

# DDL

1.

```
Estación(Identificador, Latitud, Longitud, Altitud)
Muestra(IdentificadorEstacion, Fecha, Temperatura mínima, Temperatura máxima,
Precipitaciones, Humedad mínima, Humedad máxima, Velocidad del viento mínima,
Velocidad del viento máxima)
```

create database meteorologia;  
use meteorologia;

create table estacion(id varchar(40), latitud decimal, longitud decimal, altitud decimal,  
primary key(id));

create table provincia(id\_estacion varchar(40), fecha date, temp\_min decimal, temp\_max  
decimal, precipitaciones boolean, hum\_min int, hum\_max int, vel\_vient\_min int,  
vel\_vient\_max int, foreign key (id\_estacion ) references estacion(id) on delete cascade on  
update cascade, PRIMARY KEY(fecha));

2.

Columna	Tipo
ClaveLibro	INT
Título	VARCHAR(60)
Idioma	VARCHAR(15)
Formato	VARCHAR(15)
ClaveTema	SMALLINT
Nombre(tema)	VARCHAR(40)
ClaveAutor	INT
Nombre(autor)	VARCHAR(60)
ClaveEditorial	SMALLINT
Nombre(editorial)	VARCHAR(60)
Dirección(editorial)	VARCHAR(60)
Teléfono(editorial)	VARCHAR(15)
ClaveEjemplar	INT
NúmeroOrden	SMALLINT
Edición	SMALLINT
Ubicación	VARCHAR(15)
Categoría(libro)	CHAR
ClaveSocio	INT
Nombre(socio)	VARCHAR(60)
Dirección(socio)	VARCHAR(60)
Teléfono(socio)	VARCHAR(15)
Categoría(socio)	CHAR
Fecha_prestamo	DATE
Fecha_devolucion	DATE
Notas	BLOB

El segundo ejemplo consiste en modelar una biblioteca. Este era el esquema:

```
Libro(ClaveLibro, Título, Idioma, Formato, ClaveEditorial)
Tema(ClaveTema, Nombre)
Autor(ClaveAutor, Nombre)
Editorial(ClaveEditorial, Nombre, Dirección, Teléfono)
Ejemplar(ClaveEjemplar, ClaveLibro, NúmeroOrden, Edición, Ubicación, Categoría)
Socio(ClaveSocio, Nombre, Dirección, Teléfono, Categoría)
Préstamo(ClaveSocio, ClaveEjemplar, NúmeroOrden, Fecha_prestamo,
Fecha_devolucion, Notas)
Trata_sobre(ClaveLibro, ClaveTema)
Escrito_por(ClaveLibro, ClaveAutor)
```

create database biblioteca;  
use biblioteca;

create table editorial(id smallint, nombre varchar(60), direccion varchar(60), telefono  
varchar(15), PRIMARY KEY(id));

```
create table libro(id varchar(60), titulo varchar(15), idioma varchar(15), formato smallint,  
id_editorial smallint, foreign key (id_editorial ) references editorial(id) on delete cascade on  
update cascade, PRIMARY KEY(id));
```

```
create table tema(id smallint, nombre varchar(40), PRIMARY KEY(id));
```

```
create table autor(id int, nombre varchar(60), PRIMARY KEY(id));
```

```
create table ejemplar(id int, num_orden smallint, edicion smallint, ubicacion varchar(15),  
categoria char, id_libro varchar(60), foreign key (id_libro) references libro(id) on delete  
cascade on update cascade, PRIMARY KEY(id));
```

```
create table socio(id int, nombre varchar(60), direccion varchar(60), telefono varchar(15),  
categoria char, PRIMARY KEY(id));
```

```
create table prestamo(id_socio int, id_ejemplar int, num_orden int, fechaprestamo date,  
fecha_devolucion date, notas blob, foreign key (id_socio) references socio(id) on delete  
cascade on update cascade,foreign key (id_ejemplar) references ejemplar(id) on delete  
cascade on update cascade);
```

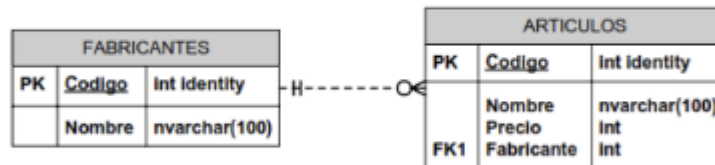
```
create table trata_sobre(id_libro varchar(60), id_tema smallint, foreign key (id_libro)  
references libro(id) on delete cascade on update cascade, foreign key (id_tema) references  
tema(id) on delete cascade on update cascade);
```

```
create table escrito_por(id_libro varchar(60), id_autor int, foreign key (id_libro) references  
libro(id) on delete cascade on update cascade,foreign key (id_autor) references autor(id) on  
delete cascade on update cascade);
```

3.

### EJERCICIO 3

#### La Tienda de Informática



Fabricantes(codigo, nombre)

Articulos(codigo, nombre, precio, fabricante)

```
create database informatica;
```

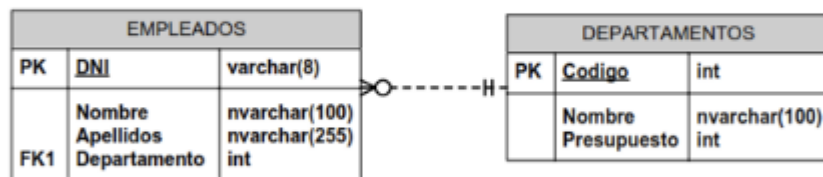
```
use informatica;
```

```
create table fabricantes(codigo int, nombre varchar(100), PRIMARY KEY(codigo));
```

```
create table articulos(codigo int, nombre varchar(100), precio int, fabricante int, foreign key  
(fabricante) references fabricantes(codigo) on delete cascade on update cascade,  
PRIMARY KEY(codigo));
```

4.

## Empleados



Empleados(dni, nombre, apellidos, id\_departamento)  
departamento(codigo, nombre, presupuesto)

```
create database empresa;  
use empresa;
```

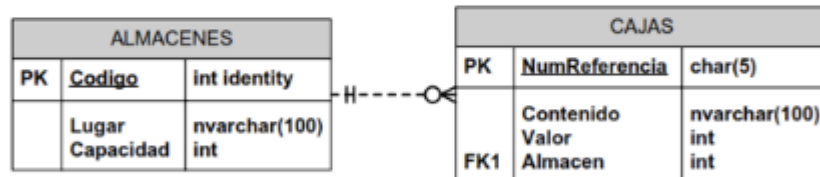
```
create table departamentos(codigo int, nombre varchar(100), presupuesto int, PRIMARY  
KEY(codigo));
```

```
create table empleados(dni varchar(8), nombre varchar(100), apellidos varchar(255),  
cod_departamento int, foreign key (cod_departamento ) references departamentos(codigo)  
on delete cascade on update cascade, PRIMARY KEY(dni));
```

5.

## EJERCICIO 5

### Los Almacenes



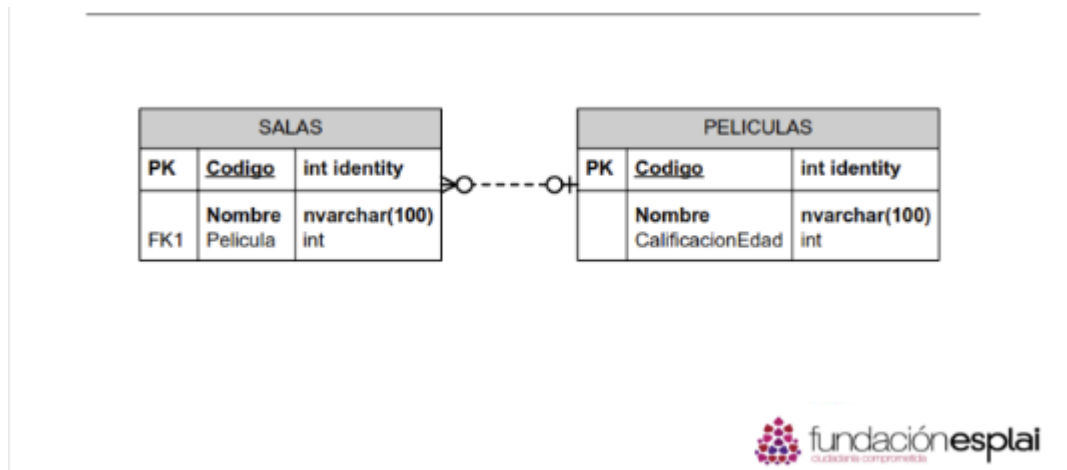
Almacenes(codigo,lugar,capacidad)  
Cajas(num\_ref,contenido,valor,almacen)

```
create database almacenes;  
use almacenes;
```

```
create table almacenes(codigo int, lugar varchar(100), capacidad int, PRIMARY  
KEY(codigo));
```

```
create table cajas(num_referencia varchar(5), contenido varchar(100), valor int, almacen int,  
foreign key (almacen) references almacenes(codigo) on delete cascade on update cascade,  
PRIMARY KEY(num_referencia));
```

6.



Peliculas(codigo, nombre, calificacion\_edad)

Salas(codigo, nombre, **pelicula**)

```
create database salas;
```

```
use salas;
```

```
create table peliculas(codigo int, nombre_pelicula varchar(100), calificacion_edad  
int, PRIMARY KEY(codigo));
```

```
create table salas(codigo int, nombre varchar(100), codigo_pelicula int, PRIMARY  
KEY(codigo),
```

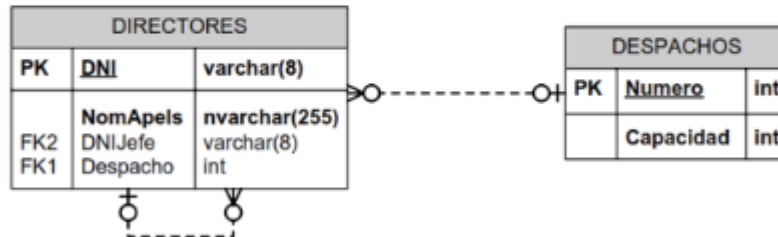
```
foreign key (codigo_pelicula) references peliculas(codigo) on delete cascade on update  
cascade
```

```
);
```

7.

## EJERCICIO 7

### Los Directores



```
create database directores;  
use directores;
```

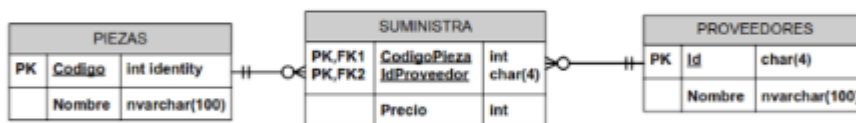
```
create table despachos(numero int, capacidad int, PRIMARY KEY(numero));
```

```
create table directores(dni varchar(8), nom_apels varchar(255), dni_jefe varchar(8),  
despacho int, foreign key (despacho) references despachos(codigo) on delete cascade on  
update cascade, foreign key (dni_jefe ) references directores(dni) on delete cascade on  
update cascade, PRIMARY KEY(dni));
```

8.

## EJERCICIO 8

### Piezas y Proveedores



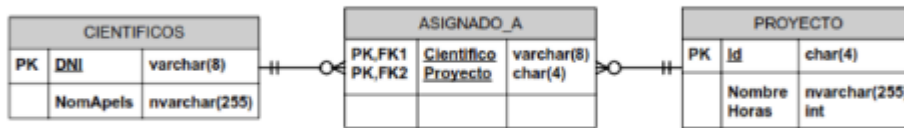
```
create database piezas;
use piezas;
```

```
create table piezas(codigo int, nombre varchar(100),PRIMARY KEY(codigo));
create table proveedores(codigo int, nombre varchar(100),PRIMARY KEY(codigo));
```

```
create table suministra(codigo_pieza int,codigo_proveedor int,precio double,PRIMARY
KEY(codigo_pieza,codigo_proveedor),
foreign key (codigo_pieza) references piezas(codigo) on delete cascade on update cascade,
foreign key (codigo_proveedor) references proveedores(codigo) on delete cascade on
update cascade
);
```



9.



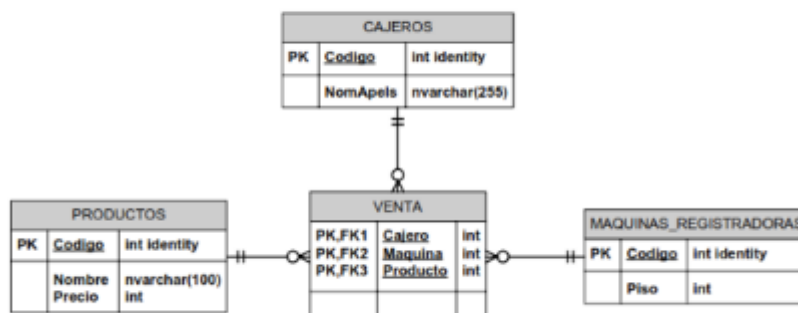
```
create database proyecto_cientifico;
use proyecto_cientifico;
```

```
create table cientificos(dni varchar(8), nom_apels varchar(255), PRIMARY KEY(dni));
```

```
create table proyecto(id varchar(4), nombre varchar(255), horas int, PRIMARY KEY(id));
```

```
create table asignado_a(dni_cientifico varchar(8), id_proyecto varchar(4), foreign key
(dni_cientifico) references cientificos(dni) on delete cascade on update cascade, foreign key
(id_proyecto ) references proyecto(id) on delete cascade on update cascade, PRIMARY
KEY(dni_cientifico, id_proyecto));
```

10.



```
create database cajeros;
use cajeros;
```

```
create table cajeros(codigo int, nombre varchar(255),PRIMARY KEY(codigo));
```

```
create table maquinas_registradoras(codigo int, piso int,PRIMARY KEY(codigo));
```

```
create table productos(codigo int, nombre varchar(255), precio int,PRIMARY KEY(codigo));
```

```

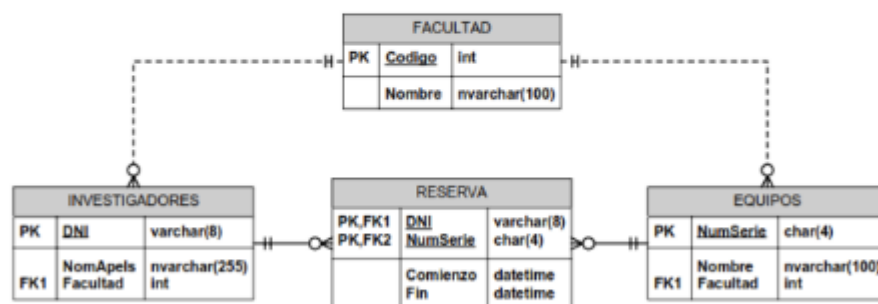
create table suministra(codigo_cajeros int,codigo_maquinas int,codigo_productos
int,PRIMARY KEY(codigo_cajeros,codigo_maquinas,codigo_productos),
foreign key (codigo_cajeros) references cajeros(codigo) on delete cascade on update
cascade,
foreign key (codigo_maquinas) references maquinas_registradoras(codigo) on delete
cascade on update cascade,
foreign key (codigo_productos) references productos(codigo) on delete cascade on update
cascade
);

```

11.

## EJERCICIO 11

### Los Investigadores



```

create database investigadores;
use investigadores;

```

```

create table facultad(codigo int, nombre varchar(100), PRIMARY KEY(codigo));

```

```

create table investigadores(dni varchar(8), nom_apels varchar(255), facultad int, foreign key
(facultad) references facultad(codigo) on delete cascade on update cascade, PRIMARY
KEY(dni));

```

```

create table equipos(num_serie varchar(4), nombre varchar(100), facultad int, foreign key
(facultad) references facultad(codigo) on delete cascade on update cascade, PRIMARY
KEY(num_serie));

```

```

create table reserva(dni_cientifico varchar(8), num_equipo varchar(4), comienzo date, fin
date, foreign key (dni_cientifico) references científicos(dni) on delete cascade on update

```

cascade, foreign key (num\_equipo) references equipos(num\_serie) on delete cascade on update cascade, PRIMARY KEY(dni\_cientifico, num\_equipo));

12.

Se desea tener una base de datos con la siguiente información acerca de los alumnos de una academia de idiomas donde se imparten varios cursos:

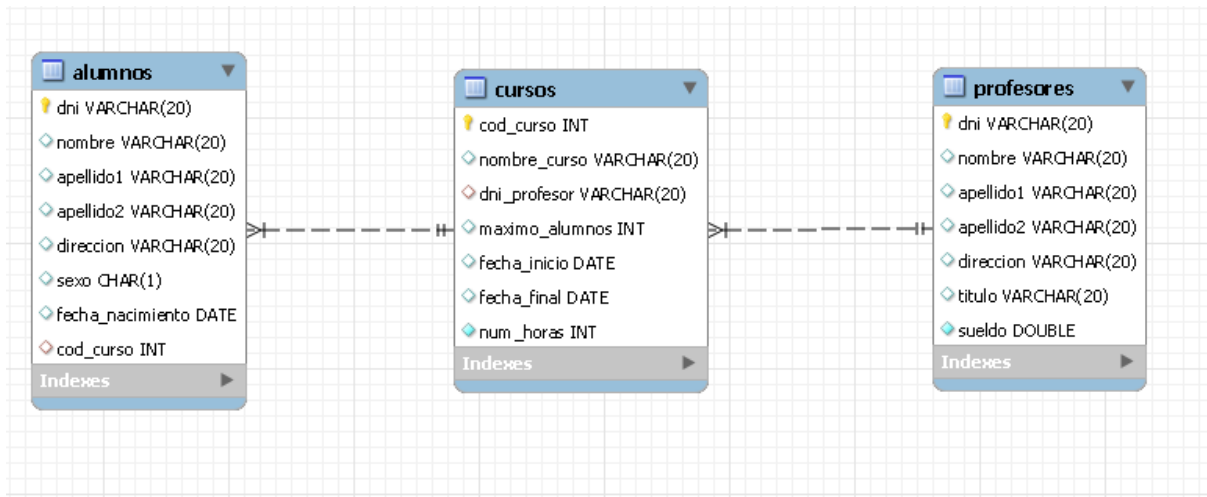
- Información acerca de los **alumnos**, que constará de su nombre y apellidos, un código único para cada alumno, su dirección, fecha de nacimiento y sexo ('H' o 'M').
- Estos alumnos estarán matriculados en un sólo **curso** cada uno. Cada curso tendrá un nombre, un código único que lo identifica, el número máximo de alumnos recomendado, un profesor, la fecha de inicio y de finalización del curso y el número de horas del curso.
- Los **profesores** tendrán una ficha en la academia donde se especifican sus datos personales y los datos del contrato. Es decir, nombre, DNI, dirección, titulación, cuota por hora.

1. Generar las siguientes tablas para guardar esta información

TABLA ALUMNOS	TABLA CURSOS	TABLA PROFESORES
Column Name	Column Name	Column Name
NOMBRE	NOMBRE_CURSO	NOMBRE
APELLIDO1	COD_CURSO	APELLIDO1
APELLIDO2	DNI_PROFESOR	APELLIDO2
DNI	MAXIMO_ALUMNOS	DNI
DIRECCION	FECHA_INICIO	DIRECCION
SEXO	FECHA_FIN	TITULO
FECHA_NACIMIENTO	NUM_HORAS	GANA
CURSO		

En las definiciones establecer las siguientes restricciones:

- No es posible dar de alta un alumno si no se matricula en un curso.
- La información del número de horas del curso es imprescindible para almacenarlo.
- El campo GANA de la tabla PROFESORES no puede estar en ningún caso vacío.
- Dos cursos no pueden llamarse igual. Lo mismo le pasa a los profesores.
- Podemos identificar las tuplas de las tablas CURSOS mediante el atributo CODIGO y PROFESORES y ALUMNOS usando el DNI.
- Cumplir la relación normal entre fecha comienzo y fecha fin (orden cronológico).
- Los valores para el atributo sexo son sólo M y H (en mayúsculas).
- Se ha de mantener la regla de integridad de referencia.



```
create database escuela;
```

```
use escuela;
```

```
create table profesores (dni varchar(20) , nombre varchar(20), apellido1 varchar(20),
apellido2 varchar(20), direccion varchar(20), titulo varchar(20), sueldo double not null,
unique (nombre, apellido1, apellido2),primary key(dni));
```

```
create table cursos (cod_curso int , nombre_curso varchar(20) unique, dni_profesor
varchar(20), maximo_alumnos int, fecha_inicio date, fecha_final date, num_horas int not null
,primary key(cod_curso),
foreign key (dni_profesor) references profesores(dni)
);
```

```
create table alumnos (dni varchar(20), nombre varchar(20), apellido1 varchar(20), apellido2
varchar(20), direccion varchar(20), sexo char(1) check (sexo in ('M', 'H')), fecha_nacimiento
date, cod_curso int,primary key(dni),
foreign key (cod_curso) references cursos(cod_curso)
);
```