

Ejercicio 4.1

NIF - CHAR (9)

Nombre - VARCHAR (60)

Apellidos - VARCHAR (60)

Telefono - CHAR (12)

Fecha de nacimiento - DATE

Fecha de alta - DATE

Ejercicio 4.2

```
CREATE TABLE TDepartamentos (  
  nombre_dpto VARCHAR(255) NOT NULL,  
  cod_dpt INT PRIMARY KEY  
);
```

```
CREATE TABLE Empleados (  
  nombre VARCHAR (255) NOT NULL,  
  apellidos VARCHAR (255) NOT NULL,  
  dirección VARCHAR(255( NOT NULL,  
  cod_dpto INT,  
  Fecha_nac DATE,  
  Cod_emp INT PRIMARY KEY,  
  FOREIGNKEY (cod_dpto) REFERENCES Departamentos ( cod_dpto)  
);
```

Ejercicio 4.3

```
ALTER TABLE Empleados  
DROP COLUMN fecha_nac;
```

```
ALTER TABLE Empleados  
ADD COLUMN correo_electronico VARCHAR(255);
```

```
ALTER TABLE Empleados  
ALTER COLUMN direccion VARCHAR(260);
```

Ejercicio 4.4

```
CREATE TABLE TSocio (  
Nombre VARCHAR (255) NOT NULL,  
Apellidos VARCHAR (255) NOT NULL,  
Fecha de alta DATE NOT NULL,  
DNI CHAR (9) PRIMARY KEY
```

```
);
```

```
CREATE INDEX idx_socio_fecha_apellidos_nombre  
ON TSocio (Fecha_de_alta DESC, Apellidos, Nombre);
```

Ejercicio 4.5

Los siguientes datos son candidatos a definirse como tipos definidos por el usuario:

```
CREATE TYPE typCNombre AS VARCHAR (60);
```

```
CREATE TYPE typnAutorID AS CHAR (10);
```

```
CREATE TYPE typnLibroID AS CHAR (10);
```

EJERCICIOS PROPUESTOS

1.

SQL-92: Fue una de las primeras versiones estandarizadas de SQL. Introdujo el concepto de joins, vistas, transacciones y más. Esta versión del estándar SQL introdujo nuevas características como la capacidad de crear vistas actualizables, la capacidad de crear tablas temporales y la capacidad de crear procedimientos almacenados. También se agregaron nuevas funciones de agregación como COUNT(DISTINCT) y AVG(DISTINCT) .

SQL:1999: Esta versión introdujo soporte para tipos de datos complejos (como arrays y estructuras), expresiones regulares, y funciones analíticas. Esta versión del estándar SQL agregó soporte para objetos complejos y jerarquías de objetos. También se agregaron nuevas características como la capacidad de definir restricciones de integridad referencial en cascada y la capacidad de definir funciones definidas por el usuario ¹.

SQL:2003: Se centró en la mejora de la interoperabilidad entre diferentes sistemas de bases de datos y estandarizó las características de objetos. Esta versión del estándar SQL agregó soporte para XML y mejoró el soporte para objetos complejos. También se agregaron nuevas características como la capacidad de definir funciones de ventana y la capacidad de definir expresiones regulares .

SQL:2008: Introdujo mejoras significativas en las funciones analíticas, soporte para expresiones de tabla y mejoras en la manipulación de grandes conjuntos de datos. Esta versión del estándar SQL agregó soporte para secuencias y mejoró el soporte para objetos complejos. También se agregaron nuevas características como la capacidad de definir tablas temporales globales y la capacidad de definir procedimientos almacenados recursivos .

SQL:2011: Realizó pequeñas modificaciones y mejoras, pero no introdujo cambios drásticos en comparación con SQL:2008. Esta versión del estándar SQL agregó soporte para intervalos de tiempo y mejoró el soporte para objetos complejos. También se agregaron nuevas características como la capacidad de definir tablas temporales persistentes y la capacidad de definir procedimientos almacenados anónimos

SQL:2016: Introdujo soporte para JSON, mejoras en la gestión de la concurrencia, y funciones de fila de generación.

[Comparación de versiones de SQL Server \(licendi.com\)](https://www.licendi.com/comparacion-de-versiones-de-sql-server/)

[Revisión de SQL Server 2022 frente a versiones anteriores \(jotelulu.com\)](https://jotelulu.com/revisión-de-sql-server-2022-frente-a-versiones-anteriores/)

[Situación actual de los estándares SQL3 y SQL/MM | ComputerWorld](https://www.computerworld.com/es/actualidad/analisis/situacion-actual-de-los-estandares-sql3-y-sql-mm/)

2, 3, y 5

```
CREATE TYPE typCNombre AS VARCHAR (60);
```

```
CREATE TYPE typnAutorID AS CHAR (10);
```

```
CREATE TYPE typnLibroID AS CHAR (10);
```

```
CREATE TYPE typcApellidos AS CHAR (60);
```

**CREATE INDEX TLibroTema_nLibroID_nTemaID
ON TLibroTema (nLibroID, nTemaID);**

CREATE TABLE TLibroTema (
PRIMARY KEY (nLibroID, nTemaID);
);

**CREATE INDEX TTema_nTemaID
ON TTema (nTemaID);**

CREATE TABLE TTema (
nTemaID CHAR (10) PRIMARY KEY;
typCNombre NOT NULL;
);

**CREATE INDEX TLibro_typnLibro
ON TLibro (typnLibro);**

CREATE TABLE TLibro (
typnLibro PRIMARY KEY,
cTitulo VARCHAR(20) NOT NULL,
nAnyoPublicacion VARCHAR(20) NOT NULL,
nEditorialID CHAR(20) NOT NULL
);

**CREATE INDEX TEditorial_nEditorialID
ON TEditorial (nEditorialID);**

CREATE TABLE TEditorial (
nEditorialID CHAR(10) PRIMARY KEY,
nPaisID CHAR(10),
FOREIGN KEY (nPaisID) REFERENCES TPais(nPaisID)

);

**CREATE INDEX TPais_nPaisID
ON TPais (nPaisID);**

CREATE TABLE TPais(
nPaisID CHAR (10) PRIMARY KEY;
typCNombre NOT NULL;
);

**CREATE INDEX TAutorPais_nAutorID_nPaisID
ON TAutorPais (nAutorID, nPaisID);**

CREATE TABLE TAutorPais(
PRIMARY KEY (nAutorID, nPaisID);
);

**CREATE INDEX TAutor_typnAutorID
ON TAutor (typnAutorID);**

CREATE TABLE TAutor(
typnAutorID PRIMARY KEY;
typCNombre NOT NULL;
typcApellidos NOT NULL;
);

**CREATE INDEX TLibroAutor_nLibroID_nAutorID
ON TLibroAutor (nLibroID, nAutorID);**

CREATE TABLE TLibroAutor(
PRIMARY KEY (nLibroID, nAutorID);
);

**CREATE INDEX TEjemplar_cSignatura
ON TEjemplar (cSignatura);**

CREATE TABLE TEjemplar (
 cSignatura VARCHAR(15) PRIMARY KEY,
 nLibroID INT,
 FOREIGN KEY (nLibroID) REFERENCES TLibro(nLibroID)
);

**CREATE INDEX idx_TPrestamo_cSignatura_cNIF_dPrestamo
ON TPrestamo (cSignatura, cNIF, dPrestamo);**

CREATE TABLE TPrestamo (
 cSignatura VARCHAR(60),
 cNIF VARCHAR(20),
 dPrestamo DATE,
 PRIMARY KEY (cSignatura, cNIF, dPrestamo)
);

**CREATE INDEX idx_TSocio_cNIF
ON TSocio (cNIF);**

CREATE TABLE TSocio (
 cNIF CHAR(10) PRIMARY KEY,
 typCNombre VARCHAR(60) NOT NULL,
 typcApellidos VARCHAR(60) NOT NULL,
 cDirección VARCHAR(60) NOT NULL,
 cTelefono CHAR(9) NOT NULL,
 dNacimiento DATE NOT NULL,
 dAlta DATE NOT NULL
);

4

```
ALTER TABLE TAuthor  
ADD dNacimiento DATE,  
    dFallecimiento DATE;
```