# UD01 - EL ALMACENAMIENTO DE LA INFORMACIÓN

## Bases de Datos

# 1 Almacenamiento de la información

Todas las aplicaciones informáticas trabajan en última instancia con datos o información que deben ser almacenados en un medio físico, como discos duros, memorias flash o DVD. Estos medios forman una jerarquía que distingue entre tres niveles de almacenamiento: primario, secundario e intermedio.

## 1.1 Almacenamiento primario

Se refiere a aquellos medios sobre los que la CPU del ordenador puede acceder directamente y, por tanto, más rápidamente. Son la memoria principal o memoria RAM y las memorias caché de primer y segundo nivel, más pequeñas, pero más rápidas.

## 1.2 Almacenamiento secundario

Se refiere a dispositivos más lentos, pero de mayor capacidad, como los discos ópticos y magnéticos, o las memorias flash. Para acceder a los datos de la CPU debe copiarlos previamente a almacenamiento primario.

El almacenamiento secundario de más amplio uso es el disco, ya sea magnético o en estado sólido, aunque las cintas se usan sobre todo para copias de seguridad por su estabilidad, capacidad y durabilidad.

## 1.3 Almacenamiento intermedio

Cuando se necesita transferir información, es decir, bloques de disco a memoria principal, se suele realizar mediante la reserva de buffers dentro de la memoria principal para agilizar la transferencia. De este modo, mientras la CPU proceso datos de un buffer puede leer o escribir en otro. El uso de este tipo de almacenamiento es muy común en sistemas de bases de datos.

# 2 Sistemas de archivos

A continuación, se muestran los conceptos relacionados con archivos, tanto en lo relativo a su contenido como a su organización lógica y física.

**Registros**

La información se almacena en forma de registros, que son colecciones de valores o elementos de información relacionados, cada uno de los cuales corresponde a un campo del registro. Por ejemplo, un registro de un alumno incluiría campos como el nombre, fecha de nacimiento o teléfono. A su vez, cada campo tiene un tipo de datos que especifica el tipo de valores que puede tomar. La unión de estos campos y sus tipos determina el tipo o formato del registro.

**Archivos**

Se puede definir un fichero como un conjunto de registros, grabados sobre un soporte que pueda ser leído por el ordenador. Los ficheros son importantes porque son la unidad básica de información utilizada por cualquier programa, incluso los sistemas gestores de bases de datos. Todos los datos son, en última instancia gestionados mediante ficheros por cuatro

operaciones básicas: consultas o lectura, inserción, modificación y borrado. Cualquier otra operación más compleja es una combinación de las anteriores.

## 2.1 Organización de archivos

El término organización de ficheros se aplica a la forma en que se colocan los datos contenidos en los registros de cada fichero sobre el soporte informático (disco, cinta, etc.) durante su grabación.

Existen dos formas básicas de organización de ficheros, secuencial y relativa. En la organización secuencial los registros se van grabando unos a continuación de los otros, mientras que en la organización relativa los registros se graban en las posiciones que les corresponda según el valor que guarden en el campo denominado clave, que permite identificar el registro dentro del fichero.

**Organización secuencial**

Es el tipo más básico de organización. Los registros se colocan secuencialmente uno a continuación del otro y los registros nuevos se añaden al final del fichero.

La inserción es muy eficiente, ya que siempre se realiza en el último bloque disponible. Sin embargo, la búsqueda de datos resulta más complicada ya que requiere una búsqueda lineal hasta llegar al registro buscado. En cuanto al borrado es muy ineficiente ya que deja huecos en el fichero al no reutilizar ese espacio. La modificación es igualmente complicada al tener que buscar el registro y realizar su reescritura.

Con el fin de mejorar las prestaciones de la organización secuencial surgen una serie de organizaciones que son una variante de ésta:

- Organización secuencial indexada: Los registros se graban en un fichero secuencialmente, pero se pueden recuperar de manera directa gracias a la utilización de un fichero adicional, llamado índice, que contiene la información de la posición que ocupa cada registro.
- Organización secuencial encadenada: Permite ordenar los registros con un orden lógico distinto del físico gracias a la utilización de campos adicionales llamados punteros.

**Organización relativa**

En este tipo de organización los registros se graban en orden según el valor de uno de sus campos llamado campo de ordenación. Normalmente se usa un campo especialmente denominado campo clave, cuyos valores son distintos para cada registro.

En estos archivos la lectura es muy eficiente cuando se hace por el campo de ordenación o campo clave. La búsqueda es igualmente eficiente al poder usar la búsqueda binaria para encontrar un registro concreto.

La inserción es costosa, ya que debe mantenerse el orden, lo que puede implicar desplazamiento de registros para insertar uno nuevo. La modificación no ofrece problemas si el campo clave no cambia, si por el

contrario, este campo se actualiza deberá tratarse como una nueva inserción.

# 3 Sistemas de Bases de Datos

Inicialmente, cuando las primeras empresas y organizaciones empezaron a usar sistemas informáticos trabajaban con sistemas de ficheros. Es decir, cada equipo trabajaba con sus propios datos y programas y se encargaba de su mantenimiento y gestión. Al principio el sistema funcionó pero con el tiempo y, sobre todo, con el incremento de la cantidad de información así como de los usuarios que la manejaban surgieron problemas, que finalmente llevaron a la organización de la información mediante un sistema más ordenado y manejable basado en la centralización de la gestión y la organización de los datos en forma de bases de datos.

Los principales problemas relacionados con el uso de ficheros como sistemas de almacenamiento de información son los siguientes:

- Separación y aislamiento de los datos: Cuando los datos se separan en distintos ficheros es más complicado acceder a ellos, ya que el programador de aplicaciones debe sincronizar el procesamiento de los distintos ficheros implicados para asegurar que se extraen los datos correctos.
- Duplicación de datos: La redundancia de datos existente en los sistemas de ficheros hace que se desperdicie espacio de almacenamiento y, lo que es más importante, puede llevar a que se pierda la consistencia de los datos.
- Dependencia de datos: Ya que la estructura física de los datos se encuentra codificada en los programas de aplicación, cualquier cambio en dicha estructura es difícil de realizar.
- Formatos de ficheros incompatibles: Debido a que la estructura de los ficheros se define en los programas de aplicación, es completamente dependiente del lenguaje de programación.
- Consultas fijas y proliferación de programas de aplicación: Desde el punto de vista de los usuarios, la aparición fue un gran avance con respecto a los sistemas manuales. Como consecuencia creció la necesidad de realizar distinto tipo de consultas. Sin embargo, los sistemas de ficheros son totalmente dependientes del programador, es decir, para cada nueva consulta era necesario un nuevo desarrollo.
- Control de concurrencia: El acceso de varios clientes al mismo fichero genera inconsistencias, ya que un cliente puede consultar un fichero mientras otro lo está modificando.
- Autorizaciones: Los errores en los permisos de ficheros pueden hacer que un mismo cliente pueda modificar un dato en un fichero y no en otro en el que esté repetido.
- Catálogo: Resulta complicado saber dónde están los distintos datos. No existe un esquema general que muestre la organización de la información.

**Definición de Bases de Datos**

Una base de datos es un conjunto de datos almacenados entre los que existen relaciones lógica y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

## 3.1 Arquitectura de sistemas de bases de datos

En 1975, el comité ANSI-SPARC (American National Standard Institute – Standards Planning and Requirements Committee) propuso un estándar para la creación de bases de datos basado en una arquitectura de tres niveles, que resulta muy útil a la hora de conseguir tres características:

- Nivel interno: Describe la estructura física de la base de datos mediante un esquema interno. Este esquema describe todos los detalles para el almacenamiento de la base de datos, así como su acceso. Se habla de ficheros, discos, directorios, etc.
- Nivel global o conceptual: Describe la estructura de toda la base de datos para una comunidad de usuarios mediante un esquema conceptual. Este esquema oculta detalles de las estructuras de almacenamiento y se concentra en describir entidades, atributos, relaciones y restricciones.
- Nivel externo: Describe varios esquemas externos o vistas, donde cada uno de ellos es una porción de la base de datos que interesa a un grupo de usuarios determinados y oculta el resto de la base de datos

La arquitectura de tres niveles es útil para explicar el concepto de independencia de datos, que podemos definir como la capacidad para modificar el esquema en un nivel del sistema sin tener que modificar el esquema del nivel inmediato superior. La independencia se puede definir en dos tipos:

- Independencia lógica: Capacidad para modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación.
- Independencia física: Capacidad para modificar el esquema interno sin tener que alterar el esquema conceptual o los externos.

## 3.2 Modelos de datos

Las bases de datos consisten entonces en los datos concretos referentes a un sistemas o parte del mundo que hemos modelado. Estos datos son sencillos de manejar cuando son unos pocos, pero cuando su volumen crece se requiere el uso de distintos modelos para facilitar el diseño de las mismas.

**Definición de modelo de datos**

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones que existen entre ellos y sus restricciones.

Un modelo de datos proporciona mecanismos de abstracción para representar una parte del mundo cuyos datos nos interesan. Dicha representación, realizada en términos de un modelo dado, recibe el nombre de esquema y el conjunto de datos que representa la base de datos.

### 3.2.1 Tipos de modelos de datos

**Modelos conceptuales**

Se usan para describir datos en el nivel global. Con este modelo representamos los datos de forma parecida a como nosotros los captamos en el mundo real. Este tipo de modelos tienen una capacidad de estructuración bastante flexible y permiten especificar restricciones de datos explícitamente. Existen diferentes modelos de este tipo, pero el más utilizado por su sencillez y eficiencia es el modelo Entidad-Relación.

**Modelos lógicos**

Se utilizan para describir datos a nivel global, pero de un modo más lógico, es decir, más cercano a la máquina. Estos modelos utilizan tablas de registros para representar los objetos modelados y sus relaciones. A diferencia de los modelos de datos conceptuales, se usan para especificar la estructura lógica global de las bases de datos y para proporcionar una descripción estructurada y cercana a la implementación.

- Modelo relacional: En este modelo se representan los datos y sus relaciones entre estos, a través de una colección de tablas, en las cuales sus filas (tuplas) equivalen a cada uno de los registros que contendrá la base de datos, y las columnas corresponden a los atributos de cada registro. Actualmente es el modelo lógico más extendido.
- Modelo jerárquico: En este modelo la información se almacena de forma jerárquica. Fue el primer modelo usado y es especialmente útil cuando la aplicación maneja estructuras de árbol y son estructuras estables. Actualmente está en desuso dado su ámbito de aplicación.
- Modelo en red: Se trata de una variante del modelo jerárquico donde se permite que un nodo tenga varios padres. Al igual que el anterior tiene un ámbito de aplicación reducido y por lo tanto está en desuso actualmente.

**Modelos lógicos avanzados**

Se trata de modelos relativamente recientes, y cada vez más utilizados, sobre todo en aplicaciones específicas que manejan nuevos y más complejos tipos de datos.

- Modelos de datos orientados a objetos: Utilizados especialmente en aplicaciones programadas bajo paradigma de orientación a objetos. Trata de almacenar en la base de datos no solo los datos, sino también la funcionalidad asociada.
- Modelos de datos declarativos: Suelen usarse en bases de conocimiento, que no son más que bases de datos con mecanismos de consulta en los que el trabajo de extracción de información a partir de los datos recae en realizada sobre el sistema informático, en lugar de sobre el usuario.

# 4 Sistemas Gestores de Bases de Datos

**Definición**

El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos y proporciona acceso controlado a la misma. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos.

**Objetivos**

- Asegurar los tres niveles de abstracción: físico, lógico y externo.
- Permitir la independencia física y lógica de los datos.
- Garantizar la consistencia de los datos, ya que puede haber datos duplicados o derivados que deben mantener sus valores de forma coherente.
- Ofrecer seguridad de acceso a los datos por parte de usuarios y grupos.
- Gestión de transacciones de forma que se garantice la ejecución de un conjunto de operaciones críticas como una sola operación.
- Permitir la concurrencia de usuarios sobre los mismos datos mediante bloqueos que mantienen la integridad de los mismos.

## 4.1 Funciones del sistemas gestor de bases de datos

Para la consecución de los objetivos anteriores la mayoría de los SGBD comerciales y libres incorporan las siguientes características y funciones:

- Catálogo: Donde se almacenan las descripciones de los datos y sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe la base de datos.
- Garantizar la integridad: Dispone de mecanismos que garantizan que todas las actualizaciones correspondientes a una determinada transacción se realicen o no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos.
- Permitir actualizaciones: Asegurar que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente.
- Recuperación de datos: Permitir recuperar las bases de datos en caso de que ocurra algún suceso imprevisto que afecte o destruya la base de datos.
- Integración: Ser capaz de integrarse con algún software de comunicación.
- Cumplir restricciones: Proporcionar los medios necesarios para garantizar que, tanto los datos de la base de datos como los cambios que se realizan sobre estos datos, sigan ciertas reglas. Se puede considerar como otro modo de proteger la base de datos pero, además de tener que ver con la seguridad, tiene otras implicaciones.
- Herramientas de administración: Proporcionar herramientas que permitan administrar la base de datos de modo efectivo, lo que implica un diseño óptimo de las mismas.

## 4.2 Componente de un SGBD

Los elementos que proporcionan los servicios anteriores no se pueden generalizar, ya que varían mucho según la tecnología. Sin embargo, normalmente todo SGBD incluye los siguientes:

- Lenguajes de datos: Empleados para la comunicación con la base de datos, se distinguen tres tipos según su funcionalidad.
    - Lenguaje de definición de datos (DDL)
    - Lenguaje de control de datos (DCL)
    - Lenguaje de manipulación de datos (DML)
- Diccionario de datos: Se trata de esquemas que describen el contenido del SGBD incluyen los distintos objetos con sus propiedades.
- Optimizador de consultas: Permite determinar la estrategia óptima para la ejecución de consultas.
- Gestión de transacciones: Encargado de realizar el procesamiento de las transacciones.
- Planificador: Para programar y automatizar la realización de ciertas operaciones y procesos.
- Copias de seguridad: Para garantizar que la base de datos se puede devolver a un estado consistente en caso de que se produzca algún fallo o error grave.

## 4.3 Tipos de SGBD

Existen numerosos SGBD en el mercado que podemos clasificar según los siguientes criterios:

- Modelo lógico en el que se basan
    - Jerárquico
    - En red
    - Relacional
    - Objeto-relacional
    - Orientado a objetos
- Número de usuarios
    - Monousuario: solo permiten un usuario.
    - Multiusuario: permiten la conexión de varios usuarios.
- Número de sitios
    - Centralizados: en un solo servidor o equipo.
    - Distribuidos: en varios equipos que pueden ser homogéneos y heterogéneos.
- Ámbito de aplicación
    - Propósito general: orientados a toda clase de aplicaciones.
    - Propósito específico: centradas en un tipo específico de aplicaciones.
- Lenguajes soportados
    - SQL Estándar
    - NoSQL o nuevo lenguaje de consulta.

## 4.4 Sistemas gestores de bases de datos comerciales y libres

Con la llegada de internet, el software libre se ha consolidado como alternativa, técnicamente viable y económicamente sostenible al software comercial, contrariamente a lo que a menudo se piensa, convirtiéndose el software libre como otra alternativa para ofrecer los mismos servicios a un coste cada vez más reducido.

Sin embargo, debe tenerse en cuenta que lo que comúnmente se denomina software libre no significa que sea gratuito ya que muchas empresas ofrecen servicios de soporte y mantenimiento para productos, pero cuyo código sigue siendo accesible.

En cuanto a alternativas libres podemos optar por PostgreSQL frente a alternativas de pago como Oracle o SQL Server. Aunque cada vez son más los fabricante que ofrecen versiones gratuitas, con limitaciones de sus productos como es el caso de Oracle y su versión express.

Con independencia de la versión de producto también disponemos, tanto en los productos libres como los comerciales, de cada vez más documentación en línea que facilita enormemente su aprendizaje. Sin embargo, es importante tener en cuenta que el uso de un software libre suele implicar mayor conocimiento del producto, y por lo tanto un personal más cualificado.

En definitiva, usar software libre o no es una elección del consumidor. Debe considerar estos factores y otro factores de la manera que mejor se adapte a sus necesidades.

# 5 Bases de datos centralizadas y distribuidas

En un sistema de bases de datos centralizado todos los componentes residen en un único lugar físico. Los clientes acceden al sistema a través de distintas interfaces que se conectan al servidor. Esta situación a pesar de estar muy extendida no está exenta de problemas, ya que podemos sufrir los típicos problemas de cuellos de botella, por saturación de peticiones, o de falta de disponibilidad, por caída del servidor central.

Por esta razón una solución cada vez más extendida es una arquitectura donde se opta por un esquema distribuido en el que los componentes se distribuyen en distintos computadores comunicados a través de una red de cómputo. Dichos sistemas se conocen con el nombre de Sistemas Gestores de Bases de Datos Distribuidas.

La principales ventajas de estos sistemas son:

- Mejora de rendimiento: Cuando grandes bases de datos se distribuyen las consultas y transacciones se hacen menos complejas al afectar a una solo localización en muchas ocasiones siendo la propia bases de datos local, y no saturan el resto de localizaciones.
- Fiabilidad: Al distribuirse los datos es más fácil asegurar la fiabilidad del sistema, ya que si un sitio es poco probable que afecte a todas las transacciones.
- Disponibilidad: Al igual que el punto anterior, antes caída de un sitio tendremos otros disponibles para acceder a la información. Mejora especialmente aplicando replicación.

Por el contrario, distribuir implica una mayor complejidad en el diseño, implementación y gestión de los datos.

## 5.1 Técnicas de distribución

Diseñar bases de datos distribuidas implica decidir sobre cómo fragmentarlas, cómo replicarlas y cómo distribuirlas, además del diseño previo de la base de datos en sí.

**Fragmentación**

Consideraremos que nuestra base de datos está formada por unidades lógicas que son las relaciones o tablas. Cada una de las tablas podrá dividirse almacenando sus datos filtrando por algún campo que consideremos, lo cual se denominaría fragmentación horizontal. O bien repartiendo los atributos en distintas localizaciones según necesidades de acceso, lo cual se denominaría fragmentación vertical.

La información sobre la fragmentación se guarda en lo que se denomina esquema de fragmentación, que es una definición de todos los atributos de la base de datos y cómo están fragmentados.

Fragmentar permite acercar los datos al consumidor, reducir el tráfico de red y mejorar la disponibilidad de los datos.

**Replicación**

Es fundamentalmente útil para mejorar la disponibilidad de los datos. El caso más extremo es la replicación en todos los sitios de la misma copia de la base de datos, que también es el caso de mayor disponibilidad, aunque puede reducir considerablemente la eficiencia en operaciones de actualización dado que cada operación debe realizar en todas las localizaciones.

La replicación distribuye carga, acelera consultas y mejora la disponibilidad, pero sobre todo, es óptima para sistemas con muchas lecturas. Sin embargo, requiere muchas más actualizaciones con el consiguiente consumo de almacenamiento.