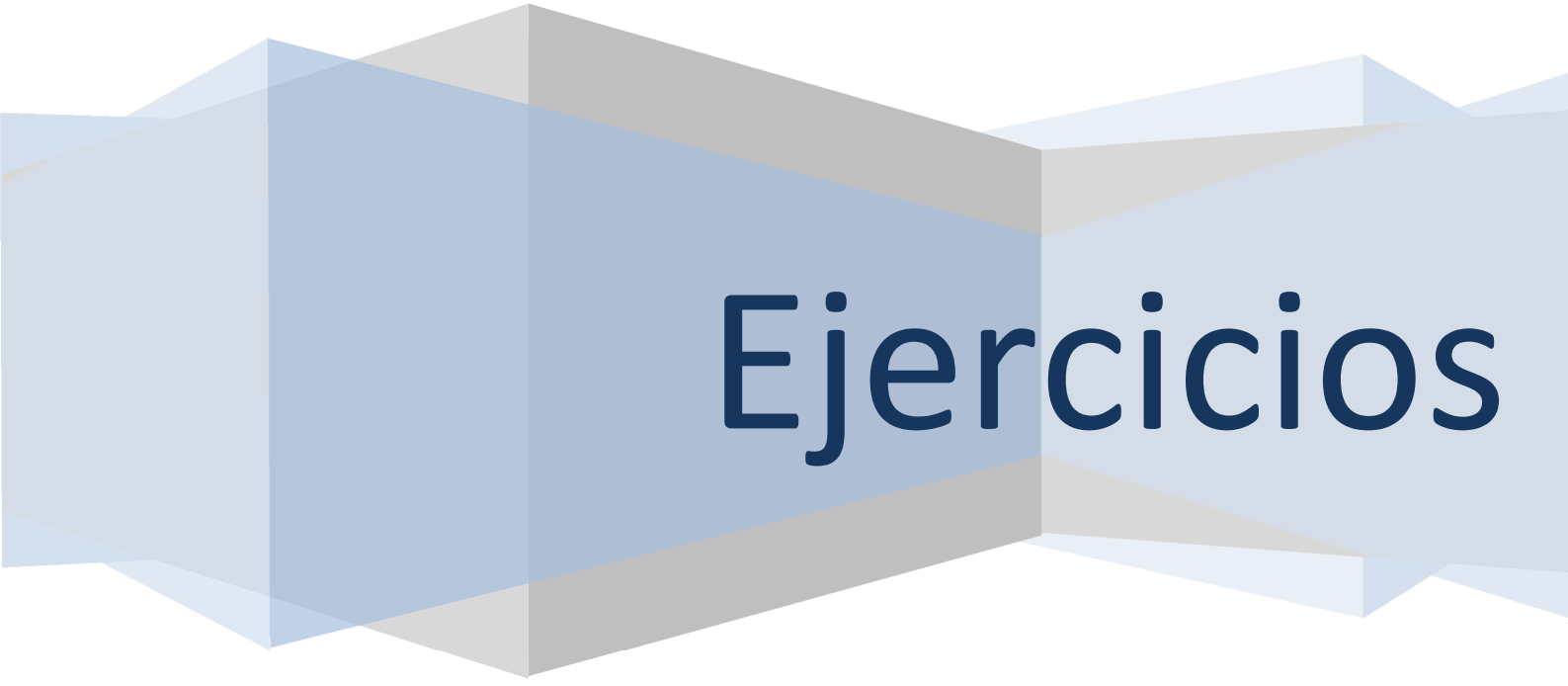
BASES DE DATOS

Administración de Desarrollo de Aplicaciones Multiplataforma
Administración de Desarrollo de Aplicaciones Web
GESTIÓN DE BASES DE DATOS
Administración de Sistemas Informáticos en Red

MODELO FÍSICO
LENGUAJE DE DESCRIPCIÓN DE DATOS

Luis Dorado
Vanesa Martínez
Pablo Bahillo
Alba Tortosa



Contenido

1 Lenguaje de definición de datos (DDL) I

1.1 Creación de un BD

Crea una base de datos que vamos a usar para los ejercicios sencillos, en ella vas a crear todas las tablas de este primer apartado. La vas a llamar “ejercicios”.
Nota: Comprueba que exista antes de crearla.

SOLUCIÓN:

```
1 •  create database if not exists ejercicios;
```

1.2 Creación de tablas sin restricciones

1.2.1 Implementa las siguientes tablas en MySQL

EMPLEADOS
DNI: VARCHAR(9) (PK)
nombre: VARCHAR(30) (NN)
apellidos: VARCHAR(60) (NN)
email: VARCHAR(60)
telefono: INTEGER (NN)
sueldos: DECIMAL(10,2) (NN)
puesto: VARCHAR(30) (NN)

LIBROS
ISBN: numero (PK)
titulo: VARCHAR(60) (NN)
tipo: VARCHAR(30) (NN)
autor: VARCHAR(90)
precio: DECIMAL(8,2) (NN)

FACTURAS
CÓDIGO: num (PK)
destinatario: VARCHAR(90) (NN)
cuenta: INTEGER (NN)
importe: DECIMAL(10,2) (NN)
fecha_hora: DATETIME (NN) (UNQ)

NOTA: Comprueba que las tablas se creen en la BD que has creado antes ejercicio.

SOLUCIÓN:

```
USE ejercicio;
CREATE TABLE empleados(
DNI VARCHAR(30),
nombre VARCHAR(30),
email VARCHAR(30),
telefono INTEGER,
sueldo DECIMAL (10,2),
puesto VARCHAR(30)
);
```

```
CREATE TABLE libros(
ISBN INT,
titulo VARCHAR(60),
tipo VARCHAR(30),
autor VARCHAR(90),
precio DECIMAL(8,2)
);
```

```
CREATE TABLE codigo(
CODIGO INT,
destinatario VARCHAR(90),
cuenta INTEGER,
importe DECIMAL(10,2),
fecha_hora DATETIME
);
```

1.2.2 Implementa las siguientes tablas en MySQL

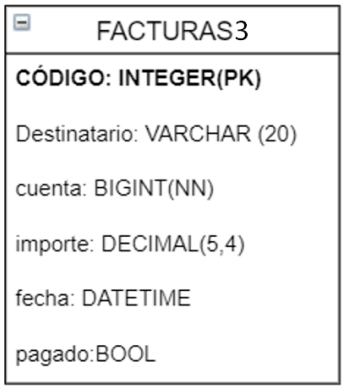
FACTURAS2
CÓDIGO: INTEGER
Destinatario: VARCHAR (20)
cuenta: BIGINT
importe: DECIMAL(5,4)
fecha: DATETIME
pagado:BOOL

SOLUCIÓN:

```
CREATE TABLE FACTURAS(
CODIGO INTEGER,
destinatario VARCHAR(20),
cuenta BIGINT,
importe DECIMAL(5,4),
fecha DATETIME,
pagado BOOL
);
```

1.3 Creación tablas con restricciones

1.3.1 Implementa las siguientes tablas en MySQL con restricciones

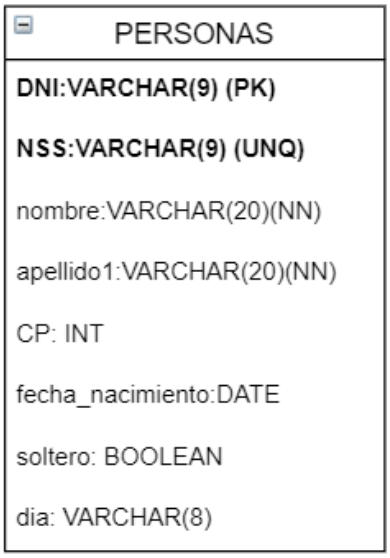


NOTA: El valor por defecto del campo datetime tendrá por defecto el valor de la fecha actual para ello usa la función CURTIME()

SOLUCIÓN:

```
CREATE TABLE FACTURAS2(
CODIGO INTEGER,
destinatario VARCHAR(20),
cuenta BIGINT NOT NULL,
importe DECIMAL(5,4),
fecha DATETIME DEFAULT (CURTIME()),
pagado BOOL,
PRIMARY KEY(CODIGO)
);
```

1.3.2 Implementa las siguientes tablas en MySQL con restricciones



NOTA: Ten en cuenta que NSS es clave candidata y el atributo soltero tiene que tener por defecto 'TRUE'

SOLUCIÓN:

```
CREATE TABLE PERSONAS(
DNI VARCHAR(9),
NSS VARCHAR(9),
nombre VARCHAR (20)NOT NULL,
apellido VARCHAR(20) NOT NULL,
CP INT,
fecha_nacimiento DATE,
soltero BOOLEAN DEFAULT('TRUE'),
dia VARCHAR(8),
PRIMARY KEY (DNI),
UNIQUE(NSS)
);
```

1.4 Borrado de tablas: Elimina las tablas creadas en ej. 1.3.1 y 1.3.2

SOLUCIÓN:

```
DROP TABLE PERSONAS;
DROP TABLE FACTURAS2;
```

1.5 Modificación de las tablas

1.5.1 Cambia el nombre del atributo "importe" del ej. 1.2.2 por "importe_total"

SOLUCIÓN:

```
ALTER TABLE FACTURAS RENAME COLUMN importe TO importe_total;
```

1.5.2 Cambia el nombre de la tabla FACTURAS por FACTUR

SOLUCIÓN:

```
RENAME TABLE FACTURAS TO FACTUR;
```

1.5.3 Cambia el nombre del atributo "cuenta" del ej. 1.2.2 por "cuenta_destinatario"

SOLUCIÓN:

```
ALTER TABLE FACTUR RENAME COLUMN cuenta TO cuenta_destinatario;
```

1.5.4 Añade una columna a la tabla libros llamada editorial que sea un varchar(30)

SOLUCIÓN:

```
ALTER TABLE libros ADD COLUMN editorial VARCHAR(30);
```

1.5.5 Elimina la columna email de la tabla EMPLEADOS

SOLUCIÓN:

```
ALTER TABLE EMPLEADOS DROP COLUMN email;
```

1.5.6 Añade a libros una clave primaria la cual tiene que ser ISBN

SOLUCIÓN:

```
ALTER TABLE libros ADD PRIMARY KEY (ISBN);
```

1.5.7 Añade a empleados una clave primaria que sea la unión de DNI y nombre

SOLUCIÓN:

```
ALTER TABLE empleados ADD PRIMARY KEY(DNI,nombre);
```

1.5.8 Borra la clave primaria antes creada y crea como clave primaria solo DNI y como clave candidata nombre

SOLUCIÓN:

```
ALTER TABLE empleados DROP PRIMARY KEY;
ALTER TABLE empleados ADD PRIMARY KEY (DNI);
ALTER TABLE empleados MODIFY COLUMN nombre VARCHAR(30) NOT NULL;
ALTER TABLE empleados ADD UNIQUE(nombre);
```

1.5.9 Muestra todas las tablas que tienes en la BD de ejercicio

SOLUCIÓN:

```
SHOW TABLES;
```

1.5.10 Describe cada una de los campos de los que está formada la tabla empleados

SOLUCIÓN:

```
DESCRIBE empleados;
```

	Field	Type	Null	Key	Default	Extra
▶	DNI	varchar(30)	NO		NULL	
	nombre	varchar(30)	NO	PRI	NULL	
	email	varchar(30)	YES		NULL	
	telefono	int	YES		NULL	
	sueldo	decimal(10,2)	YES		NULL	
	puesto	varchar(30)	YES		NULL	

1.5.11 Muestra el código de creación de la tabla empleados

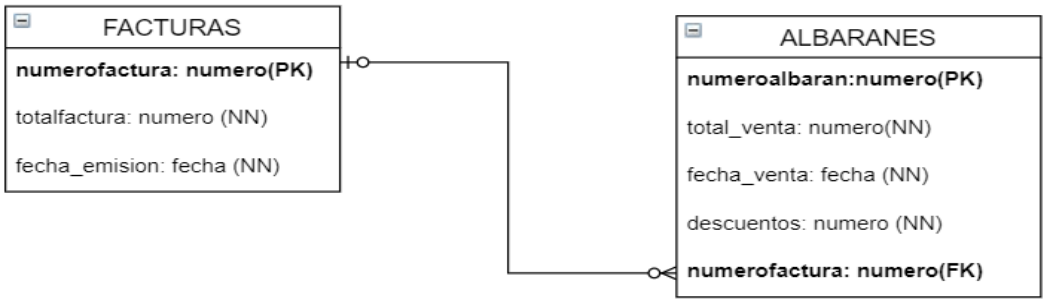
SOLUCIÓN:

```
SHOW CREATE TABLE empleados;
```

2 Lenguaje de definición de datos DDL II

2.1 Introducción a la restricción de clave ajena

2.1.1 Implementa el siguiente diagrama relacional en MySQL (ALBARANES)



- Crea una base de datos llamada **ejercom** y lleva a cabo las siguientes tablas y relaciona las tablas mediante FOREIGN KEY.
- Elige los dominios que creas oportunos para cada uno de los atributos, y el atributo descuento por defecto tendrá un valor de 5%. Recuerda al finalizar comprobar si las has creado dentro de la BD hecha anteriormente.

SOLUCIÓN:

```
CREATE DATABASE IF NOT EXISTS ejercom;
use ejercom;
CREATE TABLE FACTURAS(
    numerofactura INT PRIMARY KEY,
    totalfactura INT NOT NULL,
    fecha_emision DATE
);

CREATE TABLE albaranes(
numalbaran INT PRIMARY KEY,
total_venta INT,
fecha_venta DATE,
descuentos decimal(3,2),
numerofactura INT NOT NULL,
CONSTRAINT  falbaranes FOREIGN KEY (numerofactura)  REFERENCES FACTURAS(numerofactura)
);
```

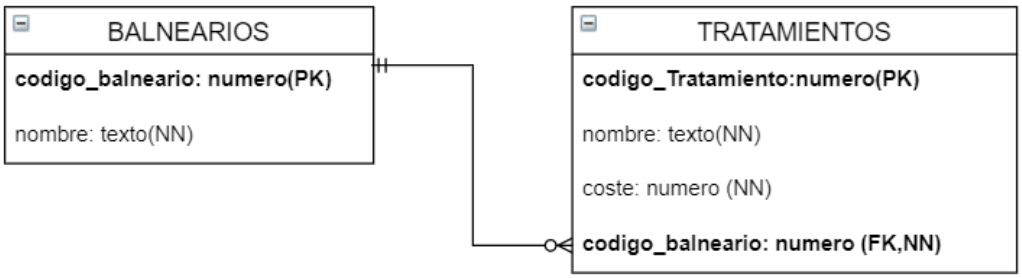
2.1.2 Borra la llave ajena del ejercicio anterior

SOLUCIÓN:

```
ALTER TABLE albaranes DROP FOREIGN KEY falbaranes;
```

2.2 Implementación de Relaciones 1:N

2.2.1 Implementa el siguiente diagrama relacional en MySQL (BALNEARIOS)

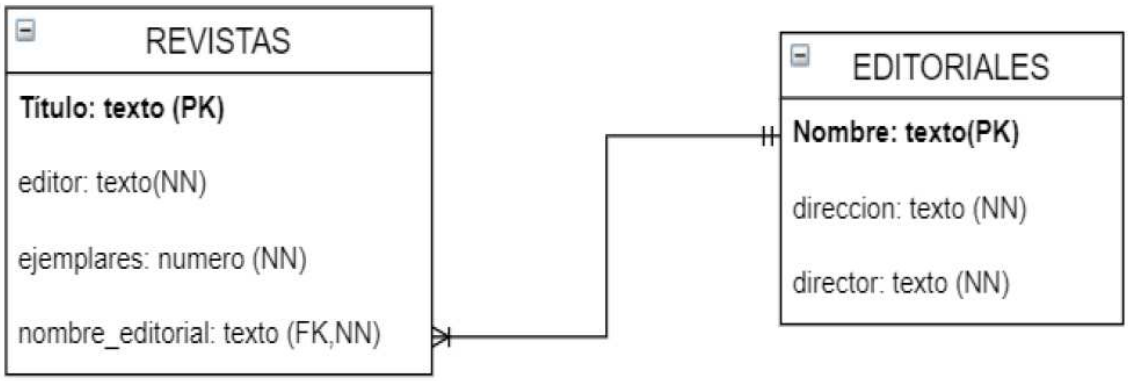


NOTA: El coste será un entero sin signo

SOLUCIÓN:

```
use ejercom;
CREATE TABLE balnearios(
codigo_balneario INT PRIMARY KEY,
nombre VARCHAR(30)
);
CREATE TABLE TRATAMIENTOS(
codigo_tratamiento INT PRIMARY KEY,
nombre VARCHAR(30),
coste  INTEGER UNSIGNED,
codigo_balneario INT NOT NULL,
CONSTRAINT fcodigo_balneario FOREIGN KEY (codigo_balneario) REFERENCES balnearios(codigo_balneario)
);
```

2.2.2 Implementa el siguiente diagrama relacional en MySQL (REVISTAS)

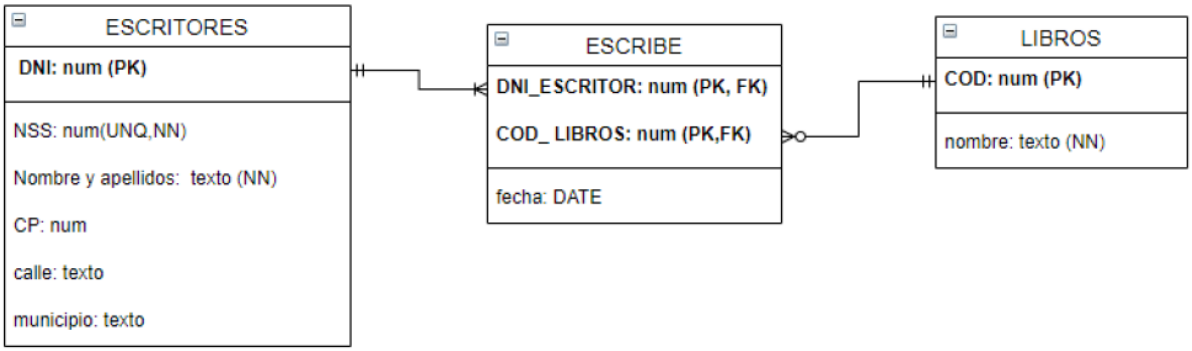


SOLUCIÓN:

```sql
create table editoriales (
    Nombre varchar(20),
    direccion varchar(100) not null,
    director varchar(20) not null,
    primary key (Nombre)
);

create table revistas (
    Titulo varchar(50),
    editor varchar(50) not null,
    ejemplares int not null,
    nombre_editorial varchar(50),
    primary key (Titulo),
    foreign key (nombre_editorial) references editoriales(Nombre)
);
```

2.3 Implementación de Relaciones M:N

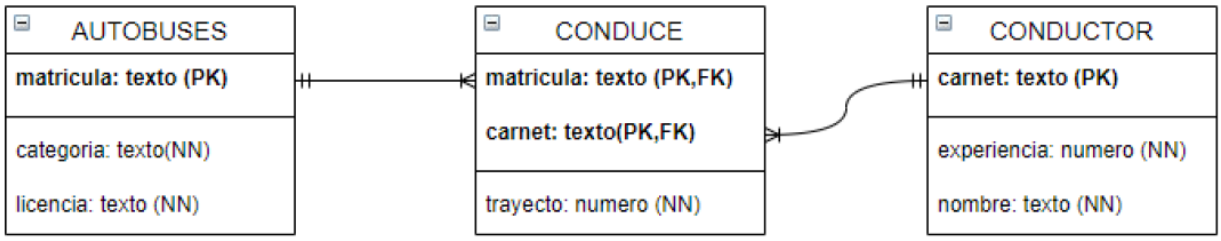
2.3.1 Implementa el siguiente diagrama relacional en MySQL (ESCRITORES – LIBROS)



SOLUCIÓN:

```sql
create table escritores (
    DNI varchar(9),
    NSS varchar(9) not null,
    nombre_y_apellidos varchar(9) not null,
    CP int,
    calle varchar(50),
    municipio varchar(50),
    primary key (DNI),
    unique (NSS)
);

create table libros (
    cod int,
    nombre varchar(100) not null,
    primary key (cod)
);

create table escritores_libros (
    DNI varchar(9),
    cod int,
    fecha date,
    primary key (DNI, cod),
    foreign key (DNI) references escritores(DNI),
    foreign key (cod) references libros(cod)
);
```

2.3.2 Implementa el siguiente diagrama relacional en MySQL (CONDUCTORES)

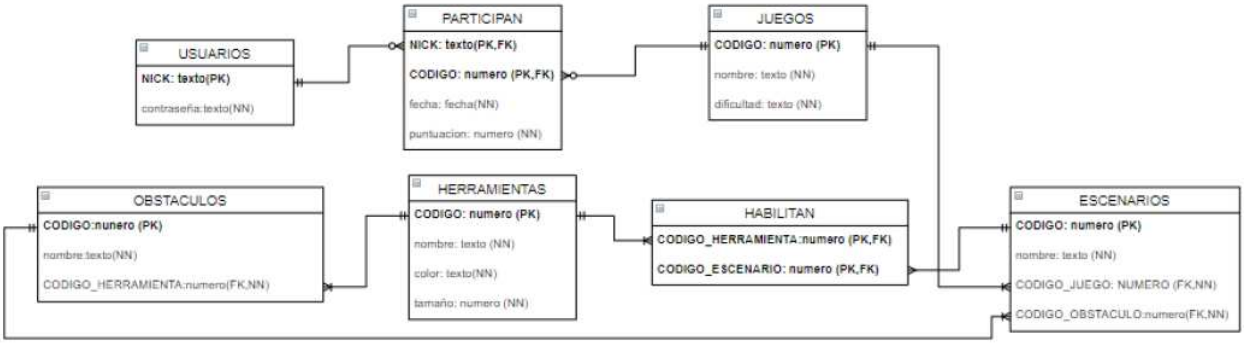


SOLUCIÓN:

```sql
create table autobuses (
    matricula varchar(7),
    categoria varchar(20) not null,
    licencia varchar(50) not null,
    primary key (matricula)
);

create table conductor (
    carnet varchar(20),
    experiencia int not null,
    nombre varchar(50) not null,
    primary key (carnet)
);

create table autobuses_conductor (
    matricula varchar(7),
    carnet varchar(20),
    trayecto int not null,
    primary key (matricula, carnet),
    foreign key (matricula) references autobuses(matricula),
    foreign key (carnet) references conductor(carnet)
);
```

2.4 Implementación de BBDDs I (1:N y M:N)

2.4.1 Implementa el siguiente diagrama relacional en MySQL (VIDEOJUEGOS)

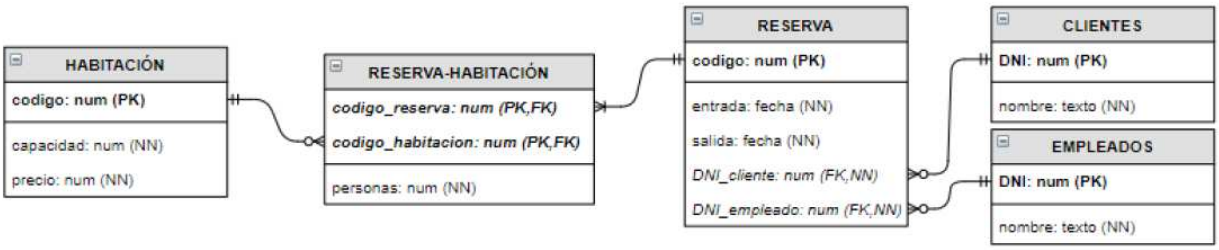


SOLUCIÓN:

```sql
create table usuarios (
    NICK varchar(10),
    contrasenna varchar(20) not null,
    primary key (NICK)
);

create table juegos (
    CODIGO int,
    nombre varchar(20) not null,
    dificultad varchar(15) not null,
    primary key (CODIGO)
);

create table herramientas (
    CODIGO int,
    nombre varchar(20) not null,
    color varchar(10) not null,
    tamano int not null,
    primary key (CODIGO)
);

create table obstaculos (
    CODIGO int,
    nombre varchar(20) not null,
    CODIGO_HERRAMIENTA int not null,
    primary key (CODIGO),
    foreign key (CODIGO_HERRAMIENTA) references herramientas(CODIGO)
);
```

```sql
create table escenarios (
    CODIGO int,
    nombre varchar(20) not null,
    CODIGO_JUEGO int not null,
    CODIGO_OBSTACULO int not null,
    primary key (CODIGO),
    foreign key (CODIGO_JUEGO) references juegos(CODIGO),
    foreign key (CODIGO_OBSTACULO) references obstaculos(CODIGO)
);

create table usuarios_juegos (
    NICK varchar(10),
    CODIGO int,
    fecha date not null,
    puntuacion int not null,
    primary key (NICK, CODIGO),
    foreign key (NICK) references usuarios(NICK),
    foreign key (CODIGO) references juegos(CODIGO)
);

create table herramientas_escenarios (
    CODIGO_HERRAMIENTA int,
    CODIGO_ESCENARIO int,
    primary key (CODIGO_HERRAMIENTA, CODIGO_ESCENARIO),
    foreign key (CODIGO_HERRAMIENTA) references herramientas(CODIGO),
    foreign key (CODIGO_ESCENARIO) references escenarios(CODIGO)
);
```

2.4.2 Implementa el siguiente diagrama relacional en MySQL (HOTEL)



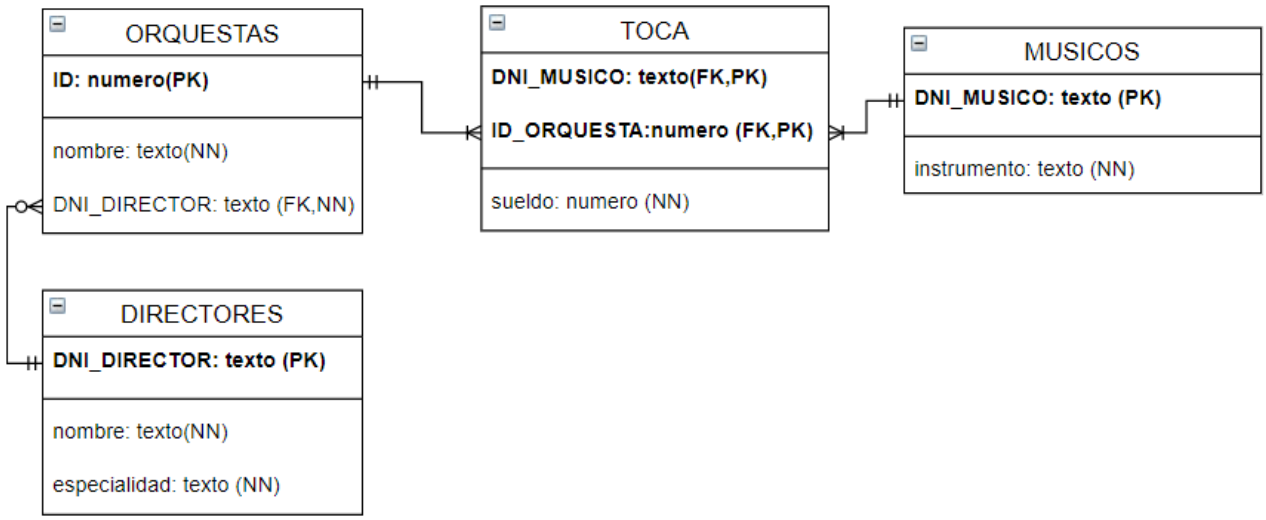
SOLUCIÓN:

```sql
create table habitacion (
    codigo int,
    capacidad int not null,
    precio int not null,
    primary key (codigo)
);

create table clientes (
    DNI int,
    nombre varchar(20) not null,
    primary key (DNI)
);

create table empleados (
    DNI int,
    nombre varchar(20) not null,
    primary key (DNI)
);
```

```sql
create table reserva (
    codigo int,
    entrada date not null,
    salida date not null,
    DNI_cliente int not null,
    DNI_empleado int not null,
    primary key (codigo),
    foreign key (DNI_cliente) references clientes(DNI),
    foreign key (DNI_empleado) references empleados(DNI)
);
```

```sql
create table reserva_habitacion (
    codigo_reserva int,
    codigo_habitacion int,
    personas int not null,
    primary key (codigo_reserva, codigo_habitacion),
    foreign key (codigo_reserva) references reserva(codigo),
    foreign key (codigo_habitacion) references habitacion(codigo)
);
```

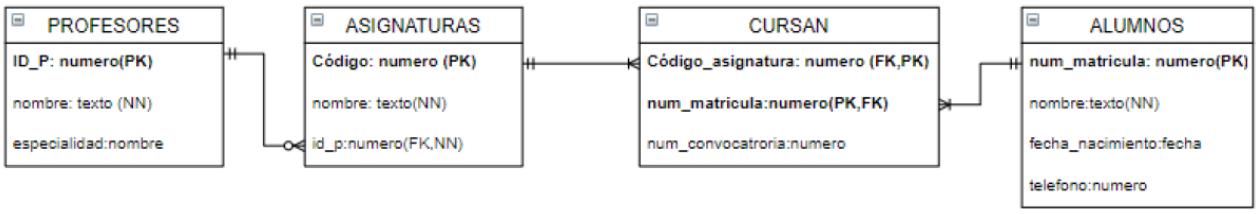
2.4.3 Implementa el siguiente diagrama relacional en MySQL (ORQUESTA)



SOLUCIÓN:

```sql
create table directores (
    DNI_DIRECTORES varchar(9),
    nombre varchar(20) not null,
    especialidad varchar(20) not null,
    primary key (DNI_DIRECTORES)
);

create table orquestas (
    ID int,
    nombre varchar(20) not null,
    DNI_DIRECTOR varchar(9) not null,
    primary key (ID),
    foreign key (DNI_DIRECTOR) references directores(DNI_DIRECTORES)
);

create table musicos (
    DNI_MUSICO varchar(9),
    instrumento varchar (20) not null,
    primary key (DNI_MUSICO)
);

create table orquestas_musicos (
    DNI_MUSICO varchar(9),
    ID_ORQUESTA int,
    sueldo int not null,
    primary key (DNI_MUSICO, ID_ORQUESTA),
    foreign key (DNI_MUSICO) references musicos(DNI_MUSICO),
    foreign key (ID_ORQUESTA) references orquestas(ID)
);
```

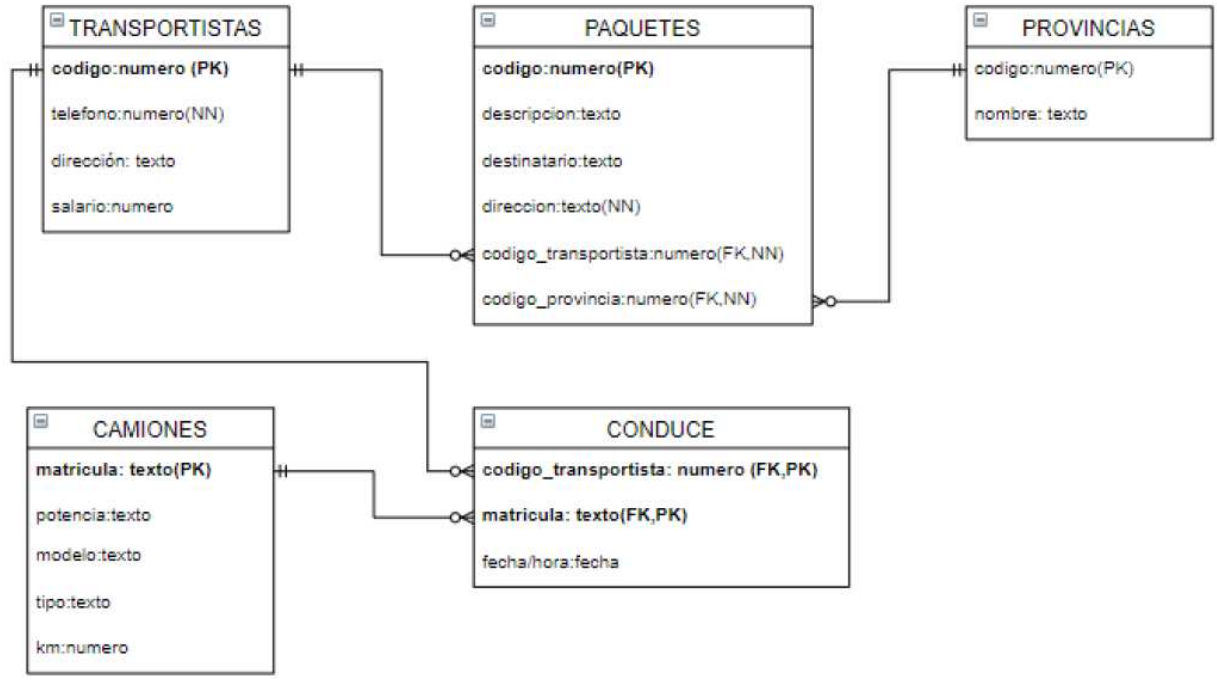
2.4.4 Implementa el siguiente diagrama relacional en MySQL (ESCUELA)



SOLUCIÓN:

```sql
create table profesores(
    ID_P int,
    nombre varchar(50) not null,
    especialidad varchar(30),
    primary key (ID_P)
);

create table asignaturas(
    Codigo int,
    nombre varchar(100) not null,
    id_p int not null,
    primary key (id_p),
    foreign key (id_p) references profesores(ID_P)
);

create table alumnos(
    num_matricula int,
    nombre varchar(50) not null,
    fecha_nacimiento date,
    telefono int,
    primary key (num_matricula)
);

create table alumnos_asignaturas(
    codigo_asignatura int,
    num_matricula int,
    num_convocatoria int,
    primary key (codigo_asignatura, num_matricula),
    foreign key (codigo_asignatura) references asignaturas(Codigo),
    foreign key (num_matricula) references alumno(num_matricula)
);
```

2.4.5 Implementa el siguiente diagrama relacional en MySQL (EMPRESA TRANSPORTE)



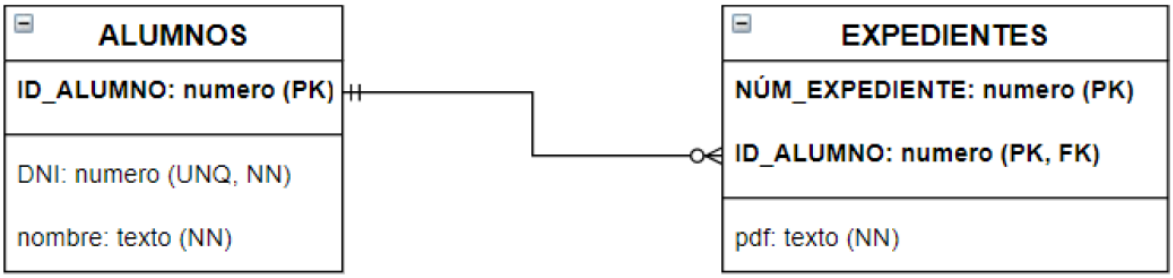
SOLUCIÓN:

```sql
create table camiones(
    matricula varchar(7),
    potencia varchar(10),
    modelo varchar(20),
    tipo varchar(20),
    km int,
    primary key (matricula)
);

create table transportistas(
    codigo int,
    telefono int not null,
    direccion varchar(100),
    salario int,
    primary key (codigo)
);

create table provincias(
    codigo int,
    nombre varchar(20),
    primary key (codigo)
);
```

```sql
create table paquetes(
    codigo int,
    descripcion varchar(200),
    destinatario varchar(100),
    direccion varchar(100) not null,
    codigo_transportista int not null,
    codigo_provincia int not null,
    primary key (codigo),
    foreign key (codigo_transportista) references transportistas(codigo),
    foreign key (codigo_provincia) references provincias(codigo)
);

create table transportistas_camiones(
    codigo_transportista int,
    matricula varchar(7),
    fecha_hora date,
    primary key (codigo_transportista, matricula),
    foreign key (codigo_transportista) references transportista(codigo),
    foreign key (matricula) references camiones(matricula)
);
```

2.5 Implementación de entidades débiles

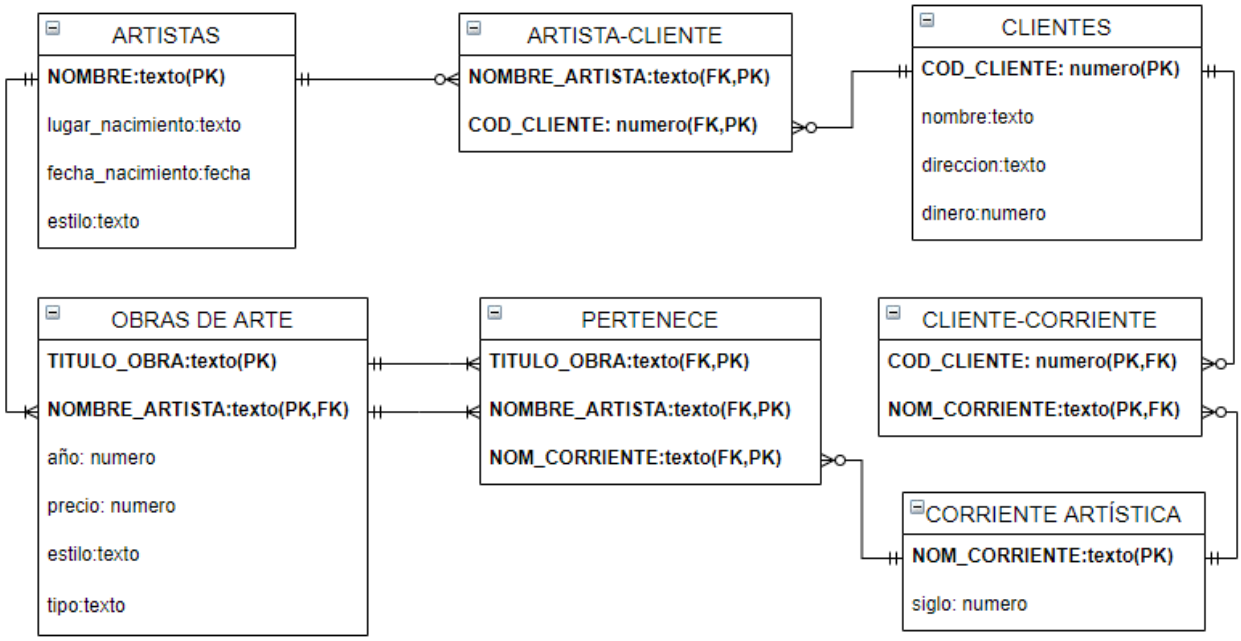
2.5.1 Implementa el siguiente diagrama relacional en MySQL (ALUMNOS – EXPEDIENTES)



SOLUCIÓN:

```sql
create table alumnos(
    ID_ALUMNO int,
    DNI int unique,
    nombre varchar(50),
    primary key (ID_ALUMNO)
);

create table expedientes(
    NUM_EXPEDIENTE int,
    ID_ALUMNO int,
    pdf varchar(200),
    primary key (NUM_EXPEDIENTE, ID_ALUMNO),
    foreign key (ID_ALUMNO) references alumnos(ID_ALUMNO)
);
```

2.5.2 Implementa el siguiente diagrama relacional en MySQL (GALERÍA DE ARTE)



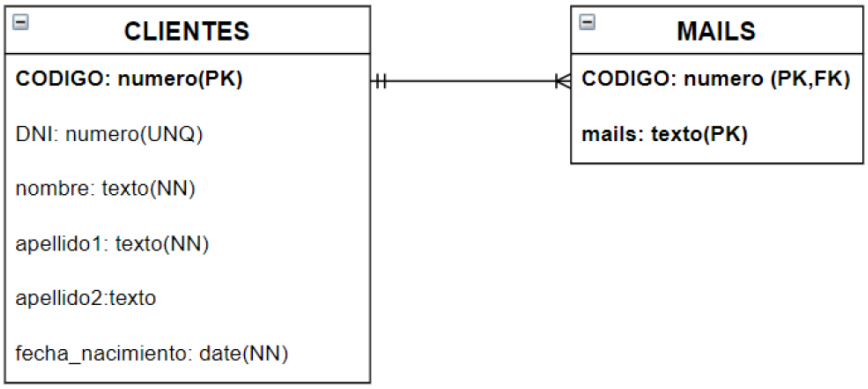
SOLUCIÓN:

```sql
create table artistas (
    NOMBRE varchar(50),
    lugar_nacimiento varchar(100),
    fecha_nacimiento date,
    estilo varchar(100),
    primary key (NOMBRE)
);

create table obras_de_arte (
    TITULO_OBRA varchar(100),
    NOMBRE_ARTISTA varchar(50),
    anno year,
    precio int,
    estilo varchar(100),
    tipo varchar(100),
    primary key (TITULO_OBRA, NOMBRE_ARTISTA),
    foreign key (NOMBRE_ARTISTA) references artistas(NOMBRE)
);

create table clientes (
    COD_CLIENTE int,
    nombre varchar(50),
    direccion varchar(100),
    dinero int,
    primary key (COD_CLIENTE)
);
```

```sql
create table corriente_artistica (
    NOM_CORRIENTE varchar(20),
    siglo int,
    primary key (NOM_CORRIENTE)
);

create table artista_cliente (
    NOMBRE_ARTISTA varchar(50),
    COD_CLIENTE int,
    primary key (NOMBRE_ARTISTA, COD_CLIENTE),
    foreign key (NOMBRE_ARTISTA) references artistas(NOMBRE),
    foreign key (COD_CLIENTE) references clientes(COD_CLIENTE)
);

create table cliente_corriente (
    COD_CLIENTE int,
    NOM_CORRIENTE varchar(20),
    primary key (COD_CLIENTE, NOM_CORRIENTE),
    foreign key (COD_CLIENTE) references clientes(COD_CLIENTE),
    foreign key (NOM_CORRIENTE) references corriente_artistica(NOM_CORRIENTE)
);

create table pertenece (
    TITULO_OBRA varchar(100),
    NOMBRE_ARTISTA varchar(50),
    NOM_CORRIENTE varchar(20),
    primary key (TITULO_OBRA, NOMBRE_ARTISTA, NOM_CORRIENTE),
    foreign key (TITULO_OBRA) references obras_de_arte(TITULO_OBRA),
    foreign key (NOMBRE_ARTISTA) references obras_de_arte(NOMBRE_ARTISTA),
    foreign key (NOM_CORRIENTE) references corriente_artistica(NOM_CORRIENTE)
);
```

2.6 Implementación de atributos multivaluados

2.6.1 Implementa el siguiente diagrama relacional en MySQL (MAILS)



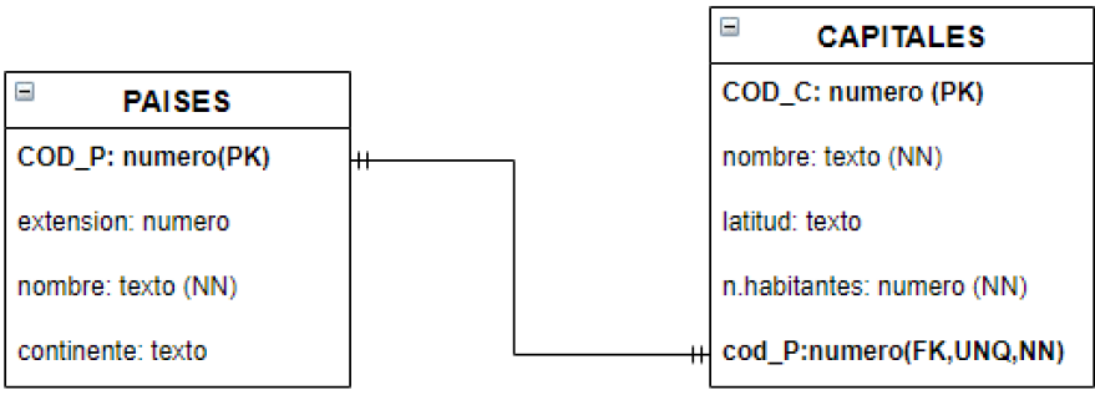
SOLUCIÓN:

```sql
create table clientes (
    CODIGO int,
    DNI int unique,
    nombre varchar(20) not null,
    apellido1 varchar(50) not null,
    apellido2 varchar(50),
    fecha_nacimiento date not null,
    primary key (CODIGO)
);

create table mails (
    CODIGO int,
    mails varchar(1000),
    primary key (CODIGO, mails),
    foreign key (CODIGO) references clientes(CODIGO)
);
```

2.7 Implementación de relaciones uno a uno

2.7.1 Ambas entidades participan con cardinalidad (1, 1)

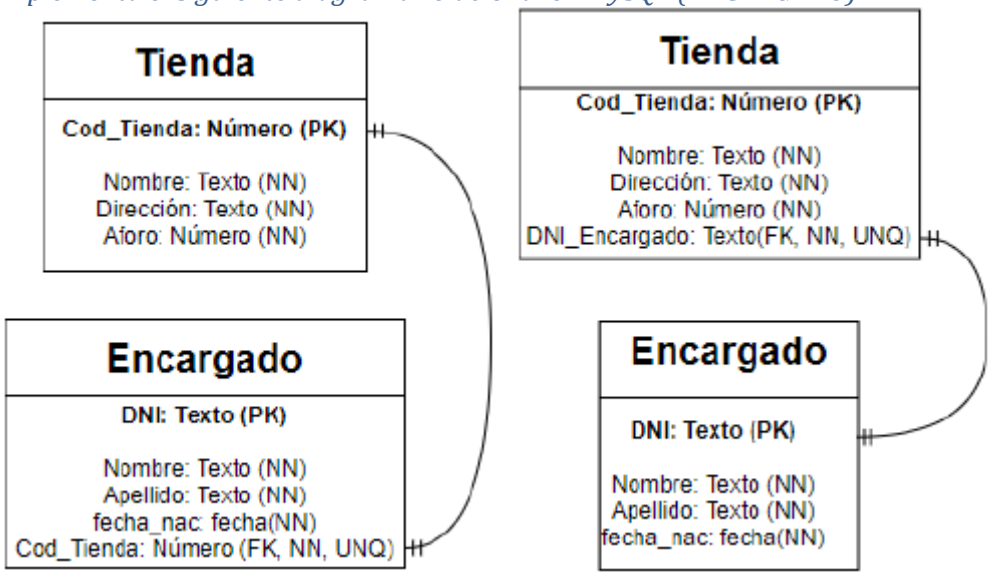
2.7.1.1 Implementa el siguiente diagrama relacional en MySQL (PAÍSES – CAPITALES)



SOLUCIÓN:

```sql
create table paises (
    COD_P int,
    extension int,
    nombre varchar(50) not null,
    continente varchar(20),
    primary key (COD_P)
);

create table capitales (
    COD_C int,
    nombre varchar(200) not null,
    latitud varchar(8),
    num_habitantes int not null,
    COD_P int not null unique,
    primary key (COD_C),
    foreign key (COD_P) references paises(COD_P)
);
```

2.7.1.2 Implementa el siguiente diagrama relacional en MySQL (ENCARGADO)

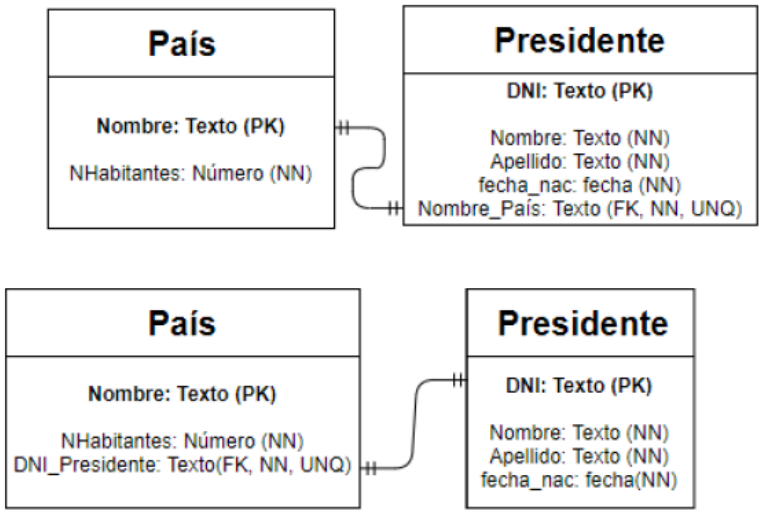


SOLUCIÓN:

```
create table tienda (
    Cod_Tienda int,
    nombre varchar(100) not null,
    direccion varchar(100) not null,
    aforo int not null,
    primary key (Cod_Tienda)
);

create table encargado (
    DNI varchar(9) not null,
    nombre varchar(50) not null,
    apellido varchar(100) not null,
    fecha_nac date not null,
    Cod_Tienda int not null unique,
    primary key (DNI),
    foreign key (Cod_Tienda) references tienda(Cod_Tienda)
);
```

```
create table tienda (
    Cod_Tienda int,
    nombre varchar(100) not null,
    direccion varchar(100) not null,
    aforo int not null,
    DNI varchar(9) not null unique,
    primary key (Cod_Tienda),
    foreign key (DNI) references encargado(DNI)
);

create table encargado (
    DNI varchar(9) not null,
    nombre varchar(50) not null,
    apellido varchar(100) not null,
    fecha_nac date not null,
    primary key (DNI)
);
```

2.7.1.3 Implementa el siguiente diagrama relacional en MySQL (PRESIDENTE)



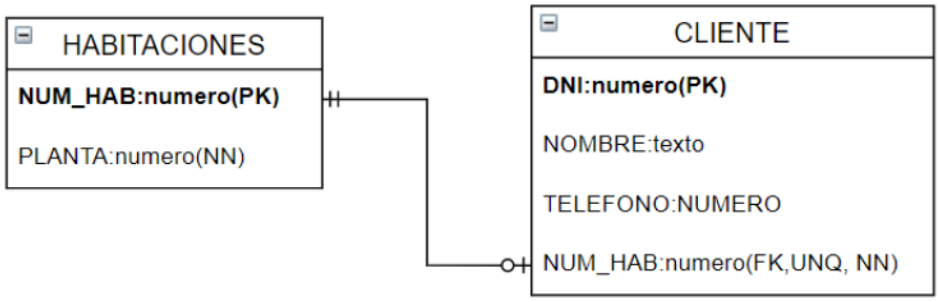
SOLUCIÓN:

```sql
create table pais (
    nombre varchar(100),
    NHabitantes int not null,
    primary key (nombre)
);

create table presidente (
    DNI varchar(9),
    nombre varchar(20) not null,
    apeliido varchar(50) not null,
    fecha_nac date not null,
    nombre_pais varchar(100) not null unique,
    primary key (DNI),
    foreign key (nombre_pais) references pais(nombre)
);
```

```sql
create table pais (
    nombre varchar(100),
    NHabitantes int not null,
    DNI_presidente varchar(9),
    primary key (nombre),
    foreign key (DNI_presidente) references presidente(DNI)
);

create table presidente (
    DNI varchar(9),
    nombre varchar(20) not null,
    apeliido varchar(50) not null,
    fecha_nac date not null,
    primary key (DNI)
);
```

2.7.2 Una entidad participa con cardinalidad (0, 1)

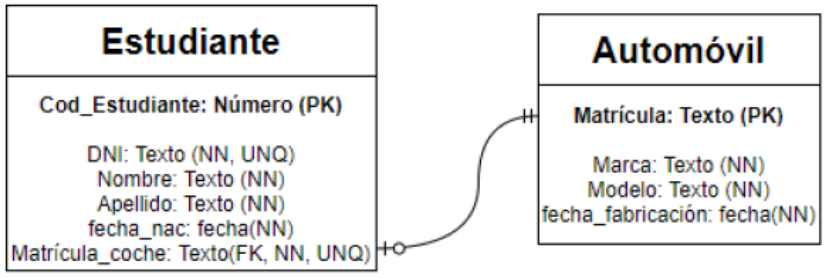
2.7.2.1 Implementa el siguiente diagrama relacional en MySQL (HABITACIÓN)



SOLUCIÓN:

```sql
create table habitaciones (
    num_hab int,
    plante int not null,
    primary key (num_hab)
);

create table cliente (
    DNI int,
    nombre varchar(20),
    telefono int,
    num_hab int not null unique,
    primary key (DNI),
    foreign key (num_hab) references habitaciones(num_hab)
);
```

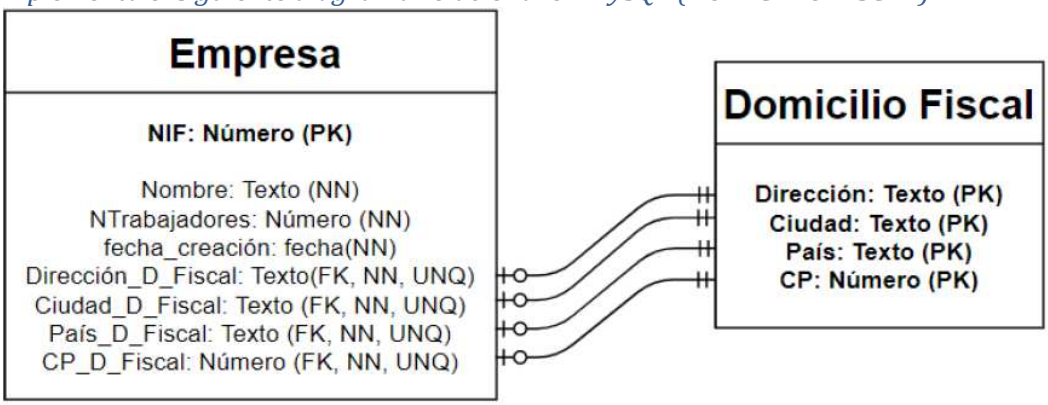
2.7.2.2 Implementa el siguiente diagrama relacional en MySQL (AUTOMÓVIL – ESTUDIANTE)



SOLUCIÓN:

```sql
create table automovil (
    matricula varchar(7),
    marca varchar(50) not null,
    modelo varchar(50) not null,
    fecha_fabricacion date not null,
    primary key (matricula)
);

create table estudiante (
    Cod_estudiante int,
    DNI varchar(9) not null unique,
    nombre varchar(20) not null,
    apellido varchar(50) not null,
    fecha_nac date not null,
    matricula_coche varchar(7) not null unique,
    primary key (Cod_estudiante),
    foreign key (matricula_coche) references automovil(matricula)
);
```

2.7.2.3 Implementa el siguiente diagrama relacional en MySQL (DOMICILIO FISCAL)

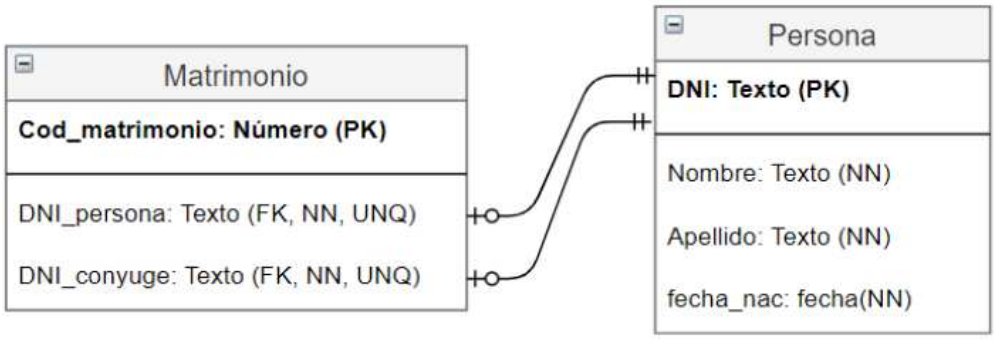


SOLUCIÓN:

```sql
create table domicilio_fiscal (
    direccion varchar(100),
    ciudad varchar(50),
    pais varchar(50),
    CP int,
    primary key (direccion, ciudad, pais, cp)
);

create table empresa (
    NIF int,
    nombre varchar(50) not null,
    ntrabajadores int not null,
    fecha_creacion date not null,
    direccion_D_fiscal varchar(100),
    ciudad_D_fiscal varchar(50),
    pais_D_fiscal varchar(50),
    CP_D_fiscal int,
    primary key (NIF),
    foreign key (direccion_D_fiscal) references domicilio_fiscal(direccion),
    foreign key (ciudad_D_fiscal) references domicilio_fiscal (ciudad),
    foreign key (pais_D_fiscal) references domicilio_fiscal(pais),
    foreign key (CP_D_fiscal) references domicilio_fiscal(cp)
);
```

2.7.3 Ambas entidades participan con cardinalidad (0, 1)

2.7.3.1 Implementa el siguiente diagrama relacional en MySQL (Matrimonio)



SOLUCIÓN:

```
create table persona (
    DNI varchar(9),
    nombre varchar(20) not null,
    apellido varchar(50) not null,
    fecha_nac date not null,
    primary key (DNI)
);

create table Matrimonio (
    Cod_matrimonio int,
    DNI_Persona varchar(9) not null unique,
    DNI_Conyuje varchar(9) not null unique,
    primary key (Cod_matrimonio),
    foreign key (DNI_Persona) references persona(DNI),
    foreign key (DNI_Conyuje) references persona(DNI)
);
```

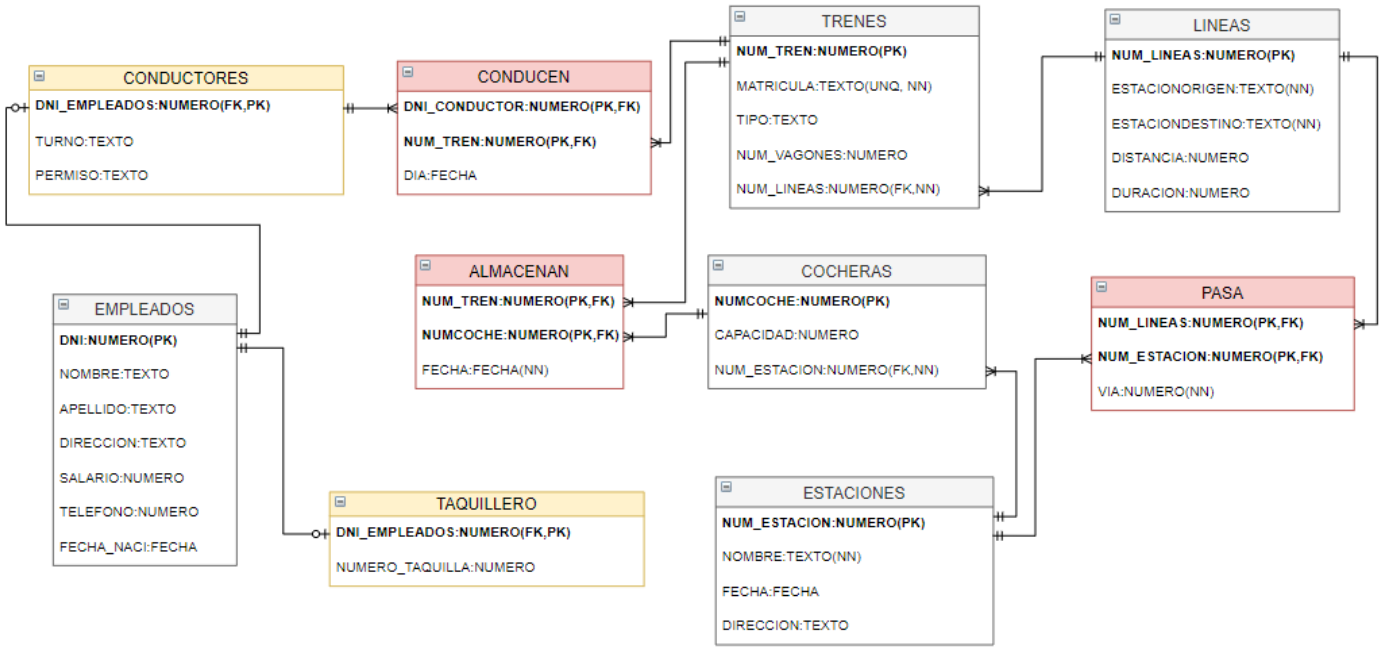
3 Lenguaje de definición de datos (DDL) III

3.1 Restricciones CHECK y AUTOINCREMENT

3.1.1 Implementa el siguiente diagrama relacional en MySQL (TRENES)

Implementa el siguiente esquema en MySQL teniendo en cuenta las siguientes condiciones:

- El atributo **num_tren** de la tabla trenes tiene que ser asignado de manera automática y de forma incremental. El atributo **tipo** de dicha tabla solo puede tomar dos valores: doble y sencillo.
- El atributo **turno** de la tabla conductores tiene que tomar solo 3 valores: noche, día, partido.
- El atributo **num_estacion** de la tabla estaciones no puede ser menor que 1 ni mayor que 20



SOLUCIÓN:

```
CREATE TABLE EMPLEADOS(
DNI VARCHAR(9),
NOMBRE VARCHAR(20) NOT NULL,
APELLIDOS VARCHAR(40) NOT NULL,
DIRECCION VARCHAR(40),
TELEFONO INT,
FECHA_NACIMIENTO DATE,
PRIMARY KEY(DNI)
);
 CREATE TABLE CONDUCTORES(
DNI_CONDUCTOR VARCHAR(9),
TURNO ENUM ('NOCHE','DIA','PARTIDO'),
PRIMARY KEY(DNI_CONDUCTOR),
FOREIGN KEY(DNI_CONDUCTOR) REFERENCES EMPLEADOS(DNI)
);
 CREATE TABLE TAQUILLERO(
DNI_TAQUILLERO VARCHAR(9),
NUM_TAQUILLA INT,
PRIMARY KEY(DNI_TAQUILLERO),
FOREIGN KEY(DNI_TAQUILLERO) REFERENCES EMPLEADOS(DNI)
);
```

```
CREATE TABLE LINEAS(
   NUM_LINEAS INT,
   ESTANCION_ORIGEN VARCHAR(20),
   ESTACION_DESTINO VARCHAR(20),
   DISTANCIA INT,
   DURACION INT,
   primary key(NUM_LINEAS)
);
CREATE TABLE TRENES(
   NUM_TRENES INT PRIMARY KEY AUTO_INCREMENT,
   MATRICULA VARCHAR(7) UNIQUE,
   TIPO ENUM('DOBLE', 'SENCILLO'),
   NUM_VAGONES INT,
   CODIGO_LINEAS INT,
   FOREIGN KEY (CODIGO_LINEAS) REFERENCES LINEAS(NUM_LINEAS)
);
```

```
CREATE TABLE CONDUCEN(
   DNI_CONDUCTOR VARCHAR(9),
   NUM_TRENES INT,
   DIA INT,
   PRIMARY KEY (DNI_CONDUCTOR, NUM_TRENES),
   FOREIGN KEY (DNI_CONDUCTOR) REFERENCES CONDUCTORES(DNI_CONDUCTOR),
   FOREIGN KEY (NUM_TRENES) REFERENCES TRENES(NUM_TRENES)
);
CREATE TABLE ESTACIONES(
   NUM_ESTACION INT,
   NOMBRE VARCHAR(20),
   FECHA DATE,
   DIRECCION VARCHAR(50),
   PRIMARY KEY(NUM_ESTACION),
   CONSTRAINT ck_ESTACIONES CHECK (NUM_ESTACION > 1 AND NUM_ESTACION<20)
    );
 CREATE TABLE ALMACENAN(
    NUM_COCHE INT,
    NUM_TREN INT,
    FECHA DATE,
    PRIMARY KEY (NUM_COCHE,NUM_TREN),
    FOREIGN KEY (NUM_COCHE) REFERENCES COCHERAS(NUM_COCHE),
    FOREIGN KEY (NUM_TREN) REFERENCES TRENES(NUM_TRENES)
 );
```

```
CREATE TABLE PASA(
   NUM_ESTACION INT,
   NUM_LINEAS INT,
   VIA INT,
   PRIMARY KEY (NUM_ESTACION, NUM_LINEAS),
   FOREIGN KEY (NUM_ESTACION) REFERENCES ESTACIONES(NUM_ESTACION),
   FOREIGN KEY (NUM_LINEAS) REFERENCES LINEAS(NUM_LINEAS)
);
CREATE TABLE COCHERAS(
   NUM_COCHE INT PRIMARY KEY,
   CAPACIDAD INT,
   NUM_ESTACION INT,
   FOREIGN KEY (NUM_ESTACION) REFERENCES ESTACIONES(NUM_ESTACION)
);
```

*3.1.1.1 Modifica el ejercicio TRENES para que el atributo **num_lineas** de la tabla **LINEAS** tenga la siguiente restricción.*

Modifica el ejercicio TRENES para que el atributo **num_lineas** de la tabla **LINEAS** tenga la siguiente restricción: **que sea mayor o igual que uno y menor o igual que 25**

SOLUCIÓN:

```
alter table LINEAS ADD CONSTRAINT CK_LINEAS CHECK(NUM_LINEAS >=1 AND NUM_LINEAS <=25);
```

*3.1.1.2 Modifica el ejercicio TRENES para que el atributo **DIA** de la tabla **CONDUCEN** sea de tipo **DATE***

SOLUCIÓN:

```
ALTER TABLE CONDUCEN MODIFY COLUMN DIA DATE;
```

*3.1.1.3 Modifica el ejercicio TRENES para que el atributo **DIA** siempre sea mayor o igual que el **2020-05-25***

SOLUCIÓN:

```
ALTER TABLE CONDUCEN ADD CONSTRAINT CK_CONDUCEN CHECK (DIA >= '2020-05-25');
```

3.1.1.4 Restricciones con nombre y acciones ante una infracción de integridad en tabla ALMACENAN.

Haz lo que sea necesario para que las claves ajenas y primarias de almacenan tengan nombre y para que las claves ajenas tengan acciones ante una infracción de integridad para que al actualizar o borrar lo hagan en cascada.

SOLUCIÓN:

```
DROP TABLE ALMACENAN;
  CREATE TABLE ALMACENAN(
        NUM_COCHE INT,
        NUM_TREN INT,
        FECHA DATE);
ALTER TABLE ALMACENAN ADD CONSTRAINT PK_ALMACENAN primary key(NUM_COCHE, NUM_TREN);
ALTER TABLE ALMACENAN ADD CONSTRAINT FK_COCHE foreign key (NUM_COCHE) REFERENCES COCHERAS(NUM_COCHE) ON UPDATE CASCADE ON DELETE CASCADE;
ALTER TABLE ALMACENAN ADD CONSTRAINT FK_TREN foreign key (NUM_TREN) REFERENCES TRENES(NUM_TRENES) ON UPDATE CASCADE ON DELETE CASCADE;
```

*3.1.1.5 Borra la tabla **ALMACENAN** para eliminar la relación M:N*

SOLUCIÓN:

```
DROP TABLE ALMACENAN
```

*3.1.1.6 Modifica la tabla **TRENES** y añade una columna **NUM_COCHE** que será un INT*

SOLUCIÓN:

```
ALTER TABLE TRENES ADD COLUMN NUM_COCHE INT
```

*3.1.1.7 Modifica la columna anterior (**NUM_COCHE** de **TRENES**) para que sea NOT NULL*

SOLUCIÓN:

```
ALTER TABLE TRENES MODIFY COLUMN NUM_COCHE INT NOT NULL
```

*3.1.1.8 Crea una clave ajena sobre el campo anterior (**NUM_COCHE** de **TRENES**) para que referencie a COCHERAS (Nueva relación 1:N)*

SOLUCIÓN:

```
ALTER TABLE TRENES ADD CONSTRAINT fk_trenes_cocheras
    FOREIGN KEY (NUM_COCHE) REFERENCES COCHERAS(NUM_COCHE)
```

3.1.1.9 Modifica la tabla TAQUILLA para que el atributo NUM_TAQUILLA sea incremental. ¿Qué ocurre?

SOLUCIÓN:

```
ALTER TABLE TAQUILLERO MODIFY COLUMN NUM_TAQUILLA INT AUTO_INCREMENT;
```

Da error porque no NUM_TAQUILLA no es llave primaria el AUTO_INCREMENT solo está permitido para la llave primaria.

3.1.1.10Borra la tabla de EMPLEADOS: ¿Qué pasa?

SOLUCIÓN:

```
DROP TABLE EMPLEADOS;
```

Da error no se puede borrar esa tabla porque hay claves foráneas en otras tablas que dependen de su llave primaria.