



## Programación con Java

### TAREA 14

<b>Índice.....</b>	<b>1</b>
<b>Introducción.....</b>	<b>2</b>
<b>Creación de Bases de Datos.....</b>	<b>3</b>
Ejercicio 01:.....	3
Ejercicio 02:.....	4
Ejercicio 03:.....	5
Ejercicio 04:.....	6
Ejercicio 05:.....	7
Ejercicio 06:.....	8
Ejercicio 07:.....	9
Ejercicio 08:.....	10
Ejercicio 09:.....	11
Ejercicio 10:.....	12
Ejercicio 11:.....	13
Ejercicio 12:.....	14
<b>Webgrafía.....</b>	<b>15</b>

Con esta práctica iniciaremos con las bases de datos empezando por lo básico; comenzamos con los comandos para iniciar y finalizar la función del servidor MySQL, junto con la creación y gestión de los usuarios y, por último, la creación de bases de datos, tablas y su arquitectura.

# Creación de Bases de Datos

---

## Ejercicio 01:

En el primer caso convertiremos un modelo racional a una base de datos con Visual Studio Code y luego lo ejecutaremos en HeidiSQL.



### CÓDIGO

```
CREATE DATABASE ejercicio01 ;
USE ejercicio01;

CREATE TABLE estacion (
  identificador INT AUTO_INCREMENT,
  latitud INT,
  longitud INT,
  altitud INT,
  PRIMARY KEY (identificador));

CREATE TABLE muestra (
  idestacion INT,
  fecha DATE,
  temperaturamin INT,
  temperaturamax INT,
  precipitaciones INT,
  humedadmin INT,
  humedadmax INT,
  velocidadmin INT,
  velocidadmax INT,
  FOREIGN KEY (idestacion)
  REFERENCES estacion (identificador)
  ON DELETE CASCADE ON UPDATE CASCADE);
```

### BASES DE DATOS

Nombre ^	Filas	Tamaño
 estacion	0	16,0 KiB
 muestra	0	32,0 KiB

# Creación de Bases de Datos

## Ejercicio 02:

### CÓDIGO

```
CREATE DATABASE ejercicio02;
USE ejercicio02;

CREATE TABLE tema (
  clave_tema SMALLINT,
  nombre VARCHAR(40),
  PRIMARY KEY (clave_tema)
);

CREATE TABLE autor (
  clave_autor INT,
  nombre VARCHAR(60),
  PRIMARY KEY (clave_autor)
);

CREATE TABLE editorial (
  clave_editorial INT,
  nombre VARCHAR(60),
  direccion VARCHAR(60),
  telefono VARCHAR(15),
  PRIMARY KEY (clave_editorial)
);

CREATE TABLE libro (
  clave_libro INT,
  titulo VARCHAR(60),
  idioma VARCHAR(15),
  formato VARCHAR(15),
  clave_editorial INT,
  PRIMARY KEY (clave_libro),
  FOREIGN KEY (clave_editorial)
  REFERENCES editorial (clave_editorial)
  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE ejemplar (
  clave_ejemplar INT,
  clave_libro INT,
  numero_orden SMALLINT,
  edicion SMALLINT,
  ubicacion VARCHAR(15),
  categoria CHAR,
  PRIMARY KEY (clave_ejemplar),
  FOREIGN KEY (clave_libro)
  REFERENCES libro (clave_libro)
  ON DELETE CASCADE ON UPDATE CASCADE
);




CREATE TABLE socio (
  clave_socio INT,
  nombre VARCHAR(60),
  direccion VARCHAR(60),
  telefono VARCHAR(15),
  categoria CHAR,
  PRIMARY KEY (clave_socio)
);

CREATE TABLE prestamo (
  clave_socio INT,
  clave_ejemplar INT,
  numero_orden SMALLINT,
  fecha_prestamo DATE,
  fecha_devolucion DATE,
  notas BLOB,
  FOREIGN KEY (clave_socio)
  REFERENCES socio (clave_socio)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (clave_ejemplar)
  REFERENCES ejemplar (clave_ejemplar)
  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE trata_sobre (
  clave_libro INT,
  clave_tema SMALLINT,
  FOREIGN KEY (clave_libro)
  REFERENCES libro (clave_libro)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (clave_tema)
  REFERENCES tema (clave_tema)
  ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE escrito_por (
  clave_libro INT,
  clave_autor INT,
  FOREIGN KEY (clave_libro)
  REFERENCES libro (clave_libro)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (clave_autor)
  REFERENCES autor (clave_autor)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

### BASES DE DATOS

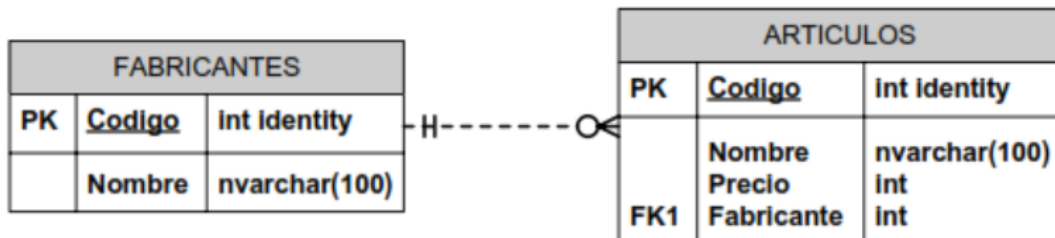
Nombre ^	Filas	Tamaño
 autor	0	16,0 KiB
 editorial	0	16,0 KiB
 ejemplar	0	32,0 KiB
 escrito_por	0	48,0 KiB
 libro	0	32,0 KiB
 prestamo	0	48,0 KiB
 socio	0	16,0 KiB
 tema	0	16,0 KiB
 trata_sobre	0	48,0 KiB

# Creación de Bases de Datos

## Ejercicio 03:

Ahora las conversiones las haremos desde una modelo racional, igualmente pero, a partir de las tablas.

### MODELO RACIONAL



### CÓDIGO

```
CREATE DATABASE ejercicio03;
USE ejercicio03;

CREATE TABLE fabricantes (
  codigo INT,
  nombre NVARCHAR(100),
  PRIMARY KEY (codigo)
);

CREATE TABLE articulos (
  codigo INT,
  nombre NVARCHAR(100),
  precio INT,
  fabricante INT,
  PRIMARY KEY (codigo),
  FOREIGN KEY (fabricante)
  REFERENCES fabricantes (codigo)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

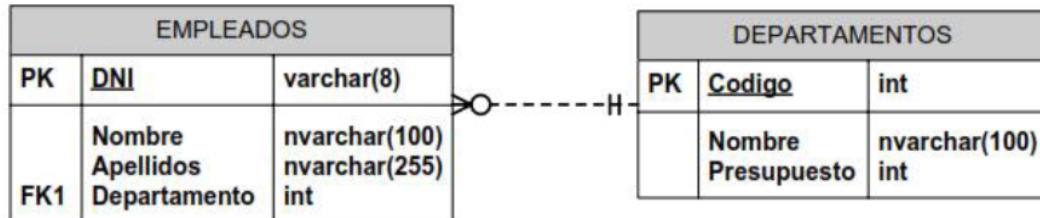
### BASES DE DATOS

Nombre ^	Filas	Tamaño
articulos	0	32,0 KiB
fabricantes	0	16,0 KiB

# Creación de Bases de Datos

## Ejercicio 04:

### MODELO RACIONAL





### CÓDIGO

```
CREATE DATABASE ejercicio04;
USE ejercicio04;

CREATE TABLE departamentos (
  codigo INT,
  nombre NVARCHAR(100),
  presupuesto INT,
  PRIMARY KEY (codigo)
);

CREATE TABLE empleados (
  dni NVARCHAR(8),
  nombre NVARCHAR(100),
  apellidos NVARCHAR(255),
  departamento INT,
  PRIMARY KEY (dni),
  FOREIGN KEY (departamento)
  REFERENCES departamentos (codigo)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

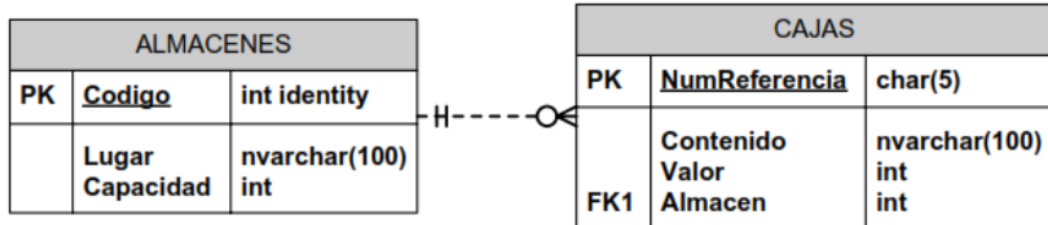
### BASES DE DATOS

Nombre ^	Filas	Tamaño
 departamentos	0	16,0 KiB
 empleados	0	32,0 KiB

# Creación de Bases de Datos

## Ejercicio 05:

### MODELO RACIONAL





### CÓDIGO

```
CREATE DATABASE ejercicio05;
USE ejercicio05;

CREATE TABLE almacenes (
    codigo INT,
    lugar NVARCHAR(100),
    capacidad INT,
    PRIMARY KEY (codigo)
);

CREATE TABLE cajas (
    num_referencia CHAR(5),
    contenido NVARCHAR(100),
    valor INT,
    almacen int,
    PRIMARY KEY (num_referencia),
    FOREIGN KEY (almacen)
    REFERENCES almacenes (codigo)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

### BASES DE DATOS

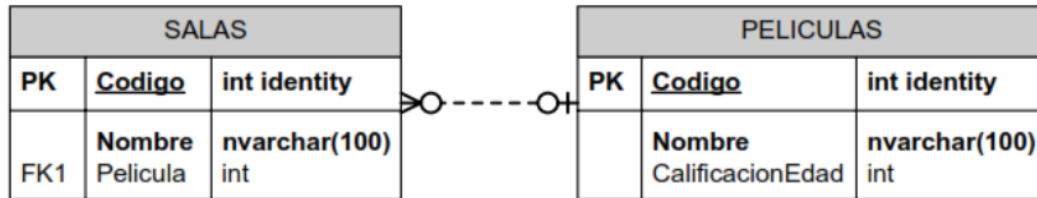
Nombre ^	Filas	Tamaño
 almacenes	0	16,0 KiB
 cajas	0	32,0 KiB



# Creación de Bases de Datos

## Ejercicio 06:

### MODELO RACIONAL





### CÓDIGO

```
CREATE DATABASE ejercicio06;
USE ejercicio06;

CREATE TABLE peliculas (
  codigo INT,
  nombre NVARCHAR(100),
  restriccion_edad INT,
  PRIMARY KEY (codigo)
);

CREATE TABLE salas (
  codigo INT,
  nombre NVARCHAR(100),
  pelicula INT,
  PRIMARY KEY (codigo),
  FOREIGN KEY (pelicula)
  REFERENCES peliculas (codigo)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

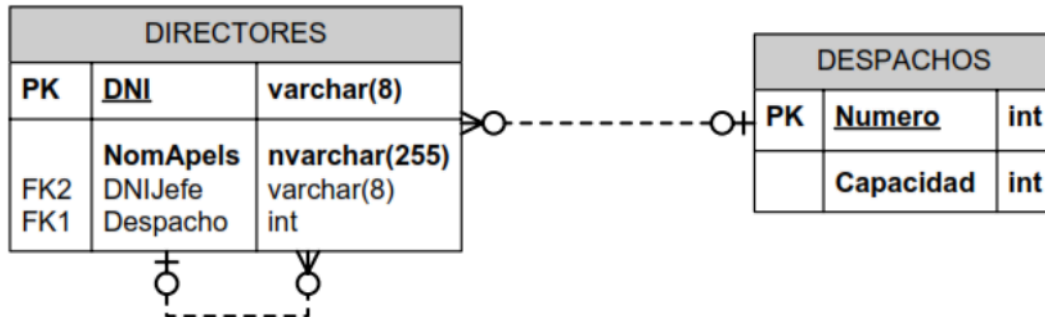
### BASES DE DATOS

Nombre ^	Filas	Tamaño
 peliculas	0	16,0 KiB
 salas	0	32,0 KiB

# Creación de Bases de Datos

## Ejercicio 07:

### MODELO RACIONAL





### CÓDIGO

```
CREATE DATABASE ejercicio07;
USE ejercicio07;

CREATE TABLE despachos (
    numero INT,
    capacidad INT,
    PRIMARY KEY (numero)
);

CREATE TABLE directores (
    dni VARCHAR(8),
    nombre_apellidos NVARCHAR(255),
    dni_jefe VARCHAR(8),
    despacho INT,
    PRIMARY KEY (dni),
    FOREIGN KEY (dni_jefe)
    REFERENCES directores (dni)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (despacho)
    REFERENCES despachos (numero)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

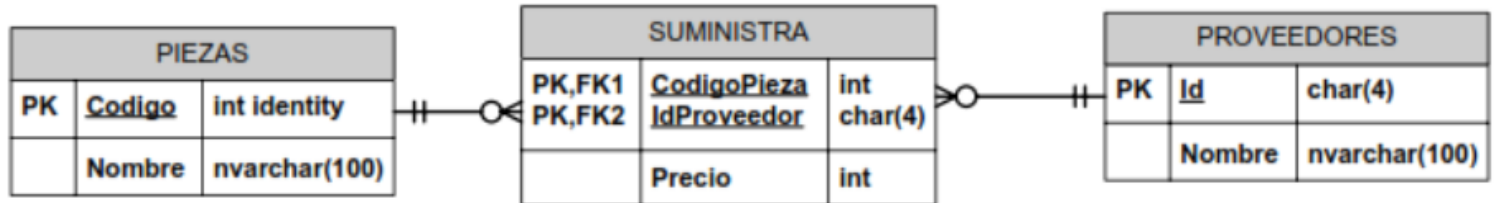
### BASES DE DATOS

Nombre ^	Filas	Tamaño
 despachos	0	16,0 KiB
 directores	0	48,0 KiB

# Creación de Bases de Datos

## Ejercicio 08:

### MODELO RACIONAL



### CÓDIGO




```
CREATE DATABASE ejercicio08;
USE ejercicio08;

CREATE TABLE piezas (
    codigo INT,
    nombre NVARCHAR(100),
    PRIMARY KEY (codigo)
);

CREATE TABLE proveedores (
    id CHAR(4),
    nombre NVARCHAR(100),
    PRIMARY KEY (id)
);

CREATE TABLE suministra (
    codigo_pieza INT,
    id_proveedor CHAR(4),
    precio INT,
    PRIMARY KEY (codigo_pieza, id_proveedor),
    FOREIGN KEY (codigo_pieza)
    REFERENCES piezas (codigo)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (id_proveedor)
    REFERENCES proveedores (id)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

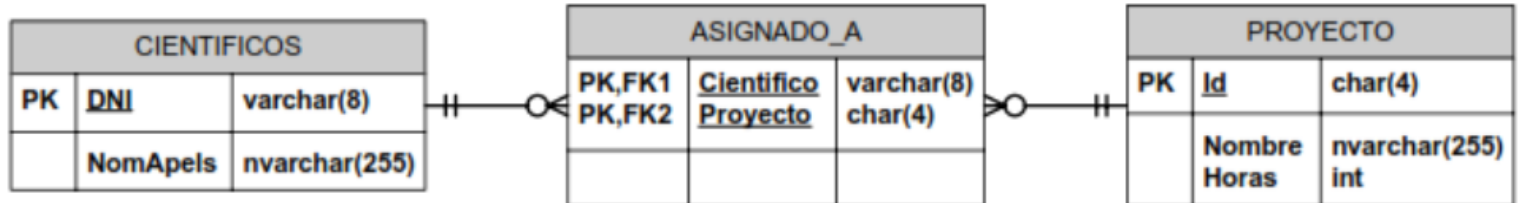
### BASES DE DATOS

Nombre ^	Filas	Tamaño
 piezas	0	16,0 KiB
 proveedores	0	16,0 KiB
 suministra	0	32,0 KiB

# Creación de Bases de Datos

## Ejercicio 09:

### MODELO RACIONAL



### CÓDIGO




```
CREATE DATABASE ejercicio09;
USE ejercicio09;

CREATE TABLE científicos (
    dni VARCHAR(8),
    nombre_apellidos NVARCHAR(255),
    PRIMARY KEY (dni)
);

CREATE TABLE proyectos (
    id CHAR(4),
    nombre NVARCHAR(255),
    horas INT,
    PRIMARY KEY (id)
);

CREATE TABLE asignado_a (
    científico VARCHAR(8),
    proyecto CHAR(4),
    PRIMARY KEY (científico, proyecto),
    FOREIGN KEY (científico)
    REFERENCES científicos (dni)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (proyecto)
    REFERENCES proyectos (id)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

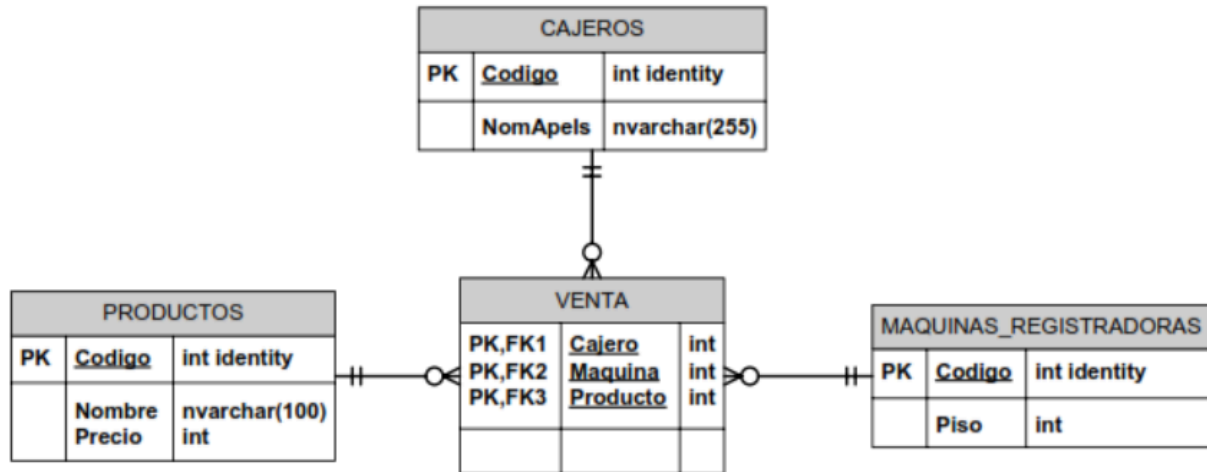
### BASES DE DATOS

Nombre ^	Filas	Tamaño
 asignado_a	0	32,0 KiB
 científicos	0	16,0 KiB
 proyectos	0	16,0 KiB

# Creación de Bases de Datos

## Ejercicio 10:

### MODELO RACIONAL



### CÓDIGO

```
CREATE DATABASE ejercicio10;
USE ejercicio10;



CREATE TABLE cajeros (
    codigo INT,
    nombre_apellidos NVARCHAR(255),
    PRIMARY KEY (codigo)
);

CREATE TABLE maquinas_registradoras (
    codigo INT,
    piso INT,
    PRIMARY KEY (codigo)
);

CREATE TABLE productos (
    codigo INT,
    nombre NVARCHAR(100),
    precio INT,
    PRIMARY KEY (codigo)
);

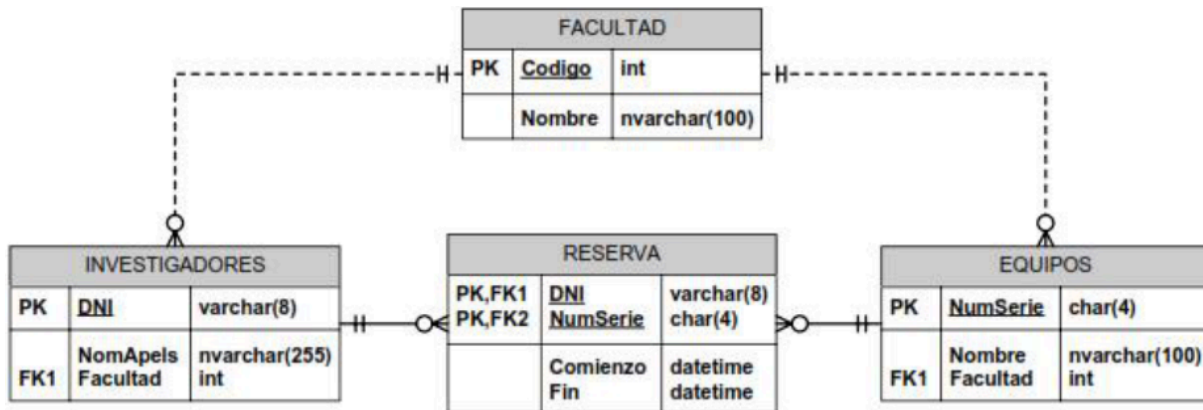
CREATE TABLE venta (
    cajero INT,
    maquina INT,
    producto INT,
    PRIMARY KEY (cajero, maquina, producto),
    FOREIGN KEY (cajero)
    REFERENCES cajeros (codigo)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (maquina)
    REFERENCES maquinas_registradoras (codigo)
    ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (producto)
    REFERENCES productos (codigo)
    ON DELETE CASCADE ON UPDATE CASCADE
);
```

### BASES DE DATOS

Nombre ^	Filas	Tamaño
 cajeros	0	16,0 KiB
 maquinas_regi...	0	16,0 KiB
 productos	0	16,0 KiB
 venta	0	48,0 KiB

## Ejercicio 11:

### MODELO RACIONAL



### BASES DE DATOS

Nombre ^	Filas	Tamaño
equipos	0	32,0 KiB
facultades	0	16,0 KiB
investigadores	0	32,0 KiB
reserva	0	32,0 KiB

### CÓDIGO

```
CREATE DATABASE ejercicio11;
USE ejercicio11;

CREATE TABLE facultades (
  codigo INT,
  nombre NVARCHAR(100),
  PRIMARY KEY (codigo)
);

CREATE TABLE investigadores (
  dni VARCHAR(8),
  nombre_apellidos NVARCHAR(255),
  facultad INT,
  PRIMARY KEY (dni),
  FOREIGN KEY (facultad)
  REFERENCES facultades (codigo)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE equipos (
  num_serie NVARCHAR(4),
  nombre NVARCHAR(100),
  facultad INT,
  PRIMARY KEY (num_serie),
  FOREIGN KEY (facultad)
  REFERENCES facultades (codigo)
  ON DELETE CASCADE ON UPDATE CASCADE
);




CREATE TABLE reserva (
  dni VARCHAR(8),
  num_serie NVARCHAR(4),
  comienzo DATETIME,
  fin DATETIME,
  PRIMARY KEY (dni, num_serie),
  FOREIGN KEY (dni)
  REFERENCES investigadores (dni)
  ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (num_serie)
  REFERENCES equipos (num_serie)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

# Creación de Bases de Datos

## Ejercicio 12:

Para el último ejercicio, tendremos que crear la base de datos a través de una información que nos dirán en el título, luego nos indicarán las restricciones y verificaciones que debe tener.

### BASES DE DATOS

Nombre ^	Filas	Tamaño
 alumnos	0	32,0 KiB
 cursos	0	64,0 KiB
 profesores	0	32,0 KiB

### CÓDIGO

```
CREATE DATABASE ejercicio12;
USE ejercicio12;

CREATE TABLE profesores (
dni VARCHAR(8),
nombre NVARCHAR(60),
apellidos NVARCHAR(255),
direccion VARCHAR(60),
titulo NVARCHAR(60),
gana INT NOT NULL,
PRIMARY KEY (dni),
UNIQUE (nombre)
);
```

```
CREATE TABLE cursos (
nombre_curso NVARCHAR(100),
cod_curso INT,
dni_profesor VARCHAR(8),
max_alumnos int,
fecha_inicio DATE,
fecha_fin DATE,
num_horas INT,
PRIMARY KEY (cod_curso),
INDEX (num_horas),
UNIQUE (nombre_curso),
CHECK (fecha_inicio < fecha_fin),
FOREIGN KEY ( dni_profesor)
REFERENCES profesores (dni)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE TABLE alumnos (
dni VARCHAR(8),
nombre NVARCHAR(60),
apellidos NVARCHAR(255),
direccion VARCHAR(60),
sexo CHAR,
fecha_nacimiento DATE,
curso INT,
PRIMARY KEY (dni),
INDEX (curso),
FOREIGN KEY (curso)
REFERENCES cursos (cod_curso)
ON DELETE CASCADE ON UPDATE CASCADE
-- CHECK (sexo = 'H' OR sexo = 'M')
);
```

- Cómo utilizar CHECK  
[CHECK SQL](#)
- Diferencia entre NVARCHAR Y VARCHAR :  
[Diferencias](#)
- Tipos de datos SQL :  
[Información sobre SQL](#)
- El copiloto de confianza como ayuda siempre, importante:  
[ChatGPT](#)