



Desarrollo de Aplicaciones Multiplataforma
Bases de Datos

UD3.- El modelo relacional

2023-24



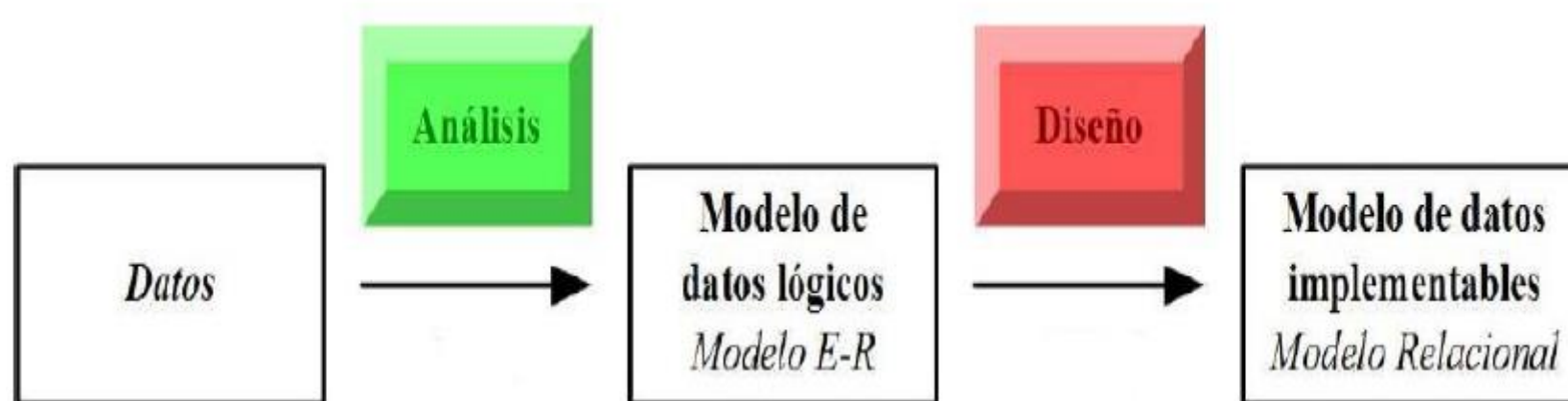
ÍNDICE

- Diseño lógico basado en el modelo relacional
- Estructura del modelo relacional
- Relaciones
 - Propiedades de las relaciones
 - Tipos de relaciones
 - Claves
- Esquema de una base de datos relacional y su representación
- Reglas de integridad
 - Nulos
 - Regla de integridad de la entidad
 - Regla de integridad referencial
 - Reglas de negocio
- Transformación del diagrama E-R al esquema relacional
- Notaciones para la representación de esquemas relacionales

**DISEÑO LÓGICO BASADO EN EL MODELO RELACIONAL**

El **diseño lógico** es la etapa del diseño de una BD en la que:

- Se obtiene la representación de la estructura de la base de datos en términos de almacenamiento (tablas).
- Implica la aplicación de unas reglas de transformación de los elementos del modelo conceptual de datos.





MODELO RELACIONAL

- Propuesto por **Codd** en los 70.
- Es el modelo más extendido.
- Está basado en la teoría de conjuntos, los datos se estructuran en forma de **relaciones** (tablas).
- Objetivo fundamental del modelo:
 - Mantener la **independencia de esta estructura lógica** respecto al modo de almacenamiento y a otras características físicas.
- El modelo relacional (MR) se caracteriza por:
 - Ser sencillo y uniforme (colección de tablas y lenguajes declarativos)
 - Tener una sólida fundamentación teórica: el modelo está definido con rigor matemático
 - Ser independiente del almacenamiento físico y de las aplicaciones.



MODELO RELACIONAL

- Los primeros sistemas relacionales tardaron diez años en aparecer en el mercado.
 - **System R** (1974)
 - Uno de los primeros SGBD relacionales, fue desarrollado por IBM como un proyecto de investigación y fue origen de **SQL**.
 - Fue origen de otros SGBD relacionales como DB2, SQL/DS e Ingress.
 - **Query By Example –QBE-** (1978) un lenguaje de consulta relacional que ofrece una interfaz de consulta gráfica.
 - **Oracle** (1979) primer SGBDR relacional comercial, el cual soporta como lenguaje de definición y manipulación de datos el SQL.
 - **Ingress** (1980) desarrollado por la universidad de Berkeley. Utilizaba un lenguaje de consulta basado en el cálculo relacional (QUEL). Fue muy utilizado en las universidades.



MODELO RELACIONAL

- SGBD relacionales
 - Oracle
 - MySQL
 - SQL Server

ORACLE®



PostgreSQL





MODELO RELACIONAL**Objetivos:****➤ Independencia física**

- La forma de almacenar los datos, no debe influir en su manipulación lógica. Si el almacenamiento físico cambia, los usuarios no tienen ni siquiera porque enterarse, seguirán funcionando sus aplicaciones.

➤ Independencia lógica

- Las aplicaciones que utilizan la base de datos no deben ser modificadas por que se modifiquen elementos de la base de datos. Es decir, añadir, borrar y suprimir datos, no influye en las vistas de los usuarios.

➤ Flexibilidad

- La base de datos ofrece fácilmente distintas vistas en función de los usuarios y aplicaciones.

➤ Uniformidad

- Las estructuras lógicas siempre tienen una única forma conceptual (las tablas)

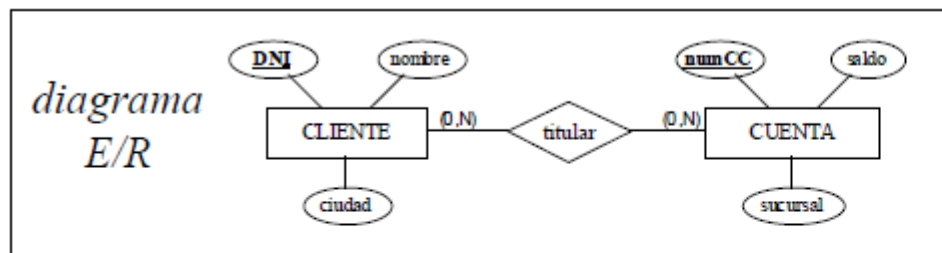
➤ Sencillez

- Facilidad de manejo (algo cuestionable).



MODELO RELACIONAL

Aproximación intuitiva al modelo relacional:



*Esquema relacional (simplificado)
de la Base de Datos bancaria*

Cliente

<u>DNI</u>	Nombre	Ciudad
17567984	Severino Martínez	Huesca
19465278	Salustiano Pérez	Teruel
25468724	Indalecio López	Zaragoza
18234587	Pedro García	Calatayud
17235465	Antonio Sánchez	Zaragoza

Cuenta

<u>numCC</u>	Sucursal	Saldo
123	23	2567
194	23	125874
237	14	654875
257	18	25984
100	22	0

Titular

<u>DNI</u>	<u>numCC</u>
17567984	123
18234587	123
25468724	257
25468724	194
17235465	237



ESTRUCTURA DEL MODELO RELACIONAL

➤ **Parte estructural**

- Utiliza una estructura de datos muy sencilla, la relación.

➤ **Parte manipulativa**

- Compuesta por un conjunto de operadores que permiten manejar la estructura anterior, el Álgebra Relacional

➤ **Teoría de las Dependencias Funcionales y de la Normalización**

- Compuesta por un conjunto de definiciones que permite estudiar las dependencias entre los atributos de las relaciones y proporciona métodos para un correcto diseño de las bases de datos relacionales.



RELACIONES

Elementos

▪ RELACIÓN

- Es la estructura básica del modelo relacional. Se representa mediante una **tabla**.
- La idea de relación según el modelo de Codd es diferente de una relación en el modelo Entidad/Relación de Chen.

▪ ATRIBUTO

- Representa las **propiedades** de la relación. Se representa mediante una **columna**.

▪ TUPLA

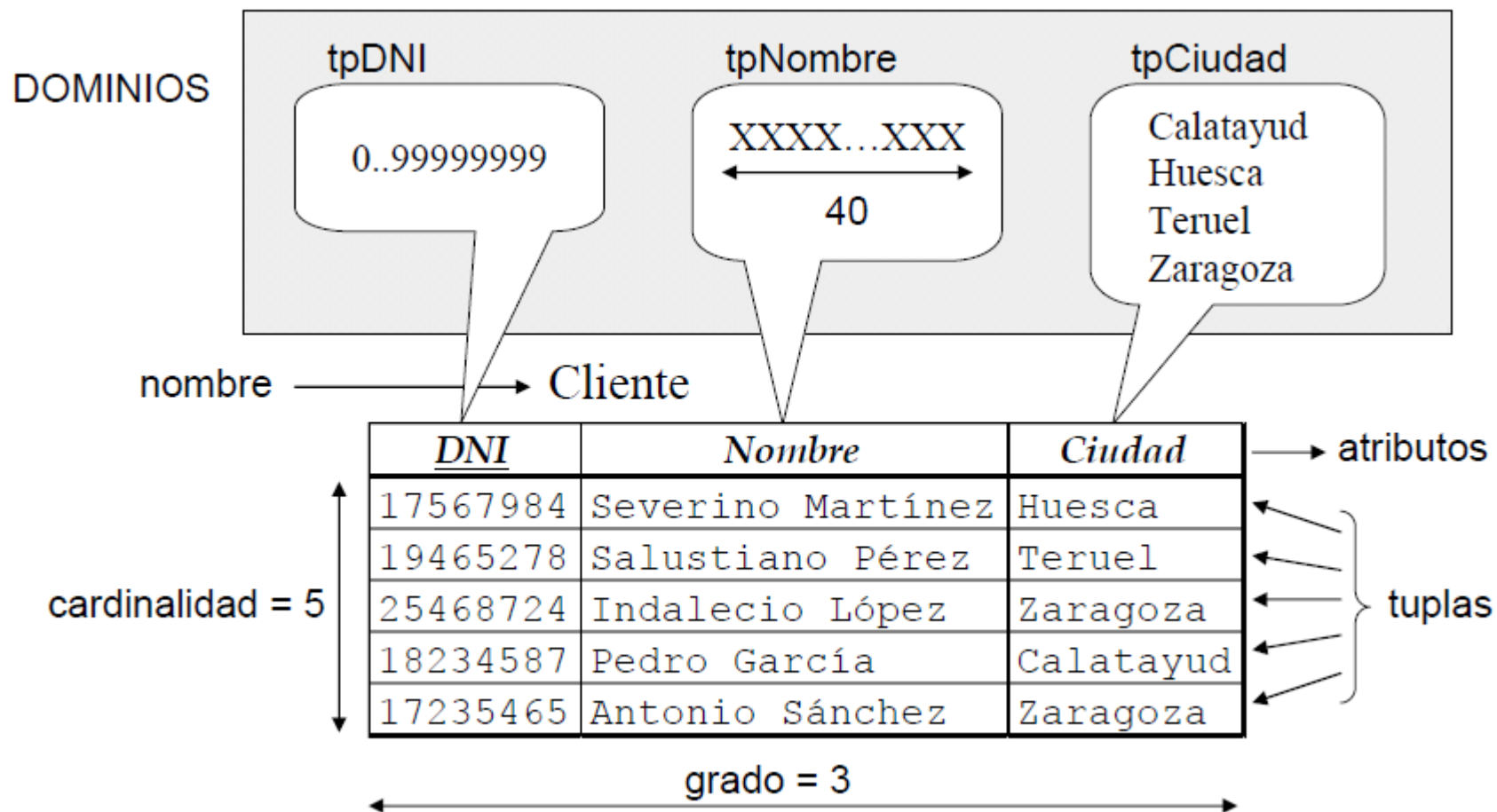
- Es una ocurrencia de la relación. Se representa mediante una **fila**.

▪ DOMINIO

- Es el conjunto válido de valores que toma un atributo. Existen con independencia de cualquier otro elemento.
- El número de columnas que tiene una relación recibe el nombre de **grado de la relación**, y el **número de filas** recibe el nombre de **cardinalidad** de la relación.

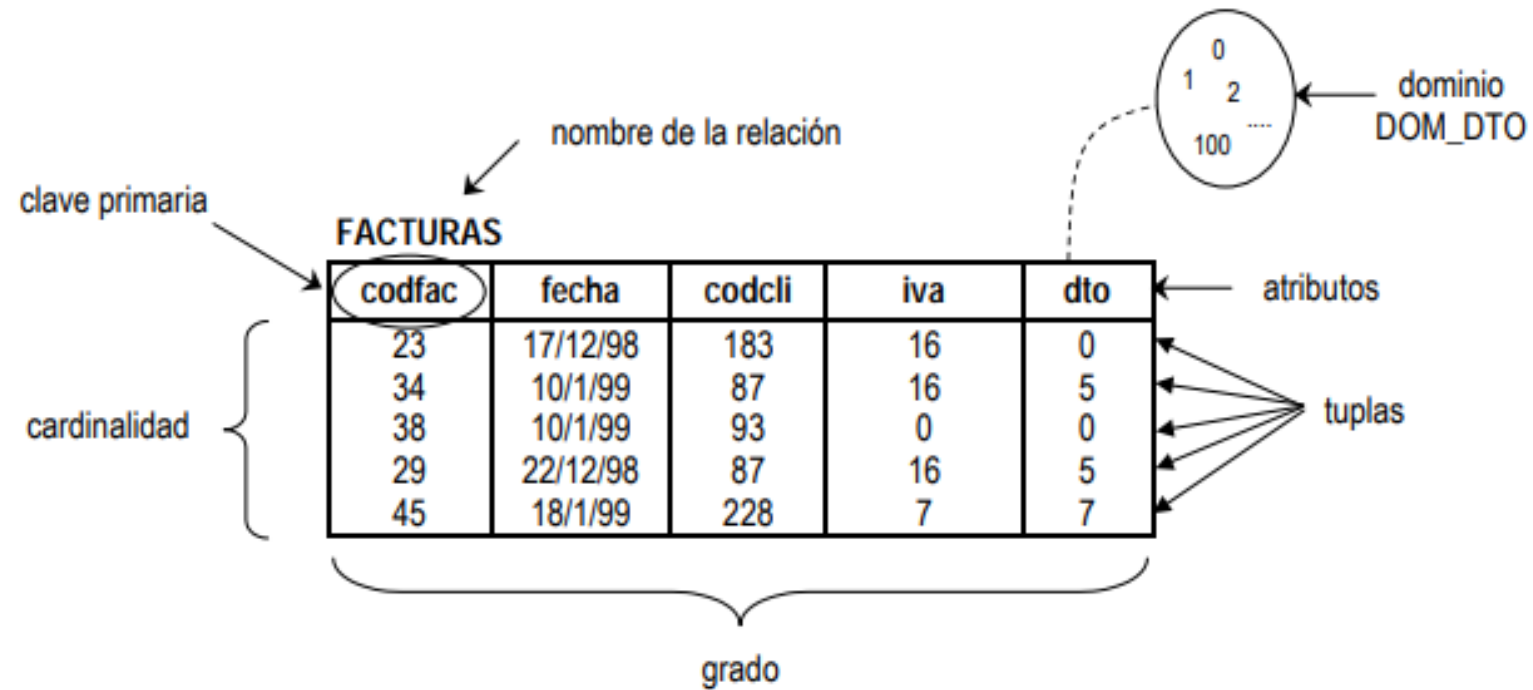


RELACIONES





RELACIONES



FACTURAS(codfac, fecha, codcli, iva, dto)



RELACIONES

Relación

Una relación está formada por dos partes:

- Cabecera (**esquema** o intensión).
 - Está formada por un conjunto **fijo de atributos**.
 - Está constituida por:
 - a. El nombre del conjunto (tabla).
 - b. El nombre de los atributos (columnas de la tabla).
 - c. Los dominios de los que toman valores.
- Cuerpo (extensión)
 - Está formado por un conjunto de **tuplas**.
 - **Grado de la relación:** número de atributos que posee.
 - **Cardinalidad:** número de tuplas de la relación.

Empleados				Esquema
<i>DNI</i>	<i>nombre</i>	<i>apellido</i>	<i>suelo</i>	
40.444.255	Juan	García	2.000	Extensión
33.567.711	Marta	Roca	2.500	
55.898.425	Carlos	Buendía	1.500	

esquema

EMPLEADOS (DNI, nombre, apellido, sueldo)



RELACIONES

Propiedades de las **relaciones (tablas)**:

- Todas las tuplas tienen las mismas columnas.
- Cada tupla debe ser única y no pueden existir tuplas duplicadas.
- El orden de las tuplas no importa.
- Cada atributo debe ser único y no pueden existir duplicados, además estará identificado por un nombre.
- Cada atributo sólo puede tomar **un valor** en cada tupla (los valores de los atributos son **atómicos**).
 - No se admiten atributos multivaluados.
- Los valores de un atributo deben pertenecer al dominio que representa.

PERSONA					
NOMBRE	APELLIDO ₁	APELLIDO ₂	DIRECCION	EDAD	TELEFONO_FIJO
Alfonso	Bonillo	Sierra	C/ Isla de Arosa, 7	38	948 363366 677586952
Sebastián	López	Ojeda	C/ Islas Vírgenes, 21	37	948 192021
Narciso	Jáimez	Toro	C/ Isla de Java, 11	38	948 111213
Gonzalo	Fernández	Hernández	C/ Ibiza, 10 C/ Calasparra, s/n	29	968 353535
Silvia	Thomas	Barrós	C/ Islas del Caribe, 19	34	958 190405

No es plana!!! Los atributos no tienen valores atómicos!!!



RELACIONES

Dominio

- Contiene todos los posibles valores que puede tomar un determinado atributo.
- Conjunto **finito** de valores **homogéneos** y **atómicos**, con un nombre asociado.
- Un **dominio**
 - Se identifica por un nombre.
 - Tiene un número finito de valores.
 - Todos los valores son del mismo tipo.
- Cada dominio puede definirse de dos maneras:
 - Extensión (dando sus posibles valores):
 - días de la semana = {lunes, martes, miércoles, ... sábado, domingo}
 - Intensión (mediante un tipo de datos):
 - peso = decimal



RELACIONES

Ejemplo: dominios de la relación OFICINA

OFICINA			
ATRIBUTO	NOMBRE DEL ATRIBUTO	DESCRIPCIÓN	DEFINICIÓN
Numero	NUM_OFICINA	Posibles valores de número de oficina	3 caracteres; rango 01- 099
Calle	NOM_CALLE	Nombres de calles de España	25 caracteres
Área	NOM_AREA	Nombres de áreas de las poblaciones de España	20 caracteres
Población	NOM_POBLACION	Nombres de las poblaciones de España	15 caracteres
Teléfono	NUM_TEL	Números de teléfono de España	9 caracteres
Fax	NUM_FAX	Números de fax de España	9 caracteres



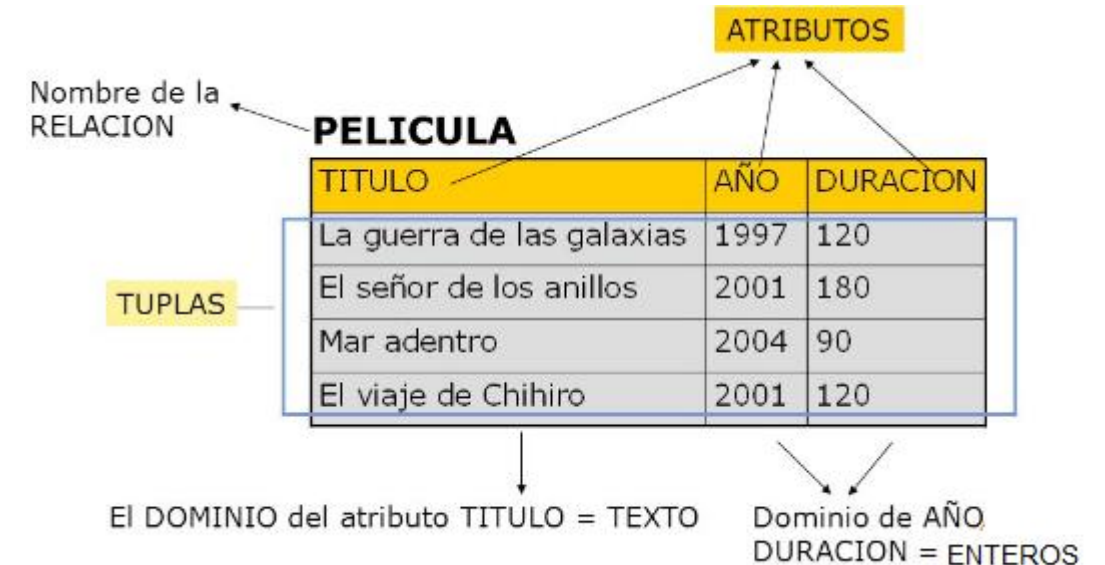
RELACIONES

Atributos

- Cada **atributo** de una base de datos relacional se define sobre un dominio.
- Un atributo representa una propiedad de una relación.
- Un atributo toma valores de un dominio.
- Varios atributos distintos (de la misma o de diferentes relaciones) pueden tomar sus valores del mismo dominio

- Notación:

$D = \text{Dom}(A) \Rightarrow D$ es el dominio de **A**





RELACIONES

Tabla vs. relación

- Una **relación** es un concepto abstracto de origen matemático.
- Una **tabla** es una forma de representar (implementar) una relación (una estructura de datos).
- Las **columnas** son los campos o atributos que definen la tabla. Cada campo está definido por:
 - **Nombre**: que describe los datos almacenados en él.
 - **Dominio**: que indica el tipo de valores que contendrá dicho campo.
- Cada **atributo** está definido sobre un **dominio**.



RELACIONES

•La **cabecera o esquema** de una **relación** se expresa como:

relación (**nombre_campo_1:dominio_1, nombre_campo_2:dominio_2,...,**
nombre_campo_m: dominio_m)

PERSONA (**NOMBRE: cadena(30), APELLIDO1: cadena(30), APELLIDO2: cadena(30),**
DIRECCION: cadena(100), EDAD: número, TELEFONO_FIJO: cadena(9))

Ejemplo de una **tupla**:

(Alfonso, Bonillo, Sierra, C/
Isla de Arosa, 7. Pamplona, 38,
948 363366)

PERSONA					
NOMBRE	APELLIDO1	APELLIDO2	DIRECCIÓN	EDAD	TELEFONO_FIJO
Alfonso	Bonillo	Sierra	C/ Isla de Arosa, 7. Pamplona	38	948 363366
Sebastián	López	Ojeda	C/ Islas Vírgenes, 21. Olite	37	948 192021
Narciso	Jáimez	Toro	C/ Isla de Java, 11. Carcastillo	38	948 111213
Gonzalo	Fernández	Hernández	C/ Ibiza, 10. Murcia	29	968 353535
Diego	Rodríguez	Gracia	C/ Menorca, 17. Valencia	36	96 0151617
Silvia	Thomas	<u>Barrós</u>	C/ Islas del Caribe, 19. Granada	34	958 190405
Miguel Ángel	Pérez	Martínez	C/ Isla Cristina, 12. Murcia	31	968 343434
Alberto	Domínguez	Vega	C/ Formentera, 9. Sevilla	35	95 0123454
Manuel	Rubia	Mateos	C/ Mallorca, 6. Jaén	36	953 100908



RELACIONES

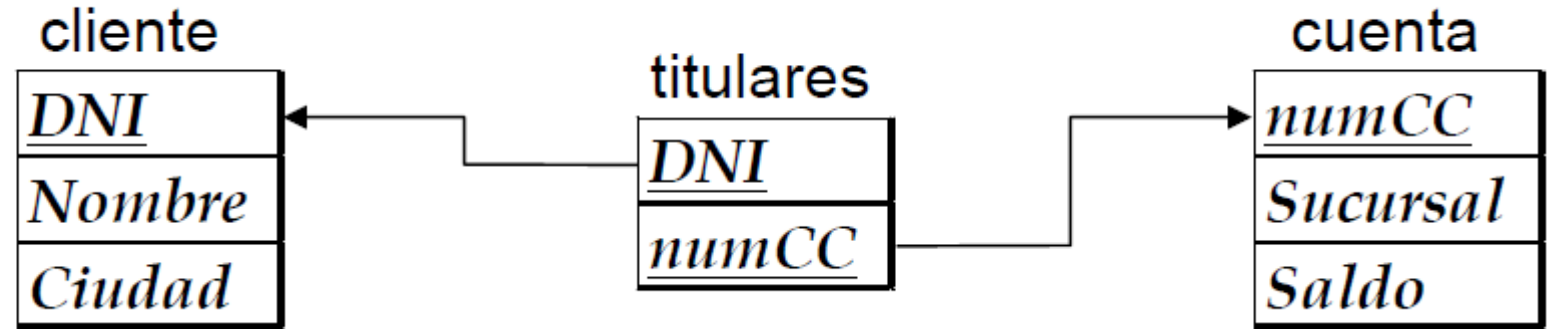
- ✓ NOMBRE es una cadena de caracteres de a lo sumo 30 caracteres.
- ✓ DIRECCION es una cadena de caracteres de a lo sumo 100 caracteres.
- ✓ EDAD es un número.
- ✓ TELEFONO_FIJO es una cadena de caracteres de longitud 9 caracteres.
- Los valores de los campos son **atómicos** (indivisibles).
 - En este caso las direcciones no parecen atómicas.
 - Un valor se considera atómico o no según las restricciones del sistema de información que tengamos.
 - **En unos casos nos interesa tener la dirección lo más descompuesta posible para poder tener más posibilidades de combinación de los campos y en otros casos nos interesa más tener toda la dirección en un único campo**



RELACIONES

ejemplo de B.D. Bancaria:

diagrama relacional



Dominios:

tpDNI = 0..999999999;

tpNombre = cadena(50);

esquemas de relación:

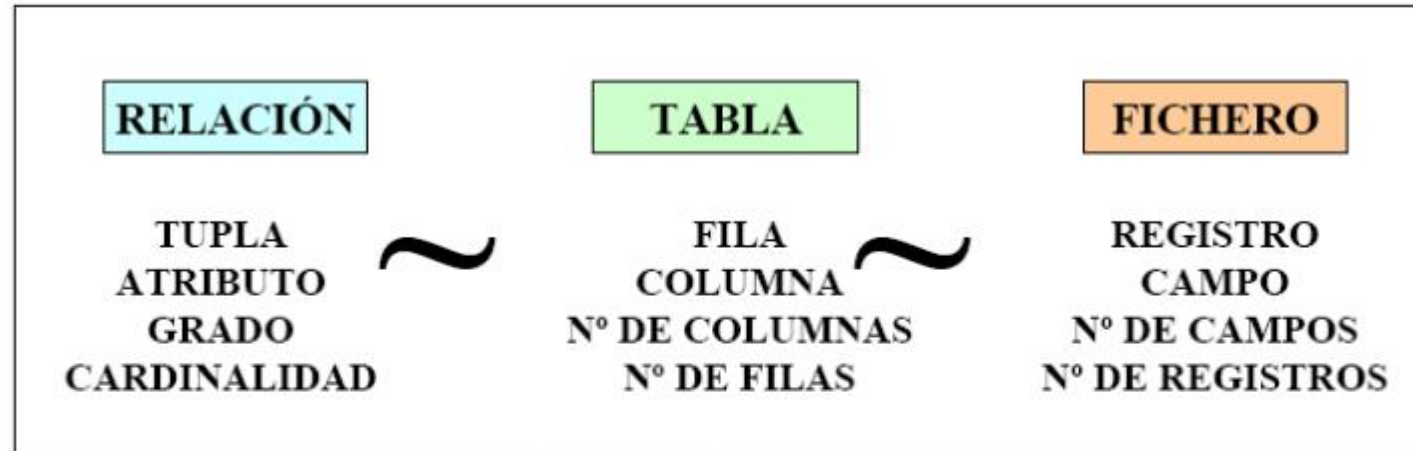
cliente = (DNI : **tpDNI**; Nombre : **tpNombre**; Ciudad : cadena(20));

cuenta = (numCC : entero; Sucursal : cadena(20); saldo : real);

titulares = (DNI : tpDNI; numCC : entero);



RELACIONES



Modelo
Relacional
(teoría)

SGBD
Relacionales
(implementación)

Sistemas
de Ficheros
Clásicos

!CUIDADO!, una relación no es una tabla. Ni una tabla es un fichero. Existen diferencias entre los conceptos.



CLAVES

Claves:

- **Clave de una Relación**, es aquel o aquellos atributos que determinan de forma unívoca y mínima a una tupla o fila de la relación. Siempre tiene que existir al menos una clave (en el peor de los casos formada por todos los atributos), ya que no pueden existir filas duplicadas.
- Así una clave debe cumplir dos requisitos:
 - Identificación unívoca de cada fila de la tabla.
 - No redundancia: no se puede descartar ningún atributo de la clave para identificar la fila.
- Tipos de claves de claves en el modelo relacional: **candidata, primaria, alternativa y ajena**.



CLAVES

Claves candidatas:

- Es un conjunto mínimo y no vacío de atributos que identifican unívoca cada tupla de una relación. Puede haber varias. Siempre hay una clave candidata en el peor de los casos toda la fila:
 - EMPLEADO (cod_emp, nif, nombre, apellidos)
 - Cod_emp y nif son claves candidatas

Clave primaria:

- Es aquella clave candidata que elige el usuario para identificar las tuplas de la relación.
 - EMPLEADO (cod_emp, nif, nombre, apellidos)
 - Elegimos Cod_emp como clave primaria

Claves alternativas:

- Las claves candidatas que no han sido escogidas como clave primaria.



Cuando analicemos la clave principal debemos tener en cuenta que:

- Sus valores siempre han de ser conocidos (no nulos).
- La memoria que ocupen ha de ser mínima.
- Su codificación ha de ser sencilla.
- Se utilizará en otras tablas para crear “interrelaciones”, mediante “claves ajenas”.

EMPLEADO					
NUM_EMPLEADO	DNI	OFICINA	TELEFONO_FIJO	TELEFONO_MOVIL	NUM_FAX
1234	12345678	005	948 235689	655 191919	948 363366
1235	23456789	002	948 235698	655 212019	948 192021
1236	13467925	006	948 457812	655 242563	948 111213
1237	25836941	007	976 456985	655 256985	976 353535
1238	42589637	004	978 191921	655 658965	978 151617

- Dos claves candidatas: NUM_EMPLEADO y DNI.
- Clave primaria: DNI.
- Clave alternativa: NUM_EMPLEADO



CLAVES

Clave ajena:

- Está formada por una o más columnas de una tabla cuyos valores corresponden con los de la clave primaria de otra o la misma tabla.
- Representan **relaciones o asociaciones entre tablas**.
- Destacar que la clave ajena y la correspondiente clave primaria han de estar definidas sobre los mismo dominios.
- Puede tomar valores nulos.

TABLA LIBROS

ID	TITULO	ID_AUTOR
1	Cinco horas con Mario	2
2	El Quijote	3
3	Las Ratas	2
4	Rinconete y Cortadillo	3
5	La insoportable levedad del ser	1

TABLA AUTORES

ID	NOMBRE	NACIONALIDAD
1	Milan Kundera	Chequia
2	Miguel Delibes	España
3	Miguel de Cervantes	España



CLAVES

Las **claves primarias y ajenas** cumplen estas propiedades:

- Las **claves ajenas** son esenciales en el Modelo Relacional ya que permiten enlazar distintas tablas de la base de datos.
- Una clave ajena y la clave primaria de la tabla referenciada asociada han de estar definidas sobre los mismos dominios.
- Una tabla puede poseer más de una clave ajena, de hecho tendrá una clave ajena por cada tabla referenciada de la cual dependa.
- Una tabla puede no tener ninguna clave ajena.
- Una clave ajena puede relacionar una tabla consigo misma (relaciones reflexivas).



ESQUEMA DE UNA BASE DE DATOS RELACIONAL

- Una base de datos relacional es un conjunto de relaciones.
- Para representar el **esquema de una base de datos relacional** se debe dar:
 - El nombre de sus relaciones
 - Los atributos de las relaciones
 - Los dominios sobre los que se definen los atributos
 - Las claves primarias y las claves ajenas

```
CLIENTES(codcli, nombre, dirección, codpostal, codpue)
VENEDORES(codven, nombre, dirección, codpostal, codpue, codjefe)
PUEBLOS(codpue, nombre, codpro)
PROVINCIAS(codpro, nombre)
ARTÍCULOS(codart, descrip, precio, stock, stock_min, dto)
FACTURAS(codfac, fecha, codcli, codven, iva, dto)
LÍNEAS_FAC(codfac, línea, cant, codart, precio, dto)
```



ESQUEMA DE UNA BASE DE DATOS RELACIONAL

Grafos relacionales

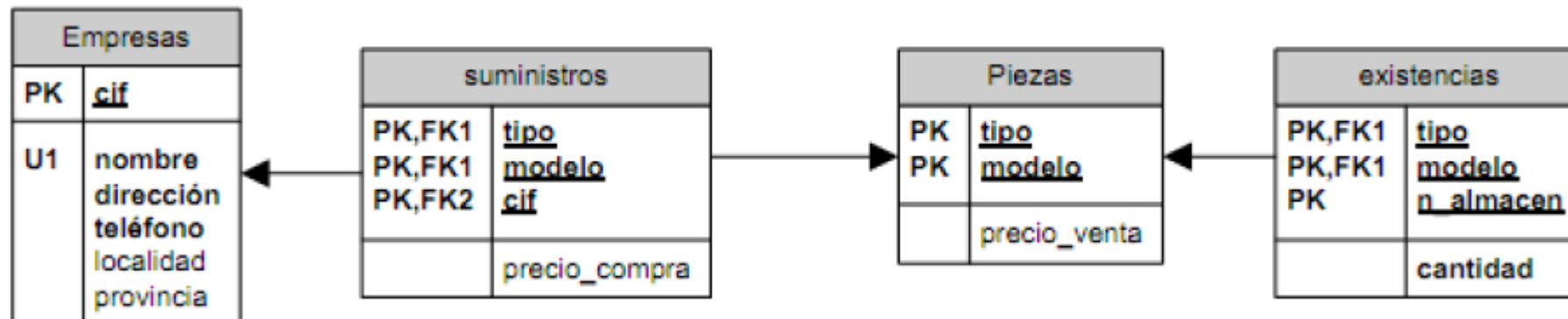
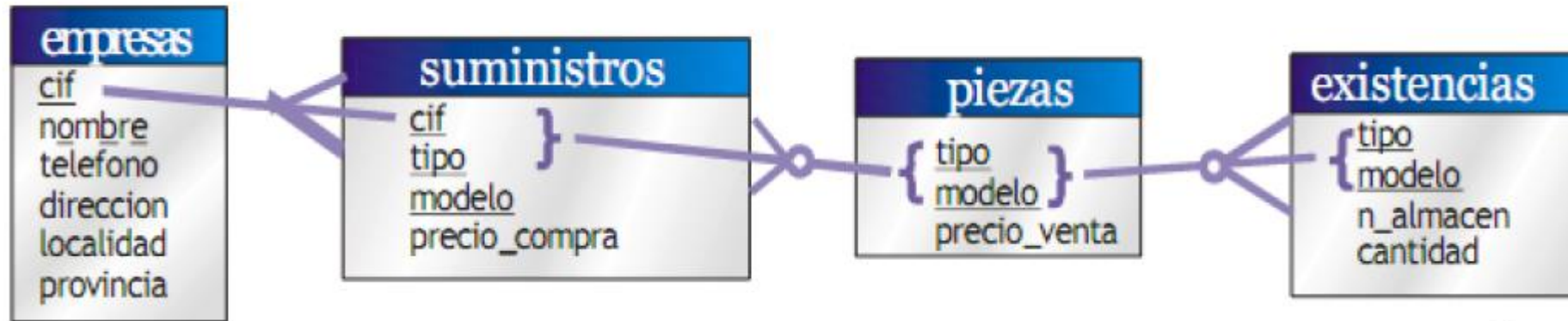
- Es un esquema relacional donde hay líneas que enlazan las claves ajenas con la relación que referencian.





ESQUEMA DE UNA BASE DE DATOS RELACIONAL

Esquemas relacionales derivados del modelo entidad/relación





REGLAS DE INTEGRIDAD

Reglas de integridad

- Restricciones que los datos almacenados deben cumplir para garantizar que son correctos.
 - Nulo.
 - Regla de integridad de la entidad.
 - Regla de integridad referencial.
 - Reglas de negocio.
- Las **restricciones inherentes** vienen impuestas por el propio modelo de datos relacional.
 - No puede haber dos tuplas iguales
 - El orden de las tuplas no es significativo.
 - El orden de los atributos no es significativo.
 - Cada atributo sólo puede tomar un único valor del dominio subyacente.



REGLAS DE INTEGRIDAD

Concepto de **nulo** en el modelo relacional:

- Se puede definir el **valor nulo** como una marca utilizada para representar información desconocida, inaplicable etc.
- Un atributo es NULO (null) cuando dicho atributo es desconocido -> se utiliza el valor **NULL**.

Ejemplos: "no se sabe la dirección de una persona", "se desconoce su fecha de nacimiento", etc.

- EMPLEADO (1234, 12345678, 005, null, 655191919, 948363366)
 - En este caso no conocemos el valor del campo "TELEF_FIJO".
- OFICINA (005, "Islas Vírgenes, 19", null, null, 948191919, 948363366)
 - No conocemos ni el campo "NOM_AREA", ni el "NOM_POBLACION".



REGLAS DE INTEGRIDAD

Integridad de la entidad

- Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo.
- Es el mecanismo que garantiza la identificación y **unicidad** de las tuplas en una relación.

PEDIDO			
DNI_CLIENTE	COD_TIENDA	FECHA_PEDIDO	CUANTIA
23456987	005	19/04/05	1904€
22335566	007	08/02/06	806€
34562358	019	21/05/06	2525€
23569856	012	20/06/05	235€
23456987	005	19/04/05	1904€



Clave primaria:
DNI_CLIENTE,
COD_TIENDA y
FECHA_PEDIDO



REGLAS DE INTEGRIDAD

Integridad referencial

- Se encargará de resolver cuestiones como las siguientes:
 - ❖ ¿Qué ocurriría si introdujésemos en un pedido el código de un proveedor que no existe en la BD?
 - ❖ ¿Qué ocurrirá con los pedidos que contienen un código de proveedor cuando ese proveedor fue dado de baja en la BD porque ya no nos suministra?
 - ❖ ¿Deberíamos borrar todos los pedidos de ese proveedor?
 - ❖ ¿Deberíamos permitir pedidos con un proveedor que ya no existe?

Regla de integridad referencial

- **Todo valor no nulo de una clave ajena, debe coincidir necesariamente con algún valor de la clave primaria a que hace referencia.**
 - Dicho de otra manera, si una relación R2 (**relación que referencia**) tiene un atributo que es la clave primaria de otra relación R1 (**relación referenciada**), todo valor de dicho atributo debe coincidir con un valor de la clave primaria de R1 o ser nulo. El atributo en cuestión es, por tanto, una **clave ajena** de la relación R2.

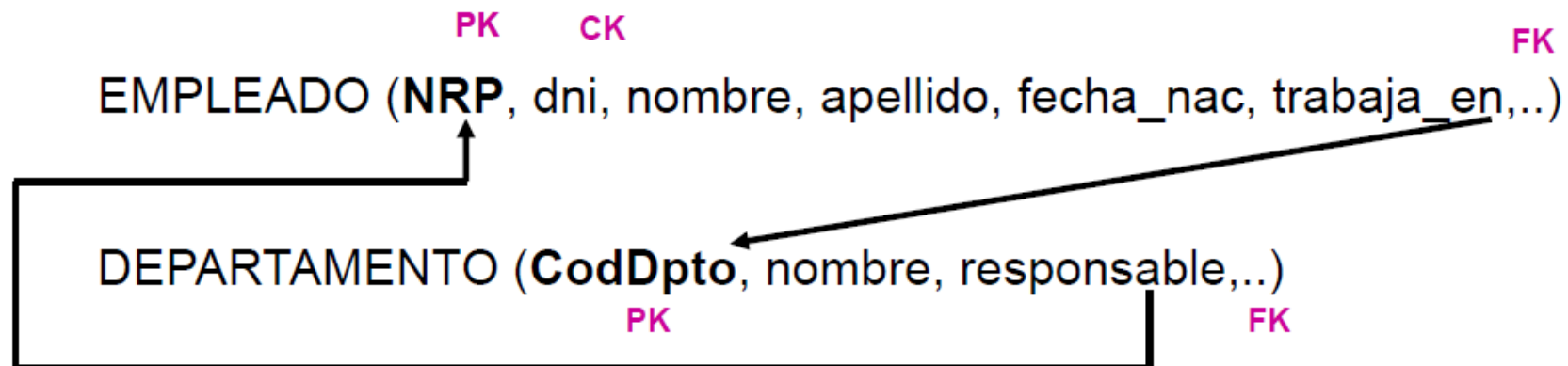
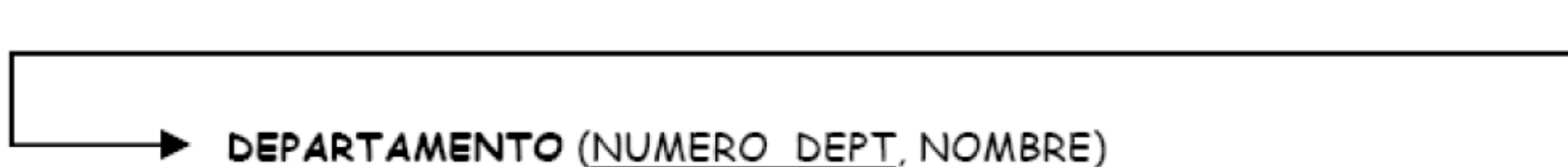


REGLAS DE INTEGRIDAD

Clave ajena o foránea (FOREIGN KEY)

- Los valores de la clave ajena en la tabla hija deben corresponderse con los valores de la clave principal en la tabla padre o bien ser nulos.
- Los nombres de los atributos relacionados no tienen por qué ser iguales.

EMPLEADO (NUM_EMPLEADO, NOMBRE, SALARIO, TELEFONO, *DEPARTAMENTO*)





REGLAS DE INTEGRIDAD

DEPARTAMENTO

NumDept	Nombre
1	Contabilidad
2	Ventas
3	Compras

EMPLEADO

NumEmpleado	Nombre	Salario	Telefono	Departamento
1	Antonio Lopez	1.300 €	976 111 222	1
2	Carmen Garcia	1.100 €	976 222 333	1
3	Felipe Sanchez	1.100 €	976 333 444	2
4	Manuel Izquierdo	1.300 €	976 444 555	3
5	Inmaculada López	1.400 €	976 555 666	4

ERROR



REGLAS DE INTEGRIDAD

Tras definir las claves ajenas, hay que determinar las consecuencias que tienen las operaciones de borrado y modificación realizadas sobre tuplas de **la relación referenciada**. Opciones:

- **Operación restringida (RESTRICT):**
 - El borrado o la modificación de tuplas de la relación que contiene la clave primaria **referenciada** sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena.
 - *Para borrar una editorial de la BD no tendría que haber ningún libro que estuviese publicado por dicha editorial, en caso contrario el sistema impediría el borrado.*
- **Operación con transmisión en cascada (CASCADE):**
 - El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo el borrado o modificación en **cascada** de las tuplas de la relación que contiene la clave ajena.
 - *Al modificar el nombre de una editorial en la relación EDITORIAL, se tendría que modificar también dicho nombre en todos los libros de nuestra BD publicados por dicha editorial.*



REGLAS DE INTEGRIDAD

- **Operación con puesta a nulos (SET NULL):**
 - El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia.
 - Sólo es posible cuando el atributo que es clave ajena admite el valor nulo.
 - *Cuando se borra una editorial, a los libros que ha publicado dicha editorial y que se encuentran en la relación LIBROS se les coloque el atributo nombre_e a nulo.*
- **Operación con puesta a valor por defecto (SET DEFAULT):**
 - El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner el valor por defecto a la clave ajena de la relación que referencia.
 - El valor por defecto se define al crear la tabla correspondiente.
- **Operación que desencadena un procedimiento de usuario:**
 - El borrado o modificación de tuplas de la tabla referenciada pone en marcha un procedimiento definido por el usuario.
- **Nota:** La opción seleccionada en caso de borrado es independiente de la de modificación.



REGLAS DE INTEGRIDAD

Ejemplo

Codprof	Nombre	Apellido	nif
035	Juan	Lopez	222222
036	Raquel	Lopez	333333
039	Ana	Lopez	444444

Cod_al	Nombre	Curso	Codprof
a21	Julio	1º	035
a23	Brad	3º	035
a24	Martin	5º	039

Si modificamos el código del profesor 035 por 094, en la tabla alumnos se modificará en las dos primeras filas o bien el sistema no nos permitirá modificar este código, esta operación la realizará el SGDB



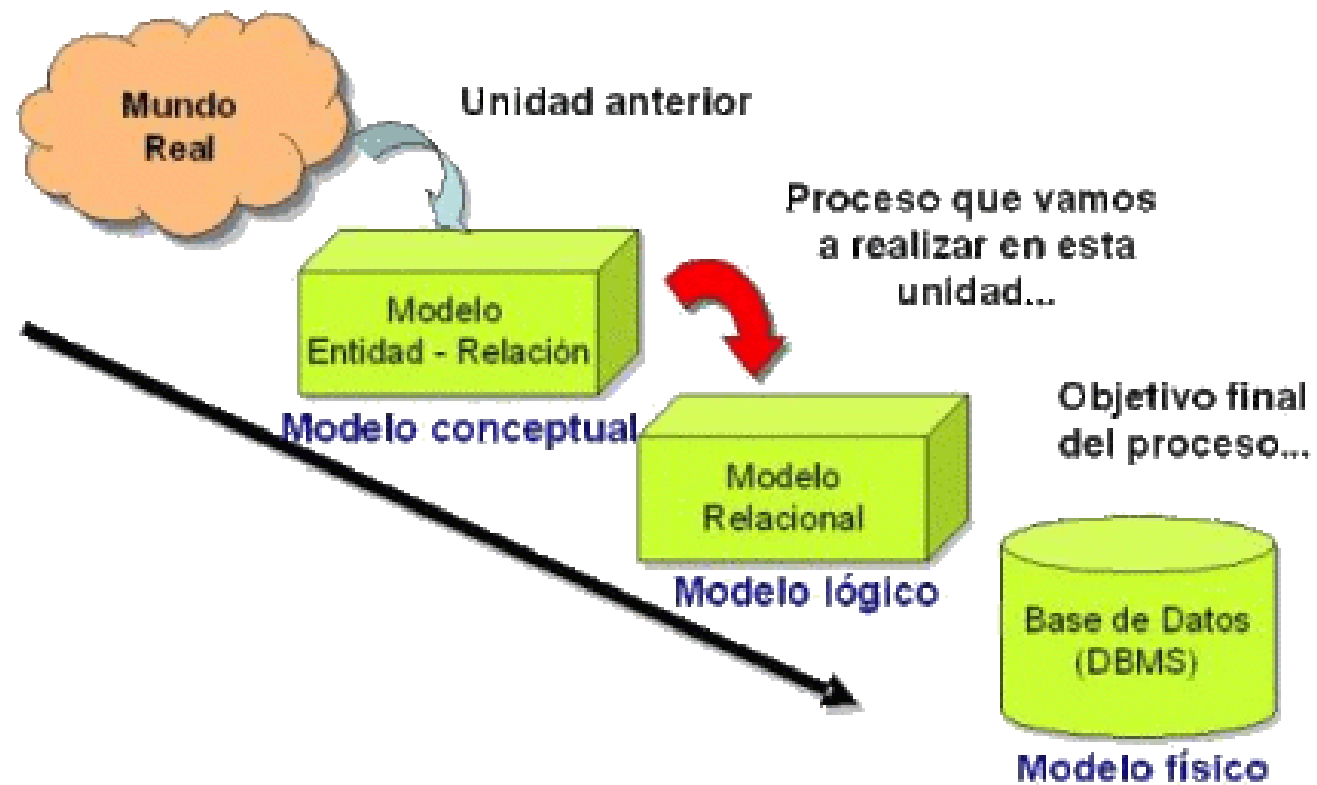
REGLAS DE INTEGRIDAD

Reglas de negocio

- **Comprobación o validación (CHECK)**
 - Por cada valor que se inserta o actualiza comprueba si es correcto.
- **Disparador**
 - Pequeños programas grabados en la base de datos que se ejecutan automáticamente cuando se cumple una determinada condición.
 - Sirven para realizar una serie de acciones cuando ocurre un determinado evento (cuando se añade una tupla o se borra un dato, un usuario abre una conexión, etc.).



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL





TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

- El modelo E/R opera con conceptos, necesitamos obtener las **tablas finales** que implementaremos en la BD y éstas las proporciona el modelo relacional.
- Hay que seguir una serie de técnicas para transformar las distintas entidades y asociaciones del modelo E/R en las correspondientes tablas y relaciones del modelo relacional.
 - Toda **entidad** se transforma en una **relación (tabla)**
 - Las **interrelaciones N:M** se transforman en una **relación (tabla)**
 - Las **interrelaciones 1:N** dan lugar a una **propagación de clave** (se crea una **clave ajena**).



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de las entidades

- Cada tipo de **entidad** se convierte en una **relación o tabla**. La tabla se llamará igual que el tipo de entidad de donde proviene.
- Cada **atributo** de una entidad se transforma en una **columna** de la relación.
 - El atributo identificativo principal pasa a ser **la clave primaria** de la relación, subrayada (PRIMARY KEY).
 - Los atributos identificadores alternativos, deben ser **valores únicos** (UNIQUE), también se podrá indicar si se desea que no puedan ser valores nulos (NOT NULL).
- A cada entidad del modelo E/R le corresponderá una tabla en el modelo relacional y se mantendrán tanto los atributos como la clave primaria.



Producto (cod_prod, nombre, precio, stock)

Almacen, (cod_alm nombre, calle, portal, tfno)



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de atributos de entidades

- Atributos **univaluados** dan lugar a un atributo de la relación.
- Atributos **multivaluados** dan lugar a una nueva tabla cuya clave es la clave primaria de la entidad junto con el nombre del atributo.
- Atributos **compuestos** se transforman en los atributos que los componen

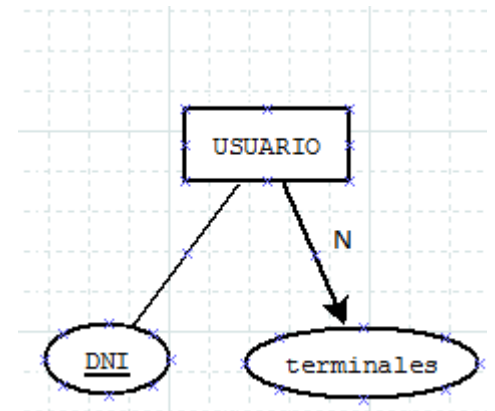
Ejemplo

- Un usuario de una aplicación informática utiliza varios terminales:

USUARIO(DNI, nombre,...)



TERMINAL(DNI, terminal)





TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

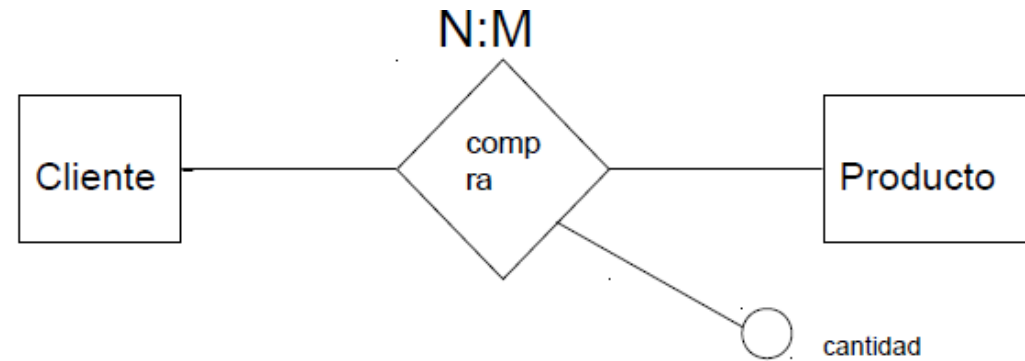
Transformación de interrelaciones N:M

- Se transforma en una tabla que tendrá como **clave primaria** la concatenación de los atributos identificadores de las entidades que relaciona.
- Cada uno de los atributos que forman la clave primaria son **claves ajenas** que referencian a las claves primarias de las entidades interrelacionadas (FOREIGN KEY).
- Si la interrelación posee **atributos**, éstos pasan a formar parte de la nueva tabla.



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Ejemplo



cliente (código_cliente, nombre, apellidos...)

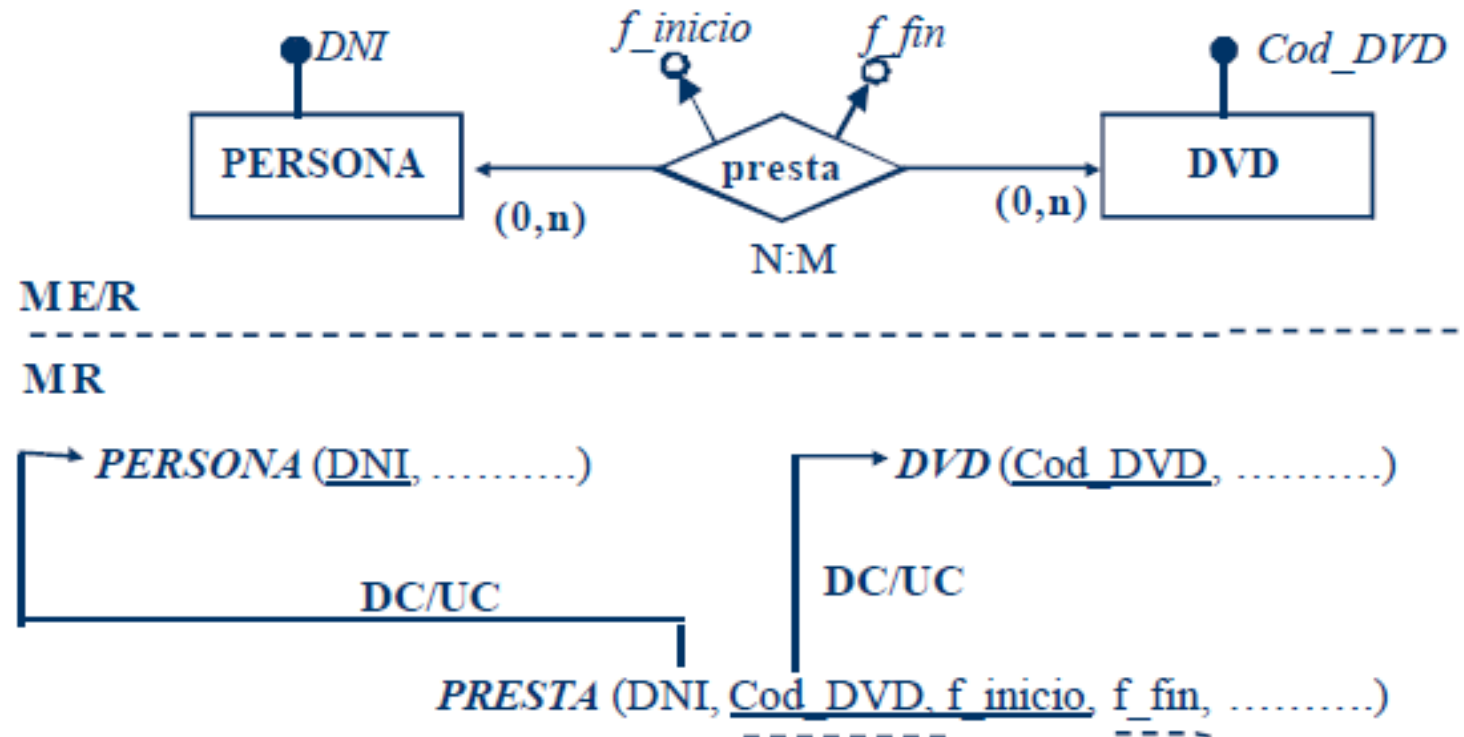
producto (código_producto, nombre, precio...)

compra (código_cliente, código_producto, cantidad,...)



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de interrelaciones – DIMENSIÓN TEMPORAL



En el caso de atributos con una **dimensión temporal** tanto si son multivaluados como univaluados, hay que estudiar la semántica concreta del problema para determinar cuales son los atributos que forman parte de la clave.



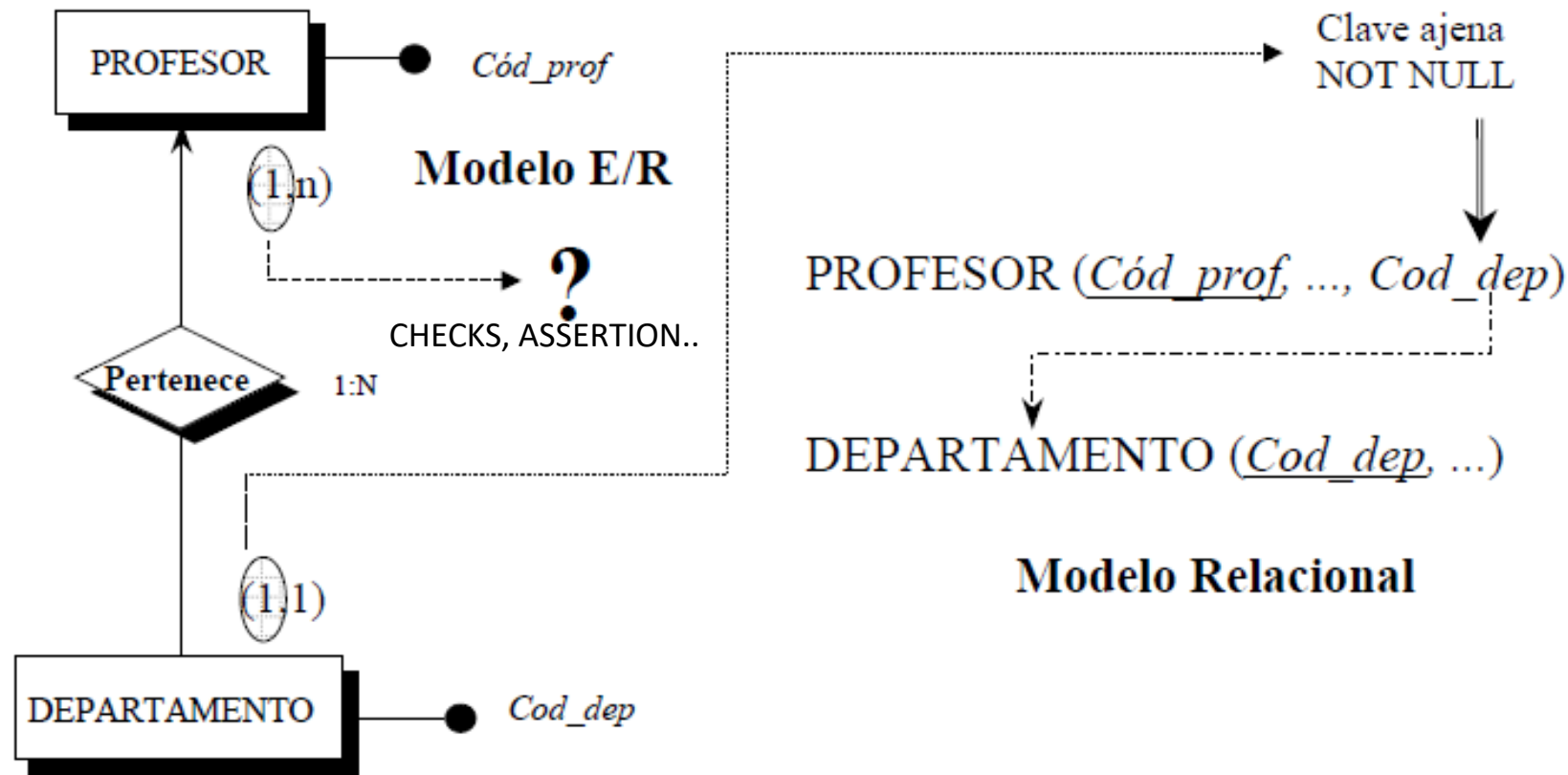
TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de interrelaciones 1:N

- a) **Propagar la clave** del tipo de entidad que tiene de cardinalidad máxima 1 a la que tiene N, desapareciendo el nombre de la interrelación.
- Esta clave será ajena.
 - Admitirá nulos si la cardinalidad es (0,1).
 - Si existen atributos en la interrelación, éstos también se propagarán.
- b) **Transformarlo en una relación**, como si se tratase de una interrelación N:M.
- La clave primaria de la relación creada es sólo la clave primaria de la tabla a la que le corresponde la cardinalidad N.
 - Lo normal es el primer caso pero puede ser apropiado en los casos siguientes:
 - Cuando el número de valores interrelacionados de la entidad que propaga su clave es muy pequeño y por tanto existirían muchos valores nulos en la clave propagada.
 - Cuando se prevé que dicha interrelación en un futuro se convertirá en una de tipo N:M.
 - Cuando la interrelación tiene atributos propios y no deseamos propagarlos.



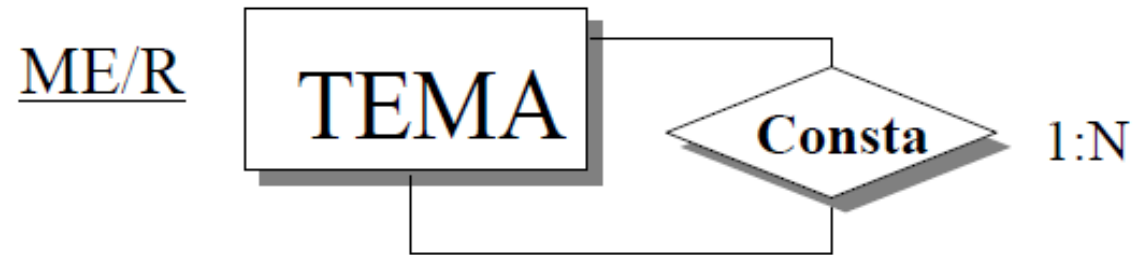
TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Ejemplo Interrelaciones 1:N (cardinalidad mínima 1 en el Padre)



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Ejemplo Interrelaciones 1:N REFLEXIVAS

Solución a: *propagación de clave*

MR

→ TEMA (Cód_tema, ..., Cod_tema_sup)
(borrado: puesta a nulos; modificación: cascada)

Solución b: *nueva tabla*

→ TEMA (Cód_tema, ...)
CONSTA (Cód_tema, Cod_tema_sup...)
(borrado y modificación: cascada)

Nulos no permitidos

Transformación de una interrelación 1:N en una nueva tabla



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de Interrelación 1:1

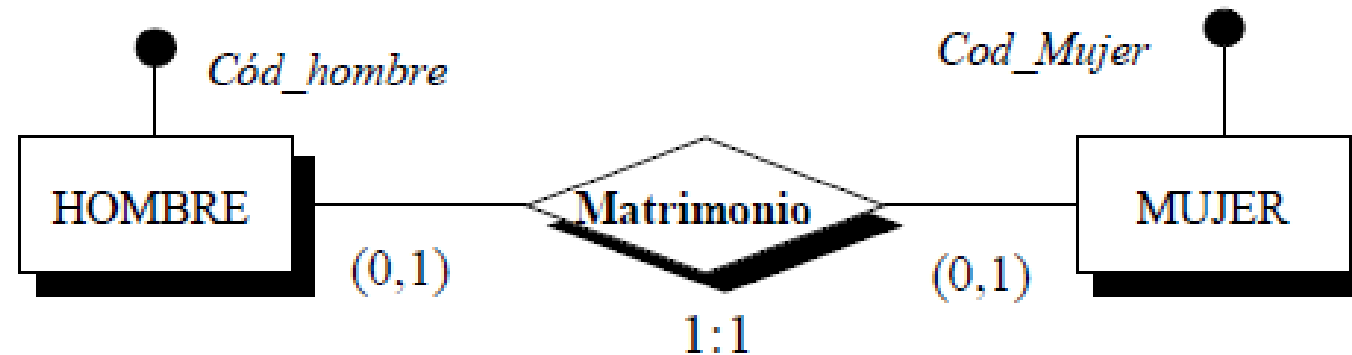
- Es un caso particular de una N:M o también de una 1:N, por lo tanto se puede aplicar la regla de crear una relación o aplicar la de propagar la clave correspondiente.
- En este último caso hay que observar que en una interrelación 1:1, la **propagación de la clave** puede efectuarse en ambos sentidos.
- Los **criterios** para aplicar una u otra regla se basan en las cardinalidades mínimas.
 - Si la relación es entre entidades con cardinalidades (0,1) y (0,1), es conveniente transformar la interrelación 1:1 en una relación.
 - Si la relación es entre entidades con cardinalidades (0,1) y (1,1), conviene propagar la clave de la entidad con cardinalidades (1,1) a la tabla resultante de la entidad de cardinalidades (0,1).
 - Si la relación es entre entidades con cardinalidades (1,1) y (1,1), se puede propagar la **clave** de cualquiera de ellas a la tabla resultante de la otra.



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de Interrelación 1:1. Cardinalidades (0,1) y (0,1)

Modelo E/R



Modelo Relacional

MATRIMONIO (Cod_hombre, Cod_mujer)

HOMBRE (Cod_hombre)

MUJER (Cod_mujer)

Clave alternativa

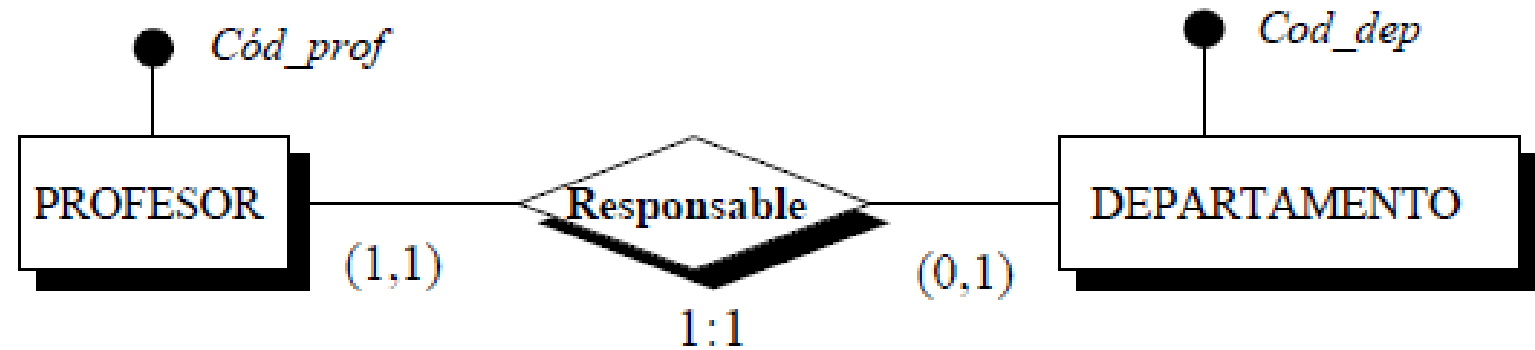
(UNIQUE, NOT NULL)



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de Interrelación 1:1. Cardinalidades (1,1) y (0,1)

Modelo E/R



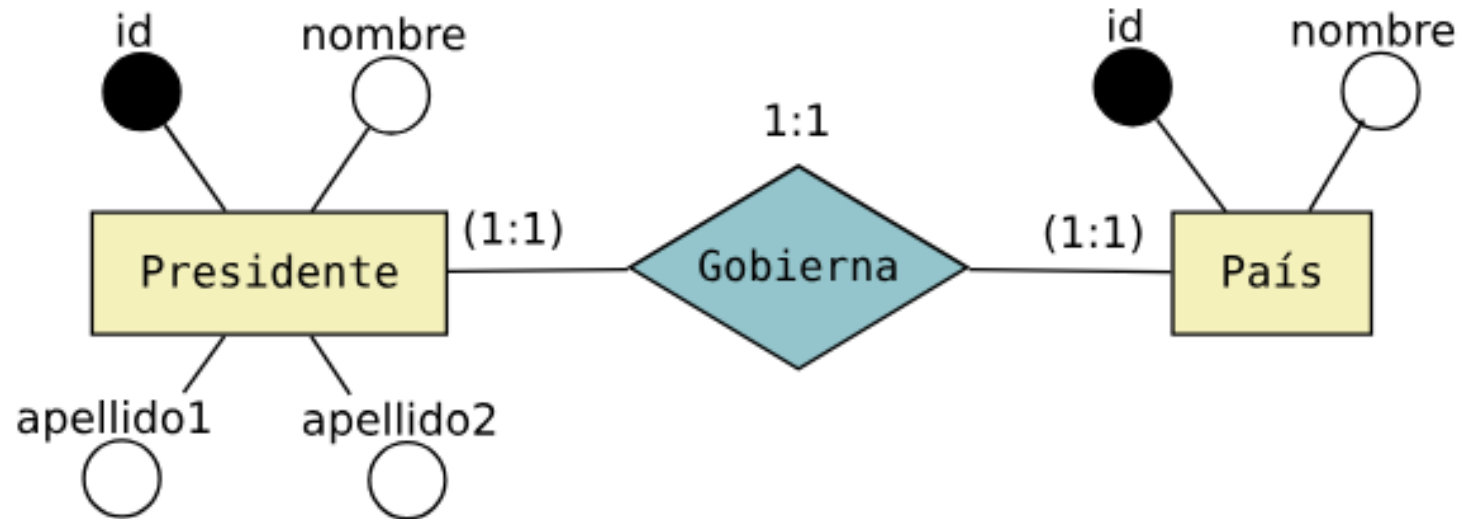
Modelo Relacional

PROFESOR (*Cód_prof*)DEPARTAMENTO (*Cód_dep*, ..., *Cód_prof*)

Clave ajena
NOT NULL



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de Interrelación 1:1. Cardinalidades (1,1) y (1,1)**Opción A:**PRESIDENTE(**id**, nombre, apellido1, apellido2)PAÍS(**id**, nombre, *id_presidente*)*id_presidente*: Clave Ajena de PRESIDENTE(id)*id_presidente*: UNIQUE y NOT NULL**Opción B:**PAÍS(**id**, nombre)PRESIDENTE(**id**, nombre, apellido1, apellido2, *id_país*)*id_país*: Clave Ajena de PAÍS(id)*id_país*: UNIQUE y NOT NULL

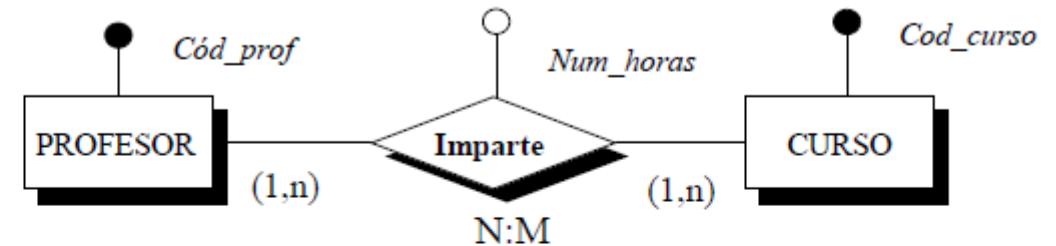


TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

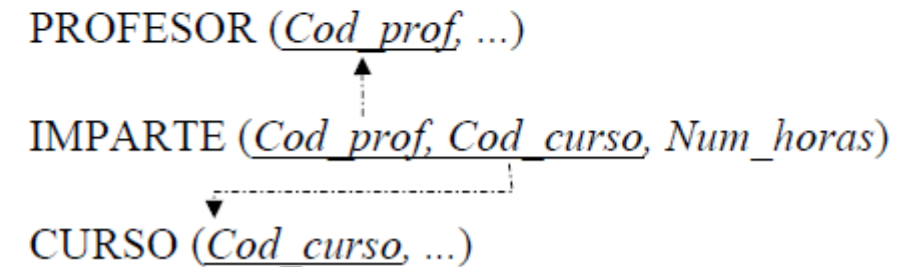
Transformación de atributos de interrelaciones

- Si la interrelación se transforma en una relación o tabla, todos sus atributos pasan a ser columnas de la relación.
- Si la interrelación se transforma mediante propagación de clave, sus atributos migran junto a la clave a la relación que corresponda.

Modelo E/R



Modelo Relacional





TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de dependencias en identificación y en existencia

- **Dependencia en existencia**
 - Se propaga la clave, creando una **clave ajena**, con nulos no permitidos, en la relación de la entidad dependiente, con la característica de obligar a una modificación en cascada (ON UPDATE CASCADE) y a un borrado en cascada (ON DELETE CACADE)
- **Dependencia en identificación**
 - La clave primaria de la relación en la que se ha transformado la entidad débil debe estar formada por la concatenación de las claves de las dos entidades participantes en la interrelación.



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de dependencias en identificación y en existencia

En Existencia:

Cod_edición: Clave Primaria

Cod_curso: **Solo clave Ajena**

En Identificación:

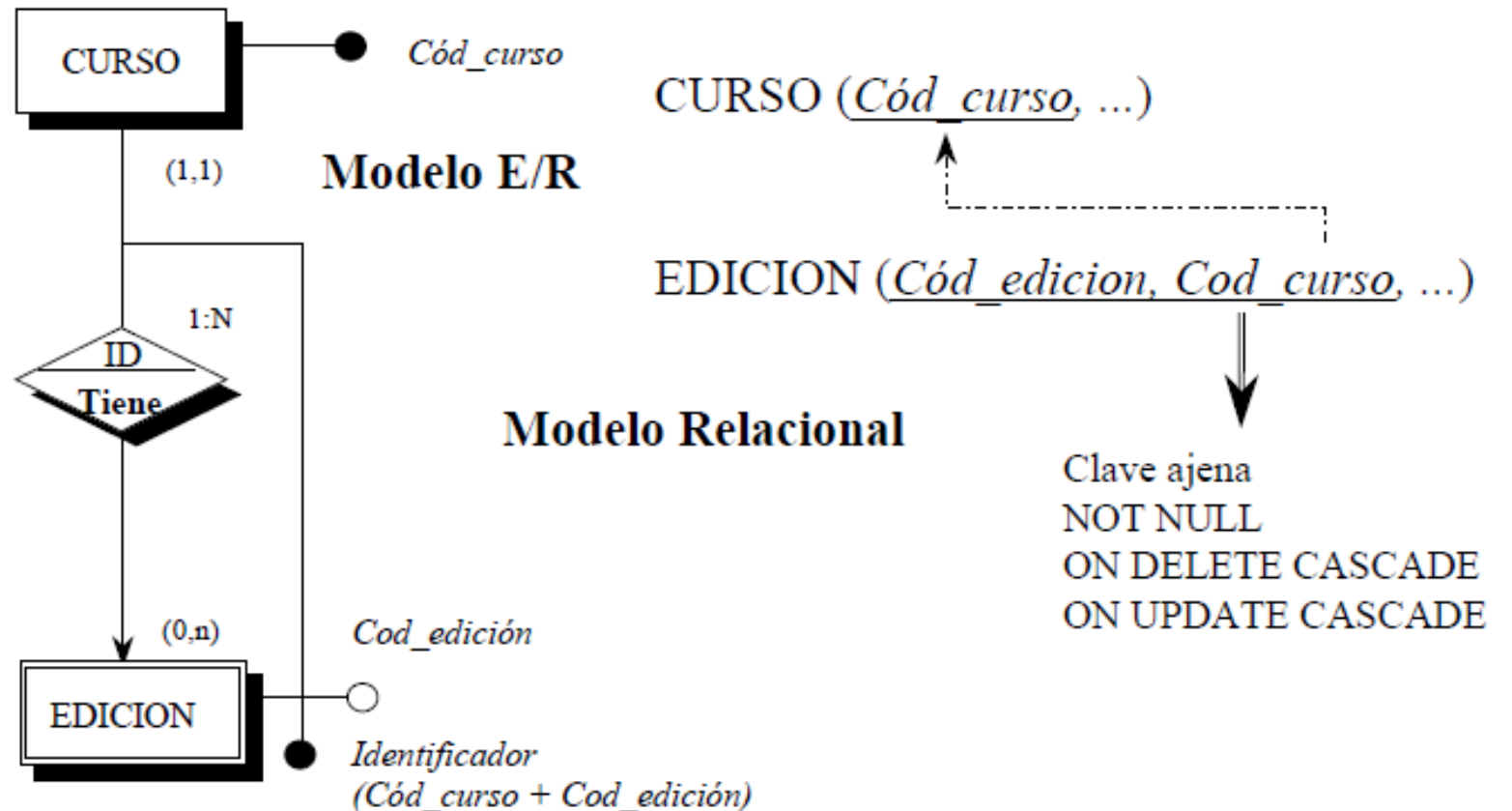
Clave Primaria y Ajena:

Cod_curso + *Cod_edición*

Restricciones Adicionales:

Cod_curso:

- NOT NULL
- ON UPDATE CASCADE
- ON DELETE CASCADE





TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de relaciones ternarias

- **Relaciones N:N:N**

- Nueva relación (tabla) que incluye los atributos propios de la relación.
- La **clave primaria** es la concatenación de las claves primarias de las tablas surgidas al transformar las entidades que forman parte de la relación.
- Cada atributo que forma parte de la clave primaria es una **clave ajena** respecto a la tabla de la que el atributo es clave primaria.

- **Relaciones N:N:1**

- Nueva relación (tabla).
- La **clave primaria** es la concatenación de las claves primarias de las tablas que corresponden a la cardinalidad N.
- Se incluyen como atributos los atributos propios de la relación más la clave primaria de la tabla que corresponde a la cardinalidad 1.
- Cada uno de los atributos que forman la clave primaria de esta tabla (y los atributos añadidos de la relación de cardinalidad 1) son **claves ajenas**.



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de relaciones ternarias

• Relaciones N:1:1

- Nueva relación (tabla).
- La **clave primaria** es la clave primaria de la tabla que corresponde a la cardinalidad N junto con la clave primaria de una de las tablas con cardinalidad 1.

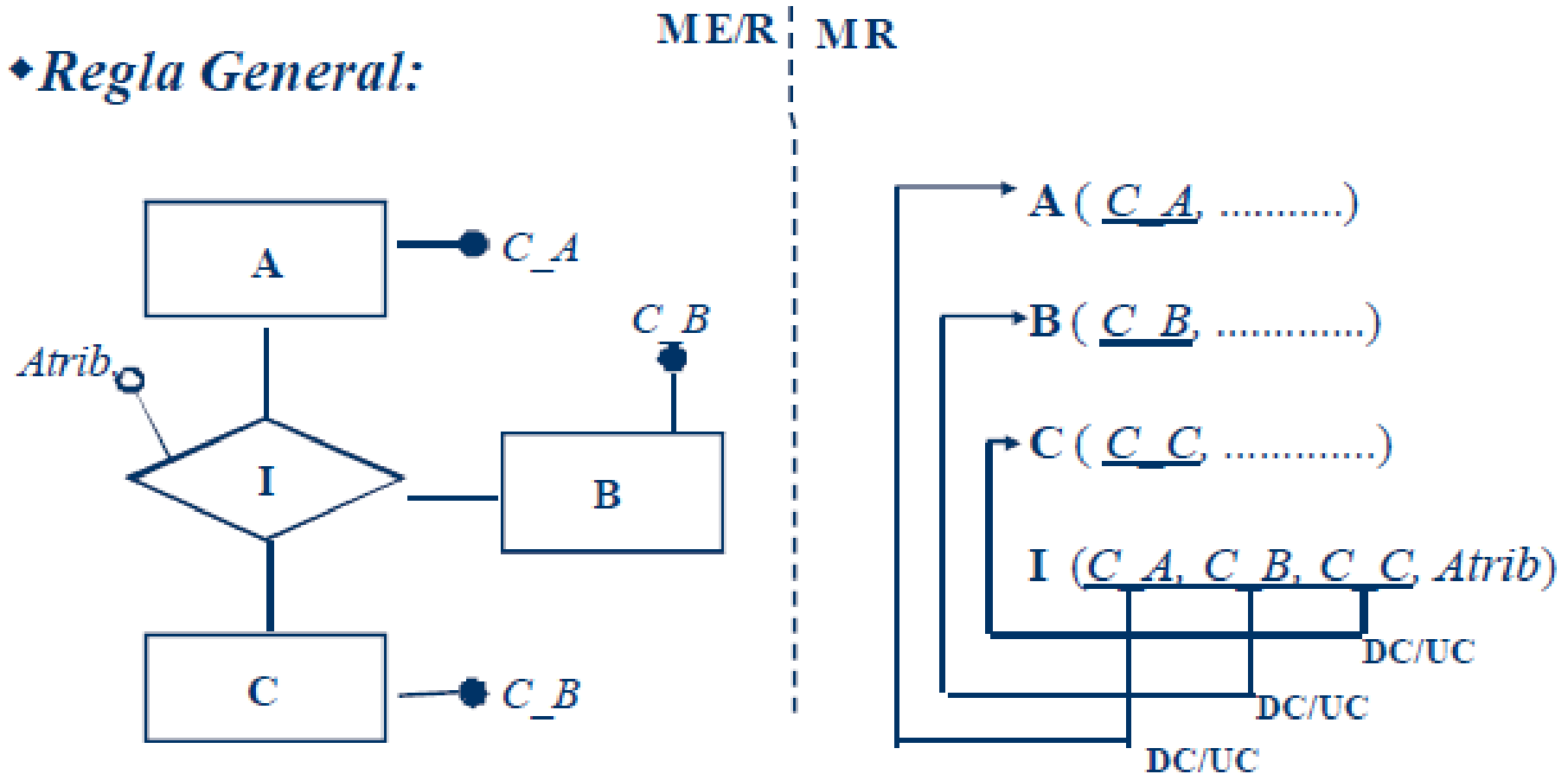
• Relaciones 1:1:1

- Nueva relación (tabla).
- La **clave primaria** es la clave primaria de dos de las tablas.



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

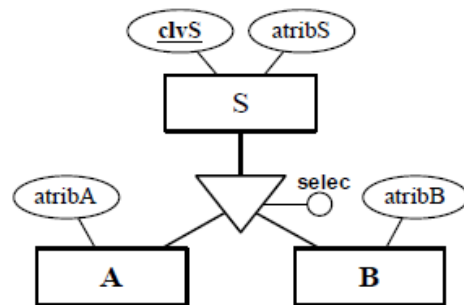
Transformación de relaciones ternarias

♦ *Regla General:*



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Transformación de tipos y subtipos (generalización/especialización)



AtribA	A...	<u>clvS</u>	selec	AtribS...	AtribB	B...

AtribA	A...	<u>clvS</u>	selec	AtribS...	AtribB	B...

①

<u>clvS</u>	selec	AtribS...

AtribA	A...	<u>clvS</u>

<u>clvS</u>	AtribB	B...

②

AtribA	A...	<u>clvS</u>	selec	AtribS...

<u>clvS</u>	selec	AtribS...	AtribB	B...

③

+ tabla
con resto



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Opciones:

- a) Englobar todos los atributos de la entidad y sus subtipos en **una sola relación**, añadiendo un atributo indicando el tipo.
 - Adoptaremos esta solución cuando los subtipos se diferencien en muy pocos atributos y las interrelaciones que los asocian con el resto de las entidades del esquema sean las mismas para todos los subtipos.
- b) Crear **una tabla para el supertipo y tantas tablas como subtipos** haya cuya clave la heredarán del supertipo y sus atributos correspondientes.
 - Ésta es la solución adecuada cuando existan muchos atributos distintos entre los subtipos y se quieren mantener los atributos comunes en una relación.
 - La tabla creada para el supertipo podrá contener el atributo discriminante de la jerarquía

**TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL**

- c) Considerar solo relaciones distintas para cada subtipo, que contengan además de los atributos propios, los atributos comunes. No crear entidad con el supertipo.
- Se elegiría esta opción cuando se dieran las mismas condiciones que en el caso anterior y los accesos realizados sobre los datos de los distintos subtipos siempre afectan a atributos comunes.

1ª solución

Empleados (cod_emp, Nif, nombre, categoria, tipo, pulsaciones, titulo, años_exp)

2ª solución

Empleados(cod_emp, Nif, nombre, categoria,...)

Secretario(cod_emp, pulsaciones)

Ingeniero(cod_emp, titulo)

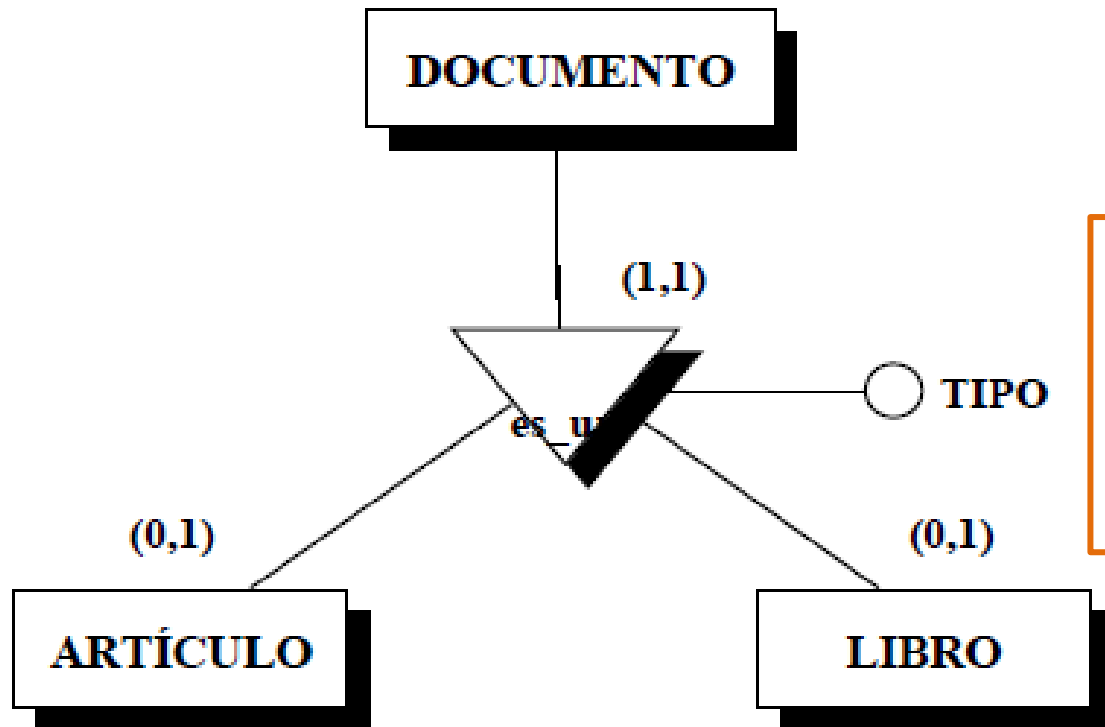
Tecnico(cod_emp, años_exp)



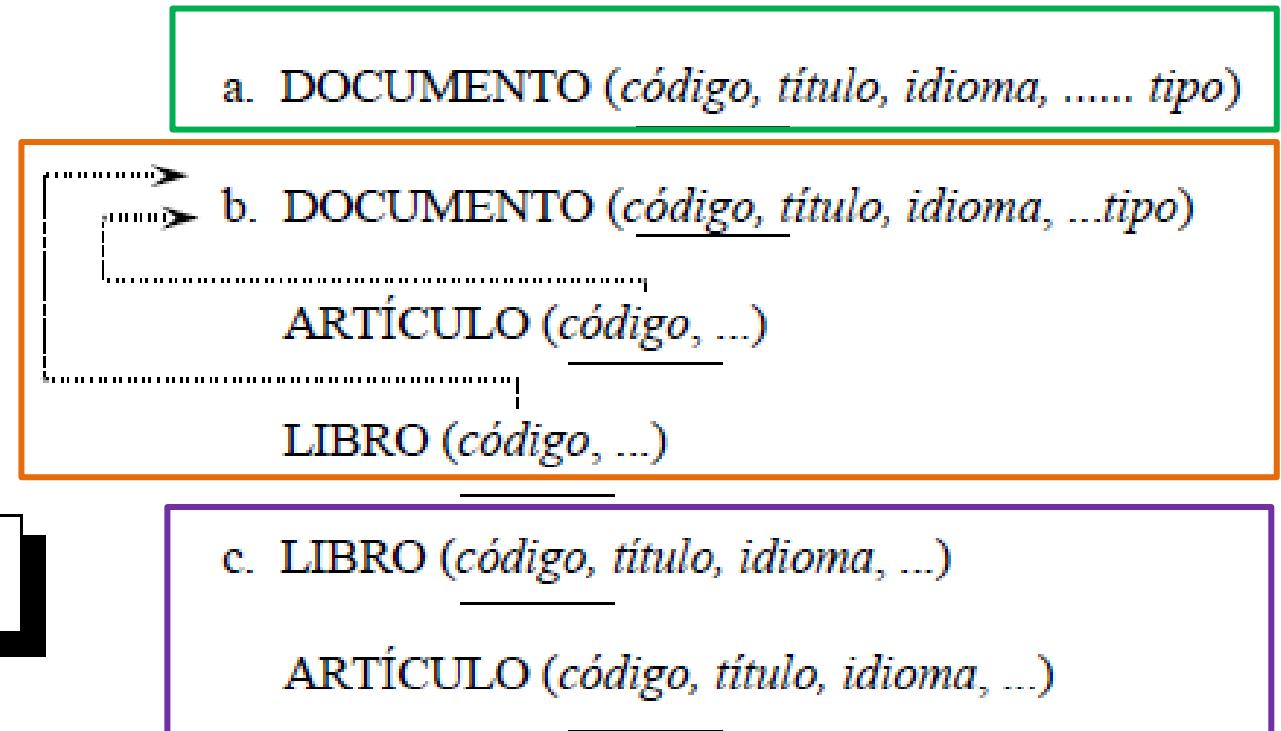
TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

EJEMPLO CON LAS TRES POSIBLES OPCIONES:

ME/R



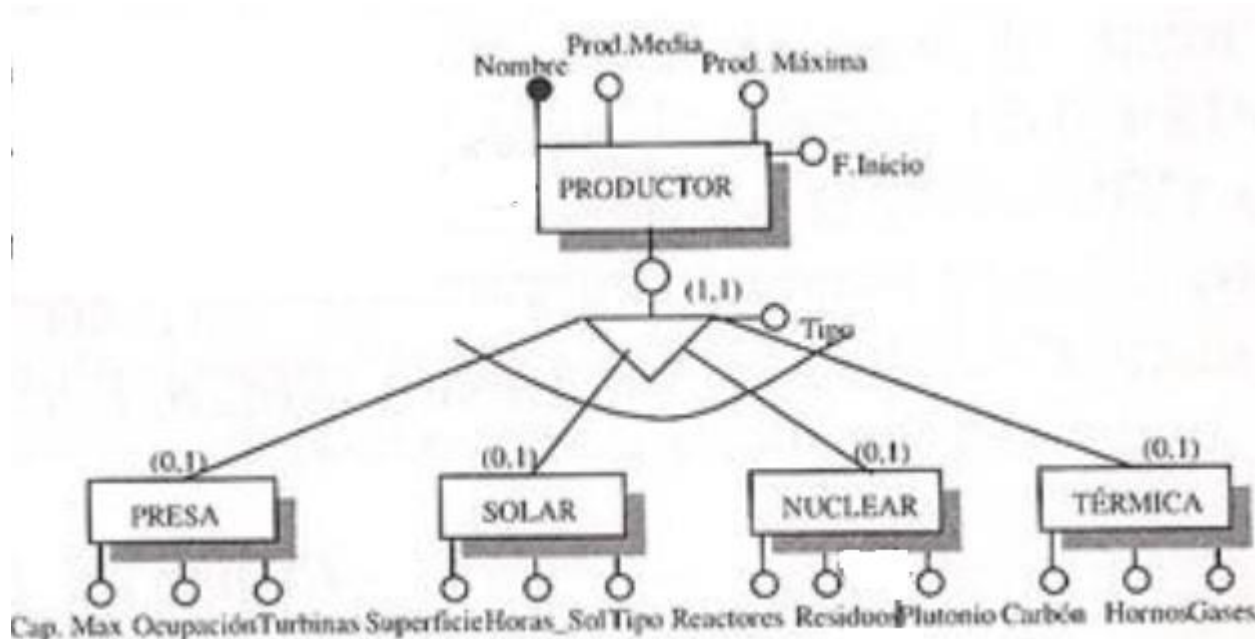
MR





TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Nota: el campo tipo de la tabla productor no puede ser nulo ya que la jerarquía es total



PRODUCTOR (nombre, prod_med, prod_max, finicio, tipo)

PRESA(nombre, cap_max, ocupacion, turbinas)

SOLAR(nombre, superficie, horas_sol, tipo)

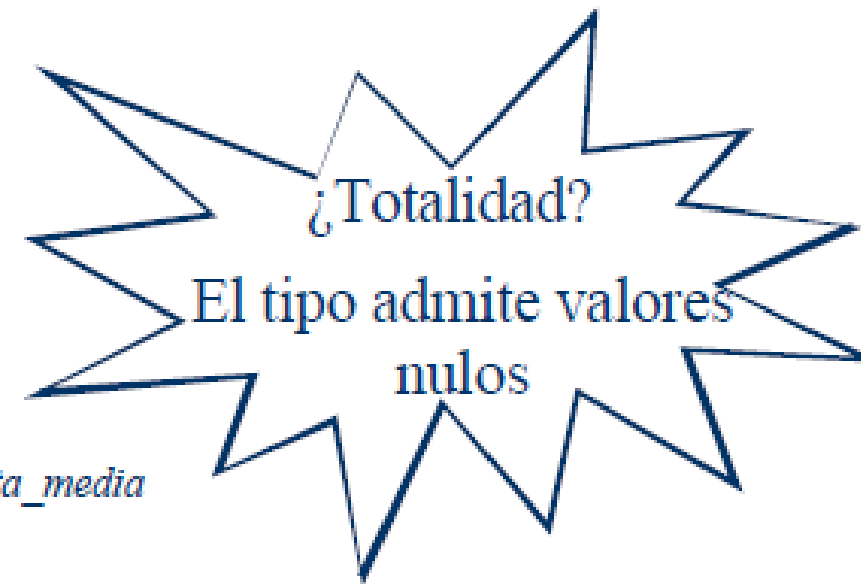
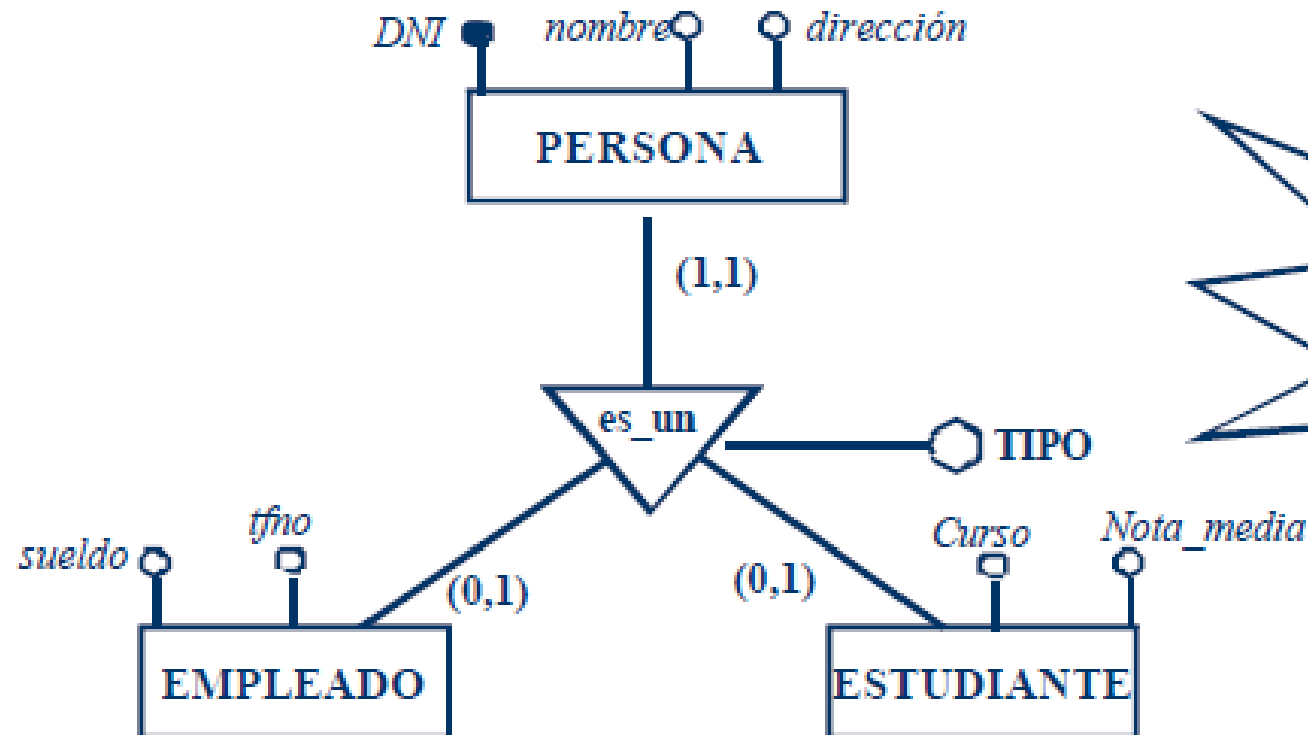
NUCLEAR(nombre, reactores, residuos, plutonio)

TERMICA(nombre, carbon, hornos, gases)



TRANSFORMACIÓN DEL DIAGRAMA E-R AL ESQUEMA RELACIONAL

Ejemplo Jerarquía Parcial con solapamiento



ME/R

MR



NOTACIONES PARA LA REPRESENTACIÓN DE ESQUEMAS RELACIONALES

a) Utilizaremos la siguiente notación:

- # delante de los atributos que formen la clave primaria.
- Clave ajena **subrayada** y escribiremos debajo el nombre de la tabla a la que se refiere.
- Si un atributo es clave alternativa se utilizará subrayado discontinuo.
- Si tiene valor único o no admite valores nulos : VNN {Atr1,..., Atrn}

b) De forma alternativa:

- TABLA_X (AX1, AX2, ..., AXm): nombre de la tabla y atributos
- [PK | CP] {AX1, ..., AXn}: clave primaria
- [FK | CAj] {AXo, ..., AXp} Ref a TABLA_Y: claves ajenas
- [AK | CAIt] {AXq, ..., Ar} claves alternativas.
- VNN{AXs,..., AXt} no admiten valores nulos.
- UNIQUE{AXu,..., AXz} no admiten valores repetidos.