

Bases de Datos I

16/09/2020

Resumen Cap 1 16/09

IS-501

José Inestroza

Introducción

Una base de datos es una colección de datos relacionados. 1 Por datos, nos referimos a hechos conocidos que pueden

ser registrados y que tengan un significado implícito.

Esta colección de datos relacionados con un significado implícito es una base de datos.

Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algún aspecto del mundo real, a veces llamado minimundo o el universo del discurso (UoD). Cambios en el mini mundo se reflejan en la base de datos.

- Una base de datos es una colección de datos lógicamente coherente con algunos sentido. Una variedad aleatoria de datos no se puede denominar correctamente base de datos.

- Una base de datos está diseñada, construida y poblada con datos para un propósito específico.

Tiene un grupo previsto de usuarios y algunas aplicaciones preconcebidas en que estos usuarios están interesados.

En otras palabras, una base de datos tiene alguna fuente de la que se derivan los datos, en cierto grado de interacción con eventos en el mundo real, y una audiencia que está activamente interesada en sus contenidos. Los usuarios finales de una base de datos pueden realizar transacciones comerciales o pueden ocurrir eventos que provocan cambios en la información de la base de datos.

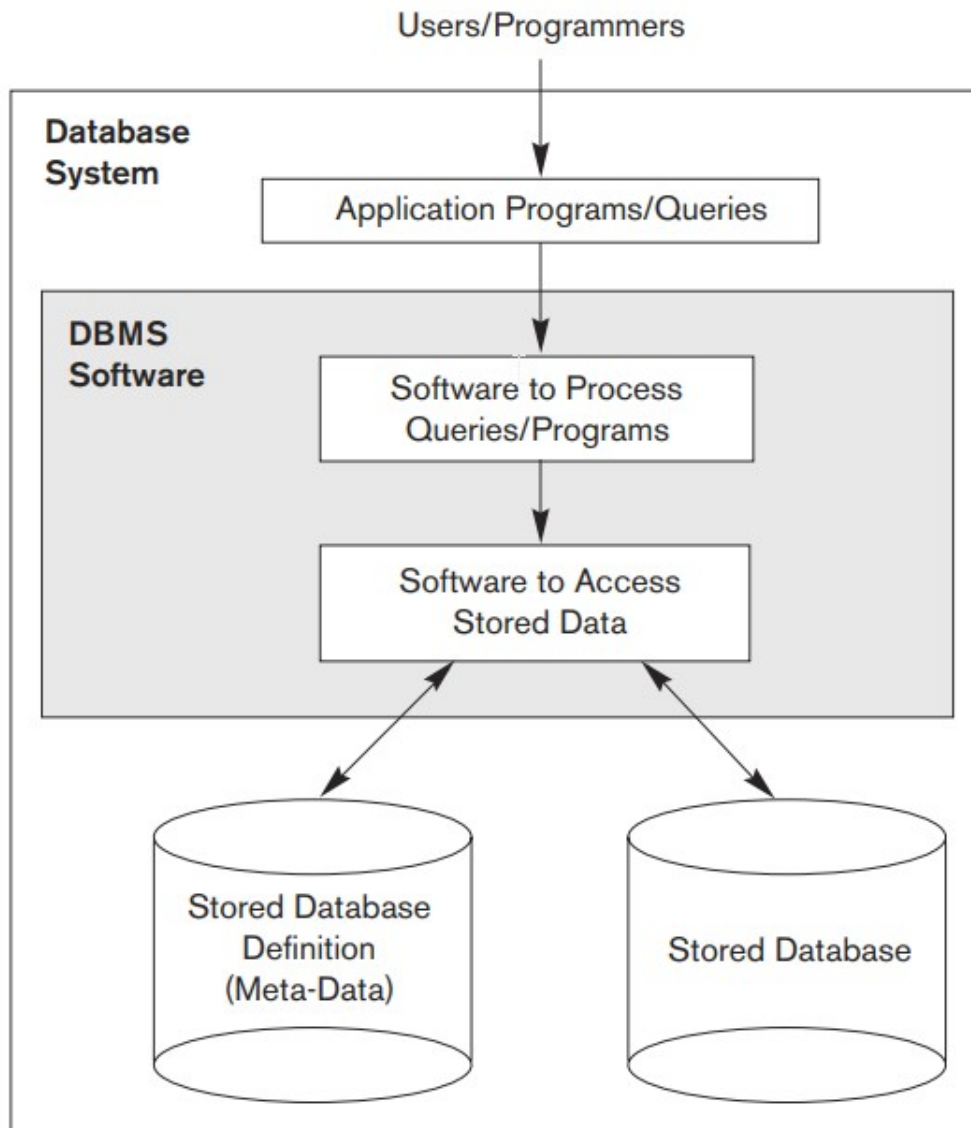
Una base de datos puede generarse y mantenerse manualmente o puede estar informatizada.

Se puede crear y mantener una base de datos computarizada ya sea por un grupo de programas de aplicación escritos específicamente para esa tarea o por un sistema de administración de base de datos. Un sistema de administración de bases de datos (DBMS) es un sistema computarizado que permite

Usuarios para crear y mantener una base de datos. El DBMS es un software de propósito general sistema que facilita los procesos de definición, construcción, manipulación y compartir bases de datos entre varios usuarios y aplicaciones. Definiendo una base de datos implica especificar los tipos de datos, estructuras y restricciones de los datos que se almacenado en la base de datos. La definición de la base de datos o la información descriptiva también almacenado por el DBMS en forma de catálogo de base de datos o diccionario; se llama metadatos.

La construcción de la base de datos es el proceso de almacenar los datos en algunos medio de almacenamiento controlado por el DBMS. Manipular una base de datos incluye funciones tales como consultar la base de datos para recuperar datos específicos, actualizar la base de datos para reflejar los cambios en el mini mundo y generar informes a partir de los datos. Compartir una base de datos permite que varios usuarios y programas accedan a la base de datos simultáneamente. Un programa de aplicación accede a la base de datos enviando consultas o solicitudes de datos al DBMS. Una consulta normalmente hace que se recuperen algunos datos; una transacción puede hacer que se lean algunos datos y que se escriban algunos datos en la base de datos.

Otras funciones importantes proporcionadas por el DBMS incluyen proteger la base de datos y mantenerlo durante un largo período de tiempo. Es posible escribir un conjunto personalizado de programas para crear y mantener la base de datos, creando de hecho un software DBMS de propósito especial para una aplicación específica, como reservas de aerolíneas. Para completar nuestras definiciones iniciales, llamaremos a la base de datos y al software DBMS juntos un sistema de base de datos.



Estos requisitos se documentan en detalle y se transforman en un diseño conceptual que se puede representar y manipular utilizando algunas herramientas informáticas para que pueda ser fácilmente mantenido, modificado y transformado en una implementación de la base de datos. Luego, el diseño se traduce a un diseño lógico que puede expresarse en un modelo de datos implementado en un DBMS comercial. La etapa final es el diseño físico, durante el cual se proporcionan especificaciones adicionales para almacenar y acceder a la base de datos. El diseño de la base de datos está implementado, poblado con datos reales y mantenidos continuamente para reflejar el estado del mini mundo.

Características del enfoque de base de datos

En procesamiento de archivos tradicional, cada usuario define e implementa los archivos necesarios para una aplicación de software específica como parte de la programación de la aplicación.

En el enfoque de la base de datos, un único repositorio mantiene los datos que se definen una vez y luego varios usuarios acceden repetidamente a través de consultas, transacciones y programas de aplicación. Las principales características del enfoque de base de datos frente a El enfoque de procesamiento de archivos es el siguiente:

- Naturaleza autodescriptiva de un sistema de base de datos
- Aislamiento entre programas y datos, y abstracción de datos
- Soporte de múltiples vistas de los datos
- Intercambio de datos y procesamiento de transacciones multiusuario

Describimos cada una de estas características en una sección separada.

Naturaleza autodescriptiva de un sistema de base de datos

Una característica fundamental del enfoque de base de datos es que el sistema de base de datos contiene no solo la base de datos en sí, sino también una definición o descripción completa de la estructura y las limitaciones de la base de datos.

Esta definición se almacena en el catálogo DBMS, que contiene información como la estructura de cada archivo, el tipo y formato de almacenamiento de cada elemento de datos y varias restricciones sobre los datos. La información

almacenados en el catálogo se denominan metadatos y describen la estructura de la base de datos primaria, algunos tipos más nuevos de sistemas de bases de datos, conocidos como sistemas NOSQL, no requieren metadatos; Más bien los datos se almacena como datos autodescriptivos que incluyen los nombres y valores de los elementos de datos juntos en una estructura.

Un software DBMS de propósito general

El paquete no está escrito para una aplicación de base de datos específica. Por tanto, debe referirse al catálogo para conocer la estructura de los archivos en una base de datos específica, como el tipo y formato de los datos a los que accederá.

Aislamiento entre programas y datos, y abstracción de datos

En el procesamiento de archivos tradicional, la estructura de los archivos de datos está incrustada en los programas de aplicación, por lo que cualquier cambio en la estructura de un archivo puede requerir cambiar todos los programas que acceden a ese archivo.

La estructura de los archivos de datos se almacena en el catálogo DBMS por separado de los programas de acceso. Llamamos a esta propiedad independencia de datos de programa. La característica que permite la independencia de los datos del programa y el funcionamiento del programa se llama abstracción de datos. Un DBMS proporciona a los usuarios un concepto

Representación de datos que no incluye muchos de los detalles de cómo son los datos almacenados o cómo se implementan las operaciones. De manera informal, un modelo de datos es un tipo de abstracción de datos que se utiliza para proporcionar esta representación conceptual.

El modelo utiliza conceptos lógicos, como objetos, sus propiedades y sus interrelaciones, que pueden ser más fáciles de entender para la mayoría de los usuarios que el almacenamiento en computadora.

Los usuarios de la base de datos y los programas de aplicación se refieren a la representación conceptual de los archivos, y el DBMS extrae los detalles del almacenamiento de archivos del catálogo cuando sean necesarios para los módulos de acceso a archivos DBMS. Muchos Se pueden utilizar modelos de datos para proporcionar esta abstracción de datos a los usuarios de la base de datos. En las bases de datos relacionales y orientadas a objetos, el proceso de abstracción incluye no solo la estructura de los datos, sino también las operaciones sobre los datos. Estas operaciones

proporcionar una abstracción de las actividades del mini mundo comúnmente comprendidas por los usuarios.

Soporte de múltiples vistas de los datos

Una base de datos suele tener muchos tipos de usuarios, cada uno de los cuales puede requerir una perspectiva o vista de la base de datos. Una vista puede ser un subconjunto de la base de datos o puede contener datos virtuales que se derivan de los archivos de la base de datos pero que no se almacenan explícitamente.

Es posible que algunos usuarios no necesiten saber si los datos a los que hacen referencia están almacenados o derivados. Un DBMS multiusuario cuyos usuarios tienen una variedad de aplicaciones distintas debe proporcionar facilidades para definir múltiples vistas.

Intercambio de datos y procesamiento de transacciones multiusuario

Un DBMS multiusuario, como su nombre lo indica, debe permitir que varios usuarios accedan a la base de datos al mismo tiempo. Esto es esencial si los datos para múltiples aplicaciones integrados y mantenidos en una sola base de datos. El DBMS debe incluir software de control de concurrencia para asegurar que varios usuarios que intentan actualizar los mismos datos lo hagan de forma controlada para que el resultado de las actualizaciones sea correcto.

El DBMS debe garantizar que solo un agente pueda acceder a cada puesto a la vez para asignación a un pasajero. Estos tipos de aplicaciones generalmente se denominan en línea, aplicaciones de procesamiento de transacciones (OLTP). Un papel fundamental del multiusuario

El software DBMS es para asegurar que las transacciones concurrentes operen correctamente y eficientemente.

Una transacción es un programa o proceso en ejecución que incluye una o más bases de datos accesos, como lectura o actualización de registros de bases de datos. Se supone que cada transacción debe ejecutar un acceso a la base de datos lógicamente correcto si se ejecuta en su totalidad sin interferencia de otras transacciones. El DBMS debe hacer cumplir varias propiedades de la transacción. La propiedad de aislamiento asegura que cada transacción parece ejecutarse de forma aislada de otras transacciones, a pesar de que cientos de las transacciones pueden estar ejecutándose al mismo tiempo. La propiedad de atomicidad asegura que o se ejecutan todas las operaciones de la base de datos en una transacción o ninguna.

Actores en escena

Para una pequeña base de datos personal, una persona normalmente define, construye y manipula la base de datos, que sí y no compartir. Identificamos a las personas cuyos trabajos implican el uso diario de una gran base de datos; los llamamos los actores en escena; se les puede llamar trabajadores detrás de escena, aquellos que trabajan para mantener el entorno del sistema de base de datos, pero que no están activamente interesados en el contenido de la base de datos como parte de su trabajo diario.

Administradores de bases de datos

En cualquier organización donde muchas personas usan los mismos recursos, es necesario un administrador jefe para supervisar y administrar estos recursos.

En un entorno de base de datos, el recurso principal es la propia base de datos y el recurso secundario es el DBMS y software relacionado. La administración de estos recursos es responsabilidad del administrador de la base de datos (DBA). El DBA es responsable de autorizar el acceso a la base de datos, coordinando y monitoreando su uso, y adquiriendo software y recursos de hardware según sea necesario. El DBA es

responsable de problemas como violaciones de seguridad y tiempo de respuesta deficiente del sistema. En organizaciones grandes, el DBA es asistido por un personal que realiza estas funciones.

Diseñadores de bases de datos

Los diseñadores de bases de datos son responsables de identificar los datos que se almacenarán en la base de datos y de elegir las estructuras adecuadas para representar y almacenar estos datos. Estas tareas se realizan principalmente antes de que la base de datos se implemente y se llene de datos. Es responsabilidad de los diseñadores de bases de datos comunicarse con todos los posibles usuarios de la base de datos para comprender sus requisitos y crear un diseño que cumpla con estos requisitos. Los diseñadores de bases de datos suelen interactuar con cada grupo potencial de los usuarios y desarrollar vistas de la base de datos que cumplan con los datos y el procesamiento requisitos de estos grupos. Luego, cada vista se analiza e integra con el vistas de otros grupos de usuarios. El diseño final de la base de datos debe ser capaz de soportar los requisitos de todos los grupos de usuarios.

Usuarios finales

Los usuarios finales son las personas cuyos trabajos requieren acceso a la base de datos para realizar consultas, actualizar y generar informes; la base de datos existe principalmente para su uso.

Hay varias categorías de usuarios finales:

- Los usuarios finales ocasionales acceden ocasionalmente a la base de datos, pero pueden necesitar información diferente cada vez. Utilizan una interfaz de consulta de base de datos sofisticada y usuarios de base de datos para especificar sus solicitudes y suelen ser gerentes de nivel medio o alto u otros navegadores ocasionales.
- Los usuarios finales ingenuos o paramétricos constituyen una parte considerable de la base de datos usuarios finales. Su principal función laboral gira en torno a la consulta constante y actualizar la base de datos, utilizando tipos estándar de consultas y actualizaciones llamadas transacciones enlatadas, que han sido cuidadosamente programadas y probado. Muchas de estas tareas ahora están disponibles como aplicaciones móviles para su uso con dispositivos móviles. Las tareas que realizan estos usuarios son variadas. Unos pocos ejemplos son: Los clientes bancarios y los cajeros controlan los saldos de las cuentas y registran los retiros. y depósitos. Los agentes de reservaciones o clientes de aerolíneas, hoteles y compañías de alquiler de automóviles verifican la disponibilidad para una solicitud determinada y realizan reservas. Los empleados de las estaciones receptoras de las empresas de transporte ingresan al paquete identificaciones mediante códigos de barras e información descriptiva mediante botones actualizar una base de datos central de paquetes recibidos y en tránsito. Los usuarios de redes sociales publican y leen elementos en sitios web de redes sociales.
- Los usuarios finales sofisticados incluyen ingenieros, científicos, analistas de negocios y otros que se familiaricen completamente con las instalaciones del DBMS para implementar sus propias aplicaciones para satisfacer sus complejos requisitos.
- Los usuarios independientes mantienen bases de datos personales mediante el uso de paquetes de programas listos para usar que brindan recursos basados en menús o gráficos fáciles de usar interfaces. Un ejemplo es el usuario de un paquete de software financiero que almacena una variedad de datos financieros personales.

Un DBMS típico proporciona múltiples facilidades para acceder a una base de datos. Usuarios finales ingenuos necesita aprender muy poco sobre las facilidades que ofrece el DBMS; simplemente tienen comprender las interfaces de usuario de las aplicaciones móviles o las transacciones estándar diseñado e implementado para su uso. Los usuarios ocasionales aprenden solo algunas instalaciones que pueden usar repetidamente. Los usuarios sofisticados intentan aprender la mayoría de las instalaciones de DBMS para lograr sus complejos requisitos. Los usuarios independientes suelen convertirse muy competente en el uso de un paquete de software específico.

Analistas de sistemas y programadores de aplicaciones (Ingenieros de software)

Los analistas de sistemas determinan los requisitos de los usuarios finales, especialmente los ingenuos y usuarios finales paramétricos y desarrollar especificaciones para transacciones enlatadas estándar que cumplan estos requisitos. Los programadores de aplicaciones implementan estas especificaciones como programas; luego prueban, depuran, documentan y mantienen estos archivos enlatados actas. Dichos analistas y programadores, comúnmente conocidos como softwa los desarrolladores o ingenieros de software deben estar familiarizados con la gama completa de capacidades que ofrece el DBMS para realizar sus tareas.

Trabajadores detrás de la escena

Además de aquellos que diseñan, usan y administran una base de datos, otros están asociados con el diseño, desarrollo y operación del software y sistema DBMS. Por lo general, estas personas no están interesadas en el contenido de la base de datos así mismo los llamamos los trabajadores detrás de escena, e incluyen a los siguientes categorías:

- Los diseñadores e implementadores de sistemas DBMS diseñan e implementan Módulos e interfaces DBMS como paquete de software. Un DBMS es muy sistema de software complejo que consta de muchos componentes o módulos, incluyendo módulos para implementar el catálogo, procesamiento de lenguaje de consulta, procesamiento de interfaz, acceso y almacenamiento en búfer de datos, control de concurrencia y manejo de recuperación y seguridad de datos. El DBMS debe interactuar con otro software del sistema, como el sistema operativo y los compiladores para varios lenguajes de programación.
- Los desarrolladores de herramientas diseñan e implementan herramientas: los paquetes de software que Facilitar el modelado y diseño de bases de datos, el diseño de sistemas de bases de datos y desempeño mejorado. Las herramientas son paquetes opcionales que a menudo se compran por separado. Incluyen paquetes para diseño de bases de datos, rendimiento monitoreo, lenguaje natural o interfaces gráficas, creación de prototipos, simulación y generación de datos de prueba. En muchos casos, los proveedores de software independientes desarrollar y comercializar estas herramientas.
- Operadores y personal de mantenimiento (personal de administración del sistema) son responsables del funcionamiento real y el mantenimiento del hardware y entorno de software para el sistema de base de datos.

Aunque estas categorías de trabajadores detrás de escena son fundamentales para hacer el sistema de base de datos disponible para los usuarios finales, normalmente no utilizan la base de datos contenidos para sus propios fines.

Ventajas de utilizar el enfoque DBMS

En esta sección discutimos algunas ventajas adicionales de usar un DBMS y el capacidades que debe poseer un buen DBMS. Estas capacidades son adicionales a las cuatro características principales. El DBA debe utilizar estas capacidades para lograr una variedad de objetivos relacionados con el diseño, administración y uso de una gran base de datos multiusuario.

Controlar la redundancia

En el desarrollo de software tradicional que utiliza procesamiento de archivos, cada grupo de usuarios mantiene sus propios archivos para manejar sus aplicaciones de procesamiento de datos. Esta redundancia al almacenar los mismos datos varias veces conduce a varios problemas.

Primero, existe la necesidad de realizar una única actualización lógica, como ingresar datos en un nuevo alumno, varias veces: una vez por cada archivo donde se registran los datos del alumno. Esto conduce a la duplicación de esfuerzos. En segundo lugar, el espacio de almacenamiento se desperdicia cuando el mismo los datos se almacenan repetidamente y este problema puede ser grave para grandes bases de datos. En tercer lugar, los archivos que representan los mismos datos pueden volverse inconsistentes. Esto puede pasar porque se aplica una actualización a algunos de los archivos pero no a otros. Incluso si una actualización, como agregar un nuevo alumno, se aplica a todos los archivos apropiados, Los datos relacionados con el estudiante pueden seguir siendo inconsistentes porque se aplican las actualizaciones independientemente por cada grupo de usuarios.

En el enfoque de base de datos, las vistas de diferentes grupos de usuarios se integran durante diseño de base de datos. Idealmente, deberíamos tener un diseño de base de datos que almacene cada elemento lógico de datos, como el nombre de un estudiante o la fecha de nacimiento, en un solo lugar en la base de datos. Esto se conoce como normalización de datos y garantiza la coherencia y ahorra espacio de almacenamiento. Sin embargo, en la práctica, a veces es necesario utilizar redundancia controlada para mejorar el rendimiento de las consultas. Al colocar todos los datos juntos, no tenemos que buscar en varios archivos para recopilar estos datos. Esto se conoce como desnormalización. En tales casos, el DBMS debe tener la capacidad de controlar esta redundancia para evitar inconsistencias entre los archivos. Estos controles se pueden especificar en el DBMS durante el diseño de la base de datos y automáticamente aplicado por el DBMS este tipo de error se puede ingresar si no se controla la redundancia. ¿Puedes decir qué parte es inconsistente?

Restringir el acceso no autorizado

Cuando varios usuarios comparten una gran base de datos, es probable que la mayoría de los usuarios no autorizado para acceder a toda la información de la base de datos.

Por ejemplo, datos financieros tales como salarios y bonificaciones a menudo se considera confidencial, y solo las personas autorizadas pueden acceder a dichos datos. Además, algunos usuarios solo pueden tener permiso para recuperar datos, mientras que otros pueden recuperar y actualizar.

Por lo tanto, también debe controlarse el tipo de operación de acceso, recuperación o actualización. Por lo general, los usuarios o grupos de usuarios reciben números de cuenta protegidos por contraseñas, que pueden utilizar para acceder a la base de datos. Un DBMS debería proporcionar un subsistema de seguridad y autorización, que el DBA utiliza para crear cuentas y para especificar restricciones de cuenta. Entonces, el DBMS debería hacer cumplir estas restricciones automáticamente. Por ejemplo, solo el personal

del DBA puede tener permitido usar ciertos software privilegiado, como el software para crear nuevas cuentas. Similar,
Los usuarios paramétricos pueden tener acceso a la base de datos solo a través de aplicaciones predefinidas o transacciones enlatadas desarrolladas para su uso.

Proporcionar almacenamiento persistente para objetos de programa

Las bases de datos se pueden utilizar para proporcionar almacenamiento persistente para los objetos y datos del programa. Esta es una de las principales razones de los sistemas de bases de datos orientados a objetos. Los lenguajes de programación suelen tener estructuras de datos complejas, como estructuras o definiciones de clases en C ++ o Java. Los valores de las variables del programa u objetos se descartan una vez que un programa termina, a menos que el programador los almacene explícitamente en archivos permanentes, lo que a menudo implica convertir estas estructuras en un formato adecuado para el almacenamiento de archivos. Cuando surge la necesidad de leer estos datos una vez más, el programador debe convertir del formato de archivo al programa estructura variable u objeto. Por tanto, un objeto complejo en C ++ se puede almacenar de forma permanente en un DBMS orientado a objetos. Se dice que tal objeto es persistente, ya que sobrevive a la finalización de la ejecución del programa y puede luego ser recuperado directamente por otro programa.

El almacenamiento persistente de objetos de programa y estructuras de datos es una función importante de los sistemas de bases de datos. Los sistemas de bases de datos tradicionales a menudo sufrían el llamado problema de desajuste de impedancia, ya que las estructuras de datos proporcionadas por el
Los DBMS eran incompatibles con las estructuras de datos del lenguaje de programación. Los sistemas de bases de datos orientados a objetos suelen ofrecer compatibilidad de estructura de datos con uno o más lenguajes de programación orientados a objetos.

Proporcionar estructuras de almacenamiento y búsqueda Técnicas para un procesamiento de consultas eficiente

Los sistemas de bases de datos deben proporcionar capacidades para ejecutar consultas y actualizaciones. Debido a que la base de datos generalmente se almacena en disco, el DBMS debe proporcionar estructuras de datos especializadas y técnicas de búsqueda para acelerar la búsqueda en disco de los registros deseados. Los archivos auxiliares llamados índices se utilizan a menudo para este propósito.

Los índices se basan típicamente en estructuras de datos de árbol o estructuras de datos hash que son convenientemente modificados para la búsqueda de disco. Para procesar los registros de la base de datos necesarios mediante una consulta en particular, esos registros deben copiarse del disco a la memoria principal.

Por lo tanto, el DBMS a menudo tiene un módulo de almacenamiento en búfer o caché que mantiene las partes de la base de datos en los búferes de memoria principal. En general, el sistema operativo es responsable del almacenamiento en búfer de disco a memoria. Sin embargo, debido a que el almacenamiento en búfer de datos es crucial para el rendimiento del DBMS, la mayoría de los DBMS hacen su propio almacenamiento en búfer de datos.

El módulo de procesamiento y optimización de consultas del DBMS es responsable de elegir un plan de ejecución de consultas eficiente para cada consulta en función de los estructuras de almacenamiento. La elección de qué índices crear y mantener es parte de diseño y ajuste de la base de datos física, que es una de las responsabilidades del DBA personal.

Proporcionar respaldo y recuperación

Un DBMS debe proporcionar instalaciones para recuperarse de fallas de hardware o software. El subsistema de respaldo y recuperación del DBMS es responsable de la recuperación. Por ejemplo, si el sistema informático falla en medio de una transacción de actualización compleja, el subsistema de recuperación es responsable de asegurarse de que la base de datos esté restaurado al estado en el que estaba antes de que la transacción comenzara a ejecutarse. Copia de seguridad en disco también es necesario en caso de una falla catastrófica del disco.

Proporcionar múltiples interfaces de usuario

Debido a que muchos tipos de usuarios con diferentes niveles de conocimientos técnicos utilizan una base de datos, un DBMS debe proporcionar una variedad de interfaces de usuario. Estos incluyen aplicaciones para usuarios móviles, lenguajes de consulta para usuarios ocasionales, interfaces de lenguaje de programación para programadores de aplicaciones, formularios y códigos de comando para usuarios paramétricos, e interfaces basadas en menús e interfaces de lenguaje natural para usuarios independientes. Tanto las interfaces de estilo de formularios como las interfaces controladas por menús se conocen comúnmente como interfaces gráficas de usuario (GUI). Muchos lenguajes y entornos especializados existen para especificar GUI. Capacidades para proporcionar interfaces GUI web a una bases de datos, o habilitar una base de datos para la Web, también son bastante comunes.

Representar relaciones complejas entre datos

Una base de datos puede incluir numerosas variedades de datos que están interrelacionados en muchas formas. Una DBMS debe tener la capacidad de representar una variedad de relaciones complejas entre los datos, para definir nuevas relaciones a medida que surgen, y para recuperar y actualizar los datos relacionados de manera fácil y eficiente.

Hacer cumplir las restricciones de integridad

La mayoría de las aplicaciones de bases de datos tienen ciertas restricciones de integridad que deben mantener los datos. Un DBMS debe proporcionar capacidades para definir y hacer cumplir estas limitaciones. El tipo más simple de restricción de integridad implica especificar un dato escriba para cada elemento de datos. Un tipo de restricción más complejo que ocurre con frecuencia implica especificar que un registro en un archivo debe estar relacionado con registros en otros archivos. El registro debe estar relacionado con el registro de un curso. Esto se conoce como integridad referencial restricción. Otro tipo de restricción especifica la unicidad de los valores de los elementos de datos, Esto es conocida como restricción de clave o unicidad. Estas restricciones se derivan de la significado o semántica de los datos y del minimundo que representa. Es el responsabilidad de los diseñadores de la base de datos para identificar las restricciones de integridad durante diseño de base de datos. Algunas restricciones se pueden especificar al DBMS y automáticamente en vigor. Es posible que sea necesario verificar otras restricciones mediante programas de actualización o en la hora de entrada de datos. Para aplicaciones típicas de gran tamaño, se acostumbra llamar reglas comerciales a estas restricciones. Un elemento de datos puede ingresarse por error y aún así satisfacer las restricciones de integridad especificadas. En el modelo Entidad-Relación, una relación debe involucrar al menos a dos entidades. Reglas que pertenecen a un dato específico modelo se denominan reglas inherentes del modelo de datos.

Permitir inferencias y acciones

Usar reglas y activadores

Algunos sistemas de bases de datos brindan capacidades para definir reglas de deducción para inferir nueva información de los hechos almacenados en la base de datos. Tales sistemas se llaman sistemas de base de datos deductivos. . Estos pueden ser especificados declarativamente como reglas, que cuando son compiladas y mantenidas por el DBMS puede determinar a todos los estudiantes en libertad condicional. En un DBMS tradicional, se tendría que escribir un código de programa de procedimiento explícito para admitir tales aplicaciones. Pero si las reglas del mini mundo cambian, generalmente es más conveniente cambiar las reglas declaradas reglas de deducción que recodificar programas de procedimiento. En la base de datos relacional actual sistemas, es posible asociar disparadores con tablas. Un disparador es una forma de regla activado por actualizaciones a la tabla, lo que resulta en realizar algunas operaciones adicionales en algunas otras tablas, enviar mensajes, etc. Los procedimientos más complicados para hacer cumplir las reglas se denominan popularmente procedimientos almacenados; se vuelven parte de la definición general de la base de datos y se invocan de manera apropiada cuando se cumplen ciertas condiciones. Los sistemas de bases de datos activos proporcionan una funcionalidad más poderosa, que proporcionan reglas activas que pueden iniciar acciones automáticamente cuando ciertos ocurren eventos y condiciones.

Implicaciones adicionales del uso y enfoque de base de datos

Esta sección analiza algunas implicaciones adicionales del uso del enfoque de base de datos que puede beneficiar a la mayoría de las organizaciones. Potencial para hacer cumplir los estándares. El enfoque de la base de datos permite al DBA definir y hacer cumplir los estándares entre los usuarios de bases de datos en una organización grande. Esto facilita la comunicación y cooperación entre varios departamentos, proyectos y usuarios dentro de la organización. El DBA Puede hacer cumplir los estándares en un entorno de base de datos centralizado más fácilmente que en un entorno en el que cada grupo de usuarios tiene el control de sus propios archivos de datos y software.

Reducción del tiempo de desarrollo de aplicaciones.

Una característica de venta principal del enfoque de base de datos es que el desarrollo de una nueva aplicación, como la recuperación de ciertos datos de la base de datos para imprimir un nuevo informe: lleva muy poco tiempo. Diseño e implementar una gran base de datos multiusuario desde cero puede llevar más tiempo que escribir una única aplicación de archivo especializada. Sin embargo, una vez que una base de datos está activa y en ejecución, generalmente se requiere mucho menos tiempo para crear nuevas aplicaciones utilizando las instalaciones de DBMS.

Flexibilidad.

Puede ser necesario cambiar la estructura de una base de datos a medida que cambian los requisitos. Los DBMS modernos permiten ciertos tipos de cambios evolutivos en la estructura de la base de datos sin afectar los datos almacenados y los programas de aplicación existentes.

Disponibilidad de información actualizada.

Un DBMS hace que la base de datos esté disponible a todos los usuarios. Tan pronto como se aplique la actualización de un usuario a la base de datos, todos los demás usuarios pueden ver esta actualización inmediatamente. Esta disponibilidad de información actualizada es esencial para muchas aplicaciones de procesamiento de transacciones, como los sistemas de reserva o bases de datos bancarias, y es posible gracias a los subsistemas de control de concurrencia y recuperación de un DBMS.

Economías de escala.

El enfoque DBMS permite la consolidación de datos y aplicaciones, reduciendo así la cantidad de superposición inútil entre las actividades de personal de procesamiento de datos en diferentes proyectos o departamentos, así como redundancias entre aplicaciones. Esto permite que toda la organización invierta en más potentes procesadores, dispositivos de almacenamiento o equipos de red, en lugar de tener cada uno departamento compra su propio equipo (de menor rendimiento). Esto reduce en general costos de operación y manejo.

Aplicaciones de bases de datos tempranas que utilizan jerárquica y sistemas de red

Muchas de las primeras aplicaciones de bases de datos mantenían registros en grandes organizaciones como corporaciones, universidades, hospitales y bancos. En muchas de estas aplicaciones, había un gran número de registros de estructura similar.

Uno de los principales problemas de los primeros sistemas de bases de datos era la combinación de relaciones conceptuales con el almacenamiento físico y la ubicación de registros en disco. Por lo tanto, estos sistemas no proporcionaron suficiente abstracción de datos y datos del programa. capacidades de independencia. En particular, las nuevas consultas que

requirió una organización de almacenamiento diferente para un procesamiento eficiente fueron bastante difíciles para implementar de manera eficiente. También fue laborioso reorganizar la base de datos cuando se realizaron cambios en los requisitos de la aplicación.

Otra deficiencia de los primeros sistemas era que solo proporcionaban programación interfaces de idioma. Esto hizo que la implementación fuera lenta y costosa nuevas consultas y transacciones, ya que los nuevos programas tenían que ser escritos, probados y depurado. La mayoría de estos sistemas de bases de datos se implementaron en grandes y costosas computadoras mainframe que comenzaron a mediados de la década de 1960 y continuaron durante las décadas de 1970 y 1980. Los principales tipos de sistemas tempranos se basaron en tres paradigmas principales: sistemas jerárquicos, sistemas basados en modelos de red y sistemas de archivos invertidos.

Proporcionar abstracción de datos y flexibilidad de aplicaciones con bases de datos relacionales

Las bases de datos relacionales se propusieron originalmente para separar el almacenamiento físico de datos de su representación conceptual y para proporcionar una base matemática para la representación y consulta de datos. El modelo de datos relacionales también introdujo lenguajes de consulta de alto nivel que proporcionaron una alternativa al lenguaje de programación interfaces, lo que hace que sea mucho más rápido escribir nuevas consultas. Los sistemas relacionales se dirigieron inicialmente a las mismas aplicaciones que los sistemas anteriores y proporcionaron flexibilidad para desarrollar nuevas consultas rápidamente y reorganizar la base de datos como los requisitos cambiaron. Por lo tanto, la abstracción de datos y la independencia de los datos del programa mejoraron mucho en comparación con los sistemas anteriores.

Los primeros sistemas relacionales experimentales desarrollados a finales de la década de 1970 y los sistemas comerciales de gestión de bases de datos relacionales (RDBMS) introducidos en el principios de la década de 1980 fueron bastante lentos, ya que no utilizaban punteros de almacenamiento físico o ubicación de registros para acceder a registros de datos relacionados. Con el desarrollo de nuevas técnicas de almacenamiento e indexación y mejor procesamiento y optimización de consultas, su desempeño mejoró. Con el tiempo, las bases de datos relacionales se convirtieron en el tipo dominante de sistema de bases de datos para las aplicaciones de bases de datos tradicionales. Las bases de datos relacionales ahora existen en casi todos los tipos de computadoras, desde pequeñas computadoras personales a grandes servidores.

Aplicaciones orientadas a objetos y la necesidad para bases de datos más complejas

La aparición de lenguajes de programación orientados a objetos en la década de 1980 y la necesidad de almacenar y compartir objetos complejos y estructurados condujo al desarrollo de

Bases de datos orientadas a objetos (OODB). Inicialmente, las OODB se consideraban un competidor de las bases de datos relacionales, ya que proporcionaban estructuras de datos más generales. Ellos también

incorporó muchos de los paradigmas útiles orientados a objetos, como datos abstractos tipos, encapsulación de operaciones, herencia e identidad de objeto sin embargo, La complejidad del modelo y la falta de un estándar temprano contribuyeron a su uso limitado. Ahora se utilizan principalmente en aplicaciones especializadas, como ingeniería sistemas de diseño, publicación multimedia y fabricación. A pesar de las expectativas que tendrán un gran impacto, su penetración general en el mercado de productos de bases de datos sigue siendo baja. Además, muchos conceptos orientados a objetos se incorporaron en las versiones más nuevas de DBMS relacionales, lo que llevó a sistemas de gestión de bases de datos, conocidos como ORDBMS.

Intercambio de datos en la web para comercio electrónico con XML

La World Wide Web proporciona una gran red de computadoras interconectadas.

Los usuarios pueden crear páginas web estáticas utilizando un lenguaje de publicación web, como el lenguaje de marcado de hipertexto (HTML), y almacenar estos documentos en servidores web donde otros usuarios (clientes) pueden acceder a ellos y verlos a través de navegadores web. Los documentos se pueden vincular a través de hipervínculos, que son indicadores de otros documentos.

A partir de la década de 1990, el comercio electrónico (e-commerce) surgió como un importante aplicación en la Web. Gran parte de la información crítica sobre el comercio electrónico Web páginas son datos extraídos dinámicamente de DBMS, como información de vuelos, precios de productos y disponibilidad de productos. Se desarrollaron una variedad de técnicas para permitir el intercambio de datos extraídos dinámicamente en la Web para mostrarlos en la Web páginas. EXtended Markup Language (XML) es un estándar para intercambiar datos entre varios tipos de bases de datos y páginas web.

XML combina conceptos

a partir de los modelos utilizados en sistemas de documentos con conceptos de modelado de bases de datos.

Ampliación de las capacidades de la base de datos para nuevas aplicaciones

El éxito de los sistemas de bases de datos en las aplicaciones tradicionales animó a los desarrolladores de otros tipos de aplicaciones a intentar utilizarlos. Tales aplicaciones

tradicionalmente utilizaba su propio software especializado y estructuras de datos y archivos.

Los sistemas de bases de datos ahora ofrecen extensiones para soportar mejor los requisitos especializados de algunas de estas aplicaciones. Los siguientes son algunos ejemplos de estas aplicaciones:

1. Aplicaciones científicas que almacenan grandes cantidades de datos resultantes de experimentos científicos en áreas como la física de altas energías, el mapeo del genoma humano y el descubrimiento de estructuras proteicas.
2. Almacenamiento y recuperación de imágenes, incluidas noticias escaneadas o fotografías personales, imágenes fotográficas de satélite e imágenes de procedimientos médicos como radiografías y pruebas de resonancia magnética (resonancia magnética).
3. Almacenamiento y recuperación de videos, como películas y videoclips de noticias o cámaras digitales personales.
4. Aplicaciones de minería de datos que analizan grandes cantidades de datos para buscar las ocurrencias de patrones o relaciones específicas, y para identificar patrones inusuales en áreas como la detección de fraudes con tarjetas de crédito.
5. Aplicaciones espaciales que almacenan y analizan ubicaciones espaciales de datos, como información meteorológica, mapas utilizados en los sistemas de información geográfica y sistemas de navegación para automóviles.
6. Aplicaciones de series de tiempo que almacenan información como datos económicos en puntos regulares en el tiempo, como ventas diarias y nacionales brutos mensuales cifras de productos.

Rápidamente se hizo evidente que los sistemas relacionales básicos no eran muy adecuados para muchos de estas aplicaciones, generalmente por una o más de las siguientes razones:

- Se necesitaban estructuras de datos más complejas para modelar la aplicación que la simple representación relacional.
- Se necesitaban nuevos tipos de datos además de los numéricos y de caracteres básicos tipos de cadenas.
- Fueron necesarias nuevas operaciones y construcciones de lenguaje de consulta para manipular los nuevos tipos de datos.
- Se necesitaban nuevas estructuras de almacenamiento e indexación para una búsqueda eficiente en los nuevos tipos de datos.

Esto llevó a los desarrolladores de DBMS a agregar funcionalidad a sus sistemas. Algunas funcionalidades eran de propósito general, como la incorporación de conceptos de bases de datos orientadas a objetos en sistemas relacionales. Otra funcionalidad tenía un propósito especial, en la forma de módulos opcionales que podrían utilizarse para aplicaciones específicas.

Aparición de sistemas de almacenamiento de Big Data y bases de datos NOSQL

En la primera década del siglo XXI, la proliferación de aplicaciones y plataformas como sitios web de redes sociales, grandes empresas de comercio electrónico, búsqueda web índices, y el almacenamiento / respaldo en la nube llevaron a un aumento en la cantidad de datos almacenados en grandes bases de datos y servidores masivos. Se necesitaban nuevos tipos de sistemas de bases de datos para administrar estas enormes bases de datos, sistemas que proporcionarían búsquedas rápidas y recuperación, así como almacenamiento confiable y seguro de tipos de datos no tradicionales, como publicaciones en redes sociales y tweets. Algunos de los requisitos de estos nuevos

sistemas fueron no es compatible con los DBMS relacionales SQL (SQL es el modelo de datos estándar y lenguaje para bases de datos relacionales). El término NOSQL generalmente se interpreta como No Solo SQL, lo que significa que en sistemas que gestionan grandes cantidades de datos, algunos de los datos se almacenan usando sistemas SQL, mientras que otros datos se almacenarían usando NOSQL, dependiendo de los requisitos de la aplicación.

Cuándo no usar un DBMS

A pesar de las ventajas de usar un DBMS, hay algunas situaciones en las que un DBMS puede implicar costos generales innecesarios en los que no se incurriría en procesamiento de archivos tradicional. Los costos generales de usar un DBMS se deben a la siguiente:

- Alta inversión inicial en hardware, software y capacitación.
- La generalidad que proporciona un DBMS para definir y procesar datos.
- Gastos generales para proporcionar funciones de seguridad, control de simultaneidad, recuperación e integridad.

Por lo tanto, puede ser más conveniente desarrollar aplicaciones de bases de datos personalizadas en las siguientes circunstancias:

- Aplicaciones de base de datos simples y bien definidas que no se espera que cambien en absoluto.
- Requisitos estrictos en tiempo real para algunos programas de aplicación que pueden no se cumple debido a la sobrecarga de DBMS.
- Sistemas integrados con capacidad de almacenamiento limitada, donde un propósito general DBMS no encajaría.
- Sin acceso de múltiples usuarios a los datos.

Algunas industrias y aplicaciones han optado por no utilizar DBMS. Por ejemplo, muchas herramientas de diseño asistido por computadora (CAD) utilizadas por ingenieros mecánicos y civiles tienen software de administración de datos y archivos patentado que está diseñado para las manipulaciones internas de dibujos y objetos 3D. Similar, Los sistemas de comunicación y conmutación diseñados por empresas como AT&T fueron primeras manifestaciones de software de base de datos que se hizo para funcionar muy rápido con datos organizados jerárquicamente para un acceso rápido y enrutamiento de llamadas. Las implementaciones de GIS a menudo implementan sus propios esquemas de organización de datos para implementar funciones relacionadas con el procesamiento de mapas, contornos físicos, líneas, polígonos, etc.

Resumen

Definimos una base de datos como una colección de datos relacionados, donde los datos significa hechos registrados. Una base de datos típica representa algún aspecto del mundo real y se utiliza con fines específicos por uno o más grupos de usuarios. Un DBMS es un paquete de software generalizado para implementar y mantener una computadora base de datos. La base de datos y el software juntos forman un sistema de base de datos. Identificamos varias características que distinguen el enfoque de la base de datos del tradicional aplicaciones de procesamiento de archivos, y discutimos las principales categorías de bases de datos usuarios, o los actores en la escena. Observamos que además de los usuarios de la base de datos, Hay varias categorías de personal de apoyo, o trabajadores detrás de escena, en un entorno de base de datos. Presentamos una lista de capacidades que debe proporcionar el software DBMS para el DBA, los diseñadores de bases de datos y los usuarios finales

para ayudarlos a diseñar, administrar y utilizar una base de datos. Luego dimos una breve perspectiva histórica sobre la evolución de las aplicaciones de bases de datos. Señalamos el rápido crecimiento reciente de las cantidades y tipos de datos que deben almacenarse en bases de datos, y discutimos la aparición de nuevos sistemas para el manejo de aplicaciones de “big data”. Finalmente, discutimos la sobrecarga costos de usar un DBMS y discutimos algunas situaciones en las que puede no ser ventajoso usar uno.