

UD 3 :

Diseño Entidad/Relacional



IES Maestro de Cva.
Isabel Guerrero

Contenidos

- ❖ **Representación del problema**
- ❖ Modelo de datos
- ❖ Diagramas E/R
- ❖ El modelo E/R ampliado
- ❖ El modelo relacional
- ❖ Transformación E/R al modelo relacional
- ❖ Normalización

Representación del problema

- ❖ Una **base de datos** representa la información contenida en algún dominio del mundo real.
- ❖ El **diseño de base** de datos consiste en **extraer** todos los **datos** relevantes de un **problema**.
- ❖ Extraer datos → Análisis → Modelización (Esquema)
- ❖ 1º Reuniones: Informáticos y Usuarios → Especificación de Requisitos del Software (ERS)

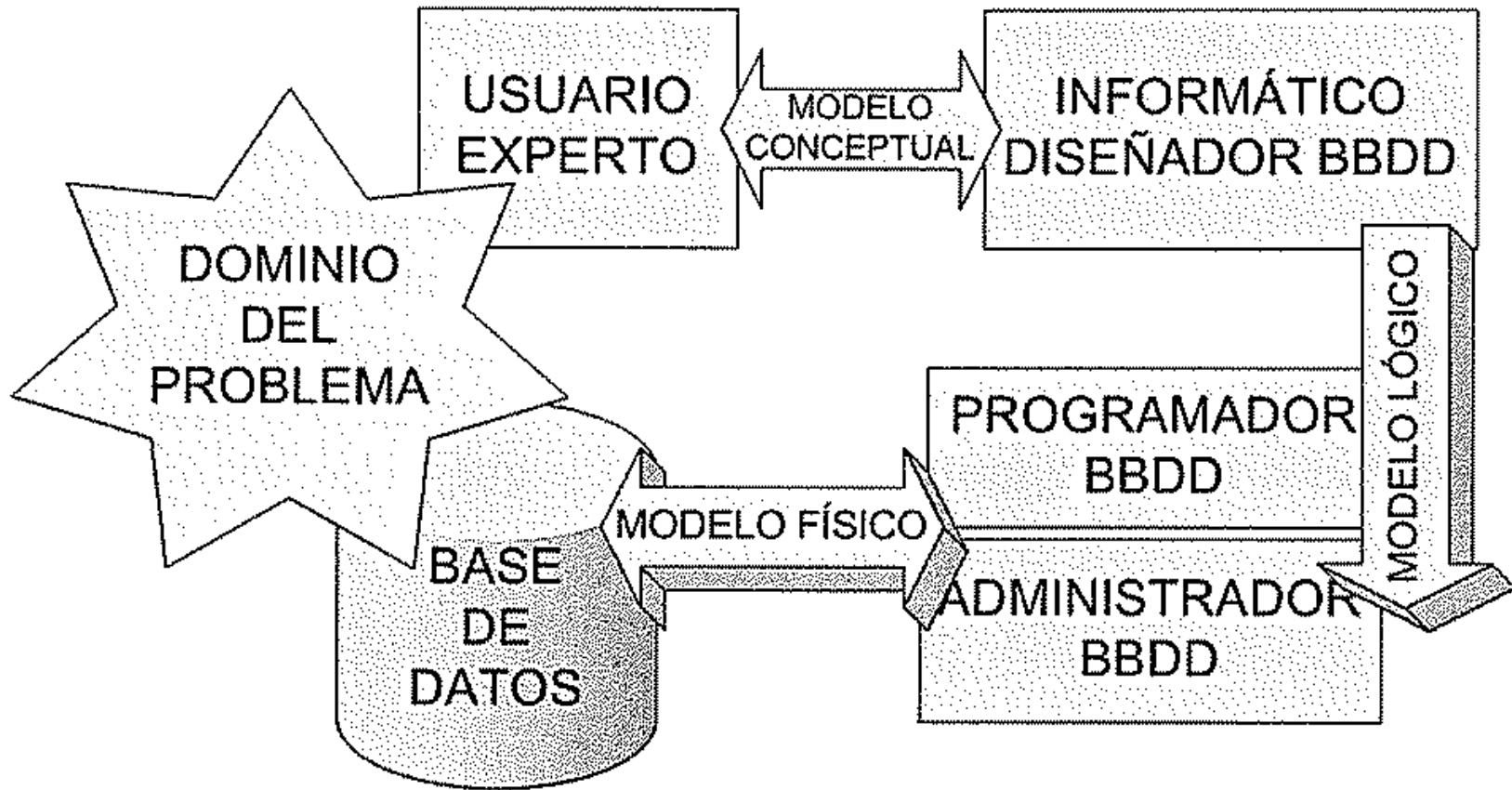
Contenidos

- ❖ Representación del problema
- ❖ **Modelo de datos**
- ❖ Diagramas E/R
- ❖ El modelo E/R ampliado
- ❖ El modelo relacional
- ❖ Transformación E/R al modelo relacional
- ❖ Normalización

Modelo de datos

- ❖ La **modelización** consiste en **representar** el problema y generar un mapa donde estén identificados **todos los objetos** de la base de datos.
- ❖ La persona que realiza la **modelización** es un **analista informático**, se ha de contar con la experiencia de un futuro **usuario** de la BD.
- ❖ Hay que **modelar** siguiendo unas directrices o **estándares**.
- ❖ La base de datos estará implementada en un **SGBD** concreto, hay que tenerlo en cuenta.

Modelo de Datos:



Modelo de datos: tipos

- ❖ **El modelo conceptual.** Es un modelo que tiene un gran poder expresivo para poder comunicarse con un usuario que no es experto en informática.

Modelo E/R. (Informático ↔ Usuario)

- ❖ **El modelo lógico.** Este modelo es más técnico que el anterior. **Modelo Relacional.**
- ❖ **Modelo físico.** Es el resultado de aplicar el modelo lógico a un SGBD concreto. Está expresado en un lenguaje tipo SQL (Sublenguaje DDL : Lenguaje de Definición de Datos)

Contenidos

- ❖ Representación del problema
- ❖ Modelo de datos
- ❖ **Diagramas E/R**
- ❖ El modelo E/R ampliado
- ❖ El modelo relacional
- ❖ Transformación E/R al modelo relacional
- ❖ Normalización

Diagrama E/R

- ❖ Representa el **modelo conceptual** se usará el modelo Entidad/Relación.
- ❖ Este modelo consiste en plasmar el resultado del análisis del problema mediante diagramas entidad-relación.
- ❖ Representan las definiciones necesarias para comprender el *modelo entidad/relación*.

Entidad

- ❖ Representa cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso.
- ❖ Por ejemplo: coche, casa, empleado, cliente, etc.
- ❖ Las **entidades** se representan gráficamente mediante **rectángulos** y su nombre aparece en el interior (generalmente en singular).
- ❖ Un nombre de entidad solo puede aparecer una vez en el diagrama.

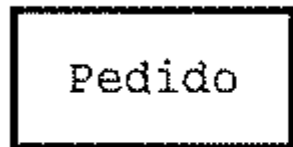
Cliente

Empleado

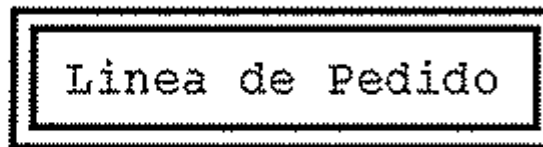
Producto

Entidad: Fuertes y Débiles

- ❖ Hay dos *tipos de entidades*:
 - **Fuertes**
 - **Débiles.**
- ❖ Una **entidad fuerte** es una entidad existe por méritos propios.
- ❖ Una **entidad débil** es una entidad cuya existencia depende de la existencia de otra entidad.
- ❖ *Las entidades débiles* se representan mediante un cuadro doble.



Entidad Fuerte



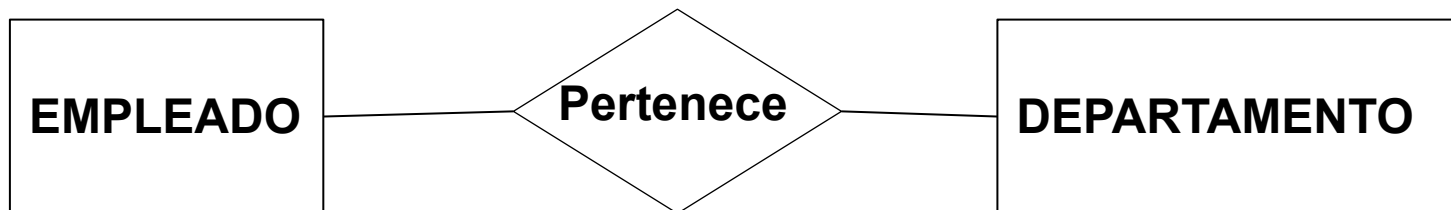
Entidad Débil

Ocurrencia de una entidad

- ❖ Es una instancia de una determinada entidad, esto es, una unidad del conjunto que representa la entidad.
- ❖ Ejemplo: La entidad "coche" tiene varias instancias, una de ellas es el vehiculo "seat ibiza con matrícula 1222FHD de color negro y con 5 puertas".

Relación

- ❖ Una **relación** (o interrelación), es una correspondencia o asociación entre dos o más entidades.
- ❖ Cada **relación** tiene un **nombre** que describe su función.
- ❖ Las relaciones se representan gráficamente mediante **rombos** y su nombre aparece en el interior (corresponde a un verbo).



Relación

- ❖ Las relaciones están clasificadas según su ***grado***.
- ❖ *El grado es el número de entidades que participan en la relación.*
 - Relaciones binarias
 - Relaciones ternarias
 - Relaciones unarias
 - Relaciones n-arias

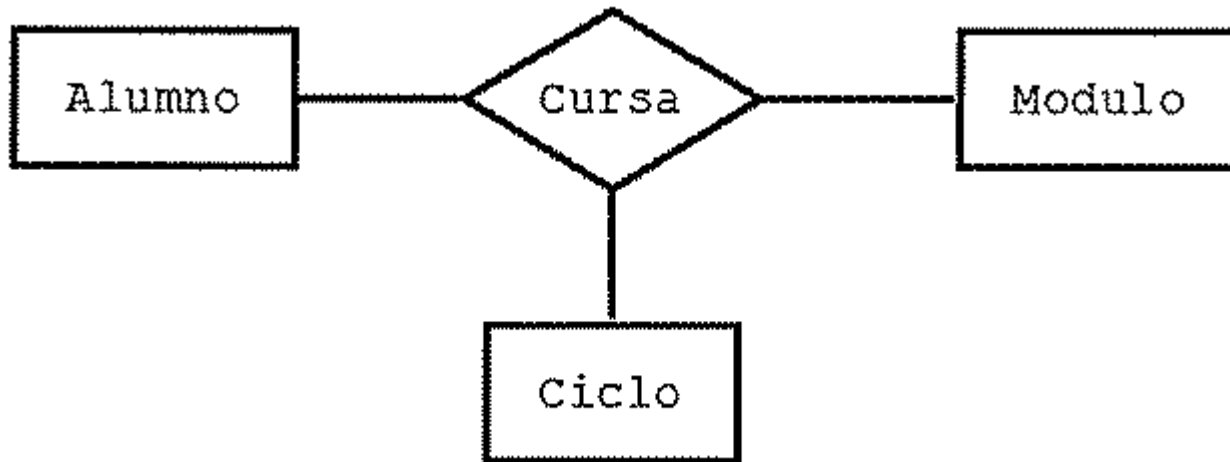
Relaciones Binarias

- ❖ **Relaciones binarias:** (grado 2), son aquellas que se dan entre dos entidades.



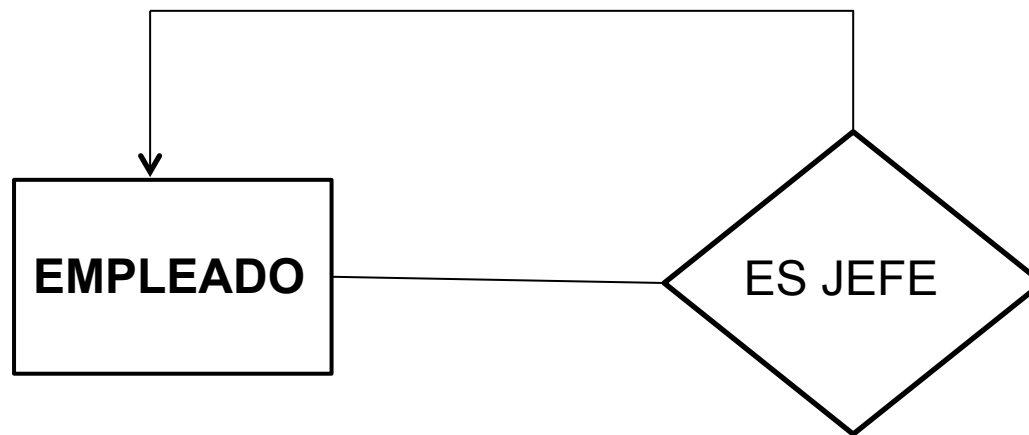
Relaciones ternarias

- ❖ **Relaciones ternarias:** (grado 3), son aquellas que se dan entre tres entidades.



Relaciones Unarias

- ❖ **Relaciones unarias o reflexivas:** (grado 1), Es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. El nombre de estos papeles es importante para determinar la función de cada participación.



Relaciones n-arias

- ❖ **Relaciones n-arias:** (grado >3) Son aquellas donde participan más de 3 entidades.
- ❖ Aparecen en muy raras ocasiones, puesto que generalmente se pueden descomponer en varias de grado 2 o de grado 3.

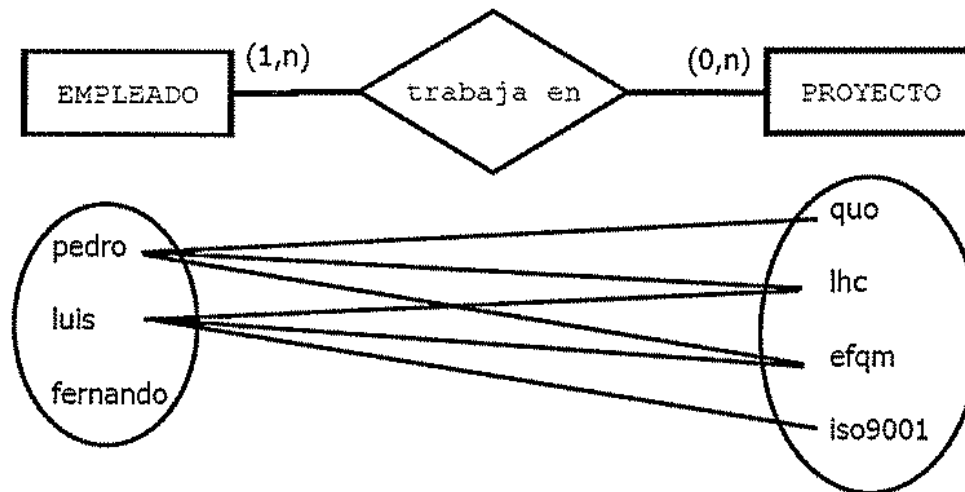
Cardinalidad de una entidad

- ❖ La **cardinalidad de una entidad** u ocurrencia de una entidad, indica, mediante una pareja de números, el mínimo y máximo número de veces que puede aparecer en la relación asociada a otra ocurrencia de entidad.

Participación	Significado
(0,1)	Mínimo cero, máximo uno
(1,1)	Mínimo uno, máximo uno
(0,n)	Mínimo cero, máximo n (Muchos)
(1,n)	Mínimo uno, máximo n (Muchos)

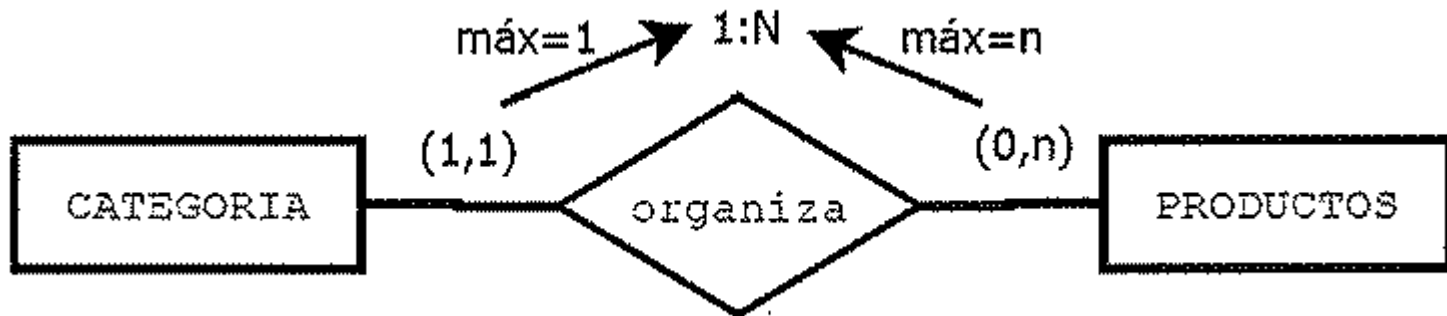
Cardinalidad de una entidad

- ❖ Por ejemplo, los empleados pueden trabajar para varios proyectos, o pueden estar de vacaciones (sin proyecto).
- ❖ Por otro lado, en un proyecto trabajan de 1 a varios trabajadores. En este caso, la participación de proyecto es de $(0,n)$, puesto que un empleado puede tener asignados de 0 a n proyectos.
- ❖ La participación del empleado es de $(1,n)$ puesto que en un proyecto puede haber de 1 a n empleados. De esta manera, se indica al lado de la entidad proyecto, el par $(0,n)$ y al lado de la entidad empleado el par $(1,n)$.



Cardinalidad de una relación

- ❖ La **cardinalidad de una relación** se calcula a través de las cardinalidades de las entidades que participan en ella.
- ❖ Se toman el número **máximo** de la cardinalidad de cada una de las entidades de la relación.
- ❖ En el ejemplo siguiente, tendría una cardinalidad de 1: N, puesto que por el lado de las categorías, el máximo de (1,1) es 1, y por el lado de los productos, el máximo de (0,n) es N.

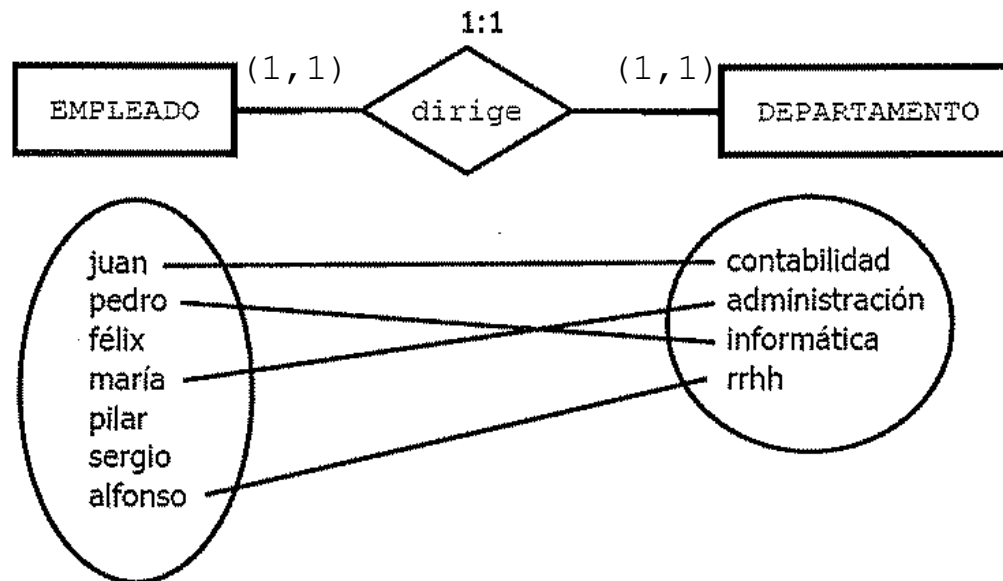


Tipos de cardinalidad de la relación

- ❖ **Cardinalidad 1: 1**
- ❖ **Cardinalidad 1:N (o 1:Muchos)**
- ❖ **Cardinalidad M:N (o Muchos:Muchos)**

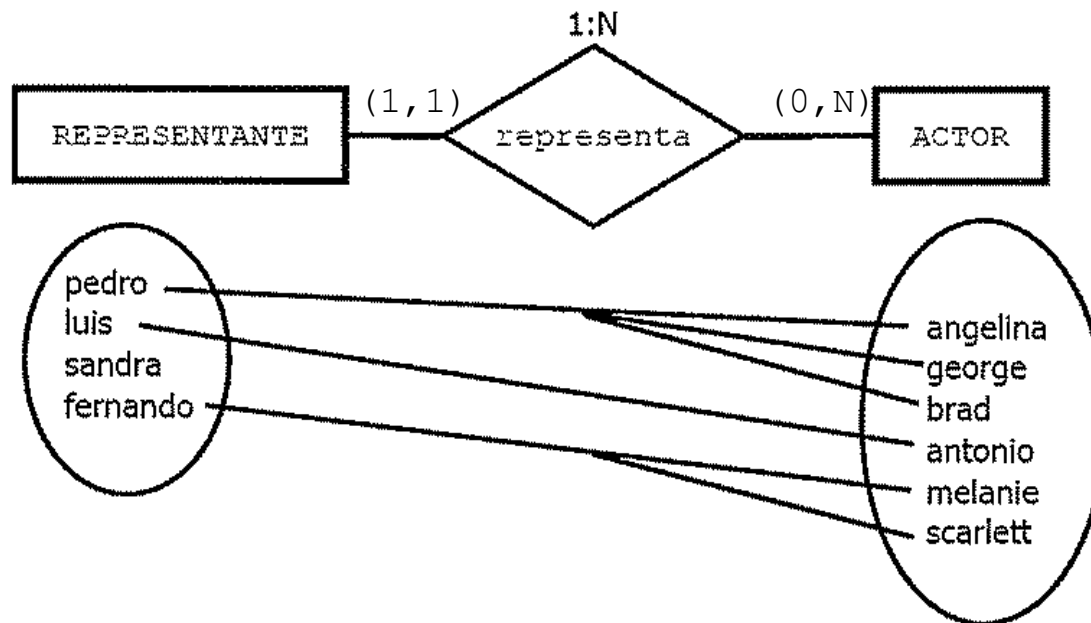
Cardinalidad 1:1

- ❖ Esta cardinalidad especifica que una entidad A puede estar vinculada mediante una relación a una y solo una ocurrencia de otra entidad B. A su vez una ocurrencia de la entidad B solo puede estar vinculada a una ocurrencia de la entidad A.
- ❖ Por ejemplo, un empleado solo puede ser jefe de un departamento, y un departamento solo puede tener un jefe. Su cardinalidad es 1:1



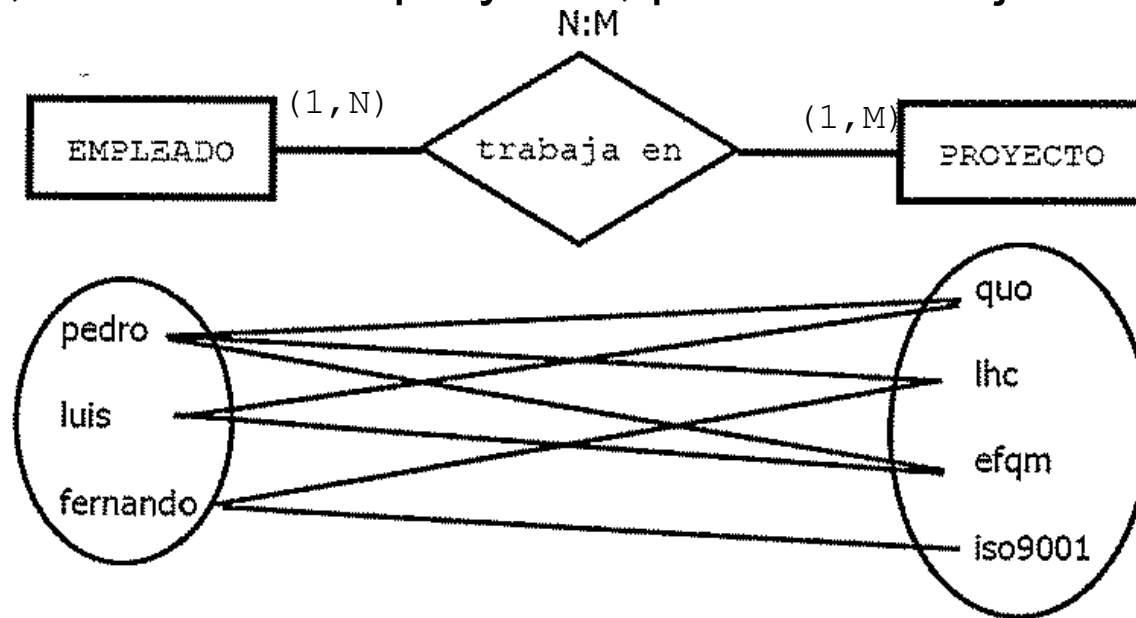
Cardinalidad 1:N (o 1:Muchos)

- ❖ Esta relación especifica que una entidad A puede estar vinculada mediante una relación a varias ocurrencias de otra entidad B. Sin embargo, una de las ocurrencias de la entidad B solo puede estar vinculada a una ocurrencia de la entidad A.
- ❖ Por ejemplo, un representante gestiona las carreras de varios actores, y un actor solo puede tener un representante .

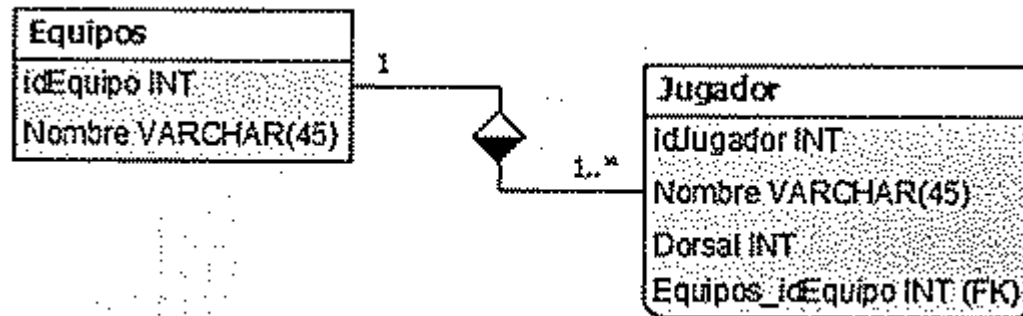


Cardinalidad M:N (o Muchos : Muchos)

- ❖ Esta cardinalidad especifica que una entidad A puede estar vinculada mediante una relación a varias ocurrencias de la entidad B, y a su vez, una ocurrencia de la entidad B puede estar vinculada a varias de la entidad A.
- ❖ Por ejemplo, un empleado puede trabajar para varios proyectos; al mismo tiempo, en un mismo proyecto, pueden trabajar varios empleados.



Otras notaciones

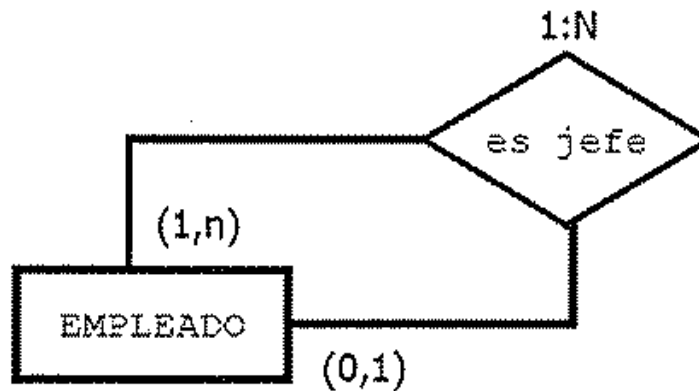


Cardinalidad de relaciones no binarias

- ❖ Para calcular la cardinalidad de una relación ternaria se tomará una de las tres entidades y se combinan las otras dos.
- ❖ A continuación, se calcula la participación de la entidad en la combinación de las otras dos.
- ❖ Posteriormente, se hará lo mismo con las otras dos entidades.
- ❖ Finalmente, tomando los máximos de las participaciones se generan las cardinalidades.

Cardinalidad de las relaciones reflexivas

- ❖ En las relaciones reflexivas, la misma entidad juega dos papeles distintos en la relación. Para calcular su cardinalidad hay que extraer las participaciones según los dos roles existentes.
- ❖ ¿Cuántos subordinados puede tener un jefe? Un jefe puede tener un mínimo de 1 y un máximo de n : $(1,n)$
- ❖ ¿Cuántos jefes puede tener un subordinado? Un mínimo de 0 (un empleado sin jefes sería el responsable de la empresa) y un máximo de 1 (suponiendo una estructura, típicamente piramidal): $(0,1)$.
- ❖ La cardinalidad sería $1 \cdot N$

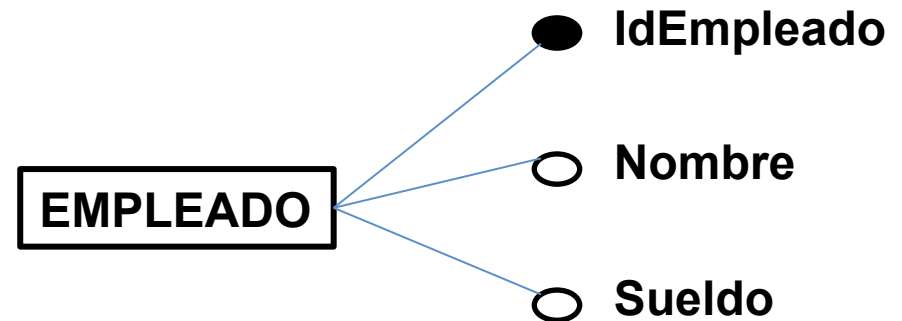
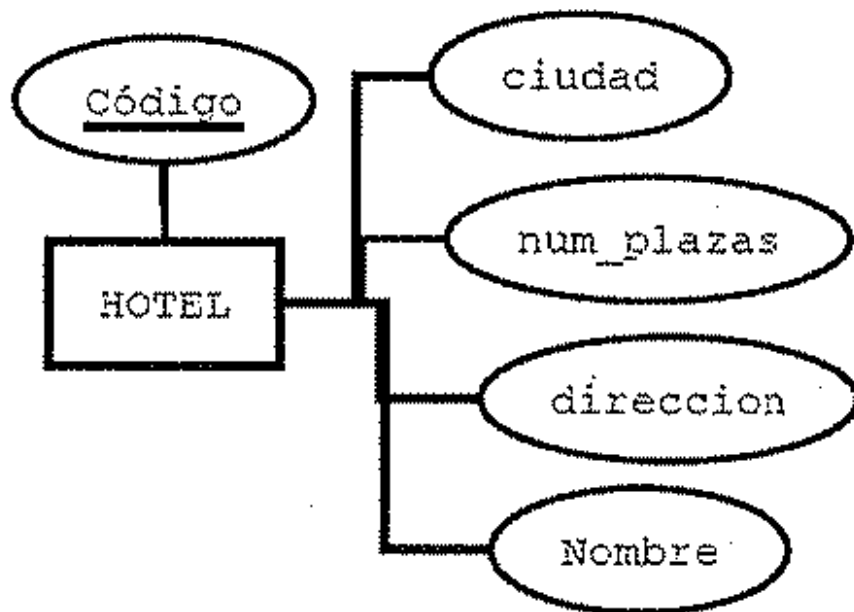


Atributos y Dominios

- ❖ **Atributos** de una entidad: son las características o propiedades que tiene dicha entidad.
- ❖ Se representan mediante elipses conectadas directamente a la entidad o con un círculo.

Atributos y Dominios

- ❖ Por ejemplo, para representar la entidad **HOTEL**, son necesarias sus características, esto es, el número de plazas disponibles, su dirección, la ciudad donde se encuentra, etc. Y para un



Atributo Clave

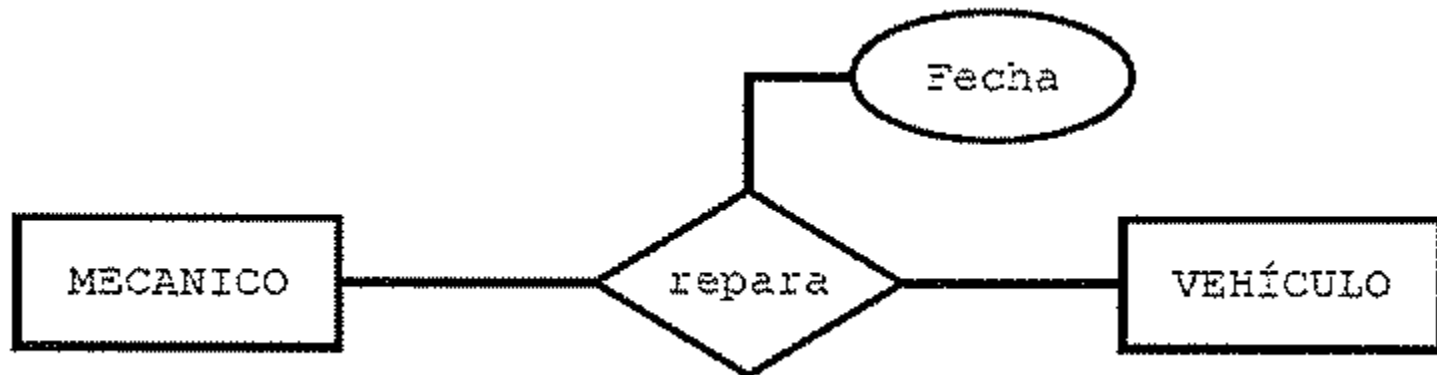
- ❖ **Clave** : designa un atributo que cuyo contenido no se puede repetir en la entidad. (**subrayado o con un círculo negro**)
- ❖ Identifica unívocamente a una entidad, es decir, que con la sola referencia a un campo clave se tiene acceso al resto de atributos de forma directa.
- ❖ Ejemplo: El DNI es el campo clave de una persona, pues cada persona tiene un DNI que es único.

Atributo clave

- ❖ Todas las entidades fuertes deben tener un atributo clave.
- ❖ **Clave atómica:** cuando la clave está formada por un único atributo.
- ❖ Ejemplo: DEPARTAMENTO: IdDepartamento, EMPLEADO: DNI, etc.
- ❖ **Clave compuesta:** la clave estará formada por varios atributos, en este caso, se dice que la clave de la entidad es la suma de esos atributos
- ❖ Por ejemplo, para identificar una VENTA, tendremos que saber el cliente, el artículo vendido y la fecha en la que se ha vendido.

Atributo de relación

- ❖ Un **atributo de relación** es aquel que es propio de una relación y que no puede ser cedido a las entidades que intervienen en la relación.
- ❖ Por ejemplo, un mecánico repara un vehículo, la reparación se realiza en una determinada fecha.



Dominios

- ❖ Cada una de las características que tiene una entidad pertenece a un **dominio**.
- ❖ El **dominio** representa la naturaleza del dato
- ❖ **Tipos de dominios:**
 - Número entero
 - Cadena de caracteres
 - Número real.
 - Booleano (Verdadero/Falso)
 - Más complejas: fecha o hora (con minutos y segundos).
 - ...

Dominios

- ❖ Por ejemplo, los siguientes atributos de la entidad empleado pertenecen a los siguientes dominios:

Atributo	Dominio
DNI	Cadena de Caracteres de longitud 10
Nombre	Cadena de Caracteres de longitud 50
Fecha_Nacimiento	Fecha
Dirección	Cadena de Caracteres de longitud 100
Sueldo	Números reales
Número de hijos	Números enteros
Departamento	Departamentos

Tipos de Atributos

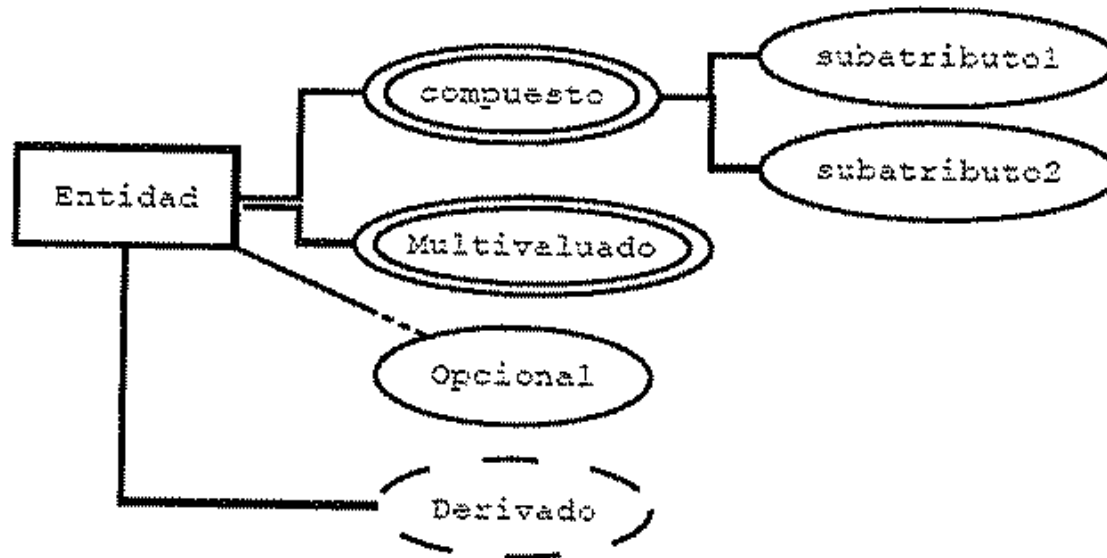
- ❖ **Atributos obligatorios:** Un atributo debe tomar un valor obligatoriamente.
- ❖ **Atributos opcionales:** Un atributo puede no tomar un valor porque sea desconocido en un momento determinado. En este caso, el atributo tiene un valor *nulo*.
- ❖ **Atributos compuestos:** Un atributo compuesto es aquel que se puede descomponer en atributos más sencillos, por ejemplo, el atributo hora_de_salida se puede descomponer en dos (hora y minutos).

Tipos de Atributos

- ❖ **Atributos univaluados:** Un atributo que toma un único valor.
- ❖ **Atributos multivaluados:** Estos atributos pueden tomar varios valores, por ejemplo el atributo teléfono puede tomar los valores de un teléfono móvil y un teléfono fijo.
- ❖ **Atributo derivado:** Son aquellos cuyo valor se puede calcular a través de otros atributos. Por ejemplo, el atributo Edad, se puede calcular a partir de la fecha de nacimiento de una persona.

Tipos de Atributos

- ❖ Al igual que con la mayoría de las notaciones, no existe unanimidad a la hora de dibujar en un diagrama los tipos de atributos. Una de las más extendidas entre los diseñadores de bases de datos es la siguiente:



Las entidades débiles

- ❖ Las entidades **débiles dependen** de una entidad **fuerte** mediante una **relación**.
- ❖ La **relación** que **une** ambas entidades también es **débil**, puesto que también desaparece si desaparece la entidad fuerte.
- ❖ La relación tiene una dependencia que puede ser de dos tipos:
 - **Dependencia de existencia**
 - **Dependencia de identificación.**

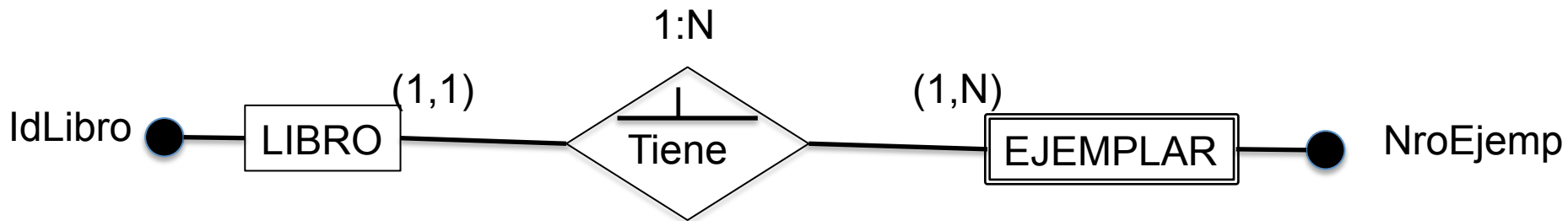
Entidades Débiles: Dependencia en Existencia

- ❖ **Dependencia de existencia:** Este tipo de dependencia expresa que, las ocurrencias de una entidad débil, no tienen ningún sentido en la base de datos sin la presencia de las ocurrencias de la entidad fuerte con la que están relacionadas.
- ❖ Por ejemplo, las transacciones que se dan en una cuenta bancaria, no tienen sentido si no existe la cuenta bancaria a la que están asociadas.



Entidades Débiles: Dependencia en Identificación

- ❖ **Dependencia de identificación:** Este tipo se produce cuando, además de la dependencia de existencia, la entidad débil necesita a la fuerte para poder identificarse.



Contenidos

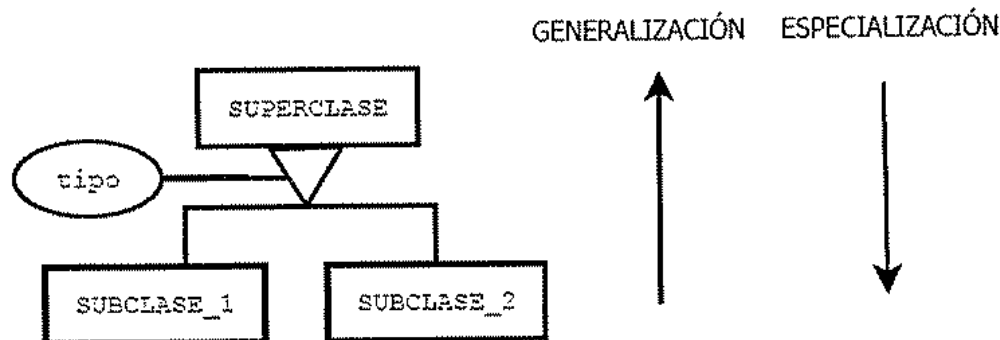
- ❖ Representación del problema
- ❖ Modelo de datos
- ❖ Diagramas E/R
- ❖ **El modelo E/R ampliado**
- ❖ El modelo relacional
- ❖ Transformación E/R al modelo relacional
- ❖ Normalización

El modelo E/R ampliado

- ❖ El modelo **Entidad-Relación Extendido**, o **Ampliado**, incorpora todos los elementos del modelo entidad relación incluyendo los conceptos:
 - Subclase
 - Superclase
 - Especialización
 - Generalización .

Generalización y Especialización

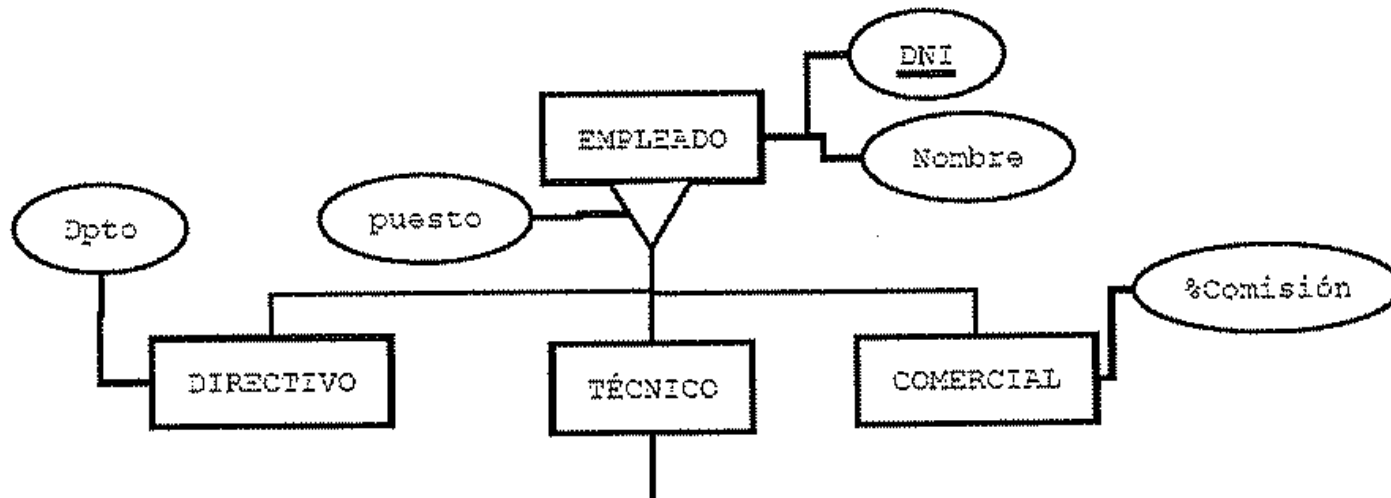
- ❖ Todas las propiedades y atributos de la entidad genérica **superclase** son heredadas por las **subclases**.
- ❖ Además, cada subclase tendrá sus propios atributos, independientes de la generalización.
- ❖ La relación de generalización se representa mediante un triángulo Isósceles pegado por la base a la entidad:



Generalización y Especialización

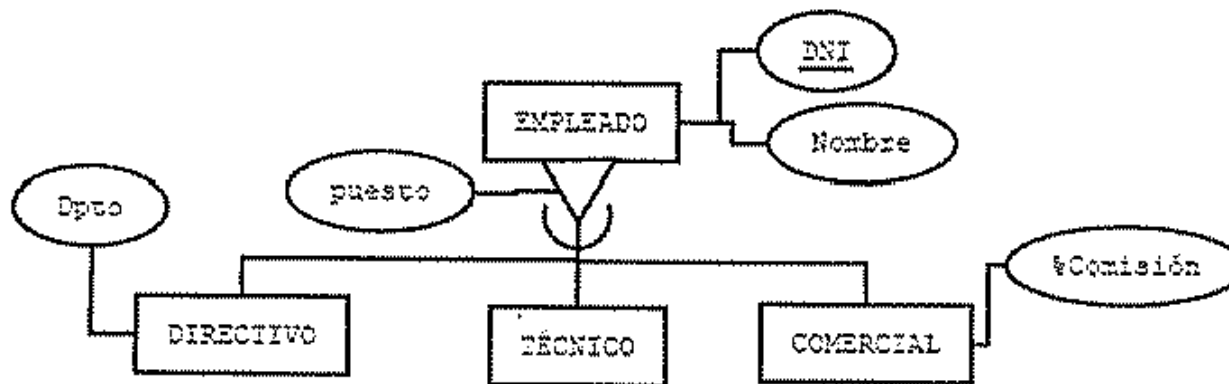
- ❖ En la figura siguiente Empleado es la superclase y los directivos, comerciales y técnicos son subclase
- ❖ En la relación se añade un atributo que indica cómo debe interpretarse la relación de la superclase con la subclase.
- ❖ La generalización Empleado que puede ser un directivo, un técnico o un comercial.
- ❖ Cada subentidad tiene sus propios atributos y relaciones, pero todas heredan los atributos nombre y DNI de la entidad padre (Empleado).

Generalización y Especialización



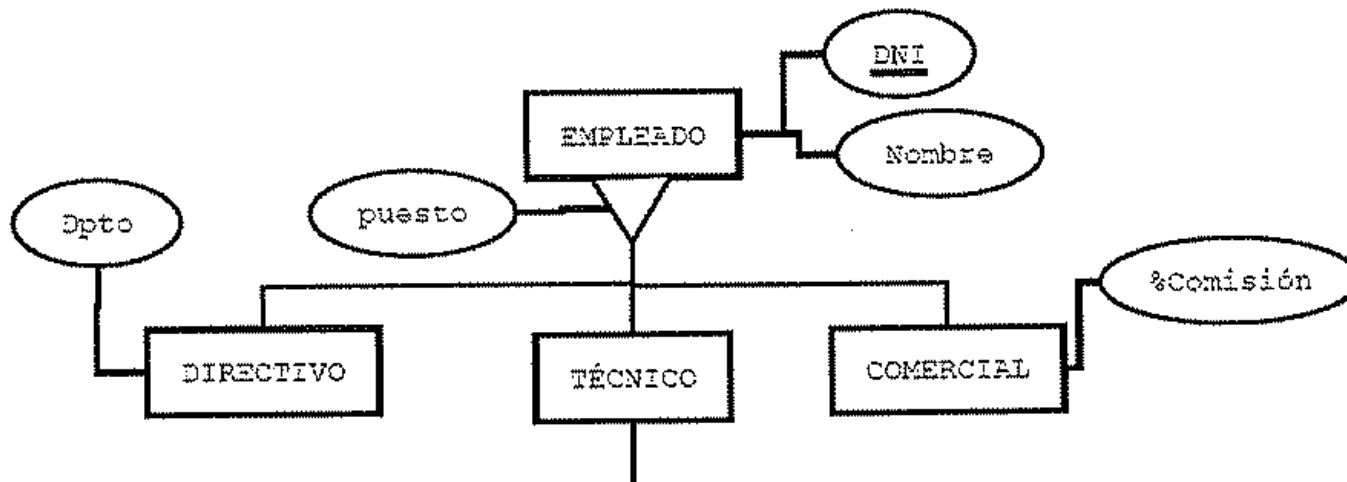
Tipos de especialización

- ❖ **Especialización Exclusiva:** En este caso, cada una de las ocurrencias de la superclase solo puede materializarse en una de las especializaciones.
- ❖ Por ejemplo, si un empleado es un directivo, no puede ser un técnico o un comercial.
- ❖ Para representar esta **especialización exclusiva**, el triángulo de la jerarquía lleva un **arco**.



Tipos de especialización

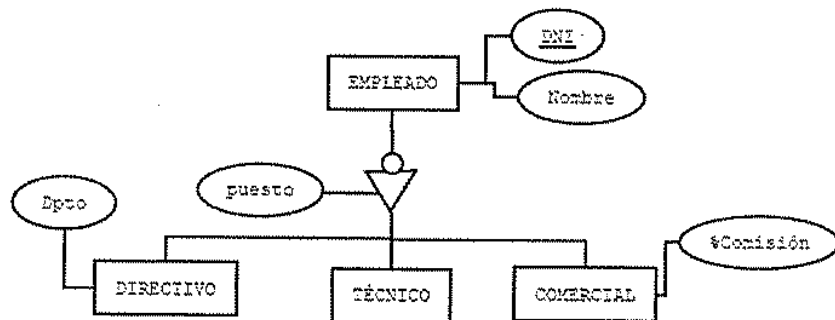
- ❖ **Especialización Inclusiva:** Se produce cuando las ocurrencias de la superclase pueden materializarse a la vez en varias ocurrencias de las subclases.
- ❖ En este caso, el empleado directivo, podría ser también técnico y comercial.
- ❖ Se representa **sin el arco**.



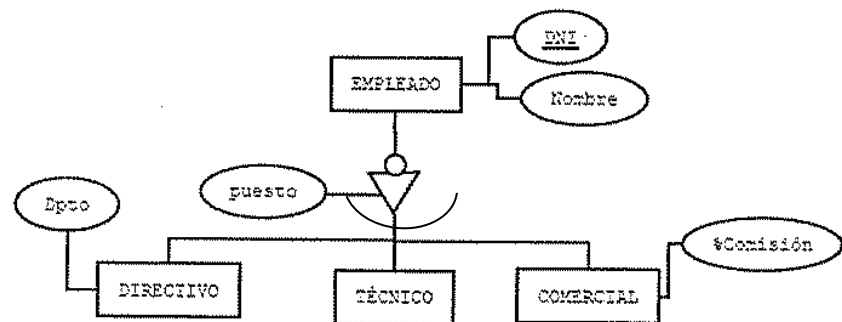
Tipos de especialización

- ❖ **Especialización Total:** Se produce cuando la entidad superclase tiene que materializarse obligatoriamente en una de las especializaciones.
- ❖ Se representan añadiendo un pequeño **círculo al triángulo** de la generalización.

Especialización inclusiva total
(debe ser una o varias)



Especialización exclusiva total
(debe ser una)

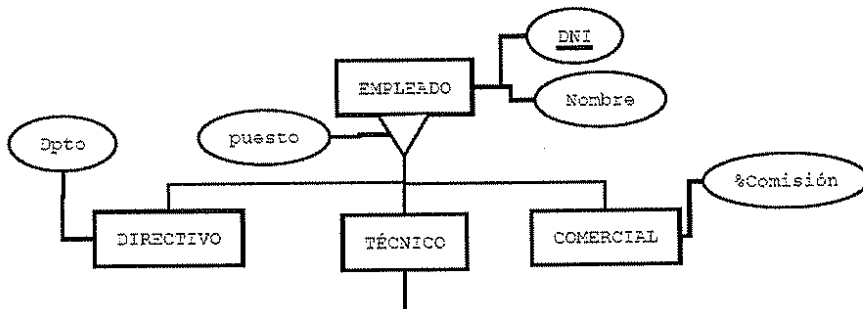


Tipos de especialización

- ❖ **Especialización Parcial:** La entidad superclase no tiene por qué materializarse en una de las especializaciones (es opcional).
- ❖ Se representa **sin el pequeño círculo**.

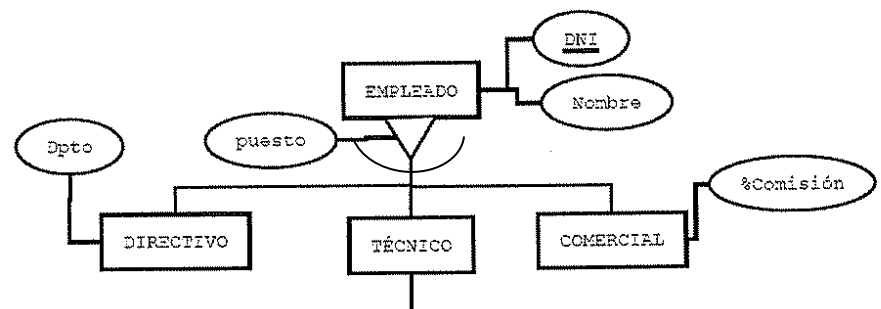
Especialización inclusiva parcial

(puede ser una, varias o ninguna)



Especialización exclusiva

(puede ser una o ninguna)



Construcción de un diagrama E /R

1. Leer varias veces el problema hasta memorizarlo.
2. Obtener una lista inicial de candidatos a entidades, relaciones y atributos.
3. Se realiza siguiendo los siguientes consejos:
 - Identificar las **entidades**.
 - Se anotan las **entidades**, sin especificar si son **fuertes** o **débiles**.
 - Extraer los **atributos** de cada entidad, identificando aquellos que pueden ser **clave**.
 - **Tipo de atributo**, seleccionando si es opcional, obligatorio, multivaluado, compuesto o derivado.

Construcción de un diagrama E /R

- Identificar las **generalizaciones** si se obtiene un atributo que es aplicable a más de una entidad (generalización/especialización).
- También es posible que los nombres comunes contengan muy poca información y no sea posible incluirlas como entidades. (Autor de un Libro).
- Extraer los **dominios** de los atributos.
- **Identificar las relaciones.** Se pueden ver extrayendo los **verbos** del texto del problema.
- **Identificar los atributos de cada relación.** Se suelen distinguir, al igual que los de entidad, por ser adjetivos, teniendo en cuenta que para que sean de relación, solo deben ser aplicables a la relación y **no** a ninguna de las de las **entidades relacionadas**.
- Una vez identificada las relaciones, hay que afinar cómo afecta la relación a las entidades implicadas. Este es el momento de distinguir las **fuertes** de las **débiles**

Construcción de un diagrama E /R

4. Averiguar las **participaciones** y **cardinalidades**.
5. Poner todos los elementos listados en el paso 2 en un mapa.
6. Refinar el diagrama hasta que se eliminen todas las incoherencias posibles.
7. Si hay dudas sobre el enunciado o sobre los requisitos: entrevista usuario.

Contenidos

- ❖ Representación del problema
- ❖ Modelo de datos
- ❖ Diagramas E/R
- ❖ El modelo E/R ampliado
- ❖ **El modelo relacional**
- ❖ Transformación E/R al modelo relacional
- ❖ Normalización

El modelo relacional

- ❖ Es el modelo más utilizado para modelar problemas reales.
- ❖ Es un modelo lógico que construye una estructura de los datos.
- ❖ A partir de este modelo se podrá realizar el modelado físico en el Sistema Gestor de Base de Datos que se desee.

El modelo relacional: Características

❖ Características fundamentales del modelo relacional:

- La **relación** es el elemento fundamental del modelo. Los usuarios ven la base de datos como una colección de relaciones.
- El modelo relacional es independiente de la forma en que se almacenan los datos y de la forma de representarlos; por tanto, la base de datos se puede representar en cualquier SGBD.

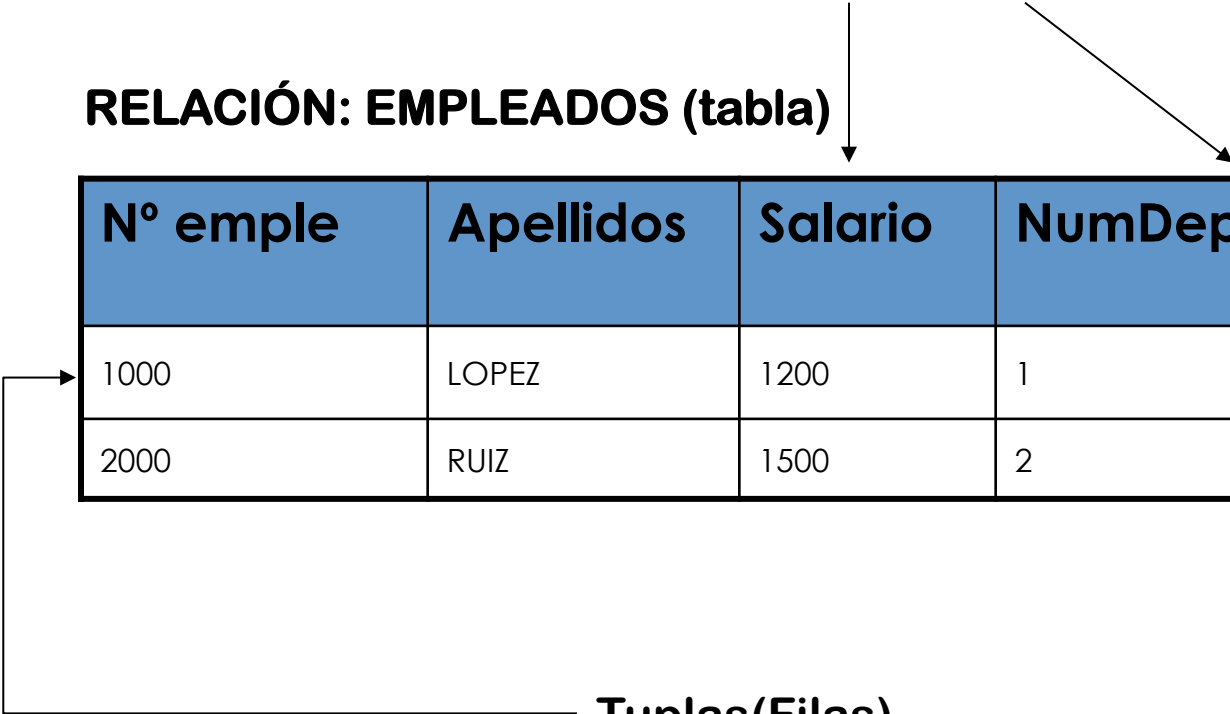
Relaciones, Atributos y Dominios

- ❖ Se define una **relación** como un conjunto de atributos, cada uno de los cuales pertenece a un **dominio**, y que posee un nombre que identifica la relación.
- ❖ Se representa gráficamente por una tabla con columnas (atributos) y filas (tuplas).
- ❖ **Atributo**: Características que describen a una entidad o relación.
- ❖ **Dominio**: Conjunto de valores permitidos para un atributo. Por ejemplo, cadenas de caracteres, números enteros, los valores Sí o No, etc.

Relaciones

Atributos (Columnas)

RELACIÓN: EMPLEADOS (tabla)



Nº emple	Apellidos	Salario	NumDepart	Fecha Alta
1000	LOPEZ	1200	1	01/01/2005
2000	RUIZ	1500	2	01/02/2005

Tuplas (Filas)

Modelo relacional: Restricciones

❖ **Restricciones de semántica:** Condiciones que deben cumplir los datos para su correcto almacenamiento.

❖ **Tipos:**

- Restricciones de clave
- Restricciones de valor único
- Restricciones de integridad referencial
- Restricciones de dominio
- Restricciones de verificación
- Restricción de valor NULO

Modelo relacional: Restricciones

- ❖ **Restricciones de clave** : Es el conjunto de atributos que identifican de forma única a una entidad.
- ❖ **Restricciones de valor único**: Es una restricción que impide que un atributo tenga un valor repetido.
 - Todos los atributos clave cumplen esta restricción.
 - No obstante es posible que algún atributo no sea clave y requiera una restricción de valor único.
 - Por ejemplo, el número de bastidor de un vehículo no es clave (lo es la matrícula) y sin embargo, no puede haber ningún número de bastidor repetido

Modelo relacional: Restricciones

- ❖ **Restricciones de integridad referencial:** Se da cuando una tabla tiene una referencia a algún valor de otra tabla. En este caso la restricción exige que exista el valor referenciado en la otra tabla.
- ❖ **Por ejemplo:**
 - No se puede poner una nota a un alumno que no exista.
 - No se puede vender un artículo que no esté creado.
 - ...

Modelo relacional: Restricciones

- ❖ **Restricciones de dominio:** Esta restricción exige que el valor que puede tomar un campo esté dentro del dominio definido.
- ❖ Por ejemplo:
 - Si se establece que un campo DNI pertenece al dominio de los números de 9 dígitos + 1 letra, no es posible insertar un DNI sin letra, puesto que la restricción obliga a poner al menos 1 letra.
 - Si se establece que un salario es de tipo real, sólo puede incluir números y el punto decimal

Modelo relacional: Restricciones

- ❖ **Restricciones de verificación:** Esta restricción permite comprobar si un valor de un atributo es válido conforme a una expresión.
 - Por ejemplo, que el atributo SEXO tenga el valor “H” o “M”.
- ❖ **Restricción de valor NULO:** Un atributo puede ser obligatorio si no admite el valor NULO o NULL, es decir, el valor falta de información o desconocimiento. Si admite como valor el valor NULL, el atributo es opcional.

Modelo relacional: Clave

- ❖ Una **clave** es un conjunto de atributos que identifican de forma única una ocurrencia de entidad.
- ❖ En este caso, las claves pueden ser:
 - **Simple** (atómicas): está formada por un solo atributo
 - **Compuestas**: formada por varios atributos.

Modelo relacional: Tipos de Clave

- ❖ **Clave Candidata:** son todos los atributos que identifican a cada ocurrencia de una entidad.
 - Por ejemplo: entidad EMPLEADOS, sus claves candidatas son: DNI y NRO Seguridad Social.
- ❖ **Clave Primaria:** Es la clave candidata elegida por el diseñador como clave definitiva
 - En el ejemplo anterior se elegiría el DNI por ser la más representativa para un empleado.
- ❖ **Clave ajena o foránea:** Es un atributo de una entidad, que es clave en otra entidad.
 - Por ejemplo, en la entidad NOTA tiene el atributo: DNI del alumno, que es clave en otra entidad. En este caso, el DNI en la tabla notas es una clave foránea o ajena de la tabla ALUMNOS.

Contenidos

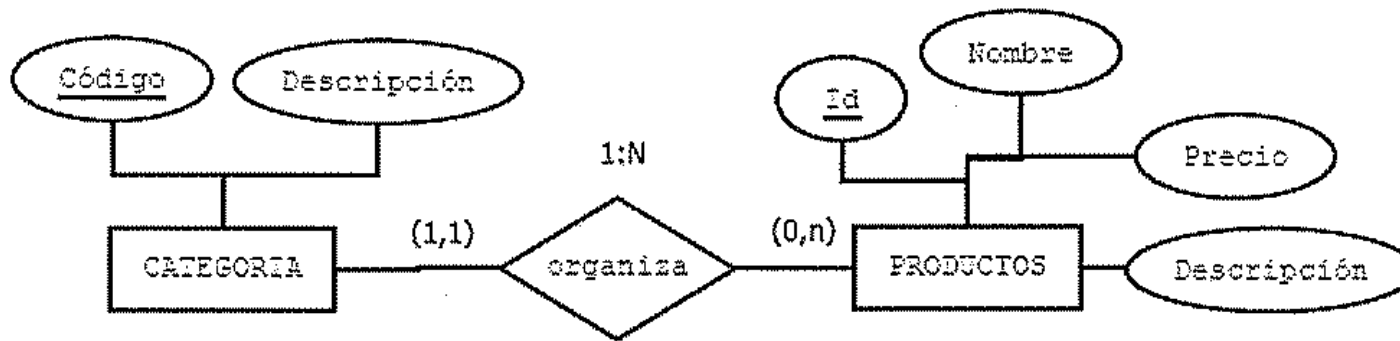
- ❖ Representación del problema
- ❖ Modelo de datos
- ❖ Diagramas E/R
- ❖ El modelo E/R ampliado
- ❖ El modelo relacional
- ❖ **Transformación E/R al modelo relacional**
- ❖ Normalización

Transformación de entidades fuertes

- ❖ Para cada **entidad** A (entidad fuerte con atributos: a_1, a_2, \dots, a_n) se crea una **tabla** A (con el nombre en plural) .
- ❖ Tendrá n **columnas** correspondientes a los atributos de A.
- ❖ Los **campos calculados** no se pasaran a las tablas en el modelo relacional.
- ❖ Cada **fila** de la tabla A corresponde a una ocurrencia de la entidad A.
- ❖ La **clave primaria** de la tabla A la forman los **atributos clave** de la entidad A.
- ❖ La clave primaria irá subrayada.

Transformación de entidades fuertes

❖ Ejemplo:



❖ La transformación sería:

CATEGORIAS(Código, Descripción)

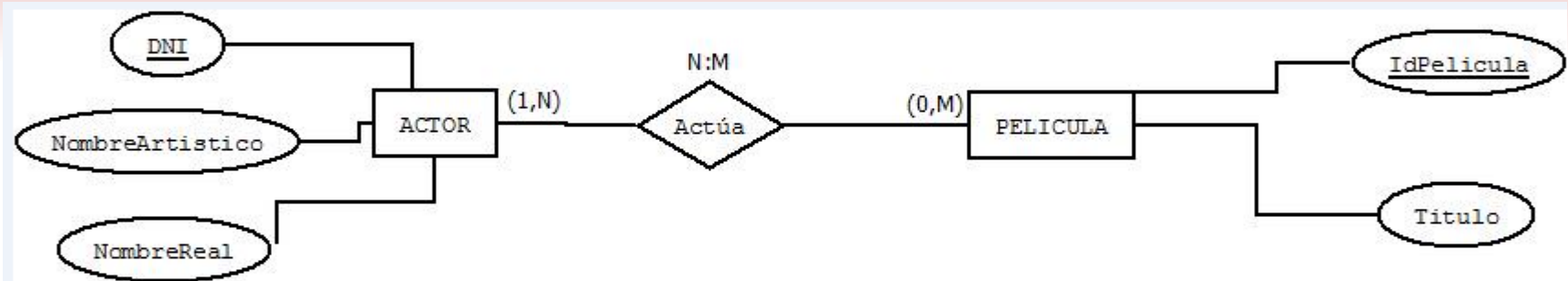
PRODUCTOS (Id,Nombre,Precio,Descripción)

Transformación de relaciones N:M

- ❖ Se crea una tabla con todos los campos claves de las entidades relacionadas y los atributos de la relación.
- ❖ La **clave primaria** de la tabla generada es la suma de los atributos claves de las entidades relacionadas (puede haber algún atributo de la relación que forme parte de la clave, por ejemplo una fecha)
- ❖ Cada clave incorporada a la tabla, será una clave ajena que referencia a la tabla de la que se importa.
- ❖ Cuando la relación es N^{aria} la transformación es igual que en las relaciones binarias.

Transformación de Relaciones N:M

❖ Ejemplo:



❖ Transformación:

ACTORES(DNI,NombreArtístico,Nombrereal)

PELICULAS(IdPelicula, Titulo)

ACTORES_PELICULAS(DNI, IdPelicula)

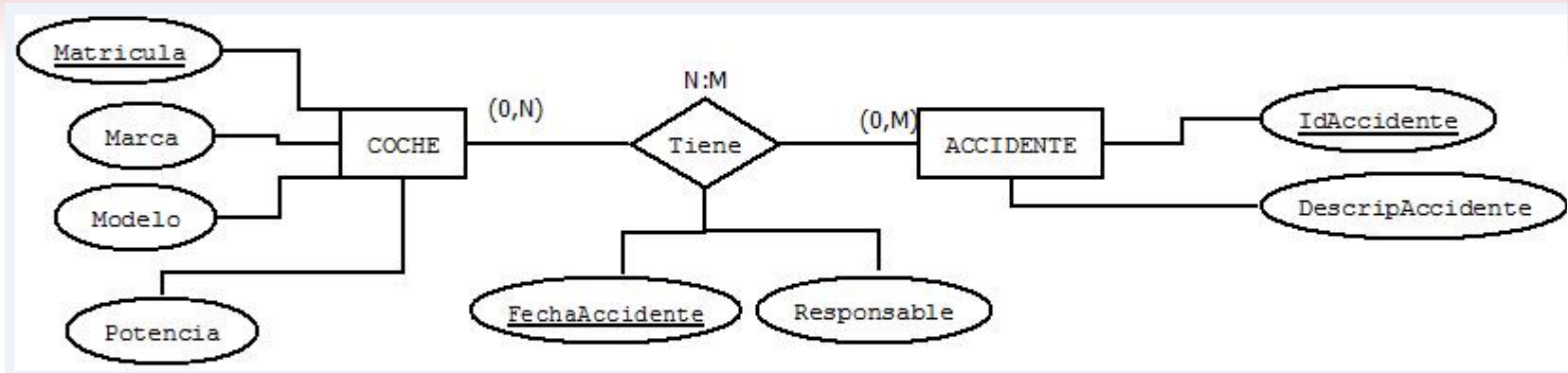
Clave Primaria: DNI

Claves Ajenas: DNI(ACTORES)

IdPelicula(PELICULAS)

Transformación de Relaciones N:M

❖ Ejemplo:



❖ Transformación:

COCHES(Matricula, Marca, Modelo, Potencia)

ACCIDENTES(IdAccidente, DescripcionAccidente)

ACCID_COCHES(Matricula,
IdAccidente, FechaAccidente, Responsable)

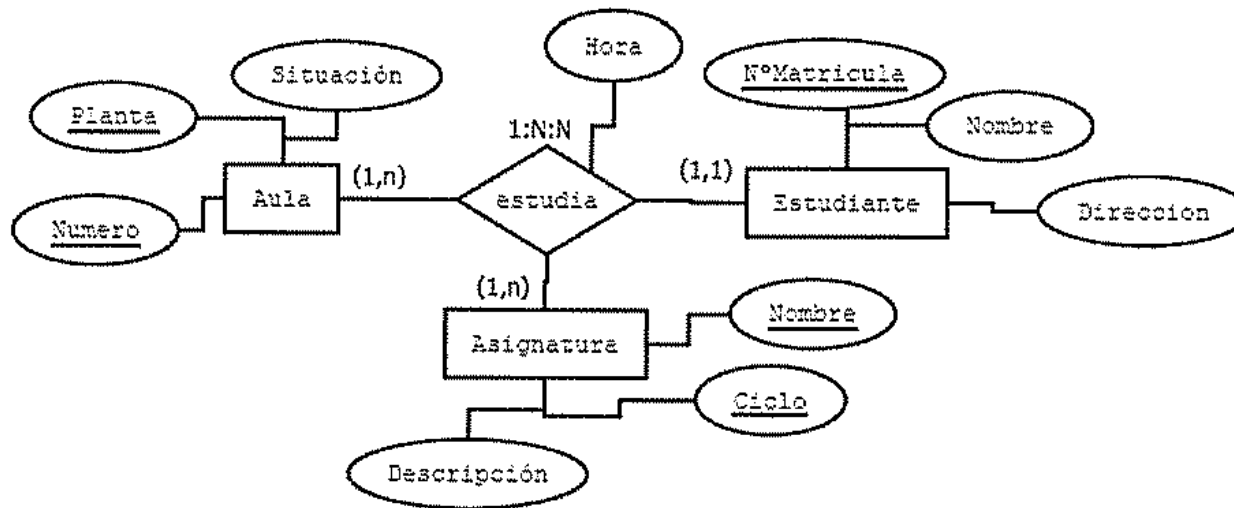
Clave Primaria: Matricula+IdAccidente+FechaAccidente

Claves ajenas: Matricula (COCHES)

IdAccidente(ACCIDENTES)

Transformación de Relaciones N:M

❖ Ejemplo:



❖ Transformación:

AULAS (Numero, Planta, Situación)

Clave Primaria: Numero

ESTUDIANTES (NºMatricula, Nombre, Dirección)

Clave Primaria: NºMatricula

ASIGNATURAS (Nombre, Ciclo, Descripción)

Clave Primaria: Nombre+Ciclo

ESTUDIOS (Numero, Planta, NºMatricula, Nombre, Ciclo, Hora)

Clave Primaria: Numero+Planta+NºMatricula,+Nombre+Ciclo

Claves Ajenas: Nº Matricula(ESTUDIANTES)

Numero+Planta (AULA)

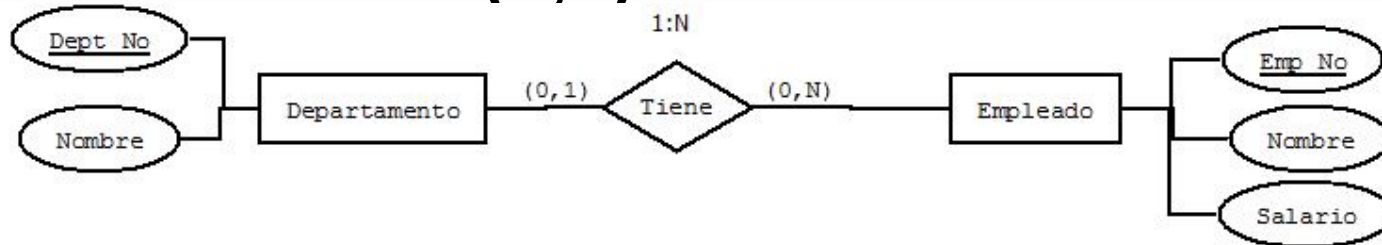
Nombre+Ciclo (ASIGNATURAS)

Transformación de Relaciones 1:N

- ❖ Se añade a la tabla de la entidad que actúa con participación máxima N, la clave de la entidad que actúa con participación máxima 1 como **clave ajena o foránea**.
- ❖ Si además, la relación tuviera atributos se importarían también a la entidad que actúa con participación máxima N.

Transformación de Relaciones 1:N

- ❖ Si tenemos una relación 1:N en la que la cardinalidad sea (0,1):



- ❖ La transformación se realiza creando una **nueva tabla** para evitar los nulos en la tabla EMPLEADOS:

DEPARTAMENTOS(Dept_No,Nombre)

Clave Primaria: Dept_No

EMPLEADOS(Emp_No,Nombre,Salario)

Clave Primaria:Emp_No

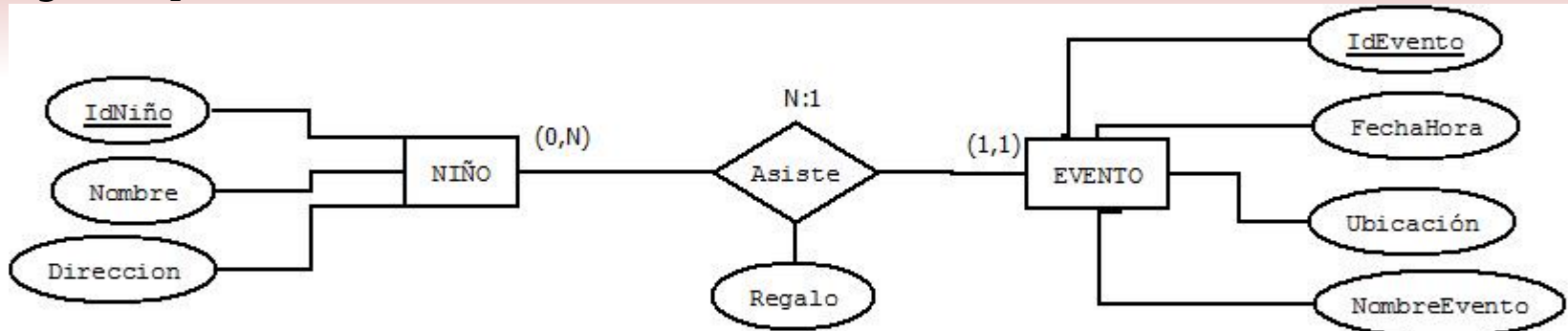
EMPLE_DEPART(Emp_No,Dept_No)

Clave Primaria: Emp_No

Clave Ajena: Dept_No

Transformación de Relaciones 1:N

❖ Ejemplo:



❖ Transformación:

NIÑOS(IdNiño,Nombre,Direccion,**IdEvento,Regalo**)

Clave Primaria: IdNiño

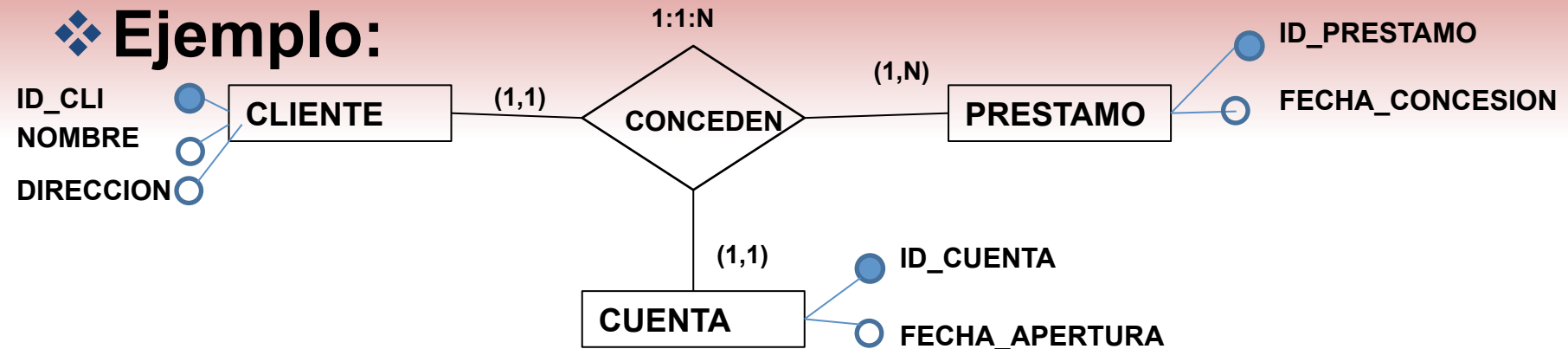
Claves ajenas: IdEvento (EVENTOS)

EVENTOS (IdEvento,FechaHora,Ubicación,NombreEvento);

Clave Primaria: IdEvento

Transformación de Relaciones 1:N

❖ Ejemplo:



❖ Transformación:

CLIENTES(ID_CLI, NOMBRE, DIRECCION)

Clave Primaria: ID_CLI

CUENTAS(ID_CUENTA, FECHA_APERTURA)

Clave Primaria: ID_CUENTA

PRESTAMOS(ID_PRESTAMO, FECHA_CONCESION, ID_CLI, ID_CUENTA)

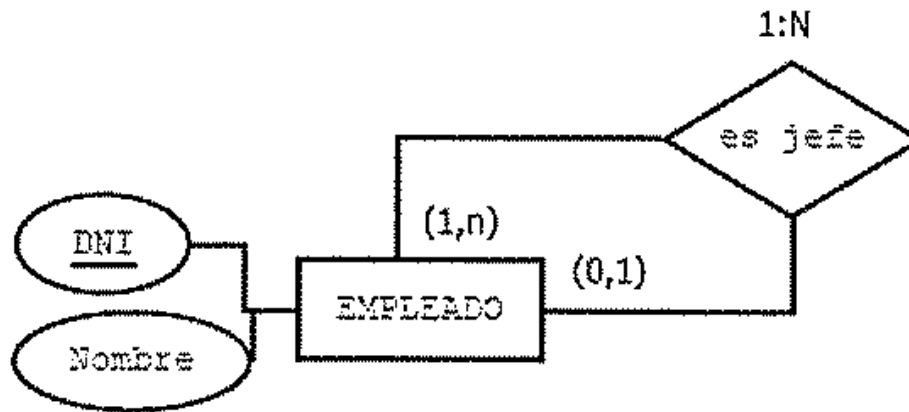
Clave Primaria: ID_PRESTAMO

Claves Ajenas: ID_CLI(CLIENTES)

ID_CUENTA(CUENTAS)

Transformación de las Relaciones reflexivas

- ❖ **Cardinalidad 1:N:** Se crearía en la tabla una clave ajena de la clave principal

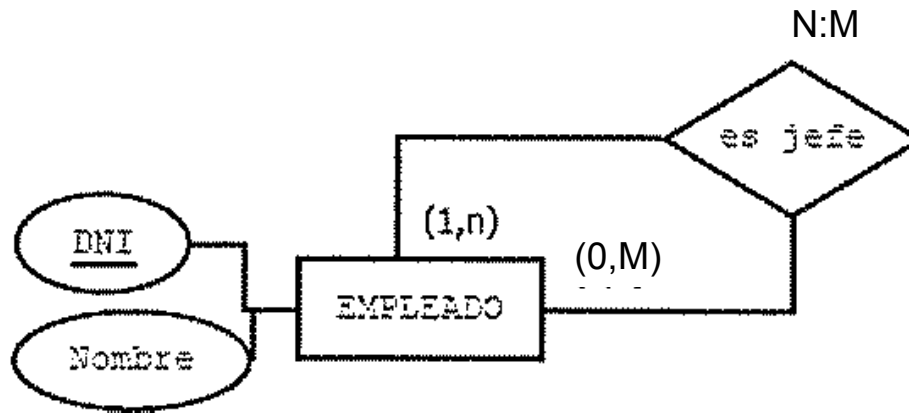


EMPLEADOS(DNI, Nombre, **DNI_JEFE);**

Clave ajena: DNI_JEFE(EMPLEADO)

Transformación de las Relaciones reflexivas

- ❖ **Cardinalidad N:M:** Se crearía una nueva tabla que tendrá como clave principal la clave primaria dos veces.



EMPLEADOS(DNI, Nombre)

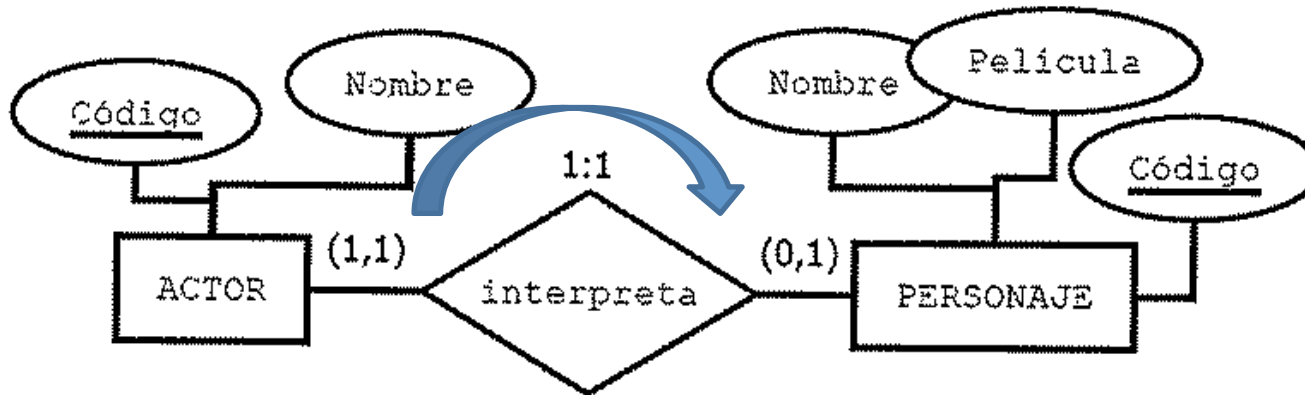
JEFES(DNI EMPLE, DNI JEFE)

Clave ajena: **DNI_EMPLE(EMPLEADOS)** es el empleado
DNI_JEFE(EMPLEADOS) son los jefes

Transformación de las Relaciones

1:1- (1,1) y (0,1)

- ❖ Depende de la cardinalidad y la transformación se realiza propagando la clave de la entidad con cardinalidad (1,1) a la de (0,1)



ACTORES(CodigoActor,Nombre)

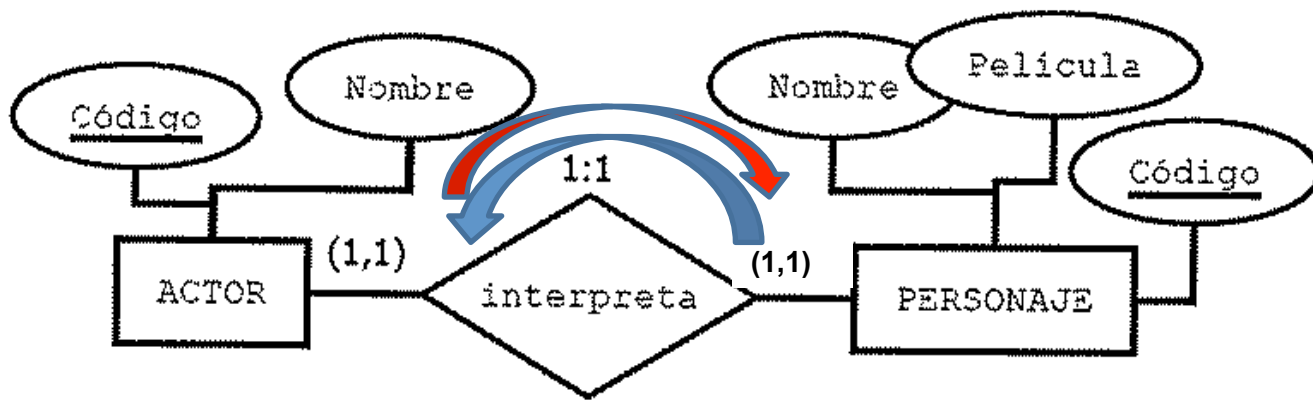
PERSONAJES(CódigoPersonaje,Nombre,Pelicula,CodigoActor)

Claves ajenas: CodigoActor(ACTORES)

Transformación de las Relaciones

1:1- (1,1) y (1,1)

- ❖ Cuando las cardinalidades son las dos (1,1), la clave se propaga indistintamente.

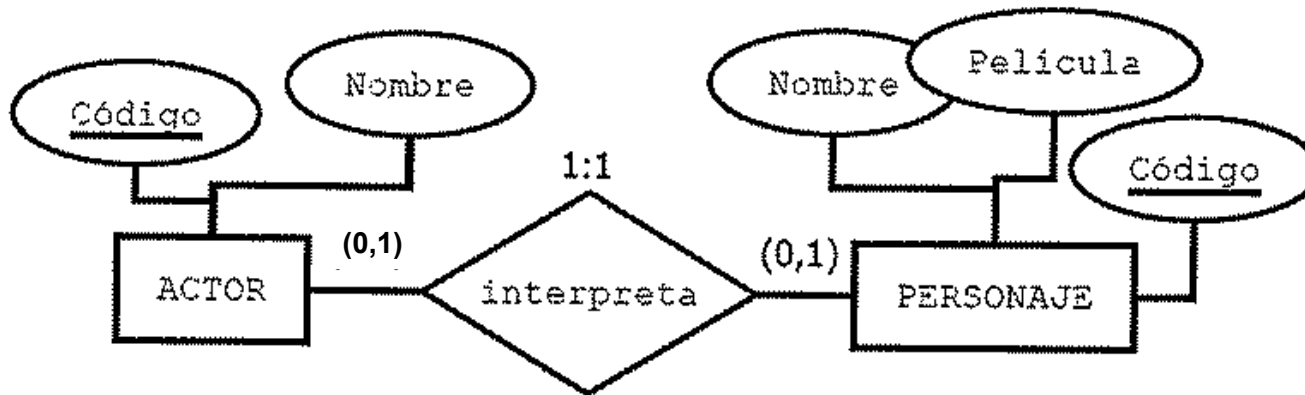


- ❖ Caso 1º:
ACTORES(CodigoActor,Nombre)
PERSONAJES(CódigoPersonaje,Nombre,Pelicula,CodigoActor**)**
Claves ajenas: CodigoActor(ACTORES)
- ❖ Caso 1º:
ACTORES(CodigoActor,Nombre,CódigoPersonaje**)**
Claves ajenas: CodigoActor(PERSONAJES)
PERSONAJES(CódigoPersonaje,Nombre,Pelicula)

Transformación de las Relaciones

1:1- (0,1) y (0,1)

- ❖ Cuando las cardinalidades son las dos (0,1), se creará una nueva tabla.



ACTORES(CodigoActor,Nombre)

PERSONAJES(CódigoPersonaje,Nombre,Pelicula)

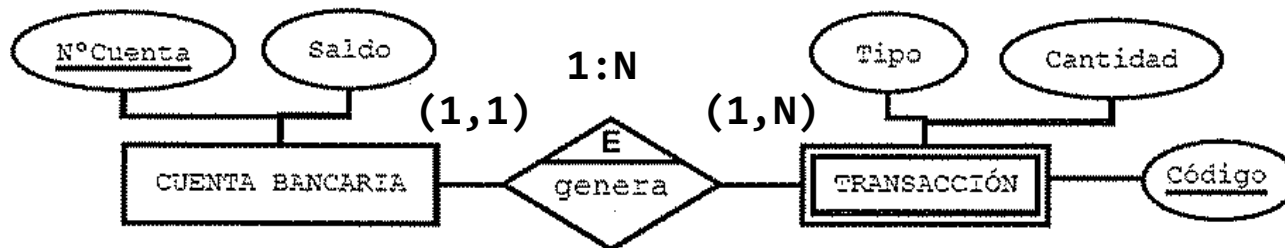
ACTOR_PERSONAJE(CodigoActor, CódigoPersonaje)

Claves ajenas: CodigoActor(ACTORES)

CodigoPersonaje(PERSONAJES)

Transformación de entidades débiles

- ❖ **Relaciones de dependencias en existencia:** La clave de la entidad fuerte pasa como clave ajena en la débil.



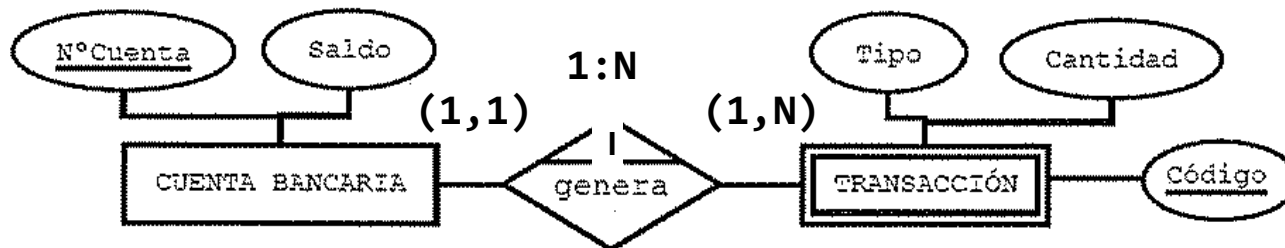
CUENTAS(Nro_cuenta,Saldo)

TRANSACCIONES(Codigo,Nro_cuenta, tipo,Cantidad)

Claves Ajenas: Nro_cuenta(CUENTAS)

Transformación de entidades débiles

- ❖ **Relaciones de dependencias en identificación:**
La clave de la entidad fuerte pasa como parte de la clave primaria en la débil.



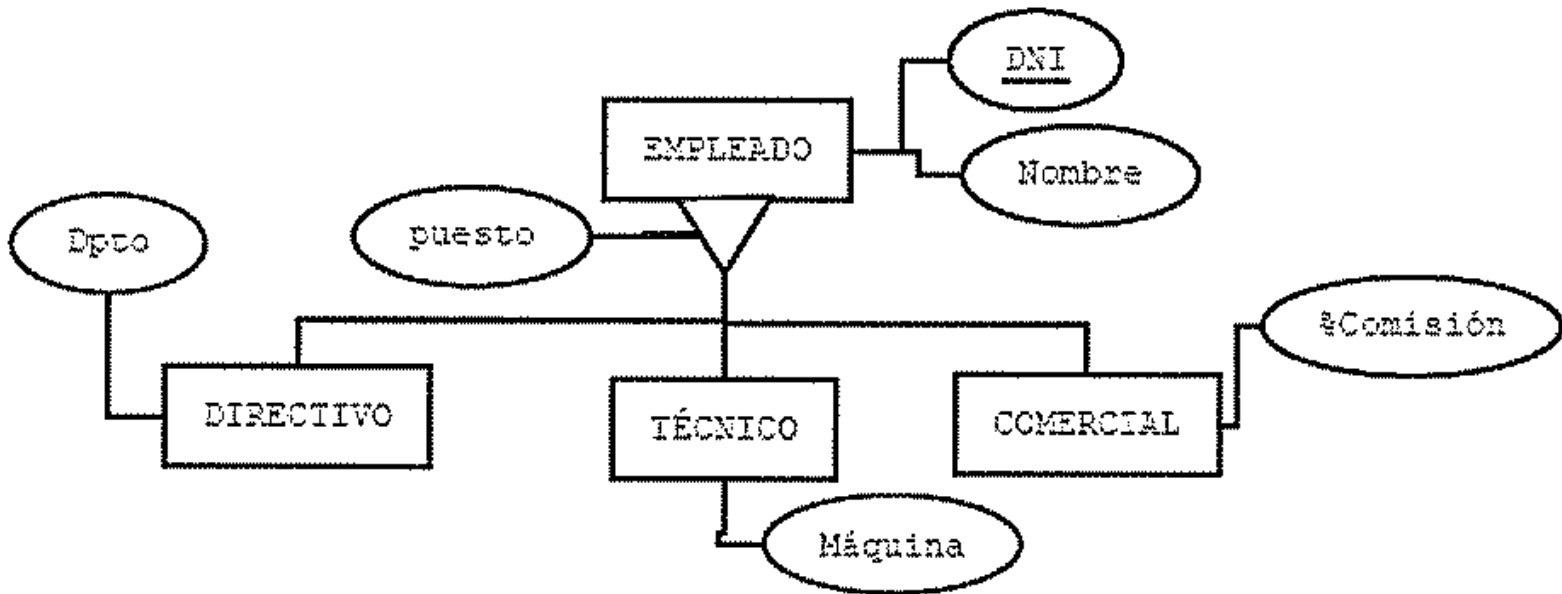
CUENTAS(Nro cuenta,Saldo)

TRANSACCIONES(Codigo, Nro cuenta,tipo,Cantidad,)

Claves Ajenas: Nro_cuenta(CUENTAS)

Transformación de las Generalización

- ❖ Para transformar las generalizaciones se puede elegir entre 5 opciones. Cada opción se adaptará mejor o peor a los diferentes tipos de especialización (Exclusiva, Inclusiva, Total, Parcial).



Transformación de las Generalización: Opción 1

- ❖ Se puede crear una tabla para la superclase y otras tantas para cada subclase, incorporando el campo clave de la superclase a las tablas de las subclases.
- ❖ Soportan : especializaciones exclusivas, no inclusivas.

EMPLEADOS(DNI,Nombre,Puesto)

DIRECTIVOS(DNI,Dpto)

Claves ajenas: DNI(EMPLEADOS)

TECNICOS(DNI,Máquinas)

Claves ajenas: DNI(EMPLEADOS)

COMERCIALES(DNI,Comisión)

Claves ajenas: DNI(EMPLEADOS)

Transformación de las Generalización: Opción 2

- ❖ Se puede crear una tabla para cada subclase incorporando todos los atributos de la clase padre, y no crear una tabla para la superclase.
- ❖ Soportan : especializaciones exclusivas e inclusivas.

DIRECTIVOS(DNI,Nombre,Dpto)

TECNICOS(DNI,Nombre,Máquinas)

COMERCIALES(DNI,Nombre,Comisión)

Transformación de las Generalización: Opción 3

- ❖ Se puede crear una sola tabla para la superclase, incorporando los atributos de todas las subclases y añadir una columna donde le indiquemos el tipo de empleado.
- ❖ Soportan : especializaciones exclusivas y no inclusivas.

**EMPLEADOS(DNI, Nombre,Puesto,
Dpto, Máquinas, Comisión)**

Transformación de las Generalización: Opción 4

- ❖ Se puede crear una sola tabla para la super clase como en la opción anterior, pero en lugar de añadir un campo "tipo", se añaden varios campos que indiquen si cumple un perfil.
- ❖ Soportan : especializaciones exclusivas e inclusivas (mas orientadas a las inclusivas).

**EMPLEADOS(DNI, Nombre,Dpto,Máquinas,
Comisión, EsDirectivo,
EsTécnico,EsComercial)**

Transformación de las Generalización: Opción 5

- ❖ Se puede crear una tabla para la super clase, pero en lugar de añadir un campo "tipo", se añaden varios campos que indiquen si cumple un perfil.
- ❖ Soportan : especializaciones exclusivas e inclusivas (orientado a las inclusivas).
- ❖ **EMPLEADOS(DNI,Nombre, EsDirectivo, EsTécnico,EsComercial)**
DIRECTIVOS(DNI,Dpto)
Claves ajenas: DNI(EMPLEADOS)
- TECNICOS(DNI,Máquinas)**
Claves ajenas: DNI(EMPLEADOS)
- COMERCIALES(DNI,Comisión)**
Claves ajenas: DNI(EMPLEADOS)

Contenidos

- ❖ Representación del problema
- ❖ Modelo de datos
- ❖ Diagramas E/R
- ❖ El modelo E/R ampliado
- ❖ El modelo relacional
- ❖ Transformación E/R al modelo relacional
- ❖ **Normalización**

Normalización

- ❖ Habitualmente, el diseño de una base de datos termina en el paso del modelo entidad-relación al modelo relacional.
- ❖ En algunos casos, hay que realizar sucesivos refinamientos en el diseño, para alcanzar la calidad deseada.
- ❖ Uno de los parámetros que mide la calidad de una base de datos es la **forma normal en la que se encuentra su diseño**.
- ❖ El proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales se llama **normalización**.

Normalización

❖ Objetivos:

1. Almacenar en la base de datos cada dato solo una vez, es decir, evitar la **redundancia de datos**. De esta manera se reduce el espacio de almacenamiento y facilita la actualización de los datos.
2. Obtener una estructura lógica adecuada que venga a establecer un equilibrio entre las exigencias de los usuarios y la eficiencia.

Normalización

- ❖ Hay definidas 6 formas normales:
 - Codd definió la **1FN, 2FN y 3FN**.
 - Con posterioridad, se definieron la forma normal de **BOYCE/CODD (FNBC)** y las **4FN y 5FN**.
- ❖ Cada una agrupa a las anteriores, de forma que, por ejemplo, la forma normal 3 cumple la forma normal 2 y la forma normal 1.

Normalización:

Dependencia funcional

- ❖ **Dependencia funcional:** Se dice que un atributo **Y** depende funcionalmente de otro atributo **X**, o que $X \rightarrow Y$, si cada valor de **X** tiene asociado en todo momento un único valor de **Y**. También se dice que **X implica Y**
- ❖ Por ejemplo:

PRODUCTOS (Idproducto, Nombre, Precio)

IdProducto \rightarrow **Nombre**, puesto que un identificador de producto solo puede tener asociado un único nombre, dicho de otro modo, a través del identificador de producto se localiza un único nombre.

Normalización:

Dependencia funcional

❖ Por ejemplo:

VENTAS (IdProducto, IdCliente, FechaVenta,
NombreProducto, PrecioVenta, Unidades)

NombreProducto no tiene dependencia funcional con **FechaVenta**, puesto que para un mismo nombre de producto puede tener asociado varias fechas para cada venta de ese producto.

NombreProducto si tiene dependencia funcional con **IdProducto**.

Normalización:

Dependencia funcional completa

- ❖ **Dependencia funcional completa:** Dado una combinación de atributos **X**(**X1** , **X2** , ••), se dice que **Y** tiene dependencia funcional completa de **X**, o que **X** => **Y**, si depende funcionalmente de **X**, pero no depende de ningún subconjunto del mismo.
- ❖ Por ejemplo:

COMPRAS (**IdProducto**, **IdProveedor**, **FechaCompra**, **NombreProducto**,
Cantidad)

- ❖ **IdProducto, IdProveedor** => **FechaCompra**, puesto que la **FechaCompra** es única para la combinación de **IdProducto** y **IdProveedor** (se puede hacer un pedido al día de cada producto a cada proveedor), y sin embargo, se pueden hacer varios pedidos del mismo producto a diferentes proveedores, es decir, **IdProducto** no tiene dependencia funcional completa de **Fecha**.
- ❖ **IdProducto, IdProveedor** no tiene dependencia funcional completa con **NombreProducto**, porque **NombreProducto** depende funcionalmente de **IdProducto**.

Normalización:

Dependencia funcional transitiva

- ❖ **Dependencia funcional transitiva:** Dada la tabla T, con atributos (X,Y,Z), donde $X \rightarrow Y$, $Y \rightarrow Z$, se dice que X depende transitivamente de Z, o que, $X \twoheadrightarrow Z$.
- ❖ Ejemplo 1:
PRODUCTOS (IdProducto, NombreProducto, Fabricante, PaísFabricante)
 $\text{IdProducto} \rightarrow \text{Fabricante}$
 $\text{Fabricante} \rightarrow \text{PaísFabricante}$
 $\text{IdProducto} \twoheadrightarrow \text{País}$, es decir, IdProducto depende transitivamente de PaísFabricante.

Normalización:

Primera Forma Normal (1FN)

- ❖ En esta forma normal no se permite que en una tabla haya atributos que puedan tomar más un valor.

EMPLEADOS

nss	nombre	puesto	salario	emails
111	Juan Pérez	Jefe de Área	3000	<u>juanp@ecn.es</u> <u>jefe2@ecn.es</u>
222	José Sánchez	Administrativo	1500	<u>jsanchez@ecn.es</u>
333	Ana Díaz	Administrativo	1500	<u>adiaz@ecn.es</u> <u>ana32@gmail.com</u>

Normalización:

Primera Forma Normal (1FN)

- ❖ **SOLUCION1:** Duplicar los registros con valores repetidos en EMPLEADOS (No es el mas adecuado).

EMPLEADOS

nss	nombre	puesto	salario	email
111	Juan Pérez	Jefe de Área	3000	juanp@ecn.es
111	Juan Pérez	Jefe de Área	3000	jefe2@ecn.es
222	José Sánchez	Administrativo	1500	jsanchez@ecn.es
333	Ana Díaz	Administrativo	1500	adiaz@ecn.es
333	Ana Díaz	Administrativo	1500	ana32@gmail.com

Normalización:

Primera Forma Normal (1FN)

❖ **SOLUCIÓN 2:** separar el atributo que viola la 1FN en otra tabla EMAILS

nss	nombre	puesto	salario
111	Juan Pérez	Jefe de Área	3000
222	José Sánchez	Administrativo	1500
333	Ana Díaz	Administrativo	1500
...

EMPLEADOS

nss	email
111	juanp@ecn.es
111	jefe2@ecn.es
222	isanchez@ecn.es
333	adiaz@ecn.es
333	ana32@gmail.com
...	...

CORREOS

EMPLEADOS(NSS,NOMBRE,PUESTO,SALARIO)

CORREOS (CORREO,NSS)

Clave Ajena: NSS (EMPLEADOS)

Normalización:

Segunda Forma Normal (2FN)

- ❖ Un diseño se encuentra en **2FN** si está en **1FN** y además, cada atributo que no forma parte de la clave tiene dependencia funcional completa de la clave principal y no con parte de la clave.
- ❖ Ejemplo:

COMPRAS (IdProducto, IdProveedor, FechaCompra,
NombreProducto, Cantidad)

- ❖ **IdProducto** → **NombreProducto**, es decir, como el Nombre depende funcionalmente del Código no es dependencia funcional completa, no está en 2FN.

Normalización: Segunda Forma Normal (2FN)

- ❖ **SOLUCIÓN:** Si no existe la tabla PRODUCTOS, crear una nueva tabla de esta forma:

COMPRAS (IdProducto, IdProveedor, FechaCompra,
Cantidad)

Clave Ajena: IdProducto(PRODUCTOS)

PRODUCTOS(IdProducto, NombreProducto)

Normalizacion:

Tercera Forma Normal (3FN)

- ❖ Un diseño se encuentra en **3FN** si está en **2FN** y además, no habrá ningún atributo no clave que dependa funcionalmente de otro atributo no clave clave.
- ❖ Ejemplo:
PRODUCTOS (IdProducto, Nombre, Fabricante, PaísFabr, TelefFabr)
 - IdProducto → Fabricante (un producto siempre es del mismo fabricante)
 - Fabricante → PaísFabr (un fabricante siempre tiene es del mismo país)
 - Fabricante → TelefFabr (un fabricante siempre tiene el mismo teléfono)
- ❖ Para el mismo fabricante el PaísFabr, TelefFabr son los mismos, luego dependen funcionalmente del fabricante.
- ❖ Cada vez que aparezca un producto que sea del mismo fabricante los datos país y teléfono se van a repetir.

Normalizacion: Tercera Forma Normal (3FN)

❖ **SOLUCIÓN:** dividir la tabla en dos

PRODUCTOS (IdProducto, Nombre, IdFabricante)

Clave Ajena: IdFabricante(FABRICANTES)

**FABRICANTES(IdFabricante, Fabricante, PaisFabr,
TelefFabr)**

Normalización :

Tercera Forma Normal (3FN)

❖ Tenemos la siguiente tabla:

VEHICULOS(IdVehiculo, Matricula, Marca, Modelo, TipoSeguro, DescripcionTipo, CantidadTipo);

<u>IdVehiculo</u>	Matricula	Marca	Modelo	TipoSeguro	DescripcionTipo	CantidadTipo
1	M1	MARCA 1	MODELO 1	O	OBLIGATORIO	300
2	M2	MARCA 2	MODELO 2	T	TODO RIESGO	700
3	M3	MARCA 1	MODELO 3	O	OBLIGATORIO	300
4	M4	MARCA2	MODELO 1	T	TODO RIESGO	700

Normalización:

Tercera Forma Normal (3FN)

❖ Vemos en la tabla anterior lo siguiente:

1. Cada vez que aparece una marca que se llame igual se repite dicho nombre, esta repetición hace que se utilice más memoria de la necesaria (por ej. , si el nombre de la marca ocupa, por ej, 100 caracteres, podemos en realidad cambiarlo por un n° que ocupa menos), se almacenarían las marcas en otra tabla. Además, a la hora de insertar nuevas marcas y actualizarlas estaría centrado en esa tabla, no tendríamos que ir a todas las tablas donde aparezcan las marcas. (Sirve igual para modelo).
2. Podemos observar que cada vez que aparece un tipo de seguro, la descripción y la cantidad son los mismos (cuando aparece tipo de seguro “O” , la descripción es “OBLIGATORIO” y la cantidad es 300. En este caso hay una dependencia funcional transitiva. No estaría en 3FN.

Normalizacion :

Tercera Forma Normal (3FN)

❖ SOLUCIÓN:

VEHICULOS(IdVehiculo, Matricula, IdMarca, IdModelo, TipoSeguro)

Claves Ajenas: IdMarca (MARCAS)

IdModelo(MODELOS)

TipoSeguro(TIPOSEGUROS)

TIPOSEGUROS(TipoSeguro, DescripcionTipo, CantidadTipo);

MARCAS (IdMarca , Marca)

MODELOS (IdModelo, Modelo)

Normalización:

Forma Normal de Boyce-Codd (FNBC)

- ❖ El problema de la **3FN** es que no maneja relaciones que:
 - Tiene varias claves candidatas,
 - Esas claves candidatas son compuestas, y
 - Las claves candidatas se solapan (o sea, tienen por lo menos un atributo en común).
- ❖ Por ello, se define la **FNBC** (Forma Normal de Boyce-Codd) para el caso en el que existan más de una clave candidata, y que se cumplan dichas condiciones. Hay que introducir, por comodidad, el concepto de determinante, que se define como el atributo del cual depende funcionalmente algún otro atributo.
- ❖ Así pues, se define la **FNBC** como:

*Una relación está en **FNBC** si y sólo si todo determinante es una clave candidata.*
- ❖ Se debe advertir que en el caso en el que no se den las condiciones anteriores, o no exista más de una clave candidata (sólo la clave primaria) la FNBC es completamente equivalente a la 3FN.

Normalización:

Forma Normañ de Boyce-Codd (FNBC)

❖ Ejemplo:

EstudProfesAlumno (Estudiante,Profesor,Asignatura).

- ❖ Donde una tupla significa que un estudiante E recibe la asignatura A impartida por el profesor P y en la cual se cumple:
 - para cada asignatura, un estudiante tiene un solo profesor.
 - cada profesor imparte sólo una asignatura.
 - cada asignatura es impartida por varios profesores.
- $E, A \multimap P$
 $P \multimap A$
- ❖ Por lo que $E, P \rightarrow A$ es una llave candidata de la relación, ya que determina a todos los atributos (E,P,A) y $E, A \rightarrow P$ es una llave candidata.
- ❖ Ambas llaves (EA y EP) son compuestas y se solapan, por lo que debe analizarse la FNBC

Normalización:

Forma Normañ de Boyce-Codd (FNBC)

- ❖ Esta relación está en 3FN, pero no en FNBC, ya que el determinante P no es llave de la relación.
- ❖ Esta relación presenta anomalías de actualización, por ejemplo:
- ❖ Eliminación: Si queremos eliminar la información de que Rdguez estudia Física, perdemos la información de que el profesor Hdez imparte Física.
- ❖ Estos problemas pueden eliminarse sustituyendo la relación original EAP por dos relaciones:

EstudianteProfesor (Estudiante,Profesor)

PorfesorAsignatura (Profesor,Asignatura)

Normalización FN4 y FN5

- ❖ La Cuarta y Quinta Formas Normales tratan con datos multivaluados (múltiples valores).
- ❖ Un dato multivaluado puede corresponder a una relación muchos a muchos, como con EMPLEADOS y HABILIDADES, o a una relación muchos a uno, como con los HIJOS de un EMPLEADO (asumiendo que sólo un pariente es empleado).

Normalización FN4 y FN5

- ❖ En un sentido, la Cuarta y Quinta Formas Normales están alrededor de claves compuestas.
- ❖ Estas formas normales procuran minimizar el número de campos involucrados en una clave compuesta.
- ❖ En nuestros diseños no vamos a utilizar las FN4 y FN5.