



BASES DE DATOS (IG18 Semipresencial)

Diseño Lógico de Bases de Datos Relacionales.

Lledó Museros / Ismael Sanz
museros@icc.uji.es / isanz@icc.uji.es



1. Introducción
2. Metodología de diseño lógico en el modelo relacional
3. Normalización



➤ Fases en el Diseño de BBDD

Especificación de requisitos

DISEÑO CONCEPTUAL

Esquema conceptual

DISEÑO LÓGICO

← Normalización

Esquema lógico

DISEÑO FÍSICO

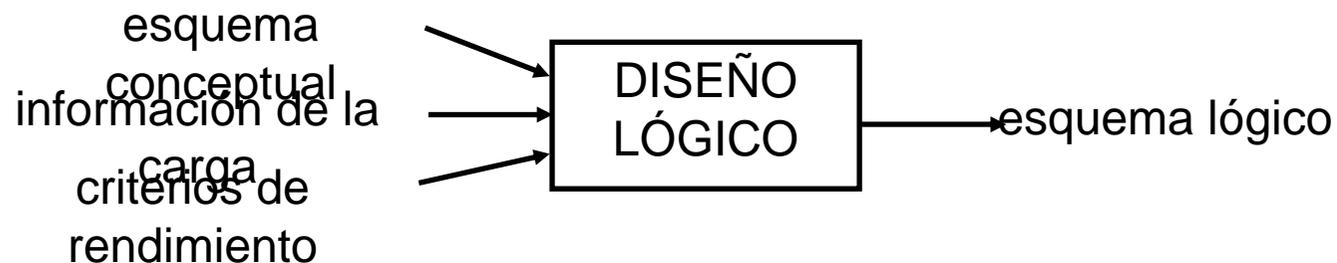
Esquema físico



- Obviamente, si se utiliza el modelo relacional, el objetivo será obtener un **esquema lógico relacional**
- Dicho esquema estará formado por:
 - Un conjunto de relaciones o tablas (en tercera forma normal).
 - Claves primarias para cada tabla
 - Claves ajenas de las distintas tablas y sus reglas de borrado y modificación
 - Reglas de integridad necesarias



- **Diseño lógico:** conversión del esquema conceptual de datos en un esquema lógico.
- **Objetivo:** obtener una representación que use de la manera más eficiente posible los recursos para la estructuración de datos y el modelado de restricciones disponibles en el modelo lógico.



- **Información de la carga**
 - Volumen de la BBDD
 - Conocimiento de consultas y transacciones a realizar, y su frecuencia.
- **Criterios de Rendimiento**
 - Tiempo de respuesta medio o máximo
 - Espacio de almacenamiento ocupado por la BBDD
 - Utilización de CPU o tiempo de E/S



1. Introducción
2. Metodología de diseño lógico en el modelo relacional
3. Normalización

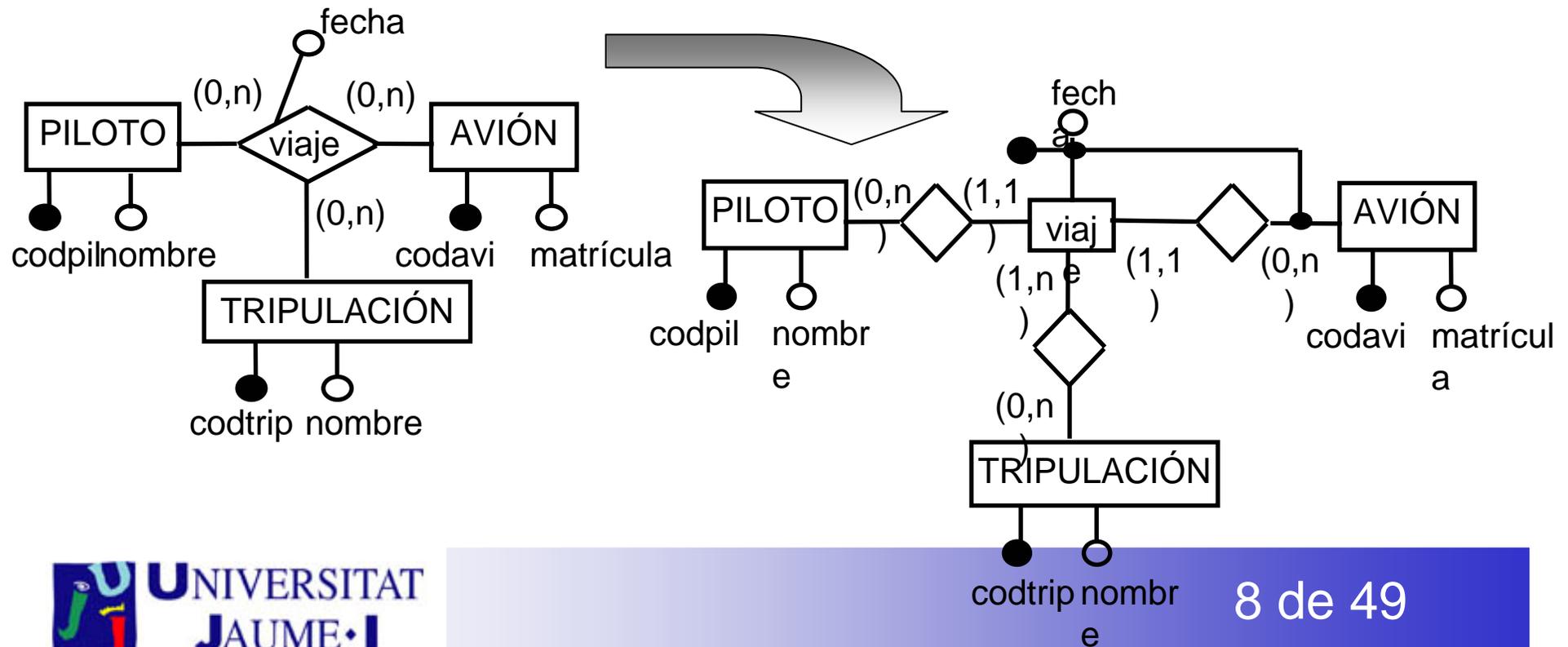


1. Refinar el esquema conceptual
2. Derivar un conjunto de relaciones (tablas)
3. Validar cada esquema mediante la normalización



1. Refinar el esquema conceptual

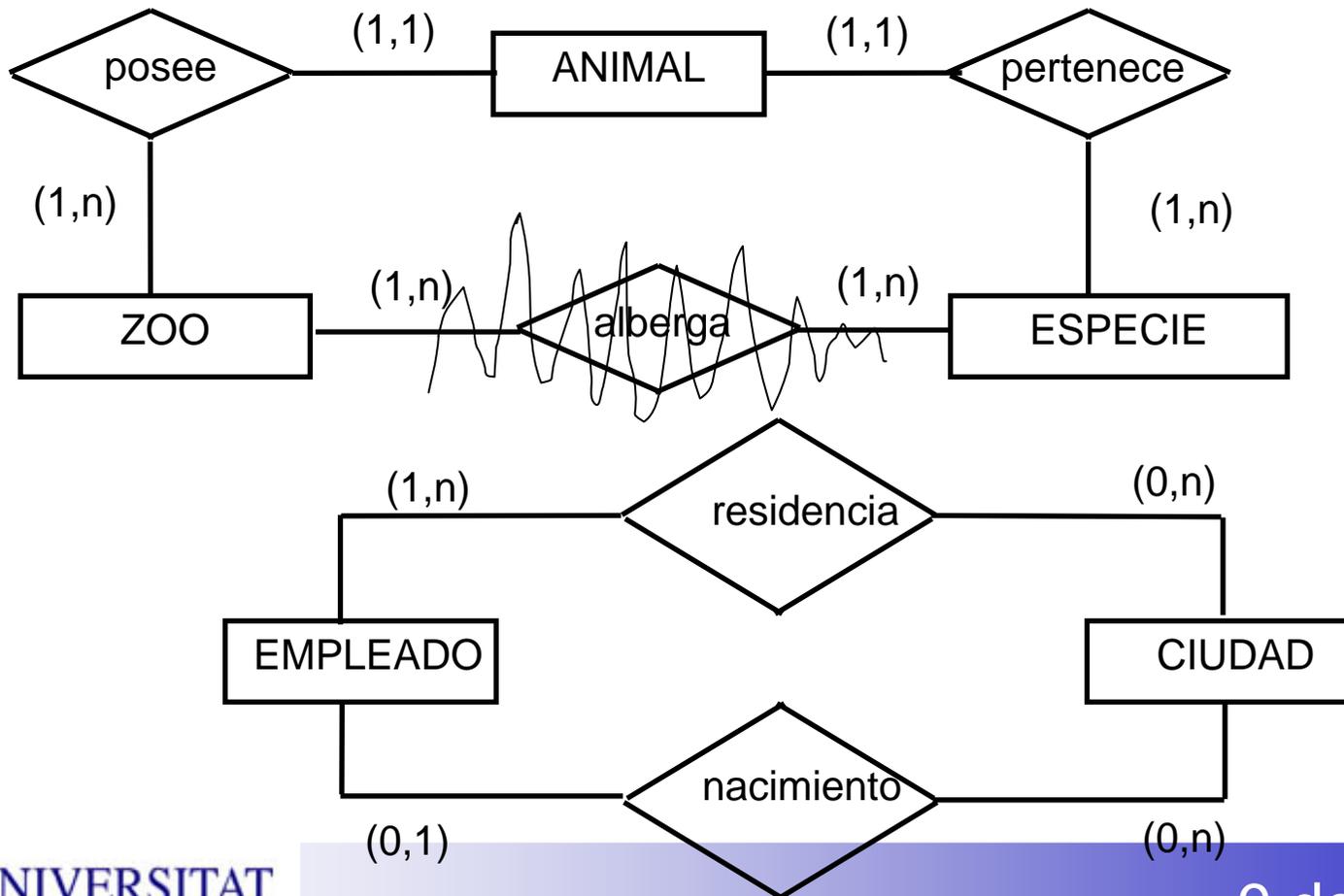
- a) Sustituir cada relación entre tres o más entidades por una entidad intermedia. La cardinalidad de las nuevas relaciones binarias dependerá de su significado. Si la relación sustituida tiene atributos, éstos serán los atributos de la nueva entidad.





1. Refinar el esquema conceptual

b) Eliminar las relaciones redundantes



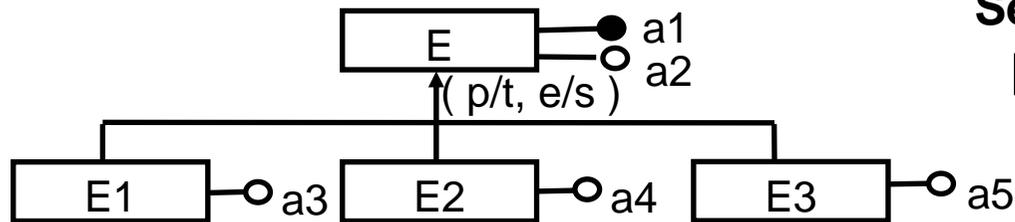


1. Refinar el esquema conceptual

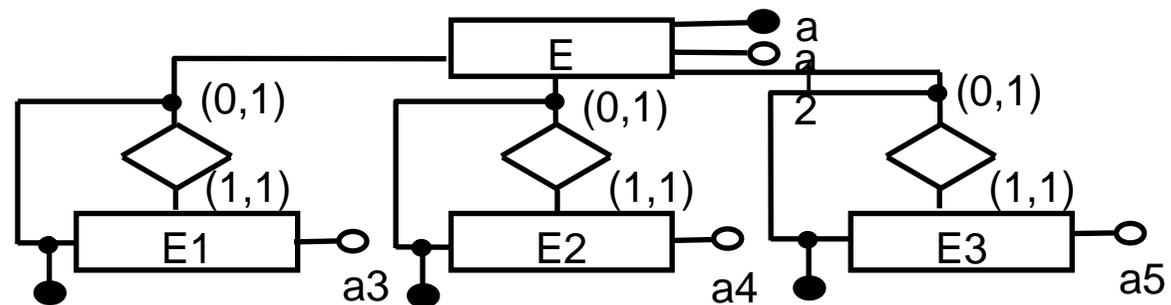
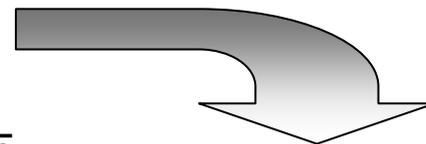
c) Transformar las clasificaciones

Ejemplo:

Clasificación Original



Se puede transformar en...



Más adelante se verá con más detalle todas las posibles transformaciones.



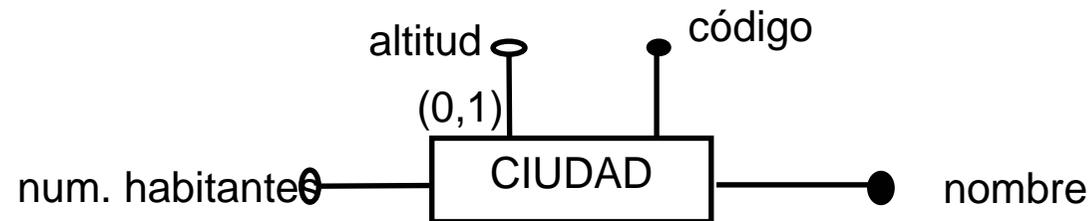
2. Derivar un conjunto de relaciones (tablas)

- En el modelo relacional sólo se pueden representar tablas.
- Todos los elementos del esquema conceptual se deben transformar en tablas:
 - Entidades
 - Clasificaciones (jerarquías de generalización)
 - Relaciones: de uno a uno, de uno a muchos y de muchos a muchos



2. Derivar un conjunto de relaciones: **Entidades**

- Cada **entidad** del esquema conceptual se transforma en una relación base (tabla).
- Los **atributos** de la entidad se convierten en los atributos de la tabla.
- De entre los **identificadores** de la entidad se debe escoger uno como clave primaria de la tabla.



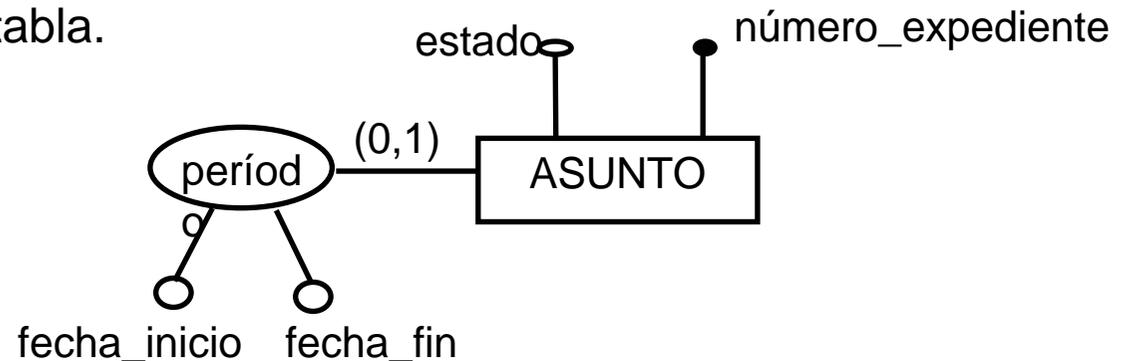
CIUDAD(nombre, codigo, num_habitantes, altitud)

Codigo es clave alternativa y altitud acepta nulos



2. Derivar un conjunto de relaciones: Entidades

- Cada componente de un atributo compuesto se convierte en un atributo de la tabla.



ASUNTO(numero expediente, estado, fecha_inicio, fecha_fin)

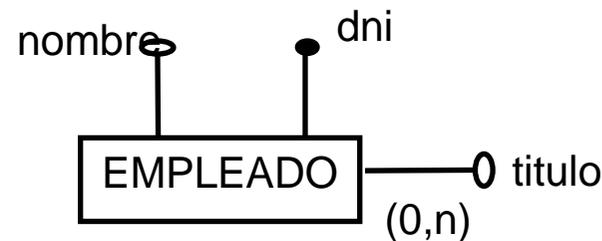
fecha_fin acepta nulos

También es posible transformar el atributo compuesto en uno solo que contenga todos los atributos simples (por ejemplo el atributo compuesto *dirección* formado por *calle*, *número* y *piso* se puede transformar en uno sólo (de tipo cadena de caracteres) que incluya a los tres.



2. Derivar un conjunto de relaciones: **Entidades**

- Por cada atributo de cardinalidad máxima mayor que uno se incluye una tabla dentro de la tabla, como un atributo más.

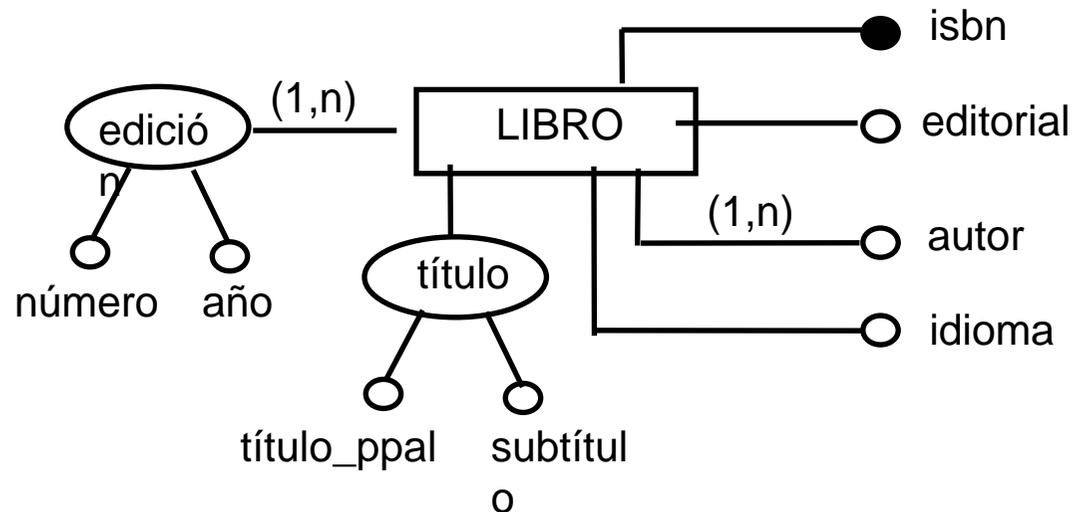


EMPLEADO(dni, nombre, TITULO(titulo))

¿titulo acepta NULOS?



2. Derivar un conjunto de relaciones: Entidades



LIBRO(isbn, editorial, **AUTOR**(autor), idioma, título_ppal, subtítulo, **EDICION**(número, año))



2. Derivar un conjunto de relaciones: Entidades

isbn	editorial	AUTOR	idioma	titulo_ppal	subtitulo	EDICIÓN	
1234543	Ediciones DFX	autor	castellano	las BBDD relacionales	un enfoque práctico	número	año
		Juan H. Gema P. Sara M.				1	1995
						2	2002
11223CDA	Trumpets	autor	castellano	las BBDD objeto-relacionales y orientadas a objetos	Aprenda en unos días	número	año
		Ferran P.				1	1999
						2	2000
11223CDA	Casd and Fert	autor	inglés	DataBase Systems	Relational model	número	año
		Mike G.				1	2001
1412SCD-X	Casd and Fert	autor	inglés	Computer Systems	Hardware and Software	número	año
		Mike R. Rose k.				1	1991
						2	1993
						3	1999
						4	2003



2. Derivar un conjunto de relaciones:

Clasificaciones (jerarquías de generalización)

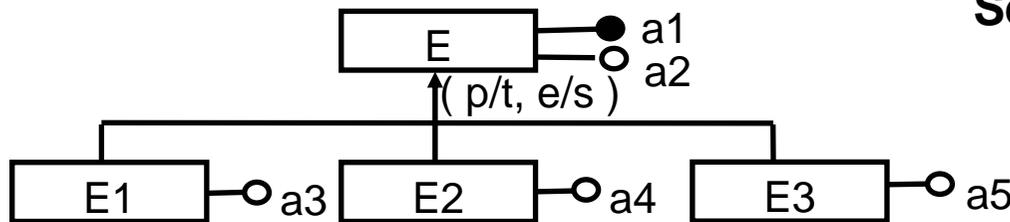
- El modelo relacional no puede representar clasificaciones
- Por ello hay que modificar el esquema entidad/relación
- Existen 3 posibles alternativas:
 1. No eliminar ninguna de las entidades, transformando las subentidades en entidades que se relacionan con la entidad padre
 2. Eliminación de la entidad genérica, las subentidades heredan sus características
 3. Integración de las subentidades en la entidad padre



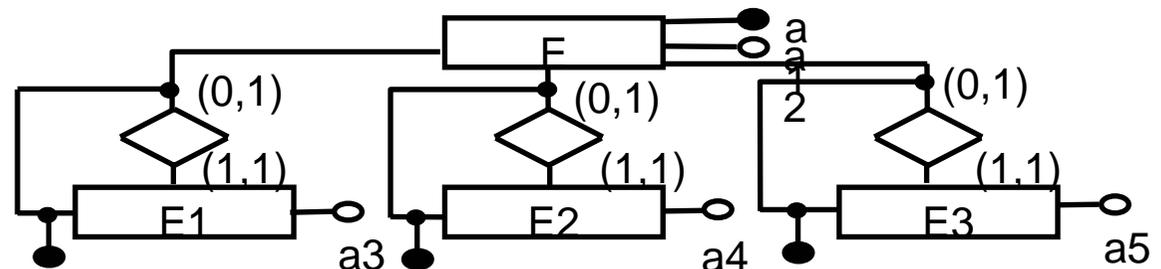
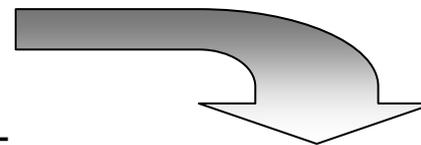
2. Clasificaciones (jerarquías de generalización)

1. Transformar las subentidades en entidades que se relacionan con la entidad padre: Se obtiene una tabla por cada entidad y subentidad. Sirve para cualquier tipo de jerarquía (t/p, e/s).

Clasificación Original



Se puede transformar en...



Cuyo modelo Lógico es:

$E(\underline{a1}, a2)$

$E1(\underline{a1}, a3)$

$E2(\underline{a1}, a4)$

$E3(\underline{a1}, a5)$

NULOS

BORRADO MODIFICACION

$E1.a1, E2.a1, E3.a1$ son claves ajenas a E NO

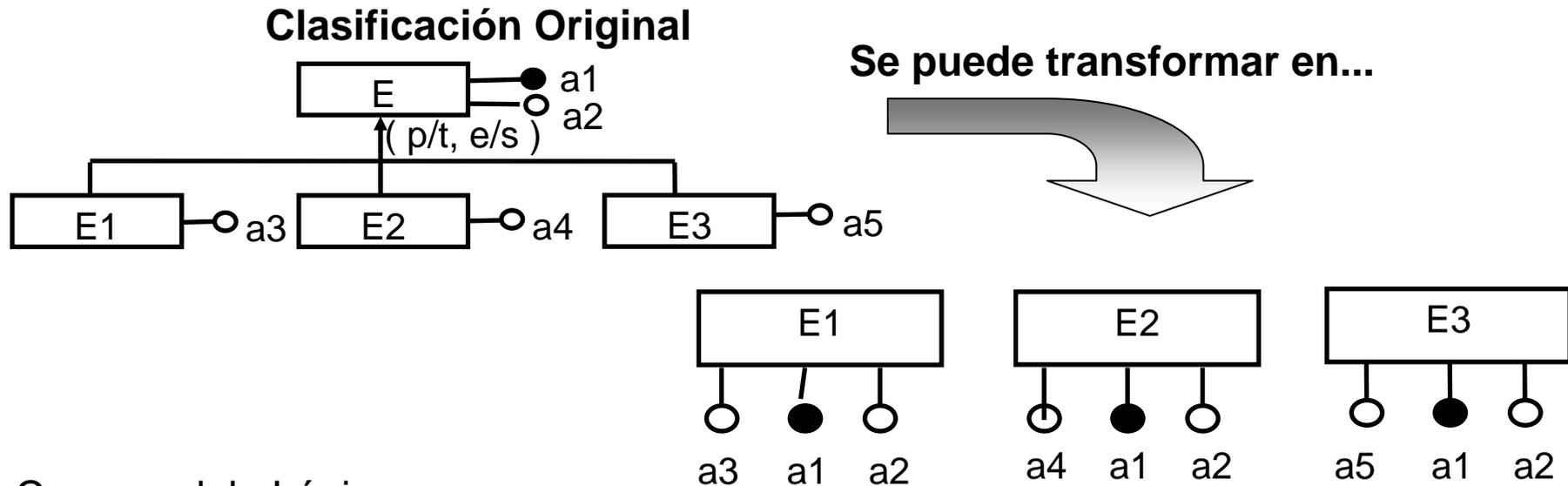
Prop.

Prop.



2. Clasificaciones (jerarquías de generalización)

2. Eliminar la entidad genérica: Se obtiene una tabla por cada subentidad. Sirve para cualquier tipo de jerarquía (t/p, e/s).



Cuyo modelo Lógico es:

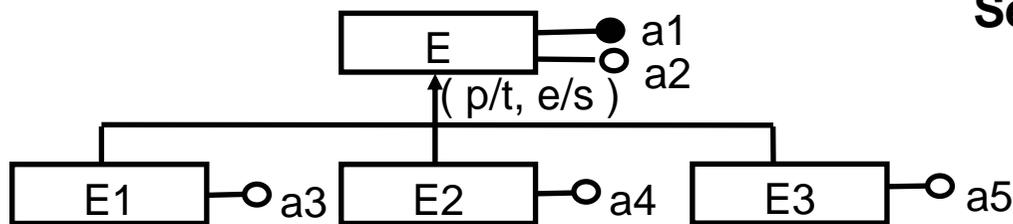
$E1(\underline{a1}, a2, a3)$ $E2(\underline{a1}, a2, a4)$ $E3(\underline{a1}, a2, a5)$



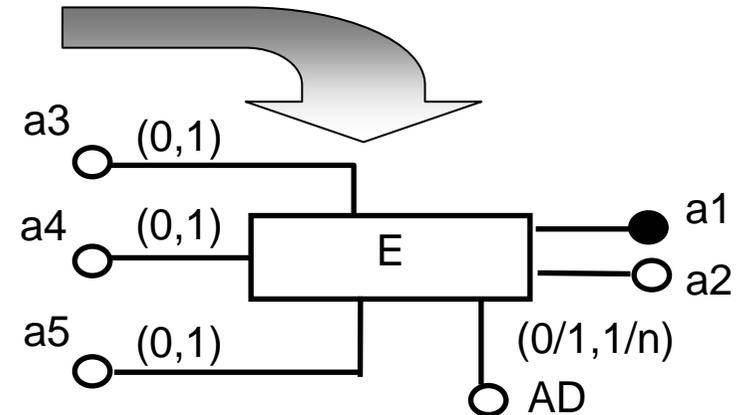
2. Clasificaciones (jerarquías de generalización)

- Integrar las subentidades en la entidad padre: Se obtiene sólo una tabla. Sólo sirve para jerarquías totales y exclusivas.

Clasificación Original



Se puede transformar en...



Cuyo modelo Lógico es:

$E(\underline{a1}, a2, a3, a4, a5, \text{tipo})$ si es exclusiva

$a3, a4, a5$ aceptan nulos, tipo acepta nulos si es parcial

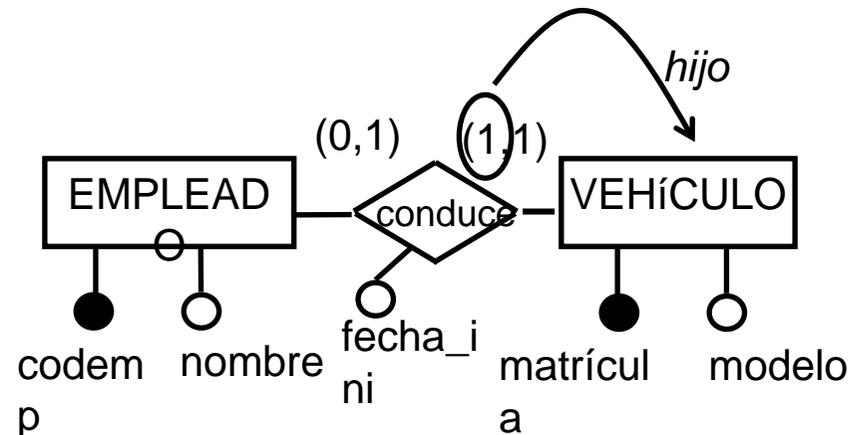
$E(\underline{a1}, a2, a3, a4, a5, AD(\text{tipo}))$ si es superpuesta

$a3, a4, a5$ aceptan nulos



2. Derivar un conjunto de relaciones: **Relaciones binarias (1:1)**

- Por cada relación binaria (1:1) incluir la clave primaria de la tabla correspondiente a la entidad padre en la tabla de la entidad hijo como una clave ajena. ¿Y los atributos de la relación?
- **Ejemplo 1:** Una empresa desea almacenar información sobre sus empleados y sobre los coches que conducen. Sólo algunos empleados conducen coches. Todos los coches están asignados a un (y sólo un) empleado.





2. Derivar un conjunto de relaciones: **Relaciones binarias (1:1). Ejercicio:**

A partir del esquema conceptual anterior, se han tomado dos opciones para modelarlo:

- El empleado E1 trabaja en las oficinas, se llama Juan.
- El empleado E2 es taxista, se llama Juan y conduce un renault de matricula HAB 2398 desde enero 2003
- El empleado E3 es taxista, se llama Ana y conduce un ford con matricula KLM 2343 desde Julio de 2003.
- El empleado E0 se llama Rosa y es la gerente

Opción 1

EMPLEADO

<u>codemp</u>	nombre	matricula	fecha_ini

VEHICULO

<u>matricula</u>	modelo

Opción 2

VEHICULO

<u>matricula</u>	modelo	codemp	fecha_ini

EMPLEADO

<u>codemp</u>	nombre

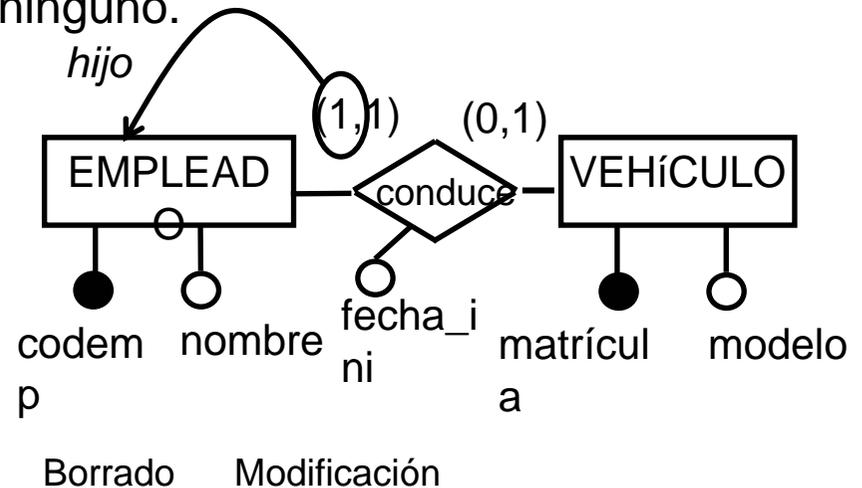
Rellena las tablas de la opción 1 y 2. ¿Cuál es mejor? ¿Encuentras algún problema?

Escribe una sentencia **SELECT** que obtenga los datos de las personas que conducen algún coche.



2. Derivar un conjunto de relaciones: **Relaciones binarias (1:1)**

- **Ejemplo 2:** Una empresa desea almacenar información sobre sus empleados y sobre los coches que conducen. Todos los empleados conducen coches. Los coches pueden estar asignados a un (y sólo un) empleado o no estar asignados a ninguno.



VEHICULO(matricula, modelo)

EMPLEADO(codemp, nombre, matricula, fecha_ini)

matricula

Nulos

Borrado

Modificación

EMPLEADO → VEHICULO

matricula es clave alternativa ¿fecha_ini acepta nulos?



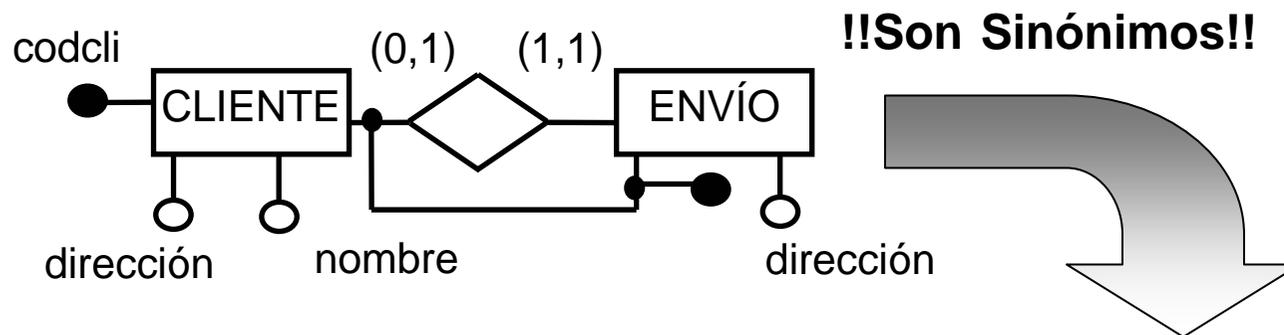
2. Relaciones binarias (1:1)

- Una pequeña diferencia entre los dos ejemplos da lugar a dos esquemas lógicos con diferencias notables.
 - ¿Y si las dos entidades participan con cardinalidad (0,1)?
 - ¿Y si son ambas (1,1)?
 - Otra posibilidad es sustituir la interrelación por una tabla que hereda las claves primarias de las dos entidades. La clave primaria de la nueva tabla dependerá del significado de los datos y de las cardinalidades de las entidades.
-
- **Ejercicio:** Reescribe los cuatro ejemplos utilizando una tabla para cada entidad y una tercera para la interrelación.



2. Relaciones binarias (1:1)

- Los **identificadores externos** pueden tratarse de dos formas:
- Juntar las dos entidades en una sólo
 - Crear dos entidades añadiéndole a la que tiene el identificador externo el identificador de la otra entidad (que será una clave ajena a ésta)
- **OJO:** Si las dos entidades son sinónimos, integrarlas en una sola tabla.



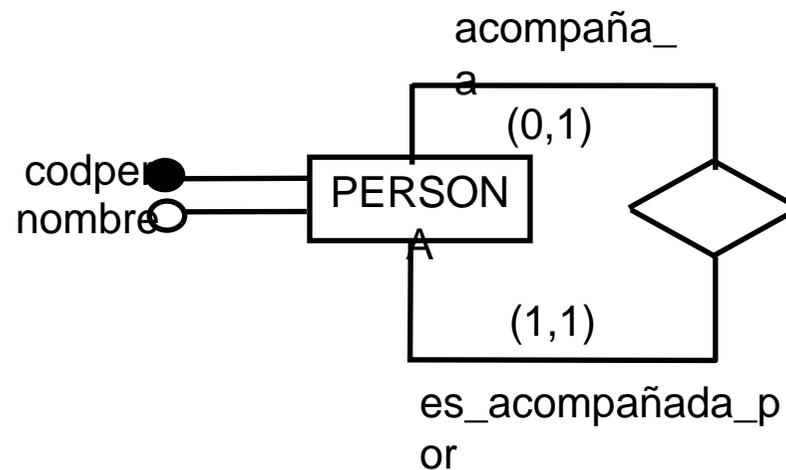
CLIENTE(codcli, dirección, nombre, dirección_envío)

ENVÍO es un **entidad débil** porque no tiene atributos que le sirvan como identificador.



2. Relaciones binarias (1:1)

- Las relaciones de una entidad consigo misma se tratan del mismo modo que las binarias:



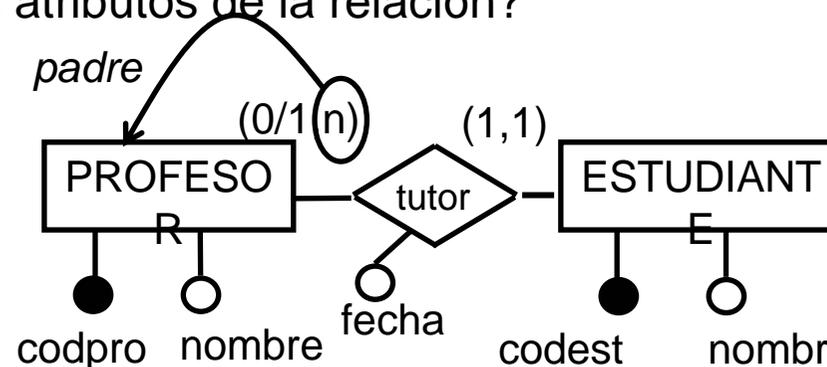
- **Ejercicio:** Escribe cuál sería el esquema relacional



2. Derivar un conjunto de relaciones: **Relaciones binarias (1:n)**

- Por cada relación binaria (1:n) incluir la clave primaria de la tabla correspondiente a la entidad padre en la tabla de la entidad hijo (será una clave ajena). ¿Y los atributos de la relación?

- **Ejemplo 1:**



PROFESOR(codpro, nombre)

ESTUDIANTE(codest, nombre, codpro, fecha)

codpro

NULOS Borrado Modificación

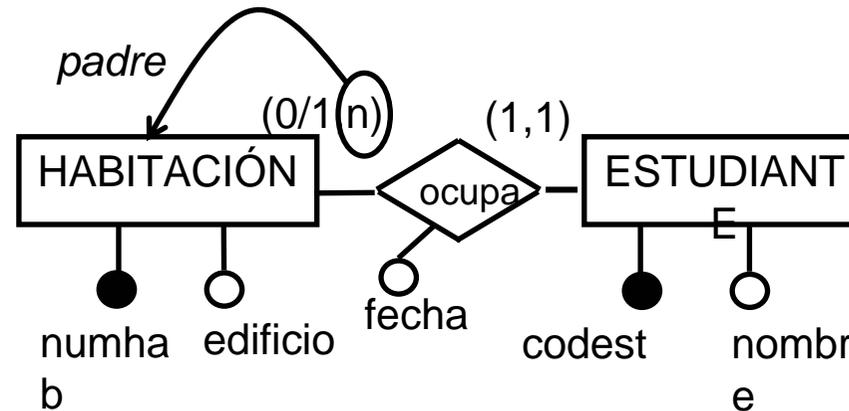
ESTUDIANTE → PROFESOR

¿fecha acepta nulos?



2. Relaciones binarias (1:n)

➤ Ejemplo 2:



HABITACIÓN(numhab, edificio)

ESTUDIANTE(codest, nombre, numhab, fecha)

codpro

NULOS Borrado Modificación

ESTUDIANTE → HABITACIÓN

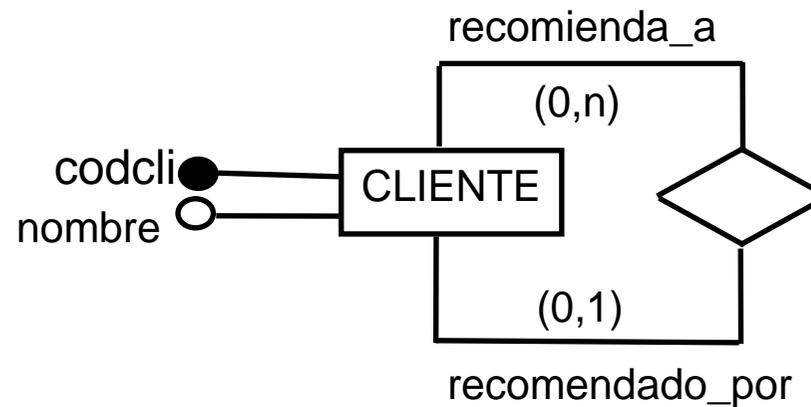
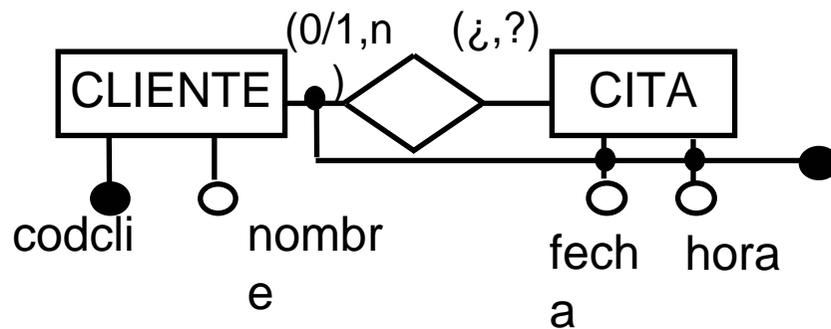
¿fecha acepta nulos?

¿Y si hay muy pocos estudiantes que viven en una habitación del campus?



2. Relaciones binarias (1:n)

- **Ejercicio:** ¿cuál es la cardinalidad de la entidad CITA?
- ¿cómo transformarías los dos ejemplos?





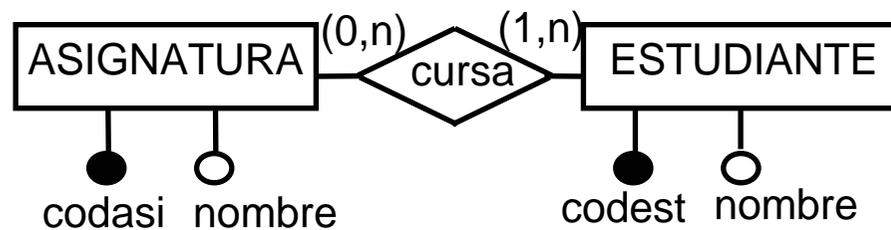
2. Derivar un conjunto de relaciones: **Relaciones binarias (m:n)**

- Por cada relación binaria (m:n) incluir una nueva tabla con una clave ajena a cada una de las tablas correspondientes a las entidades participantes.
- La clave primaria de la nueva tabla dependerá del significado de los datos.
- ¿Y los atributos de la relación? Se añaden a la tabla correspondiente a la relación



2. Relaciones binarias (m:n)

➤ Ejemplo 1



ASIGNATURA(codasi, nombre)

ESTUDIANTE(codest, nombre)

CURSA(codest, codasi)

	codest		
CURSA	—————>	ESTUDIANTE	

Nulos Borrado Modificación

	codasi		
CURSA	—————>	ASIGNATURA	

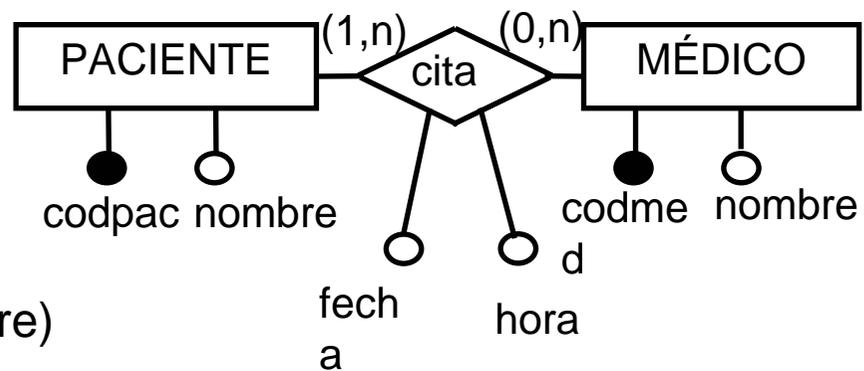
Nulos Borrado Modificación

¿Cuál es la clave primaria de CURSA?



2. Relaciones binarias (m:n)

➤ Ejemplo 2



PACIENTE(codpac, nombre)

MÉDICO(codmed, nombre)

CITA(codmed, fecha, hora, codpac)

CITA $\xrightarrow{\text{codest}}$ PACIENTE

Nulos Borrado Modificación

CITA $\xrightarrow{\text{codasi}}$ MÉDICO

Nulos Borrado Modificación

¿Cuál es la clave primaria de CITA?



Resumen de la correspondencia entre esquemas para las relaciones

	<i>Relación 1:1</i>	<i>Relación 1:n</i>	<i>Relación n:m</i>
<i>Integrar las dos tablas correspondientes a cada una de las entidades participantes en la relación binaria, en una sola tabla.</i>	Es lo más aconsejable cuando ambas entidades tienen <u>el mismo identificador</u> . Los atributos de la relación binaria también estarán en la tabla. OJO: es posible que algunos atributos deban aceptar nulos.	Para este tipo de relaciones binarias no se puede escoger esta opción.	Para este tipo de relaciones binarias no se puede escoger esta opción.
<i>Poner una clave ajena en la tabla correspondiente a una de las entidades participantes en la relación binaria.</i>	La clave ajena <u>se puede poner</u> en cualquiera de las tablas. La tabla que recibe la clave ajena también recibe los atributos de la relación binaria. OJO: es posible que algunos atributos deban aceptar nulos.	La clave ajena <u>se debe poner</u> en la tabla correspondiente a la entidad que participa en la relación binaria con <u>cardinalidad máxima 1</u> . Los atributos de la relación binaria se ponen como atributos en la tabla que recibe la clave ajena. OJO: es posible que algunos atributos deban aceptar nulos.	Para este tipo de relaciones binarias no se puede escoger esta opción.
<i>Añadir al esquema una nueva tabla en la que se refleje la relación binaria.</i>	Es lo más aconsejable cuando ambas entidades participan en la relación de forma opcional y hay pocas ocurrencias de la misma. Esta nueva tabla tiene una clave ajena a cada una de las dos tablas y también los atributos de la relación binaria.	La nueva tabla tiene una clave ajena a cada una de las dos tablas y también los atributos de la relación binaria. La clave primaria de la nueva tabla será la clave ajena que hace referencia a la tabla de la entidad que participa en la relación binaria con <u>cardinalidad máxima 1</u> .	Esta nueva tabla tiene una clave ajena a cada una de las dos tablas y también los <u>atributos</u> de la relación binaria. La clave primaria variará según el significado de la relación binaria (hay que "meditarla").



3. Validar cada esquema mediante la normalización

- La teoría de la normalización, desarrollada por Codd en 1972, permite mejorar el diseño lógico de una Base de datos relacional. Aunque es una técnica para diseñar bases de datos relacionales, no se utiliza como estrategia de diseño de base de datos sino para verificar esquemas relacionales.
- Se fundamenta en las Formas Normales, que son un conjunto de restricciones que deben cumplir las relaciones.
- Ventajas:
 - Evita anomalías en inserciones, modificaciones y borrados.
 - Mejora la independencia de datos.



1. Introducción
2. Metodología de diseño lógico en el modelo relacional
3. Normalización



3. Normalización. Ejemplo

- Se desea almacenar información sobre las facturas que realizan los clientes de una empresa así como los artículos que compran:

```
FACTURAS (codfac, fecha, iva, dto, codcli, nombre, población)
```

```
LINEAS (codfac, codart, cant, dto, precio, descrip, precio_almacen)
```

	codfac		Nulos	Borrado	Modif.
LINEAS	----->	FACTURAS	No	P	P



3. Normalización. Ejemplo

Facturas

codfac	fecha	iva	dto	codcli	nombre	población

Lineas

codfac	codart	cant	dto	precio	descrip	precio almacen



- **Ejemplo. Escribe en las dos tablas los siguientes datos:**
 - La factura 3125 es del 23/09/2003, con un 16% de IVA y del cliente 355 cuyo nombre es Juan Diaz de Castellón. Se vendieron 3 tuercas de doble paso (cod. 345T) por un precio de 2.25 Euros y 2 tornillos de rosca reforzada (cod. T554) por 1 Euro.
 - La factura 3126 es del 12/12/2003, con un 16% de IVA y del cliente 354 cuyo nombre es Rosa Fernández de Almazora. Se vendieron 3 tornillos de rosca reforzada (cod. T554) por 1.2 Euros.
 - La factura 3127 es del 15/12/2003, con un 16% de IVA y del cliente 355 cuyo nombre es Juan Diaz de Castellón. Se vendieron 2 tornillos de rosca reforzada (cod. T554) por 1.1 Euros.
 - Se ha comprado un nuevo artículo en el almacén con código 445A, arandelas cuadradas con un precio de 0.87 Euros.
 - El cliente 355 devuelve las tres tuercas y solicita que se rectifique la factura 3125.
 - El cliente 355 ha cambiado su domicilio a Villareal.
 - El cliente 354 realiza la devolución de la factura 3126.
 - Se ha hecho un nuevo cliente, Ferran Sabater de Castellón con código 337.



➤ Ejemplo:

- ¡¡ Existen dependencias entre los datos, **no sólo de la clave primaria!!**
- Indica cuales són esas dependencias:

```
FACTURAS (codfac, fecha, iva, dto, codcli, nombre, población)

LINEAS (codfac, codart, cant, dto, precio, descrip, precio_almacen)

          codfac                Nulos  Borrado  Modif.
LINEAS  ----->  FACTURAS    No      P      P
```

- Genera **¡Anomalías en las actualizaciones de datos!**



➤ Ejemplo:

CLIENTES(codcli, nombre, población)

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

	codcli		Nulos	Borrado	Modif.
FACTURAS	----->	CLIENTES	Sí	A	P

ARTICULOS(codart, descrip, precio_almacen)

LINEAS(codfac,codart, cant, dto, precio)

	codfac		Nulos	Borrado	Modif.
LINEAS	----->	FACTURAS	No	P	P
	codart		Nulos	Borrado	Modif.
LINEAS	----->	ARTICULOS	No	R	P



➤ Dependencia funcional

- Dada una relación (tabla) R, el atributo Y de R depende funcionalmente del atributo X de R

R.X → R.Y

si X determina el valor de Y, es decir, un valor Y en R está asociado a cada valor X en R. Tanto X como Y puede ser atributos compuestos.

Ejemplo:

```
CLIENTE (codcli, nombre, codpostal, población)

codpostal -----> población
```



- **Dependencia funcional. Observaciones:**
 - Si el atributo X es una clave primaria (o alternativa) de R , entonces todos los atributos Y de la relación dependen funcionalmente de X , por la definición de clave primaria (o alternativa)
 - La dependencia funcional es una noción semántica (depende del significado de los datos)
 - Cada dependencia funcional es una clase especial de regla de integridad
 - Cada dependencia funcional representa una relación uno a muchos.



➤ Primera Forma Normal (1FN):

- Una relación está en 1FN si, y sólo si, todos sus dominios contienen valores atómicos.

PRODUCTO

<u>codprod</u>	nombre	VERSIÓN		
		número	fecha	ventas
LH4	Ladrillo hueco	1	1/3/1996	30.000
		2	1/8/1998	50.000
		3	1/2/2000	13.000
LP7	Ladrillo perforado	1	1/6/1996	70.000
		2	1/12/2000	

grupos repetitivos
(valores no atómicos)

¡NO ESTA EN 1FN!

Hay atributos que no son atómicos



➤ Primera Forma Normal (1FN):

```
PRODUCTO (codprod, nombre, VERSIÓN (número, fecha, ventas))
```

Se extrae de la tabla anidada añadiéndole la clave primaria a la tabla principal. La clave primaria de la nueva tabla será la unión de la que tenía más la de la tabla principal.

```
PRODUCTO (codprod, nombre, descripción)
VERSIÓN (codprod, número, fecha, ventas)

          codprod          Nulos Borrado Modificación
VERSIÓN -----> PRODUCTO
```

¡OJO! La nueva tabla hereda parte de la clave primaria

Ejercicio: Rellena los huecos que faltan en la clave ajena.



➤ Segunda Forma Normal (2FN):

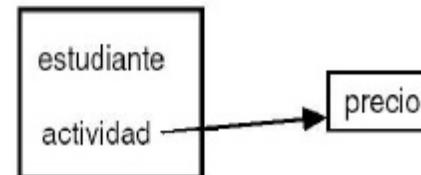
Una relación esta en 2FN si, y sólo si, está en 1FN y, además, cada atributo no clave depende completamente de a clave primaria (no depende de algún subconjunto)

estudiante	actividad	precio
100	Tenis	1500
100	Yoga	1500
200	Tenis	1500
300	Escalada	5000

misma actividad, mismo precio.

¡No está en 2FN!

actividad → precio





➤ Segunda Forma Normal (2FN):

```
INSCRIPCIÓN(estudiante, actividad, precio)
```

Se extraen los atributos dependientes en otra tabla. En la primera tabla se deja el atributo del que dependían los otros que será clave ajena a la nueva tabla.

```
INSCRIPCIÓN(estudiante, actividad)
```

```
ACTIVIDAD(actividad, precio)
```

```
                actividad                Nulos  Borrado  Modificación  
INSCRIPCIÓN  ----->  ACTIVIDAD
```

Ejercicio: Rellena los huecos que faltan en la clave ajena



➤ Tercera Forma Normal (3FN):

Una relación está en 3FN si, y sólo si, está en 2FN y, además, cada atributo no clave no depende transitivamente de la clave primaria.

inquilino	edificio	alquiler
100	E04	50.000
200	E13	50.000
300	E09	65.000
400	E04	50.000

mismo edificio, mismo alquiler.

¡No está en 3FN!





➤ Tercera Forma Normal (3FN):

```
INQUILINO (inquilino, edificio, alquiler)
```

Se extraen los atributos dependientes en otra tabla. En la primera tabla se deja el atributo del que dependían otros que será clave ajena a la nueva tabla

```
INQUILINO(inquilino, edificio)
```

```
EDIFICIO(edificio, alquiler)
```

```
                edificio                Nulos  Borrado  Modificación  
INQUILINO  ----->  EDIFICIO
```

Ejercicio: Rellena los huecos que faltan en la clave ajena



BASES DE DATOS (IG18 Semipresencial)

Diseño Lógico de Bases de Datos Relacionales

¿DUDAS?

Lledó Museros / Ismael Sanz
museros@icc.uji.es / isanz@icc.uji.es