

Bases de Datos

**Traducción de Diagramas
Entidad-Relación a esquemas
relacionales**



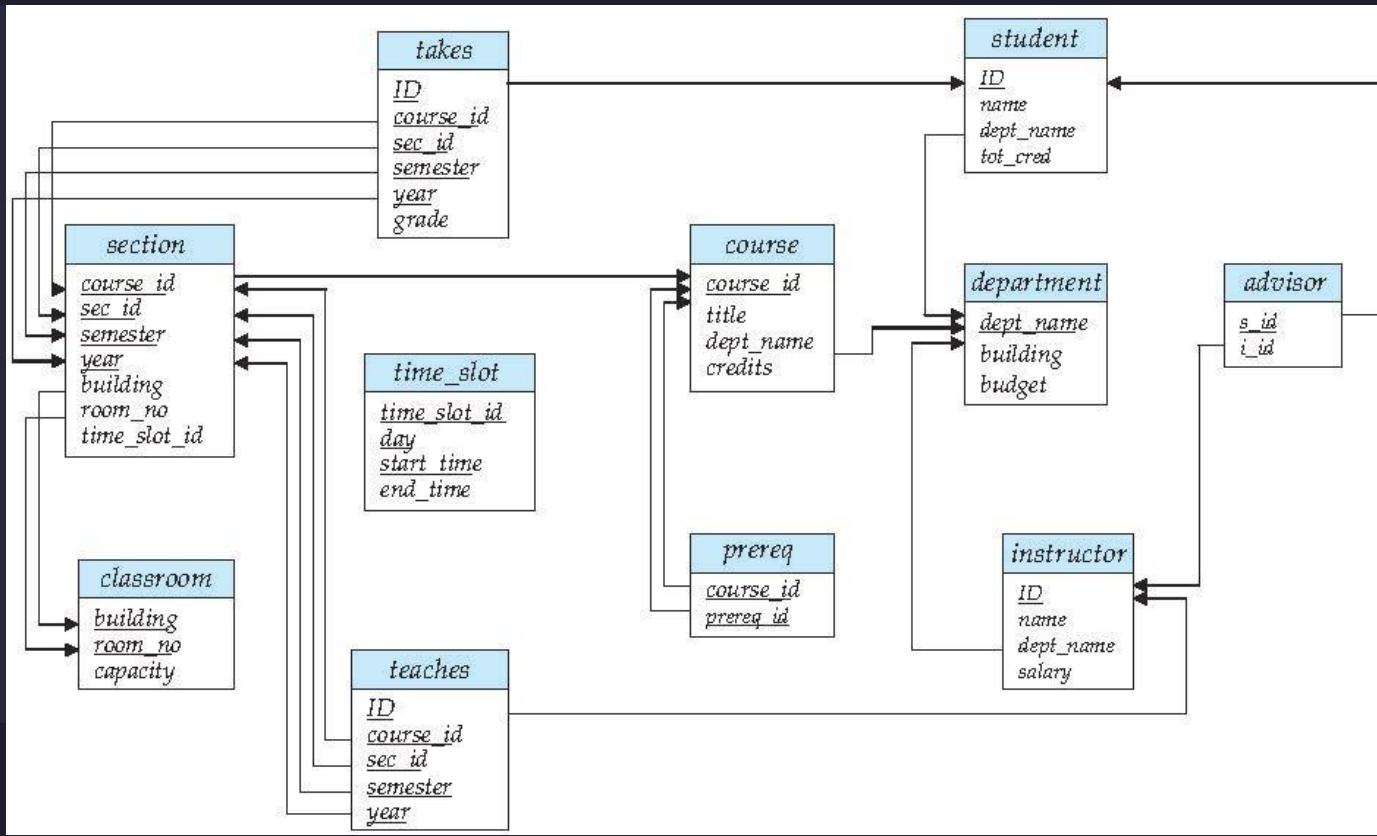
De modelo Entidad-Relación a modelo relacional

modelo relacional: tablas

esquema de relación en DER → tabla de BD

1. cómo son los esquemas de modelo relacional
2. restricciones de integridad

Diagrama para BD universitaria



Diseño de una BD relacional

Un buen DER :D

Almacenar toda la información en una sola relación resulta en redundancia

Ej.: dos estudiantes con el mismo instructor

Necesidad de valores nulos

Ej.: representar un estudiante sin supervisor

Notación de tablas

Notación del modelo relacional

Ej.: En la Universidad tenemos instructores y cada uno tiene identificador, nombre, nombre de departamento y salario....

A_1, A_2, \dots, A_n son atributos.

$R = (A_1, A_2, \dots, A_n)$ es un **esquema** de relación

Ej.: Instructor = (ID, name, dept_name, salary)

Dominio de los atributos

conjunto de valores permitidos para cada atributo (rango, o tipo)
tienen que ser atómicos
(para poder hacer consultas)

El valor *null*

El valor especial *null* es un miembro de todo dominio, significa que el valor es desconocido o no existe

Si para una tupla no tenemos el valor de un atributo por algún motivo, podemos poner *null* como valor para ese atributo.

Relaciones

relación r : tabla r

- columnas $D_1, D_2, \dots D_n$
- filas: elemento $t \in r$:

cada fila es una tupla

Ej.: $\text{persona} \subseteq \text{string} \times \text{integer} \times \text{integer}$.

tupla: ("Jorge Pérez", 51, 18003567)

Notación de relaciones

relación: tabla

- columnas $D1, D2, \dots Dn$
- filas: elemento $t \in r$:

cada fila es una tupla

Ej.: $\text{persona} \subseteq \text{string} \times \text{integer} \times \text{integer}$

tupla: ("Jorge Pérez", 51, 18003)

nombres de relaciones
con **minúscula**, nombres
de esquema con
mayúscula

Ejemplo de relación en un modelo relacional

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Terminología

informal	formal
tabla	relación
encabezado de columna	atributo
valores posibles en una columna	dominio
fila	tupla
definición de tabla	esquema de relación
tabla poblada	estado de la relación

Definición de BD relacional

conjunto de esquemas de relación

1. definimos los conceptos del problema mediante esquemas de relación,
2. definimos los datos como tablas asociadas a esos esquemas,
3. podemos poblar las tablas, consultarlas y alterarlas.

Claves
Slbrschtz cap 2.3.

Superclaves

Sea $K \subseteq R$, R esquema de relación; K es una superclave de R si los valores para K son suficientes para identificar una tupla única en cada posible relación $r(R)$

Ej.: instructor(ID, name, dept name, salary)

$\{ID\}$ e $\{ID, name\}$ son superclaves de instructor.

Claves candidatas y claves primarias

K es clave candidata, si K es mínima:
para todo atributo de K, si se lo
quito, K deja de ser superclave

Ejemplo: {ID} clave candidata de instructor

¡Atención! No confundir clave candidata
con superclave de cardinalidad mínima

Claves candidatas y claves primarias

Una de las claves candidatas es elegida para ser la clave primaria

Claves foráneas

Restricción de clave foránea (o de integridad referencial): el valor en una relación debe aparecer en otra.

Ejemplo:

`instructor(ID, name, dept name, salary)`

`department(dept name, building, budget)`

El valor de dept name en instructor debe aparecer en department

Claves foráneas

En otras palabras:

Los valores de uno o más atributos en una tupla de la relación referenciante aparecen en uno o más atributos de una tupla en la relación referenciada.

Claves foráneas

En otras palabras:

Los atributos referenciados en la relación referenciada suelen formar una clave primaria del esquema de la relación referenciada.

Claves foráneas

En otras palabras:

Generalizando aún más: los atributos referenciados de la relación referenciada suelen formar una clave candidata del esquema de la relación referenciada.

Notación

relación: $r, s, u, r1, r2, \dots$

esquema de relación: $R, S, U, R1, R2, \dots$

Sea $t \in r, r(R), A \in R$, $t[A]$ es el valor de t en A .

Sea $t \in r, r(R)$, $t[i]$ es el valor de t en el atributo i -ésimo de R .

Traducción de DER a tablas
Slbrschtz cap 2.9.

Principio básico

Para cada conjunto de entidades y para cada conjunto de relaciones de la BD hay una única tabla, con el nombre del conjunto de entidades o del conjunto de relaciones correspondiente

Cada tabla tiene varias columnas, cada una de las cuales tiene un nombre único

Reglas de traducción de DER a tablas

necesitamos:

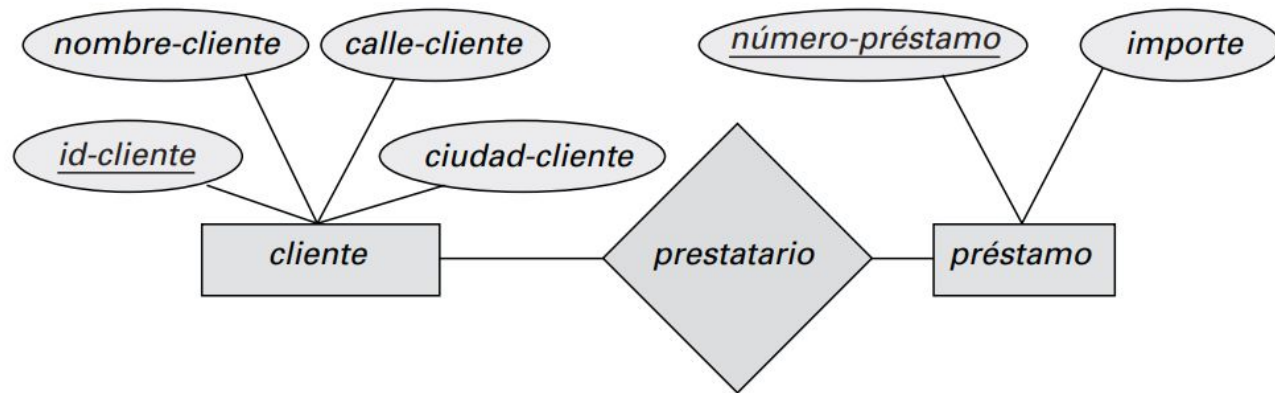
- Para cada entidad (CE) y relación (CR) hay un esquema relacional único
- Identificar claves primarias
- Identificar claves foráneas

Entidad fuerte, atributos simples

Una entidad (CE) fuerte sin atributos compuestos ni multi-valorados se traduce a un esquema relacional con los mismos atributos

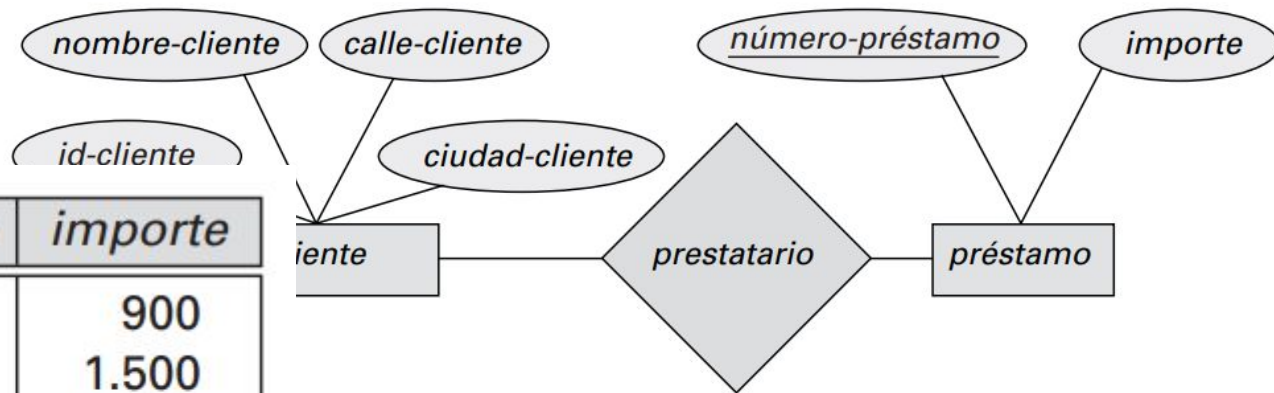
La clave primaria del CE se convierte en la clave primaria del esquema relacional.

Entidad fuerte, atributos simples



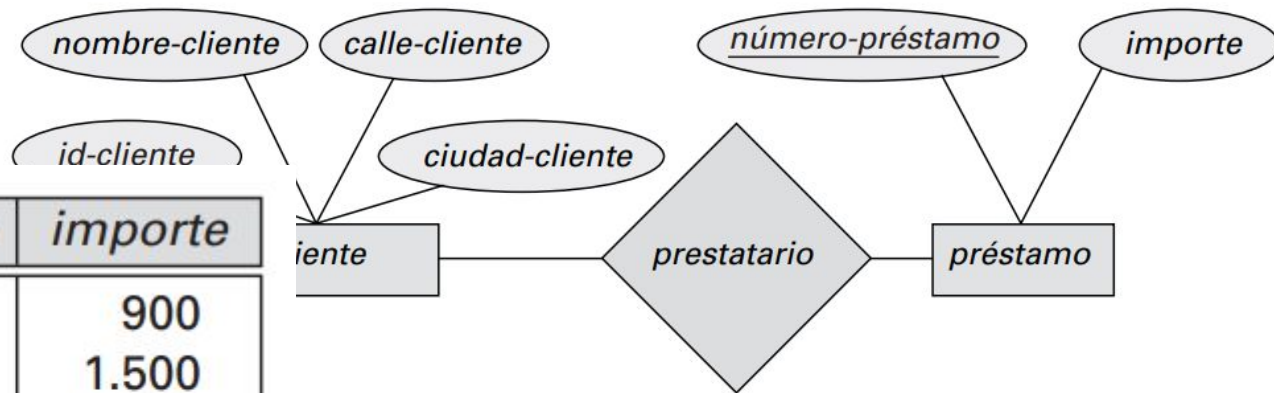
Entidad fuerte, atributos simples

<u>número-préstamo</u>	<u>importe</u>
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500



Entidad fuerte, atributos simples

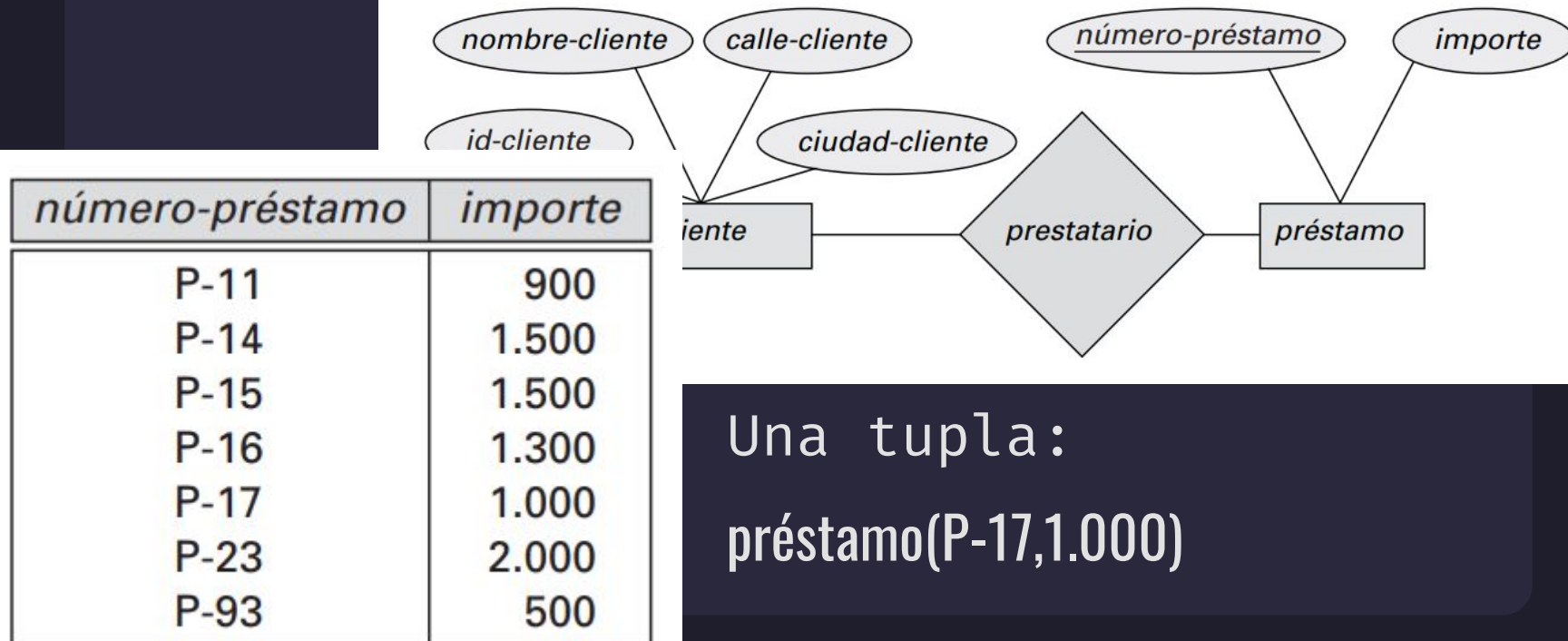
<u>número-préstamo</u>	importe
P-11	900
P-14	1.500
P-15	1.500
P-16	1.300
P-17	1.000
P-23	2.000
P-93	500



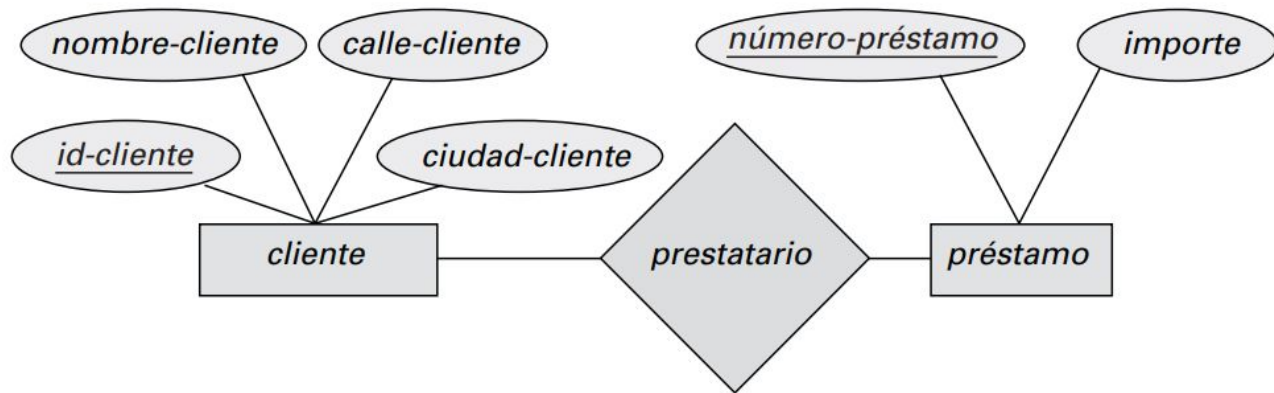
se traduce a:

Préstamo(número-préstamo, importe)

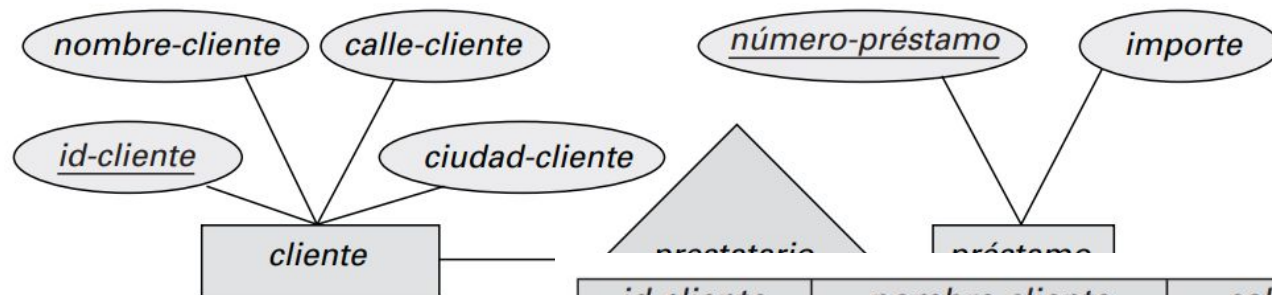
Entidad fuerte, atributos simples



Entidad fuerte, atributos simples

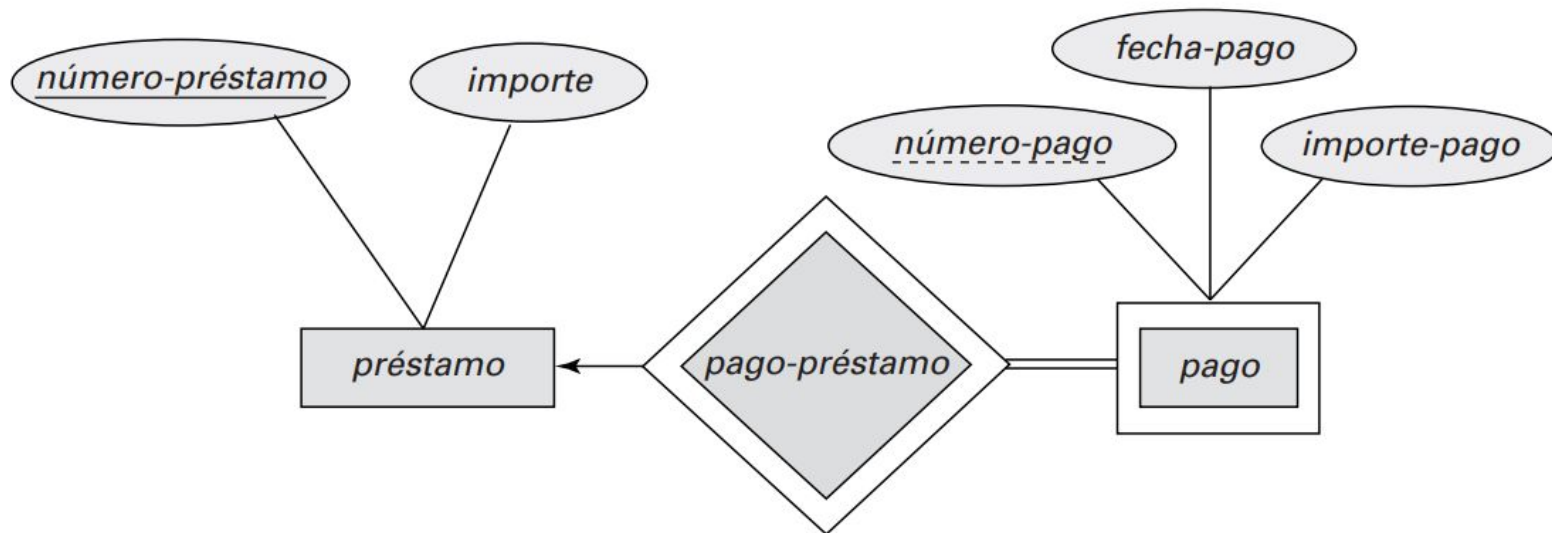


Entidad fuerte, atributos simples

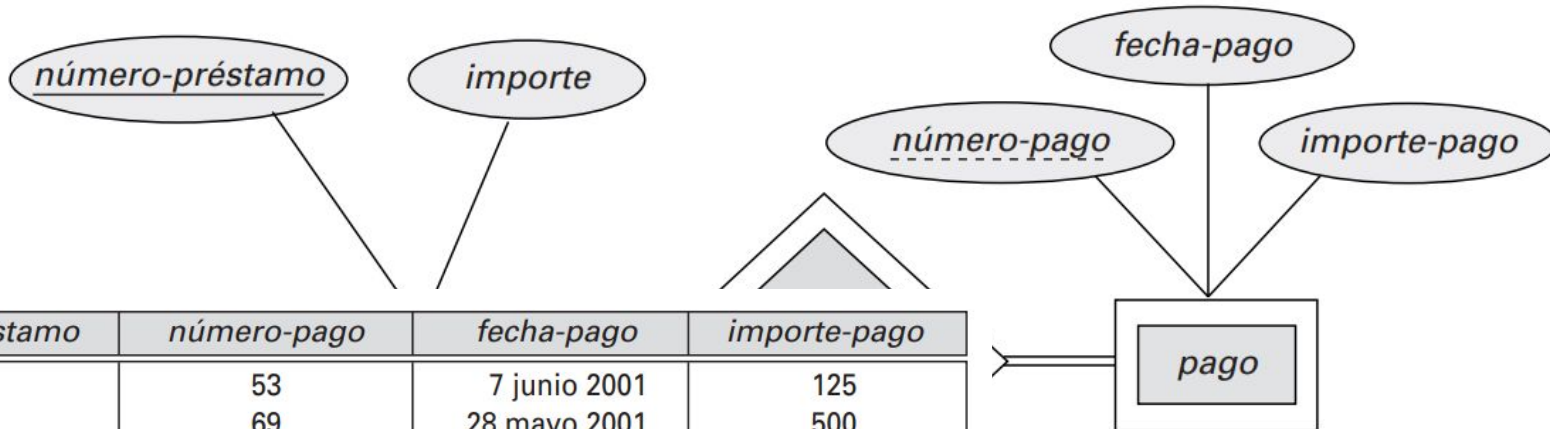


<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
01.928.374	Gómez	Carretas	Cerceda
18.273.609	Abril	Preciados	Valsaín
19.283.746	González	Arenal	La Granja
24.466.880	Pérez	Carretas	Cerceda
32.112.312	Santos	Mayor	Peguerinos
33.557.799	Fernández	Jazmín	León
33.666.999	Rupérez	Ramblas	León
67.789.901	López	Mayor	Peguerinos
96.396.396	Valdivieso	Goya	Vigo

Entidad débil, atributos simples



Entidad débil, atributos simples



<i>numero-préstamo</i>	<i>numero-pago</i>	<i>fecha-pago</i>	<i>importe-pago</i>
P-11	53	7 junio 2001	125
P-14	69	28 mayo 2001	500
P-15	22	23 mayo 2001	300
P-16	58	18 junio 2001	135
P-17	5	10 mayo 2001	50
P-17	6	7 junio 2001	50
P-17	7	17 junio 2001	100
P-23	11	17 mayo 2001	75
P-93	103	3 junio 2001	900
P-93	104	13 junio 2001	200

Atributos compuestos

Se crea un atributo separado para cada uno de los atributos componentes

- no se crea una columna separada para el propio atributo compuesto

Atributos compuestos



Atributos compuestos



`biblioteca(nombre, calle, número, ciudad)`

Atributos compuestos

Se crea un atributo separado para cada uno de los atributos componentes

- no se crea una columna separada para el propio atributo compuesto

... o sí!

redundancia pero legibilidad y eficiencia

Atributos compuestos

Se crea un atributo separado para cada uno de los atributos componentes

- no se crea una columna separada para el propio atributo compuesto

... o sí!

redundancia pero legibilidad y eficiencia

También están los nombres de columna!!!

Atributos multivalorados

Los atributos suelen ser columnas, pero...
Para un **atributo multivalorado** se crea una **tabla nueva**, con una columna que tiene la **clave primaria** del conjunto de entidades o conjunto de relaciones del que es atributo

Atributos multivalorados



Atributos multivalorados



libro(título, ISBN, editorial, edición)

libro-autor(ISBN, autor)

Atributos multivalorados



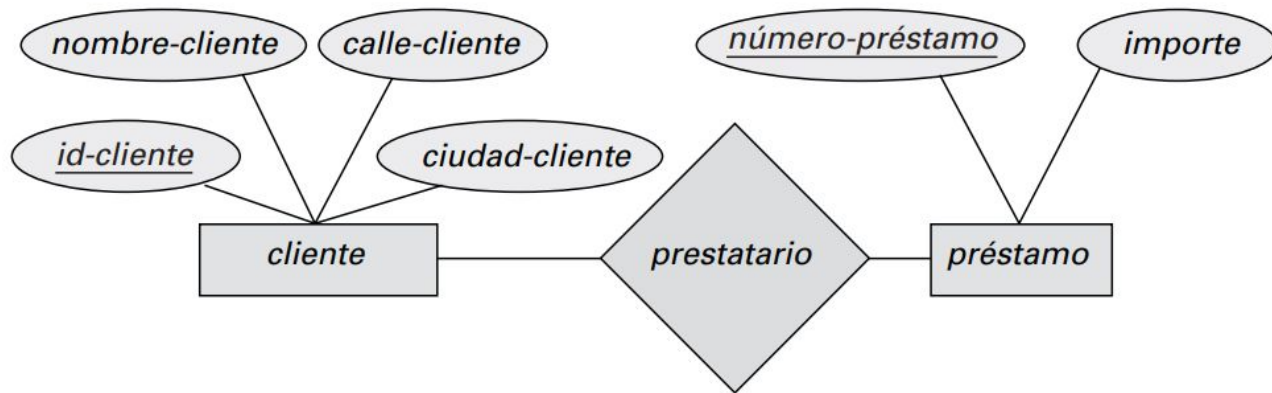
libro(título, ISBN, editorial, edición)

libro-autor(ISBN, autor)

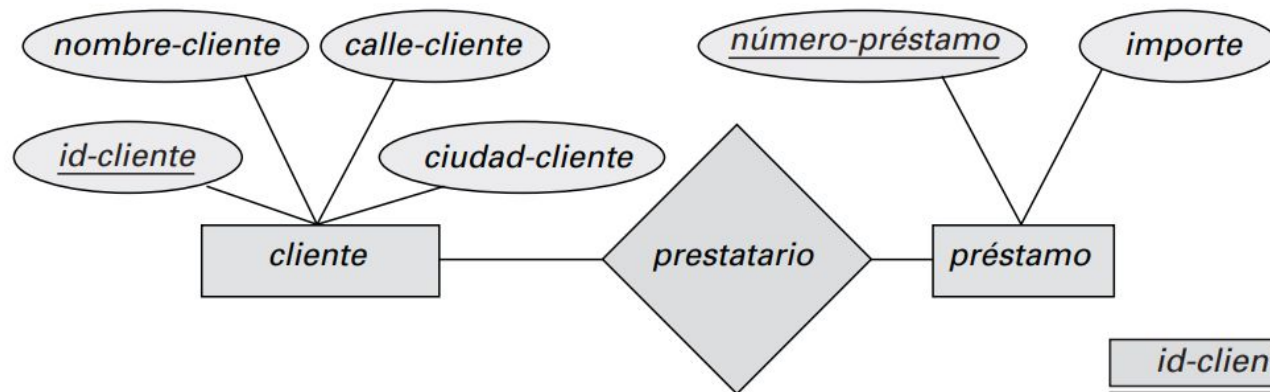
Con restricción de
integridad referencial

For libro-autor **foreign key** ISBN
references libro

Relación (CR)



Relación (CR)



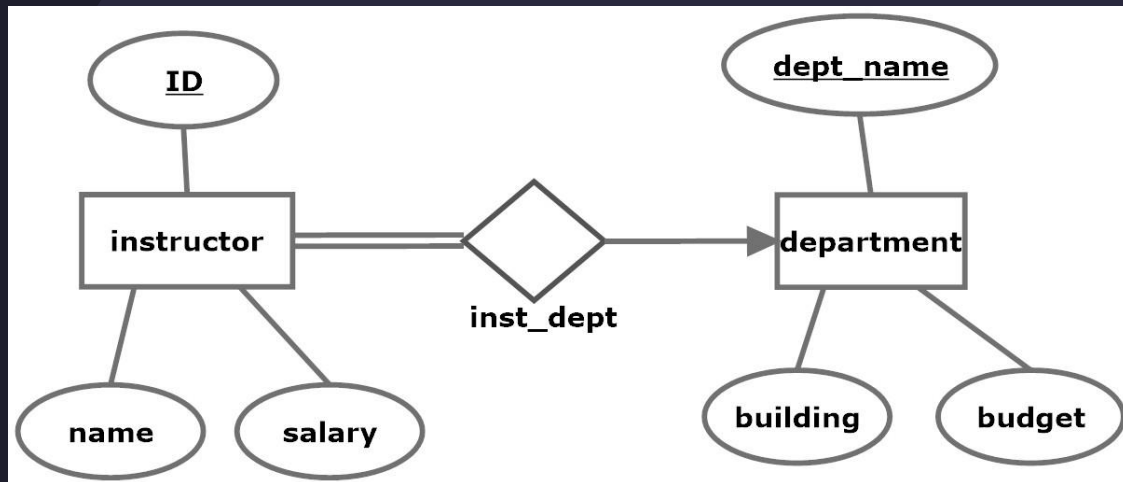
<i>id-cliente</i>	<i>número-préstamo</i>
01.928.374	P-11
01.928.374	P-23
24.466.880	P-93
32.112.312	P-17
33.557.799	P-16
55.555.555	P-14
67.789.901	P-15
96.396.396	P-17

Relaciones varios a varios

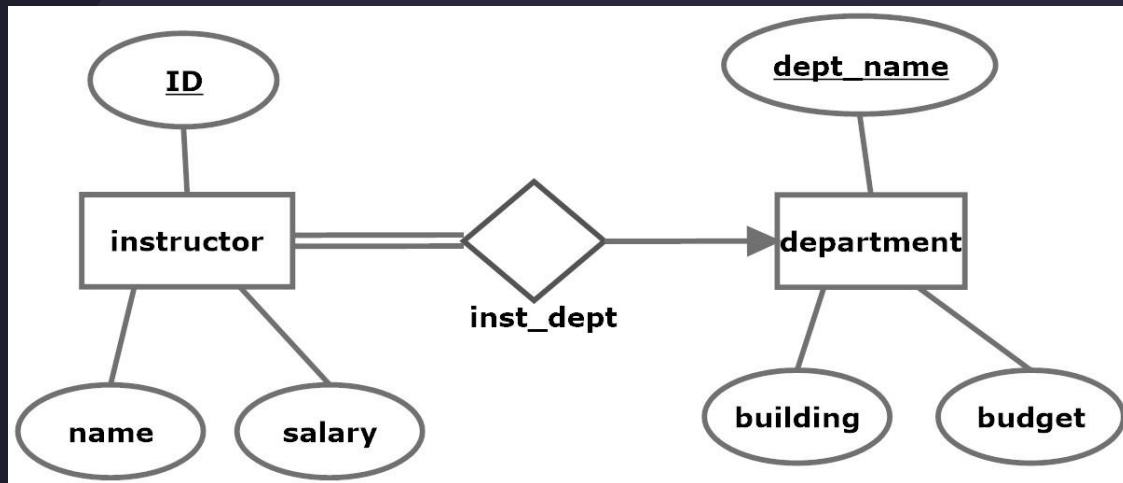
Representamos una relación (CR) varios a varios con un esquema con atributos para las claves primarias de las dos entidades (CE) y los atributos simples de la relación

- la clave primaria es la unión de las claves primarias de las entidades

Relaciones varios a varios



Relaciones varios a varios



`instructor(ID, name, salary, dept_name)`

For instructor **foreign key** dept_name **references** department

Relaciones uno a varios

Una relación (CR) varios a uno se representa agregando atributos extra en la entidad (CE) del lado varios, con la clave primaria del lado uno.

Relaciones uno a varios

- La clave primaria de la relación es la clave primaria de la entidad del lado varios.
- Se crea una restricción de clave foránea de la relación que referencia a la clave primaria de la entidad del lado varios.

Relaciones uno a varios (observación)

Si la participación es parcial en el lado varios, aplicar la regla anterior puede resultar en valores nulos.

Esto sucede cuando a una entidad del CE del lado varios no le corresponde ninguna entidad del CE del lado uno.

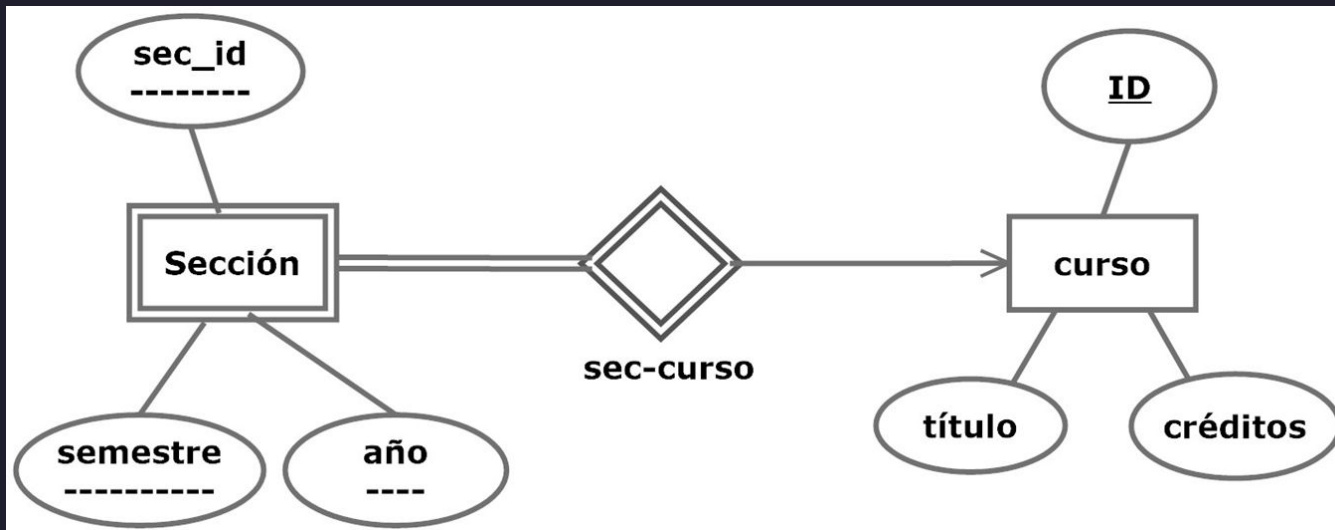
Relaciones uno a uno

Una relación (CR) uno a uno se representa agregando al esquema de una de las entidades la clave primaria de la otra

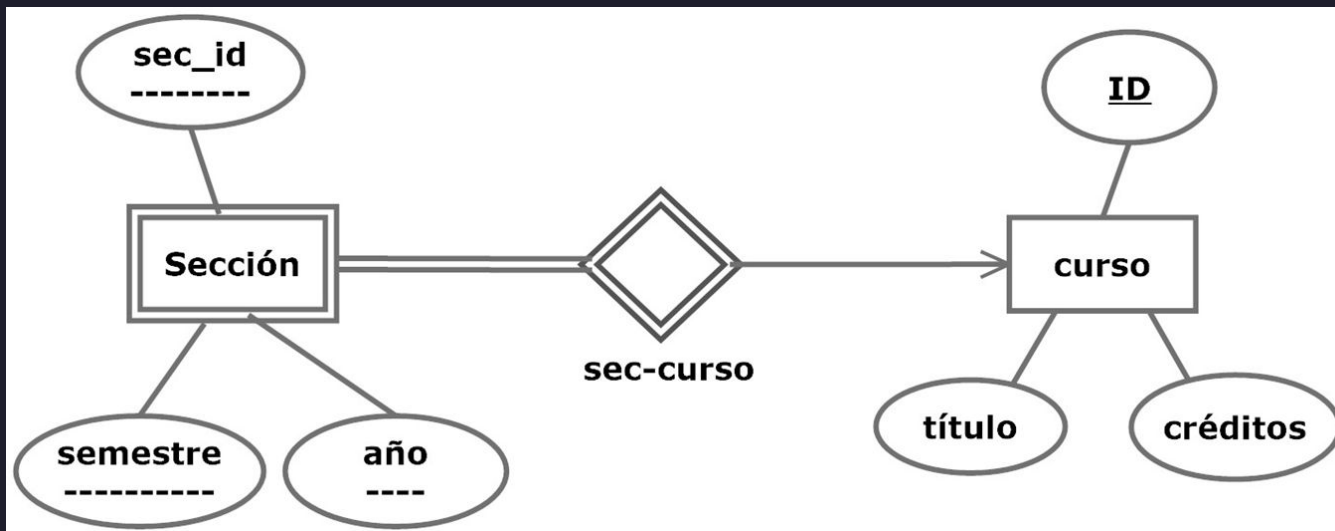
Relaciones uno a uno

- La clave primaria de la relación puede ser la de cualquiera de las entidades
- Se crea restricción de clave foránea que referencia la clave primaria de la entidad que no se tomó de base para crear el esquema

Relaciones uno a uno



Relaciones uno a uno



sección(ID, sec id, semestre, año)

For sección **foreign key** ID **references** curso

Sutilezas de representación

(que no vamos a ver ahora)

- Generalización
- Agregación

Restricciones de integridad

Para qué sirven las restricciones de integridad

asegurar que los usuarios (autorizados)
no puedan modificar la base de datos de
forma que se pierda la consistencia

Cómo son las restricciones de integridad

predicados arbitrarios

¡complicados de verificar!

en la práctica, sólo restricciones
verificables con poca sobrecarga

Cómo son las restricciones de integridad

predicados arbitrarios

¡complicados de verificar!

en la práctica, sólo restricciones
verificables con poca solución

Veremos las de SQL y
luego pensaremos
también en términos de
restricciones para
otros problemas

Tipos de restricciones

- Al modelo entidad-relación
 - Claves (Silberschtz cap 2.3)
 - Relaciones (cap 2.2) (aridad, total, parcial)
- Dominios (cap 6.1)
- Integridad referencial (cap 6.2)
- Aserciones (*asertos*) (cap 6.3)
- Disparadores (*triggers*) (cap 6.4)
- Dependencia funcional (cap 7)
- Seguridad (cap 6.5 a 6.7)

Propiedades de relaciones como restricciones

¿Cómo podemos pensar las reglas de transformación a tablas de las relaciones en términos de restricciones?

Propiedades de relaciones como restricciones

¿Cómo podemos pensar las reglas de transformación a tablas de las relaciones en términos de restricciones?

- relación total, relación parcial
- Uno a muchos, muchos a muchos

Restricciones sobre dominios

```
create domain número-cuenta char(10)
```

```
constraint
```

```
comprobación-número-cuenta-nulo
```

```
check(value not null)
```


Restricciones de integridad referencial

Tratan de que las diferentes tablas mantengan las referencias que tenemos en el Diagrama Entidad - Relación

Restricciones de integridad referencial



Restricciones de integridad referencial



libro(título, ISBN, editorial, edición)

libro-autor(ISBN, autor)

Con restricción de
integridad referencial

For libro-autor **foreign key** ISBN
references libro

Qué pasa cuando modifico la BD?



Aserciones

La suma de los importes de los préstamos de cada sucursal debe ser menor que la suma de los saldos de sus cuentas.

```
create assertion restricción-suma check
(not exists (select * from sucursal
  where (select sum(importe) from préstamo
    where préstamo.nombre-sucursal = sucursal.nombre-sucursal)
  >= (select sum (importe) from cuenta
    where préstamo.nombre-sucursal = sucursal.nombre-sucursal))))
```

Disparadores

modelo evento-condición-acción

1. evento que causa
2. comprobación del disparador
3. condición que se debe cumplir para ejecutar el disparador
4. acciones que ejecuta el disparador

Disparadores

```
create trigger descubierto after update on cuenta  
referencing new row as nfila  
for each row  
when nfila.saldo < 0  
begin atomic  
    insert into prestatario  
        (select nombre-cliente, número-cuenta  
         from impositor  
         where nfila.número-cuenta = impositor.número-cuenta);  
    insert into préstamo values  
        (nfila.número-cuenta, nfila.nombre-sucursal, – nfila.saldo)  
    update cuenta set saldo = 0  
        where cuenta.número-cuenta = nfila.número-cuenta  
end
```

Seguridad

De lectura, de escritura, de borrado...

Privilegios, autorizaciones, vistas

Cifrado y autenticación

Dependencia funcional

Más adelante!

Y cuando no hay reglas?



/THANKS!

/DO YOU HAVE ANY QUESTIONS?

youremail@freepik.com
+91 620 421 838
yourwebsite.com



CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**

> Please keep this slide for attribution

