

SISTEMAS XESTORES DE BASES DE DATOS

Índice

1.	Sistemas xestores de bases de datos	2
1.1	Introdución	2
1.2	Obxectivos dos Sistemas Xestores de Bases de Datos	2
1.3	Funcións dos Sistemas Xestores de Bases de Datos	2
1.3.1	Definición de datos	3
1.3.2	Manipulación de datos	3
1.3.3	Seguridade e integridade dos datos	3
1.3.4	Xestión de transaccións e acceso concorrente	4
	Propiedades das transaccións	5
1.3.5	Outras funcións dos Sistemas Xestores de Bases de Datos	5
1.4	Compoñentes dun Sistema Xestor de Bases de Datos	5
1.5	Tipos de Sistemas Xestores de Bases de Datos	7
1.5.1	En función do modelo de datos	7
1.5.2	En función do tipo de licenza	7
1.5.3	En función do número de usuarios	8
1.5.4	En función da distribución dos seus compoñentes	8
	A distribución dos datos nun SXBD distribuído	8

1. Sistemas xestores de bases de datos

1.1 Introducción

Se dispoñemos dunha base de datos, necesitamos un conxunto de ferramentas para poder acceder ós datos. A forma de acceso vai a depender da tarefa a realizar, por exemplo:

- O administrador engadirá usuarios, modos de acceso, reorganizará a BD cando sexa necesario,...
- Os analistas e programadores deberán coñecer a estrutura da BD para poder facer as aplicacións e programas.
- Os usuarios finais utilizarán a BD a través de aplicacións ou linguaxes de consulta interactivos.

Para poder realizar todo isto necesitamos un sistema xestor de base de datos (**SXBD**) que se pode definir como o **“conxunto de ferramentas que subministra a todos (administrador, analistas, programadores, usuarios) os medios necesarios para describir, recuperar e manipular os datos almacenados na BD, mantendo a seguridade, integridade e confidencialidade dos mesmos”**.

1.2 Obxectivos dos Sistemas Xestores de Bases de Datos

Nun ambiente multiusuario o SXBD ofrece á empresa un control centralizado da súa información. Os obxectivos que se plantexan nestes sistemas están relacionados coa intención de evitar os problemas que existían noutros sistemas de información:

- Evitar a redundancia dos datos, eliminando así a inconsistencia dos mesmos.
- Manter a integridade dos datos realizando as validacións necesarias cando se realicen modificacións na base de datos e actualizando os datos que poidan estar repetidos.
- Mellorar os mecanismos de seguridade dos datos.
- Asegurar a independencia dos programas e os datos, é dicir, a posibilidade de modificar a estrutura da base de datos (esquema) sen necesidade de modificar os programas das aplicacións que manexan eses datos.
- Mellorar a eficacia de acceso ós datos, en especial no caso de consultas imprevistas.

1.3 Funcións dos Sistemas Xestores de Bases de Datos

As principais funcións que debe realizar un SXBD son:

- A definición dos datos.
- A manipulación dos datos.

- Garantir a seguridade e integridade dos datos.
- A xestión das transaccións e o acceso concorrente.

1.3.1 Definición de datos

A Linguaxe de Definición de Datos (LDD) permítelle ao SXBD describir e definir os esquemas da base de datos. Esta linguaxe debe permitir:

- A creación de obxectos conceptuais (esquema conceptual).
- A descrición de estruturas físicas (esquema físico).
- A definición de vistas (esquema externo).

Para unha base de datos existente, a LDD debe permitir tamén:

- A modificación da descrición de obxectos conceptuais.
- O borrado de obxectos conceptuais.
- A modificación e borrado de camiños de acceso.
- A modificación das vistas.

Unha LDD está composta por un conxunto de comandos que actúan sobre os obxectos conceptuais e sobre os obxectos físicos.

O conxunto das descricións de obxectos dunha base de datos coñécese co nome de dicionario de datos ou catálogo e almacénase xunto cos datos.

1.3.2 Manipulación de datos

A función de manipulación de datos encárgase de todas as operacións de intercambio de datos entre os usuarios e a base de datos. Esta función faise coa axuda da Linguaxe de Manipulación de Datos (L.M.D.), que está composta por un conxunto de comandos que nos permiten a consulta ou posta ó día (inserción, modificación e borrado) dos datos dunha base de datos.

As L.M.D. funcionan de forma diferente, segundo o modelo de datos:

- Nos modelos en rede e xerárquicos son linguaxes procedementais, é dicir, os programadores deben indicar o camiño a seguir para acceder ós datos solicitados.
- No modelo relacional son linguaxes declarativas (non procedementais) e non é necesario indicar como se ten que acceder ós datos, é suficiente indicar que é o que se desexa obter.

1.3.3 Seguridade e integridade dos datos

Trátase de garantir a coherencia e validez dos datos, comprobando que só os usuarios autorizados poidan efectuar as operacións correctas sobre a base de datos. Isto conséguese mediante:

- Un control sobre os usuarios que acceden á base de datos e os tipos de operacións que están autorizados a realizar. Este control chámase xestión de autorizacións e permite crear ou bo-

rrar usuarios e conceder ou retirar dereitos para poder realizar determinados tipos de operacións (por exemplo: para crear obxectos, borrar obxectos, modificar datos, etc.).

- A validación das operacións realizadas cos datos. Este control faise mediante un conxunto de regras chamadas restricións de integridade. Existen varios tipos de restricións de integridade, como por exemplo as restricións de integridade referencial, que imponen que as modificacións realizadas sobre algúns datos obriguen a realizar modificacións doutros datos cos que están relacionados (por exemplo, se se modifica o código dun artigo debería modificarse ese código en todos os pedidos que soliciten o artigo).
- Unha protección dos datos contra os accesos mal intencionados e os fallos. Os accesos mal intencionados sóense evitar coa asignación de palabras de paso (password) ós usuarios, a definición de vistas, protección física dos datos (encriptado dos datos), etc. Con respecto ós fallos causados por manipulacións incorrectas, ou accidentes lóxicos ou físicos, os SXBD soen dispoñer de utilidades de recuperación dos datos despois dun fallo.

A correcta utilización de todas estas operacións de seguridade e integridade constitúe unha tarefa esencial do Administrador da base de datos (xestión de usuarios e os seus dereitos, xestión de vistas e recuperación da base de datos despois dun fallo).

1.3.4 Xestión de transaccións e acceso concorrente

O principal obxectivo da implantación dunha base de datos nun sistema de información dunha organización é poñer a disposición dun gran número de usuarios un conxunto integrado de datos. O SXBD debe protexer os datos cando estes sexan manipulados simultaneamente por diferentes usuarios.

Unha transacción defínese como unha unidade lóxica de tratamento (conxunto de ordenes) que aplicada a un estado coherente da base de datos déixaa, de novo, nun estado coherente despois de facer as modificacións. Unha transacción só se pode executar completamente ou ser anulada. Por exemplo, unha transacción bancaria na que se traspasan 1000 euros dunha conta corrente a unha conta de aforro. Esta transacción inclúe dúas accións distintas: unha resta no saldo da conta corrente e unha suma na conta de aforro. ¿Que pasaría se despois de executar a primeira operación prodúcese un fallo e non se chega a completar a segunda operación?.... Ó agrupar as dúas operacións nunha transacción garántese que ou se executan as dúas operacións, ou non se executa ningunha.

Unha transacción pode finalizar de dúas maneiras distintas:

- **Con éxito:** nese caso as actualizacións de que consta a transacción se gravan (commit), e dicir, se fan permanentes.
- **Con fracaso:** nese caso ten que ser restaurado o estado inicial no que se atopaba a base de datos antes de que empezara a executarse a transacción. As actualizacións de que consta a transacción deberán, polo tanto, desfacerse (rollback).

Os SXBD xestionan as transaccións de forma implícita ofrecendo ademais aos usuarios unha serie de instrucións para xestionalas dunha maneira explícita.

Para controlar a execución simultánea de transaccións é necesario unha boa xestión de acceso concorrente que controle que a execución paralela de transaccións dá o mesmo resultado que unha execución secuencial. Para controlar a execución paralela, a solución clásica consiste en bloquear momentaneamente os datos utilizados por unha transacción ata que finalice a ac-

tualización e as demais transaccións que demandan estes datos, cando finaliza a transacción, quedan desbloqueados.

A consecuencia de utilizar as técnicas de bloqueo é o risco do interbloqueo (chamado bloqueo mortal), que sucede cando dúas ou mais transaccións se encontran á espera de que queden liberados datos que ten bloqueados outra transacción que espera polos datos que están bloqueados polas transaccións anteriores. O SXBD debe estar capacitado para detectar esta situación e desbloqueala.

Propiedades das transaccións

As propiedades principais que deben cumprir as transaccións se coñecen polas siglas ACID (Atomicity, Consistency, Isolability, Durability), que fan referencia a:

- **Atomicidade:** no sentido que xa temos especificado anteriormente de que, ou ben se executan todas as sentencias dunha transacción, ou ningunha.
- **Mantemento da consistencia:** a execución dunha transacción ten que deixar a base de datos nun estado consistente.
- **Illamento:** xa que unha transacción non mostra os cambios que produce ata que finaliza.
- **Persistencia:** Unha vez que a transacción finaliza con éxito, os seus efectos perduran na base de datos.

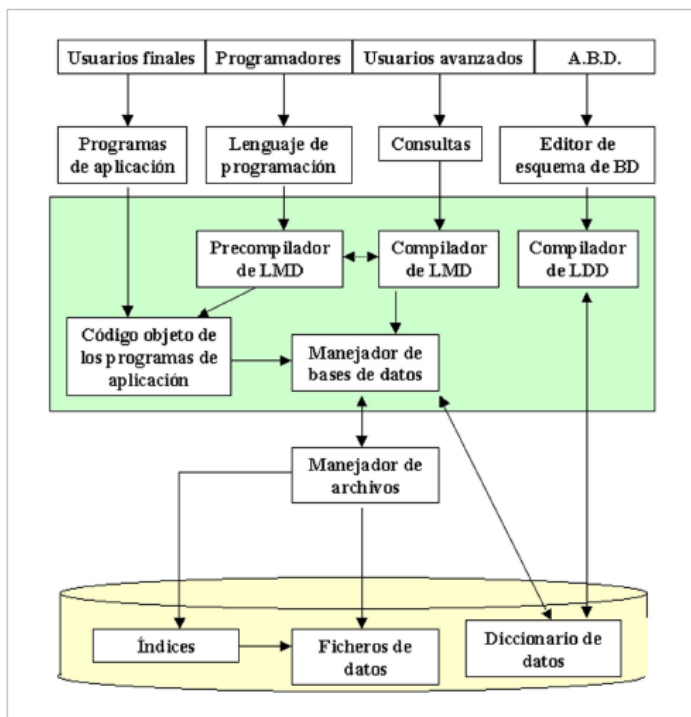
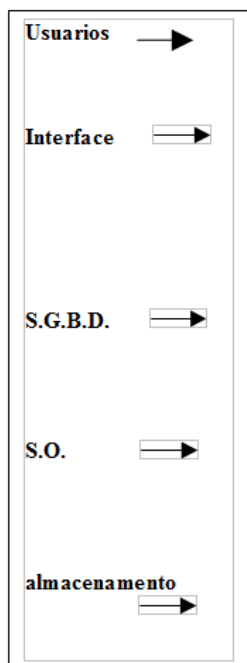
1.3.5 Outras funcións dos Sistemas Xestores de Bases de Datos

Outras funcións que deben realizar os SXBD son:

- Garantir un tempo de resposta idóneo para o diálogo entre os usuarios e a base de datos nos procesos en liña.
- Capacidade de auditoría. Para o administrador da base de datos é moi importante coñecer quen accede á base de datos e que operacións realiza. Un bo SXBD debe poder proporcionarlle esa información.

1.4 Compoñentes dun Sistema Xestor de Bases de Datos

A maior parte das funcións dun sistema de bases de datos son realizadas polos distintos módulos nos que se divide un SXBD, pero existen algunhas funcións elementais que deben ser feitas polo sistema operativo. Por exemplo, a base de datos e o dicionario de datos almacénanse en disco; o acceso ó disco é controlado polo sistema operativo que programa a entrada-saída do disco; o sistema xestor de bases de datos interactúa co sistema operativo cando se require acceso ó disco.. Se moitos usuarios comparten o mesmo sistema, o sistema operativo programará as distintas solicitudes de acceso ó disco, incluíndo as do sistema xestor de bases de datos.



O almacenamento físico está composto de:

- **Ficheiros de datos:** gardan os datos da base de datos.
- **Diccionario de datos:** é onde se describe toda a información almacenada na base de datos. É dicir, "contén datos acerca dos datos" (metadatos). En xeral, este diccionario de datos pode ser visto como unha base de datos que contén as definicións de outros obxectos do sistema, en lugar de datos. En especial, os esquemas e transformacións, ademais das restricións de seguridade e de integridade. O diccionario de datos é fundamental.
- **Índices:** estruturas que permiten acceder rapidamente ós datos.
- **Datos estatísticos:** información estatística dos datos da B.D. O procesador de consultas usa esa información para xerar consultas eficientes, mediante o optimizador de consultas.

Os compoñentes do SXBD pódense clasificar de forma moi simplificada en:

- **Compilador LMD** (linguaxe de manipulación de datos): traduce as instrucións na linguaxe de consultas a instrucións de baixo nivel. Este linguaxe é o que entende o motor da avaliación de consultas. Non só traduce se non que optimiza as peticións.
- **Precompilador LMD:** converte as instrucións da LMD incorporadas nun programa a chamadas a procedementos normais da linguaxe anfitrión. Interactúa co compilador da LMD para xerar o código.
- **Intérprete LDD** (linguaxe de definición de datos): interpreta as instrucións LDD e gárdasas no diccionario de datos, nas táboas correspondentes.
- **Manexador da base de datos** (motor), que é o responsable das seguintes tarefas:
 - Avaliación de consultas. Executa as instrucións xeradas polo compilador da LMD.
 - Xestión de seguridade e de autorizacións. Comproba a corrección dos datos e verifica a autorización dos usuarios para acceder ós datos.

- Implantación da integridade. O manexador da bases de datos comprobará as restricións de integridade cada vez que se vai facer unha actualización.
- Interacción co manexador de ficheiros. Os datos sen procesar almacénanse no disco mediante o sistema de ficheiros proporcionado polo sistema operativo. O manexador de bases de datos traduce as diferentes peticións en LMD a comandos do sistema de arquivos. Así o manexador da base de datos encárgase realmente do almacenamento, recuperación e actualización dos datos da base de datos.
- **Xestor de memoria intermedia:** pasa os datos dos dispositivos de almacenamento secundario á memoria principal e indica cales deles convén conservar na memoria caché.
- **Xestor de transaccións:** verifica que a base de datos estea sempre nun estado consistente a pesar dos erros ó gravar novos datos e tamén se encarga de xestionar os accesos concurrentes ós datos.

Dunha forma resumida, podemos dicir que o SXBD recibe as peticións dos usuarios, que se comunican con el mediante unha interface adecuada ó seu nivel de coñecementos no manexo de bases de datos e da súa responsabilidade. Procesa estas peticións optimizando a forma de acceder ós datos e solicita ó sistema operativo que se transfiran eses datos dende o disco á memoria principal para poder envialos ós usuarios que os solicitaron.

1.5 Tipos de Sistemas Xestores de Bases de Datos

1.5.1 En función do modelo de datos

En función do modelo de datos empregado, os SXBD pódense clasificar en:

- **Xerárquico.**- Son aqueles que empregan o modelo de datos xerárquico. O SXBD xerárquico máis coñecido no mercado é IMS que utiliza a linguaxe de consulta DL/I.
- **En rede.**- Son aqueles que empregan o modelo de datos en rede. O modelo en rede mais seguido é Codasyl. Algúns SXBD Codasyl existentes no mercado son: DMS 1100, IDMS e IDS.
- **Relacional.**- Son aqueles que empregan o modelo de datos relacional. Algúns dos SXBD relacionais existentes son: Oracle, Informix, Sybase, Ms SQL Server.
- **Orientados a obxectos.**- Son aqueles que empregan o modelo de datos orientado a obxectos. Neste tipo de SXBD compre mencionar que hoxe en día existen sistemas que implementan unha solución intermedia entre o modelo orientado a obxectos e o modelo relacional xa que permiten o almacenamento de obxectos inda que o funcionamento interno segue sendo relacional, este tipo de SXBD son chamados obxecto-relacionais.
- **NoSQL.**- Son aqueles que empregan algún dos modelos de datos NoSQL.

1.5.2 En función do tipo de licenza

En función do tipo de licenza para a súa utilización, os SXBD pódense clasificar en:

- **Software libre.** Son aqueles SXBD que, tal e como mandan as directrices do software libre, poden ser copiados, estudados, modificados, utilizados libremente con calquera fin e redis-

tribuídos con ou sen cambios ou melloras. Por exemplo: MySQL, PostgreSQL, DB2 ExpressC , ..

- **Software propietario.** Son aqueles SXBD nos que a redistribución ou modificación dos mesmos está prohibida e o seu uso require permiso expreso do titular do software, normalmente mediante a adquisición dunha licenza. Por exemplo: Oracle, MS SQL Server, Informix e DB2 de IBM.
- **Software propietario, pero gratuíto.** Son aqueles SXBD nos que a redistribución ou modificación dos mesmos está prohibida e o seu uso require permiso expreso do titular do software mediante unha licenza, neste caso gratuíta. Nesta categoría adoitan atoparse versións limitadas de SXBD propietarios, como por exemplo Microsoft SQL Express ou Oracle Express Edition.

1.5.3 En función do número de usuarios

Outro criterio para clasificar os SXBD é o número de usuarios aos que dá servizo o sistema. Podémolos clasificar en:

- **Sistemas monousuario.** Son aqueles que só poden atender a un usuario á vez. O seu principal uso dáse nos ordenadores persoais.
- **Sistemas multiusuario.** Son aqueles que poden atender a varios usuarios ao mesmo tempo. A maior parte dos SXBD pertencen a este grupo.

1.5.4 En función da distribución dos seus compoñentes

En función da distribución dos seus compoñentes, os SXBD pódense clasificar en:

- **Centralizados.** Son aqueles nos que tanto o propio SXBD como as bases de datos que xestiona residen por completo nunha soa máquina. Isto non evita que poidan atender a múltiples usuarios á vez.
- **Distribuídos.** Son aqueles nos que as bases de datos e o propio software do SXBD poden estar distribuídos en varios sitios conectados por unha rede. Os SXBD distribuídos homoxéneos utilizan o mesmo SXBD en múltiples sitios. Unha tendencia recente consiste en crear software para ter acceso a varias bases de datos autónomas preexistentes almacenadas en SXBD distribuídos heteroxéneos. Isto dá lugar aos SXBD federados ou sistemas multibase de datos nos que os SXBD participantes teñen certo grado de autonomía local. Moitos SXBD distribuídos empregan unha arquitectura cliente-servidor.

A distribución dos datos nun SXBD distribuído

Unha das decisións máis importantes que o deseñador de bases de datos distribuídas debe tomar é o posicionamento dos datos no sistema e o esquema baixo o cal o desexa facer. Para isto existen catro alternativas principais: centralizada, replicada, particionada ou fragmentada, e híbrida.

- **Centralizada.** É moi similar ao modelo de Cliente/Servidor no sentido que a base de datos está centralizada nun lugar e os usuarios están distribuídos. Este modelo só brinda a vantaxe de ter o procesamento distribuído xa que en sentido de dispoñibilidade e fiabilidade dos datos non se gaña nada.

- **Replicadas.** O esquema de BDD de replicación consiste en que cada nodo debe ter a súa copia completa da base de datos. É fácil ver que este esquema ten un alto custo no almacenamento da información. Debido a que a actualización dos datos debe ser realizada en todas as copias, tamén ten un alto custo de escritura, pero todo isto vale a pena si temos un sistema no que se vai a escribir poucas veces e ler moitas, e onde a dispoñibilidade e fiabilidade dos datos sexa de máxima importancia.
- **Particionadas ou fragmentadas.** Este modelo consiste en que só hai unha copia de cada elemento, pero a información está distribuída a través dos nodos. En cada nodo alóxase un ou máis fragmentos disxuntos da base de datos. Como os fragmentos non se replican isto diminúe o custo de almacenamento, pero tamén sacrifica a dispoñibilidade e fiabilidade dos datos. Algo que se debe tomar en conta cando se desexa implementar este modelo é a **política de fragmentación da información**. A fragmentación pódese realizar de tres formas:
 - **Horizontal:** Os fragmentos son subconxuntos das filas ou tuplas dunha táboa (restrinxir).
 - **Vertical:** Os fragmentos son subconxuntos das columnas ou atributos dunha táboa (proxectar).
 - **Mixto:** Almacénanse fragmentos produto de restrinxir e proxectar unha táboa.

Unha vantaxe significativa deste esquema é que as consultas (SQL) tamén se fragmentan polo que o seu procesamento é en paralelo e máis eficiente, pero tamén se sacrifican certas operacións que involucren varios fragmentos da base de datos (como poden ser os produtos cartesianos ou os *joins*).

Para que unha fragmentación sexa correcta debe cumprir coas seguintes regras:

- **Debe ser Completa:** Si unha relación R se fragmenta en R_1, R_2, \dots, R_n , cada elemento de datos de R debe estar nalgún R_i .
- **Debe ser Reconstruible:** Debe ser posible definir unha operación relacional que a partir dos fragmentos (R_i) obteña de novo a relación R .
- **Os fragmentos deben ser Disxuntos:** Si a fragmentación é horizontal, un elemento que está en R_i non pode estar en R_k (para k distinto a i). Se a fragmentación é vertical, é necesario que se repitan as claves primarias para o conxunto de atributos que non son clave primaria.
- **Híbrida.** Este esquema simplemente representa a combinación do esquema de partición e replicación. Particiónase a relación e á vez os fragmentos están selectivamente replicados a través do sistema de BDD.