

Bases de Datos

Clase 6: Diseño de Bases de Datos

Hasta ahora

Podemos comenzar a diseñar una base de datos, pero
¿lo estamos haciendo bien?

Pero un error en la modelación puede ser muy costoso!

Por ejemplo: olvidar añadir una relación

Esta Clase

Metodología para especificar nuestro problema

Construir una aplicación con un DBMS

1. Modelar los requerimientos (trabajo conceptual, con diagramas)
2. Diseño del esquema e implementación (tablas, atributos, llenar la base de datos)
3. Programar aplicación usando el DBMS (mucho más fácil si el trabajo anterior fue bien hecho)

Construir una aplicación con un DBMS

- 1. Modelar los requerimientos (trabajo conceptual, con diagramas)**
2. Diseño del esquema e implementación (tablas, atributos, llenar la base de datos)
3. Programar aplicación usando el DBMS (mucho más fácil si el trabajo anterior fue bien hecho)

Construir una aplicación con un DBMS

Las estadísticas dicen que un 87,4% de las aplicaciones tienen que volver a mejorar su base de datos en la tercera etapa!

(Las estadísticas también dicen que el 89,4% de las estadísticas se inventan en el momento)

Diseño conceptual de la BD

Orientado a:

- Qué entidades modelar
- Cómo relacionar esas entidades
- Restricciones en nuestro dominio
- Cómo lograr un buen diseño

Diseño conceptual de la BD

Orientado a:

- Qué entidades modelar
- Cómo relacionar esas entidades
- Restricciones en nuestro dominio
- Cómo lograr un buen diseño

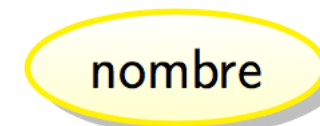
Aprenderemos a usar el modelo **entidad relación** (E/R)

Diagramas E/R

Entidad



Atributo

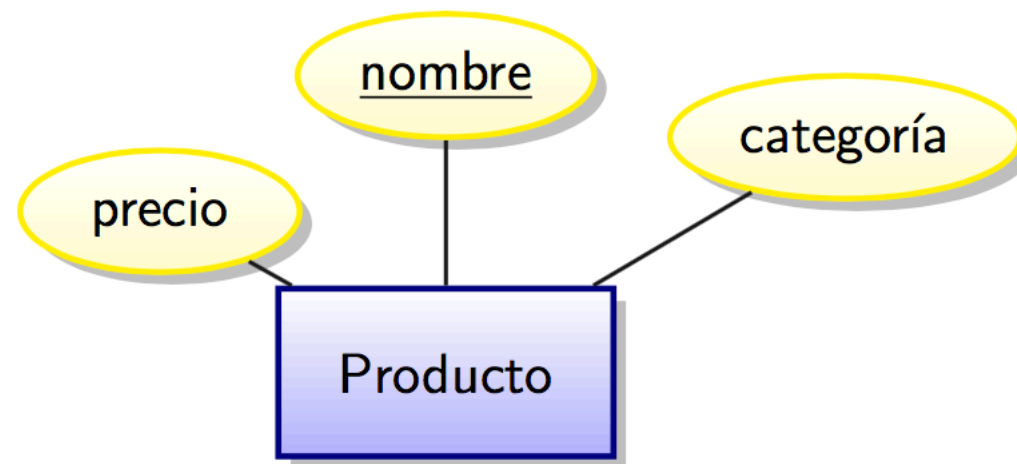


Relación



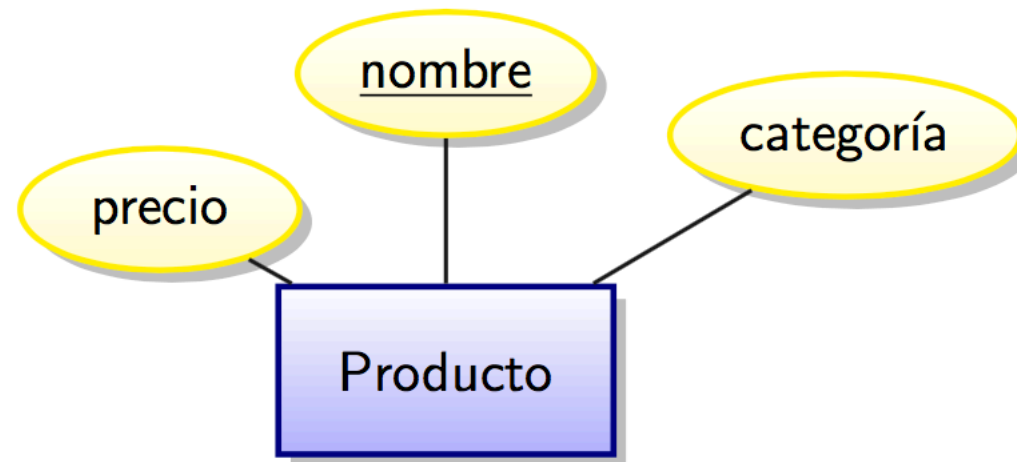
Diagramas E/R

Entidad con sus atributos



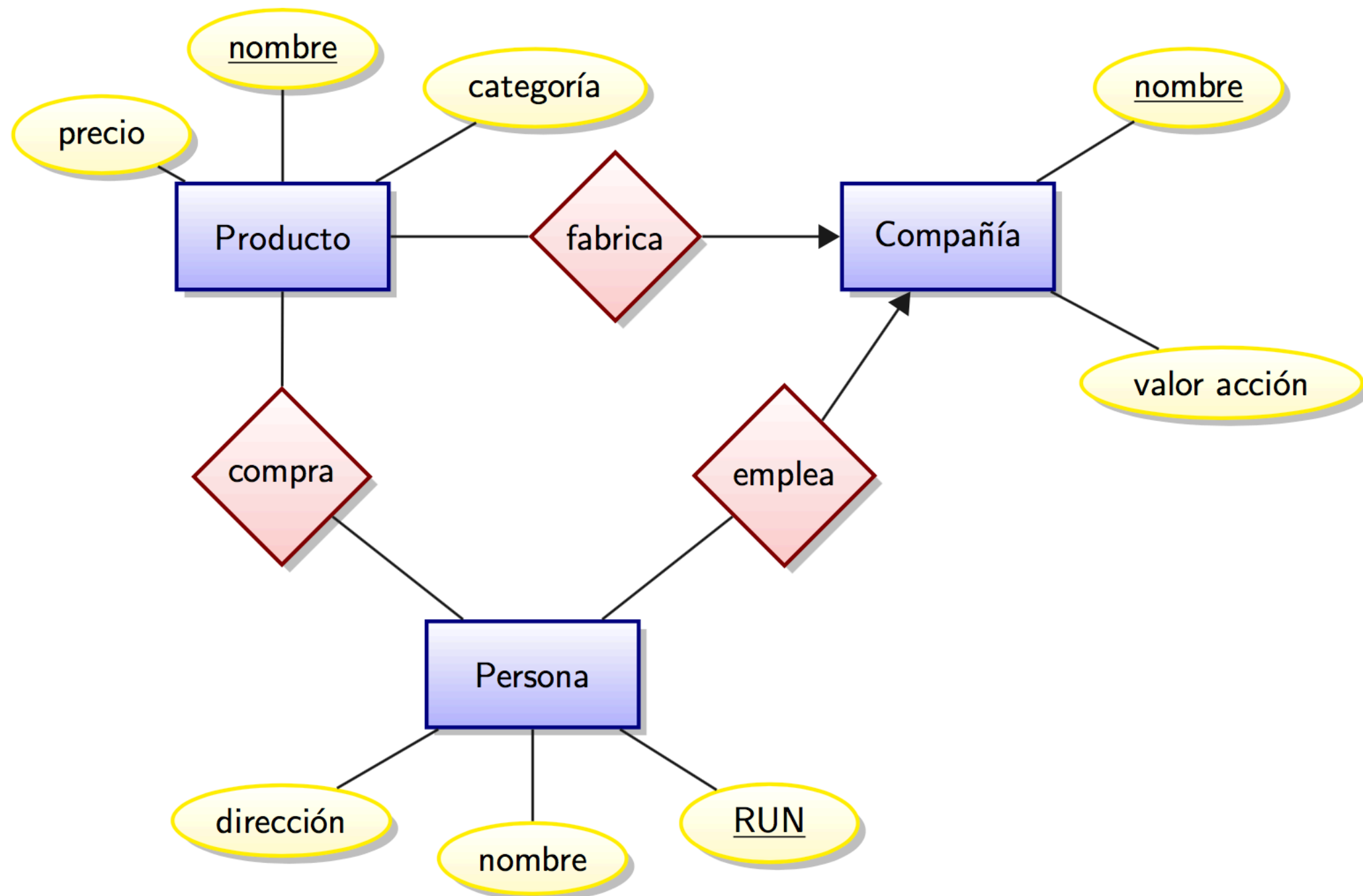
Diagramas E/R

Entidad con sus atributos



Obligatorio: cada entidad debe tener una llave

Diagramas E/R



Relaciones

Sean A y B conjuntos, una relación (binaria) R es un subconjunto de $A \times B$

Relaciones

Sean A y B conjuntos, una relación (binaria) R es un subconjunto de $A \times B$

$$A = \{1, 2, 3\}, B = \{a, b, c, d\}$$

Relaciones

Sean A y B conjuntos, una relación (binaria) R es un subconjunto de $A \times B$

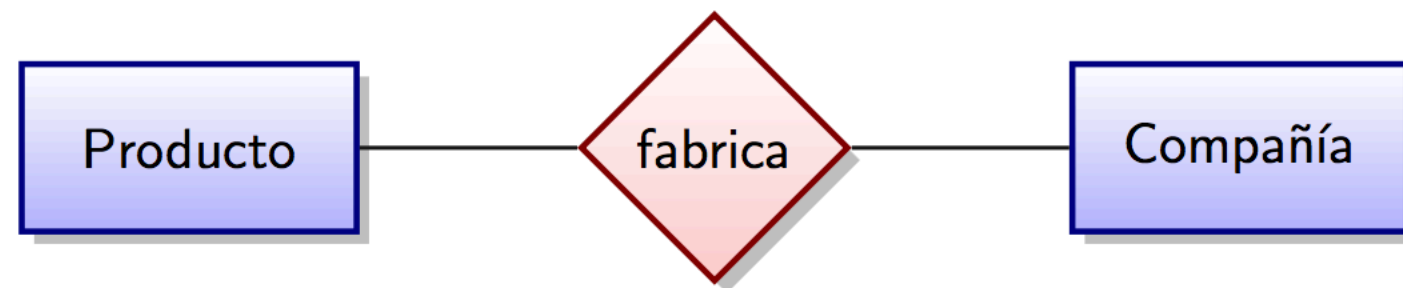
$$A = \{1, 2, 3\}, B = \{a, b, c, d\}$$

$$R = \{(1, a), (1, c), (2, b)\}$$

Relaciones

Multiplicidad

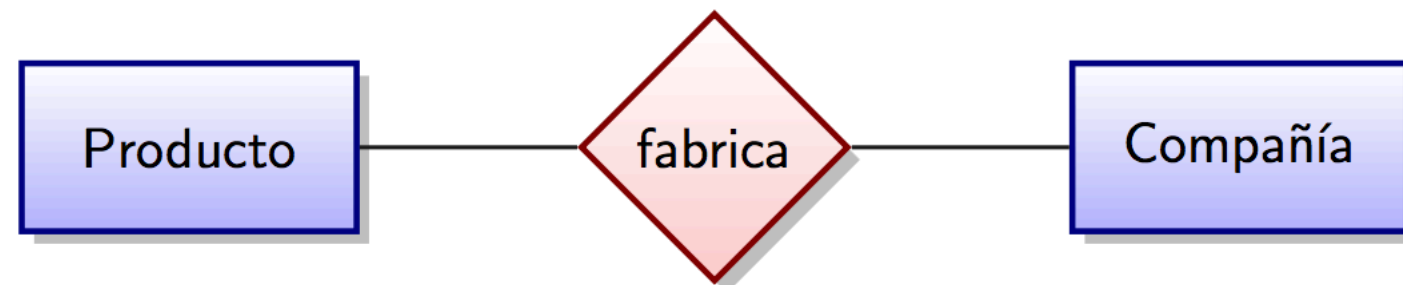
Varios a varios:



Cada producto tiene **muchas** compañías y cada compañía tiene **muchos** productos

Relaciones

Intuición

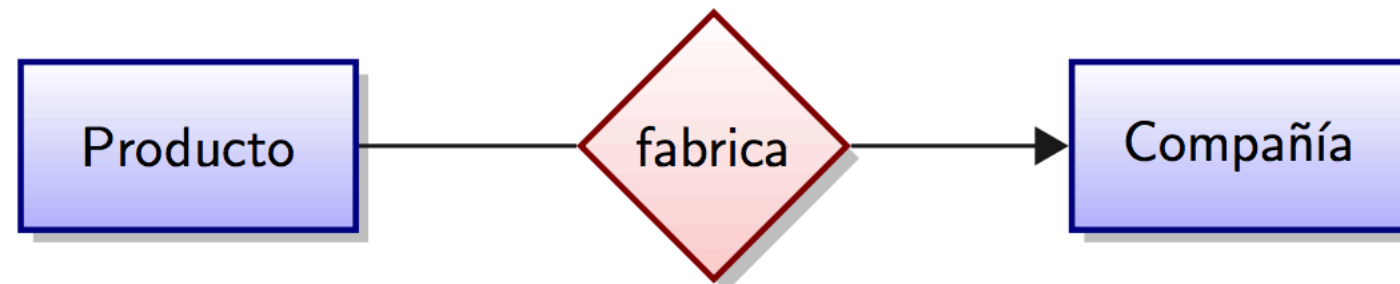


La llave de **Fábrica** se compone de la llave de **Producto** y de **Compañía**

Relaciones

Multiplicidad

Varios a uno:

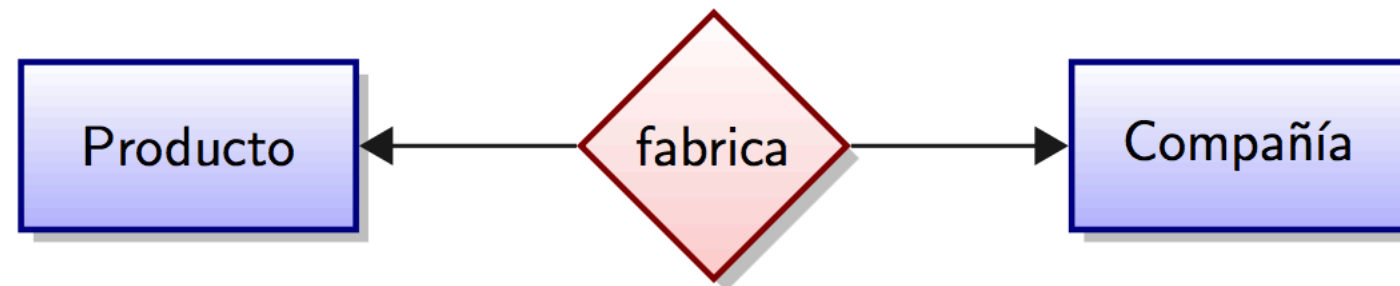


Cada producto tiene **una única** compañía, pero una compañía tiene **más de un** producto

Relaciones

Multiplicidad

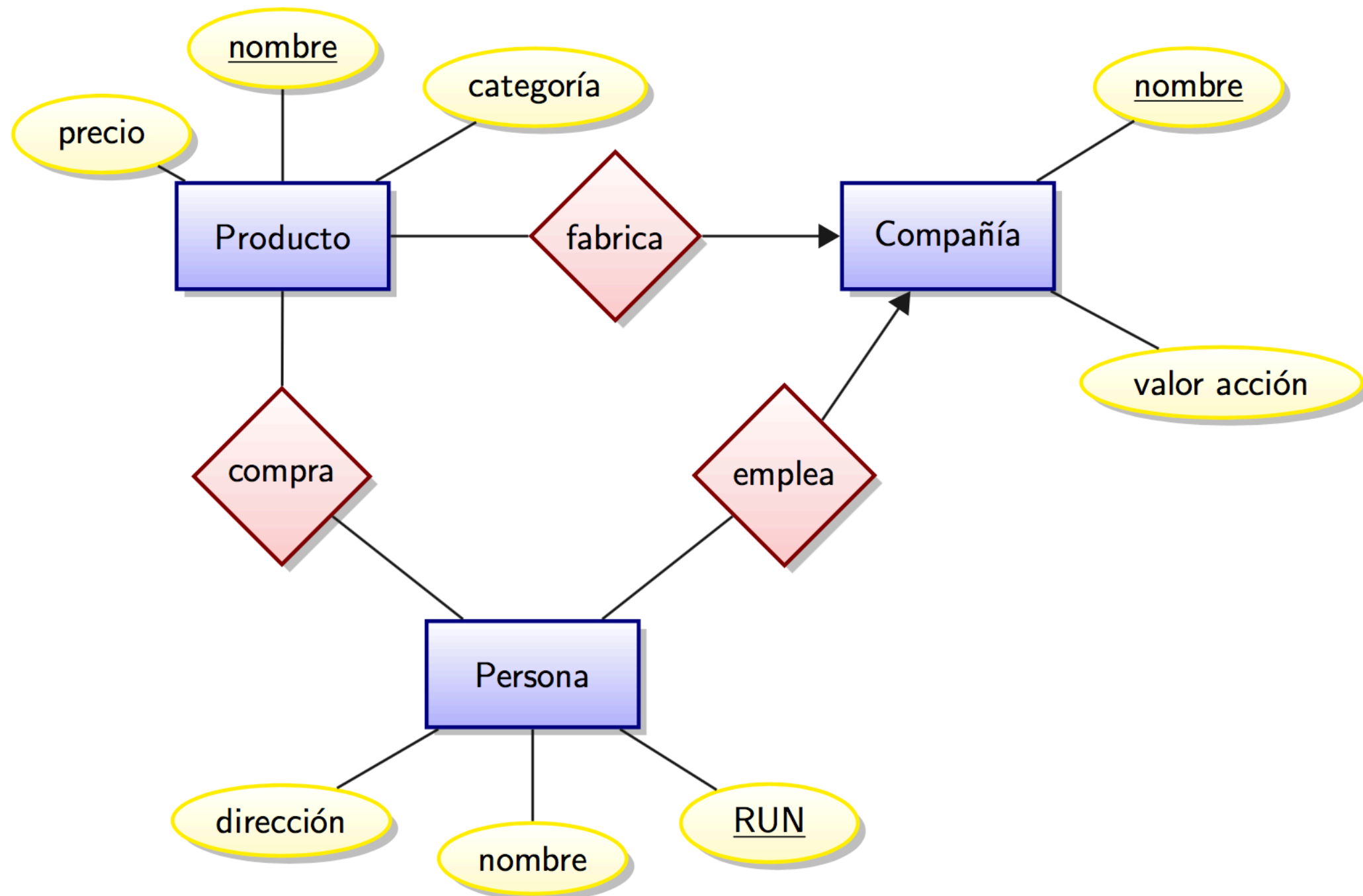
Uno a uno:



Cada producto tiene **una única** compañía, y cada compañía tiene un **único** producto

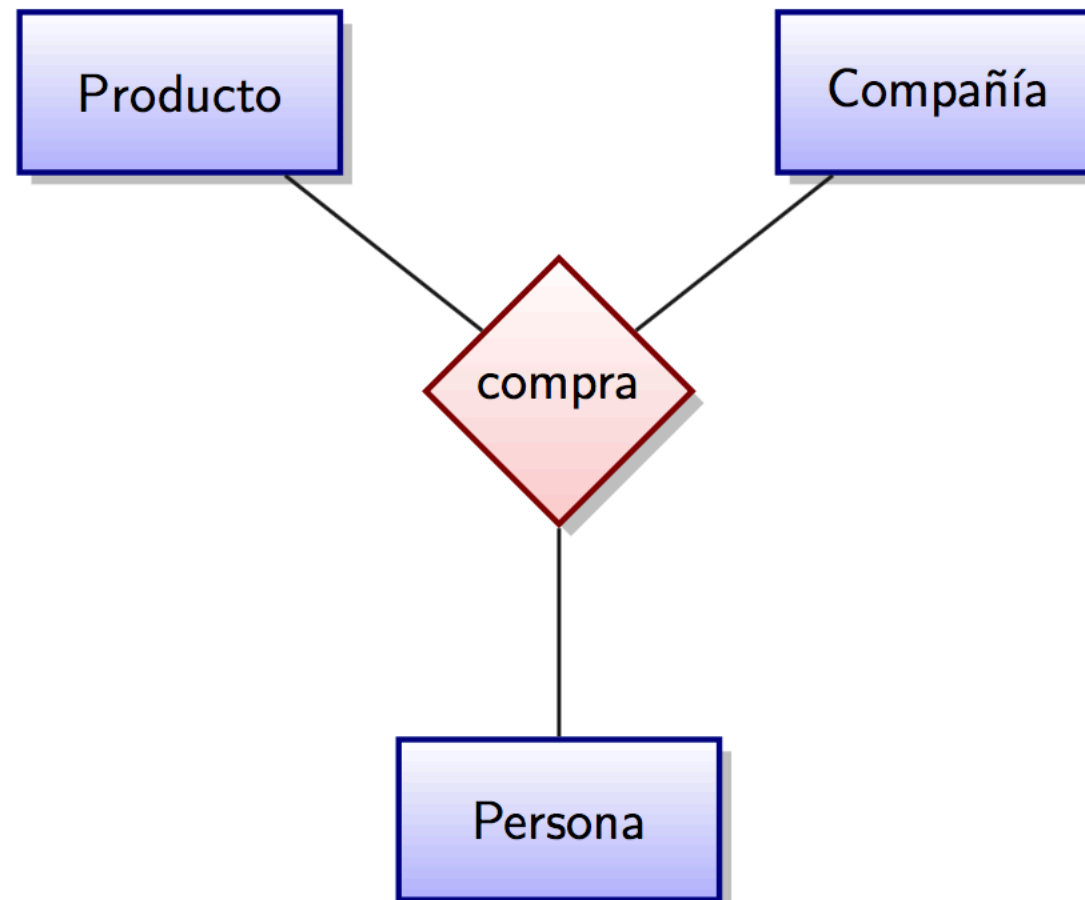
Diagramas E/R

Ejemplo



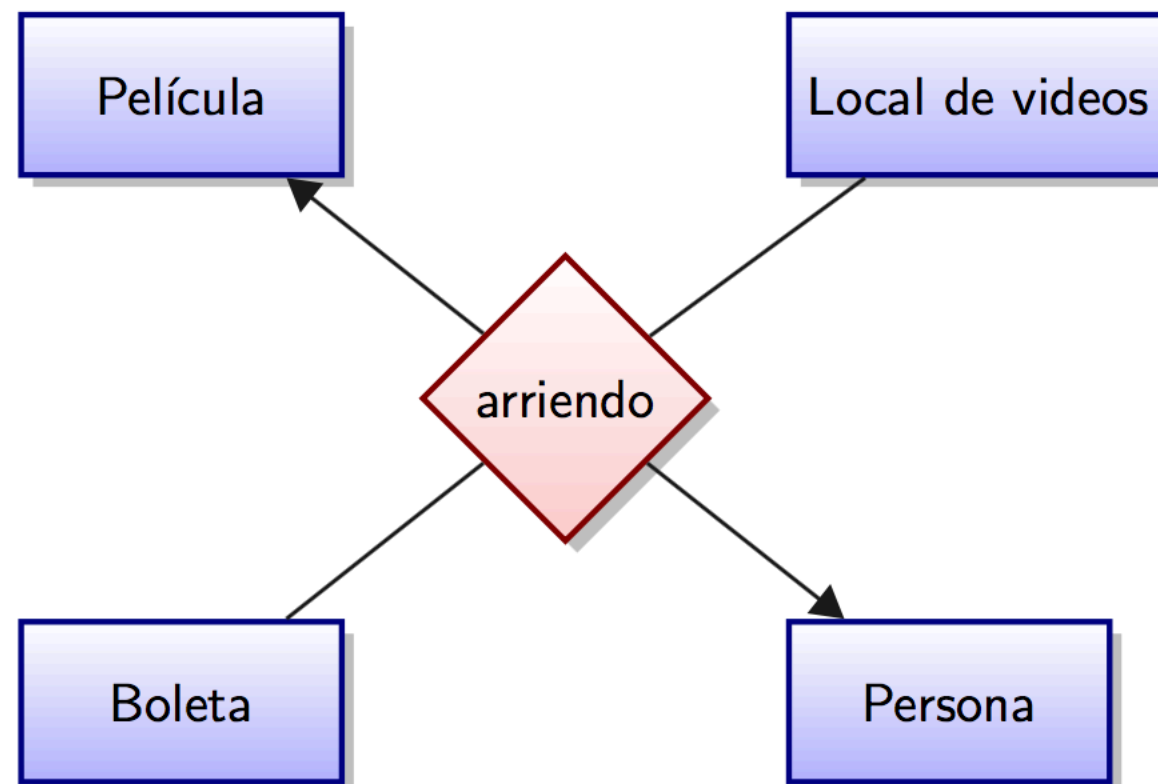
Relaciones múltiples

Una relación puede involucrar a más de 2 entidades

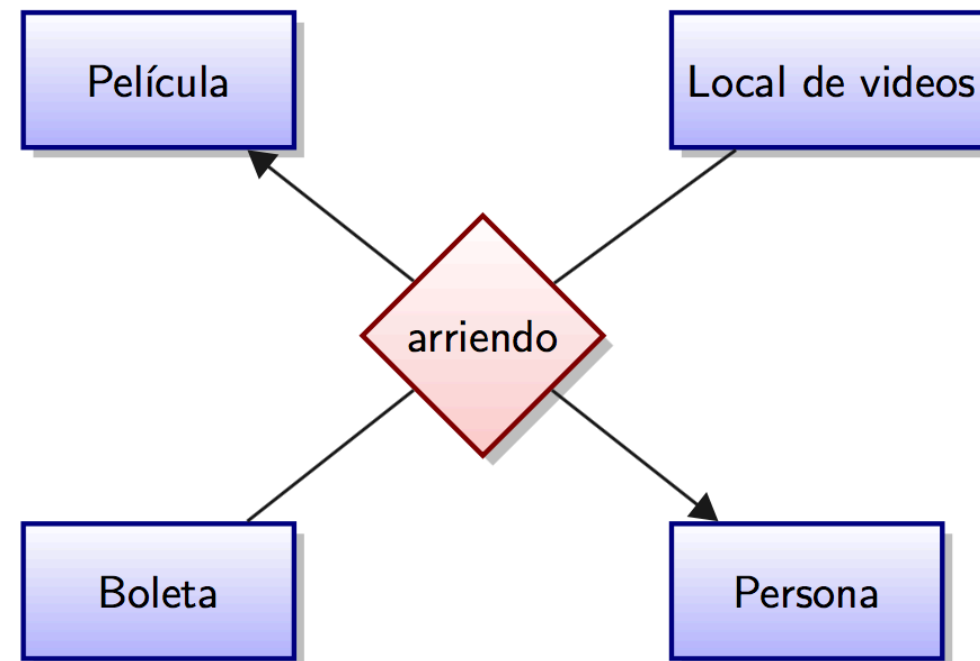


Dependencia multifuncional

¿Qué significa esto?



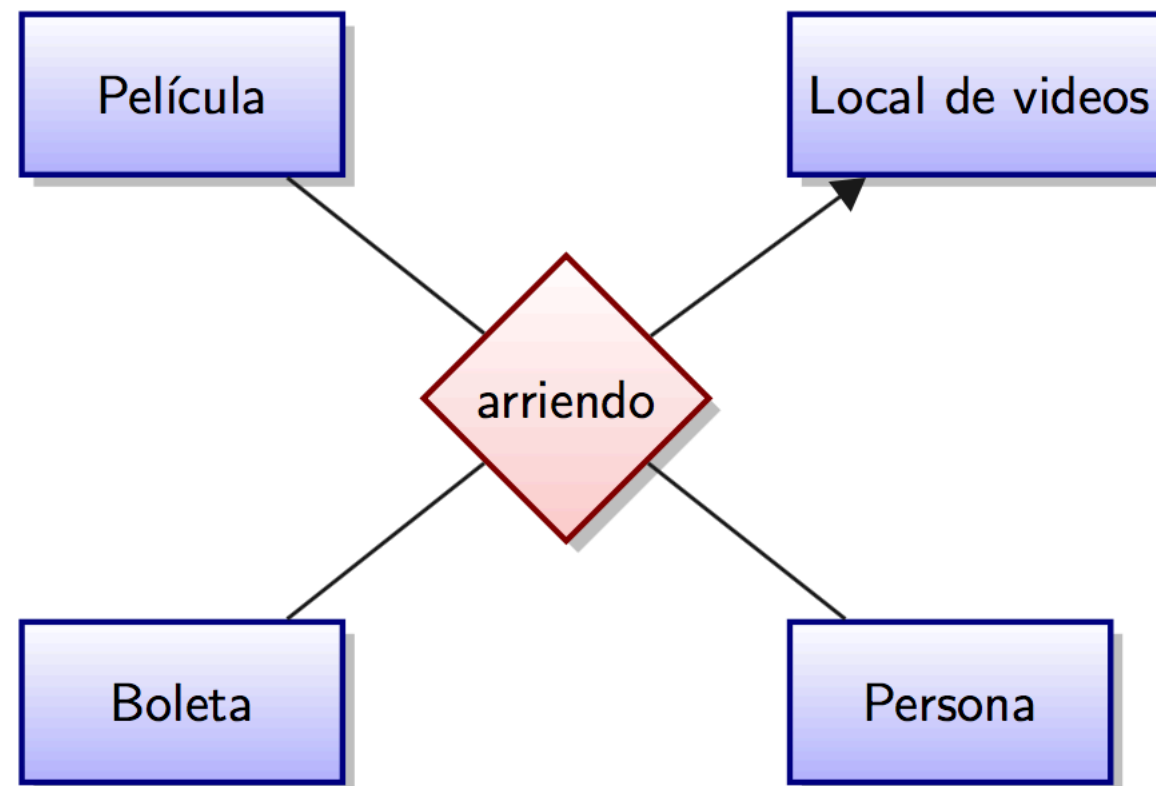
Dependencia multifuncional



- Puedo determinar la película con la boleta, la persona y el local
- Puedo determinar la persona con la película, la boleta y el local

Dependencia multifuncional

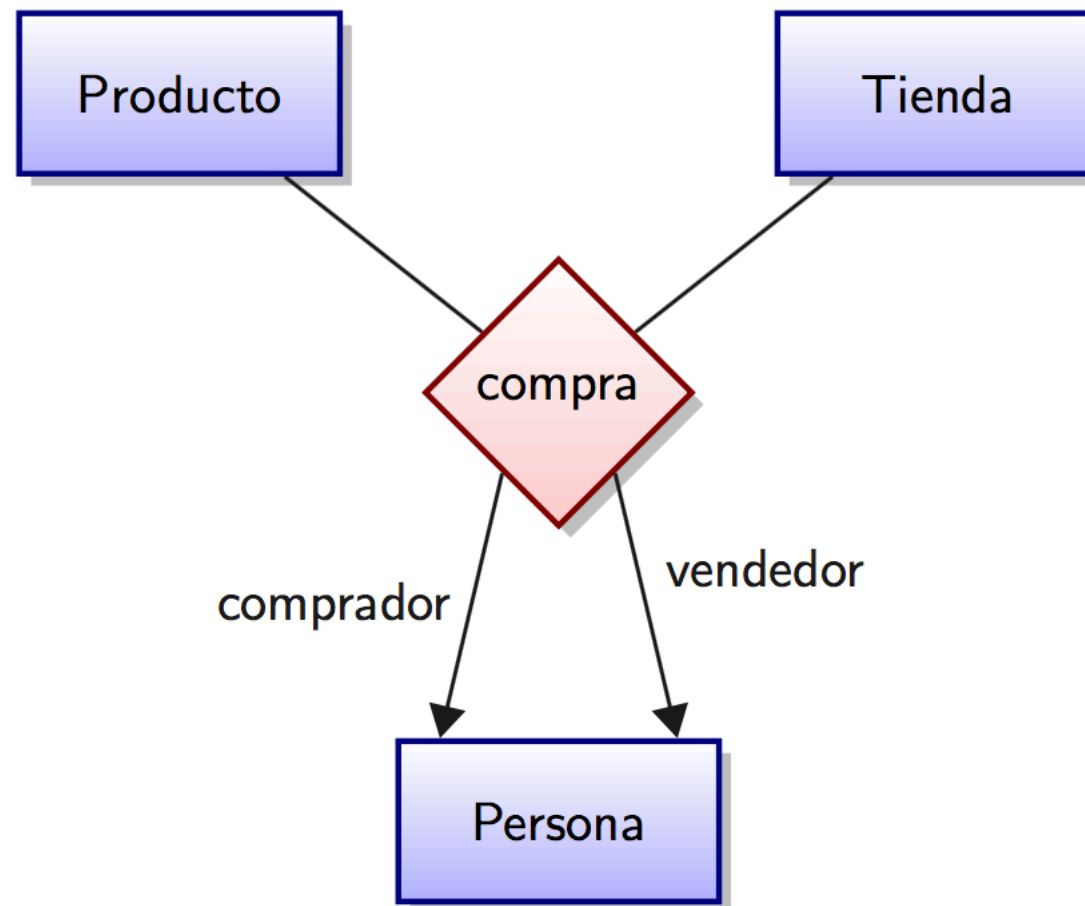
Quiero decir “la boleta determina la tienda”



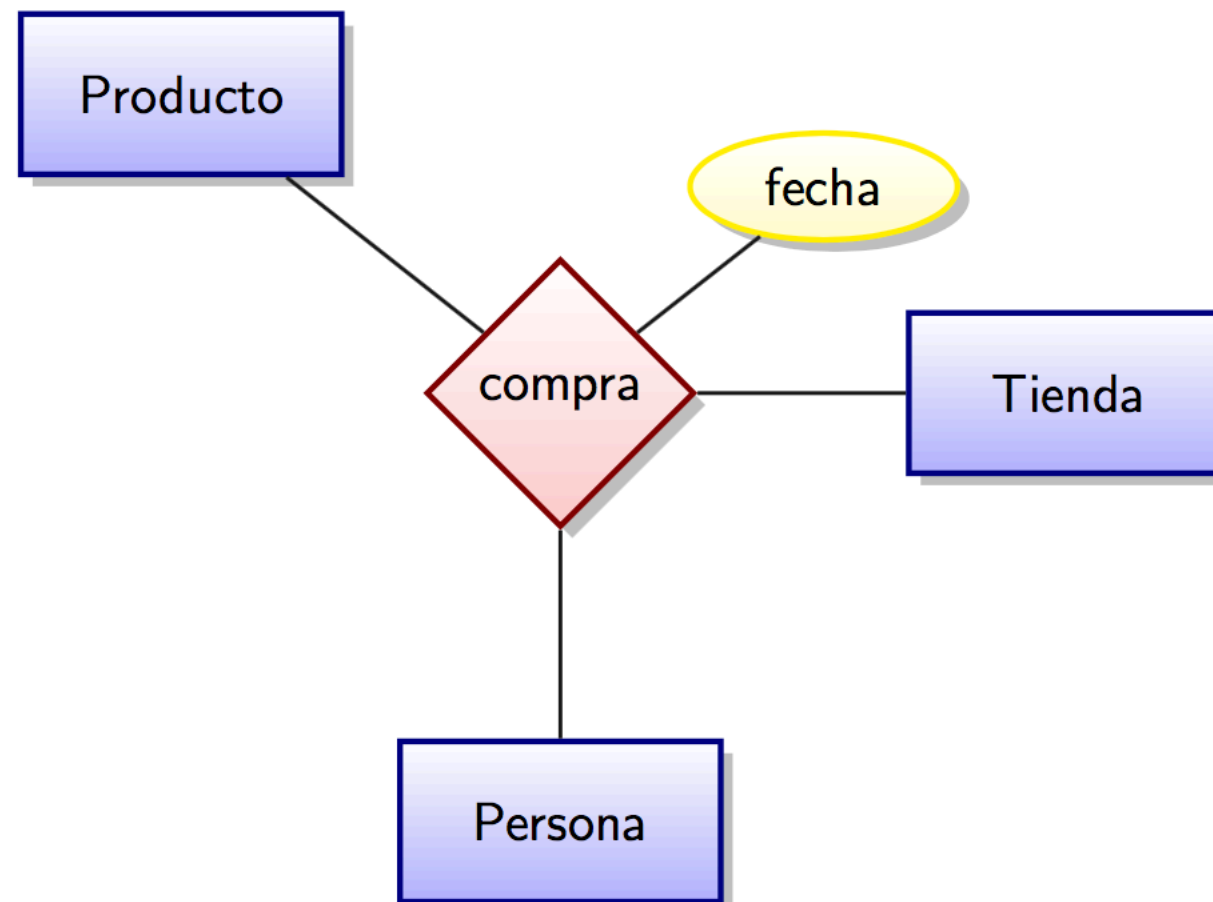
¿Por qué esto está incompleto?

Roles

Una entidad puede participar más de una vez en la relación



Atributos en relaciones



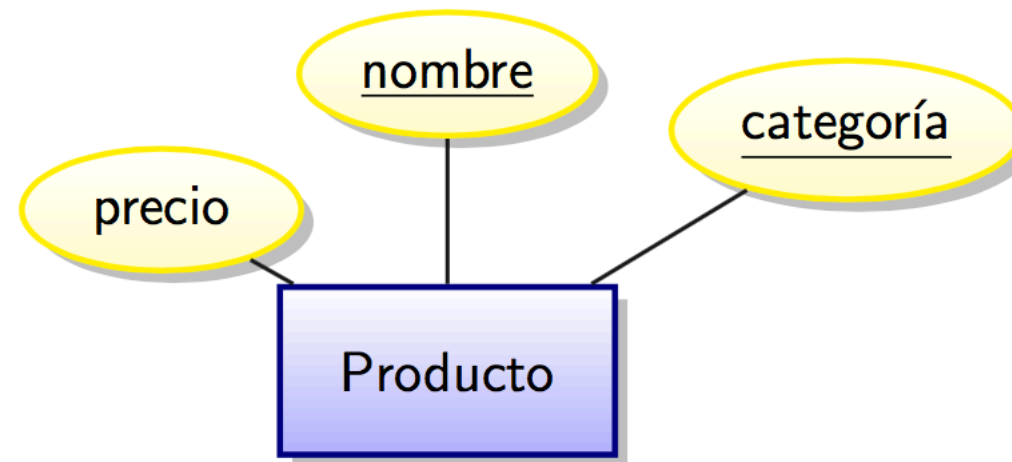
De E/R a esquema relacional

¡La transformación es prácticamente automática!

Punto de partida: cada entidad y relación es una tabla

De E/R a esquema relacional

Entidad a tabla

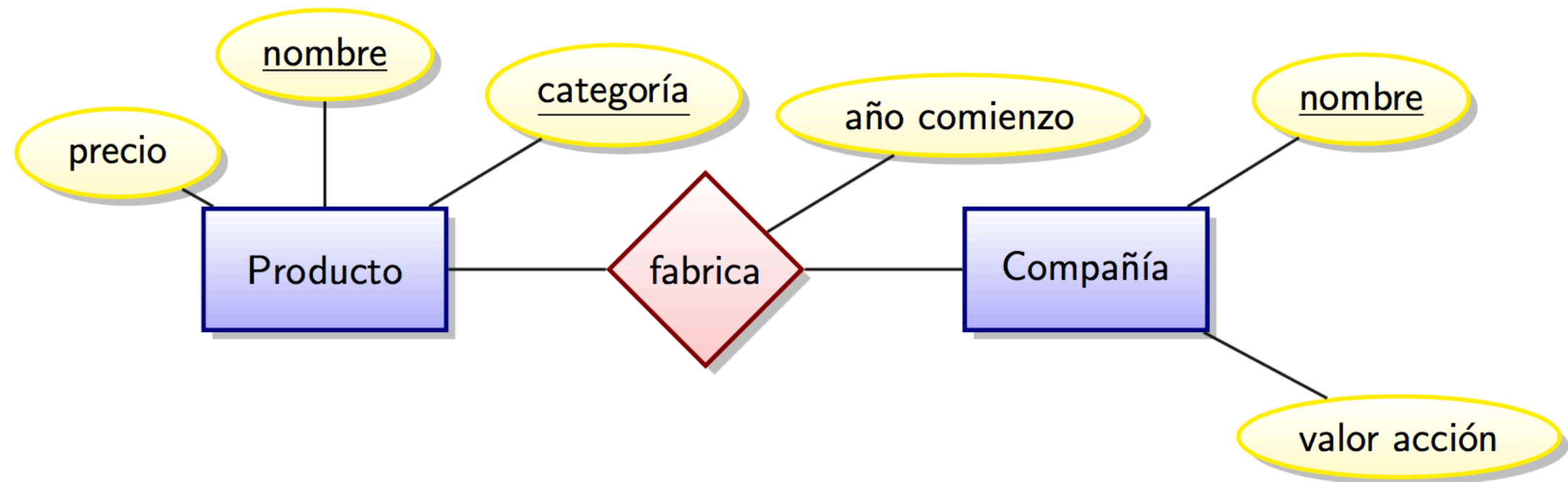


Producto(nombre, categoría, precio)

nombre	categoría	precio
Polaroid SX-70	Fotografía	\$100.000
Trípode	Fotografía	\$20.000

De E/R a esquema relacional

Relación a tabla



`Fabrica(Producto.nombre, categoría, Compañía.nombre, año_c)`

De E/R a esquema relacional

Relación a tabla

Fabrica(Producto.nombre, categoría, Compañía.nombre, año_c)

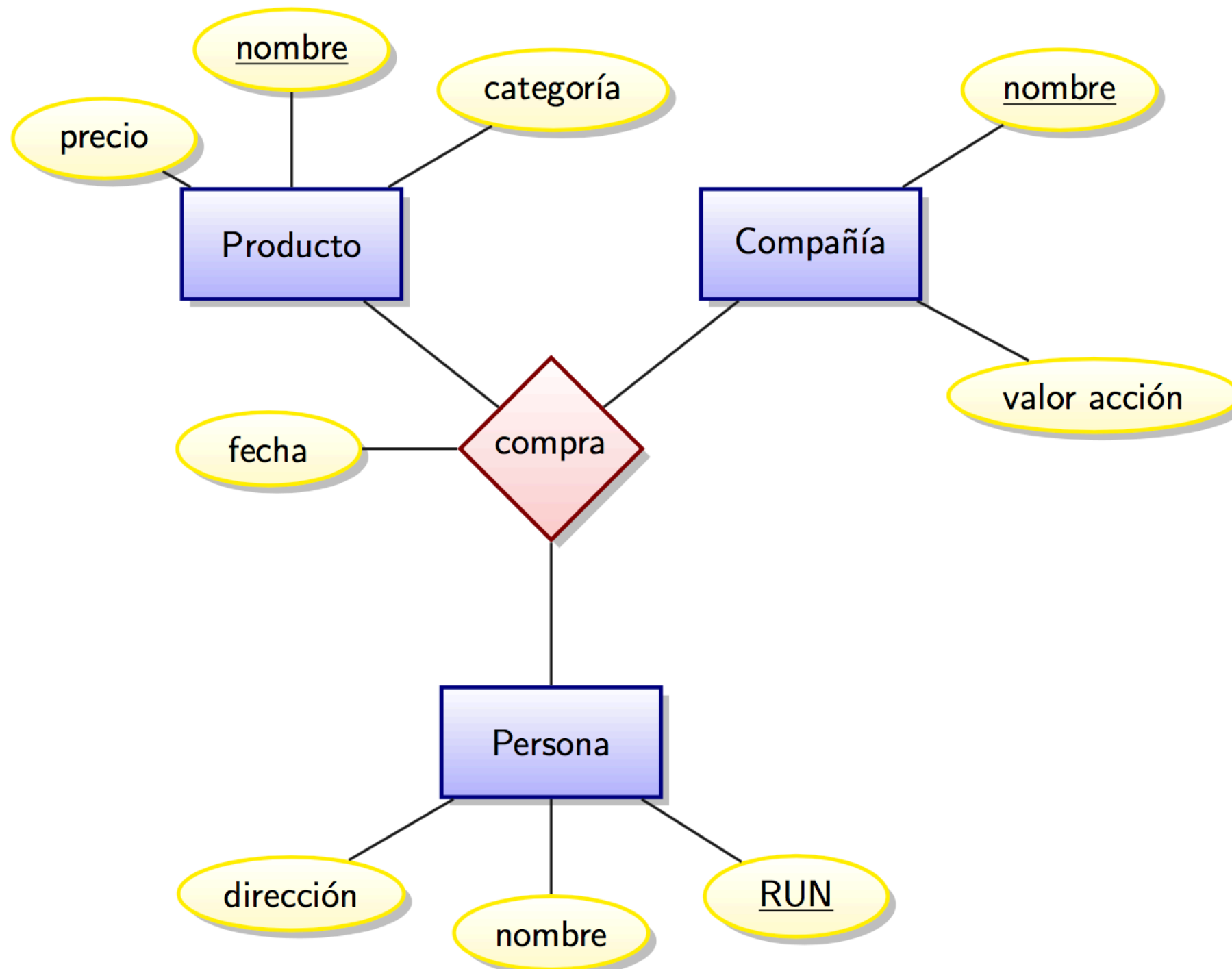
P.nombre	categoría	C.nombre	año_c
Polaroid SX-70	Fotografía	Polaroid SA	1974

De E/R a esquema relacional

En general

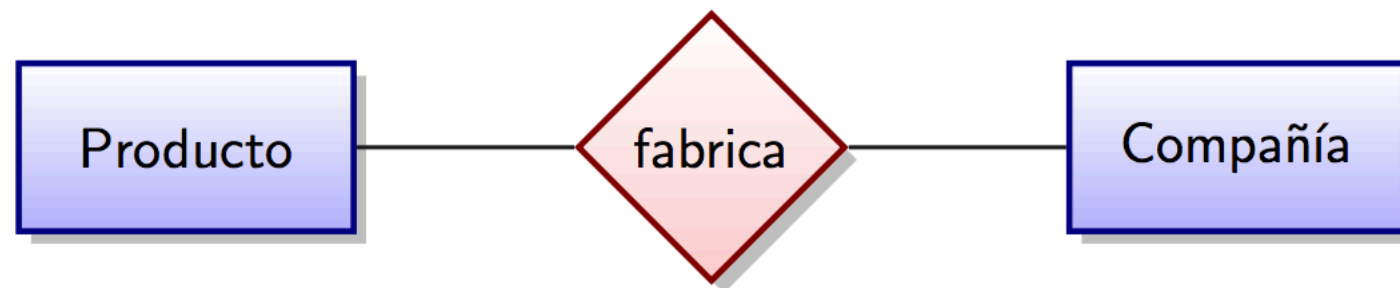
- Una tabla por cada relación
- Los atributos de la relación son atributos de la nueva tabla
- Para cada entidad (o rol) en la relación, son atributos sus llaves primarias (generalmente)

Ejemplo



Llaves de relaciones

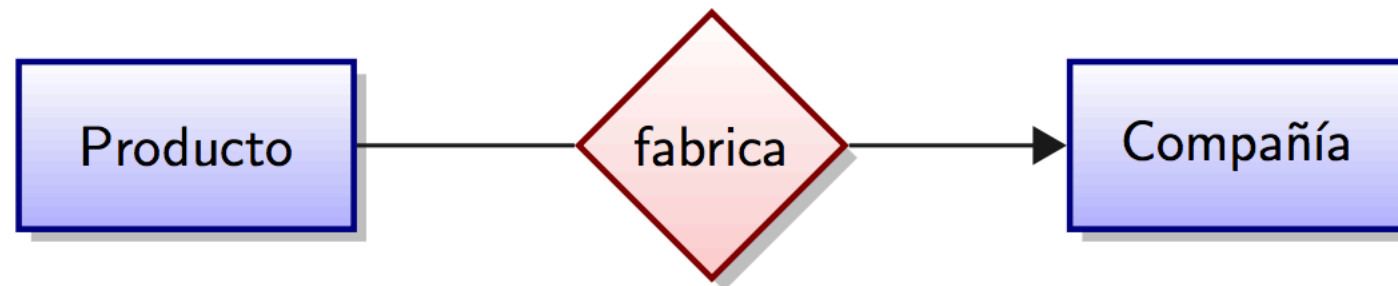
Varios a varios



La llave de fabrica son las llaves de Producto y de Compañía

Llaves de relaciones

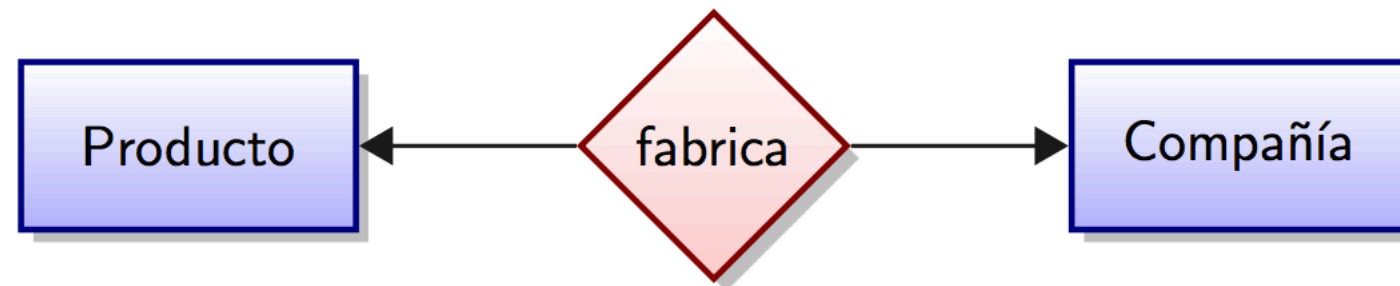
Varios a varios



La llave de fabrica es la llave de Producto ¿Por qué?

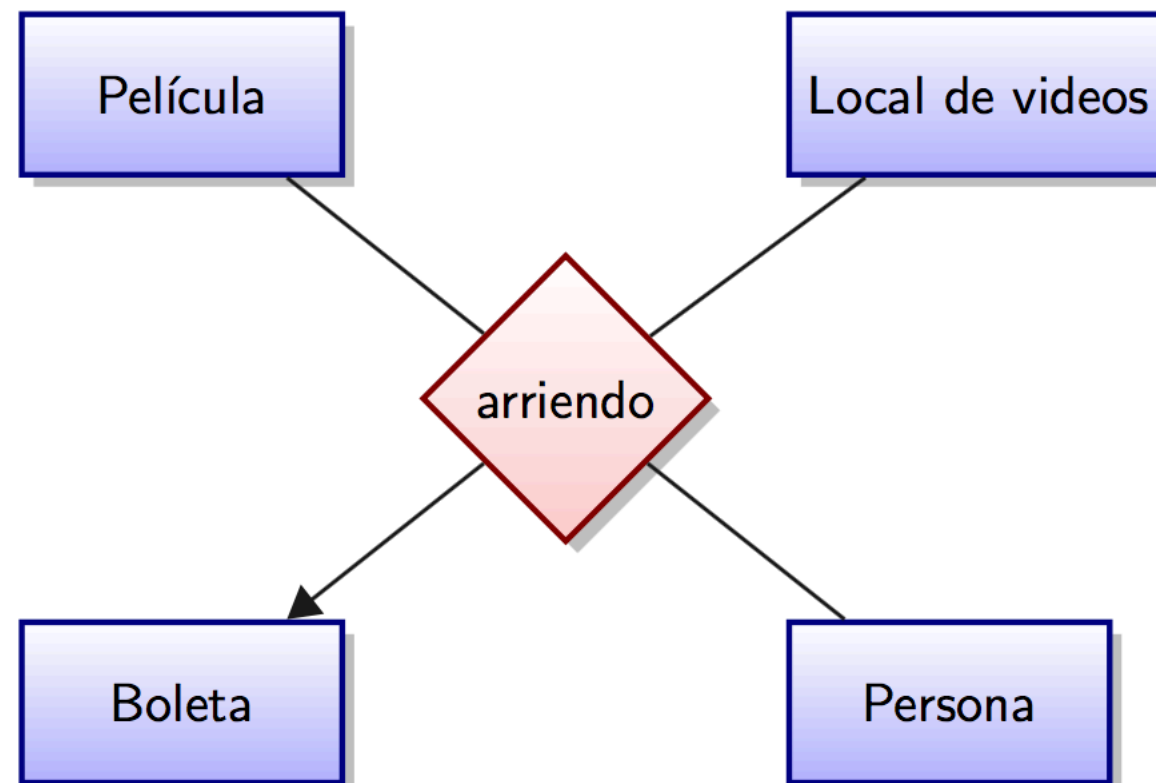
Llaves de relaciones

Varios a varios



La llave de fabrica es la llave de Producto o la de Compañía

Llaves de relaciones múltiples

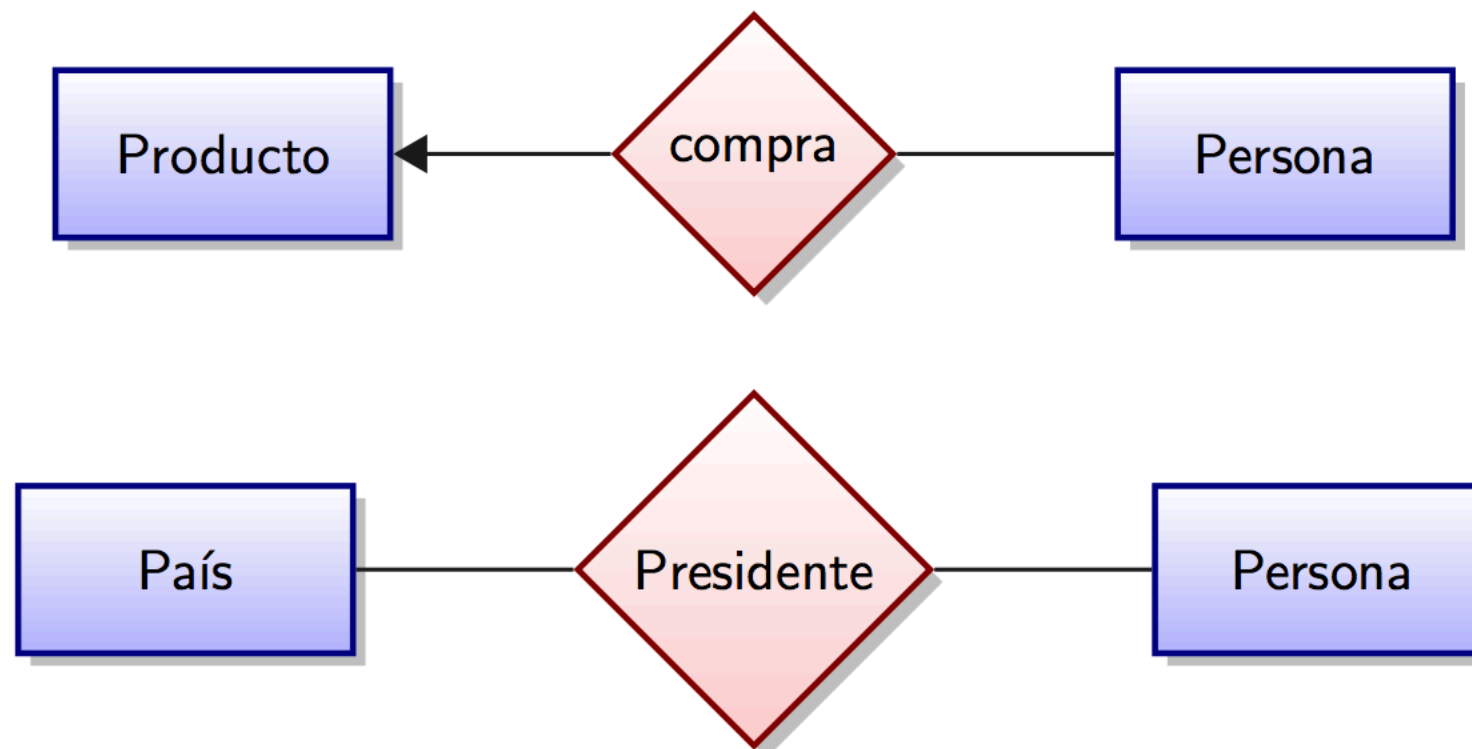


Acá no necesitamos la llave de Boleta en la relación arriendo ¿Por qué?

Principios básicos del diseño

Fidelidad al problema

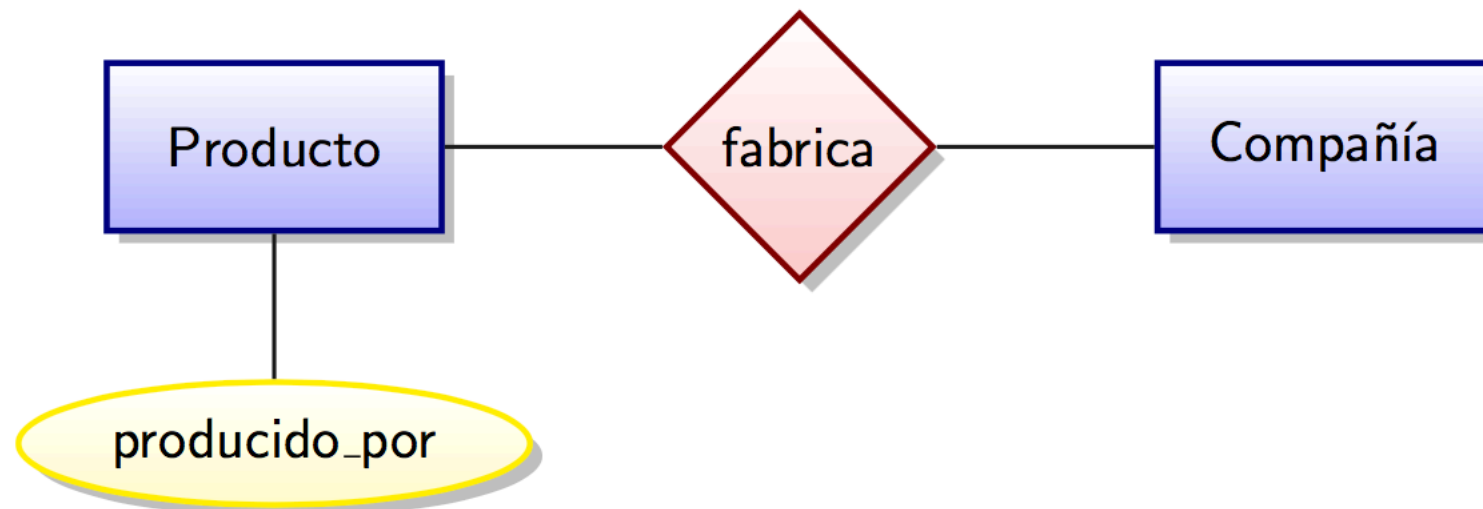
¿Qué está mal?



Principios básicos del diseño

Evitar redundancia

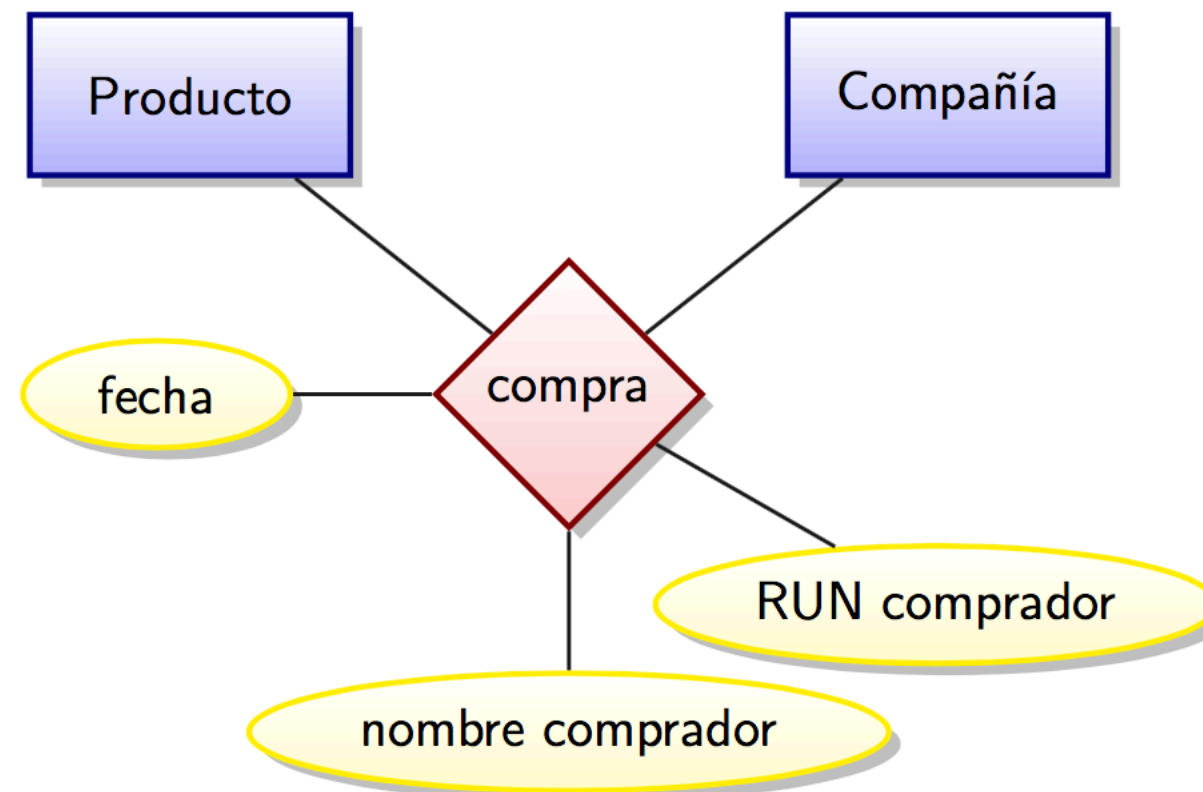
Algo como esto, puede generar **anomalías**



Principios básicos del diseño

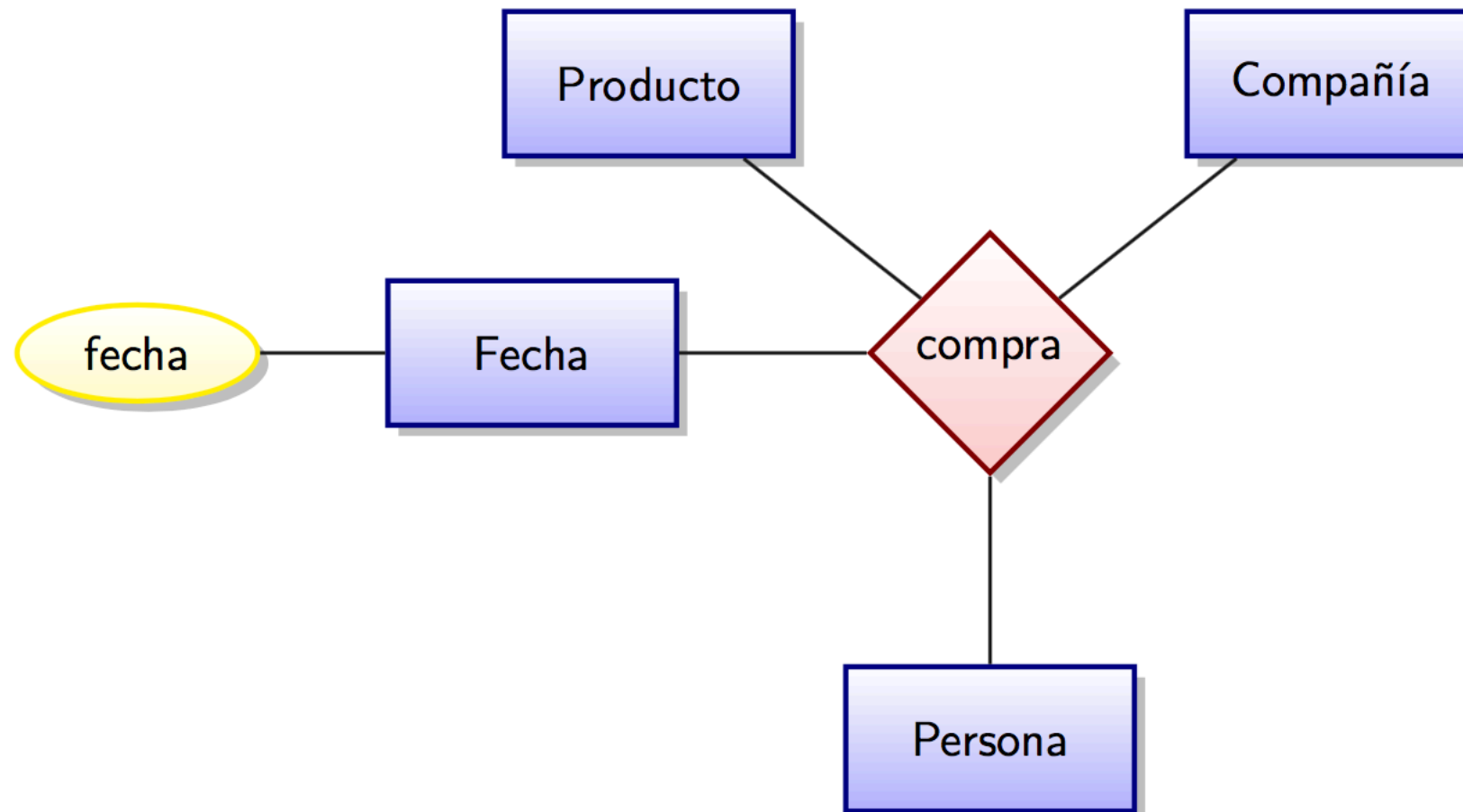
Elegir entidades y relaciones correctamente

¿Qué está mal?



Principios básicos del diseño

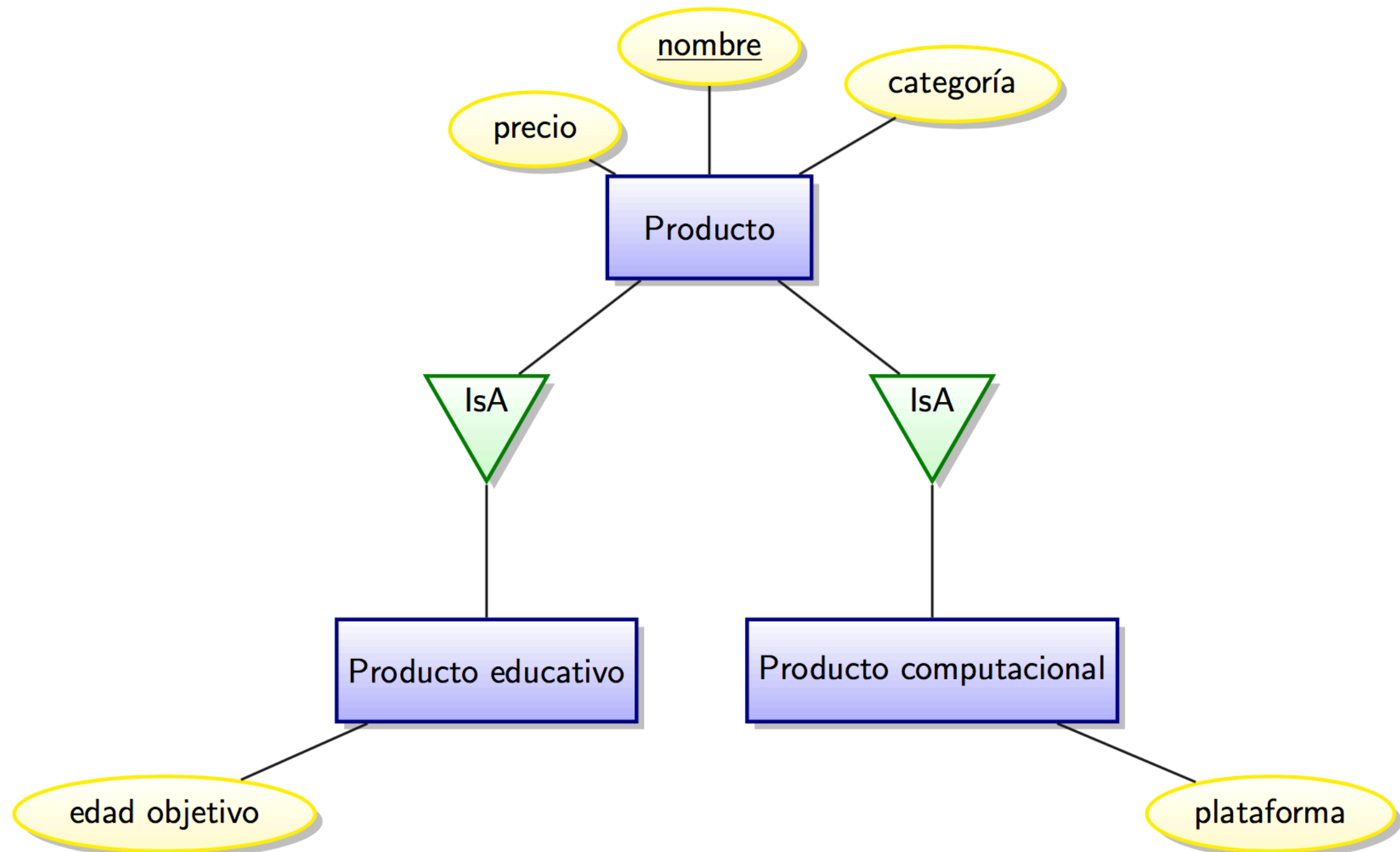
No complicar más de lo necesario



Subclases (IsA)

- Algunas entidades son casos especiales de otra
- Similar a la herencia en orientación a objetos
 - Ej. Todo estudiante es también una persona
- Se hace con la relación IsA (EsUn en Inglés)

Subclases (IsA)



Subclases (IsA)

Una idea de transformación

Producto(nombre, precio, categoría)

ProductoEducativo(nombre, edad_objetivo)

ProductoComputacional(nombre, plataforma)

Subclases (IsA)

Una idea de transformación

- Las entidades hijo tienen la misma primary key que la entidad padre
- Las entidades hijo tienen sólo los atributos adicionales
- Para obtener la versión completa hago un join

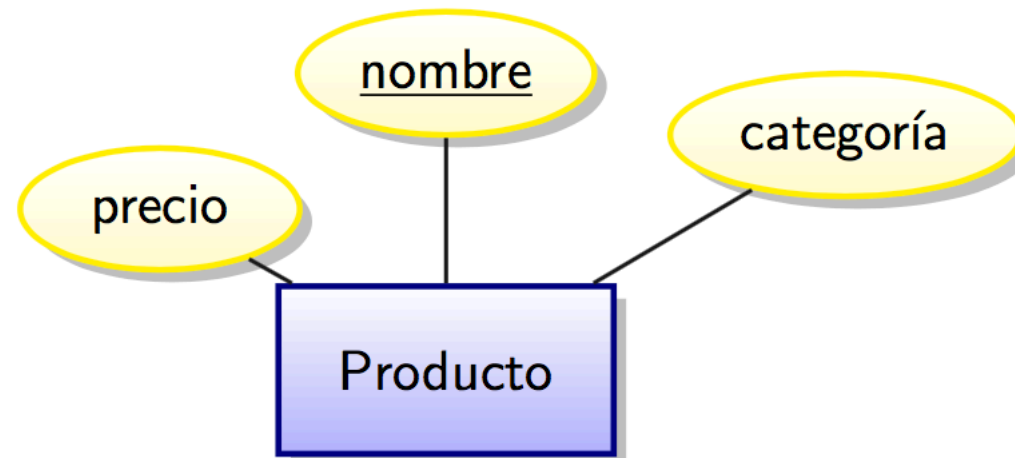
Restricciones de integridad

Encontrar restricciones de integridad es parte importante de la modelación:

- Llaves: el RUT es valor único
- Único: sólo puedo tener un padre
- Participación: toda compañía debe tener al menos un empleado
- Referenciales: si se trabaja en una compañía, esta debe existir
- Numéricas: la edad de las personas debe estar entre 0 y 150 años

Restricciones de integridad

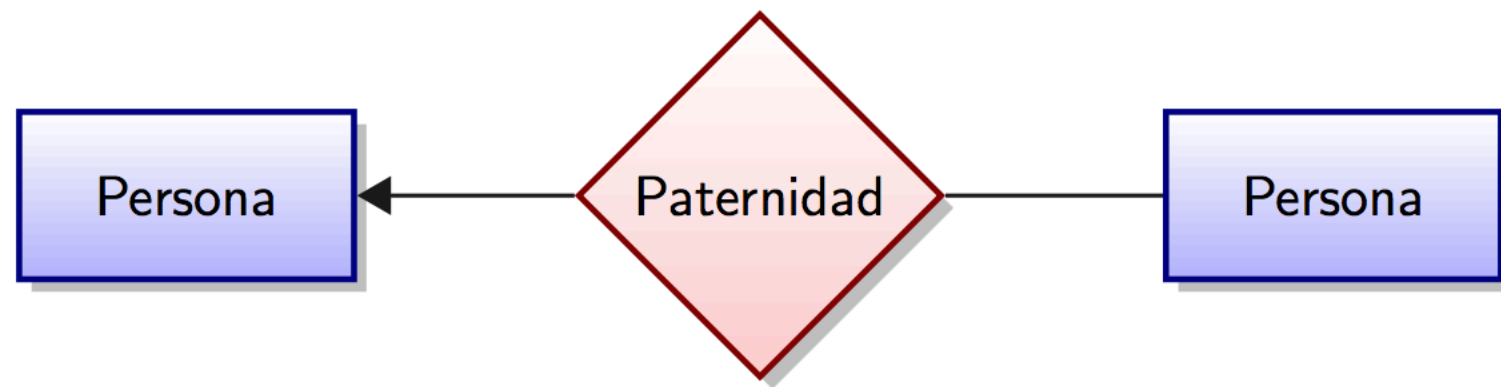
Llaves



No se puede especificar si hay más de una llave

Restricciones de integridad

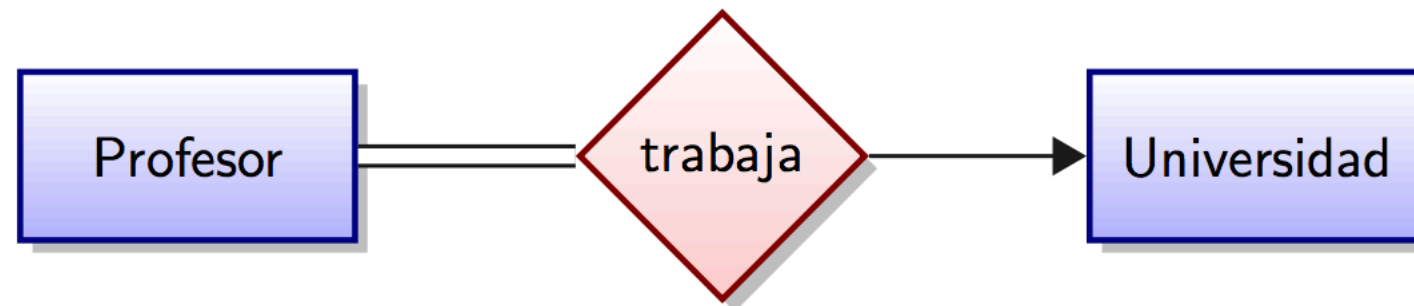
Valor único



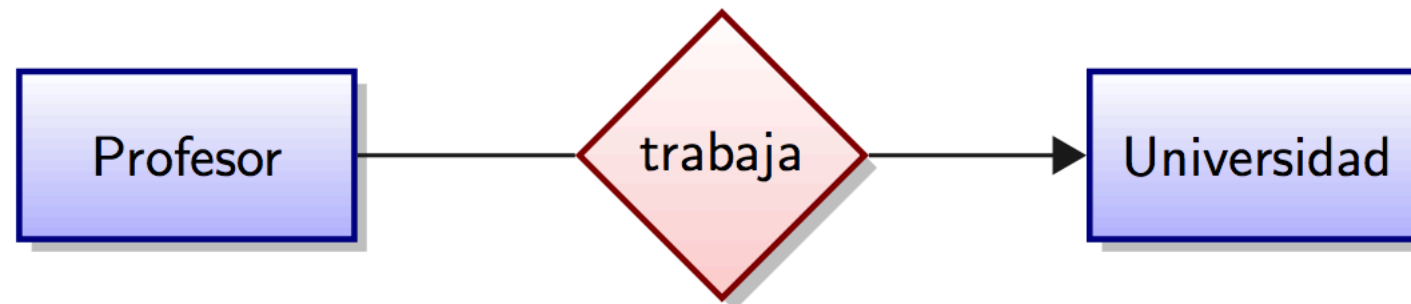
Restricciones de integridad

Participación + Llave

Cada profesor necesariamente trabaja en una única universidad

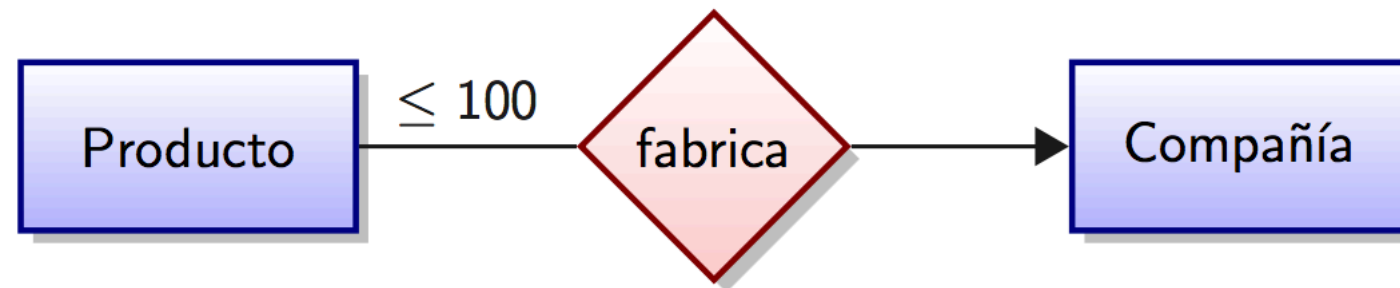


Cada profesor puede trabajar en una única universidad (pero puede estar sin trabajo!)



Restricciones de integridad

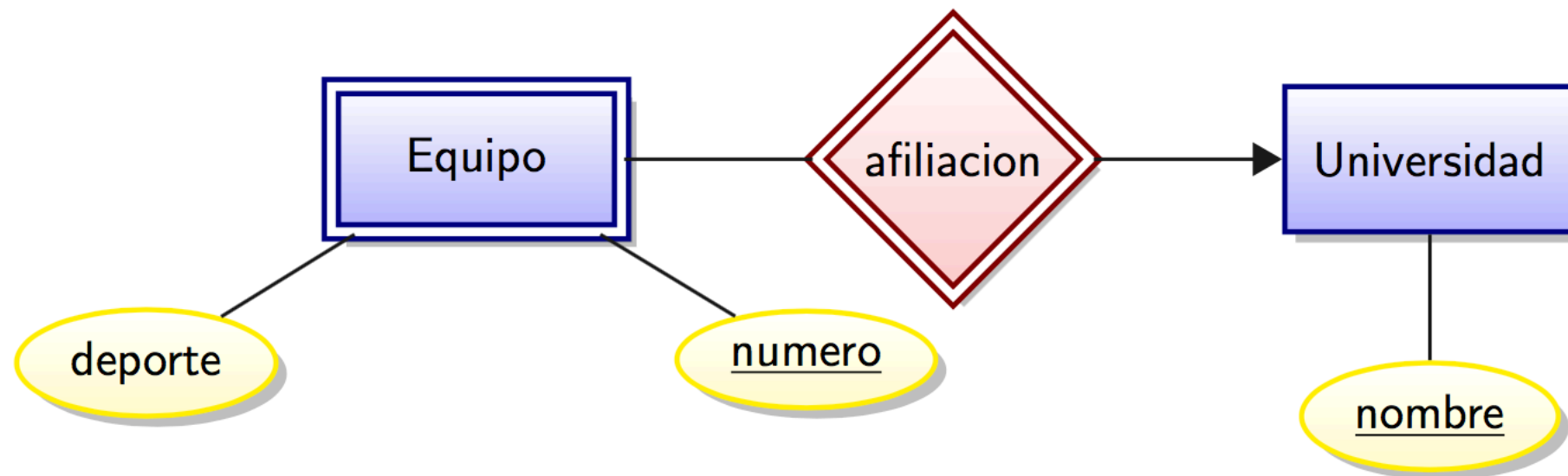
Restricciones numéricas



Restricciones de integridad

Entidades débiles

Aquí, la llave de equipo es su número y la llave de universidad



El equipo no tiene sentido si no existe la universidad!