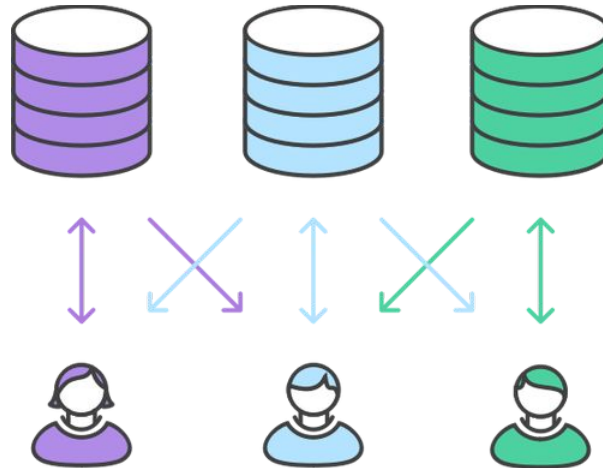


# Bases de Datos I

Tecnatura Universitaria en Inteligencia Artificial

## EJERCICIOS



## EJERCICIO 1

Necesitamos crear una base de datos llamada EstudiantesDB, que contenga una tabla llamada Estudiantes. Esta tabla tiene que incluir las columnas:

- EstudiantelD
- Nombre
- Apellido
- Fecha de Nacimiento
- Legajo
- Teléfono
- Dirección
- Apodo

Primero en un papel/lapiz, Documento de texto / Hoja de cálculo plantee:

1. La columna y el tipo de dato que elegiría.
2. Cuál elegiría como clave primaria.

Una vez que sepamos cual es nuestra clave primaria y los tipos de datos que vamos a usar, abrir el SQL Server Manager Studio y realizar lo siguiente:

1. Crear una base de datos llamada EstudiantesDB.
2. Crear una Tabla llamada Estudiantes dentro de EstudiantesDB.
3. Llenar la tabla con al menos 8 estudiantes.
4. Agregar una columna más llamada **email** y elegir el tipo de dato adecuado.
5. Eliminar una columna **Apodo** ( la institución se dio cuenta que no era muy apropiado tener el apodo de un alumno cargado en la base)
6. Agregar una columna más llamada **email** y elegir el tipo de dato adecuado.
7. Cargar los emails de los estudiantes (sin modificar ninguno de los otros campos)

Para crear la base, la tabla, para agregar / eliminar columnas utilizamos DDL (Data Definition Language - Lenguaje de definición de datos) y para modificar o actualizar un dato utilizamos DML (Data Manipulation Language - Lenguaje de manipulación de datos).

¿Cuales fueron las sentencias que utilizaron para campo punto en la diapo anterior?

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.

## EJERCICIO 2

1. Crear una base de datos LIBRERIA.  
Va a estar conformada por 8 tablas como las de las siguientes diapos.
2. Plantear las tablas con sus tipos de datos y restricciones correspondientes.
3. Una vez planteadas, crear las tablas utilizando lenguaje SQL.



## AUTORES

- autor\_id VARCHAR(11)
- nombre VARCHAR(40)
- apellido VARCHAR(20)
- telefono VARCHAR(12)
- direccion VARCHAR(40)
- ciudad VARCHAR(20)
- provincia CHAR(2)

## VENTAS

- local\_id CHAR(4)
- factura\_nro VARCHAR(20)
- fecha DATETIME
- cantidad SMALLINT
- forma\_pago VARCHAR(12)
- libro\_id VARCHAR(6)

## LOCALES

- local\_id VARCHAR(5)
- nombre VARCHAR(15)
- direccion VARCHAR(15)
- ciudad VARCHAR(10)
- provincia VARCHAR(20)
- c\_postal VARCHAR(15)

¿Qué problema nos podría traer declarar el local\_id de ventas como **char**?

¿Qué criterio tomamos para utilizar **varchar** en c\_postal y no un tipo de dato **int**?

¿Cuál es el rango que abarca SMALLINT? ¿Cuál es la diferencia con INTEGER?

## EDITORIALES

- editorial\_id \_\_\_\_\_
- nombre VARCHAR(40)
- ciudad VARCHAR(20)
- provincia CHAR(2)
- pais VARCHAR(30)

## LIBROS

- libro\_id \_\_\_\_\_
- titulo \_\_\_\_\_
- categoria VARCHAR(20)
- editorial\_id \_\_\_\_\_
- precio DOUBLE
- comentario VARCHAR(300)
- fecha\_public \_\_\_\_\_

## LIBRO\_AUTOR

- autor\_id \_\_\_\_\_
- libro\_id VARCHAR(10)

Completar los campos \_\_\_\_\_ con el dato que crean que corresponde

## EMPLEADOS

- empleado\_id CHAR(9)
- nombre VARCHAR(20)
- apellido VARCHAR(30)
- puesto\_id SMALLINT
- editorial\_id CHAR(4)
- fecha\_ingreso DATETIME

## PUESTOS

- puesto\_id SMALLINT
- descripcion VARCHAR(50)



**Consideraciones:** La lógica del negocio indica que un mismo número de factura se puede repetir en varios locales y en un mismo local, pero con distintos códigos de libros y la combinación de autor/libro es única.

