

BASES DE DATOS

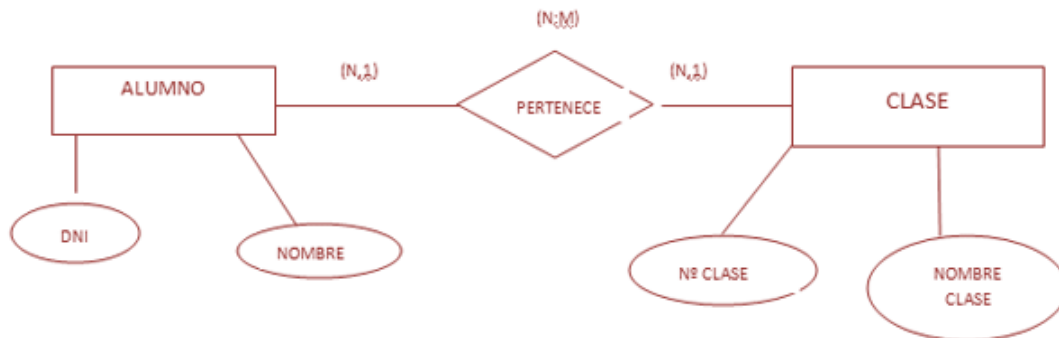


EL MODELO ENTIDAD-RELACIÓN (III)

1 Paso del diagrama E/R al Modelo Relacional

Una vez que tenemos el esquema conceptual con el modelo Entidad/Relación, el siguiente paso es definir el modelo lógico de datos. A continuación veremos las reglas básicas para transformar un Modelo E/R a un Modelo Relacional.

Para pasar a tablas todos los datos sin dejarnos nada y que las tablas tengan sentido por si solas tenemos que seguir unos pasos.



Para este modelo de entidad-relación el paso a tablas quedaría de la siguiente forma:

Tabla alumno	DNI(clave primaria)	nombre
---------------------	---------------------	--------

Tabla clase	Nº clase (clave primaria)	Nombre clase
--------------------	---------------------------	--------------

Tabla pertenece	DNI (clave foránea)	Nº clase (clave foránea)
------------------------	---------------------	--------------------------

}
 Clave primaria

El paso a tablas es un factor sumamente importante de cara a la construcción de una base de datos para la web.

1.1 Transformación de entidades

Las entidades del modelo E/R se pasan al Modelo Relacional según estas reglas:

- Toda entidad se convierte en una tabla, cuyo nombre será el mismo que el de la entidad, pero en plural.
- Cada atributo de la entidad será una columna de la tabla y se mantendrán los mismos nombres.
- Los identificadores principales pasan a ser claves primarias.
- Los identificadores alternativos pasan a ser claves alternativas.

1.2 Transformación de relaciones

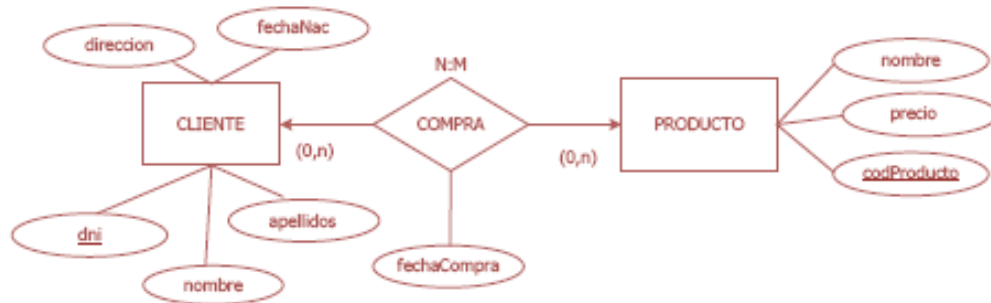
En general se transforma cada relación del modelo E/R en una tabla en el Modelo Relacional, pero no siempre es así. Hay que analizar cada caso y su tipo de correspondencia.

RELACIONES N:M

La relación se transforma en una nueva tabla que tendrá como clave primaria la concatenación de los identificadores principales de las entidades que asocia.

Si la relación tuviera atributos, estos pasarían a ser atributos de la nueva tabla. Además, hay que estudiar, si estos atributos de la relación deben formar parte de la clave primaria de la nueva tabla.

Ejemplo



1) Transformamos las entidades CLIENTE y PRODUCTO: Recordamos que al pasar una entidad a tabla mantenemos el nombre de la entidad pero en plural.

NOTA: Utilizamos la notación nombre-tabla(atributo1, atributo2,atributoN)

CLIENTES (DNI, NOMBRE, APELLIDOS, DIRECCION, FECHANAC)

PRODUCTOS (CODPRODUCTO, NOMBRE, PRECIO)

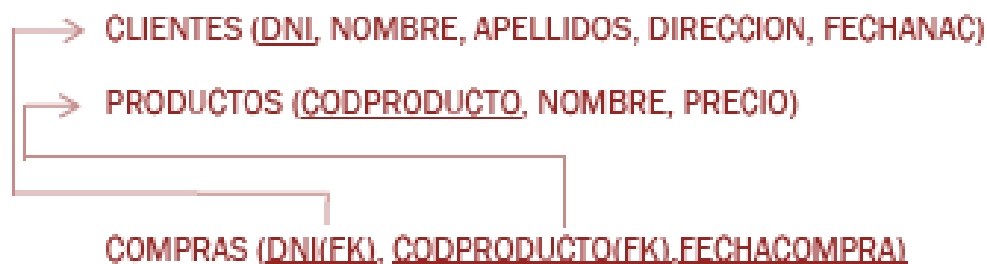
2) Transformamos la relación COMPRA:

COMPRAS (DNI(FK), CODPRODUCTO(FK), FECHACOMPRA)

Vemos que la clave primaria de la tabla está formada por los identificadores principales de las entidades que asocia la relación (DNI y CODPRODUCTO) que además son claves ajenas lo que denotamos con **FK** (*Foreign Key*).

En este ejemplo FECHACOMPRA, atributo de la relación, también formará parte de la clave primaria puesto que un mismo cliente puede comprar un mismo producto varias veces, y es la fecha de compra la que permitirá distinguir dichas compras.

Gráficamente la solución final se expresa:



Donde las flechas indican la propagación de la clave. Salen de la clave ajena en la tabla hija y entra en la tabla padre a la que hace referencia.

RELACIONES 1:N

Hay dos posibilidades:

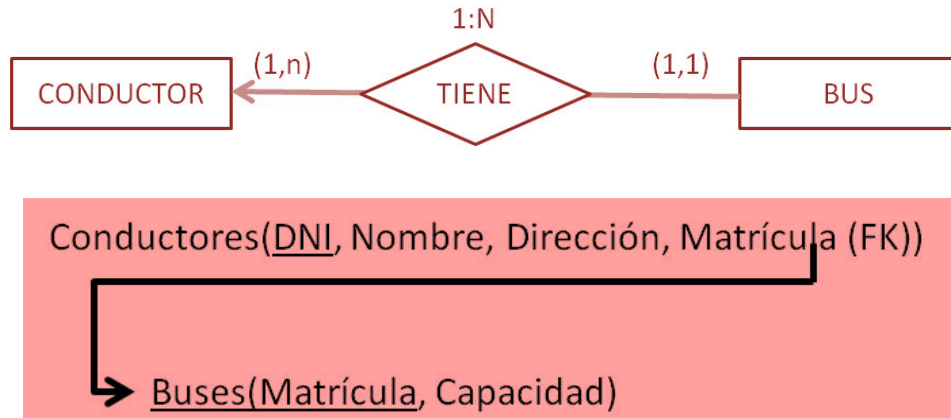
1. Propagar la clave

Se escoge esta solución cuando la entidad que tiene cardinalidad máxima 1, también tiene cardinalidad mínima 1. En este caso:

- La clave de la entidad que tiene a su lado la cardinalidad máxima 1 se propaga a la tabla resultante de la otra entidad (la del lado N).
- Si la relación tiene atributos, estos también pasan a formar parte de la tabla de cardinalidad máxima N.

Ejemplo

Cada conductor conduce un único autobús, pero un autobús puede tener varios conductores.



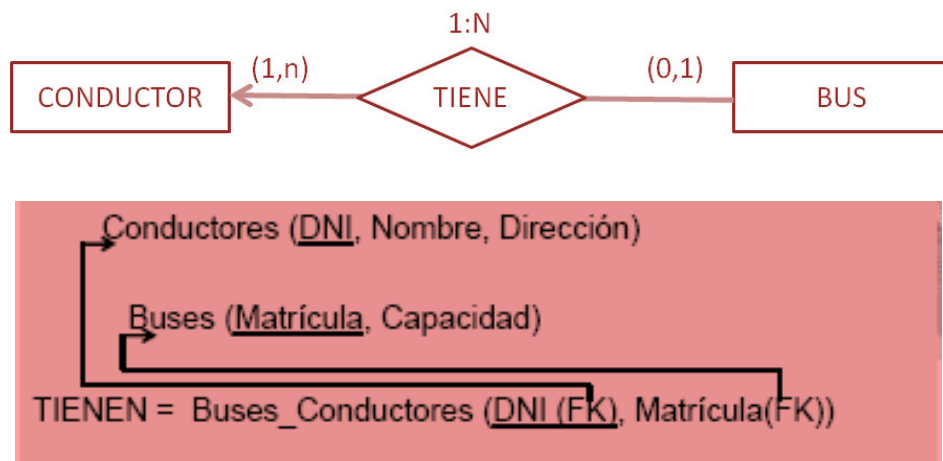
2. Transformar la interrelación en una nueva tabla

Se hace cuando la entidad que tiene a su lado la cardinalidad máxima 1, tiene cardinalidad mínima 0.

- Se construye una nueva tabla correspondiente a la relación, formada por los identificadores principales de las entidades que une la relación y los atributos de la relación, si los hubiera.
- La clave primaria de esta nueva tabla será el identificador principal de la entidad que a su lado tiene la cardinalidad máxima N.
- La tabla tendrá como claves ajenas los identificadores principales de las entidades que participan en la relación.

Ejemplo

Un conductor puede no tener asignado un autobús, pero si lo tiene sólo puede tener 1. Un autobús puede tener varios conductores.



RELACIONES 1:1

Se pueden dar tres casos según las cardinalidades de las entidades:

- Caso 1: Cardinalidades (1,1) en las dos entidades.
- Caso 2: Cardinalidades (0,1) en las dos entidades.
- Caso 3: Cardinalidades (1,1) en una entidad y (0,1) en la otra.

1) Cardinalidades (1,1) en ambas entidades

Opción 1 No es necesario crear una nueva tabla. Podemos propagar la clave principal, de cualquiera de las entidades a la tabla resultante de la otra entidad, en forma de clave ajena). En general, se propaga la clave primaria de la tabla que tiene accesos más frecuentes a la tabla menos utilizada.

Opción 2 Se podrían unir las dos entidades en una única tabla tomando como clave principal la clave primaria de cualquiera de las entidades. En general, se elige la clave primaria de la entidad que se considere más importante.

El identificador principal de la otra entidad podría utilizarse como clave alternativa.

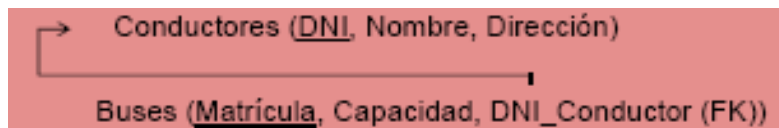
Ejemplo

Cada conductor tiene su autobús y sólo lo conduce él. Dado un autobús sólo tiene un conductor.

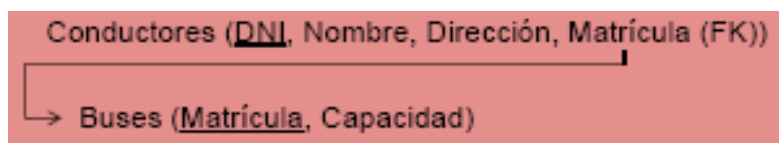


Tenemos cuatro soluciones posibles:

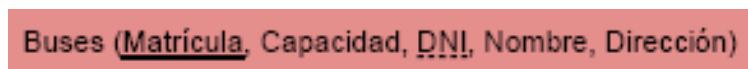
- Aplicando la opción 1 propagando de Conductores a Buses



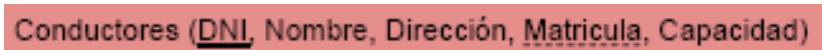
- Aplicando la opción 1 propagando de Buses a Conductores



- Aplicando la opción 2 unimos ambas en la tabla Buses y pasamos a ella los atributos de Conductores, con DNI como clave alternativa.



- Aplicando la opción 2 unimos ambas en la tabla Conductores y pasamos a ella los atributos de Buses, con Matrícula como clave alternativa.

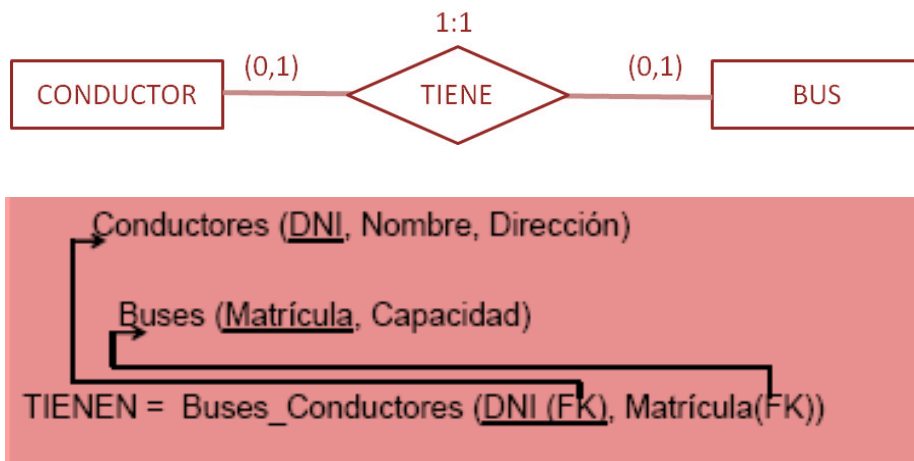


2) Cardinalidades (0,1) en ambas entidades

Construimos una nueva tabla correspondiente a la relación que estará formada por los identificadores principales de las entidades y por los atributos de la relación, si los tuviera. Tomamos como clave principal de la nueva tabla el identificador principal de una de las entidades y el de la otra puede ser clave alternativa.

Ejemplo

Cada conductor puede tener un autobús que sólo conduce él o no. Dado un autobús puede tener un conductor fijo o no

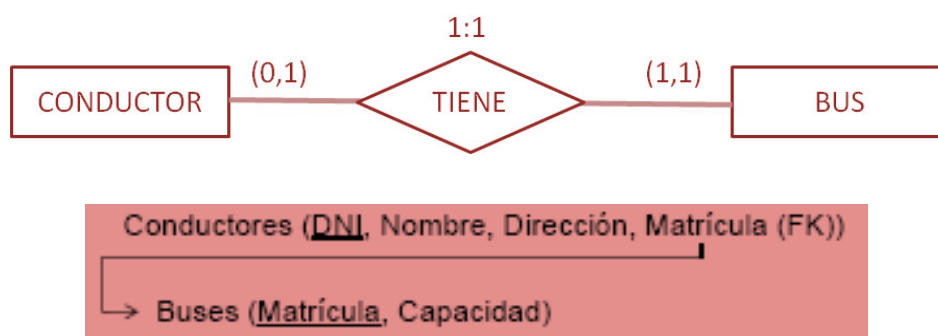


3) Cardinalidades (0,1) y (1,1) en las entidades

En este caso conviene propagar la clave principal de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad con cardinalidad (0,1). En esta tabla sería clave ajena y nunca tendría valores nulos.

Ejemplo

Cada conductor tiene un autobús que sólo conduce él. Dado un autobús puede tener un conductor fijo o no

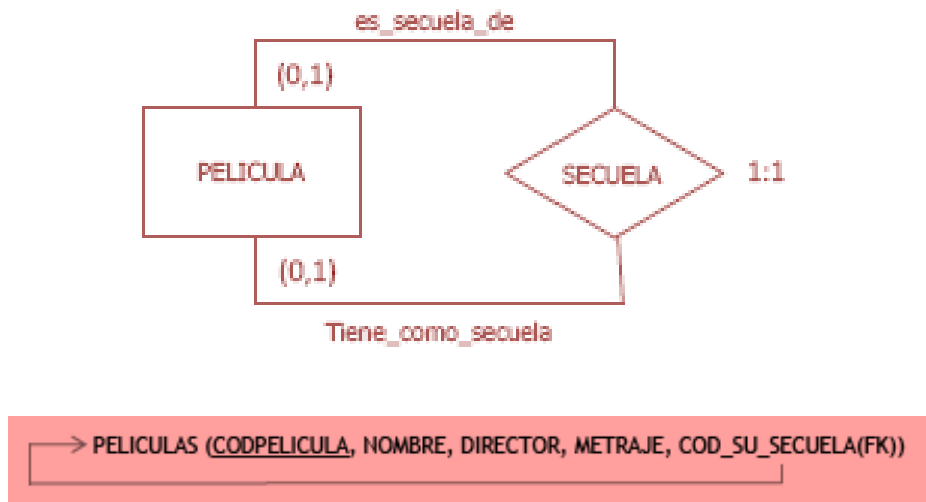


RELACIONES REFLEXIVAS 1:1

Resulta una única tabla que será la asociada a la entidad. La clave principal de la tabla se propaga a sí misma, por lo que aparecerá dos veces: una como clave primaria y otra como clave ajena. A una de ellas le cambiamos el nombre.

Ejemplo:

Una película A puede tener secuela en otra B y entonces la película B es secuela a la película A. Tenemos así la relación PELICULA-SECUELA-PELICULA.



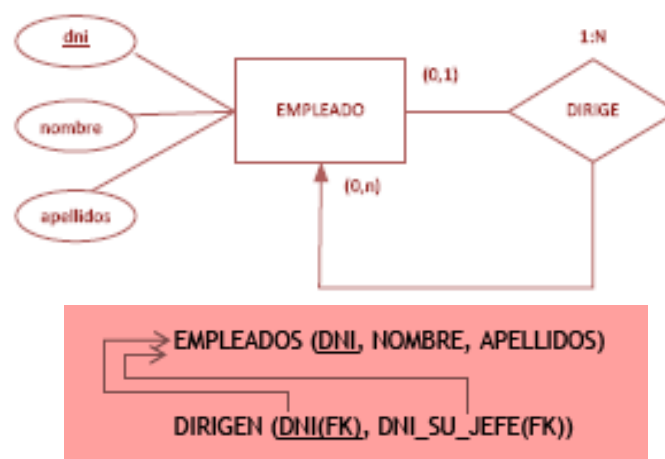
RELACIONES REFLEXIVAS 1:N

Tenemos dos posibles casos:

- 1) Si la entidad que tiene a su lado la cardinalidad máxima 1 también tiene cardinalidad mínima 1, es decir tenemos (1,1), entonces hacemos igual que si fuera reflexiva 1:1
- 2) Si la entidad que tiene a su lado la cardinalidad máxima 1, tiene cardinalidad mínima cero, (0,1), entonces creamos una nueva tabla para la relación, en la que:
 - El identificador principal de la entidad estará repetido dos veces con nombre diferente y serán claves ajenas de la misma tabla.
 - Los atributos de la relación pasan a ser atributos de la tabla.
 - La clave primaria será el identificador principal del lado N.

Ejemplo

Sea la relación EMPLEADO-DIRIGE-EMPLEADO en la que un empleado puede dirigir a muchos empleados (si es jefe) o a ninguno (si no es jefe). Un empleado puede ser dirigido por un jefe o por ninguno (si él es el jefe o si todavía no tiene jefe).



RELACIONES REFLEXIVAS N:M

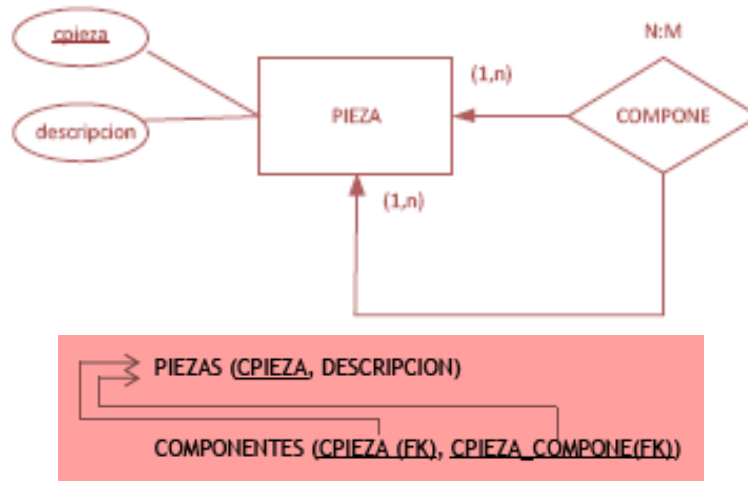
Se tratan como las relaciones binarias.

La tabla resultante de la relación contendrá dos veces el identificador principal de la entidad (con distintos nombres), más los atributos de la relación si los hubiera.

La clave primaria de la nueva tabla será la combinación de las dos, que también serán claves ajenas de la misma tabla.

Ejemplo

Una pieza está compuesta por otras piezas lo que nos da la relación PIEZA-COMPONE-PIEZA.



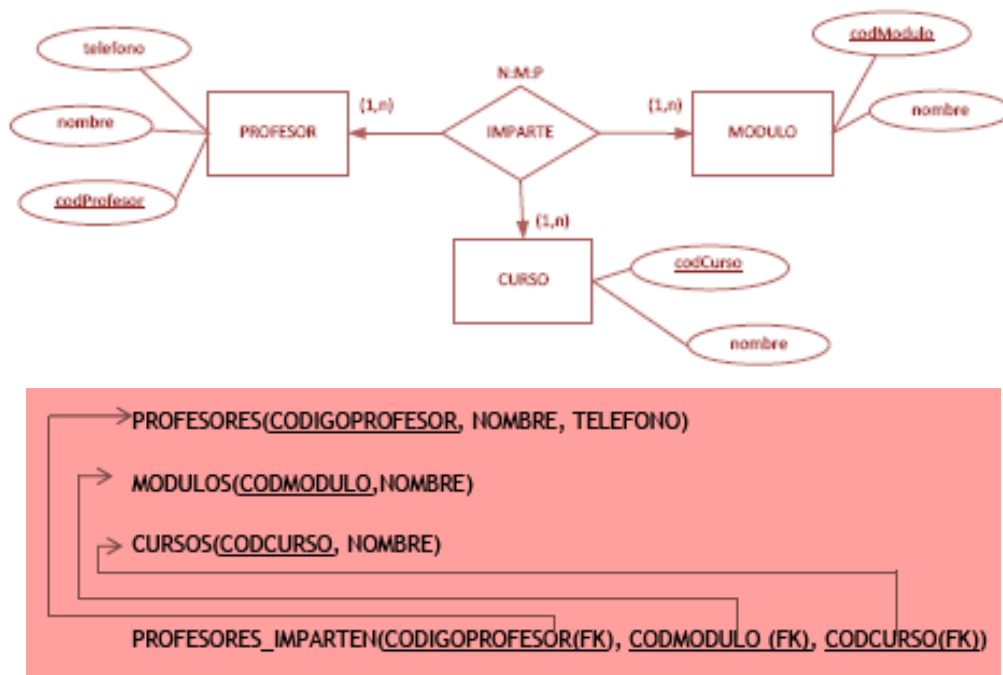
RELACIONES TERNARIAS

Son las que agrupan a 3 entidades, pero extendemos el procedimiento a las relaciones que asocien 4 ó más entidades.

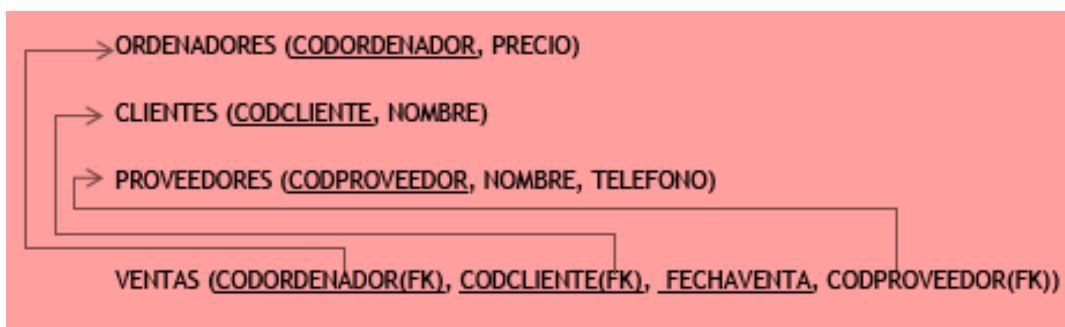
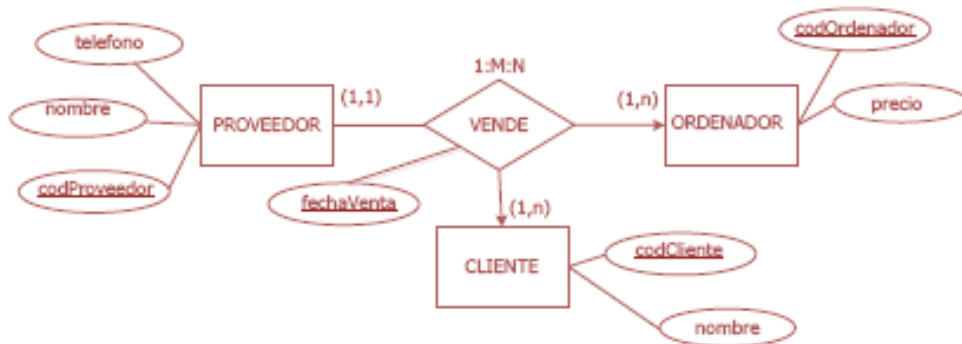
Para pasar al modelo de datos relacional la relación se convierte en una tabla nueva, que tiene los atributos propios de la relación más los identificadores principales de todas las entidades que asocia.

- Si la relación es M:N:P, es decir, si todas las entidades participan con cardinalidad máxima N, entonces la clave primaria de la tabla resultante es la unión de los identificadores de las entidades que relaciona.

Ejemplo



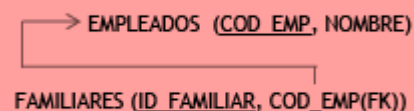
- Si la relación es 1:M:N, es decir, si la cardinalidad máxima que aparece en el lado de una de las entidades es 1, entonces el identificador principal de esta entidad no formará parte de la clave primaria de la tabla resultante, pero sí de la tabla como un atributo más.



RELACIONES ENTRE UNA ENTIDAD FUERTE Y UNA DÉBIL

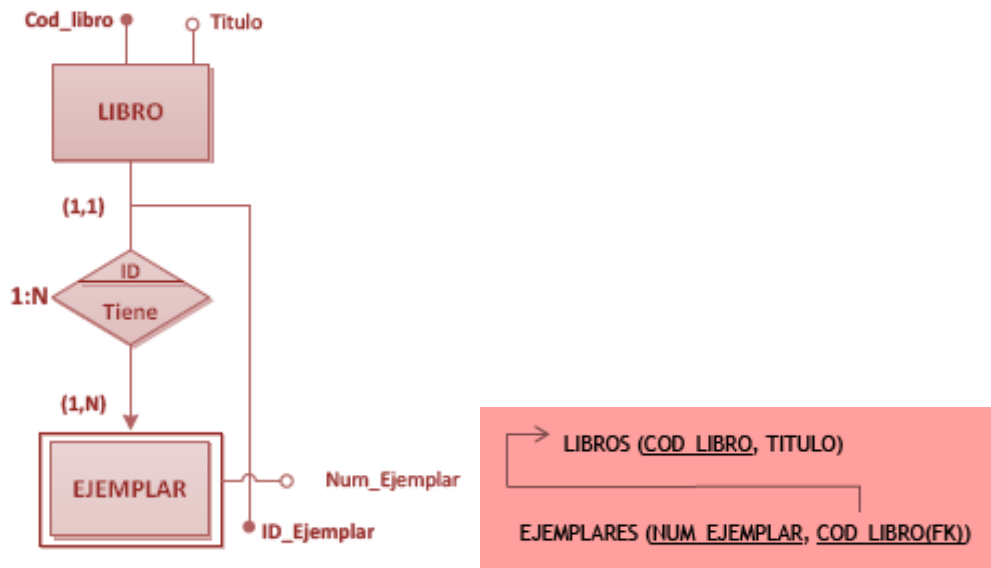
Dependencia en Existencia

Se propaga el identificador principal de la entidad fuerte a la tabla resultante de la entidad débil, pasando a ser clave ajena de ella. Es la misma transformación que se hace en la relación 1:N con una cardinalidad (1,1).



Dependencia en Identificación

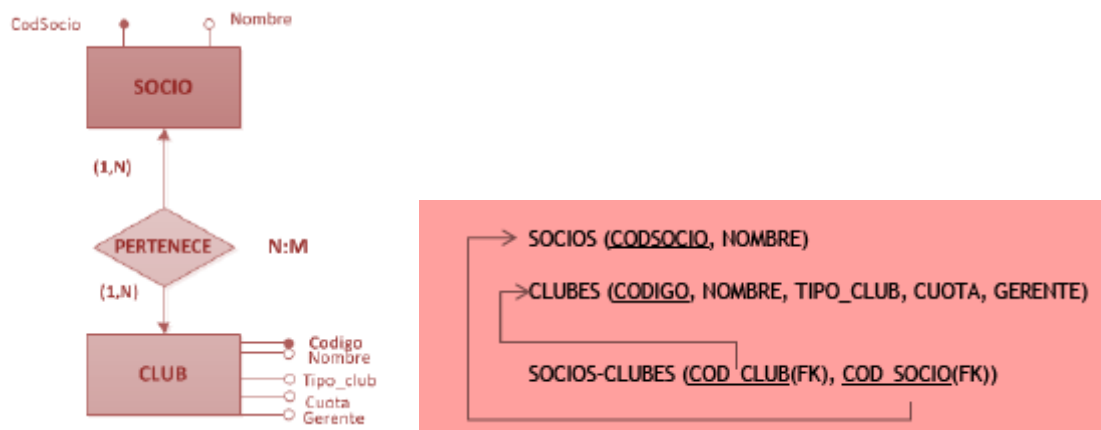
Se transforma como una relación 1:N con una cardinalidad (1,1) con la diferencia de que la tabla que se obtiene de la entidad débil tiene por clave primaria la propia sino también la de la entidad fuerte que se ha propagado.



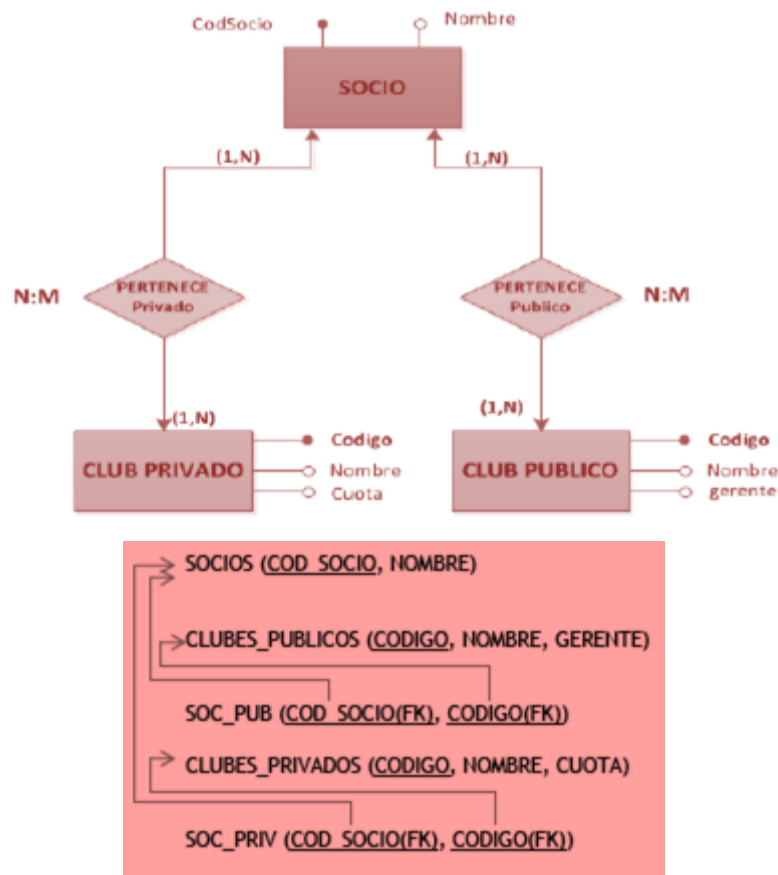
TRANSFORMACIÓN DE JERARQUÍAS

El modelo relacional no dispone de mecanismos para la representación de las relaciones jerárquicas por lo que debemos eliminar este tipo de relaciones al transformarlas al modelo relacional. Se aplicará una de las siguientes opciones:

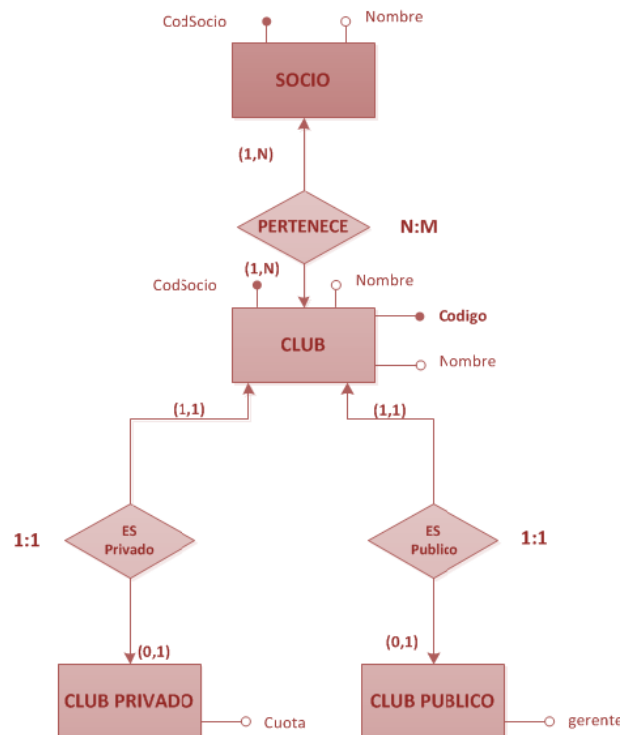
1. Integrar todas las entidades en una única tabla eliminando los subtipos. Esta tabla contendrá todos los atributos del supertipo y de los subtipos, así como los atributos necesarios que permitan distinguir a qué subtipo pertenece cada atributo. Si los subtipos participan en relaciones, estas deben mantenerse con la nueva entidad. Esta opción se puede utilizar con cualquier tipo de jerarquía. Es muy sencilla de aplicar, pero se pueden generar muchos valores nulos en los atributos opcionales propios de cada subtipo.

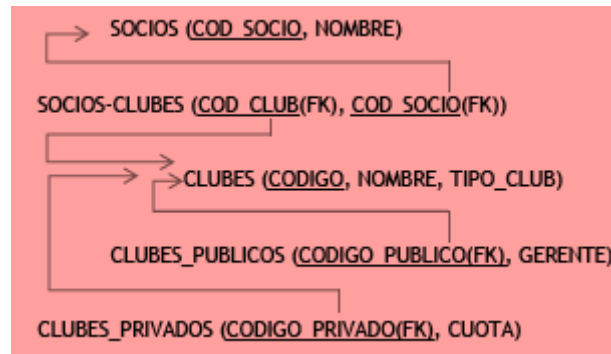


2. Integrar todas las entidades en una única, eliminando el supertipo y transfiriendo sus atributos a los subtipos. Las relaciones que tuviera el supertipo pasan a relacionarse con cada uno de los subtipos. La clave del supertipo pasa a cada uno de los subtipos. Esta opción sólo puede ser aplicada a las relaciones que son exclusivas y totales. Tiene como desventaja que hace que se duplique información y que el número de relaciones aumente.



3. Mantener supertipo y subtipos. Es como si insertáramos una relación 1:1 entre el supertipo y cada uno de los subtipos. Los atributos se mantienen en cada entidad y cada subtipo se identificará con la clave ajena del supertipo.





1.3 Restricciones que no pueden plasmarse en el diseño lógico

Muchos sistemas gestores de bases de datos no implementan el modelo relacional completo. Por ello debemos incorporar en el modelo relacional documentación con la suficiente información que permita incorporar a la base de datos mecanismos que tengan en cuenta las restricciones señaladas.

Algunas consideraciones que es conveniente recoger serían:

1. Restricciones que no figuran en el enunciado original pero que se consideran adecuadas o convenientes.
2. Las cardinalidades mínimas 1 en relaciones N:M y 1:N (algunas se controlan con la restricción NOT NULL cuando se realiza la propagación de la clave).
3. Cardinalidades máximas conocidas en las relaciones binarias N:M, 1:N y relaciones ternarias. Por ejemplo: (1,3).
4. Exclusividad y totalidad en las generalizaciones.
5. Inserción y borrado en las generalizaciones.
6. Etc.

Los esquemas relacionales deben completarse con las anotaciones apropiadas, referidas a las consideraciones de este apartado.

2 Reglas de integridad

Los SGBD deben proporcionar la forma de definir las restricciones de integridad de usuario de una base de datos y una vez definida, debe velar por su cumplimiento. Las reglas de integridad del modelo, en cambio, no se deben definir para cada base de datos concreta, porque se consideran preestablecidas para todas las base de datos de un modelo. Un SGBD de un modelo determinado debe velar por el cumplimiento de las reglas de integridad preestablecidas por su modelo.

Regla de integridad de unicidad de la clave primaria

La regla de integridad de unicidad está relacionada con la definición de clave primaria que establece que toda clave primaria que se elija para una relación no debe tener valores repetidos.

Regla de integridad de entidad de la clave primaria

La regla de integridad de entidad de la clave primaria dispone que los atributos de la clave primaria de una relación no pueden tener valores nulos. Esta regla es necesaria para que los valores de las claves primarias puedan identificar las tuplas

individuales de las relaciones. Si las claves primarias tuviesen valores nulos, es posible que algunas tuplas no se pudieran distinguir.

Regla de integridad referencial

La regla de integridad referencial está relacionada con el concepto de clave foránea, lo que determina que todos los valores que toma una clave foránea deben ser valores nulos o valores que existen en la clave primaria que referencia. La necesidad de esta regla es debido a que las claves foráneas tienen por objetivo establecer una conexión con la clave primaria que referencian.

Para cada clave ajena de la BD hay que contestar a estas preguntas

1) **¿Tiene sentido que la clave ajena acepte valores nulos?**

2) **¿Qué ocurre si se intenta borrar la tupla referenciada por la clave ajena?**

En este caso podemos:

-**Restringir**: No se permite modificar el valor de la clave primaria de la tupla referenciada.

-**Propagar**: se modifica el valor de la clave primaria de la tupla referenciada y se propaga la modificación a las tuplas que la referencian mediante clave ajena.

-**Anular**: se modifica la tupla referenciada y las tuplas que la referenciaban ponen a nulo el valor la clave ajena (sólo si esta acepta nulos)

3) **¿Qué ocurre si se intenta modificar el valor de la clave primaria de la tupla referenciada por la clave ajena?**

Regla de integridad de dominio

Está relacionada con la noción de dominio. Establece dos condiciones.

- Un valor no nulo de un atributo A debe pertenecer al dominio del atributo A
- Un operador determinado sólo se puede aplicar sobre valores que tengan dominios que le sean adecuados.

3 Formas normales

Las bases de datos mal diseñadas tienen problemas de:

- Almacenamiento redundante (varias copias de la misma información)
- Pérdidas no deseadas de información al modificar tuplas
- La base entra en un estado no consistente al borrar una tupla
- Imposibilidad de almacenar cierta información

Las Formas Normales son un grupo de reglas que aseguran que un conjunto de Relaciones (tablas) que forman una BDD, tengan un diseño adecuado.

Existen 5 formas normales, más la denominada de Boyce-Codd. De ellas, las tres primeras son imprescindibles en cualquier BD, la de Boyce-Codd se utiliza también frecuentemente y el resto son casos poco habituales.

Vamos a utilizar la tabla BIBLIOTECA que vemos a continuación como ejemplo para ir aplicando las distintas formas normales.

BIBLIOTECA					
CodLibro	Titulo	Autor	Editorial	NombreLector	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez Gómez, Juan	15/04/2005
1004	Visual Basic 5	E. Petroustos	Anaya	Ríos Terán, Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	Roca, René	16/04/2005
1006	Oracle University	Nancy Greenberg y Priya Nathan	Oracle Corp.	García Roque, Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez Gómez, Juan	18/04/2005

3.1 Primera forma normal (1FN)

“Una relación está en primera forma normal (1FN) si y sólo si todos sus atributos son atómicos”. Un atributo es atómico si sus elementos se pueden considerar como unidades indivisibles. O lo que es lo mismo: en una celda sólo debe existir un dato.

BIBLIOTECA no cumple el requisito de la Primera Forma Normal (1NF) de sólo tener campos atómicos, pues el nombre del lector es un campo que puede (y conviene) descomponerse en primer apellido, segundo apellido y nombre.

BIBLIOTECA 1FN							
Cod Libro	Titulo	Autor	Editorial	Primer Apellido	Segundo Apellido	Nombre Lector	FechaDev
1001	Variable compleja	Murray Spiegel	McGraw Hill	Pérez	Gómez	Juan	15/04/2005
1004	Visual Basic 5	E. Petroustos	Anaya	Ríos	Terán	Ana	17/04/2005
1005	Estadística	Murray Spiegel	McGraw Hill	Roca		René	16/04/2005
1006	Oracle University	Nancy Greenberg y Priya Nathan	Oracle Corp.	García	Roque	Luis	20/04/2005
1007	Clipper 5.01	Ramalho	McGraw Hill	Pérez	Gómez	Juan	18/04/2005

3.2 Dependencias funcionales: Segunda forma normal (2FN)

“Una relación *R* se encuentra en segunda forma normal (2ªFN), si y sólo si, está en 1ªFN y todos los atributos no clave dependen funcionalmente de manera completa de la clave primaria”.

El atributo *Y* depende funcionalmente del atributo *X* si y sólo si para un valor concreto de *X* obtenemos siempre el mismo valor de *Y*.

Se habla de **dependencia funcional completa**, cuando X está formado por más de un atributo y el atributo Y depende de X en su totalidad, no de una parte de los atributos de X.

- Si una Relación (tabla) tiene como clave primaria un solo atributo, esta se encuentra en 2ªFN.
- Para comprobar si una tabla está en 2ªFN, se comprueban uno a uno todos los atributos que no son clave, y si no tienen dependencia funcional completa de la clave principal, se extrae ese atributo a una nueva tabla junto con el o los atributos que forman parte de la clave de la que dependen.
- Al finalizar el proceso, de una Relación original que no estaba en 2ªFN pueden aparecer varias relaciones que si están en 2ªFN además de la relación original debidamente actualizada.
- La Relación original pierde los atributos no clave que se extraen a otras tablas pero NO los atributos que forman parte de la clave principal.
- En las nuevas tablas los atributos clave serán los no dependientes.

En BIBLIOTECA, el título es completamente identificado por el código del libro, pero el nombre del lector en realidad no tiene dependencia de este código, por tanto estos datos deben ser trasladados a otra tabla.

BIBLIOTECA 2FN			
CodLibro	Título	Autor	Editorial
1001	Variable compleja	Murray Spiegel	McGraw Hill
1004	Visual Basic 5	E. Petroustos	Anaya
1005	Estadística	Murray Spiegel	McGraw Hill
1006	Oracle University	Nancy Greenberg y Priya Nathan	Oracle Corp.
1007	Clipper 5.01	Ramalho	McGraw Hill

LECTORES 2FN			
CodLector	PrimerApellido	SegundoApellido	NombreLector
501	Pérez	Gómez	Juan
502	Ríos	Terán	Ana
503	Roca		René
504	García	Roque	Luis

Hemos creado una tabla para contener los datos del lector y también tuvimos que crear la columna **CodLector** para identificar unívocamente a cada uno. Sin embargo, esta nueva disposición de la base de datos necesita que exista otra tabla para mantener la información de qué libros están prestados a qué lectores. Esta tabla se muestra a continuación:

PRÉSTAMOS 2FN		
CodLibro	CodLector	FechaDev
1001	501	15/04/2005
1004	502	17/04/2005
1005	503	16/04/2005
1006	504	20/04/2005
1007	501	18/04/2005

3.3 Tercera forma normal (3FN)

“Una relación está en tercera forma normal (3FN) si y sólo si está en 2FN y todos los atributos no claves dependen de manera no transitiva de la clave primaria”

Es decir, los atributos que no son clave no pueden depender de otro atributo no clave. Esto significa que las columnas en la tabla deben contener solamente información sobre la entidad definida por la clave primaria y, por tanto, las columnas en la tabla deben contener datos acerca de una sola cosa.

- Si una Relación (tabla) solo tiene un atributo no clave, esta se encuentra en 3ªFN.
- Para comprobar si una tabla está en 3ªFN, se comprueban uno a uno todos los atributos que no son clave, y si tienen dependencia funcional de otro atributo no clave, se extraen ambos a una nueva tabla.
- Al finalizar el proceso, de una Relación original que no estaba en 3ªFN pueden aparecer varias relaciones que si están en 3ªFN además de la relación original debidamente actualizada.
- La Relación original pierde los atributos no clave dependientes, manteniendo los otros.
- En las nuevas tablas los atributos clave serán los no dependientes.

En nuestra BIBLIOTECA vemos que, además de información del libro, hay información del autor y de la editorial.

LIBROS 3FN	
CodLibro	Título
1001	Variable compleja
1004	Visual Basic 5
1005	Estadística
1006	Oracle University
1007	Clipper 5.01

AUTORES 3FN	
CodAutor	Autor
801	Murray Spiegel
802	E. Petroustsos
803	Nancy Greenberg y Priya Nathan
804	Ramalho

EDITORIALES 3FN	
CodEditorial	Editorial
901	McGraw Hill
902	Anaya
903	Oracle Corp.

Aunque hemos creado nuevas tablas para que cada una tenga sólo información acerca de una entidad, también hemos perdido la información acerca de qué autor ha escrito qué libro y las editoriales correspondientes, por lo que debemos crear otras tablas que relacionen cada libro con sus autores y editoriales.

LIBROS/AUTORES 3FN	
CodLibro	CodAutor
1001	801
1004	802
1005	801
1006	803
1006	804
1007	806

LIBROS/EDITORIALES 3FN	
CodLibro	CodEditorial
1001	901
1004	902
1005	901
1006	903
1007	901

Podemos comprobar que las tablas Lectores y Préstamos no necesitan ninguna modificación pues ya se encontraban en 3FN.

LECTORES 3FN			
CodLector	PrimerApellido	SegundoApellido	NombreLector
501	Pérez	Gómez	Juan
502	Ríos	Terán	Ana
503	Roca		René
504	García	Roque	Luis

PRÉSTAMOS 3FN		
CodLibro	CodLector	FechaDev
1001	501	15/04/2005
1004	502	17/04/2005
1005	503	16/04/2005
1006	504	20/04/2005
1007	501	18/04/2005

3.4 Forma normal de Boyce-Codd (FNBC)

“Una relación está en forma normal Boyce-Codd (FNBC) si y solo si está en 3FN y cada dependencia funcional no trivial tiene una clave candidata como determinante”

En términos menos formales, una tabla está en FNBC si está en 3FN y los únicos determinantes son claves candidatas.

Se define determinante en una relación (tabla) a un atributo del cual depende de forma funcional y completa, cualquier otro atributo de la relación.

Ejemplo

Consideremos una empresa donde un trabajador puede trabajar en varios departamentos. En cada departamento hay varios responsables, pero cada trabajador sólo tiene asignado uno. Tendríamos una tabla con las columnas:

IDTrabajador, IDDepartamento, IDResponsable

La única clave candidata es IDTrabajador (que será por tanto la clave primaria). Si añadimos la limitación de que el responsable sólo puede serlo de un departamento, este detalle produce una dependencia funcional ya que:

Responsable → Departamento

Por lo tanto hemos encontrado un determinante (IDResponsable) que sin embargo no es clave candidata. Por ello, esta tabla no está en FNBC. En este caso la redundancia ocurre por mala selección de clave. La repetición del par [IDDepartamento + IDResponsable] es innecesaria y evitable.

Solamente en casos raros una tabla en 3NF no satisface los requerimientos de la FNBC.

4 Normalización de modelos relacionales

Las tablas que definen el modelo relacional, pueden presentar ciertas anomalías, como la existencia de redundancia, incoherencia y problemas para mantener la integridad de los datos, por ello las tablas resultantes del esquema relacional deben seguir determinadas pautas que eviten esas anomalías y garanticen un resultado adecuado a las necesidades planteadas inicialmente por el cliente.

Para conseguir esto se aplica la Normalización. La normalización es una estrategia de abajo a arriba y que busca conseguir los objetivos siguientes:

- Eficiencia.
- Facilidad de uso y actualización de las tablas creadas.
- Eliminar anomalías de actualización, inserción y borrado.
- Conservar la información original (descomposición sin pérdida de información).
- Conservar las dependencias funcionales originales (descomposición sin pérdida de dependencias funcionales).
- No crear dependencias nuevas o relaciones inexistentes.

Dependencia funcional

Es una relación entre atributos de una misma tabla. Veamos un ejemplo:

Supongamos la tabla EMPLE, donde tenemos los campos SALARIO_NETO, RETENCIÓN y SALARIO_BRUTO. El salario neto es igual al salario bruto menos la retención. En realidad nos sobra uno de los 3 campos (por ejemplo el neto). Salario neto depende de salario base y las retenciones dependen del salario base.

Salario Neto --> Salario Base

Retenciones --> Salario Base

Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino los modelos mentales del usuario.

La **normalización** consiste en comprobar que cada tabla cumple unas reglas entre la clave principal y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización, si una regla no se cumple, la relación se debe descomponer en varias tablas que sí la cumplan.

Las dependencias funcionales son de primordial importancia a la hora de encontrar y eliminar la redundancia de los datos almacenados en las tablas. Un atributo depende de otro si para cada valor del primero existe un único valor del segundo.

Los tipos de dependencias son:

- **Completa.** Cuando un atributo depende de un par de atributos de manera conjunta, y no de manera individual.

Ejemplo: Supongamos una tabla de empresas y sueldos donde una persona puede estar en varias empresas con diferentes sueldos. Por tanto NOMBRE y EMPRESAS determinan SUELDO de manera completa. El sueldo que aparecerá depende de la persona que sea y de la empresa que sea.

- **Parcial.** Cuando un atributo depende de un único atributo.

Ejemplo: Mismo caso anterior, pero una persona solo puede trabajar en una sola empresa. SUELDO sólo está determinado por NOMBRE.

- **Transitiva.** Cuando un atributo depende de otro, que a su vez depende de un tercero.

Ejemplo: tabla NOMBRE, DIRECCIÓN y CÓDIGO POSTAL. DIRECCIÓN depende de NOMBRE y CÓDIGO POSTAL depende de DIRECCIÓN. Por tanto CÓDIGO POSTAL tiene una dependencia transitiva con respecto a NOMBRE.

- **Multivaluada.** Un atributo multievaluado depende funcionalmente de otro atributo.

Ejemplo: la tabla NOMBRE y la tabla TITULACIONES. Una persona puede tener varias titulaciones, por lo que TITULACIÓN presenta una dependencia multivaluada con respecto a NOMBRE.

Se dice que una relación está en una forma normal determinada si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente, teniendo en cuenta que es un proceso que crece, es decir, que para estar en una determinada forma normal debe estar en la forma normal anterior (obviamente excepto la primera). Una regla convierte una relación/tabla en dos o más relaciones.

Índice de contenidos

1 Paso del diagrama E/R al Modelo Relacional	2
1.1 Transformación de entidades	2
1.2 Transformación de relaciones	2
1.3 Restricciones que no pueden plasmarse en el diseño lógico	12
2 Reglas de integridad	12
3 Formas normales	13
3.1 Primera forma normal (1FN)	14
3.2 Dependencias funcionales: Segunda forma normal (2FN)	14
3.3 Tercera forma normal (3FN)	16
3.4 Forma normal de Boyce-Codd (FNBC)	18
4 Normalización de modelos relacionales	19