

# Tema 4.- Diseño Físico

Entrega Ejercicios 3 al 6

Tercero: Instituto (instituto.sql)

```
/* Fichero instituto.sql */  
USE instituto;  
DROP TABLE IF EXISTS Profesores;  
DROP TABLE IF EXISTS Modulos;  
DROP TABLE IF EXISTS Matriculas;  
DROP TABLE IF EXISTS Alumnos;  
DROP TABLE IF EXISTS Grupos;
```

```
CREATE TABLE Profesores(  
    Dni CHAR(9) PRIMARY KEY,  
    Nombre VARCHAR (50),  
    Direccion TEXT,  
    Tfno CHAR(9)  
);
```

```
CREATE TABLE Modulos(  
    Codigo CHAR(8) PRIMARY KEY,  
    Nombre VARCHAR (50),  
    Dni_profesor CHAR(9),  
    Foreign Key (Dni_profesor) REFERENCES Profesores(Dni) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE Grupos(  
    Curso CHAR (1),  
    Letra CHAR (1),  
    Aula CHAR (4),  
    Delegado CHAR (9),  
    PRIMARY KEY(Curso, Letra)  
);
```

```
CREATE TABLE Alumnos(  
    N_expdte CHAR (9) PRIMARY KEY,  
    Nombre VARCHAR (20),  
    Apellidos VARCHAR (40),  
    Fecha_nac DATE,  
    Curso CHAR (1),  
    Letra CHAR (1),
```

```
Foreign Key (Curso, Letra) REFERENCES Grupos(Curso, Letra) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);  
  
ALTER TABLE `grupos` ADD FOREIGN KEY (`Delegado`) REFERENCES  
`alumnos`(`N_expdt`) ON UPDATE CASCADE ON DELETE RESTRICT;
```

```
CREATE TABLE Matriculas(  
    Modulo CHAR (8),  
    Alumno CHAR (9),  
    PRIMARY KEY (Modulo, Alumno),  
    Foreign Key (Modulo) REFERENCES Modulos(Codigo) ON UPDATE CASCADE ON  
DELETE CASCADE,  
    Foreign Key (Alumno) REFERENCES Alumnos(N_expdt) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

#### Cuarto: Liga (liga.sql)

```
/* Fichero liga.sql */  
USE liga;  
DROP TABLE IF EXISTS Presidentes;  
DROP TABLE IF EXISTS Equipos;  
DROP TABLE IF EXISTS Jugadores;  
DROP TABLE IF EXISTS Partidos;  
DROP TABLE IF EXISTS Goles;
```

```
CREATE TABLE Presidentes(  
    Dni CHAR (9) PRIMARY KEY,  
    Nombre VARCHAR (20),  
    Apellidos VARCHAR (40),  
    Fecha_nac DATE NULL,  
    Anno YEAR  
);
```

```
CREATE TABLE Equipos(  
    Codigo CHAR (9) PRIMARY KEY,  
    Nombre VARCHAR (50),  
    Estadio VARCHAR (50),  
    Aforo VARCHAR (4) NULL,  
    Fundacion DATE NULL,  
    Ciudad VARCHAR (20),  
    Presidente CHAR (9),  
    Foreign Key (Presidente) REFERENCES Presidentes(Dni) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE Partidos(  
    Codigo CHAR (8) PRIMARY KEY,  
    Fecha DATE,  
    Equipo_Local CHAR (9),  
    Equipo_Visitante CHAR (9),  
    Foreign Key (Equipo_Local) REFERENCES Equipos(Codigo) ON UPDATE  
CASCADE ON DELETE RESTRICT,  
    Foreign Key (Equipo_Visitante) REFERENCES Equipos(Codigo) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE Jugadores(  
    Codigo CHAR (9) PRIMARY KEY,  
    Nombre VARCHAR (60),  
    Fecha DATE,  
    Posicion ENUM ('portero', 'defensa', 'centrocampista', 'delantero'),  
    Cod_equipo CHAR (9),  
    Foreign Key (Cod_equipo) REFERENCES Equipos(Codigo) ON UPDATE CASCADE  
ON DELETE RESTRICT  
);
```

```
CREATE TABLE Goles(  
    Cod_Partido CHAR (8),  
    Minuto SMALLINT,  
    Descripcion TEXT NULL,  
    Cod_jugador CHAR (9),  
    PropiaPuerta BOOLEAN,  
    PRIMARY KEY (Cod_Partido, Minuto),  
    Foreign Key (Cod_Partido) REFERENCES Partidos(Codigo) ON UPDATE  
CASCADE ON DELETE CASCADE,  
    Foreign Key (Cod_jugador) REFERENCES Jugadores(Codigo) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

### Quinto: Startrek (startrek.sql)

```
/* Fichero startrek.sql */  
  
USE startrek;  
DROP TABLE IF EXISTS Personajes;  
DROP TABLE IF EXISTS Klingons;  
DROP TABLE IF EXISTS Vulcanos;  
DROP TABLE IF EXISTS Humanos;  
DROP TABLE IF EXISTS Actores;  
DROP TABLE IF EXISTS Depende;  
DROP TABLE IF EXISTS Aparece;  
DROP TABLE IF EXISTS Peliculas;  
DROP TABLE IF EXISTS Participa;
```

```
DROP TABLE IF EXISTS Capítulos;  
DROP TABLE IF EXISTS Visitas;  
DROP TABLE IF EXISTS Planetas;  
DROP TABLE IF EXISTS Naves;  
DROP TABLE IF EXISTS Lanzaderas;
```

```
CREATE TABLE Personajes(  
    Nombre VARCHAR(20),  
    Grado ENUM ('capitán','teniente','almirante'),  
    PRIMARY KEY(Nombre)  
);
```

```
CREATE TABLE Depende(  
    Mando VARCHAR(20),  
    Subordinado VARCHAR(20),  
    PRIMARY KEY(Mando,Subordinado),  
    FOREIGN KEY (Mando) REFERENCES Personajes(Nombre) ON UPDATE CASCADE ON  
DELETE CASCADE,  
    FOREIGN KEY (Subordinado) REFERENCES Personajes(Nombre) ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```

```
CREATE TABLE Actores(  
    Codigo CHAR(10),  
    Nombre VARCHAR(20),  
    Nacionalidad VARCHAR(20),  
    Fecha_nac DATE,  
    Personaje VARCHAR(20),  
    PRIMARY KEY(Codigo),  
    FOREIGN KEY (Personaje) REFERENCES Personajes(Nombre) ON UPDATE CASCADE  
ON DELETE RESTRICT  
);
```

```
CREATE TABLE Películas(  
    Título VARCHAR(20),  
    Director VARCHAR(20),  
    Año CHAR(10),  
    Protagonista VARCHAR(20),  
    PRIMARY KEY(Título),  
    FOREIGN KEY (Protagonista) REFERENCES Personajes(Nombre) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE Aparece(  
    Pelicula VARCHAR(20),  
    Personaje VARCHAR(20),
```

```
PRIMARY KEY(Pelicula, Personaje),  
FOREIGN KEY (Pelicula) REFERENCES Películas(Titulo) ON UPDATE CASCADE  
ON DELETE CASCADE,  
FOREIGN KEY (Personaje) REFERENCES Personajes(Nombre) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Capitulos(  
    Temporada INTEGER,  
    Orden VARCHAR (20),  
    Titulo VARCHAR (20),  
    Fecha DATE,  
    PRIMARY KEY(Temporada, Orden)  
);
```

```
CREATE TABLE Participa(  
    Personaje VARCHAR(20),  
    Cap_temporada INTEGER,  
    Cap_orden VARCHAR(20),  
    PRIMARY KEY(Personaje, Cap_temporada, Cap_orden)  
);
```

```
CREATE TABLE Planetas(  
    Codigo CHAR(10),  
    Nombre VARCHAR (20),  
    Galaxia VARCHAR (20),  
    Problema TEXT,  
    PRIMARY KEY(Codigo)  
);
```

```
CREATE TABLE Naves(  
    Codigo CHAR(10),  
    N_tripulantes VARCHAR(20),  
    PRIMARY KEY(Codigo)  
);
```

```
CREATE TABLE Visitas(  
    Codigo CHAR(10),  
    Cap_temporada INTEGER,  
    Cap_orden VARCHAR(20),  
    Cod_planeta CHAR(20),  
    Cod_nave CHAR(20),  
    PRIMARY KEY(Codigo)  
);
```

```
CREATE TABLE Lanzaderas(  
  Cod_nave CHAR(10),  
  Numero VARCHAR(20),  
  Capacidad VARCHAR(20),  
  PRIMARY KEY(Cod_nave,Numero),  
  FOREIGN KEY (Cod_nave) REFERENCES Naves(Codigo) ON UPDATE CASCADE ON  
DELETE CASCADE  
);
```

```
CREATE TABLE Humanos(  
  Personaje VARCHAR(20),  
  Fecha_nac VARCHAR(20),  
  Ciudad VARCHAR(20),  
  PRIMARY KEY(Personaje),  
  FOREIGN KEY (Personaje) REFERENCES Personajes(Nombre) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Vulcanos(  
  Personaje VARCHAR(20),  
  Fecha_grad VARCHAR(20),  
  Mentor VARCHAR(20),  
  PRIMARY KEY(Personaje),  
  FOREIGN KEY (Personaje) REFERENCES Personajes(Nombre) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

```
CREATE TABLE Klingons(  
  Personaje VARCHAR(20),  
  Fecha_combate VARCHAR(20),  
  Planeta VARCHAR(20),  
  PRIMARY KEY(Personaje),  
  FOREIGN KEY (Personaje) REFERENCES Personajes(Nombre) ON UPDATE CASCADE  
ON DELETE CASCADE  
);
```

## Sexto: Super (super.sql)

```
/* Fichero super.sql */  
  
USE super;  
DROP TABLE IF EXISTS Zonas;  
DROP TABLE IF EXISTS Repartidores;  
DROP TABLE IF EXISTS Clientes;  
DROP TABLE IF EXISTS Pedidos;  
DROP TABLE IF EXISTS Categorías;  
DROP TABLE IF EXISTS Productos;
```

```
DROP TABLE IF EXISTS Incluye;
```

```
CREATE TABLE Zonas(  
    Codigo_postal CHAR(7),  
    PRIMARY KEY(Codigo_postal)  
);
```

```
CREATE TABLE Clientes(  
    Dni CHAR(9),  
    Nombre VARCHAR(20),  
    Direccion VARCHAR(40),  
    Tfno VARCHAR(20),  
    Mail VARCHAR(20),  
    Password CHAR(15),  
    Zona VARCHAR(50),  
    PRIMARY KEY(Dni),  
    FOREIGN KEY(Zona) REFERENCES Zonas(Codigo_postal) ON UPDATE CASCADE  
ON DELETE RESTRICT  
);
```

```
CREATE TABLE Pedidos(  
    Codigo CHAR(10),  
    Fecha DATE,  
    Direccion VARCHAR(40),  
    Importe BOOLEAN,  
    Tarjeta BOOLEAN,  
    Fecha_cad_tjta DATE,  
    Dni_cliente CHAR(9),  
    PRIMARY KEY(Codigo),  
    FOREIGN KEY(Dni_cliente) REFERENCES Clientes(Dni) ON UPDATE CASCADE  
ON DELETE RESTRICT  
);
```

```
CREATE TABLE Repartidores(  
    Codigo CHAR(10),  
    Nombre VARCHAR(20),  
    Matricula VARCHAR(20),  
    Zona VARCHAR(50),  
    PRIMARY KEY(Codigo),  
    FOREIGN KEY(Zona) REFERENCES Zonas(Codigo_postal) ON UPDATE CASCADE  
ON DELETE RESTRICT  
);
```

```
CREATE TABLE Categorias(  
    Nombre VARCHAR(20),  
    Observaciones TEXT,  
    Condiciones ENUM ('frío', 'congelado', 'seco'),  
    PRIMARY KEY(Nombre)  
);
```

```
CREATE TABLE Productos(  
    Nombre VARCHAR(20),  
    Marca VARCHAR(20),  
    Origen CHAR(20),  
    Volumen VARCHAR(20),  
    Peso INTEGER,  
    Foto BOOLEAN,  
    Caracteristicas TEXT,  
    Stock VARCHAR(20),  
    Cod_categoria CHAR(10),  
    PRIMARY KEY(Nombre,Marca),  
    FOREIGN KEY(Cod_categoria) REFERENCES Categorias(Nombre) ON UPDATE  
CASCADE ON DELETE RESTRICT  
);
```

```
CREATE TABLE Incluye(  
    Cod_pedido CHAR(10),  
    Prod_nombre VARCHAR(20),  
    Prod_marca VARCHAR(20),  
    Cantidad INTEGER,  
    PRIMARY KEY(Cod_pedido,Prod_nombre,Prod_marca),  
    FOREIGN KEY(Cod_pedido) REFERENCES Pedidos(Codigo) ON UPDATE CASCADE  
ON DELETE CASCADE,  
    FOREIGN KEY(Prod_nombre) REFERENCES Productos(Nombre) ON UPDATE  
CASCADE ON DELETE CASCADE  
);
```