

Diseño conceptual: Modelo entidad-relación

Bases de Datos

Curso 2018-2019

Jesús Correas – jcorreas@ucm.es

**Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid**

Bibliografía

- Bibliografía básica:

- ▶ R. Elmasri, S.B. Navathe. **Fundamentals of Database Systems** (6a Ed). Addison-Wesley, 2010. (en español: **Fundamentos de Sistemas de Bases de Datos** (5a Ed). Addison-Wesley, 2007).
Capítulos 3 y 4 (5ª ed. en español) o 7 y 8 (6ª ed.).

- Bibliografía complementaria:

- ▶ A. Silberschatz , H. F. Korth, S. Sudarshan. **Fundamentos de bases de datos** (5a Ed), McGraw-Hill, 2006.
Capítulo 6. (en especial el modelo EER: Sección 6.7).

Contenido

- Introducción. El modelo entidad-relación.
- Entidades y atributos.
 - ▶ Superclaves y claves candidatas. Entidades débiles.
 - ▶ Tipo de entidad.
- Diagramas ER.
- Relaciones.
 - ▶ Atributos de relación.
 - ▶ Tipo de relación.
 - ▶ Grado de una relación.
 - ▶ Relaciones recursivas.
 - ▶ Restricciones: cardinalidad y participación.
 - ▶ Restricciones y entidades débiles.
- Diagramas ER extendidos.
 - ▶ Especialización y generalización. Herencia de atributos.
 - ▶ Agregaciones.

Introducción. El modelo entidad-relación.

- Este modelo nos permite representar la información de una BD en el **nivel conceptual**.
- Se obtiene a partir del análisis inicial del sistema, **pero este proceso no es directo**.
 - ▶ Un buen diseño requiere estudiar las necesidades del sistema y la información de análisis.
- El diseño de una BD debe ser **conciso, fácil de comprender y mantener**, y debe permitir un diseño lógico y físico **eficiente**.
- Se pueden obtener varios diseños *candidatos* para un mismo análisis.
- Los elementos fundamentales del modelo ER son:
 - ▶ Entidades, tipos de entidad.
 - ▶ Atributos.
 - ▶ Relaciones, tipos de relación y restricciones sobre las relaciones.
- El modelo ER se representa mediante los **diagramas ER**.

Introducción. El modelo entidad-relación.

- Los pasos a seguir para diseñar el modelo ER son:
 - ▶ Elegir los tipos de entidad y atributos.
 - ▶ Elegir los tipos de relación.
 - ▶ Definir las restricciones.
- Vamos a introducir los elementos del modelo con un **ejemplo**:
 - ▶ Debemos diseñar una base de datos para gestionar una empresa de preparación de platos preparados.

Introducción. Ejemplo introductorio.

- Algunas de las especialidades de nuestra empresa:



- Algunos requerimientos del sistema son los siguientes:

Introducción. Ejemplo introductorio.

Requerimientos:

- ▶ En la empresa trabajan **empleados** de los que se necesita saber su nombre, número de la SS, puesto que ocupan (cocinero, pinche, motorista...), teléfonos (fijo y móviles) y edad.
- ▶ La empresa tiene varios **establecimientos**.
- ▶ Existe una relación jerárquica entre los empleados.
- ▶ La empresa elabora diferentes **platos**, que tienen un código, una descripción (*maki roll*, *nigiri&sashimi*, etc.) y un precio.
- ▶ Cada plato está compuesto por **ingredientes** en cierta cantidad.
- ▶ Cada empleado debe ser capaz de preparar entre uno y 8 platos.
- ▶ Debemos saber los platos que sabe preparar cada empleado.
- ▶ Necesitamos la información de los **clientes** (teléfono, dirección, email, pedidos que han realizado).
- ▶ Cada **pedido** se identifica por un número, y debemos mantener la siguiente información: fecha del pedido, platos solicitados, cantidad
- ▶ Se debe conocer el **proveedor** que proporciona cada ingrediente en cada establecimiento, etc.




Entidades



- Son los elementos básicos del modelo ER.
- Representan **una “cosa” del mundo real**, distinguible de todas las demás.
- Puede ser un objeto físico (un vehículo, una persona) o conceptual (un curso, un préstamo).
- Una entidad determinada se define mediante un **conjunto de atributos**: propiedades que lo describen.
- Un **tipo de entidad** define un **conjunto de entidades que comparten los mismos atributos**.
- Cada entidad tiene sus propios valores para cada atributo.
- **Ejemplo**: Una entidad “empleado” con los siguientes atributos:

NSS:	1234567890
Nombre:	Andrés Sánchez García
Puesto:	Pinche
Edad:	25
Teléfono:	600123456

Atributos

- Cada atributo tiene asociado un **dominio**: conjunto de valores que puede tomar. 
- Cada entidad de un tipo de entidad tiene asociado un valor a cada atributo.
- Los atributos pueden ser de distintos tipos:
 - ▶ **simples** (atómicos) o **compuestos** (se pueden descomponer en atributos más pequeños, ej. nombre completo es el nombre y apellidos).
 - ▶ **monovalorados** o **multivalorados** (pueden tomar varios valores para una misma entidad, ej. color de un coche con varios tonos).
 - ▶ **primitivos** o **derivados** (se pueden calcular a partir de otros atributos o entidades, ej. fecha de nacimiento y edad).
- Si una entidad no tiene un valor para un atributo, toma el valor **nulo** (ej., si un empleado no tiene teléfono).

Superclaves, claves candidatas, clave primaria

- Las entidades de un tipo de entidad deben ser **distinguibles** de las demás entidades del mismo tipo.
 - ▶ Como las entidades están definidas por los valores de sus atributos, algunos de estos atributos deben poder ser utilizados para identificar una entidad particular en el conjunto de entidades de ese tipo.
- Una **superclave** es un subconjunto de los atributos de un tipo de entidad **que identifican unívocamente a cada entidad** de ese tipo. 
 - ▶ Cualquier superconjunto de una superclave también es superclave.
 - ▶ Interesa encontrar un conjunto de atributos lo mas pequeño posible que nos permita identificar unívocamente cada entidad.
- Una **clave candidata** es una superclave que no contiene ningún subconjunto que sea superclave. 
- La **clave primaria** de una entidad es una clave candidata seleccionada por el diseñador.

Superclaves, claves candidatas, clave primaria


- **Ejemplo:** Entidad empleado:

- ▶ {NIF}, {NSegSocial}, {Teléfono} (móvil), {NIF, Nombre}, {NSegSocial, Nombre, Edad},...
- ▶ No son superclaves: {Edad}, {Nombre, Edad}...
- ▶ Claves candidatas: {NIF}, {NSegSocial}, {Teléfono}

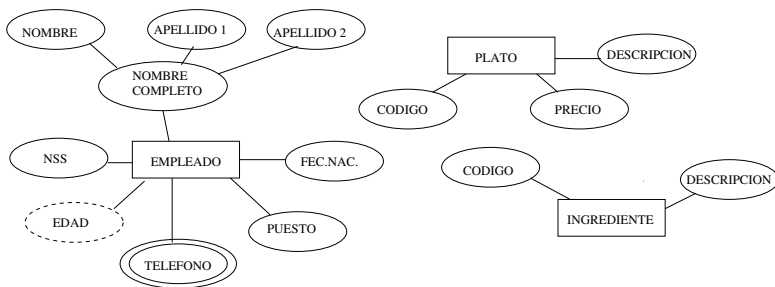
- Normalmente cada entidad tiene **un atributo que es clave**, pero es posible que la clave esté formada por más de un atributo.
- Un tipo de entidad es **débil** si no hay ningún conjunto de atributos que sea superclave. Estudiaremos este caso más adelante.



Diagramas ER

- Representación gráfica de la estructura de la BD en un nivel conceptual.
- Elementos básicos:
 - ▶ **Tipos de entidades:** Se representan mediante una caja.
 - ★ Para las entidades **débiles** se utiliza una **caja doble**.
 - ▶ **atributos:** Se representan mediante una **elipse** conectada con una línea a su entidad.
 - ★ Los atributos de la **clave**  recen **subrayados**.
 - ★ En los atributos **multivalorados** se utiliza una **elipse doble**.
 - ★ Cada componente de un atributo compuesto se representa como atributo suyo (conectado mediante una línea).
 - ★ Los atributos **derivados** se representan mediante una **elipse discontinua**.
 - ▶ **Tipos de relaciones:** Se representan mediante un **rombo** unido con líneas a las entidades asociadas.

Ejemplo de diagrama ER: entidades y atributos



EMPLEADO:

{⟨0101, (Juan, García, Pérez), 21, {666111222, 911234567}, Pinche, 01.01.95⟩,
⟨0202, (José, Sanz, Sanz), 26, {666123231}, Metre, 01/01/90⟩, ...}

PLATO: {⟨PL001, California maki, 8.50€⟩, ⟨PL002, Nigiri, 3.00€⟩,
...}

INGREDIENTE: {⟨IN001, Salmon⟩, ⟨IN002, Arroz⟩, ⟨IN003, Wasabi⟩, ...}

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes.**

¿Cómo se puede resolver utilizando elementos básicos de ER?

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes.**
 - ¿Cómo se puede resolver utilizando elementos básicos de ER?
- ¿Haciendo que **ingrediente** sea atributo de **plato**?

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
`<PL001, California maki, 8.50€, IN001, salmon >`
`<PL001, California maki, 8.50€, IN002, Rice >`

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
⟨PL001, California maki, 8.50€, **IN001, salmon** ⟩
⟨PL001, California maki, 8.50€, **IN002, Rice** ⟩ → **redundancia**.

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
`<PL001, California maki, 8.50€, IN001, salmon >`
`<PL001, California maki, 8.50€, IN002, Rice >` → **redundancia**.
 - (2) Hacer el atributo ingrediente **multivalorado**:
`<PL001, California maki, 8.50€, {<IN001, Salmon>, <IN002, E`
`<PL002, Nigiri, 3.00€, {<IN003, Wasabi>, <IN002, Arroz>, ...} >`



Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
`<PL001, California maki, 8.50€, IN001, salmon >`
`<PL001, California maki, 8.50€, IN002, Rice >` → **redundancia**.
 - (2) Hacer el atributo ingrediente **multivalorado**:
`<PL001, California maki, 8.50€, {<IN001, Salmon>, <IN002, Egg>} >`
`<PL002, Nigiri, 3.00€, {<IN003, Wasabi>, <IN002, Arroz>, ...} >`
★ Cada ingrediente tiene atributos → **redundancia**.

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
`<PL001, California maki, 8.50€, IN001, salmon >`
`<PL001, California maki, 8.50€, IN002, Rice >` → redundancia.
 - (2) Hacer el atributo ingrediente **multivalorado**:
`<PL001, California maki, 8.50€, {<IN001, Salmon>, <IN002, Egg>} >`
`<PL002, Nigiri, 3.00€, {<IN003, Wasabi>, <IN002, Arroz>, ...} >`
 - ★ Cada ingrediente tiene atributos → redundancia.
 - ★ El mismo ingrediente puede estar en varios platos → redundancia.

Relaciones

- En las entidades del ejemplo existen *relaciones implícitas*. Por ejemplo: **entre un plato y sus ingredientes**.
¿Cómo se puede resolver utilizando elementos básicos de ER?
- **¿Haciendo que ingrediente sea atributo de plato?** Un plato tiene **varios ingredientes**. Posibles soluciones:
 - (1) Repetir el plato para cada uno de los ingredientes:
`<PL001, California maki, 8.50€, IN001, salmon >`
`<PL001, California maki, 8.50€, IN002, Rice >` → **redundancia**.
 - (2) Hacer el atributo ingrediente **multivalorado**:
`<PL001, California maki, 8.50€, {<IN001, Salmon>, <IN002, Egg>} >`
`<PL002, Nigiri, 3.00€, {<IN003, Wasabi>, <IN002, Arroz>, ...} >`
 - ★ Cada ingrediente tiene atributos → **redundancia**.
 - ★ El mismo ingrediente puede estar en varios platos → **redundancia**.
- ▶ Además, un ingrediente **no podría estar en la BD** si no está asociado a ningún plato.
- **Solución:**

refinar el diseño y establecer relaciones entre las entidades.

Relaciones

Una **relación** es una **asociación entre entidades**.

- Un **tipo de relación** entre tipos de entidades E_1, \dots, E_n define las **asociaciones entre entidades de cada tipo de entidad**.
- Es una **relación matemática**: un subconjunto del producto cartesiano $E_1 \times \dots \times E_n$:

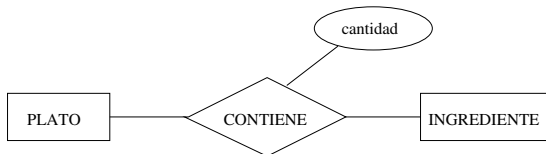
$$\{(e_1, \dots, e_n) \mid e_1 \in E_1, \dots, e_n \in E_n\}$$

- Cada elemento de este conjunto es una **instancia del tipo de relación**.
- **Ejemplo**: Se puede definir una relación `contiene` entre las entidades `plato` e `ingrediente`:

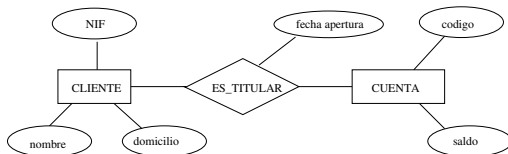


Atributos de relaciones

- Cada plato puede contener un ingrediente dado en una cantidad diferente: `cantidad` es un **atributo de la relación**.
- Una relación **puede tener atributos**.

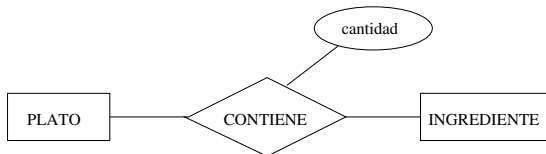


- Los atributos solo deben asociarse a una relación si no se pueden añadir a ninguna de las entidades participantes:

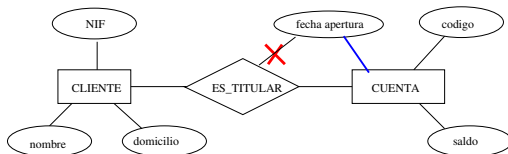


Atributos de relaciones

- Cada plato puede contener un ingrediente dado en una cantidad diferente: `cantidad` es un **atributo de la relación**.
- Una relación **puede tener atributos**.



- Los atributos solo deben asociarse a una relación si no se pueden añadir a ninguna de las entidades participantes:



Relaciones binarias y ternarias

- Las relaciones se pueden establecer **entre dos o más entidades**. Lo vemos con un **ejemplo**:
- Se quiere que la BD guarde información sobre las compras de ingredientes a proveedores en cada establecimiento.
- Intervienen tres entidades:
 - ▶ ingrediente,
 - ▶ establecimiento, con atributos `codigo` y `direccion`,
 - ▶ proveedor, con atributos `CIF`, `RazonSocial` y `direccion`.
- Debemos establecer relaciones entre estas entidades para conocer los proveedores que abastecen de cada ingrediente en cada establecimiento.
- Podemos crear **dos relaciones**:
 - ▶ **abastece** entre `proveedor` y `establecimiento`,
 - ▶ **proporciona** entre `proveedor` e `ingrediente`.

Relaciones binarias y ternarias

- Con las relaciones **abastece** y **proporciona** **no es posible** contestar a la siguiente pregunta: **¿qué proveedor proporciona un ingrediente dado a un establecimiento determinado?**
 - ▶ Cada relación contiene información por separado, pero **no se puede combinar directamente**.

ingrediente:

`{⟨IN001, Patata⟩, ⟨IN002, Huevo⟩, ⟨IN003, Leche⟩, ...}`

establecimiento:

`{⟨ES001, (C/Ponzano 3, 28001, Madrid)⟩,
⟨ES002, (C/Real s/n, 28140, Colmenar)⟩, ...}`

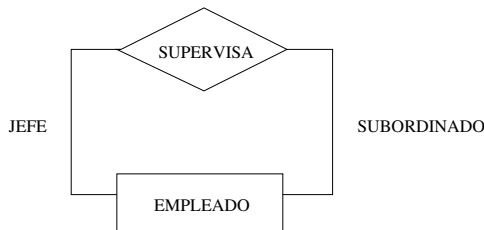
proveedor:

`{⟨CIF01, Granja Laurel, (C/Laurel s/n, 28200, Alcorcón)⟩,
⟨CIF02, Agrícola Roma, (Av/La Paz 30, 46400, Teruel)⟩, ...}`

- Para resolver este problema se debe establecer una **relación entre las tres entidades**.
- El **grado** de una relación es el número de entidades que participan.

Relaciones recursivas.

- También se pueden representar **relaciones recursivas**.
- En una **relación recursiva** un tipo de entidad participa en la relación varias veces con diferentes **roles**.
- Las líneas que unen la entidad con la relación se etiquetan con cada rol.
- Por ejemplo: la relación *supervisa* entre un empleado (el jefe) y otro empleado (el subordinado).



Restricciones: cardinalidad y participación.

- En los diagramas ER se puede añadir información para representar **restricciones** sobre las combinaciones de entidades que participan en una relación.
- Se utilizan para limitar el número de veces que las entidades pueden participar en el conjunto de relaciones.
- **Ejemplos:**
 - ▶ “Un plato contiene ingredientes (al menos uno).”
 - ▶ “Cada empleado debe ser capaz de preparar entre uno y 8 platos.”
 - ▶ (en un taller mecánico) “Un cliente puede ser propietario de varios vehículos, pero un vehículo tiene un único propietario”
- Hay dos tipos principales de restricciones:
 - ▶ Restricciones de **cardinalidad** (o “razón de cardinalidad”).
 - ▶ Restricciones de **participación**.

Restricciones de cardinalidad.

- Especifica el número máximo de relaciones en el que una entidad (de un tipo de entidad) puede aparecer en el conjunto de relaciones.

Ejemplo: *“Un cliente puede ser propietario de varios vehículos, pero un vehículo tiene un único propietario”*

- ▶ Una entidad `cliente` puede aparecer relacionada con varias entidades `vehículo`.
- ▶ Pero una entidad `vehículo` puede aparecer en un elemento de la relación `posee` con una entidad `cliente`.
- Podemos referirnos a valores concretos o genéricos de cardinalidad:
 - ▶ Hay muchas relaciones en las que se usa cardinalidad **1**.
 - ▶ Cuando nos referimos a “*varios*” utilizamos un **N** genérico, en el que el valor exacto de N no se especifica.
 - ▶ Un caso particular de “varios” es un valor concreto mayor a 1.



- ▶ Observa **dónde se sitúa cada restricción de cardinalidad** en el diagrama.

Restricciones de cardinalidad. Razón de cardinalidad

- La **razón de cardinalidad** de un tipo de relación binario especifica el **número máximo de instancias de la relación** en las que una entidad puede participar.
- **Ejemplo:** la relación `cliente:vehículo` tiene una razón de cardinalidad **1:N** : *cada cliente puede poseer varios vehículos, pero un vehículo solo tiene un propietario*
- Razones de cardinalidad posibles en relaciones binarias entre entidades A y B (A:B):
 - ▶ **Uno a uno (1:1):** cada elemento de A se relaciona con **a lo sumo un** elemento de B y viceversa.
 - ▶ **Uno a varios (1:N):** cada elemento de A se relaciona con **cualquier número** de elementos de B, pero cada elemento de B se relaciona con **a lo sumo un** elemento de A.
 - ▶ **Varios a uno (N:1):** cada elemento de A se relaciona con **a lo sumo un** elemento de B, pero cada elemento de B se relaciona con **cualquier número** de elementos de A.
 - ▶ **Varios a varios (M:N):** cada elemento de A se relaciona con **cualquier número** de elementos de B y viceversa.

Restricciones de cardinalidad. Ejemplos.

- **Uno a uno:** relación `dirige` entre `empleado` y `departamento` (un dpto. solo puede tener un director y un empleado puede ser director de a lo sumo un dpto.).



- **Uno a varios:** relación `realiza` entre `cliente` y `pedido` (un cliente puede hacer varios pedidos pero un pedido solo puede hacerlo un cliente).



- **Varios a varios:** relación `contiene` entre `plato` e `ingrediente` (un plato puede contener varios ingredientes, y un ingrediente puede aparecer en varios platos).



Restricciones de cardinalidad. Regla general.

Sea R un tipo de relación binari entre los tipos de entidad E_1 y E_2 . **La cardinalidad de E_1 es n if:**

dada una entidad $e_2 \in E_2$, existen a lo sumo n entidades $e_1 \in E_1$ tales que $\langle e_1, e_2 \rangle \in R$.

- En esta regla, n puede ser 1 o N (cualquier número), o incluso un número concreto mayor a 1.
- **Ejemplo:** “Todo empleado debe ser capaz de preparar entre 1 y 8 platos.”¹



¹las restricciones de participación no se muestran en este diagrama.

Restricciones de cardinalidad. Notación alternativa.

- **Uno a uno:** relación `dirige` entre `empleado` y `departamento` (un dpto. solo puede tener un director y un empleado puede ser director de a lo sumo un dpto.).



- **Uno a varios:** relación `realiza` entre `cliente` y `pedido` (un cliente puede hacer varios pedidos pero un pedido solo puede hacerlo un cliente).



- **Varios a varios:** relación `contiene` entre `plato` e `ingrediente` (un plato puede contener varios ingredientes, y un ingrediente puede aparecer en varios platos).



Restricciones de cardinalidad en relaciones ternarias.

- Una relación ternaria puede tener muchas razones de cardinalidad diferentes: $M:N:Q$, $M:N:1$, $N:1:1$, $1:1:1$ (y sus permutaciones).

- Ejemplo:** En una relación ternaria entre `establecimiento`, `ingrediente` y `proveedor`:

- ▶ Si “todos los proveedores pueden suministrar a cualquier establecimiento varios ingredientes y un ingrediente puede ser suministrado a un establecimiento por varios proveedores”:

es una relación $M:N:Q$

- ▶ Pero si “un establecimiento solo puede ser abastecido de un ingrediente por un solo proveedor”:

es una relación $M:N:1$

En este caso el “1” corresponde al proveedor.

Sea R una relación k -aria. **La cardinalidad de E_j es n si:**

fijados $e_1 \in E_1, \dots, e_{i-1} \in E_{i-1}, e_{i+1} \in E_{i+1}, \dots, e_k \in E_k$ cualesquiera, se verifica que existen a lo sumo n entidades $e_{j1}, \dots, e_{jn} \in E_j$ tales que $\langle e_1, \dots, e_{i-1}, e_{ij}, e_{i+1}, \dots, e_k \rangle \in R, 1 \leq j \leq n$.

Restricciones de participación.

- La **restricción de participación** indica si cada elemento de un tipo de entidad debe participar **obligatoriamente** en la relación o no.
- Corresponde al **número mínimo** de instancias de la relación en las que una entidad debe participar. Hay dos tipos:
 - ▶ **participación total (1)**: cada entidad del tipo de entidad debe participar **en al menos una instancia de la relación**.
Ejemplo: Todo pedido es realizado por un cliente.
 - ▶ **participación parcial (0)**: algunas entidades del tipo de entidad **pueden no participar en ninguna instancia de la relación**.
Ejemplo: Puede haber ingredientes que no formen parte de ningún plato.
- Se representan en el diagrama ER mediante una **doble línea** si la participación es **total**, o una **línea simple** si es **parcial**.
- **Notación alternativa**: cada línea de la relación contiene un par (mín, máx) que indica el número de veces **mínimo y máximo** que aparece la entidad en la relación.

Restricciones de participación. Ejemplos.

- Entre cliente y pedido: “un cliente puede realizar varios pedidos (o ninguno), un pedido está asociado obligatoriamente a un (y solo un) cliente”²:



- Entre plato e ingrediente: “Todo plato tiene al menos un ingrediente, pero puede haber ingredientes no contenidos en ningún plato”:



²Observa que con la notación alternativa (a la izquierda) la N está en el otro lado de la relación (Elmasri, sec. 3.7.4).

Notación alternativa combinando restricciones de cardinalidad y participación.

- También se denomina **notación (min,max)** (see Elmasri, 6ª ed, sec. 7.7.4).
- Combina cardinalidad y participación en un par, **but cambia la ubicación de las restricciones de cardinalidad.**
- **Ejemplos:**
 - ▶ Entre cliente y pedido:



- ▶ Entre plato e ingrediente:



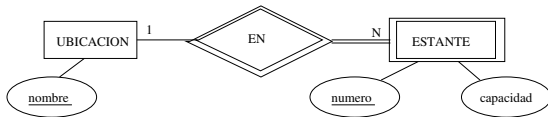
- Es **más expresiva**: permite expresar restricciones de participación diferentes de 0 (parcial) y 1 (total).
- Tiene un **significado diferente** en relaciones ternarias.

Tipos de entidad débiles.

- Un **tipo de entidad débil** es la que no tiene ningún conjunto de atributos que pueda actuar como clave primaria:
 - ▶ Pueden existir varias entidades diferentes con exactamente los mismos valores para todos los atributos.
- Debe estar asociado siempre a otra entidad, denominada **entidad identificadora** o propietaria, mediante una **relación identificadora**.
- La **relación identificadora** siempre debe tener las siguientes características:
 - ▶ Razón de cardinalidad 1:N (de la entidad identificadora a la entidad débil).
 - ▶ La **participación** de la entidad débil debe ser **total**.
- El tipo de entidad débil sí debe tener atributos que formen una **clave parcial** que discrimine entre las distintas entidades débiles que se corresponden con cada entidad identificadora.

Tipos de entidad débiles.

- En el diagrama ER los tipos de entidad débiles se representan mediante una caja doble y las relaciones identificadoras mediante un rombo doble.
- Ejemplo:** *“Los ingredientes se almacenan en **estantes** situados en varias **ubicaciones** (frigorífico, congelador, despensa...). Los estantes de cada ubicación están numerados comenzando en 1. Por tanto, puede haber varios estantes con el mismo número pero en diferentes ubicaciones.”*



Modelo ER extendido.

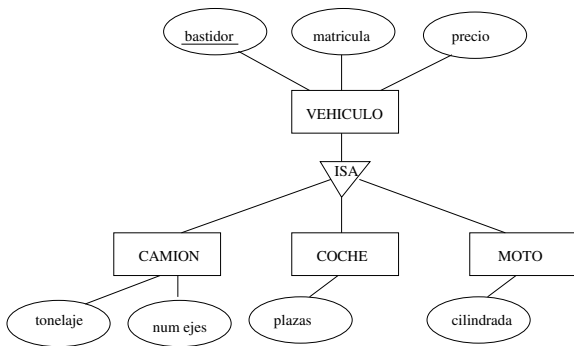
- En el **modelo ER extendido (EER)** se añaden otros elementos para dar más expresividad a los diagramas de diseño de una BD:
 - ▶ Especialización y Generalización
 - ▶ Conjuntos de entidades de nivel más alto y más bajo
 - ▶ Herencia de atributos
 - ▶ Agregación

Modelo EER. Especialización y Generalización.

- Durante el diseño del modelo ER se pueden identificar subgrupos de entidades que tienen características comunes, pero diferentes de las demás entidades del tipo.
- La **especialización** es una técnica de diseño **descendente**:
 - ▶ Se parte de un conjunto de entidades inicial.
 - ▶ Se van identificando subgrupos (**subclases**) de entidades que son **especializaciones** de la inicial.
- La **generalización** es la técnica de diseño inversa:
 - ▶ Se parte de entidades que comparten características comunes de forma que se pueden **generalizar** en una única **superclase**.
- En el diagrama EER no se hace distinción de estas dos técnicas.
- Se utiliza un elemento común a ambas: una figura triangular denominada **ISA** (*es-un*) para identificar las relaciones clase/subclase.
- Las subclases **heredan los atributos y la participación en relaciones de las superclases**.

Modelo EER. Especialización y Generalización. Ejemplo.

- **Ejemplo:** se puede iniciar el diseño utilizando la entidad `vehículo` y después especializarlo para cada tipo: `camión`, `coche` o `moto`.
- Los atributos y relaciones del nivel superior **se heredan** en los niveles inferiores.



Modelo EER. Agregaciones.

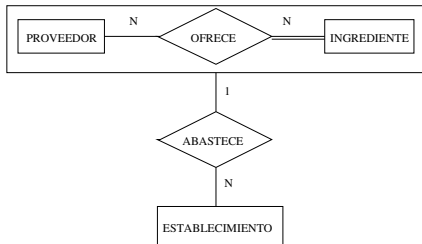
- El modelo ER solo permite establecer relaciones entre entidades. No es posible conectar directamente una relación con otra relación.
- Hay algunos casos en los que puede ser necesario tratar una relación **como una entidad de un nivel más alto.**
- En el modelo EER se puede considerar un conjunto de componentes (varias entidades y una relación que las une) **como si fueran una única entidad.**
- Para representarlo se encierra este conjunto en un rectángulo y se conecta este rectángulo a otras relaciones del diagrama EER.

Modelo EER. Agregaciones. Ejemplo

- En el ejemplo de la empresa de platos preparados, supongamos que tenemos una relación `ofrece` que indica para cada proveedor los ingredientes que ofrece.



- Si nos indican que “un establecimiento es abastecido por un proveedor **de los ingredientes que ofrece**”, la relación ternaria `abastece` vista anteriormente ahora puede relacionar el establecimiento con una agregación:



Del diseño conceptual al diseño lógico

- El modelo ER no lo suelen implementar directamente los sistemas de gestión de BD.
- El diseño conceptual que proporciona el modelo ER se debe convertir en un diseño lógico de más bajo nivel, el **modelo relacional**.
- El proceso de conversión al modelo relacional es sistemático siguiendo una serie de reglas que veremos en el próximo tema.
- Algunas herramientas y frameworks CASE generan de forma (semi-)automática el esquema DDL de una BD a partir de un modelo ER (o los diagramas de clases UML).