

TEMA 1: **INTRODUCCIÓN A** **LAS BASES DE DATOS**

David Bataller Signes

ÍNDICE

1. ENTIDADES, ATRIBUTOS Y VALORES.

2. FICHEROS.

3. BASES DE DATOS (BD).

3.1 MODELOS DE BD.

4. LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS (SGBD).

4.1 EVOLUCIÓN DE LOS SGBD

4.2 FUNCIONES QUE DEBE GARANTIZAR UN SGBD

4.3 COMPONENTES FUNCIONALES DE LOS SGBD

4.4 ARQUITECTURA DE LOS SGBD

Los datos que se utilizan de manera informatizada se guardan, habitualmente, en bases de datos (BD).

Para abordar el estudio de las BD i de los programas que sirven para gestionarlos, los sistemas gestores de bases de datos (SGBD), es imprescindible entender previamente algunos conceptos teóricos fundamentales, referentes a los datos y a su representación. A parte de los conceptos relativos a los datos y a las bases de datos, hay que conocer que representaciones de la información se utilizan habitualmente así como la evolución de los programas en este ámbito.

Para poder trabajar con BD hay que saber distinguir tres ámbitos diferenciados, pero que a su vez están fuertemente relacionados, los cuales hacen referencia a la realidad, a su conceptualización y a su representación informática posterior. Consideramos tres “mundos”:

- El mundo real. Está constituido por los objetos de la realidad que nos interesan y con los cuales vamos a trabajar.
- El mundo conceptual. Es el conjunto de conocimientos o informaciones obtenidas gracias a la observación de la parte del mundo real que nos interesa. Un mismo mundo real puede dar lugar a diferentes mundos conceptuales dependiendo de la manera de percibir la realidad, o dependiendo de los intereses del observador.
- El mundo de las representaciones. Está formado por las representaciones informáticas, o datos, del mundo conceptual necesarias para poder trabajar.

A partir de esto definimos los datos como representaciones informáticas de la información disponible, relativa a los objetos del mundo real. Y el mundo de las representaciones está formado por los datos informatizados con los que trabajamos.

La conversión de las concepciones en datos no es automática sino que requiere pasar por dos fases de diseño que son:

- Fase de diseño lógico. Se trabaja con el modelo abstracto de datos obtenido al final de la etapa de diseño conceptual, para traducirlo al

modelo de datos utilizado por el SGBD con el que se quiere implementar y mantener la futura BD.

- Fase de diseño físico. Se pueden hacer modificaciones sobre el esquema obtenido en la fase de diseño anterior para incrementar la eficiencia en algunas de las operaciones que se hagan con los datos.

Debido a todo esto vemos que una misma realidad se puede representar de maneras distintas dependiendo de las decisiones de diseño tomadas y de la tecnología usada.

1. Entidades, atributos y valores.

Tres elementos caracterizan fundamentalmente las informaciones:

- Entidades. Son los objetos del mundo real que se conceptualizan. Son identificables, es decir, se distinguen los unos de los otros. Y nos interesa algunas (como mínimo una) de sus propiedades.
- Atributos. Son las propiedades de las entidades que nos interesan.
- Valores. Son los contenidos concretos de los atributos.

En un principio, los atributos deberían almacenar un solo valor en cada instante. De esta manera nuestros modelos serán compatibles con el modelo lógico de datos más utilizado, el modelo relacional.

Se pueden distinguir dos tipos de entidad. La entidad tipo, que es un tipo genérico de entidad (abstracción) que hace referencia a una clase de cosas, por ejemplo, las motos, en general. Y la entidad instancia que se refiere a la conceptualización de un objeto concreto del mundo real, como una moto en concreto, diferenciable del resto de objetos del mismo tipo gracias a alguna propiedad.

A veces el valor de un atributo es desconocido o no existe. Para representar esta circunstancia, el atributo debe admitir el valor nulo (indica que no hay ningún valor asociado a un atributo determinado de una entidad instancia concreta). Para que un atributo admita el valor nulo se debe especificar esta posibilidad a la hora de definir el dominio.

Atributo identificador. Es aquel que permite distinguir inequívocamente cada entidad instancia del resto, debido a que su valor es único, y no se repite en diferentes entidades instancia.

Hay veces que un solo atributo no es suficiente para identificar inequívocamente por lo que se recurre a la combinación de los valores de dos o más atributos de la misma entidad. Por esto aparece el concepto de clave.

Clave. Atributo o conjunto de atributos que permiten identificar inequívocamente las instancias de una entidad.

Todo atributo identificador es al mismo tiempo una clave. Pero los atributos que forman parte de un conjunto de más de un atributo que actúan como clave de la entidad no son identificadores, ya que, por si mismos no son capaces de identificar las entidades instancia.

Ni los atributos identificadores ni los que forman parte de una clave pueden admitir el valor nulo porque no servirían para distinguir las entidades instancia.

2. Ficheros.

La representación informática más frecuente en el ámbito de las BD es la representación tabular (en forma de tabla), la cual se implementa habitualmente en ficheros que se estructuran en registros y campos. Por esto también se puede definir las BD como conjuntos de ficheros interrelacionados (o que almacenan datos que están interrelacionados).

Cada tabla representa una entidad genérica y está estructurada en filas y columnas.

Cada fila representa una entidad genérica y cada columna representa un atributo.

Cada celda (intersección de una fila y de una columna) almacena el valor que tiene el atributo de la entidad instancia determinada.

La implementación informática de las representaciones tabulares se materializa mediante los ficheros de datos. Se entiende por fichero a la

implementación informática de una tabla, con los datos estructurados en registros (implementación de cada entidad instancia, equivale a una fila de la tabla) y campos (implementación de cada atributo, equivale a una columna de la tabla).

3. Bases de datos (BD).

Si partimos de un número concreto de entidades tipo necesitaremos, como mínimo, el mismo número de tablas para representarlas. Estas tablas o ficheros no serán objetos inconexos sino que deben estar interrelacionados.

Definimos interrelaciones como, las informaciones que permiten asociar las entidades entre ellas.

Las interrelaciones entre los registros de dos o más tablas se hacen mediante campos del mismo tipo de dato que almacena los mismos valores.

La interrelación entre ficheros implica que los cambios de valor de los campos que sirven para interrelacionarlos deben quedar reflejados en todos los ficheros implicados, para mantener la coherencia de los datos.

Una BD consiste en un conjunto de ficheros de datos interrelacionados o unos conjuntos estructurados de datos organizados en entidades interrelacionadas.

Aunque una definición más completa sería que es la representación informática de los conjuntos de entidades instancia correspondientes a diferentes entidades tipo y de las relaciones entre estas. Este conjunto estructurado de datos debe ser utilizado de manera compartida y simultanea por una pluralidad de usuarios de diferente tipo.

3.1 Modelos de BD.

Los modelos de datos son unos conjuntos de herramientas lógicas para describir los datos, sus interrelaciones, su significado y las restricciones a aplicar para garantizar su coherencia.

Todos los modelos de BD, en general, proporcionan tres tipos de herramientas:

- Estructuras de datos. Elementos con los que se construyen las BD, como tablas, árboles, etc.
- Reglas de integridad. Restricciones que los datos deberán respetar, como por ejemplo tipo de dato, dominios, llaves, etc.
- Operaciones a realizar con los datos. Altas, bajas, modificaciones y consultas, como mínimo.

Los modelos de datos más utilizados a lo largo del tiempo han sido los siguientes, expuestos en orden de aparición:

1. Jerárquico. Las BD jerárquicas se concibieron al principio de los años sesenta, y aún se utilizan gracias al buen rendimiento y la mejor estabilidad que proporcionan con grandes volúmenes de información.

Las BD jerárquicas almacenan la información en una estructura jerárquica que podemos imaginar con una forma de árbol invertido, donde cada nodo padre puede tener diferentes hijos. El nodo superior, que no tiene padre, se conoce como raíz. Y los nodos que no tienen hijos se llaman hojas.

2. En red. Al principio de los años setenta, en el mercado, fueron surgiendo BD que seguían un modelo en red, similar al modelo jerárquico, con registros interrelacionados mediante una estructura en forma de árbol invertido, pero más flexible, ya que permitía que los nodos tuvieran más de un solo padre.

3. Relacional. A partir de los años ochenta, se empezaron a comercializar gran cantidad de BD que aplicaban este modelo. Los datos se estructuran en representaciones tabulares, llamadas tablas, que representan entidades tipo del mundo conceptual, y que están formadas por filas y columnas. Las columnas forman los campos, que implementan los atributos, es decir, las características que nos interesan de las entidades. Y las filas son los registros, que implementan las entidades instancia,

constituidas por los conjuntos de los valores que presentan los campos correspondientes a cada instancia.

El modelo relacional se basa en la lógica de predicados y en la teoría de conjuntos (Áreas de la lógica y de las matemáticas). Actualmente, es el sistema más ampliamente utilizado para modelizar datos.

4. Relacional con objetos/orientado a objetos. El modelo de datos relacionales con objetos es una extensión del modelo relacional en sentido estricto.

Este nuevo modelo admite la posibilidad de que los tipos de datos sean, además los tradicionales, tipos abstractos de datos (TAD). Con esta particularidad, acercan los sistemas de BD relacionales al paradigma de la programación orientada a objetos.

Los modelos estrictamente orientados a objetos definen las BD en términos de objetos, de sus propiedades y, lo que es más innovador, de sus operaciones. Los objetos con una misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías. Las operaciones de cada clase se especifican en términos de procedimientos predefinidos, llamados métodos.

Actualmente han aparecido las BD multidimensionales que son BD diseñadas para optimizar el procesamiento analítico en línea, conocido como OLAP (On-Line Analytical Processing). La característica más destacable del procesamiento OLAP es la estructuración de los datos en los llamados cubos OLAP (estructura de datos que permite accesos rápidos a la información y que se organiza esta información en diversas perspectivas o dimensiones).

Un caso particular de BD multidimensional son las BD multivalor. Las BD multivalor suelen diseñar BD donde los atributos almacenan listas de valores, a diferencia de las BD relacionales donde los atributos son monovalor. Muchas veces se conoce a las BD multivalor como BD post-relacionales.

Últimamente, y por influencia de las técnicas de desarrollo de software orientadas al objeto, han aparecido algunos modelos semánticos como

nuevas extensiones del modelo ER originario, entre las que destaca especialmente el UML (unified modeling language).

La notación del lenguaje UML es diagramática, como la notación del modelo ER. Los diagramas UML se pueden utilizar para expresar los diseños conceptuales, tal como se hace con los diagramas ER.

Otro tipo son las BD distribuidas, donde hacemos referencia a la utilización de datos almacenados en diferentes ubicaciones, a veces muy distantes entre sí, pero que al mismo tiempo están conectadas, mediante una red de comunicaciones.

4. Los SGBD.

Los SGBD son un tipo de software que tiene como finalidad la gestión y el control de las BD.

Es interesante conocer la evolución de este tipo de software a lo largo de su historia y los objetivos que todos ellos perseguían. También hay que destacar las nociones relativas a la arquitectura de los SGBD como a las aplicaciones. También ver las diferentes tipologías de usuarios y administradores de BD, y lenguajes que han usado para comunicarse con los sistemas gestores.

4.1 Evolución de los SGBD.

Para entender mejor por qué los SGBD son hoy en día tal como los conocemos, conviene repasar brevemente su historia. Al igual que en otros ámbitos de software (como, por ejemplo, en el de los sistemas operativos), la evolución de los SGBD ha sido, a menudo, intrínsecamente ligada a la evolución del hardware.

Años cincuenta: procesamiento secuencial

Inicialmente, el único hardware disponible para almacenar la información consistía en paquetes de cintas perforadas y en cintas magnéticas. Estos dispositivos sólo se podían leer de forma secuencial y, por tanto, el software de la época estaba limitado por esta circunstancia. Como el tamaño de los datos a procesar era muy superior a la de la memoria

principal de las computadoras, los programas sólo podían realizar procesos por lotes, de la siguiente forma:

- Obteniendo los datos en un orden determinado (desde una o más cintas).
- Haciendo algún cálculo sobre los datos.
- Escribiendo el resultado (en una nueva cinta).

Años sesenta y setenta: sistemas centralizados

Durante la mayor parte de las dos décadas de los años sesenta y setenta, los SGBD tuvieron una estructura centralizada, como correspondía a los sistemas informáticos entonces: un gran ordenador para cada organización que se le pudiera costear, y una red de terminales no inteligentes, sin capacidad propia para procesar datos.

Inicialmente, sólo se utilizaban para gestionar procesos por lotes con grandes volúmenes de datos. Posteriormente, con la aparición de los terminales, a veces conectados mediante la línea telefónica, se fueron elaborando aplicaciones transaccionales, por ejemplo para reservar y comprar billetes en líneas de transportes, o por realizar operaciones financieras.

Los programas aún estaban muy ligados al nivel físico, y se debían modificar siempre que se hacían cambios en el diseño de la BD, ya que estos cambios implicaban, a su vez, modificaciones en la estructura física de la BD. El personal que realizaba estas tareas tenía que estar altamente cualificado.

Años ochenta: SGBD relacionales

Aunque desde el principio de los años setenta ya se había definido el modelo relacional, y el acceso no procedimental a los datos organizados siguiendo este modelo, no fue hasta los años ochenta cuando fueron apareciendo SGBD relacionales en el mercado.

La razón de esta demora en el uso de los sistemas relacionales fue el pobre rendimiento que ofrecían inicialmente los productos relacionales en comparación con las BD jerárquicas y en red. Pero la innovación en el hardware, primero con los miniordenadores y posteriormente con los microordenadores, supuso un cierto abaratamiento de la informática y su extensión a muchas más organizaciones.

Hasta entonces, el trabajo de los programadores que trabajaban con BD prerrelacionales había sido demasiado pesada, ya que, por un lado, debían codificar sus consultas de manera procedimental, y de otra, tenían que estar pendientes de su rendimiento y hacer consideraciones de índole física a la hora de codificarlas.

Pero debido a la expansión de la informática que tuvo lugar durante la década que comentamos, era necesario simplificar el desarrollo de las aplicaciones. Los SGBD lo lograron, todo independizando los programas de los aspectos físicos de los datos.

Además, la aparición del lenguaje de consulta estructurado (structured query language, SQL) y, sobre todo, su estandarización a partir del año 1986 facilitar enormemente el uso de los sistemas relacionales y, por tanto, su implantación masiva.

Finalmente, las BD relacionales pudieron competir, incluso, en materia de rendimiento con las jerárquicas y con las estructuradas en red, con lo cual terminaron reemplazando sus competidoras en la mayoría de los casos.

Años noventa: BD distribuidas, arquitecturas cliente / servidor, y lenguajes de cuarta generación

Como ya sabemos, los primeros sistemas de BD eran centralizados: todos los datos del sistema estaban almacenadas en un único gran ordenador al que se podía acceder desde diferentes terminales. Pero el éxito gradual de los ordenadores personales, cada vez más potentes y con precios más competitivos, junto con el desarrollo de las redes, posibilitó la distribución de una misma BD en diferentes ordenadores (o nodos).

En función del número de SGBD utilizados, los sistemas distribuidos pueden ser de dos tipos:

- Homogéneos, si todos los nodos utilizan el mismo SGBD. Las interacciones entre los diferentes nodos son más sencillas. Pero las actualizaciones del sistema gestor implicarán necesariamente a todos los nodos.
- Heterogéneos, si cada nodo utiliza un SGBD diferente. Las interacciones entre los diferentes nodos pueden ser más complicadas. Pero habrá más flexibilidad a la hora de actualizar el sistema gestor de cada nodo.

La tecnología utilizada habitualmente en la distribución de BD es la arquitectura cliente/servidor. Actualmente, todos los SGBD comerciales están adaptados a esta realidad. El funcionamiento de los sistemas basados en este tipo de arquitectura es, esquemáticamente, lo siguiente: dos procesos se ejecutan en un mismo sistema o en dos de diferentes, de tal manera que uno hace de cliente y el otro de servidor.

Finalmente, durante los años noventa, la implantación en todas las BD, incluso en pequeños sistemas personales, motivó la aparición de los llamados lenguajes de cuarta generación, los cuales se siguen utilizando en la actualidad.

Se trata de lenguajes muy sencillos, pero al mismo tiempo muy potentes, especializados en el desarrollo de aplicaciones centradas en el acceso a BD. Ofrecen muchas facilidades para definir, generalmente de manera visual, ventanas desde las que se pueden consultar, introducir, modificar o borrar datos, incluso en entornos cliente/servidor.

Tendencias actuales: orientación a objetos, Internet, y elementos multimedia

Actualmente, los SGBD relacionales acaparan el mercado. Han evolucionado de tal manera que ya no tienen competidores en rendimiento, fiabilidad o seguridad. Por otra parte, ya no necesitan habitualmente tareas de mantenimiento planificadas que conlleven la

parada periódica. Por tanto, la disponibilidad que ofrecen es muy elevada, ya que se acerca a las veinticuatro horas de todos los días del año.

Pero, al mismo tiempo, todos están inmersos en un complejo proceso de transformación con el fin de adaptarse a las innovaciones tecnológicas de más éxito:

Las tecnologías multimedia. Los tipos de datos que tradicionalmente admitían los SGBD ahora se ven incrementados por otros nuevos que permiten almacenar imágenes y sonidos.

La orientación a objetos. Este paradigma de la programación ha terminado influyendo en la orientación de muchos SGBD, que siguen alguna de las dos líneas esbozadas a continuación:

- SGBD relacionales con objetos, los cuales admiten tipos abstractos de datos (TAD), además de los tipos tradicionales.
- SGBD orientados a objetos, que estructuran los datos en clases, las cuales comprenden tanto los datos (atributos) como las operaciones sobre ellas (métodos).

Las clases se estructuran jerárquicamente, de tal manera que las de niveles inferiores (subclases) heredan las propiedades de las de niveles superiores (Superclases).

Internet. Hoy en día, la mayoría de SGBD profesionales incorporan los recursos necesarios para apoyar a los servidores de páginas web dinámicas (es decir, con acceso a los datos contenidos en un SGBD alojado en el servidor web correspondiente).

Otra línea de innovación que siguen algunos SGBD es el trabajo con los llamados almacenes de datos (data warehouse). Estos almacenes consisten en réplicas elaboradas de los datos generados por el funcionamiento cotidiano de la organización o la empresa de que se trate, durante un cierto período de tiempo para realizar análisis estratégicos de índole financiera, de mercados, etc.

El lenguaje de marcas extensible (XML) también influye en el mundo de los SGBD, aunque inicialmente no se puede concebir como una tecnología

para dar servicio a las BD, sino para estructurar documentos muy grandes. Pero esta capacidad para almacenar los datos de que se compone un documento lo hacen susceptible de ser utilizado, también, en el ámbito de las BD.

Hay dos maneras de incorporar la tecnología XML en los SGBD:

- Los SGBD relacionales con soporte para XML
- Los sistemas gestores para BD nativas XML

La utilización de datos estructurados mediante el estándar XML resulta especialmente interesante en el intercambio de información entre sistemas basados en plataformas poco compatibles entre ellas.

4.2 Funciones que debe garantizar un SGBD

Todos los SGBD del mercado quieren alcanzar una serie de objetivos y ofrecer una serie de funcionalidades, con más o menos acierto, que actualmente se consideran indispensables para el buen funcionamiento de cualquier sistema de información:

- Posibilitar las consultas no predefinidas de cualquier complejidad.

Los usuarios autorizados de un SGBD deben poder plantear directamente al sistema cualquier consulta sobre los datos almacenados, de la complejidad que sea necesaria, respetando, eso sí, las reglas sintácticas para que la sentencia sea correcta.

A continuación, el SGBD debe ser capaz de responder él mismo a la consulta formulada, sin que sea necesario recurrir a ninguna aplicación externa.

- Garantizar la independencia física y la independencia lógica de los datos.

Hay que garantizar la máxima independencia física de los datos respecto a los procesos usuarios, en general (es decir, tanto en cuanto a las consultas interpretadas por el SGBD como a los programas externos que acceden a la BD), de tal manera que se puedan llevar a cabo todo tipo de cambios tecnológicos de índole física para mejorar el rendimiento, sin que ello

implique tener que modificar ni las consultas a la BD ni las aplicaciones que acceden.

De manera similar, también es deseable la independencia lógica de los datos, la que implica que las modificaciones en la descripción lógica de la BD no deben impedir la ejecución normal los procesos usuarios no afectados por ellas.

Y, con respecto a la independencia lógica de los datos, incluso puede interesar que convivan diferentes visiones lógicas de una misma BD, en función de las características concretas de los diferentes usuarios o grupos de usuarios.

- Evitar o solucionar los problemas derivados de la redundancia.

La repetición de los datos (redundancia) es peligrosa, ya que cuando se actualizan pueden perder la integridad. Cuando se modifica el valor de un dato que está repetido, se modifican simultáneamente los valores de sus repeticiones para que se mantenga la coherencia entre todas.

Otro tipo de duplicidad admisible es la que constituyen los llamados datos derivados. Se trata de datos almacenados en la BD, que en realidad son el resultado de cálculos realizados con otros datos también presentes en la misma BD.

- Proteger la integridad de los datos.

Además de la redundancia, hay muchos otros motivos que pueden dañar la consistencia de los datos, tales como los errores humanos, las deficiencias en la implementación de los algoritmos de las aplicaciones, las averías de los soportes físicos de almacenamiento i las transacciones incompletas como consecuencia de las interrupciones del suministro eléctrico.

Los SGBD deben proteger la integridad de los datos en todos estos casos. Para ello disponen, por un lado, de las reglas de integridad, también llamadas restricciones, y de otra, los sistemas de restauración basados en copias de seguridad.

Mediante las reglas de integridad, el sistema valida automáticamente ciertas condiciones al producirse una actualización de datos, y lo autoriza si las cumple, o deniega el permiso de lo contrario.

Los SGBD también proporcionan herramientas para realizar periódicamente copias de seguridad de los datos que permiten restaurar los datos dañados y devolverlos a un estado consistente.

- Permitir la concurrencia de usuarios.

Un objetivo fundamental de todo SGBD es posibilitar de manera eficiente el acceso simultáneo a la BD por parte de muchos usuarios. Además, esta necesidad ya no está circunscrita sólo a grandes compañías o en administraciones públicas con muchos usuarios, sino que cada vez es más frecuente, debido a la expansión de Internet y el éxito de las páginas dinámicas, alojadas en servidores web que deben incorporar un SGBD.

Para tratar correctamente los problemas derivados de la concurrencia de usuarios, los SGBD utilizan fundamentalmente dos técnicas: las transacciones y los bloqueos.

Una transacción consiste en un conjunto de operaciones simples que se han de ejecutar como una unidad.

Un bloqueo consiste en impedir el acceso a determinados datos durante el tiempo en que sean utilizadas para una transacción. Así se consigue que las transacciones se ejecuten como si estuvieran aisladas, de tal manera que no se producen interferencias entre ellas.

- Contribuir a la seguridad de los datos.

La expresión seguridad de los datos hace referencia a su confidencialidad. A menudo, el acceso a los datos no debe ser libre o, al menos, no lo debe ser totalmente.

Los SGBD deben permitir definir autorizaciones de acceso a las BD, todo estableciendo permisos diferentes en función de las características del usuario o del grupo de usuarios.

Estos mecanismos de seguridad requieren que cada usuario se pueda identificar. Lo más frecuente es utilizar un nombre de usuario y una contraseña asociada para cada usuario.

Otro aspecto a tener en cuenta al hablar de la seguridad de los datos es su encriptación. Muchos SGBD ofrecen esta posibilidad, en alguna medida. Las técnicas de encriptación permiten almacenar la información utilizando códigos secretos que no permiten acceder a los datos a personas no autorizadas.

4.3 Componentes funcionales de los SGBD

El software que conforma los SGBD se divide en diferentes módulos, encargados de las respectivas funcionalidades que debe garantizar el sistema. Los componentes funcionales de los SGBD más importantes son el gestor de almacenamiento y el procesador de consultas.

Gestor de almacenamiento

El gestor de almacenamiento proporciona la interfaz entre los datos, consideradas a bajo nivel, y las consultas y los programas que acceden a la BD. Los datos se almacenan en disco utilizando el sistema de archivos que proporciona el sistema operativo utilizado. Y el gestor traduce las instrucciones DML a órdenes comprensibles por sistemas de archivos a bajo nivel.

Los componentes del gestor de almacenamiento son los siguientes:

- Gestor de autorizaciones y de integridad. Comprueba que se satisfagan tanto las restricciones de integridad como las autorizaciones de los usuarios para acceder a los datos.
- Gestor de transacciones. Asegura que la BD se mantenga en un estado de consistencia a pesar de los fallos del sistema, y también que las transacciones concurrentes no interfieran entre ellas.
- Gestor de archivos. Gestiona la reserva de espacio de almacenamiento en disco y las estructuras de datos utilizadas para representar la información almacenada en disco.

- Gestor de memoria intermedia. Transfiere los datos desde el disco a la memoria principal, y decide qué datos se deben tratar en memoria caché. Permite al sistema tratar con datos de tamaño muy superior a la de la memoria principal.

Procesador de consultas

El procesador de consultas ayuda al SGBD a simplificar el acceso a los datos.

Los componentes del procesador de consultas son los siguientes:

- Intérprete DDL. Interpreta las instrucciones de tipo DDL y registra las definiciones en el diccionario de datos.
- Compilador DML. Traduce las instrucciones DML formuladas en un lenguaje de consultas (normalmente, SQL) a una serie de instrucciones a bajo nivel que puede interpretar el motor de evaluación de consultas. Al realizar la traducción mencionada, un buen compilador DML también se encargará de hacer una optimización de consultas eligiendo, entre todas las alternativas, la de menor coste.
- Motor de evaluación de consultas. Ejecuta las instrucciones de bajo nivel generadas por el compilador DML.

4.4 Arquitectura de los SGBD

En 1975, el comité ANSI/X3/SPARC propuso una arquitectura para los SGBD estructurada en tres niveles de abstracción (interno, conceptual y externo), que resulta muy útil para separar los programas de aplicación de la BD considerada desde un punto de vista físico.

Por otra parte, también resulta interesante examinar la arquitectura los SGBD desde un punto de vista funcional, ya que conociendo los diferentes componentes, y los flujos de datos y de control podemos entender el funcionamiento de los SGBD, y estaremos en mejores condiciones de utilizarlos de una manera óptima.

Para gestionar las BD, los SGBD deben conocer su estructura (es decir, las entidades, los atributos y las interrelaciones que contiene, etc.). Los SGBD necesitan disponer de una descripción de las BD que deben gestionar. Esta definición de la estructura recibe el nombre de esquema de la BD, y debe estar constantemente al alcance del SGBD para que éste pueda cumplir sus funciones.

De acuerdo con el estándar ANSI/X3/SPARC, debería haber tres niveles de esquemas:

- En el nivel externo se sitúan las diferentes visiones lógicas que los procesos usuarios (programas de aplicación y usuarios directos) tienen de las partes de la BD que utilizan. Estas visiones se denominan esquemas externos.
- En el nivel conceptual hay una sola descripción lógica básica, única y global, que llamamos esquema conceptual, y que sirve de referencia para en el resto de esquemas.
- En el nivel físico hay una única descripción física, que llamamos esquema interno.

Al definir un esquema externo, sólo se incluirán los atributos y entidades que interesen.

En el esquema conceptual, se describirán las entidades tipo, sus atributos, las interrelaciones y también las restricciones o reglas de integridad.

El esquema interno o físico contendrá la descripción de la organización física de la BD: caminos de acceso (índices, hashing, apuntadores ...), codificación de los datos, gestión del espacio, tamaño de la página, etc.