

## Unidad 1 -

### Tema 1. - INTRODUCCIÓN A LAS BASES DE DATOS

1. ¿QUÉ ES UNA BASE DE DATOS?
2. BASES DE DATOS Y SGDB
3. CARACTERÍSTICAS DE UNA BASE DE DATOS
4. MODELOS Y TIPOS DE BASES DE DATOS
5. ¿QUÉ SE QUIERE DECIR CUANDO SE HABLA DE UN SISTEMA DE BASE DE DATOS RELACIONAL?
  - 5.1 PROBLEMAS DE LAS BASES DE DATOS SIMPLES
6. BASES DE DATOS RELACIONALES
7. RELACIONES
  - 7.1 TIPO DE RELACIONES
8. PLANIFICACION DE LAS BASES DE DATOS

#### 1. ¿QUÉ ES UNA BASE DE DATOS?

---

El mundo real está formado por *entidades* con las que interactuamos, como personas o documentos. En el marco de una empresa las entidades son clientes, proveedores, facturas; en la sanidad, historial clínico, paciente, etc. Para dominar con eficacia la parcela del mundo real con la que interrelacionamos, necesitamos almacenar las entidades como datos útiles para nuestra actividad. Si, por ejemplo, tenemos, más de treinta clientes, y gestionamos una media de cinco facturas diarias, tendremos que recurrir a algún sistema de gestión de datos, como pueden ser los clásicos ficheros, en vez de fiarnos de nuestra propia memoria. En todos los ámbitos siempre se utiliza algún **sistema de gestión de la información**, desde la elemental agenda telefónica hasta las sofisticadas bases de datos de grandes organizaciones administrativas.

Tradicionalmente, se han utilizado tarjetas u hojas que se agrupaban en archivadores. En las empresas de hace décadas era habitual la presencia de grandes archivadores que clasificaban de algún modo la información que se gestionaba en la empresa. Un archivador podría guardar las fichas de los clientes, otro archivador las fichas de los proveedores, otro las de los productos a vender, etc. El conjunto de todas esos archivadores con sus fichas sería una antigua base de datos, es decir, una colección de datos relacionados entre sí, almacenados según unos criterios precisos, necesarios para la actividad que se debe desempeñar .

#### 2. BASES DE DATOS Y SGDB

---

El sistema de gestión de datos manual, basado en fichas y archivadores, tiene evidentes inconvenientes respecto a un sistema automatizado. Los ordenadores ofrecen una impresionante capacidad de gestión de información, de manera extremadamente rápida. Por este motivo, existen programas para gestionar bases de datos por ordenador, llamados sistemas de gestión de bases de datos, también conocidos con el acrónimo SGBD (Sistemas de Gestión de Bases de Datos). Es importante distinguir entre una base de datos y un SGBD (Sistema de Gestión de Bases de Datos).

*Una **base de datos** es un conjunto de datos organizados y relacionados lógicamente para un uso determinado, por ejemplo, una base de datos con la dirección y teléfono de nuestros clientes y nuestros proveedores, una base de datos con el autor, título y editorial de los libros de una biblioteca, etc.*

*Por el contrario, un **sistema de gestión de bases de datos** o **SGBD** es un programa informático que permite crear, modificar y mantener una base de datos.. Un SGBD puede trabajar con distintas bases de datos (la base de datos de discos, la base de datos con direcciones y teléfonos, la base de datos de libros, etc.).*

### 3. CARACTERÍSTICAS DE UNA BASE DE DATOS

---

Una base de datos es un sistema informatizado cuyo propósito principal es mantener información y hacer que esté disponible en el momento requerido. Esta información es persistente dentro del sistema, es decir, una vez introducida en él, se mantiene hasta que el usuario decida eliminarla.

Como su propio nombre indica, los programas de gestión de bases de datos realizan una serie de operaciones, que permiten el acceso y la manipulación de los datos almacenados en el ordenador.

Estas operaciones son:

- **Creación:** Consiste en definir la estructura de la base de datos, es decir, el número de campos, el nombre tipo y tamaño.
- **Altas:** Introducción de datos.
- **Bajas:** Borrado de datos
- **Modificaciones:** Cambios en los datos ya introducidos.
- **Consultas.** Búsqueda de datos bajo criterios definidos por el usuario.
- **Ordenación:** Clasificación de los datos por uno o varios criterios.
- **Informes:** Impresión de listados siguiendo determinados criterios.
- **Formularios:** Son plantillas de entrada de datos que se utilizan para rellenar nuevos registros de la base de datos

Los sistemas de bases de datos se diseñan para manejar grandes cantidades de información. El manejo de datos incluye, tanto la **definición** de las estructuras para el

almacenamiento de la información, como los mecanismos para el **manejo** de la misma. Algunas de las **ventajas** de usar las bases de datos son:

- **Independencia física:** La gestión de los datos se hace a alto nivel, es decir, por comandos o menús fácilmente inteligibles por el usuario, quedando ocultos todos los detalles físicos. Las aplicaciones de bases de datos no necesitan conocer el formato de los registros y archivos que procesa. Sólo saber cuantos datos y de que tipo se necesitan procesar de la base de datos. El SGBD es el que localiza los datos en los archivos.
- **Evitan la redundancia:** La redundancia es la repetición de datos idénticos. Por ejemplo, las bibliotecas suelen estar provistas de ficheros de autores y ficheros de títulos, que duplican la información. Una base de datos debe permitir consultar los datos por cualquier concepto, sin que ello implique que se repitan en distintas localizaciones. Para ello se descompone un mismo fichero en varios, tanto como sea necesario para evitar repeticiones indeseables. Veremos esto con detalle un poco más adelante.
- **Evitan la inconsistencia.** Directamente relacionado con lo anterior. Cuando existen varias copias de un dato, puede ocurrir que haya inconsistencia, es decir, que las copias no concuerden ( por ejemplo, al modificar un dato en un sitio y no en las copias). Las bases de datos bien diseñadas evitan este problema.
- **Evita programación para las tareas más comunes:** Los programas de bases de datos incorporan las funciones básicas sobre datos que, de otra manera, deberían codificarse mediante un lenguaje de programación.
- **Menor coste:** Los programas de bases de datos son productos estándar, que siempre resultan más baratos que el desarrollo de una aplicación a medida.
- **Es posible aplicar restricciones de seguridad para el acceso a los datos.** Se pueden configurar diferentes perfiles de usuario, con distintos privilegios que permitirán el acceso a distintos grupos de datos, y con diferentes posibilidades de operación. Así, unos usuarios podrán leer sólo una parte de los datos, otros podrán leerlos y modificarlos, otros podrán crear nuevas estructuras de datos, etc..

## 4. MODELOS Y TIPOS DE BASES DE DATOS

---

Existen muchos modelos de bases de datos, es decir, diferentes maneras en que los programas de bases de datos organizan los datos. Existen tres modelos:

- ❑ **Modelo jerárquico**
- ❑ **Modelo de red**
- ❑ **Modelo relacional.**

Actualmente sólo se emplea el modelo relacional, concebido por Codd, basado en su álgebra relacional.

## 5. ¿QUÉ SE QUIERE DECIR CUANDO SE HABLA DE UN SISTEMA DE BASE DE DATOS RELACIONAL?

Se refiere a una manera de organizar la información que contiene la base de datos.

En los **sistemas relacionales**, la información se representa mediante **tablas**. Las tablas están formadas por filas, denominadas **registros**, que equivalen a los casos de entidades almacenados y por columnas, denominadas **campos** representación de los atributos de los tipos de entidades. La intersección de una fila y de una columna se llama **dato** o elemento de la tabla.

Una tabla es una estructura de datos que contiene información común. Por ejemplo, una tabla llamada «abogados» podría estar construida por campos tales como: nombre del abogado, apellidos, dirección, especialidad, etc. Las tablas pueden, a su vez, estar relacionadas entre si, lo cual genera una visión completa del sistema.

Si tuvieras que pensar un modo de representar la información que contiene una agenda, seguro que terminarías utilizando una estructura similar a una tabla como la mostrada en la figura 1.1

Nombre	Apellido	Dirección	Población	Provincia	c P	Teléfono
Andrés	Jiménez	Castelao, 9	Ourense	Ourense	38981	38-38-38
Enrique	Guzmán	Fuenlabrada, 22	Sada	A Coruña	34256	34-34-34
Antonio	Redondo	Orillamar, 69	Vigo	Pontevedra	36200	36-36-36
María	Buendía	Azabache,70	Lugo	Lugo	35654	35-35-35

Figura 1.1 Tabla de una agenda

En esta tabla podemos ver los datos, con una estructura organizada. Recuerda ahora la definición de una base de datos (conjunto de información *organizada* lógicamente) ¿Podremos considerar, ahora, que nuestra tabla es una base de datos?

En los programas de bases de datos relacionales existe una manera de limitar el acceso del usuario a partes específicas de las bases de datos, mediante la realización de las llamadas **vistas**. Las vistas son imágenes lógicas de las bases de datos que no tienen por qué corresponderse con la estructura real de la base de datos, es decir, con tablas auténticamente existentes. Las vistas no se diferencian en absoluto para el usuario de las tablas propiamente dichas, pero es la manera de concentrarse exclusivamente en las columnas que le interesan o de impedirle el acceso a las que no le interesan.

En la mayoría de los casos, una única tabla no satisface todos nuestros problemas y será necesario utilizar más de una. Por este motivo a la hora de diseñar las tablas (de definir los campos que va a contener) hay que evitar la duplicidad de la información. Por ejemplo, sería una redundancia almacenar los datos personales de un cliente en cada uno de los pedidos que efectúe. Deberíamos tener una tabla con los datos fijos del cliente, y otra tabla con los datos de cada uno de sus pedidos. El enlace entre las diferentes tablas se realiza por medio de un campo común que contiene la misma información en todas y cada una de las tablas que se desean relacionar. En el caso que acabamos de mencionar, el campo común deberá ser un código que le asignemos al cliente.

## 5.1 PROBLEMAS DE LAS BASES DE DATOS SIMPLES

Las tablas son estructuras de datos muy sencillas y entendidas por todos. En algunas ocasiones una base de datos muy simple, con una única tabla, nos puede solucionar muchos problemas, por ejemplo, en el caso de nuestra agenda telefónica. Sin embargo, en otros casos no será suficiente, porque se nos presentarán algunos problemas.

Pensemos en el siguiente ejemplo: una tabla de los empleados de una empresa con los siguientes campos:

- Nombre
- Apellidos
- Dirección
- Código postal
- Mes de pago
- Fecha pago
- Salario bruto

La tabla de la figura 1.2 la utilizaremos para conocer la cantidad y el día en que vamos a pagar a nuestros empleados cada mes. Por ello, tenemos los datos de todos nuestros empleados repetidos en varios registros (filas) a lo largo de la tabla.

Nombre	Apellido	Dirección	C-P.	Mes	Fecha Pago	Sal. Bruto
Andrés	Guzmán	Generalitat	11111	Enero	31/01/98	2800000
Anastasio	González	Reinosa	22222	Enero	30/01/98	3500000
Zacarías	Tarraís	Trafalgar	33333	Enero	29/01/98	1875000
Enrique	Casero	Azabache	44444	Enero	28/01/98	2000000
Zacarías	Tarrals	Trafalgar	33333	Febrero	29/02/98	1875000
Andrés	Guzmán	Generalitat	11111	Febrero	31/02/98	2800000
Anastasio	González	Reinosa	22222	Febrero	30/02/98	3500000

Figura 1.2.

Centrémonos en un empleado en concreto: Zacarías Tarraís. Si te fijas con atención, observarás que todos los datos de este empleado coinciden, menos los referentes al mes y fecha de pago. Es decir, los registros que hay en la tabla de cada empleado contienen la misma información sobre los apellidos, la dirección, la ciudad y el salario bruto y difieren en el mes y en la fecha de pago.

Si hemos dicho que el objetivo de la tabla es conocer el **importe** y la **fecha de pago** de cada empleado cada mes, esta tabla satisface nuestras necesidades, pero con un coste alto, ya que hay que repetir una gran cantidad de datos para obtener información. El resto de información del registro 5 no vale para nada, ya que tenemos esa información en el registro 3.

- Otro problema que surge junto a la **duplicación de información** es la dificultad que encierra la modificación de un dato. Imagina que Andrés Guzmán cambia de dirección, de Castela, 9 a Soage Hermida, 6. Tendrías que seguir un proceso para poder cambiar este dato y la información de la tabla sea correcta. Tendrías que buscar todos los registros en que apareciera este empleado, y modificar, uno a uno, el campo dirección. Parece claro que cuando tuvieras los datos de 12 meses de pagos de un empleado (contando con solo 1 año), este proceso se alargaría ya demasiado. No tiene sentido almacenar así la información.
- Un problema adicional es que, si se te olvida, o se te pasa modificar dicho dato en uno de los registros, se perderá la **coherencia de los datos**. Esto es, la misma tabla indicará que

Andrés Guzmán vive en dos direcciones, provocando confusión a la hora de darle tratamiento a los datos.

## 6. BASES DE DATOS RELACIONALES

En la figura 1.2, nos damos cuenta de que lo normal es que un empleado trabaje varios meses o años en la empresa y, por tanto, hubiera varios registros dedicados a él. Esto hace que haya datos repetidos para cada empleado.

Veamos otro ejemplo que ilustre los problemas de la información redundante. En la figura 1.3, tenemos una tabla con los libros vendidos en una tienda de libros de informática, y con los clientes que los han adquirido. Esta tabla podría ser la que usamos para enviar las facturas a nuestros clientes al terminar el mes.

ISBN	Título	Autor	Cod_cli	Nom_Cli	Dirección	Fecha Venta
111111	Manual de ms-dos	José Luis Gamuza	0001	Carolina	Orillamar, 33	31/01/94
222222	Manual de Word	Edward Jones	0002	Jesús	Urzáiz, 55	14/02/94
333333	Manual de Excel	Venerando Solís	0004	Susana	Coya, 44	24/01/94
444444	Manual de Access	Alberto Rodríguez	0003	Alicia	Anónimo,123	31/01/94
111111	Manual de ms-dos	José Luis Gamuza	0004	Susana	Coya, 44	09/01/94
333333	Manual de Excel	Venerando Solís	0003	Alicia	Anónimo,123	23/01/94
222222	Manual de Word	Edward Jones	0001	Carolina	Orillamar, 33	31/01/94

Figura 1.3.

Volvemos a tener el mismo problema de la duplicación de la información, tanto de los datos del cliente cada vez que compra un artículo, como del artículo en sí. Por lo tanto estamos como en el ejemplo anterior, el tener que introducir la misma información más de una vez.

En los distintos libros comprados por un mismo cliente aparecen repetidos todos los datos personales, volviendo a ser poco operativo tratar la información de este modo.

¿Cómo podemos solucionar este problema? Supón que, en vez de esta única tabla, utilizamos dos, y en esas dos tablas almacenamos información distinta pero con un campo en común. Supongamos, por ejemplo la estructura de las dos tablas que se muestran a continuación:

ISBN	Título	Autor	Cod-Cli	Fecha_venta
------	--------	-------	---------	-------------

Figura 1.4. Tabla de Artículos.

Cod_Cli	Nombre	Dirección
---------	--------	-----------

Figura 1.5. Tabla de clientes.

Ahora ya no está repetida la información de los clientes en cada libro que hemos vendido (el único dato que aparece es un número de cliente que le hemos asignado según el

número que tiene asignado en la contabilidad) aunque sigue repetida la información de cada libro (título y autor).

Si analizamos los datos existentes en estas dos tablas, veremos que ambas tablas se pueden enlazar por el campo **Cod\_Cli**. En la tabla de artículos vendidos tenemos números de clientes que coinciden con los números de la tabla de clientes.

En nuestro caso, nuestras tablas podrían llamarse **Clientes** y **Artículos** respectivamente.

Si nos centramos en los datos de un cliente en particular, si éste cambia de domicilio, lo único que tendríamos que hacer es ir a la tabla de clientes y modificar este dato. De este modo, evitaríamos tener que buscarlo por todos los registros, como ocurría con el ejemplo anterior. Podríamos, también, completar la tabla añadiendo datos personales pertinentes como **Apellidos**, **Cod-Postal**, **Teléfono**, u otros que fueran necesarios y que sólo fueran dependientes del cliente, no de sus compras.

<b>Cod_Cli</b>	<b>Nombre</b>	<b>Apellidos</b>	<b>Dirección</b>	<b>Cod-Postal</b>	<b>Teléfono</b>
----------------	---------------	------------------	------------------	-------------------	-----------------

*Figura 1.5.a. Tabla de clientes.*

## 7. RELACIONES

Un aspecto fundamental de las bases de datos es su capacidad para relacionar sus elementos entre entre sí. Todas las bases de datos de organizaciones como empresas, escuelas, asociaciones de vecinos, federaciones deportivas, colegios profesionales, etc, todas implican enlaces entre entidades que las componen. En cuanto se necesita una organización y clasificación de la información mínimamente compleja, es necesario recurrir a una base de datos relacional

En nuestro primer ejemplo de base de datos con varias tablas, vamos a ver la forma en que una persona utilizaría estas dos tablas para acceder a su información, Para ello, supongamos que deseamos localizar el teléfono del cliente que nos ha pedido el Manual Imprescindible de Word, ya que queremos avisarle de que no dispondremos de él hasta dentro de dos semanas. Los pasos que seguiríamos serían:

1. En la tabla de **Artículos** buscaríamos el registro que contenga los datos de este libro.
2. Una vez en él, localizaríamos el código del cliente que nos lo ha solicitado (**Cod\_Cli**) .
3. Con este código de cliente, iríamos a la segunda tabla (tabla **Clientes**) y localizaríamos el registro correspondiente a este código.
4. Por último, buscaríamos el contenido del campo **Teléfono** en dicho registro.

De hecho, lo que hemos hecho ha sido relacionar las dos tablas mediante el único campo que tienen en común: el código de cliente (**Cod\_Cli**). Pues bien, el modo en que trabaja el ordenador es similar, pero necesita que le indiquemos cómo ha de realizar los pasos anteriores; esto es, necesita que le indiquemos qué relación existe entre las dos tablas utilizadas,

Así hemos utilizado nuestra primera base de datos relacional de dos tablas. Como ves, una base de datos relacional no es más que un conjunto de tablas relacionadas entre sí.

Aunque la solución dada a nuestro ejemplo ha reducido la cantidad de datos repetidos, en la tabla de almacén vemos que todavía se repiten los datos de los libros cada vez que se

vende uno. De este modo, los datos del Manual de Word (autor, ISBN, etc.) se encuentran tres veces en la misma tabla de la figura 1.3.

ISBN	Título	Autor	Cod_cli	Nom_Cli	Dirección	Fecha Venta
111111	Manual de ms-dos	José Luis Gamuza	0001	Carolina	Orillamar, 33	31/01/94
222222	Manual de Word	Edward Jones	0002	Jesús	Urzáiz, 55	14/02/94
333333	Manual de Excel	Venerando Solís	0004	Susana	Coya, 44	24/01/94
444444	Manual de Access	Alberto Rodríguez	0003	Alicia	Anónimo,123	31/01/94
111111	Manual de ms-dos	José Luis Gamuza	0004	Susana	Coya, 44	09/01/94
333333	Manual de Excel	Venerando Solís	0003	Alicia	Anónimo,123	23/01/94
222222	Manual de Word	Edward Jones	0001	Carolina	Orillamar, 33	31/01/94

Figura 1.3.

Como puedes imaginar, la solución vuelve a ser dividir esa tabla en otras dos relacionadas. Las dos figuras que mostramos a continuación pueden ser una solución a este problema.

ISBN	Título	Autor
------	--------	-------

Fig. 1.6 Tabla de artículos - ALMACEN

ISBN	Cod-Cli	Fecha_venta
------	---------	-------------

Fig, 1.7. Tabla de artículos vendidos - VENTAS

La tabla de la figura 1.6, que llamaremos **ALMACEN** es para almacenar información de los distintos artículos que están a la venta. Si introduces información te darás cuenta de que no tienes información repetida en ningún registro.

A su vez, la tabla de la figura 1.7, que llamaremos **VENTAS**, sólo cuenta con los campos que indican qué artículo ha comprado un cliente determinado.

La relación existente entre estas dos tablas viene dada por el campo **ISBN** que al ser único para cada libro nos sirve para no repetir los datos de la tabla **ALMACEN**. Observa bien las tres tablas que hemos obtenido al final (figuras 1.5, 1.6 y 1.7, Clientes, Almacén y ventas, respectivamente). Verás que no hay ningún dato que esté repetido en las distintas tablas, excepto los que se utilizan para relacionar unas tablas con otras.

La tabla **Clientes** está relacionada con la tabla **Ventas** mediante el campo **Cod\_Cli**. Por otra parte, la tabla **Almacén** también está relacionada con la tabla **Ventas** mediante el campo **ISBN**. Estas relaciones permiten combinar los datos de las tablas **Clientes** y **Almacén** utilizando la tabla **Ventas** como enlace.

Estas dos últimas tablas que hemos diseñado podrían – deberían – completarse con algunos campos más. Los campos que tienen hasta ahora proceden de la división de la tabla de la que partimos inicialmente. Pero si examinamos las tablas pensando en la función que deben cumplir, que es guardar la información necesaria para la gestión de las ventas de libros, deberemos completarlas.

Así, la tabla **ALMACEN** podría incluir los campos **PVP** y **Existencias**. El campo **PVP** servirá para facturar al cliente el importe correspondiente. El campo **Existencias** nos indicará si disponemos de un libro en el almacén, y debería actualizarse cada vez que compremos/vendamos ese libro.



ISBN	Título	Autor	PVP	Existencias
------	--------	-------	-----	-------------

Fig. 1.6 Tabla Almacen

Y la tabla **Ventas** *debería* incluir el campo **Num-Pedido** por varias razones:

- es práctica habitual que los pedidos lleven una numeración.
- podrá servirnos como **clave**, o identificador único del registro. Esto lo estudiaremos con detalle más adelante.

También deberemos incluir el campo **Cantidad**, para indicar las unidades de un mismo libro que un mismo cliente compra en una misma fecha.

ISBN	Cod-Cli	Num-Pedido	Cantidad	Fecha_venta
------	---------	------------	----------	-------------

Fig. 1.7 Tabla VENTAS

## 7.1 TIPO DE RELACIONES

La primera norma para establecer relaciones entre tablas es que el campo o columna que se utiliza para enlazar no debe tener repeticiones, sino que debe ser un identificador único: Número de la Seguridad Social, NIF, Código de Artículo, Numero de Expediente, etc. A este campo o columna que permite la identificación de cada registro se le denomina **clave primaria**. También puede haber claves **compuestas** cuando están formadas por varios campos: apellidos y nombre. Para establecer una relación es necesario que las dos tablas tengan una información común, es decir, un campo clave similar o idéntico que sirva de enlace. Si esta condición se cumple, pueden establecerse tres tipos de relaciones:

### • *Uno a uno (1:1)*

Son las relaciones que se establecen entre entidades que tienen una correspondencia uno a uno, es decir, una entidad A está asociada a lo sumo con una entidad B y una entidad B está asociada con una entidad en A.

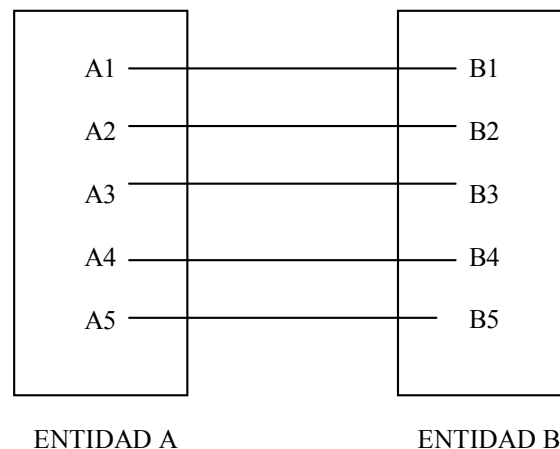


Fig. 1.8: Relación 1:1

Ej: Consideremos dos TABLAS, una llamada EMPLEADO y otro SUELDO. En EMPLEADOS se guarda la información personal de cada empleado de cada empresa y en SUELDO la información para hallar el sueldo de cada persona. Cada registro de empleado sólo va a estar relacionado con un registro de Sueldo.

Cod	nombre	direccion	telefono		Cod	sueldo-bru	horas-ext	retenc
123A	Pepa Pérez	C/Lista, 5	234566	→	123A	230000	0	19%
134A	Luis Sánchez	C/Sol 7	345355	→	134A	230000	20	19%
135A	Sara López	C/Perez 3	454545	→	135A	260000	12	10%
137A	Luis García	C/Soledad 7	437355	→	137A	200000	0	19%
139A	Sara López	C/Calvo, 3	454567	→	139A	230000	0	19%
140A	Angel Pua	C/Luna 7	464666	→	140A	280000	30	20%
142A	Ana Cal	C/Pisa 3	676767	→	142A	230000	0	19%

Figura 1.9: La tabla EMPLEADO mantiene una relación 1:1 con SUELDO

## • **Uno a muchos (1:N)**

Son las relaciones uno a muchos que se establecen entre una entidad A y una serie de ocurrencias de otra entidad B que se relacionan con la primera. Una entidad en A está asociada con un número cualquiera de entidades de B. Una entidad de en B, sin embargo, sólo está asociada con una entidad en A. Es el caso de los empleados en relación con un departamento: existen varios empleados en un mismo departamento pero no viceversa, pues no existen varios departamentos para un mismo empleado .

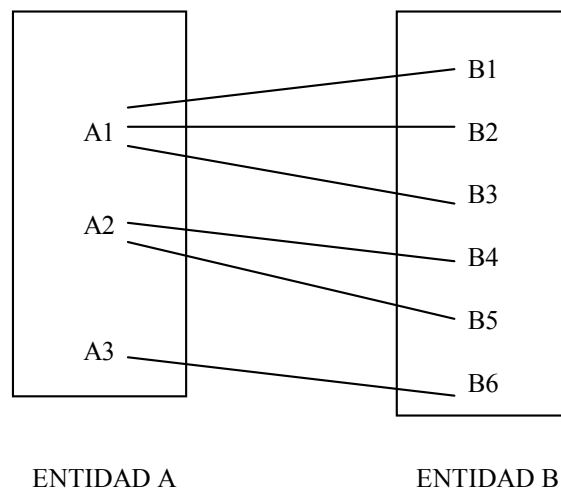


Fig. 1.10: Relación 1:N

Ej: Supongamos dos tablas llamadas PROVEEDOR y ARTICULO. Supongamos que en la empresa que queremos modelar, un proveedor puede proveer más de un artículo, pero un mismo artículo sólo lo suministra un proveedor. En este caso estamos en una relación uno a muchos.

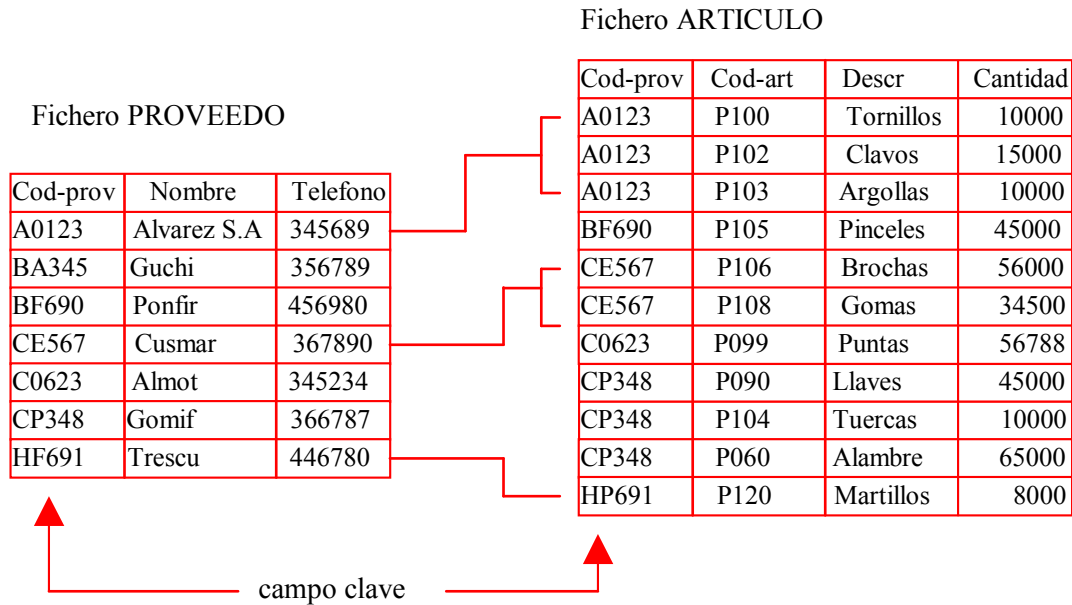
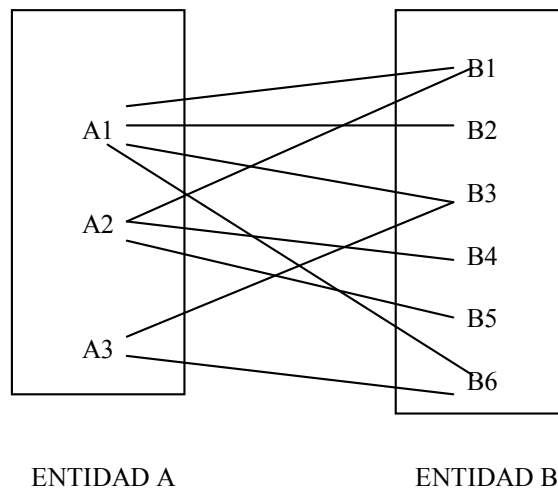


Fig. 1.11: Proveedores se relaciona con Artículo en una relación 1:N

### • Muchos a muchos (N:M)

Finalmente, existen las relaciones muchos a muchos, como puede ser el caso de los alumnos y asignaturas de un centro educativo, o los empleados de una empresa en relación con proyectos de esa empresa. Una entidad A esta asociada con un número cualquiera de entidades en B y una entidad B esta asociada con un número cualquiera de entidades en A.



Por ejemplo: un empleado puede ser asignado a más de un proyecto, y en un proyecto pueden participar más de un empleado. Es importante señalar que este tipo de relaciones muchos a muchos, tan habituales en el mundo real, no pueden trasladarse directamente a la implementación de los datos en un gestor de bases de datos. Las relaciones muchos a muchos se descompondrán en dos relaciones uno a muchos (1:N), en las que existe una nueva tabla que nos permite relacionar las dos tablas iniciales.

La tabla ASIGNACIÓN utiliza los campos **clave** de las tablas principales como **claves foráneas** para realizar la asignación de proyectos a empleados. Cada empleado pertenece a un unico departamento.

Codigo empleado	Nombre	Apellidos	Alta	Sueldo	Codigo departamento
P450	Juan	Sánchez López	01/12/96	2300000	DC1
P451	María	Rivero Mosteiro	12/02/97	3000000	DI1
P453	Rosa	Perez Roma	11/04/97	3500000	DA1
P455	Rafael	Moreno Peña	17/05/98	2500000	DI1
P456	Carlos	Roca Fria	19/04/98	3500000	DI1
P457	Manolo	Casas Vila	02/05/98	5000000	DP1

Figura 1.12 . La tabla EMPLEADO

CÓDIGO PROYECTO	NOMBRE PROYECTO
PRY0023	Desarrollo nuevo producto
PRY0024	informatización oficina
PRY0025	Campaña de lanzamiento
PRY0026	Formación

Figura 1.14 La tabla PROYECTO mantiene una relación N.N con EMPLEADO, aunque para ello necesita una tabla intermedia, la tabla ASIGNACION (véase figura 1.15).

CÓDIGO EMPLEADO	CÓDIGO PROYECTO
P450	PRY0024
P451	PRY0024
P451	PRY0025
P453	PRY0026

Figura 1.15. La tabla ASIGNACIÓN sirve para relacionar PROYECTO y EMPLEADO y, de esta forma, descomponer la relación muchos a muchos en dos relaciones uno a muchos.

## 8. PLANIFICACION DE LAS BASES DE DATOS

El trabajo de planificación es esencial para obtener un diseño adecuado de una base de datos, con tablas y relaciones que representen adecuadamente la organización real.

Para ello es necesario dividir y clasificar la información de forma en que se conviertan en unidades independientes y autónomas y establecer las relaciones. Esto, que hemos visto de modo intuitivo en los ejemplos precedentes, pueden realizarse de modo más formal:

- Empleando la metodología de diseño **Entidad/Relación**, con lo que obtendremos tablas normalizadas (que cumplen las reglas de normalización básicas) o bien
- Aplicando sobre nuestros datos iniciales, las reglas de *normalización*, que permiten comprobar que el diseño de tablas es correcto.