Unidad de trabajo nº. 1:
Sistemas de almacenamiento de la información

1.1.- Introducción

Según el *Diccionario* de la Real Academia Española *(DRAE),* informática es el «Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores». El diccionario de Cambridge University Press define *information technology (IT)* como «la ciencia y la actividad de utilizar ordenadores y otras herramientas electrónicas para almacenar y enviar información». En ambos casos, el objeto de la disciplina es la información, y el objetivo su gestión.

Definimos *sistema de información* como el conjunto de procedimientos y funciones dirigidos a la recogida, elaboración, evaluación, almacenamiento, recuperación, condensación y distribu- ción de informaciones dentro de una organización.

Antes de que surgieran las bases de datos el procesamiento automatizado de información se hacía mediante ficheros. Las aplicaciones eran orientadas al proceso (el esfuerzo se enfocaba al tratamiento que los datos recibían en una aplicación concreta). Los ficheros se diseñaban a medida para cada sistema de información, sin que existiera un formato común.

Esta aproximación no contemplaba la gestión de la información a medio o largo plazo. Una organización disponía de varias aplicaciones que, en algunos casos, trataban la misma información (ejemplo: el *software* utilizado por el departamento de recursos humanos debía gestionar un fichero con datos de empleados, mientras la aplicación de contabilidad mantenía otro fichero distinto con los mismos datos organizados de otra forma). Surgían los siguientes problemas:

- Redundancia de datos (duplicidad innecesaria de información).
- Mal aprovechamiento del espacio de almacenamiento.
- Aumento en el tiempo de proceso.
- Inconsistencia de información debida a la redundancia (si un dato cambiaba en el fichero de una aplicación, no cambiaba en los demás).
- Aislamiento de la información (imposibilidad de transferirla a otros programas a no ser que se desarrollara un *software* de migración específico). Había, en definitiva, una gran falta de flexibilidad originada en la dependencia total de la estructura física de los datos.

1.2. Ficheros

Las aplicaciones gestoras de bases de datos se encargan de configurar una estructura óptima de almacenamiento de información con mínima intervención por parte del

usuario. No obstante, es interesante completar la perspectiva histórica con una breve descripción teórica sobre organización de ficheros.

1.2.1. Tipos de fichero según su estructura de almacenamiento

En relación con su contenido, encontramos los siguientes tipos básicos de fichero:

- *Texto plano*. Almacenan secuencias de caracteres correspondientes a una codificación de- terminada (ASCII, Unicode, EBCDIC, etc.). Son legibles mediante un *software* de edición de texto como el Bloc de Notas de Windows o el Vi de Linux.

  Ejemplos: los ficheros de texto con extensión .txt, los .csv de valores separados por comas, los .htm y .html correspondientes a páginas web, los de lenguajes de marcas .xml o .rss.

- *Binarios*. Contienen información codificada en binario para su procesamiento por parte de aplicaciones. Su contenido resulta ilegible en un editor de texto.

  Ejemplos: archivos ejecutables (.exe), documentos de aplicaciones (.pdf, .docx, .xlsx, .pptx), ficheros de imagen, audio o vídeo (.jpg, .gif, .mp3, .avi, .mkv), archivos de sistema (.dll).

PARA SABER MÁS

El sistema de codificación de caracteres más popular es el código ASCII (*American Standard Code for Information Interchange*, código estándar estadounidense para intercambio de información), que define 256 caracteres distintos (todas las combinaciones de 8 bits, es decir, 2^8 posibilidades). Al- gunos de ellos, llamados caracteres de control, no representan símbolos concretos, sino que se encargan de definir acciones como el borrado, el salto de línea o el tabulador.

Cuando se utilizan ficheros de texto plano para almacenar información se pueden clasificar de acuerdo a su organización interna:

- *Secuenciales*. La información se escribe en posiciones físicamente contiguas. Para acceder a un dato hay que recorrer todos los anteriores.

```
00789521T#Paula#Sanz#González#619554687$50687452Y#José      Luis#García#Viñals#
667859621$38546998X#Javier#Peinado#Martín#666932541$09653801B#Ruth#Lázaro#
Cardenal#689330247%
```

Figura 1.2. Fichero secuencial con información sobre clientes

Por cada contacto se ha decidido estructurar la información en cinco datos independientes: NIF, nombre, primer apellido, segundo apellido y número de teléfono. Nótese que en este caso el programador ha decidido utilizar la almohadilla (#) como separador de datos, el dólar ($) como separador de contactos y el tanto por ciento (%) como marca de fin de fichero.

• *De acceso directo o aleatorio*. Cada línea de contenido se organiza con unos tamaños fijos de dato. Se puede acceder directamente al principio de cada línea.

```
00789521TPaula          Sanz          González        619554687
50687452YJosé Luis      García        Viñals          667859621
38546998XJavier         Peinado       Martín          666932541
09653801BRuth           Lázaro        Cardenal        689330247
```

Figura 1.3. Fichero de acceso directo con información sobre clientes

En esta ocasión cada contacto ocupa una línea del fichero (al final de cada una el sistema operativo incluirá uno o dos caracteres de salto de línea invisibles para el usuario), y cada dato utiliza un número de caracteres fijo, aunque no lo ocupe totalmente (en el ejemplo se reservan 15 caracteres para el nombre, aunque en el caso de Ruth solo se utilicen 4).

Como todos los clientes ocupan el mismo espacio en el fichero, podemos acceder fácilmente a cualquiera de ellos multiplicando la posición en la que se encuentra menos una por el número de caracteres que mide cada línea. Por ejemplo, si el fichero se ha creado en un sistema Windows y queremos acceder al tercer cliente, tendremos que restar uno a su posición ($3 - 1 = 2$) y multiplicar el valor resultante por la longitud de la línea (63 caracteres más los dos caracteres de salto de línea que incluye el sistema operativo, es decir, 65). Como $2 \times 65 = 130$, la información del tercer cliente se encontrará en la posición 131.

La contrapartida a esta facilidad de posicionamiento es que el tamaño del fichero crece considerablemente respecto a su versión secuencial.

• *Indexados*. Generalmente en un fichero de acceso aleatorio la información se almacena en el orden en que se da de alta. Incluso aunque se consiguiera introducir dicha información de acuerdo a algún criterio de ordenación concreto, en algunas ocasiones es útil poder ordenarla por varios criterios distintos. En el ejemplo anterior es posible que necesitemos un listado de clientes ordenado por NIF y otro por apellido. Para dar solución a este problema se creó la organización indexada, que consiste en la existencia de uno o varios archivos adjuntos que ordenan el dato (llamado clave) por el que se desea ordenar el fichero y lo relacionan con la localización de la línea correspondiente.

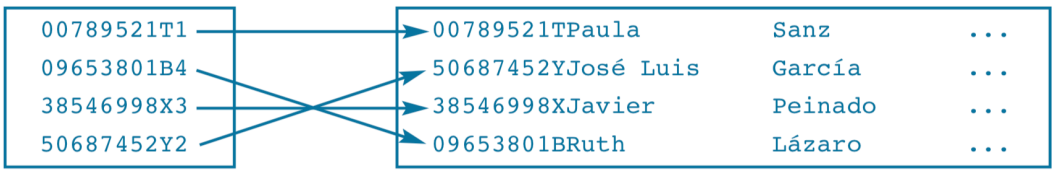

Figura 1.4. Fichero de índice por NIF de cliente y fichero de clientes original

En la figura 1.4 los NIF aparecen ordenados. Tras cada uno de ellos se ha añadido el número de línea del fichero principal donde se encuentra la información asociada. Si una aplicación *software* quisiera listar los clientes ordenados por NIF, recorrería

secuencialmente el fichero de índice, y al final de cada línea encontraría la línea del fichero principal que debe leer para encontrar a cada cliente.

El siguiente fichero indexaría los clientes ordenados por su primer apellido:

Figura 1.5 Fichero de índice por primer apellido de cliente y fichero de clientes original

Aunque se utilice en este caso para simplificar el ejemplo, generalmente el acceso a cada posición no lo marca el número de línea, sino un puntero a la celda de memoria correspondiente.

Actividad propuesta 1.1

Crear manualmente un fichero índice que ordene el ejemplo de la figura 1.3 por primer apellido, segundo apellido y nombre.

1.2.2. Tipos de soporte de almacenamiento

De acuerdo a la organización física de los datos, diferenciamos entre dos tipos de soportes:

• *Secuenciales*. Para acceder a un dato hay que recorrer todo el contenido del soporte previo a dicho dato (ejemplo: cintas magnéticas).

Figura 1.6

Cinta magnética para guardar información de respaldo

Direccionables. Se puede acceder directamente a un dato sin tener que recorrer todos los

anteriores (ejemplo: disco duro).

En un soporte direccionable se puede implementar un acceso secuencial, directo o indexado, mientras que en un soporte secuencial solo se podrá implementar un acceso secuencial.

1.3. Bases de datos

La evolución lógica de los problemas derivados del uso de ficheros fue estandarizar el acceso a la información, de modo que un diseño físico concreto sirviera para todas las aplicaciones de una organización. Este nuevo enfoque se centraba en los datos y no en el proceso, es decir, se estructuraba el almacenamiento de dichos datos con independencia de las aplicaciones que los fueran a utilizar. Se eliminaba la redundancia y se favorecía la transferencia de información entre aplicaciones. Aparecía el concepto de base de datos.

1.3.1. Definición

Volviendo nuevamente al *DRAE,* este nos dice que *base de datos* es un término relacionado con el mundo de la informática, y lo define como «Conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información». Adoración de Miguel y Mario Piattini ofrecen una definición más precisa:

> Colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos y su definición (estructura de la BD), única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos.

1.3.2. Tipos de bases de datos

Un *modelo de base de datos* es la arquitectura mediante la que se almacena e interrelaciona la información que se va a gestionar. La clasificación habitual de bases de datos toma como punto de partida el modelo subyacente:

- *Jerárquico*. Es el más antiguo. Refina la idea de fichero indexado, creando una estricta relación de jerarquía entre los datos de varios ficheros, motivo por el que presenta serias limitaciones semánticas. Relacionado con grandes máquinas (*mainframes*), su implantación comercial más conocida es IMS de IBM.
- *En red*. Introduce mejoras respecto al modelo jerárquico (mayor independencia y flexibilidad de los datos) a costa de aumentar el nivel de complejidad. Implantaciones: CODASYL, IDMS/DB de Computer Associates (actualmente CA Technologies).
- *Relacional*. Representa la información en forma de entidades y relaciones entre ellas, evitando rutas preconcebidas para localizar los datos y huyendo de la rigidez de los modelos previos. Cada entidad y cada relación aparece en forma de tablas bidimensionales (con filas y columnas). Es el modelo más extendido desde hace décadas, gracias a compañías como Oracle, IBM o Microsoft (que posteriormente evolucionaron hacia el modelo objeto-relacional), aunque hoy en día podemos encontrar bases de datos relacionales puras, como MySQL o SAP Sybase.
- *Orientado a objetos*. Aplica a los datos el paradigma de la orientación a objetos (OOP, *object-oriented programming*). Irrumpió con fuerza en los años noventa debido a las nuevas necesidades de almacenamiento de las bases de datos relacionales (imágenes, documentos, ficheros de audio y vídeo). Implantaciones:Versant, db4o, InterSystems, Objectivity.
- *Objeto-relacional*. En los últimos años los fabricantes de bases de datos relacionales han incorporado a su *software* diversas capacidades de las bases de

datos orientadas a objetos, creando modelos híbridos con base relacional. Ejemplos: Oracle, Microsoft SQL Server, IBM DB2, IBM Informix, PostgreSQL.

- Otros modelos.

  – *Documental*. Destinado al almacenamiento e indexación de grandes documentos.

  – *Orientado al documento*. Gestionan datos provenientes de documentos previamente estructurados, generalmente de lenguajes de marcas (XML, JSON).

  – *Multidimensional*. Orientado al tratamiento de la información mediante algoritmos de inteligencia artificial.

  – *Deductivo*.Almacena reglas de inferencia mediante las que genera deducciones a partir de unos datos determinados.

Actividad propuesta 1.2

Con la ayuda de Internet, elaborar una lista de bases de datos comerciales y libres relacionando cada una de ellas con su modelo de datos correspondiente.

Si, en cambio, utilizamos como criterio la ubicación física de la información, podemos diferenciar entre dos grandes tipos de bases de datos:

- *Centralizadas*. La base de datos reside en una sola máquina, típicamente el servidor de base de datos.
- *Distribuidas*. La información se reparte por distintos servidores, generalmente alejados físicamente. Un ejemplo sería la base de datos de una compañía de seguros, concebida a partir de los datos de la oficina central y de los de todas sus sucursales. Su implantación exige hacer un fuerte hincapié en aspectos de *networking* y seguridad.

PARA SABER MÁS

Símbolo de base de datos

En gráficos, esquemas y literatura informática las bases de datos se representan mediante un cilindro apoyado en su base. Es una referencia a las antiguas memorias de tambor magnético, de similar aspecto físico, como se ve en el siguiente anuncio publicado en la revista *Scientific American* en 1953:

Figura 1.9

Memoria de tambor

1.4. Sistemas gestores de bases de datos

El sistema gestor de bases de datos (SGBD) es el *software* que el fabricante pone a disposición del usuario para manejar sus bases de datos. Nuevamente, De Miguel y Piattini (1993) nos definen el término con más detalle:

Un conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios no informáticos como a los analistas, programadores, o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su seguridad.

En el mercado hay una amplia tipología de SGBD que corresponde con el modelo de base de datos subyacente.

1.4.1. Componentes del SGBD

Generalizando, podemos encontrar la siguiente enumeración de componentes en la mayoría de los SGBD:

- *Datos.*Almacenados de forma eficiente en ficheros del sistema operativo.
- *Herramientas de acceso a los datos.* Un lenguaje de programación mediante el que los usuarios técnicos puedan crear, leer y modificar la información, así como un diccionario de datos que albergue los metadatos, es decir, la información sobre el diseño de cada base de datos. Como mínimo, se ofrecerá una interfaz de línea de comandos mediante la que acceder a estas herramientas.
- *Utilidades.* Herramientas adicionales para gestión de *backups*, estadísticas, tareas programadas, mantenimiento de usuarios, grupos y permisos, etc.
- *Entornos gráficos.* Simplifican la gestión del SGBD y sirven como alternativa a la línea de comandos.

1.4.2. Funciones del SGBD

A pesar de la gran variedad de modelos y soluciones comerciales, podemos enumerar una serie de funciones comunes a un gran número de SGBD:

- Recuperar y modificar la información de los ficheros que conforman la base de datos de forma transparente para el usuario.
- Garantizar la integridad de los datos, impidiendo inconsistencias semánticas.

- Ofrecer un lenguaje de programación mediante el que interaccionar con la información.
- Proveer el diccionario de datos.
- Solucionar los conflictos derivados de accesos concurrentes a la información.
- Gestionar transacciones, garantizando la unidad de varias instrucciones de escritura relacionadas entre sí.
- Incluir utilidades de *backup*.
- Proporcionar mecanismos de seguridad para evitar accesos y operaciones indebidos.

Estos aspectos se desarrollarán en las unidades de trabajo siguientes.

PARA SABER MÁS

Según la consultora estadounidense Gartner, en el año 2013 la empresa Oracle Corporation contaba con la mayor cuota de mercado relativa a sis- temas gestores de bases de datos con sus SGBD Oracle Database y MySQL, consolidando la tendencia de los últimos años. Tras ella se encon- traban Microsoft (con SQL Server), IBM (con Informix y DB2) y SAP (con Sybase Adaptive Server Enterprise y Sybase IQ).

El *ranking* mensual de la página web www.db-engines.com estableció el siguiente índice de popularidad para gestores relacionales y objeto-relacionales en septiembre de 2019:

CUADRO 1.1. SGBD(O)R ordenados por índice de popularidad

352 systems in ranking, September 2019

Rank			DBMS	Database Model	Score		
Sep 2019	Aug 2019	Sep 2018			Sep 2019	Aug 2019	Sep 2018
1.	1.	1.	Oracle	Relational, Multi-model	1346.66	+7.18	+37.54
2.	2.	2.	MySQL	Relational, Multi-model	1279.07	+25.39	+98.60
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1085.06	-8.12	+33.78
4.	4.	4.	PostgreSQL	Relational, Multi-model	482.25	+0.91	+75.82
5.	5.	5.	MongoDB	Document	410.06	+5.50	+51.27
6.	6.	6.	IBM Db2	Relational, Multi-model	171.56	-1.39	-9.50
7.	7.	7.	Elasticsearch	Search engine, Multi-model	149.27	+0.19	+6.67
8.	8.	8.	Redis	Key-value, Multi-model	141.90	-2.18	+0.96
9.	9.	9.	Microsoft Access	Relational	132.71	-2.63	-0.69
10.	10.	10.	Cassandra	Wide column	123.40	-1.81	+3.85

Fuente: www.db-engines.com