

## **Conceptos del sistema de base de datos y Arquitectura**

La arquitectura de los paquetes DBMS ha evolucionado de los primeros sistemas monolíticos, donde el conjunto El paquete de software DBMS era un sistema estrechamente integrado, para el DBMS moderno paquetes que son de diseño modular, con una arquitectura de sistema cliente / servidor. El crecimiento reciente en la cantidad de datos que requieren almacenamiento ha llevado a sistemas de bases de datos con arquitecturas distribuidas compuestas por miles de computadoras que administran los almacenes de datos. tipos de máquinas servidor: servidores web, servidores de bases de datos, servidores de archivos, aplicaciones servidores, etc. Los entornos de computación en la nube actuales consisten en miles de grandes servidores que administran los llamados big data para los usuarios de la Web. En una arquitectura DBMS cliente / servidor básica, la funcionalidad del sistema se distribuye entre dos tipos de mdulos.1 Un mdulo cliente esta diseñado típicamente para que se ejecutará en un dispositivo móvil, estación de trabajo de usuario o computadora personal (PC). Normalmente, los programas de aplicación y las interfaces de usuario que acceden a la base de datos se ejecutan en el módulo cliente. Por lo tanto, el módulo de cliente maneja la interacción del usuario y proporciona las interfaces fáciles de usar, como aplicaciones para dispositivos móviles, o GUI (interfaces gráficas de usuario) basadas en formularios o menús para PC. El otro tipo de módulo, llamado un módulo de servidor, generalmente maneja el almacenamiento de datos, el acceso, la búsqueda y otras funciones.

### **Modelos de datos, esquemas e instancias**

Una característica fundamental del enfoque de la base de datos es que proporciona nivel de abstracción de datos. La abstracción de datos generalmente se refiere a la supresión de detalles de la organización y el almacenamiento de datos, y el resaltado de las características esenciales para una mejor comprensión de los datos. Una de las principales características del enfoque de la base de datos es apoyar la abstracción de datos para que diferentes usuarios puedan percibir datos en su nivel de detalle preferido.

**Un modelo de datos:** una colección de conceptos que se puede utilizar para describir la estructura de una base de datos: proporciona los medios necesarios Por estructura de una base de datos nos referimos a los tipos de datos, relaciones y restricciones que se aplican a los datos. La mayoría de los modelos de datos también incluyen conjunto de operaciones básicas para especificar recuperaciones y actualizaciones en la base de datos.para lograr esta abstracción.

Por otro lado, operaciones genéricas para insertar, eliminar, modificar o recuperar cualquier tipo de objeto a menudo se incluyen en los datos básicos, los conceptos para especificar el comportamiento son fundamentales para la orientación a objetos. modelos de datos pero también se están incorporando en más tradicionales modelos de datos. En los datos de modelo relacionales básicos, existe una disposición para adjuntar comportamiento a las relaciones en forma de módulos almacenados persistentes, conocidos popularmente como procedimientos almacenados

### **Categorías de modelos de datos**

Se han propuesto muchos modelos de datos, que podemos categorizar según los tipos de conceptos que utilizan para describir la estructura de la base de datos. De alto nivel o los

modelos de datos conceptuales proporcionan conceptos que se acercan a la forma en que muchos usuarios perciben los datos, mientras que los modelos de datos físicos o de bajo nivel proporcionan conceptos que describen los detalles de cómo se almacenan los datos en los medios de almacenamiento de la computadora, generalmente magnéticos discos. Los conceptos proporcionados por modelos de datos físicos generalmente están pensados para computadoras especialistas, no para usuarios finales.

Los modelos de datos representativos ocultan muchos detalles de los datos almacenamiento en disco, pero se puede implementar directamente en un sistema informático. Los modelos de datos conceptuales utilizan conceptos como entidades, atributos y relaciones. Una entidad representa un objeto o concepto del mundo real, como un empleado o un proyecto. del minimundo que se describe en la base de datos. Un atributo representa algunos propiedad de interés que describe con más detalle una entidad, como el nombre del empleado o salario. Una relación entre dos o más entidades representa una asociación entre las entidades, por ejemplo, una relación de trabajo entre un empleado y un proyecto.

Los modelos de datos representativos o de implementación son los modelos que se utilizan con mayor frecuencia en los DBMS comerciales tradicionales. Estos incluyen el relacional ampliamente utilizado modelo de datos, así como los llamados modelos de datos heredados: la red y modelos jerárquicos, que se han utilizado ampliamente en el pasado. Los modelos de datos representativos representan datos mediante el uso de estructuras de registros y, por lo tanto, son a veces llamados modelos de datos basados en registros.

Podemos considerar el modelo de datos de objetos como un ejemplo de una nueva familia de modelos de datos de implementación más cercanos a modelos de datos conceptuales. Un estándar para bases de datos de objetos llamado el modelo de objetos ODMG ha sido propuesto por el Grupo de gestión de datos de objetos (ODMG). Los modelos de datos también se utilizan con frecuencia como modelos conceptuales de alto nivel, particularmente en el dominio de la ingeniería de software.

Los modelos de datos físicos describen cómo se almacenan los datos como archivos en la computadora al representar información como formatos de registro, orden de registro y rutas de acceso. Una ruta de acceso es una estructura de búsqueda que hace que la búsqueda de una base de datos particular registros eficientes, como indexación o hash. Otra clase de modelos de datos se conoce como modelos de datos autodescriptivos. El almacenamiento en sistemas basados en estos modelos combina la descripción de los datos con los propios valores de los datos. En los DBMS tradicionales, la descripción (esquema) está separada de los datos. Estos modelos incluyen XML así como muchos los almacenes de clave-valor y los sistemas NOSQL que se crearon recientemente para administrar big data.

### **Esquemas, instancias y de la base de datos**

En un modelo de datos, es importante distinguir entre la descripción del base de datos y la propia base de datos. La descripción de una base de datos se llama esquema de base de datos, que se especifica durante el diseño de la base de datos y no se espera cambiar con frecuencia. Un esquema mostrado se denomina diagrama de esquema. Un diagrama de esquema muestra solo algunos aspectos de un esquema, como los nombres de tipos de registros y elementos de datos, y algunos tipos de restricciones. Otros aspectos no son especificado en el diagrama de esquema; Los datos de la base de datos en un momento determinado se denominan base de datos instantánea. También se denomina conjunto actual de ocurrencias o instancias en la base de datos. En un estado de base de datos dado, cada construcción de esquema tiene su propia corriente conjunto de instancias

e pueden construir muchos estados de la base de datos para corresponder a un esquema de base de datos particular. Cada vez que insertamos o eliminamos un registro o cambiamos el valor de un elemento de datos en un registro, cambiamos un estado de la base de datos a otro estado. Cuando definimos una nueva base de datos, especificamos su esquema de base de datos solo al DBMS. En este punto, el estado de la base de datos correspondiente es el estado vacío con sin datos. Obtenemos el estado inicial de la base de datos cuando la base de datos está poblada o cargada con los datos iniciales. A partir de ese momento, cada vez que se actualice la operación se aplica a la base de datos, obteniendo otro estado de la base de datos. En cualquier punto con el tiempo, la base de datos tiene un estado actual.

El DBMS es parcialmente responsable de garantizar que cada estado de la base de datos sea un estado válido, es decir, un estado que satisface la estructura y las restricciones especificadas en el esquema. Por lo tanto, especificar un esquema correcto para el DBMS es extremadamente importante y el esquema debe estar diseñado con sumo cuidado. El DBMS almacena las descripciones del esquema, construcciones y restricciones, también llamadas metadatos, en el catálogo de DBMS, por lo que el software DBMS puede consultar el esquema siempre que lo necesite. El esquema a veces se llama intensión, y un estado de base de datos se llama una extensión del esquema, no se supone que el esquema cambie con frecuencia, no es infrecuente que los cambios deban aplicarse ocasionalmente al esquema como los requisitos de la aplicación cambian.

### **Arquitectura de tres esquemas e independencia de datos**

Tres de las cuatro características importantes del enfoque de base de datos son:

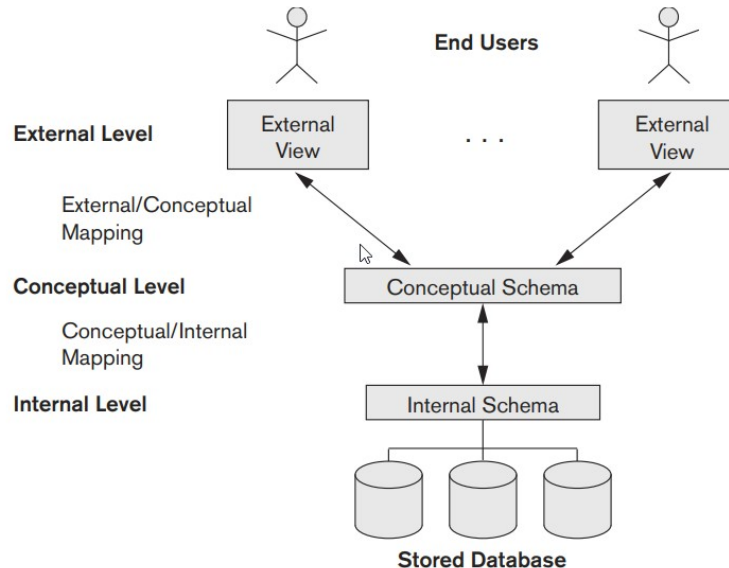
1. El uso de un catálogo para almacenar la descripción de la base de datos (esquema) de modo que sea autodescritivo.
2. Aislamiento de programas y datos (programa-datos e independencia del funcionamiento del programa).
3. Soporte de múltiples vistas de usuarios.

### **La arquitectura de tres esquemas**

El objetivo de la arquitectura de tres esquemas, es separar las aplicaciones de usuario de la base de datos física. En esta arquitectura, los esquemas pueden definirse en los tres niveles siguientes:

1. El nivel interno tiene un esquema interno, que describe la física estructura de almacenamiento de la base de datos. El esquema interno utiliza datos físicos modelo y describe los detalles completos de almacenamiento de datos y rutas de acceso para la base de datos.
2. El nivel conceptual tiene un esquema conceptual, que describe la estructura de toda la base de datos para una comunidad de usuarios. El esquema conceptual esconde los detalles de las estructuras de almacenamiento físico y se concentra en describir entidades, tipos de datos, relaciones, operaciones de usuario y restricciones. Por lo general, se utiliza un modelo de datos representacional para describir el esquema conceptual cuando está implementado el sistema de base de datos. Este esquema conceptual de implementación es a menudo se basa en un diseño de esquema conceptual en un modelo de datos de alto nivel.
3. El nivel externo o de vista incluye una serie de esquemas externos o puntos de vista. Cada esquema externo describe la parte de la base de datos que le interesa

a un grupo de usuarios en particular y oculta el resto de la base de datos de ese grupo de usuarios. Como en el nivel anterior, cada esquema externo se implementa típicamente utilizando un modelo de datos representacional, posiblemente basado en un modelo externo diseño de esquemas en un modelo de datos conceptual de alto nivel.



Algunos DBMS más antiguos pueden incluir detalles a nivel físico en el esquema conceptual. La arquitectura ANSI de tres niveles tiene un lugar importante en la base de datos desarrollo tecnológico porque separa claramente el nivel externo de los usuarios, el nivel conceptual de la base de datos y el nivel de almacenamiento interno para diseñar una base de datos. En la mayoría de los DBMS que apoyan las vistas del usuario, los esquemas externos se especifican en el mismo modelo de datos que describe la información a nivel conceptual (por ejemplo, un DBMS relacional como Oracle o SQLServer utilizan SQL para esto).

Observe que los tres esquemas son solo descripciones de datos; los datos reales se almacenan sólo a nivel físico. En la arquitectura de tres esquemas, cada grupo de usuarios se refiere a su propio esquema externo. Por lo tanto, el DBMS debe transformar una solicitud especificada en un esquema externo en una solicitud contra el esquema conceptual, y luego en una solicitud en el esquema interno para procesar sobre la base de datos almacenada. Si la solicitud es una recuperación de la base de datos, los datos extraídos de la base de datos almacenada deben ser reformateado para que coincida con la vista externa del usuario. Los procesos de transformación las solicitudes y los resultados entre niveles se denominan asignaciones. Estas asignaciones pueden ser requiere mucho tiempo, por lo que algunos DBMS, especialmente aquellos que están destinados a admitir pequeños bases de datos: no admiten vistas externas. Incluso en tales sistemas, sin embargo, es necesario transformar las solicitudes entre los niveles conceptual e interno.

### Independencia de datos

La arquitectura de tres esquemas se puede utilizar para explicar mejor el concepto de datos independencia, que se puede definir como la capacidad de cambiar el esquema en un nivel de un sistema de base de datos sin tener que cambiar el esquema en el siguiente nivel superior nivel. Podemos definir dos tipos de independencia de datos:

1. La independencia lógica de los datos es la capacidad de cambiar el esquema conceptual sin tener que cambiar esquemas externos o programas de aplicación. Podemos cambiar el esquema conceptual para expandir la base de datos (agregando un tipo de registro o elemento de datos), para cambiar las restricciones o para reducir la base de datos (eliminando un tipo de registro o elemento de datos). En el último caso, esquemas externos que se refieren solo a los datos restantes no deberían verse afectados. Después de la

El esquema conceptual se somete a una reorganización lógica, los programas de aplicación que hacen referencia a las construcciones del esquema externo deben funcionar como antes. Los cambios en las restricciones se pueden aplicar al esquema conceptual sin afectando los esquemas externos o programas de aplicación.

2. La independencia de los datos físicos es la capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual. Por tanto, tampoco es necesario cambiar los esquemas externos. Los cambios en el esquema interno pueden ser necesario porque se reorganizaron algunos archivos físicos, por ejemplo, creando estructuras de acceso adicionales, para mejorar el rendimiento de la recuperación o actualizar. Si los mismos datos que antes permanecen en la base de datos, no debemos hay que cambiar el esquema conceptual.

En general, la independencia de los datos físicos existe en la mayoría de las bases de datos y entornos de archivos donde los detalles físicos, como la ubicación exacta de los datos en el disco, y los detalles del hardware de la codificación, ubicación, compresión, división, fusión de registros, etc., están ocultos para el usuario. Por otro lado, la independencia lógica de los datos es más difícil de lograr porque permite cambios estructurales y de restricciones sin afectar los programas de aplicación, una requisito mucho más estricto.

Siempre que tengamos un DBMS de varios niveles, su catálogo debe ampliarse para incluir información sobre cómo mapear solicitudes y datos entre los distintos niveles. El DBMS utiliza software adicional para realizar estas asignaciones haciendo referencia a la asignación información en el catálogo. La independencia de los datos se produce porque cuando el esquema es cambiado en algún nivel, el esquema en el siguiente nivel superior permanece sin cambios; solamente se cambia el mapeo entre los dos niveles. Por tanto, no es necesario cambiar los programas de aplicación que se refieren al esquema de nivel superior.

## **Lenguajes e interfaces de bases de datos**

El DBMS debe proporcionar lenguajes e interfaces apropiados para cada categoría de usuarios.

### **Lenguajes DBMS**

Una vez que se completa el diseño de una base de datos y se elige un DBMS para implementar en la base de datos, el primer paso es especificar esquemas conceptuales e internos para la base de datos y cualquier mapeo entre los dos. En muchos DBMS donde no se mantiene una separación estricta de niveles, un lenguaje, llamado lenguaje de definición de datos (DDL), es utilizado por el DBA y por los diseñadores de bases de datos para definir ambos esquemas. Los DBMS tendrá un compilador DDL cuya función es procesar declaraciones DDL en para identificar descripciones de las construcciones del esquema y almacenar el esquema descripción en el catálogo DBMS. En los DBMS donde se mantiene una clara separación entre lo conceptual y lo niveles internos, el DDL se utiliza para especificar solo el esquema conceptual. Otro lenguaje, el lenguaje de definición de almacenamiento (SDL), se utiliza para especificar el esquema. Las

asignaciones entre los dos esquemas se pueden especificar en cualquiera de estos lenguajes. En la mayoría de los DBMS relacionales de hoy, no existe un lenguaje específico que realiza el papel de SDL. En cambio, el esquema interno se especifica mediante una combinación de funciones, parámetros y especificaciones relacionadas con el almacenamiento de archivos. Estos permiten el personal de DBA para controlar las opciones de indexación y la asignación de datos al almacenamiento. Por un verdadero arquitectura de tres esquemas, necesitaríamos un tercer lenguaje, la definición de vista lenguaje (VDL), para especificar las vistas del usuario y sus asignaciones al concepto esquema, pero en la mayoría de los DBMS, el DDL se utiliza para definir tanto conceptual como externo esquemas. En los DBMS relacionales, SQL se utiliza en el rol de VDL para definir usuarios o vistas de la aplicación como resultados de consultas predefinidas. Una vez que se compilan los esquemas de la base de datos y la base de datos se llena con datos, los usuarios deben tener algunos medios para manipular la base de datos. Manipulaciones típicas incluyen la recuperación, inserción, eliminación y modificación de los datos. El DBMS proporciona un conjunto de operaciones o un lenguaje llamado lenguaje de manipulación de datos (DML) para estos fines. En los DBMS actuales, los tipos de lenguajes anteriores normalmente no se consideran lenguajes distintos; en su lugar, se utiliza un lenguaje integral integral que incluye construcciones para la definición del esquema conceptual, la definición de la vista y la manipulación de datos. La definición de almacenamiento generalmente se mantiene separada, ya que se utiliza para definir estructuras de almacenamiento físico para ajustar el rendimiento del sistema de base de datos, que es generalmente realizado por el personal de DBA.

El SDL era un componente en las primeras versiones de SQL, pero se ha eliminado del lenguaje para mantenerlo en los niveles conceptuales y externos solamente. Hay dos tipos principales de DML. Un LMD de alto nivel o sin procedimiento puede ser utilizada por sí solo para especificar operaciones complejas de bases de datos de forma concisa. Muchos DBMS Permiten que las declaraciones DML de alto nivel se ingresen de forma interactiva desde una pantalla monitor o terminal o para integrarse en un lenguaje de programación de propósito general. En el último caso, las declaraciones DML deben identificarse dentro del programa para que puedan ser extraídas por un precompilador y procesadas por el DBMS. Un DML de bajo nivel o de procedimiento debe estar integrado en una programación de propósito general idioma. Este tipo de DML generalmente recupera registros u objetos individuales de la base de datos y procesa cada uno por separado. Por lo tanto, necesita usar programación construcciones de lenguaje, como bucles, para recuperar y procesar cada registro de un conjunto de registros. Los DML de bajo nivel también se denominan DML de registro a la vez debido a esto propiedad. Los DML de alto nivel, como SQL, pueden especificar y recuperar muchos registros en una sola declaración DML; por lo tanto, se les llama set-at-a-time o set-orientado DML. Una consulta en un DML de alto nivel a menudo especifica qué datos recuperar en lugar de que cómo recuperarlo; por lo tanto, estos lenguajes también se denominan declarativos. Siempre que los comandos DML, ya sean de alto o bajo nivel, están integrados en un Lenguaje de programación de propósito general, ese lenguaje se llama lenguaje anfitrión y el DML se denomina sublenguaje de datos.

Por otro lado, un alto nivel

El DML utilizado de forma interactiva independiente se denomina lenguaje de consulta. En general, tanto los comandos de recuperación como los de actualización de un DML de alto nivel se pueden usar de forma interactiva y, por lo tanto, se consideran parte del lenguaje de consulta. Los usuarios finales ocasionales suelen utilizar un lenguaje de consulta de alto nivel para especificar sus solicitudes, mientras que los programadores

usan el DML en su forma incrustada. Para usuarios ingenuos y paramétricos, generalmente existen interfaces fáciles de usar para interactuar con la base de datos; Estos también pueden ser utilizados por usuarios ocasionales u otros que no quieran aprender el detalles de un lenguaje de consulta de alto nivel. A continuación, discutimos estos tipos de interfaces.

## **Interfaces DBMS**

Las interfaces fáciles de usar proporcionadas por un DBMS pueden incluir lo siguiente:

### **Interfaces basadas en menús para clientes web o navegación.**

Estas interfaces presentan al usuario listas de opciones (llamadas menús) que guían al usuario a través de la formulación de una solicitud. Los menús eliminan la necesidad de memorizar los comandos y sintaxis de un lenguaje de consulta; más bien, la consulta se compone paso a paso seleccionando opciones de un menú que muestra el sistema. Derribar los menús son una técnica muy popular en las interfaces de usuario basadas en web. Ellos son también se utiliza a menudo en interfaces de navegación, que permiten al usuario mirar a través de los contenidos de una base de datos de forma exploratoria y no estructurada.

### **Aplicaciones para dispositivos móviles.**

Estas interfaces ofrecen a los usuarios móviles acceso a sus datos. Las aplicaciones tienen interfaces programadas integradas que normalmente permiten a los usuarios iniciar sesión con su nombre de cuenta y contraseña; las aplicaciones luego proporcionan un menú limitado de opciones para el acceso móvil a los datos del usuario, así como opciones como pagar facturas (para bancos) o hacer reservas (para sitios web de reservas).

### **Interfaces basadas en formularios.**

Una interfaz basada en formularios muestra un formulario a cada usuario. Los usuarios pueden completar todas las entradas del formulario para insertar nuevos datos, o pueden completar solo ciertas entradas, en cuyo caso el DBMS recuperará datos coincidentes para las entradas restantes. Los formularios generalmente se diseñan y programan para usuarios ingenuos como interfaces para transacciones enlatadas. Muchos DBMS tienen lenguajes de especificación de formularios, que son lenguajes especiales que ayudan a los programadores a especificar tales formas. Formularios SQL \* es un lenguaje basado en formularios que especifica consultas utilizando un formulario diseñado en conjunto con el esquema de base de datos relacional. Oracle Forms es un componente del paquete de productos Oracle que proporciona un amplio conjunto de funciones para diseñar y construir aplicaciones que utilizan formularios. Algunos sistemas tienen utilidades que definen una forma al permitir el usuario final construye de forma interactiva un formulario de muestra en la pantalla.

### **Interfaces gráficas de usuario.**

Una GUI generalmente muestra un esquema al usuario en forma de diagrama. Luego, el usuario puede especificar una consulta manipulando el diagrama. En muchos casos, las GUI utilizan tanto menús como formularios. Interfaces de lenguaje natural. Estas interfaces aceptan solicitudes escritas en inglés o en algún otro idioma e intentan comprenderlas.

## **Interfaces de lenguaje natural**

La interfaz generalmente tiene su propio esquema, que es similar a la base de datos conceptual esquema, así como un diccionario de palabras importantes. La interfaz de lenguaje natural se refiere a las palabras de su esquema, así como al conjunto de palabras estándar de su diccionario, que se utilizan para interpretar la solicitud. Si la interpretación tiene éxito, La interfaz genera una consulta de alto nivel correspondiente a la solicitud de lenguaje natural y lo envía al DBMS para su procesamiento; de lo contrario, se inicia un diálogo con el usuario para aclarar la solicitud.

### **Búsqueda de base de datos basada en palabras clave.**

Son algo similares a la búsqueda web motores, que aceptan cadenas de palabras en lenguaje natural (como inglés o español) y compararlos con documentos en sitios específicos (para motores de búsqueda locales) o en la Web. páginas en la Web en general (para motores como Google o Ask). Usan predefinidos indexa las palabras y utiliza funciones de clasificación para recuperar y presentar los documentos resultantes en un grado decreciente de coincidencia. Estas interfaces de consulta textual de "forma libre" son aún no es común en las bases de datos relacionales estructuradas, aunque un área de investigación llamada La consulta basada en palabras clave ha surgido recientemente para bases de datos relacionales.

**Entrada y salida de voz.** Uso limitado de voz como consulta de entrada y voz como respuesta a una pregunta o resultado de una solicitud se está convirtiendo en algo común. Aplicaciones con vocabularios limitados, como consultas de directorio telefónico, vuelo llegada / salida, y la información de la cuenta de la tarjeta de crédito, permiten el habla para entrada y salida para permitir a los clientes acceder a esta información. La entrada de voz se detecta utilizando una biblioteca de palabras predefinidas y se utiliza para configurar los parámetros que se suministran a las consultas. Para la salida, tiene lugar una conversión similar de texto o números a voz.

### **Interfaces para usuarios paramétricos.**

Los usuarios paramétricos, como los cajeros de banco, a menudo tienen un pequeño conjunto de operaciones que deben realizar repetidamente. Analistas de sistemas y los programadores diseñan e implementan una interfaz especial para cada clase conocida de usuarios ingenuos. Por lo general, se incluye un pequeño conjunto de comandos abreviados, con el objetivo de minimizar el número de pulsaciones de teclas necesarias para cada solicitud.

### **Interfaces para el DBA.**

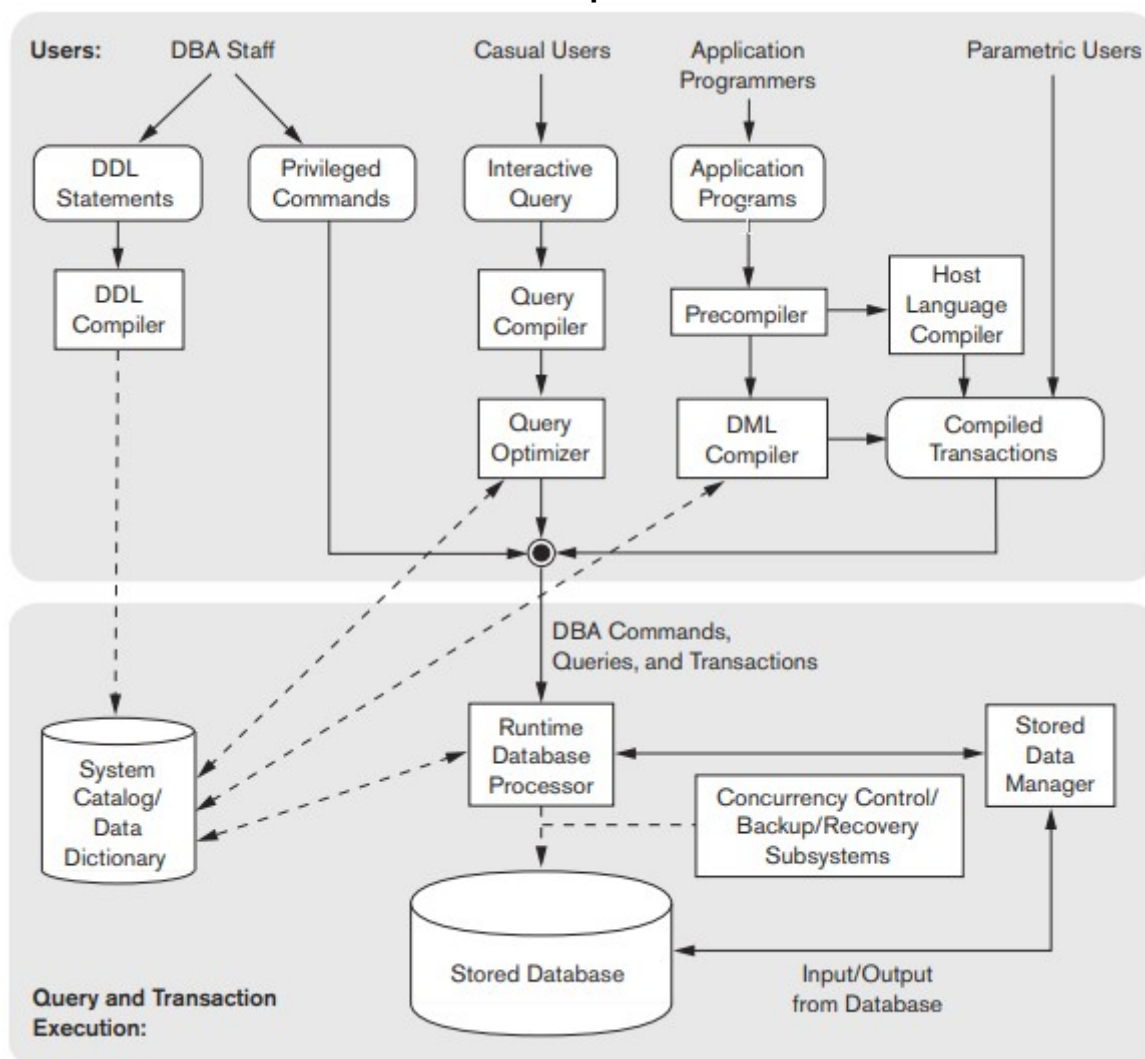
La mayoría de los sistemas de bases de datos contienen comandos privilegiados que solo puede utilizar el personal de DBA. Estos incluyen comandos para crear cuentas, configuración de parámetros del sistema, concesión de autorización de cuenta, cambio de esquema y reorganización de las estructuras de almacenamiento de una base de datos.

### **El entorno del sistema de base de datos**

Un DBMS es un sistema de software complejo. Los tipos de componentes de software que constituyen un DBMS y los tipos de software de sistemas informáticos con los que interactúa el DBMS son:



## Módulos de componentes DBMS



La parte superior de la figura se refiere a los distintos usuarios del entorno de la base de datos y sus interfaces. La parte inferior muestra los módulos internos del DBMS responsables del almacenamiento de datos y procesamiento de transacciones. La base de datos y el catálogo DBMS generalmente se almacenan en disco. Acceso al disco está controlado principalmente por el sistema operativo (SO), que programa el disco leer/escribir. Muchos DBMS tienen su propio módulo de administración de búfer para programar la lectura/escritura del disco, porque la administración del almacenamiento de búfer tiene un considerable efecto sobre el rendimiento. Reducir la lectura/escritura del disco mejora considerablemente el rendimiento. Un módulo de administrador de datos almacenados de nivel superior del DBMS controla el acceso a la información DBMS que se almacena en el disco, ya sea que sea parte de la base de datos o el catálogo.

Primero la parte superior muestra interfaces para el personal de DBA, usuarios ocasionales que trabajan con interfaces interactivas para formular consultas, aplicaciones programadores que crean programas utilizando algunos lenguajes de programación host, y los usuarios paramétricos que realizan la entrada de datos funcionando proporcionando parámetros a predefinidos. El personal de DBA trabaja en la definición de la base de datos y la ajusta haciendo cambios en su definición utilizando DDL y otros comandos privilegiados. El compilador DDL procesa las definiciones de esquemas, especificadas en el DDL, y almacena descripciones de los esquemas (metadatos) en el catálogo DBMS. El catálogo incluye información como los nombres y tamaños de archivos, nombres y tipos de datos de elementos de datos, detalles de almacenamiento de cada archivo,

información de mapeo entre esquemas y restricciones.

Usuarios ocasionales y personas con necesidad ocasional de información de la base de datos interactuar usando la interfaz de consulta interactiva, se muestran las interacciones móviles o basadas en menús o formularios que se utilizan normalmente para generar la consulta interactiva automáticamente o para acceder a transacciones enlatadas. Estas consultas se analizan y validan para verificar que la sintaxis de la consulta sea correcta, nombres de archivos y elementos de datos, y así sucesivamente mediante un compilador de consultas que los compila en un formulario interno. Esta consulta interna está sujeta a optimización de consultas. Entre otras cosas, el optimizador de consultas se ocupa de la reordenación y posible reordenación de operaciones, eliminación de redundancias y uso de algoritmos de búsqueda eficientes durante la ejecución. Eso consulta el catálogo del sistema para obtener información estadística y física sobre el almacena datos y genera código ejecutable que realiza las operaciones necesarias para la consulta y realiza llamadas en el procesador en tiempo de ejecución.

Los programadores de aplicaciones escriben programas en lenguajes host como Java, C o C ++

que se envían a un precompilador. El precompilador extrae comandos DML desde un programa de aplicación escrito en un lenguaje de programación anfitrión. Estos comandos se envían al compilador DML para su compilación en código objeto para la base de datos acceso. El resto del programa se envía al compilador del lenguaje anfitrión. El objeto

Los códigos para los comandos DML y el resto del programa están vinculados, formando un transacción enlatada cuyo código ejecutable incluye llamadas a la base de datos en tiempo de ejecución procesador. También es cada vez más común utilizar lenguajes de programación como como PHP y Python para escribir programas de base de datos. Las transacciones enlatadas se ejecutan repetidamente por usuarios paramétricos a través de PC o aplicaciones móviles; estos usuarios simplemente proporcionan los parámetros de las transacciones. Cada ejecución se considera una transacción. Un ejemplo es una transacción de pago bancaria donde el número de cuenta, beneficiario, y la cantidad se pueden proporcionar como parámetros.

En la parte inferior el procesador de base de datos en tiempo de ejecución ejecuta (1) los comandos privilegiados, (2) los planes de consulta ejecutables y (3) las transacciones enlatadas con parámetros de tiempo de ejecución. Funciona con el catálogo del sistema y puede actualizarlo. con estadísticas. También funciona con el administrador de datos almacenados, que a su vez utiliza servicios del sistema operativo para realizar entradas / salidas de bajo nivel (lectura / escritura) operaciones entre el disco y la memoria principal. El procesador de base de datos en tiempo de ejecución maneja otros aspectos de la transferencia de datos, como la gestión de búferes en la memoria. Algunos DBMS tienen su propio módulo de gestión de búfer, mientras que otros dependen del sistema operativo para la gestión del búfer. Hemos mostrado control de concurrencia y los sistemas de respaldo y recuperación por separado como un módulo en esta figura. Son integrado en el funcionamiento del procesador de base de datos en tiempo de ejecución para fines de Gestión de transacciones. Es común que el programa cliente que accede al DBMS se ejecute en un equipo o dispositivo separado del equipo en el que reside la base de datos. El primero se denomina computadora cliente que ejecuta el software cliente DBMS y el segundo es llamado servidor de base de datos. En muchos casos, el cliente accede a una computadora intermedia, llamado servidor de aplicaciones, que a su vez accede al servidor de la base de datos. Por otro lado, si el sistema informático es principalmente dedicado a ejecutar el servidor de la base de datos, el DBMS controlará la memoria

principal de las páginas del disco. El DBMS también interactúa con compiladores para lenguajes de programación de host de uso general y con servidores de aplicaciones y programas cliente que se ejecutan en máquinas separadas a través de la interfaz de red del sistema.

### **Utilidades del sistema de base de datos**

Además de poseer los módulos de software que se acaban de describir, la mayoría de los DBMS tienen utilidades de base de datos que ayudan al DBA a administrar el sistema de base de datos. Las utilidades comunes tienen los siguientes tipos de funciones:

- **Cargando.** Se utiliza una utilidad de carga para cargar archivos de datos existentes, como texto archivos o archivos secuenciales, en la base de datos. Por lo general, el formato actual (origen) del archivo de datos y la estructura del archivo de la base de datos deseada (destino) se especifican en la utilidad, que luego reformatea automáticamente los datos y los almacena en la base de datos. Con la proliferación de DBMS, la transferencia de datos desde un DBMS a otro se está volviendo común en muchas organizaciones. Algunos de los proveedores ofrecen herramientas de conversión que generan los programas de carga adecuados, dadas las descripciones de almacenamiento de la base de datos de origen y destino existente. (esquemas internos).
- **Copia de seguridad.** Una utilidad de respaldo crea una copia de respaldo de la base de datos, generalmente por volcar toda la base de datos en cinta u otro medio de almacenamiento masivo. La copia de seguridad se puede utilizar para restaurar la base de datos en caso de un disco catastrófico. fracaso. Las copias de seguridad incrementales también se utilizan a menudo, donde solo cambian desde se registra la copia de seguridad anterior. La copia de seguridad incremental es más compleja, pero ahorra espacio de almacenamiento.
- **Reorganización del almacenamiento de la base de datos.** Esta utilidad se puede utilizar para reorganizar un conjunto de archivos de base de datos en diferentes organizaciones de archivos y crear un nuevo acceso a caminos para mejorar el rendimiento.
- **Supervisión del desempeño.** Dicha utilidad monitorea el uso de la base de datos y proporciona estadísticas al DBA. El DBA utiliza las estadísticas para tomar decisiones como si reorganizar o no archivos o si agregar o quitar índices para mejorar el rendimiento.

Otras utilidades pueden estar disponibles para ordenar archivos, manejar la compresión de datos, monitorear el acceso de los usuarios, interactuar con la red y realizar otras funciones.

### **Herramientas, entornos de aplicación, e instalaciones de comunicaciones**

Otras herramientas suelen estar disponibles para los diseñadores de bases de datos, los usuarios y el DBMS CASE Tools se utilizan en la fase de diseño de los sistemas de bases de datos. Otra herramienta que puede ser

bastante útil en las grandes organizaciones es un sistema de diccionario de datos ampliado (o repositorio de datos). Además de almacenar información de catálogo sobre esquemas y restricciones, el diccionario de datos almacena otra información, como decisiones de diseño, estándares de uso, descripciones de programas de aplicación e información del usuario. Tal sistema es también llamado depósito de información. Se puede acceder a esta información directamente por usuarios o el DBA cuando sea necesario. Una utilidad de diccionario de datos es similar al DBMS catálogo, pero incluye una variedad más amplia de información y se accede principalmente por usuarios en lugar del software DBMS. Entornos de desarrollo de aplicaciones, como PowerBuilder (Sybase)

o JBuilder (Borland), han sido bastante populares. Estos sistemas proporcionan un entorno para desarrollar aplicaciones de bases de datos e incluyen instalaciones que ayudan en muchas facetas de los sistemas de bases de datos, incluido el diseño de bases de datos, el desarrollo de GUI, las consultas y actualización y desarrollo de programas de aplicación. El DBMS también necesita interactuar con el software de comunicaciones, cuya función es permitir que los usuarios en ubicaciones remotas del sitio del sistema de base de datos accedan a la base de datos a través de terminales de computadora, estaciones de trabajo o computadoras personales. Estas están conectados al sitio de la base de datos a través de hardware de comunicaciones de datos como enrutadores de Internet, líneas telefónicas, redes de larga distancia, redes locales o dispositivos de comunicación por satélite. Muchos sistemas de bases de datos comerciales tienen comunicación paquetes que funcionan con el DBMS. El sistema integrado de comunicaciones de datos y DBMS se denomina sistema DB / DC. Además, algunos DBMS distribuidos son distribuidos físicamente en varias máquinas. En este caso, se necesitan redes de comunicaciones para conectar las máquinas. Suelen ser redes de área local (LAN), pero también pueden ser otros tipos de redes.

## **Centralizado y Cliente / Servidor Arquitecturas para DBMS**

### **Arquitectura de DBMS centralizada**

Las arquitecturas de los DBMS han seguido tendencias similares a las de las arquitecturas de sistemas informáticos generales. Las arquitecturas más antiguas usaban computadoras centrales para proporcionar el procesamiento principal de todas las funciones del sistema, incluida la aplicación del usuario programas y programas de interfaz de usuario, así como toda la funcionalidad DBMS. La razón era que en los sistemas más antiguos, la mayoría de los usuarios accedían al DBMS a través de terminales de computadora que no tenían capacidad de procesamiento y solo proporcionaban capacidades de visualización. Por lo tanto, todo el procesamiento se realizó de forma remota en la carcasa del sistema informático el DBMS, y solo la información de visualización y los controles se enviaron desde la computadora a los terminales de pantalla, que estaban conectados al ordenador central a través de varios tipos de redes de comunicaciones. Al principio, los sistemas de bases de datos utilizaron estas computadoras de manera similar a como habían usado terminales de pantalla, de modo que el propio DBMS seguía siendo un DBMS centralizado en el que toda la funcionalidad del DBMS, la ejecución del programa de aplicación y el procesamiento de la interfaz de usuario se llevaban a cabo en una máquina. Gradualmente, los sistemas DBMS comenzaron a explotar la potencia de procesamiento disponible en el lado del usuario, lo que condujo a arquitecturas DBMS cliente / servidor.

### **Arquitecturas básicas de cliente / servidor**

Primero, discutimos la arquitectura cliente / servidor en general; luego discutimos como es aplicado a los DBMS. La arquitectura cliente / servidor se desarrolló para hacer frente a entornos informáticos en los que una gran cantidad de PC, estaciones de trabajo, servidores de archivos, impresoras, servidores de bases de datos, servidores web, servidores de correo electrónico y otro software y los equipos están conectados a través de una red. La idea es definir servidores especializados con funcionalidades específicas.

Los servidores web o de correo electrónico también entran en la categoría de servidores especializados. Muchos clientes pueden acceder a los recursos proporcionados por servidores especializados máquinas. Las máquinas cliente proporcionan al usuario las

interfaces adecuadas para utilizar estos servidores, así como con potencia de procesamiento local para ejecutar aplicaciones locales.

Este concepto puede trasladarse a otros paquetes de software, con programas especializados, como un paquete CAD (diseño asistido por computadora), que se almacenan en máquinas servidor y ser accesible para múltiples clientes.

### **Arquitectura cliente / servidor a nivel lógico**

Algunas máquinas serían solo sitios de clientes (por ejemplo, dispositivos móviles o estaciones de trabajo / PC que solo tienen software de cliente instalado). Otras máquinas serían servidores dedicados y otras tendrían ambos funcionalidad de cliente y servidor. El concepto de arquitectura cliente / servidor asume un marco subyacente que consta de muchas PC / estaciones de trabajo y dispositivos móviles, así como un número menor de máquinas servidor, conectadas a través de redes inalámbricas o LAN y otros tipos de Red de computadoras. Un cliente en este marco suele ser una máquina de usuario que proporciona capacidades de interfaz de usuario y procesamiento local. Cuando un cliente requiere acceso a la funcionalidad adicional, como el acceso a la base de datos, que no existe en el cliente, se conecta a un servidor que proporciona la funcionalidad necesaria. Un servidor es un sistema que contiene tanto hardware como software que puede proporcionar servicios al cliente. máquinas, como acceso a archivos, impresión, archivo o acceso a bases de datos. En general, algunas máquinas instalan solo software de cliente, otras solo software de servidor, y aún así otros pueden incluir software tanto de cliente como de servidor, sin embargo, es más común que el software de cliente y servidor se ejecute normalmente en máquinas independientes. Se crearon dos tipos principales de arquitecturas DBMS básicas en este marco cliente / servidor subyacente: dos niveles y tres niveles.

### **Arquitecturas de cliente / servidor de dos niveles para DBMS**

En los sistemas de administración de bases de datos relacionales (RDBMS), muchos de los cuales comenzaron como sistemas centralizados, los componentes del sistema que se movieron primero al del lado del cliente eran la interfaz de usuario y los programas de aplicación. Porque SQL proporcionaron un lenguaje estándar para RDBMS, esto creó un punto lógico de división entre cliente y servidor. Por lo tanto, la funcionalidad de consulta y transacción relacionada con el procesamiento de SQL permaneció en el lado del servidor. En tal arquitectura, el servidor a menudo se denomina servidor de consultas o transacción servidor porque proporciona estas dos funcionalidades. En un RDBMS, el servidor es también llamado servidor SQL. Los programas de la interfaz de usuario y los programas de aplicación pueden ejecutarse en el lado del cliente.

Cuando se requiere acceso a DBMS, el programa establece una conexión al DBMS (que está en el lado del servidor); una vez que se crea la conexión, el cliente El programa puede comunicarse con el DBMS. Un estándar llamado Open Database La conectividad (ODBC) proporciona una interfaz de programación de aplicaciones (API), que permite que los programas del lado del cliente llamen al DBMS, siempre que tanto el cliente como las máquinas servidor tienen instalado el software necesario. La mayoría de los proveedores de DBMS proporcionan controladores ODBC para sus sistemas. Un programa cliente realmente puede conectarse a varios RDBMS y enviar solicitudes de consulta y transacción utilizando la API ODBC, que luego se procesan en los sitios del servidor. Los resultados de la consulta se envían de nuevo al programa cliente, que puede procesar y mostrar los resultados según sea necesario. Un relacionado También se ha definido el estándar para el lenguaje de programación Java, llamado JDBC. Esto permite que los programas cliente de Java accedan a uno o más DBMS a través de una interfaz estándar. Las arquitecturas que se describen aquí se denominan arquitecturas de dos niveles porque los componentes de software se distribuyen en dos sistemas: cliente y

servidor. Las ventajas de esta arquitectura son su simplicidad y compatibilidad perfecta con los sistemas. La aparición de la Web cambió los roles de clientes y servidores, liderando a la arquitectura de tres niveles.

### **Arquitecturas de tres y n niveles para aplicaciones web**

Muchas aplicaciones web utilizan una arquitectura llamada arquitectura de tres niveles, que agrega una capa intermedia entre el cliente y el servidor de la base de datos. Esta capa intermedia o nivel medio se denomina servidor de aplicaciones o Web. servidor, dependiendo de la aplicación. Este servidor juega un papel intermediario al ejecutar programas de aplicación y almacenar reglas comerciales (procedimientos o restricciones) que se utilizan para acceder a los datos del servidor de la base de datos. También puede mejorar seguridad de la base de La capa de presentación muestra información al usuario y permite entrada de datos. La capa de lógica empresarial maneja las reglas y restricciones intermedias antes los datos se pasan al usuario o al DBMS. La capa inferior incluye todos servicios de gestión de datos. La capa intermedia también puede actuar como servidor web, que recupera los resultados de la consulta del servidor de la base de datos y los formatea en dinámicos Páginas web que el navegador web visualiza en el lado del cliente. datos comprobando las credenciales de un cliente antes de enviar una solicitud al servidor de la base de datos. Los clientes contienen interfaces de usuario y navegadores web. El servidor intermedio acepta solicitudes del cliente, procesa la solicitud y envía consultas y comandos de la base de datos al servidor de la base de datos, y luego actúa como un conducto para pasar (parcialmente) datos procesados desde el servidor de la base de datos a los clientes, donde se puede procesar más y filtrar para presentarlo a los usuarios. Así, el usuario la interfaz, las reglas de la aplicación y el acceso a los datos actúan como los tres niveles. También se han propuesto otras arquitecturas. Es posible dividir las capas entre el usuario y los datos almacenados en componentes más finos, dando así se elevan a arquitecturas de n niveles, donde n puede ser de cuatro o cinco niveles. Normalmente, el negocio La capa lógica se divide en varias capas. Además de distribuir programación y datos a través de una red, las aplicaciones de n niveles ofrecen la ventaja de que cualquier nivel puede ejecutarse en un procesador apropiado o plataforma de sistema operativo y puede ser manejado de forma independiente. Proveedores de ERP (planificación de recursos empresariales) y CRM (gestión de relaciones con el cliente) los paquetes a menudo utilizan una capa de middleware, que da cuenta de los módulos de front-end (clientes) que se comunican con varios bases de datos back-end (servidores). Los avances en la tecnología de cifrado y descifrado hacen que sea más seguro transferir datos confidenciales del servidor al cliente en forma cifrada, donde se descifrarán. lo último se puede hacer por hardware o por software avanzado. Varias tecnologías para la compresión de datos también ayudan a transferir grandes cantidades de datos de servidores a clientes a través de redes cableadas e inalámbricas.

### **Clasificación de la base de datos y Sistemas de gestión**

Se pueden utilizar varios criterios para clasificar los DBMS. El primero es el modelo de datos en el que se basa el DBMS. El modelo de datos principal utilizado en muchos comerciales actuales DBMS es el modelo de datos relacionales, y los sistemas basados en este modelo son conocidos como sistemas SQL. El modelo de datos de objetos se ha implementado en algunos sistemas comerciales pero no ha tenido un uso generalizado. Recientemente, los llamados big data Los sistemas, también conocidos como sistemas de almacenamiento de clave-valor y sistemas NOSQL, utilizan varios modelos de datos: basados en documentos, basados en gráficos, basados en columnas y clave-valor. modelos de datos. Muchas aplicaciones heredadas todavía se ejecutan en sistemas de bases de datos basados en modelos de datos jerárquicos y de red.

Los DBMS relacionales evolucionan continuamente y, en particular, se han incorporando muchos de los conceptos que se desarrollaron en bases de datos de objetos. Esta ha dado lugar a una nueva clase de DBMS denominados DBMS relacionales de objetos. Podemos categorizar los DBMS en función del modelo de datos: relacional, objeto, objeto-relacional, NOSQL, valor-clave, jerárquico, de red y otros. Algunos DBMS experimentales se basan en XML (eXtended Markup Language) modelo, que es un modelo de datos estructurado en árbol. Estos se han llamado XML nativo DBMS. Varios DBMS relacionales comerciales han agregado interfaces XML y almacenamiento a sus productos. El segundo criterio utilizado para clasificar los DBMS es el número de usuarios admitidos por el sistema. Los sistemas de un solo usuario admiten solo un usuario a la vez y en su mayoría son utilizados con PC. Los sistemas multiusuario, que incluyen la mayoría de DBMS, admiten múltiples usuarios concurrentes. El tercer criterio es el número de sitios en los que se distribuye la base de datos. Un DBMS está centralizado si los datos se almacenan en un solo sitio de computadora. Centralizado DBMS puede soportar múltiples usuarios, pero el DBMS y la base de datos residen totalmente en un solo sitio informático. Un DBMS distribuido (DDBMS) puede tener la base de datos real y software DBMS distribuido en muchos sitios conectados por una red informática. Los grandes sistemas de datos a menudo se distribuyen masivamente, con cientos de sitios. Los datos son a menudo se replica en varios sitios para que la falla de un sitio no genere algunos datos indisponibles.

Los DDBMS homogéneos utilizan el mismo software DBMS en todos los sitios, mientras que Los DDBMS heterogéneos pueden utilizar diferentes software DBMS en cada sitio. También es posible desarrollar software de middleware para acceder a varias preexistentes autónomas bases de datos almacenadas en DBMS heterogéneos. Esto conduce a un DBMS federado (o sistema de base de datos múltiple), en el que los DBMS participantes están débilmente acoplados y tener cierto grado de autonomía local. Muchos DDBMS utilizan la arquitectura cliente-servidor. El cuarto criterio es el costo. Es difícil proponer una clasificación de DBMS basado en el costo. Hoy tenemos productos DBMS de código abierto (gratuitos) como MySQL y PostgreSQL que son compatibles con proveedores externos con servicios adicionales. Los principales productos RDBMS están disponibles como versiones de copia de 30 días de examen gratuito así como versiones personales, que pueden costar menos de \$ 100 y permitir una buena cantidad de funcionalidad. Los sistemas gigantes se venden en forma modular con componentes para manejar la distribución, la replicación, el procesamiento paralelo, la capacidad móvil, etc. y con una gran cantidad de parámetros que se deben definir para la configuración. Además, se venden en forma de licencias: las licencias de sitio permiten el uso ilimitado del sistema de base de datos con cualquier número de copias ejecutándose en el cliente sitio. Otro tipo de licencia limita el número de usuarios concurrentes o el número de asientos de usuario en una ubicación. Versiones independientes para un solo usuario de algunos sistemas como Microsoft Access se vende por copia o se incluye en la configuración general de un computadora de escritorio o portátil. Además, las funciones de almacenamiento de datos y minería, así como soporte para tipos de datos adicionales, están disponibles a un costo adicional. Es posible pagar anualmente millones de dólares por la instalación y el mantenimiento de grandes sistemas de bases de datos. También podemos clasificar un DBMS en función de los tipos de opciones de ruta de acceso para almacenar archivos. Una familia conocida de DBMS se basa en estructuras de archivos invertidas. Finalmente, un DBMS puede ser de propósito general o especial. Cuando el rendimiento es una consideración primordial, un DBMS de propósito especial se puede diseñar y construir para una aplicación específica; tal sistema no se puede utilizar para otras aplicaciones sin cambios principales. Muchas reservas de aerolíneas y sistemas de

directorios telefónicos desarrollados en el pasado son DBMS para propósitos especiales. Estos entran en la categoría de online sistemas de procesamiento de transacciones (OLTP), que deben admitir una gran cantidad de transacciones concurrentes sin imponer demoras excesivas. Analicemos brevemente el criterio principal para clasificar los DBMS: los datos modelo. El modelo de datos relacionales representa una base de datos como una colección de tablas, donde cada tabla se puede almacenar como un archivo separado. La mayoría de las bases de datos relacionales utilizan el lenguaje de consulta llamado SQL y admite una forma limitada de vistas de usuario. El modelo de datos de objetos define una base de datos en términos de objetos, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías (o gráficos acíclicos). Las operaciones de cada clase se especifican en términos de procedimientos predefinidos llamados métodos. Los DBMS relacionales han ampliado sus modelos para incorporar bases de datos de objetos, conceptos y otras capacidades; estos sistemas se conocen como relacionales de objetos o sistemas relacionales extendidos.

Los sistemas de big data se basan en varios modelos de datos, con los siguientes cuatro datos modelos más comunes. El modelo de datos clave-valor asocia una clave única con cada valor (que puede ser un registro u objeto) y proporciona un acceso muy rápido a un valor dado su clave. El modelo de datos del documento se basa en JSON (Java Script Notación de objetos) y almacena los datos como documentos, que se parecen un poco objetos complejos. El modelo de datos de gráficos almacena objetos como nodos de gráficos y las relaciones entre objetos como bordes de gráficos dirigidos. Finalmente, los datos basados en columnas Los modelos almacenan las columnas de filas agrupadas en páginas de disco para un acceso rápido y permitir múltiples versiones de los datos. El modelo XML ha surgido como un estándar para el intercambio de datos en la Web y se ha utilizado como base para implementar varios prototipos de sistemas XML nativos. XML utiliza estructuras de árbol jerárquicas. Combina conceptos de bases de datos con conceptos a partir de modelos de representación de documentos. Los datos se representan como elementos; con el uso de etiquetas, los datos se pueden anidar para crear estructuras de árbol complejas. Este modelo se parece conceptualmente al modelo de objetos, pero utiliza una terminología diferente. Se han agregado capacidades XML a muchos productos DBMS comerciales. modelos de datos importantes, ahora conocidos como modelos de datos heredados, son la red y los modelos jerárquicos. El modelo de red representa los datos como tipos de registro y también representa un tipo limitado de relación 1: N, llamado tipo de conjunto. Una relación 1: N, o uno a muchos, relaciona una instancia de un registro con muchos registros instancias que utilizan algún mecanismo de enlace de puntero en estos modelos. El modelo, también conocido como modelo CODASYL DBTG, 14 tiene un lenguaje de registro en tiempo asociado que debe estar integrado en un lenguaje de programación host. El DML de red se propuso en el Informe del Grupo de tareas de base de datos (DBTG) de 1971 como extensión del lenguaje COBOL. El modelo jerárquico representa los datos como estructuras de árbol jerárquicas. Cada jerarquía representa una serie de registros relacionados. No existe un idioma estándar para modelo jerárquico. Un DML jerárquico popular es DL / 1 del sistema IMS. Dominó el mercado de DBMS durante más de 20 años entre 1965 y 1985. Su DML, llamado DL / 1, fue un estándar industrial de facto durante mucho tiempo.

## Resumen

Distinguimos tres categorías principales un modelo de datos:

- Modelos de datos conceptuales o de alto nivel (basados en entidades y relaciones)
- Modelos de datos físicos o de bajo nivel



- Modelos de datos representativos o de implementación (basados en registros, orientados a objetos)

Distinguimos el esquema, o descripción de una base de datos, de la propia base de datos. El esquema no cambia muy a menudo, mientras que el estado de la base de datos cambia cada hora en que se insertan, eliminan o modifican los datos. Luego describimos el esquema de tres Arquitectura DBMS, que permite tres niveles de esquema:

- Un esquema interno describe la estructura de almacenamiento físico de la base de datos.
- Un esquema conceptual es una descripción de alto nivel de toda la base de datos.
- Los esquemas externos describen las vistas de diferentes grupos de usuarios.

Un DBMS que separa limpiamente los tres niveles debe tener asignaciones entre los esquemas para transformar las solicitudes y los resultados de las consultas de un siguiente nivel. La mayoría de los DBMS no separan completamente los tres niveles. Usamos el Arquitectura de tres esquemas para definir los conceptos de datos lógicos y físicos independencia. Luego discutimos los principales tipos de lenguajes e interfaces que admiten los DBMS. Se utiliza un lenguaje de definición de datos (DDL) para definir el esquema conceptual de la base de datos. En la mayoría de los DBMS, el DDL también define las vistas del usuario y, a veces, las estructuras de almacenamiento; en otros DBMS, existen lenguajes o funciones separados para especificar el almacenamiento estructuras. Esta distinción se está desvaneciendo en las implementaciones relacionales de hoy, con SQL sirviendo como un lenguaje general para realizar múltiples funciones, incluida la vista definición. La parte de definición de almacenamiento (SDL) se incluyó en las primeras versiones de SQL, pero ahora normalmente se implementa como comandos especiales para el DBA en relaciones DBMS. El DBMS compila todas las definiciones de esquema y almacena sus descripciones en el catálogo DBMS. Se utiliza un lenguaje de manipulación de datos (DML) para especificar recuperaciones de bases de datos y actualizaciones. Los DML pueden ser de alto nivel (orientados a conjuntos, no procedimentales) o de bajo nivel (orientados a registros, procedimentales). Se puede incrustar un DML de alto nivel en una programación de host idioma, o puede usarse como un idioma independiente; en el último caso es a menudo llamado lenguaje de consulta. Discutimos diferentes tipos de interfaces proporcionados por DBMS y los tipos de Usuarios de DBMS a los que se asocia cada interfaz. Luego discutimos el entorno del sistema de base de datos, módulos de software DBMS típicos y DBMS utilidades para ayudar a los usuarios y al personal de DBA a realizar sus tareas. Continuamos con una descripción general de las arquitecturas de dos y tres niveles para la base de datos aplicaciones.

Finalmente, clasificamos los DBMS de acuerdo con varios criterios: modelo de datos, número de usuarios, número de sitios, tipos de rutas de acceso y costo. Discutimos la disponibilidad de DBMS y módulos adicionales, sin costo en forma de código abierto software a configuraciones cuyo mantenimiento cuesta anualmente millones. También señaló la variedad de acuerdos de licencia para DBMS y productos relacionados. La clasificación principal de los DBMS se basa en el modelo de datos. Brevemente discutió los principales modelos de datos utilizados en los DBMS comerciales actuales.