

# Capítulo 2

## Modelo relacional y traducción de diagramas ER a esquemas relacionales



# Transición de Modelo ER a Modelo Relacional

- **Situación:**

- Asumimos que se dispone de un buen diseño de BD usando modelo ER.
- En la práctica se usan otros modelos para definir consultas y para alterar los datos.
  - El modelo relacional es uno de los más usados.
  - En el mismo se mantienen y consultan **tablas de información**.

- **Problema:** ¿Cómo aprovechar el modelo ER haciendo la transición desde él a un modelo relacional?

- **Propósito:** En esta sección consideramos los siguientes asuntos:

- Comprender qué son esquemas de modelo relacional y algunas de las restricciones de integridad más comunes para ellos,
- Resolver el problema anterior.

# Esquemas e instancias relacionales

- **Atributos:**

- En la Universidad tenemos instructores y cada uno tiene identificador, nombre, nombre de departamento y salario....
- Atributos: En el enunciado: *ID, name, dept\_name, salary*
- **Notación:**  $A_1, A_2, \dots, A_n$  son atributos.

- **Esquemas relacionales:**

- $R = (A_1, A_2, \dots, A_n)$  es un **esquema de relación**
- **Ejemplo:**
  - *Instructor* = (ID, name, dept\_name, salary)

# Esquemas e instancias relacionales

- ***Significado e implicaciones***
  - Dado un *enunciado en lenguaje natural*, se puede identificar atributos primero,
  - y luego se pueden proponer esquemas de relación para distintos *conceptos* del problema actual.
  - Cómo hacer este trabajo es tema de un capítulo posterior.

# Esquemas e instancias relacionales

- Dados conjuntos  $D_1, D_2, \dots, D_n$  una **relación**  $r$  es un subconjunto de  $D_1 \times D_2 \times \dots \times D_n$ 
  - Las relaciones son conjuntos (el orden entre las tuplas no importa).
  - Se puede pensar una relación como una **tabla** con columnas para los dominios  $D_1, D_2, \dots, D_n$  respectivamente.
  - Un elemento  $t \in r$  es una **tupla** representado mediante una fila de la tabla.
- **Ejemplo:**  $persona \subseteq string \times integer \times integer$ . Una tupla puede ser: *(“Jorge Pérez”, 51, 18003567)*
- **Notación:** nombres de relaciones comienzan con minúscula, nombres de esquema comienzan con mayúscula.
  - Esto es para saber si me estoy refiriendo a una relación o a un esquema.

# Esquemas e instancias relacionales

- Notar que el *concepto de relación* en el **modelo relacional**; no tiene nada que ver con el concepto de relación en el modelo ER.
  - Es más bien como están acostumbrados a trabajar en matemática.
- ***Significado e implicaciones***
  - Se puede expresar la **información manejada** por una organización como un conjunto de relaciones o tablas.
  - En cada columna se deciden los dominios con los que se trabaja para los valores.

# Esquemas e instancias relacionales

- Ahora que vimos los conceptos de esquema de relación y de relación surge la pregunta:
  - ¿Cómo puedo juntar esos conceptos y qué se gana con hacerlo?
- **Notación:**  $r(R)$  significa  $r$  es una relación con esquema de relación  $R$ .
  - O sea, las columnas de  $r$  tienen como nombres los atributos de  $R$ .
- **Ejemplo:**
  - $persona(Persona)$
  - $Persona = (nombre, edad, DNI)$
- También lo puedo decir así:
  - $persona(nombre, edad, DNI)$

# Esquemas e instancias relacionales

- *Implicaciones de la definición anterior*
  - Los conceptos del problema que se asociaban a esquemas de relación ahora se terminan asociando también a relaciones o tablas de la BD.
  - Los dominios de las tablas pasan a ser los *dominios de los atributos* del esquema relacional.
  - Además es más fácil de comprender una tabla donde las columnas tienen los nombres de un esquema de relación.



# Ejemplo de Relación (instructor)

<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

attributes  
(or columns)

tuples  
(or rows)

# Tipos de atributos

- El conjunto de valores permitidos para cada atributo se llama **dominio del atributo**.
- **Exigencia:**
  - Los valores de los atributos se requiere que sean **atómicos**; esto es, indivisibles.
  - Esto se refiere al uso que se hace de los atributos.
    - O sea que en las consultas o restricciones de integridad no me voy a poner a dividir el valor de un atributo en partes.
    - Esto va a simplificar la descripción de consultas o restricciones de integridad.
- **Ejemplo:** Está mal decir *persona(nombre, DNI)* si voy a hacer consultas como “listar los apellidos de personas que tienen DNI”.

# Tipos de atributos

- **Implicaciones de la exigencia anterior:**
  - En un esquema de relación no puedo tener atributos multivalorados.
  - En un esquema de relación no puedo tener atributos compuestos.
  - Aquí surge una diferencia importante con el diseño de entidad-relación.

# Tipos de atributos

- **Situación:** A veces no sabemos el valor de un atributo o aun no existe.
- **Problema:** ¿Cómo podemos reflejar esto en la base de datos?
- **Solución:** El valor especial **null** es un miembro de todo dominio.
  - **null** 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.
- **Ejemplo:** para un empleado *teléfono celular de contacto*. Podemos no saberlo o no existir en cuyo caso le asignamos el valor null.

# Terminología

<u>Informal Terms</u>		<u>Formal Terms</u>
Table		Relation
Column Header		Attribute
All possible Column Values		Domain
Row    fila		Tuple
Table Definition		Schema of a Relation
Populated Table		State of the Relation

# Bases de datos relacionales

- Un **esquema de una BD relacional** es un conjunto de esquemas de relación.

$$R_1, \dots, R_n$$

- Una **base de datos relacional** consiste de múltiples relaciones.

$$r_1(R_1), \dots, r_n(R_n)$$

- **Implicaciones:**

- Primero definimos los conceptos del problema mediante esquemas de relación.
- Luego definimos los datos como tablas asociadas a esos esquemas.
- Luego podemos poblar las tablas, consultarlas y alterarlas.

# Bases de datos relacionales

- **Nomenclatura:**
  - **Nombre de relación:** expresado en minúscula,  $r, s, u, r_1, r_2, \dots$
  - **Nombre de esquema de relación:** expresado en mayúscula  $R, S, U, R_1, R_2, \dots$
- **Notación:** Sea  $t \in r, r(R), A \in R$ .  $t[A]$  es el valor de  $t$  en  $A$ .
- **Notación:** Sea  $t \in r, r(R)$ .  $t[i]$  es el valor de  $t$  en atributo  $i$ -ésimo de  $R$ .

# Superclaves

- **Propósito:** Ahora vamos a ver algunos tipos de **restricciones de integridad** muy comunes cuando se trabaja con el modelo relacional.
- **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  $r(R)$ .
  - **Observar** que se habla de relaciones para el problema o situación del mundo real que está siendo considerando.
- **Ejemplo:** *instructor(ID, name, dept name, salary)*  
 $\{ID\}$  e  $\{ID, name\}$  son superclaves de *instructor*.



# Claves candidatas y claves primarias

- Una superclave  $K$  es una **clave candidata**, si  $K$  es minimal.
  - Para todo atributo de  $K$  si se lo quito a  $K$  dejo de tener una superclave.
  - **Ejemplo:** {ID} clave candidata de *instructor*.
  - **¡Atención!** No confundir clave candidata con superclave de cardinalidad mínima.
- Una de las claves candidatas es elegida para ser la **clave primaria**.
- **Notación:**
  - Se indican los atributos de una clave primaria para un esquema de relación  $R$  **subrayando** los atributos de  $R$  que forman 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*
  - Relación **referenciante** y relación **referenciada**.
    - ❑ *Instructor* es la referenciante y *department* es la referenciada.
    - ❑ **For instructor foreign key dept name references department**

# Claves foráneas

- **Generalizando y siendo un poco más precisos:**
  - **Restricción de clave foránea:** 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.
  - Los atributos referenciados en la relación referenciada suelen formar una **clave primaria** del esquema de la relación referenciada.
  - **Generalizando aun más:** los atributos referenciados de la relación referenciada suelen formar una **clave candidata** del esquema de la relación referenciada.

# Esquema de BD universitaria

*classroom(building, room number, capacity)*

*department(dept name, building, budget)*

*course(course id, title, dept name, credits)*

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

*section(course id, sec id, semester, year, building, room number, time slot id)*

*teaches(ID, course id, sec id, semester, year)*

*student(ID, name, dept name, tot cred)*

*takes(ID, course id, sec id, semester, year, grade)*

*advisor(s ID, i ID)*

*time slot(time slot id, day, start time, end time)*

*prereq(course id, prereq id)*

# Claves foráneas

- **Ejercicio:** En el esquema relacional anterior hacer claves foráneas para *takes*
  - **Ayuda:** las materias (*course*) se dividen en secciones (*section*); los alumnos se asignan a secciones (para cada sección se toma una evaluación); para esto se usa *takes*.

# Claves foráneas

- **Solución:**

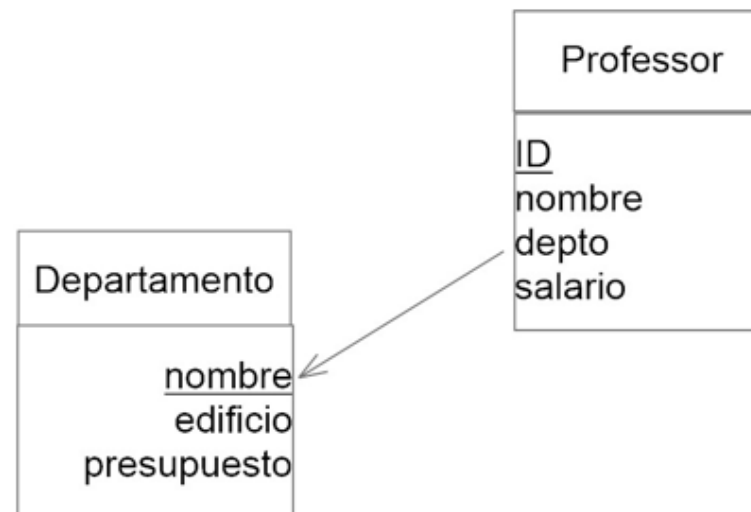
- **For takes foreign key *ID* references *student***
- **For takes foreign key *course id, sec id, semester, year* references *section***

# Claves foráneas

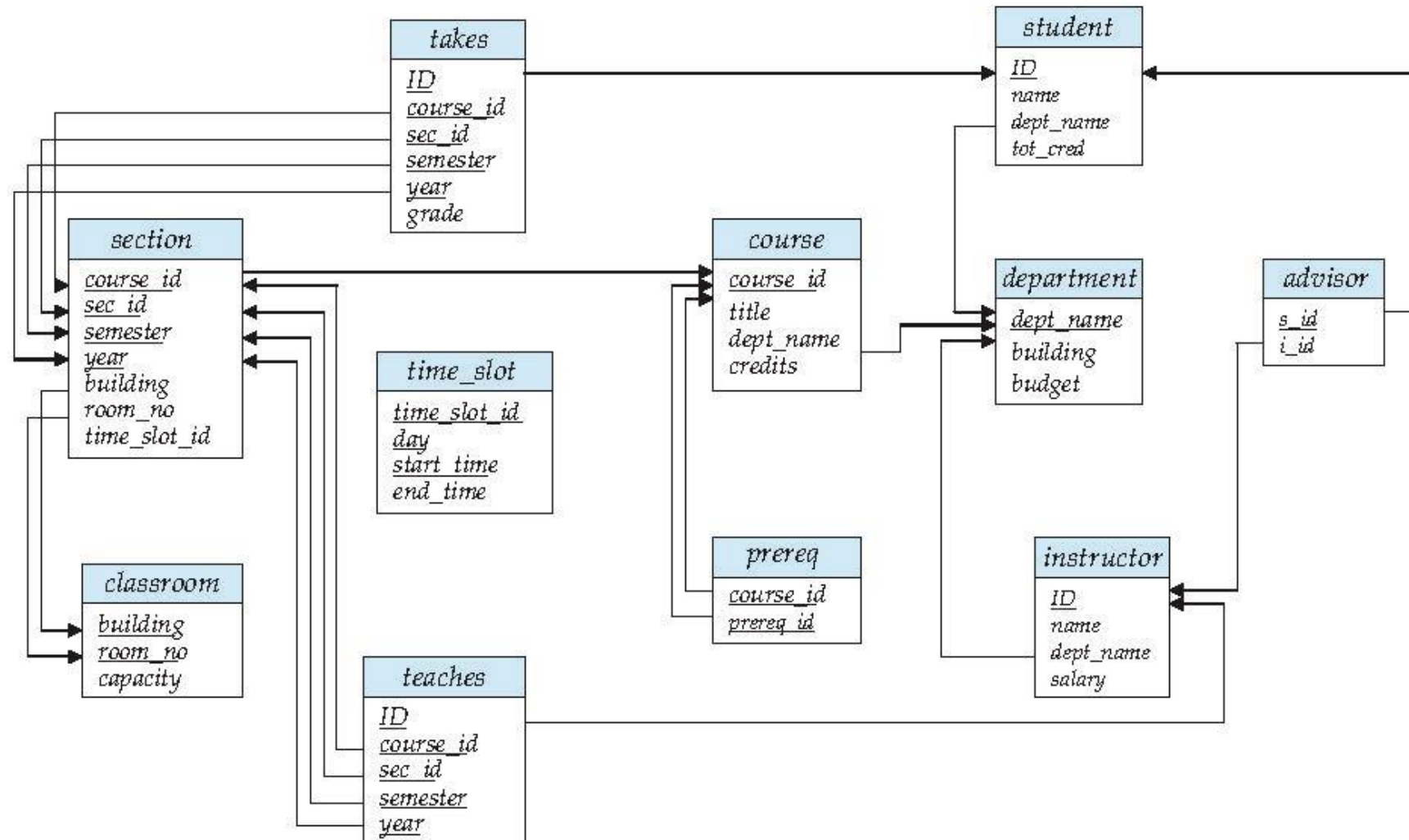
- **Representación gráfica:**

- ❑ Los esquemas se representan con rectángulos conteniendo nombre de esquema y nombres de atributos.
- ❑ Los atributos de clave primaria se subrayan.
- ❑ Las restricciones de clave foránea se representan mediante flechas que van de atributos referenciantes (de esquema referenciante) a atributos referenciados (de esquema referenciado).

- **foreign key** *depto of Profesor* **references** *nombre of Departamento*



# Diagrama de esquema para BD universitária





# Diseño de BD relacional

- **Problema:** ¿Cómo hacer un buen diseño de BD relacional?
  - Esto es lo mismo que encontrar un buen esquema de una BD relacional.
- **Mal diseño:**
  - *univ (instructor -ID, name, dept\_name, salary, student\_Id, ..)*
- Almacenar toda la información en una sola relación resulta en:
  - **Repetición de la información**
    - **Ejemplo:** dos estudiantes con el mismo instructor
  - **Necesidad de valores nulos**
    - **Ejemplo:** representar un estudiante sin supervisor

# Diseño de BD relacional

- **Solución 1:** Aprovechar un buen diseño de esquema ER.
  - Si se tiene un buen diseño de ER, se lo puede transicionar a un buen diseño relacional.
  - Este es el enfoque a usar en esta parte del curso.
- **Solución 2:** Obtener un buen diseño relacional a partir de los atributos y las restricciones de integridad del problema.
  - La **teoría de normalización** trabaja con esta idea y trata con cómo diseñar buenos esquemas de BD relacionales.
  - Este es asunto de un capítulo posterior.

# Reducción a Esquemas Relacionales

- **Problema:** Se tiene un buen diseño de esquema de BD usando modelado-ER, pero en la práctica se usan esquemas de BD relacionales.
  - O sea que necesito un buen diseño de esquema de BD relacional.
- **Solución:** Traducción de diagrama ER a un conjunto esquemas relacionales con claves primarias y restricciones de clave foránea.
  - Hay **reglas de traducción** que suelen resolver un porcentaje alto del problema.
  - Pero esas reglas no contemplan todos los casos.

# Reducción a Esquemas Relacionales

- *Estudiaremos:*

- Reglas de traducción automáticas.

- *Situación más complicada:*

- Cuando *no hay* regla de traducción automática que se pueda usar, pero:
  - Hay alguna idea para hacer la traducción ¿cómo evaluarla?
  - O hay más de una alternativa para hacer la traducción que se puede aplicar, ¿cómo decidir cuál de ellas conviene usar?
- En ambos casos usar **criterios de decisión** que nos ayudarán a escoger entre opciones de traducción o a mejorar una **idea de traducción**.
  - Redundancia de información
  - Facilidad de comprensión
  - Facilidad para hacer consultas

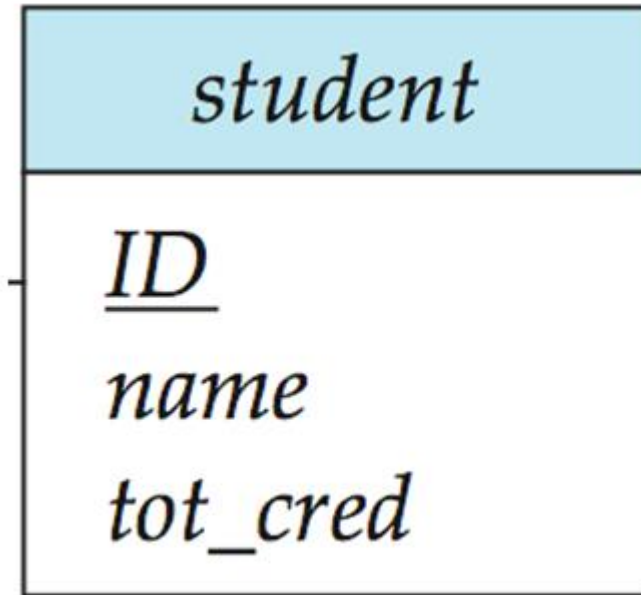
# Reducción a Esquemas Relacionales

- **Propiedad de la traducción de diagramas ER a tablas a respetar:**
  - Para cada CE y CR hay un esquema relacional único al cual se asigna el CE / CR.
- ***Si se la cumple se gana en:***
  - Simplicidad
  - Localización de la traducción en un esquema

# Reducción a Esquemas Relacionales

- *Otras informaciones que se deben obtener junto con los esquemas de relación obtenidos por la traducción automática:*
  - **Claves primarias** de esquemas de relación.
    - Para esto se usa la información de claves primarias de CE.
  - **Restricciones de clave foránea.**
    - Para esto se usa el tipo de elemento de modelo ER a mapear (p.ej. CR, CE débiles, atributo multivalorado, etc.).

# Reducción a Esquemas Relacionales



***Se mapea a:***

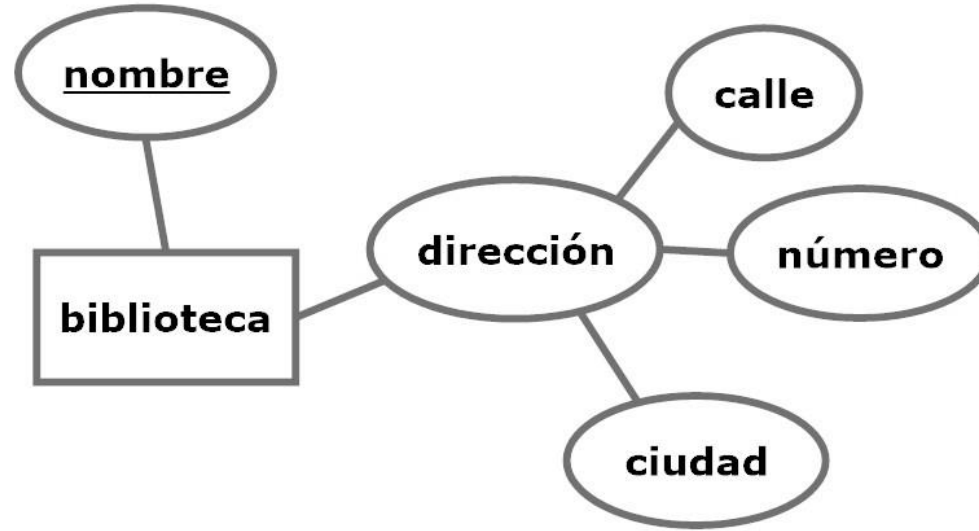
*student*(*ID*, *name*, *tot\_cred*)

# Reducción a Esquemas Relacionales

- **Regla CEF-1:** Un CE fuerte que no involucra atributos compuestos ni atributos multi-valorados se mapea a un esquema relacional con los mismos atributos.
  - La clave primaria del CE se convierte en la clave primaria del esquema relacional.



# Reducción a Esquemas Relacionales



*Se mapea a:*

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

# Reducción a Esquemas Relacionales

- **Regla CEF-2:** Un CE fuerte que no involucra atributos/subatributos multivalorados se mapea a un esquema relacional con los mismos atributos simples y los subatributos hoja de los atributos compuestos.

# Reducción a Esquemas Relacionales



*Se mapea a:*

*libro(título, ISBN, editorial, edición)*

*libro-autor(ISBN, autor)*

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

# Reducción a Esquemas Relacionales

- **Aclaración:** Cada valor del atributo multivalorado mapea a una tupla separada en la tabla del esquema *EM*.
  - Por ejemplo un libro con ISBN 970-17-0256-5 y autores con DNIs: 11122256 y 15345678 se mapea a dos tuplas: (970-17-0256-5, 11122256 ) y (970-17-0256-5, 15345678)

# Reducción a Esquemas Relacionales

- **Regla CEF-3:** un atributo multivalorado  $M$  simple de un CE  $E$  es representado por un esquema separado  $EM$ .
  - $EM$  tiene atributos correspondientes a la clave primaria de  $E$  y un atributo correspondiente al atributo multivalorado  $M$ .
  - Todos los atributos de  $EM$  forman su clave primaria.
  - Se pone una **restricción de clave foránea** desde  $EM$  que referencia a la clave primaria de  $E$ .

# Reducción a Esquemas Relacionales

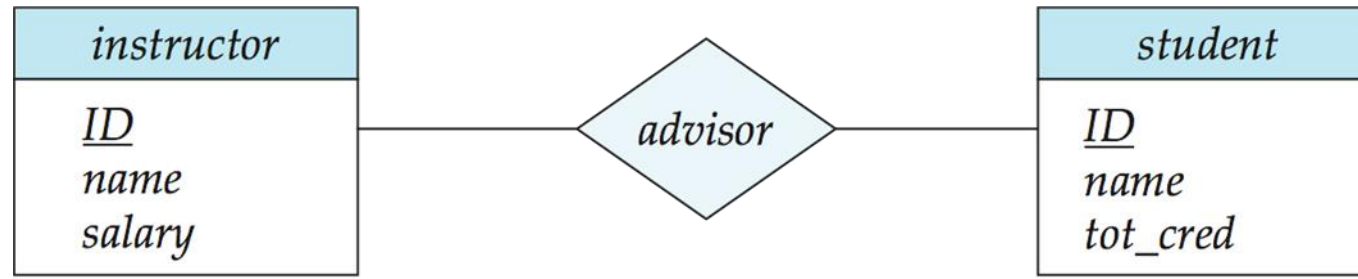
- **Decisión:** Los *atributos derivados* no son explícitamente representados en el modelo de datos relacional.
  - Se verá que si se los necesita una forma de computarlos es por medio de consultas.

# Reducción a Esquemas Relacionales

- **Ejercicio:** reducir a tablas el siguiente conjunto de entidades

<i>instructor</i>
<u><i>ID</i></u>
<i>name</i>
<i>first_name</i>
<i>middle_initial</i>
<i>last_name</i>
<i>address</i>
<i>street</i>
<i>street_number</i>
<i>street_name</i>
<i>apt_number</i>
<i>city</i>
<i>state</i>
<i>zip</i>
{ <i>phone_number</i> }
<i>date_of_birth</i>
<i>age</i> ( )

# Reducción a Esquemas Relacionales



**Se mapea a:**

*Advisor* = (*i-id*, *s-id*)

**For Advisor foreign key *i-id* references *instructor***

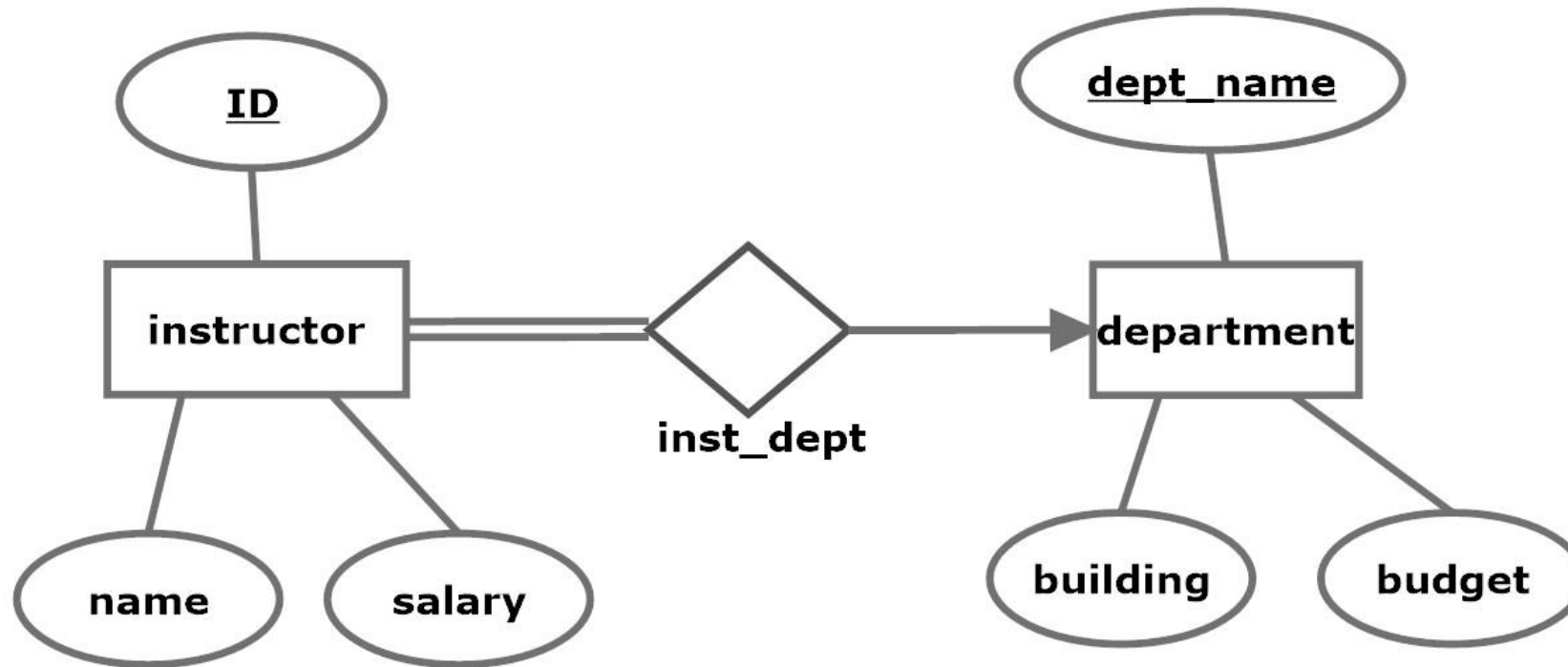
**For Advisor foreign key *s-id* references *student***



# Reducción a Esquemas Relacionales

- **Regla CR1:** un CR varios-varios es representado con un esquema con atributos para las claves primarias de los dos CE participantes y todos los atributos descriptivos del CR (que no son multivalorados).
  - la clave primaria del esquema del CR es la unión de las claves primarias de los CE que participan en el CR.
  - Para cada CE que participa en el CR se crea restricción de clave foránea que referencia clave primaria de CE.

# Reducción a Esquemas Relacionales



*CR inst\_dept se mapea a:*

*instructor(ID, name, salary, dept\_name)*

**For instructor foreign key dept\_name references department**

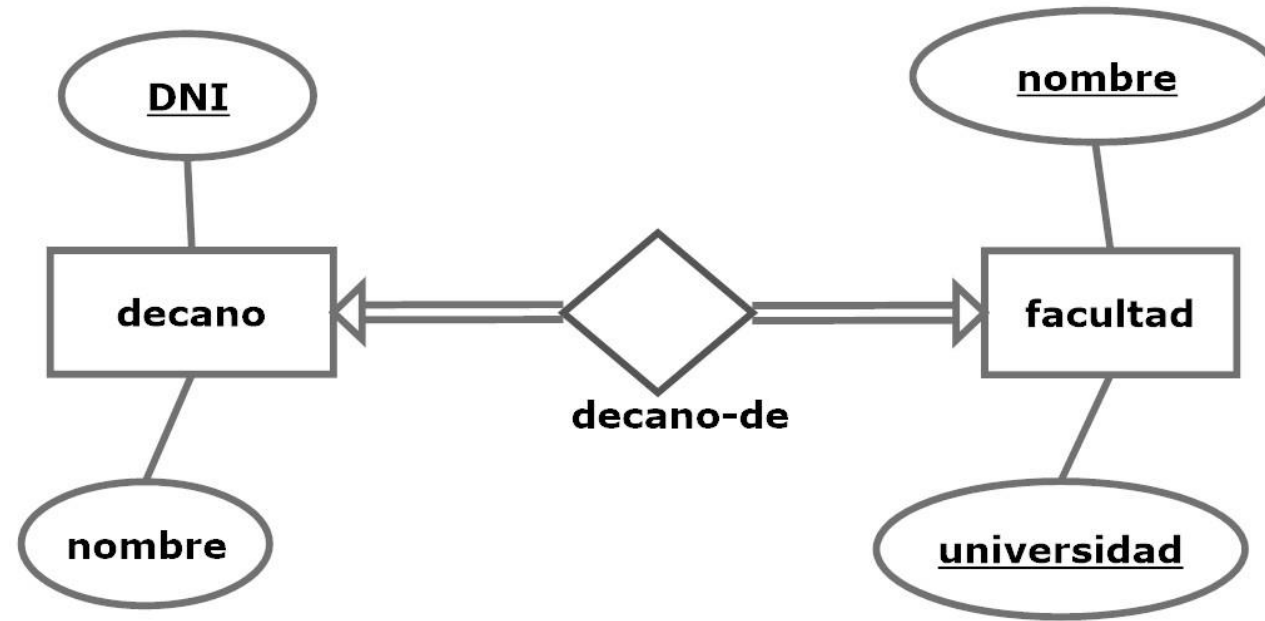
# Reducción a Esquemas Relacionales

- **Regla CR2:** un CR varios a uno o uno a varios que es **total** en el lado varios puede ser representado agregando atributos extra en el CE del lado varios, conteniendo la clave primaria del lado uno.
  - La clave primaria del CR es la clave primaria del CE del lado varios.
  - Se crea restricción de clave foránea de CR que referencia a clave primaria de CE de lado varios.

# Reducción a Esquemas Relacionales

- 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.

# Reducción a Esquemas Relacionales



*El CR decano\_de se mapea a:*

Las siguientes respuestas son igualmente válidas:

*decano(DNI, nombre, nombreFacultad, universidad )*

**For decano foreign key nombreFacultad, universidad  
references facultad**

y

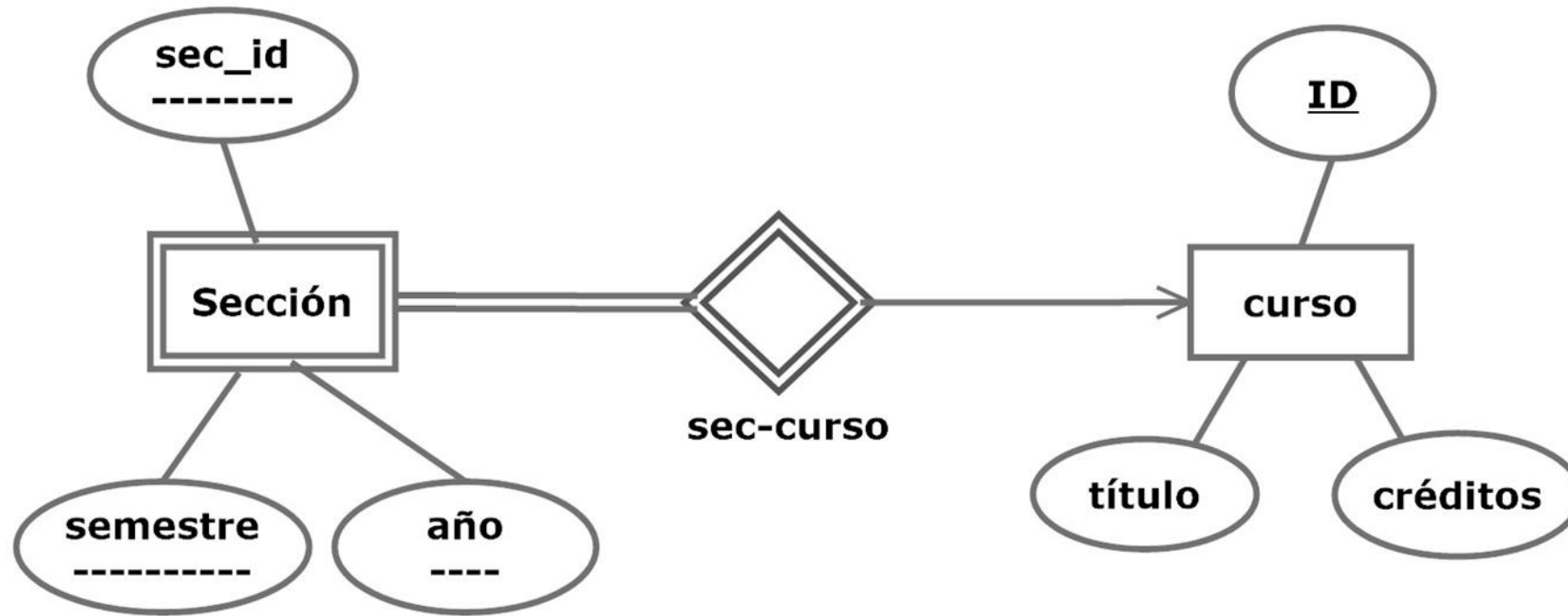
*facultad (nombre, universidad, DNI)*

**For facultad foreign key DNI references decano**

# Reducción a Esquemas Relacionales

- **Regla CR3:** Un CR uno a uno con participación total en ambos lados puede ser mapeado agregando al esquema resultante de traducir uno de los CE participantes los atributos de la clave primaria del otro CE.
  - La clave primaria de cualquier CE puede ser elegida como la clave primaria del CR.
  - Se crea restricción de clave foránea de esquema relacional asociado al CR que referencia clave primaria de otro CE (el que no se tomo de base para hallar el esquema asociado al CR).

# Reducción a Esquemas Relacionales



*El CE débil sección se mapea a:*

*sección(ID, sec\_id, semestre, año)*

**For sección foreign key ID references curso**

# Reducción a Esquemas Relacionales

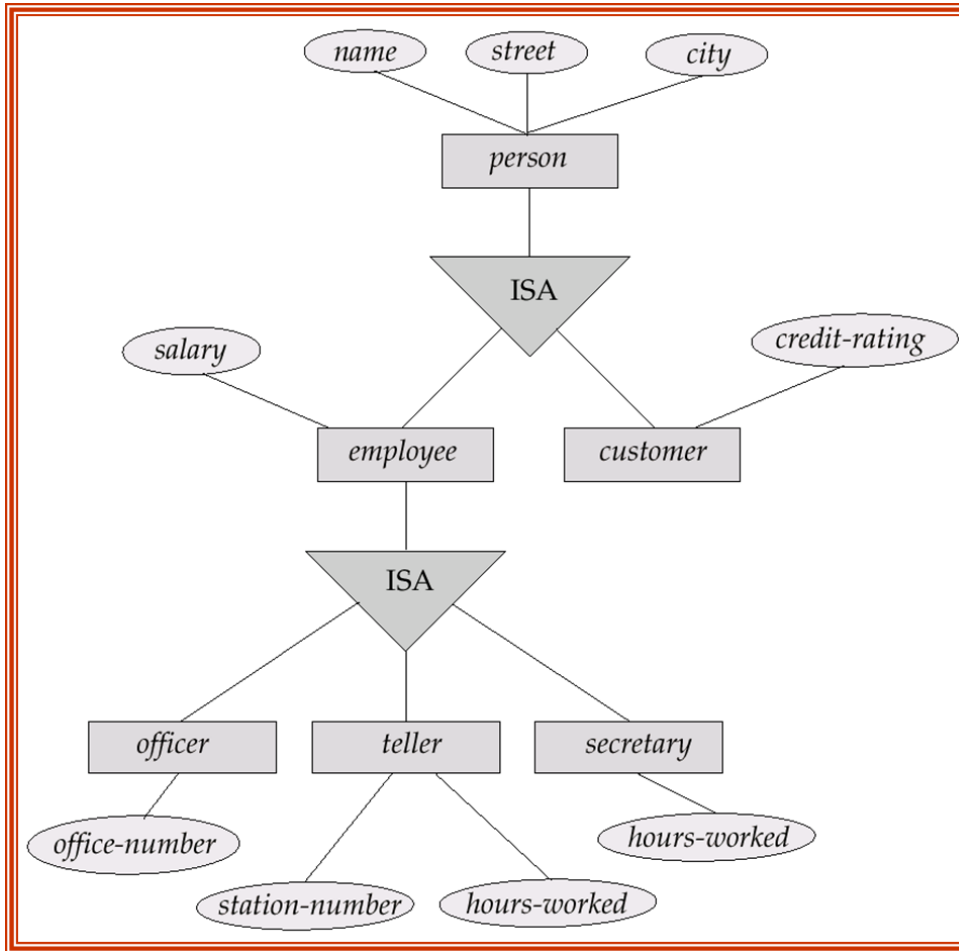
- **Regla CED:** un CE débiles se mapea a una tabla que incluye columnas para la clave primaria del CE identificador más los atributos (no multi-valorados) del CE débiles (achataando jerarquías de atributos compuestos si es necesario).
  - La clave primaria del CE identificador más el discriminador del CE débil forman la clave primaria del esquema relacional de la traducción.
  - Para atributos de esquema de CE débil que provienen de CE identificadora se agrega restricción de clave foránea desde esquema de CE débil a CE identificador.
  - El CR identificador no se mapea.



# Reducción a Esquemas Relacionales

- Para traducir generalización iremos proponiendo y evaluando ***ideas de traducción***.
- Primero consideramos la situación más fácil donde el CE generalización no está relacionado con otros CE.

# Reducción a Esquemas Relacionales



*Primera idea de mapeo de los CE que participan en generalización.*

*person(name, street, city)*  
*customer(name, credit-rating)*  
*Employee(name, salary)*

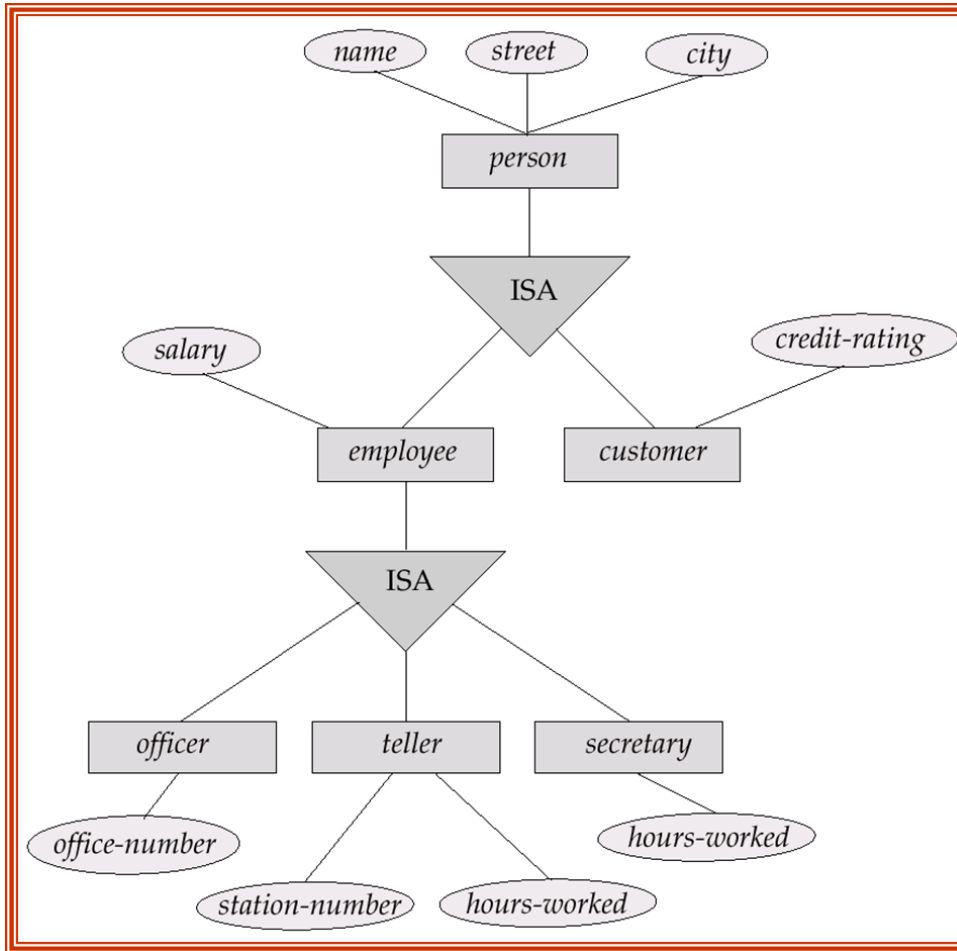
# Reducción a Esquemas Relacionales

- **Idea G1:** traducción de generalización a tablas.
  - Formar una tabla para el CE de nivel más alto (la generalización)
  - Formar una tabla para cada CE especialización que incluye la clave primaria del CE generalización y los atributos locales.

# Reducción a Esquemas Relacionales

- *Evaluación de Idea G1*: obtener toda la información de CE de especialización requiere acceder a dos tablas.
- *¿Cómo evitar tener que acceder a dos tablas?*
- **Idea G2**: formar una tabla para cada CE especialización con los atributos locales y heredados del CE generalización.
  - Es más eficiente tener todo en una tabla.

# Reducción a Esquemas Relacionales



*Usando la idea G2 los CE se mapean a:*

*person(name, street, city)*  
*customer(name, street, city, credit-rating)*  
*Employee(name, street, city, salary)*

# Reducción a Esquemas Relacionales

- *Evaluación de idea G2*: si dos o más especializaciones contienen la misma entidad, entonces los atributos que no son clave primaria del CE generalización son almacenados redundantemente.
- *Si generalización es total se puede evitar ese tipo de redundancia.*
- Porque la tabla del CE generalización no se requiere para almacenar información.

# Reducción a Esquemas Relacionales

- **Idea G3**: Si la generalización es total formar una tabla para cada CE especialización con los atributos locales y heredados; no formar tabla para la generalización.
- *La idea G3 es óptima cuando*
  - la generalización es total y disjunta.

# Reducción a Esquemas Relacionales

- **Resumiendo:** Cuando el CE generalización no está relacionado con otros CE: ¿Cómo conviene hacer la traducción?
- **Respuesta:**
  - Generalización total y disjunta: usar idea G3.
  - Generalización no total y disjunta: usar idea G2.
    - No va a haber redundancia entre los CE especializaciones.
    - No hay que consultar dos tablas para obtener toda la información de una especialización.
  - Generalización no disjunta: usar idea G1
- Tener en cuenta igual que hay pros y contras con cada propuesta.
  - En la práctica es necesario contar con conocimiento del problema puntual a resolver.

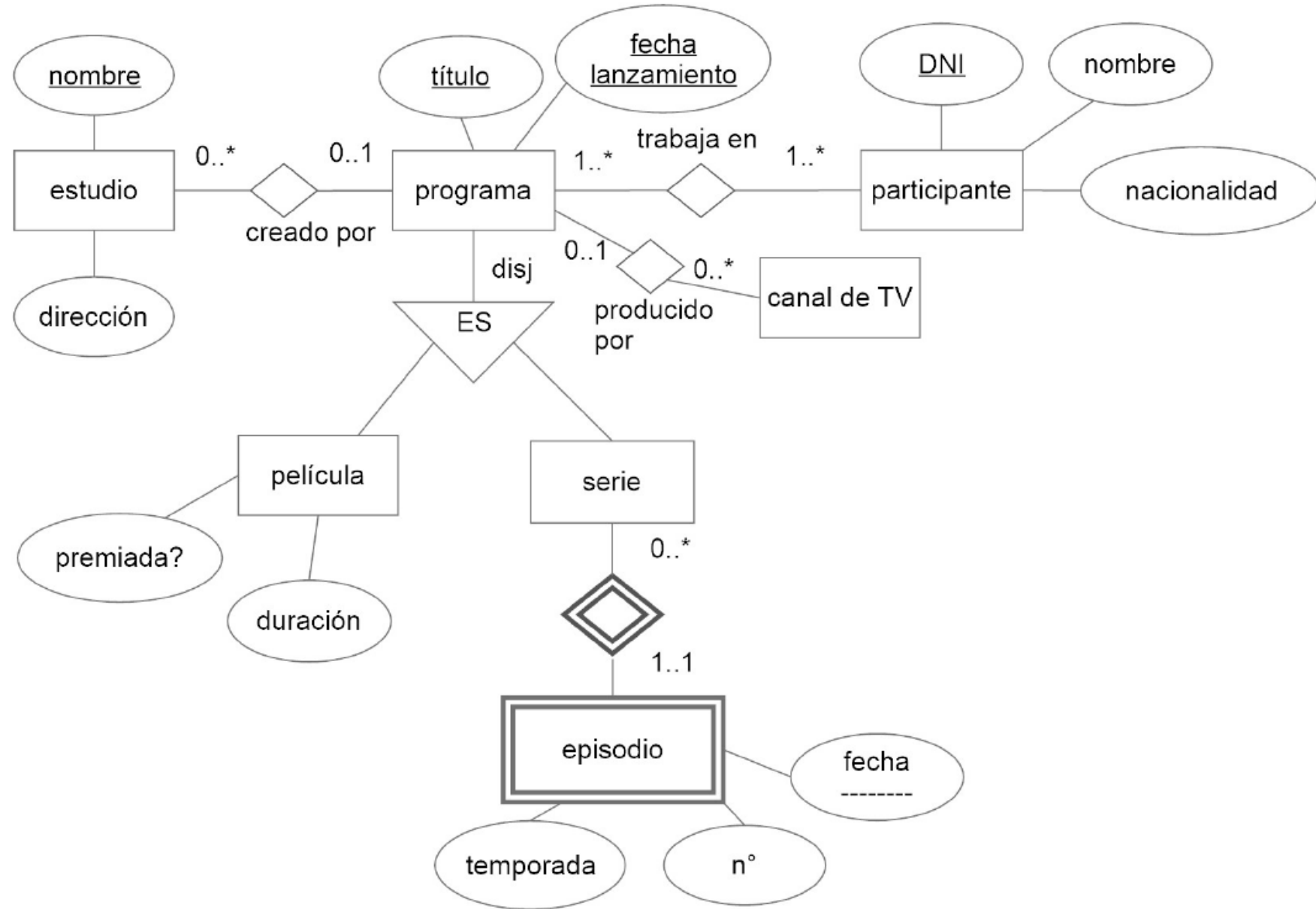


# Reducción a Esquemas Relacionales

- Ahora hacemos ejemplos donde un CE generalización está relacionado por CR con al menos un CE.
- **Ejemplo 1:** BD de programas de TV.
  - Se tiene una BD de programas de TV; los estudios son estudios cinematográficos que crean películas o series. Un programa puede ser producido por un canal de TV (por ej. Para shows, para programas de noticias, etc.), o ser creado por un estudio cinematográfico (por ej. para películas o series).

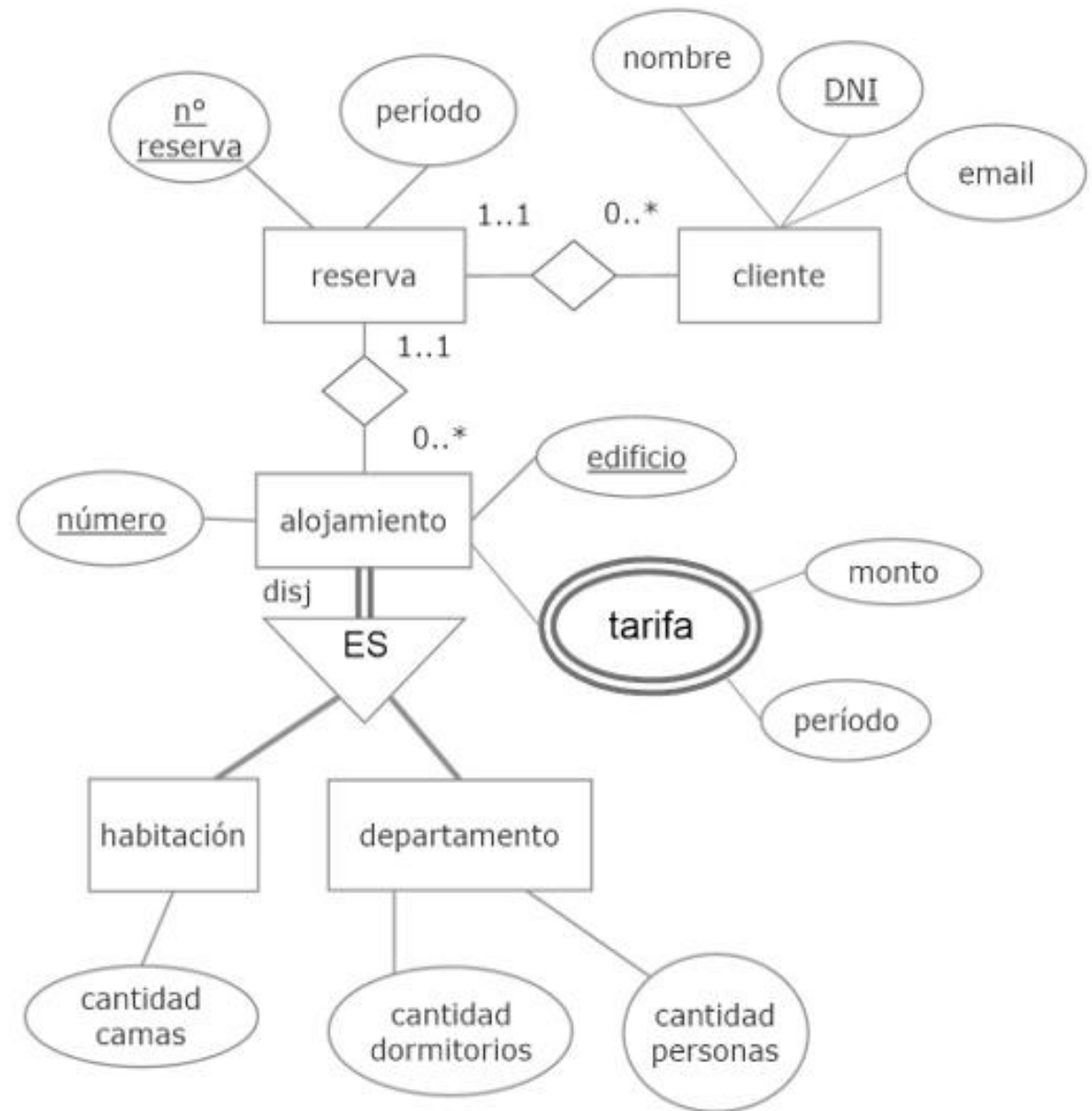
# Reducción a Esquemas Relacionales

- Como programa está relacionado con 3 CE conviene traducirlo a una tabla:
- Programa(título, fecha lanzamiento, nombre, nombre canal TV)
- Además tabla de *programa* contiene *nombre* y *nombre* de canal de TV; esto es para reflejar *creado por* y *producido por*.
- Película (título, fecha lanzamiento, duración, premiada?).
- Serie(título, fecha lanzamiento)
- Completar lo que falta.



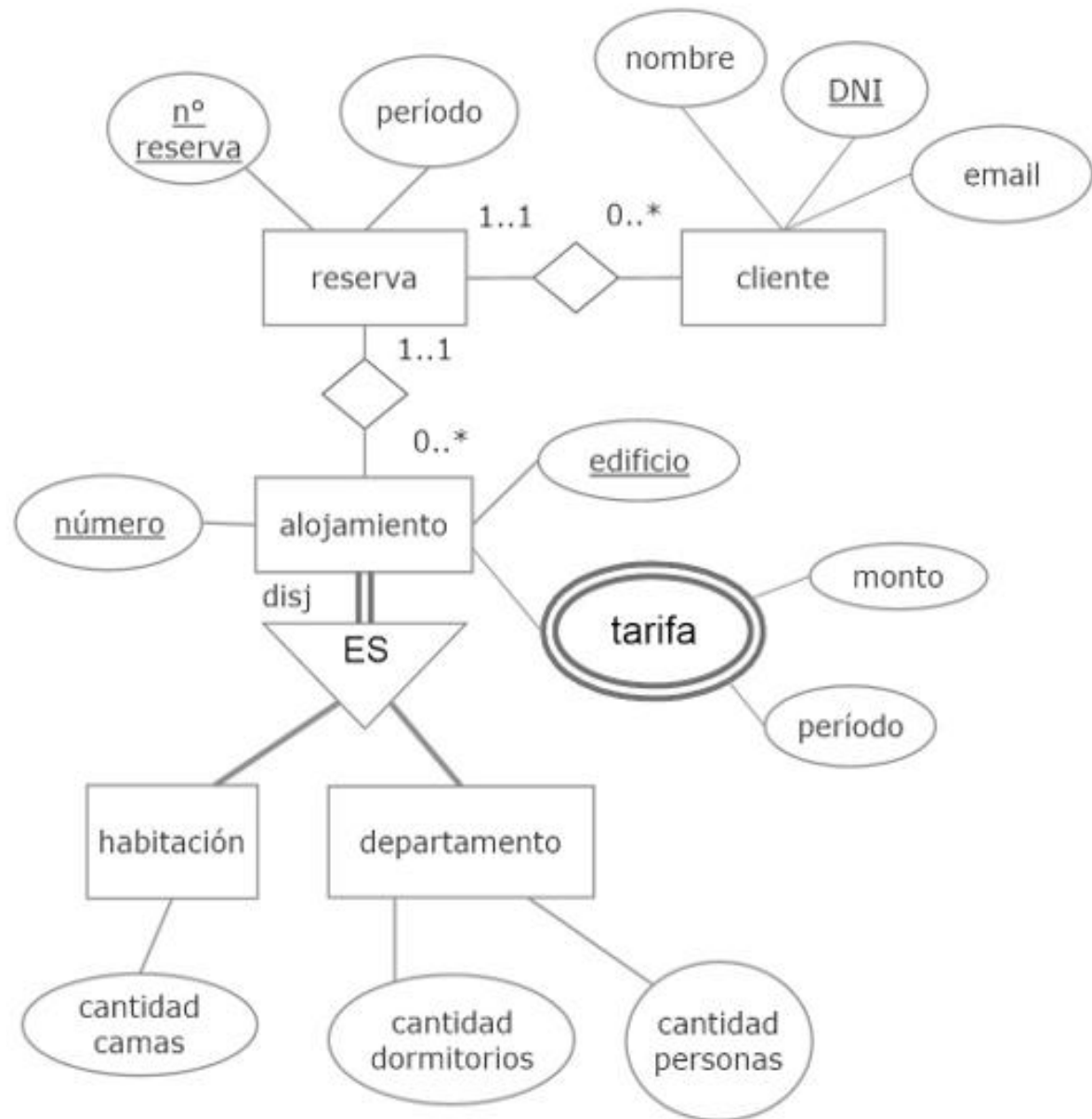
# Reducción a Esquemas Relacionales

- **Ejemplo 2:** BD de complejo turístico



# Reducción a Esquemas Relacionales

- Necesito tabla de *alojamiento* porque está relacionada con *reserva* y necesito reflejar atributo multi-valorado.
- alojamiento(número, edificio)
- tarifa(monto, período, número, edificio)
- reserva(n° reserva, período, DNI, número, edificio)
- departamento(número, edificio, cantidad dormitorios, cantidad personas)
- habitación(número, edificio, cantidad camas)
- Completar lo que está faltando.



# Reducción a Esquemas Relacionales

- Otra forma de resolver el ejercicio anterior es agregar más atributos a *alojamiento* y sacar tablas *habitación* y *departamento*:
  - *alojamiento*(número, edificio, tipo, cantidad camas, cantidad dormitorios, cantidad personas)
- El *tipo* puede ser: *habitación* o *departamento*.
- Si tipo *habitación* los campos de departamento *cantidad de dormitorios* y *cantidad personas* son nulos; si tipo es *departamento*, el campo *cantidad camas* es nulo.