MÓDULO

Bases de Datos

UNIDAD 3

Diseño Físico de Bases de Datos

ÍNDICE

- Introducción
- Modelo Relacional. Definiciones, contenido y notación.
- Obtención del modelo relacional.

Introducción

Edgar Frank Codd definió las bases del modelo relacional a finales de los 60.

Lo que intentaba era evitar que los usuarios de las bases de datos, tuvieran que verse obligadas a aprender los entresijos internos del sistema. Pretendía que los usuarios trabajaran de forma sencilla e independientemente del funcionamiento físico de la base de datos en sí.

Fue un enfoque revolucionario.

El modelo relacional

Los objetivos que perseguía con su modelo eran:

- ✓ **Independencia física**. La forma de almacenar los datos, no debe influir en su manipulación lógica. Si la forma de almacenar los datos cambia, lo usuarios no tienen que verse afectados en su trabajo. Esto permite que los usuarios se concentre en qué quieren consultar en la base de datos, en lugar de cómo consultarlo.

El modelo relacional

- ✓ **Independencia lógica.** Las aplicaciones que utilizan la base de datos, no deben ser modificadas porque 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.

Estructura de las bases de datos relacionales

El elemento fundamental es lo que se conoce como relación, aunque más habitualmente se le llama **tabla**.

Codd definió las relaciones utilizando un lenguaje matemático, pero se pueden asociar a la idea de tabla (de filas y columnas) ya que es más fácil de entender.

Estructura de las bases de datos relacionales

Las relaciones constan de:

- Atributos. Cada una de las propiedades de los datos que se almacenan en la relación (nombre, dni,…)
- Tuplas. Cada elemento de la relación. Por ejemplo si se tiene una relación que almacena personas, una tupla representaría cada una de las personas almacenadas (Ana, Pedro, Carmen,…).

Puesto que una relación se representa como una tabla, se puede decir que las columnas de las tablas son los atributos y las filas, las tuplas.

Estructura de las bases de datos relacionales

Tupla

Es cada una de las filas de la relación. Se corresponde con la idea clásica de **registro**. Representa por tanto cada elemento individual de esa relación. Tiene que cumplir que:

- Cada tupla se debe corresponder con un elemento del mundo real.
- No puede haber dos tuplas iguales (con todos los valores iguales).

Estructura de las bases de datos relacionales

Dominio

Un dominio contiene todos los posibles valores que puede tomar un determinado atributo. Dos atributos distintos pueden tener el mismo dominio.

Es un conjunto finito de valores del mismo tipo. A los dominios se les asigna un nombre y así podemos referirnos a ese nombre en más de un atributo.

Estructura de las bases de datos relacionales

Grado

Indica el tamaño de una relación en base al número de columnas (atributos) de la misma. Lógicamente cuanto mayor es el grado de una relación, mayor es su complejidad al manejarla.

Cardinalidad

Número de tuplas de una relación

Propiedades de las tablas

Las tablas deben cumplir una serie de propiedades:

- Cada tabla tiene un nombre distinto.
- Cada atributo de la tabla toma un solo valor en cada tupla.
- Los nombres de los atributos de una tabla son distintos, aunque pueden repetirse en tablas diferentes.
- Cada tupla es única. No hay tuplas duplicadas.
- El orden de los atributos, no importa.
- El orden de las tuplas, no importa.

Claves

Clave candidata

Conjunto de atributos que identifican unívocamente cada tupla de una relación.

Son las columnas cuyos valores no se repiten en ninguna otra tupla de esa tabla.

Toda tabla en el modelo relacional debe tener al menos una clave candidata.

Claves

Clave primaria

Clave candidata que se escoge como identificador de las tuplas. Se elige la que identifique mejor a cada tupla en el contexto de la base de datos.

Por ejemplo un campo *DNI* sería clave candidata en una tabla de clientes, pero si tiene un campo *código cliente*, éste sería mejor como clave primaria.

Clave alternativa

Cualquier clave candidata que no sea primaria

Claves

Clave ajena

Son los datos de atributos de una tabla cuyos valores están relacionados con atributos de otra tabla.

Estos atributos sirven para relacionar unas tablas con otras

Claves

Profesor	Código Profesor
Ángel	1
Martina	2
Carmen	3

Expediente	Alumno	Cod_Prof
175	Andrés	2
102	Sofía	3
305	Pablo	2
419	Julia	1
533	Inés	1

El atributo Cod_Prof en la segunda tabla, es una clave ajena. Sirve para relacionar el alumno con el profesor.

Semántica

En el modelo relacional existen ciertas reglas semánticas

- ✓ Clave principal (primary key). Marca uno o más atributos como identificadores de la tabla.
- ✓ Unicidad (unique). Impide que los valores de los atributos marcados de esa forma puedan repetirse. Esta restricción debe indicarse en todas las claves alternativas.
- ✓ Obligatoriedad (not null). Prohíbe que el atributo marcado de esta forma quede vacío, es decir, que pueda contener un valor nulo.

Semántica

- ✓ Integridad referencial (foreign key). Sirve para indicar una clave ajena. Sólo pueden contener valores que coincidan con la clave principal de la tabla que relacionan.

Semántica

En principio las entidades fuertes del modelo E/R son transformadas al modelo relacional siguiendo estas instrucciones:

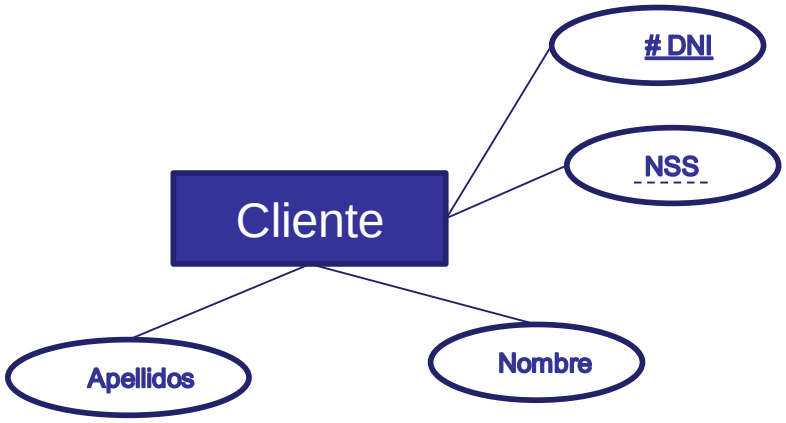
-Entidades. Pasan a ser tablas.

-Atributos. Se convierten en columnas o atributos de tabla.

-Identificadores principales. Pasan a ser claves primarias.

-Identificadores candidatos. Pasan a ser claves candidatas.

Transformación de las entidades fuertes



Modelo E/R

Cliente (DNI, NSS, Nombre, Apellidos)
CP {DNI}
UNI{NSS}
VNN{NSS}

Modelo relacional

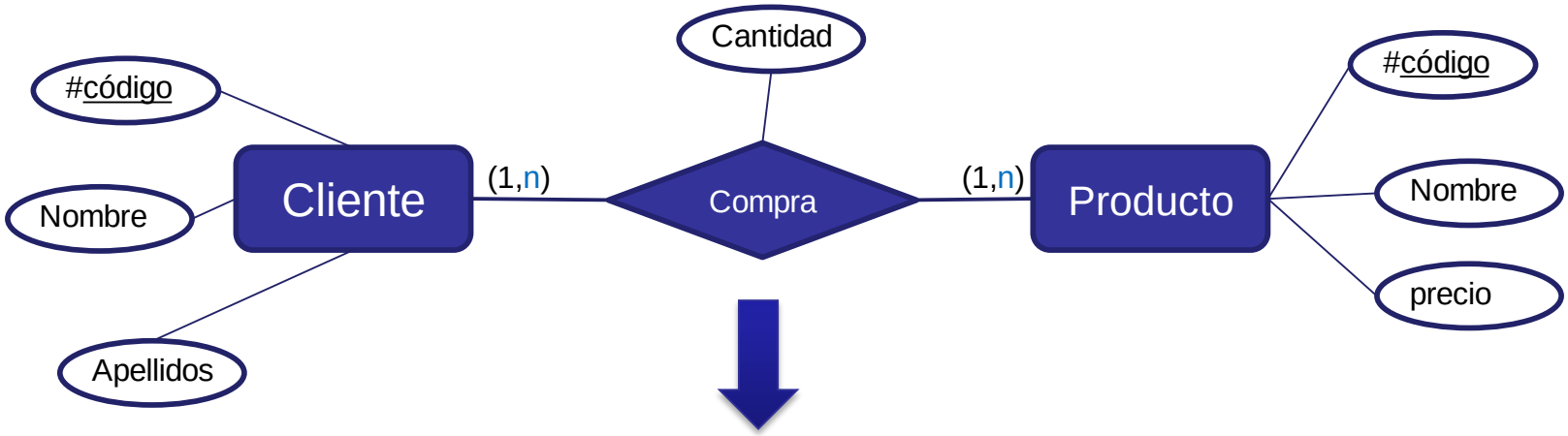
Transformación de las relaciones

Hay que tener en cuenta todos los casos.

- **Relaciones en las que dos o más entidades tienen cardinalidad máxima N**. El proceso de transformación obliga a crear una nueva tabla que contendrá las claves primarias de las entidades que participan en la relación. Éstas, actuarán de clave principal de la nueva tabla.

  Además todos estos atributos importados, serán claves ajenas respecto a las entidades de las que provienen. Si la relación tenía atributos, éstos formarán parte de la tabla.

Transformación de las relaciones. Cardinalidad máxima N a N



Cliente (código, nombre, apellidos)

Producto (código, nombre,precio)

Compra(código_cliente, código_producto, cantidad)

Transformación de las relaciones. Cardinalidad máxima N a N

En este caso **surgen 3 nuevas tablas:**

- **Cliente** (código, nombre, apellidos,…)

  CP {código} → *Clave Principal*

- **Producto** (código, nombre, precio,…)

  CP {código} → *Clave Principal*

- **Compra**(código_cliente, código_producto, cantidad)

  CP {código_cliente, código_producto} → *Clave Principal*

  CAj {código_cliente} referencia a Cliente → *Clave Ajena*

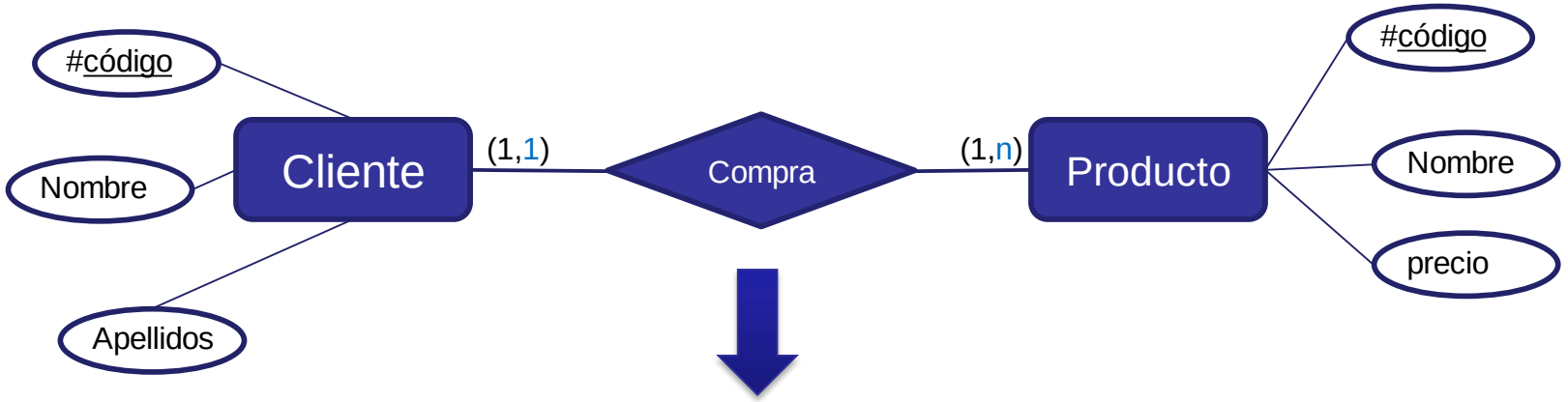
  CAj {código_producto} referencia a Producto → *Clave Ajena*

Transformación de las relaciones

- **Relaciones en las que sólo hay una entidad con cardinalidad máxima N**. Son las relaciones de (1,N) y en este caso no se crea ninguna entidad nueva. Simplemente, a la tabla que tiene cardinalidad N, se le añadirán el atributo clave principal (que actuarán como clave ajena) de la entidad con cardinalidad 1.

Transformación de las relaciones. Cardinalidad máxima 1 a N

- **Relaciones en las que sólo hay una entidad con cardinalidad N.**



Cliente (código, nombre, apellidos)
CP{código}

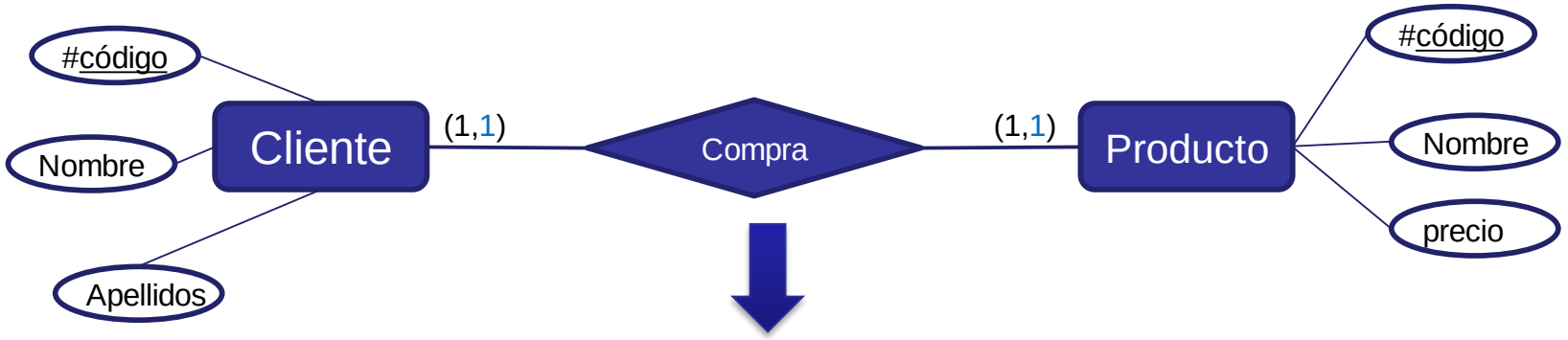
Producto (código, nombre, precio, código_cliente)
CP {código}
CAj {código_cliente} referencia a Cliente

Transformación de las relaciones

- **Relaciones en las que todas las entidades participan con cardinalidad 1 como máximo:** en este caso se procede como en el caso de (1,N) con la diferencia de que sería indiferente en qué entidad añadimos la clave ajena.

Transformación de las relaciones. Cardinalidad máxima 1 a 1

Opción 1

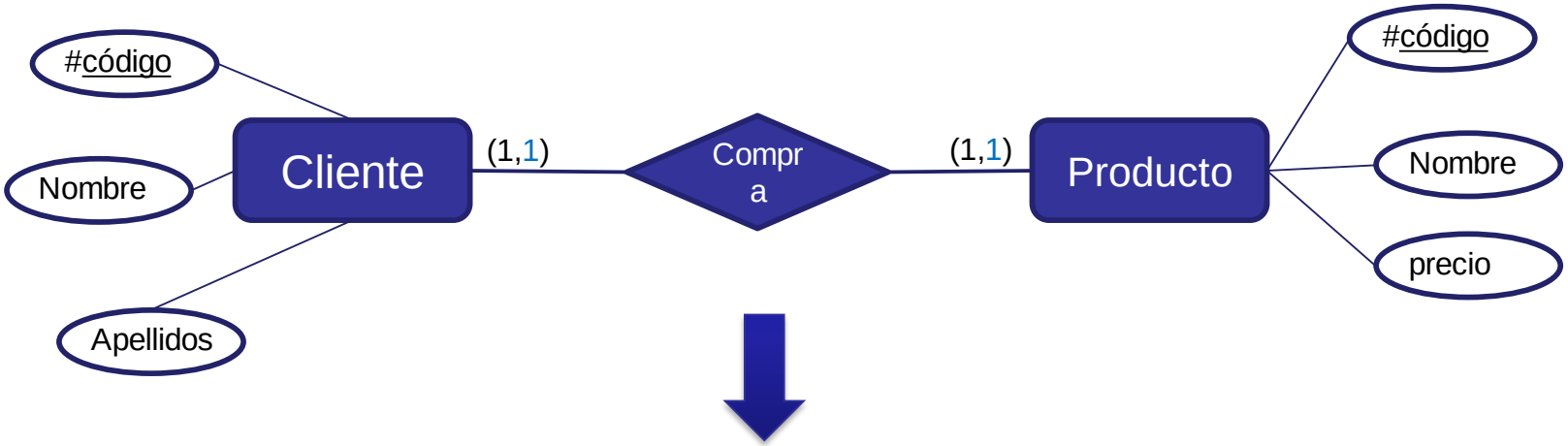


Cliente (código, nombre, apellidos)
CP{código}

Producto (código, nombre, precio, código_cliente)
CP {código}
CAj {código_cliente} referencia a Cliente
UNI {código_cliente}

Transformación de las relaciones. Cardinalidad máxima 1 a 1

Opción 1



Cliente (código, nombre, apellidos, codigo_producto)
CP{código}
CAj {código_producto} referencia a Producto
UNI {código_producto}

Producto (código, nombre, precio)
CP {código}

Transformación de las relaciones

- **Relaciones recursivas:** se tratan de la misma forma que las otras, solo que un mismo atributo puede figurar dos veces en una tabla como resultado de la transformación.

  Aunque se trate del mismo atributo, no podrá utilizar el mismo nombre.

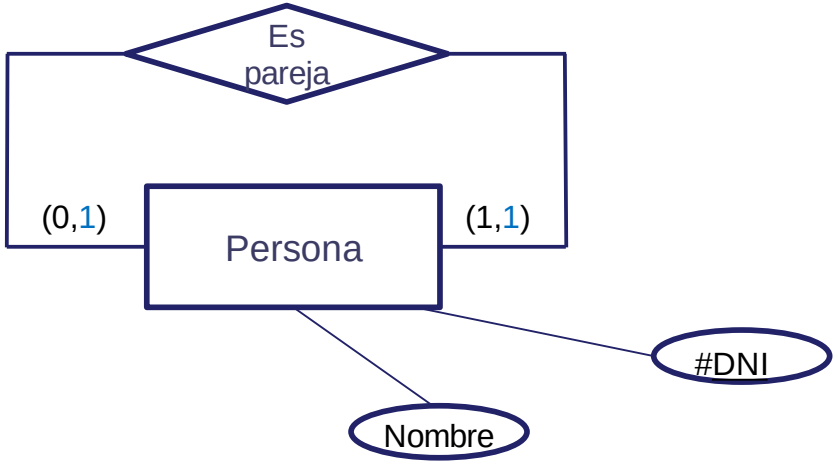
Transformación de las relaciones. Relaciones recursivas

Persona (DNI, nombre, DNI_pareja)

CP{DNI}

CAj{DNI_pareja} referencia a Persona

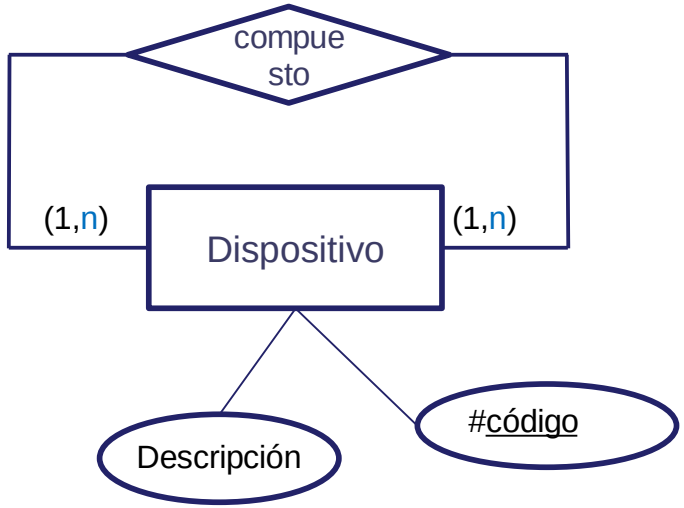
UNI {DNI_pareja}



Transformación de las relaciones

Dispositivo (código, Descripción)

CP{código}



Compuesto (cod1, cod2)

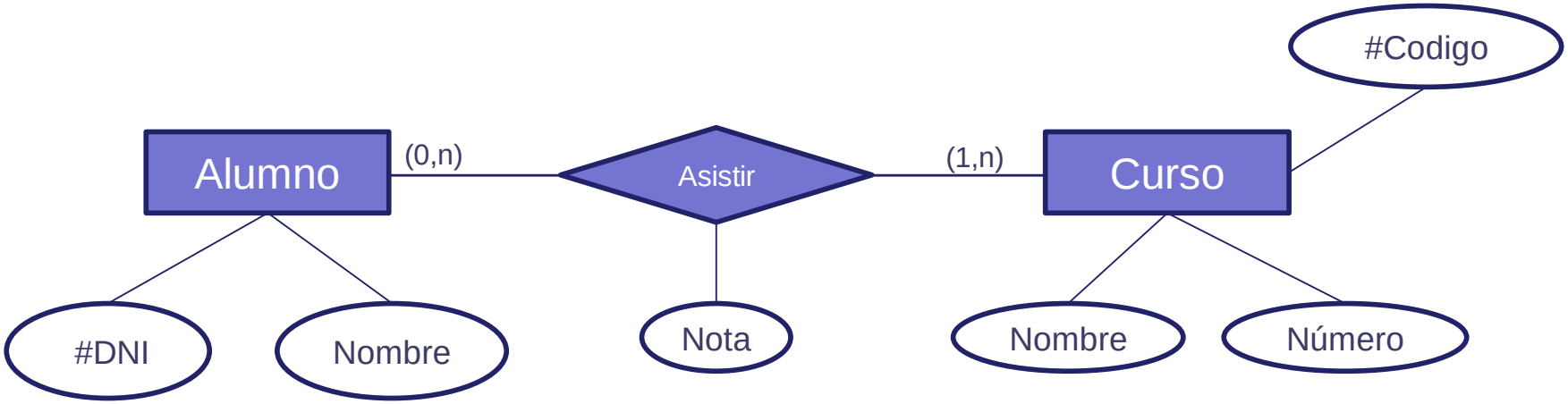
CP {cod1, cod2}

CAj {cod1} referencia a Dispositivo

CAj {cod2} referencia a Dispositivo

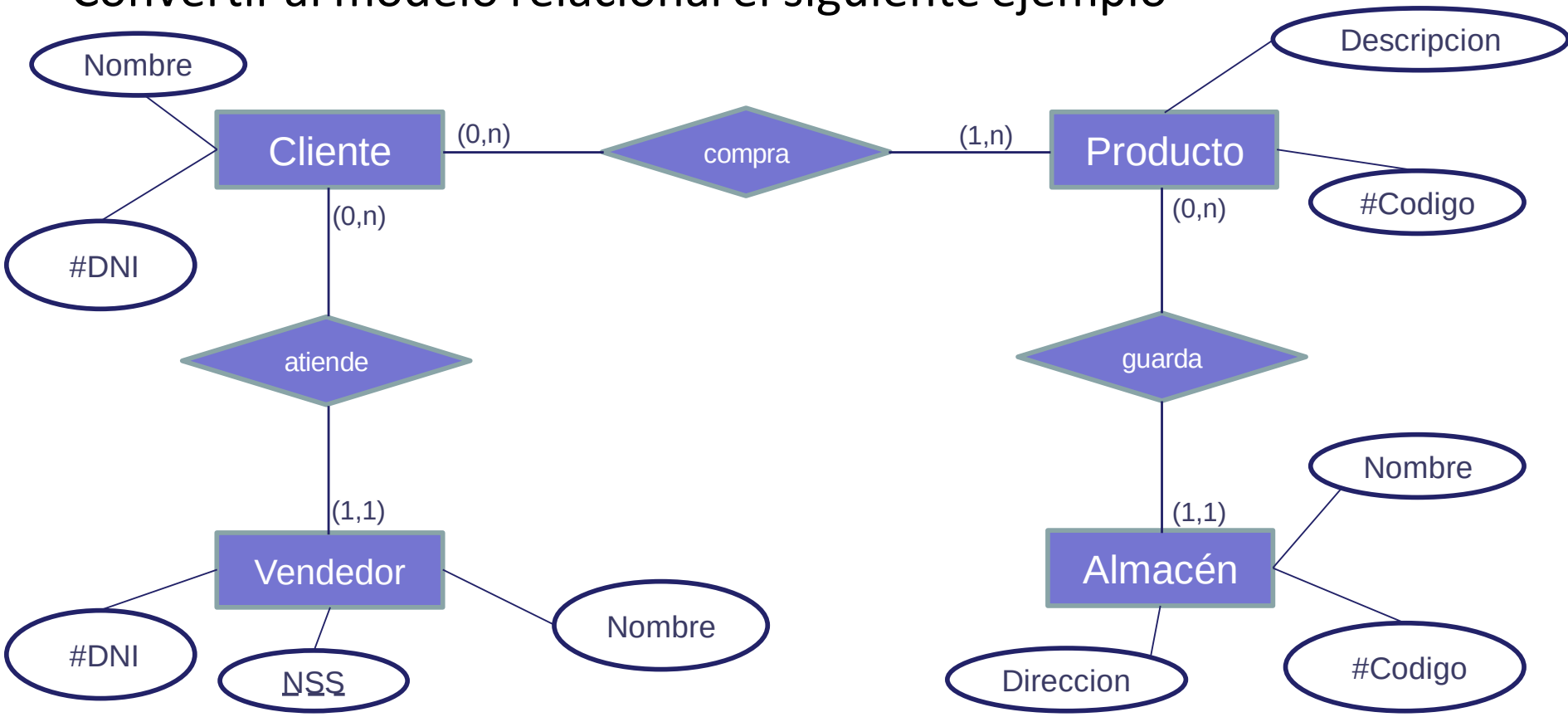
Ejercicio

Convertir al modelo relacional el siguiente ejemplo



Ejercicio

Convertir al modelo relacional el siguiente ejemplo



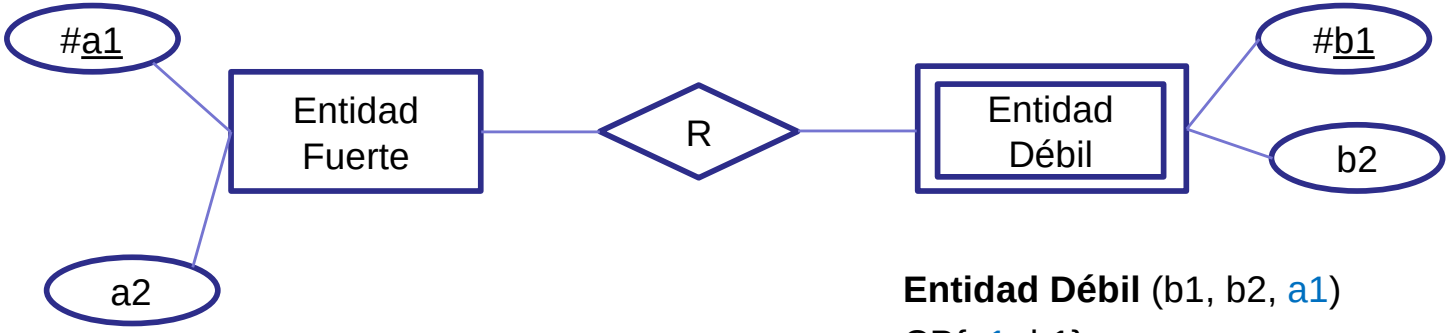
Ejercicio

Convertir al modelo relacional el ejercicio de zoológico.

Transformación de las entidades débiles

Toda entidad débil incorpora una relación implícita con una entidad fuerte.

Esta relación no necesita incorporarse como una nueva tabla en el modelo relacional. Basta con añadir como atributo y clave ajena en la entidad débil, el identificador de la fuerte.



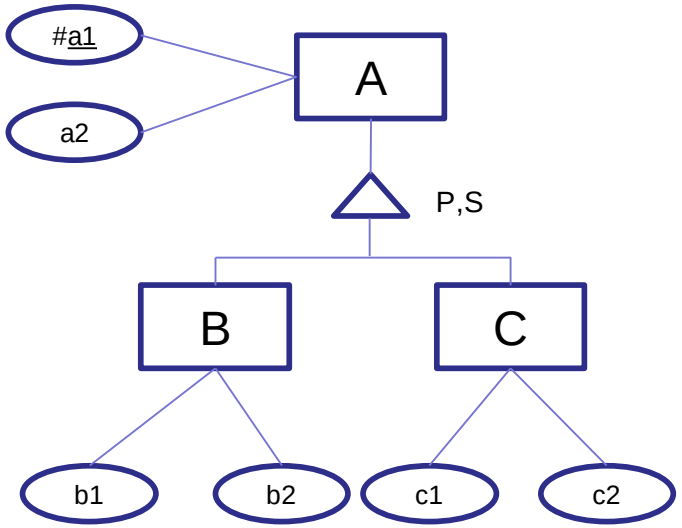
Entidad Fuerte (a1, a2)
CP{a1}

Entidad Débil (b1, b2, a1)
CP{a1, b1}
CAj {a1} referencia a Entidad Fuerte

Transformación de la Generalización/Especialización

Sólo cuando la especialización/generalización es parcial y solapada existe una transformación en relaciones totalmente adecuada.

En los demás casos es necesario incluir ciertas restricciones que no se verán.



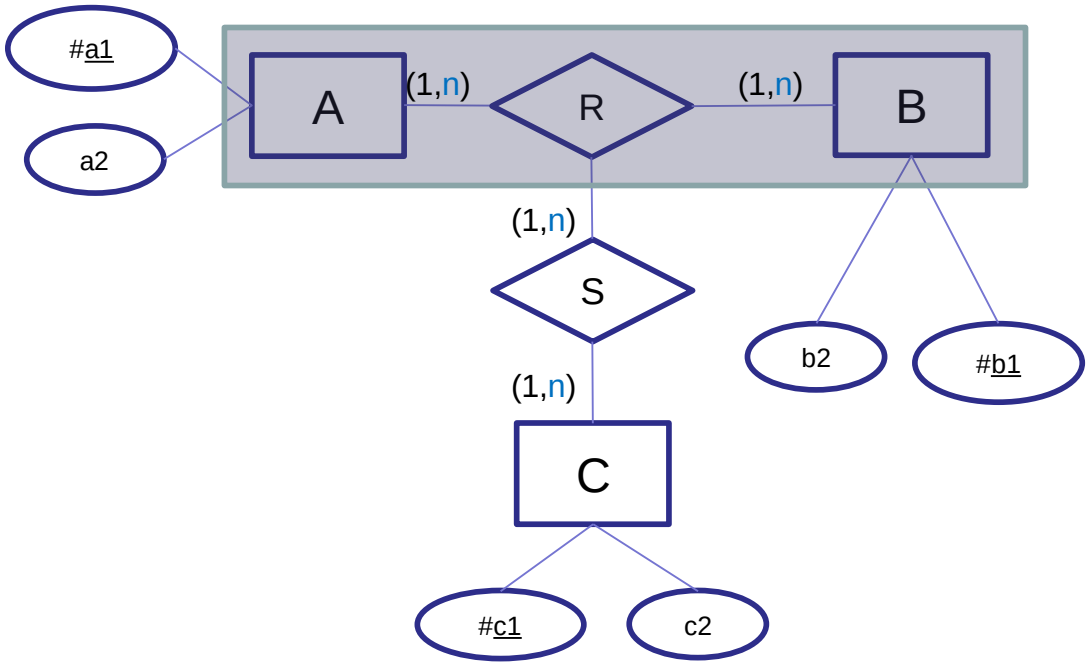
A(a1,a2)
CP{a1}

B (b1, b2, a1)
CP{a1}
CAj {a1} hace referencia a A

C (c1, c2, a1)
CP{a1}
CAj {a1} hace referencia a A

Transformación de la agregación

La agregación considera las entidades relacionadas por una relación junto con ésta, como una entidad.



A(a1,a2) **B** (b1, b2) **C** (c1, c2)

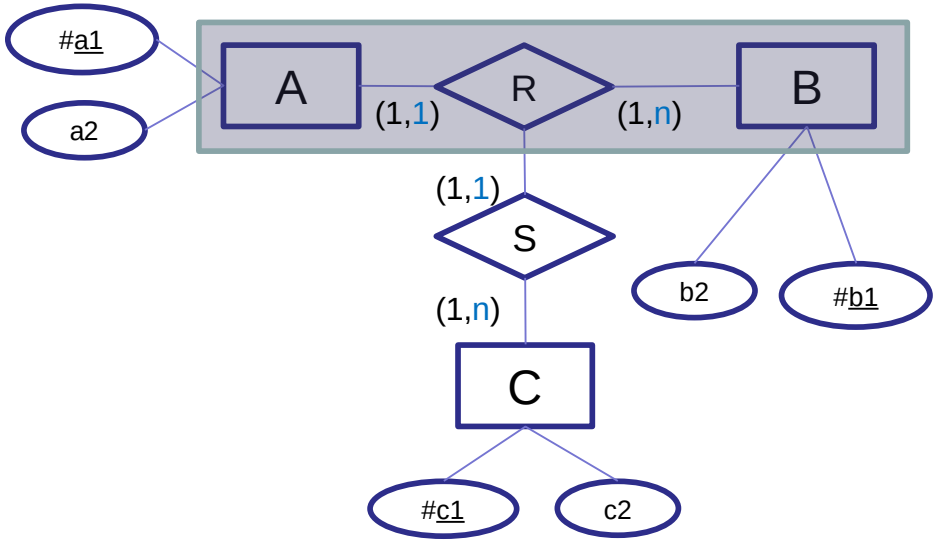
CP{a1} CP{b1} CP{c1}

R (a1, b1)
CP{a1,b1}
CAj{a1} hace referencia a A
CAj{b1} hace referencia a B

S (c1, a1, b1)
CP{c1, a1, b1}
CAj {a1, b1} hace referencia a R
Caj {c1} hace referencia a C

Transformación de la agregación

La agregación considera las entidades relacionadas por una relación junto con ésta, como una entidad.



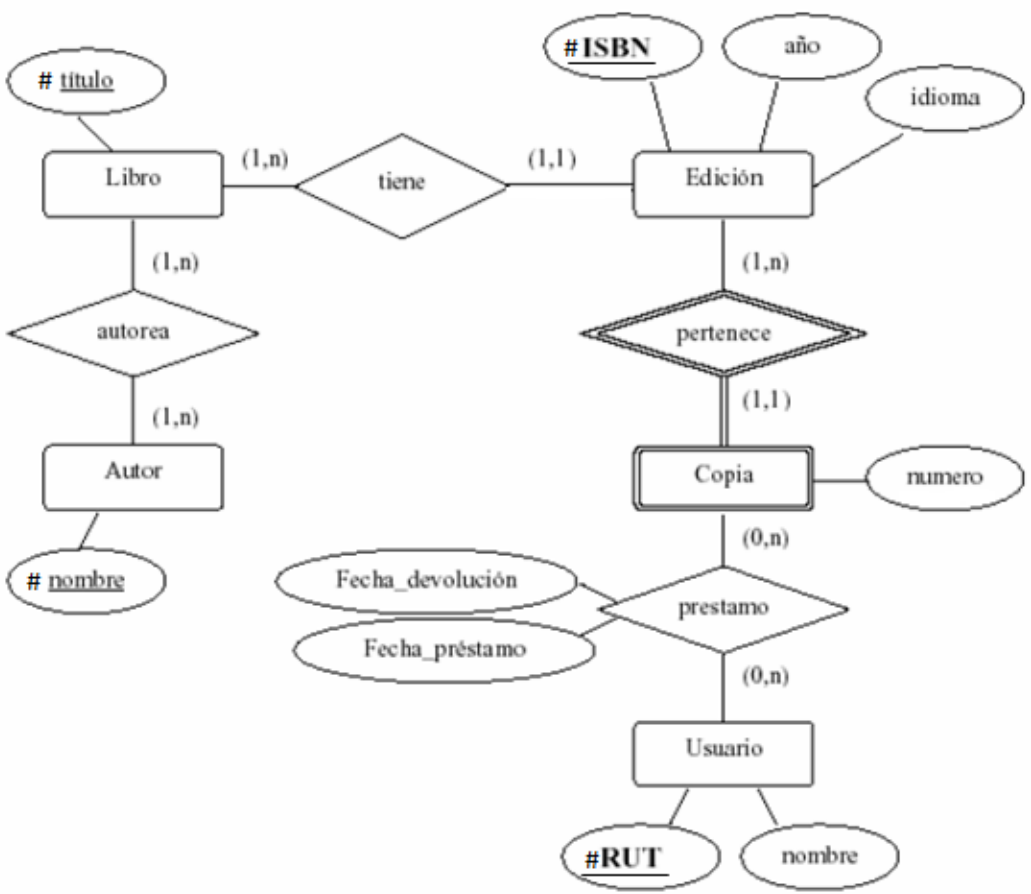
A(a1,a2)
CP{a1}

B (b1, b2, a1)
CP{b1}
CAj {a1} referencia a A

C (c1, c2, b1)
CP{c1}
CAj {b1} referencia a B

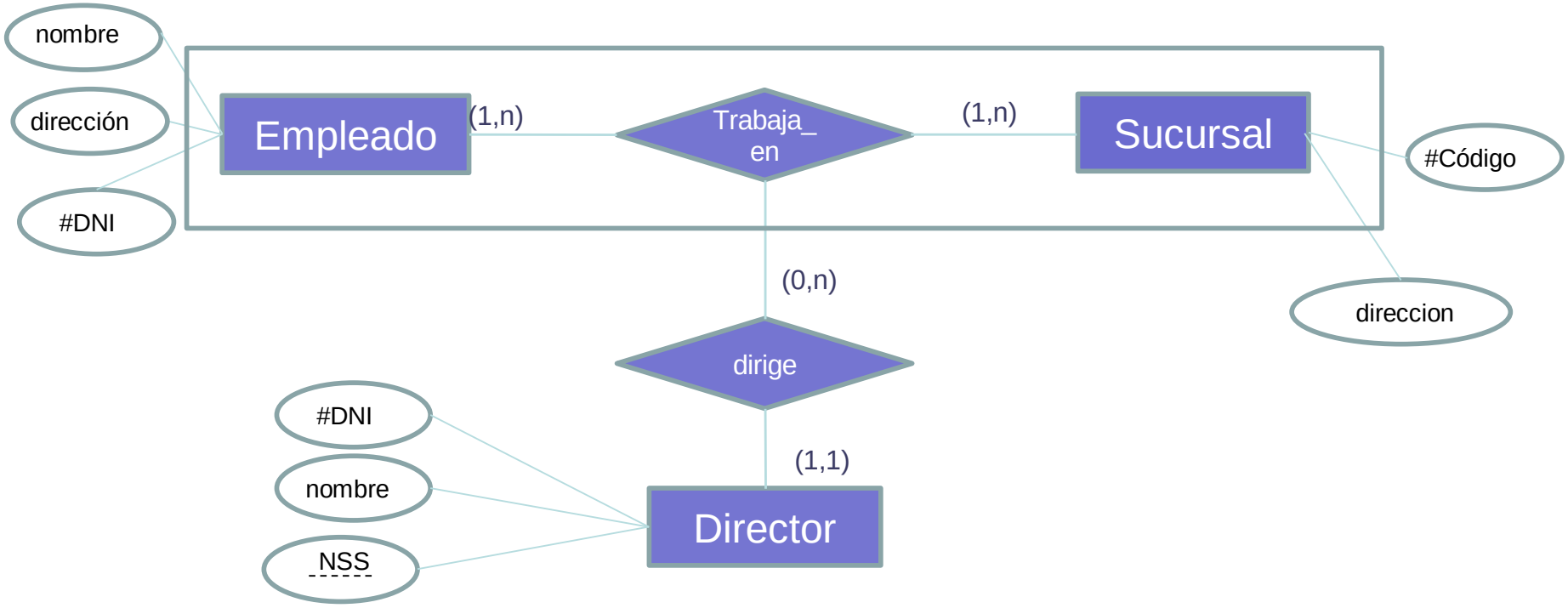
Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación



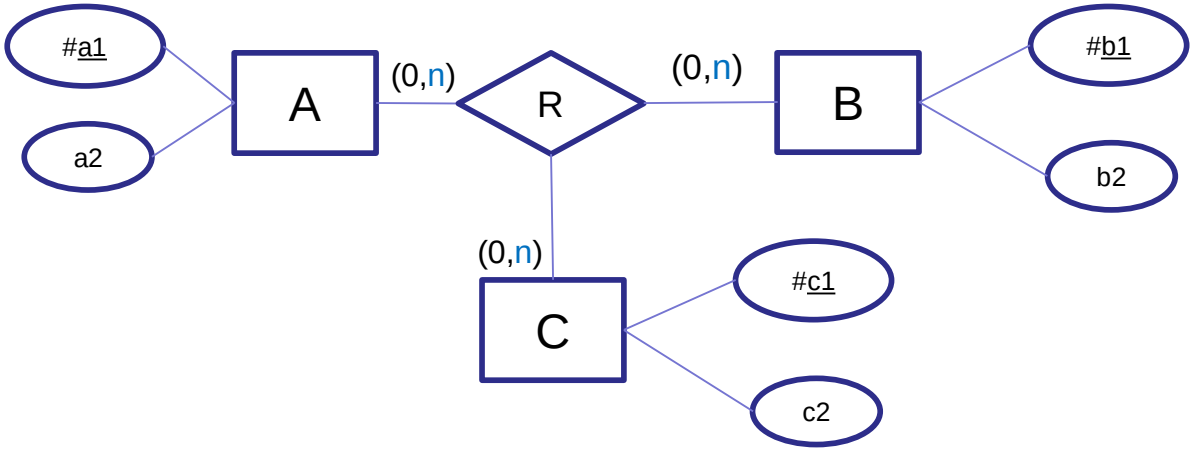
Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación.



Transformación de relaciones ternarias

Relaciones ternarias N:N:N



A(a1,a2)
CP{a1}

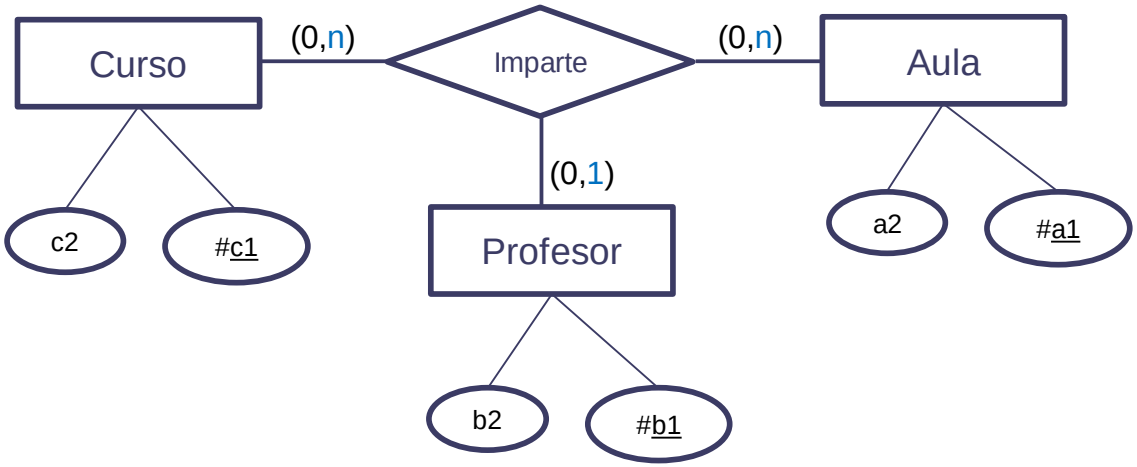
B (b1, b2)
CP{b1}

C (c1, c2)
CP {c1}

R(a1,b1,c1)
CP{a1, b1, c1}
CAj{a1} Referencia a A
CAj{b1} Referencia a B
CAj{c1} Referencia a C

Transformación de relaciones ternarias

Relaciones ternarias 1:N:N

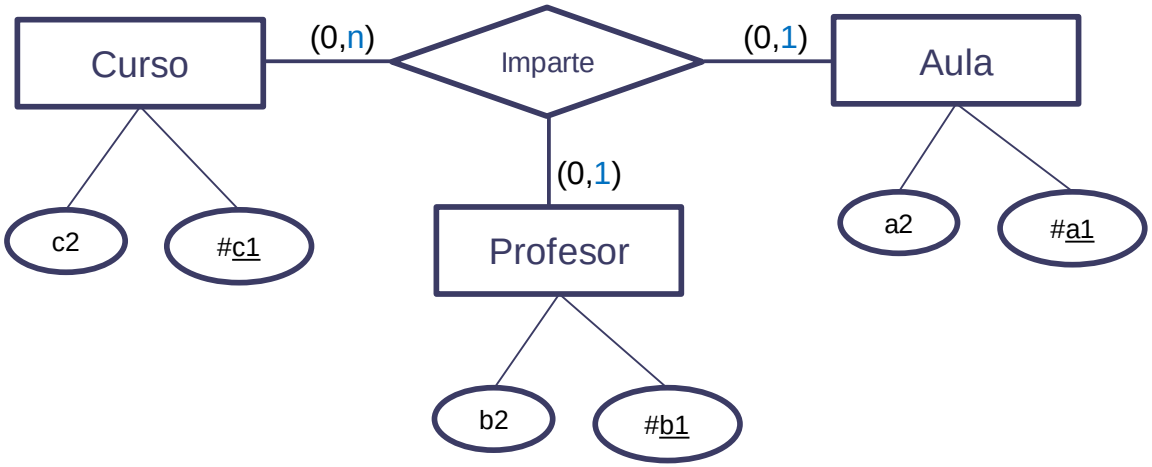


Regla General: se deberá definir una restricción de VNN sobre todo atributo correspondiente a una CAj que no esté presente en la CP de la relación que representa a la relación ternaria

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{a1, c1}
VNN {b1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

Transformación de relaciones ternarias

Relaciones ternarias 1:1:N. Opción 1

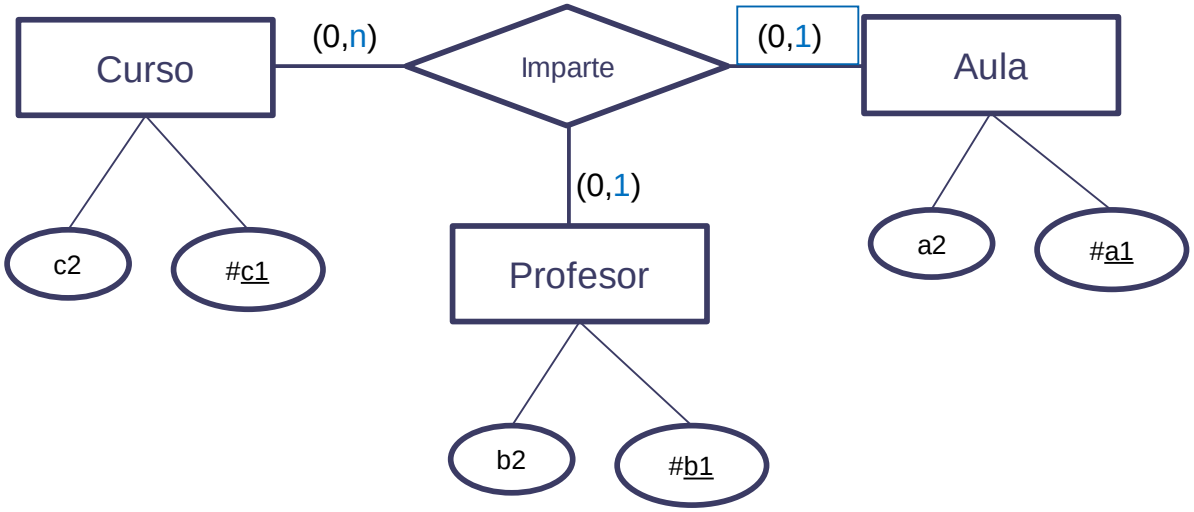


Debido a la presencia de dos cardinalidades máximas 1, existen dos claves candidatas a CP. Para la clave alternativa tendrá que definirse las restricciones de unicidad y VNN.

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{b1, c1}
VNN {a1}
UNI {a1, c1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

Transformación de relaciones ternarias

Relaciones ternarias 1:1:N. Opción 2

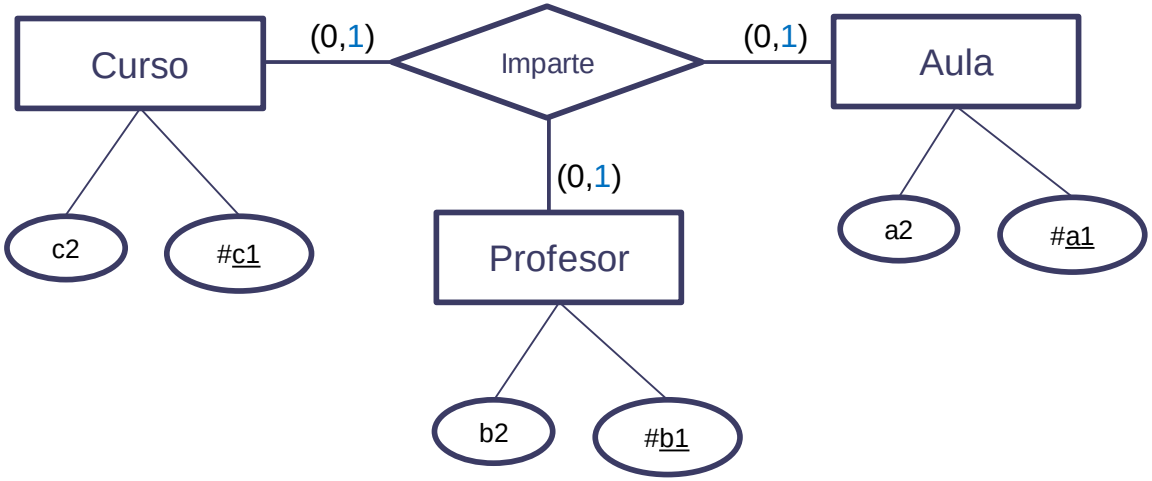


Debido a la presencia de dos cardinalidades máximas 1, existen dos claves candidatas a CP. Para la clave alternativa tendrá que definirse las restricciones de unicidad y VNN.

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{a1, c1}
VNN {b1}
UNI {b1, c1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

Transformación de relaciones ternarias

Relaciones ternarias 1:1:1. Opción 1

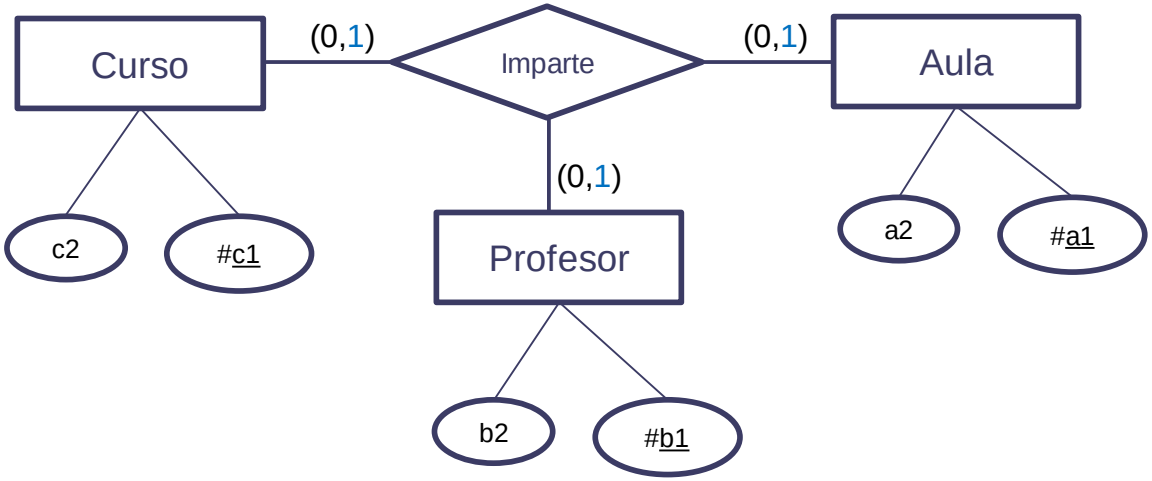


Existen tres cardinalidades máximas 1, por lo que hay tres claves candidatas para la relación R; cualquiera puede ser CP teniendo que definirse para las otras dos una restricción de unicidad.

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{a1, b1}
VNN {c1}
UNI {a1, c1}
UNI {b1, c1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

Transformación de relaciones ternarias

Relaciones ternarias 1:1:1. Opción 2

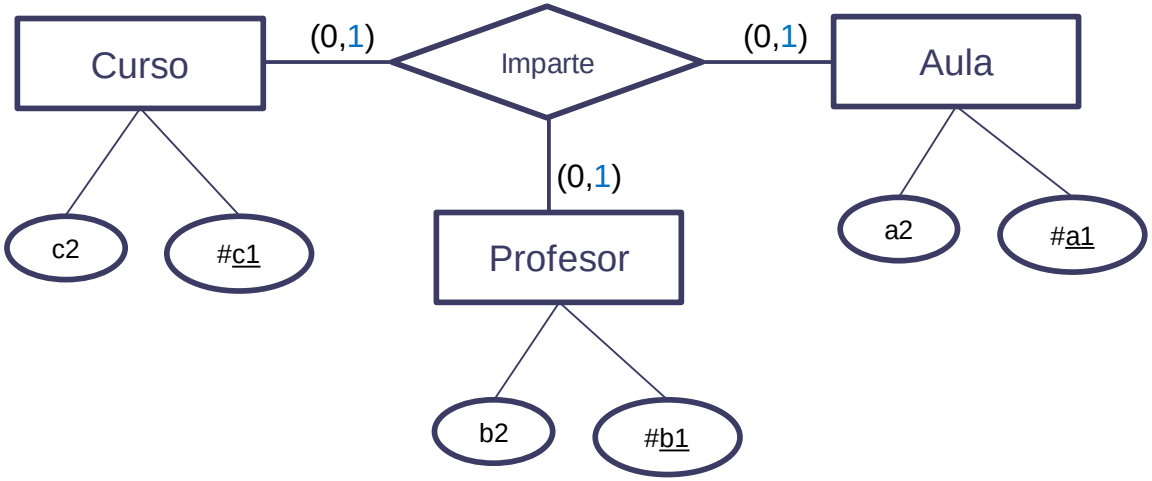


Existen tres cardinalidades máximas 1, por lo que hay tres claves candidatas para la relación R; cualquiera puede ser CP teniendo que definirse para las otras dos una restricción de unicidad.

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{a1, c1}
VNN {b1}
UNI {a1, b1}
UNI {b1, c1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

Transformación de relaciones ternarias

Relaciones ternarias 1:1:1. Opción 3

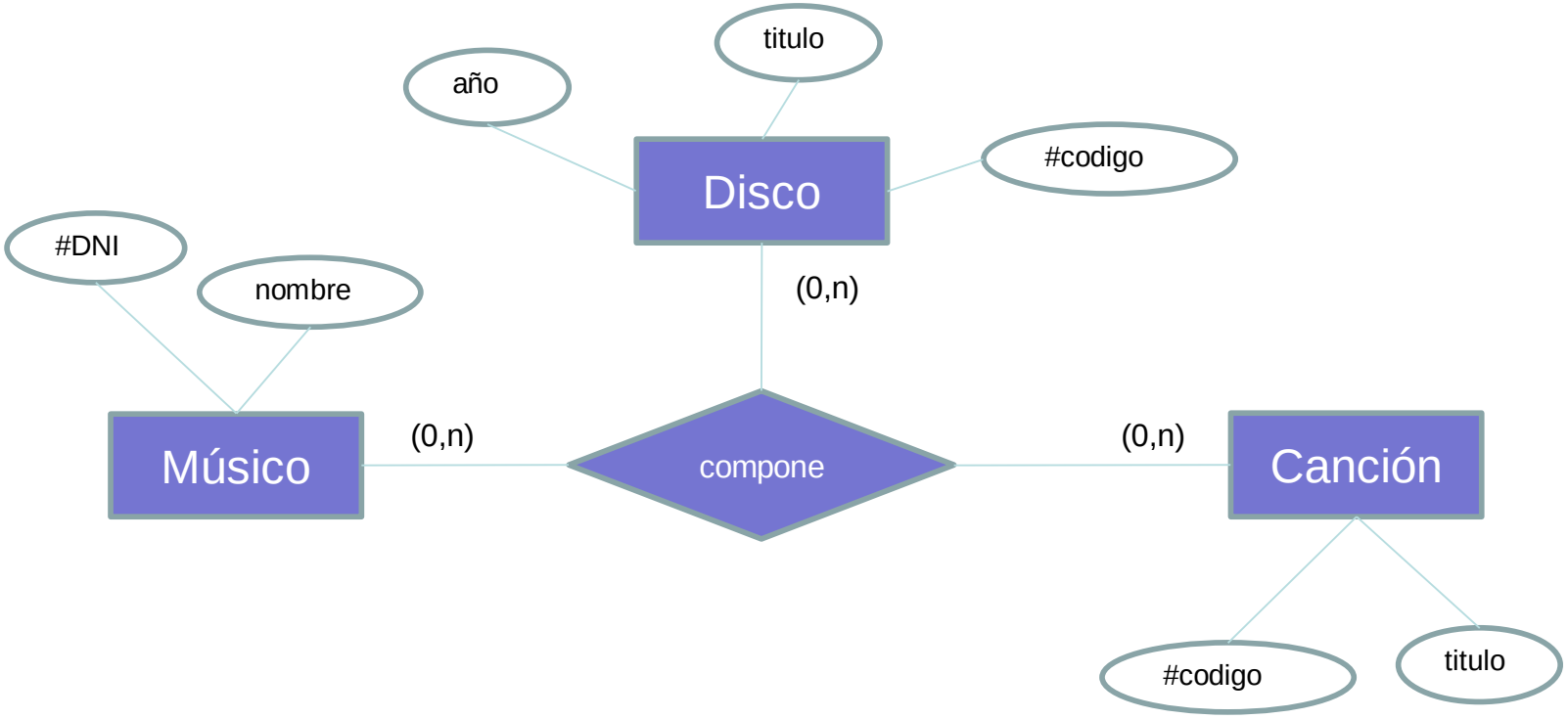


Existen tres cardinalidades máximas 1, por lo que hay tres claves candidatas para la relación R; cualquiera puede ser CP teniendo que definirse para las otras dos una restricción de unicidad.

Aula(a1,a2)
CP{a1}
Profesor (b1, b2)
CP{b1}
Curso (c1, c2)
CP {c1}
Imparte(a1,b1,c1)
CP{b1, c1}
VNN {a1}
UNI {a1, b1}
UNI {a1, c1}
CAj{a1} Referencia a Aula
CAj{b1} Referencia a Profesor
CAj{c1} Referencia a Curso

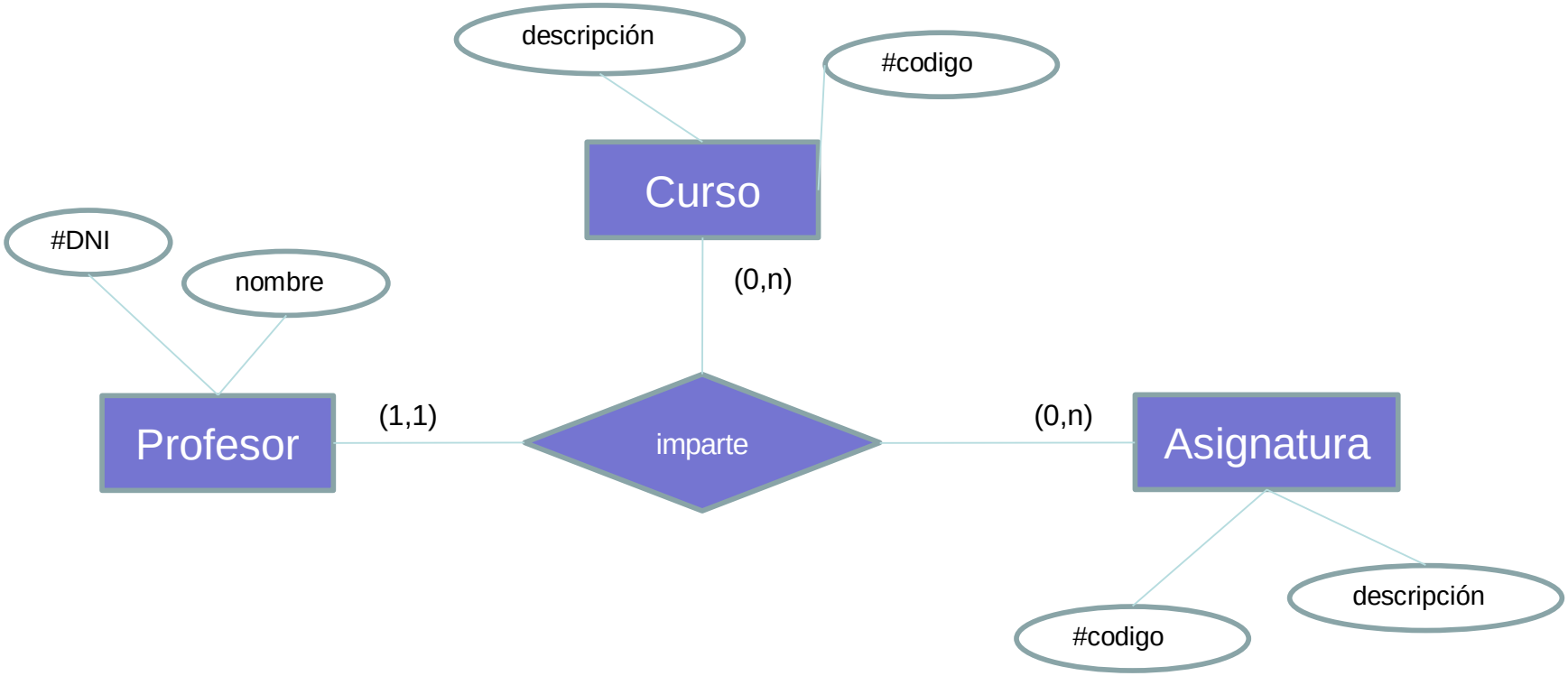
Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación



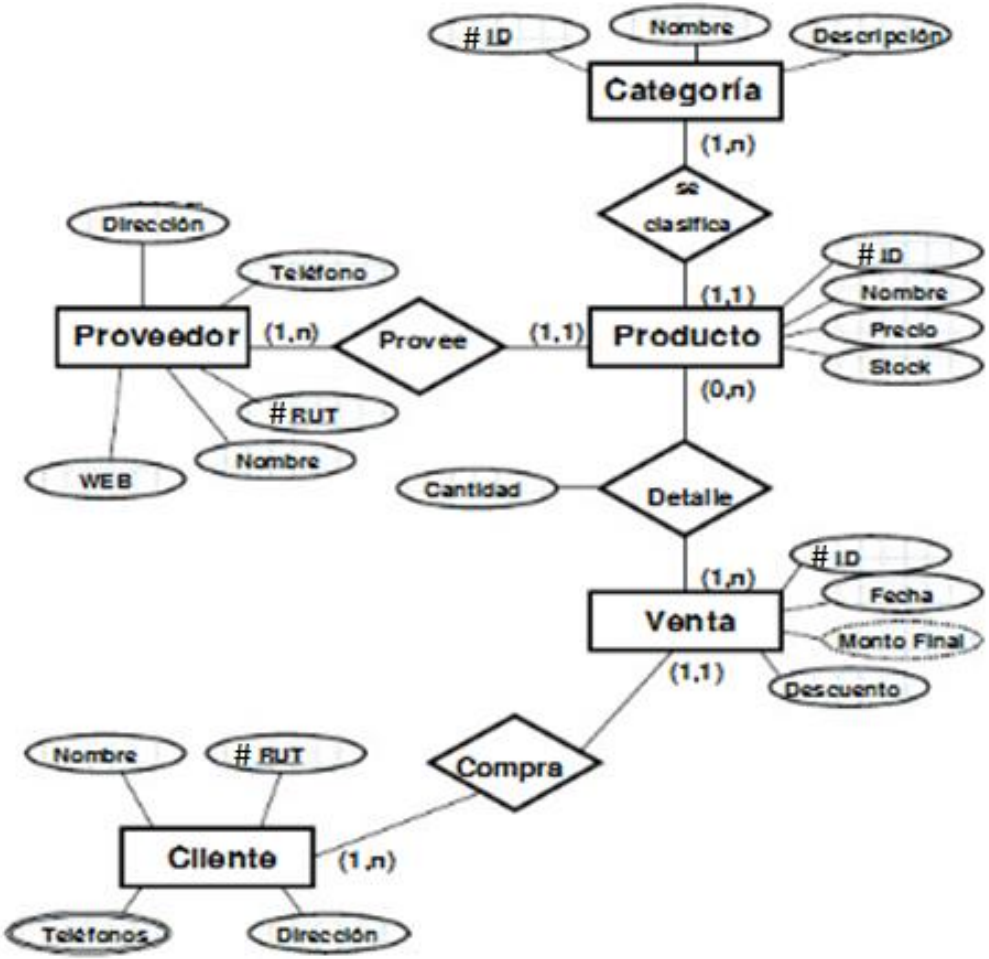
Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación



Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación.



Ejercicio

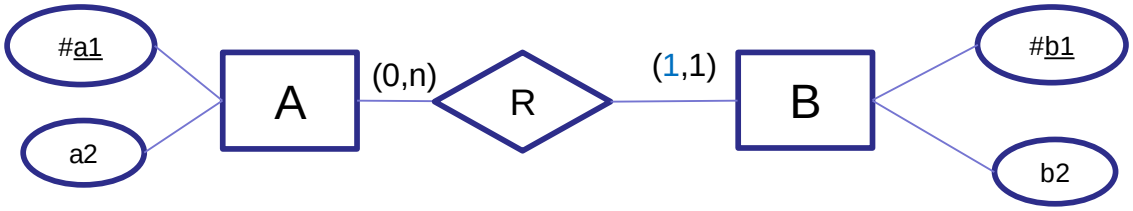
Convierte al modelo relacional el ejercicio:

- Gestión biblioteca

Restricciones de existencia

Las restricciones de existencia se dan cuando la cardinalidad mínima de la relación es 1.

Esta restricción suele indicarse con la propiedad Valor No Nulo (VNN) del atributo que se utiliza para indicar la relación.



A(a1,a2, b1)

CP{a1}

CAj{b1} hace referencia a B

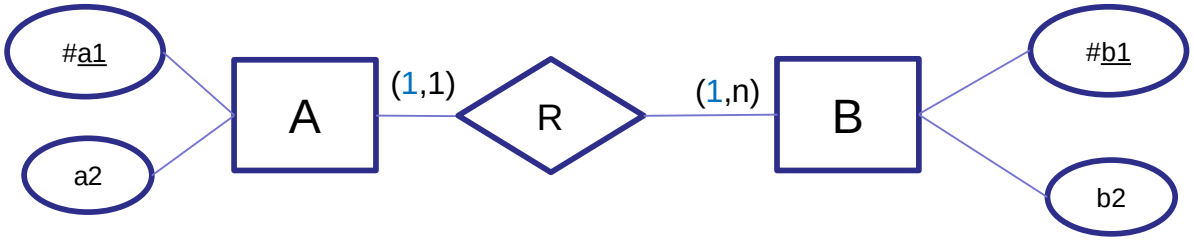
VNN {b1}

B (b1, b2)

CP{b1}

Restricciones de existencia

En el caso de que tenga dos restricciones de integridad, una de las restricciones habría que indicarla a través de una expresión del cálculo relacional o bien con SQL.



B (b1, b2)
CP{b1}
CAj {a1} hace referencia a A
VNN {a1}

Cálculo relacional tabla A

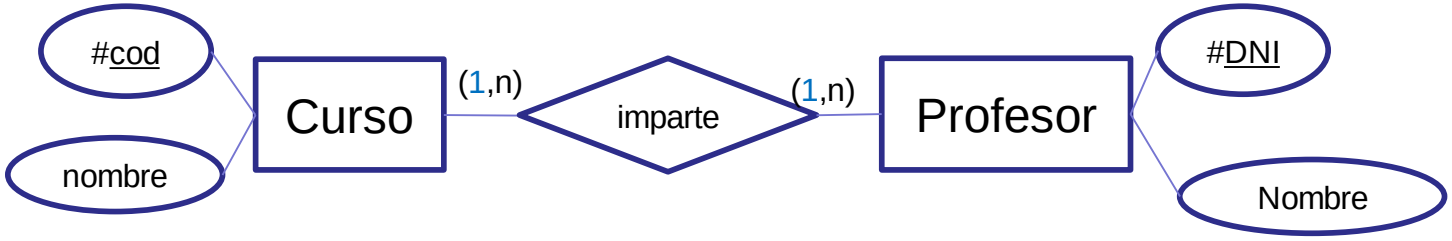
A(a1,a2)
CP{a1}

Ax: A, Bx: B

$\forall$ Ax(A(Ax) $\rightarrow$ $\exists$ Bx(B(Bx) $\wedge$ Bx.a1 = Ax.a1))

Restricciones de existencia

Cuando la cardinalidad máxima es de n:n y aparece una nueva tabla, la restricción de existencia se vincula a la nueva tabla



Curso (cod, nombre)

CP{cod}

Imparte (cod_curso, DNI_prof)

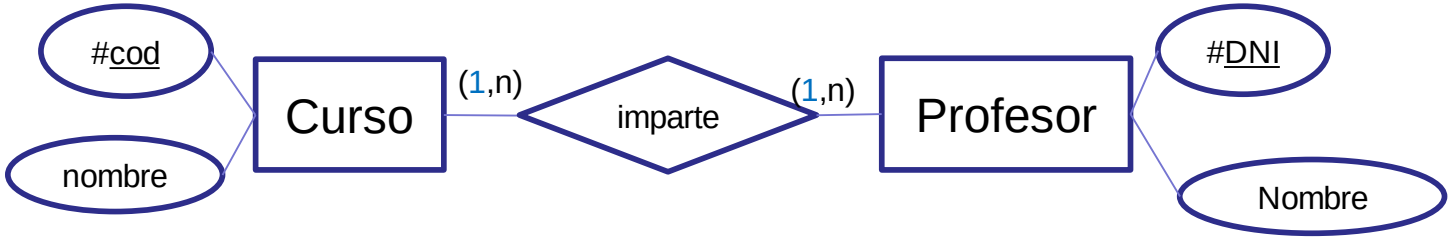
CP{cod_curso, DNI_prof}

Cx: Curso, Ix: Imparte

$\forall$ Cx(Curso(Cx) $\rightarrow$ $\exists$ Ix(Imparte(Ix) $\wedge$ Ix.cod_curso = Cx.cod))

Restricciones de existencia

Cuando la cardinalidad máxima es de n:n y aparece una nueva tabla, la restricción de existencia se vincula a la nueva tabla



Profesor (dni, nombre)

CP{dni}

Px: Profesor, Ix: Imparte

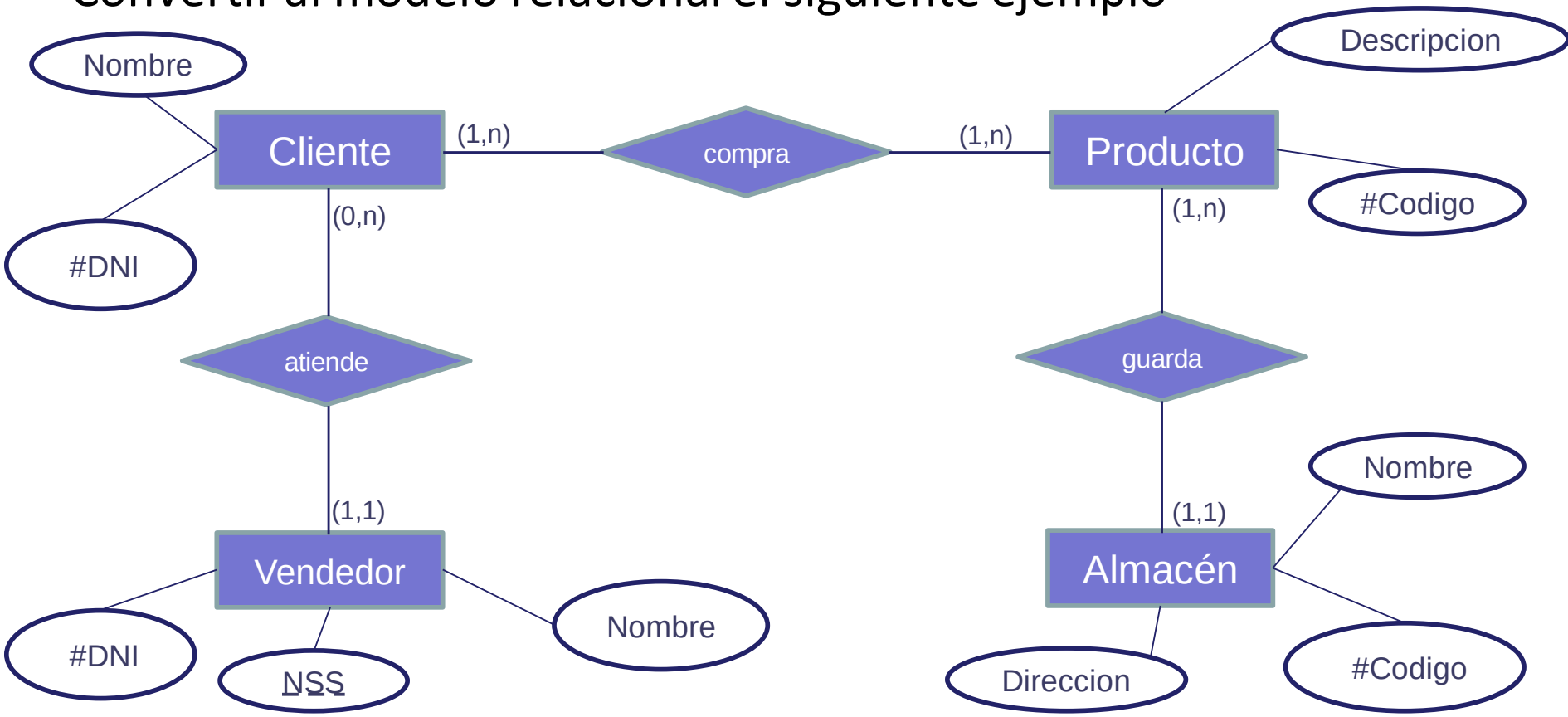
$\forall$ Px(Profesor(Px) $\rightarrow$ $\exists$ Ix(Imparte(Ix) $\wedge$ Ix.DNI_prof = Px.dni))

Imparte (cod_curso, DNI_prof)

CP{cod_curso, DNI_prof}

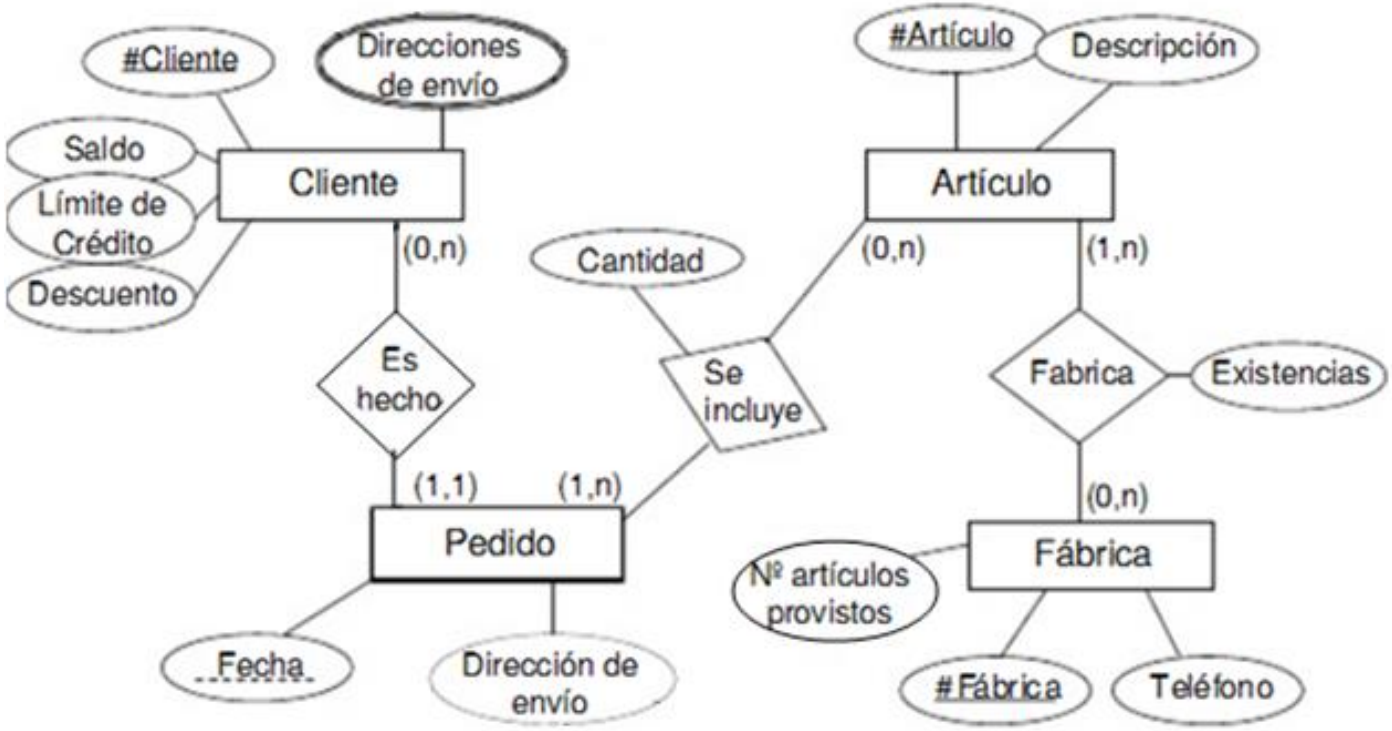
Ejercicio

Convertir al modelo relacional el siguiente ejemplo



Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación.



Ejercicio

Convierte al modelo relacional el siguiente diagrama de Entidad/Relación