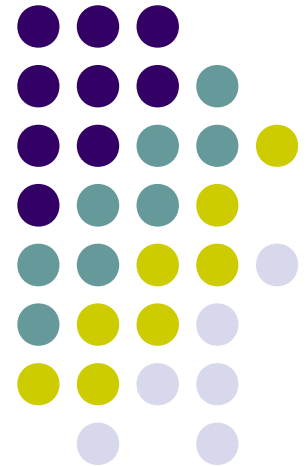


# Bases de datos

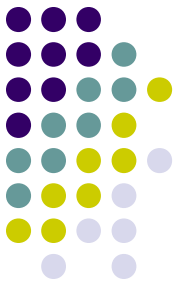
## Tema II – La base de datos relacional - RDMBS



Universidad Nacional del Litoral  
**FACULTAD DE INGENIERÍA  
Y CIENCIAS HÍDRICAS**



# RDBMS *Relational DataBase Management System*



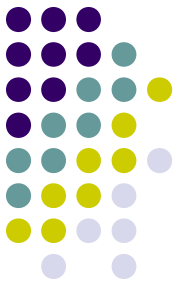
## Componentes conceptuales

**En una BD se trata de establecer conjuntos de relaciones entre dos o más elementos conceptuales.**

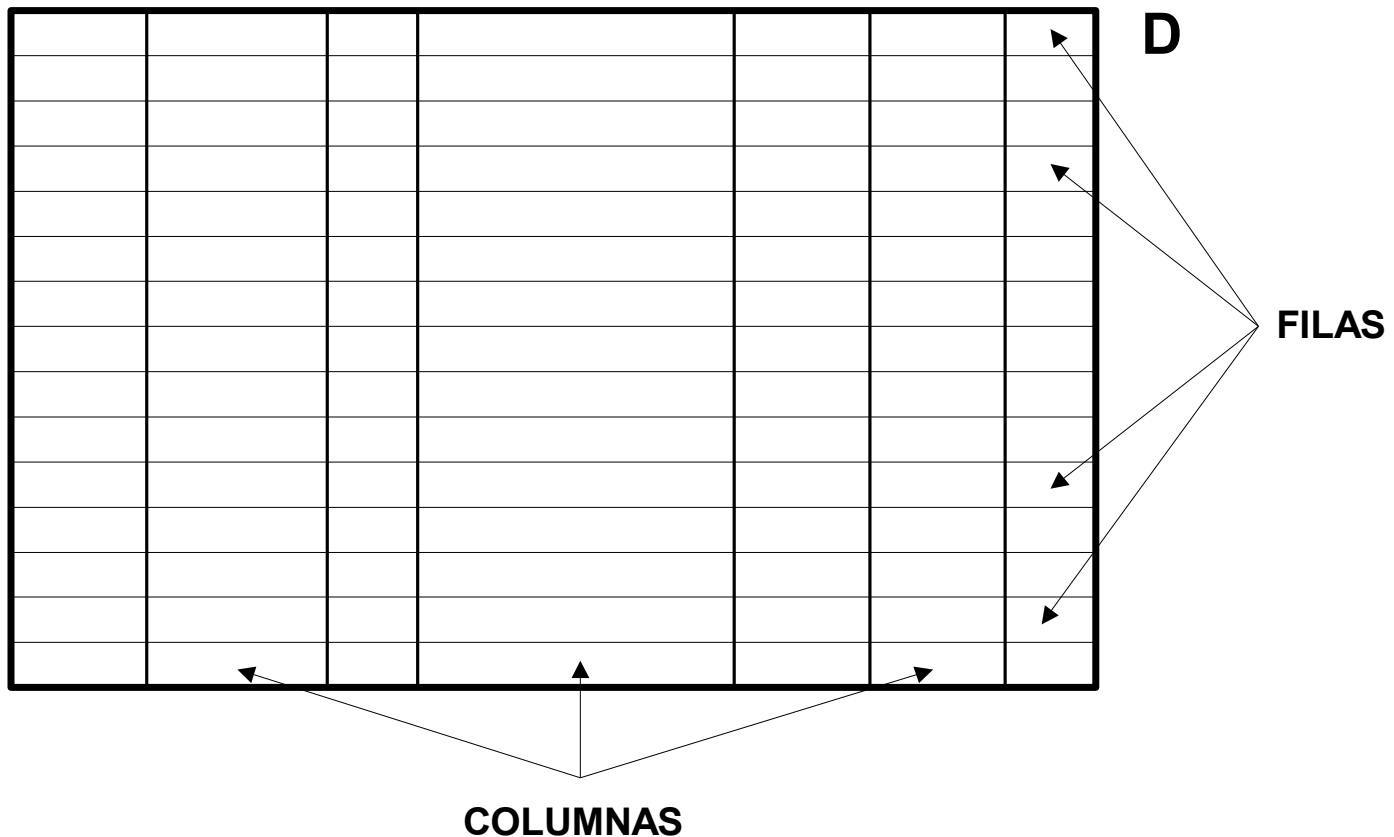
- 1. Los elementos conceptuales si bien son abstractos, representan “algo”.**
- 2. Las relaciones no resultan tan tangibles ya que no pueden ser medidas o clasificadas con claridad como los atributos de un individuo. Una relación no existe como algo separado o tangible.**

**La relación se debe tratar **del mismo modo que los datos**. Por lo tanto, esta relación se convierte en algo tan concreto como aquellos datos. La forma de trabajar con ambos, datos y relación, en la base de datos es la TABLA o MATRIZ.**

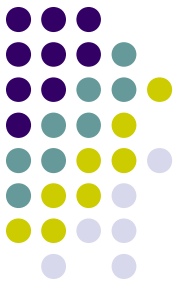
# RDBMS *Relational DataBase Management System*



## La TABLA (o matriz)



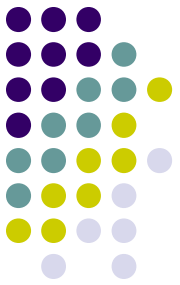
# RDBMS *Relational DataBase Management System*



## Relación 1-1

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Para cada fila de una matriz, existirá una fila correspondiente en la otra matriz.

# RDBMS *Relational DataBase Management System*



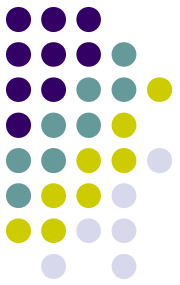
## Relación 1-1

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Para cada fila de una matriz, existirá una fila correspondiente en la otra matriz.

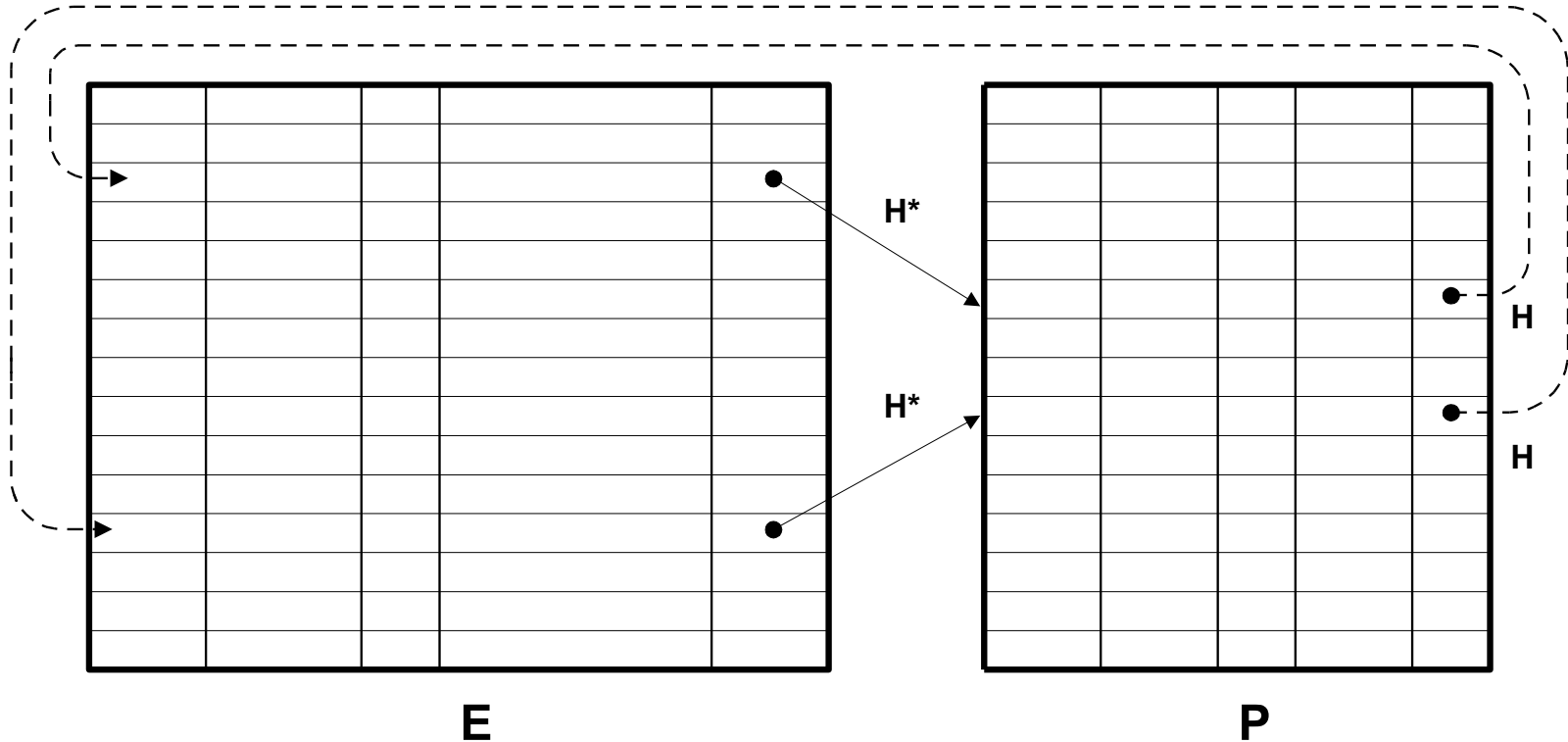
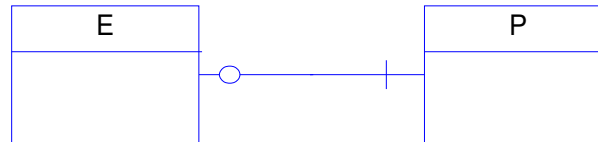
### **Solución**

- Se establecen punteros entre las filas correspondientes.

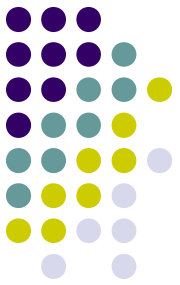
# RDBMS *Relational DataBase Management System*



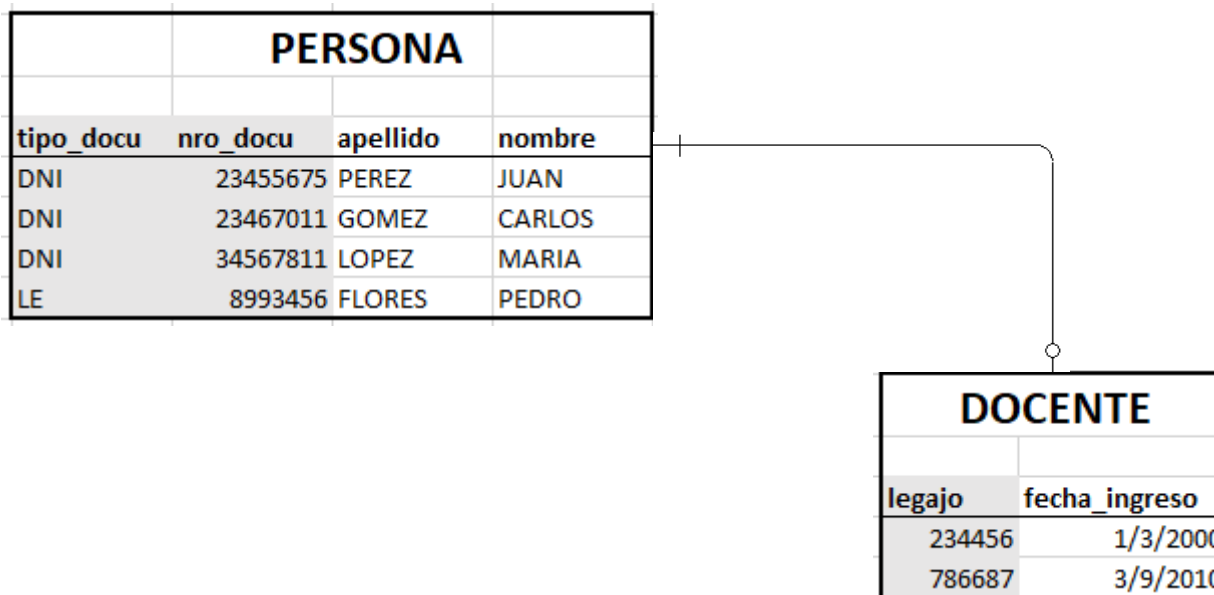
## Relación 1-1



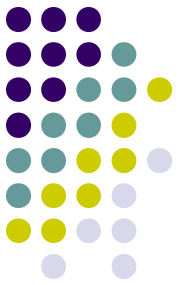
# RDBMS *Relational DataBase Management System*



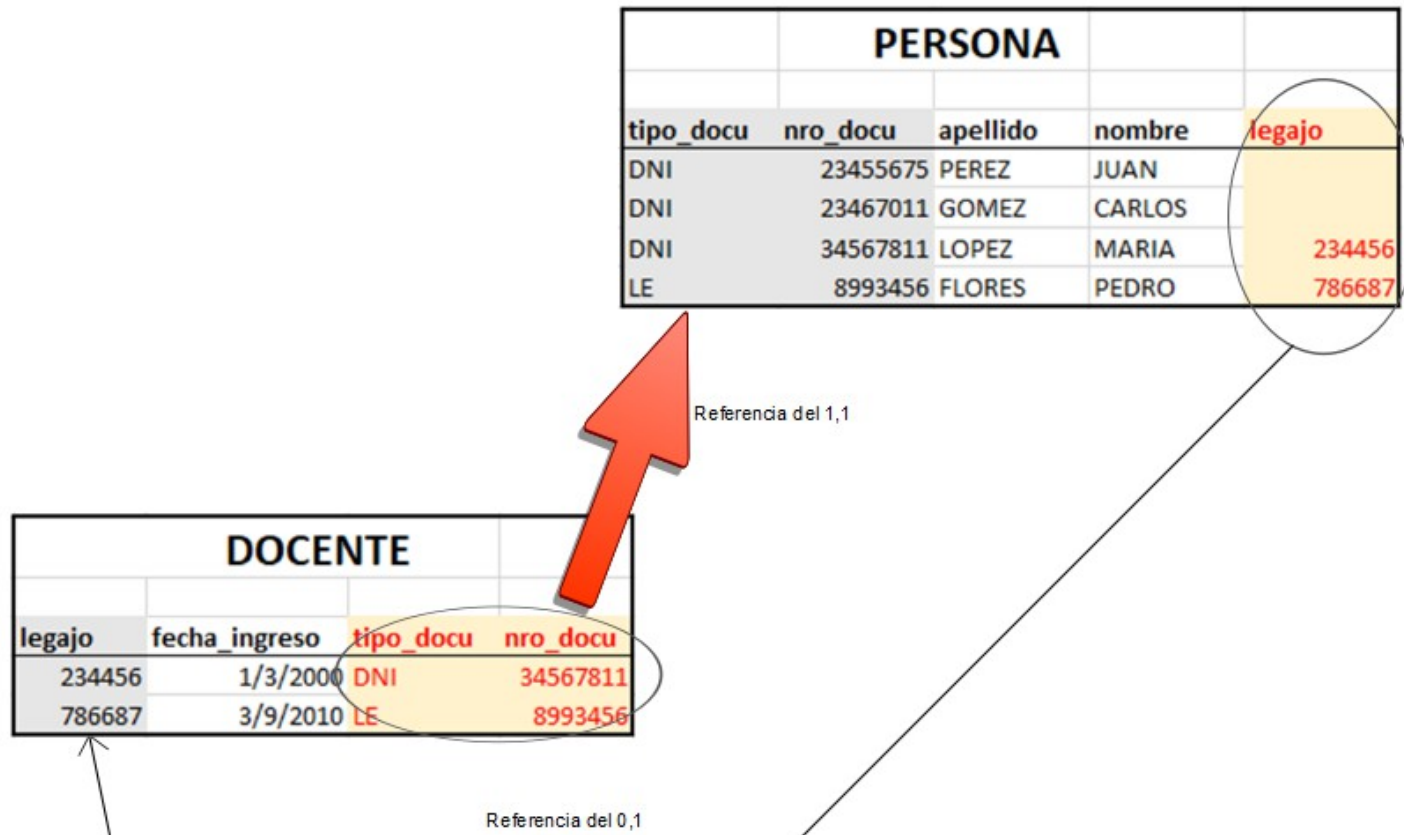
## Relación 1-1



# RDBMS *Relational DataBase Management System*

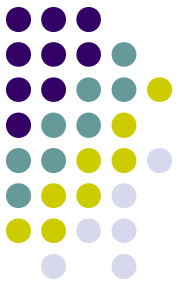


## Relación 1-1





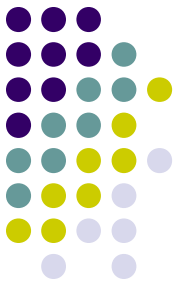
# RDBMS *Relational DataBase Management System*



## Relación n-1

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Muchas filas de una matriz pueden tener correspondencia con una misma fila de la otra matriz.

# RDBMS *Relational DataBase Management System*



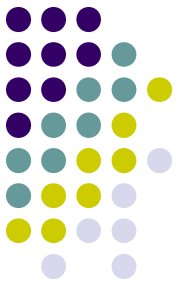
## Relación n-1

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Muchas filas de una matriz pueden tener correspondencia con una misma fila de la otra matriz.

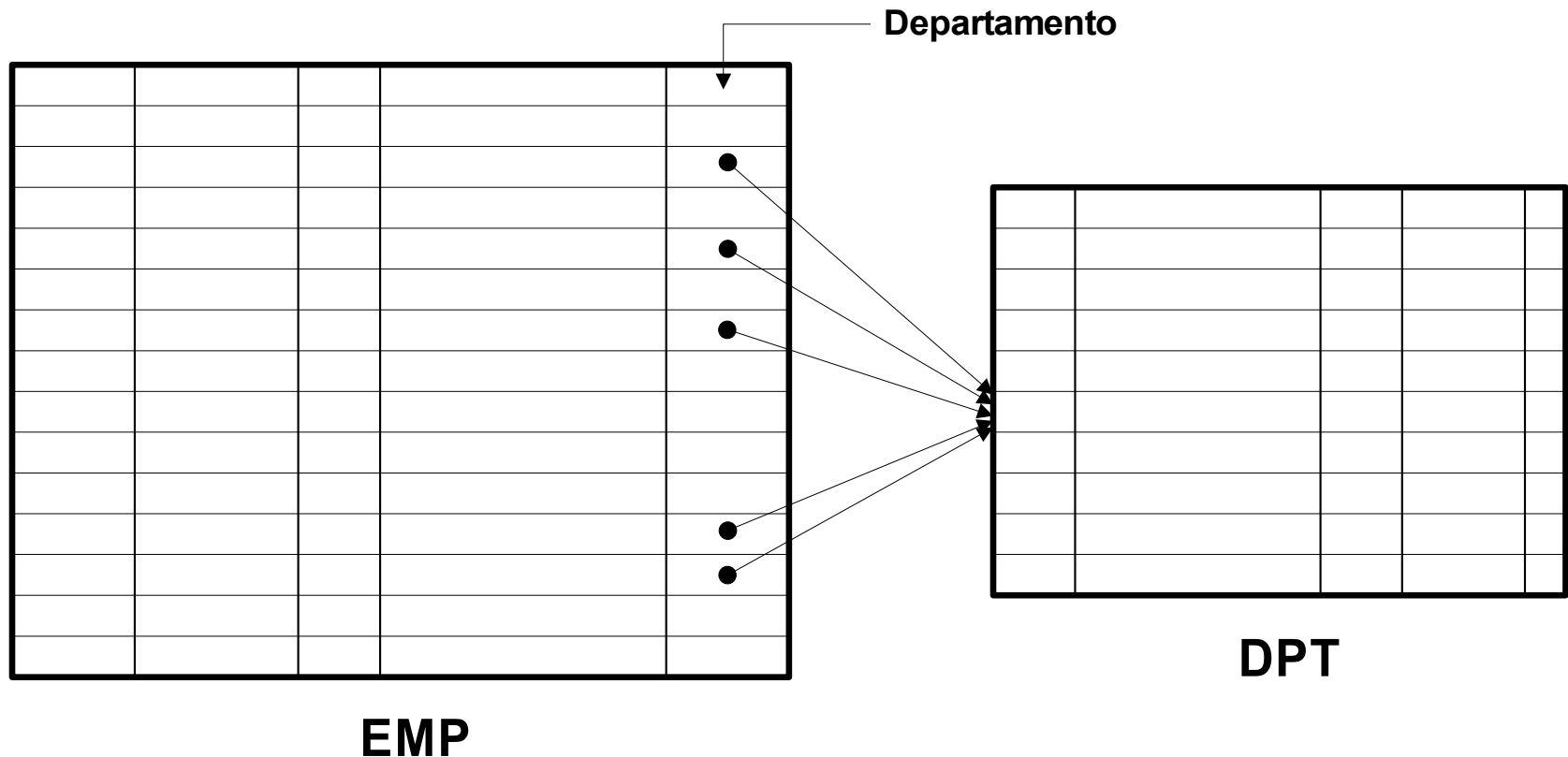
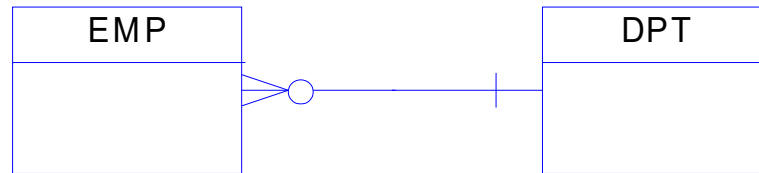
### **Solución**

- Se establecen punteros entre las filas correspondientes del lado de la matriz n

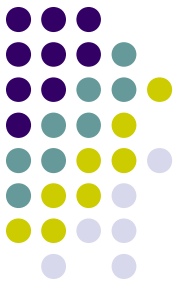
# RDBMS *Relational DataBase Management System*



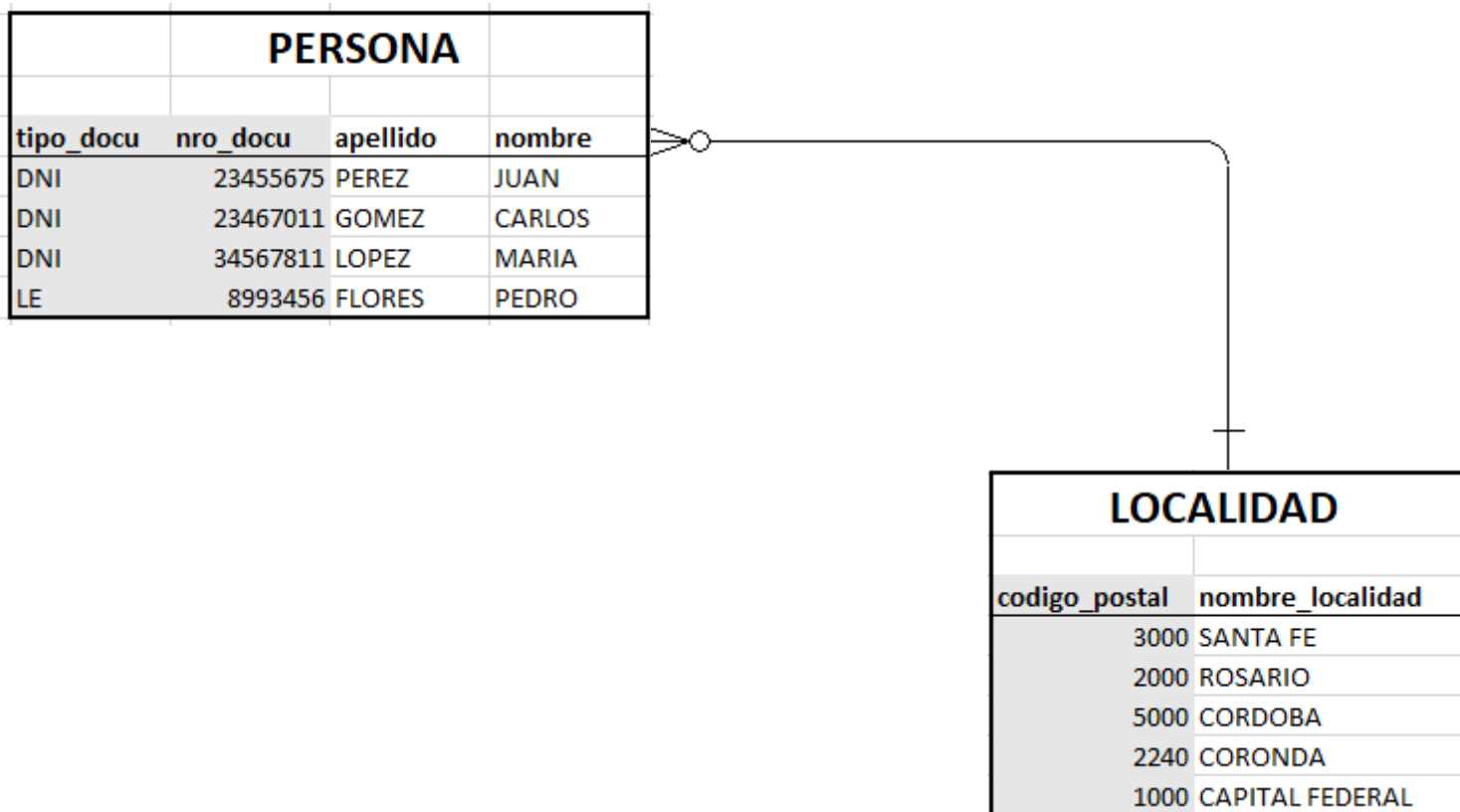
## Relación n-1



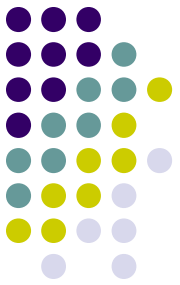
# RDBMS *Relational DataBase Management System*



## Relación n-1



# RDBMS *Relational DataBase Management System*



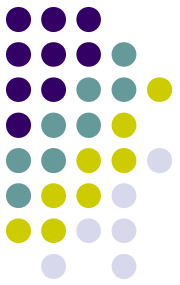
## Relación n-1

PERSONA				
tipo_docu	nro_docu	apellido	nombre	codigo_postal
DNI	23455675	PEREZ	JUAN	3000
DNI	23467011	GOMEZ	CARLOS	2240
DNI	34567811	LOPEZ	MARIA	3000
LE	8993456	FLORES	PEDRO	5000

Referencia del n,1

LOCALIDAD	
codigo_postal	nombre_localidad
3000	SANTA FE
2000	ROSARIO
5000	CORDOBA
2240	CORONDA
1000	CAPITAL FEDERAL

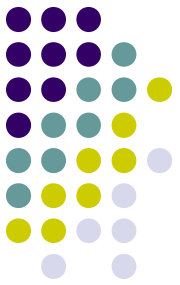
# RDBMS *Relational DataBase Management System*



## Relación 1-n

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Una fila de una matriz puede tener correspondencia con muchas filas de la otra matriz.

# RDBMS *Relational DataBase Management System*



## Relación 1-n

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Una fila de una matriz puede tener correspondencia con muchas filas de la otra matriz.

### **Solución**

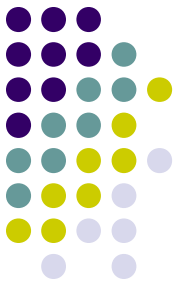
- Se crea la **MATRIZ RELACIONAL**

## A decorative graphic in the bottom right corner consisting of a grid of colored dots in shades of purple, teal, yellow, and light blue, arranged in a pattern that tapers to the right.

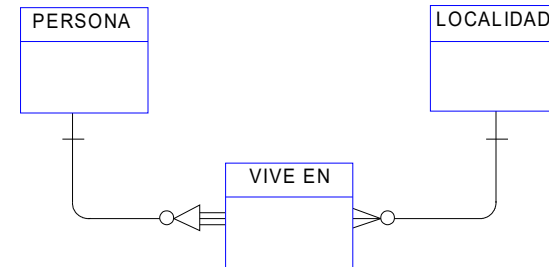




# RDBMS *Relational DataBase Management System*



## Relación 1-n



PERSONA			
tipo_docu	nro_docu	apellido	nombre
DNI	23455675	PEREZ	JUAN
DNI	23467011	GOMEZ	CARLOS
DNI	34567811	LOPEZ	MARIA
LE	8993456	FLORES	PEDRO

LOCALIDAD	
codigo_postal	nombre_localidad
3000	SANTA FE
2000	ROSARIO
5000	CORDOBA
2240	CORONDA
1000	CAPITAL FEDERAL



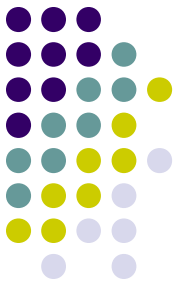
Referencia (dependiente)



Referencia 1,1

VIVE EN		
tipo_docu	nro_docu	codigo_postal
DNI	23455675	3000
DNI	23467011	2240
DNI	34567811	3000
LE	8993456	5000

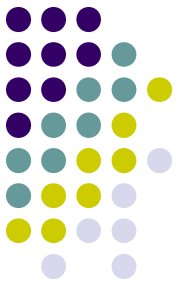
# RDBMS *Relational DataBase Management System*



## Relación m-n

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Una fila de una matriz puede tener correspondencia con muchas filas de la otra matriz.

# RDBMS *Relational DataBase Management System*



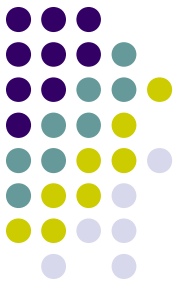
## Relación m-n

- Se consideran dos poblaciones. Cada una de ellas, está representada por su propia matriz.
- Una fila de una matriz puede tener correspondencia con muchas filas de la otra matriz.

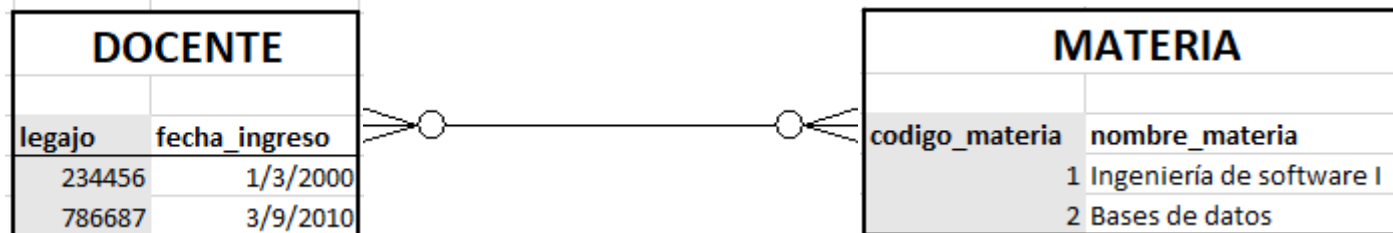
### **Solución**

- Se crea la **MATRIZ DE RELACION**.

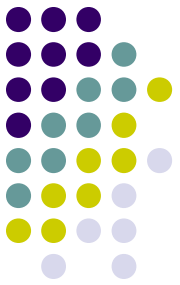
# RDBMS *Relational DataBase Management System*



## Relación m-n



# RDBMS *Relational DataBase Management System*

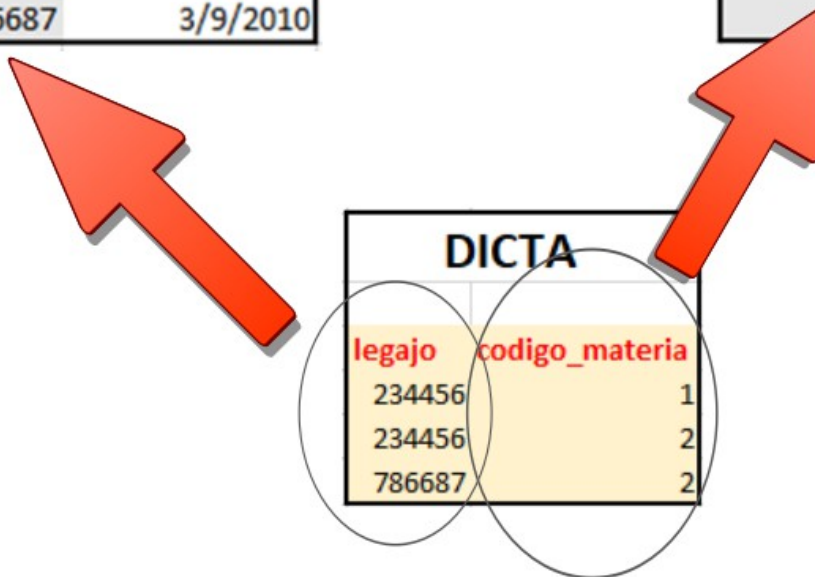


## Relación m-n

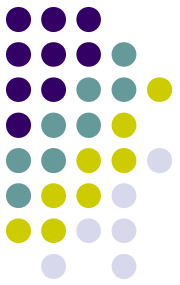
DOCENTE	
legajo	fecha_ingreso
234456	1/3/2000
786687	3/9/2010

MATERIA	
codigo_materia	nombre_materia
1	Ingeniería de software I
2	Bases de datos

DICTA	
legajo	codigo_materia
234456	1
234456	2
786687	2



# RDBMS *Relational DataBase Management System*

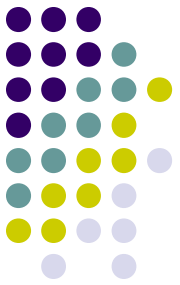


## Relación m-n (con atributos)

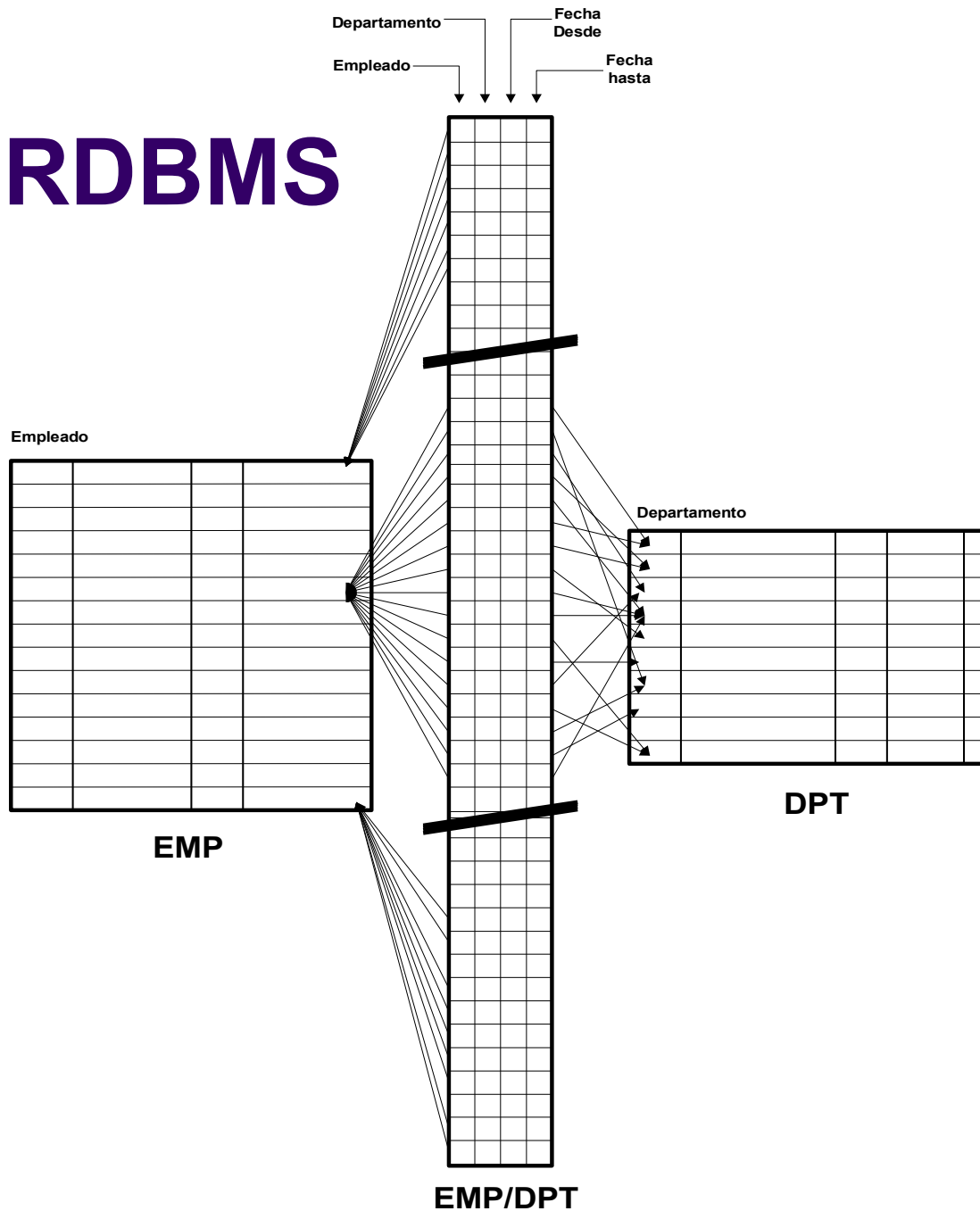
FACTURA	
nro_factura	fecha_factura
1000000	3/4/2020
1000002	5/4/2020

ARTICULO		
codigo_articulo	nombre_articulo	precio
1	CUADERNO	50
2	LAPIZ	9
3	GOMA	6
4	REGLA	25
5	CARPETA	45

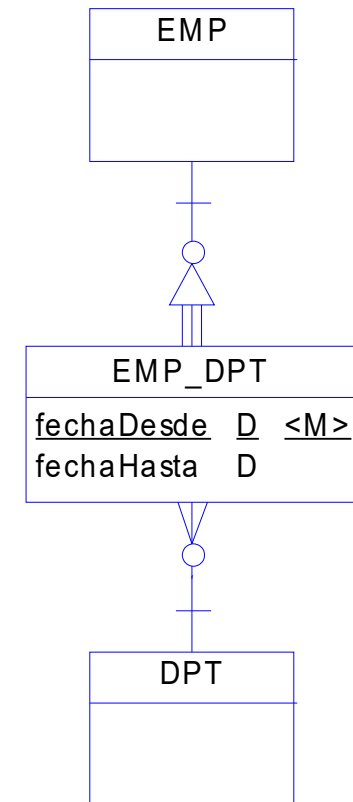
FORMADA POR			
nro_factura	codigo_articulo	cantidad	
1000000	1	1	1
1000000	3	3	2
1000000	5	5	2
1000000	4	4	1
1000002	5	5	1
1000002	2	2	4
1000002	3	3	5
1000002	1	1	3

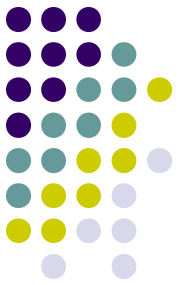


# RDBMS



## Relación m-n



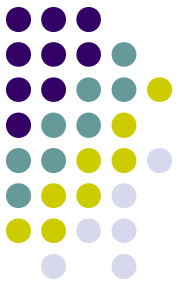


## La base de datos “alumnado”

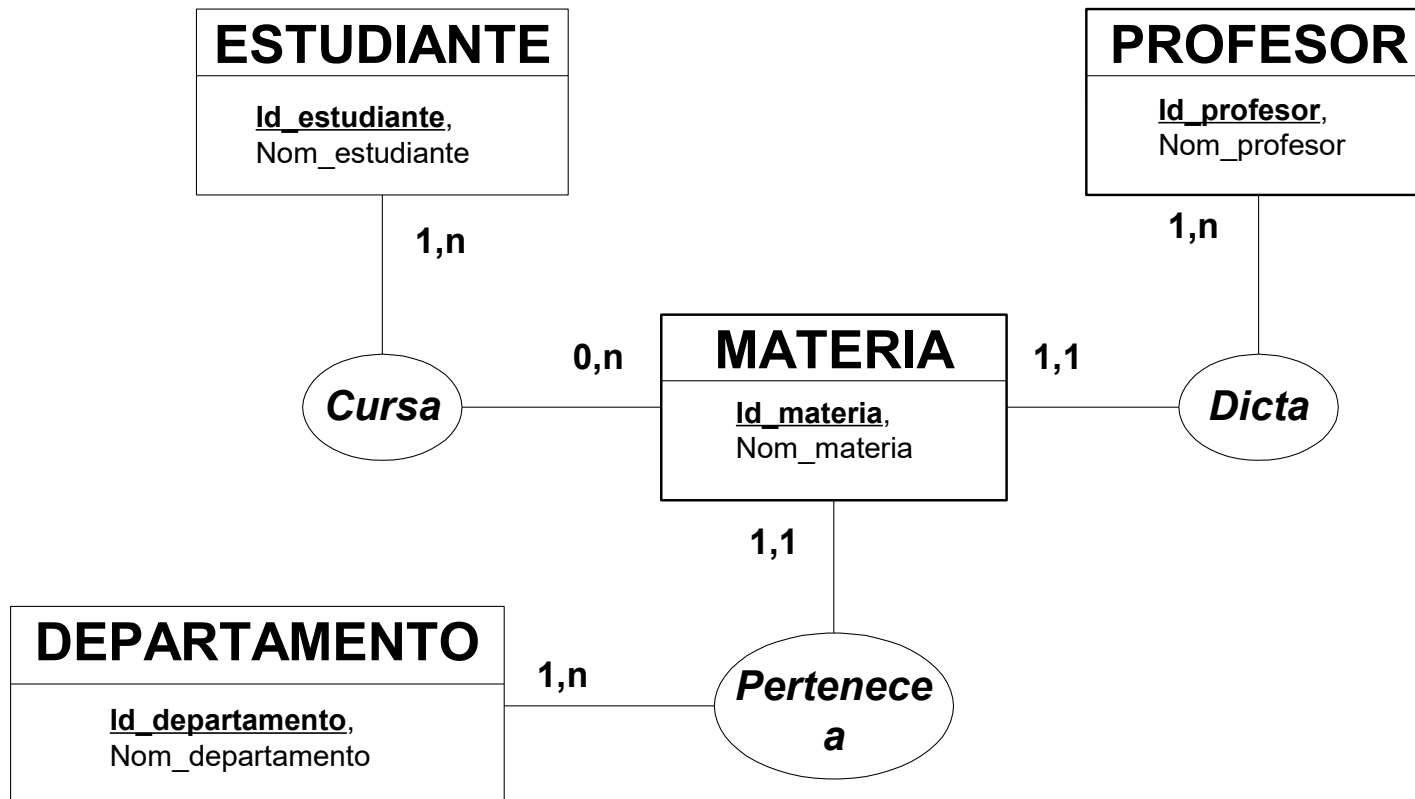
- Un **PROFESOR** dicta al menos una **MATERIA**
- Una **MATERIA** es dictada por uno y solamente un **PROFESOR**
- Un **ALUMNO** puede estar inscripto y cursar varias **MATERIAS**
- Una **MATERIA** depende de uno y solamente un **DEPARTAMENTO**



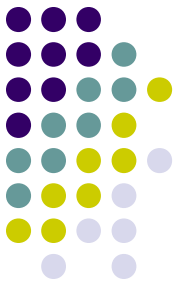
# RDBMS *Relational DataBase Management System*



## La base de datos “alumnado”



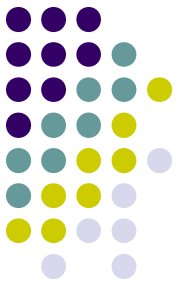
# RDBMS *Relational DataBase Management System*



## La base de datos “alumnado”

ID Materia	Nombre Materia	ID Departa- mento	Nombre Departa- mento	ID Profesor	Nombre Profesor	ID Estudiante	Nombre Estudiante

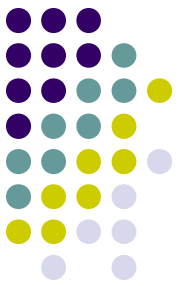
# RDBMS *Relational DataBase Management System*



## El registro compuesto

<b>Id Materia</b>	<b>Nombre Materia</b>	<b>Id Depto.</b>	<b>Nombre Depto.</b>	<b>Id Profesor</b>	<b>Nombre Profesor</b>	<b>Id Estudiante</b>	<b>Nombre Estudiante</b>
10725	Ing. de Software	100	Sistemas	1	CERI, José	666	SCRUTTI, Claudio
						127	MORETTI, A. R.
						.....	.....
						25466	ROSSI, Fabiana C.
10726	Bases de datos	100	Sistemas	2	CERI, José	1433	SANTO, Gabriel M.
						25466	ROSSI, Fabiana c.
						.....	.....
						3155	DEYRO, B.M.W.

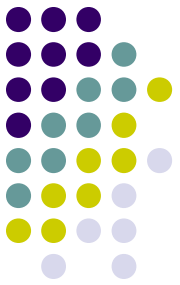
# RDBMS *Relational DataBase Management System*



## El registro compuesto

<b>Id Materia Estudiante</b>	<b>Id Materia</b>	<b>Nombre Materia</b>	<b>Id Depto.</b>	<b>Nombre Depto.</b>	<b>Id Profesor</b>	<b>Nombre Profesor</b>	<b>Id Estudiante</b>	<b>Nombre Estudiante</b>
10725-1	10725	Ing. de Software	100	Sistemas	1	CERI, José	666	SCRUTTI, Claudio
10725-2	10725	Ing. de Software	100	Sistemas	1	CERI, José	127	MORETTI, A. R.
.....	10725	Ing. de Software	100	Sistemas	1	CERI, José	.....	.....
10725-n	10725	Ing. de Software	100	Sistemas	1	CERI, José	25466	ROSSI, Fabiana C.
10726-1	10726	Bases de datos	100	Sistemas	2	CERI, José	1433	SANTO, Gabriel M.
10726-2	10726	Bases de datos	100	Sistemas	2	CERI, José	25466	ROSSI, Fabiana c.
.....	10726	Bases de datos	100	Sistemas	2	CERI, José	.....	.....
10726-n	10726	Bases de datos	100	Sistemas	2	CERI, José	3155	DEYRO, B.M.W.

# RDBMS *Relational DataBase Management System*



**continúa...**

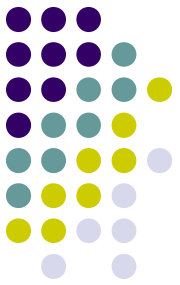
***No normalizada***




***1FN***


[illegible]

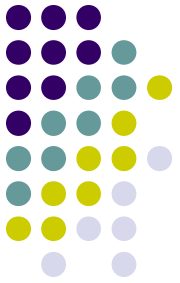
# RDBMS *Relational DataBase Management System*



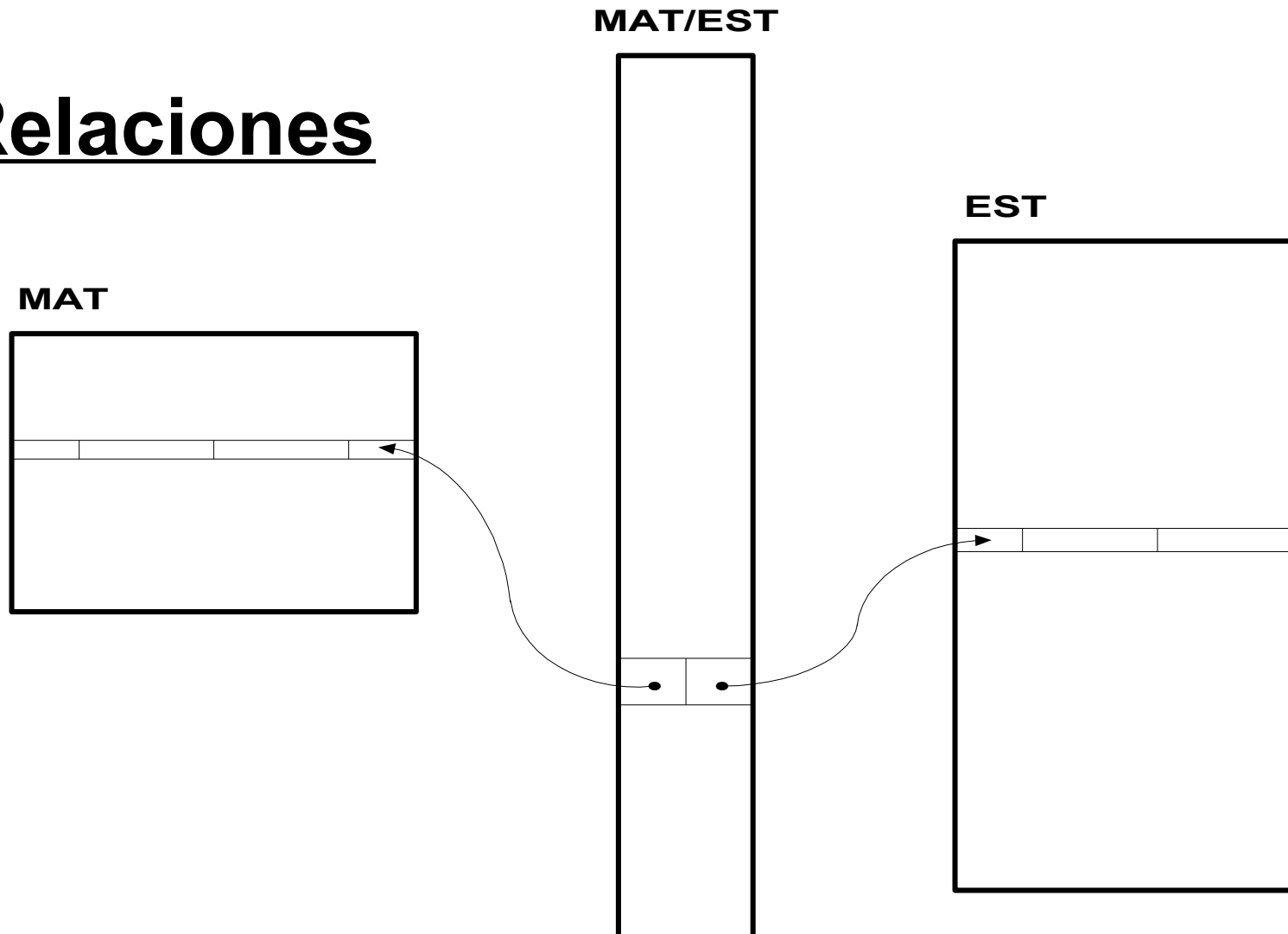
## Descomposición

- Cada nueva tabla está en la primera forma normal.
- Cada tupla tiene una clave.
- La eficiencia ganada en espacio da como resultado menores requerimientos de éste, y menor número de columnas para cada sub-relación.
- Cada relación tiene por lo general menos filas, dado que el número de entidades para cada sub-relación es a menudo menor que el de la relación original.

# RDBMS *Relational DataBase Management System*

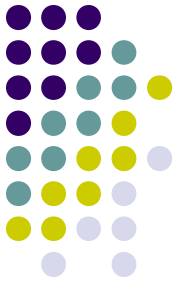


## Relaciones



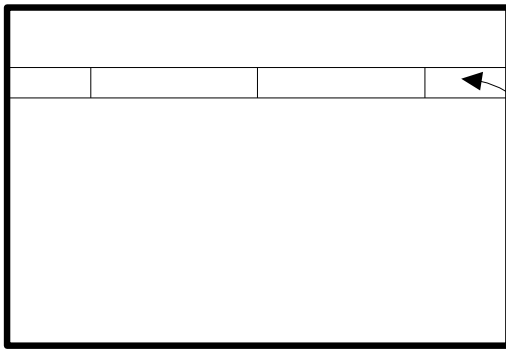


# RDBMS *Relational DataBase Management System*

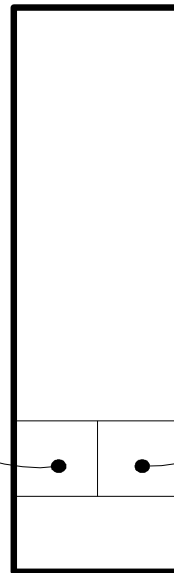


## Relaciones

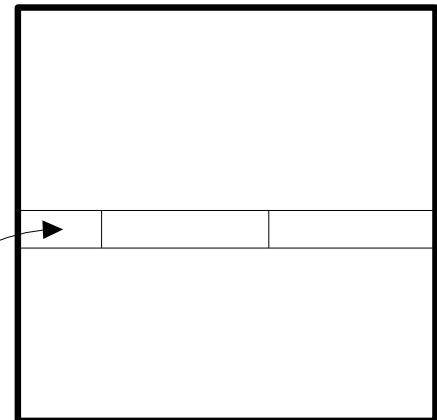
**MAT**



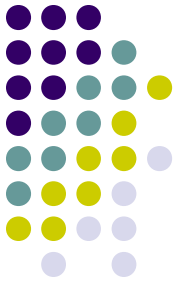
**MAT/PROF**



**PROF**

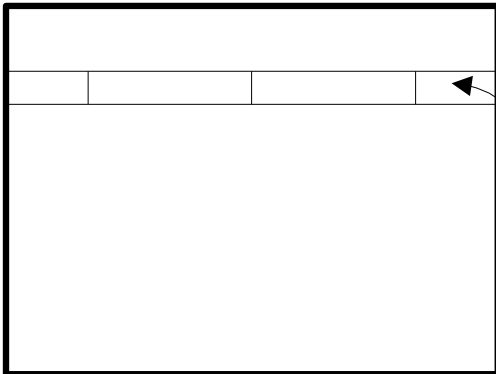


# RDBMS *Relational DataBase Management System*

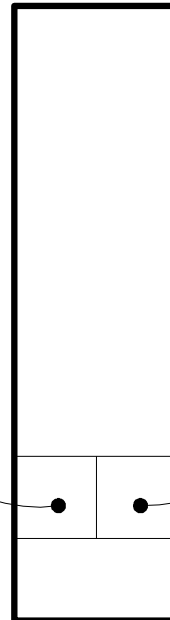


## Relaciones

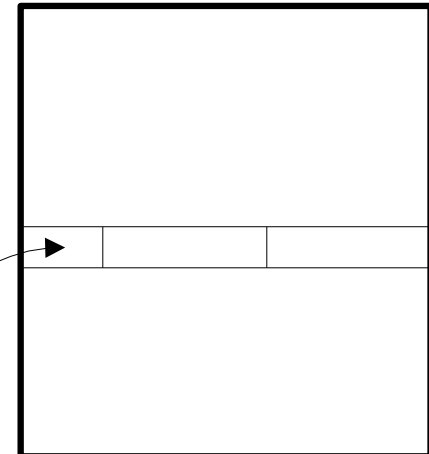
DEP



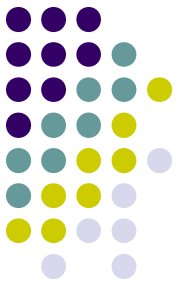
DEP/MAT



MAT



# RDBMS *Relational DataBase Management System*



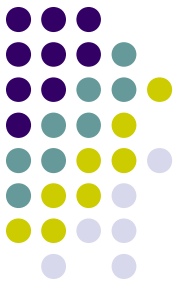
## Tablas de entidades

- ESTUDIANTE (*Id\_estudiante*, Nom\_estudiante)
- PROFESOR (*Id\_profesor*, Nom\_profesor)
- MATERIA (*Id\_materia*, Nom\_materia)
- DEPARTAMENTO (*Id\_departamento*, Nom\_departamento)

## Tablas relacionales

- EST\_MAT (relación entre estudiante y materia)
- PROF\_MAT (relación entre profesor y materia)
- DEP\_MAT (relación entre departamento y materia)

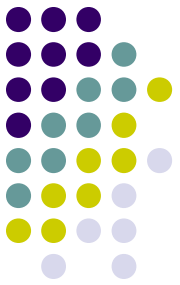
# RDBMS *Relational DataBase Management System*



## Depuración de las tablas relacionales

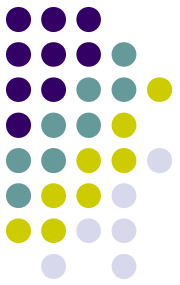
- **Por cada uno de los símbolos utilizados en el modelo conceptual de datos, existirá una tabla en el modelo físico (una por cada entidad y una por cada relación).??**

# RDBMS *Relational DataBase Management System*



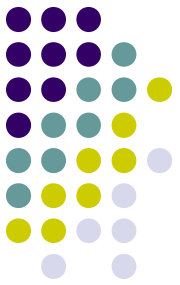
## Depuración de las tablas relacionales

- **Por cada uno de los símbolos utilizados en el modelo conceptual de datos, existirá una tabla en el modelo físico (una por cada entidad y una por cada relación).??**
- **NO. Depende de la funcionalidad de la relación y de la parcialidad de participación de las entidades en la relación en cuestión.**



## Depuración de las tablas relacionales

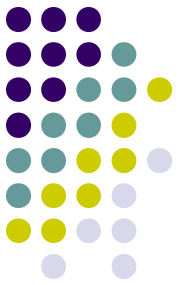
- EST\_MAT (m,n): un estudiante puede cursar muchas materias y una materia puede ser cursada por muchos estudiantes.
- PROF\_MAT (1,n): una materia es dictada por uno y solamente un profesor, mientras que un profesor puede dictar varias materias.
- DEP\_MAT (1,n): una materia pertenece a uno y solamente un departamento, mientras que un departamento tiene varias materias.



## Depuración de las tablas relacionales

- **EST\_MAT (m,n)**: un estudiante puede cursar muchas materias y una materia puede ser cursada por muchos estudiantes.
- **PROF\_MAT (1,n)**: una materia es dictada por uno y solamente un profesor, y un profesor puede dictar varias materias.
- **DEP\_MAT (1,n)**: un material pertenece solamente un departamento y un departamento tiene varias materias.

**DEPENDENCIA  
FUNCIONAL**

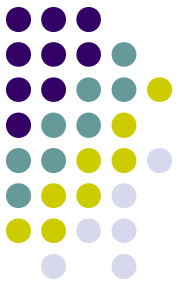


## Depuración de las tablas relacionales

- EST\_MAT (m,n): un estudiante puede cursar muchas materias y una materia puede ser cursada por muchos estudiantes. **TABLA DE RELACIÓN**
- PROF\_MAT (1,n): una materia es dictada por uno y solamente un profesor, pero un profesor puede dictar varias materias. **DEPENDENCIA FUNCIONAL**
- DEP\_MAT (1,n): un departamento tiene solamente un departamento y un departamento tiene varias materias. **DEPENDENCIA FUNCIONAL**



# RDBMS *Relational DataBase Management System*



## Depuración de las tablas relacionales

**MATERIA** quedará conformada por:

Atributos

*Id\_materia*

*Nom\_materia*

*Id\_profesor*

*Id\_departamento*

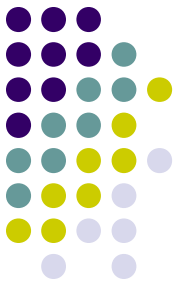
Integridad

*Clave primaria: Id\_materia*

*Clave ajena Id\_profesor REFERENCIANDO a PROFESOR*

*Clave ajena Id\_departamento REFERENCIANDO a  
DEPARTAMENTO*

# RDBMS *Relational DataBase Management System*



## Depuración de las tablas relacionales

**MATERIA** quedará conformada por:

Atributos

*Id\_materia*

*Nom\_materia*

*Id\_profesor*

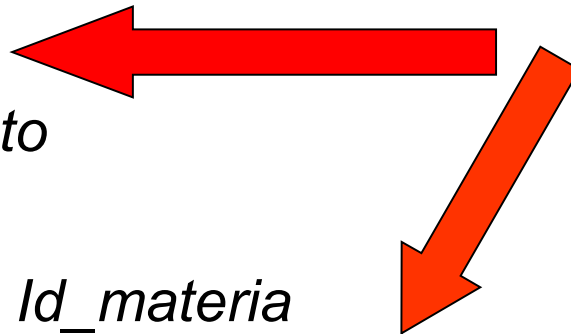
*Id\_departamento*

Integridad

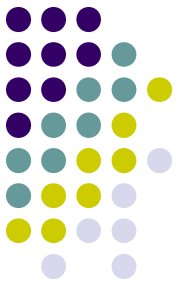
*Clave primaria: Id\_materia*

*Clave ajena Id\_profesor REFERENCIANDO a PROFESOR*

*Clave ajena Id\_departamento REFERENCIANDO a DEPARTAMENTO*



# RDBMS *Relational DataBase Management System*



## Depuración de las tablas relacionales

**MATERIA** quedará conformada por:

Atributos

*Id\_materia*

*Nom\_materia*

*Id\_profesor*

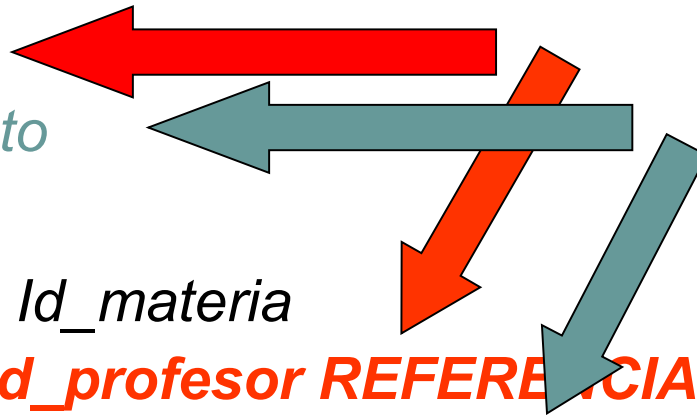
*Id\_departamento*

Integridad

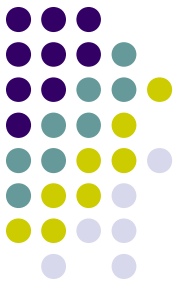
Clave primaria: *Id\_materia*

*Clave ajena Id\_profesor REFERENCIANDO a PROFESOR*

*Clave ajena Id\_departamento REFERENCIANDO a DEPARTAMENTO*



# RDBMS *Relational DataBase Management System*



## Depuración de las tablas relacionales

**EST\_MAT** estará compuesta por:

Atributos

*Id\_estudiante*

*Id\_materia*

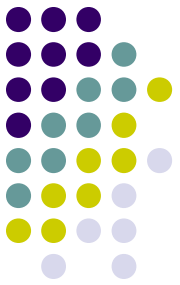
Integridad

*Clave primaria: Id\_estudiante + Id\_materia*

*Clave ajena Id\_estudiante REFERENCIANDO a ESTUDIANTE*

*Clave ajena Id\_materia REFERENCIANDO a MATERIA*

# RDBMS *Relational DataBase Management System*

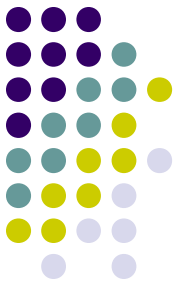


## Transformación del modelo E-R al MFD

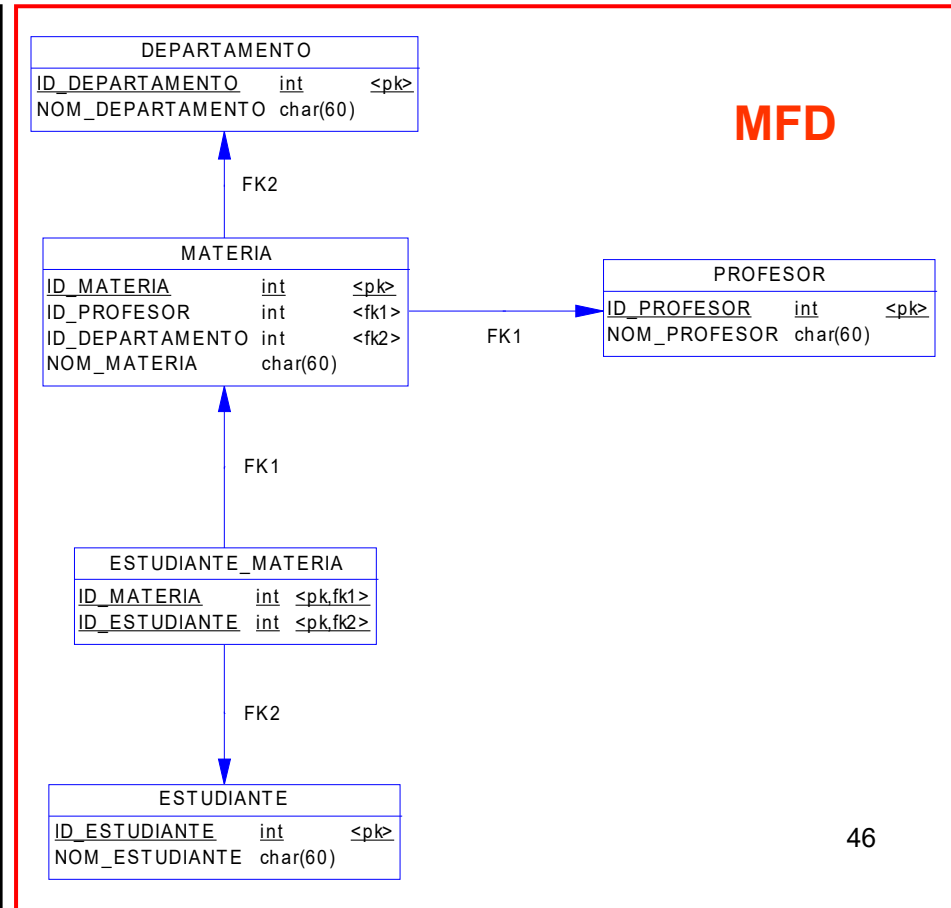
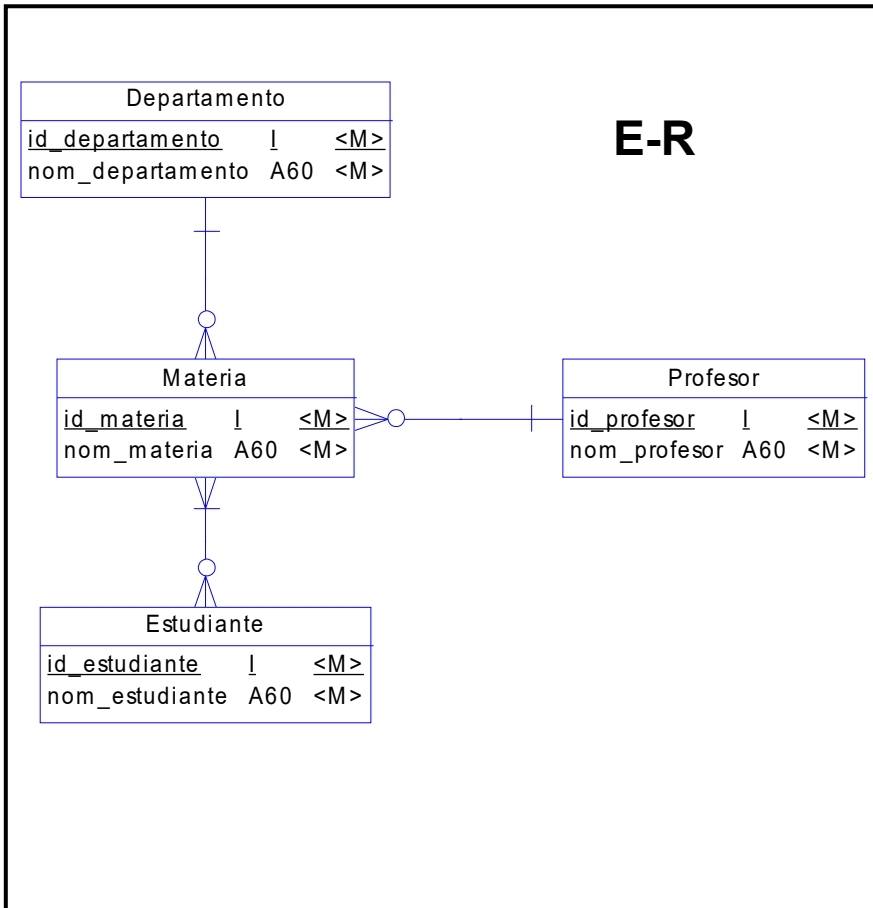
Los elementos que se transformarán en tablas serán:

- Las entidades
- Las relaciones que sean de funcionalidad m,n (muchos a muchos)
- Las relaciones del tipo 0,1 – 0,1
- Las relaciones que poseen atributos (que necesariamente serán m,n o bien 0,1 – 0,1)
- Las relaciones en las que participen más de 2 entidades
- Para el caso de las jerarquías de clasificación pueden:
  - Generar solamente la tabla correspondiente a la entidad padre
  - Generar la tabla de la entidad padre y tablas para cada uno de sus hijos
  - Generar solamente las tablas de las entidades hijo

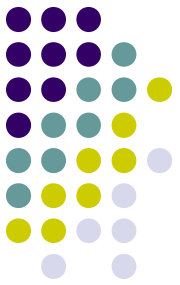
# RDBMS *Relational DataBase Management System*



## Transformación del modelo E-R al MFD



# RDBMS *Relational DataBase Management System*

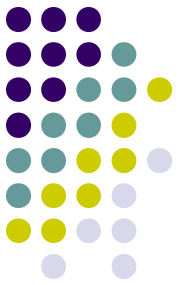


## Diagrama de clases → Modelo de datos

Los elementos que se transformarán en tablas son:

- Las clases
- Las asociaciones que sean de funcionalidad m,n (muchos a muchos)
- Las asociaciones con atributos (clases asociativas)
- Para el caso de las generalizaciones (relaciones de herencia) se puede:
  - Generar sólo la tabla correspondiente a la entidad padre (generalmente una clase abstracta)
  - Generar la tabla de la entidad padre y las tablas para cada una de las clases hijas
  - Generar solo las tablas de las clases hijas

# RDBMS *Relational DataBase Management System*

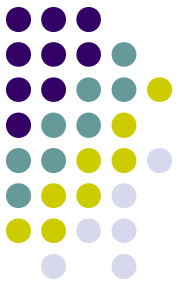


## Diagrama de clases → Modelo de datos

- Las relaciones de composición implican una relación de dependencia: la clave de la tabla dependiente estará conformada por la clave de la tabla principal más sus atributos identificatorios propios
- Las relaciones de agregación no modifican el modo en que se pasan las relaciones al modelo físico de datos. Dependen exclusivamente de la cardinalidad de las relaciones



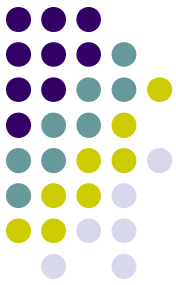
# RDBMS *Relational DataBase Management System*



## Claves de negocio vs. id

- En el mundo de objetos, las instancias de cada clases se identifican a través de “id”. La persistencia de esas instancias en filas de tablas de bases de datos relacionales, se puede realizar utilizando identificadores autogenerados (id).
- Para mantener la integridad se debe definir las claves de negocio como claves alternativas únicas.

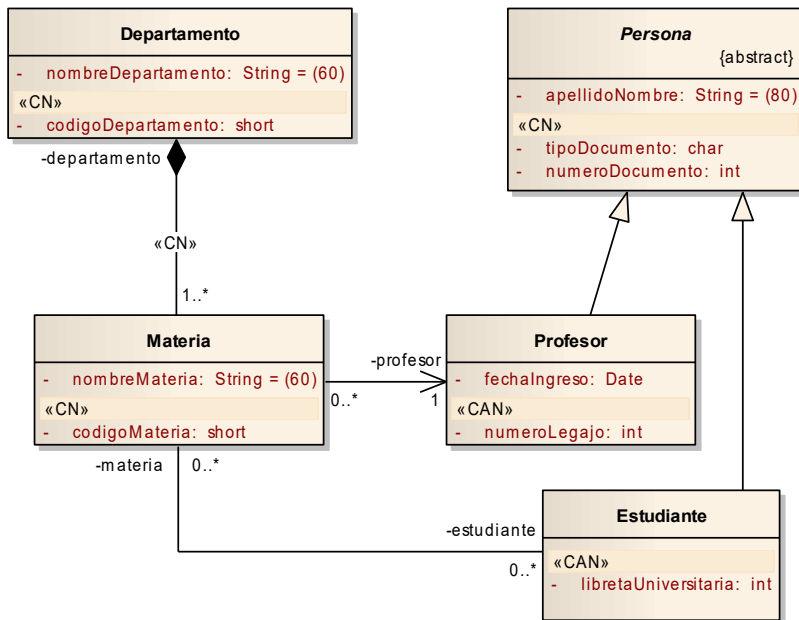
# RDBMS *Relational DataBase Management System*



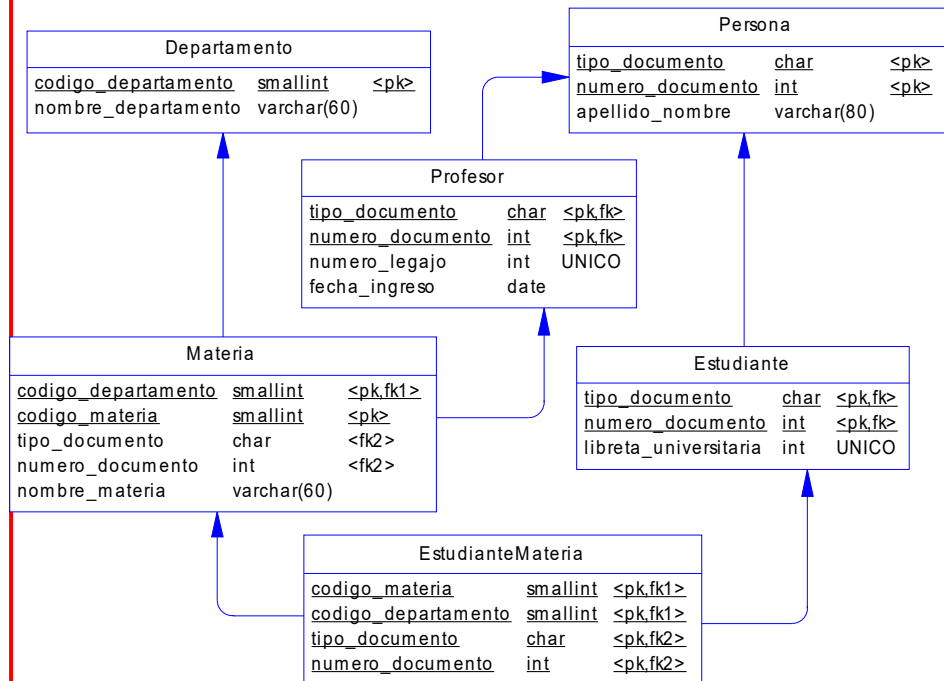
## Transformación de diagrama de clases al MFD

### Diagrama de clases

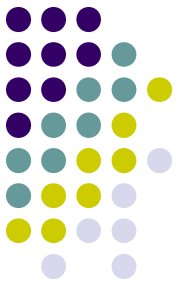
class Clases - MFD



### MFD con Claves de Negocio



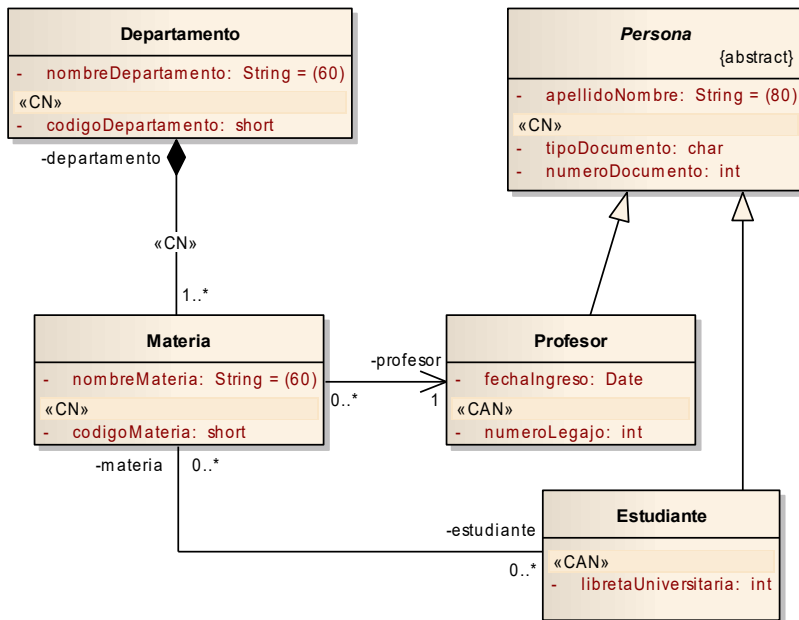
# RDBMS *Relational DataBase Management System*



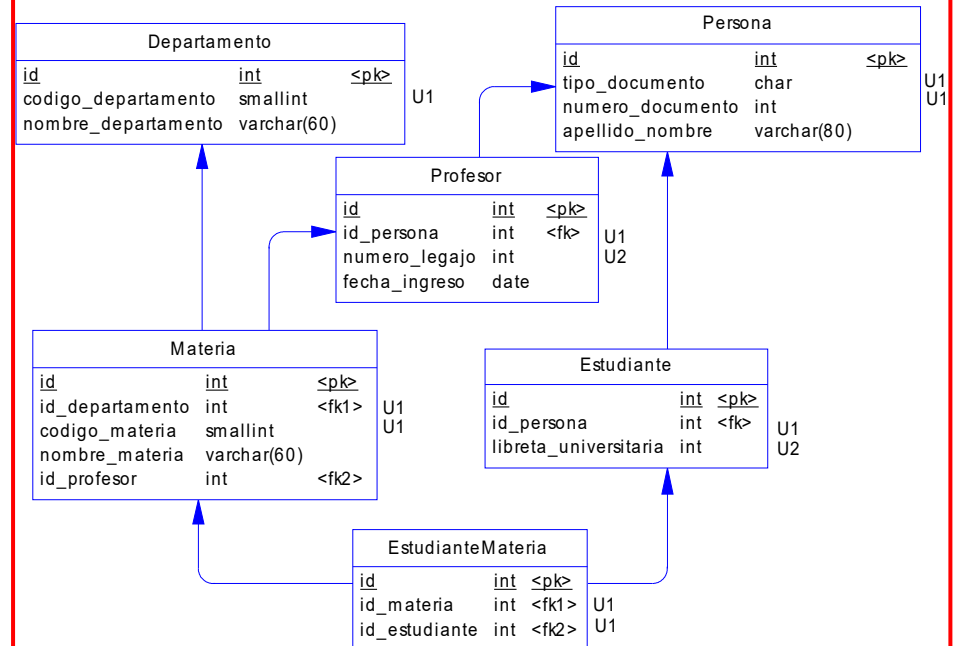
## Transformación de diagrama de clases al MFD

### Diagrama de clases

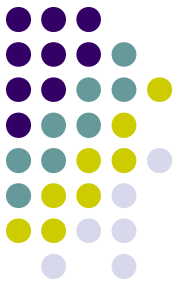
class Clases - MFD



### MFD con id



# RDBMS *Relational DataBase Management System*



## Principios de la RDBMS

### **Regla 0: Gestión de una base de datos relacional.**

Todo sistema que se anuncie como un sistema de gestión de base de datos relacional, debe ser capaz de manejar bases de datos exclusivamente con sus capacidades relacionales.

### **Regla 1: Representación de la información.**

Toda la información de una base de datos relacional, se representa explícitamente en el ámbito lógico y exactamente de una forma: mediante valores en tablas.

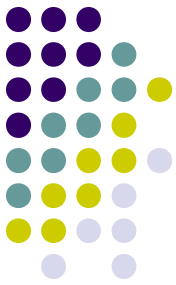
### **Regla 2: Garantía de accesibilidad lógica.**

Todos y cada uno de los datos de una base de datos, relacional tienen la garantía de ser accesibles lógicamente mediante el recurso de una combinación de: el nombre de la Tabla, el valor de la clave primaria y el nombre de la columna.

### **Regla 3: Representación sistemática de la información que falta.**

Los valores nulos (que son distintos de la cadena vacía de caracteres o de la cadena de caracteres en blanco, y distintos de cero o de cualquier otro número) tienen la existencia en los sistemas de gestión de bases de datos totalmente relacionales, para representar la información que falta y la información que no es aplicable, de forma sistemática e independiente del tipo de dato.

# RDBMS *Relational DataBase Management System*



## Principios de la RDBMS

### **Regla 4: Sub-lenguaje de datos completo.**

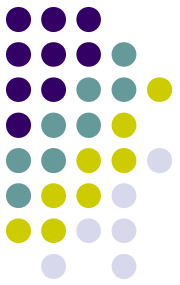
Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal. Sin embargo, debe haber, al menos, un lenguaje cuyas instrucciones puedan expresarse por alguna sintaxis bien definida, como cadenas de caracteres, y que sea completo, soportando todos los términos siguientes:

- Definición de Datos
- Definición de Vistas
- Manejo de Datos
- Limitaciones de integridad
- Autorización o permisos
- Límites de transacción (inicio y fin para hacer permanentes los cambios y deshacer los cambios no permanentes)

### **Regla 5: Inserción, actualización y borrado de alto nivel.**

La capacidad de manejar una relación de base o una relación derivada como un único operador, se aplica no sólo a la recuperación de datos, sino también a la inserción, a la actualización y al borrado de datos.

# RDBMS *Relational DataBase Management System*



## Principios de la RDBMS

### **Regla 6: Independencia de los datos físicos.**

Los programas de aplicaciones y las actividades terminales, permanecerán lógicamente inalterados siempre que se realicen cambios en las representaciones de almacenamiento o en los métodos de acceso.

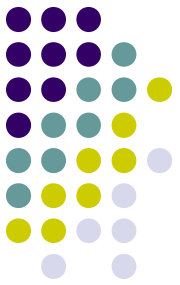
### **Regla 7: Independencia de los datos lógicos.**

Los programas de aplicaciones y las actividades finales permanecerán lógicamente inalterados cuando se llevan a cabo cambios en las tablas de base que conservan la información de cualquier tipo que permita teóricamente su inalterabilidad.

### **Regla 8: Independencia de la integridad.**

Las limitaciones de integridad, específicas de una base de datos en particular, deben ser definibles en un sub-lenguaje de definición de datos y almacenables en el catálogo o diccionario.

# RDBMS *Relational DataBase Management System*



## Características de las TABLAS

- Son homogéneas en sus columnas. Todos los elementos de la columna son de la misma clase o tipo de dato.
- Cada elemento, es un único número o una cadena de caracteres – **1FN** - por lo tanto, un elemento determinado por fila y columna será un único valor y no un grupo.
- Todas las filas de una tabla, deben ser distintas (no se permiten duplicaciones) – **1FN** - .
- El orden de las filas dentro de una tabla, es indiferente.
- A las columnas de una tabla, se les asigna nombres distintos, y el orden de las columnas dentro de una tabla, es indiferente.
- Los tipos de tablas son las de **base**, las **temporales** y las **virtuales**