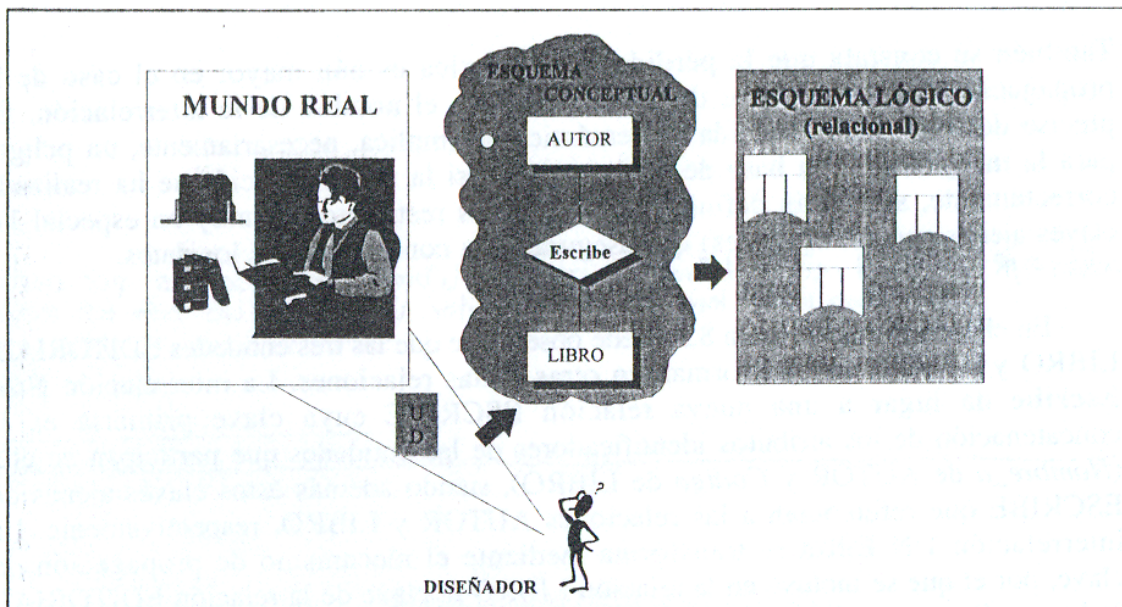


PASO DEL MODELO ENTIDAD-RELACIÓN AL MODELO RELACIONAL

En la **fase de análisis de los datos** se ha obtenido el **Modelo Entidad/Relación**. Dicho modelo corresponde al ámbito conceptual, al mundo de las ideas. Como toda fase de análisis, se centra en **Qué** datos hay que reflejar y no en **cómo**. En la fase de diseño, debe traducirse a una estructura que pueda ser manejada por el ordenador.



Al modelo de datos obtenido en el análisis se le denomina **Modelo Conceptual de Datos**. Y al modelo obtenido en el diseño se le denomina **Modelo lógico de datos**.

El mecanismo que utiliza el ordenador para representar y tratar la información se denomina **Sistema Gestor de Bases de Datos (S.G.B.D.)**.

Mientras el modelo conceptual se centra en reflejar la realidad recogida en los requisitos del sistema que corresponden a los datos, el modelo lógico se centra en adaptar esa solución a las limitaciones que impone el S.G.B.D. que se vaya a utilizar.

Existen diferentes opciones para construir el Modelo lógico de datos: jerárquico, Codasyl o relacional. Nos centraremos en el **Modelo Relacional** por ser el más utilizado.

El proceso consiste en traducir el Modelo E/R al Modelo Relacional. A este proceso también se le denomina **Paso a Tablas**.

1. Modelo de Datos Relacional

A finales de los años sesenta, el Dr. Codd propuso un nuevo modelo para los Sistemas de Gestión de Bases de Datos, que marcó un importante hito en la evolución de esta tecnología. Este modelo, que ha aparecido en el mercado a principios de los

años ochenta, y que en la actualidad siguen la mayoría de las bases de datos existentes, tiene como objetivos:

- Independencia física
- Independencia lógica
- Flexibilidad
- Uniformidad
- Sencillez

a la vez que presenta un sólido fundamento teórico que permite desarrollar un **álgebra** y un **cálculo** relacionales, así como toda la teoría de la normalización.

Así, el Modelo Relacional está basado en la teoría propuesta por Codd, donde los datos se estructuran lógicamente en tablas o relaciones.

1. Terminología del modelo relacional

Representa la información en base al uso de una **representación tabular plana**. Una tabla bidimensional mediante la cual se representa tanto los objetos como las relaciones entre ellos existentes en el dominio del problema.

Nombre			
atributo 1	atributo 2	atributo n
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx
xxxxxxx	xxxxxxx	xxxxxxx

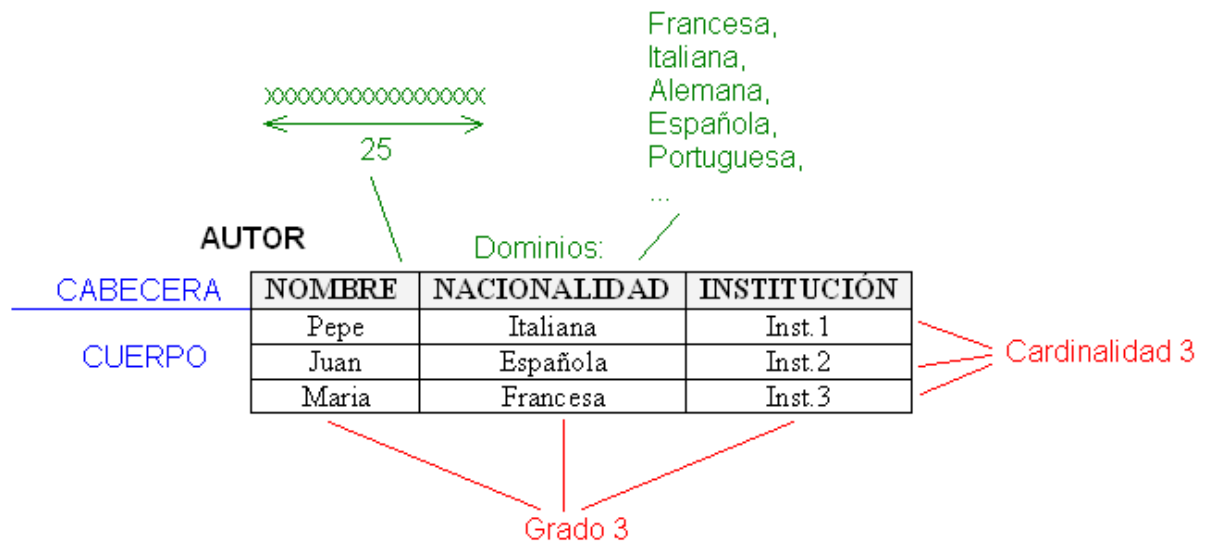
donde:

Nombre: Nombre de la relación

Atributo1, atributo2,... atributo n: Propiedades de la tabla (columnas)

Tupla1, tupla2,... tupla n: Valores de cada una de las columnas

Ejemplo:



- **Dominio:** Conjunto finito de valores **homogéneos** (todos son del mismo tipo) y **atómicos** (son indivisibles, sino pierden la semántica) caracterizados por su nombre. Tienen un tipo de datos asociado. En definitiva, de donde los atributos toman sus valores. Varios atributos pueden tomar valores del mismo dominio.
- **Dominio compuesto:** Combinación de dominios simples.

Ejemplo: Tres dominios → Día, Mes y Año Nombre y Apellidos
Dominio compuesto → Fecha NombreApell
- **Grado:** Número de atributos de la relación (nº de columnas)
- **Cardinalidad:** Número de tuplas de la relación (nº filas).
- **Cabecera:** Atributos de la tabla con sus dominios asociados (Estática)
- **Cuerpo:** Conjunto de tuplas que varían con el tiempo (Dinámica)

Una tabla es una matriz rectangular que presenta las siguientes propiedades:

- Cada **entrada** de la tabla, representa un **ítem de datos**.
- Todos los **ítems** de datos **de una columna son de la misma clase** y, por tanto, están definidos en el mismo dominio de datos y representan una misma propiedad o característica en el dominio del problema.
- **Cada columna** de la tabla tiene asignado un **nombre único** en el conjunto de columnas de esa tabla, aunque pueden existir tablas diferentes con columnas de igual nombre.
- Para cada tabla **todas las filas son diferentes**, no se admiten filas duplicadas.

- Tanto las **filas como las columnas** pueden ser **consideradas en cualquier secuencia** sin afectar, por ello, ni al contenido de la información ni a la representación semántica de la misma.

Una tabla o matriz rectangular se le denomina **relación**, a las filas de la misma se les denomina **tuplas** y al conjunto de sus columnas **dominio** de la relación. Así, una base de datos relacional estará formada por un conjunto de relaciones.

2. Esquema de la relación

Esquema de relación

De acuerdo con la definición matemática de relación: subconjunto del producto cartesiano de n dominios, existe un gran paralelismo entre esta definición y el concepto relacional de tabla (relación). Anteriormente se ha definido su modo de representación gráfico.' y en este apartado se muestra la notación empleada para describir una tabla, para lo que se necesita incluir mi nuevo concepto: esquema, como esquema de base de datos con el significado de diseño lógico de la misma y como esquema de relación (tabla) que es el conjunto de todos los atributos (campos) y sus dominios. Es decir, un esquema de relación muestra la estructura lógica de una tabla en el modelo relacional.

Un esquema de relación R está formado por un conjunto de atributos A_n y a cada uno de estos le corresponde un dominio D_n .

$$R(A) = R(A_1:D_1; A_2:D_2; A_3:D_3; \dots A_n:D_n)$$

Ejemplo:

ALUMNO (NIF: valores del NIF; nombre: nombres posibles; primer apellido: apellidos posibles;... teléfono: lista de números de teléfono).

Para simplificar, en las notaciones siguientes se elimina el dominio cuando sea obvio.

Alumnos (**#NIF**, nummat, nombre, primer apellido,... teléfono)

Alumnos (**#NUMMAT**, NIF, nombre, primer apellido,... teléfono)

3. Claves

Existen diferentes tipos de claves:

- **Clave candidata:** Conjunto de atributos que identifican **unívoca** (no se repita la clave) y **mínimamente** (no exista ningún subconjunto que cumpla la propiedad de unicidad) cada tupla de la relación. Características:
 - Al menos hay una por tabla y está formada por el conjunto de todos los campos.
 - Ha de tener el mínimo número posible de campos. Entre uno y el total de campos de la tabla.
 - No han de poseer campos superfluos en la clave candidata.

Ejemplo: En la relación:

EMPLEADOS (#CODIGO, DNI, NOMBRE, DIRECCION, TEL, SUELDO, COMISION,...)

podrían ser claves candidatas tanto el CODIGO como el DNI, una de éstas se elige como clave primaria (por ejemplo, el CODIGO), mientras que el resto se denominan claves alternativas.

- **Clave primaria:** La clave candidata que se escoge para identificar las tuplas de la relación. Si sólo existe una clave candidata esta será la primaria. Tiene las siguientes características:
 - Es única y mínima
 - Existe y es conocida.
 - Es NO NULA
 - No es ambigua
 - No se permiten claves primarias redundantes (repetidas)
 - Los campos que forman la clave primaria reciben el nombre de primos.

El modelo relacional presenta la **regla de integridad de entidad**, que establece que ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo. Se entiende por valor nulo una marca que sirve para indicar información desconocida o inaplicable.

CODJUGADOR	CLUB	JUGADOR	DORSAL	COD PAIS	CODDEMAR
713	CELTA	JOSEMA	NC	ESPAÑA	DEFENSA
716	CELTA	LAKABEG	NC	ESPAÑA	DEFENSA
719	CELTA	OSCAR VALES	NC	ESPAÑA	DEFENSA
721	CELTA	PATXI SALINAS	NC	ESPAÑA	DEFENSA
731	CELTA	MICHEL	NC	ESPAÑA	DEFENSA
743	CELTA	BERGES	NC	ESPAÑA	DEFENSA
786	CELTA	SANCHEZ	NC	ESPAÑA	DELANTERO
792	CELTA	GELI	NC	ESPAÑA	MEDIO
892	CELTA	DEL SOLAR	NC	ESPAÑA	MEDIO
905	CELTA	ITO	NC	ESPAÑA	MEDIO
912	CELTA	KARPIN	NC	ESTONIA	MEDIO
914	CELTA	REVIVO	NC	ISRAEL	DELANTERO
910	CELTA	RATKOVIC	NC	YUGOSLAVIA	MEDIO

Clave Primaria:

Única Mínima Conocida Pública No redundante

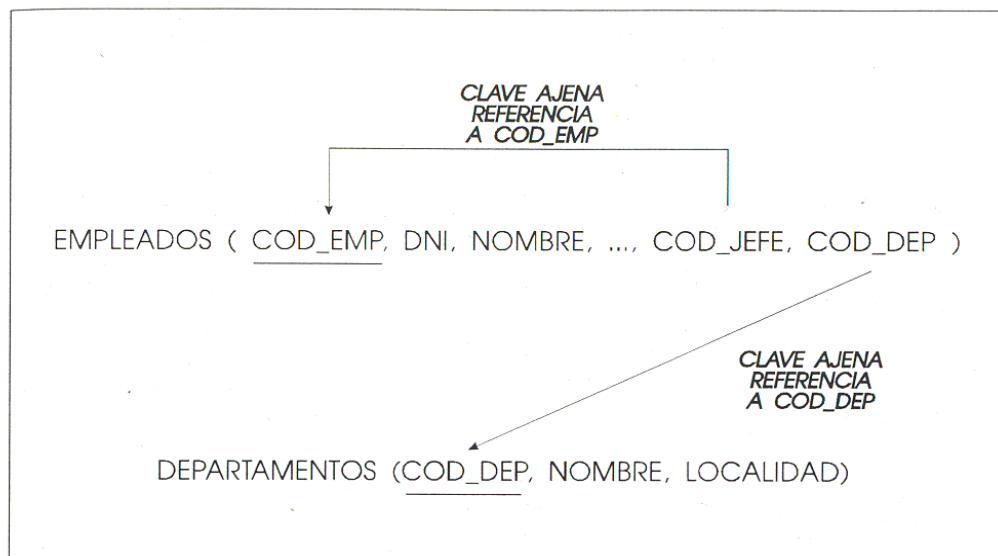
Figura 3.4. Características de la clave primaria.

- **Claves alternativas:** Elegida la clave primaria, el resto de claves candidatas reciben el nombre de claves alternativas.
- **Clave ajena:** En una relación R_2 , la clave ajena es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la clave primaria de una relación R_1 . Las claves ajenas, al compartir campos, permiten relacionar tablas, por lo que facilitan la relación entre las mismas, para originar nuevas tablas o vistas.
- Características:

- Su dominio ha de ser el mismo que el dominio de la clave primaria de la tabla a la que haga referencia.
- Se permite que los nombres de los campos sean diferentes de la clave ajena y que el nombre del campo de la clave primaria y el de la tabla asociada sean diferentes.
- Su valor puede ser duplicado o ser nulo.
- Los campos que la forman pueden formar parte de la clave primaria de la tabla a la que pertenecen.
- Una tabla que no se relacione con otras no tiene claves ajenas.

Ejemplo: Claves ajenas que presenta la relación EMPLEADO.

opciones:



Para las claves ajenas el modelo relacional especifica la **regla de integridad referencial** que establece que los valores de la clave ajena o bien coinciden con los de la clave primaria a la que referencia o bien son nulos. Por tanto, para cada clave ajena se debe especificar si puede o no tomar valores nulos. Además, es necesario determinar las consecuencias de ciertas operaciones (borrado y modificación) realizadas sobre tuplas de la relación referenciada; pudiéndose distinguir, en principio, las siguientes opciones:

- **Operación restringida (RESTRICT):** El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena.

Ejemplo: Para borrar un departamento no debería haber ningún empleado que estuviese trabajando en dicho departamento, si no, el sistema impediría el borrado.

- **Operación con transmisión en cascada (CASCADE):** El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contiene la clave ajena.

Ejemplo: Al modificar el código de un departamento en la relación DEPARTAMENTO, se debe modificar también dicho código en todos los empleados de nuestra base de datos que trabajen en dicho departamento.

- **Operación con puesta a nulos (SET NULL):** El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo **poner a nulos** los valores de las claves ajenas de la relación que referencia. Esta opción, obviamente, sólo es posible cuando el atributo que es clave ajena admite el valor nulo.

Ejemplo: Cuando se borra un departamento, a los empleados que trabajan en dicho departamento y que se encuentran en la relación EMPLEADOS se les coloque el atributo COD_DEP a nulos.

- **Operación con puesta a valor por defecto (SET DEFAULT):** El borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo **poner al valor por defecto** a la clave ajena de la relación que referencia; valor por defecto que habría sido definido al crear la tabla correspondiente.
- **Operación que desencadena un procedimiento de usuario:** En este caso, el borrado o la modificación de tuplas de la tabla referenciada pone en marcha un procedimiento definido por el usuario.

La opción seleccionada en caso de borrado es independiente de la de modificación.

4. Índices

Desde el punto de vista de la representación de los datos un índice es un componente que permite mostrar los datos con un determinado orden, por lo que es útil su empleo en consultas y actualizaciones, en las que primero se ordenan los datos y luego se seleccionan o actualizan, ya que los índices están asociados a proyecciones sobre relaciones (subconjuntos de atributos).

Como los índices contienen información ya almacenada en las tablas, son objetos redundantes en las bases de datos, pero presentan ventajas en cuanto a la velocidad cuando hay que efectuar recorridos por grandes tablas, ya que se recorren los índices, que contienen mucha menos información que las tablas de las que proceden con lo que las búsquedas y selecciones se realizan en mucho menos tiempo que si se ha de recorrer la totalidad de la tabla.

Los índices se crean y borran en tiempo de ejecución, cuando se necesiten, pudiéndose almacenar en la base de datos como un objeto más de la misma. Los índices se actualizan cada vez que se modifican los datos de la tabla.

El modelo relacional, a diferencia de los modelos precedentes -jerárquico y en red-, permite que los usuarios de la base de datos creen aquellos índices que necesiten. En cualquier caso hay que hacer notar que la creación, modificación o eliminación de índices no aporta nada desde la óptica de los datos, pero sí desde su velocidad de recuperación. Al menos, en cada tabla se crea un índice, que es su clave primaria.

2. Traducción de Esquemas E/R a Esquemas Relacionales

Mediante una serie de reglas fáciles de entender y de aplicar se realiza la traducción de los esquemas conceptuales realizados mediante el E/R a **esquemas lógicos** basados en el **modelo relacional**.

El proceso de traducción de esquemas conceptuales a lógicos consiste en la aplicación, **por pasos**, de una serie de **reglas** que, aplicadas a los esquemas conceptuales, transforman los objetos de estos esquemas en objetos pertenecientes a los esquemas lógicos.

La aplicación de las reglas va a dar lugar a la transformación de los tipos de entidad y los tipos de relación que forman parte de los esquemas conceptuales, en **tablas**, los únicos objetos que intervienen en los modelos lógicos relacionales.

En el paso del Modelo E/R al relacional se **pierde semántica**, puesto que tanto las entidades como las interrelaciones se transforman en relaciones, ya que en el modelo relacional sólo existe la relación para representar ambos tipos de objetos. También se constata pérdida de semántica en la propagación de clave, pero en este caso desaparece incluso el nombre de la interrelación 1:N. La pérdida de semántica no implica, sin embargo, un peligro para la integridad de la base de datos, ya que se definen restricciones de integridad referencial que aseguren la conservación de la misma.

1. Transformación de tipos de entidad

Todos los tipos de entidad presentes en el esquema conceptual se transformarán en tablas en el esquema relacional manteniendo, el número y tipo de atributos, así como la característica de identificador de estos atributos.

Ejemplo:



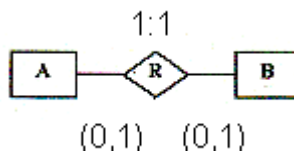
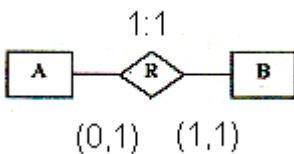
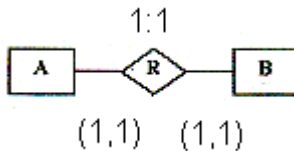
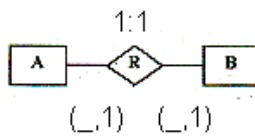
Los atributos claves de las tablas se representarán subrayados, en negrita o con el símbolo # delante del nombre.

Ejemplo: En el caso del ejemplo de la relación entre Libros y Autores, una vez aplicada esta regla se obtendrá el siguiente esquema relacional:

Libro (#ISBN, edición, editorial, páginas, fecha)
Autor (#DNI, nombre, apellidos)

2. Transformación de tipos de relación uno a uno

El proceso de transformación de los tipos de relación binarias en los cuales los tipos de entidad participan con **cardinalidad máxima uno** va a depender del valor de la **cardinalidad mínima** con la cual participa cada tipo de entidad en el tipo de relación. En base a este valor se pueden presentar los siguientes casos:

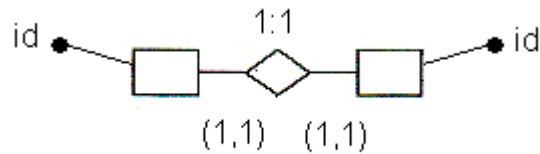


1. **Los dos tipos de entidad** participan de forma completa en el tipo de relación. Es decir, los dos tipos de entidad participan **con cardinalidad mínima uno**.
2. **Uno de los dos tipos de entidad** participa de forma parcial en el tipo de relación. Es decir, **con cardinalidad mínima cero**.
3. **Los dos tipos de entidad** participan de forma parcial en la relación. Es decir, las dos **con cardinalidad mínima cero**.

Aplicaremos una regla diferente para cada caso:

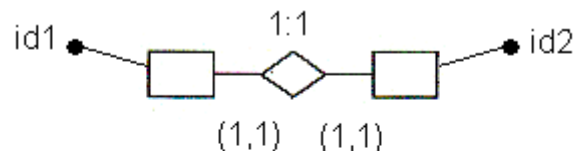
Si en un tipo de relación binaria los dos tipos de entidad participan de forma completa, es decir, ambos tipos de entidad participan con las cardinalidades mínima y máxima igual a uno, entonces:

- Si los dos tipos de entidad tienen la misma clave:
 1. Los dos tipos de entidad se transforman en una única tabla formada por la agregación de los atributos de los dos tipos de entidad.
 2. La clave de la tabla es el identificador de los tipos de entidad (es el mismo en ambas).



<u>id</u>	

- Si **los dos tipos de entidad tienen distinta clave**, cada tipo de entidad se transforma en una tabla y,
 1. Cada tabla tendrá como clave principal el identificador de cada uno de los tipos de entidad de los cuales se deriva.
 2. Cada tabla tendrá como clave ajena el identificador del otro tipo de entidad con el cual está relacionado.

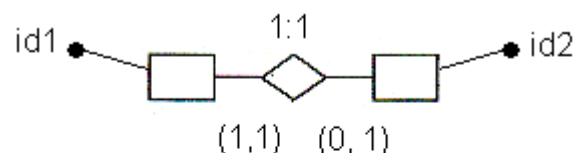


<u>id1</u>		(id2)

<u>id2</u>		(id1)

Si en un tipo de relación binaria 1:1 alguno de los tipos de entidad participa de forma parcial (cardinalidad mínima igual a 0), entonces, cada tipo de entidad se transforma en una tabla y se procede de alguna de las dos formas siguientes:

1. El identificador del tipo de entidad que participa de forma completa pasa como atributo de la tabla correspondiente a la transformación del otro tipo de entidad. En este caso este atributo no podrá tomar valores nulos para diferentes tuplas de esta tabla, no generándose ninguna tabla para el tipo de relación.

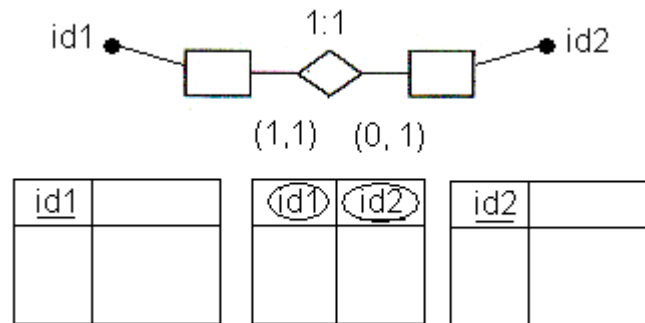


<u>id1</u>	

<u>id2</u>		(id1)

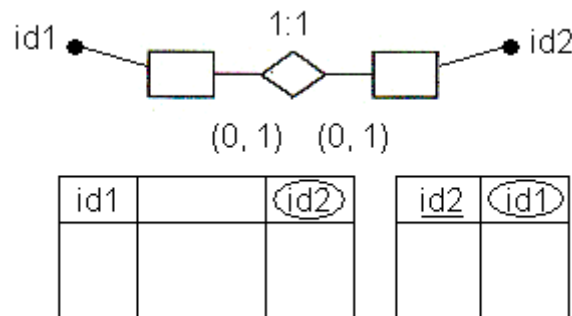
2. Se construye una nueva tabla correspondiente al tipo de relación formada por los atributos identificadores de los dos tipos de entidad. Se considerará como clave de esta tabla al identificador del tipo de entidad que participa de forma parcial en el tipo de interrelación. Los atributos de esta tabla mantendrán

referencias con las claves de las tablas correspondientes a la transformación de los tipos de entidad.



Si en un tipo de relación binaria ambos tipos de entidad participan de forma parcial, cada uno de ellos se transforma en una tabla y se procede de alguna de las dos formas siguientes:

1. Los identificadores de cada uno de los tipos de entidad pasan a formar parte como atributos de las tablas correspondientes al otro tipo de entidad. Estos atributos actuarán como claves ajenas en estas tablas.

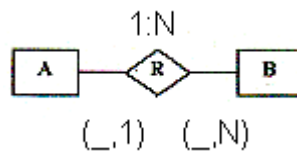


2. Se construye una nueva tabla correspondiente al tipo de relación y cuyos atributos serán los identificadores de los dos tipos de entidad. La clave principal de la tabla generada será el identificador de uno de los tipos de entidad y, necesariamente, se definirá como clave candidata al identificador del otro tipo de entidad.

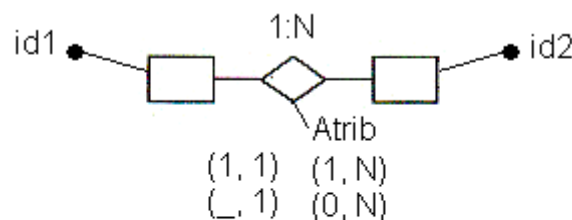
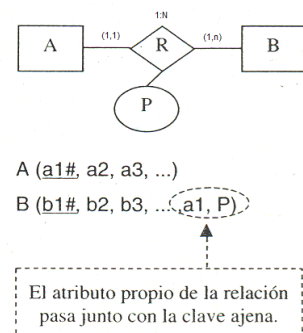
El primer criterio de transformación es el más acertado cuando el tipo de relación no tenga atributos que lo cualifiquen. En caso contrario, cuando el tipo de relación tenga atributos asociados, el segundo criterio es más acertado, pues aporta una mayor claridad a la representación lógica del problema.

3. Transformación de tipos de relación uno a muchos

La transformación de esquemas conceptuales en los que estén presentes tipos de **relación binarias** en los que un tipo de entidad participa con **cardinalidad máxima uno** y el otro tipo de entidad participa con **cardinalidad máxima muchos**, es muy similar a la transformación descrita anteriormente. Se aplicarán reglas **diferentes en función de las cardinalidades mínimas** con las cuales participa cada tipo de entidad en el tipo de relación.



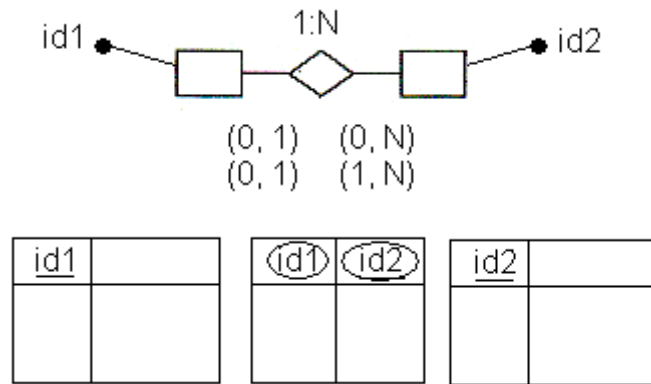
Si en un tipo de relación binaria 1:N ambos tipos de entidad participan de forma total, o el tipo de entidad que interviene con cardinalidad máxima muchos participa de forma parcial, entonces, cada tipo de entidad se transforma en una tabla, el identificador del tipo de entidad que participa con cardinalidad máxima uno pasa a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima muchos. Este atributo será definido como clave ajena de esta tabla manteniendo una referencia con la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima uno. Si el tipo de interrelación tuviera atributos asociados, estos atributos pasan a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima muchos.



Si en un tipo de 1:N ambos tipos participan de el tipo de entidad con cardinalidad participa de entonces cada se transforma en genera una correspondiente al tipo de relación. Esta tabla estará formada por los identificadores de los tipos de entidad que intervienen en el tipo de relación y por todos los atributos asociados al tipo de relación. La clave principal de esta tabla será el atributo identificador correspondiente al tipo de entidad que interviene con cardinalidad máxima muchos, y será necesario definir como claves ajenas los atributos identificadores correspondientes a los dos tipos de entidad.

<u>id1</u>		<u>id2</u>	<u>id1</u>	Atrib

relación binaria de entidad forma parcial, o que interviene máxima uno forma parcial, tipo de entidad una tabla y se nueva tabla

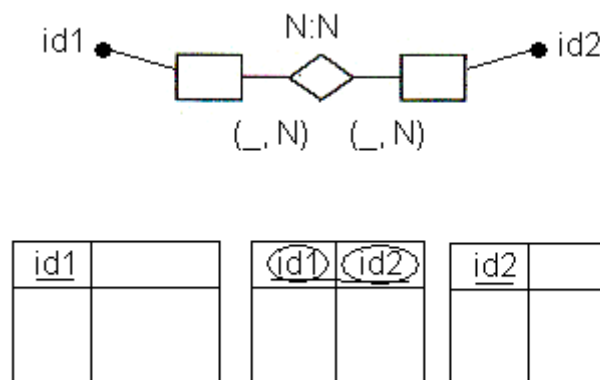
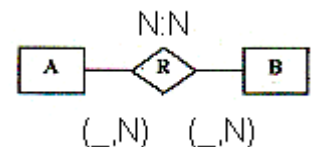


En las entidades débiles, la clave de la entidad fuerte debe ir la primera y, a continuación, los discriminadores de la débil. **Ambos elementos serán claves.**

4. Transformación de tipos de relación muchos a muchos

En esta clase de relación el proceso de transformación no depende de la cardinalidad mínima con la que interviene cada tipo de entidad en la relación, sino que siempre se aplicará la siguiente regla:

En un tipo de relación binaria N:N cada tipo de entidad se transforma en una tabla y se genera una nueva tabla para representar al tipo de relación. Esta tabla estará formada por los identificadores de los tipos de entidad que intervienen en el tipo de relación y por todos los atributos asociados al tipo de relación. La clave principal de esta tabla será la agregación de los atributos identificadores correspondientes a los tipos de entidad que intervienen en el tipo de relación.



Es importante el orden de concatenación de los atributos en la clave compuesta.

Orden de los atributos en las claves compuestas

Claves compuestas son las formadas por la agregación de varios atributos. La definición de una clave compuesta supone un problema añadido, no trivial, para el diseñador de la base de datos.

El orden de los atributos que forman parte de una clave compuesta deber ser, transparente para el usuario final, pero es importante para el diseñador de la base de datos.

El que un conjunto de atributos se defina como clave implica una serie de compromisos que afectan tanto a la visión lógica como física de la base de datos:

- No pueden existir dos objetos (tuplas de la tabla) con el mismo valor de ellos.
- Al identificar sin ambigüedad a una tupla, los métodos de acceso a esta información se realizarán, generalmente, en base a criterios de selección en los cuales sólo intervenga uno o todos los atributos que forman parte de la clave.
- El diseñador establecerá una organización física tal que estos procedimientos de acceso tengan un desempeño aceptable.

Si nos centramos en los aspectos físicos y consideramos que se organiza un índice por la clave para un acceso rápido a la información, el orden de los atributos en este índice es importante. No es lo mismo, ni se obtendrán los mismos resultados.

Ejemplo: Si el índice se define como la agregación de carnet# y matricula que matricula y carnet#. En el primer caso las entradas del índice estarán ordenados por los identificadores de los conductores, mientras que en el segundo lo estarán ordenadas por los de los autobuses.

Condu-Auto	(carnet#, matricula, trayecto)
Auto-Condu	(matricula, carnet#, trayecto)

El diseñador de la base de datos deberá tener en cuenta los requerimientos operacionales impuestos y definidos en el análisis de requerimientos del sistema para la definición del orden de los atributos.

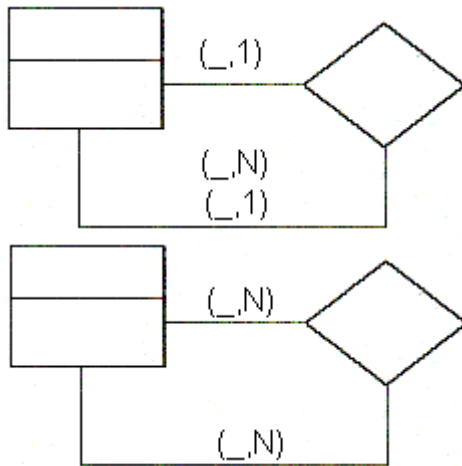
5. Transformación de tipos de relación N-arias

En el proceso de transformación de las relaciones en las que intervienen más de dos tipos de entidad se debe aplicar las mismas reglas: Cada tipo de entidad se transforma en una tabla y el tipo de relación se transforma en una nueva tabla cuyos atributos son los atributos identificadores de los tipos de entidad participantes en el tipo de relación y los atributos asociados al tipo de relación. La clave de esta tabla será la agregación de los atributos identificadores de los tipos de entidad.

Si todas las **cardinalidades máximas es de muchos** $N:N:N$, todos los identificadores forman la clave. Cuando algún tipo de entidad participa en el tipo de relación con **cardinalidad máxima uno** $1:N:N$ el identificador de este tipo de entidad no pasa a formar parte de la clave de la tabla que se deriva del tipo de relación, así el tipo de relación está cualificado por el atributo que identifica a este tipo de entidad.

6. Transformación de tipos de relación reflexivas

Los tipos de **relaciones reflexivas** son **relaciones binarias** en las que únicamente interviene un tipo de entidad. Se pueden presentar dos casos:



1. El tipo de entidad participa en uno de sus papeles con cardinalidad máxima uno, y en el otro papel con cardinalidad máxima muchos, o bien ambos participan con cardinalidad máxima uno.
2. El tipo de entidad participa, en sus dos papeles, con cardinalidad máxima muchos.

En el **primer caso**, se procede de una de las dos formas siguientes:

1. Se genera una tabla para el tipo de entidad, y en esta tabla se añade como clave ajena el identificador del tipo de entidad (cuando la cardinalidad mínima es uno).
2. Se genera una tabla para el tipo de entidad y otra para el tipo de relación formada por los atributos asociados al tipo de relación, el identificador del tipo de entidad el cual será la clave de la tabla, representando al objeto que forma parte de otros objetos, y de nuevo el identificador del tipo de entidad representando al objeto del cual forma parte otros objetos de menor categoría (cuando la cardinalidad mínima es cero)

En el **segundo caso**:

Se genera una tabla para la entidad y otra tabla para la de relación, siendo los atributos de esta tabla:

- Los atributos asociados al tipo de relación
- El identificador del tipo de entidad desempeñando uno de los papeles en el tipo de relación.
- El identificador del tipo de entidad desempeñando el otro papel en el tipo de relación.

La clave de esta tabla será el identificador del tipo de entidad (por duplicado).

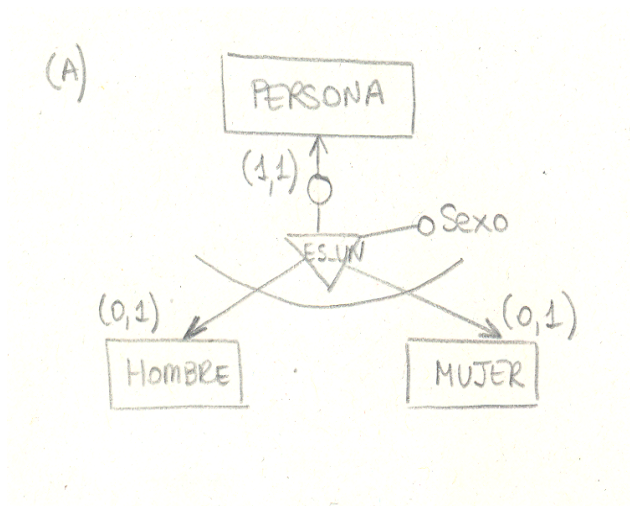
7. Eliminación de las relaciones jerárquicas

En el modelo conceptual se pueden representar las relaciones jerárquicas existentes entre las entidades pero estas no se pueden representar en el modelo lógico relacional.

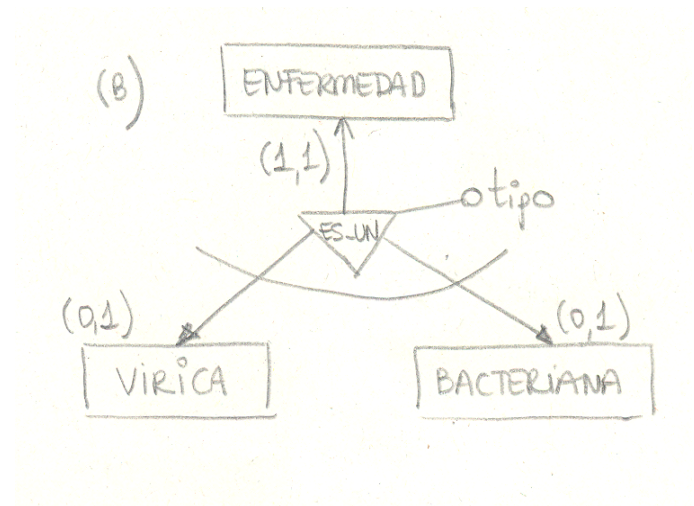
El modelo relacional no dispone de mecanismos fáciles de usar que permitan la representación de relaciones jerárquicas y, por tanto, es necesaria la eliminación de las relaciones jerárquicas como paso previo al proceso de transformación de los esquemas conceptuales a relacionales.

En el **proceso de eliminación** de las relaciones jerárquicas se deberá aplicar alguna de las reglas que se describen y cuya elección va a depender de:

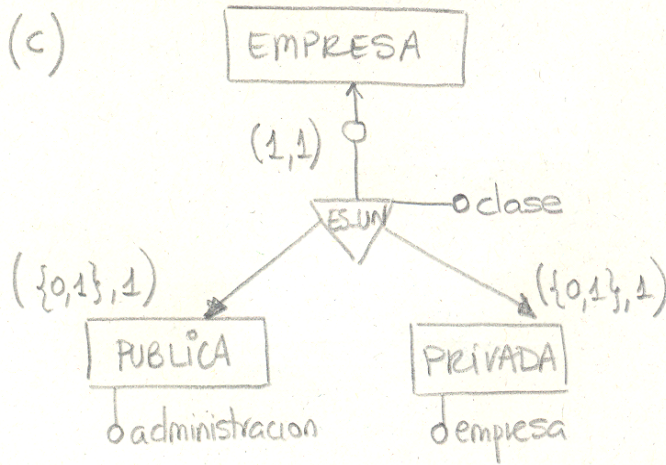
- La **magnitud de la especialización** que los subtipos de entidad tienen con respecto al supertipo o tipo de entidad más general. La especialización supone que los subtipos de entidad tienen asociados diferentes atributos que diferencian a un subtipo de entidad del resto de los subtipos que mantienen un tipo de relación jerárquica con un mismo supertipo de entidad.
- El **tipo de especialización** que representa el tipo de relación jerárquica, la cual puede ser alguna de las siguientes: total exclusiva, parcial exclusiva, total inclusiva y parcial inclusiva.



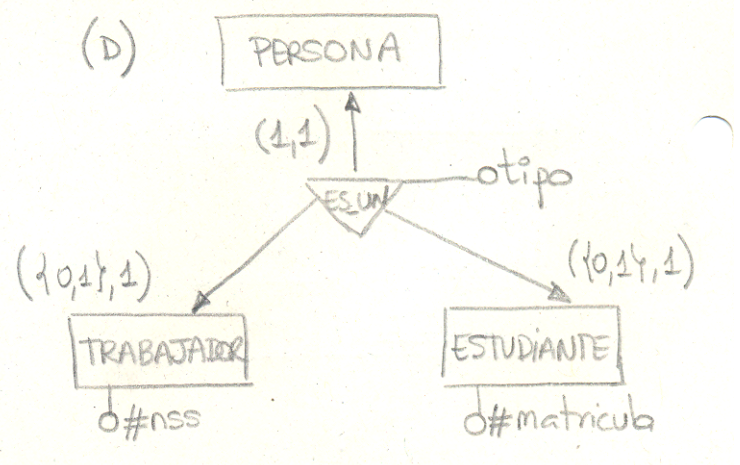
Exclusiva; Total sin solapamiento



Exclusiva; parcial sin solapamiento



Inclusiva; Total con solapamiento



Inclusiva; Parcial con solapamiento

Guías de referencia

Como las relaciones jerárquicas son un caso especial, se pueden dar algunas guías que sirvan de referencia, pero en la mayoría de los casos, va a depender del problema en concreto. Estas guías son las siguientes:

- Se creará una tabla para la entidad supertipo. A no ser que tenga tan pocos atributos que dejarla sea una complicación.
- Si la entidad subtipo no tiene atributos y no está relacionada con ninguna otra entidad => desaparece.
- Si la entidad subtipo tiene algún atributo => se crea una tabla. Si no tiene clave propia hereda la de la entidad supertipo.
- Si la relación es exclusiva, el atributo que genera la jerarquía se incorpora en la tabla de la entidad supertipo. Si se ha creado una tabla para cada una de las entidades subtipo, no hace falta incorporar dicho atributo a la entidad supertipo.