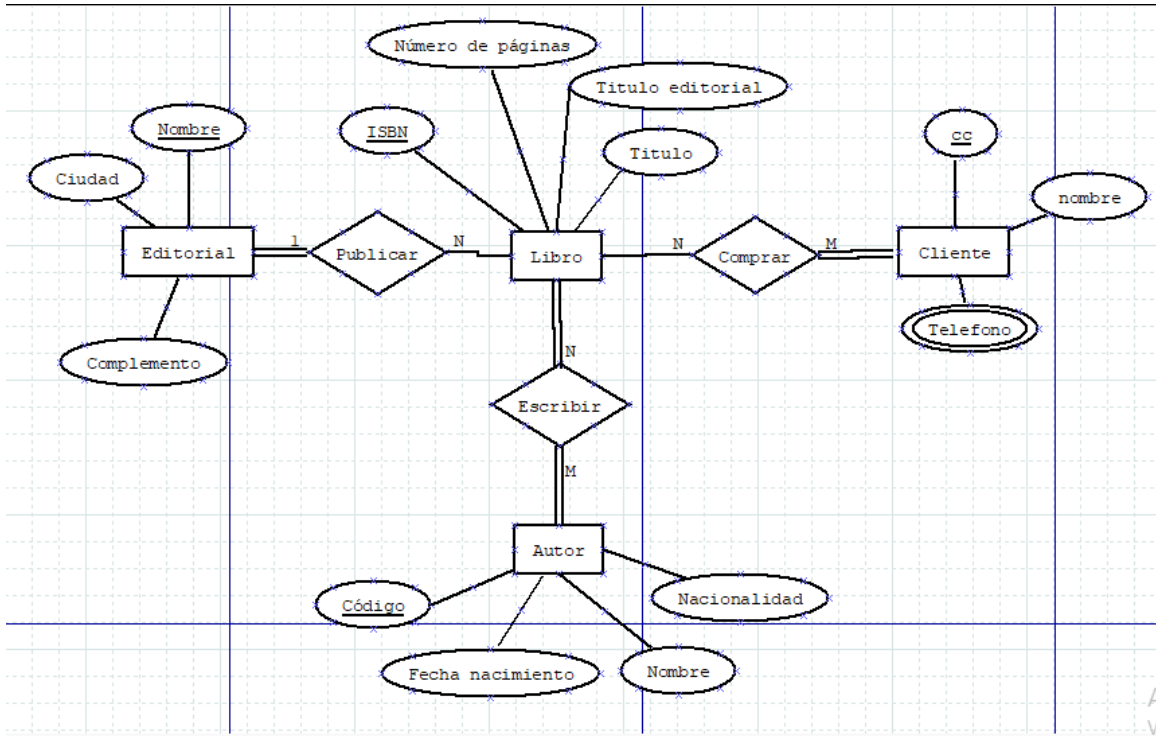
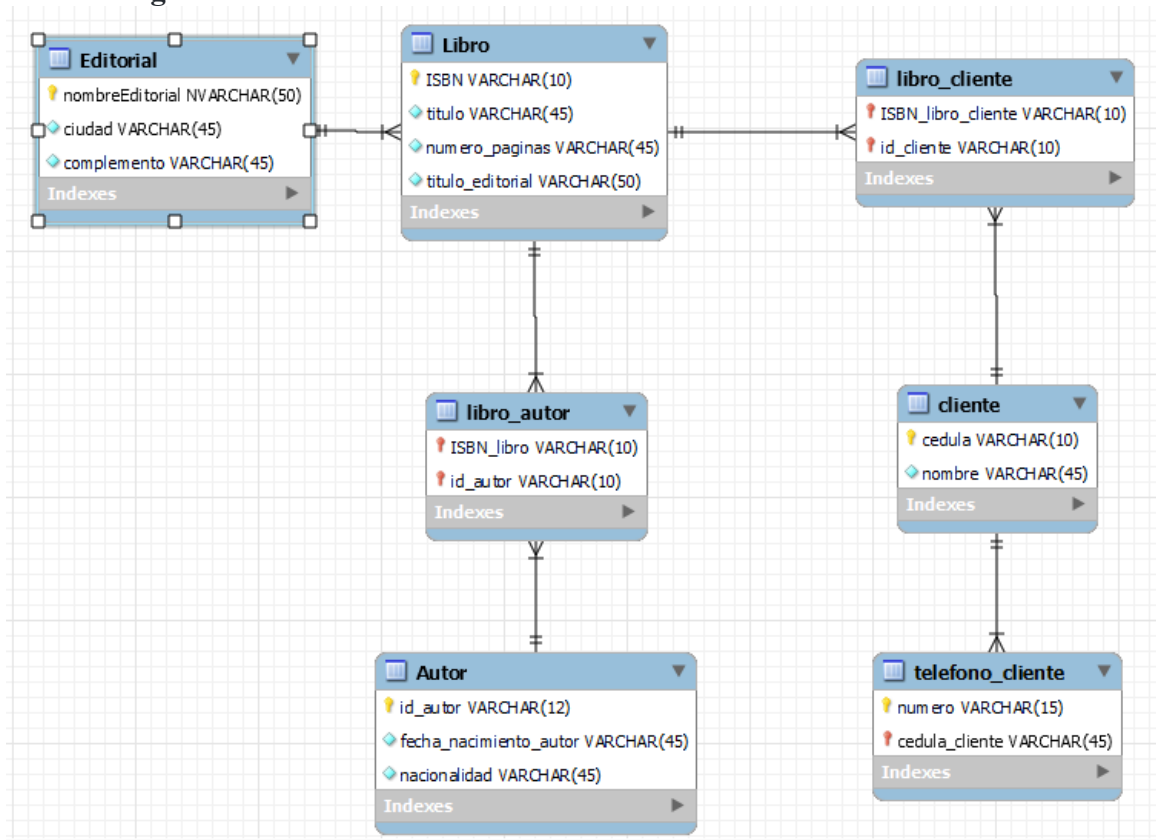


1. Diagrama E-R Librería



2. Diagramar en workbench



3. Sentencias de SQL la representación del modelo relacional.

-- Table `libreria`.`editorial`

```
CREATE TABLE IF NOT EXISTS `libreria`.`editorial` (  
  `nombreEditorial` VARCHAR(50) NOT NULL,  
  `ciudad` VARCHAR(45) NOT NULL,  
  `complemento` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`nombreEditorial`))  
ENGINE = InnoDB;
```

-- Table `libreria`.`libro`

```
CREATE TABLE IF NOT EXISTS `libreria`.`libro` (  
  `ISBN` VARCHAR(10) NOT NULL,  
  `titulo` VARCHAR(45) NOT NULL,  
  `numero_paginas` VARCHAR(45) NOT NULL,  
  `titulo_editorial` VARCHAR(50) NOT NULL,  
  PRIMARY KEY (`ISBN`),  
  CONSTRAINT `titulo_editorial`  
    FOREIGN KEY ()  
    REFERENCES `libreria`.`editorial` ()  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `libreria`.`cliente`

```
CREATE TABLE IF NOT EXISTS `libreria`.`cliente` (  
  `cedula` VARCHAR(10) NOT NULL,  
  `nombre` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`cedula`))  
ENGINE = InnoDB;
```

-- Table `libreria`.`telefono_cliente`

```
CREATE TABLE IF NOT EXISTS `libreria`.`telefono_cliente` (  
  `numero` VARCHAR(15) NOT NULL,  
  `cedula_cliente` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`numero`, `cedula_cliente`),
```

```
INDEX `cedula_cliente_idx` (`cedula_cliente` ASC) VISIBLE,  
CONSTRAINT `cedula_cliente`  
  FOREIGN KEY (`cedula_cliente`)  
  REFERENCES `libreria`.`cliente` (`cedula`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `libreria`.`libro_cliente`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `libreria`.`libro_cliente` (  
  `ISBN_libro_cliente` VARCHAR(10) NOT NULL,  
  `id_cliente` VARCHAR(10) NOT NULL,  
  PRIMARY KEY (`ISBN_libro_cliente`, `id_cliente`),  
  INDEX `id_cliente_idx` (`id_cliente` ASC) VISIBLE,  
  CONSTRAINT `id_cliente`  
    FOREIGN KEY (`id_cliente`)  
    REFERENCES `libreria`.`cliente` (`cedula`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `ISBN_libro_cliente`  
    FOREIGN KEY (`ISBN_libro_cliente`)  
    REFERENCES `libreria`.`libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

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

```
CREATE TABLE IF NOT EXISTS `libreria`.`autor` (  
  `id_autor` VARCHAR(12) NOT NULL,  
  `fecha_nacimiento_autor` VARCHAR(45) NOT NULL,  
  `nacionalidad` VARCHAR(45) NOT NULL,  
  `nombre` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_autor`))  
ENGINE = InnoDB;
```

```
-- -----  
-- Table `libreria`.`libro_autor`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `libreria`.`libro_autor` (  
  `ISBN_libro` VARCHAR(10) NOT NULL,  
  `id_autor` VARCHAR(12) NOT NULL,  
  PRIMARY KEY (`ISBN_libro`, `id_autor`),  
  INDEX `id_autor_idx` (`id_autor` ASC) VISIBLE,  
  CONSTRAINT `id_autor`  
    FOREIGN KEY (`id_autor`)  
    REFERENCES `libreria`.`autor` (`id_autor`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `ISBN_libro`  
    FOREIGN KEY (`ISBN_libro`)  
    REFERENCES `libreria`.`libro` (`ISBN`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```

`ISBN_libro` VARCHAR(10) NOT NULL,
`id_autor` VARCHAR(10) NOT NULL,
PRIMARY KEY (`ISBN_libro`, `id_autor`),
INDEX `id_autor_idx` (`id_autor` ASC) VISIBLE,
CONSTRAINT `ISBN_libro`
  FOREIGN KEY (`ISBN_libro`)
  REFERENCES `libreria`.`libro` (`ISBN`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `id_autor`
  FOREIGN KEY (`id_autor`)
  REFERENCES `libreria`.`autor` (`id_autor`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

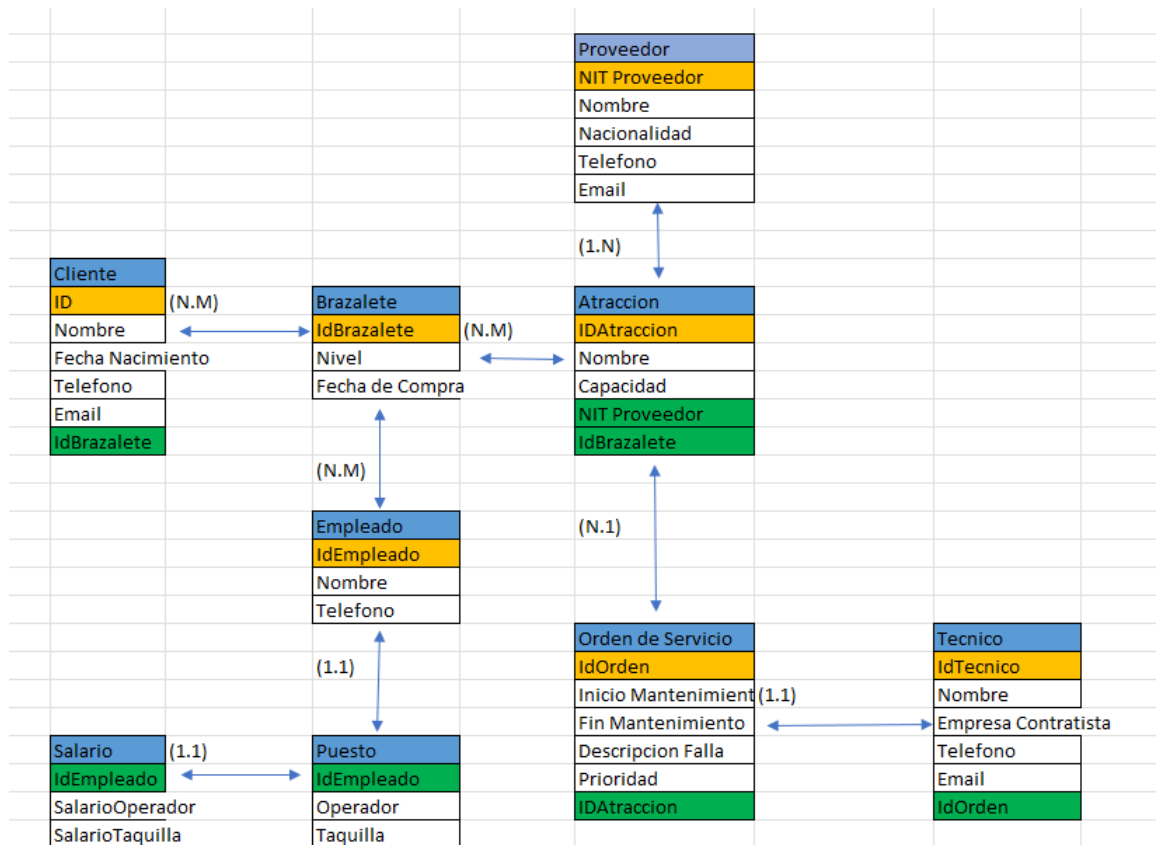
```

Segunda actividad

Modelo entidad relación

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.

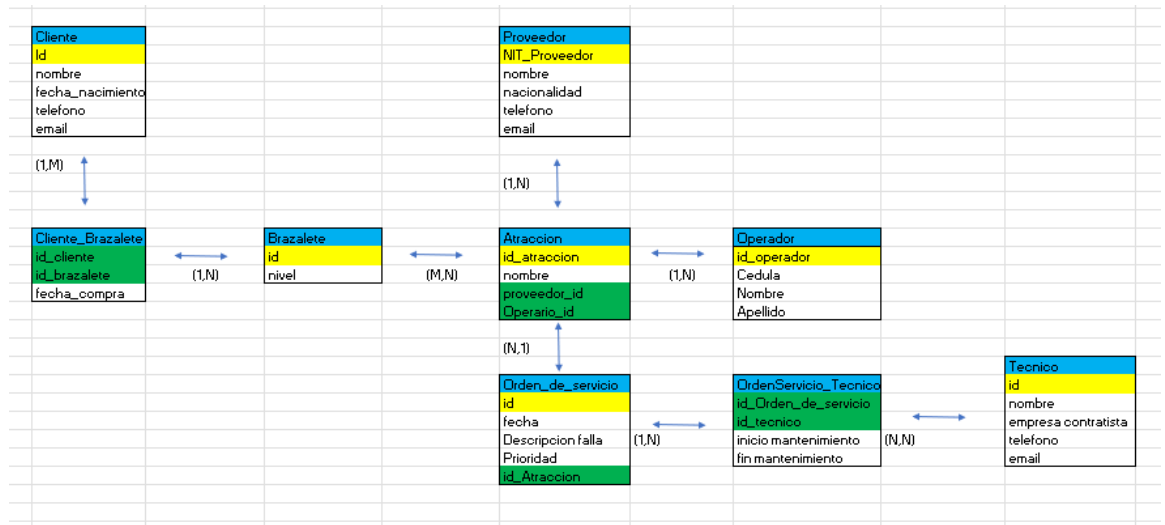


Corrección

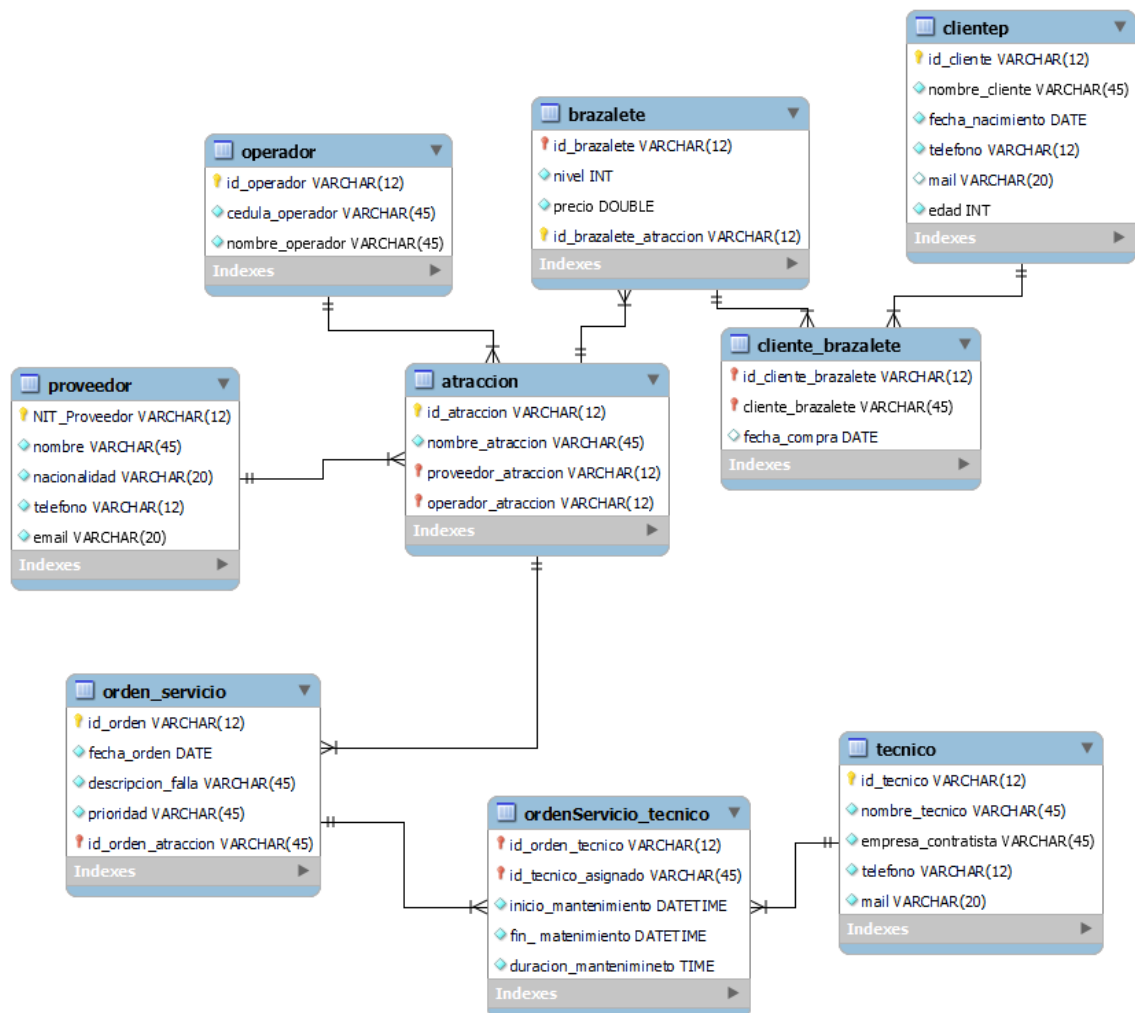
El MER plantea no establece una relación entre atracción y un empleado con cargo de operador, por lo tanto, se agrega una tabla operador de tal manera que se cumpla la premisa de que las atracciones deben ser manipuladas por un operador.

Es necesario una tabla intermedia para registrar los brazaletes asignados por los clientes, por este motivo se agrego la tabla cliente brazaletes con los atributos id_cliente , id_ brazaletes y fecha de compra del brazaletes tal como se indicaba en la entidad brazaletes 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.



MER parque norte



Sentencias SQL para base de datos parquenorte

-- Table `parquenorte`.`proveedor`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`proveedor` (  
  `NIT_Proveedor` VARCHAR(12) NOT NULL,  
  `nombre` VARCHAR(45) NOT NULL,  
  `nacionalidad` VARCHAR(20) NOT NULL,  
  `telefono` VARCHAR(12) BINARY NOT NULL,  
  `email` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`NIT_Proveedor`))  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`clientep`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`clientep` (  
  `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) NULL,  
  `edad` INT NOT NULL,  
  PRIMARY KEY (`id_cliente`))  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`operador`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`operador` (  
  `id_operador` VARCHAR(12) NOT NULL,  
  `cedula_operador` VARCHAR(45) NOT NULL,  
  `nombre_operador` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_operador`))  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`tecnico`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`tecnico` (  
  `id_tecnico` VARCHAR(12) NOT NULL,  
  `nombre_tecnico` VARCHAR(45) NOT NULL,  
  `empresa_contratista` VARCHAR(45) NOT NULL,
```

```
`telefono` VARCHAR(12) NOT NULL,  
`mail` VARCHAR(20) NOT NULL,  
PRIMARY KEY (`id_tecnico`))  
ENGINE = InnoDB;
```

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

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`atraccion` (  
  `id_atraccion` VARCHAR(12) NOT NULL,  
  `nombre_atraccion` VARCHAR(45) NOT NULL,  
  `proveedor_atraccion` VARCHAR(12) NOT NULL,  
  `operador_atraccion` VARCHAR(12) NOT NULL,  
  PRIMARY KEY (`id_atraccion`, `proveedor_atraccion`, `operador_atraccion`),  
  INDEX `proveedor_atraccion_idx` (`operador_atraccion` ASC, `proveedor_atraccion`  
ASC) VISIBLE,  
  CONSTRAINT `operador_atraccion`  
    FOREIGN KEY (`operador_atraccion`)  
    REFERENCES `parquenorte`.`operador` (`id_operador`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `proveedor_atraccion`  
    FOREIGN KEY (`operador_atraccion`, `proveedor_atraccion`)  
    REFERENCES `parquenorte`.`proveedor` (`NIT_Proveedor`, `NIT_Proveedor`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
-----  
-- Table `parquenorte`.`brazalete`  
-----
```

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`brazalete` (  
  `id_brazalete` VARCHAR(12) NOT NULL,  
  `nivel` INT NOT NULL,  
  `precio` DOUBLE NOT NULL,  
  `id_brazalete_atraccion` VARCHAR(12) NOT NULL,  
  PRIMARY KEY (`id_brazalete`, `id_brazalete_atraccion`),  
  CONSTRAINT `id_brazalete_atraccion`  
    FOREIGN KEY (`id_brazalete`)  
    REFERENCES `parquenorte`.`atraccion` (`id_atraccion`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`cliente_brazalete`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`cliente_brazalete` (  
  `id_cliente_brazalete` VARCHAR(12) NOT NULL,  
  `cliente_brazalete` VARCHAR(45) NOT NULL,  
  `fecha_compra` DATE NULL,  
  PRIMARY KEY (`id_cliente_brazalete`, `cliente_brazalete`),  
  INDEX `cliente_brazalete_idx` (`cliente_brazalete` ASC) VISIBLE,  
  CONSTRAINT `id_cliente_brazalete`  
    FOREIGN KEY (`id_cliente_brazalete`)  
      REFERENCES `parquenorte`.`clientep` (`id_cliente`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `cliente_brazalete`  
    FOREIGN KEY (`cliente_brazalete`)  
      REFERENCES `parquenorte`.`brazalete` (`id_brazalete`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`orden_servicio`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`orden_servicio` (  
  `id_orden` VARCHAR(12) NOT NULL,  
  `fecha_orden` DATE NOT NULL,  
  `descripcion_falla` VARCHAR(45) NOT NULL,  
  `prioridad` VARCHAR(45) NOT NULL,  
  `id_orden_atraccion` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_orden`, `id_orden_atraccion`),  
  INDEX `id_orden_atraccion_idx` (`id_orden_atraccion` ASC) VISIBLE,  
  CONSTRAINT `id_orden_atraccion`  
    FOREIGN KEY (`id_orden_atraccion`)  
      REFERENCES `parquenorte`.`atraccion` (`id_atraccion`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `parquenorte`.`ordenServicio_tecnico`

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`ordenServicio_tecnico` (  
  `id_orden` VARCHAR(12) NOT NULL,  
  `id_cliente` VARCHAR(12) NOT NULL,  
  `id_atraccion` VARCHAR(45) NOT NULL,  
  `id_tecnico` VARCHAR(12) NOT NULL,  
  `fecha_servicio` DATE NOT NULL,  
  `descripcion_falla` VARCHAR(45) NOT NULL,  
  `prioridad` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_orden`, `id_cliente`, `id_atraccion`, `id_tecnico`, `fecha_servicio`, `descripcion_falla`, `prioridad`),  
  INDEX `id_cliente` (`id_cliente` ASC) VISIBLE,  
  INDEX `id_atraccion` (`id_atraccion` ASC) VISIBLE,  
  INDEX `id_tecnico` (`id_tecnico` ASC) VISIBLE,  
  INDEX `fecha_servicio` (`fecha_servicio` ASC) VISIBLE,  
  INDEX `descripcion_falla` (`descripcion_falla` ASC) VISIBLE,  
  INDEX `prioridad` (`prioridad` ASC) VISIBLE,  
  CONSTRAINT `id_orden`  
    FOREIGN KEY (`id_orden`)  
      REFERENCES `parquenorte`.`orden_servicio` (`id_orden`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `id_cliente`  
    FOREIGN KEY (`id_cliente`)  
      REFERENCES `parquenorte`.`cliente_brazalete` (`id_cliente_brazalete`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `id_atraccion`  
    FOREIGN KEY (`id_atraccion`)  
      REFERENCES `parquenorte`.`atraccion` (`id_atraccion`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `id_tecnico`  
    FOREIGN KEY (`id_tecnico`)  
      REFERENCES `parquenorte`.`tecnico` (`id_tecnico`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
`id_orden_tecnico` VARCHAR(12) NOT NULL,  
`id_tecnico_asignado` VARCHAR(45) NOT NULL,  
`inicio_mantenimiento` DATETIME NOT NULL,  
`fin_mantenimiento` DATETIME NOT NULL,  
`duracion_mantenimiento` TIME NOT NULL,  
PRIMARY KEY (`id_orden_tecnico`, `id_tecnico_asignado`),  
INDEX `id_tecnico_asignado_idx` (`id_tecnico_asignado` ASC) VISIBLE,  
CONSTRAINT `id_orden_tecnico`  
  FOREIGN KEY (`id_orden_tecnico`)  
    REFERENCES `parquenorte`.`orden_servicio` (`id_orden`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
CONSTRAINT `id_tecnico_asignado`  
  FOREIGN KEY (`id_tecnico_asignado`)  
    REFERENCES `parquenorte`.`tecnico` (`id_tecnico`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```