

Bases de Datos

Unidad 2: Diseño Lógico de Bases de Datos

UNIDAD 2: DISEÑO LÓGICO DE BASES DE DATOS

- 1. Modelo de datos**
- 2. La representación del problema: los diagramas E/R**
 - 2.1 Entidades**
 - 2.2 Atributos y tipos**
 - 2.3 Relaciones**
 - 2.4 Cardinalidad**
 - 2.5 Debilidad**
- 3. Integridad referencial y dependencia funcional**
- 4. El modelo E/R ampliado.**
- 5. Construcción de un diagrama E/R**
- 6. El modelo relacional**
 - 6.1 Características de una relación.**
 - 6.2 Claves primarias y claves ajenas**
 - 6.3 Restricciones**
 - 6.4 Representación del esquema relacional**
 - 6.5 Paso del diagrama E/R al modelo relacional.**
- 7. Normalización.**

1. Modelo de datos

Un modelo de datos es un conjunto de herramientas y reglas para representar los datos, las relaciones entre éstos y las restricciones de una base de datos.

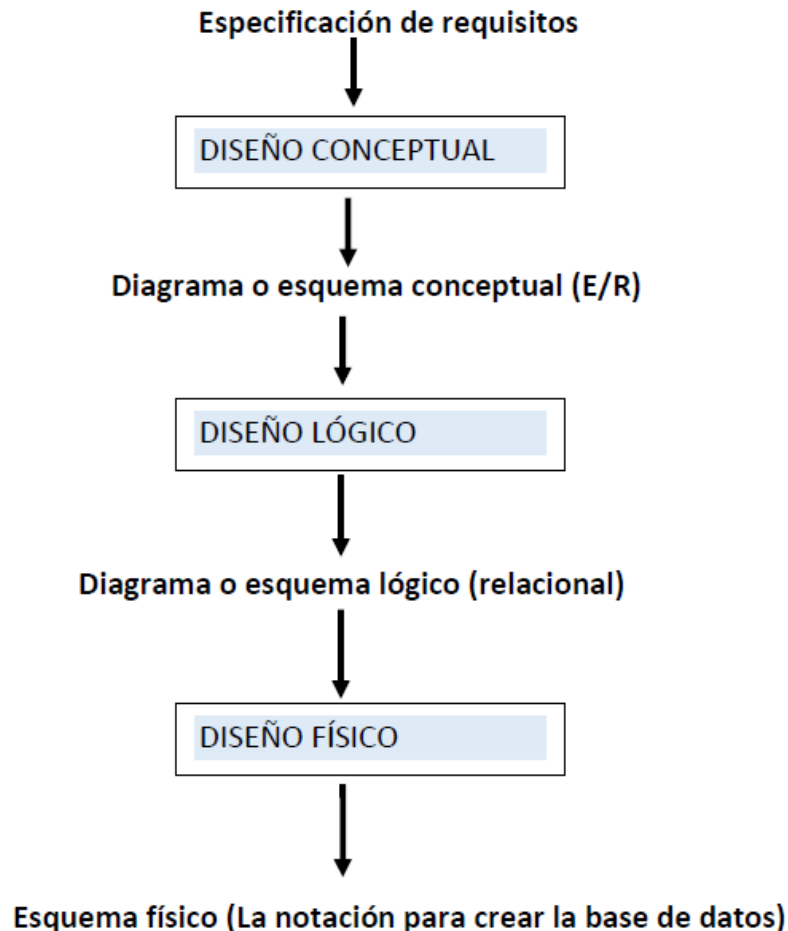
Fundamentalmente se han utilizado los siguientes modelos de datos:

- ☐ Entidad-Relación
- ☐ Relacional
- ☐ Jerárquico
- ☐ En red
- ☐ Orientado a Objetos
- ☐ Relacional orientado a objetos



2. Los diagramas E/R

El proceso de diseño de una base de datos, desde la descripción del funcionamiento del sistema hasta la implementación de la BD

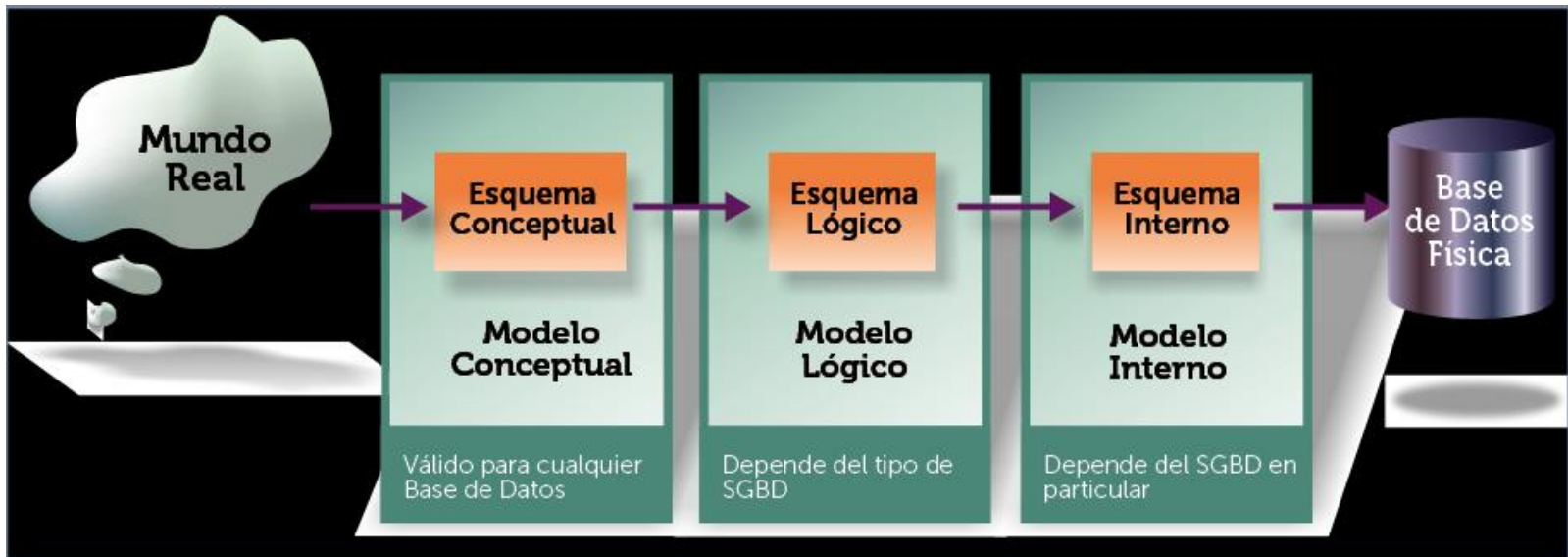


2. Los diagramas E/R

El **diseño/modelo conceptual** parte de las especificaciones de requisitos de usuario y su resultado es el **esquema conceptual** de la base de datos.

El **diseño/modelo lógico** parte del esquema conceptual y da como resultado un **esquema lógico**.

El **diseño/modelo físico/interno** es el nivel más bajo que contiene los registros almacenados, los métodos de representación, los campos de datos e índices. Solo hay un esquema interno por cada base de datos.



2. Los diagramas E/R

Diferencias esenciales entre el modelo lógico y el conceptual

- El **modelo conceptual** es independiente del SGBD que se vaya a utilizar.
- El **modelo lógico** depende de un tipo de SGBD en particular.
- El **modelo conceptual** es el más cercano al usuario.
- El **modelo lógico** es el encargado de establecer **el paso entre el modelo conceptual y el modelo físico del sistema** (el que utiliza internamente el ordenador).

2. Los diagramas E/R

- ❑ El modelo Entidad-Relación es un modelo puramente conceptual.
- ❑ Representa el funcionamiento de un sistema de información mediante un diagrama Entidad Relación (E/R).
- ❑ Facilita enormemente el diseño de una base de datos. Es muy representativo del funcionamiento del sistema de información.
- ❑ Toma como referencia la percepción que tenemos del funcionamiento del mundo real:
 - Esa percepción se basa en entidades que actúan sobre otras entidades haciendo procesos.
 - Consta de una colección de objetos básicos llamados **entidades** y de unas **relaciones** establecidas entre dichas entidades.

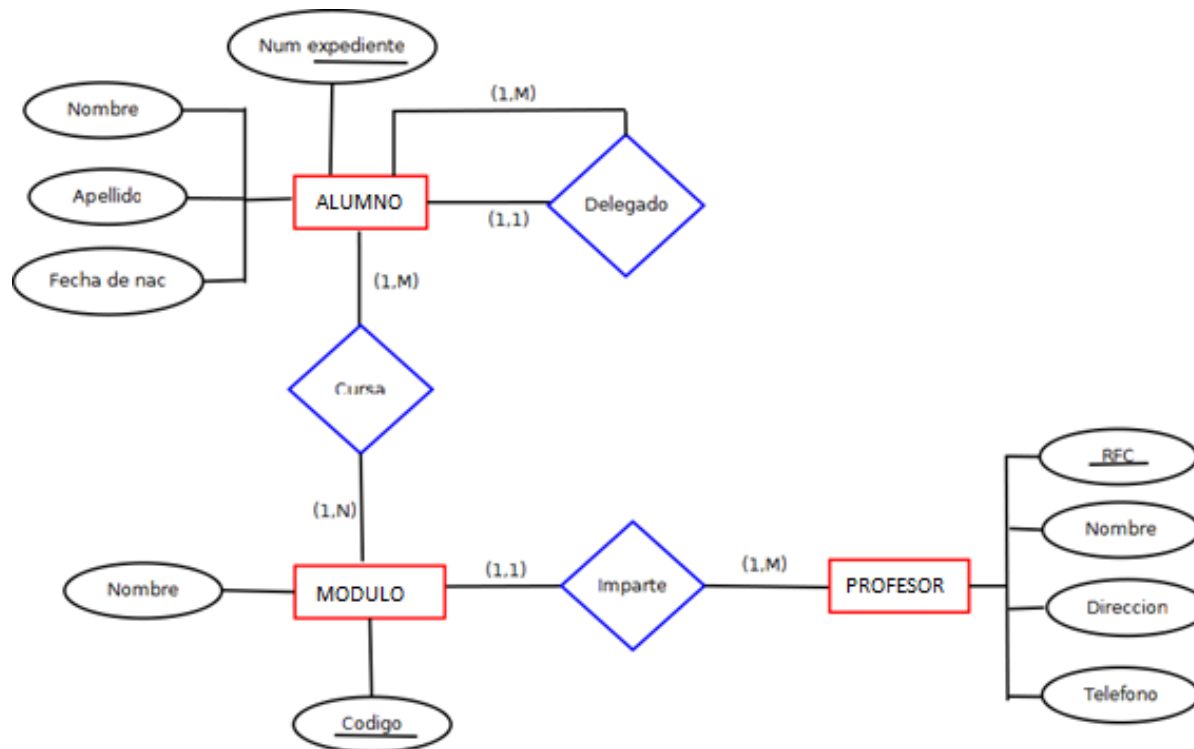
2. Los diagramas E/R

Los modelos conceptuales deben ser buenas herramientas para representar la realidad, por lo que deben poseer las siguientes cualidades:

- ☐ **Expresividad:** deben tener suficientes conceptos para expresar perfectamente la realidad.
- ☐ **Simplicidad:** deben ser simples para que los esquemas sean fáciles de entender.
- ☐ **Minimalidad:** cada concepto debe tener un significado distinto.
- ☐ **Formalidad:** todos los conceptos deben tener una interpretación única, precisa y bien definida.

2. Los diagramas E/R

- ❑ Existen diferentes modelos E/R y diagramas de representación para modelar los requisitos .
- ❑ Vamos a utilizar el modelo de Chen. El siguiente diagrama E/R usa las normas de representación de Chen.



2.1 Entidades

- ❑ Las entidades son uno de los elementos usados en los diagramas E/R.
- ❑ ***Una entidad es un objeto, sujeto o concepto sobre el que se recoge información básica en el sistema para poder realizar los procesos que se requieran.***
- ❑ En un sistema de información que permite gestionar el funcionamiento de un centro de estudios, serían entidades:

ALUMNO

PROFESOR

ASIGNATURA

CURSO

¿Qué otras entidades pueden darse en un centro de estudios?

¿Y en un concesionario de coches?

2.1 Entidades

- ❑ *Una entidad se representa en un diagrama E/R mediante un rectángulo.*

ALUMNO

PROFESOR

CURSO

MODULO

2.2 Atributos y tipos

Un atributo es una propiedad o una característica de una entidad.

Como veremos más adelante, las relaciones también pueden tener atributos.

Por ejemplo, la entidad ALUMNO puede tener los atributos:

- ☐ Numero
- ☐ Nombre
- ☐ Apellidos
- ☐ Fecha Nac.
- ☐ Poblacion

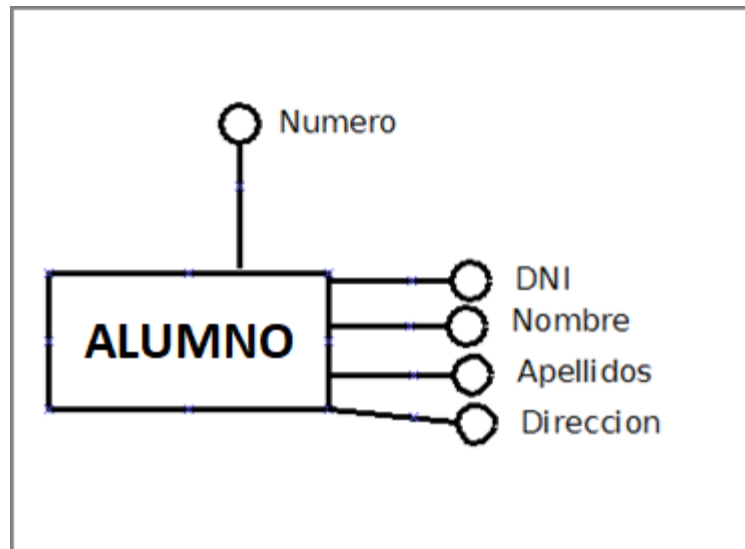


The screenshot shows a web form titled 'Datos Alumno' with a Windows logo icon. The form is divided into sections. The first section, 'DATOS PERSONALES', contains fields for 'Nombre' (Francisco), 'Apellidos' (Meneu), 'Fecha nac.' (06/07/2009), 'Población' (Betxi), and 'Código Postal' (12549). Below this, there are fields for 'Nombre Padre' (Paco), 'Nombre Madre' (Maria), 'Teléfono Padre' (00000000), and 'Teléfono Madre' (1111111111). At the bottom, there are tabs for 'Personal' and 'Académico'.

DATOS PERSONALES	
	Nombre Francisco
	Apellidos Meneu
Fecha nac.	06/07/2009
Población	Código Postal
Betxi	12549
Nombre Padre	Nombre Madre
Paco	Maria
Teléfono Padre	Teléfono Madre
00000000	1111111111
Personal	Académico

2.2 Atributos y tipos

Los atributos de una entidad, se representan mediante pequeños círculos unidos a la entidad por una línea. Al lado de cada círculo se escribe el nombre del atributo.



2.2 Atributos y tipos

Dominios de los atributos

El **dominio** de un atributo es todo el conjunto de valores que se pueden asignar a ese atributo.

Ejemplos de atributos y dominios de una entidad **EMPLEADO**:

Atributo	Dominio
DNI	Cadena de caracteres de longitud 9
Nombre	Cadena de caracteres de longitud 20
Apellidos	Cadena de caracteres de longitud 30
Antigüedad	Fecha
Salario	Número real con dos decimales
Categoría	Enumerado de Categorías
JornadaCompleta	Verdadero/Falso

Ejercicio 1:

Indica cual sería el dominio de cada uno de los siguientes atributos de la entidad PERSONA:

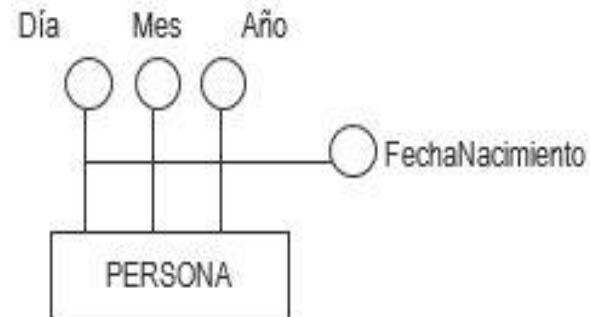
- *Fecha de nacimiento*
- *Localidad de nacimiento*
- *Edad*
- *EsMayorDeEdad*
- *DNI*
- *Teléfono*
- *Nombre*
- *Apellidos*

*¿Qué atributos de los anteriores consideras innecesarios?
¿Por qué?*

2.2 Atributos y tipos

Tipos de atributos:

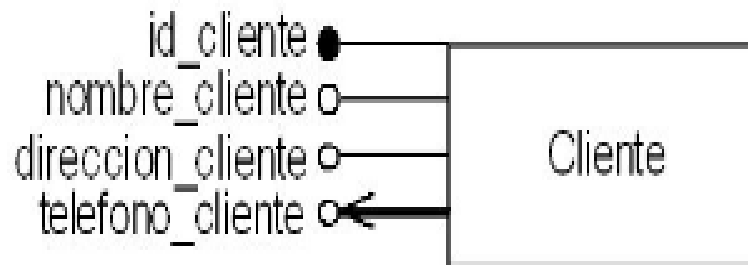
Atributos simples y atributos compuestos: Un atributo es simple si su contenido no se considera dividido en partes, por ejemplo NOMBRE. Es compuesto si admite dividirse en partes. Por ejemplo, FECHA podría ser compuesto si se considera que de FECHA se puede usar aisladamente DÍA, MES y AÑO.



2.2 Atributos y tipos

Tipos de atributos:

Atributos monovaluados y atributos multivaluados: Un atributo es monovaluado si admite para cada elemento de la entidad un solo valor, por ejemplo nombre de una persona sería monovaluado. Si un atributo admite una lista de valores para cada elemento, sería multivaluado, por ejemplo si un atributo de la entidad CLIENTE fuese teléfono_cliente, éste podría ser atributo multivaluado.



2.2 Atributos y tipos

Tipos de atributos:

Atributos obligatorios y atributos opcionales: Un atributo es obligatorio si para todo elemento debe contener algún valor y es opcional si puede haber elementos que no tengan asignado ningún valor para ese atributo. Por ejemplo, el atributo **Aficiones** podría ser opcional para una entidad **CLIENTE**.

Un atributo opcional se representa:



Anota 4 atributos (entre obligatorios y opcionales) de la entidad COCHE de un concesionario, e indica cuáles son los obligatorios y cuáles los opcionales.

2.2 Atributos y tipos

Atributos derivados y no derivados: Un atributo es derivado si se puede obtener a partir de los datos contenidos en otros atributos y es no derivado si su valor no depende de ningún otro atributo. Un atributo derivado podría ser IMPORTE DE VENTA si los valores para ese atributo se obtuviesen a través de los atributos UNIDADES VENDIDAS y PRECIO UNIDAD.

No es recomendable usar atributos derivados.

Piensa en otros ejemplos de atributos derivados (que no sean fecha de nacimiento y edad o esMayorDeEdad).

2.2 Atributos y tipos

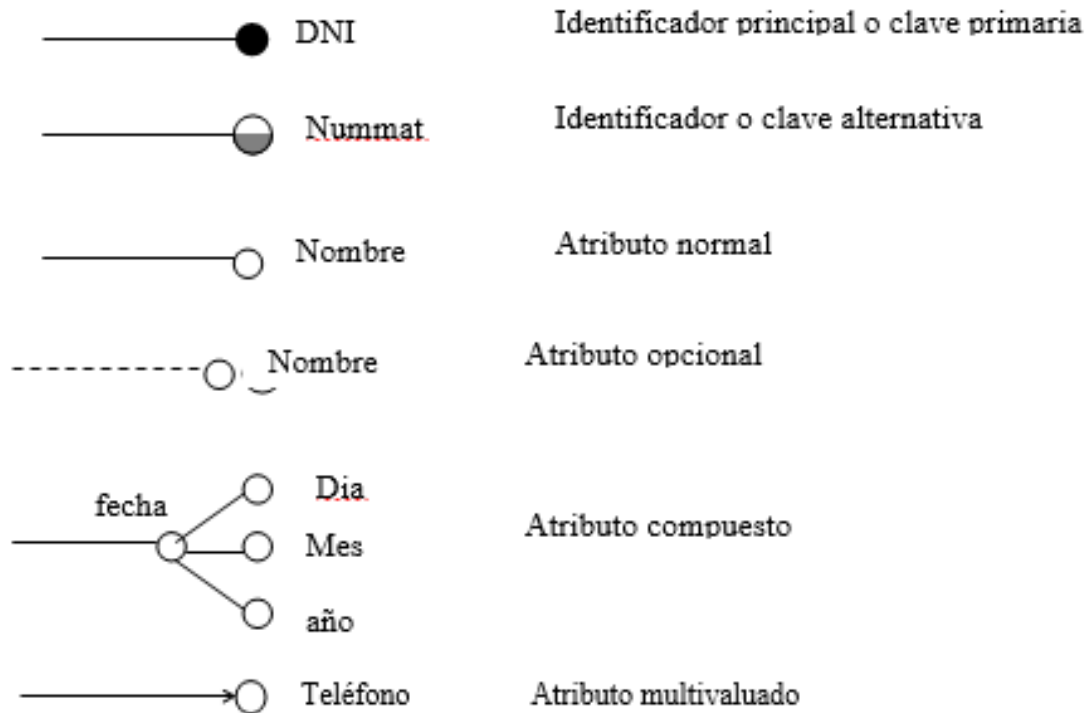
Clave : Una clave sirve para identificar de forma única a cada elemento de una entidad. Una clave puede estar formada por un solo atributo o por varios. En una clave no se pueden repetir valores, es decir, no puede haber dos elementos de la misma entidad con la misma clave. En una entidad puede haber dos tipos de clave:

- **Clave primaria o principal:** Dentro de los conjuntos de atributos que pueden permitir identificar a los elementos de una entidad, debería ser la que se considera más adecuada en base a una serie de requisitos: **simplicidad, longitud, representatividad, estabilidad**.
- **Clave secundaria o alternativa.** Puede haber varias en una entidad pero no se debe abusar de estas claves. Serán todas aquellas que decidamos, aparte de la primaria.

Anota al menos 3 claves para cada una de las siguientes entidades (PERSONA, COCHE y TITULACIÓN). Ordénalas en base a los requisitos descritos con anterioridad.

2.2 Atributos y tipos

Representación de los atributos:



Ejercicio 2:

Justifica si los siguientes atributos serían obligatorio-opcional, compuesto-simple, derivado-no derivado, monovaluado-multivaluado.

- *Fecha de nacimiento*
- *Localidad de nacimiento*
- *Edad*
- *EsMayorDeEdad*
- *DNI*
- *Teléfonos*
- *Nombre*
- *Apellidos*

2.3 Relaciones

Relación:

Es una asociación entre varias entidades a través de una acción realizable entre esas entidades.

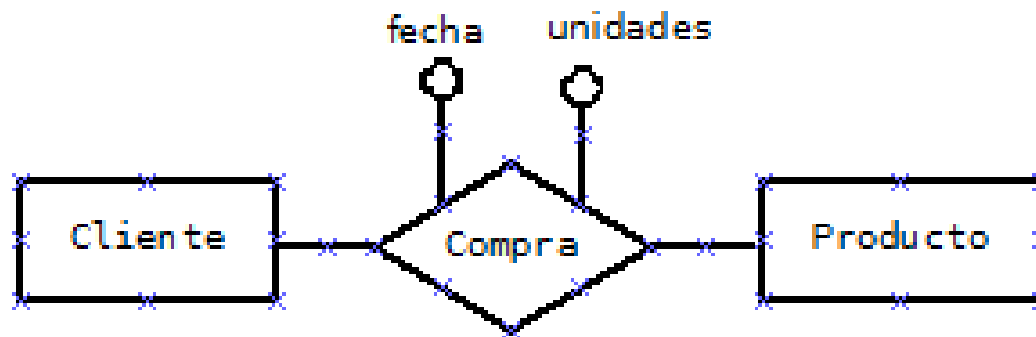
- **COMPRAR** (entre **CLIENTE** y **PRODUCTO**)
- **CURSAR** (entre **ALUMNO** y **MODULO**)
- **SER_HIJO** (entre **ALUMNO** y **PADRE**).
- **SER_JEFE** (**EMPLEADO** consigo misma)
- **COMPRAR** (entre las entidades **CLIENTE**, **PRODUCTO**, **VENDEDOR**)

Anota diferentes ejemplos de relaciones que se pueden dar entre las entidades de un concesionario de coches.

2.3 Relaciones

Relación binaria o de grado dos :

Cuando se da entre dos entidades.



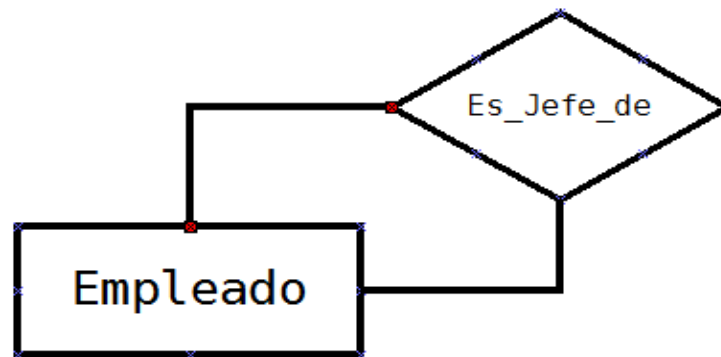
Vemos aquí que las relaciones también pueden tener atributos.

¿Puede haber relaciones que no impliquen a dos entidades?

2.3 Relaciones

Relación unaria, reflexiva o de grado uno:

Cuando se da entre elementos de la misma entidad, es decir, un elemento de una entidad se relaciona con uno o más elementos de la misma entidad.

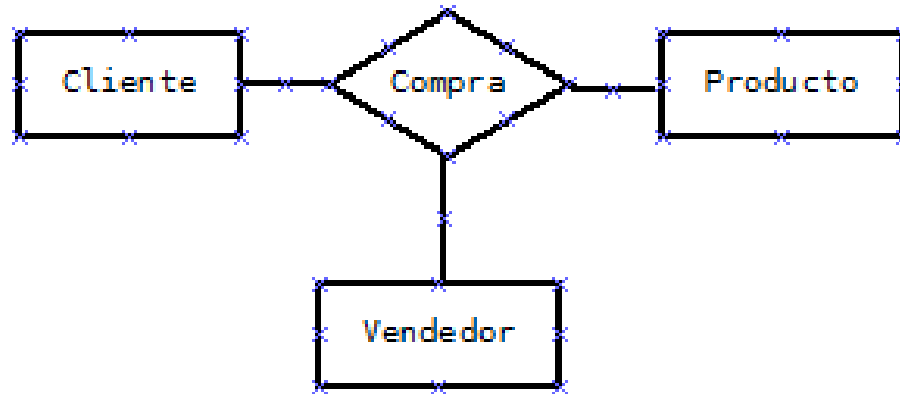


Piensa en varios ejemplos de relaciones unarias.

2.3 Relaciones

Relación ternaria o de grado tres:

Cuando se da entre tres entidades



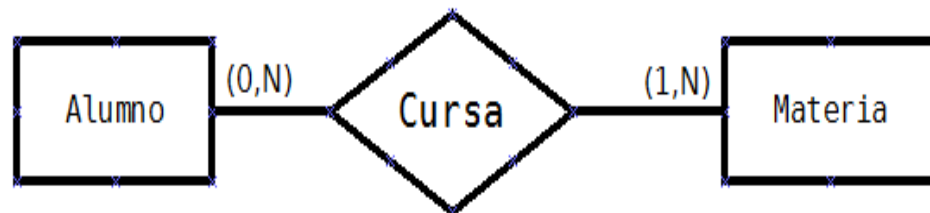
2.4 Cardinalidad

Ocurrencia: Es una unidad del conjunto de elementos que representa una entidad. Para la entidad ALUMNO, una ocurrencia de ALUMNO es un alumno concreto.

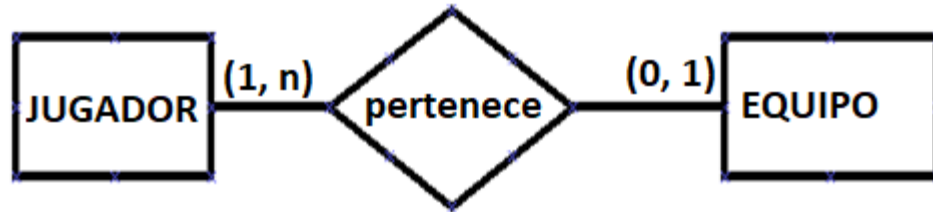
La cardinalidad de una entidad A respecto de otra B en una relación, indica el número mínimo y máximo de ocurrencias de la entidad A que pueden estar relacionadas con una ocurrencia de la entidad B.

La **cardinalidad** se indica mediante una pareja números encerrados entre paréntesis. El primer número indica el mínimo número de ocurrencias relacionadas (será siempre un valor 0 o 1). El segundo número indica el máximo número de ocurrencias relacionadas (será siempre un valor 1 o N para muchos).

La cardinalidad de una entidad se representa al lado de la entidad con la que se relaciona. Es decir, al lado opuesto de la relación.



2.4 Cardinalidad



Interpretación:

Un jugador pertenece como mínimo a ningún equipo (0) y como máximo a un equipo (1).

A un equipo pertenece como mínimo un jugador (1) y como máximo varios (n).

2.4 Cardinalidad

Las cardinalidades que se pueden dar en las relaciones son:

Cardinalidad	Significado
(0,1)	Mínimo cero, máximo uno
(1,1)	Mínimo uno, máximo uno
(0,N)	Mínimo cero, máximo muchos
(1,N)	Mínimo uno, máximo muchos

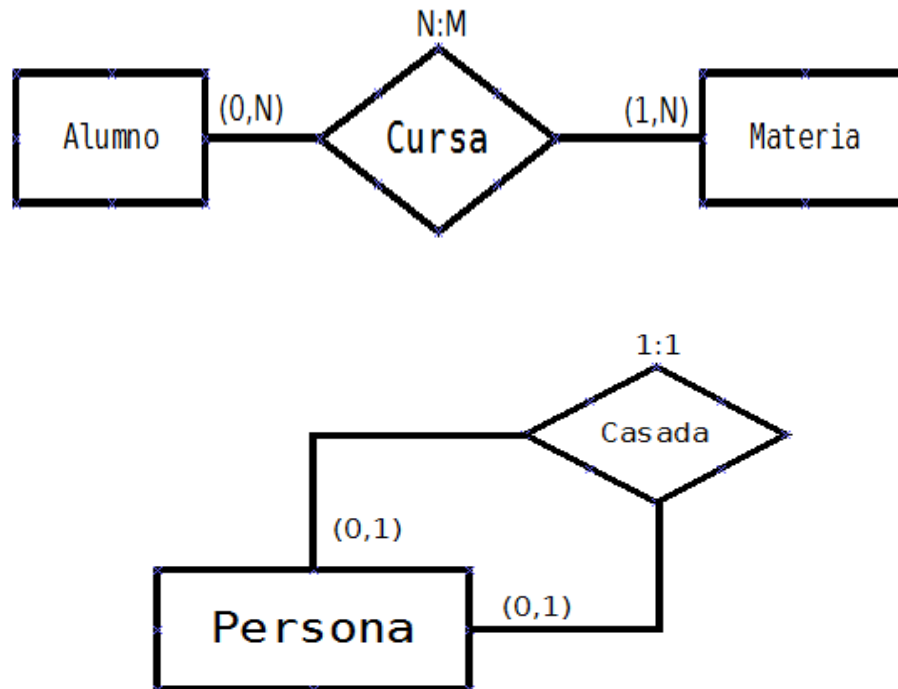
2.5 Tipo de correspondencia

Tipo de correspondencia o relación de cardinalidad: Expresa el número máximo de elementos u ocurrencias que se pueden llegar a relacionar entre las entidades de una relación.

- **Uno a uno (1:1):** Sería el caso de la relación CASADO entre las entidades PERSONA y PERSONA. Un persona podrá estar casada con otra persona pero no con muchas.
- **Uno a muchos (1:N):** Sería el caso de la relación PERTENECE entre las entidades MUNICIPIO y PROVINCIA. Un municipio sólo puede pertenecer a una provincia, mientras que a una provincia pertenecen muchos municipios.
- **Muchos a muchos (N:M):** Sería el caso de la relación COMPRA entre las entidades PRODUCTOS y CLIENTES. Un cliente puede comprar varios productos y un mismo tipo de producto será comprado por varios clientes.

2.5 Tipo de correspondencia

Representación de cardinalidad y tipo de correspondencia



2.5 Tipo de correspondencia

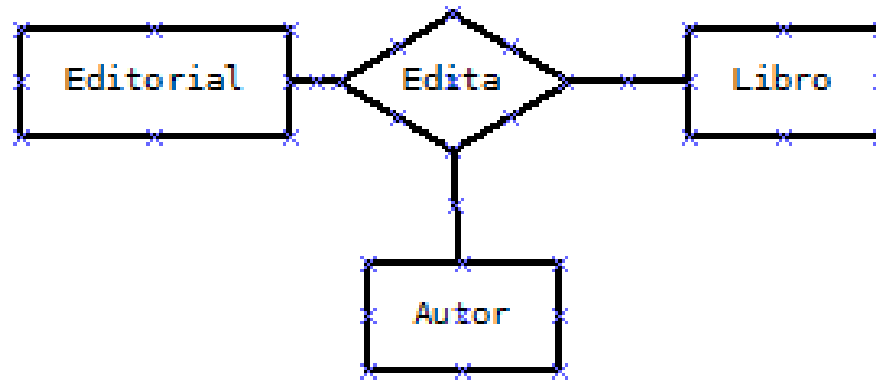
Ejercicio 1:

En un supermercado hay productos organizados en categorías. Cada producto pertenece a una única categoría. Están previstas categorías que aún pueden no tener productos. Calcula las cardinalidades de cada entidad y el tipo de correspondencia y represéntalos en el esquema E/R.

2.5 Tipo de correspondencia

Ejercicio 2:

Obtén la cardinalidad de cada una de las entidades en la siguiente relación:



Entidades fuertes

Son aquellas que tienen existencia por sí mismas, es decir, su existencia **no depende** de la existencia de otras entidades. Por ejemplo, en una base de datos hospitalaria, la existencia de instancias concretas de la entidad DOCTOR no depende de la existencia de instancias u objetos de la entidad PACIENTE.

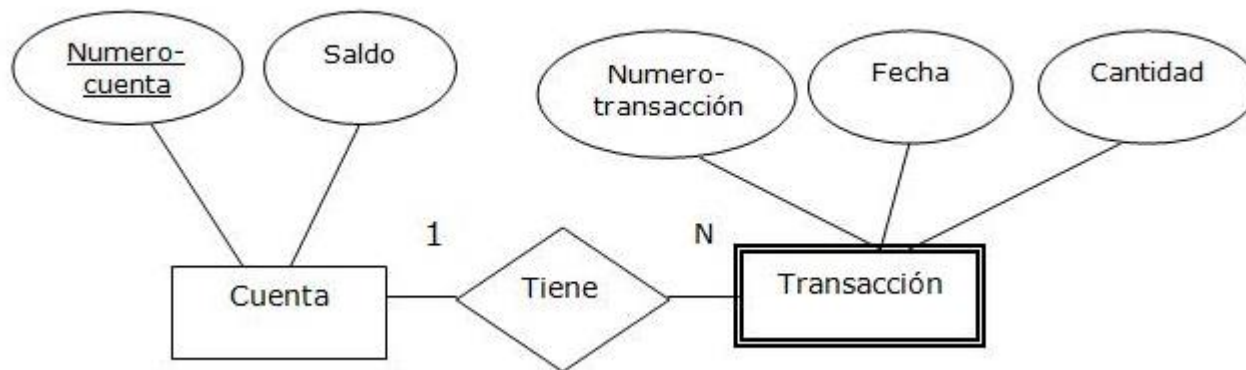
Entidades débiles

Son aquellas cuya existencia **depende** de la existencia de otras instancias de entidad. Por ejemplo, consideremos las entidades PACIENTE y FAMILIAR. La existencia de una instancia concreta de la entidad FAMILIAR depende de la existencia de una instancia de la entidad PACIENTE.

2.6 Debilidad

Por ejemplo, en la gestión de la operativa bancaria, se pueden realizar transacciones asociadas a una cuenta. Si CUENTA es una entidad y TRANSACCION es otra entidad, CUENTA sería entidad fuerte y TRANSACCION una entidad débil, respecto de la anterior.

¿Por qué?



En el modelo E/R una entidad débil se representa con el nombre de la entidad encerrado en un rectángulo doble.

2.6 Debilidad

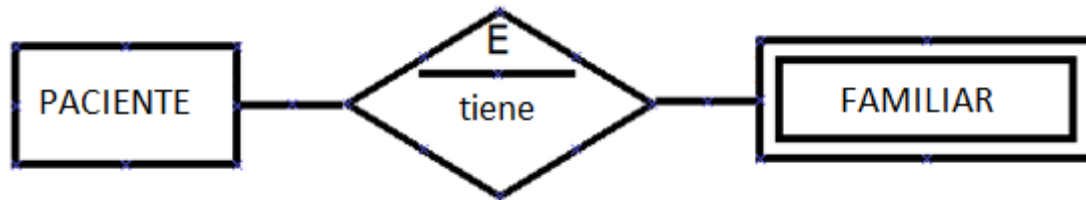
Una entidad es débil frente a otra que es fuerte, cuando para que exista un elemento de la débil es necesario que exista un elemento de la fuerte.



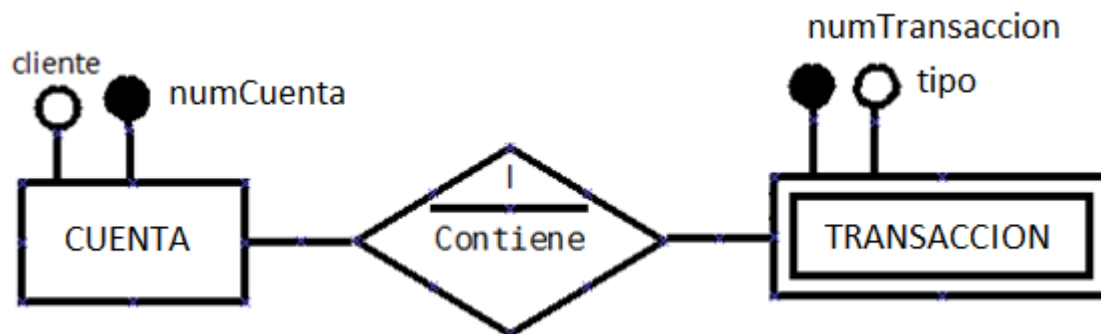
2.6 Debilidad

Dependencias en relaciones de debilidad:

Dependencia en existencia: Las ocurrencias de una entidad débil no tienen ningún sentido en la base de datos sin una ocurrencia de la entidad fuerte con la que están relacionadas.



Dependencia en identificación: Además de la dependencia en existencia, la entidad débil necesita a la fuerte para poder crear una clave a partir de la clave que tiene la entidad fuerte. Es decir, en el ejemplo, cada transacción se identificaría con numTransaccion y numCuenta.



3.- El modelo E/R ampliado

Modelo E/R ampliado: Recoge todos los conceptos y especificaciones del modelo E/R y añade otros para mejorar el diseño de las bases de datos.

- **Superclase:** Es una entidad genérica de la que derivan otras entidades. La superclase tiene unos atributos que van a tener también las entidades que derivan de ellas.
- **Subclase:** Es una entidad que deriva de una entidad genérica o superclase. La subclase va a tener los atributos de la superclase más unos atributos específicos.

Por ejemplo, EMPLEADO sería una superclase y OPERARIO y ENCARGADO serían subclases de ésta. En un centro de estudios PERSONA podría ser una superclase mientras ALUMNO y PROFESOR serían subclases.

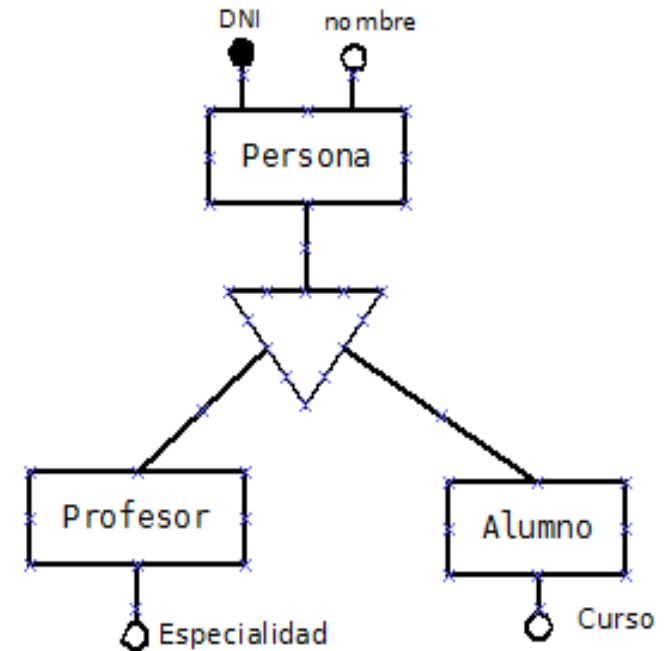
Anota diferentes ejemplos de pares de superclase-subclases.

3.- El modelo E/R ampliado

Generalización es el proceso de construir una superclase a partir de las características comunes o que comparten varias subclases del sistema de información.

Una generalización se representa mediante un triángulo invertido que une la superclase y las subclases.

Piensa y anota diferentes atributos para cada una de las 3 clases mostradas.



3.- El modelo E/R ampliado

La especialización es el proceso inverso a la generalización. En la especialización se trata de buscar:

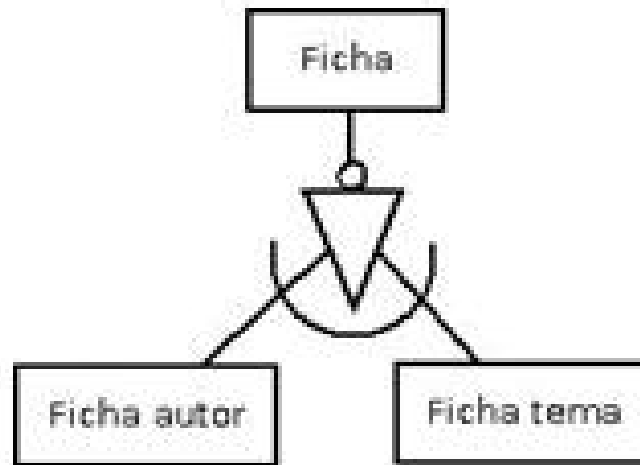
- Los atributos específicos de las subclases.
- Las restricciones de existencia de elementos de las entidades.

Conforme a las restricciones de existencia de elementos de las entidades, nos podemos encontrar con los siguientes tipos de especialización o generalización:

- ☐ **Especialización o generalización exclusiva total**
- ☐ **Especialización o generalización exclusiva parcial**
- ☐ **Especialización o generalización solapada total**
- ☐ **Especialización o generalización solapada parcial**

3.- El modelo E/R ampliado

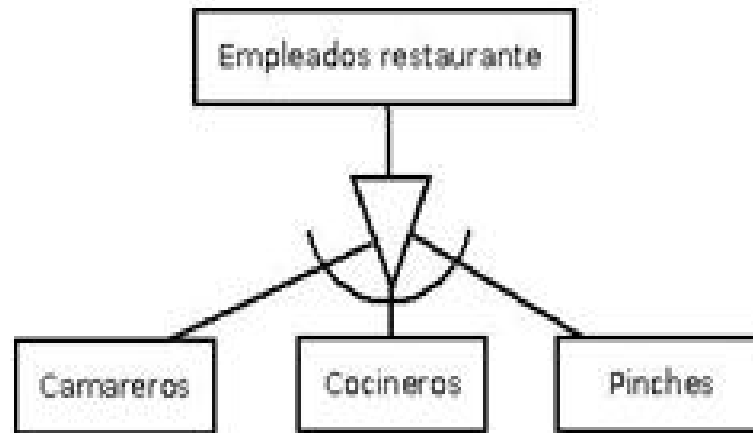
Especialización exclusiva total: Por ser exclusiva, un elemento de la superclase sólo puede estar en una subclase. Por ser total, todos los elementos de la superclase están en alguna de las subclases.



Piensa en diferentes ejemplos de relaciones donde esté presente la especialización exclusiva total.

3.- El modelo E/R ampliado

Especialización exclusiva parcial: Por ser exclusiva, un elemento de la superclase sólo puede estar en una subclase. Por ser parcial, no tienen porque estar todos los elementos de la superclase en alguna de las subclases (gerente, mesero, barman,...).



Piensa en diferentes ejemplos de relaciones donde esté presente la especialización exclusiva parcial.

3.- El modelo E/R ampliado

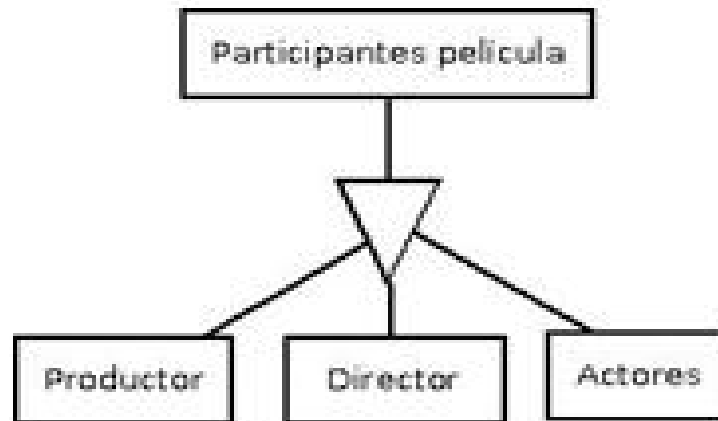
Especialización solapada total: Por ser solapada, un elemento de la superclase podría pertenecer a varias subclases. Por ser total, todos los elementos de la superclase están en alguna de las subclases.



Piensa en diferentes ejemplos de relaciones donde esté presente la especialización solapada total.

3.- El modelo E/R ampliado

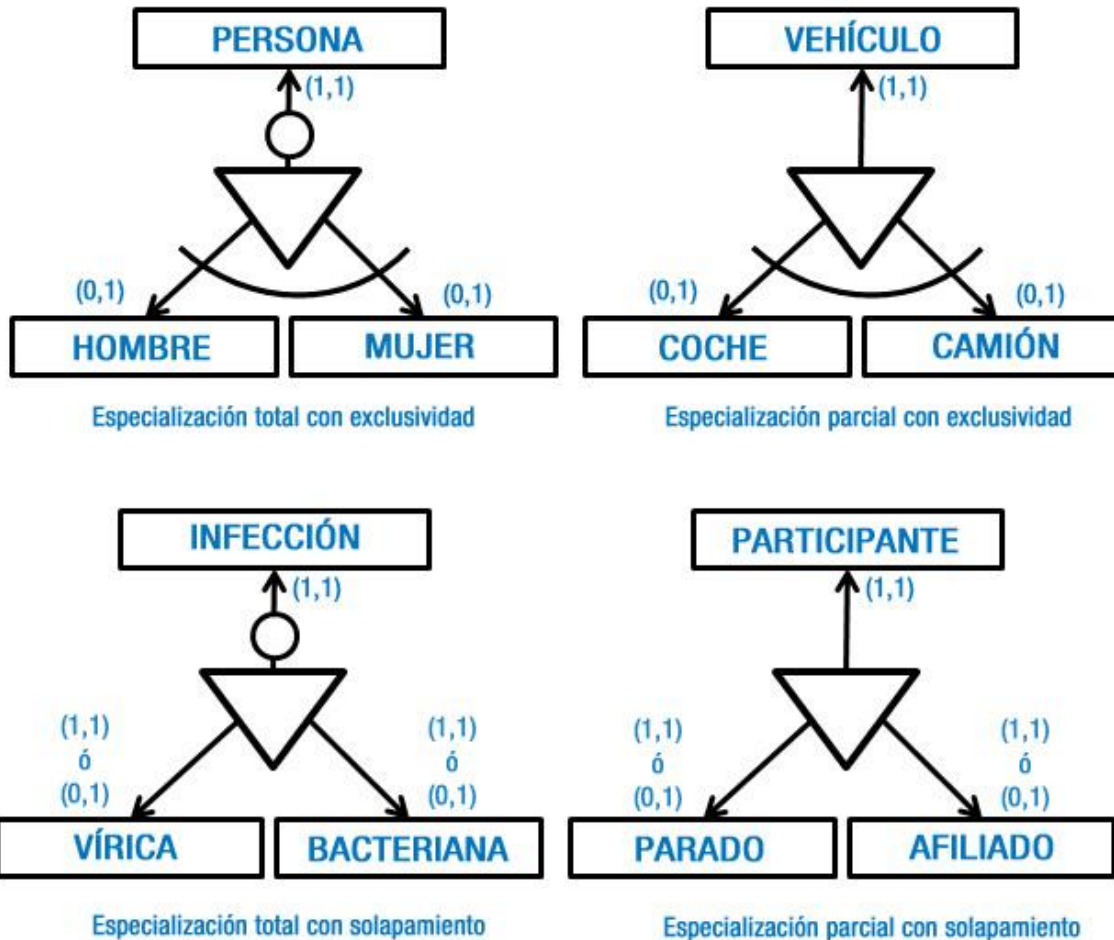
Especialización solapada parcial: Por ser solapada, un elemento de la superclase podría pertenecer a varias subclases. Por ser parcial, no tienen porque estar todos los elementos de la superclase están en alguna de las subclases (dobles, maquilladores, ...).



Piensa en diferentes ejemplos de relaciones donde esté presente la especialización solapada parcial.

3.- El modelo E/R ampliado

Cardinalidades de la especialización:



4.- Construcción de un diagrama E/R

Es un proceso **metodológico**, es decir, seguiremos un método para llegar al resultado. Los pasos a seguir serán:

1. Leer el documento varias veces hasta entender bien el problema y tener clara toda la información de que disponemos.

2. Crear un listado de candidatos a entidades, relaciones y atributos:

- Identificar las entidades. Los sujetos básicos en el sistema.
- Buscar los atributos de cada entidad. Proponer la clave principal de cada uno. Establecer los tipos de atributos (compuestos, multivaluados, opcionales, derivados). Establecer sus dominios (Fecha, numero real con dos decimales, cadena de caracteres de longitud 9, V/F...)
- Identificar las generalizaciones y especializaciones (tipos de especializaciones exclusiva total, solapada parcial...)
- Identificar las relaciones de debilidad, entidades fuertes y débiles. Dependencias de existencia o de identificación.

4.- Construcción de un diagrama E/R

3. Averiguar las cardinalidades y los tipos de correspondencia en cada relación.

4. Revisar lo obtenido para:

- Eliminar entidades derivadas.
- Ver si es necesario añadir entidades a alguna relación.
- Ver si algunos atributos de una entidad se deben agrupar como atributos de una nueva entidad.

5. Realizar una distribución de las entidades y representar sus relaciones en el diagrama así como los atributos.

6. Volver a leer el problema para ver si nos hemos dejado algo. Revisar que toda la información está representada en el esquema y refinarlo si es necesario.

5. El modelo relacional

El modelo relacional es el más apropiado, en la actualidad, para representar la estructura de una base de datos. Ello es debido a:

- ☐ Es un modelo sencillo, potente y flexible para el diseño de una base de datos.
- ☐ Tiene una base matemática en el álgebra relacional. Cualquier operación sobre elementos del modelo relacional deriva de una operación del álgebra relacional.
- ☐ Su base en el álgebra relacional ha facilitado la construcción del lenguaje SQL para manipular los datos.
- ☐ La mayoría de los SGBD son relacionales, se basan en este modelo.

5.1 Características de una relación

El elemento principal del modelo relacional es la RELACIÓN.

Una relación es una tabla donde cada elemento de la relación es una fila y se le denomina **tupla de la relación** y cada propiedad, atributo o característica de los elementos es una **columna**.

city_id	city	country_id	last_update
307	Linz	9	2006-02-15 04:45:25
308	Lipetsk	80	2006-02-15 04:45:25
309	Livorno	49	2006-02-15 04:45:25
310	Ljubertsy	80	2006-02-15 04:45:25
311	Loja	28	2006-02-15 04:45:25
312	London	102	2006-02-15 04:45:25

Tupla o fila

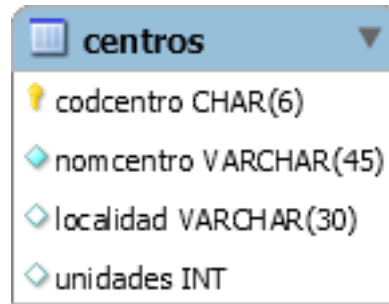
Columna, propiedad o atributo

No debes confundir el concepto de relación en el modelo relacional con el concepto de relación en el modelo E/R

5.1 Características de una relación

Al conjunto de valores que puede tomar una columna se le denomina **dominio**. Un dominio puede ser:

- ❑ **General:** si los valores pueden ser todos los existentes dentro del tipo de dato correspondiente a la columna.



- ❑ **Restringido:** si sólo puede tomar valores dentro de un rango de un dominio general, por ejemplo, números reales comprendidos entre 0 y 10.

Si tenemos un atributo de tipo Booleano y solo puede tomar los valores Verdadero y Falso, ¿de qué tipo es el dominio?

Restricciones inherentes al modelo (establecidas por el propio modelo) :

- ☐ No puede haber dos tuplas o filas que tengan el mismo contenido en todas sus columnas. ¿Por qué?
- ☐ Ninguna columna que sea clave primaria (restricción de usuario) admite nulos. ¿¿Por qué??
- ☐ Ninguna columna que sea clave primaria admite valores repetidos en las tuplas. ¿¿¿Por qué???
- ☐ Ninguna columna que sea clave alternativa admite valores repetidos en las tuplas. ¿¿¿¿Por ...

Restricciones de usuario (el modelo relacional permite que el usuario las establezca) :

- ☐ Clave primaria (PRIMARY KEY)
- ☐ Unicidad o clave alternativa (UNIQUE)
- ☐ Obligatoriedad (NOT NULL)
- ☐ Clave ajena (FOREIGN KEY)
- ☐ Verificación o chequeo (CHECK)
- ☐ Aserciones o asertos (ASSERTION)
- ☐ Disparadores (TRIGGER)



Clave primaria o principal (PRIMARY KEY)

Es un conjunto de atributos o columnas que identifican de forma única a cada tupla de una relación (a cada fila de una tabla).

Se debe declarar clave primaria en cualquier tabla, aunque no es obligatorio hacerlo.

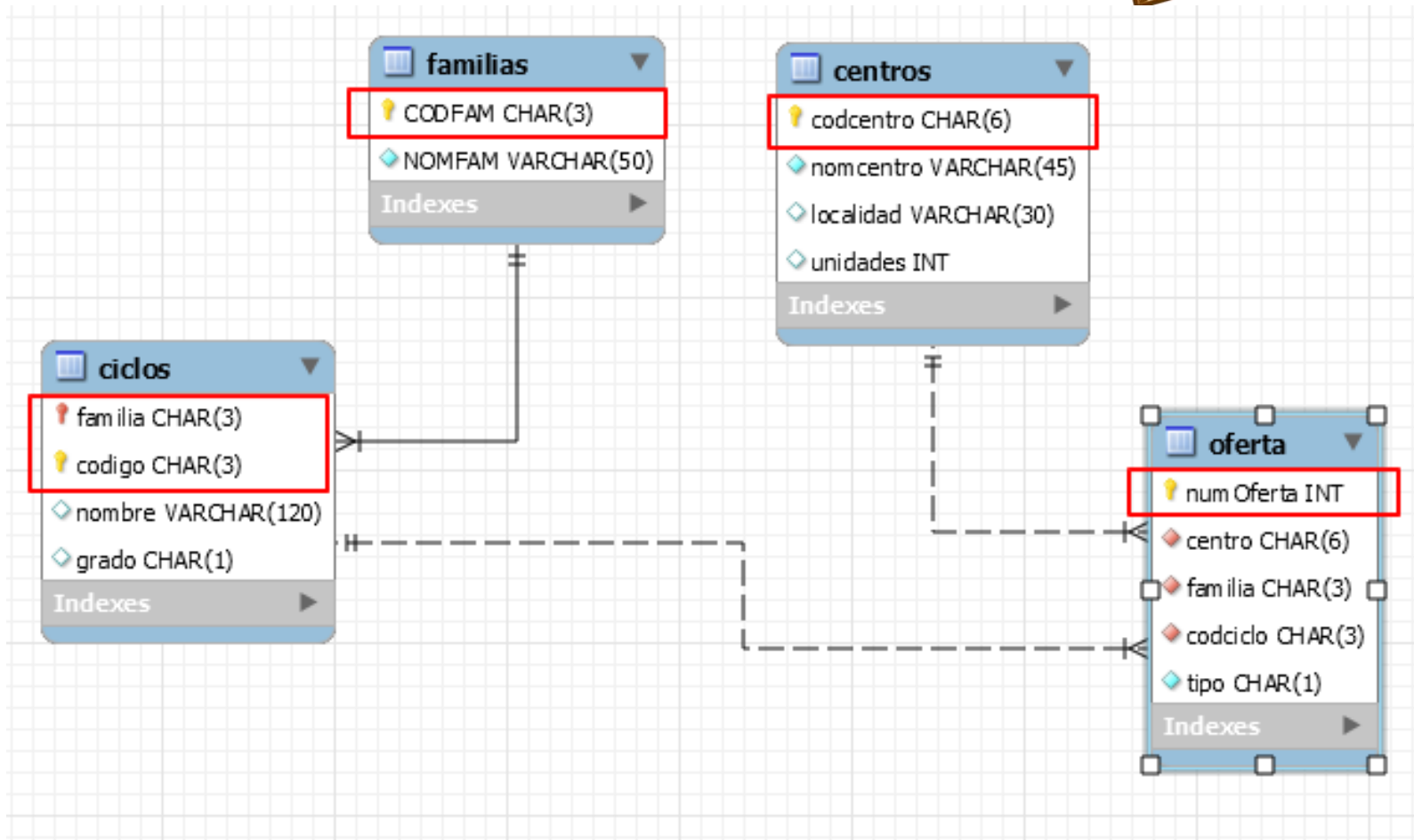
Sólo puede definirse una clave primaria en una tabla y debe ser, dentro de las columnas que puedan servir para identificar a cada tupla, la columna o el conjunto de columnas que se considere mejor para identificar de forma única a cada tupla o elemento de la tabla.

Sobre las claves primarias quedan establecidas las restricciones inherentes comentadas anteriormente.

5.3 Claves primarias y ajenas



Clave primaria o principal (PRIMARY KEY)

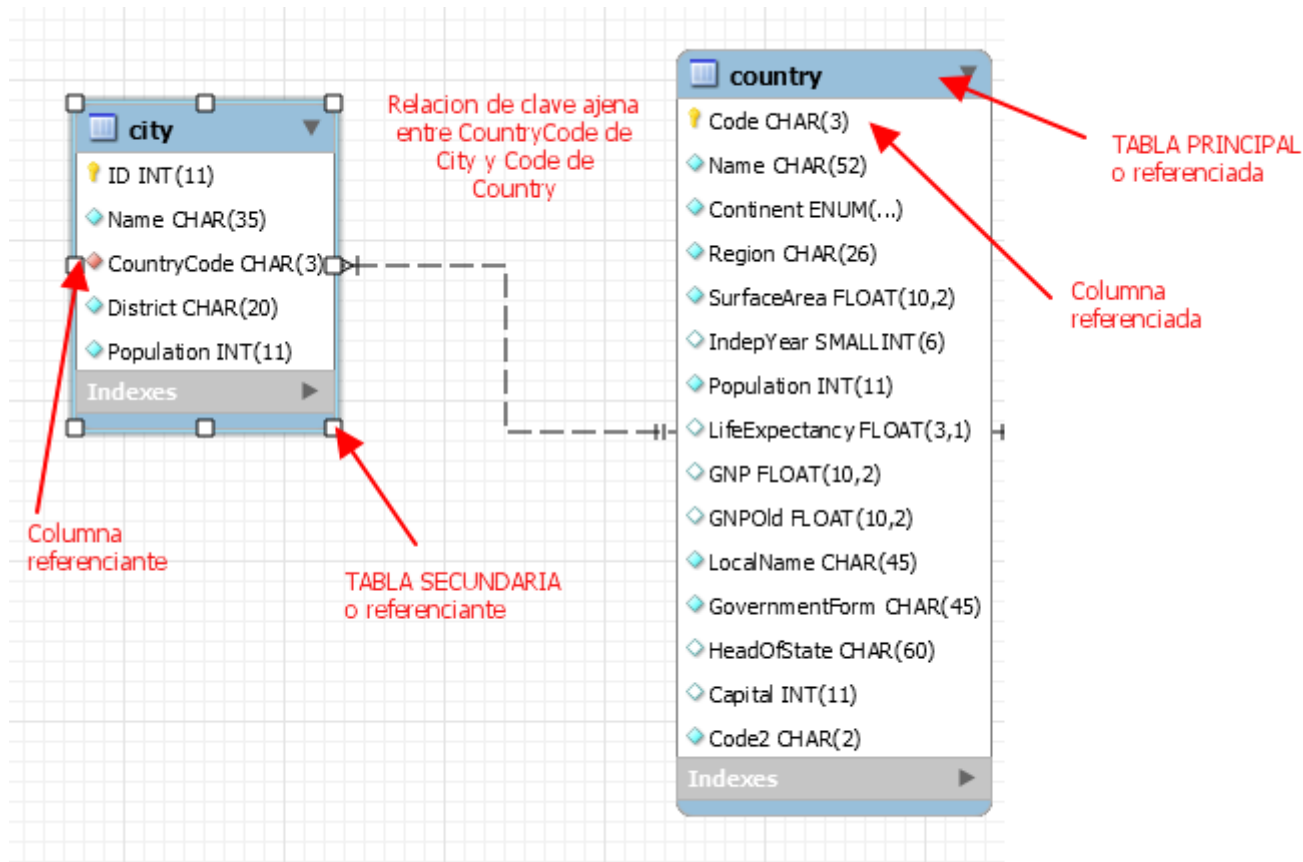


Clave ajena (FOREIGN KEY)

Sirve para indicar que uno o más atributos que forman clave ajena en una tabla (tabla secundaria en la relación o referenciante) están relacionados con uno o más atributos de otra tabla (principal en la relación o referenciada) que forman clave primaria o clave alternativa en esa otra tabla.

Por ejemplo, si tenemos una tabla **COUNTRY** que contiene datos de todos los países del mundo y una tabla **CITY** que contiene datos de ciudades del mundo, podrá haber una relación de clave ajena entre **CITY (tabla secundaria)** con **COUNTRY (tabla principal)** para controlar el país al que pertenece cada ciudad.

5.3 Claves primarias y ajenas



5.4 Integridad Referencial

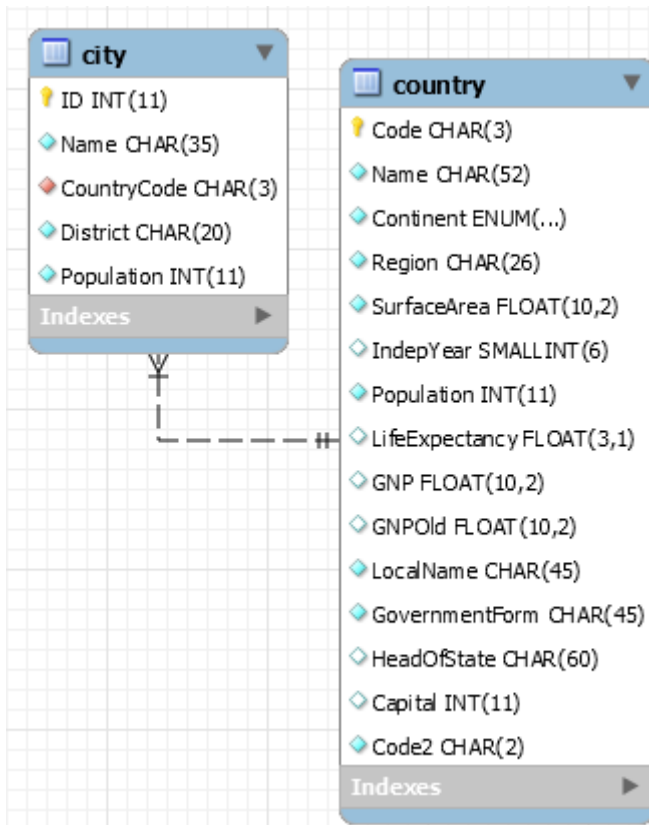
Las **restricciones de integridad referencial** son las que permiten que el SGBD controle incoherencias entre los datos cargados en la clave ajena y los datos existentes en la clave primaria de la tabla principal.

Las restricciones de integridad referencial actúan cuando:

- ☐ Se inserta una nueva fila en la tabla secundaria.
- ☐ Se modifica el valor de la clave ajena en la tabla secundaria.
- ☐ Se borra una fila en la tabla principal.
- ☐ Se modifica la clave primaria en la tabla principal.

5.4 Integridad Referencial

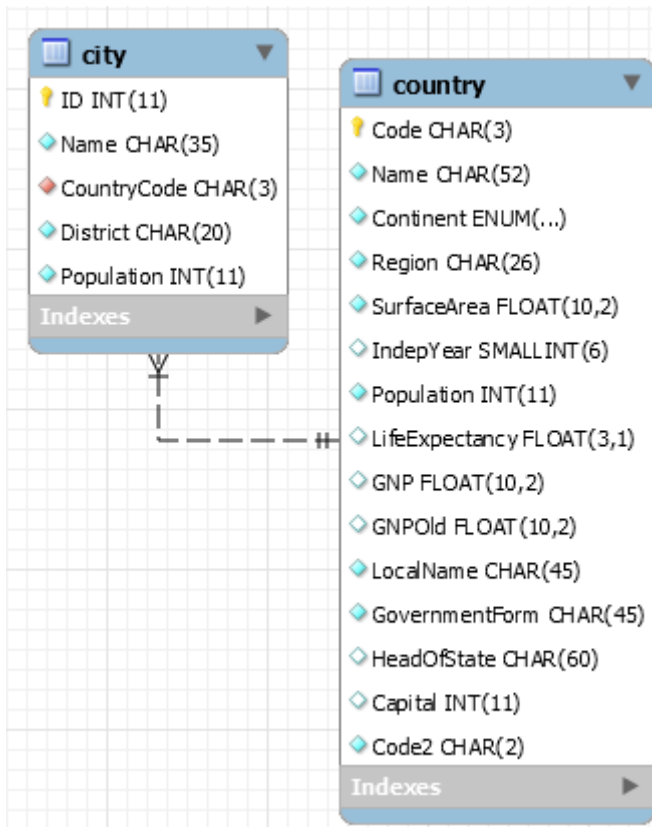
Cuando se inserta una nueva fila en la tabla secundaria.



Al insertar una nueva **city** se comprobaría que el **CountryCode** de la nueva ciudad esté cargado en **Code** de algún **Country**. Si no lo está, se rechaza la inserción.

5.4 Integridad Referencial

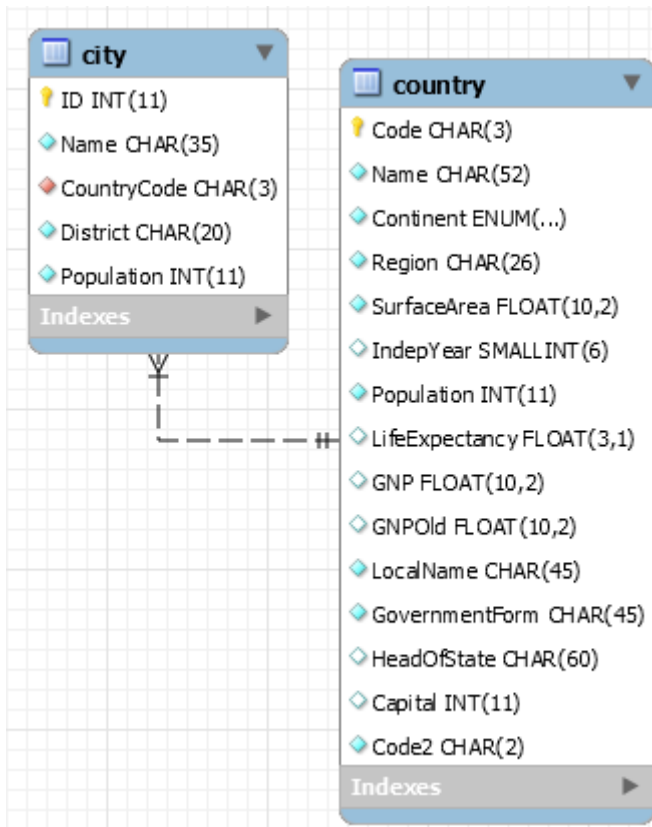
Al modificar una clave ajena en una fila en la tabla secundaria.



Al modificar el contenido de una CITY, se comprueba que el nuevo valor cargado en la clave ajena **CountryCode** exista en la clave primaria **Code** de tabla principal COUNTRY. Si no existe se rechaza la modificación y queda la fila con el valor anterior

5.4 Integridad Referencial

Al eliminar una fila en la tabla principal.



Borrado en cascada (BC): Si se elimina un país, se eliminan todas las ciudades del país.

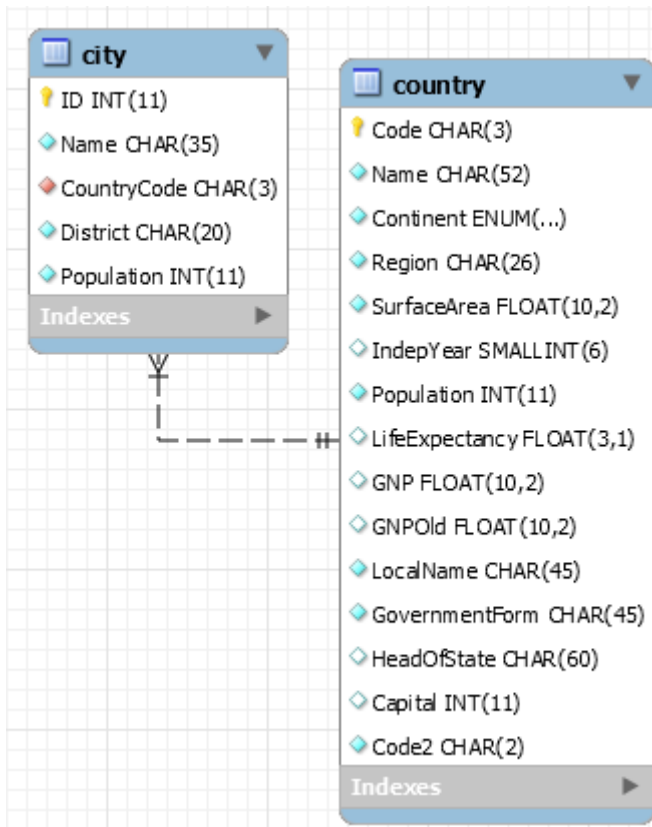
Borrado restringido (BR): Si se trata de eliminar un país y hay ciudades de ese país en la tabla CITY, no se permite la eliminación.

Borrado con puesta a nulos (BN): Si se trata de eliminar un país y hay ciudades de ese país en la tabla CITY, se elimina el país y en la columna clave ajena (countrycode) de CITY de todas las ciudades de ese país, se carga NULL.

Borrado con puesta a valor por defecto (BD): Si se trata de eliminar un país y hay ciudades de ese país en la tabla CITY, se elimina el país y en la columna clave ajena (countrycode) de CITY de todas las ciudades de ese país, se carga un valor por defecto.

5.4 Integridad Referencial

Al modificar el campo clave primaria en una fila de la tabla principal.



Modificación en cascada (MC): Si se modifica el código de un país, se modifica **countrycode** de todas las ciudades del país.

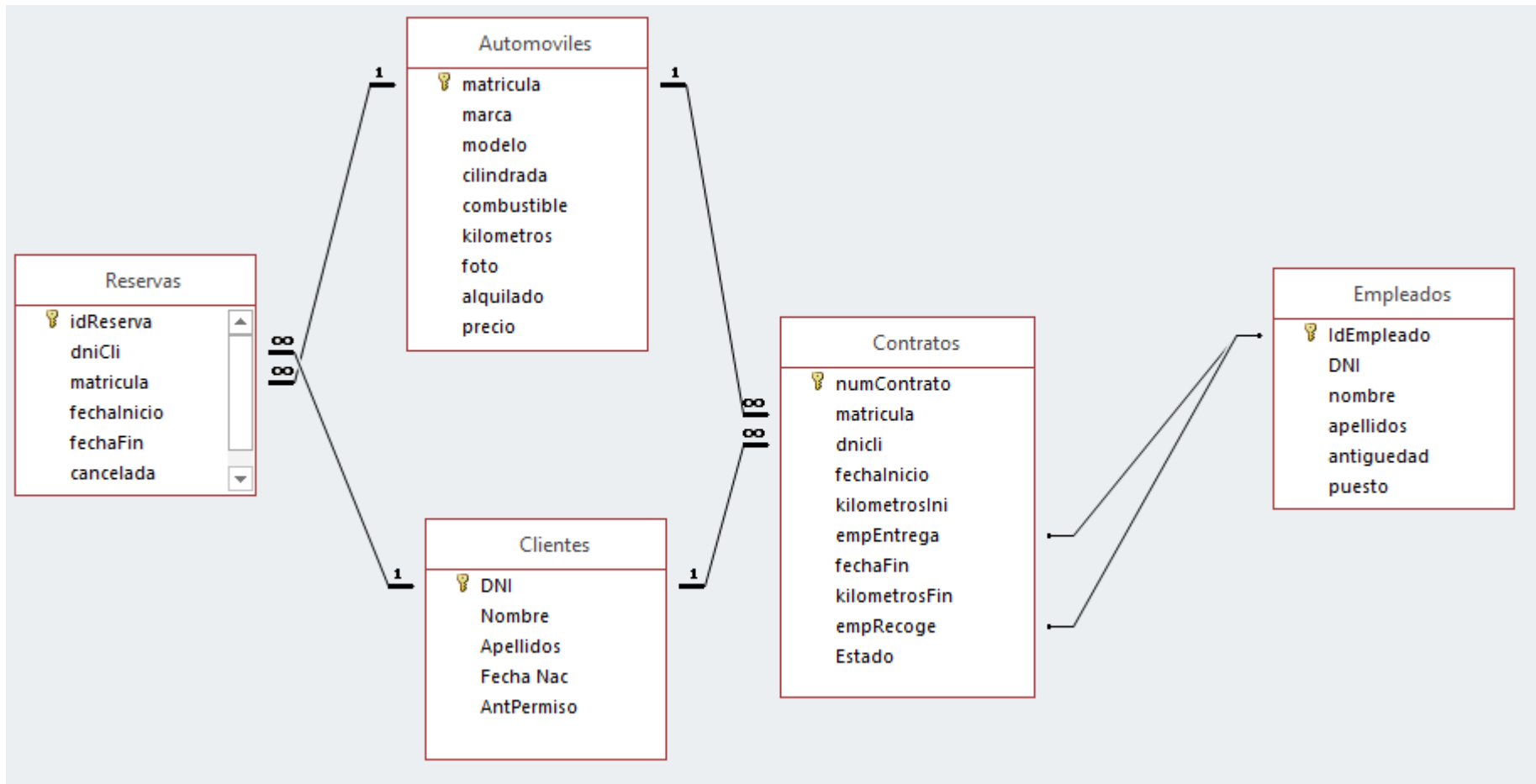
Modificación restringida (MR): Si se trata de modificar el código de un país y hay ciudades de ese país en la tabla CITY, no se permite la modificación.

Modificación con puesta a nulos (MN): Si se trata de modificar el código de un país y hay ciudades de ese país en la tabla CITY, se carga NULL en la columna clave ajena (countrycode) de CITY de todas las ciudades de ese país.

Modificación con puesta a valor por defecto (MD): Si se trata de modificar el código de un país y hay ciudades de ese país en la tabla CITY, se modifica el país y en la columna clave ajena (countrycode) de CITY de todas las ciudades de ese país, se carga un valor por defecto.

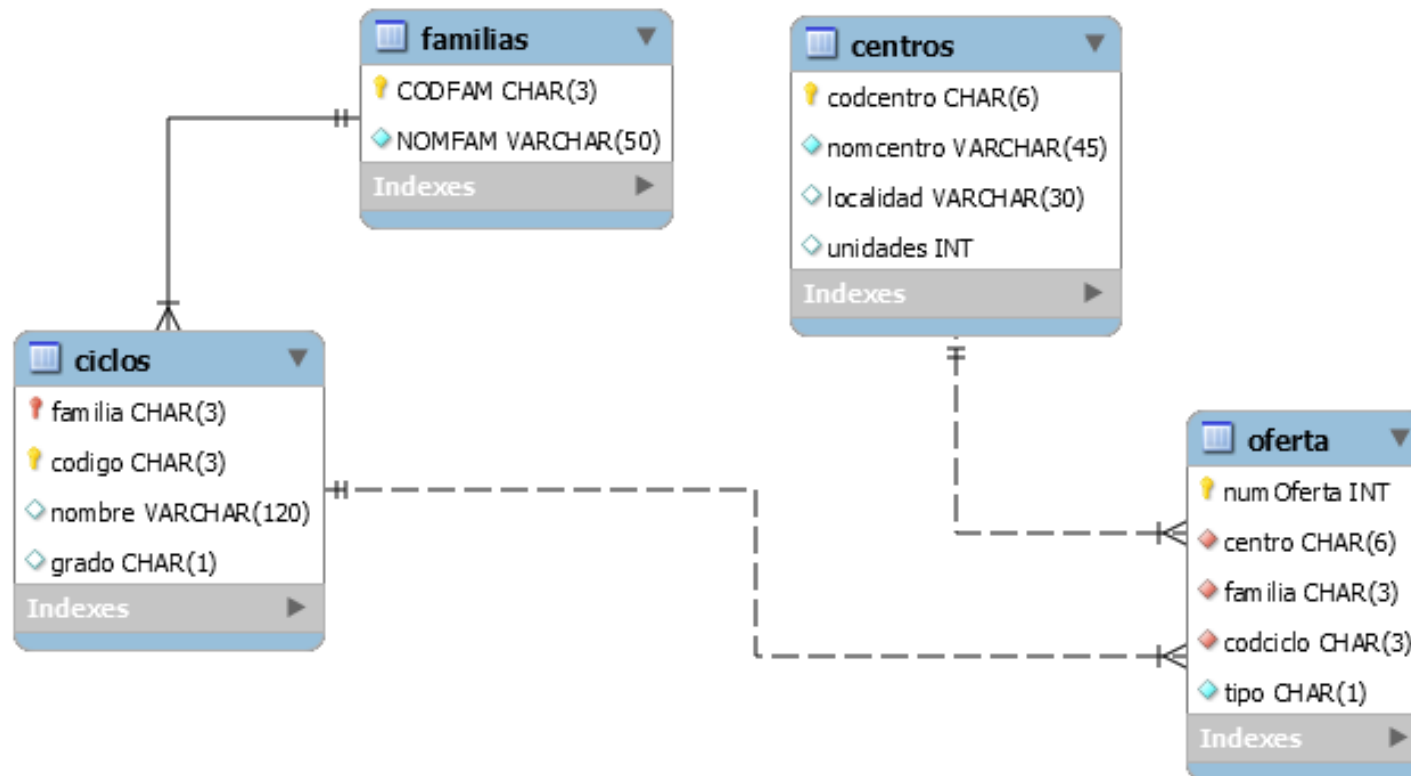
5.5 Representación del esquema relacional

Tipo de esquema 1



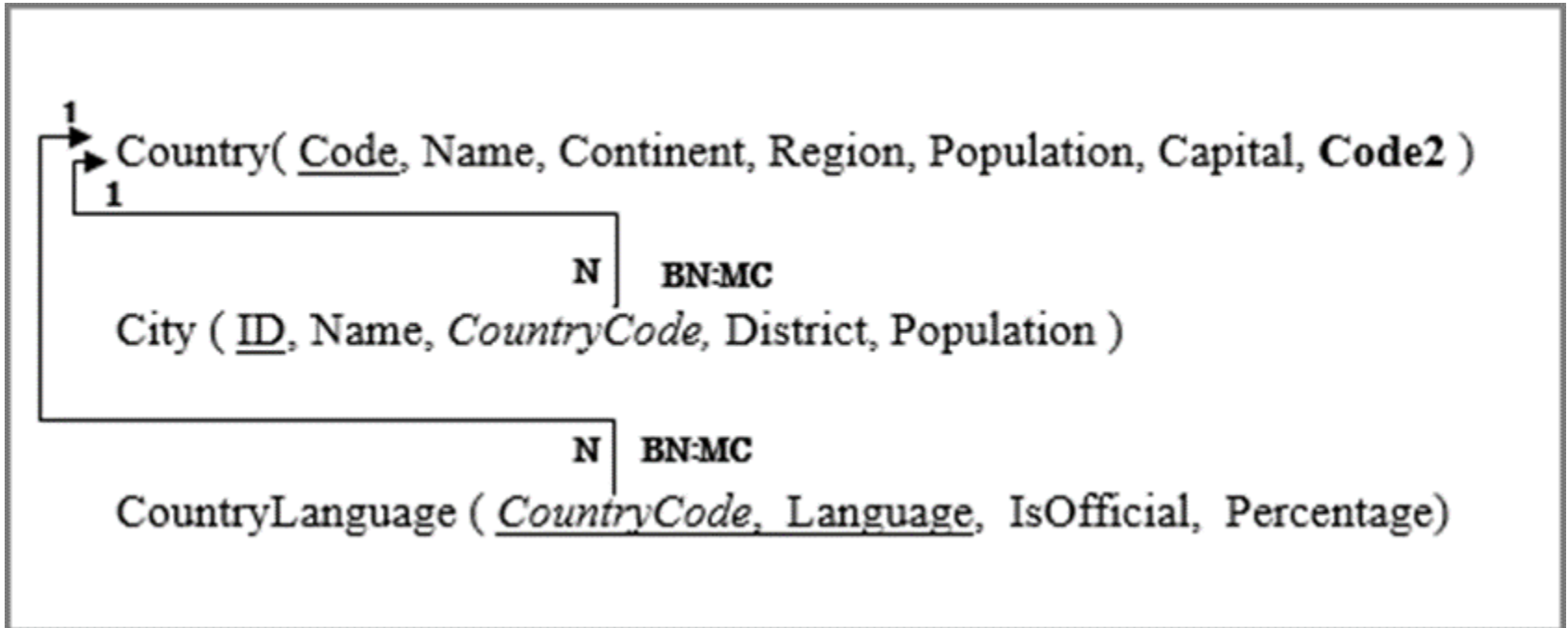
5.5 Representación del esquema relacional

Tipo de esquema 2



5.5 Representación del esquema relacional

Tipo de esquema 3



5.5 Representación del esquema relacional

Tipo de esquema 3

Se trata de un grafo donde:

Cada nodo o elemento representa **una tabla o relación** con todos sus atributos.

Se **representan las claves ajenas** a través de flechas dirigidas entre la clave ajena y la tabla donde se encuentra la clave primaria relacionada.

Cada nodo en el grafo (tabla) se representa, si es posible, con una línea de texto. Esta línea contiene, en letras mayúsculas, el nombre de la tabla y a continuación, entre paréntesis, los nombres de los atributos o columnas de la siguiente forma:

- ❑ Si un atributo es clave primaria se representa subrayado.
- ❑ Si un atributo es clave alternativa se representa en negrita. En el curso podremos representarlo doble subrayado.
- ❑ Si un atributo puede tomar valores nulos, se representa con un asterisco al final del nombre del atributo.
- ❑ Si un atributo es clave ajena se representa en letra cursiva. Lo representaremos entre comillas.

5.5 Representación del esquema relacional

Tipo de esquema 3

EMPLEADOS(numemp, nombre, apellidos, **dni**, *ncat*, salario, fecha, tlf *)

EMPLEADOS(numemp, nombre, apellidos, dni, "ncat", salario, fecha, tlf *)

Si esto es un elemento del esquema de una base de datos, representa lo siguiente:

- ☐ La línea describe una tabla llamada EMPLEADOS.
- ☐ La tabla EMPLEADOS contiene las columnas (atributos) numemp, nombre, apellidos, dni, ncat, fecha y tlf.
- ☐ La clave primaria es numemp.
- ☐ El atributo **dni** es clave alternativa.
- ☐ El atributo tlf admite valores nulos (marcado con un *).
- ☐ El atributo *ncat* es clave ajena (aunque no se ha representado con que tabla y primary key está relacionado).

5.5 Representación del esquema relacional

Tipo de esquema 3

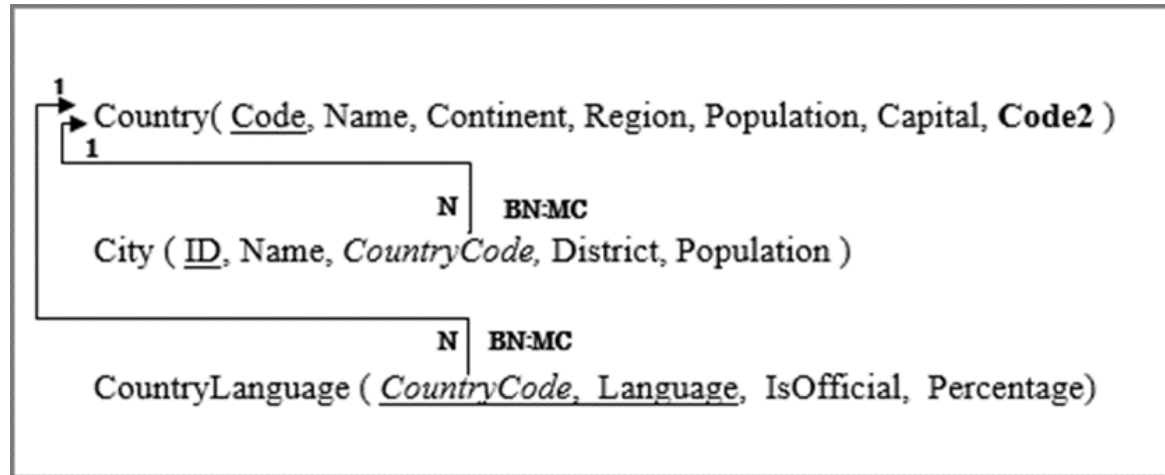
Representación de claves ajenas

- ☐ Si un atributo es clave ajena se representa en letra cursiva.
- ☐ Para representar la clave con la que está relacionada la clave ajena, se traza una flecha dirigida desde el nombre de la clave ajena hasta el nombre de la tabla que contiene la clave relacionada.
- ☐ En el origen de la flecha se deben escribir las restricciones de borrado y modificación, si es que se van a establecer.
- ☐ En la relación debe representarse la cardinalidad. Si el tipo es 1:N, se escribirá 1 al final de la flecha, es decir, en la tabla principal y se escribirá N en el origen de la flecha.
- ☐ En el modelo relacional no se permite representar cardinalidades N:M, sólo se permiten 1:1 y 1:N ¿A qué se debe esta limitación?

5.5 Representación del esquema relacional

Tipo de esquema 3

Representación de claves ajenas



- ☐ En la tabla **CountryLanguage**, la columna **CountryCode** es clave ajena y está relacionada con la primary key de la tabla **Country**.
- ☐ Por cada fila de **Country** (por cada país) puede haber muchas filas en **CountryLanguage**.
- ☐ En esa clave ajena existe una restricción de borrado con puesta a nulos, entonces cuando se borre un país, el valor **CountryCode** de **CountryLanguage** que tuviese asociado ese país se tomará el valor Null.
- ☐ En esa clave ajena también existe una restricción de modificación en cascada, por lo tanto, si se modifica el código de un país se modificará también ese código en todos los idiomas del país.

5.5 Representación del esquema relacional

Ejercicio

Representa el esquema relacional correspondiente a una base de datos sobre la red de albergues del Camino de Santiago del Norte.

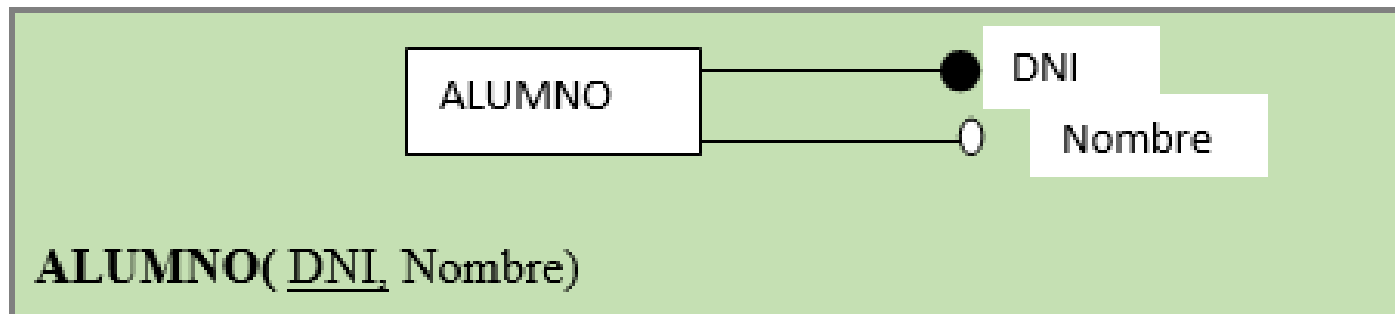
- De cada albergue se registrará su nombre, dirección, localidad y kms que faltan para el destino final (Santiago de Compostela). Existen albergues con el mismo nombre genérico (Albergue de peregrino, por ejemplo).
- Los albergues son de propiedad municipal, pertenecen a Ayuntamientos. Un determinado Ayuntamiento puede disponer de varios albergues.
- De cada Ayuntamiento debemos conocer su nombre, dirección, teléfono y URL de su web.
- En los albergues pernoctan peregrinos de los que se registra un número de tarjeta (único), su nombre y nacionalidad.
- Se debe registrar la fecha de entrada de cada peregrino en el albergue correspondiente.

UNIDAD 2: DISEÑO LÓGICO DE BASES DE DATOS

- 1. Modelo de datos**
- 2. La representación del problema: los diagramas E/R**
 - 2.1 Entidades**
 - 2.2 Atributos y tipos**
 - 2.3 Relaciones**
 - 2.4 Cardinalidad**
 - 2.5 Debilidad**
- 3. El modelo E/R ampliado.**
- 4. Construcción de un diagrama E/R**
- 5. El modelo relacional**
 - 5.1 Características de una relación.**
 - 5.2 Restricciones**
 - 5.3 Claves primarias y claves ajenas**
 - 5.4 Integridad Referencial**
 - 5.5 Representación del esquema relacional**
 - 5.6 Paso del diagrama E/R al modelo relacional.**
- 6. Normalización.**
 - 6.1 Dependencias funcionales**
 - 6.2 Formas normales**

5.6.- Paso del diagrama E/R al diagrama relacional

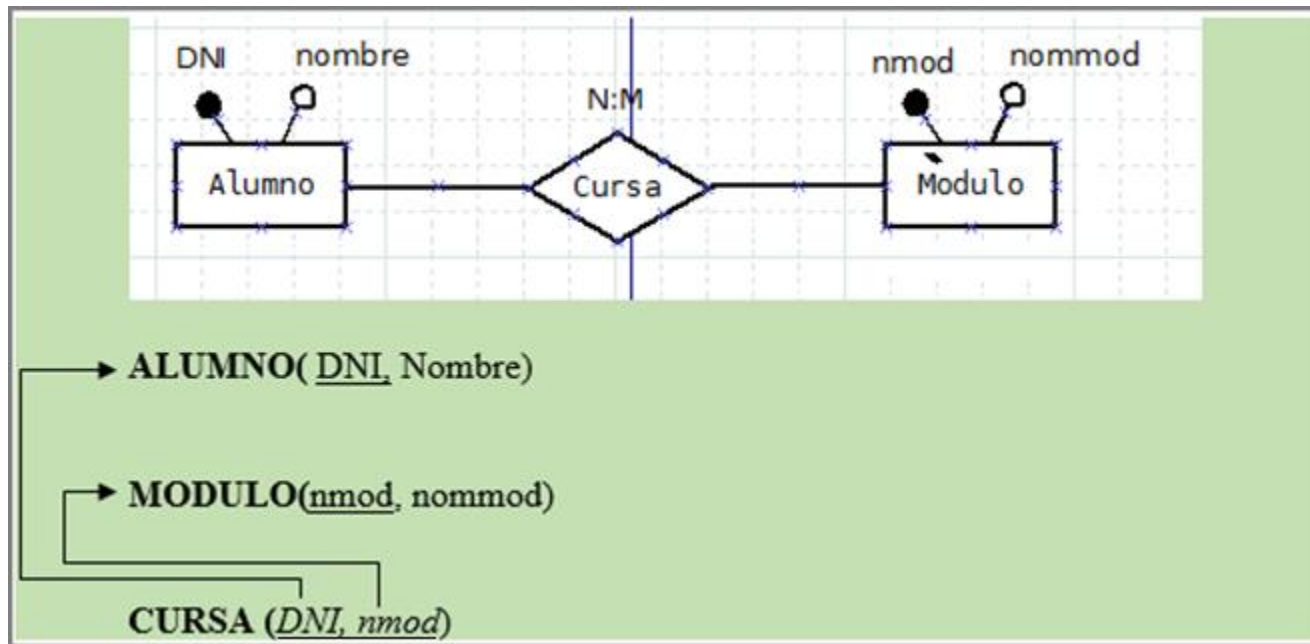
Toda entidad, sea del tipo que sea, se transforma en una tabla que contiene los mismos atributos de la entidad, excepto los multivaluados.



¿Por qué motivo no incluiremos los atributos multivaluados?

5.6.- Paso del diagrama E/R al diagrama relacional

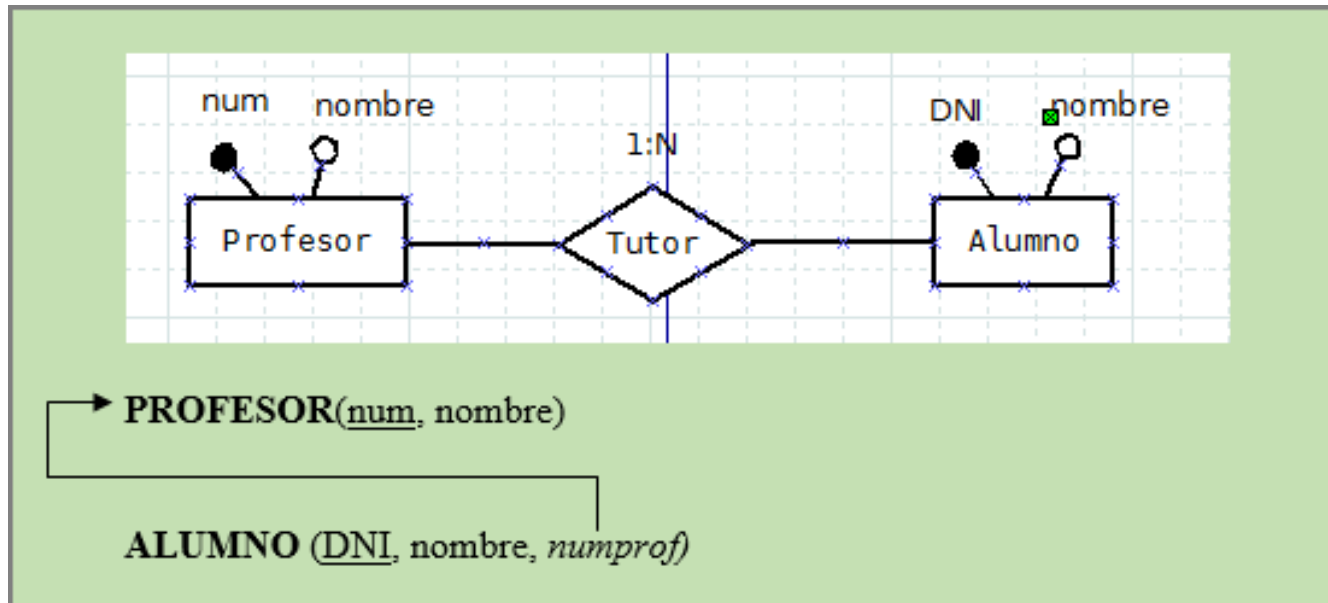
Las relaciones con tipo de correspondencia N:M entre una entidad A y una entidad B dan origen a una tabla cuya clave primaria está formada por la concatenación de las claves primarias de las tablas A y B. La tabla tendrá, además, los atributos que sean propios de la relación.



¿Por qué creamos una nueva tabla en lugar de utilizar claves foráneas?

5.6.- Paso del diagrama E/R al diagrama relacional

En una relación con cardinalidad 1:N se debe **propagar** la clave primaria de la entidad con participación máxima 1 para ser clave ajena en la tabla que tiene participación máxima N.

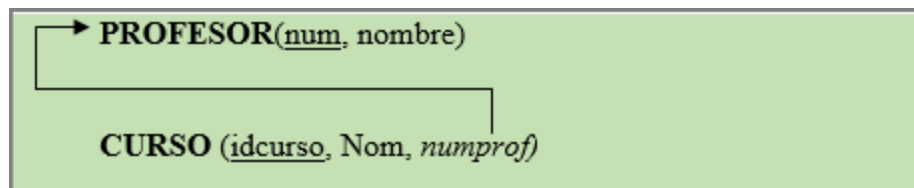
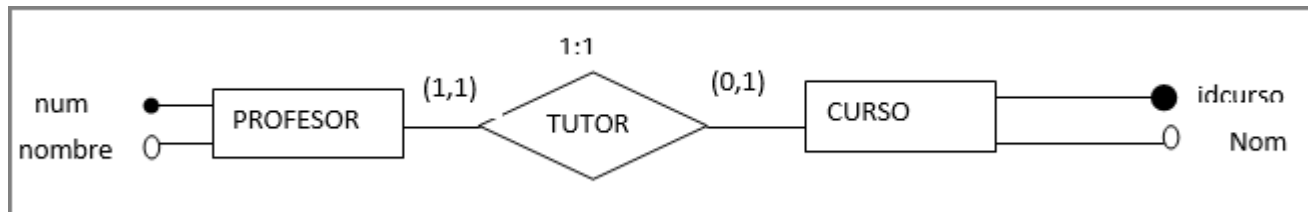


¿Podríamos propagar la clave primaria de la entidad con participación máxima N?

5.6.- Paso del diagrama E/R al diagrama relacional

Una relación con cardinalidad 1:1 entre dos entidades A y B presenta tres casos:

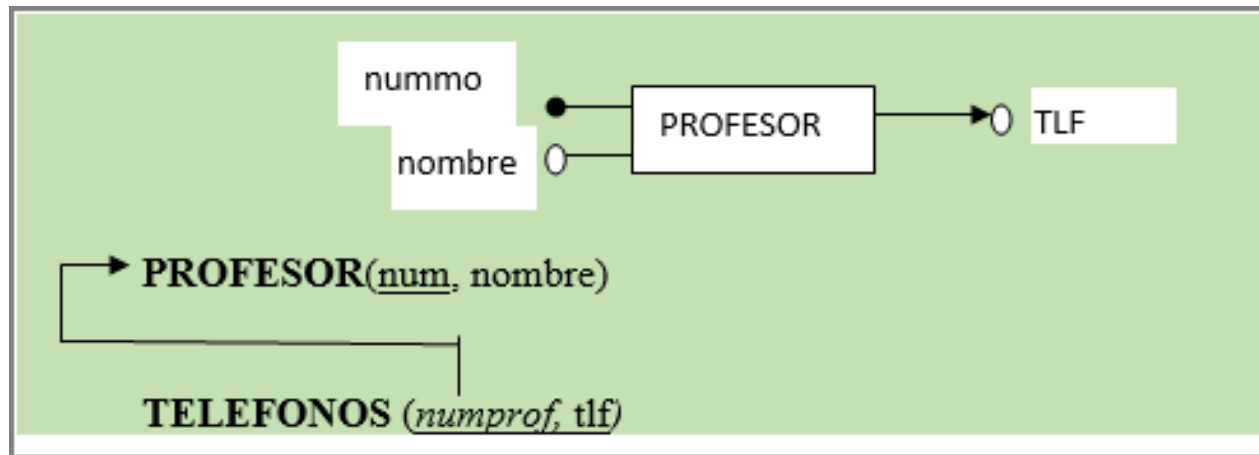
- ❑ Si la participación de A es (0,1) y la de B es (1,1), se propaga la clave primaria de B para ser clave ajena en la tabla A.



- ❑ Si la participación de A es (1,1) y la de B es (1,1) se propaga la clave primaria de cualquiera de las dos tablas como clave ajena de la otra tabla.
- ❑ Si la participación de ambas entidades es (0,1), se trata como el caso de las relaciones con tipo de correspondencia N:M.

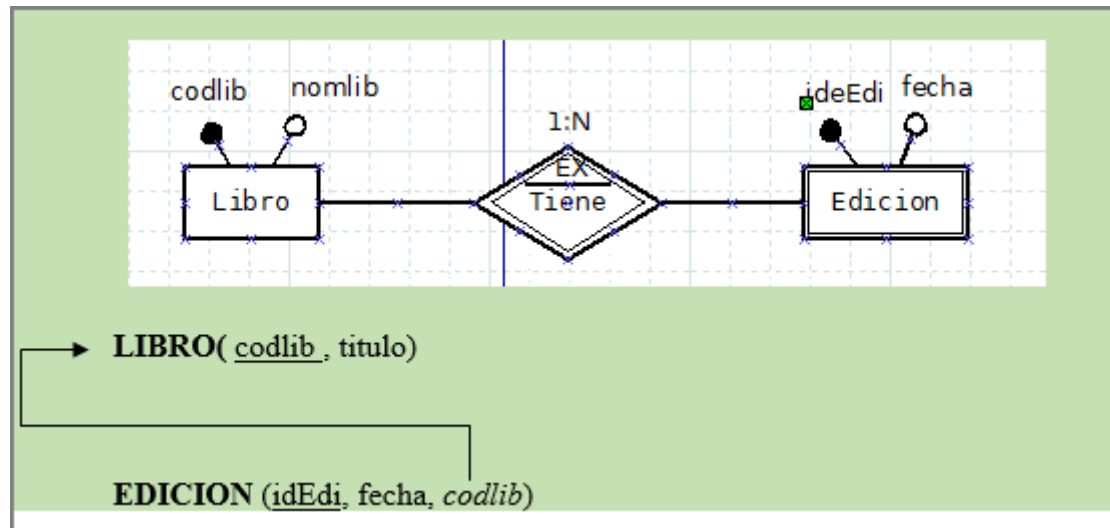
5.6.- Paso del diagrama E/R al diagrama relacional

Un **atributo multivaluado** de una entidad da lugar a una tabla formada por dos atributos: la clave de la entidad de la que forma parte y el atributo correspondiente, no siendo multivaluado en esta nueva tabla. La clave primaria será la concatenación de los dos atributos o bien un identificador nuevo elegido para esa función.



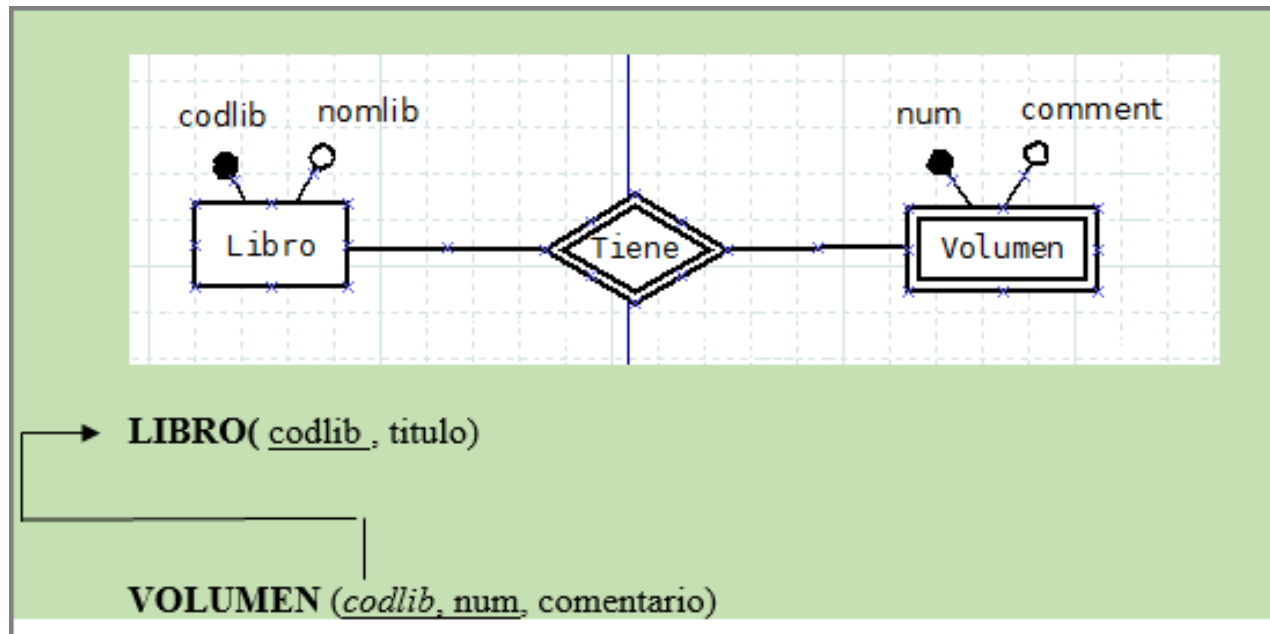
5.6.- Paso del diagrama E/R al diagrama relacional

Una relación con **dependencia en existencia** hace que se propague la clave primaria de la entidad fuerte como clave ajena en la tabla correspondiente a la entidad débil. La clave primaria en la tabla correspondiente a la entidad débil será la que se haya indicado para dicha entidad en el esquema E-R.



5.6.- Paso del diagrama E/R al diagrama relacional

Una relación con **dependencia en identificación** hace que se propague la clave primaria de la entidad fuerte como clave ajena en la tabla correspondiente a la entidad débil. La clave primaria en la tabla correspondiente a la entidad débil será la concatenación de la clave ajena propagada y el identificador de la entidad débil indicado en el esquema E-R.

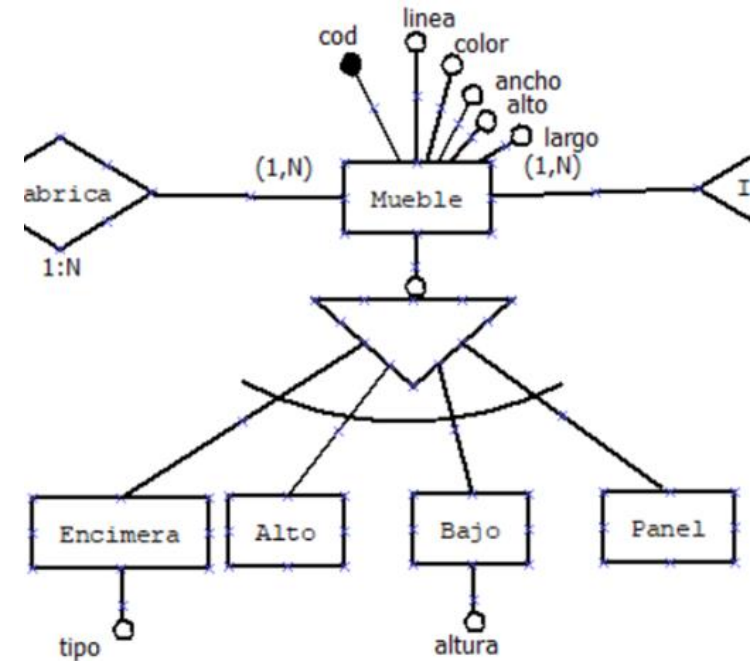


¿Cuál es la diferencia con el caso de dependencia en existencia?

5.6.- Paso del diagrama E/R al diagrama relacional

Cuando nos encontramos con una **superclase** y varias **subclases** tenemos varias opciones para transformarlo en el modelo relacional:

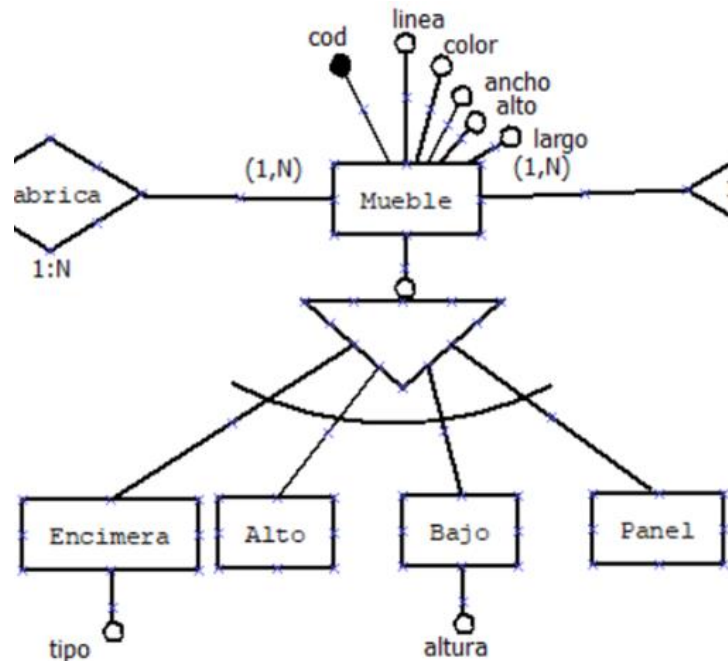
1. Englobar la superclase y las subclases en **una sola tabla** con todos los atributos, comunes y no comunes de las entidades. La tabla deberá tener un atributo “Tipo” que distinga entre subclases. Solo válido si no hay relaciones particulares de las subclases y si son exclusivas. ¿Por qué?
2. **Una tabla por cada subclase** con los atributos propios y los comunes. Solo vale para exclusivas totales. ¿¿Por qué??
3. **Una tabla para la superclase y una para cada subclase.** Las de la subclase heredan la clave primaria de la superclase como clave ajena. Vale tanto para totales como parciales al igual que solapadas y exclusivas. ¿¿¿Por qué???



5.6.- Paso del diagrama E/R al diagrama relacional

Veamos un ejemplo:

1. Englobar la superclase y las subclases en **una sola tabla** con todos los atributos, comunes y no comunes de las entidades. La tabla deberá tener un atributo “Tipo” que distinga entre subclases. **Solo válido si no hay relaciones particulares de las subclases y si son exclusivas.**

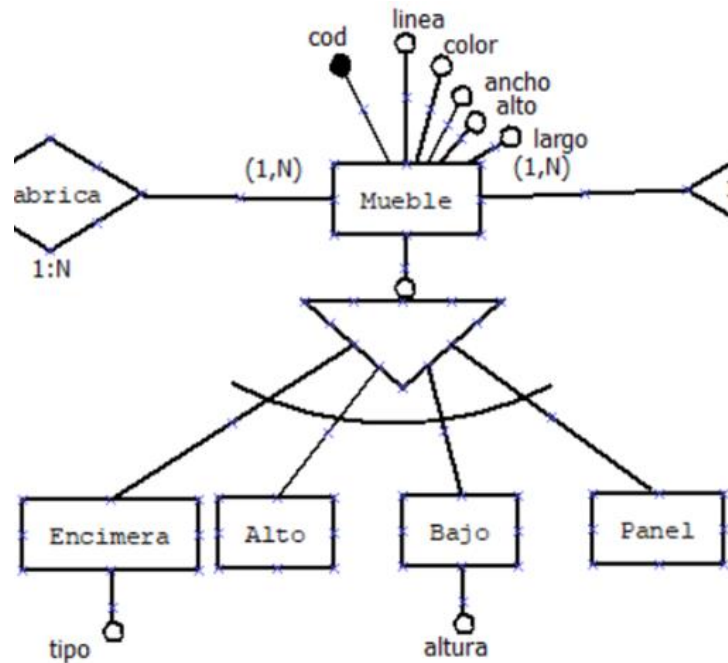


Mueble(cod, línea, color, ancho, alto, largo, tipo, altura, clase_mueble)

5.6.- Paso del diagrama E/R al diagrama relacional

Veamos un ejemplo:

2. **Una tabla por cada subclase** con los atributos propios y los comunes. Solo vale para exclusivas totales.



Encimera(cod, línea, color, ancho, alto, largo, tipo)

Alto(cod, línea, color, ancho, alto, largo)

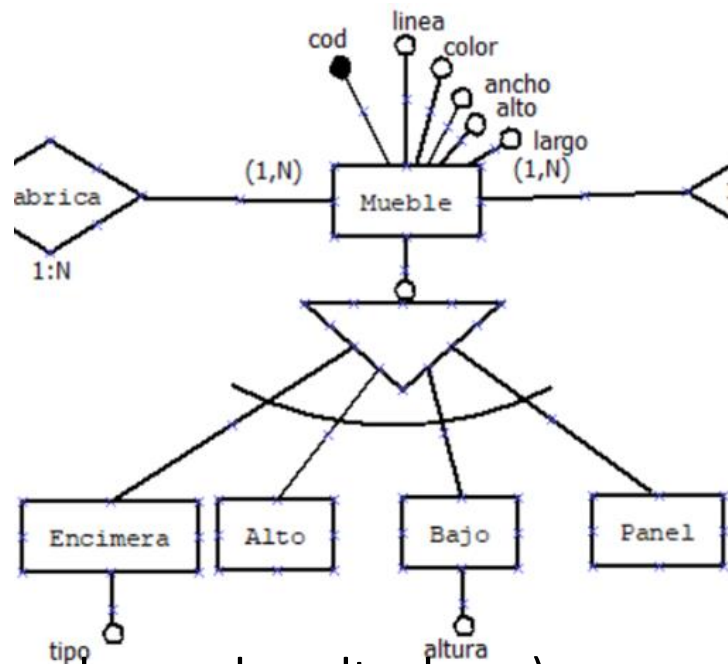
Bajo(cod, línea, color, ancho, alto, largo, altura)

Panel(cod, línea, color, ancho, alto, largo)

5.6.- Paso del diagrama E/R al diagrama relacional

Veamos un ejemplo:

3. **Una tabla para la superclase y una para cada subclase.** Las de la subclase heredan la clave primaria de la superclase como clave ajena. **Vale tanto para totales como parciales al igual que solapadas y exclusivas.**



Mueble(cod, línea, color, ancho, alto, largo)

Encimera(cod mueble, tipo)

Alto(cod mueble)

Bajo(cod mueble, altura)

Panel(cod mueble)

6- Normalización

PEDIDO

<u>Artículo</u>	<u>cliente</u>	cantidad	precio	ciudad	distancia
A1	C1	12	100	Madrid	400
A1	C2	30	100	Valencia	200
A1	C3	15	100	Alicante	80
A2	C1	35	250	Madrid	400
A2	C2	20	250	Valencia	200
A2	C4	10	250	Madrid	400
A3	C3	25	175	Alicante	80

Este diseño es inadecuado porque:

- **Hay redundancia:** Se repite innecesariamente precio, ciudad, distancia.
- **Anomalías de modificación:** Por error podríamos tener el mismo artículo con dos precios.
- **Anomalías de inserción:** A la hora de insertar un nuevo artículo podemos cometer errores ya que estamos metiendo realmente un pedido, no un artículo.
- **Anomalías de borrado:** Si borramos por ejemplo los pedidos A1-C2 y A2-C2 perdemos los datos del cliente C2.

6- Normalización

Normalización

Proceso que persigue garantizar que se cumplen una serie de normas.

En primer lugar se debe garantizar que no hay redundancia en los datos guardados.

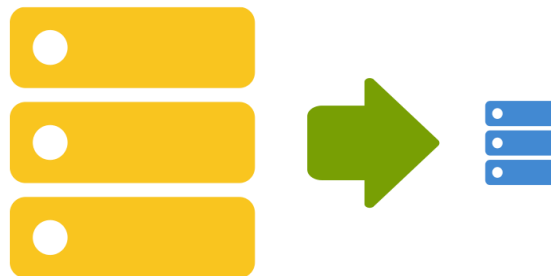
En segundo lugar se busca una coherencia en el esquema conceptual de los datos guardados.

Transformaremos las estructuras complejas originales en datos pequeños, más simples y estables.

Se busca evitar errores lógicos derivados de una estructura compleja.

Una base de datos normalizada ocupará menos espacio que una sin normalizar.

Existen varias Formas Normales, cada una derivada de la anterior. Antes de tratar las formas normales vamos a ver el concepto de dependencia funcional entre atributos o grupos de atributos.



6.1.- Dependencia funcional

Concepto de dependencia funcional:

Se dice que, en una tabla o relación, un atributo Y depende funcionalmente de otro atributo X, o que $X \rightarrow Y$, si cada valor de X tiene asociado en todo momento un único valor de Y. También se dice que X implica Y y, por tanto, que X es implicante de Y.

Dependencia funcional completa:

Dada una combinación de atributos (X_1, X_2, \dots) , se dice que Y tiene dependencia funcional completa de esos atributos si depende funcionalmente de ese conjunto pero no depende funcionalmente de un subconjunto de ellos.

Dependencia funcional transitiva:

Una dependencia funcional es **transitiva** si se cumple que “a implica b” y “b implica c” no cumpliéndose que “b implica a”.

6.1.- Dependencia funcional

Dependencia funcional:

PRODUCTO (codPro, nombre, precio, descripción)

Se produce que **codPro** → **nombre**

Dependencia funcional completa:

ARTICULO (idRevista, numeroEdicion, numArticulo, pagina)

idRevista, numeroEdicion, NumArticulo → **pagina**

Dependencia funcional transitiva:

CLIENTES(codCli, nombre, codProv, Provincia)

Hay una dependencia transitiva entre codCLI y Provincia ya que:

codCli → **codProv**

codProv → **Provincia**

Y codProv no implica CodCli.

6.2.- Formas Normales

FN1:

En esta forma normal se prohíbe que en una tabla haya atributos que puedan tomar valores múltiples. Es una forma inherente al modelo ya que en las tablas de un SGBD relacional se impide realizarlo.

<u>codEq</u>	<u>posicion</u>	Jugador
RMA	Delantero	C. Ronaldo Bale Benzema
RMA	Portero	K. Navas Casilla
FCB	Delantero	Suarez Messi Neymar

Esta tabla no está en FN1.
Una posible solución será:

Pero, ¿de qué otra forma
se podría solucionar?

codEq	posicion	Jugador
RMA	Delantero	C. Ronaldo
RMA	Delantero	Bale
RMA	Delantero	Benzema
RMA	Portero	K. Navas
RMA	Portero	Casilla
FCB	Delantero	Suarez
FCB	Delantero	Messi
FCB	Delantero	Neymar

6.2.- Formas Normales

FN2:

Un diseño de BD está en FN2 si lo está en FN1 y, además, cada atributo que no forma parte de la clave tiene dependencia completa de la clave principal.

Dicho de otra forma: *Cada atributo que no forma parte de la clave primaria no tiene dependencia de ninguna parte no completa de la clave.*

Por ejemplo:

Si tenemos una tabla de módulos cursados por los alumnos,
CURSADOS (codAlumno, codModulo, nombreModulo, notaFinal)

Esto no está en FN2 dado que un atributo que no forma parte de la clave principal (**nombreModulo**) **no tiene dependencia completa de la clave principal.**

No tiene dependencia completa debido a que nombreModulo depende funcionalmente de una parte de la clave principal (en este caso nombreModulo depende exclusivamente de codModulo y no depende de codAlumno: es decir, no depende de las dos partes de la clave).

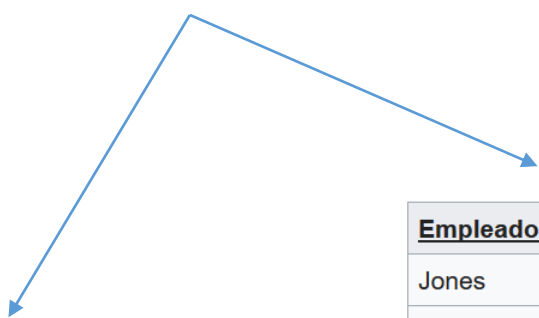
6.2.- Formas Normales

FN2: Ejemplos

En el siguiente ejemplo, “Lugar actual de trabajo” tiene dependencia funcional de una parte de la clave, de “Empleado”, luego no está en FN2.

<u>Empleado</u>	<u>Habilidad</u>	Lugar actual de trabajo
Jones	Mecanografía	114 Main Street
Jones	Taquigrafía	114 Main Street
Jones	Tallado	114 Main Street
Bravo	Limpieza ligera	73 Industrial Way
Ellis	Alquimia	73 Industrial Way
Ellis	Malabarismo	73 Industrial Way
Harrison	Limpieza ligera	73 Industrial Way

Para que esté en FN2, habría que transformarlo en dos tablas:



<u>Empleado</u>	Lugar actual de trabajo
Jones	114 Main Street
Bravo	73 Industrial Way
Ellis	73 Industrial Way
Harrison	73 Industrial Way

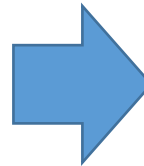
<u>Empleado</u>	<u>Habilidad</u>
Jones	Mecanografía
Jones	Taquigrafía
Jones	Tallado
Bravo	Limpieza ligera
Ellis	Alquimia
Ellis	Malabarismo
Harrison	Limpieza ligera

6.2.- Formas Normales

FN3:

Un diseño se encuentra en FN3 si está en FN2 y además, no hay ningún atributo que, no formando parte de la clave principal, dependa transitivamente de la clave principal. *Esto se puede decir de otra forma: cuando no exista ningún atributo que tenga dependencia de otro atributo que no sea clave principal.*

REPRESENTANTE				
DNI	nombre	apellidos	CodProv	Provincia
11111111B	Antonio	Arias Arias	39	Cantabria
12121212C	Blanca	Bueno Bueno	28	Madrid
13131313D	Carlos	Cruz Cruz	33	Asturias
14141414E	Diana	Diaz Diaz	28	Madrid
22222222F	Emma	Esquer Esquer	33	Asturias
23232323G	Frank	Fuerte Fuerte	48	Bizkaia
24242424H	German	García García	28	Madrid



REPRESENTANTE			
DNI	nombre	apellidos	CodProv
11111111B	Antonio	Arias Arias	39
12121212C	Blanca	Bueno Bueno	28
13131313D	Carlos	Cruz Cruz	33
14141414E	Diana	Diaz Diaz	28
22222222F	Emma	Esquer Esquer	33
23232323G	Frank	Fuerte Fuerte	48
24242424H	German	García García	28

PROVINCIA	
CodProv	NomProv
39	Cantabria
28	Madrid
33	Asturias
48	Bizkaia

6.2.- Formas Normales

FN3:

Diferencia con la FN2: En la FN2 hablamos de dependencia de la clave o parte de ella. En este caso hablamos de dependencia de cualquier atributo.

Otro ejemplo:

Si tenemos una tabla de ganadores de un torneo,
GANADORES (Torneo, Año, Ganador, FechaNacimientoGanador)

Esta tabla está en FN2. Ya que ni Ganador ni FechaNacimientoGanador dependen funcionalmente de Torneo ni de Año por separado, pero si juntas.

Sin embargo, no está en FN3 ya que FechaNacimientoGanador depende funcionalmente de Ganador.

6.2.- Formas Normales

FN3: ejemplos

La tabla: SOCIO (DNI, ciudad, país, edad)

¿Está en FN2?

No está en FN3 ya que país depende transitivamente de DNI a través de ciudad. Ya que:

- DNI → Ciudad, y Ciudad → País, pero Ciudad no implica DNI.

¿Cómo hacemos que esté en FN3?

SOCIO (DNI, Ciudad, Edad)

SITUADA (Ciudad, País)

Otro ejemplo:

EVENTO (Cod_Evento, Nom_Evento, Nom_local, Nom_Muni)

Está en FN2 porque todas dependen funcionalmente de la clave y no de una parte de ella (no se puede dividir porque solo tiene un atributo).

No está en FN3 porque Cod_evento → Nom_local, Nom_local → Nom_Muni pero Nom_local no implica Cod_evento. Dicho de otra forma, tenemos una dependencia funcional entre atributos no pertenecientes a la PK.

6.2.- Formas Normales

Clave candidata:

Sería cualquier conjunto de atributos que permiten identificar de forma única a cada elemento de la entidad, no pudiéndose hacer esa identificación con un subconjunto de atributos de la clave candidata. Por ejemplo, dentro de los atributos de ALUMNO, podrían ser claves candidatas:

- ☐ DNI
- ☐ Número de matriculación
- ☐ Nombre + apellidos + fecha nacimiento (considerando que nombre + apellidos podría repetirse y por tanto no servir para identificar a un solo elemento).

Sin embargo, el conjunto DNI + Numero de matriculación no sería clave candidata porque al menos un subconjunto de atributos de esa clave candidata (DNI) permite identificar de forma única a cada elemento de la entidad.

Clave primaria o principal: Sólo puede haber una clave primaria o principal para identificar a los elementos de una entidad. La clave primaria es, dentro de las claves candidatas, aquella que presente mejores características para ser usada como tal clave primaria.

6.2.- Formas Normales

FNBC (Forma Normal de Boyce-Codd):

Un diseño se encuentra en FNBC si está en FN3 y además todo determinante (atributo del que depende funcionalmente de manera completa algún otro atributo) es una clave candidata. Para casos en los que hay una DF entre atributos no principales pero no transitiva.

idEmp	Departamento	idResp
Emp01	Producción	Resp01
Emp02	Ventas	Resp04
Emp03	Producción	Resp03
Emp04	Producción	Resp01
Emp05	Producción	Resp02
Emp06	Ventas	Resp05
Emp07	Ventas	Resp05
Emp08	Producción	Resp01
Emp09	Producción	Resp02
Emp10	Ventas	Resp04



PERSONAL	
idEmp	idResp
Emp01	Resp01
Emp02	Resp04
Emp03	Resp03
Emp04	Resp01
Emp05	Resp02
Emp06	Resp05
Emp07	Resp05
Emp08	Resp01
Emp09	Resp02
Emp10	Resp04

No está en FNBC ya que idResp implica Departamento e idResp no es clave candidata. Observad que no hay dependencia transitiva con la clave.

RESPONSABLES	
idResp	Departamento
Resp01	Producción
Resp03	Producción
Resp02	Producción
Resp05	Ventas
Resp04	Ventas

6.2.- Formas Normales

FNBC (Forma Normal de Boyce-Codd): ejemplo

Supongamos que tenemos esta tabla:

student_id	subject	professor
101	Java	P.Java
101	C++	P.Cpp
102	Java	P.Java2
103	C#	P.Chash
104	Java	P.Java

¿Cuál sería la clave principal?

¿Cumple FN3? Si, porque no hay dependencias funcionales transitivas de la clave principal.

¿Cumple FNBC? Subject tiene una dependencia funcional de profesor. Luego no lo cumple.

¿Cómo hacemos que cumpla FNBC?