

Tema2ArquitecturadeunSGBD.pdf



BetaNT82



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



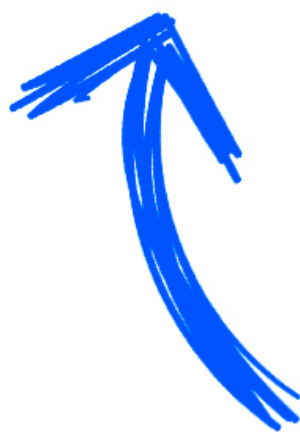
Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



Estudiar sin publi es posible.



Compra Wuolah Coins y que nada
te distraiga durante el estudio



Tema 2: Arquitectura de un SGBD

1. Una arquitectura con 3 niveles

La arquitectura en niveles se usan ya que los usuarios pueden acceder a los mismos datos pero con distintos permisos. Además de esto también nos ayudan a mantener la independencia lógica, por la organización global.

Los 3 niveles son los siguientes: Interno, Conceptual y Externo

1.1 Nivel Interno

Es el nivel más cercano al almacenamiento físico. Es la capa donde se establece la forma en que se implantan las estructuras de datos que organizan los demás niveles.

Es la representación física de la BD en el ordenador, es decir, es la que nos dice cómo están almacenados los datos y les busca el rendimiento óptimo.

Aquí se debe representar las estructuras de datos, las organizaciones de ficheros, la comunicación con el SO para gestionar el almacenamiento, la comprensión de datos, etc.

Como hemos nombrado antes, en este nivel parte de la actividad se relacionan con el SO. Las responsabilidades del SO son el nivel físico para todos, y el resto dependen del SGBD que sea y también del SO.

Se usa un lenguaje especial (DDL) para aumentar el rendimiento sin que altere la visión abstracta.

1.2 Nivel Conceptual

Es una abstracción global de la BD que agrupa todas las percepciones de los usuarios sobre ella.

Se podría entender como una visión global de los datos, y como estamos hablando de visiones del usuario, nos referimos a la estructura lógica de los datos, es decir, que datos y como se relacionan.

En este nivel se representan las entidades, relaciones y sus atributos; las restricciones; la semántica de los datos; y la información sobre seguridad e integridad.

Sirve como en un eslabón que proporciona soporte a la vista externa, pero no se conoce nada sobre el almacenamiento en este nivel.

1.3 Nivel Externo

En este nivel están todas las percepciones individuales de cada usuario, varios usuarios pueden coincidir en la percepción. Para ser más específicos, el nivel externo es cada una de las perspectivas de los usuarios, estas se pueden entender como una estructura de datos o módulo de este nivel.

El nivel externo está formado por muchos subniveles externos, en cada subnivel solo se representan las entidades, relaciones y atributos que le son de interés al usuario de ese subnivel.

El usuario elige como representar y como calcular los datos. Y los usuarios programadores desarrollan sus actividades desde este nivel.

Para realizar estas actividades necesita su propio lenguaje, que veremos más tarde.

2. Correspondencias entre niveles

Una correspondencia o transformación entre niveles es un conjunto de normas que dicen como se definen los datos de un nivel en términos de otro. Este es un mecanismo esencial para lograr la independencia.

Existen los siguientes tipos de correspondencias:

- Conceptual/Interna. Es una transformación de campos del nivel interno al nivel conceptual. Este garantiza la independencia física, ya que si se cambia el nivel interno, se cambiaría la correspondencia y se mantiene invariable el nivel conceptual.
- Externa/Conceptual. En esta transformación la independencia lógica queda asegurada. El nivel que no varía es el externo, y el que sufre cambios es el conceptual.
- Externa/Externa. Algunos SGBD nos permite hacer transformaciones entre un nivel externo con otro nivel externo, esto sigue manteniendo la

independencia lógica. Aquí podemos decir que existen dos niveles externos: el dependiente (el que no cambia) y el subyacente (el que varía).

Las transformaciones se hacen con lenguajes para el SGBD, es decir, que son comprensibles por él. Lo más común en una base de datos es que sea guardada como metadatos de la SGBD. El SGBD deja recursos al DBA(administrador) para que pueda intervenir como para modificar criterios o variar el almacenamiento.

Las transformaciones en las que intervenga el nivel externo son escritas por el usuario, el lenguaje que se suele usar para ello es el DDL, pero es recomendable que lo haga coordinándose con el DBA.

Para el nivel interno se suelen usar métodos que hacen llamadas al SO, para que así sean portables y que cada SO resuelva la llamada como crea, aunque esto no ocurre siempre, ya que depende de que SGBD usemos.

3. Lenguajes de una BD

Existen distintas arquitecturas, pero según la arquitectura ANSI/SPARC nos dicen que debe haber un lenguaje específico para los datos y realizar sobre ellos : definición, control y manipulación. A este lenguaje lo denominamos DSL (sublenguaje) y lo implementa el SGBD.

El DLL (sublenguaje de definición de datos) es un subconjunto del DSL para definir estructuras y esquemas en la BD.

El DML (sublenguaje de manipulación de datos) es un subconjunto de DSL para:

- Meter datos en esquemas.
- Modificar datos.
- Eliminar datos.
- Consultar datos y estructuras.

El DCL (sublenguaje de control de datos) es un subconjunto de DSL que permite gestionar los requisitos de acceso y otras tareas administrativas.

La arquitectura ANSI/SPARC recomienda que como mínimo se tenga 1 DLL, 1 DML y 1 DCL para cada nivel. En la realidad cada sublenguaje tiene una implementación única y debido a la separación de los lenguajes para cada

nivel que se realiza según los privilegios que tiene cada distinto tipo de usuario, se discrimina quien puede ejecutar qué.

Queda otro lenguaje que es el lenguaje de la aplicación o anfitrión, debido a que existen múltiples lenguajes se han realizado una gran variedad de aplicaciones, aunque han surgido distintos idiomas que han intentado centralizarse como SQL. Pero como aun así se han usado una gran variedad han tenido que surgir mecanismos que permiten usar las distintas estructuras que proporciona cada lenguaje para implementar el DSL. Esto también aplica de forma inversa, es decir, que surgen mecanismos para adaptar los métodos del DSL para cada aplicación, esto es el llamado acoplamiento.

3.1 Acoplamiento

Según el acoplamiento podemos dividir los lenguajes en dos tipos:

- Débilmente Acoplados. Son lenguajes de propósito general, en las que se distinguen cuales son sentencias del lenguaje y cuales para acceder a la BD.
- Fuertemente acoplados. Son lenguajes y herramientas de propósito específico que tienen un DSL y herramientas para facilitar el desarrollo de aplicaciones de BD. En estas se coge el DSL y se extiende para que haga control de flujo, operaciones E/S e interfaz.

3.2 Alternativas del acoplamiento débil

Se clasifican según su enfoque principal

- Uso de API's de acceso a BD. El SGBD proporciona una biblioteca para el lenguaje anfitrión, así puede acceder a la BD desde el código fuente del anfitrión. Con esta opción para el programa final se debe enlazar la biblioteca con el código del objeto. Ejemplo: ODBC, JDBC.
- DSL inmerso en el código fuente del lenguaje anfitrión. El código es híbrido entre el lenguaje anfitrión y el DSL, después el SGBD traduce ese código en uno que se enlaza con la biblioteca que otorga una API pero en este lo realiza el SGBD y no el usuario. Ejemplo: SQL inmerso en C, SQLJ.

3.3 Alternativas del acoplamiento fuerte

- Los propietarios de cada empresa han realizado una alternativa como PL/SQL de Oracle

- Ejecución de Java sobre una máquina virtual implantada en el SGBD. Esto proporciona una gran capacidad

3.4 Realidad en la implementación

No se ha hecho caso a las recomendaciones ANSI/SPARC.

En la realidad lo que se ha realizado es adaptarse a los lenguajes realizados por fabricantes de SGBD.

Estos lenguajes no realizan separaciones entre niveles, ni sublenguajes. Esto permite que los usuarios programadores y DBA's usen el mismo lenguaje pero con distintas restricciones.

4. Enfoques para la arquitectura de un SGBD

En la era de la información el papel del SGBD se ha convertido en uno muy importante, por lo que se han realizado avances sobre este, para que sean más complejos y tengan mayor funcionalidad, incluso algunos pueden ejercer la función de administrador sobre ellos mismos. Durante las distintas épocas se han pasado por distintos esquemas.

4.1 Arquitectura Centralizada

Toda la gestión y procesamiento recae en los servidores centrales. Cada terminal accedía a la BD, cada computador tenía instalado las aplicaciones pertinentes y el SGBD, y con estos cada computador interactúa con el servidor.

Este esquema tiene serios problemas, el más importante es que el precio del computador principal es muy elevado.

También tiene ventajas, que son que cada usuario tiene un gran manejo de la interacción con su propia terminal, y que el precio de las terminales no son tan caros.

4.2 Arquitectura Distribuida

La solución que se puso en su momento fue la aproximación Cliente-Servidor. Se basa en dos servidores, un servidor para la BD y otro de escucha, este

último sirve de enlace entre los PC's y el servidor de la BD. Mientras que los clientes tienen el programa de la aplicación y el servidor de la BD

4.3 Arquitectura de 3 niveles

Este enfoque surgió debido a Internet. Los 3 niveles que tienen son nivel de servidor de datos, nivel de servidor de aplicaciones y nivel de cliente.

4.3.1 Nivel de servidor de datos

Se organiza una serie de nodos en los que usando SGBD's distribuida se gestiona una BD distribuida y global. Este se usa cuando se van a tener datos almacenados locales y luego datos en sedes centrales, así las peticiones de una sede de cierto dato se traducen como peticiones a la sede donde se encuentra el dato. El DBA adopta una estrategia que distribuye los datos según su generación, y su frecuencia de acceso. Otra tarea suya es que copias se ubican en cada sede y ponerles una frecuencia de sincronización.

4.3.2 Nivel de servidor de aplicaciones

Tiene un elevado coste de mantenimiento para actualizar las configuraciones de los PC's clientes y para su distribución e instalación. Se usan clientes ligeros que pueden ejecutar aplicaciones conforme a estándares, a veces, incluso solo ejecutan la interfaz de usuario que interactúa con la parte lógica de la app.

El mantenimiento y actualización es menor que el requerido en el enfoque Cliente/Servidor, pero tiene que tener estrategias más sofisticadas ya que gran parte de la funcionalidad (gestión de seguridad, parte lógica) no es ejecutado por el cliente. Por eso se realizan configuraciones excesivas sobre el hardware y el software.

4.3.3 Nivel de cliente

Se poseen PC's ligeros con configuraciones basadas en estándares abiertos y portables, que hacen que la dependencia del hardware y el SO se reduzca. Una traza de lo que haría el cliente es:

1. El usuario llama a una aplicación del servidor de apps
2. El usuario interacciona con la interfaz (creada con appletJava) y está interfaz interactúa con la parte lógica
3. Se generan peticiones a los diferentes servidores de BD

4.3.4 Ventajas e Inconveniente

Ventajas:

- Reducción en el mantenimiento de los clientes
- Facilidad y flexibilidad para que el usuario opere con las apps.
- Software reutilizable y personalizable, por lo que baja el tiempo de desarrollo de las apps.

Inconvenientes:

- Mayor complejidad de configuración y administración en la configuración & administración del servidor, y en el desarrollo de las apps.

5. El administrador de la BD

En este apartado vamos a detallar todas la funciones que tiene el administrador de la BD, que lo vamos a llamar DBA.

5.1 Elaboración del esquema conceptual

Para realizar el esquema el DBA debe realizar las siguientes etapas:

1. Análisis de necesidades de información
2. Identificación de datos operativos
3. Elaboración de esquema lógico
4. Implantación del esquema en ddl. El SGBD guardará el archivo fuente como la versión compilada.

5.2 Decidir la estructura de almacenamiento en el nivel interno

El DBA debe elegir como guardará los datos con el DDL, para ello realizará un esquema interno y su transofrmación conceptual/interna en el DDL interno. Lo más común es que el DDL interno no esté disponible y este se defina en el DDL conceptual más lo que sea necesario para definirlo.

5.3 Conexión con usuarios

Como el nivel externo se relaciona con la visión que tenga el usuario se puede tener dos enfoques para realizar esta conexion:

- Una visión semántica particular. El DBA debe recoger e integrar esta visión en el esquema
- Representación específica de la visión en la estructura de datos. El DBA debe apoyar al desarrollador de apps para realizar la correspondencia de buena forma en el DDL.

5.4 Definir restricciones de Integridad

El esquema conceptual va a ser el que define la estructura que va a guardar los datos válidos y la que va a evitar que se guarden datos incompatibles. El DBA debe plasmar las restricciones en el esquema de este nivel.

5.5 Definir e Implantar la política de seguridad

Debido a los diferentes usuarios que puede tener una BD y los distintos permisos que pueden tener estos, se debe determinar que usuario puede acceder a que dato. El DBA debe implementar esto y monitorizar el cumplimiento de esta.

5.6 Definir e implementar la estrategia de recuperación frente a fallos

Si en la BD ocurre algún error, no se debe perder información. Para ello el DBA usando el SGBD y demás herramientas informáticas debe tomar precauciones para que esto no ocurra. Existen combinaciones de SO's y SGBD's que permiten que si un servidor falla todos los demás se den cuenta, un ejemplo es la estructura de cluster.

Para los fallos en el almacenamiento existen estructuras, con redundancia, que cuando un dato se pierde por un fallo se permite recuperarlo.

Además el DBA debe realizar copias de seguridad de forma periódica para asegurar la no pérdida de información.

En caso de que la información que se pierde sea por las conexiones de las redes, el DBA debe poder restaurar la base al estado anterior de que se iniciase esa comunicación.

5.7 Optimización del rendimiento

El DBA se debe encargar de que el SGBD funcione de la forma más eficiente, para ello reorganiza el espacio no utilizado y las operaciones para que se

ejecuten de forma más rápida.

En caso de tener unas necesidades mayores a los recursos disponibles el DBA debe intentar convencer a los propietarios de aumentar los recursos, si esto no es posible, debe equilibrar la carga de almacenamiento y procesamiento según las prioridades de la BD.

5.8 Monitorizar el SGBD

Como hemos visto el SGBD es una herramienta muy importante para el BD, por ello debe realizar un seguimiento de la actividad del sistema(supervisar acceso de los usuarios a la BD, comprobar niveles de uso del almacenamiento, evaluar eficiencia). Actualmente, los SGBD proporcionan herramientas que informan de esto para que el DBA no deba prestar atención a una sola tarea.