipula una atracción y una atracción es manipulada por una o más atracciones.

5. Orden de servicio y atracción: Una orden de servicio le pertenece a una atracción y una atracción puede tener una o más ordenes de servicio.
6. Orden de servicio técnico y técnico: un técnico le asignan una o más ordenes de servicio y una orden de servicio puede ser atendida por uno más técnicos.

Propuesta de MER con respecto a comentarios y correcciones

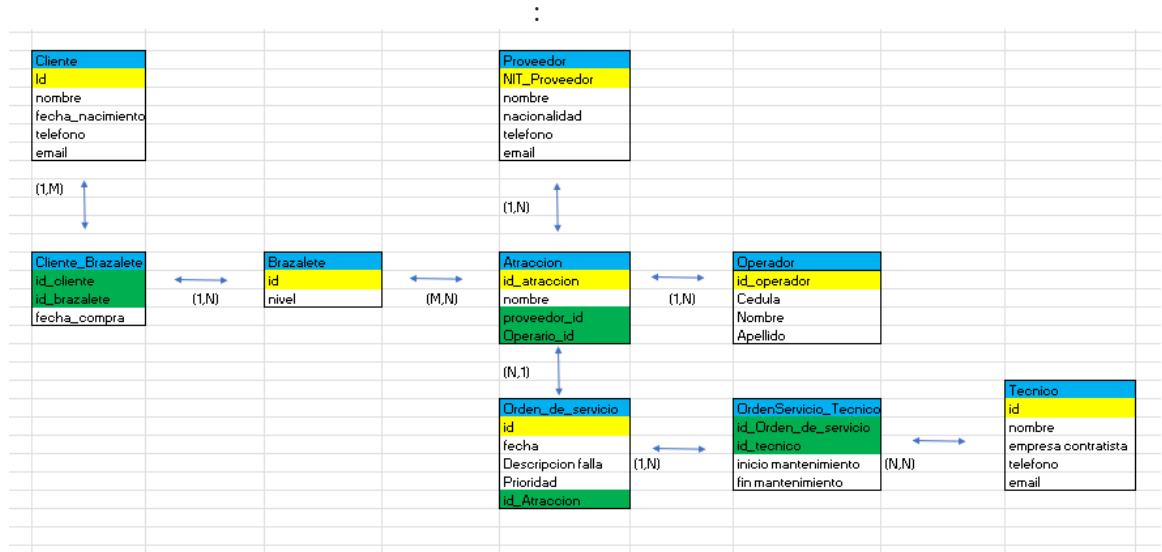


Figura 4: MER corregido.

Modelo entidad relación elaborado en MYSQL workbench.

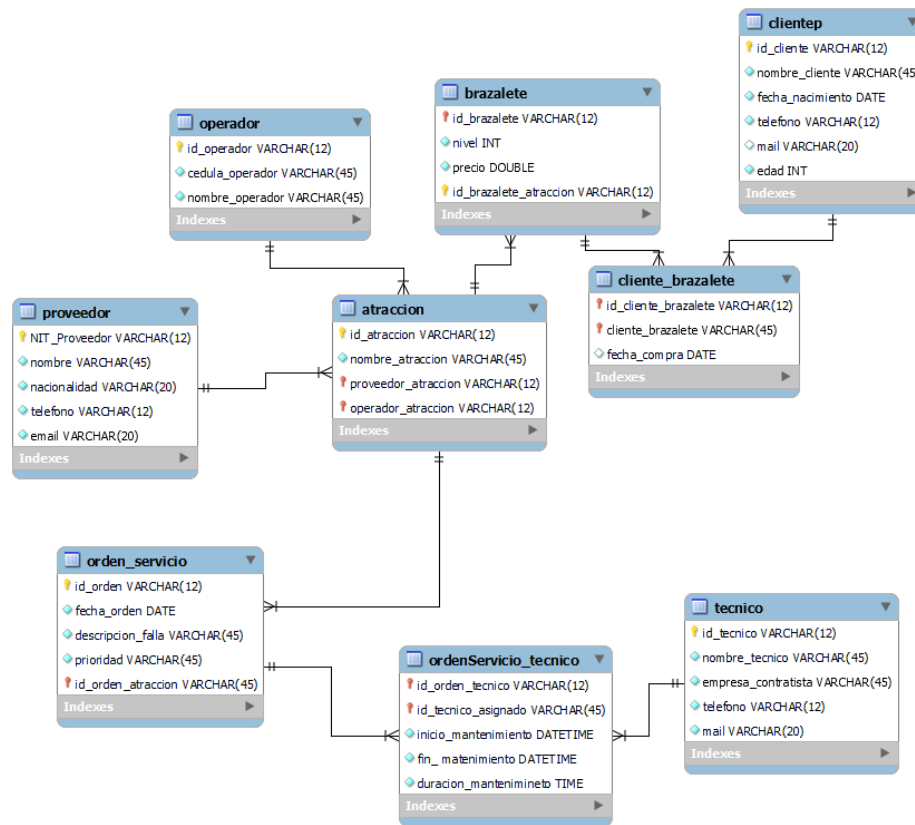


Figura 5: MER elaborado en MYSQL workbench.

Sentencias SQL para base de datos parquenorte

```

-----
-- Table parquenorte.cliente
-----
CREATE TABLE IF NOT EXISTS parquenorte.cliente(
id_cliente VARCHAR(12) NOT NULL,
nombre_cliente VARCHAR (45) NOT NULL,
fecha_nacimiento DATE NOT NULL,
telefono VARCHAR (12) NOT NULL,
mail VARCHAR (20) NOT NULL,
edad INT NOT NULL,
PRIMARY KEY (id_cliente)
);

-----
-- Table parquenorte.brazaletes
-----
CREATE TABLE IF NOT EXISTS parquenorte.brazaletes(
id_brazaletes VARCHAR(12) NOT NULL,
nivel INT NOT NULL,

```

```
precio DOUBLE,  
id_brazalete_atraccion VARCHAR(12),  
PRIMARY KEY (id_brazalete)  
);  
ALTER TABLE parquenorte.brazalete RENAME COLUMN id_brazalete_atraccion TO idAtraccion;  
ALTER TABLE parquenorte.brazalete ADD FOREIGN KEY (idAtraccion) REFERENCES  
parquenorte.atraccion(id);
```

```
-- -----  
-- Table parquenorte.cliente_brazalete  
-- -----
```

```
CREATE TABLE IF NOT EXISTS parquenorte.cliente_brazalete(  
id VARCHAR(12) PRIMARY KEY,  
fecha_compra DATE,  
idCliente VARCHAR(12) NOT NULL,  
idBrazalete VARCHAR(12) NOT NULL,  
FOREIGN KEY (idCliente) REFERENCES cliente(id_cliente),  
FOREIGN KEY (idBrazalete) REFERENCES brazalete(id_brazalete)  
);
```

```
-- -----  
-- Table parquenorte.proveedor  
-- -----
```

```
CREATE TABLE IF NOT EXISTS parquenorte.proveedor(  
Nit VARCHAR(12),  
nombre VARCHAR(45),  
nacionalidad VARCHAR(30),  
telefono VARCHAR(12),  
email VARCHAR(20),  
PRIMARY KEY (Nit)  
);
```

```
-- -----  
-- Table parquenorte.operador  
-- -----
```

```
CREATE TABLE IF NOT EXISTS parquenorte.operario(  
id VARCHAR(12),  
cedula VARCHAR(15),  
nombre VARCHAR (45),  
PRIMARY KEY (id)  
);
```

```
-- -----  
-- Table parquenorte.atraccion  
-- -----
```

```
CREATE TABLE IF NOT EXISTS parquenorte.atraccion(  
id VARCHAR(12),
```

```

nombre VARCHAR(45) NOT NULL,
idProveedor VARCHAR (12) NOT NULL,
idOperario VARCHAR (12) NOT NULL,
    FOREIGN KEY (idProveedor) REFERENCES proveedor(Nit),
    FOREIGN KEY (idOperario) REFERENCES operario(id)
);
ALTER TABLE parquenorte.atraccion ADD PRIMARY KEY (id);
CREATE INDEX idx_atraccion_id ON parquenorte.atraccion(id);
-----
-- Table parquenorte.ordenServicio
-----
CREATE TABLE IF NOT EXISTS parquenorte.ordenServicio(
id VARCHAR(12),
fechaOrden DATE,
descripcionFalla VARCHAR(80),
prioridad VARCHAR (45),
PRIMARY KEY (id)
);
-----
-- Table parquenorte.tecnico
-----
CREATE TABLE IF NOT EXISTS parquenorte.tecnico(
id VARCHAR(12),
nombreTecnico VARCHAR(50),
empresaContratista VARCHAR(45),
telefono VARCHAR(12),
mail VARCHAR (20),
PRIMARY KEY (id)
);
-----
-- Table parquenorte.servicioTecnico
-----
CREATE TABLE IF NOT EXISTS parquenorte.servicioTecnico(
id VARCHAR(12),
idOrden VARCHAR (12),
idTecnico VARCHAR (12),
horaInicio DATETIME,
horaFin DATETIME,
duracionMantenimiento TIME,
PRIMARY KEY (id),
    FOREIGN KEY (idOrden) REFERENCES ordenServicio(id),
    FOREIGN KEY (idTecnico) REFERENCES tecnico(id)
);

```

Scripts de bases de datos parque norte con datos

SELECT * FROM parquenorte.tecnico;

	id	nombreTecnico	empresaContratista	telefono	mail
	1	Juan	Empresa sur	12345	juna@gmail.com
▶	2	Ana	Atracciones SA	23456	an@gmail.com
*	NULL	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.operario;

	id	cedula	nombre
	1	123456	July Cifuentes
▶	2	34567	la Romero
*	NULL	NULL	NULL

SELECT * FROM parquenorte.cliente;

	id_cliente	nombre_cliente	fecha_nacimiento	telefono	mail	edad
	1	july	2000-01-02	676767	j@gmail.com	23
▶	2	mario	2010-07-08	789090	m@gmail.com	13
*	NULL	NULL	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.proveedor;

	Nit	nombre	nacionalidad	telefono	email
	700067	Diversión	Ecuatoriana	67899	ED@gmail.com
▶	768967	Atracci...	Colombiana	44905	atco@gmail.com
*	NULL	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.atraccion;

	id	nombre	idProveedor	idOperario
	1	carritos	700067	1
▶	2	monta...	768967	2
*	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.brazalete;

	id_brazalete	nivel	precio	idAtraccion
	1	2	200000	1
▶	2	3	300000	2
*	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.cliente_brazalete;

	id	fecha_compra	idCliente	idBrazalete
	1	2023-02-11	1	1
▶	2	2023-01-02	2	1
*	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.ordenservicio;

	id	fechaOrden	descripcionFalla	prioridad	id_atraccion
	1	2023-01-02	mecanica	alta	1
▶	2	2023-11-02	electronica	alta	2
*	NULL	NULL	NULL	NULL	NULL

SELECT * FROM parquenorte.serviciotecnico;

	id	idOrden	idTecnico	horaInicio	horaFin	duracionMantenimiento
	1	1	2	10:30:00	11:30:00	60
▶	2	2	1	10:30:00	12:30:00	120
✱	NULL	NULL	NULL	NULL	NULL	NULL