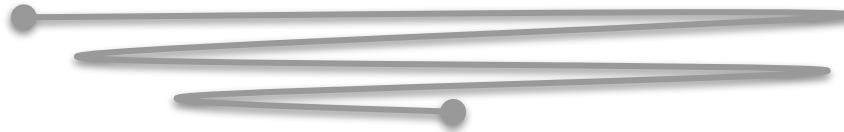


# *Laboratorio de Datos*



## *Modelo Relacional*

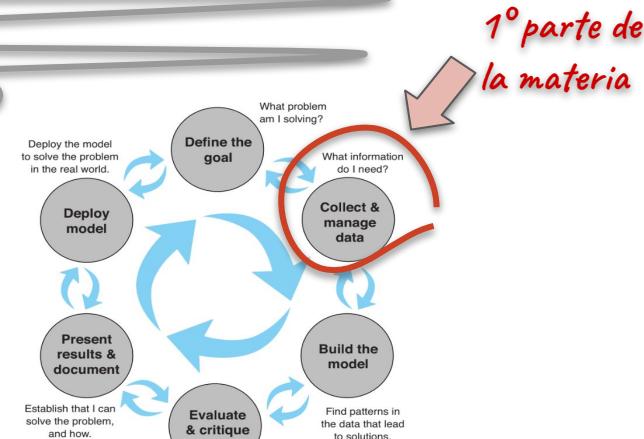


# Recorrido de la materia (hasta ahora)

- ✓ Lenguaje de programación para trabajar en nuestros proyectos



- ✓ Etapas de un proyecto de Ciencias de Datos

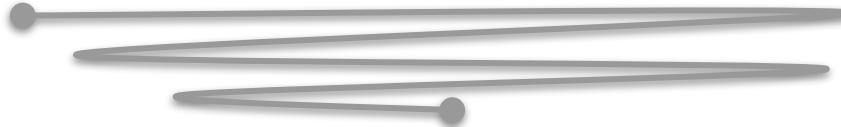


Tomado de "Practical Data Science With R - 2º Ed.", Zumel & Mount - 2020

- ✓ Modelado de Datos



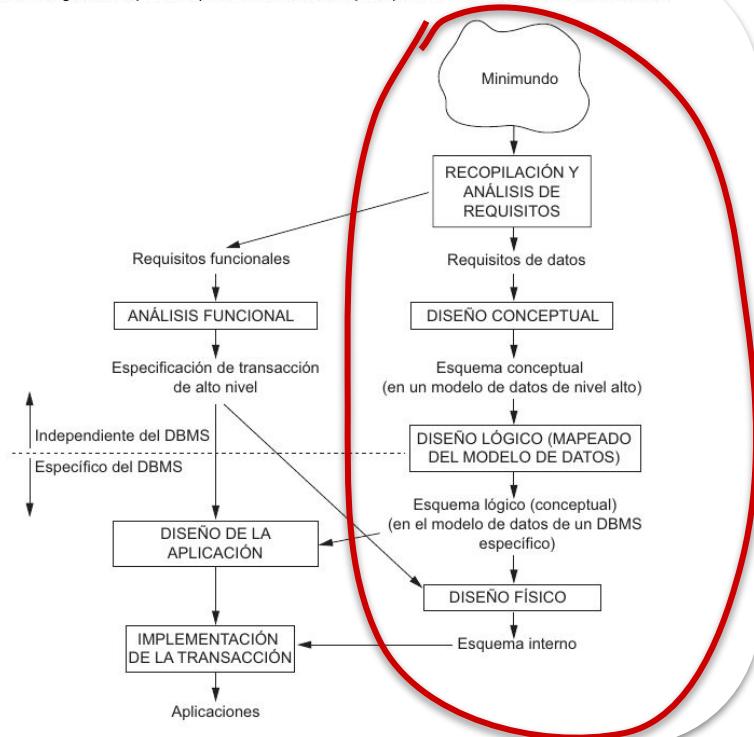
## *Clase Pasada -> Modelado de Datos*



- 1. Nos pusimos de acuerdo en algunos conceptos (dato, base de datos, etc.)*
- 2. Necesidad de representar parte del problema (recortar -> “Minimundo”)*
- 3. DER - Herramienta visual estándar que nos ayuda a explicar la estructura de los datos a modelar*

# Diseño de una BD - Etapas del Diseño

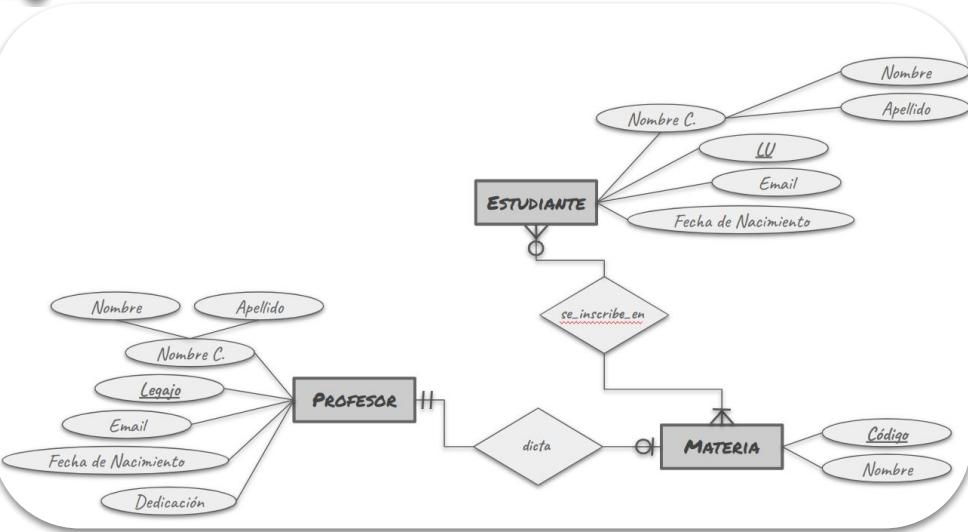
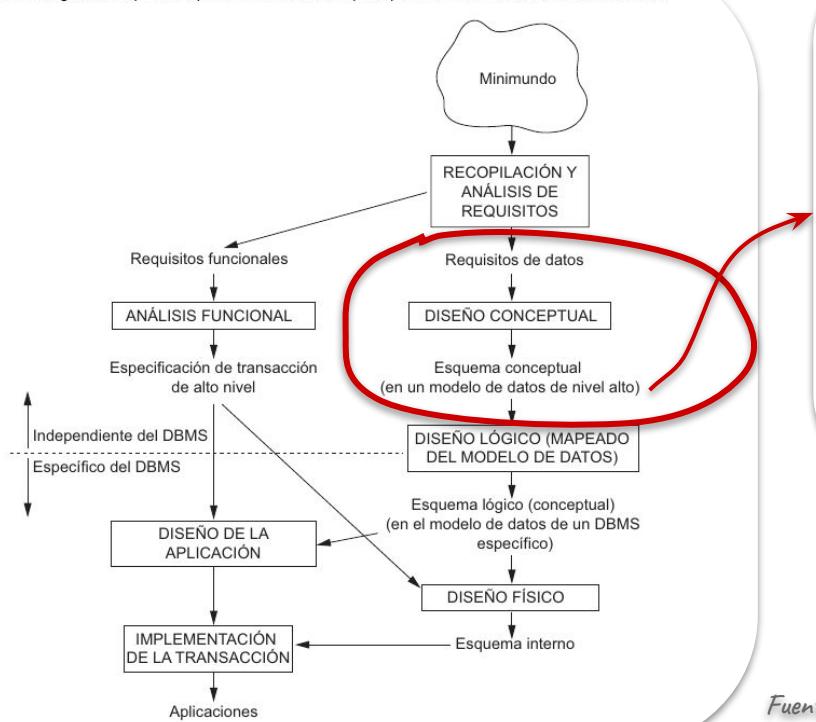
Figura 3.1. Diagrama simplificado para ilustrar las fases principales del diseño de una base de datos.



Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# Diseño de una BD - Modelo Conceptual

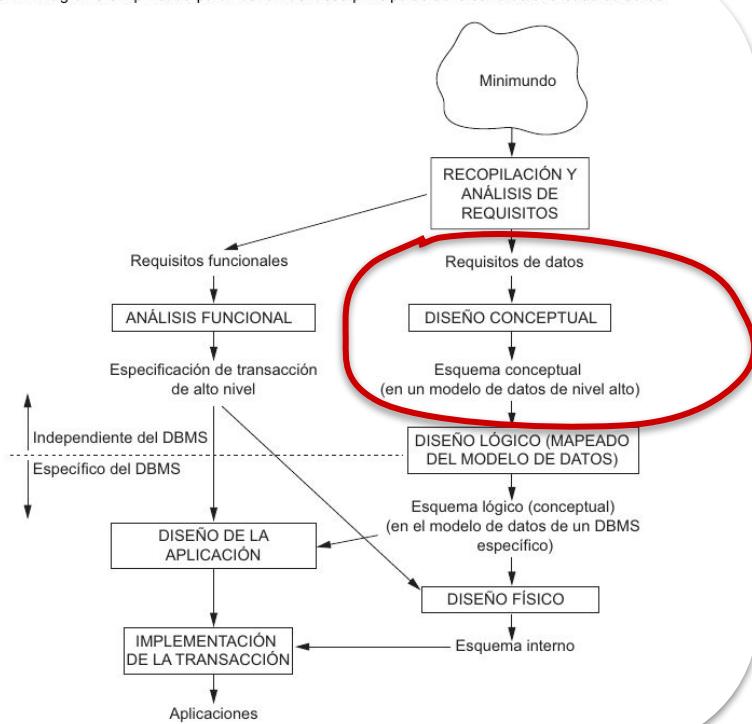
Figura 3.1. Diagrama simplificado para ilustrar las fases principales del diseño de una base de datos.



Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# Diseño de una BD - Modelo Conceptual

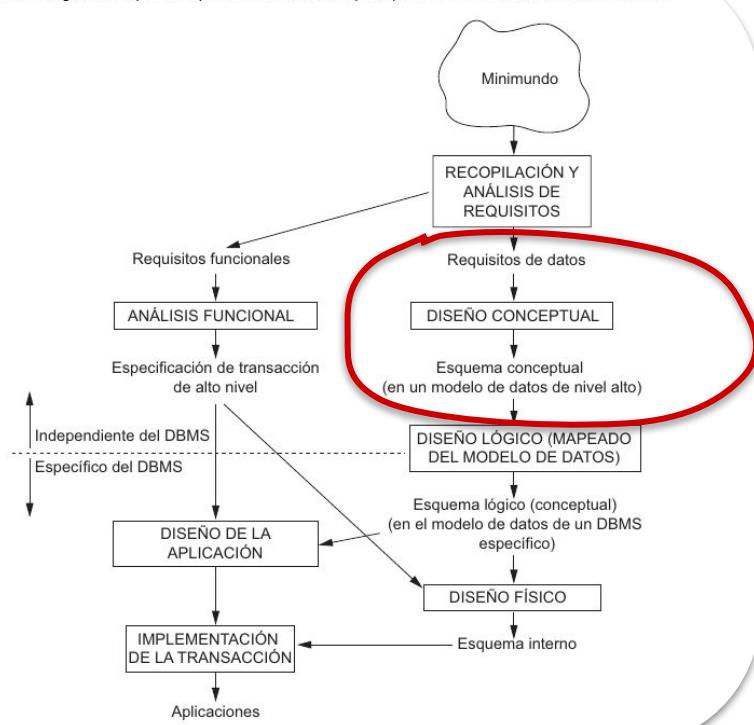
Figura 3.1. Diagrama simplificado para ilustrar las fases principales del diseño de una base de datos.



- ✓ Una conceptualización es una vista abstracta y simplificada del mundo que queremos representar.
- ✓ El conocimiento de un dominio se representa en un formalismo declarativo:
  - Conjunto de entidades, sus relaciones y restricciones, importantes para describir el problema/dominio.
- ✓ La clase pasada vimos el Modelo Entidad Relación (recordar el *DER*), pero existen otros.

# Diseño de una BD - Modelo Conceptual

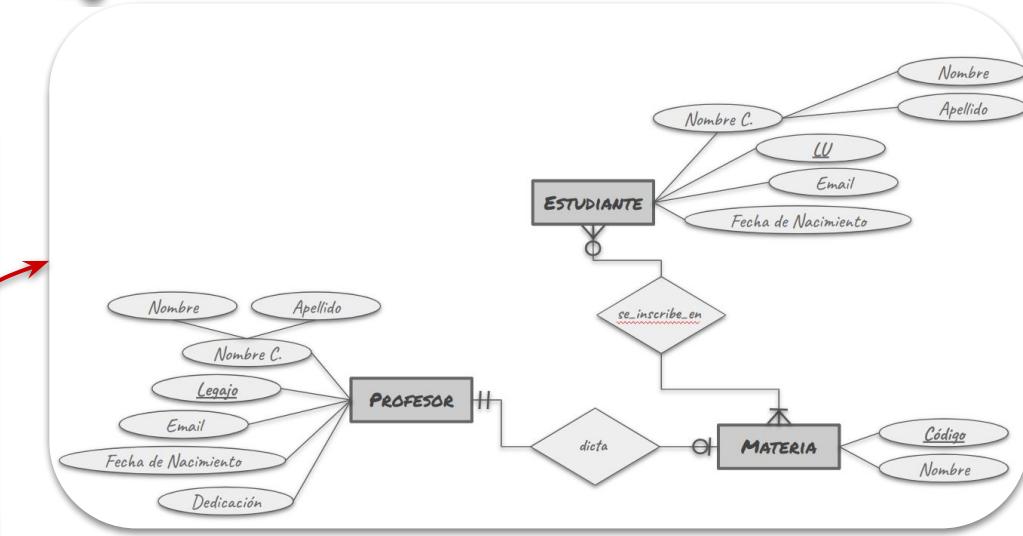
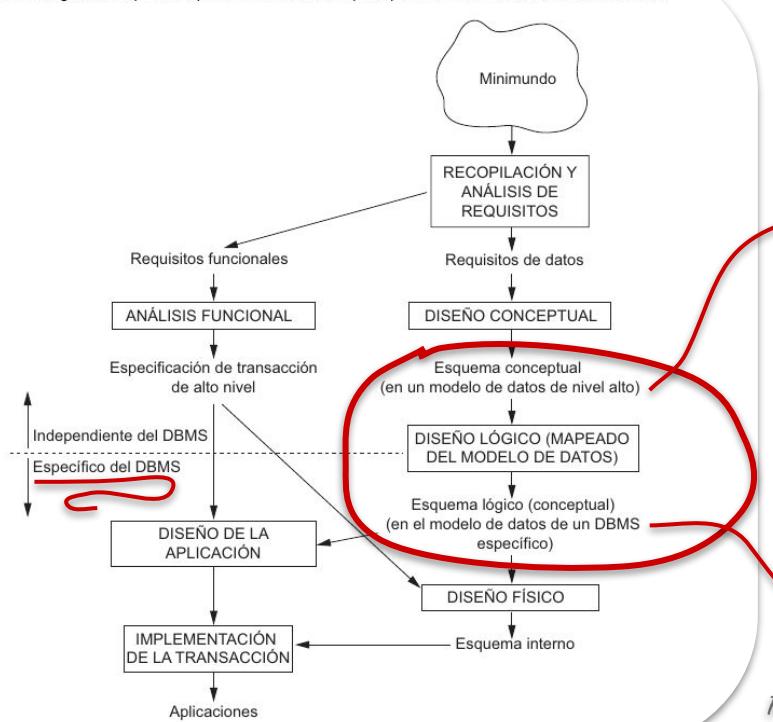
Figura 3.1. Diagrama simplificado para ilustrar las fases principales del diseño de una base de datos.



Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

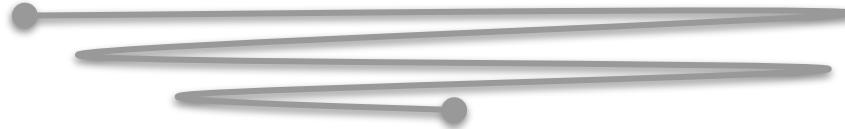
# Diseño de una BD - Modelo de Datos Relacional

Figura 3.1. Diagrama simplificado para ilustrar las fases principales del diseño de una base de datos.



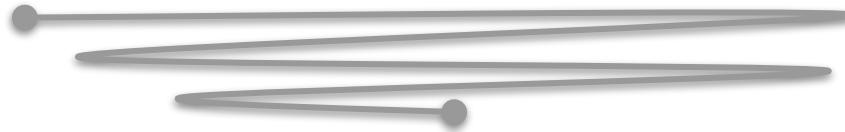
Depende de qué DBMS utilizemos ...

# *Sistemas de Administración de Bases de Datos*



*¿Qué tipos de Sistemas de Administración de Bases de Datos (DBMS) conocen?*

# Breve historia de las Bases de Datos



## Inicialmente (1960's)

- ✓ Modelos de redes
- ✓ Modelos jerárquicos

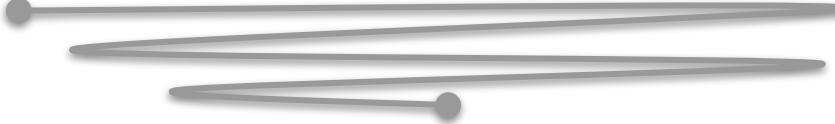
Historia ...

- ~ What goes around comes around. Reading in DB Systems, 4th Edition, 2006.
- ~ 01 - History of Databases (CMU Databases / Spring 2020)

## Mayor problema (... pero hay otros)

- ✓ No es independiente de la implementación física. Ante un cambio en la estructura de los datos hay que cambiar la API y las queries (consultas)

## Breve historia de las Bases de Datos



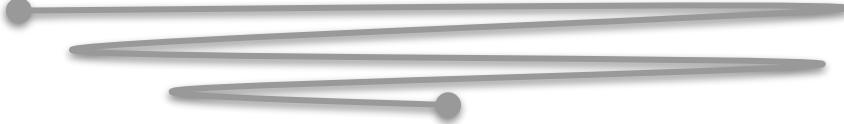
### Evolución (1970's)

- ✓ Modelos relacionales (Ted Codd)

### Historia ...

- ✓ Codd veía que en los otros modelos cada vez que algo cambiaba en el esquema de la base de datos había que reprogramar todo ...
- ✓ Modelo de Bases de Datos Relacionales (E. F. Codd) verdadera independencia física de los datos, es decir, no hay que reprogramar todo.
- ✓ Adoptado rápidamente por su simpleza y fundamentación matemática.

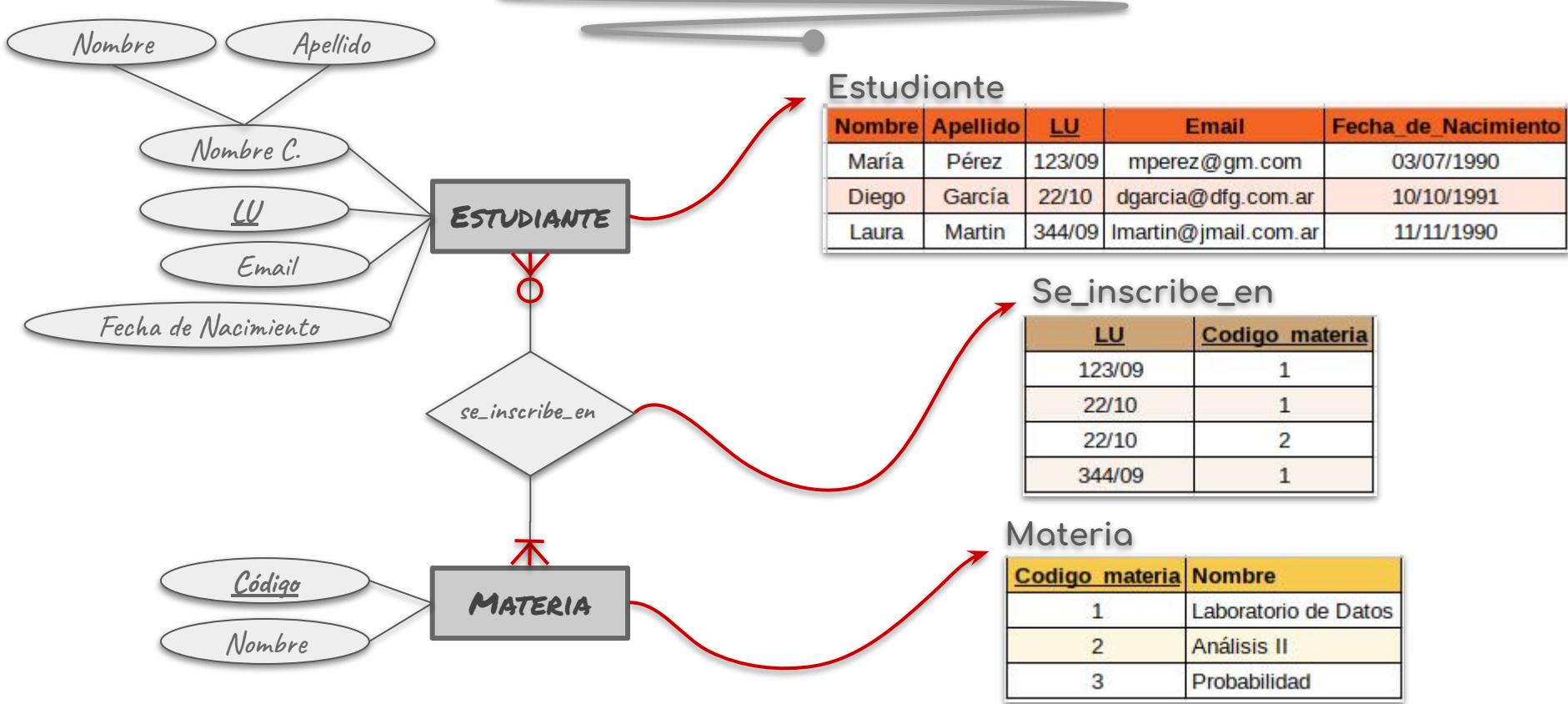
## Breve historia de las Bases de Datos



### Modelos relationales

- ✓ Tienen base teórica en la teoría de conjuntos y predicados de la lógica de primer orden
- ✓ Los datos se organizan en una estructura simple: relaciones (tablas)
- ✓ Datos se acceden a través de un lenguaje de alto nivel (operar a nivel de conjuntos)
- ✓ Almacenamiento físico queda para la implementación

# Modelo Entidad Relación -> Modelo Relacional

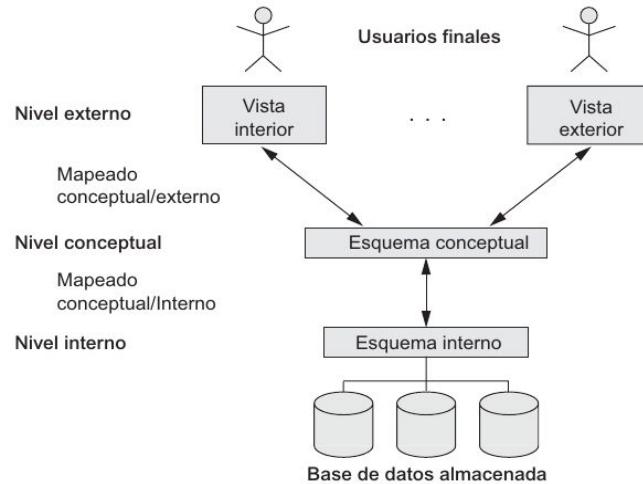


# DBMS - Arquitectura de 3 esquemas

## Objetivo (propuesta)

- ✓ Separar las aplicaciones de usuario y las bases de datos físicas

Figura 2.2. Arquitectura de tres esquemas.



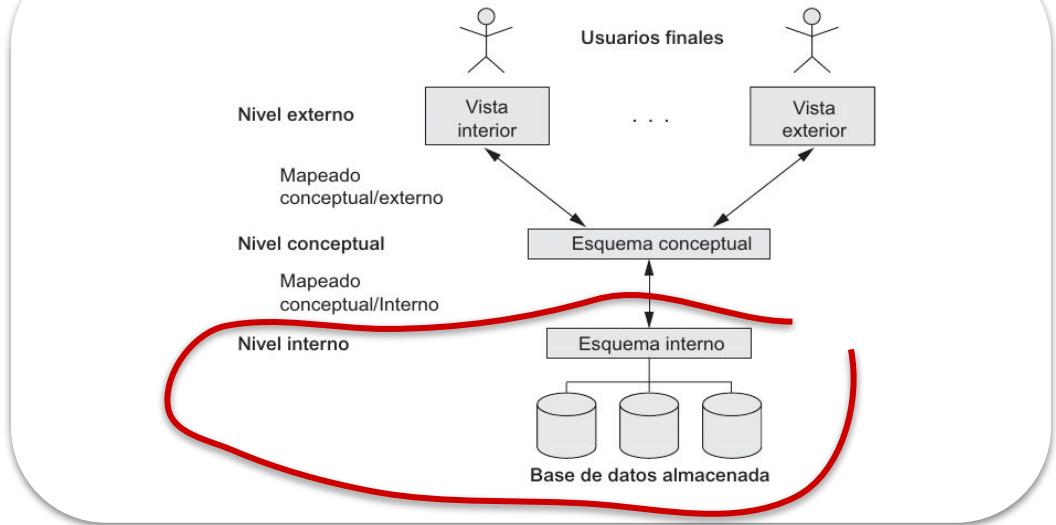
Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# DBMS - Arquitectura de 3 esquemas

## Nivel Interno

- ✓ Tiene un esquema interno, que describe la estructura de almacenamiento físico de la BD.
- ✓ El esquema interno utiliza un modelo de datos físico y describe todos los detalles del almacenamiento de datos y las rutas de acceso a la base de datos.
- ✓ En este nivel: Colección de archivos, índices y otras estructuras de almacenamiento.

Figura 2.2. Arquitectura de tres esquemas.



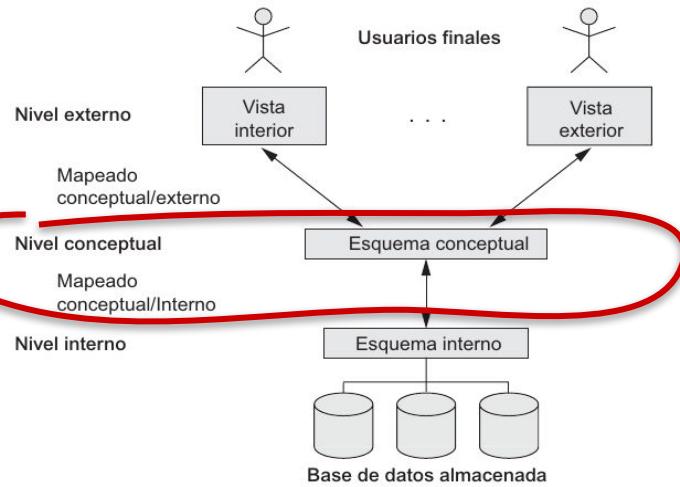
Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# DBMS - Arquitectura de 3 esquemas

## Nivel Conceptual

- ✓ Tiene un esquema conceptual, que describe la estructura de toda la base de datos para una comunidad de usuarios.
- ✓ El esquema conceptual oculta los detalles de las estructuras de almacenamiento físico y se concentra en describir las entidades, los tipos de datos, las relaciones, las operaciones de los usuarios y las restricciones.

Figura 2.2. Arquitectura de tres esquemas.



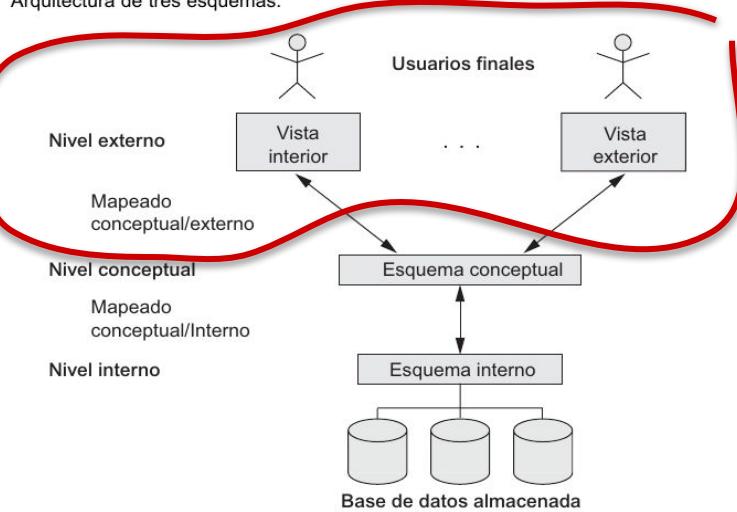
Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# DBMS - Arquitectura de 3 esquemas

## Nivel Externo

- ✓ Tiene varios esquemas externos.
- ✓ Un esquema externo describe la parte de la BD en la que un grupo de usuarios en particular está interesado y le oculta el resto de la BD.

Figura 2.2. Arquitectura de tres esquemas.



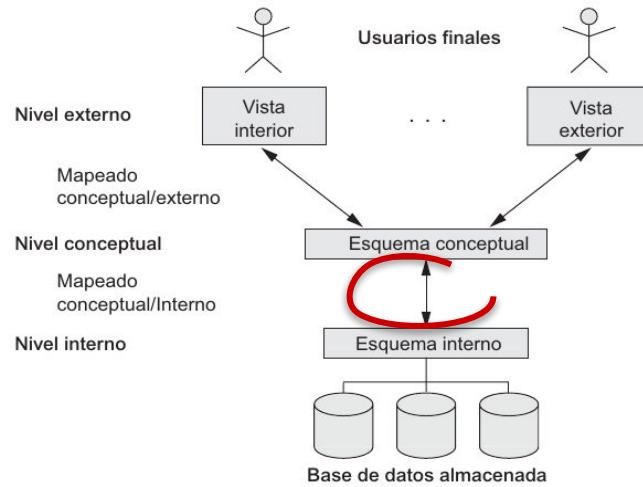
Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# DBMS - Arquitectura de 3 esquemas

## Independencia entre niveles

- ✓ Independencia física: capacidad del sistema de ejecutar cambios sobre la definición y estructura de los datos sin que esto afecte la BD conceptual.
- ✓ Ejemplo: cambiar la estructura de almacenamiento o el sistema de archivos, o la técnica de indexado sin tener que cambiar la BD conceptual que los usuarios ven/usen.

Figura 2.2. Arquitectura de tres esquemas.



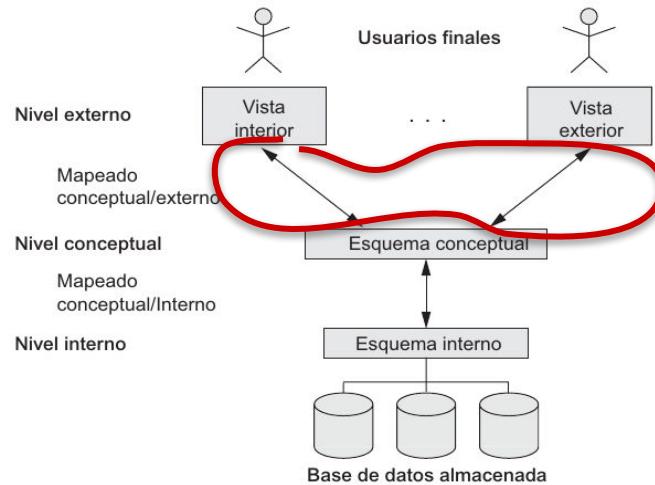
Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# DBMS - Arquitectura de 3 esquemas

## Independencia entre niveles

- ✓ Independencia lógica: capacidad del sistema de cambiar el esquema conceptual, sin cambiar la vista lógica que el usuario tiene de los datos.
- ✓ Ejemplo: agregar o borrar conceptos, relaciones, o atributos al modelo conceptual sin tener que reescribir programas de aplicación o vistas previamente definidas.

Figura 2.2. Arquitectura de tres esquemas.



Fuente: Elmasri/Navathe - Fundamentos de Sistemas de Bases de Datos, 5ta Ed., Pearson, 2007.

# Modelo Relacional

Modelo relacional representa la base de datos como una colección de relaciones (tablas)

Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
María	Pérez	123/09	mperez@gm.com	03/07/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

Materia

Codigo_materia	Nombre
1	Laboratorio de Datos
2	Análisis II
3	Probabilidad

Se\_inscribe\_en

LU	Codigo_materia
123/09	1
22/10	1
22/10	2
344/09	1

# Modelo Relacional

Informalmente:

- Posee tablas de valores donde cada fila representa una colección de valores relacionados
- Nombre de la tabla y de las columnas se utiliza para ayudar a interpretar el significado de cada uno de los valores de las filas

Diagrama que ilustra los conceptos básicos del Modelo Relacional:

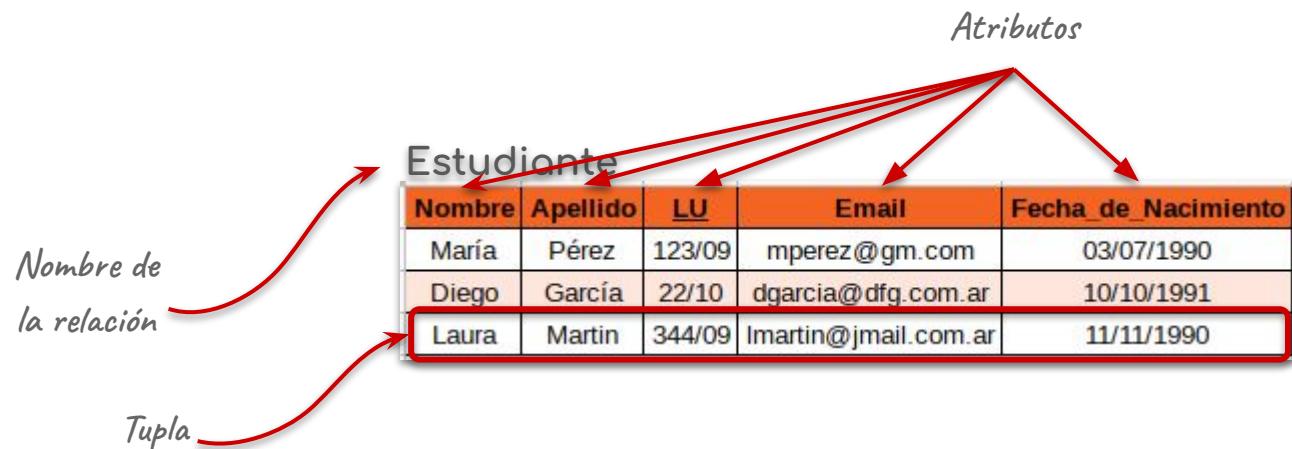
- Nombre de la tabla:** Se apunta al nombre "Estudiante" que aparece encima de la tabla.
- Fila:** Se apunta a la tercera fila de la tabla, que contiene los datos de la estudiante Laura.
- Nombres de las columnas:** Se apuntan a los nombres de las columnas: "Nombre", "Apellido", "LU", "Email" y "Fecha\_de\_Nacimiento".

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
María	Pérez	123/09	mperez@gm.com	03/07/1990
Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

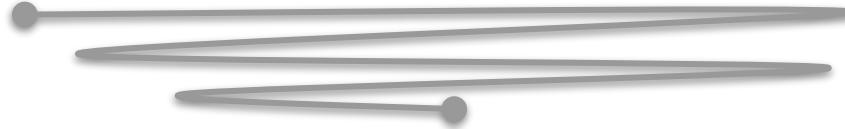
# Modelo Relacional

Más formalmente ...

- ✓ Los valores que pueden aparecer en cada atributo (columna) están representados por un dominio de posibles valores



# *Modelo Relacional*



*Dominio*

# Modelo Relacional - Dominio

Son  
definiciones lógicas  
de dominios

Dominio (D): Conjunto de valores atómicos (cada valor de un dominio es indivisible en lo que al modelo relacional se refiere).

Ejemplos.

DíasDeLaSemana

Lunes  
Martes  
Miércoles  
Jueves  
Viernes  
Sábado  
Domingo

Conjunto de días de la  
semana.

DocumentoNacionalIdentidad

1.000.000  
1.000.001  
1.000.002  
1.000.003  
1.000.004  
1.000.005  
....

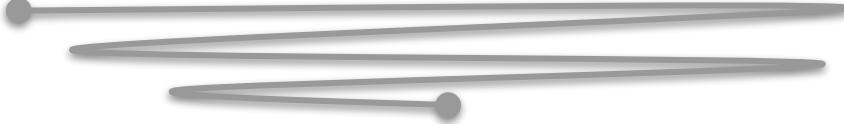
Conjunto de documentos  
nacionales de identidad  
(DNI) válidos en Argentina.

NombresDePersonas

Laura  
Sergio  
Alexia  
Josefina  
Justo  
Marisa  
....

Conjunto de nombres  
posibles de una persona.

## Modelo Relacional - Dominio



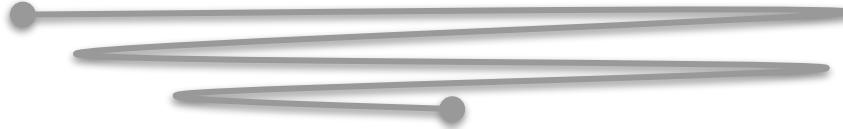
Para cada definición lógica se especifica también un tipo de dato o formato.

Ejemplos.

- ✓ *DiasDeLaSemana*. Es una cadena de caracteres que representa los nombres válidos de los días de la semana
- ✓ *DocumentoNacionalIdentidad*. Es un número entero comprendido entre 1.000.000 y 99.999.999
- ✓ *NombresDePersonas*. Es una cadena de caracteres que representa los nombres válidos de personas

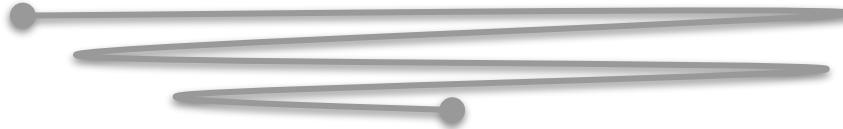
\* Un dominio cuenta con un nombre, un tipo de dato o formato. También se puede facilitar información adicional para la interpretación de sus valores. Ejemplo, un dominio numérico como *PesoPersona* puede contar con las unidades de medida, como kilogramos o libras.

# *Modelo Relacional*



*Esquema*

## Modelo Relacional - Esquema



El Esquema (o diseño) de relación describe la estructura de la relación (tabla).

### Ejemplo.

ESTUDIANTE(Nombre, Apellido, LU, EMail, Fecha\_de\_nacimiento)

Un esquema R, denotado por  $R(A_1, A_2, \dots, A_n)$ , está constituido por un nombre de relación R y una lista de atributos  $A_1, A_2, \dots, A_n$ .

También se pueden explicitar los tipos de datos de cada atributo.

ESTUDIANTE(Nombre: cadena, Apellido: cadena, LU: cadena, EMail: cadena, Fecha\_de\_Nacimiento: fecha)

# Modelo Relacional - Esquema



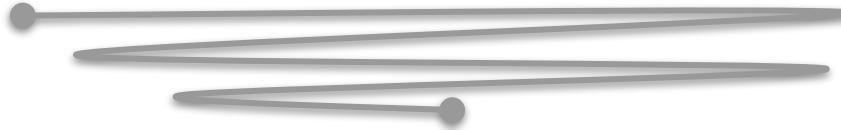
Cada atributo  $A_i$  pertenece a un dominio  $D$  y se especifica como  $\text{dom}(A_i)$

## Ejemplo.

**ESTUDIANTE**(Nombre, Apellido, LU, EMail, Fecha\_de\_nacimiento)

- $\text{dom}(\text{Nombre}) = \text{NombresDePersonas};$
- $\text{dom}(\text{Apellidos}) = \text{ApellidosDePersonas};$
- $\text{dom}(\text{LU}) = \text{LUDeLaFacultad};$
- $\text{dom}(\text{EMail}) = \text{EMailsExistentes};$
- $\text{dom}(\text{Fecha\_de\_Nacimiento}) = \text{FechasDeNacimientoDePersonasVivas}$

## Modelo Relacional - Esquema



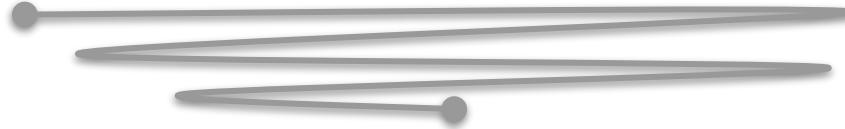
El grado (o arity) de una relación es la cantidad de atributos ( $n$ ) de la misma.

### Ejemplo.

ESTUDIANTE(Nombre, Apellido, LU, EMail, Fecha\_de\_nacimiento)

$$\text{grado}(\text{ESTUDIANTE}) = 5$$

# *Modelo Relacional*



*Estado*

# Modelo Relacional - Estado

Un estado de relación del esquema  $R(A_1, A_2, \dots, A_n)$ , también especificado como  $r(R)$ , es un conjunto de  $n$ -tuplas  $r = \{t_1, t_2, \dots, t_m\}$ . Cada tupla  $t$  es una lista ordenada de  $n$  valores  $t = \langle v_1, v_2, \dots, v_n \rangle$ , donde  $v_i$ ,  $1 \leq i \leq n$ , es un elemento de  $\text{dom}(A_i)$  o un valor especial  $\text{NULL}$

## Ejemplo.

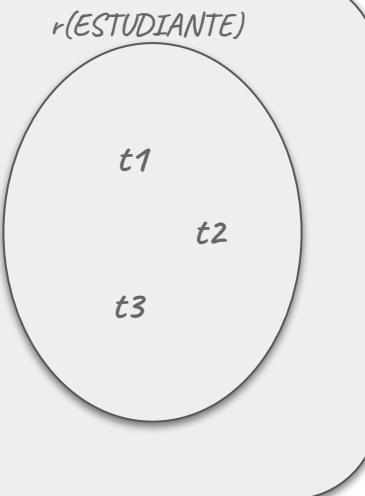
Atributos ->  $A_1$   $A_2$   $A_3$   $A_4$   $A_5$

Tuplas ↓

**Estudiante**

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
$t_1$	Maria	Pérez	123/09	mperez@gm.com	03/07/1990
$t_2$	Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
$t_3$	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

*v1 de la tupla t1*



# Modelo Relacional - Estado

El  $i$ -enésimo valor de la tupla  $t$ , que se corresponde con el atributo  $A_i$ , se referencia como  $t[A_i]$  (o  $t[i]$  si utilizamos una notación posicional).

## Ejemplo.

Atributos ->  $A_1$   $A_2$   $A_3$   $A_4$   $A_5$

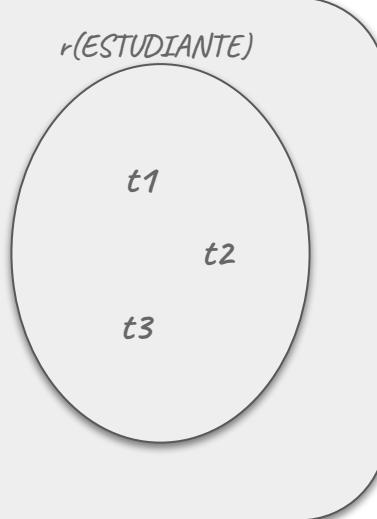
Tuplas ↓

**Estudiante**

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
$t_1$	Maria	Pérez	123/09	mperez@gm.com	03/07/1990
$t_2$	Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
$t_3$	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

$v_1$  de la tupla  $t_1$

$$t_1[1] = \text{"Maria"} \\ t_2[5] = 10/10/1991$$



# Modelo Relacional - Estado

$r(R)$  es un subconjunto del producto cartesiano de los dominios que definen  $R$

$$r(R) \subseteq (\text{dom}(A_1) \times \text{dom}(A_2) \times \dots \times \text{dom}(A_n))$$

Ejemplo.

Atributos ->  $A_1$   $A_2$   $A_3$   $A_4$   $A_5$

Tuplas ↓

**Estudiante**

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
$t_1$	Maria	Pérez	123/09	mperez@gm.com	03/07/1990
$t_2$	Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
$t_3$	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

v1 de la tupla  $t_1$

$r(\text{ESTUDIANTE})$

$t_1$

$t_2$

$t_3$

# Modelo Relacional - Estado

¿Cuántas tuplas distintas son posibles representar en el esquema  $R(A_1, A_2, \dots, A_n)$ ?

$|\text{dom}(A_1)| \times |\text{dom}(A_2)| \times \dots \times |\text{dom}(A_n)|$ . Este producto de cardinalidades de todos los dominios representa la cantidad total de tuplas distintas que pueden existir en la relación  $r(R)$ .

## Ejemplo.

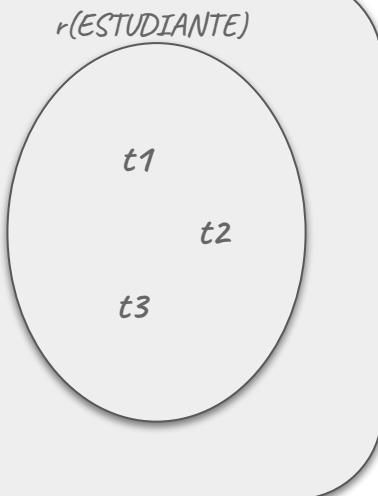
Atributos ->  $A_1$   $A_2$   $A_3$   $A_4$   $A_5$

Tuplas ↓

**Estudiante**

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
$t_1$	Maria	Pérez	123/09	mperez@gm.com	03/07/1990
$t_2$	Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
$t_3$	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

v1 de la tupla  $t_1$



# Modelo Relacional - Estado



La Biblioteca de Babel  
Jorge Luis Borges

¿Cuántas tuplas distintas son posibles representar en el esquema  $R(A_1, A_2, \dots, A_n)$ ?

$|\text{dom}(A_1)| \times |\text{dom}(A_2)| \times \dots \times |\text{dom}(A_n)|$ . Este producto de cardinalidades de todos los dominios representa la cantidad total de tuplas distintas que pueden existir en la relación  $r(R)$ .

## Ejemplo.

Atributos -> A1 A2 A3 A4 A5

Tuplas ↓

t1 t2 t3

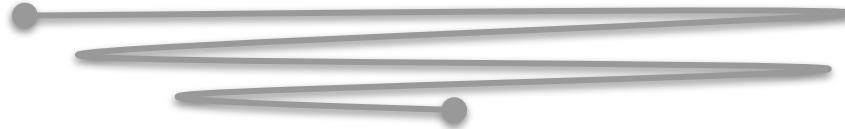
Estudiante

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
t1	Maria	Perez	123/09	mperez@gm.com	03/07/1990
t2	Diego	Garcia	22/10	dgarcia@dfq.com.ar	10/10/1991
t3	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

v1 de la tupla t1



# Modelo Relacional - Características



## Orden de las tuplas en una relación

La relación está definida como un conjunto de tuplas. Matemáticamente, los elementos de un conjunto no guardan un orden entre ellos (repito: no guardan un orden entre ellos); por tanto, las tuplas en una relación tampoco la tienen.

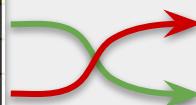
### Ejemplo.

Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Maria	Pérez	123/09	mperez@gm.com	03/07/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Maria	Pérez	123/09	mperez@gm.com	03/07/1990



Para el modelo relacional ambos estados de relación son equivalentes (son conjuntos de tuplas)

## Modelo Relacional - Características

### Orden de los valores dentro de una tupla

El orden de valores dentro de una tupla (y por consiguiente los atributos de un esquema de relación) es importante.

#### Ejemplo.

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990



Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Martin	Laura	344/09	lmartin@jmail.com.ar	11/11/1990

Para el modelo relacional estas tuplas NO SON equivalentes.

Puede darse una definición alternativa de una relación que haría innecesaria el ordenamiento de los valores de una tupla, sin embargo usaremos la definición original de relación, en la que los atributos y los valores de dentro de las tuplas están ordenados, porque simplifica mucho la notación.

# Modelo Relacional - Características

Cada valor en una tupla es un valor atómico

El valor no es divisible en componentes -> no están permitidos los atributos compuestos ni multivalor.

Gran parte de la teoría que se esconde tras el modelo relacional fue desarrollada con este principio en mente, el cual recibe el nombre de principio de Primera Forma Normal.

Ejemplo. Atributo compuesto

~~Estudiante~~

Nombre Completo	LU	Email	Fecha de Nacimiento
Maria Pérez	123/09	mperez@gm.com	03/07/1990
Diego García	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura Martin	344/09	lmartin@jmail.com.ar	11/11/1990

~~Estudiante~~

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Maria	Pérez	123/09	mperez@gm.com	03/07/1990

Los atributos compuestos deben representarse sólo por sus atributos de componente simple

# Modelo Relacional - Características

Cada valor en una tupla es un valor atómico

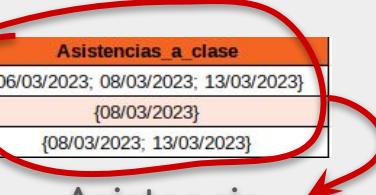
El valor no es divisible en componentes -> no están permitidos los atributos compuestos ni multivalor.

Gran parte de la teoría que se esconde tras el modelo relacional fue desarrollada con este principio en mente, el cual recibe el nombre de principio de Primera Forma Normal.

Ejemplo. Atributo multivalor

Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento	Asistencias_a_clase
Maria	Perez	123/09	mperez@gm.com	03/07/1990	{06/03/2023; 08/03/2023; 13/03/2023}
Diego	Garcia	22/10	dgarcia@dfg.com.ar	10/10/1991	{08/03/2023}
Laura	Martin	344/09	lmartin@mail.com.ar	11/11/1990	{08/03/2023; 13/03/2023}



Estudiante

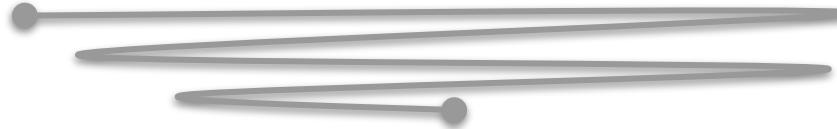
Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@mail.com.ar	11/11/1990
Diego	Garcia	22/10	dgarcia@dfg.com.ar	10/10/1991
Maria	Perez	123/09	mperez@gm.com	03/07/1990

Asistencia

LU	Asistencias_a_clase
123/09	06/03/2023
123/09	08/03/2023
123/09	13/03/2023
22/10	08/03/2023
344/09	08/03/2023
344/09	13/03/2023

Los atributos multivalor deben representarse en relaciones separadas

# Modelo Relacional - Características



## Valores NULLs en las tuplas

Los valores NULL pueden tener varios significados:

- Valor desconocido
- Valor existente pero no disponible
- Atributo no aplicable a esta tupla (Ejemplo: Número de licencia de conducir para un menor de edad)

### Ejemplo.

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento	Dirección
María	Pérez	123/09	mperez@gm.com	03/07/1990	Rivadavia 1234 - CABA
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991	NULL
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990	NULL

¿Significa que Diego y Laura tienen la misma dirección?



Durante el diseño de una base de datos es preferible evitar los valores NULL tanto como sea posible.

## Modelo Relacional - Notación



### Letras

- Un esquema de relación  $R$  de grado  $n$  se designa como  $R(A_1, A_2, \dots, A_n)$ .
- Las letras  $Q, R, S$  especifican nombres de relación.
- Las letras  $q, r, s$  especifican estados de relación.
- Las letras  $t, u, v$  indican tuplas.

# Modelo Relacional - Notación



En general, el nombre de una relación como *ESTUDIANTE* va a indicar el conjunto real de tuplas de la misma (el estado actual de la relación) mientras que *ESTUDIANTE(Nombre, Apellido, ...)* se va a referir sólo a su esquema.

## Ejemplo.

*ESTUDIANTE*



Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
María	Pérez	123/09	mperez@gm.com	03/07/1990

*ESTUDIANTE(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)*



Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
--------	----------	----	-------	---------------------

# Modelo Relacional - Notación

## Atributos

Dos atributos (en relaciones diferentes) podrían llegar a tener el mismo nombre. ¿Cómo podemos hacer para diferenciarlos?

Ejemplo.

Estudiante.LU

Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Maria	Pérez	123/09	mperez@gm.com	03/07/1990

Asistencia.LU

Asistencia

LU	Asistencias_a_clase
123/09	06/03/2023
123/09	08/03/2023
123/09	13/03/2023
22/10	08/03/2023
344/09	08/03/2023
344/09	13/03/2023

Podemos anteponer al atributo el nombre de la relación (separados por un punto seguido).

Importante: En una misma relación no pueden existir dos atributos con el mismo nombre.

# Modelo Relacional - Notación

Una  $n$ -tupla  $t$  en una relación  $r(R)$  está designada por  $t = \langle v_1, v_2, \dots, v_n \rangle$ , donde  $v_i$  es el valor correspondiente al atributo  $A_i$ . La siguiente notación se refiere a los valores componentes de las tuplas:

- Tanto  $t[A_i]$  como  $t.A_i$  (y, a veces,  $t[i]$ ) hacen referencia al valor  $v_i$  de  $t$  del atributo  $A_i$ .
- Tanto  $t[A_u, A_w, \dots, A_z]$  como  $t.(A_u, A_w, \dots, A_z)$ , donde  $A_u, A_w, \dots, A_z$  es una lista de atributos de  $R$ , hacen referencia a la subtupla de valores  $\langle v_u, v_w, \dots, v_z \rangle$  de  $t$  correspondientes a los atributos especificados en la lista.

## Ejemplo.

Atributos ->  $A_1$   $A_2$   $A_3$   $A_4$   $A_5$

Tuplas ↓

*Estudiante*

	Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
$t_1$	Maria	Pérez	123/09	mperez@gm.com	03/07/1990
$t_2$	Diego	García	22/10	dgarcia@dfq.com.ar	10/10/1991
$t_3$	Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990

*v<sub>1</sub> de la tupla t<sub>1</sub>*

$$t_1[\text{Apellido}] = \text{"Pérez"}$$

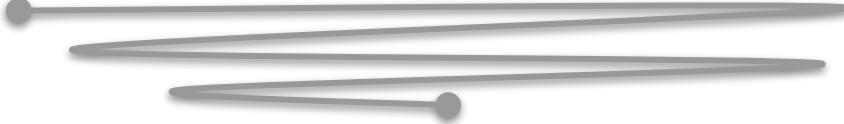
$$t_1.\text{Apellido} = \text{"Pérez"}$$

$$t_1[2] = \text{"Pérez"}$$

$$t_1[\text{Fecha_de_Nacimiento}, \text{Apellido}] = \langle 03/07/1990, \text{"Pérez"} \rangle$$

$$t_1.(\text{Fecha_de_Nacimiento}, \text{Apellido}) = \langle 03/07/1990, \text{"Pérez"} \rangle$$

# Modelo Relacional - Restricciones



## Restricciones de Dominio

Las restricciones de dominio especifican que dentro de cada tupla, el valor de un atributo  $A$  debe ser un valor atómico del dominio  $\text{dom}(A)$ .

### Ejemplo.

*ESTUDIANTE(Nombre, Apellido, LU, EMail, Fecha\_de\_nacimiento)*

$\text{dom}(\text{Nombre}) = \text{NombresDePersonas};$

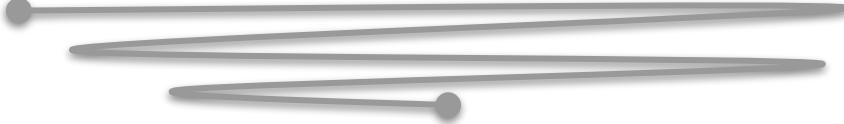
$\text{dom}(\text{Apellidos}) = \text{ApellidosDePersonas};$

$\text{dom}(\text{LU}) = \text{LUDeLaFacultad};$

$\text{dom}(\text{EMail}) = \text{EMailsExistentes};$

$\text{dom}(\text{Fecha_de_Nacimiento}) = \text{FechasDeNacimientoDePersonasVivas}$

# Modelo Relacional - Restricciones



## Restricciones de clave

Una relación está definida como un conjunto de tuplas. ¿Puede tener tuplas repetidas?

No. Por definición, todos los elementos de un conjunto son distintos; por tanto, todas las tuplas en una relación también deben serlo.

### Ejemplo. Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Maria	Perez	123/09	mperez@gm.com	03/07/1990
Diego	Garcia	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Manuel	Perez	66/09	mlopez@yahoo.com	03/07/1990

No puede haber dos tuplas con la misma combinación de valores en la totalidad de sus atributos

# Modelo Relacional - Restricciones

## Restricciones de clave

Habitualmente existen subconjuntos de atributos (de una relación  $R$ ) con la propiedad de que dos tuplas (en cualquier relación  $r$  de  $R$ ) no van a tener la misma combinación de valores para estos atributos.

### Ejemplo.

### Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Maria	Perez	123/09	mperez@gm.com	03/07/1990
Diego	Garcia	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Manuel	Perez	66/09	mlopez@yahoo.com	03/07/1990

¿Existe algún subconjunto de atributos para el cual, por definición, ningún par de tuplas va a tener los mismos valores? Si, por ejemplo  $\langle \text{Nombre}, \text{Apellido}, \text{LU} \rangle$

# Modelo Relacional - Restricciones

## Restricciones de clave

Si denominamos SK a uno de estos subconjuntos de atributos, entonces para dos tuplas cualesquiera distintas  $t_1$  y  $t_2$  en una relación  $r$  de  $R$ , tenemos la restricción:  $t_1[SK] \neq t_2[SK]$

Ejemplo. Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Maria	Perez	123/09	mperez@gm.com	03/07/1990
Diego	Garcia	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Manuel	Perez	66/09	mlopez@yahoo.com	03/07/1990

SK se denomina Superclave

Sea  $SK = < \text{Nombre}, \text{Apellido}, LU >$

Para dos tuplas cualesquiera distintas  $t_1$  y  $t_2$  en Estudiante  $\rightarrow t_1[SK] \neq t_2[SK]$

Observar que no pasa para cualquier subconjunto (ver qué pasa con  $< \text{Apellido}, \text{Fecha\_de\_Nacimiento} >$ )

Toda relación tiene al menos una superclave predeterminada ¿Cuál es? El conjunto de todos sus atributos

# Modelo Relacional - Restricciones

## Restricciones de clave

Definimos Clave como una superclave mínima (una superclave de la cual no podemos eliminar ningún atributo y que siga siendo superclave).

Ejemplo. Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento
Maria	Pérez	123/09	mperez@gm.com	03/07/1990
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990
Manuel	Pérez	66/09	mlopez@yahoo.com	03/07/1990

Dar un ejemplo de superclave y clave de Estudiante

< Nombre, Apellido, LU > es una superclave de Estudiante

< LU > es una clave de Estudiante

Importante: un conjunto de atributos que constituye una clave es una propiedad del esquema de relación; es una restricción que debe mantenerse en cada estado de relación válido del esquema. Una clave está determinada por el significado de los atributos, y la propiedad es fija en el tiempo: debe mantenerse cuando insertemos nuevas tuplas en la relación.

# Modelo Relacional - Restricciones

## Restricciones de clave

¿Puede un esquema de relación contar con más de una clave?

Sí, en ese caso, cada una de ellas recibe el nombre de clave candidata

Ejemplo. Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento	CUIL
María	Pérez	123/09	mperez@gm.com	03/07/1990	27-38345129-9
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991	20-37345989-2
Laura	Martín	344/09	lmartin@jmail.com.ar	11/11/1990	27-95349333-1
Manuel	Pérez	66/09	mlopez@yahoo.com	03/07/1990	20-38345998-3

Dar 2 ejemplos de clave candidata

<LU> es una clave candidata

<CUIL> también es una clave candidata

Es común designar una de ellas como la clave principal (clave primaria) de la relación, y será la que se utilice para identificar las tuplas en la relación. En nuestro ejemplo: <LU>

La elección de una clave candidata como clave primaria es algo arbitrario; sin embargo, es preferible elegir una que tenga un solo atributo, o un pequeño número de ellos.

# Modelo Relacional - Restricciones

## Restricciones de entidad

El valor de ninguna clave primaria puede ser **NULL**.

### Ejemplo.

### Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento	CUIL
María	Pérez	123/09	mperez@gm.com	03/07/1990	27-38345129-9
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991	20-37345989-2
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990	27-95349333-1
Manuel	Pérez	66/09	mlopez@yahoo.com	03/07/1990	20-38345998-3



Ningún valor de LU puede ser **NULL**

# Modelo Relacional - Restricciones

## Restricciones de integridad referencial

Informalmente, una tupla de una relación que hace referencia a otra relación debe hacer referencia a una tupla existente de esa relación

Ejemplo.

Profesor

Nombre	Apellido	Legajo	Email	Fecha_de_Nacimiento	Depto
Nadia	Gómez	123.456	ngomez@gm.com	03/08/1974	DC
Pedro	Díaz	345.234	pdiaz@df.com.ar	11/10/1985	DF
Natalia	Vélez	234.564	nvelez@jmail.com.ar	11/11/1990	DC

Estos códigos tienen que existir en la relación  
Departamento

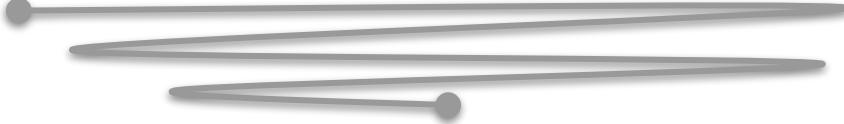
Departamento

Código Departamento	Nombre
DF	Departamento de Física
DC	Departamento de Computación
DM	Departamento de Matemática

El atributo Depto de la relación Profesor es una clave foránea (foreign key) que hace referencia a la relación Departamento

Las restricciones de integridad referencial están especificadas entre dos relaciones y se utilizan para mantener la consistencia entre las tuplas de dos relaciones

## Modelo Relacional - Restricciones



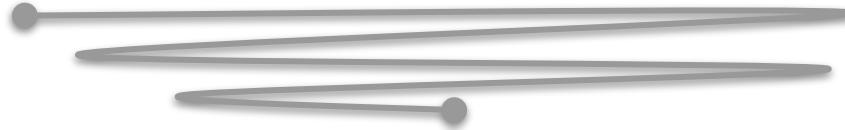
### Restricciones de integridad referencial

Formalmente, un conjunto de atributos FK en una relación R1 es una foreign key de R1 que referencia a la relación R2 si satisface las siguientes reglas:

1. Los atributos en FK tienen el mismo dominio, o dominios, que los atributos de clave principal PK de R2; se dice que los atributos FK referencian o hacen referencia a la relación R2.
2. Un valor de FK en una tupla  $t_1$  del estado actual  $r_1(R_1)$  aparece como valor de PK en alguna tupla  $t_2$  del estado actual  $r_2(R_2)$  o es NULL. En el caso anterior, tenemos que  $t_1[FK] = t_2[PK]$ , y decimos que la tupla  $t_1$  referencia o hace referencia a la tupla  $t_2$ .

Si se mantienen ambas condiciones, se establece una restricción de integridad referencial de R1 a R2.

## *Modelo Relacional - Restricciones*

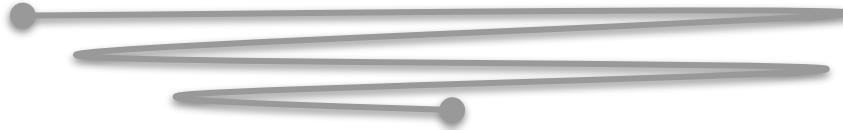


### *Restricciones de integridad semántica*

*Ejemplo. “El salario de un empleado no debe exceder el de su supervisor y que el número máximo de horas que un empleado puede trabajar a la semana es de 40 hs.”*

*Pueden implementarse dentro de las propias aplicaciones que actualizan la base de datos, o usar un lenguaje de especificación de restricciones de propósito general. Para ello existen unos mecanismos llamados triggers y aserciones.*

## *Modelo Relacional - Restricciones*



### *Restricciones de transición*

*Ejemplo. “El sueldo de un empleado sólo puede aumentar”.*

*Este tipo de restricción suele estar implementada en las aplicaciones o mediante reglas activas y triggers*

# Modelo Relacional - Restricciones

Restricciones de dependencia funcional.

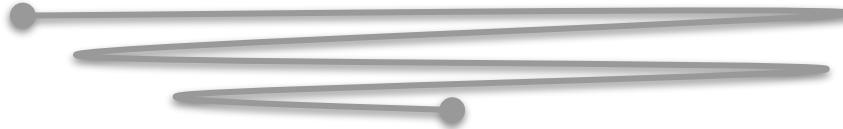
Establece una relación funcional entre dos conjuntos de atributos X e Y. Esta restricción especifica que el valor de X determina el de Y en todos los estados de una relación; está indicada como una dependencia funcional  $X \rightarrow Y$ .

## Ejemplo. Estudiante

Nombre	Apellido	LU	Email	Fecha_de_Nacimiento	CUIL
María	Pérez	123/09	mperez@gm.com	03/07/1990	27-38345129-9
Diego	García	22/10	dgarcia@dfg.com.ar	10/10/1991	20-37345989-2
Laura	Martin	344/09	lmartin@jmail.com.ar	11/11/1990	27-95349333-1
Manuel	Pérez	66/09	mlopez@yahoo.com	03/07/1990	20-38345998-3

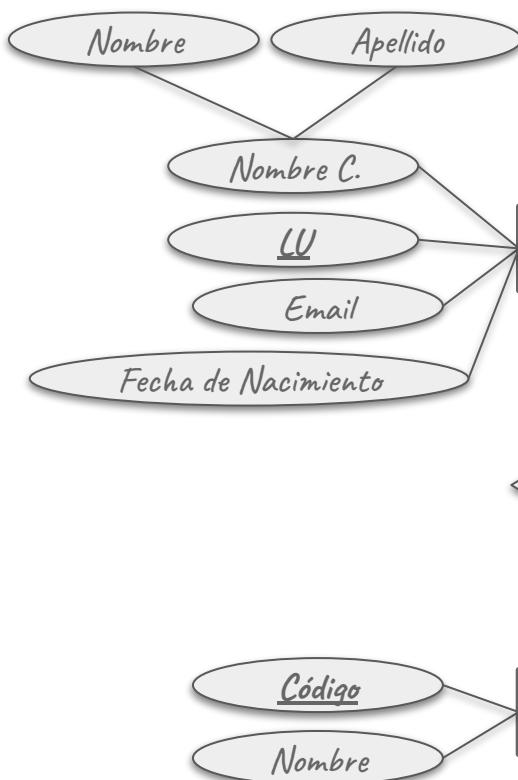
$$\left. \begin{array}{l} X = \langle LU \rangle \\ Y = \langle \text{Nombre, Apellido} \rangle \end{array} \right\} X \rightarrow Y$$

## *Modelo Relacional - Mapeo desde DER*

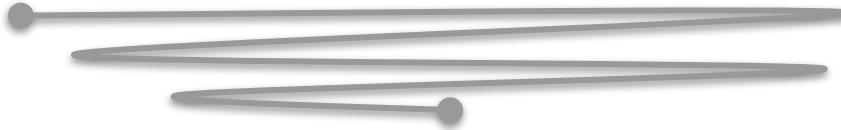


*¿Cómo pasar un DER al Modelo Relacional ?*

# Modelo Relacional - Mapeo desde DER



# *Modelo Relacional - Mapeo desde DER*

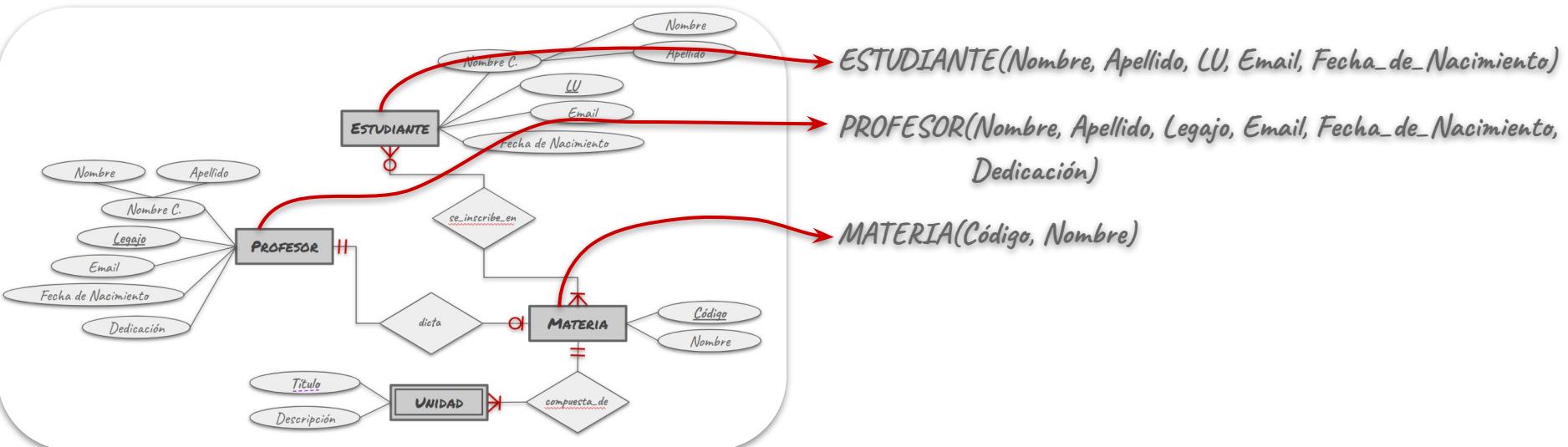


*Paso 1. Tipo de Entidades Fuertes*

# Modelo Relacional - Mapeo desde DER

## Paso 1. Tipo de Entidades Fuertes

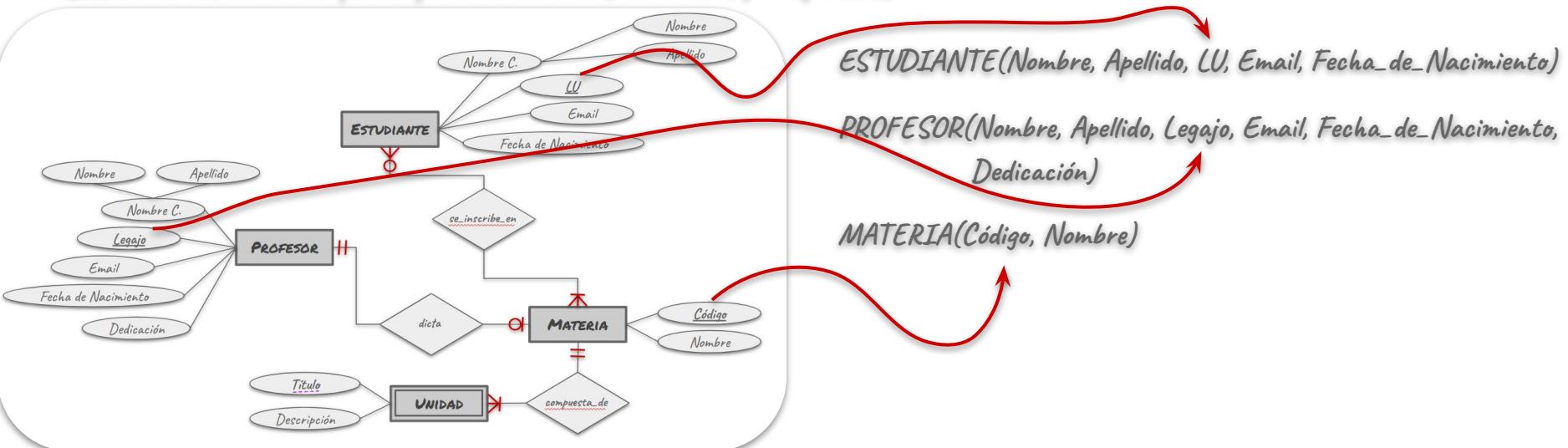
- a. Por cada tipo de entidad (fuerte)  $E$  del DER, crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).



# Modelo Relacional - Mapeo desde DER

## Paso 1. Tipo de Entidades Fuertes

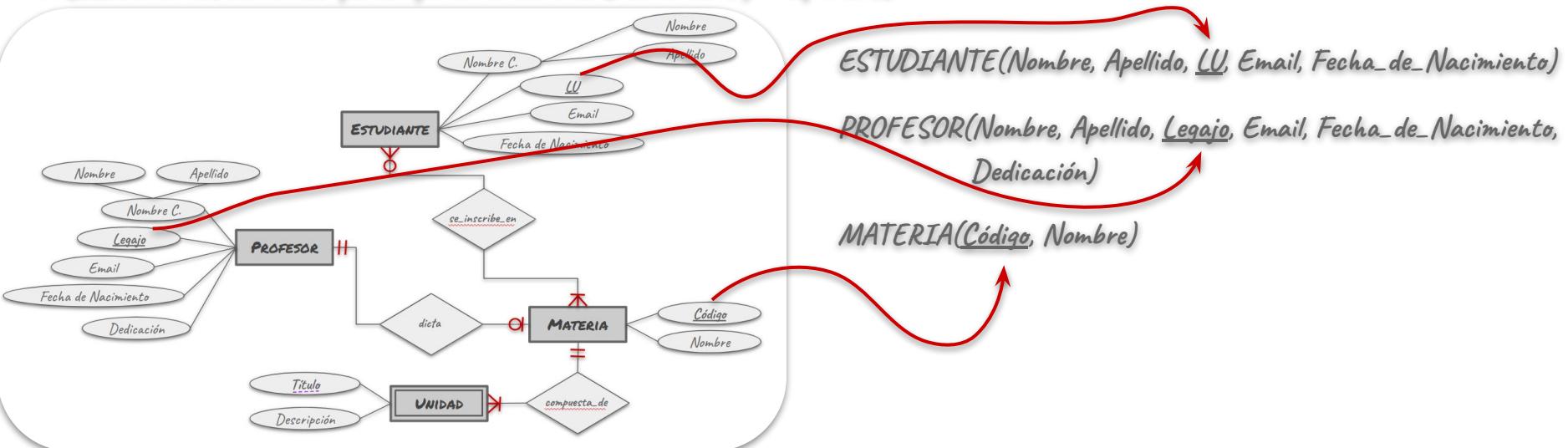
- Por cada tipo de entidad (fuerte) E del DER, crear una relación R que incluya todos los atributos simples de E (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- Seleccionar los atributos que componen la clave de E como clave principal de R.



# Modelo Relacional - Mapeo desde DER

## Paso 1. Tipo de Entidades Fuertes

- Por cada tipo de entidad (fuerte) E del DER, crear una relación R que incluya todos los atributos simples de E (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- Seleccionar los atributos que componen la clave de E como clave principal de R.



## *Modelo Relacional - Mapeo desde DER*

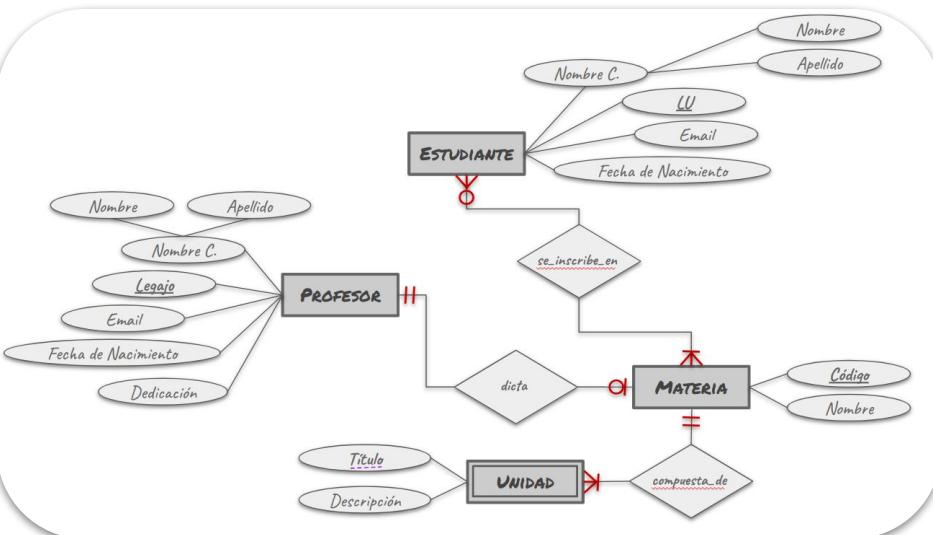


*Paso 2. Tipo de Entidades Débiles*

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- a. Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).



**ESTUDIANTE**(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)

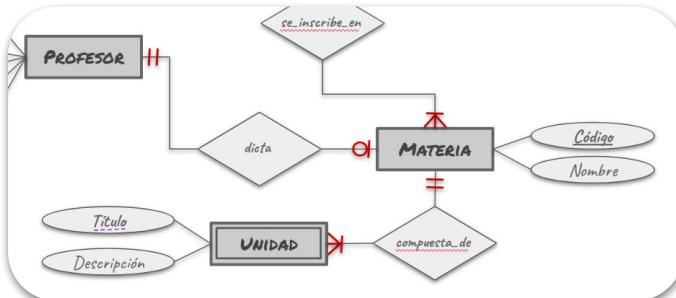
**PROFESOR**(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)

**MATERIA**(Código, Nombre)

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- a. Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).



ESTUDIANTE(Nombre, Apellido, CU, Email, Fecha\_de\_Nacimiento)

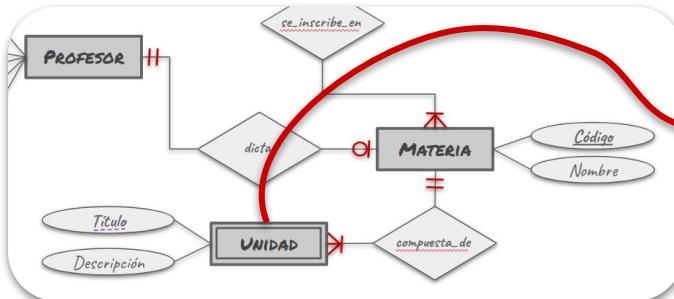
PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedicación)

MATERIA(Código, Nombre)

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- a. Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).



ESTUDIANTE(Nombre, Apellido, UU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedicación)

MATERIA(Código, Nombre)

UNIDAD(Título, Descripción)

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

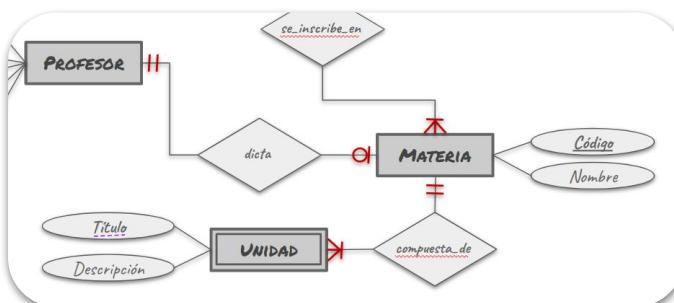
- Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- Incluir en  $R$  los atributos correspondientes a la clave de  $P$  (estos atributos van a conformar una foreign key).

ESTUDIANTE(Nombre, Apellido, CU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedicación)

MATERIA(Código, Nombre)

UNIDAD(Título, Descripción)



# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

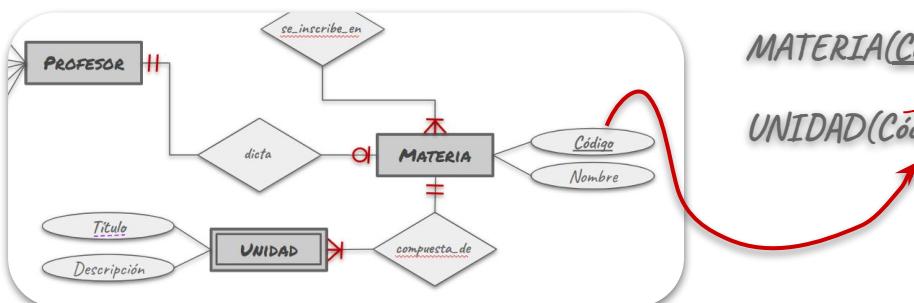
- Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- Incluir en  $R$  los atributos correspondientes a la clave de  $P$  (estos atributos van a conformar una foreign key).

ESTUDIANTE(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedicación)

MATERIA(Código, Nombre)

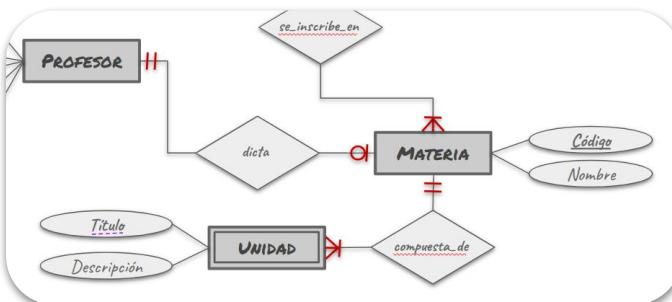
UNIDAD(Código, Título, Descripción)  
*foreign key*



# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- a. Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- b. Incluir en  $R$  los atributos correspondientes a la clave de  $P$  (estos atributos van a conformar una foreign key).
- c. Seleccionar tanto los atributos que componen la clave de  $E$ , como los atributos que componen la clave de  $P$  y designarla como clave primaria de  $R$ .



ESTUDIANTE(Nombre, Apellido, CU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedición)

MATERIA(Código, Nombre)

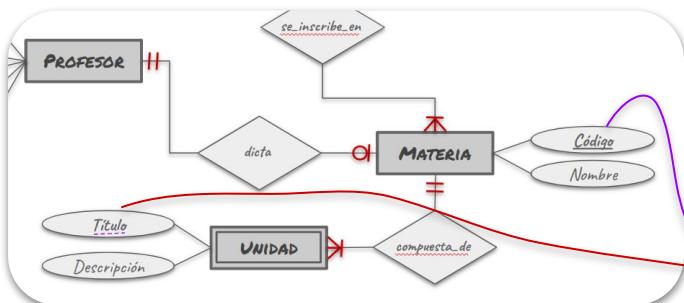
↑ foreign key

UNIDAD(Código, Título, Descripción)

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- Incluir en  $R$  los atributos correspondientes a la clave de  $P$  (estos atributos van a conformar una foreign key).
- Seleccionar tanto los atributos que componen la clave de  $E$ , como los atributos que componen la clave de  $P$  y designarla como clave primaria de  $R$ .



ESTUDIANTE(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedición)

MATERIA(Código, Nombre)

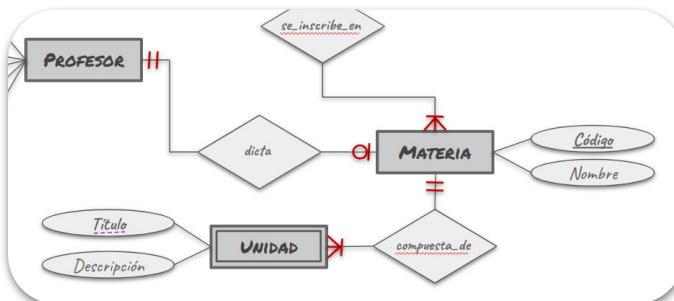
foreign key

UNIDAD(Código, Título, Descripción)

# Modelo Relacional - Mapeo desde DER

## Paso 2. Tipo de Entidades Débiles

- a. Por cada tipo de entidad (débil)  $E$  del DER y cuyo tipo de entidad propietaria es  $P$ , crear una relación  $R$  que incluya todos los atributos simples de  $E$  (para el caso de los atributos compuestos, incluir únicamente los atributos simples que los componen).
- b. Incluir en  $R$  los atributos correspondientes a la clave de  $P$  (estos atributos van a conformar una foreign key).
- c. Seleccionar tanto los atributos que componen la clave de  $E$ , como los atributos que componen la clave de  $P$  y designarla como clave primaria de  $R$ .



ESTUDIANTE(Nombre, Apellido, CU, Email, Fecha\_de\_Nacimiento)

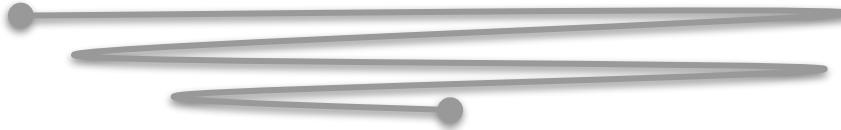
PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedición)

MATERIA(Código, Nombre)

↑ foreign key

UNIDAD(Código, Título, Descripción)

## *Modelo Relacional - Mapeo desde DER*

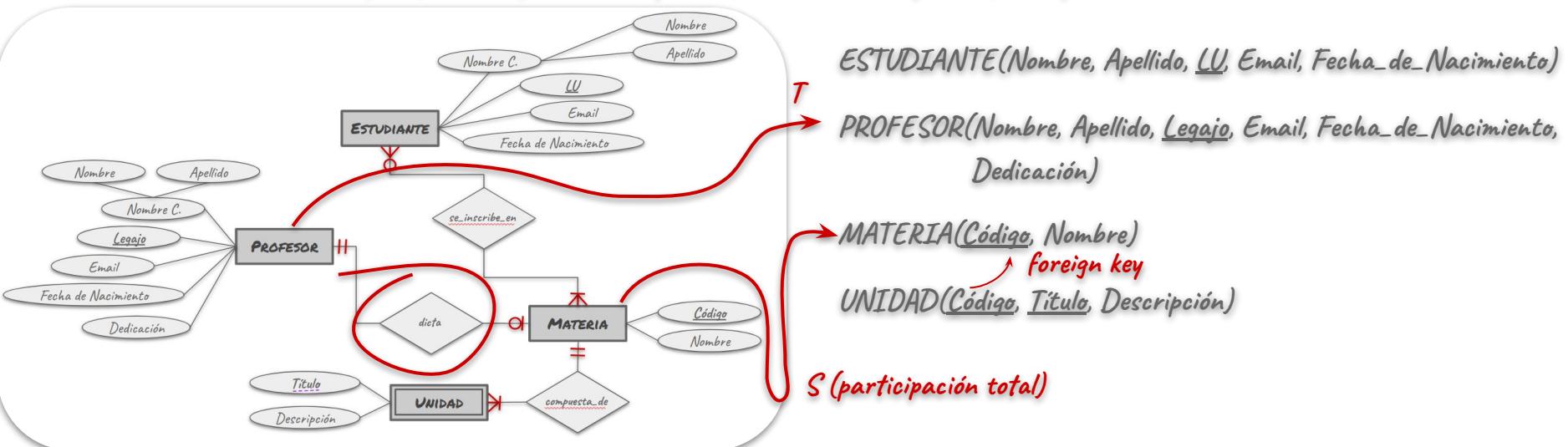


*Paso 3. Relación uno-a-uno (1:1)*

# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

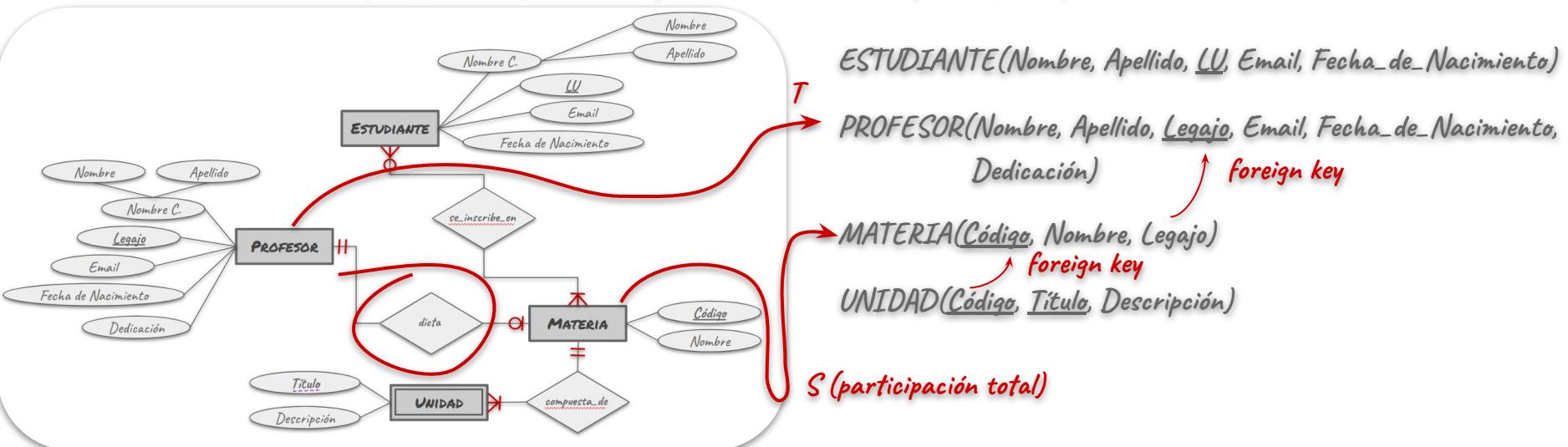
Hay 3 opciones: 1. Metodología de la foreign key. Seleccionar una de las relaciones ya mapeadas (por ejemplo, S) e incluya como foreign key en S la clave primaria de T. Lo mejor es elegir, en el papel de S, un tipo de entidad con participación total en R. Incluir todos los atributos simples (o los componentes simples de los atributos compuestos) del tipo de relación 1:1 de R como atributos de S.



# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

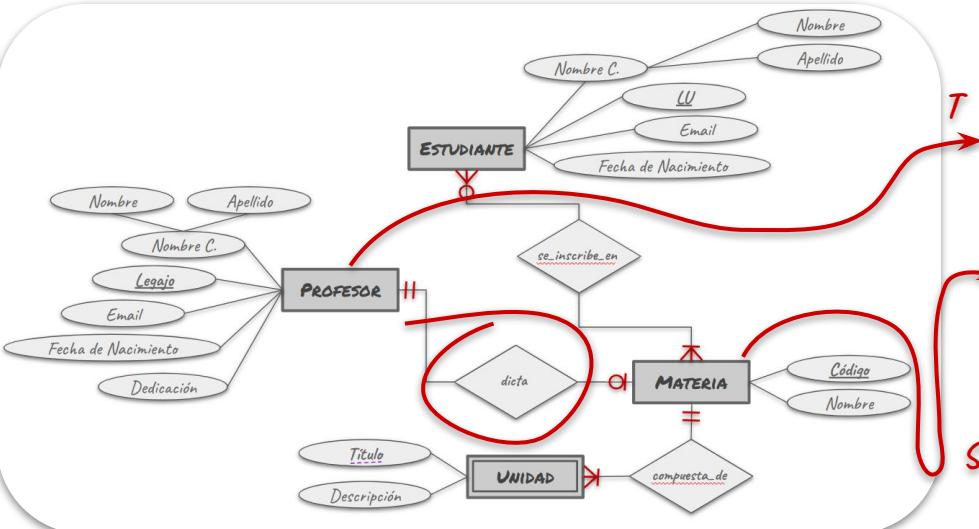
Hay 3 opciones: 1. Metodología de la foreign key. Seleccionar una de las relaciones ya mapeadas (por ejemplo, S) e incluya como foreign key en S la clave primaria de T. Lo mejor es elegir, en el papel de S, un tipo de entidad con participación total en R. Incluir todos los atributos simples (o los componentes simples de los atributos compuestos) del tipo de relación 1:1 de R como atributos de S.



# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

Hay 3 opciones: 2. Metodología de la relación mezclada. Fusionar los dos tipos de entidad (S y T) y la relación en una sola relación. Esto puede ser apropiado cuando las dos participaciones son totales.



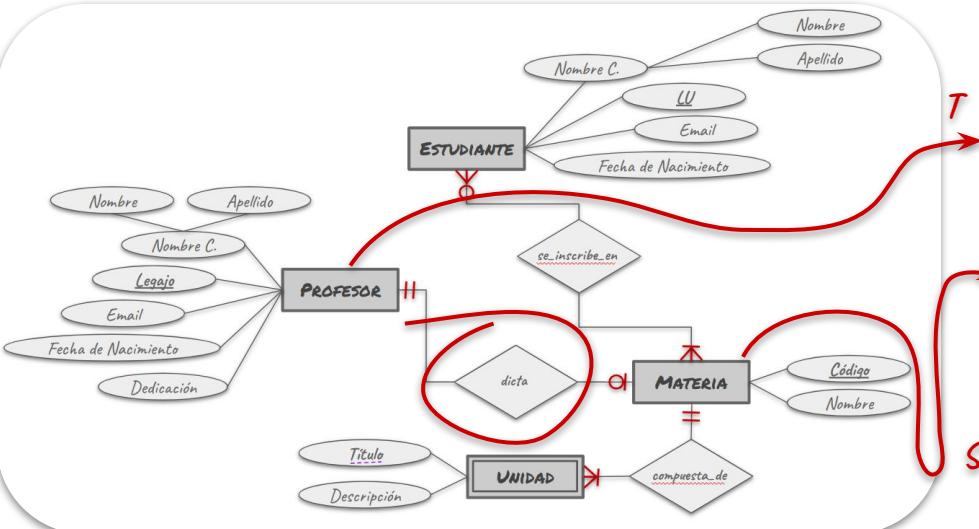
PROFESOR\_MATERIA(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación, Materia\_Codigo, Materia\_Nombre)  
ESTUDIANTE(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)  
~~PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicacion)~~  
~~MATERIA(Código, Nombre)~~  
UNIDAD(Código, Titulo, Descripción)

S → T

# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

Hay 3 opciones: 2. Metodología de la relación mezclada. Fusionar los dos tipos de entidad (S y T) y la relación en una sola relación. Esto puede ser apropiado cuando las dos participaciones son totales.



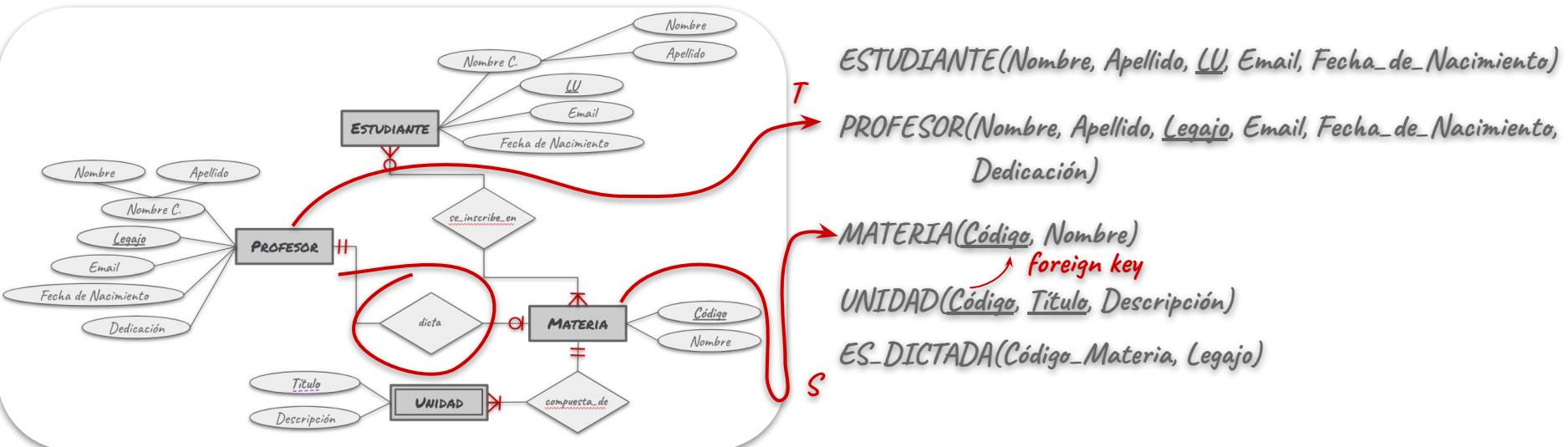
PROFESOR\_MATERIA(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación, Materia\_Codigo, Materia\_Nombre)  
ESTUDIANTE(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento)  
~~PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)~~  
~~MATERIA(Código, Nombre)~~  
UNIDAD(Código, Título, Descripción)

En nuestro caso, como NO es una participación  
-> total tiene problemas con los NULL

# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

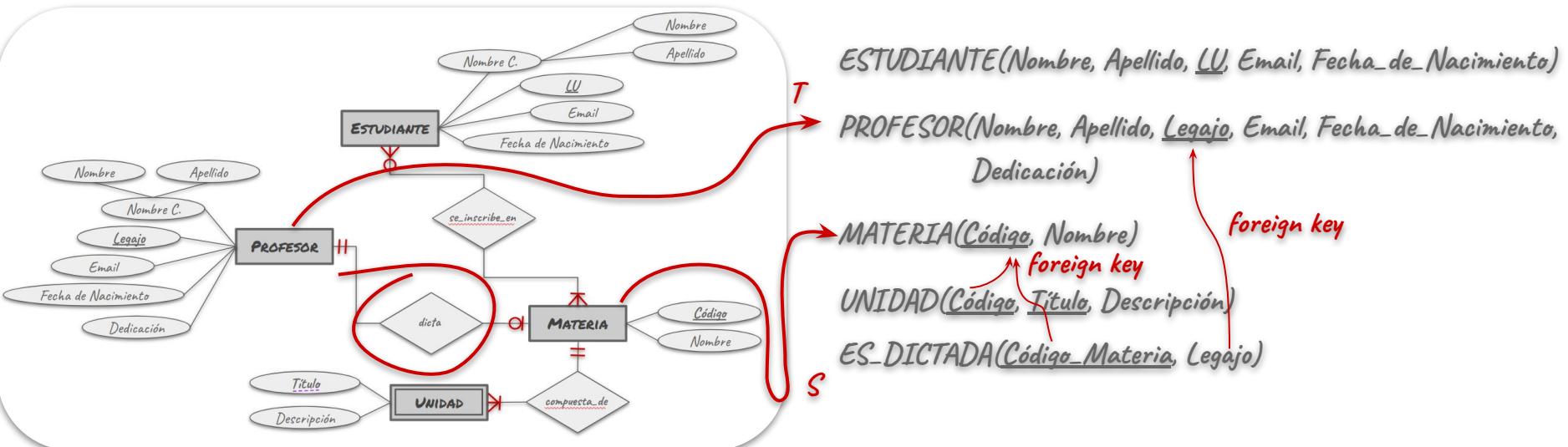
Hay 3 opciones: 3. Metodología de referencia cruzada o relación de relación. Crear una tercera relación R con el propósito de crear una referencia cruzada de las claves primarias de las relaciones S y T que representan los tipos de entidad.



# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

Hay 3 opciones: 3. Metodología de referencia cruzada o relación de relación. Crear una tercera relación R con el propósito de crear una referencia cruzada de las claves primarias de las relaciones S y T que representan los tipos de entidad.

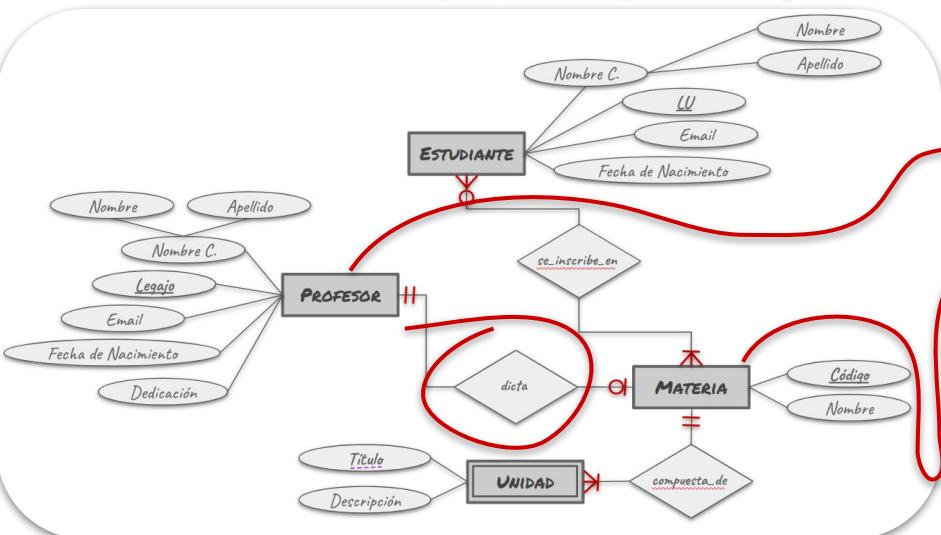


# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

En general, vamos a optar por esta primera opción

Hay 3 opciones: 1. Metodología de la foreign key. Seleccionar una de las relaciones ya mapeadas (por ejemplo, S) e incluya como foreign key en S la clave primaria de T. Lo mejor es elegir, en el papel de S, un tipo de entidad con participación total en R. Incluir todos los atributos simples (o los componentes simples de los atributos compuestos) del tipo de relación 1:1 de R como atributos de S.



ESTUDIANTE(Nombre, Apellido, LL, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)

MATERIA(Código, Nombre)

UNIDAD(Código, Título, Descripción)

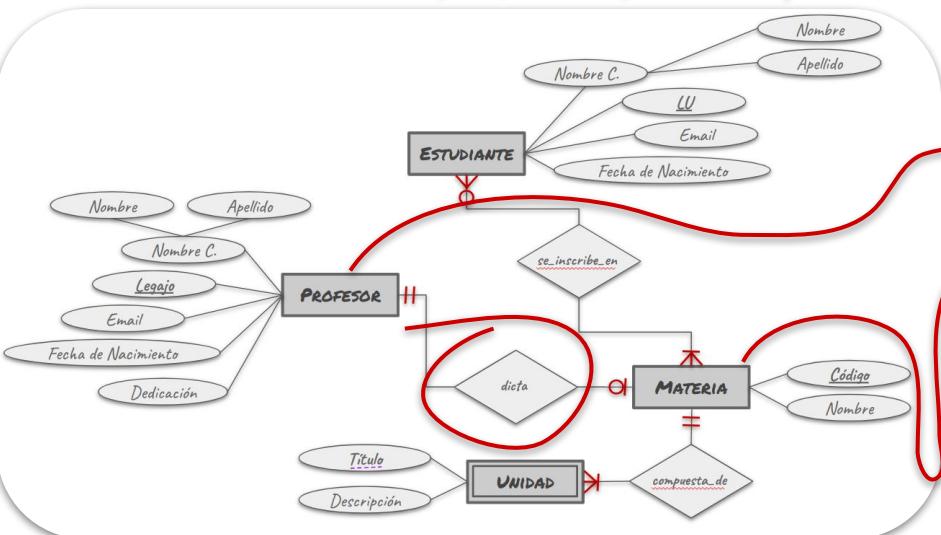
S (participación total)

# Modelo Relacional - Mapeo desde DER

## Paso 3. Relación uno-a-uno (1:1)

En general, vamos a optar por esta primera opción

Hay 3 opciones: 1. Metodología de la foreign key. Seleccionar una de las relaciones ya mapeadas (por ejemplo, S) e incluya como foreign key en S la clave primaria de T. Lo mejor es elegir, en el papel de S, un tipo de entidad con participación total en R. Incluir todos los atributos simples (o los componentes simples de los atributos compuestos) del tipo de relación 1:1 de R como atributos de S.



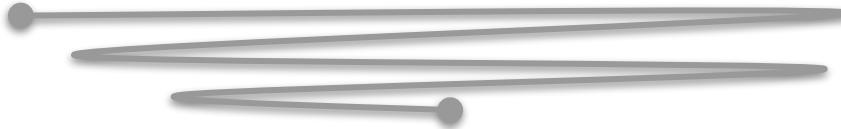
Mapa de las entidades y sus atributos resultantes:

- ESTUDIANTE**(Nombre, Apellido, LL, Email, Fecha\_de\_Nacimiento)
- PROFESOR**(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)
- MATERIA**(Código, Nombre, Legajo)
- UNIDAD**(Código, Título, Descripción)

Annotations:

- Un **Foreign key** apunta a Legajo en **PROFESOR**.
- Un **Foreign key** apunta a Código en **MATERIA**.
- Un **Foreign key** apunta a Código en **UNIDAD**.
- Un **S (participación total)** apunta a Legajo en **PROFESOR**.

## *Modelo Relacional - Mapeo desde DER*



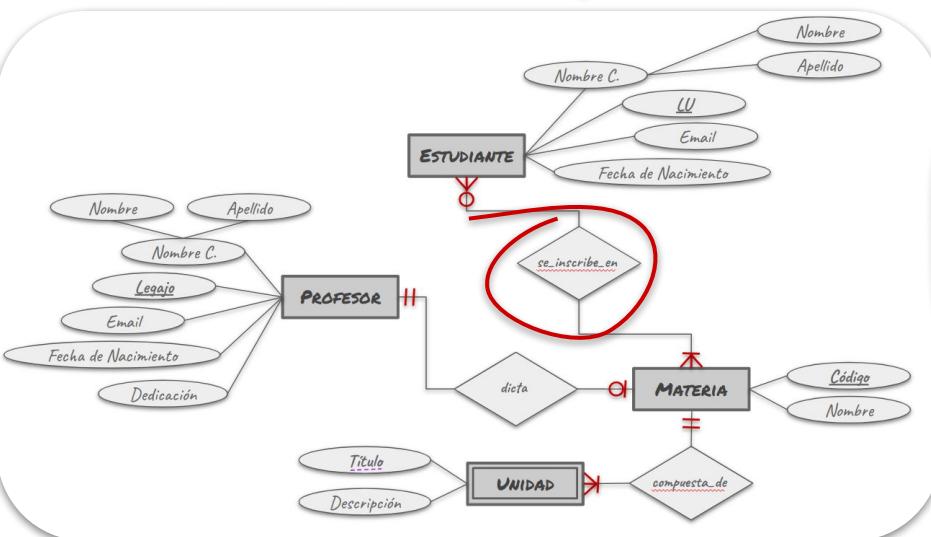
*Paso 4. Relación muchos-a-muchos (N:M)*

# Modelo Relacional - Mapeo desde DER

## Paso 4. Relación muchos-a-muchos (N:M)

Crear una nueva relación S para representar a la relación R. Incluir como atributos de la foreign key en S las claves primarias de las relaciones que representan los tipos de entidad participantes; su combinación formará la clave primaria de S. Incluya también cualesquiera atributos simples del tipo de relación M:N y los atributos de S.

foreign key



ESTUDIANTE(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)

MATERIA(Código, Nombre, Legajo)

UNIDAD(Código, Título, Descripción)

SE\_INSCRIBE\_EN(LU, Código\_Materia)

Foreign key

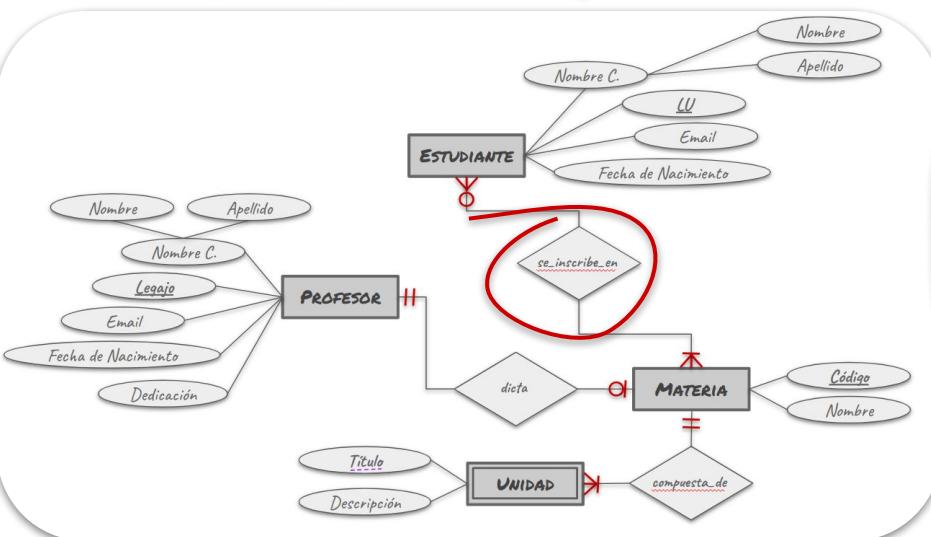
Foreign key

# Modelo Relacional - Mapeo desde DER

## Paso 4. Relación muchos-a-muchos (N:M)

Crear una nueva relación S para representar a la relación R. Incluir como atributos de la foreign key en S las claves primarias de las relaciones que representan los tipos de entidad participantes; su combinación formará la clave primaria de S. Incluya también cualesquiera atributos simples del tipo de relación M:N y los atributos de S.

foreign key



ESTUDIANTE(Nombre, Apellido, LU, Email, Fecha\_de\_Nacimiento)

PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)

MATERIA(Código, Nombre, Legajo)

UNIDAD(Código, Título, Descripción)

SE\_INSCRIBE\_EN(LU, Código\_Materia)

Foreign key

Foreign key

## *Modelo Relacional - Mapeo desde DER*

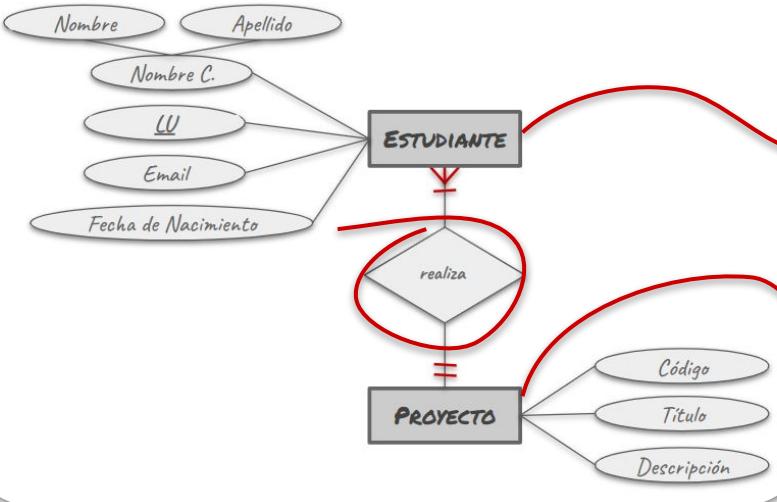


*Paso 5. Relación uno-a-muchos (1:N)*

# Modelo Relacional - Mapeo desde DER

## Paso 5. Relación uno-a-muchos (1:N)

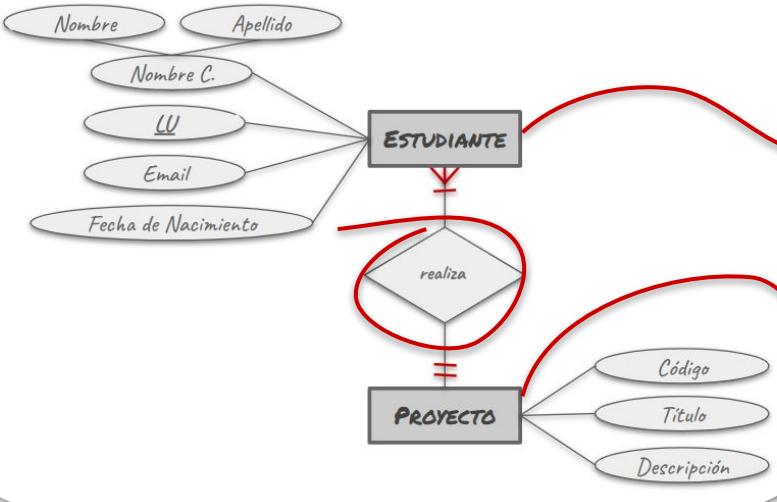
Identificar la relación  $S$  que representa el tipo de entidad participante en el lado  $N$  del tipo de relación. Incluir como foreign key en  $S$  la clave primaria de la relación  $T$  que representa el otro tipo de entidad participante en  $R$ : esto es porque cada instancia de entidad en el lado  $N$  está relacionada, a lo sumo, con una instancia de entidad del lado 1 del tipo de relación. Incluya cualesquiera atributos simples (o componentes simples de los atributos compuestos) del tipo de relación 1:N como atributos de  $S$ .



# Modelo Relacional - Mapeo desde DER

## Paso 5. Relación uno-a-muchos (1:N)

Identificar la relación  $S$  que representa el tipo de entidad participante en el lado  $N$  del tipo de relación. Incluir como foreign key en  $S$  la clave primaria de la relación  $T$  que representa el otro tipo de entidad participante en  $R$ : esto es porque cada instancia de entidad en el lado  $N$  está relacionada, a lo sumo, con una instancia de entidad del lado 1 del tipo de relación. Incluya cualesquiera atributos simples (o componentes simples de los atributos compuestos) del tipo de relación 1:N como atributos de  $S$ .



## *Modelo Relacional - Mapeo desde DER*

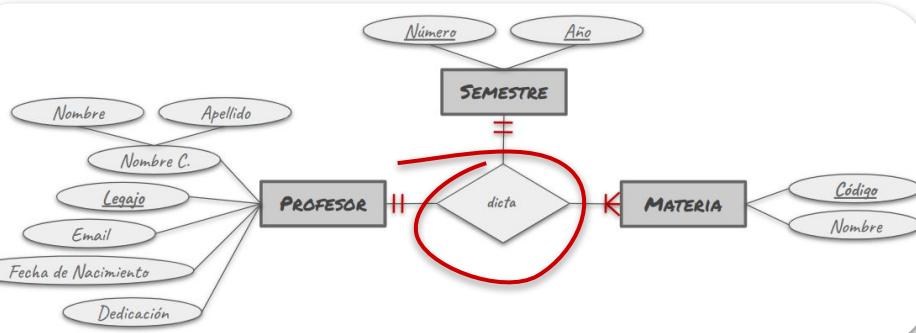


*Paso 6. Relación n-aria*

# Modelo Relacional - Mapeo desde DER

## Paso 6. Relación n-aria

Crear una nueva relación S para representar R. Incluir como atributos de la foreign key en S las claves primarias de las relaciones que representan los tipos de entidad participantes. Incluir también atributos simples del tipo de relación n-aria como atributos de S. Normalmente, la clave primaria de S es el conjunto de todas las foreign keys que refieren a los tipos de entidad participantes. No obstante, si la cardinalidad en cualquiera de los tipos de entidad E que participan en R es 1, entonces la clave primaria de S no debe incluir el atributo de la foreign key que hace referencia a E.



PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento,  
Dedicación)

MATERIA(Código, Nombre)

*foreign key*

SEMESTRE(Número, Año)

DICTA(Código\_Materia, Legajo, Número\_Semestre, Año\_Semestre)

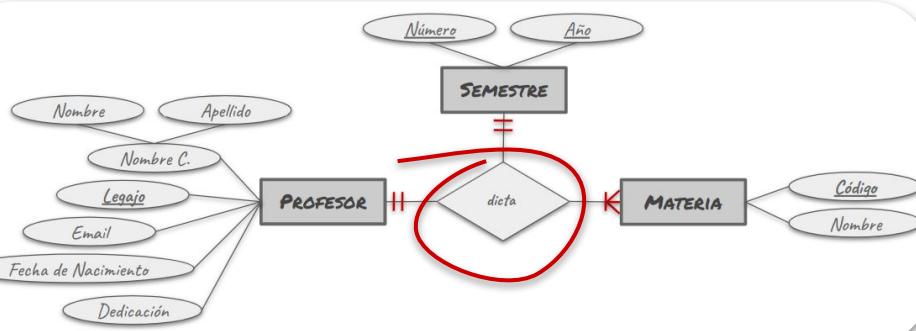
*foreign key*

*foreign key*

# Modelo Relacional - Mapeo desde DER

## Paso 6. Relación n-aria

Crear una nueva relación S para representar R. Incluir como atributos de la foreign key en S las claves primarias de las relaciones que representan los tipos de entidad participantes. Incluir también atributos simples del tipo de relación n-aria como atributos de S. Normalmente, la clave primaria de S es el conjunto de todas las foreign keys que refieren a los tipos de entidad participantes. No obstante, si la cardinalidad en cualquiera de los tipos de entidad E que participan en R es 1, entonces la clave primaria de S no debe incluir el atributo de la foreign key que hace referencia a E.



PROFESOR(Nombre, Apellido, Legajo, Email, Fecha\_de\_Nacimiento, Dedicación)

MATERIA(Código, Nombre)

foreign key

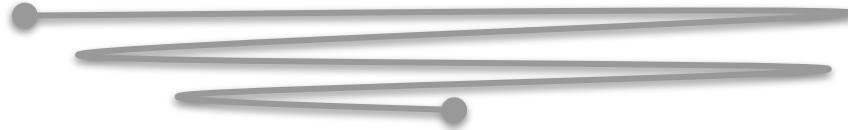
SEMESTRE(Número, Año)

foreign key

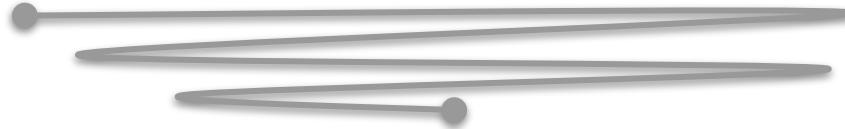
DICTA(Código\_Materia, Legajo, Número\_Semestre, Año\_Semestre)

foreign key

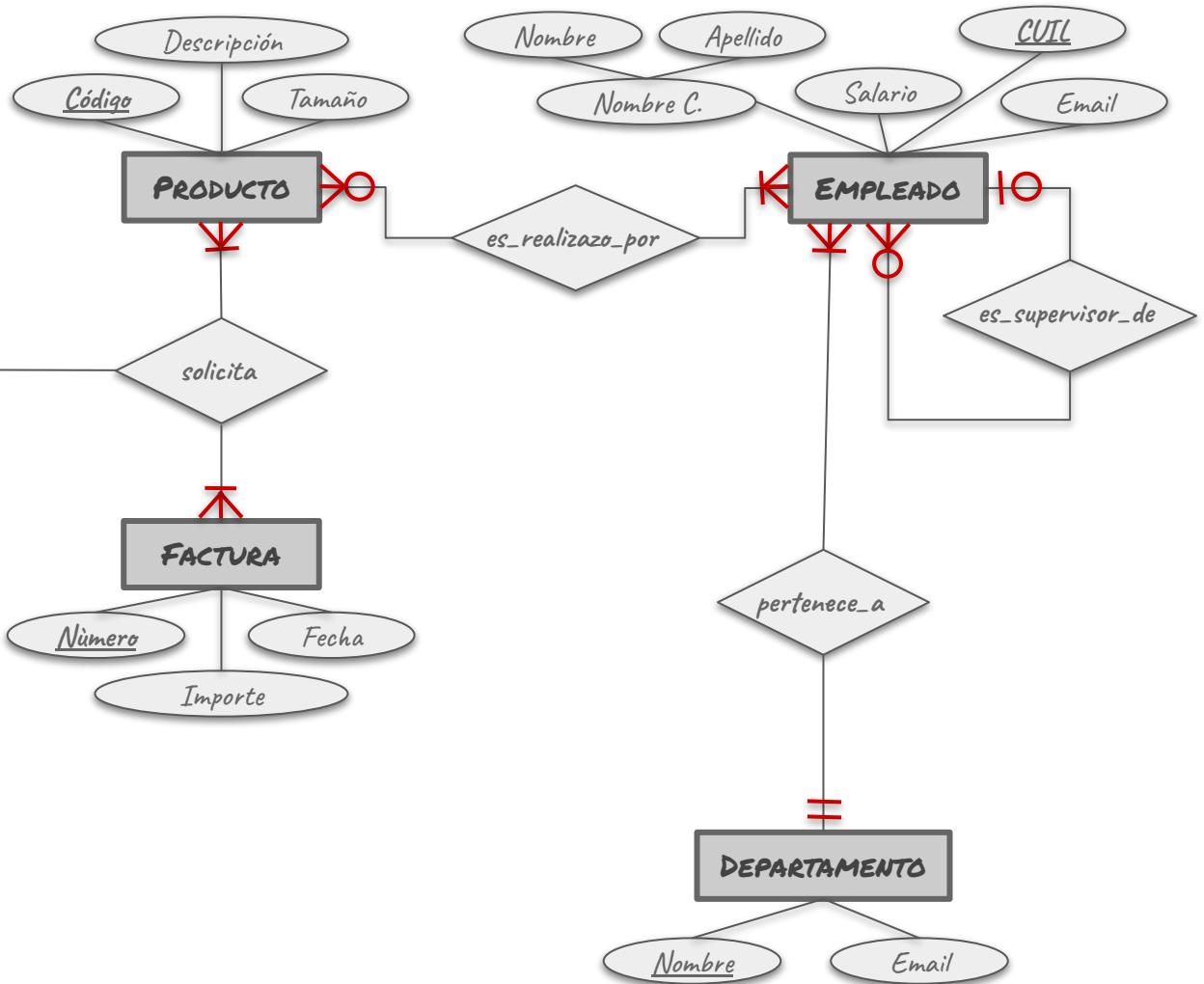
*Trabajo en equipo*



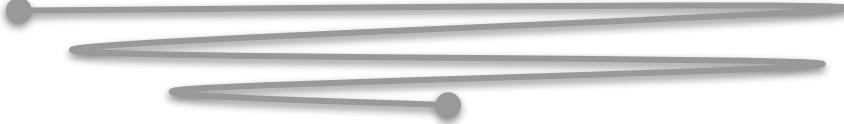
## Actividad Nro. 1 - Consigna



- ✓ Conformar grupos de 3 integrantes
- ✓ Mapear el siguiente Diagrama Entidad Relación al Modelo Relacional  
(Consultar las dudas)

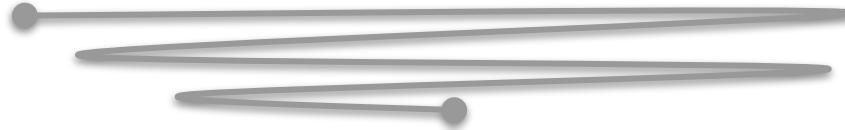


## Cierre



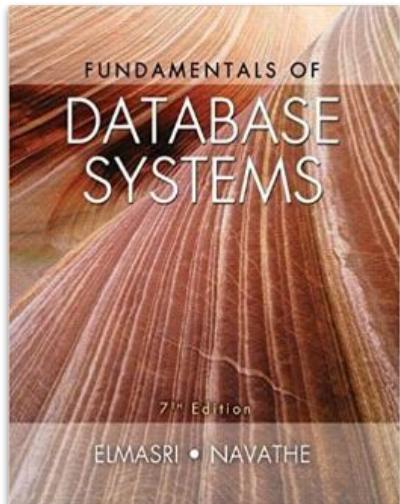
1. DER - Vista abstracta y simplificada del mundo que queremos representar, independiente del DBMS que vayamos a utilizar
2. Esquema lógico es la manera de implementar el DER en un DBMS.  
Es dependiente del DBMS que vayamos a utilizar
3. Adoptamos el modelo relacional e hicimos el pasaje del DER al modelo lógico de un DBMS relacional
4. Vimos características (y mucha notación) del modelo relacional
5. Comentamos la importancia de contar con una Arquitectura de 3 niveles para DBMS

## *Tareas para la próxima clase*

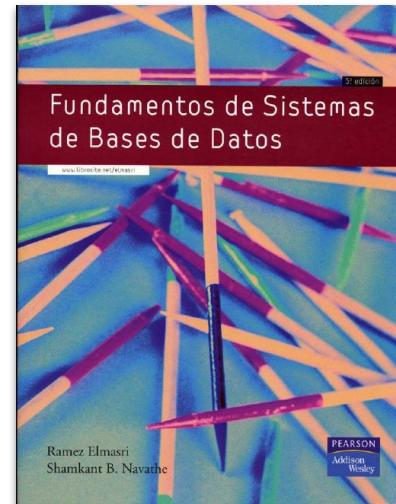


- ✓ *Resolver la guía de ejercicios de “Modelo Relacional”*

# Bibliografía



Elmasri/Navathe, *Fundamentals of Database Systems*,  
7th. Ed., Pearson, 2016.



Elmasri/Navathe, *Fundamentos de Sistemas de Bases de Datos*,  
5ta Ed., Pearson, 2007.

(Aviso. Difieren un poco en la notación)