

Tema 7

GESTORES DE BASES DE DATOS

7.1 INTRODUCCIÓN

Un gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas. Existen distintos tipos de gestores de bases de datos: relacional, jerárquico, red, ... El modelo relacional es el utilizado por casi todos los gestores de bases de datos para PC's. El modelo relacional (SGBDR) es un software que almacena los datos en forma de **tablas** (ver figura 7.1).

Código	Título	Nacionalidad	Género	Duración
1	Contacto	USA	C/Ficción	104
2	El retorno del Jedi	USA	C/Ficción	120
3	La lista de Schindler	USA	Histórica	180
4	El mundo perdido	USA	C/Ficción	123
5	Nueve meses	USA	Comedia	95
6	Algo para recordar	USA	Comedia	100
7	Two Much	ESP	Comedia	90
8	Drácula de Bram Stocker	USA	Terror	110
9	Titanic	USA	Drama	135
10	El último mohicano	USA	Drama	95

Figura 7.1: Ejemplo de una tabla

7.1.1 El problema: Sistemas de ficheros

Tradicionalmente, los datos se han organizado en ficheros. Un fichero mantiene información homogénea, dispuesta en registros. Ej.: Empleados, Clientes, Nóminas, etc. Diferentes programas pueden mantener diferentes ficheros referidos a la misma entidad (ver figura 7.2).

Estos sistemas presentan algunos problemas:

Redundancia: Normalmente es perjudicial ya que da lugar a inconsistencia, cuando un dato no se actualiza en todos los lugares donde aparece. Es el caso de datos repetidos, que aparecen en varios ficheros, o de datos calculados, que podrían obtenerse a partir de otros datos.

Rigidez de búsqueda: A cada fichero, según la manera en que más frecuentemente se accede a él, se le da una organización. Si después se necesita otro tipo de acceso, puede resultar lento trabajar con el fichero.

Dependencia de los programas: La información de dónde comienza un campo, dónde acaba, su tipo, etc. está controlada por el programa; cualquier cambio en la estructura del fichero implicaría una modificación de los programas.

Problemas de confidencialidad y seguridad: La **confidencialidad** consiste en evitar la consulta de ciertos datos a determinados usuarios mientras que el control de **seguridad** de los datos almacenados impedirá que puedan ser modificados por personas no autorizadas.

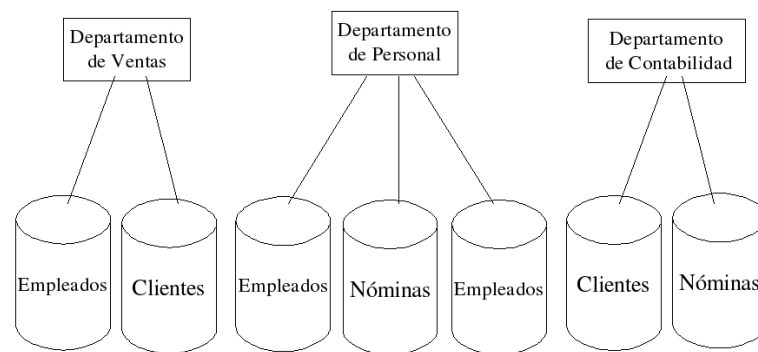


Figura 7.2: Sistema de ficheros

7.1.2 La solución: Bases de datos

Es la alternativa que aborda la solución a estos problemas. Se trata de dar una solución integral al almacenamiento y gestión de los datos, en lugar de soluciones parciales (ver figura 7.3):

- Evitar la redundancia "gratuita"
- Flexibilidad de búsqueda
- Independencia de los programas
- Seguridad y confidencialidad integral

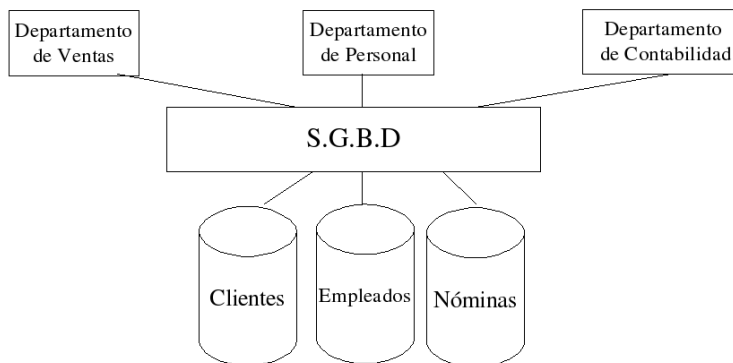


Figura 7.3: Sistema de Bases de datos

7.2 LOS USUARIOS

Hay tres clases de usuarios:

- Usuario final: Accede a la base de datos desde su PC empleando un lenguaje de consulta (DML) o a través de un programa.
 - Son usuarios que no necesitan formación técnica
 - Podrán manejar la información de forma sencilla y eficiente a través de la interfaz que se les proporcione.
- Administrador de la base de datos: Se encarga del control general del sistema de base de datos. Usualmente actúa como intermediario entre programador y usuario final.
 - Son los responsables de su seguridad e integridad
 - Requieren un amplio conocimiento de la herramienta SGBD a nivel de administración: tablas, índices, consultas, formularios, informes, macros, etc.
- Programador de aplicaciones: Encargado de escribir programas de aplicación que utilicen bases de datos (lenguaje de alto nivel, como Cobol, Clipper, VisualBasic, 4GL).
 - Pueden utilizar lenguajes de alto nivel para acceder y actualizar los datos.
 - Son capaces de implementar soluciones a medida.
 - Su conocimiento de la herramienta SGBD debe ser aún más profundo: módulos, API (application programa interface), etc.

7.3 CONCEPTOS DE BASES DE DATOS

En las bases de datos se manejan distintos conceptos que pasamos a comentar.

7.3.1 Entidades

Una entidad es una clase o categoría de objetos que poseen características diferenciadoras que los distinguen del resto. Ejemplo: Dentro de una empresa que vende complementos para el automóvil encontraremos las siguientes entidades: Artículos, Clientes, Proveedores, Pedidos, etc.

Otros ejemplos:

En una biblioteca: Libro, Socio, Autor, etc.

En una academia: Alumno, Profesor, Cursos, Asignaturas, etc.

En concesionario de automóviles: Vendedor, Cliente, Automóvil, Pedido, etc.

Las entidades consideradas en una base de datos deberán determinarse en consonancia con las necesidades. Por ejemplo, en una empresa de transportes aparecen diferentes entidades: vehículos, mercancías, transportistas, clientes, etc. No obstante, si nuestro objetivo fuere diseñar una base de datos para el control de las inspecciones técnicas de los vehículos, entonces el resto de entidades (mercancías, transportistas, clientes, etc.) no serán tenidas en cuenta.

Cada objeto perteneciente a una entidad debe poseer información suficiente para que pueda ser identificado de forma única.

En una base de datos relacional, las entidades se representan en forma de **tablas**.

7.3.2 Atributos

Toda entidad contiene un conjunto de datos, a los que llamaremos atributos o campos, que permiten describir de una manera completa y única a cada elemento de la entidad. Ejemplos:

Entidad "Clientes". Atributos: Código, DNI, Nombre y apellidos, Dirección, Teléfono, Cuenta bancaria, etc.

Entidad "Productos". Atributos: Código, Descripción, Fabricante, Color, Peso, Precio, etc.

Cada atributo se corresponde, en una base de datos relacional, con las **columnas o campos** de una tabla.

La figura 7.1 muestra un ejemplo de una tabla correspondiente a una entidad Películas. En ella se observan cinco atributos o campos (las columnas): Código, título, Nacionalidad, Género y Duración.

7.3.3 Registros

Para una entidad dada, cada entrada o aparición (cada cliente en la entidad Clientes, cada vehículo en la entidad Vehículos, etc.) se denomina registro u ocurrencia de registro. Un registro es, por tanto, una representación de un objeto perteneciente a una entidad dada.

En una base de datos relacional, los registros se corresponden con las **filas** de las tablas. Ejemplos:

La entidad Automovil con los campos N_matricula, Marca, Modelo, Color, Km, Gasolina, y un registro (o ocurrencia de registro): J-5757-M, Ford, Orion, Rojo, 45401, Super.

En una base de datos comercial, tenemos las entidades Cliente, Vendedor, Producto, etc. Para el registro Vendedor, habrá tantas ocurrencias de dicho registro como vendedores hay en la empresa.

En la figura 7.1 vemos un ejemplo de tabla (para una entidad "Películas") donde cada fila de la tabla corresponde a un registro u ocurrencia de registro.

7.3.4 Claves

Para una entidad dada, es necesario que cada ocurrencia esté descrita de manera única y diferenciada del resto de ocurrencias de esa misma entidad. Esto se consigue mediante la clave de entidad: un atributo o un conjunto de atributos de la propia entidad que identifica de manera única a cada ocurrencia de la entidad.

Ejemplo:

La entidad Cliente contiene dos atributos que perfectamente pueden identificar de manera única a cada ocurrencia: D.N.I. y Código de Cliente. Ambas son claves de entidad, puesto que no existe más de un cliente con un mismo D.N.I. o Código. Sin embargo pueden coexistir, varios con un mismo Nombre, igual Apellido, etc.

Si una clave no tiene ningún subconjunto de campos que sea a su vez clave, se dice que es una clave **candidata**. El diseñador escogerá entre las claves candidatas la más adecuada para tratarla como clave **principal o primaria**. Al resto de las claves candidatas se les llamará claves **alternativas**.

Ejemplo: Entidad "Alumno". Atributos: N_matricula, Nombre, Apellidos, DNI, Direccion, Telefono, F_nacimiento.

CLAVES

N_matricula

DNI

Nombre + Apellidos

Nombre + Apellidos + DNI

N_matricula + DNI

N_matricula + Nombre + Apellidos + DNI

CLAVES CANDIDATAS

N_matricula

DNI

Nombre + Apellidos

CLAVE PRIMARIA (O PRINCIPAL)

N_matricula

CLAVES ALTERNATIVAS

DNI

Nombre + Apellidos

Entidades débiles Cuando no hay ningún atributo (o conjunto de atributos) que identifique de manera única cada una de las ocurrencias de la entidad. Es

decir, son entidades en las que no se puede determinar ninguna clave.

Ejemplo: Los ejemplares de un mismo libro en una biblioteca. Si tenemos como atributos, Título, Autor, Editorial y Año de publicación, no podemos encontrar ningún atributo (o conjunto de atributos) que permita identificar de forma única cada ejemplar (consideremos que de un mismo libro habrá varios ejemplares), es decir, no podemos definir ninguna clave para esta entidad.

En tales casos, deberemos realizar alguna modificación para conseguir una entidad con clave. Por ejemplo, al añadir un código para cada ejemplar de libro, conseguimos una clave que permite identificar de forma única cada ejemplar, y por tanto dicha entidad deja de ser débil.

Claves foráneas A veces, en nuestra base de datos necesitamos incluir en una entidad atributos que son claves de otras entidades. Las denominamos claves foráneas. Un campo es clave foránea en una tabla cuando es clave principal en otra tabla (ver figura 7.4).

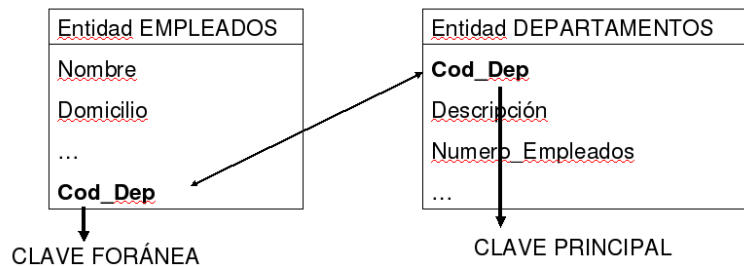


Figura 7.4: Clave foránea

7.3.5 Relaciones

Las entidades, habitualmente, no existen de forma aislada o independiente, sino que están relacionadas unas con otras. Por ejemplo:

- Una Editorial edita muchos Libros
- Un Autor escribe muchos Libros
- Un Pedido está formado por una relación Productos
- Un Profesor imparte clases a muchos Alumnos
- Un Automóvil tiene un Propietario

Para establecer una relación entre dos entidades, hacemos uso del concepto de clave foránea, visto anteriormente. Así, la tabla A se relaciona con la tabla B si la clave de la tabla A aparece como clave foránea en la tabla B (o al contrario). (Ver figura 7.4).

Grado de una relación: El grado de una relación indica la manera en que cada entidad participa en la relación. Hay tres tipos:

(1 : 1) Uno a uno	}	Relaciones simples
(1 : n) Uno a muchos		
(n : m) Muchos a muchos		Relaciones complejas

Relaciones de uno a uno (1:1) A cada ocurrencia de una entidad le corresponde una y solo una ocurrencia de la otra, y viceversa. Son relaciones simples.

Ejemplos:

Empleado-Cónyuge
Alumno-Expediente

Relaciones de uno a muchos (1:n) A cada ocurrencia de la primera entidad le pueden corresponder varias ocurrencias de la segunda, y a cada ocurrencia de la segunda le corresponde no más de una ocurrencia de la primera. Son relaciones simples.

Ejemplos:

Padre-Hijo
Departamento-Empleado

Relaciones de muchos a muchos (n:m) A cada ocurrencia de la primera entidad le pueden corresponder varias ocurrencias de la segunda, y viceversa. Un profesor da clase a muchos alumnos y un alumno tiene varios profesores, por tanto la relación Profesor-Alumno pertenece a este tipo de relación. Son relaciones complejas.

Ejemplos:

Países-Idiomas
Cliente-Productos

7.3.6 Representación gráfica

Para representar gráficamente las entidades y atributos usaremos los siguientes elementos (ver figura 7.5):

- Entidades: se representan mediante **rectángulos**
- Atributos: se representan mediante **elipses**
- Clave principal: El atributo (o atributos) considerados clave principal se encierran en **doble elipse**.
- Relaciones: se representan mediante un **rombo** unido a ambas entidades relacionadas.
- Grado de una relación: en cada extremo de una relación se indica el grado en que cada entidad participa en dicha relación. Usaremos el asterisco '*' para indicar 'muchos'.

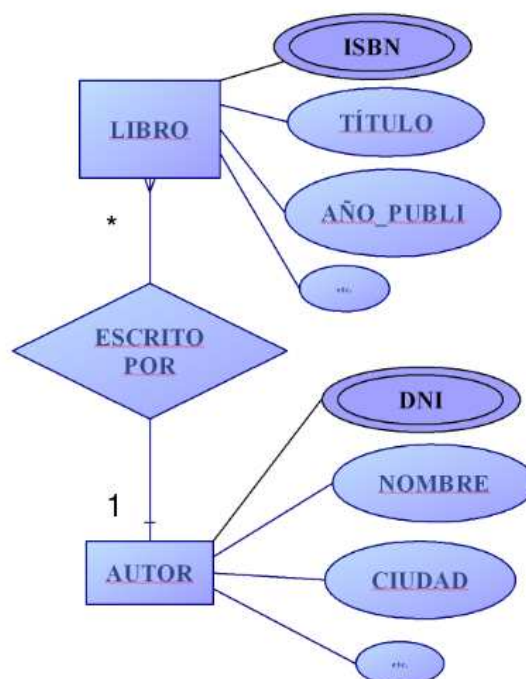


Figura 7.5: Representación gráfica

7.4 OBJETOS DE BASES DE DATOS

La mayor parte de las bases de datos que diseñemos harán uso de seis categorías de objetos: tablas, consultas, formularios, informes, macros y módulos. A continuación describimos dichas categorías:

7.4.1 TABLAS

Definimos tabla como un sistema o dispositivo de almacenamiento de los datos referentes a una **entidad**. Ejemplo: Tabla Representante para la entidad Representante, tabla Pieza ...

Las columnas se conocen con el nombre de campos o atributos y representan las distintas elementos de información disponibles en la tabla (entidad). Código, Nombre, Apellidos, Dirección, Teléfono... conformarían las columnas o campos de la tabla Representantes

Las filas (ocurrencias de registro o simplemente registros) se definen como el conjunto de atributos o campos correspondientes a un elemento de información.

Ejemplo. En la tabla Representante, cuyos campos son Código, Nombre, Apellidos, Dirección, Ciudad, Teléfono... , el siguiente conjunto de información podría constituir una fila o registro de la tabla: 2527, MANOLO, PÉREZ PEREZ, AVENIDA DEL OLIVO N° 8, JAÉN, 953222222..., constituirían una fila o registro dentro de nuestra tabla.

En la figura 7.6 vemos un ejemplo de una tabla correspondiente a películas de cine.

Código	Título	Nacionalidad	Género	Duración
1	Contacto	USA	C/Ficción	104
2	El retorno del Jedi	USA	C/Ficción	120
3	La lista de Schindler	USA	Histórica	180
4	El mundo perdido	USA	C/Ficción	123
5	Nueve meses	USA	Comedia	95
6	Algo para recordar	USA	Comedia	100
7	Two Much	ESP	Comedia	90
8	Drácula de Bram Stocker	USA	Terror	110
9	Titanic	USA	Drama	135
10	El último mohicano	USA	Drama	95

Figura 7.6: Ejemplo de una tabla

7.4.2 CONSULTAS

Cuando el usuario quiere obtener información de la base de datos, las tablas no son la mejor manera de hacerlo.

Ejemplo 1: una tabla de alumnos.

- A un profesor, le puede interesar una selección de alumnos de un determinado grupo dentro de un curso y de una titulación concreta.
- Al secretario del centro le puede interesar ver los alumnos que pertenecen a familia numerosa.

Ejemplo 2: Una librería, con una tabla de libros y otra de autores.

- Al usuario le puede interesar obtener una relación con las últimas ediciones, por ejemplo, los libros editados en 2006, indicando título del libro y nombre del autor.
- Si queremos hacer un pedido a una editorial, necesitamos una relación de los libros que dicha editorial publica, de los cuales no nos quedan existencias.

Es una de las partes más importantes dentro de una base de datos, ya que, además de servir de soporte a los datos, el cometido de un SGBD es proporcionar métodos de acceso a la información que resulten eficaces y apropiados.

Las consultas permiten personalizar o restringir el acceso a la información almacenada de acuerdo con una serie de criterios establecidos por el usuario. Estos criterios de selección son de dos tipos:

- Selección de campos: se trata de indicar qué campos han de incluirse en los resultados de la consulta, campos que pueden pertenecer a distintas tablas.
- Selección de registros: consiste en establecer ciertas condiciones de restricción que se aplicarán sobre ciertos campos (filtros) y que permitirán obtener, no todos los registros de la tabla sino una selección de ellos.

Ejemplo. Supongamos que, sobre la tabla de la figura 7.6, realizamos la siguiente consulta: Obtener Título, Nacionalidad y Género de las películas cuya

duración inferior a los 100 minutos. Analicemos la frase:

Título, Nacionalidad y Género ... constituye la selección de campos.
Duración inferior a 100 minutos ... constituye la selección de registros.

El resultado de la consulta será el mostrado en la figura 7.7.

Título	Nacionalidad	Género
Nueve meses	USA	Comedia
Two Much	ESP	Comedia
El último mohicano	USA	Drama

Figura 7.7: Resultado de la consulta

Como podemos observar, el resultado de una consulta se va a mostrar al usuario con el mismo aspecto de una tabla, pero no se trata de una tabla estática que permanece en la base de datos, sino una "tabla dinámica" que se genera en el momento en que se abre la consulta y se destruye una vez que el usuario cierra la consulta.

7.4.3 FORMULARIOS

Nos permiten manipular la información contenida en una base de datos mediante una interfaz gráfica similar a la de cualquier aplicación Windows. Mediante los formularios vemos y actualizamos la información disponible en la base de datos. Los formularios constituyen el instrumento que va a usar el usuario final de la aplicación para manipular la información (ver figura 7.8).

Los formularios pertenecen a la interfaz del usuario final y por tanto, en su diseño tendremos en cuenta una serie de características que afectan directamente al usuario final.

7.4.4 INFORMES

Se trata de presentar la información de la base de datos en un formato adecuado para su salida por impresora o cualquier dispositivo similar. Será el usuario quien elija los elementos de datos que se deben mostrar así como la forma en que han de quedar dispuestos en el informe (ver figura 7.9).

Habitualmente, los SGBD ofrecen utilidades para definir los informes de forma gráfica y sencilla.

7.4.5 MACROS

Conjunto de operaciones que nos permiten automatizar cualquier tarea repetitiva de una base de datos. Las macros se construyen encadenando una serie de acciones preconfiguradas que realizan operaciones de gestión tales como abrir o cerrar formularios e informes, ejecutar comandos de menú o examinar los registros de una tabla.

The image shows three overlapping windows from a database application. The top window is titled 'LECTORES' and contains fields for 'DNI', 'Nombre', and 'Telefono'. Below these fields is a record navigation bar showing 'Registro: 1 de 1'. The middle window is titled 'LIBRO' and contains fields for 'ISBN', 'TITULO', 'AÑO DE PUBLICACIÓN', and a large 'COMENTARIOS' text area. It also has a record navigation bar. The bottom window is titled 'PRESTAMOS' and contains fields for 'Numero', 'Fecha', 'DNI_Lector', 'ISBN_Libro', and 'Fecha devolución'. It also has a record navigation bar.

Figura 7.8: Ejemplo de formularios

LIBROS POR EDITORIAL

NOMBRE	TÍTULO	AUTOR	AÑO DE PUBLICACIÓN
Nombre	Titulo	Autor	Año de publicación
Quiso 2009	Wilson Pizarro	2009	
Nombre	Título	Autor	Año de publicación
Quiso 2014	Caro Pizarro	2014	
Nombre	Título	Autor	Año de publicación
Quiso 2014	Caro Pizarro	2014	
Nombre	Título	Autor	Año de publicación
Quiso	Wilson Pizarro	2009	
Nombre	Título	Autor	Año de publicación
Quiso de Tarea	Wilson Pizarro	2009	
Nombre	Título	Autor	Año de publicación
Quiso	Wilson Pizarro	2009	
Nombre	Título	Autor	Año de publicación
Quiso	Wilson Pizarro	2009	

Forma: 17 de abril de 2009. Página 1 de 1

Figura 7.9: Ejemplo de informe

7.4.6 MODULOS

Los módulos nos permiten automatizar tareas, al igual que las macros, pero con mayor flexibilidad y posibilidades. Un módulo es una colección de procedimientos creados mediante instrucciones en un lenguaje de programación (por ejemplo, para Microsoft Access, el lenguaje de programación de módulos es Visual Basic).

7.5 DISEÑO DE APLICACIONES DE BASES DE DATOS

Podemos diseñar y ofrecer al usuario una base de datos con todos sus objetos de estructura y operaciones: tablas, formularios, informes, etc., pero **esto no es una aplicación de bases de datos**. Una aplicación debe incorporar una **interfaz de usuario**.

La interfaz de usuario utilizará todos los objetos de la base de datos enlazándolos de forma adecuada y ofreciendo al usuario un sistema coherente de gestión de los datos. Para ello, gestores de bases de datos como Microsoft Access ofrecen ciertos recursos como son:

- Menús
- Barras de herramientas
- Macros

Etapas de diseño Las etapas que hemos de seguir para diseñar una aplicación de bases de datos son las siguientes:

1. Establecer propósito y objetivos
2. Definir la **estructura**: tablas, relaciones, etc.
3. Diseño de **operaciones**: entrada, salida, etc.
4. Interfaz de usuario: menús, macros, etc.

7.5.1 Establecer propósito y objetivos

Se trata de establecer los límites de la aplicación. No basta con decir, por ejemplo, "diseñar una base de datos para una papelería". En una papelería hay una gestión de ventas, una gestión de pedidos, una gestión de inventario, etc. Hemos de delimitar el ámbito de la aplicación estableciendo los objetivos que pretendemos alcanzar.

Por ejemplo, para una biblioteca, podríamos establecer los siguientes objetivos:

- Gestión de pedidos: facilitar la preparación y envío de pedidos a las editoriales, así como la recepción del material en respuesta a dichos pedidos.
- Gestión de préstamos: Control de libros prestados, pudiendo conocer en todo momento qué libros se han prestado, a quién y cuándo han de ser devueltos.

7.5.2 Definir la estructura

El diseño de la estructura de una base de datos no debe hacerse ?a la ligera?. Debemos seguir una metodología de trabajo que nos permita llegar desde la observación del mundo real hasta el diseño de la estructura de la base de datos, dicho de otro modo, debemos ajustarnos a un **modelo de datos**. El modelo de datos que utilizaremos (ver figura 7.10) consta de tres etapas:

- A. Elaboración del modelo conceptual
- B. Elaboración del esquema conceptual
- C. Construcción de tablas y relaciones en la base de datos

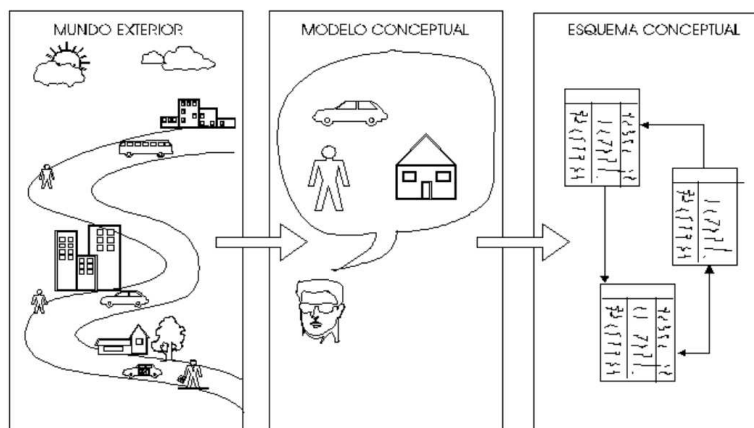


Figura 7.10: Modelo de datos

Comentaremos a continuación cada una de estas etapas.

A. Elaboración del modelo conceptual Para diseñar una base de datos, debemos acotar la parcela del mundo exterior que nos interesa. Debemos aprender, comprender y conceptualizar dicho mundo exterior transformándolo en un conjunto de ideas y definiciones que sean una imagen fiel del mundo real. Es un proceso de abstracción del mundo que nos rodea. A esta imagen del mundo exterior la llamaremos modelo conceptual.

De forma práctica, consiste en escribir las narrativas de los casos de uso de la aplicación, es decir, describir de forma exhaustiva los distintos usos que tendrá nuestra aplicación. La elaboración del modelo conceptual requiere disponer de amplio conocimiento del problema y de su contexto, y se han de tener muy en cuenta el propósito y los objetivos establecidos para la aplicación.

Siguiendo con el ejemplo de la biblioteca, los casos de uso de nuestra aplicación serán los siguientes:

Pedidos: Por fax o correo electrónico se enviará a la editorial un pedido de libros indicando: autor, título, año de edición, y número de ejemplares. Necesitaremos mantener la información postal de las editoriales, para agilizar el envío del pedido.

Recepción: A la llegada de un paquete remitido por la editorial se darán entrada en la base de datos de los libros recibidos. Se asignará un código individual a cada ejemplar.

Préstamo: Se identificará el lector así como el ejemplar de libro. Se indicará la fecha de préstamo y en la que debe ser devuelto.

Devolución: Con la identificación del ejemplar de libro se comprobará la fecha de devolución. La posible penalización se anota en la ficha del lector.

B. Elaboración del esquema conceptual Partiendo del modelo conceptual, vamos a elaborar un esquema conceptual, donde, de forma gráfica, queden expuestas las entidades consideradas, así como sus atributos y sus relaciones. Para ello deberemos seguir los siguientes pasos:

1 Entidades y atributos (Ver figura 7.11)

- Decidir qué van a ser entidades y qué van a ser atributos.
- Para descubrir entidades y atributos habremos de observar los sustantivos que aparecen en el texto.
- Como norma general podemos considerar que las entidades son todo lo que ocupe lugar en el espacio (p.e. un libro, una editorial) o en el tiempo (p.e. un pedido, un préstamo) y de lo que nos interese guardar información. Los sustantivos que no sean entidades serán, probablemente, atributos (título, ISBN, año publicación, etc.). Esta norma puede tener excepciones.
- Puede haber entidades y atributos no expresados de forma explícita en los escenarios del modelo conceptual.

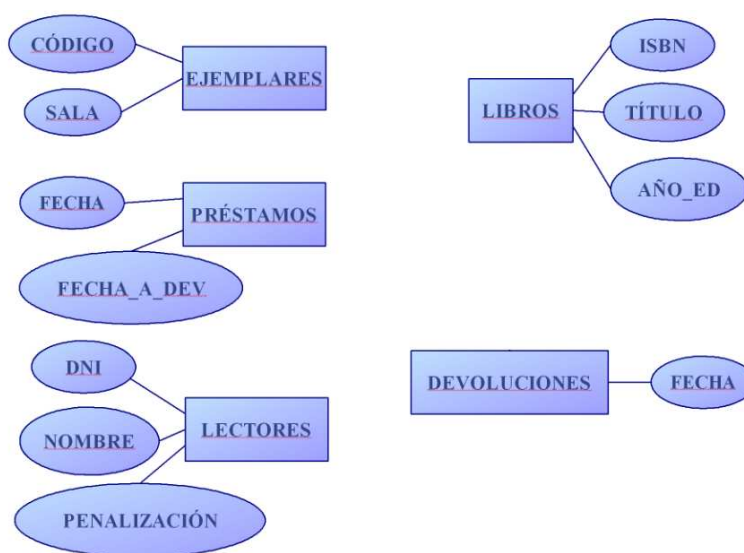


Figura 7.11: Esquema conceptual: Entidades y atributos

2 Claves e índices (Ver figura 7.12)

- Para cada entidad elegimos, entre las claves candidatas, la que va a ser clave principal.
- El campo clave será aquél por el que nos interese que esté ordenada la tabla (por el que se vaya a buscar habitualmente).
- El resto de las claves, opcionalmente podrán ser consideradas como índices, teniendo en cuenta las frecuencia con la que se vaya a buscar a través de dichas claves.

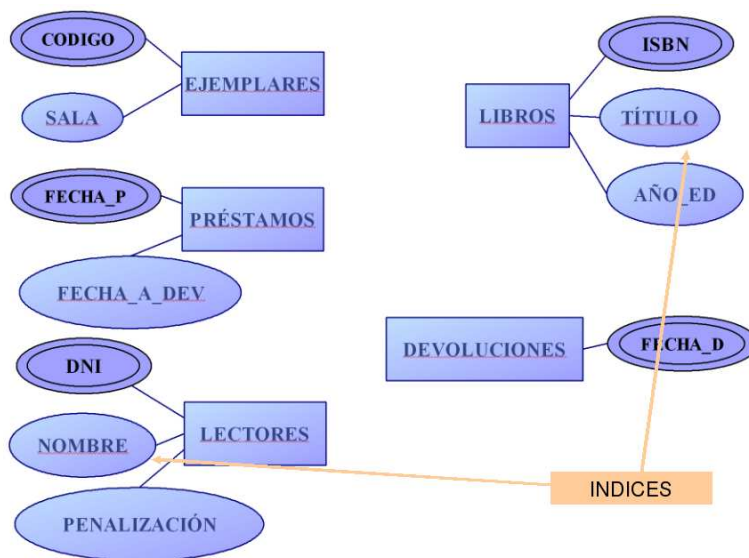


Figura 7.12: Esquema conceptual: Claves e índices

3 Relaciones (Ver figura 7.13)

- Establecemos sólo las relaciones que sean necesarias (el resto se omiten)
- Indicamos el tipo o grado de las relaciones marcando junto a cada entidad el cardinal con el que dicha entidad participa en la relación.
- Ejemplo: Autor-Libro (grado 1:n)
 - Junto a la entidad Autor marcamos un 1, ya que cada libro está escrito por 1 autor.
 - Junto a la entidad Libro marcamos una n (o un *), ya que cada autor puede escribir n libros.

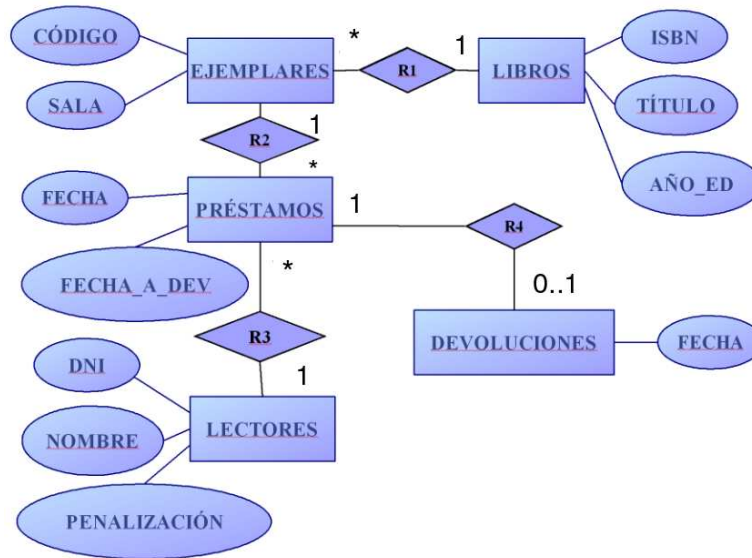


Figura 7.13: Esquema conceptual: Relaciones

C. Construcción de tablas y relaciones en la base de datos

- El último paso consiste en implementar el esquema conceptual en el gestor de bases de datos elegido (Microsoft Access, OpenOffice, etc.)
- Las entidades se implementan mediante tablas y los atributos son los campos o columnas de dichas tablas.
- Habitualmente dichos gestores proporcionan una ventana donde definir las relaciones entre entidades.
- Las relaciones simples (1:1 y 1:n) se implementan utilizando claves foráneas (ver figura 7.14)
- Las relaciones complejas (n:m) se implementan utilizando una tabla auxiliar (ver figura 7.15)

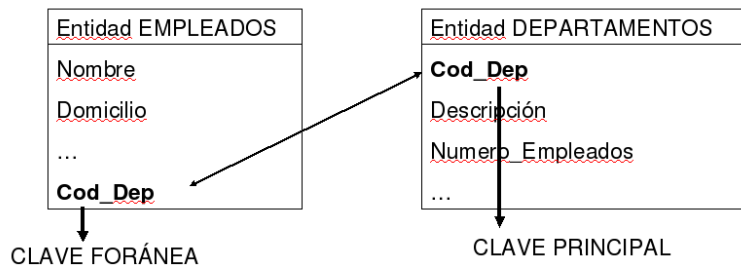


Figura 7.14: Relación 1:n entre Departamento y Empleado

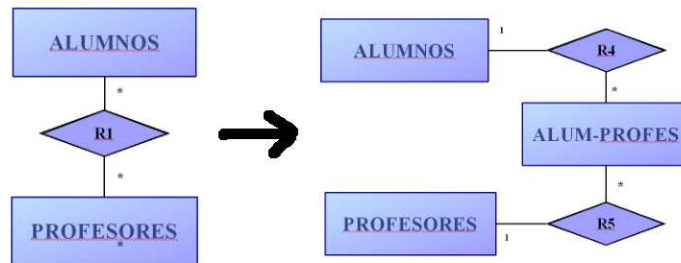


Figura 7.15: Simplificación de una relación compleja (n:m)

7.5.3 Diseño de operaciones

Una vez definida la estructura de la base de datos (parte **estática**), el siguiente paso es el diseño de las operaciones, es decir, la parte **dinámica** de la base de datos. Mediante estas operaciones accederemos a los datos para añadir, modificar, borrar, consultar, etc. Entre otras, podremos realizar las siguientes operaciones:

- Adición de registros.
- Modificaciones de registros
- Ordenación de la información
- Posibilidad de realizar determinadas operaciones matemáticas y estadísticas.
- Recuperación de registros
- Borrado de registros
- Emisión de informes

Para implementar estas operaciones haremos uso de los objetos dinámicos de la base de datos, principalmente consultas, formularios, informes y macros, comentados todos ellos en apartados anteriores.

7.5.4 Interfaz de usuario

Una base de datos no se convierte en una aplicación de bases de datos hasta que no dispone de una interfaz de usuario adaptada a su perfil, es decir, que ofrezca fácil acceso a toda la funcionalidad que la base de datos puede proporcionar (ver figura 7.16).

La interfaz de usuario utilizará todos los objetos de la base de datos enlazándolos de forma adecuada y ofreciendo al usuario un sistema coherente de gestión de los datos. Para ello nos apoyaremos en ciertos recursos de Access como son:

- Menús

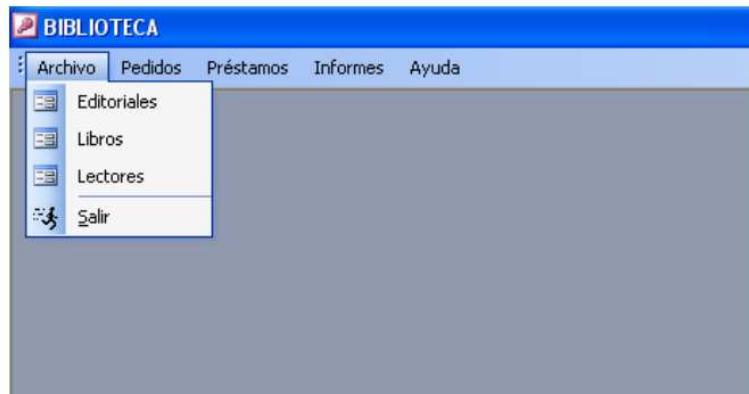


Figura 7.16: Ejemplo de interfaz de usuario

- Barras de herramientas
- Macros

Formulario de presentación Es habitual presentar un formulario previo de presentación de la aplicación con título de la aplicación, copyright, etc. No estará asociado a ninguna tabla ni consulta. Estará centrado en la pantalla y no tendrá botones de ningún tipo. Solo permanece visible unos segundos.

Sistema de menús Microsoft Access (al igual que otros SGBD) facilita la creación de menús desde los que acceder a los distintos objetos de la base de datos, principalmente, formularios e informes. Un sistema típico de menús puede tener la siguiente estructura:

- Primer nivel de menú
 - Archivo
 - ... (funciones de la aplicación)
 - Informes
 - Ayuda
- Segundo nivel de menú: Dentro de cada opción del primer menú dispondremos de un submenú (o menú de segundo nivel). Ejemplo:
 - Archivo: Editoriales, Libros, Lectores, ..., Salir
 - Pedidos: Emisión, Recepción
 - Etc.

7.6 INTEGRIDAD EN BASES DE DATOS

En una base de datos siempre va a ser una cuestión crítica el control de la integridad de los datos, es decir, controlar distintos aspectos como los siguientes:

- Impedir cualquier operación que provoque una inconsistencia de los datos (registros duplicados, registros "huérfanos" en una relación, etc.).
- En entornos multiusuario, que varios usuarios puedan coordinar el acceso a unos mismos datos en el mismo momento.
- Asegurar la integridad de dicha información cuando la base de datos se ha replicado en varios equipos.

Triggers. Los triggers o disparadores identifican las instrucciones que se ejecutan cuando se realiza una operación de inserción, actualización o supresión dentro de una base de datos. Con los triggers podemos, por ejemplo, dar un aviso justo antes de eliminar un registro, rellenar ciertos campos con valores por defecto al insertar un registro, etc.

Los "triggers" permiten asegurar la integridad de los datos ya que podemos controlar mejor las operaciones evitando la inconsistencia, incorporando controles de seguridad, etc.

Clases de integridad Podemos distinguir distintas clases de integridad dependiendo del aspecto de la base de datos que pretendamos controlar:

- Integridad de registro (Entity integrity), que consiste en evitar la duplicidad de registros (clave principal).
- Integridad de dominio (Domain integrity) que consiste en reducir los datos erróneos (tipos de datos, reglas de validación).
- Integridad referencial (Referential integrity): mantener la consistencia entre tablas relacionadas unas con otras.
- Integridad de usuario (User-defined integrity): seguridad contra acceso no deseado y acceso simultáneo a datos.
- Integridad de datos distribuidos (Distributed-data integrity): mantener la consistencia cuando una base de datos está distribuida (réplicas).

7.7 CÓMO EVALUAR UN SGBD

Los Sistemas de Gestión de Bases de Datos (SGBD) se han convertido en parte fundamental de la estrategia de las empresas. El valor de una información actualizada ha crecido tanto que las empresas que quieran incrementar o mantener su productividad deberán gestionar eficientemente todos los datos que manejan, y la mejor herramienta es un SGBD. Dado que disponemos de varias opciones, resulta imprescindible contar con elementos de juicio a la hora de optar por una u otra solución, ¿cuál se adecua mejor a nuestras necesidades?

7.7.1 Elementos de evaluación

El principal objetivo es encontrar el SGBD que sea capaz de responder adecuadamente al conjunto de aplicaciones y a las exigencias de los usuarios. Es decir, basta con saber qué queremos conseguir de un SGBD y buscar uno que destaque en eso que nosotros esperamos.

En primer lugar hemos de tener en cuenta varias características que definen un SGBD:

- Modelo de datos: relacional, jerárquico o en red.
- Lenguajes de definición y manipulación de datos (SQL)
- Herramientas de ayuda para el desarrollo: lenguajes de cuarta generación, generadores de aplicaciones, asistentes, generadores de informes, CASE, etc.

En segundo lugar hemos de evaluar el rendimiento del SGBD. Para ello hemos de definir un conjunto de pruebas mediante las cuales podamos valorar ciertas características muy concretas en cada producto. La evaluación se realiza midiendo el tiempo que tarda cada SGBD en llevar a cabo las pruebas, ya que la velocidad representa la eficiencia en la realización de cada función (ver figura 7.17).

- Capacidad de almacenamiento y recuperación de datos: velocidad de ejecución de consultas, creación de índices, importación de datos, etc.
- Protección de datos: acceso simultáneo de varios usuarios, etc.
- Control de accesos de los usuarios: intentos de acceso de usuarios no autorizados, etc.
- Consumo de recursos: memoria RAM, etc. (algunos SGBD incorporan monitores que permiten realizar un seguimiento de los recursos consumidos)

	<u>V. dBase</u>	<u>Paradox</u>	<u>CA-S.DB</u>	<u>Approach</u>	<u>Access</u>	<u>V. FoxPro</u>
ODBC	SI	SI	NO	SI	SI	SI
DDE y OLE	SI	SI	DDE	SI	SI	SI
Programación	SI	SI	NO	SI	SI	SI
Asistentes	SI	SI	NO	SI	SI	SI
Tamaño BD	8,8 MB	8,8 MB	9,2 MB	9,7 MB	5,9MB	4,1 MB
Consulta monotabla	2,18"	1,45"	19,40"	2,20	1,35"	1,50"
Consulta multitabla	2,30"	3,44"	NO	NR	4,23"	2,10"
Busqueda normal	2,27"	2,57"	9,43"	1,55	3,95"	2,56"
Búsq. campos index.	0,57"	1,70"	NO	0,45"	1,32"	1,25"
Consu. Actualización	2,27"	2,15"	NO	3,12"	3,24"	0,57"
Indexación de tabla	2,11"	3,55"	NO	4,2"	3,37"	1,50"
Precio	59.000	49.000	12.900	19.900	69.900	70.000

Figura 7.17: Ejemplo de evaluación de varios SGBD

7.7.2 SGBD libres y comerciales

A continuación mostramos una lista de los SGBD más comunmente usados, distinguiendo entre los SGBD libres y los comerciales.

Ejemplos de SGBD libres

- PostgreSQL
- MySQL

Ejemplos de SGBD comerciales

- Oracle
- DB2, Informix (IBM)
- dBase (dBI)
- Paradox (Borland)
- SQL-Server (MS)
- Access (MS)
- FoxPro (MS)

