

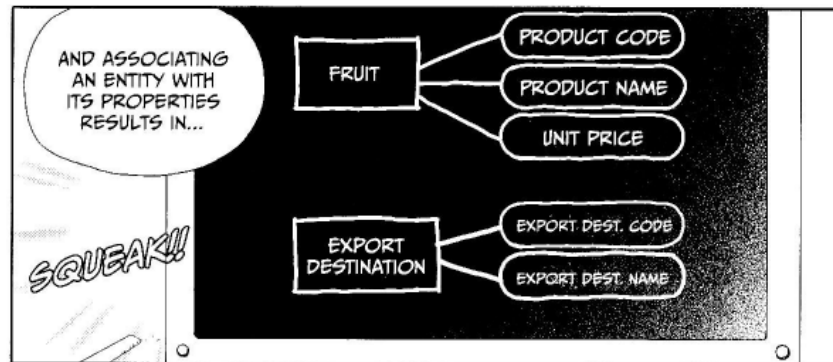
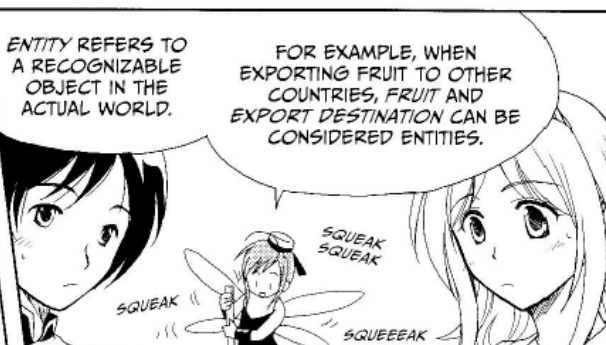
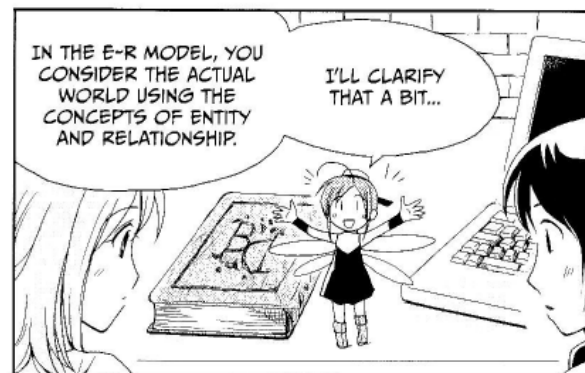
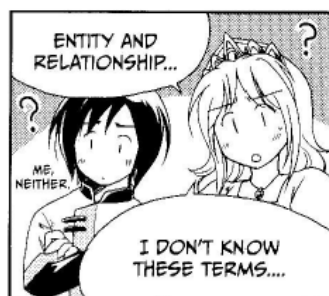
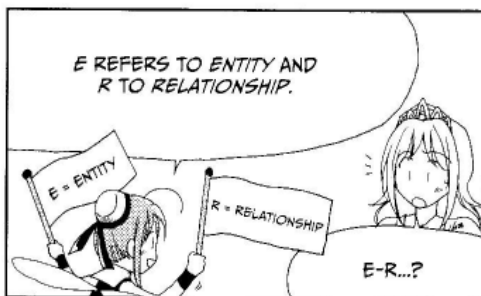
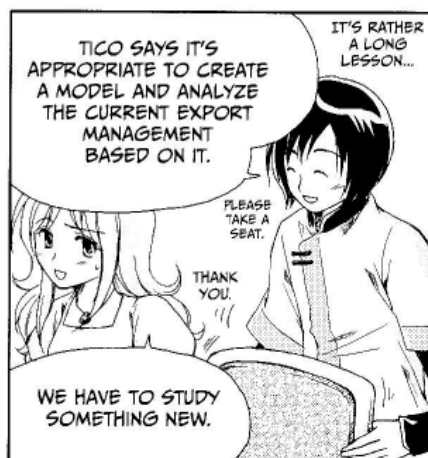
Those who are enamored of practice without theory are like a pilot who goes into a ship without rudder or compass and never has any certainty where he is going. Practice should always be based on a sound knowledge of theory.

Leonardo Da Vinci (1452-1519)

Repaso DER Modelo Relacional

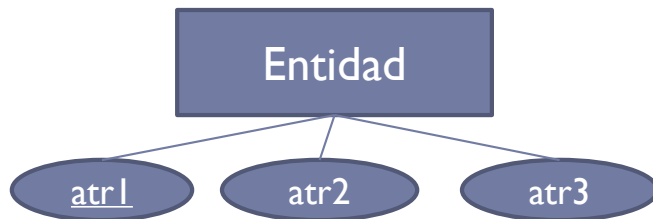
Dr. Gerardo Rossel

REPASO DE MODELO DE ENTIDAD RELACION



MER - Repaso

- ▶ **Modelo Entidad-Relación (MER):** Herramienta que permite realizar una abstracción o modelo de alguna situación de interés presente en el mundo real.
- ▶ El MER se realizará utilizando la técnica Diagramas de Entidad Relación (DER).



Grado: Número de entidades que participan de una interrelación.

- Unarias
- Binarias
- Ternarias



MER - Repaso

- ▶ **Cardinalidad:** Se refiere a la cantidad de elementos de una entidad que puedan estar interactuando a través de la interrelación, con elementos de otra entidad.

- **Uno a uno**
- **Uno a muchos**
- **Muchos a muchos**



- ▶ **Participación:** Se refiere a la cantidad de elementos de una entidad que puedan estar interactuando a través de la interrelación, con elementos de otra entidad.

- **Parcial**
- **Total**



MER – Notación

- ▶ Existe una variedad de notación para realizar un DER, dependiendo de las diferentes bibliografías:

Database Systems – The complete book
(García Molina, Ullman, Widon)



Fundamentals of Database Systems
(Elmasri, Navathe)



Basado en Fundamentals of Database Systems
(Elmasri, Navathe)



MER – Notación

► Participación parcial:

Fundamentals of Database Systems
(Elmasri, Navathe)

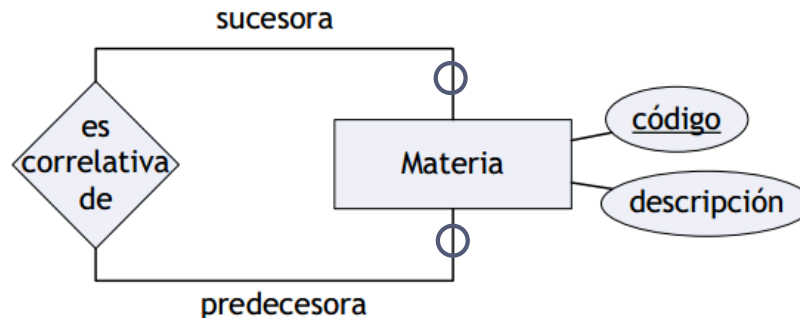


Basado en Fundamentals of Database Systems
(Elmasri, Navathe)



MER - Repaso

- ▶ Rol: Cada entidad que participa en una interrelación, lo hace en un rol particular. Este rol ayuda a explicar el significado de la interrelación
- ▶ **Importante:**
 - ▶ En relaciones unarias **SIEMPRE** deben aclararse los roles, para evitar ambigüedades, dado que se trata de roles diferentes para la misma entidad:

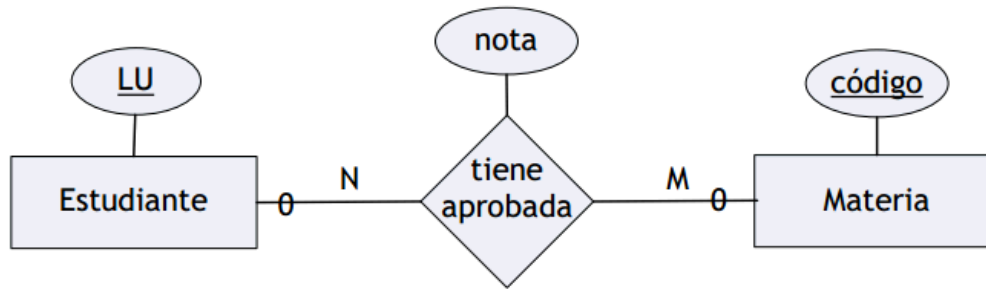


A su vez, en relaciones unarias **SIEMPRE** debe haber al menos una participación parcial

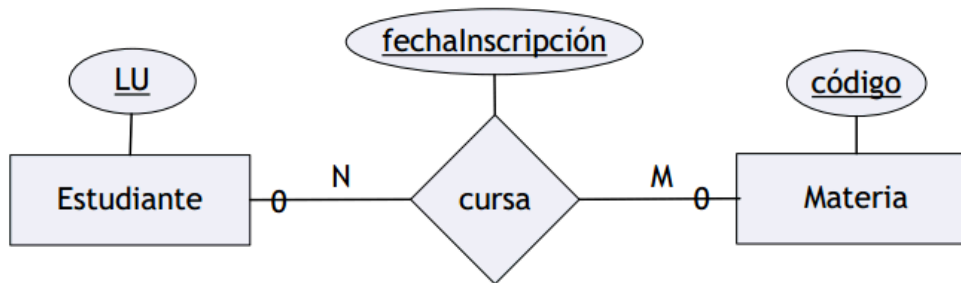
MER - Repaso

► Importante:

- Las **UNICAS** interrelaciones que admiten atributos (descriptivos o identificatorios) son las del tipo M:N.

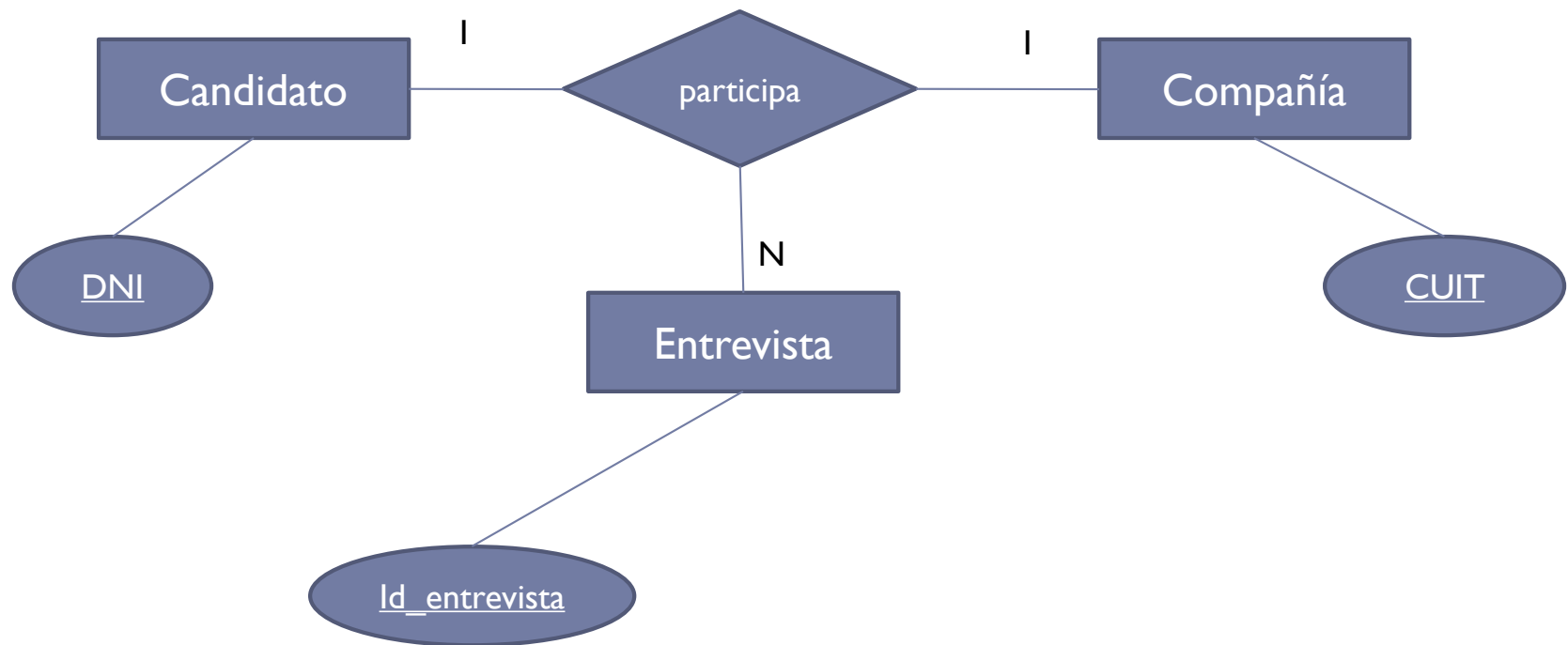


- Los atributos identificatorios en las interrelaciones, permiten que se repitan pares ordenados, pero para un atributo dado deben tener diferentes valores:

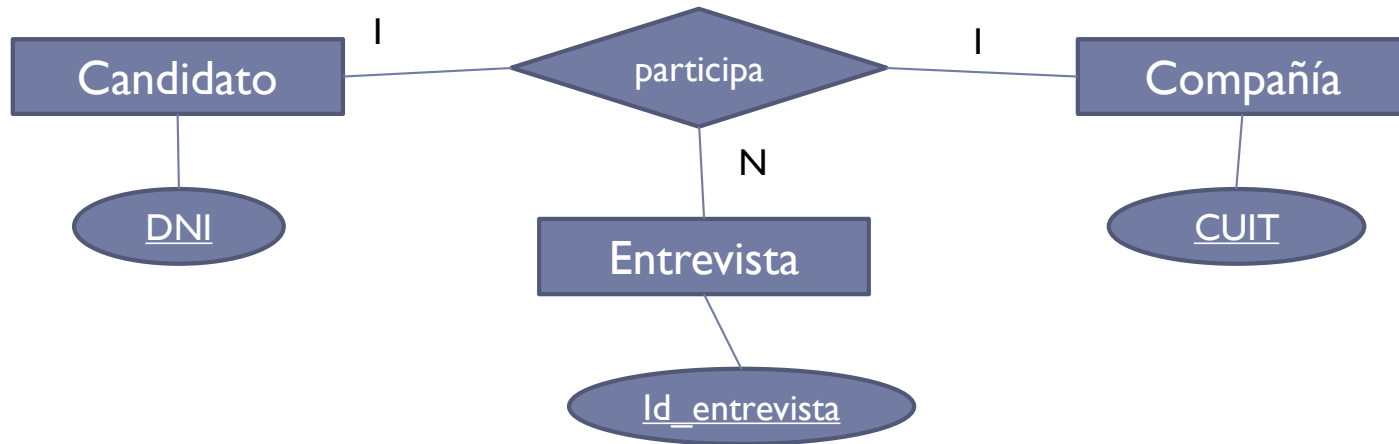


Interrelaciones Ternarias

- ▶ En este tipo de interrelaciones participan tres entidades en forma simultánea.
- ▶ En este caso la terna (ó 3-upla) posee un elemento de cada entidad participante

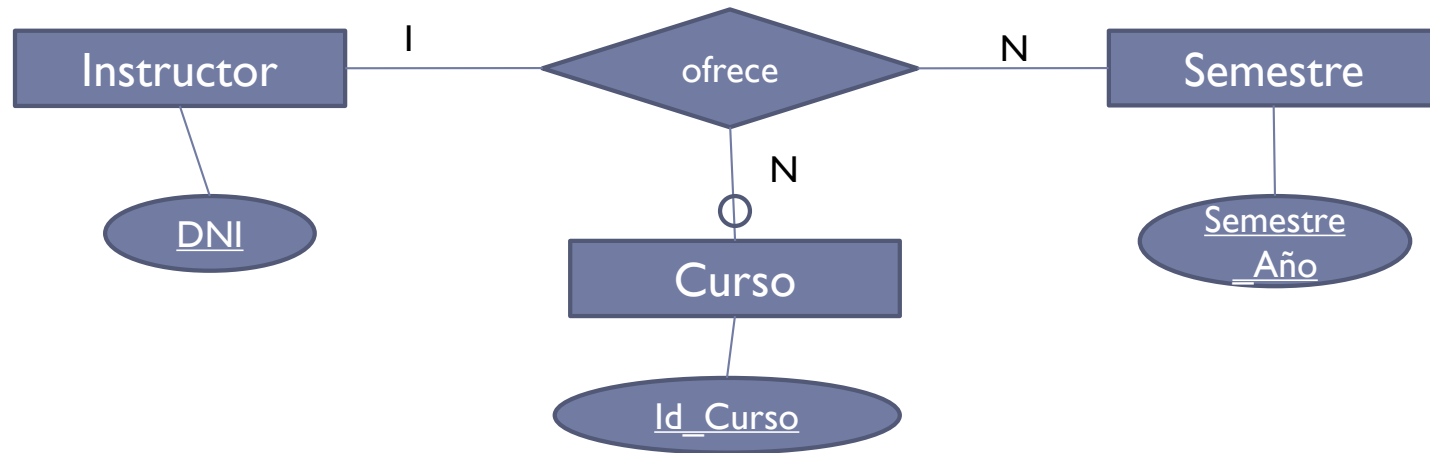


Interrelaciones Ternarias: Como se lee?



- ▶ Se toman de a **pares**:
 - ▶ “Cada candidato en una compañía puede realizar N entrevistas”
 - ▶ “Cada entrevista de un candidato corresponde a una compañía”
 - ▶ “Cada entrevista de una compañía corresponde a un candidato”

Interrelaciones Ternarias: Participación



- ▶ Para evaluar la participación, se toma **un elemento** de cada entidad:
 - ▶ “Todo elemento de la entidad Instructor debe participar al menos una vez de la interrelación”
 - ▶ “Todo elemento de la entidad Semestre debe participar al menos una vez de la interrelación”
 - ▶ “Los elementos de la entidad curso pueden no participar de la interrelación”

En un elemento de la interrelación ternaria “ofrece”, **SIEMPRE** participan elementos de las 3 entidades

Interrelaciones Ternarias: Participación

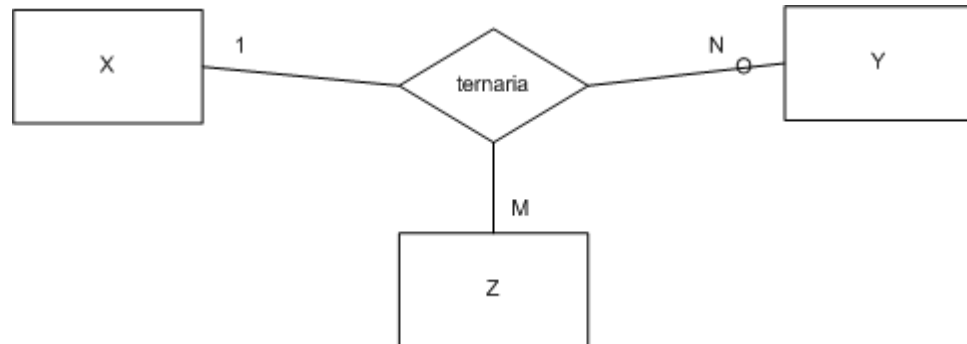
- Un elemento en una interrelación ternaria **SIEMPRE** requiere la participación de las 3 entidades. Es decir, en una terna de una interrelación ternaria, no puede tener elementos nulos:

| Instructor | Curso | Semestre |
|-------------|-------------|-------------|
| 32 | 5 | 1-2016 |
| 32 | 7 | 1-2017 |
| 35 | <i>null</i> | <i>null</i> |
| <i>null</i> | 10 | 1-2017 |



Interrelaciones Ternarias : Participación

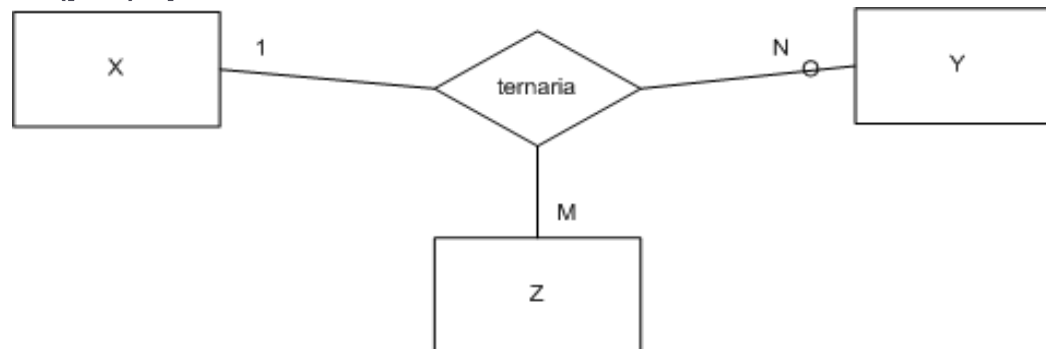
- ▶ Un elemento de la ternaria **SIEMPRE** tendrá un **elemento de cada entidad participante**
- ▶ Es decir, en la relación ternaria podremos tener:
 - ▶ $\langle x1, y2, z3 \rangle$
 - ▶ $\langle x2, y2, z3 \rangle$
- ▶ Pero **NUNCA**
 - ▶ $\langle x1, \text{null}, z2 \rangle$



Interrelaciones Ternarias

- ▶ La **cardinalidad** se define tomando de a dos entidades.

- ▶ Un par (x,z) puede estar relacionado con muchos y's
- ▶ Un par (x,y) puede estar relacionado con muchos z's
- ▶ Un par (y,z) puede estar relacionado con un solo x

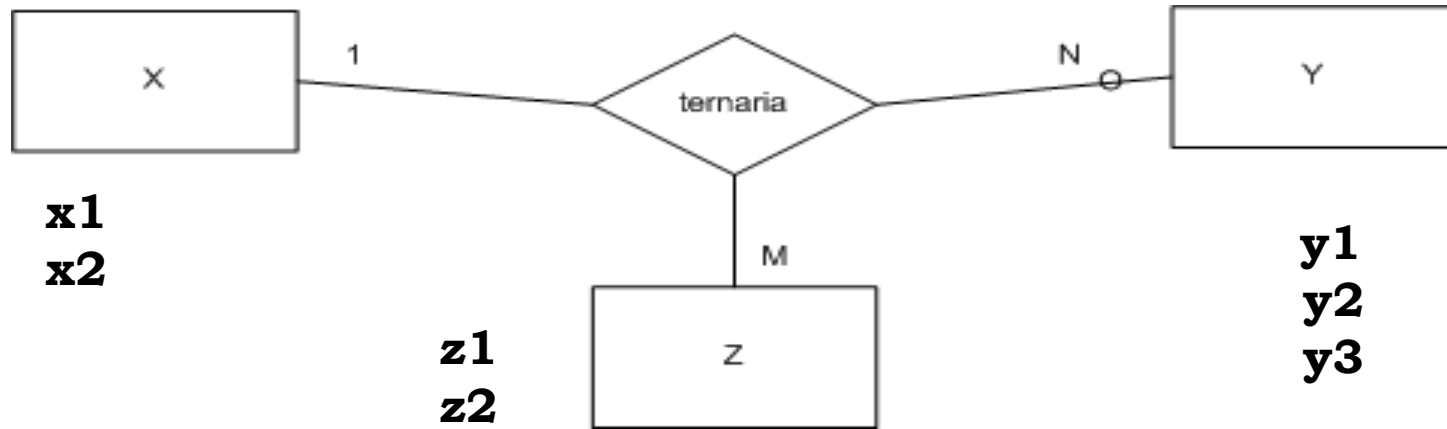


- ▶ La **participación** se define individualmente

- ▶ X participa totalmente en la ternaria
- ▶ Y participa parcialmente en la ternaria
- ▶ Z participa totalmente en la ternaria

Interrelaciones Ternarias

► Restricciones, Cardinalidad y Participación



VÁLIDA ○ **INVÁLIDA??**

$\langle x1, y1, z1 \rangle$
 $\langle x1, y1, z2 \rangle$



$\langle x1, y1, z1 \rangle$
 $\langle x1, y1, z2 \rangle$
 $\langle x2, y2, z2 \rangle$



$\langle x1, y1, z1 \rangle$
 $\langle x1, y1, z2 \rangle$
 $\langle x1, y2, z2 \rangle$
 $\langle x2, y2, z2 \rangle$



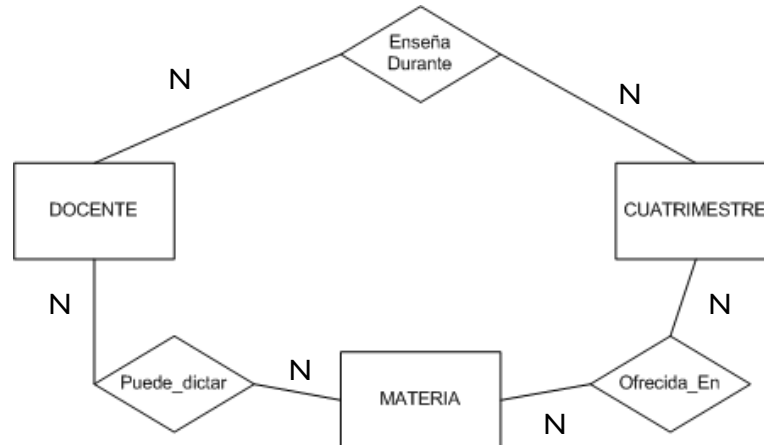
¿Interrelaciones Binarias o Ternarias?

- Los **docentes** pueden dictar potencialmente varias materias según su especialidad. Las **materias** son ofrecidas durante **cuatrimestres**, a su vez es posible que diferentes docentes dicten la misma materia en distintos cuatrimestres. Los docentes dictan sólo una materia por cuatrimestre

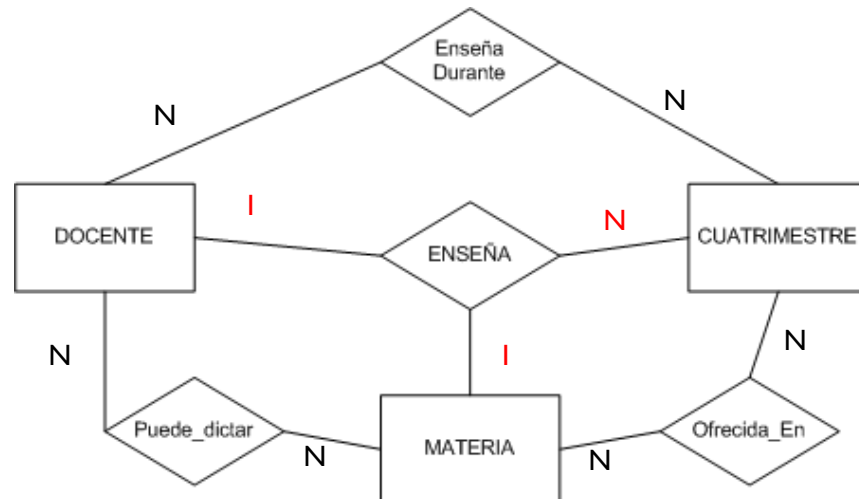


¿Interrelaciones Binarias o Ternarias?

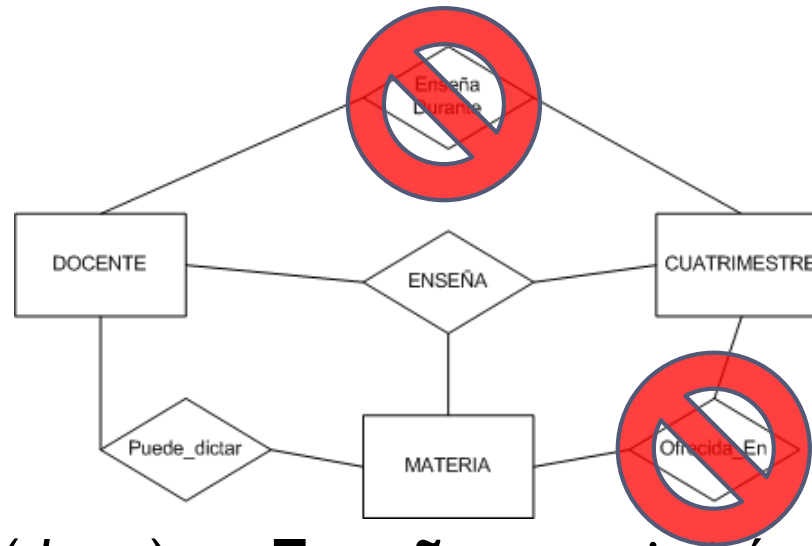
► Quedaría:



- ¿Como capturamos la información: *José enseñó Base de Datos en el 1er Cuat de 1967?*
- ¿Hay un mejor modelo?

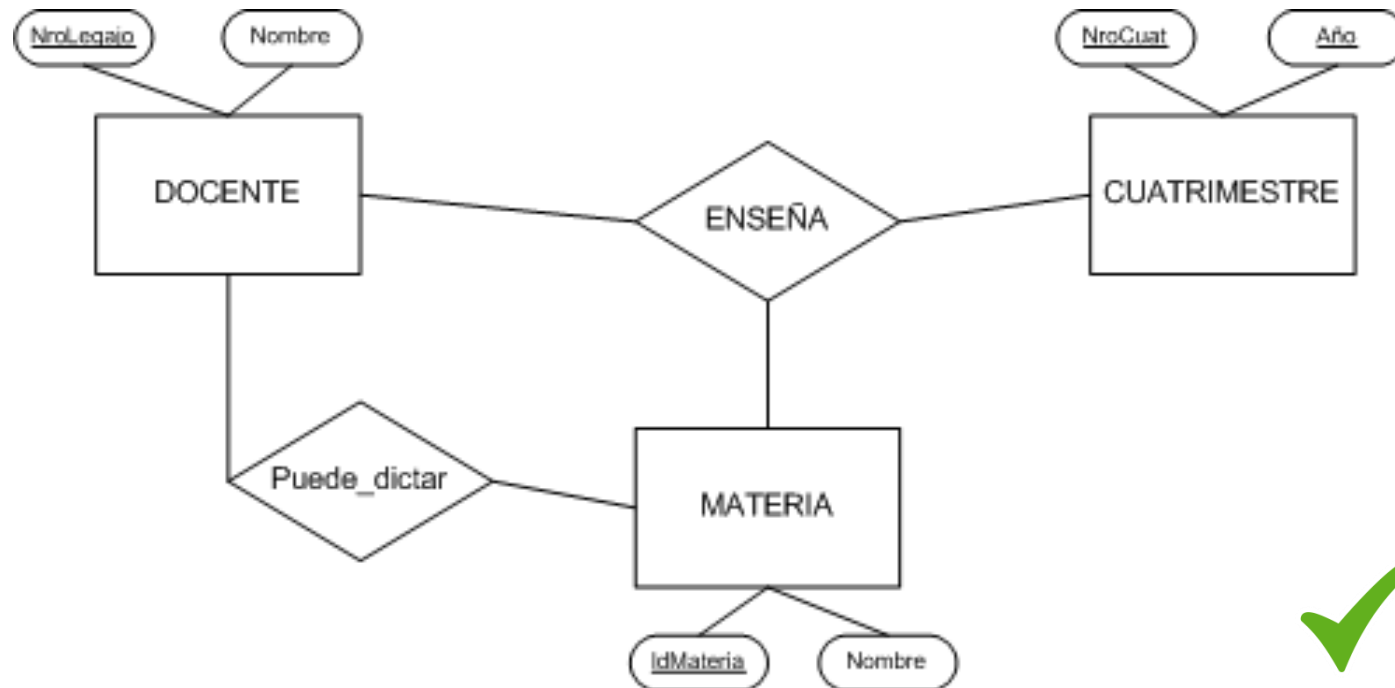


Interrelaciones Ternarias



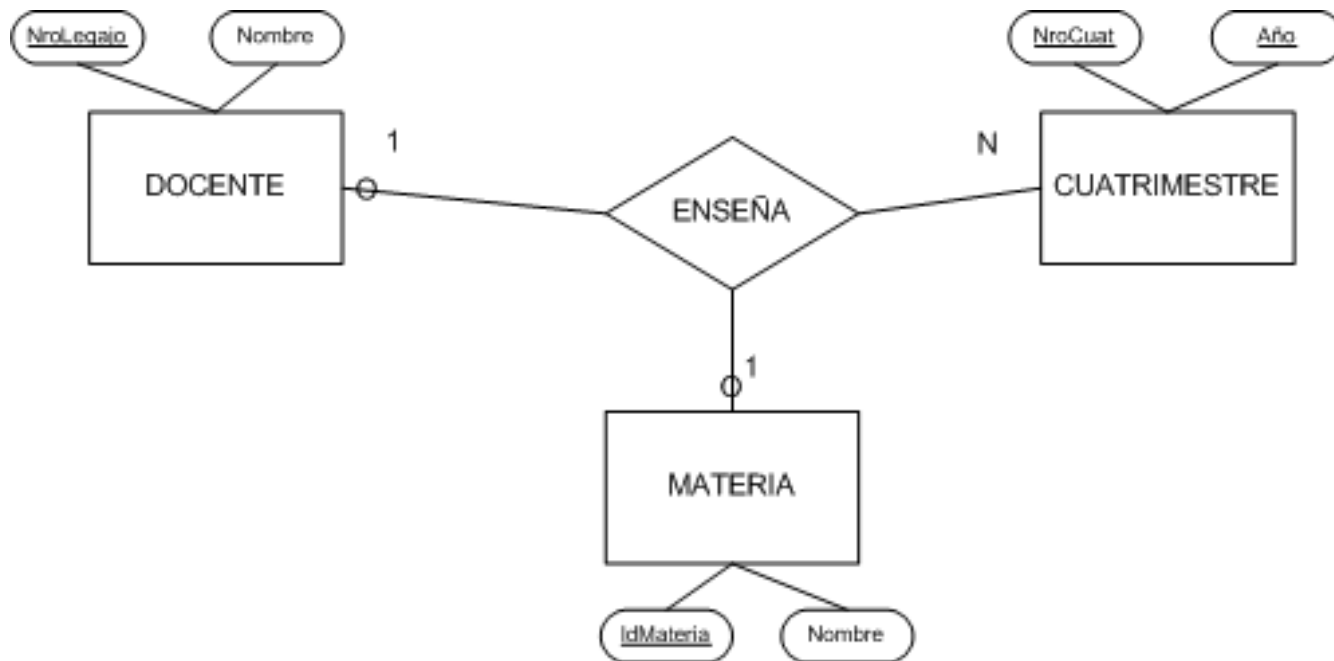
- ▶ Una instancia (d,c,m) en **Enseña** no existiría si no existieran instancias (d,c) en **Enseña_Durante**, (c,m) en **Ofrecida_En** y (d,m) en **Puede_Dictar**.
- ▶ La reversa no es válida.
- ▶ Se pueden deducir las instancias de **Enseña_Durante** y **Ofrecida_En** a partir de Enseña (**REDUNDANTES!!**).

Interrelaciones Ternarias



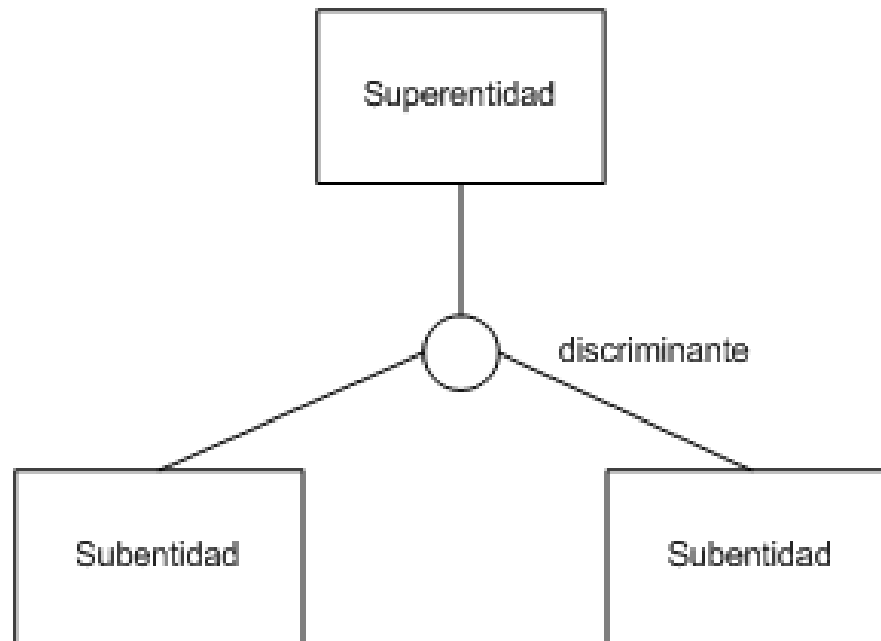
Interrelaciones Ternarias

- Agregamos entonces cardinalidad y participación
 - Estas restricciones dependen del espacio del problema en concreto.



Jerarquías

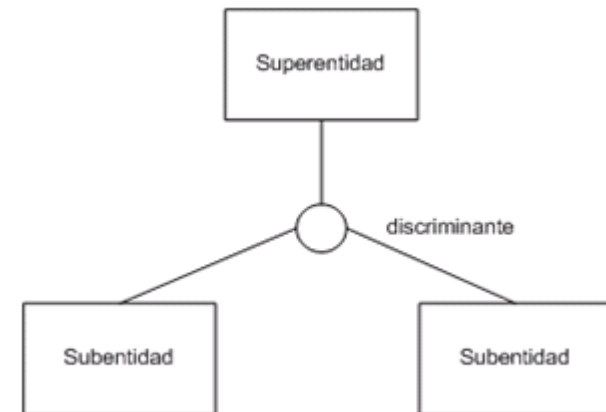
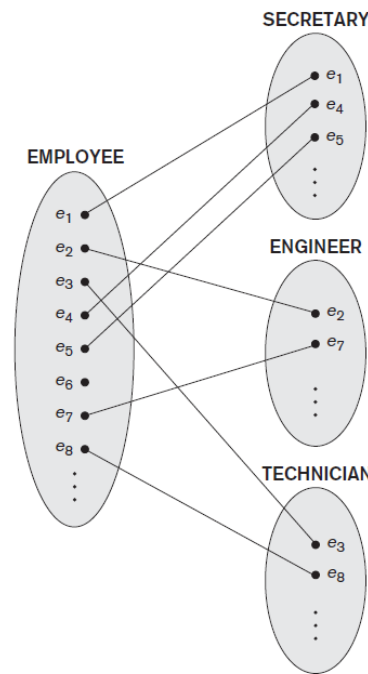
Muchas veces, un conjunto entidad puede contener entidades que poseen atributos específicos que no están asociados al resto de las entidades del conjunto. Para estos casos se definen *subentidades*, cada una de las cuales tendrá uno o varios atributos especiales



Utilizaremos la relación “es_un” para indicar que una entidad es una *subentidad* de otra

Jerarquías

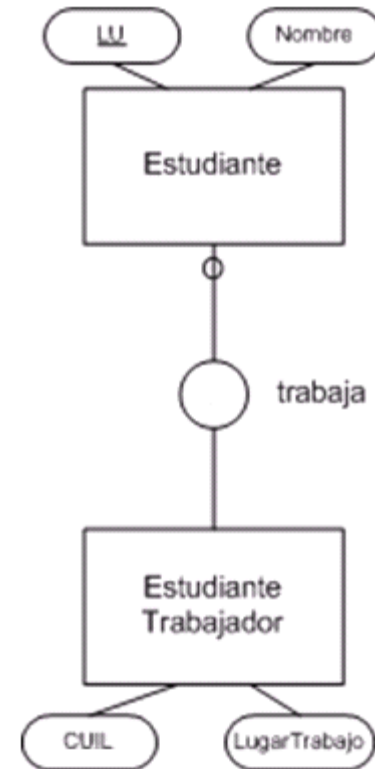
- ▶ Entidad Padre y una o más entidades hijas.
- ▶ Las entidades hijas o *subentidades* heredan la clave de la *superentidad* o entidad padre.
- ▶ Relación *es_un*
- ▶ Cobertura
 - ▶ Total
 - ▶ Parcial
- ▶ Solapamiento
 - ▶ Disjuntas
 - ▶ Con solapamiento



Jerarquías-Coberturas

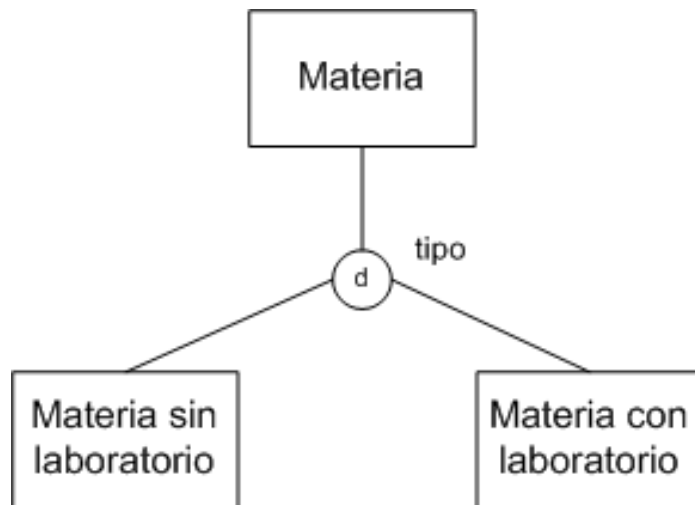


TOTAL

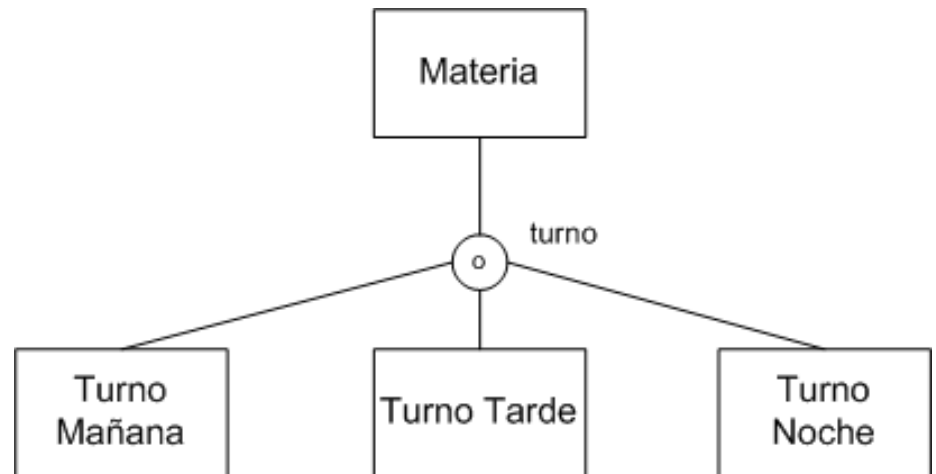


PARCIAL

Jerarquías -Solapamiento



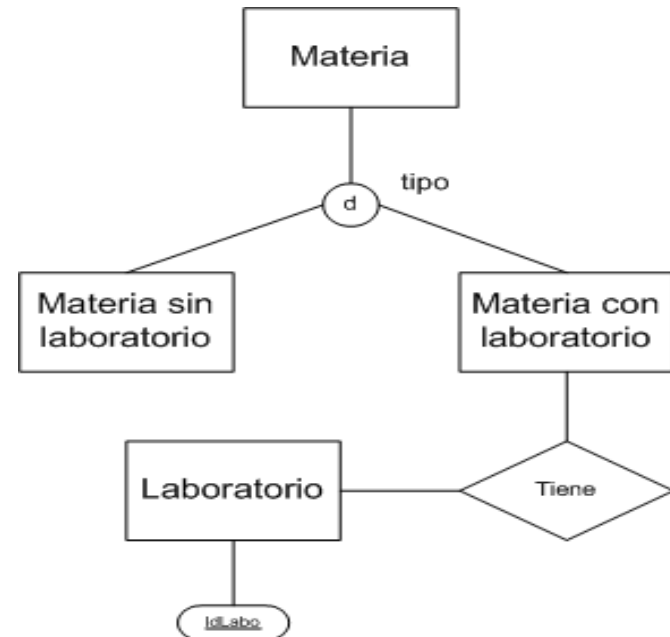
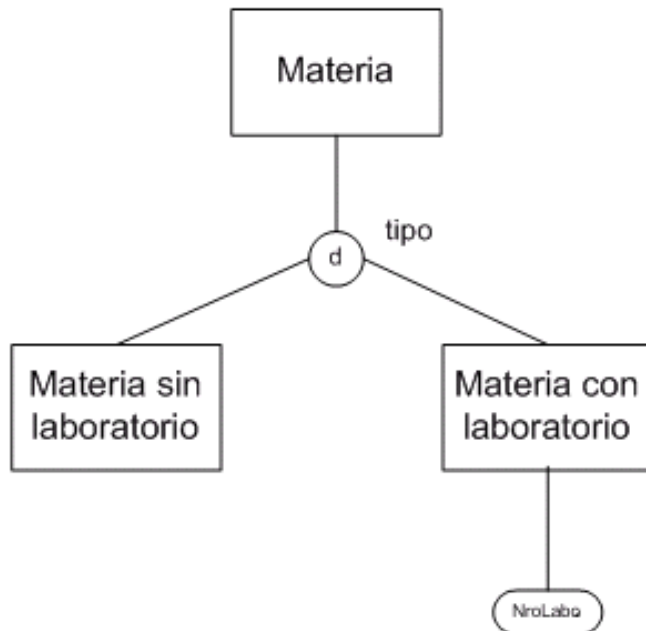
Disjunta



Con solapamiento

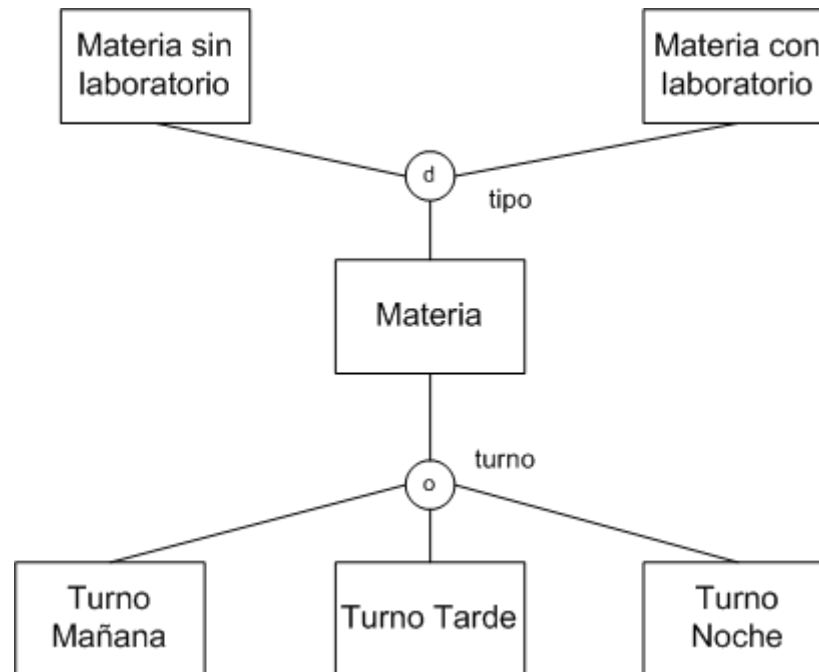
Jerarquías

- La definición de una jerarquía puede estar guiada por la *existencia de atributos distintos* y también por la *existencia de interrelaciones diferentes*



Jerarquías

- Cada jerarquía tiene una semántica. Es posible que una misma entidad tenga diferentes especializaciones con distinta semántica.



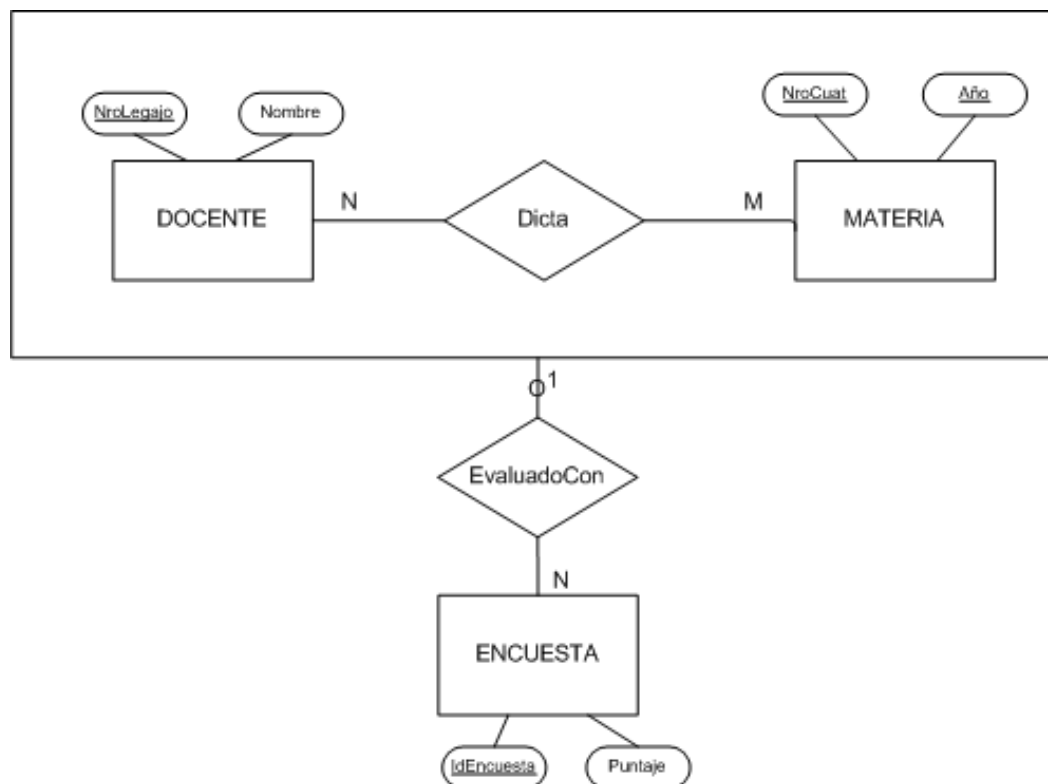
Agregación

- ▶ Se registran datos de los docentes y de las materias dictadas. Una materia puede ser dictada por muchos docentes y un docente puede dictar muchas materias. Además, en algunos cursos (no en todos) se realizan encuestas de evaluación docente. Cada encuesta corresponde a un docente que dicta una materia, y cada docente en una materia puede tener más de una encuesta (o ninguna).
- ▶ ¿Por qué no una ternaria?



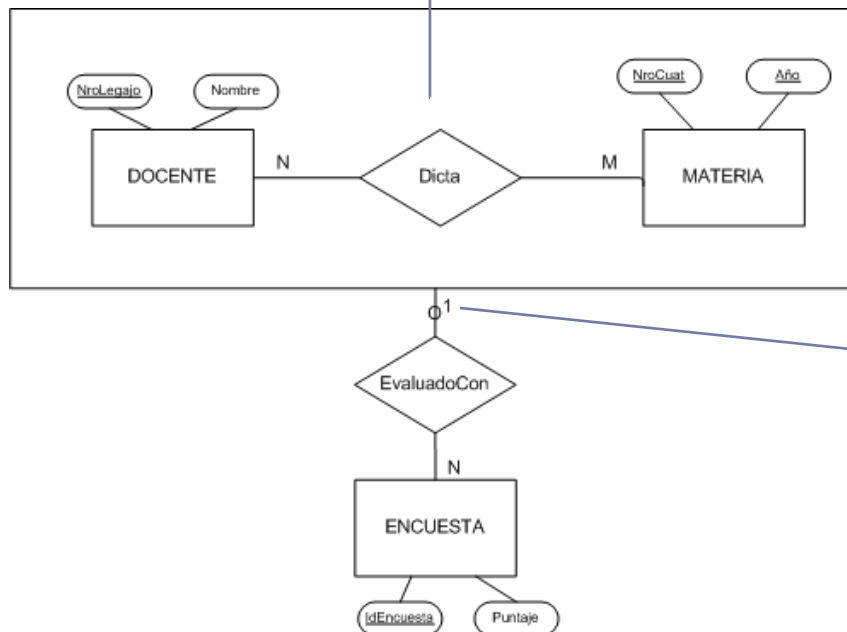
Agregación

- La agregación es una abstracción en la cual una interrelación (junto con sus entidades vinculadas) es tratada como una entidad de alto nivel y puede participar de interrelaciones



Agregación: Consideraciones

- ▶ En una relación ternaria, como se detalló anteriormente, **SIEMPRE** están los tres elementos de la terna presentes.
- ▶ En una agregación, uno de los elementos de la terna es nulo. Por eso **NO** puede modelarse a través de una relación ternaria.

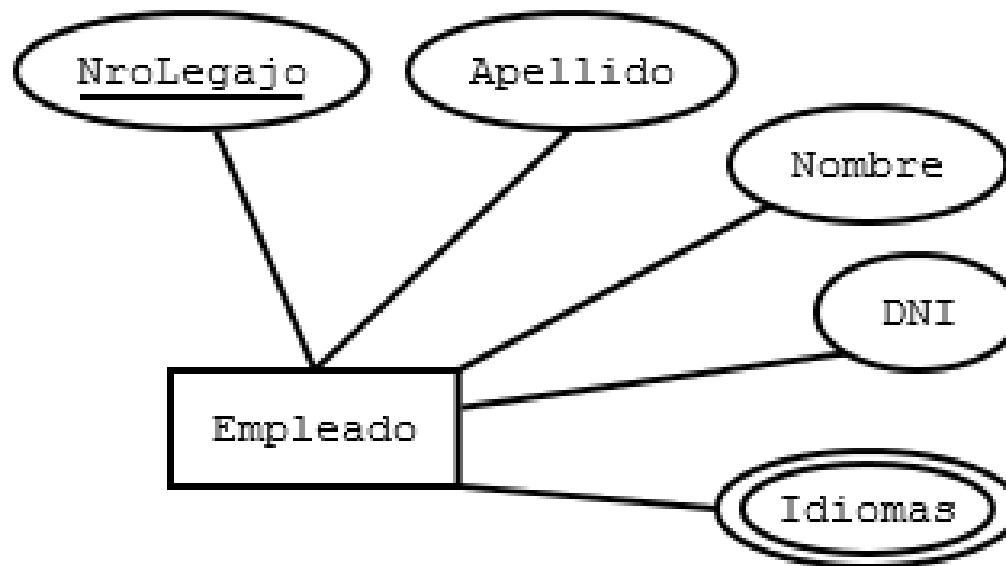


La interrelación es una entidad de alto nivel, por eso **sólo es posible** cuando se trata de relaciones M:N

Participación parcial: porque **puede NO** haber encuesta

Atributos simples y multivaluados

- ▶ **Atributo simple:** Es un atributo que contiene un valor único para cada ocurrencia de un tipo de entidad.
- ▶ **Atributo multivaluado:** Es un atributo que contiene varios valores para cada ocurrencia de un tipo de entidad





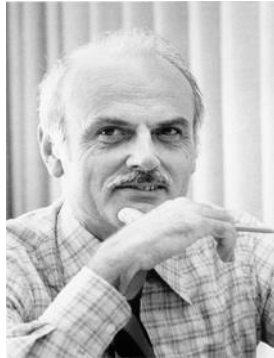
Modelo Relacional

Modelo Relacional: Contexto Histórico

- ▶ 1950...Datos mantenidos en Archivos.
- ▶ 1966-1980 Pre-relacional.
 - ▶ Modelo Jerárquico - Information Management System (IMS) de IBM (1966).
 - ▶ Modelo de Red - CODASYL DBTG: IDS (Integrated Data Store) - Charles Bachman (General Electric)
- ▶ 1970-1980: La guerra de los modelos.*CODASYL DBTG vs Relational Model*
- ▶ 1970 (junio) Ted Codd: A Relational Model of Data for Large Shared Data Banks in Communications of the ACM.
- ▶ 1971 CODASYL DBTG: publica su estándar

Debate: Ted Codd y Charles Bachman

1975 ACM SIGMOD(Special Interest Group on Management of Data).



- ▶ Nada tan complicado como la propuesta de DBTG puede ser la forma correcta de administrar datos
- ▶ Las consultas orientadas a conjuntos son dificultosas.
- ▶ El modelo de red no tiene fundamentos formales en teoría matemática

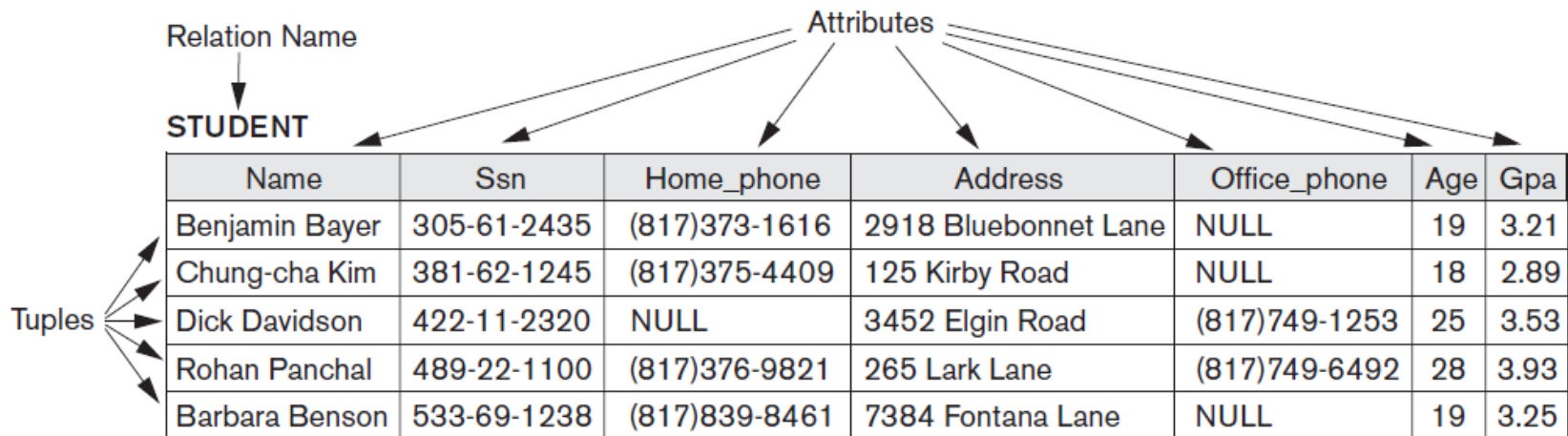


- ▶ El modelo relacional es demasiado matemático
- ▶ No se puede construir una implementación eficiente del modelo relacional
- ▶ Las aplicaciones necesitan procesar los datos de a un registro por vez.

Modelo Relacional - Definiciones

- ▶ ‘A relational model of data for large shared data banks’ (Codd, 1970).
- ▶ Representa a la base de datos como un conjunto de relaciones.
- ▶ Intuitivamente, una relación puede pensarse como una **tabla**, con **filas** y **columnas**.
 - ▶ Cada **tabla** es una relación y tiene su nombre
 - ▶ Cada columna de la tabla representa un **atributo**, asociado a un conjunto de valores posibles que puede tomar. A este conjunto es a lo que llamamos **dominio** del atributo
 - ▶ Cada fila, a la que denominaremos **tupla** está formada por un conjunto de valores de datos relacionados

Modelo Relacional



Modelo Relacional - Definiciones

- ▶ Un **dominio D** es un conjunto de valores atómicos. Por lo que respecta al modelo relacional, atómico significa indivisible.
- ▶ Una **relación** se tiene un **esquema** (o intención de la relación) y una **extensión** (estado o instancia)
- ▶ El esquema de la relación, que se escribe $R(A_1, A_2, \dots, A_n)$ consiste en un nombre de relación **R** y un conjunto de atributos $\{A_1, A_2, \dots, A_n\}$.
- ▶ La **aridad** de una relación es la cantidad de atributos que tiene.
- ▶ Un atributo **A_i** es *el nombre del rol* que ejerce algún dominio **D** en un esquema de relación. Si D es el dominio de A_i se escribe como **dom(A_i)**.

Modelo Relacional - Definiciones

- La extensión de la relación de esquema $R(A_1, A_2, \dots, A_n)$ denotada como $r(R)$ es un **conjunto de tuplas** t_i ($i = 1, 2, \dots, m$), donde cada tupla t_i es, a su vez un conjunto de pares $t_i = \{ \langle A_1:v_{i1} \rangle, \langle A_2:v_{i2} \rangle \dots \langle A_n:v_{in} \rangle \}$ y, para cada par $\langle A_j:v_{ij} \rangle$, se cumple que v_{ij} es un valor de **dom(A_j)**,
Atención con un valor especial que denominaremos nulo (**null**).
 - En esta definición de tupla no importaría el orden. Podríamos definir también si tomamos el orden de los atributos como relevante a una tupla como $t = \langle v_1, v_2, \dots, v_n \rangle$, donde cada v_i representa un valor del atributo A_i .
- Notar entonces que la extensión de una relación es un subconjunto del producto cartesiano de una lista de dominios.
 - $r(R) \subseteq (dom(A_1) \times dom(A_2) \times \dots \times dom(A_n))$

| | | | | | | | | |
|---------------|--|----------------|-------------|---------------|----------------------|---------------|-----|------|
| Relation Name | | Attributes | | | | | | |
| STUDENT | | | | | | | | |
| | | Name | Ssn | Home_phone | Address | Office_phone | Age | Gpa |
| Tuples | | Benjamin Bayer | 305-61-2435 | (817)373-1616 | 2918 Bluebonnet Lane | NULL | 19 | 3.21 |
| | | Chung-cha Kim | 381-62-1245 | (817)375-4409 | 125 Kirby Road | NULL | 18 | 2.89 |
| | | Dick Davidson | 422-11-2320 | NULL | 3452 Elgin Road | (817)749-1253 | 25 | 3.53 |
| | | Rohan Panchal | 489-22-1100 | (817)376-9821 | 265 Lark Lane | (817)749-6492 | 28 | 3.93 |
| | | Barbara Benson | 533-69-1238 | (817)839-8461 | 7384 Fontana Lane | NULL | 19 | 3.25 |

Modelo Relacional - Interpretación

- ▶ Una esquema de relación puede ser interpretado como un predicado y cada tupla puede interpretarse como valores que satisfacen. ese predicado.
- ▶ **CWA: closed world assumption.**

Modelo Relacional - Claves

- ▶ En este modelo, aparece un concepto importante
 - ▶ SuperClave
 - ▶ Clave: conjunto minimal de atributos que definen unívocamente a las tuplas.
 - ▶ Sea K una clave, E una relación y e_i, e_j tuplas
 - ▶ $\forall e_i, e_j \in E: e_i.K = e_j.K \rightarrow e_i = e_j$
- ▶ Las relaciones pueden tener varias claves
 - ▶ A las claves de la relación se las denomina *Claves Candidatas* (Candidate Keys, CK)
 - ▶ Una de ellas será elegida como *Clave Primaria* (Primary Key, PK)
- ▶ A su vez, pueden referenciar a claves de otras relaciones
 - ▶ Se conocen como *Claves Externas* (Foreign Keys, FK)

Base de Datos Relacional

- ▶ An *integrity constraint* (*constraint* for short) is basically just a boolean expression that must evaluate to TRUE. In the case of departments and employees, for example, we might have a constraint to the effect that SALARY values must be greater than zero.
- ▶ A **relational database schema** S is a set of relation schemas $S = \{R_1, R_2, \dots, R_m\}$ and a set of **integrity constraints** IC
- ▶ A **relational database state** DB of S is a set of relation states $DB = \{r_1, r_2, \dots, r_m\}$ such that each r_i is a state of R_i and such that the r_i relation states satisfy the integrity constraints specified in IC .

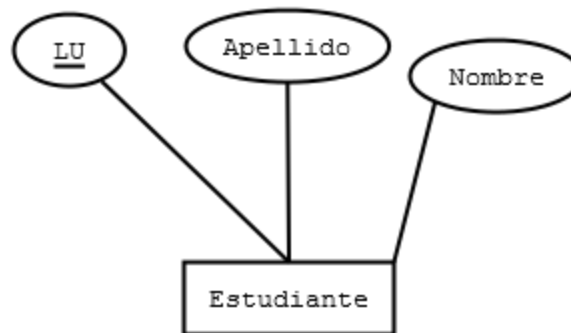
Restricciones (Constraints)

- ▶ Inherentes al modelo.
- ▶ Expresadas en la definición del esquema
 - ▶ Claves, No NULL, Dominio
 - ▶ Integridad Referencial
 - ▶ Integridad de las Relaciones.
- ▶ Las restricciones de integridad (*integrity constraints*) se especifican sobre un esquema de base de datos y se espera que se cumplan sobre todo estado válido de la base de datos

Pasaje DER a MR

Transformación del MER al MR

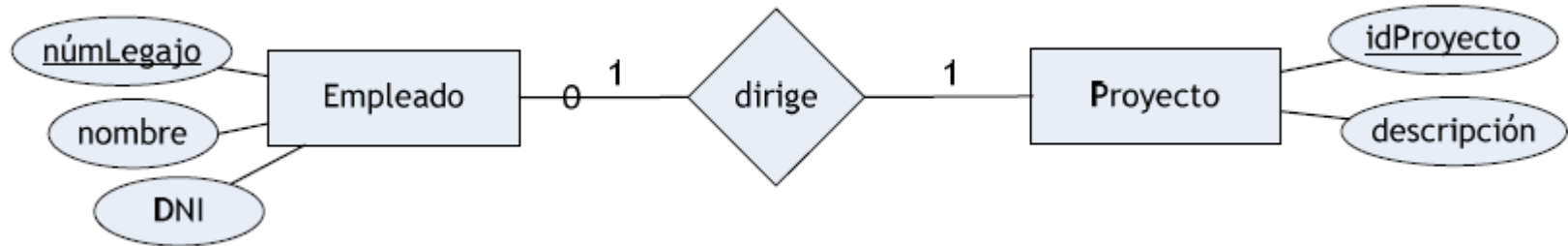
► Entidades



Estudiante(LU,Apellido,Nombre)

Transformación del MER al MR

► Interrelaciones 1 : 1



Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción, númLegajo)

PK = CK = {idProyecto}

FK = {númLegajo}

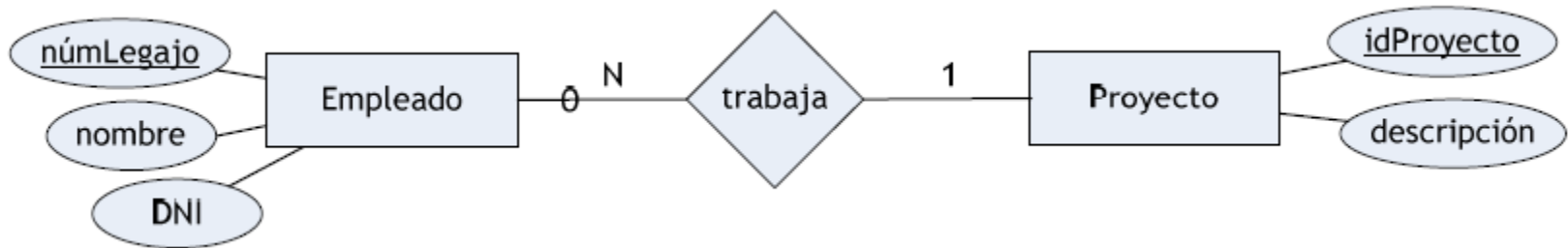
Establecemos las siguientes **restricciones adicionales**:

Empleado.númLegajo puede no estar en *Proyecto.númLegajo*

Proyecto.númLegajo debe estar en *Empleado.númLegajo*

Transformación del MER al MR

► Interrelaciones 1:N



Empleado(númLegajo, nombre, DNI, idProyecto)

PK = {númLegajo}

CK = {númLegajo, DNI}

FK = {idProyecto}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

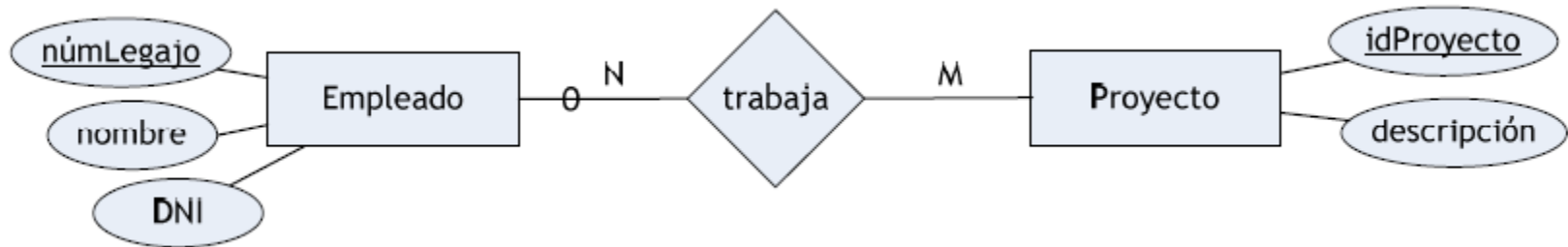
Establecemos las siguientes **restricciones adicionales**:

Empleado.idProyecto es nulo o está en *Proyecto.idProyecto*

Proyecto.idProyecto debe estar en *Empleado.idProyecto*

Transformación del MER al MR

► Interrelaciones N:M



Empleado(númLegajo, nombre, DNI)

PK = {númLegajo}

CK = {númLegajo, DNI}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Trabaja(númLegajo, idProyecto)

PK = CK = {(númLegajo, idProyecto)}

FK = {númLegajo, idProyecto}

Establecemos las siguientes **restricciones adicionales**:

Empleado.númLegajo puede no estar en *Trabaja.númLegajo*

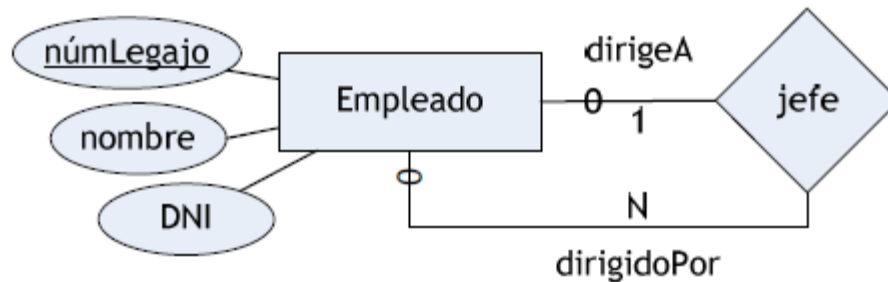
Proyecto.idProyecto debe estar en *Trabaja.idProyecto*

Trabaja.númLegajo debe estar en *Empleado.númLegajo*

Trabaja.idProyecto debe estar en *Proyecto.idProyecto*

Transformación del MER al MR

► Interrelaciones Unarias



Empleado(númeroLegajo, nombre, DNI, númeroLegajoJefe)

PK = {númeroLegajo}

CK = {númeroLegajo, DNI}

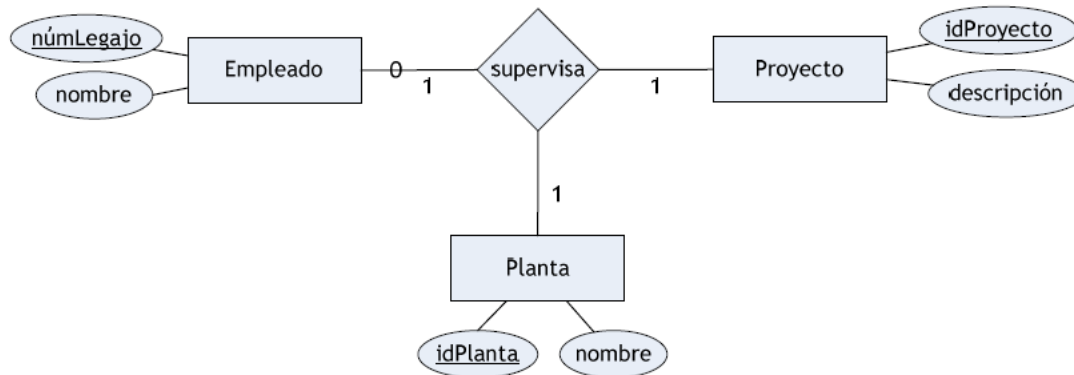
FK = {númeroLegajoJefe}

Empleado.númeroLegajoJefe puede ser nulo o debe estar en Empleado.númeroLegajo

Empleado.númeroLegajo puede no estar en Empleado.númeroLegajoJefe

Transformación del MER al MR

► Ternarias 1:1:1



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

CK = {(númLegajo, idProyecto), (númLegajo, idPlanta), (idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

Supervisa.númLegajo debe estar en Empleado.númLegajo

Supervisa.idProyecto debe estar en Proyecto.idProyecto

Supervisa.idPlanta debe estar en Planta.idPlanta

Empleado.númLegajo puede no estar en Supervisa.númLegajo

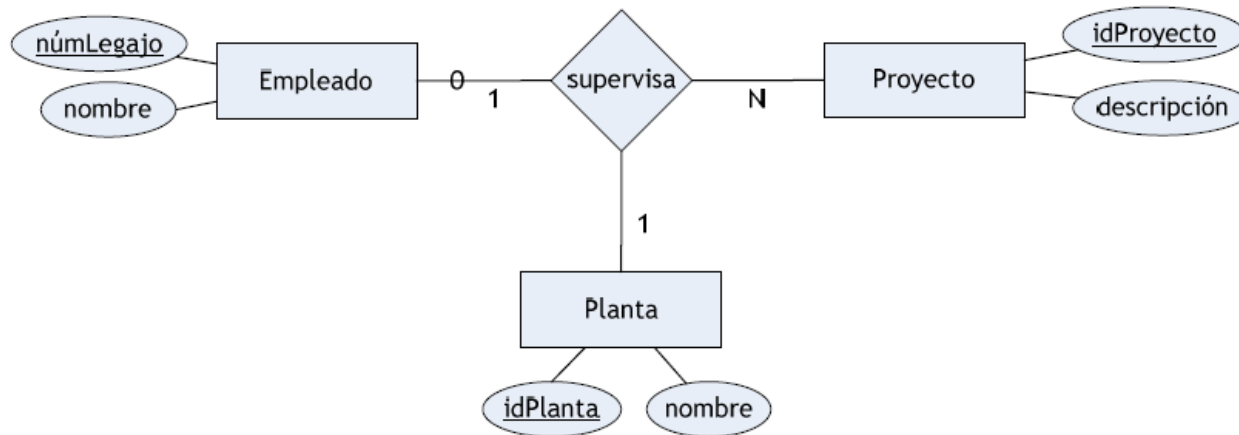
Proyecto.idProyecto debe estar en Supervisa.idProyecto

Planta.idPlanta debe estar en Supervisa.idPlanta

Transformación del MER al MR

► Ternarias 1:1:N

- Cambia el problema y ahora un empleado en una planta puede supervisar varios proyectos



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = {(númLegajo, idProyecto)}

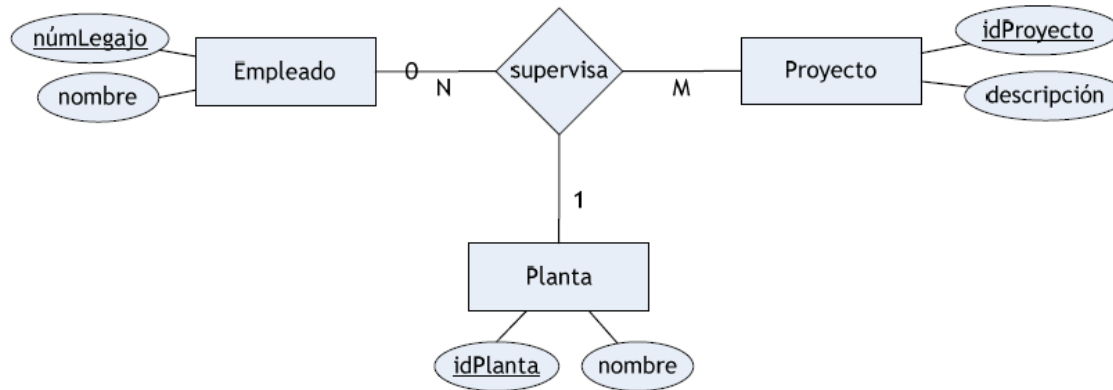
CK = {(númLegajo, idProyecto), (idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

Transformación del MER al MR

► Ternarias 1:N:M

- Cambia el problema y ahora un proyecto en una planta puede ser supervisado por varios empleados



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

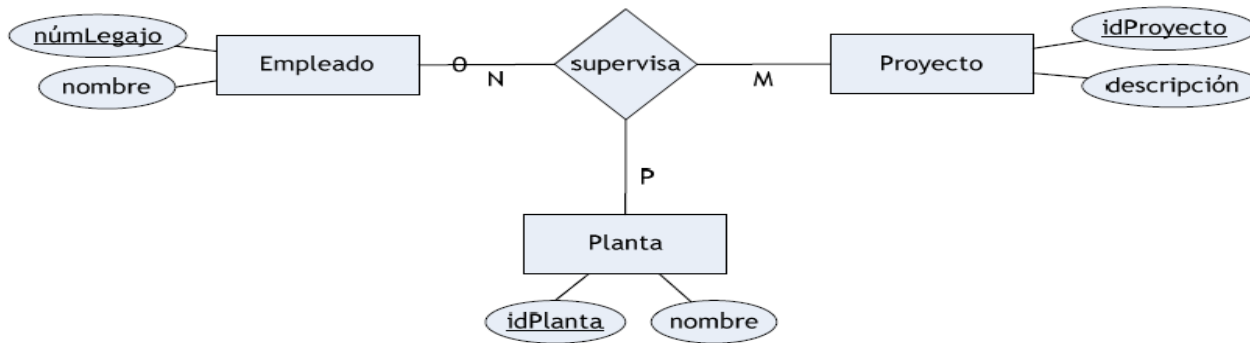
PK = CK = {(númLegajo, idProyecto)}

FK = {númLegajo, idProyecto, idPlanta}

Transformación del MER al MR

► Ternarias N:M:P

- Cambia el problema y ahora un empleado puede supervisar un proyecto en más de una planta.



Empleado(númLegajo, nombre)

PK = CK = {númLegajo}

Proyecto(idProyecto, descripción)

PK = CK = {idProyecto}

Planta(idPlanta, nombre)

PK = CK = {idPlanta}

Supervisa(númLegajo, idProyecto, idPlanta)

PK = CK = {(númLegajo, idProyecto, idPlanta)}

FK = {númLegajo, idProyecto, idPlanta}

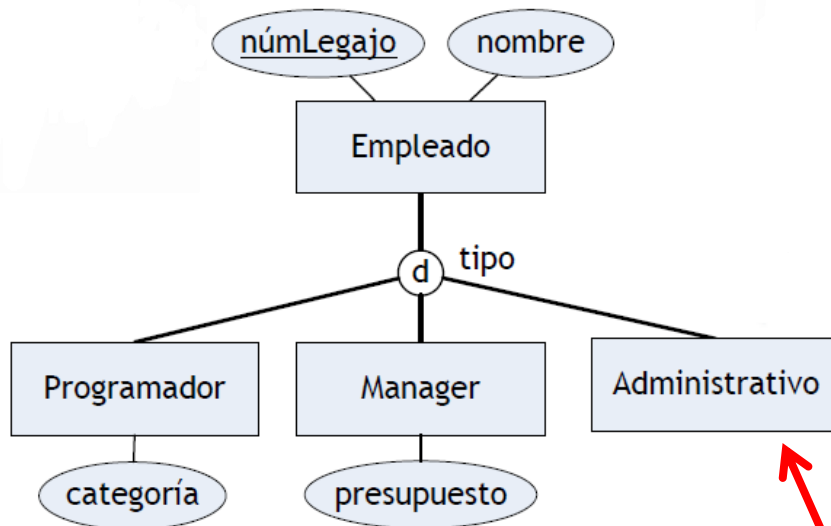
Transformación del MER al MR

▶ RESUMEN TERNARIAS

- ▶ Siempre se genera un esquema aparte para la interrelación.
- ▶ La clave del esquema dependerá de la cardinalidad.

Transformación del MER al MR

► Jerarquías- Disjunta



Se le agrega el discriminante como atributo, es el que permite particionar el conjunto de empleados

Empleado(númLegajo, nombre, tipo)
PK = CK = {númLegajo}

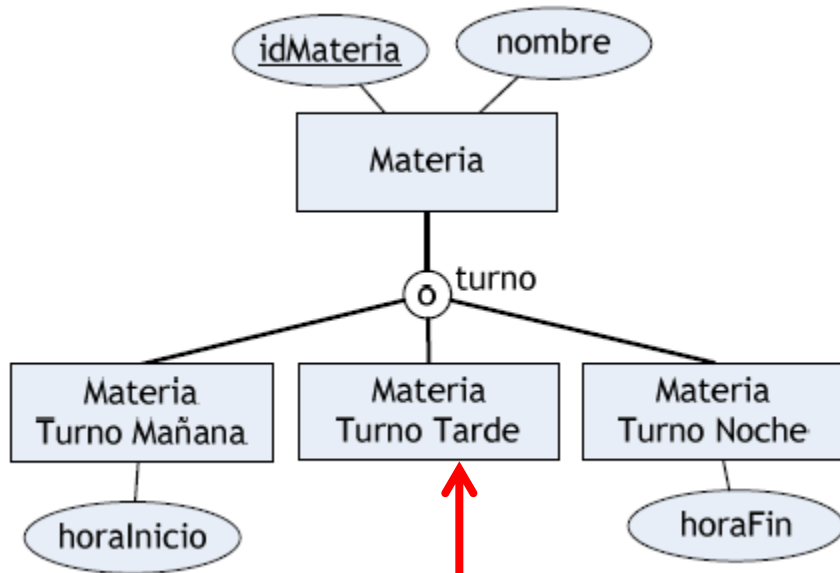
Programador(númLegajo, categoría)
PK = CK = FK = {númLegajo}

Manager(númLegajo, presupuesto)
PK = CK = FK = {númLegajo}

No tiene atributos ni relaciones no es necesario generar un esquema

Transformación del MER al MR

► Jerarquías con solapamiento



En este caso **si es necesario** generar un esquema

No se agrega el discriminante

Materia(idMateria, nombre)
PK = CK = {idMateria}

MateriaTurnoMañana(idMateria, horaInicio)
PK = CK = FK = {idMateria}

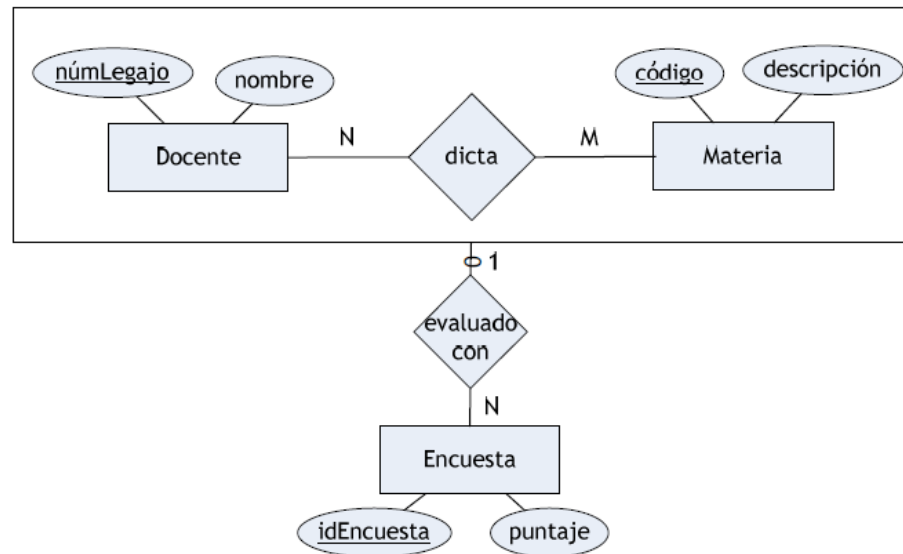
MateriaTurnoTarde(idMateria)
PK = CK = FK = {idMateria}

MateriaTurnoNoche(idMateria, horaFin)
PK = CK = FK = {idMateria}

Transformación del MER al MR

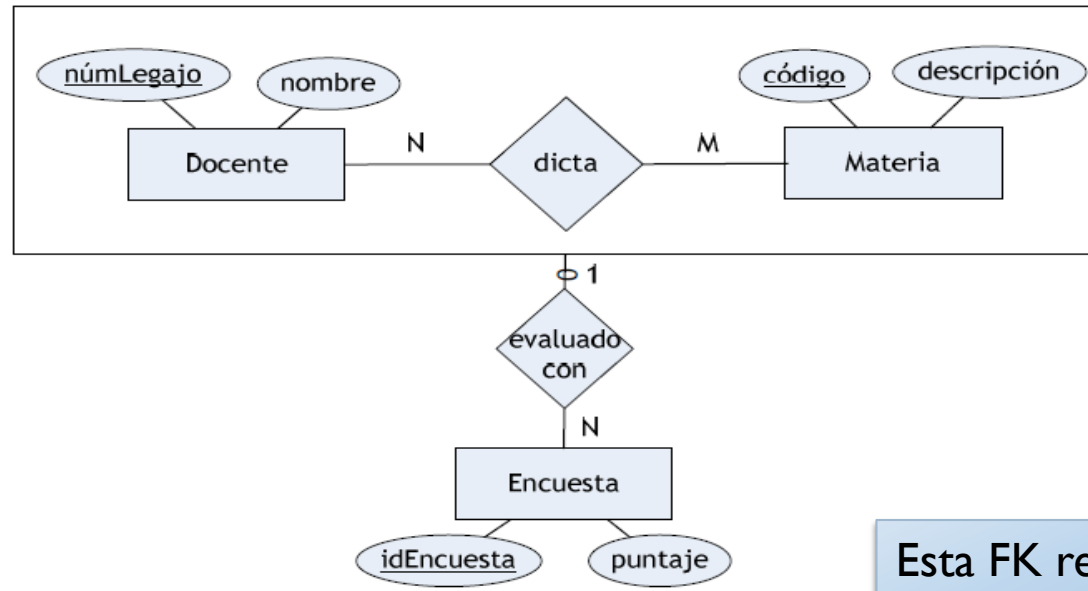
► Agregación

- En la notación de la materia sólo permitimos agregación en interrelaciones N:M.
- Las agregaciones se transforman considerando a la agregación como si fuera una entidad.



Transformación del MER al MR

► Agregación



Esta FK referencia a Dicta

Docente(númLegajo, nombre)
PK = CK = {númLegajo}

Materia(código, descripción)
PK = CK = {código}

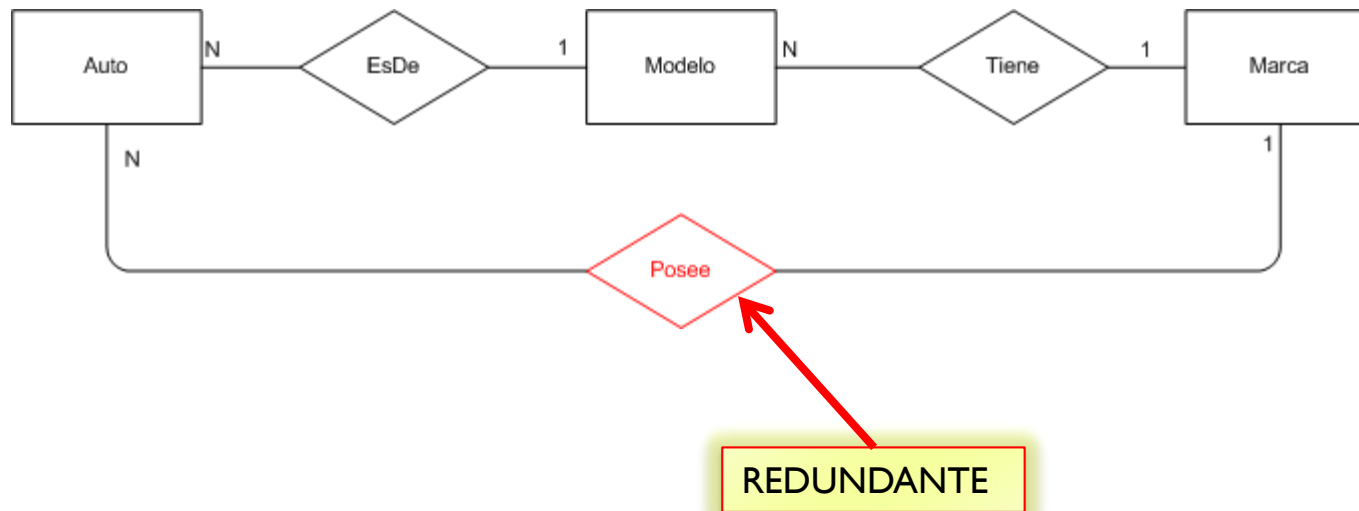
Dicta(númLegajo, código)
PK = CK = {(númLegajo, código)}
FK = {númLegajo, código}

Encuesta(idEncuesta, puntaje, númLegajo, código)
PK = CK = {idEncuesta}
FK = {(númLegajo, código)}

Consideraciones de Diseño

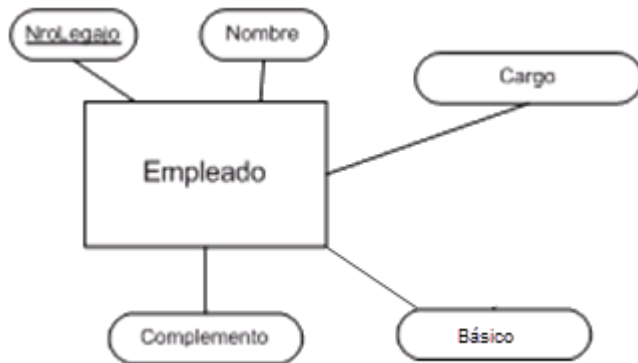
Consideraciones de Diseño

- ▶ El modelo debe respetar fielmente los requerimientos de la situación real que estamos modelando
- ▶ Evitar Redundancia



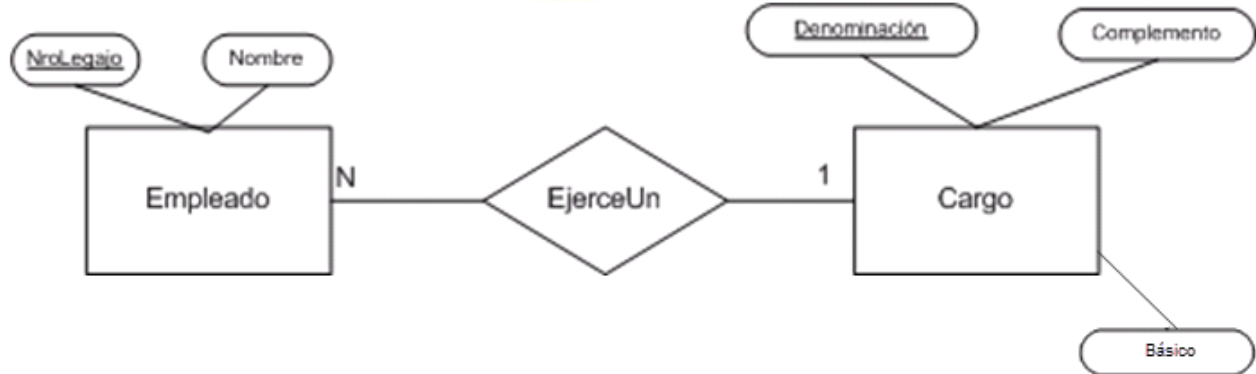
Consideraciones de Diseño

► ¿Entidad o Atributo?



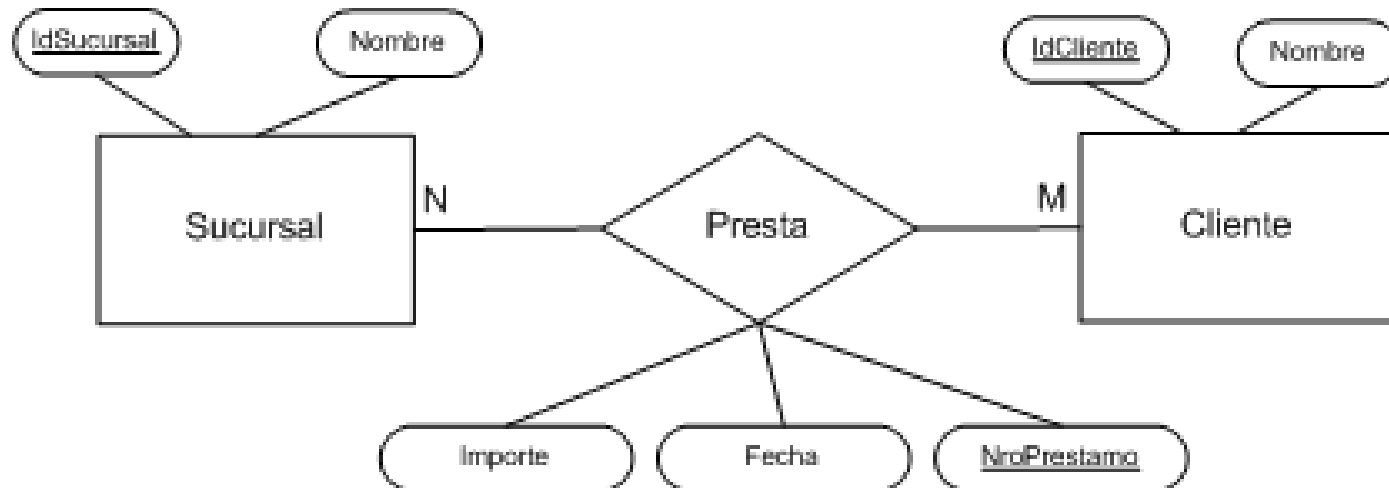
¿Más de un atributo?

¿Posibilidad de otras relaciones?



Consideraciones de Diseño

► ¿Entidad o Interrelación?

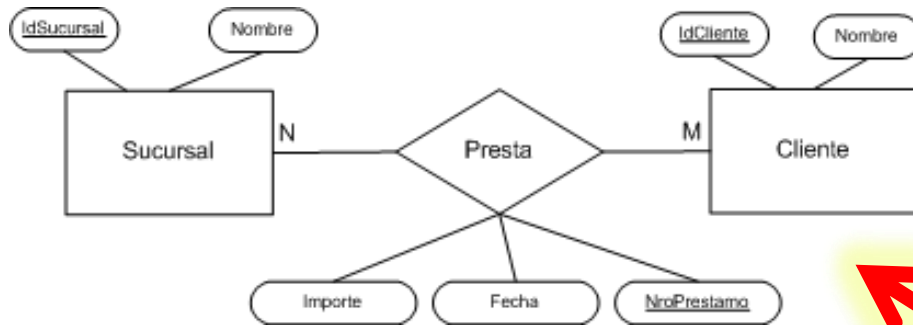


¿Cómo ubicamos los pagos?

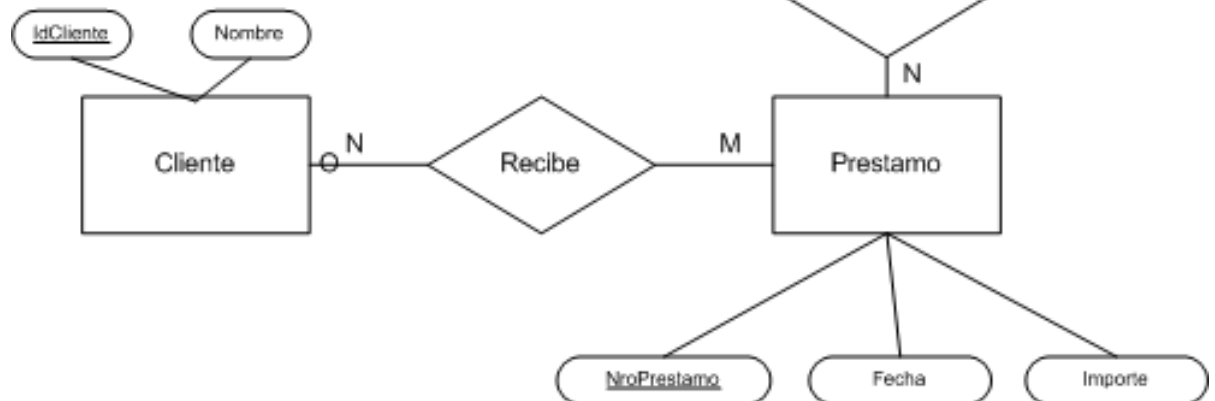
¿Qué pasa si varios clientes comparten un préstamo?

Consideraciones de Diseño

► ¿Entidad o Interrelación?

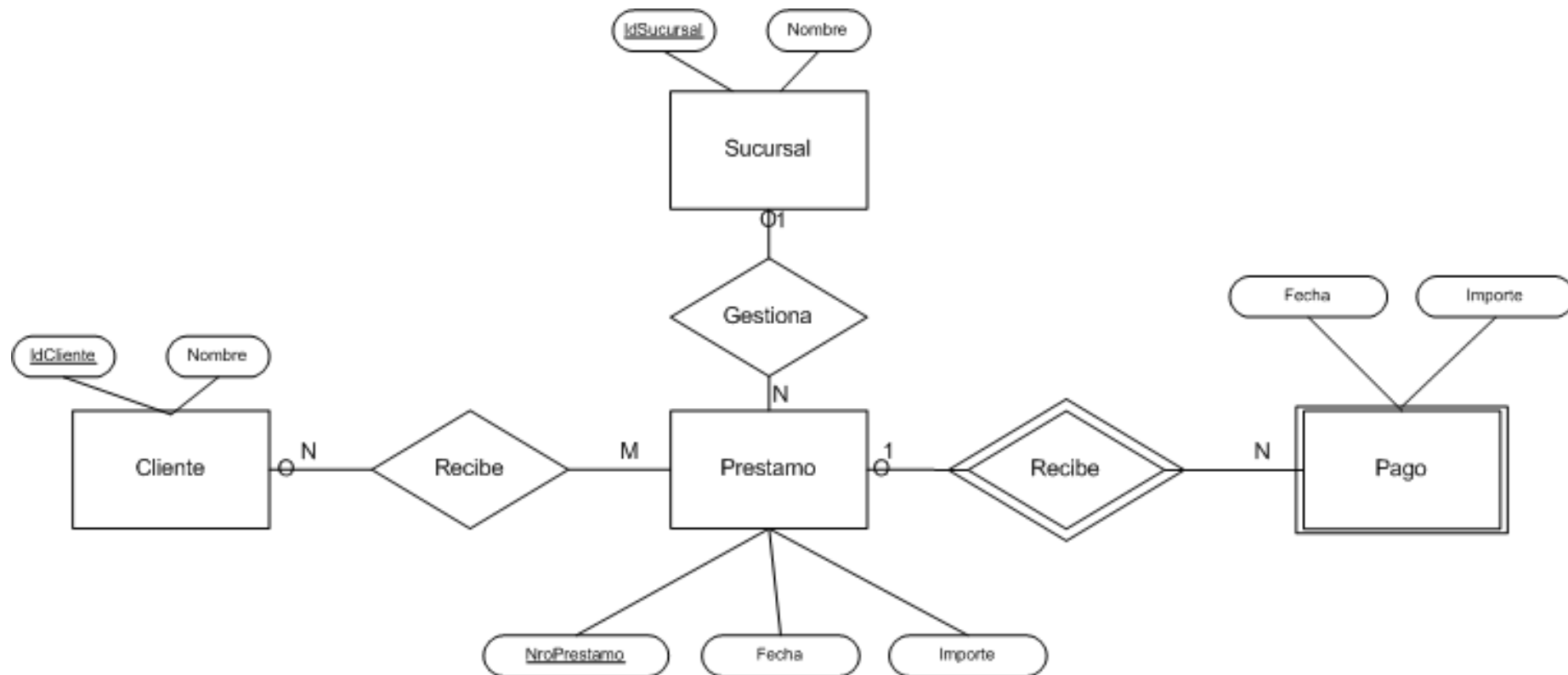


¿Qué pasa si varios clientes comparten un préstamo?



¿Cómo ubicamos los pagos?

Consideraciones de Diseño



¿Cómo ubicamos los pagos?

Consideraciones de Diseño

- ▶ Sustantivos en singular para las entidades
- ▶ Por lo general, usar verbos para las interrelaciones. Se leen de
 - ▶ Izquierda a derecha
 - ▶ Arriba hacia abajo
- ▶ Elegir buenas Claves
 - ▶ Para elegir una buena clave, debe seleccionarse un atributo que no vaya a cambiar con el tiempo.
 - ▶ Cuando la clave es muy compleja o no existe un atributo que surja del problema real, se recomienda generar una clave artificial (por ejemplo, idProducto)

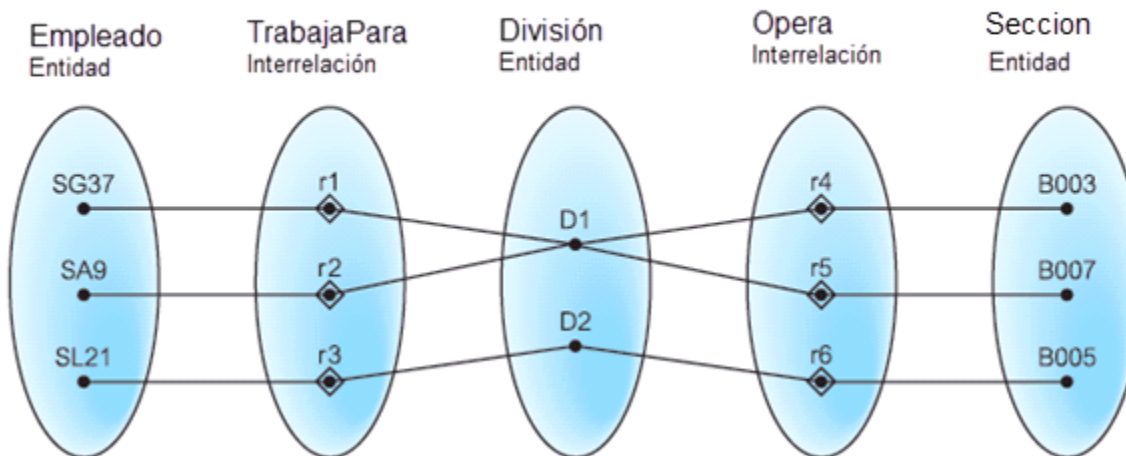
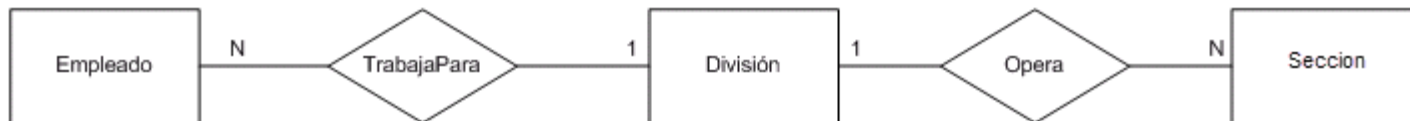
Trampas de Conexión

- ▶ Ocurren debido a una mala interpretación de las interrelaciones.
- ▶ Para identificar las trampas de conexión, debemos asegurarnos de que el **significado** de un tipo de interrelación está completamente **entendido** y claramente **definido**.
- ▶ Si no entendemos las interrelaciones, podríamos crear un modelo que no es una representación adecuada del “mundo real”.
- ▶ ***Fan Traps* o Trampa del Abanico.**
- ▶ ***Chasm Traps* o Trampa del Sumidero.**

Trampas de Conexión

► Trampa del Abanico

- El modelo representa una interrelación entre tipos de entidades, pero el camino entre algunas entidades es ambiguo
- La trampa de abanico sucede cuando salen dos o más interrelaciones 1:N en abanico desde la misma entidad.



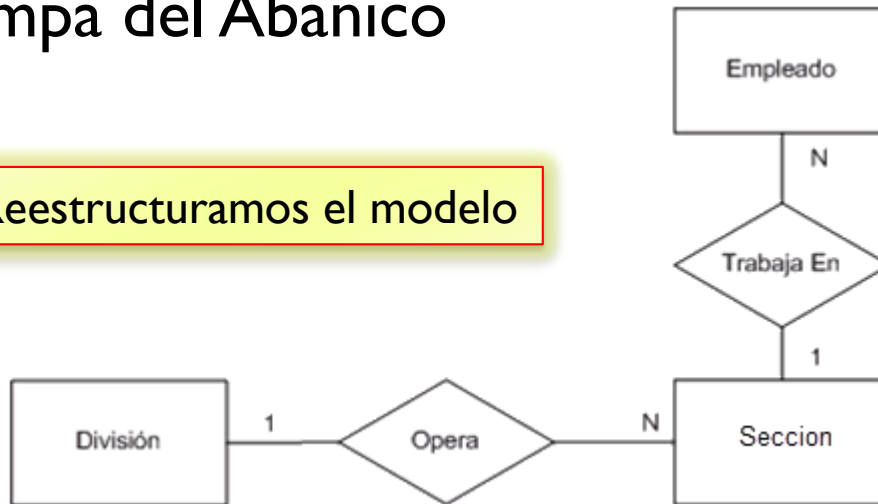
¿Qué empleados trabajan en una sección?

¿A que sección pertenece el empleado SG37?

Trampas de Conexión

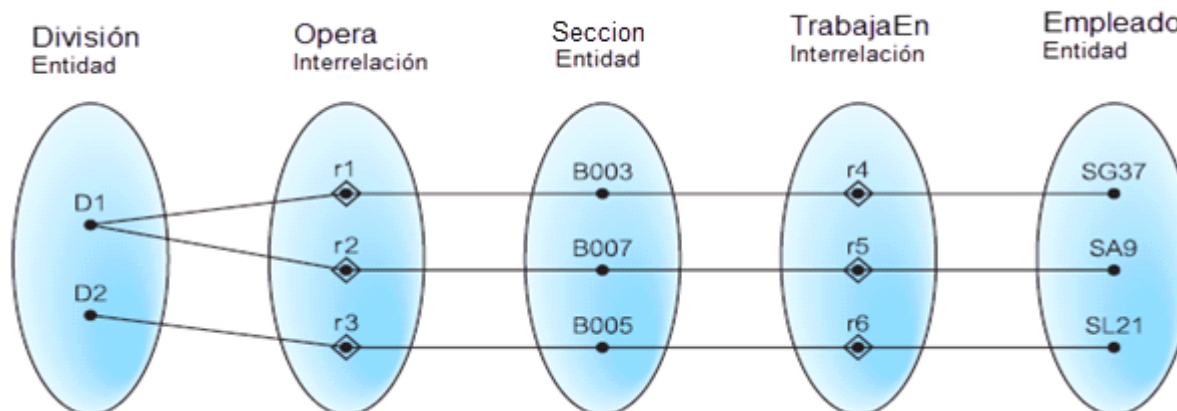
► Trampa del Abanico

Reestructuramos el modelo



¿Qué empleados trabajan en una sección?

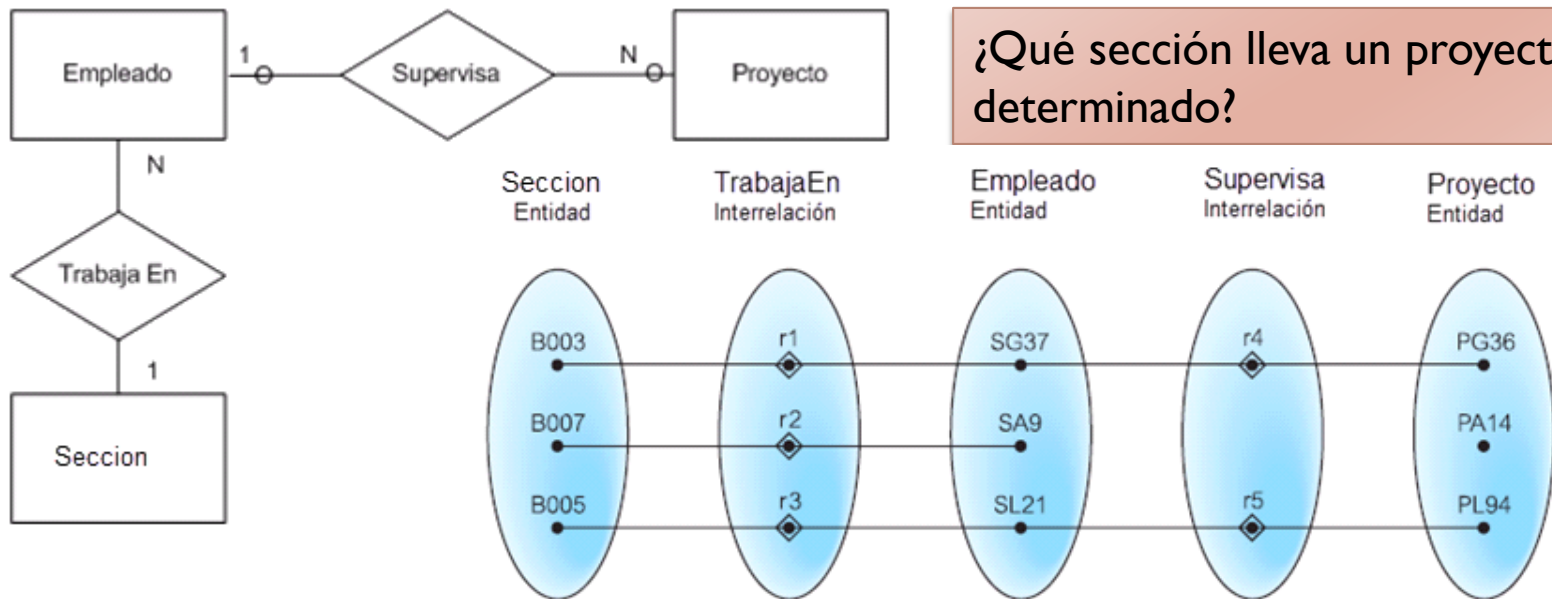
¿Qué empleados trabajan en una división?



Trampas de Conexión

► Trampa del Sumidero

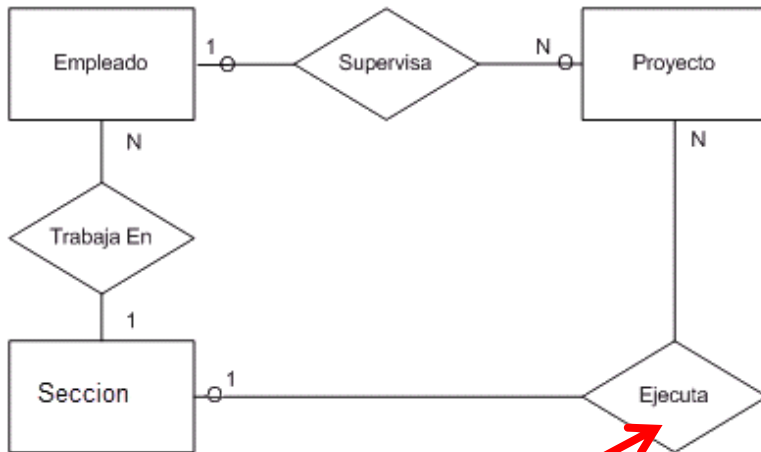
- El modelo sugiere la existencia de algún tipo de interrelación entre dos tipos de entidad, pero no existe camino entre algunas entidades.
- La trampa del sumidero puede aparecer cuando hay uno o más tipos de interrelación donde los tipos de entidad tienen una participación parcial.



En una seccion trabajan uno o más empleados, los cuales pueden supervisar cero o más proyectos llevados por la seccion, que a su vez son supervisados por cero o más empleados.

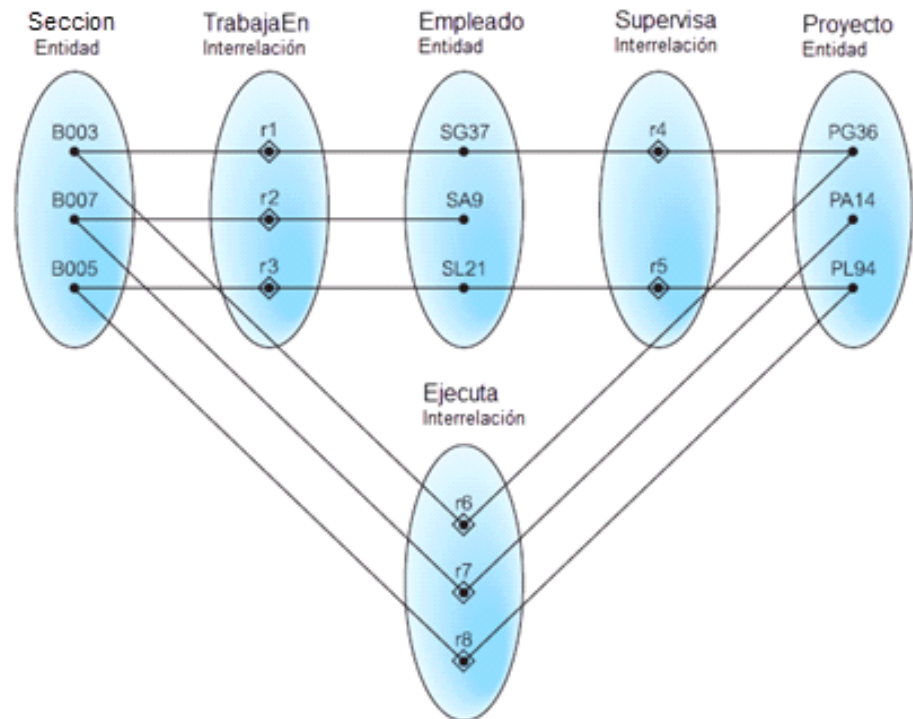
Trampas de Conexión

► Trampa del Sumidero



Reestructuramos el modelo y Agregamos una Interrelación

¿Qué sección lleva un proyecto determinado?



Bibliografía

- ▶ Database System Concepts. 4ta Edición Abraham Silberschatz, Henry F. Korth y S. Sudarshan
- ▶ Database Management Systems, Ramakrishnan/Gherke 3rd Ed.
- ▶ A Practical Approach to Design, Implementation, and Management. Thomas Connolly/Carolyn Begg 4ta Ed.
- ▶ Apunte de la Materia
- ▶ Fundamentals of Database Systems Elmasri/Navathe 7th Ed., Addison Wesley