

Diseño Conceptual de Bases de Datos

1. Modelo de datos

[DeMP99]

Es un conjunto de conceptos, reglas y convenciones que permiten describir, a distintos niveles de abstracción, la estructura (**esquema**) de la BD. Según el nivel de abstracción en el que se encuentre la estructura de la BD, los modelos que permiten su descripción (cada uno de los cuales ofrece distintos elementos de descripción) serán:

- **modelos conceptuales (alto nivel)**: facilitan la descripción global del conjunto de información con independencia de la máquina (tanto del hardware como del SGBD), por lo que sus conceptos son cercanos al mundo real (entidades, atributos, relaciones, etc.). Son modelos de análisis, no de implementación. Ej: modelo E-R, diagrama de casos de uso (UML), etc.
- **modelos lógicos**: se encuentran soportados por los SGBD y describen los datos a nivel lógico, por lo que sus conceptos son propios de cada SGBD (tablas o relaciones en el caso del modelo Relacional, objetos para un SGBDOO, etc.)
- **modelos internos (físicos)**: están orientados a la máquina, siendo sus elementos de descripción punteros, índices, agrupamientos, etc.

2. Diseño conceptual

[SKS02]

2.1. Modelo Entidad-Relación

El modelo Entidad-Relación es un modelo de datos conceptual de alto nivel. Describe la percepción de los datos dirigida a los usuarios y no pretende describir la forma en que se van a almacenar internamente.

Para su explicación nos basaremos en el siguiente ejemplo: Supongamos una compañía formada por EMPLEADOS, DEPARTAMENTOS y PROYECTOS:

- la compañía está organizada en departamentos. Cada departamento tiene un número, nombre, varias localizaciones y un empleado que lo dirige. Nos interesa la fecha en que dicho empleado comenzó a dirigir el departamento. Un departamento puede estar distribuido en varios lugares.
- un departamento controla un conjunto de proyectos, cada uno de los cuales tiene un nombre, número y se efectúa en un solo lugar.
- se almacena el nombre de cada empleado, su número de S.S, dirección, salario, sexo y fecha de nacimiento. Un empleado se asigna a un departamento, pero puede trabajar en varios proyectos, que no tienen por qué ser controlados por el mismo departamento. Se almacena el número de horas por semana que cada empleado trabaja en cada proyecto, y el supervisor de cada empleado.
- se almacena, para cada empleado, información sobre sus familiares. Se mantiene para cada familiar: nombre, sexo, fecha nacimiento y parentesco con el empleado.

2.1.1. Entidades y atributos

- **Entidad (ocurrencia, instancia)**: Es un objeto específico, concepto o cosa del "mini mundo"
- **Atributos**: Propiedades que describen una entidad. Cada entidad específica (ocurrencia de la entidad) tiene un valor concreto para cada uno de los atributos. Los atributos pueden ser:
 - ♦ **simples**: cuando toma un valor indivisible. Ej: Dni
 - ♦ **compuestos**: si pueden descomponerse en subpartes más pequeñas con significados independientes, y que son referenciados por separado
Ej: Localidad (Calle, Población, Municipio, Provincia)
 - ♦ **univaluados**: si toman un único valor para cada entidad particular.
Ej: cada entidad Persona tiene una única Edad
 - ♦ **multivaluados**: una entidad puede tener una multiplicidad de valores para un atributo. Un atributo multivaluado puede tener límites inferior y superior del número de valores para una entidad individual.
Ej: un atributo Colores para un coche
 - ♦ **derivados**: atributos relacionados que pueden obtenerse a partir de otros.
Ej: Fecha de Nacimiento y Edad. Este último sería derivado

- **Conjuntos (tipos) de Entidades:** Agrupación de entidades con los mismos atributos. Especifica una estructura común compartida por las entidades pertenecientes al tipo. Su descripción se denomina **esquema de tipo de entidad**, conteniendo el nombre del conjunto, el nombre y significado de cada uno de sus atributos y la especificación de restricciones.

Se llama **extensión** de un conjunto de entidades al conjunto de ocurrencias que posee en un momento dado, e **intensión** a la descripción de la BD por sus atributos.

- **Atributo clave de un conjunto de entidades:** Es un atributo o conjunto de atributos de un conjunto de entidades para el que cada ocurrencia tiene un valor diferente. El concepto de atributo clave establece la **restricción de unicidad** sobre cualquier extensión de un conjunto de entidades, pues prohíbe que dos entidades tengan el mismo valor para el atributo clave a la vez.
- **Dominios de los atributos:** Cada atributo simple se asocia con un conjunto de valores (dominio), que especifica el conjunto de valores que puede tomar ese atributo para cada ocurrencia de entidad.

Ej: En un primer diseño conceptual la BD de la compañía quedaría:

DEPARTAMENTO (Nombre, Número, {Dirección}, Jefe, FechaComienzoDirecc)
PROYECTO (Nombre, Número, Dirección, DptoControl)
EMPLEADO (Nombre (Fnombre, Minic, Lnombre), SSN, Sexo, Dirección, Salario, FechaNac,
Departamento, Supervisor, {TrabajaEn (Proyecto, Horas)})
FAMILIARES (Empleado, NombreDpto, Sexo, FechaNac, Relación)

Se puede observar la existencia de relaciones implícitas entre los distintos conjuntos de entidades:

- el Jefe de Departamento es un empleado que dirige el departamento
- el DptoControl es el Departamento que controla el proyecto
- el Supervisor de Empleado se refiere a otro empleado, ...

En el modelo E/R, estas referencias no se representan como atributos sino como relaciones. Inicialmente se suelen representar algunas relaciones como atributos durante el diseño del esquema conceptual, que luego se convierten a relaciones.

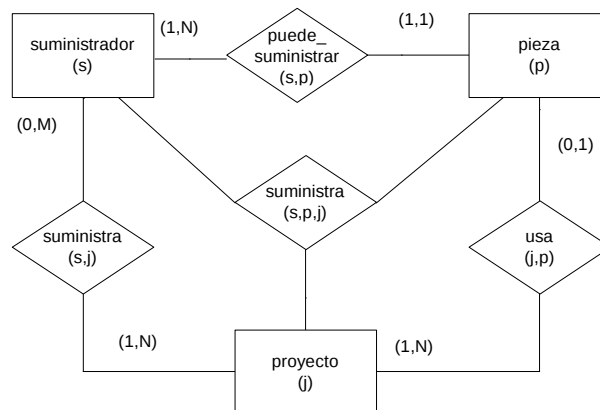
2.1.2. Relaciones

- **Relaciones y conjunto de relaciones:** Un conjunto de relaciones R entre n tipos de entidades $E_1, E_2, \dots$, En es un conjunto de asociaciones entre entidades de esos tipos. Cada instancia r_i de la relación R es una asociación de n entidades, donde la asociación incluye exactamente una entidad de cada uno de los conjuntos de entidades participantes. (Cada r_i indica que sus entidades participantes se relacionan de algún modo en el minimundo).

Ej: relación TRABAJA_PARA entre Empleado y Departamento, que asocia cada empleado con el departamento donde trabaja

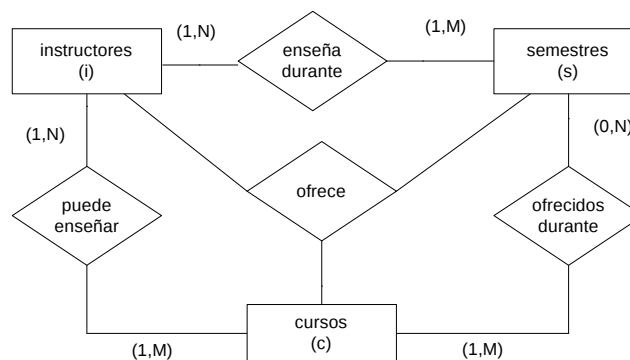
- **Grado de un conjunto de relaciones:** es el número de conjuntos de entidades participantes (binaria, ternaria, etc.).

En general, una relación ternaria representa más información que 3 relaciones binarias.



Ej: La existencia de tres relaciones (s,p), (j,p) y (s,j) no implica necesariamente que exista una instancia (s,p,j). Sin embargo, existiría siempre si las entidades tuviesen PARTICIPACION TOTAL.

A la hora de decidir entre tres relaciones binarias y una ternaria, el diseñador debe considerar la semántica concreta para la situación a representar:



OFRECE representa información de los instructores que ofrecen cursos durante ciertos semestres. Posee ocurrencias (i,s,c) cuando el instructor i ofrece el curso c durante el semestre s. PUEDE_ENSEÑAR relaciona cada curso con los instructores que pueden impartirlo. ENSEÑA_DURANTE relaciona cada semestre con los instructores que enseñan algún curso en el mismo. OFRECIDOS_DURANTE relaciona cada semestre con los cursos ofrecidos durante el mismo.

Las relaciones ternarias y binarias representan información diferente pero se relacionan entre sí.

Así, una instancia (i,s,c) de OFRECE no podrá existir si no existe una instancia (i,s) en ENSEÑA_DURANTE y una (s,c) en OFRECIDOS_DURANTE y una (i,c) en PUEDE_ENSEÑAR. Sin embargo, no tiene por qué darse lo contrario; es decir, el hecho de que existan instancias (i,s), (s,c) e (i,c) en las relaciones binarias no tiene por qué corresponderse a una instancia (i,s,c) en OFRECE.

- Se puede interpretar una relación como un atributo.

Ej: el hecho de que un Empleado trabaje en un Departamento puede verse como un atributo del mismo (donde, para cada ocurrencia de Empleado, Departamento representaría la ocurrencia Departamento donde trabaja), o bien usando un atributo multivaluado Empleados en el conjunto de entidades Departamento (cuyos valores para cada ocurrencia de Departamento sea el conjunto de ocurrencias Empleado que trabajan en ese departamento)

- Roles:** Cada conjunto de entidades que participa en un conjunto de relaciones juega un papel (rol) particular en la relación. El nombre de rol indica el papel que la ocurrencia participante juega en cada instancia de la relación.

Ej: en el conjunto de relaciones TRABAJA_PARA, Empleados juega el papel de trabajador y Departamento juega el papel de departamento.

Cuando el mismo conjunto de entidades participan más de una vez en una relación, el nombre de rol es esencial para distinguir el significado de cada participación. En este caso, la relación se dice **reflexiva**.

Ej: la relación Supervisión relaciona un empleado a un supervisor, donde tanto la entidad Empleado como la Supervisor son miembros del mismo conjunto de entidades Empleado. Empleado participa dos veces en Supervisión: una en el papel de supervisor y otra en el papel de supervisado.

- **Restricciones estructurales:** Los conjuntos de relaciones tienen ciertas restricciones que limitan las combinaciones de entidades. Estas restricciones vienen determinadas por la situación del minimundo que representa las relaciones. Las dos restricciones más habituales son:
 - ♦ **de cardinalidad**, que especifica el número de ocurrencias de los tipos de entidades que pueden participar en una relación (1:N, 1:1 ó M:N)
 - ♦ **de participación**, que indica si la existencia de una entidad depende en su existencia de otra mediante la relación. Puede ser **total** (se representa con una línea más ancha) o **parcial** (se representa con una línea simple).

Ej: si una compañía establece que cada empleado debe trabajar en un departamento, la entidad Empleado solo existirá si participa en una ocurrencia de la relación TRABAJA_PARA; es una participación total.

Ej: existe una relación DIRIGE entre Empleado y Departamento donde la entidad Empleado tiene una participación parcial porque solo un subconjunto de ocurrencias de Empleado dirige departamentos.

Como notación alternativa puede emplearse un par de números (**min,max**) que indican que para cada entidad e en E , e debe participar en al menos el **min** y como máximo el **max** de ocurrencias en R .

Ej: Empleado (0,1) dirige (1,1) Departamento significa: un Empleado dirige a lo sumo un Departamento; un Departamento tiene exactamente un Director

Ej: Empleado (1,1) trabaja (0,n) Departamento significa: un Empleado trabaja en 1, y solo 1, Departamento; en un Departamento trabaja ningún o cualquier número de empleados.

- **Atributos de los conjuntos de relaciones:** Los conjuntos de relaciones pueden tener atributos propios. Los atributos de los conjuntos de relaciones 1:1 ó 1:N se pueden incluir como atributos de uno de los conjuntos de entidades participantes.

Ej: un atributo FechaComienzo de la relación DIRIGE puede ser atributo de Empleado o Departamento (aunque conceptualmente pertenece a DIRIGE), ya que es una relación 1:1 donde cada ocurrencia de Empleado o de Departamento participa en una ocurrencia de la relación. El valor de FechaComienzo se puede determinar de modo separado o por la ocurrencia de Departamento o del Empleado.

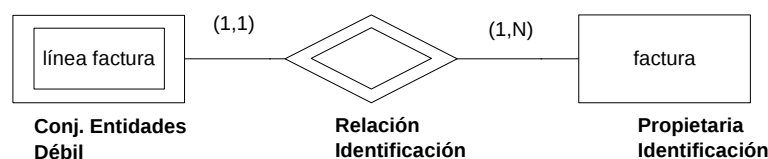
Para un conjunto de relaciones 1:N, el atributo solo se puede incluir como un atributo del conjunto de entidades del lado N de la relación

Ej: para TRABAJA_PARA, el atributo FechaComienzo solo se puede incluir en Empleado.

Si el valor de un atributo es combinación de las ocurrencias de entidades participantes en la relación, el atributo debe especificarse como de la relación. Es el caso de las relaciones M:N

Ej: Horas de TRABAJA_EN: el número de horas de un empleado que trabaja en un proyecto se determina por la combinación Empleado-Proyecto y no separadamente.

- **Conjuntos de entidades débiles:** Son conjuntos de entidades que no tienen claves propias. Sus ocurrencias se identifican relacionándolas con ocurrencias específicas de otros conjuntos de entidades, combinándolas con valores de algunos de sus atributos. A este otro conjunto de entidades se llama **propietaria de identificación**, y al conjunto de relaciones entre ellas **relación de identificación**.



Un conjunto de entidades débil tiene siempre una restricción de participación TOTAL (dependencia de existencia) porque no se puede identificar una entidad débil sin su entidad fuerte. Sin embargo, **no toda participación total da lugar a un conjunto de entidades débil.**

Un conjunto de entidades débil tiene una clave parcial, que es el conjunto de atributos que pueden identificar cada una de las entidades débiles que se relacionan con una fuerte.

Ej: el atributo NombreFamiliar es una clave parcial.

Los conjuntos de entidades débiles se pueden representar a veces como atributos compuestos o multivaluados.

Ej: se puede especificar FAMILIARES como atributo multivaluado compuesto para Empleados, cuyos atributos simples sean NombreFamiliar, FechaNacimiento, Sexo y Relación. Este atributo puede sustituir al conjunto de entidades débil FAMILIARES.

Elegiremos una u otra opción dependiendo de si el conjunto de entidades débil participa independientemente en conjuntos de relaciones o no.

2.1.3. Agregación

Una limitación del modelo E-R es que no permite expresar relaciones entre relaciones. Para ver la necesidad de tales construcciones considérese la relación ternaria *trabaja_en* entre *empleado*, *sucursal* y *trabajo* que se muestra en la Figura 1.

Para registrar los directores de las tareas realizadas por un empleado en una sucursal, una alternativa es crear una relación cuaternaria *dirige* entre *empleado*, *sucursal*, *trabajo* y *director*, tal como se muestra en la figura anterior. En la figura resultante hay información redundante, ya que cada combinación *empleado*, *sucursal*, *trabajo* en *dirige* también lo está en *trabaja_en*.

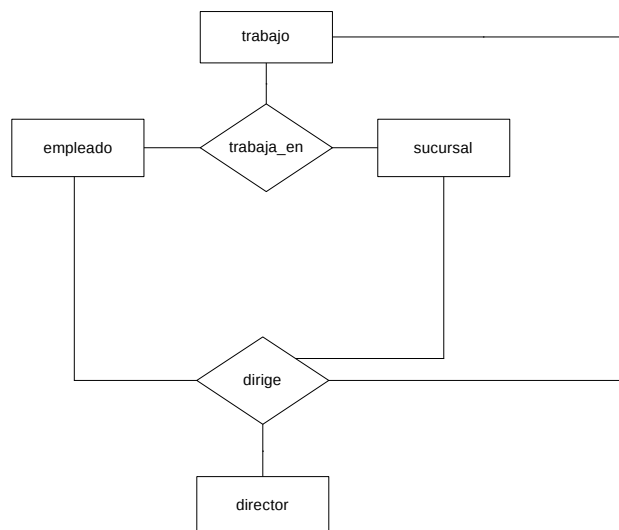


Figura 1.- Ejemplo de Relación Cuaternaria

La mejor forma de modelar esta situación es usar la agregación. La **agregación** es una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto. Así, para este ejemplo, se considera el conjunto de relaciones *trabaja_en* como un conjunto de entidades de nivel más alto denominada *TRABAJA*, y se trata de la misma forma que cualquier otro conjunto de entidades. Se puede crear entonces una relación binaria *dirige* entre *TRABAJA* y *director*. En la Figura 2 se muestra la notación empleada habitualmente para esta situación.

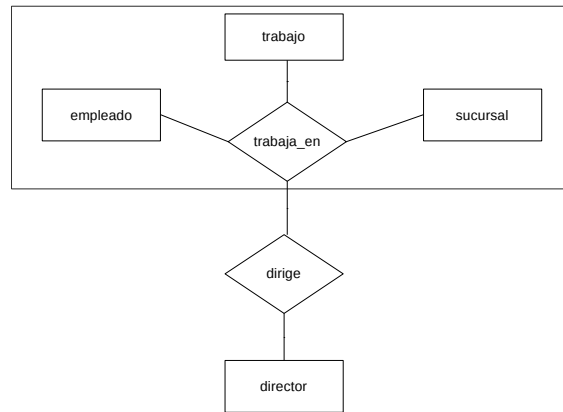
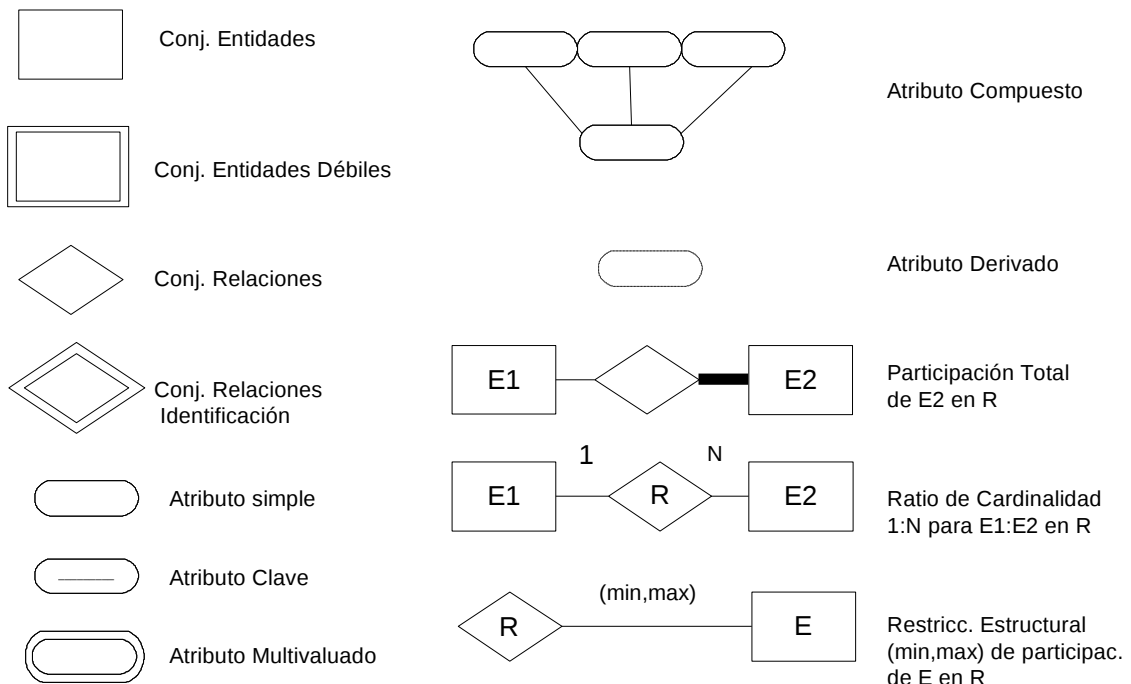
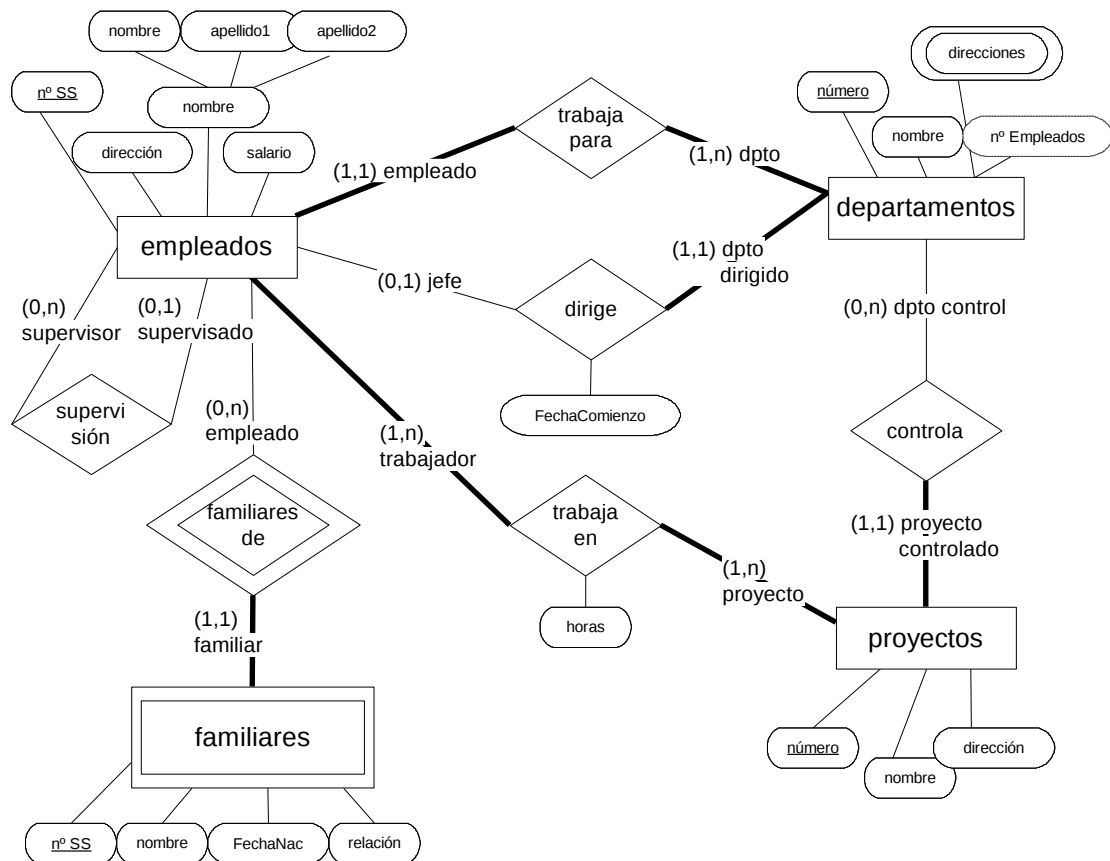


Figura 2.- Ejemplo de Agregación

2.1.4. Notación



Ej: Atendiendo a las consideraciones vistas, la base de datos Compañía se puede representar mediante el modelo E-R así:



2.1.5. Algoritmo de transformación entre ER y relacional

1. Tipos de entidad **regulares** se transforman en relaciones, eligiendo una de las claves candidatas como clave primaria
2. Tipos de entidad **débiles** se transforman en relaciones. Su clave primaria se obtiene como combinación de la clave parcial y la clave del propietario
3. Las **relaciones binarias 1:1** se resuelven introduciendo una clave foránea en una de las entidades relacionadas. En el caso de que la relación tenga atributos propios es preferible la creación de una tabla de relación.
4. Las **relaciones binarias 1:N** se resuelven introduciendo la clave foránea en el lado N de la relación. En el caso de existir atributos de la relación o de que la participación sea parcial suele ser preferible una relación adicional, conjugando las claves de las entidades participantes e incorporando los atributos de la relación.
5. Las **relaciones binarias M:N** generan siempre una relación adicional, cuya clave es la combinación de las de las entidades participantes.
6. Los **atributos multivaluados** generan relaciones, cuya clave se compone de la de la entidad y del valor del atributo.
7. Las **relaciones n-arias** derivan en relaciones.
8. Consideraciones adicionales:

Los **atributos compuestos** se representan mediante los valores de sus componentes simples.

La materialización de las relaciones desde las tablas diseñadas se realiza mediante un equijoin o un join simple:

- Join simple para 1:1 o 1:N
- 2 joins para M:N (o si las relaciones 1:N o 1:1 se resolvieron con tabla adicional)
- n joins para relaciones n-arias

2.2. Modelo Entidad Relación Extendido (EER)

[EN02, SKS02]

Surge debido a la insuficiencia del modelo Entidad-Relación para aplicaciones complejas. Incorpora conceptos de modelización semántica al modelo E/R tales como: Subclase y Superclase, Especialización/Generalización y Categorización.

2.2.1. Subclases y Superclases

Una **subclase** es un subconjunto de ocurrencias de un conjunto de entidades del mismo tipo (**superclase**) que necesitan ser representadas por separado por necesidades de la aplicación. Todas estas ocurrencias tendrán alguna característica común:

Ej: EMPLEADO $\Rightarrow$ Secretarios, Técnicos, Asalariados, ...

Los miembros de las subclases son las mismas entidades que las de la superclase pero bajo un rol específico. No podrá existir una ocurrencia en ninguna subclase que no sea miembro de la superclase, pero puede haber ocurrencias en la superclase que no pertenezcan a ninguna subclase.

La relación entre una subclase y la superclase correspondiente se dice que es de tipo IS_A (es_un).

Ej: Secretario es_un Empleado

2.2.1.1. Herencia

- a) **de atributos:** Los miembros de las subclases heredan los atributos de las superclases a las que pertenecen.

Ej: dada una especialización *Empleado* $\Rightarrow$ *Secretario*, *Técnico*, *Ingeniero*, ..., la *FechaNacimiento* de una *Secretario* es la del *Empleado* que la representa.

Además, las subclases pueden disponer de atributos propios.

Ej: *Velocidad* como atributo de *Secretario*

- b) **de relaciones:** una entidad miembro hereda todas las relaciones establecidas para su superclase. Las subclases, además, pueden establecer sus propias relaciones.

Ej: si *Empleado* tiene una relación *TrabajaEn* se aplica también a *Secretario*. Sin embargo, *Jefe* puede tener una relación *Dirige* no existente para el resto de empleados.

2.2.2. Especialización

Es el proceso de definición de un conjunto de subclases a partir de un conjunto de entidades (la superclase) basándose en alguna característica que distingue las entidades de la superclase.

Ej: {*Secretario*, *Técnico*, *Ingeniero*, *Jefe*} es una especialización de la superclase *Empleado* basándose en el *TipoTrabajo* de cada ocurrencia.

Pueden existir distintas especializaciones del mismo conjunto de entidades basándose en diferentes características.

Ej: otra especialización de *Empleados* podría ser {*Asalariados*, *PorHoras*} atendiendo al *modo de pago*.

Razones del uso de subclases en la modelización de datos

- a) ciertos atributos solo se pueden aplicar a algunas ocurrencias del conjunto de entidades superclase. Estos formarán la subclase.

Ej: dada una especialización *Empleado* $\Rightarrow$ *Secretario*, *Técnico*, *Ingeniero*, ..., *Pulsaciones_Minuto* como atributo de *Secretario*

- b) en ocasiones solo un subconjunto de las ocurrencias de la superclase participan en ciertas relaciones.

Ej: dada una especialización *Empleado* $\Rightarrow$ *Asalariado*, *Por Horas*, ..., solo las ocurrencias de *Asalariado* se relacionan con *Sindicato*.

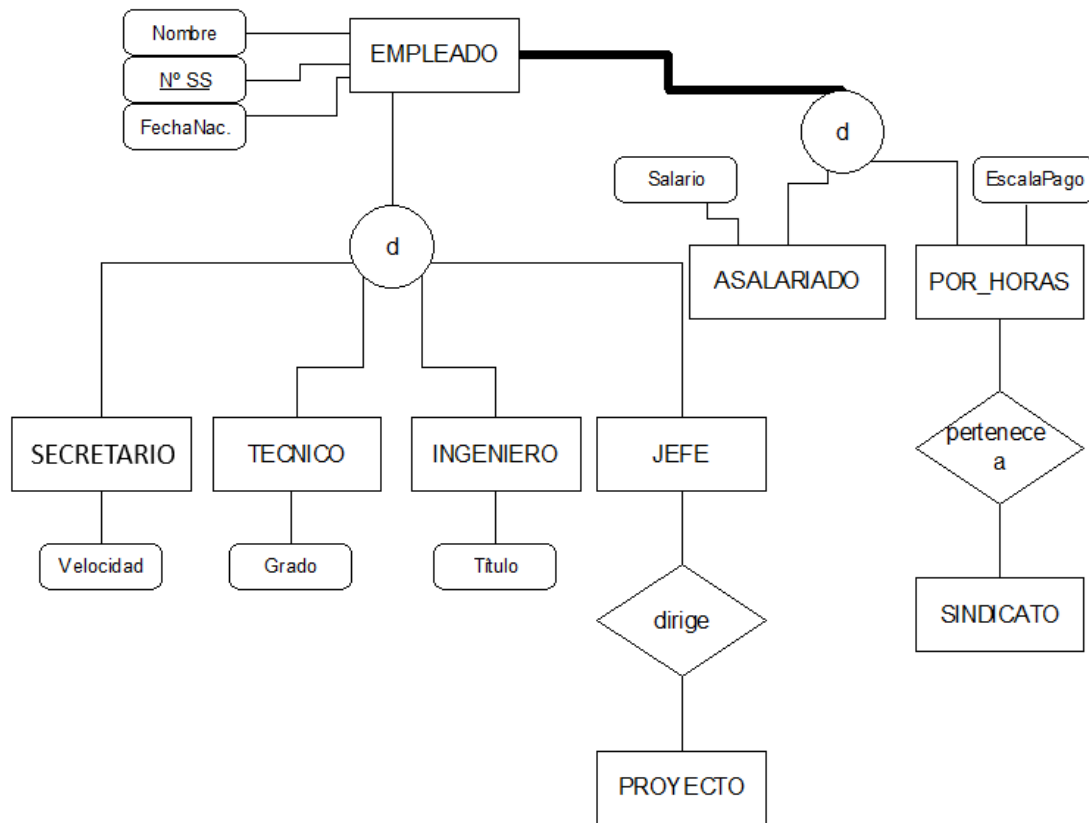


Figura 3.- Ejemplo de especializaciones / generalizaciones

2.2.3. Generalización

Es el proceso de abstracción inverso a la especialización. Se basa en suprimir las diferencias entre algunos conjuntos de entidades, identificar sus atributos (características) comunes y generalizarlos obteniendo un único conjunto de entidades superclase.

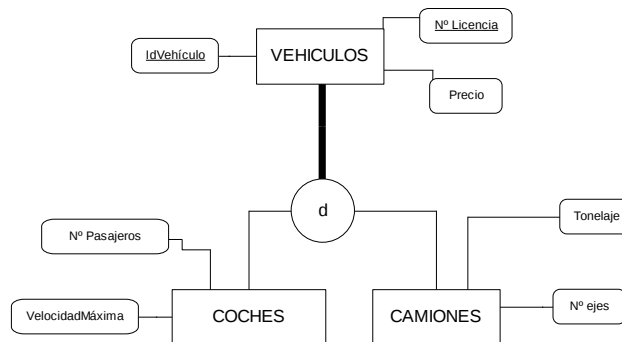


Figura 4.- Ejemplo de generalización

En este caso, la especialización es exhaustiva (no existen Vehículos que no sean Camiones o Coches) y, además, disjunta (un mismo Vehículo no puede ser de ambas clases). La especialización exhaustiva se representa con doble línea.

2.2.4. Restricciones de la Especialización y la Generalización

- a) de **disyunción**: especifica que las subclases deben ser disjuntas, es decir, una ocurrencia de la superclase podrá estar como mucho en una de las subclases. Se especifica mediante la letra **d**. La ausencia de restricción de disyunción sería el solapamiento, especificado mediante la letra **o** (por defecto).

- b) de **participación total** o **parcial**: La participación total (restricción de unión) especifica la restricción de que cada entidad en la superclase debe ser miembro de alguna subclase en la especialización. Se representa mediante una doble línea.

Ej: {*Asalariados*, *Por_Horas*} es un ejemplo de participación total.

La participación parcial se representa con una línea y permite que ocurrencias de la superclase no pertenezcan a ninguna subclase.

Ej: {*Secretario*, *Técnico*, *Ingeniero*} es un ejemplo de participación parcial.

La restricción de participación en el proceso de generalización es total ya que la superclase se deriva de las subclases por lo que solo contiene las ocurrencias de las subclases.

- c) restricciones **implícitas**: En ocasiones podemos determinar exactamente las ocurrencias que van a ser miembros de cada subclase comparando el valor de algún atributo de la superclase con una condición. En este caso, decimos que se trata de una **especialización definida mediante atributo**.

El atributo en cuestión se denomina **atributo de definición**, y la condición utilizada se denomina **predicado de definición**.

Ej: si *Empleados* tiene un **atributo de definición** *TipoTrabajo*, podemos especificar la condición de miembro de la subclase *Secretario* mediante el **predicado de definición**: *TipoTrabajo*="Secretario", la condición de miembro de la subclase *Técnico* mediante el **predicado de definición**: *TipoTrabajo*="Técnico Industrial", y así sucesivamente.

- d) restricciones **definidas por el usuario**: En este caso, los conjuntos de entidades de nivel más bajo no están restringidos mediante una condición de miembro; en cambio, las entidades se asignan a un conjunto de entidades dado por el usuario de la base de datos.

Ej: supongamos que los *Empleados* se asignan a diferentes grupos de trabajo. Un empleado dado no se asigna a un grupo automáticamente en términos de una condición que lo defina explícitamente. En su lugar, la asignación se hace de forma individual por el usuario a cargo de la decisión.

2.2.4.1. Reglas de inserción y borrado para Especialización y Generalización

1. El borrado de una entidad de una superclase implica que es borrada automáticamente de todas las subclases a las que pertenece
2. La inserción de una entidad en una superclase implica que se inserte obligatoriamente en todas las subclases definidas mediante atributo donde la entidad satisfaga el predicado de definición
3. La inserción de una entidad en una superclase de una especialización total implica que la entidad se inserte en al menos una de sus subclases.

2.2.5. Categorización

Una **categoría** es una subclase obtenida de la modelización de una UNICA relación ES-UN con más de una superclase, donde las superclases representan diferentes conjuntos de entidades.

Ej: En una BD que registre vehículos (considerando como tal únicamente coches y camiones), un propietario puede ser una persona, un banco o una compañía. Se define una categoría *PROPIETARIOS* como una subclase de la UNION de *PERSONAS*, *BANCOS* y *COMPAÑÍAS*. *VEHÍCULOS REGISTRADOS* es una subclase de la UNION de *COCHES* y *CAMIONES*.

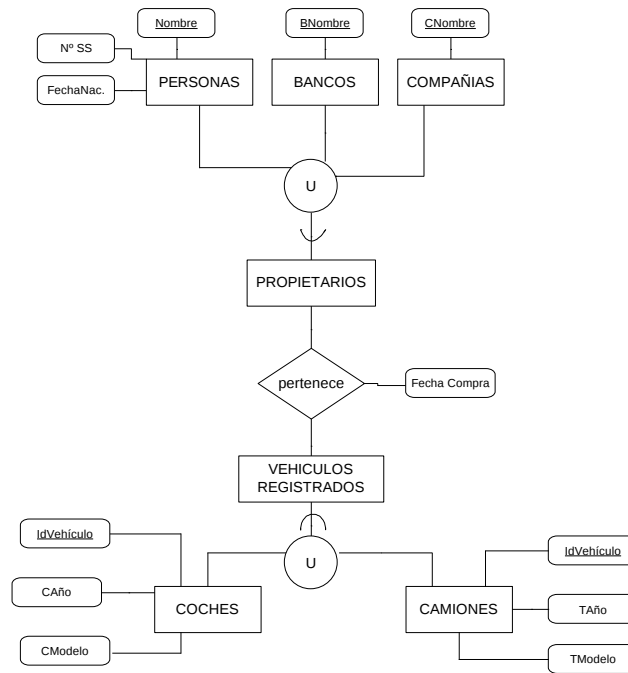


Figura 5.- Ejemplo de categorizaciones parciales

Existen claras diferencias con los elementos definidos anteriormente:

- respecto a **especialización**: una categoría tiene 2 o más superclases que representan distintos conjuntos de entidades, mientras que en las relaciones ES_UN siempre tienen una única superclase.
- respecto a **subclase compartida**: cada ocurrencia de la subclase debe existir en TODAS las superclases, es decir, la subclase es un SUBCONJUNTO DE LA INTERSECCIÓN de las superclases (Figura 6). Una categoría, sin embargo, es un SUBCONJUNTO DE LA UNION de superclases: cada entidad que es miembro de la categoría debe existir al menos una de las superclases pero no tiene por qué serlo de todas ellas (Figura 5). En la mayoría de los casos, una entidad de la categoría es miembro de exactamente una de las superclases.

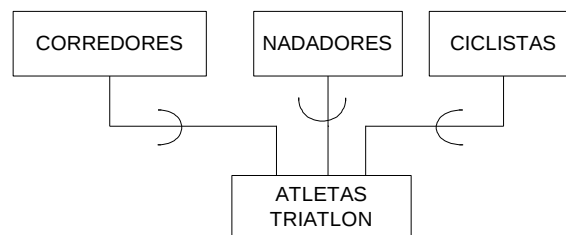


Figura 6.- Subclase compartida

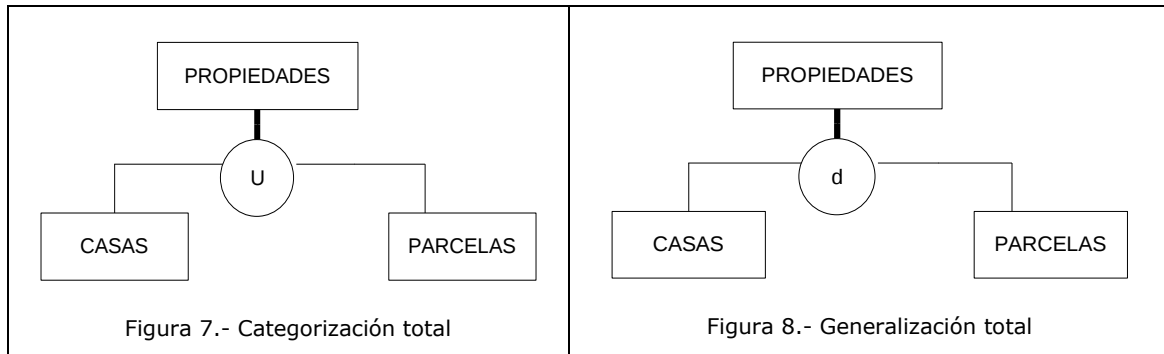
- También existe diferencia en la herencia de atributos. En el caso de las categorías es una **herencia selectiva** donde cada entidad hereda los atributos de la superclase a la que pertenece. Las subclases compartidas, en cambio, heredan TODOS los atributos de sus superclases.
- respecto a **generalización/especialización total**: Para los casos de categorización (Figura 5) y generalización (Figura 4) todos los COCHES y CAMIONES son VEHÍCULOS, mientras que VEHÍCULOS REGISTRADOS incluye solo algunos coches y algunos camiones, pero no necesariamente todos (puede hacer coches y camiones no registrados, pero sí pertenecientes a COCHES y CAMIONES)
- respecto a **generalización/especialización parcial**: Para los casos de categorización (Figura 5) y generalización (Figura 4), VEHÍCULOS puede contener otro tipo de entidades

que no sean COCHES o CAMIONES (por ejemplo, MOTOCICLETAS) mientras que VEHÍCULOS REGISTRADOS solo permite como miembros a COCHES y CAMIONES.

Una categoría puede ser **parcial** (Figura 5) o **total** (Figura 7). En las categorizaciones parciales puede haber elementos en las superclases que no son miembros de la categoría; en la total todo elemento de las superclases aparece en la categoría.

La categorización total y la especialización total son equivalentes excepto en que en la especialización las subclases tienen claves idénticas que coinciden con la de la superclase, mientras que en la categorización las superclases pueden tener claves distintas. En caso de que las superclases tengan la misma clave, son equivalentes.

Ej: para los casos de categorización total de la Figura 7 y generalización (o especialización) total de la Figura 8.



2.2.6. Conversión del modelo EER al modelo relacional

2.2.6.1. Relaciones Clase/Subclase y Especialización/Generalización

Considerando $At(R)$ = atributos de una relación R

$Pk(R)$ = clave primaria de una relación R

m subclases $\{S_1, \dots, S_m\}$ y una superclase C

$\{k, a_1, \dots, a_n\}$ los atributos de la superclase C , siendo k la clave

hay varias opciones para resolver las relaciones de subclases y especialización:

- a) **Crear una relación L_i para cada subclase S_i** , que incluya los atributos específicos de S_i y la clave primaria de la superclase C , **y una relación L para la superclase** donde se incluyan sus atributos. Es decir:

$$\begin{aligned} 1 \text{ relación } L \text{ para } C & \Rightarrow At(L) = \{k, a_1, \dots, a_n\} & Pk(L) = k \\ m \text{ relaciones } L_i \text{ para las subclases} & \Rightarrow At(L_i) = \{k, At(S_i)\} & Pk(L_i) = k \end{aligned}$$

Ej: para la Figura 3, quedaría:

Empleados = $\{N^oSS, Nombre, FechaNac\}$

Secretario = $\{N^oSS, Velocidad\}$, Técnicos = $\{N^oSS, Grado\}$, Ingeniero = $\{N^oSS, Título\}$

Mediante un join en la clave primaria entre un L_i y L se pueden obtener todos los atributos de la subclase S_i .

Se utiliza en cualquier caso de especialización / generalización.

- b) **Construir el esquema con el resultado del join anterior.** Es decir:

$$\begin{aligned} 1 \text{ relación } L_i \text{ para cada subclase} & \Rightarrow \\ At(L_i) = \{k, a_1, \dots, a_n\} \cup \{At(S_i)\} & Pk(L_i) = k \end{aligned}$$

Ej: para la Figura 3, quedaría:

Secretario = $\{N^oSS, Nombre, FechaNac, Velocidad\}$, Técnicos = $\{N^oSS, Nombre, FechaNac, Grado\}$,

Ingeniero = $\{N^oSS, Nombre, FechaNac, Título\}$, Jefe = $\{N^oSS, Nombre, FechaNac\}$

Para obtener las entidades de la superclase C se realiza una operación de **UNION EXTERIOR** (unión del esquema de las tablas), quedándose únicamente con los atributos comunes.¹

Se utiliza en caso de especialización disjunta total (pues si no es total se pierden las entidades que no son de ninguna de las subclases, y si no es disjunta entonces una entidad que pertenezca a 2 subclases tendrá almacenados los atributos heredados de modo redundante en más de un Li).

El inconveniente que tiene es que para encontrar una entidad arbitraria de C tendremos que buscar en todas las relaciones Li.

c) **Crear 1 única relación L =>**

$$At(Li) = \{k, a_1, \dots, a_n\} \cup \{At(S_1)\} \cup \dots \cup \{At(S_i)\} \cup \{t\} \quad Pk(Li)=k$$

donde **t** es un atributo que indica la subclase a la que pertenece la tupla.

Ej: para la Figura 3, quedaría:

Empleado = {NºSS, Nombre, FechaNac, Velocidad, Grado, Título, t}

Se utiliza en caso de especialización disjunta. No permite especializaciones solapadas.

d) Cuando las subclases son definidas **por atributo**, no es necesario el atributo **t** para el caso anterior pues el valor del atributo de definición ya indica la subclase a la que pertenece cada tupla.

e) **Crear 1 única relación L =>**

$$At(L) = \{k, a_1, \dots, a_n\} \cup \{At(S_1)\} \dots \cup \{At(S_i)\} \cup \{t_1, \dots, t_i\} \quad Pk(L)=k.$$

donde **tp** estará a 1 si la tupla pertenece a la subclase **p**.

Ej: para la Figura 3, quedaría:

Empleado = {NºSS, Nombre, FechaNac, Velocidad, Grado, Título, t1, t2, t3, t4}

Se utiliza en caso de especialización solapada. Genera un gran número de valores nulos.

La elección entre las opciones a) y b) depende del uso de la superclase y de las subclases (si forman parte de otras relaciones o no), del número de atributos diferentes de las subclases y del número de tuplas que contienen.

Las opciones c), d) y e) crean una única relación para representar la superclase y sus subclases. Cualquier entidad que no pertenezca a alguna de las subclases tendrá valores nulos para los atributos específicos de esas subclases.

Estas opciones no son recomendables si las subclases tienen muchos atributos específicos. En caso contrario, son preferibles a las opciones anteriores porque evitan la necesidad de hacer las operaciones de join o de unión dando lugar a una implementación más eficiente.

2.2.6.2. Subclases compartidas

Se puede utilizar cualquiera de las alternativas vistas, pero generalmente se emplea el caso a). Utilizando las opciones c), d) o e) hay información redundante.

2.2.6.3. Categorías

Hay dos casos:

- a) si las claves de las superclases NO están definidas bajo el mismo dominio => se especifica una nueva clave llamada **clave subrogada** pues no se puede usar ninguna de las superclases exclusivamente para identificar todas las entidades de la categoría.

¹ La unión exterior se creó para efectuar la unión de tuplas de dos relaciones que son compatibles *parcialmente*. Las tuplas con los mismos valores en los atributos comunes se unen y se representan una sola vez en el resultado. Los atributos de cada relación que no son compatibles con la unión se mantienen en el resultado, y las tuplas que no tienen valores para dichos atributos se rellenan con valores nulos. La unión exterior es equivalente al *join externo* si los atributos de join son *todos* los atributos comunes a ambas relaciones.

La clave subrogada se añade a las superclases (como clave foránea) para poder especificar la correspondencia de valores entre la clave subrogada y la clave de cada superclase, y se usa como clave en la categoría.

- b) si las superclases tienen la misma clave => se puede utilizar la clave de las superclases como clave para la categoría.

En cualquier caso, se crea una tabla para cada superclase y otra para la categoría. La herencia de atributos se consigue mediante un EQUIJOIN.

Ej: para la Figura 5, quedaría:

Personas = {NºSS, NºCarnet, Nombre, Dirección, **IdProp**}

Bancos = {BNombre, BDirección, **IdProp**}

Compañías = {CNombre, CDirección, **IdProp**}

Propietarios = {**IdProp**}

Vehículos Registrados = {**IdVehículo**, NªLicencia}

Coches = {**IdVehículo**, CEstilo, CModelo, CAño}

Camiones = {**IdVehículo**, TModelo, Tonelaje, TAño}

Pertenece = {**IdProp**, **IdVehículo**, FechaCompra}

BIBLIOGRAFÍA:

NOMBRE	AUTOR	AÑO	EDITORIAL	APUNTES
Fundamentos de Sistemas de Bases de datos. (3ª edic) cap. 3, 4	Ramez Elmasri Shamkant Navathe	2002	Addison-Wesley Iberoamericana	EN02
Database Systems. A practical Approach to Design, Implementation and Management (Second edition) cap 5	Thomas Connolly Carolyn Begg	1998	Addison Wesley	CBS98
Database Management System (Second edition) cap. 2	R. Ramakrishnan	1998	Mc Graw Hill	Rama98
Fundamentos de bases de datos (4ª edición) cap. 2	A. Silberschatz H.F. Korth S. Sudarshan	2002	Mc Graw Hill	SKS02
Fundamentos y modelos de bases de datos (2ª edición)	A. de Miguel M. Piattini	1999	Ra-ma	DeMP99