

## Bloque 2. Modelos de Datos y Diseño de BD Relacionales

### TEMA 1: Enfoque bases de datos

#### 1.- Que es una base de datos

Una **Base de Datos** (BD) es una **colección organizada de información** (de datos estructurados), que normalmente se almacena de forma electrónica en un sistema informático.

Un **dato** es un **hecho conocido con significado implícito que puede ser registrado.**

Las bases de datos y la tecnología de bases de datos tienen un impacto vital sobre el uso creciente de los ordenadores. El uso de bases de datos permite almacenar grandes volúmenes de datos y mantenerlos organizados de forma que su manipulación sea sencilla y su consulta sea eficiente.

Además, permite disponer de mecanismos que garanticen tanto la corrección de los datos (integridad) como su seguridad

#### 2.- Primeras nociones

- **Esquema:** es un **conjunto de elementos de datos**, al que se le da un nombre.

En una BD se suele usar los esquemas para agrupar los elementos de datos que pertenecen un mismo contexto. En general, una misma base de datos (física) puede contener **varios esquemas diferentes**

- **Sistema gestor de base de datos (SGBD):** es un conjunto de programas que permite definir, crear, manipular y controlar el acceso a la base de datos.
  - **Definir** una BD es especificar las estructuras de datos, los tipos de datos y las restricciones de los datos
  - **Crear** una BD es almacenar datos en algún medio de almacenamiento controlado por el SGBD
  - **Manipular** la BD es consultar datos, introducir/modificar/eliminar datos, para reflejar cambios en el dominio y generar informes a partir de los datos almacenados
  - **Controlar el acceso** es proporcionar seguridad, integridad, control de la concurrencia y de la recuperación después de fallos, y un catálogo de descripciones de los datos (metadatos)
- **Metadatos** describen la estructura de la BD
  - Incluyen descripciones de los datos y de las restricciones (de integridad y de seguridad) que los datos deben cumplir
  - También incluyen descripciones de los esquemas de base de datos ya creados en la BD Y más cosas...
  - Detalles acerca de los usuarios de la base de datos
  - Estadísticas de almacenamiento y de uso/acceso a datos
  - Etc.

La existencia de los metadatos proporciona a la base de datos una naturaleza autodescriptiva.

Los metadatos están agrupados y almacenados en SU propio esquema de la BD

- **Sistemas de bases de datos:** formado por la Base de Datos, el SGBD y los programas de aplicación

#### 3.- Panorama actual

- **tipos de SGBD**
  - Según el Modelo de Datos en el que está basado
    - Relacional, Red, Jerárquico,
    - Orientado a Objetos,
    - Objeto/Relacional,
    - NoSQL: Documentos, Clave-Valor, Columnar, Grafos...
  - Según el número de usuarios simultáneos que admite
    - Monousuario
    - Multiusuario
  - Según el número de lugares en que se almacenan datos
    - Centralizado

- Distribuido (SGBDD)
  - SGBDD homogéneo: mismo software de SGBD en todos los sitios
  - SGBDD heterogéneo (ej. Multi-Base de Datos o BD Federadas)
- Según su propósito
  - de propósito General
  - de propósito Específico: construido para un tipo concreto de aplicaciones
- **proveedores**
  - **SGBD Relacionales**
  - **SGBD NoSQL**

## TEMA 2: Modelo de datos.

**SI (Sistema de Información):** Colección de personas, procedimientos (reglas) y equipos diseñados, contruidos, operados y mantenidos para recoger, procesar, almacenar, recuperar y visualizar información. La Base de Datos es un componente fundamental del Sistema de Información. Para **construir** un Sistema de Información es necesario realizar un análisis de la funcionalidad y de los datos del sistema, para después hacer un diseño a través de modelos que facilitan su comprensión.

**(Diseño SW:**Modelado de Procesos funcionamiento del sistema; procesos y subprocesos y cómo transforman los datos.)

**Diseño BD:** Modelado de Datos información con la que trabaja el sistema: datos, cómo se relacionan entre sí y qué restricciones deben cumplir.

MODELADO DE DATOS: se usan para conseguir esa visión abstracta. **Conjunto de conceptos, reglas y convenciones que permiten describir y manipular datos.** Un modelo de datos es una herramienta formal para comprender y representar el mundo real. Tipos:

- Conceptuales: permite crear el Esquema Conceptual, que describe los tipos de entidad, los tipos de relación, los atributos y las restricciones
- Logicos: permite describir la estructura lógica global de la base de datos, mediante el Esquema Lógico, que es una descripción de la implementación
  - Incluye conceptos entendibles por usuarios finales, pero no lejos de organización física de datos
  - Oculta detalles de implementación, pero son conceptos implementables directamente en el sistema
- Fisicos: permite describir la estructura física global de la base de datos mediante el Esquema interno, que especifica los detalles de almacenamiento de los datos:
  - Formato y ordenamiento de registros en los ficheros de datos en disco
  - Tamaños de página, de bloque, ...
  - Estructuras de almacenamiento
  - Estructuras de acceso a los datos (ficheros índices, etc.)
  - Etc.

## TEMA 3: El proceso de diseño de base de datos.

### 1.- Diseño de base de datos

Diseñar la estructura lógica y física de los datos para satisfacer las necesidades de información de los usuarios en una organización, para un conjunto definido de aplicaciones.

**MÉTODO** de diseño para desarrollo de BD: Serie de etapas que realizar en el proceso de diseño de una base de datos, junto con herramientas que facilitan la representación de los datos en cada etapa, y las reglas que permiten el paso de una etapa a la siguiente.

**Herramientas:** Recursos para realizar operaciones que se prevén en el método:

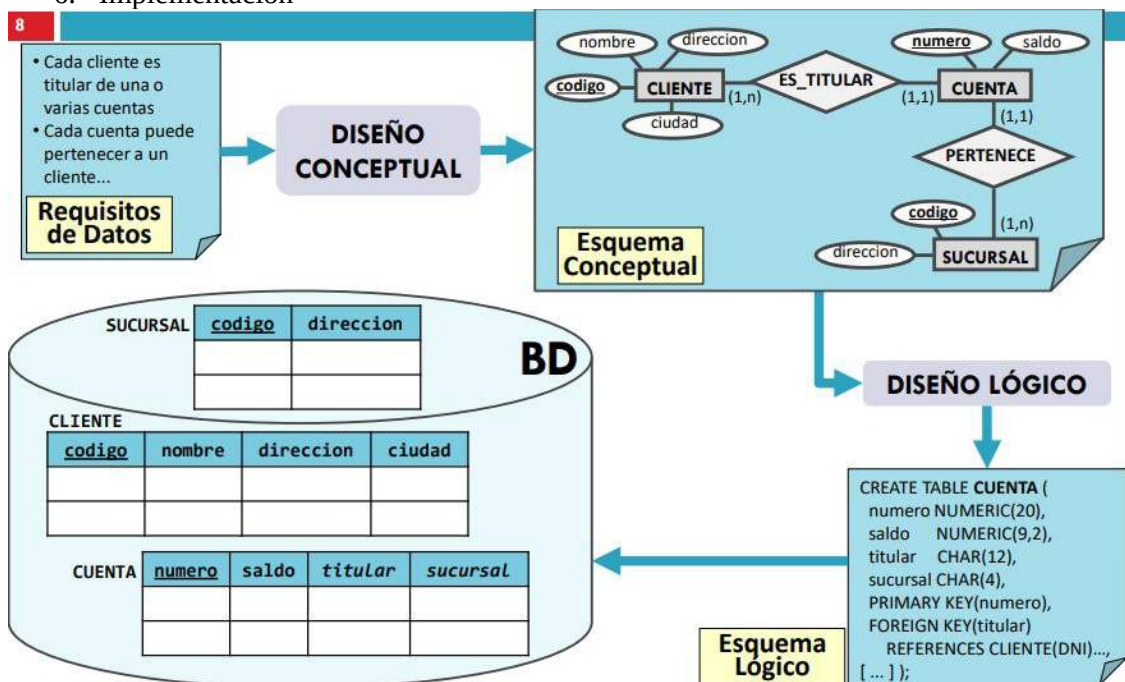
- Modelo de datos
  - Conjunto de conceptos, reglas y convenciones que permiten describir y manipular datos
  - El esquema obtenido es la visión de la realidad que tiene el diseñador
- Lenguaje de datos

- Definición de una sintaxis sobre un modelo de datos (ej. SQL)
- Otras herramientas:
  - CASE, adquisición de requisitos, diccionario de datos
  - Documentación, etc.

**Reglas:** estrategias y criterios para actuar sobre los elementos de entrada de cada etapa y conseguir las salidas (los resultados).

## 2.- Etapas de diseño

1. Recopilación de Requisitos de usuario
2. **Diseño Conceptual**
3. Elección del SGBD
4. **Diseño Lógico**
5. Diseño Físico
6. Implementación



1. **Diseño de la base de datos:** estructura y contenido de la **base de datos**
2. **Diseño del procesamiento de la base de datos:** transacciones y **aplicaciones software**.

## TEMA 4: Diseño conceptual y modelo Entidad-Relacion.

Diseño Conceptual: Es el proceso de consturir un esquema de los datos utilizados por una empresa u organización, independientemente de toda consideración física. Para ello, es necesario el **análisis meticulouso del Catálogo de Requisitos de Datos** del sistema

Incluye las siguientes fases

1. Análisis de los requisitos de datos recogidos
2. Diseño del Esquema Conceptual de datos

### ESQUEMA CONCEPTUAL:

Se obtiene al analizar los **requisitos** de datos, mediante refinamiento y estructuración progresivos

Es un resumen de los requisitos de datos: **una descripción “formal” de alto nivel** de la base de datos. Debe ser diseñado sin considerar aspectos técnicos y/o de implementación como estos:

- Usuarios, SGBD concreto, Plataformas hardware
- Lenguajes de programación, Programas de aplicación
- Restricciones de rendimiento, etc.

Es un vehículo de comunicación entre usuarios (no técnicos), diseñadores y analistas . Conceptos fáciles de entender (alto nivel de abstracción). Útil para documentar el proceso de diseño

## **ENTIDAD-RELACION**

☉ **Entidad (entity)** o instancia: cosa u objeto del mundo real con existencia propia y distinguible del resto. Puede ser física o real, o abstracta o conceptual.

**TIPO DE ENTIDAD:** Define o representa un conjunto de entidades del mismo tipo. (Rectángulo, mayúscula, singular).

Una entidad, por tanto, es una instancia de un tipo de entidad.

Para descubrir los tipos de entidad hay que combinar dos criterios:

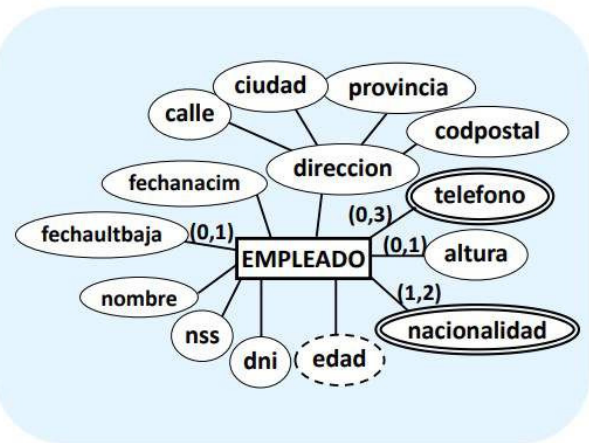
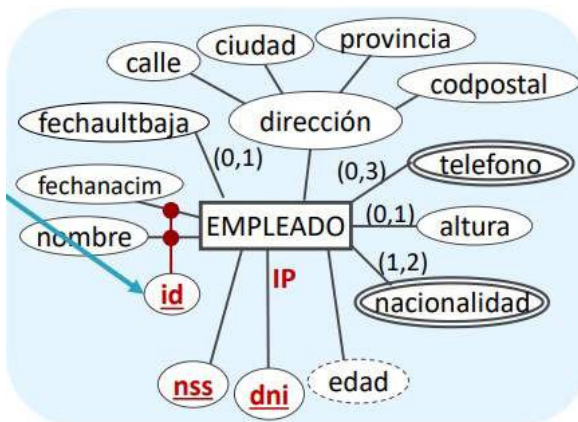
- **Criterio de categorización de objetos:** reglas basadas en el papel desempeñado por los datos
  - Concepto con más propiedades además de su propio nombre
  - Concepto que describe un conjunto de objetos con existencia autónoma
- **Criterio lingüístico:** reglas basadas en la gramática
  - Sustantivo sujeto o complemento directo

☉ **Atributo (attribute)** : Propiedad o característica de un tipo de entidad. Una instancia (una entidad) particular es descrita por los valores de sus atributos. Descubrir atributos combinando dos criterios:

- **Criterio lingüístico:**
  - Preposición o frase preposicional entre nombre de tipo de entidad y propiedad (sustantivo)
  - Sustantivo sujeto o complemento directo
- **Criterio de categorización**
  - Concepto al que se le asigna un valor
  - Concepto simple, sin otras propiedades asociadas

Tipos de atributos:

- Atributo simple: no divisible.
- Atributo compuesto: se divide en otros con significado propio.
- Atributo almacenado: valor proviene del mundo real.
- Atributo calculado: valor calculado a partir de info ya existente.
- Atributo monovalorado: solo un valor para cada instancia.
- Atributo multivalorado: mas de un valor a la vez para la misma entidad.
- Atributo obligatorio
- Atributo opcional
- Atributo clave o identificador: atributo con valor distinto para cada instancia de un tipo de entidad.
  - clave compuesta: clave formada por mas atributos. Debe ser minima. Se subrata la generica
  - clave principal: IP clave de todas las que hay que mas se use en la practica.



⊙ **Relación (relationship):** Asociación, vínculo o correspondencia entre instancias de tipos de entidad, que están relacionadas de alguna manera en el 'mundo real'.

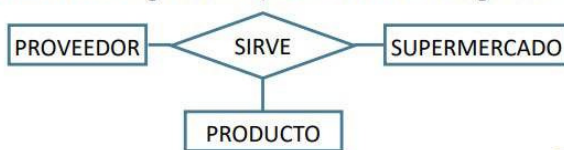
**TIPO DE RELACION:** Estructura genérica o abstracción del conjunto de relaciones existentes entre dos o más tipos de entidad. Sólo puede haber relaciones entre tipos de entidad. Un tipo de relacion puede tener distinto grado.

#### ■ **Binaria:** grado 2

■ El más frecuente



#### ■ **Ternaria:** grado 3; **Cuaternaria:** grado 4, etc.



#### ■ **Recursiva/Reflexiva:** grado 1



- **Criterio de categorización de objetos:**
  - Concepto que proporciona un vínculo lógico entre entidades
  - Concepto que hace posible seleccionar una entidad a través de una propiedad de otra entidad
- **Criterio lingüístico:** reglas basadas en la gramática
  - Verbo transitivo o frase verbal
  - Preposición o frase preposicional entre nombres de tipos de entidad ya identificados

#### **Nombres de Rol(papel)**

Todo tipo de entidad que participa en un tipo de relación juega un papel específico en la relación. Los nombres de rol se deben usar, sobre todo, en los tipos de relación recursivos, para evitar la ambigüedad y ayudar a explicar su significado

#### **Cardinalidad de tipo de entidad**

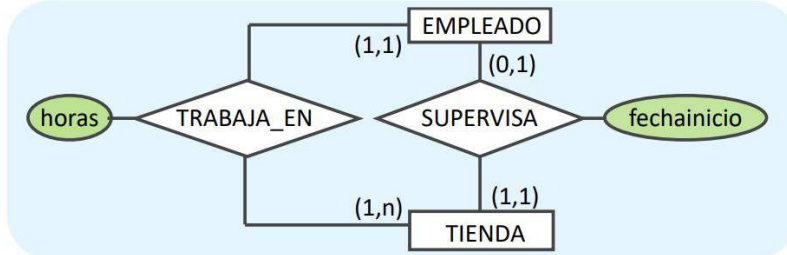
Describe las posibles combinaciones de entidades que pueden participar en las relaciones. La cardinalidad de un tipo de entidad E que participa en un tipo de relación R describe cómo participa cada instancia de E en R.

- **Cardinalidad minima** de los tipos de entidad indican cómo es la **participación** del tipo de entidad

- Participación obligatoria: Un tipo de entidad participa obligatoriamente en un tipo de relación cuando su cardinalidad mínima es 1. Todas y cada una de las instancias participan en la relación
- Participación opcional: Un tipo de entidad participa opcionalmente en un tipo de relación cuando su cardinalidad mínima es 0. Existen instancias que no participan en la relación
- **Cardinalidad máxima** de los tipos de entidad participantes **permiten clasificar** los tipos de relación
  - Uno a uno (**1:1**) Si cada instancia de un tipo de entidad se relaciona sólo con una instancia del otro tipo, y viceversa (Ambos tipos de entidad puede tener cardinalidades (0,1) o (1,1) )
  - Uno a muchos (**1:n**) Si cada instancia de uno de los tipos de entidad se relaciona con muchas instancias del otro y cada instancia del otro sólo se relaciona con una instancia del primero (Uno de los tipos de entidad tendrá cardinalidad (0,n) o (1,n). El otro tipo tendrá cardinalidad (0,1) o (1,1) ).
  - Muchos a muchos(**n:m**) Si cada instancia de uno de los tipos de entidad se relaciona con muchas instancias del otro y viceversa.

### Atributos en tipos de relación

Conceptualmente estos atributos pertenecen a la relación, pues sólo tienen sentido por la relación que hay entre los tipos de entidad. Los tipos de relación NO pueden contener claves.



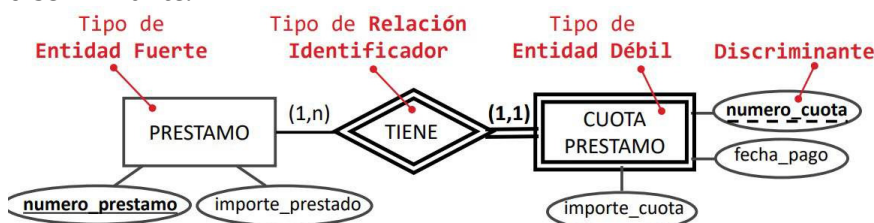
### TIPO DE ENTIDAD DÉBIL:

Un tipo de entidad es débil si entre sus atributos no se puede encontrar una clave que permita identificar de forma única todas y cada una de sus instancias.

Cada instancia de un tipo de entidad débil se identifica por su relación con una instancia de otro tipo de entidad → **entidad fuerte**.

El tipo de relación que vincula el tipo de entidad débil a su fuerte es el tipo de relación identificador. Casi siempre es de tipo **1:N** y con **cardinalidad mínima 1** (participación total) del tipo de entidad débil: jamás existirá una instancia débil no conectada a una fuerte y con **cardinalidad máxima 1**: una instancia débil sólo puede corresponder a una instancia fuerte.

El tipo de entidad fuerte presta su clave principal a la débil para que forme su clave. La clave principal del tipo de entidad débil será la **concatenación de la clave primaria de la fuerte y un discriminante**.



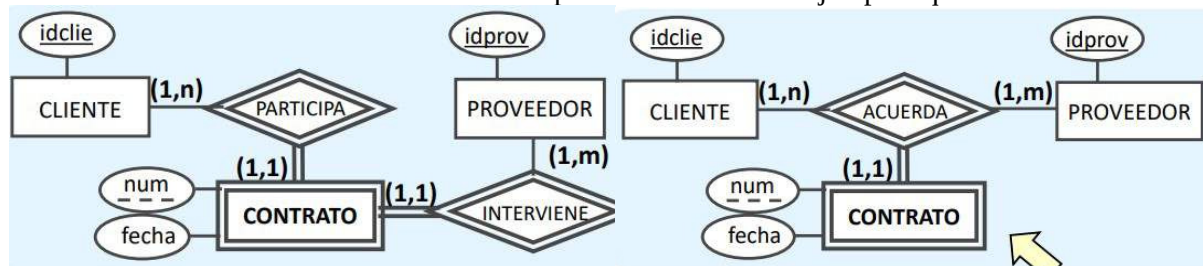
**\*Pero que un tipo de entidad tenga participación total (obligatoria, cardinalidad mínima 1) en un tipo de relación, NO implica que el tipo de entidad sea débil**

### RELACION N-ARIA



Vincula 3 o más tipos de entidad. Un tipo de relación n-aria modela relaciones que sólo existen si participan n instancias al mismo tiempo, cada una de un tipo de entidad distinto. **La relación no existiría si faltara una de las n instancias.**

**Tipo de relación n-aria falsa:** tipo de entidad débil de varios tipos de entidad. DOS tipos de relación identificador conectados al mismo tipo de entidad débil. Ejemplo equivalentes:



### MERE: jerarquías

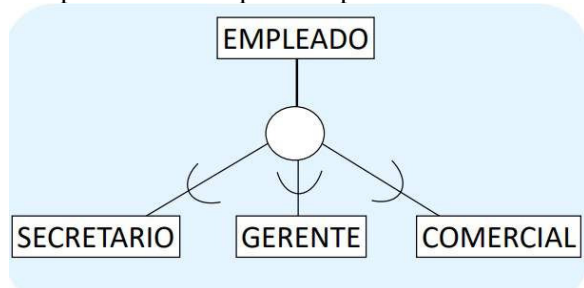
Caso especial de relación entre un tipo de entidad y varios otros tipos de entidad.

Estas jerarquías pueden formarse mediante un proceso de **especialización o bien de generalización**

⊙ Especialización: de lo general a lo específico

⊙ Generalización: de lo específico a lo general

El tipo de entidad que se especializa en otros se llama **supertipo**



La extensión (el conjunto de instancias) de un subtipo es un subconjunto de la extensión del supertipo. Una instancia de subtipo también es instancia del supertipo y es la **misma instancia**, pero con un papel específico distinto. Una instancia del supertipo **puede no ser miembro de ningún subtipo**.

Los subtipos suelen estar definidos según una característica distintiva de las instancias del supertipo: **Atributo discriminante**

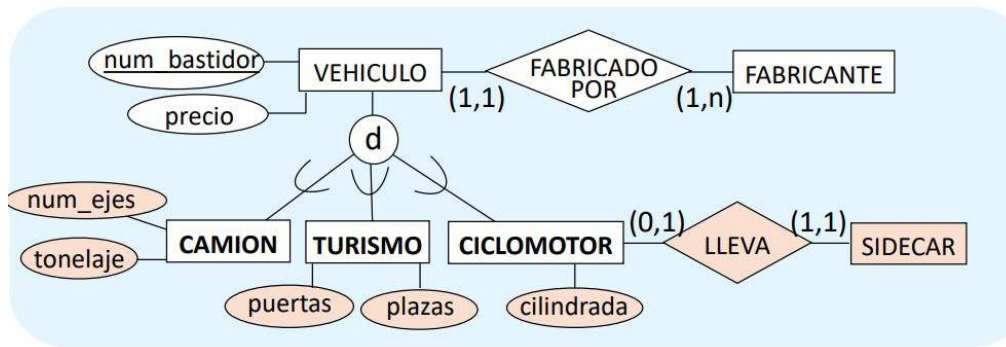
- Disyunción. Subtipos disjuntos: una instancia del supertipo puede ser a la vez miembro de, como máximo, uno de los subtipos ("d" dentro del círculo)
- Solapamiento. Subtipos solapados: una instancia del supertipo puede ser miembro de más de un subtipo a la vez ("O" dentro del círculo)
- Totalidad: toda instancia del supertipo también debe ser instancia de algún subtipo (línea doble entre el supertipo y el círculo)
- Parcialidad: es posible que alguna instancia del supertipo no pertenezca a ninguno de los subtipos (línea simple)

Las restricciones de disyunción y completitud son independientes entre sí. Dan lugar a 4 tipos de jerarquías de especialización:

- Disjunta y Total
- Disjunta y Parcial
- Solapada y Total
- Solapada y Parcial

### HERENCIA

Un subtipo puede tener atributos propios (específicos) y participar en tipos de relación por separado. Un subtipo es un tipo de entidad completo cuando se considera sus atributos y tipos de relación específicos, más los atributos y tipos de relación heredados del supertipo.



## PASOS EN EL DISEÑO CONCEPTUAL

### 1. Determinar tipos de entidad

Realizar una o más lecturas de los requisitos de datos y aplicar los **criterios lingüístico y de categorización** de objetos, para descubrir los tipos de entidad. Crear un diagrama Entidad-Relación.

### 2. Definir tipos de relación (y cardinalidades).

Volver a analizar el catálogo de requisitos de datos, aplicando los criterios lingüístico y de categorización de objetos, para descubrir los tipos de relación que conectan los tipos de entidad. Completar el Diagrama Entidad-Relación (MER) con las cardinalidades

### 3. Definir atributos y asociarlos a tipos de entidad y a los tipos de relación

☉ Tipo de ENTIDAD si el concepto ...

- Tiene asociados otros atributos de interés para el usuario
- Y/O está relacionado con otros tipos de entidad

☉ ATRIBUTO si

- No tiene más propiedad que su valor
- Y no participa en vínculos con otros tipos de entidad

☐ Distinguir Atributo Compuesto vs. Atributo Simple

☐ Distinguir Atributo Opcional vs. Atributos Derivado

### 4. Determinar los atributos clave (principal y alternativas) de los tipos de entidad.

Establecer los atributos identificadores de cada tipo de entidad (todas las claves candidatas). Si un tipo de entidad tiene varios, elegir uno como identificador principal (clave primaria)

- El que tenga menos atributos
- Aquel cuyos valores cambien menos a lo largo del tiempo
- El de valores con menor longitud
- El más empleado por los usuarios para buscar, seleccionar o identificar las instancias del tipo de entidad

tipo de entidad

☉ El resto de identificadores, que siguen siendo claves, son denominados **alternativos**.

### 5. Considerar la incorporación de restricciones de integridad.

Especificar (documentar) las **restricciones de integridad** necesarias para impedir que los datos queden incompletos, imprecisos o incoherentes

☉ Añadir semántica que no puede expresarse en el diagrama mediante conceptos del

Modelo ER

☉ Redactar Reglas de Integridad, RIn, en lenguaje natural

Reglas de dominio → intervalos numericos, tipos que puede tomar un atributo.

Reglas de integridad → condiciones especificadas en el enunciado.

### 6. Comprobar si existe (y minimizar) redundancia



1. Comprobar sinónimos y entidades ‘colgantes’
2. Eliminar las relaciones redundantes
3. Considerar la dimensión del tiempo

## 7. Validar el esquema conceptual contra las transacciones

Asegurarse de que permite la ejecución de toda operación (transacción) incluida en los requisitos. Comprobar que el esquema representa toda la información (entidades, relaciones y sus atributos) requerida por cada transacción.

**Navegación en el Esquema Conceptual:** representar (pintar) el camino tomado por cada transacción en el propio diagrama

## 8. Revisar el esquema conceptual con el usuario

comprobar que quien va a utilizar los datos (ingenieros o científicos de datos, usuarios, etc.) considera el Esquema Conceptual como una “verdadera” representación de los requisitos de datos

⊙ Esquema Conceptual - Diagrama MER

⊙ Diccionario de datos

## TEMA 5: Modelo Relacional

**Dominio:** Conjunto de valores atómicos del mismo tipo, donde toman su valor los atributos. Cada atributo está definido sobre **UN y sólo UN** dominio.  $\{\text{valores de "a"}\} \subseteq \text{Dominio("a")}$

Un dominio puede contener valores no tomados por ningún atributo

Comparaciones restringidas a Dominio: la comparación de dos atributos sólo tiene sentido si ambos toman valores del mismo dominio o compatibles. Si el SGBD soporta dominios, podrá detectar este tipo de errores

**Relación:** Tiene un nombre y representa una entidad genérica.

Representada mediante una tabla. Contiene un conjunto de **tuplas** (filas). Cada tupla representa una entidad concreta (instancia).

Compuesta de atributos (columnas) cada atributo tiene nombre (y dominio) y representa una propiedad de la entidad genérica

Una relación R sobre un conjunto de dominios D1,D2...Dn está compuesta por dos partes:

- Esquema o Cabecera: conjunto de pares (Atributo:Dominio)  $\{(a1:D1), (a2:D2) \dots (an:Dn)\}$ 
  - Cada ai tiene asociado sólo un Di
  - Los Di no tienen por qué ser distintos entre sí; por ejemplo a1 y a3 pueden tener el mismo dominio

El esquema puede incluir otras restricciones de integridad.

- Estado, Cuerpo o Instancia: conjunto de tuplas que contiene la relación en un instante concreto. Tupla = conjunto de pares (Atributo:Valor)  $\{(a1:vi1),(a2:vi2)\dots(an:vin)\}, \dots\}$  para la tupla i,  $i=1..m$

### □ Un esquema de relación:

■ Notación **sencilla** (texto, sin los nombres de dominio):

PELICULA ( idPel, titulo, director, genero, estreno, nacionalidad, duracion )

■ Notación de **tabla**:

PELICULA	idPel	titulo	director	genero	estreno	nacionalidad	duracion
----------	-------	--------	----------	--------	---------	--------------	----------

### □ Un estado de la relación, en forma de tabla:

P56	Avengers: Endgame	Joe Russo	Ciencia-ficción	2019	EEUU	181
P02	La trinchera infinita	Aitor Arregi	Drama	2019	España	147
P10	The Matrix	A.Wachowski	Ciencia-ficción	1999	EEUU	138
P22	Capitán América	Joe Russo	Ciencia-ficción	2011	EEUU	124
P47	Campeones	Javier Fesser	Comedia dramática	2018	España	124

El estado de una relacion variable en el tiempo → introduccion/Modificacion/Borrado

El esquema de relacion no suele variar.

### Propiedades de una Relación:

1. No existen tuplas repetidas: Es imposible que existan dos tuplas con igual valor en todos los atributos.
2. Las tuplas no están ordenadas
3. Los atributos no están ordenados
4. Los valores de atributos son atómicos: en una celda o casilla sólo puede haber un valor.

**Base de datos relacional:** percibida por el usuario como una colección de relaciones con diferentes cabeceras o esquemas de relación (atributos y restricciones) y cuyos estados varían con el tiempo (tuplas en cada relación).

**Todo contenido de información está representado en la BD de una y sólo una forma: como un valor explícito dentro de una posición de columna dentro de una fila dentro de una tabla.**

En una BDR distinguimos

- Esquema de la Base de Datos
  - Conjunto de esquemas de relación
  - Es la descripción de la base de datos: el Esquema Lógico
- Estado o instancia de la Base de Datos
  - Visión del contenido de la base de datos en cierto instante
  - Conjunto de estados de las relaciones de la base de datos

**Restricciones de Integridad:** al incluirlas en el esquema de la BD, el SGBD conoce las restricciones del mundo real así, el SGBD las aplica y evita cualquier configuración de datos imposible.

Toda restricción de integridad cumple que:

- Forma parte de la BD (es parte del esquema de la BD)
- Se satisface para cualquier estado (haya los datos que haya)
- No varía con el tiempo

Son específicas de cada BD particular, pero el Modelo Relacional incluye características básicas de integridad importantes y necesarias en toda BD.

### **Clave de una relación**

Sea  $R$  una relación,  $R(a_1 : D_1, a_2 : D_2, \dots, a_n : D_n)$ . Una **clave de  $R$**  es un subconjunto de sus atributos tal que cumple la restricción de Unicidad

No existen dos tuplas distintas con valores iguales para dichos atributos y también cumple la restricción de Irreductibilidad

- ⊙ no tiene atributos redundantes, es mínima
- ⊙ ningún subconjunto cumple la restricción de unicidad

Una clave es un conjunto de atributos que identifica de forma única cada tupla de la relación

Una clave puede ser **Simple** (un atributo) o **Compuesta** (estar formada por varios atributos).

Una relación puede contener **varias claves**. Si  $R$  tiene varias claves, se llaman **Claves Candidatas**, de entre todas ellas hay que elegir la clave **primaria** (candidata para identificar las tuplas de  $R$ ) y **alternativas**

### **NULL**

Es posible especificar (para toda relación) si un atributo puede o no contener nulo

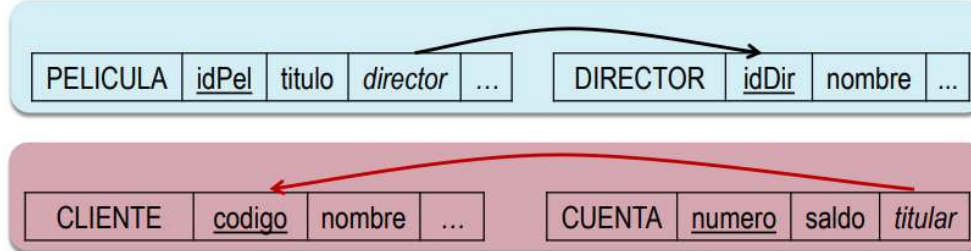
Si en una tupla, un atributo contiene un nulo, significa que el valor real del atributo es desconocido. NULO es un indicador (marca) de ausencia de información.

Ningún atributo componente de una clave primaria puede contener NULL.

Las claves alternativas sí pueden contener NULL

**CLAVE AJENA:** Expresar la conexión entre una tupla de una relación y otra tupla de otra relación. Así, para expresar el vínculo existente entre las tuplas de dos relaciones R1 y R2, es necesario añadir a una de las relaciones R2 “una copia de la clave primaria” de la otra relación R1. Ese nuevo atributo en R2 se convierte en una **clave ajena**.

□ **Diagrama Referencial:** expresión gráfica de la existencia de Claves Ajenas.



Una clave ajena también puede ser Simple o Compuesta. Una FK contendrá tantos atributos como tenga la PK a la que referencia.

Una clave ajena puede ser (o formar parte de la) clave primaria.

Una clave ajena SÍ puede contener NULL

☉ Si una tupla contiene NULL en una clave ajena, significa que esa tupla no participa en el vínculo entre las relaciones.

☉ No referencia a ninguna otra tupla.

**Restricción de Integridad Referencial (Importante):** el valor de una clave ajena debe coincidir con el valor de clave primaria de alguna tupla de la relación referenciada, o bien es completamente nulo o lo que es lo mismo: no hay valores (no nulos) de clave ajena sin correspondencia.

Una clave ajena puede referenciar a su misma relación **Auto-referencia**, significa que cierta tupla (fila) de una relación (tabla) puede estar vinculada con otra tupla distinta de la misma relación

**Camino Referencial:** recorrido del diagrama para obtener información relacionada.

**Ciclo Referencial:** Camino que empieza y acaba en la misma relación (caso especial: Auto-referencia )

**TABLA (atributo1, atributo2, atributo3, atributo4, atributo5, atributo6, atributo7)**

Admiten NULL: **atributo3, atributo6**

Clave Primaria: **atributo2**

Claves Alternativas (UNIQUE): 1. **atributo3** ; 2. (**atributo4, atributo5**);

Claves Ajenas (FOREIGN KEY):

1. (**atributo7**) Referencia\_a **UNATABLA(clave)**

2. (**atributo4, atributo5**) Referencia\_a **OTRATABLA(clave1,clave2)**

...

Derivados:

1. **atributo6 = atributo2\*20/100**

...

Comprobar:

1. **atributo1 IN ('SI', 'NO')**

2. **atributo6 > 0**

3. **atributo7 >= atributo3**

...

## TEMA 6: Diseño logico.

El objetivo principal es transformar el esquema conceptual (EC)(T4) de datos en el esquema lógico (EL)(T5) de datos.

Fases:

- Diseño logico eStandar (DLS)

### Traducción del EC independiente del SGBD específico

Se elige el Modelo de Datos Lógico, pero aún no el SGBD comercial concreto. Uso de un Modelo Lógico de Datos estándar

- Relacional ⇔
- Red
- Jerárquico
- Orientado a Objetos

Obtiene el Esquema Lógico eStandar (ELS) En el Modelo Relacional para describir el ELS se puede usar

- Pseudolenguaje [TABLA(colu1, colu2, ...) y especificaciones]
- Lenguaje de Definición de Datos (LDD) de ANSI SQL

- Diseño logico especifico (DLE)

Ya elegido el SGBD específico (comercial) adaptación del ELS al SGBD

Uso del modelo lógico de datos particular del SGBD elegido

Obtiene el Esquema Lógico Específico (ELE) y se describe con el lenguaje propio del SGBD (dialecto SQL)

## DISEÑO LOGICO ESTANDAR:

### 1. Obtener relaciones (tablas)

- ☐ Crear relaciones para representar tipos de entidad, tipos de relación, y ciertos atributos del esquema conceptual de datos
- ☐ Obtendremos relaciones (tablas) a partir de cada:

#### 1. Tipo de entidad [fuerte]

Se traduce a una relación (tabla) (usar el mismo nombre)

- ☐ La relación incluye estos atributos: uno para cada atributo simple del tipo de entidad y uno para cada componente de los atributos compuestos
- ☐ La clave primaria se deriva del identificador principal. Una clave alternativa (UNIQUE) por cada identificador alternativo

## 2. Tipo de entidad débil

- ☐ Se traduce a una relación (tabla)
- ☐ Puesto que la clave primaria del tipo de entidad débil se obtiene (total o parcialmente) de la de su tipo de entidad fuerte, NO puede ser definida hasta traducir el tipo de relación que la conecta a su fuerte

## 3. Jerarquía

- Disjunta y Total

Da lugar a varias relaciones: una relación por cada combinación supertipo/subtipo

- Tantas relaciones como subtipos
- Cada relación contendrá todos los atributos de cierto subtipo y los del supertipo

Hay que definir una restricción de integridad general (aserto) para garantizar la disyunción

Un aserto es una restricción de integridad que afecta a más de una relación y expresa una condición que sus tuplas deben cumplir

- Disjunta y Parcial

Da lugar a varias relaciones: una relación para el supertipo y una relación para cada subtipo

Las relaciones correspondientes a los subtipos:

- Contienen una clave ajena que referencia a la clave primaria de la relación correspondiente al supertipo
- La clave primaria de cada relación (de subtipo) es dicha clave ajena

Hay que definir una restricción de integridad general (aserto) para garantizar la disyunción

- Solapada y Total

Da lugar a una única relación, que corresponde al supertipo

- Debe incluir uno o más atributos discriminantes para distinguir el tipo concreto de cada tupla
- Los atributos de los subtipos han de admitir el nulo

Hay que definir restricciones de comprobación para asegurar la corrección de los datos para cada tupla según su tipo

- Solapada y Parcial

Da lugar a 2 relaciones: una relación para el supertipo y una relación para todos los subtipos

La relación que agrupa a los subtipos:

- Incluye una clave ajena que referencia a la relación correspondiente al supertipo
- La clave primaria es dicha clave ajena
- Incluye discriminante(s) para distinguir el tipo de cada tupla
- Debe contener restricciones de comprobación que garanticen la corrección de los datos
- Los atributos de los subtipos han de admitir el nulo

## 4. Tipo de relación binaria 1:N

Se traduce a una **clave ajena** que referencia desde una relación (tabla) a la otra. Hay que identificar los tipos de entidad 'padre' (participa varias veces) e 'hijo' (solo participa una vez). La clave ajena servirá para enlazar cada tupla (fila) 'hijo' con su correspondiente tupla 'padre'. Hay que añadir al tipo relación 'hijo' una copia de la clave primaria del tipo relación 'padre' (**propagación de la clave**).

Si el tipo de entidad 'hijo' es débil, se traduce igual y además, la clave ajena forma parte de la clave primaria.

Si el tipo de relación contiene atributos, se añaden como atributos en el tipo de relación 'hijo'

La cardinalidad mínima 0 del tipo de entidad 'hijo' indica que la clave ajena SÍ admite NULO y viceversa.

La cardinalidad mínima 0 del tipo de entidad 'padre' indica que puede haber tuplas (filas) en la relación 'padre' no referenciadas mediante la clave ajena La cardinalidad mínima 1 del tipo de entidad 'padre' debe representarse **mediante una restricción de integridad general (Aserto)**

### 5. Tipo de relación binaria 1:1

Se traduce a una **clave ajena**.

Obligatoria en un lado (0, 1) + (1,1) : Traducir a una clave ajena en la relación correspondiente al tipo de entidad hijo (la obligatoria)

Obligatoria en ambos lados (1,1) + (1,1): Traducir a una clave ajena, que se puede incluir en cualquiera de las dos relaciones (usar el sentido comun).

Opcional en ambos lados (0,1) + (0,1): Elegir "al azar" pero "con sentido común" los tipos de entidad 'padre' e 'hijo' y traducir a clave ajena

### 6. Tipo de relación binaria 1:1 recursiva

Decidir con base en la participación de cada tipo de entidad (cardinalidad mínima)

Obligatoria en ambos lados: Añadir a la relación correspondiente al tipo de entidad una copia de su propia clave primaria. Dicha copia es una clave ajena que referencia a la misma relación.

Opcional en ambos lados: crear una nueva relación para el tipo de relación

- Con 2 copias de la clave primaria de la relación en la que se ha traducido el tipo de entidad

- Las dos son claves ajenas a la misma relación

- Renombrar adecuadamente dichas claves ajenas (**usar roles**)

Obligatoria en un lado: Traducir a clave ajena, como en el caso obligatoria en ambos lados.

- Tipo de entidad 'padre': la de participación opcional - (0,1)

- Tipo de entidad 'hijo': la de participación obligatoria - (1,1)

### 7. Tipo de relación binaria M:N

**Crear una nueva relación**

Añadir una copia de las claves primarias de los tipos de entidad conectados: serán claves ajenas a cada una de las relaciones correspondientes a dichos tipos de entidad

⊙ Incluir atributos (columnas) para los atributos del tipo de relación

⊙ La clave primaria de la nueva relación será la concatenación de ambas claves ajenas

**Añadir un aserto por cada cardinalidad mínima 1**

Si la concatenación de ambas claves ajenas no forma una clave, significa que en el esquema conceptual se omitió un tipo de entidad conectado al tipo de relación. **Es un error de diseño.**

### 8. Tipo de relación n-aria

**Crear una nueva relación**

Incluir atributos (columnas) para los atributos del tipo de relación

⊙ Añadir una copia de las claves primarias de los tipos de entidad conectados: serán claves ajenas a cada uno de ellos

⊙ La clave primaria de la nueva relación será una de estas:

- Concatenación de las claves ajenas

- Combinación de algunas claves ajenas

- La concatenación de varias claves ajenas y algún atributo



A veces, la clave primaria puede reducirse.

Una traducción de una relación de un tipo de relación n-aria aunq tenga cardinalidad mínima 0 no puede contener nulo ninguna de las claves.

### 9. Atributo multivalorado

Crear una nueva relación

Incluir un atributo (columna) con el mismo nombre que el atributo multivalorado

⊙ Añadir una copia de la clave primaria del tipo de entidad a la que está conectado: será una clave ajena

⊙ La clave primaria de la nueva relación será uno de estos:

- La concatenación de la clave ajena y el atributo
- El atributo “en solitario”

traducción MER(modelo entidad relación) → MR (modelo relacional)

Siempre hay que tener en cuenta que, según los tipos de relación en los que participen los tipos de entidad y la existencia de jerarquías, a veces habrá que adaptar la traducción para conseguir un resultado (esquema lógico) correcto.

### 2. Validar las relaciones contra las transacciones de usuario

Asegurar que el esquema lógico (EL) soporta las transacciones requeridas, para ello “Ejecutar manualmente” las transacciones usando las relaciones (tablas), vínculos clave-primaria/clave ajena y el diccionario de datos. Si alguna no se puede resolver → ERROR (en diseño conceptual DC o traducción a MR).

### 3. Revisar las restricciones de integridad

Evitar la pérdida de semántica inherente a la traducción Esquema Conceptual → Esquema Lógico

Asegurar que se ha documentado toda restricción de integridad necesaria para impedir que la BD pueda llegar a estar incompleta, imprecisa o incoherente. Comprobar estos tipos de Restricciones de Integridad

- Datos requeridos u obligatorios

Comprobar que los atributos de tipos de entidad que no admiten NULL.

Comprobar que los atributos de tipos de relación han sido traducidos correctamente, de forma que admiten nulos o no en función de las cardinalidades mínimas

Revisar el campo “Admite nulos” de las fichas de descripción correspondientes.

- Restricciones de dominio de atributos
- Integridad de entidad
- Integridad referencial (claves ajenas)
- Restricciones generales

### 4. Validar el esquema lógico con los usuarios

Un aserto es una restricción de integridad que afecta a más de una relación y expresa una condición que sus tuplas deben cumplir. Usar como vocabulario los nombres de relación y de atributo. En el esquema lógico ya no hay tipos de entidad, ni tipos de relación, ni instancias. Tenemos relaciones, atributos, tuplas, valores... **Expresar lo que NO puede ocurrir.**