

Introducció als sistemes de gestió de bases de dades

Rafael Camps Paré

P03/05053/02048


Índex

Introducció	5
Objectius	5
1. Funcionalitat i objectius dels SGBD	7
1.1. Consultes no predefinides i complexes	7
1.2. Flexibilitat i independència	7
1.3. Integritat de les dades	8
1.4. Concurrència d'usuaris	9
1.5. Seguretat	9
1.6. Altres objectius	10
2. Arquitectura de tres nivells dels SGBD	11
2.1. Esquemes i nivells	11
2.2. Independència de les dades	13
3. Models de BD	14
4. Modalitats d'ús del llenguatge	16
5. Arquitectura externa i interna	18
6. Els SGBD comercials	21
Resum	23
Activitats	25
Exercicis d'autoavaluació	25
Solucionari	26
Exercicis d'autoavaluació	26
Glossari	26
Bibliografia	27

Introducció als sistemes de gestió de bases de dades

Introducció

En començar aquesta assignatura partim del supòsit que l'estudiant ja sap què són els sistemes de gestió de bases de dades (SGBD) i en coneix les característiques principals. Aquests continguts es van desenvolupar a l'assignatura *Bases de dades I*. En aquest mòdul didàctic, fem una breu introducció als SGBD, revisem breument alguns dels temes que ja vam explicar amb detall i n'introduïm de nous.



Els conceptes que es repassen en aquest mòdul els trobareu al mòdul "Introducció a les bases de dades" de l'assignatura *Bases de dades I*.

Objectius

Com que cal que l'estudiant recordi i complementi els coneixements generals que va adquirir en l'assignatura *Bases de dades I*, en els materials didàctics que es faciliten en aquest mòdul trobarà les eines indispensables per a assolir els objectius següents:

1. Conèixer els objectius principals dels SGBD.
2. Distingir entre els models de BD més utilitzats.
3. Reconèixer els avantatges que ofereixen les diferents modalitats d'ús del llenguatge SQL.
4. Saber identificar els trets principals de l'arquitectura dels SGBD del mercat.
5. Poder distingir els components més habituals d'un SGBD.
6. Identificar les tendències dels SGBD actuals.

1. Funcionalitat i objectius dels SGBD

Una **base de dades (BD)** és un conjunt estructurat de dades que representa entitats i les seves interrelacions. Aquesta representació informàtica integrada ha de poder ser consultada i actualitzada de forma compartida per molts usuaris de tipus diversos. El **sistema de gestió de bases de dades (SGBD)** és el programari especialitzat que en facilita la utilització, un programari complex que actualment constitueix una part essencial de tot sistema d'informació (SI).

Tot seguit recordarem els objectius i la funcionalitat dels SGBD.

1.1. Consultes no predefinides i complexes

Els usuaris poden fer consultes de qualsevol tipus i complexitat directament a l'SGBD. Aquest ha de respondre immediatament sense que estiguin preestablertes, és a dir, sense que s'hagi d'escriure, compilar i executar un programa específic per a cada consulta.

L'usuari ha de poder formular la consulta en un llenguatge i el sistema l'ha d'interpretar directament. Però també s'ha de poder escriure programes amb consultes incorporades. El llenguatge que es fa servir universalment per a assolir aquest doble objectiu (consultes no predefinides i complexes) és el llenguatge SQL, que ja coneixeu.

Vegeu el llenguatge SQL al mòdul "El llenguatge SQL" de l'assignatura *Bases de dades I*.

1.2. Flexibilitat i independència

Interessa obtenir la màxima independència entre les dades i els processos usuaris. Cal que es pugui fer tot tipus de canvis tecnològics i canvis en la descripció de la base de dades (BD) sense que s'hagi de modificar els programes d'aplicació ja escrits ni variar la manera de fer les consultes directes.

Diem que hi ha independència física de les dades quan els usuaris que hi fan consultes o actualitzacions (tant si són usuaris directes com si són programadors d'aplicacions) no necessiten conèixer les característiques físiques de la BD amb què treballen: suport físic, sistema operatiu (SO) emprat, índexs, etc. Grà-

cies a la independència física podem fer canvis de tecnologia i canvis físics per a millorar el rendiment sense afectar cap usuari.

També volem que els usuaris no hagin de fer canvis quan es modifica la descripció lògica o l'esquema de la BD; per exemple, quan s'hi afegeixen/suprimeixen entitats o interrelacions, atributs, etc. I encara més: volem que els diferents processos usuaris puguin tenir diferents visions lògiques d'una mateixa BD i que aquestes visions es puguin mantenir al màxim d'independents de la BD i entre si. Aquest tipus d'independència, que dóna flexibilitat i elasticitat als canvis lògics, s'anomena **independència lògica de les dades**.

1.3. Integritat de les dades

Els SGBD hauran d'assegurar el manteniment de la qualitat de les dades en qualsevol circumstància.

Quan tenim redundància a la BD, les actualitzacions poden provocar inconsistències o incoherències entre les dades. Convindrà, doncs, evitar-la. Però sovint ens interessa tenir-ne per raons de fiabilitat, disponibilitat, rendiment o costos de comunicacions. Per això els SGBD han de permetre que el dissenyador defineixi dades redundants, però llavors haurà de ser l'SGBD mateix el que faci automàticament l'actualització de les dades a tots els llocs on estiguin repetides.

La redundància no és l'única causa possible de pèrdua d'integritat de les dades. La correcció o la consistència de les dades es pot perdre per moltes altres raons: errades de programes, errades d'operació humana, avaria de disc, transaccions incompletes per tall de l'alimentació elèctrica, etc.

Podrem declarar per a l'SGBD un conjunt de **regles d'integritat**, o restriccions, per garantir que els programes mantindran la qualitat de les dades.

Quan l'SGBD detecta que un programa, o un usuari, intenta fer una operació que va contra les regles declarades en definir la BD, no li ho ha de permetre, i li ha de tornar un estat d'error. En dissenyar una BD per a un SI concret i escriure'n l'esquema no definirem solament les dades, sinó també les regles d'integritat.

El concepte *integritat de les dades* va més enllà de la prevenció que els programes, o els usuaris, emmagatzemin dades incorrectes. En casos d'errades o desastres també podríem perdre la integritat de les dades. L'SGBD ens ha de donar les eines per a poder reconstruir o restaurar les dades malmeses.

Els **processos de restauració*** de què tot SGBD disposa poden reconstruir la BD fins a un estat consistent, correcte, anterior a l'incident. Això s'acostuma a fer gràcies a l'obtenció de còpies periòdiques de les dades, que se solen anomenar **còpies de seguretat****, i al manteniment continu d'un **diari***** en el qual l'SGBD anota tots els canvis que es fan a la BD.

* En anglès, *restore o recovery*.

** En anglès, *backups*.

*** En anglès, *log*.

1.4. Concurrència d'usuaris

Un objectiu fonamental dels SGBD és permetre que un nombre elevat d'usuaris puguin accedir concurrentment a la mateixa BD.

Quan els accessos concurrents són tots de lectura, ens enfrontem simplement a un problema de rendiment causat per les limitacions físiques dels suports de què es disposa. Però si un o més d'un usuari actualitzen les dades, es poden produir **problemes d'interferència**, els quals poden donar lloc a l'obtenció de dades errònies i a la pèrdua d'integritat de la BD. Per a tractar els accessos concurrents, els SGBD fan servir el concepte de *transacció*.

Una **transacció** consisteix en un conjunt d'operacions simples que s'executen com una unitat indivisible. Els SGBD han d'aconseguir fer que el conjunt d'operacions d'una transacció mai no s'executi parcialment: o s'executen totes o no se n'executa cap.

Entre les transaccions que s'executen concurrentment es poden produir problemes d'interferència que facin obtenir resultats erronis o facin perdre la integritat de les dades. Per a evitar-ho es demana que l'SGBD tracti cada transacció com un procés aïllat dels altres.

1.5. Seguretat

En el camp dels SGBD, el terme *seguretat* s'acostuma a utilitzar per a fer referència als temes relatius a la confidencialitat, la privacitat, les autoritzacions, els drets d'accés, etc.

Els SGBD permeten definir autoritzacions o drets d'accés per categories: la global de tota la BD, la d'entitat i fins i tot la d'atribut. Aquests mecanismes de

seguretat es basen en la identificació de l'usuari, que s'efectua mitjançant els codis d'usuari (i grups d'usuaris) acompanyats de contrasenyes*.

* En anglès, *passwords*.

Per a tenir un grau de seguretat més elevat quan es vol mantenir les dades en secret, molts dels SGBD actuals tenen previst el xifratge* de les dades que emmagatzemen.

* En anglès, *encryption*.

1.6. Altres objectius

A mesura que els SGBD evolucionen, els dissenyadors es plantegen nous objectius per a adaptar-los a noves necessitats i a noves tecnologies. En aquests moments podríem citar com a objectius més recents els següents: !

!
Podeu veure les tendències actuals en SGBD al subapartat 2.4 del mòdul "Introducció a les bases de dades" de l'assignatura *Bases de dades I*.

- 1) Servir eficientment els *data warehouse*.
- 2) Adaptar-se al desenvolupament orientat a l'objecte.
- 3) Incorporar el temps com un element de caracterització de la informació.
- 4) Adaptar-se al món d'Internet.

2. Arquitectura de tres nivells dels SGBD

Com que ja coneixeu l'arquitectura ANSI/SPARC de tres nivells, ara només en recordem breument els aspectes més destacats.

2.1. Esquemes i nivells


Els SGBD necessiten que els donem una descripció o definició de la BD. Aquesta descripció rep el nom d'**esquema de la BD**.

L'esquema de la BD és un element fonamental de l'arquitectura d'un SGBD que permet independitzar l'SGBD de la BD. Per exemple, ens permet canviar el disseny de la BD, l'esquema, sense haver de fer cap canvi a l'SGBD.

La distinció clàssica entre dos nivells d'abstracció, el nivell lògic i el nivell físic, és ampliada per l'arquitectura ANSI/SPARC a tres nivells d'abstracció, tres nivells d'esquemes.

Vegeu els nivells lògic i físic al subapartat 3.7 del mòdul "Les dades: conceptes introductoris" de l'assignatura Bases de dades I.

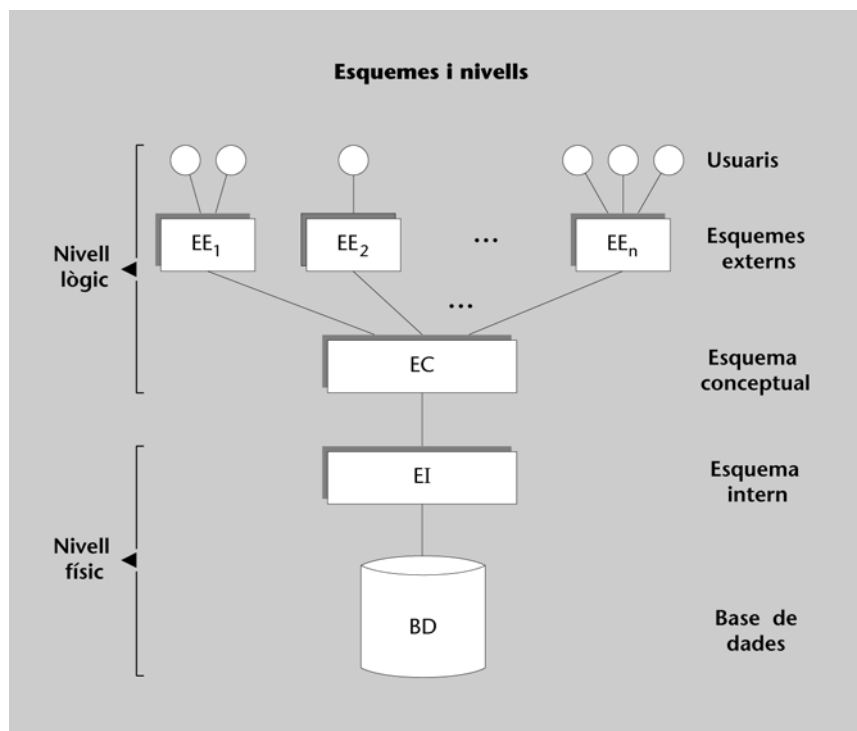
La idea bàsica de l'arquitectura ANSI/SPARC consisteix a descompondre el nivell lògic en dos: el **nivell extern** i el **nivell conceptual**. En aquesta arquitectura anomenem **nivell intern** allò que havíem anomenat *nivell físic*.

Així, doncs, d'acord amb l'arquitectura ANSI/SPARC, hi hauria els tres nivells d'esquemes que podeu veure en el gràfic de la pàgina següent: 

a) En el nivell conceptual hi ha una sola descripció lògica bàsica, única i global, que anomenem **esquema conceptual**, i que serveix de referència per a la resta d'esquemes.

A l'esquema conceptual es descriuen les entitats tipus, els seus atributs, les interrelacions i també les restriccions o regles d'integritat.

b) En el nivell extern se situen les diferents visions lògiques que els diferents processos usuaris tindran de les parts de la BD que utilitzaran. Aquestes visions s'anomenen **esquemes externs**.



Cada aplicació podrà tenir la seva visió particular, i segurament parcial, de l'esquema conceptual: els usuaris veuran la BD per mitjà d'esquemes externs apropiats a les seves necessitats. Aquests esquemes es poden considerar redefinicions de l'esquema conceptual, amb les parts i els termes que convinguin per a les necessitats de les aplicacions (o grups d'aplicacions). Alguns sistemes els anomenen *subesquemes*.

Els usuaris poden ser programes o usuaris directes.

En definir un esquema extern se citaran només aquells atributs i entitats que interessin, els podrem reanomenar, podrem definir dades derivades, podrem redefinir una entitat perquè les aplicacions que usen aquest esquema extern creguin que en són dos o podrem definir combinacions d'entitats perquè en semblin una de sola, etc.

c) En el nivell físic hi ha una descripció física única, que anomenem **esquema intern**.

L'esquema intern o físic contindrà la descripció de l'organització física de la BD: camins d'accés (índexs, *hashing*, apuntadors, etc.), codificació de les dades, gestió de l'espai, mida de la pàgina, etc.

Esquemes i nivells als SGBD relacionals

En els SGBD relacionals, és a dir, en el món de l'SQL, s'ha fixat una terminologia lleugerament diferent de l'habitual. No se separen clarament els tres nivells de descripció.

Es parla d'un sol **esquema** (*schema*), però dintre d'aquest s'inclouen descripcions de tots tres nivells. A l'*schema* es descriuen els elements d'allò que a l'arquitectura ANSI/SPARC s'anomena *esquema conceptual* (entitats tipus, atributs i restriccions) més les *vistas* (*views*), que corresponen aproximadament als esquemes externs.

El model relacional en què s'inspira l'SQL es limita a moure's en el món lògic. Per això l'estàndard ANSI-ISO de l'SQL no parla en absolut del món físic o intern; ho deixa en mans dels SGBD relacionals del mercat. Però aquests donen la possibilitat d'incloure en l'*schema* descripcions d'estructures i característiques físiques (*index*, *tablespace*, *cluster*, espais per a excedents, etc.).

2.2. Independència de les dades

L'arquitectura de tres nivells ens proporciona tots dos tipus d'independència de les dades: la independència física i la independència lògica. El canvi de dades d'un suport a un altre, o de lloc dintre d'un suport, no afecta els programes d'aplicació ni els usuaris directes perquè no modifica l'esquema conceptual ni els esquemes externs, com tampoc no els haurà d'afectar el canvi del mètode d'accés a uns registres determinats*. El canvi de format o de codificació hauria d'afectar la BD física i l'esquema intern, però no el món exterior.

Vegeu la independència física i la independència lògica al subapartat 1.2 d'aquest mòdul didàctic.

* Per exemple, eliminant un índex en arbre-B o substituint-lo per un *hashing*.

Si hi ha independència física de les dades, l'única cosa que canvia en modificar l'esquema intern són les correspondències entre l'esquema conceptual i l'intern. Òbviament, la majoria dels canvis de l'esquema intern obliguen a refer la BD real (física).

Amb relació a la independència lògica, com que en l'arquitectura ANSI/SPARC hi ha dos nivells lògics diferents, distingim les dues situacions següents: !

a) Canvis en l'esquema conceptual: aquest tipus de canvis no afectaran els esquemes externs que no facin referència a les entitats o als atributs modificats.

b) Canvis en els esquemes externs: efectuar un canvi en un esquema extern afectarà els usuaris que facin servir els elements modificats, però no hauria d'afectar els altres ni l'esquema conceptual i, en conseqüència, tampoc no hauria d'incidir en l'esquema intern ni en la BD física.

3. Models de BD

Com ja sabeu, el component fonamental per a modelitzar en un SGBD relacional són les taules o relacions, però hi ha certs tipus d'SGBD que utilitzen altres components.

El conjunt de components, o eines conceptuals, que un SGBD proporciona per a modelitzar rep el nom de **model de BD**.

Vegeu el model relacional al mòdul "El model relacional i l'àlgebra relacional" de l'assignatura Bases de dades I.

El **model relacional**, que ja coneixeu, és el que s'utilitza habitualment. Abans del model relacional van aparèixer els models jeràrquic i en xarxa. Així com en el model relacional hi ha un sol element estructural, la taula, que és formada per files i columnes, en aquests models prerelacionals hi ha dos elements bàsics: els registres i les interrelacions.


A les BD prerelacionals, les interrelacions entre els registres eren representades per punters (més o menys físics) i el programador emprava aquestes interrelacions explícitament. Aquest tipus de programació s'anomena *programació navegacional* perquè el programa explicita els camins (punter) que vol recórrer per a obtenir els conjunts de dades que l'interessen. En canvi, a les BD relacionals no s'indica el camí, sinó l'element que es vol obtenir. Aleshores es pot dir que es tracta d'una *programació declarativa*.

Comparació entre els models prerelacionals i el model relacional

Analitzem, per exemple, una BD amb informació dels departaments d'una empresa i amb informació dels empleats d'aquesta empresa. En un SGBD relacional que utilitzés SQL per a indicar que entre els departaments i els empleats hi ha la interrelació *emple at_del_dept* posaríem un atribut (clau forana) a la taula d'empleats amb el valor del codi del departament on treballa. Per a mostrar el nom del departament de cada empleat hauríem de fer una combinació (*join*) declarant la condició següent:

```
dept.codidept = empl.codidept;
```

En una BD jeràrquica o en xarxa a l'esquema de la BD hauríem d'explicitar la interrelació *empleat_del_dept* i posar-li un nom, i el programa usuari hauria de seguir per a cada empleat el camí *empleat_del_dept* per a trobar el nom del departament on treballa.

La descripció de les interrelacions, l'esquema i l'ús explícit d'aquestes pels programes fa que als SGBD prerelacionals hi hagi menys independència de dades que als sistemes relacionals. 

Al **model jeràrquic** els registres estan interrelacionats en forma d'arbres. L'SGBD clàssic d'aquest model és l'IMS/DL1 d'IBM.

Al **model en xarxa** els registres no estan limitats a ser “fills” d’un sol registre tipus. El comitè CODASYL-DBTG va proposar un estàndard basat en aquest model i ja definia tres nivells d’esquemes.

La diferència més important entre els models prerelacionals i el model relacional és que aquest es limita al nivell lògic i no fa absolutament cap consideració sobre les representacions físiques; és a dir, ens dóna una independència total física de les dades.

Aquests últims anys s’estén el **model de BD relacional amb objectes**. Es tracta d’una ampliació del model relacional en què s’afegeix la possibilitat que els tipus de dades siguin tipus abstractes de dades (TAD) definibles per l’usuari informàtic amb una estructura tan complexa com es vulgui. Les instàncies d’un TAD són els objectes.

Els tipus de dades que es poden definir als SGBD relacionals dels anys vuitanta i noranta són molt limitats. La incorporació de multimèdia –imatge i so– als SI obliga a fer que els SGBD relacionals acceptin atributs d’aquests tipus. Però en algunes aplicacions no n’hi ha prou d’incorporar tipus especialitzats en multimèdia: també necessiten tipus complexos definibles pel desenvolupador a la mida de l’aplicació.

Exemple de tipus especialitzat multimèdia

A l’entitat *alumne* ens pot interessar tenir-hi un atribut *foto* tal que el seu valor sigui una tira de bits llarguíssima resultat de la digitalització de la fotografia de l’alumne.

En definitiva, es necessiten tipus abstractes de dades, TAD. Els SGBD més recents ja incorporen aquesta possibilitat i ja s’ha iniciat un ampli mercat de TAD predefinitos o llibreries de classes.

Això ens porta a l’**orientació a l’objecte (OO)**. L’èxit de la OO al final dels anys vuitanta en el desenvolupament de programari bàsic, aplicacions al disseny en enginyeria, i en la construcció d’interfícies gràfiques d’usuari, ha fet que als anys noranta s’hagi estès a gairebé tots els camps de la informàtica.

En els SI s’inicia també l’adopció, tímida de moment, de la OO. La utilització de llenguatges com el C++ o el Java comporta que actualment els SGBD relacionals s’hi adaptin amb interfícies escaients. Aquesta incorporació dels objectes als sistemes relacionals els apropa al paradigma de la OO. Els primers SGBD relacionals que han donat aquesta possibilitat han estat Oracle (versió 8), Informix (versió 9) i IBM/DB2/UDB (versió 5).

Models de dades i models de BD

El que en aquest apartat hem anomenat *models de BD* s’acostumen a anomenar *models de dades*, ja que permeten modelitzar les dades. Però hi ha models de dades que no són usats pels SGBD del mercat; com que no disposen d’operacions, no es fan servir a les realitzacions, sinó només durant els processos d’anàlisi i disseny. El més conegut d’aquests tipus de models de dades és el model entitat relació (*entity-relationship*), o model ER, que ja coneixeu.

4. Modalitats d'ús del llenguatge

El llenguatge SQL és el llenguatge universal de comunicació amb els SGBD per a crear, consultar i mantenir les BD. Fins ara heu utilitzat l'SQL de manera directa, interactiva, sense escriure cap programa. L'SGBD disposa d'un interpretador que analitza, verifica i executa les instruccions* que li donem directament des d'un teclat. Però no és aquesta la modalitat més habitual d'ús del llenguatge SQL. El més habitual serà que vulguem fer servir l'SQL des de dintre d'un programa.

* Per exemple, SELECT, CREATE TABLE, etc.

Si volem escriure un programa d'aplicació que treballi amb BD, segurament ens interessarà fer servir el nostre llenguatge habitual de programació*. Però aquests llenguatges normalment no tenen instruccions pròpies per a les BD, per tant voldrem poder emprar l'SQL des de dintre del programa. Podrem escollir entre les dues modalitats d'SQL següents:

* Llenguatges com el Pascal, el C, el Cobol, el Basic, el Fortran, el Java, etc.

1) Les **crides a funcions**: al mercat hi ha llibreries de funcions especialitzades a demanar al servidor de BD l'execució d'instruccions SQL*. Només cal incloure dintre del programa, escrit en el llenguatge habitual, crides a les funcions apropiades. Normalment a aquestes funcions se'ls dona com a paràmetre la instrucció d'SQL que es vol executar. Seran les funcions les que s'encarregaran de donar les instruccions d'SQL a l'SGBD en temps d'execució.

* Com ara les llibreries ODBC o les JDBC.

2) L'**SQL hostatjat** consisteix a incloure directament les instruccions de llenguatge SQL en el nostre programa. Però això exigeix disposar d'un precompilador especialitzat que accepti dintre del nostre llenguatge de programació habitual les instruccions de l'SQL. Llavors es diu que l'SQL és hostatjat o incorporat* al nostre llenguatge de programació (Pascal, C, Cobol, Java, etc.), que fa de llenguatge amfitrió**. Tenim les dues variants d'SQL hostatjat que presentem a continuació:

* En anglès, *embedded*.
** En anglès, *host*.

a) L'**estàtic**: amb aquesta variant el programador escriu explícitament en el seu programa les instruccions que vol que s'executin.

b) El **dinàmic**: amb aquesta variant el programador pot escriure parcialment les instruccions que s'han d'executar i deixar-ne parts per a ser completades en temps d'execució del programa. L'SQL hostatjat dinàmic permet, doncs, escriure programes genèrics* que dialoguen amb un usuari final per a determinar què han de fer. Quan s'escriu el programa no se sap exactament quines seran les instruccions que s'hauran d'enviar a l'SGBD, i es concretaran en temps d'execució, ja que depenen del que demani l'usuari del programa. La modalitat d'SQL hostatjat estàtic té l'avantatge d'estalviar temps d'execució, ja que la verificació i la traducció es fan en temps de compilació.

* Com ara les eines d'accés visual a BD.

Fixem-nos que la primera de les modalitats (crides o funcions) és dinàmica, ja que el programa passa les instruccions SQL com el valor d'un paràmetre* i, per tant, les pot construir en temps d'execució.

* Per exemple, en forma de literal.

En un altre mòdul s'estudia com es pot accedir a una BD des d'un llenguatge de programació de propòsit general.

5. Arquitectura externa i interna

L'**arquitectura general dels SGBD** és determinada per tres elements bàsics: l'SGBD centralitzat o client/servidor (C/S), les dades centralitzades o distribuïdes i l'aprofitament o no de plataformes amb paral·lelisme. A continuació expliquem en què consisteixen aquests elements:

1) Com ja sabem, els SGBD han evolucionat des d'una arquitectura centralitzada a una de tipus C/S. Recordem que aquestes arquitectures es caracteritzen per les particularitats següents:

a) L'**arquitectura centralitzada** consta d'un sol ordinador, que pot estar connectat a una xarxa de terminals. En aquest ordinador s'executa l'SGBD i hi resideix la BD.


b) L'**arquitectura C/S** té una part de l'SGBD als ordinadors clients (la interfície amb l'usuari), i el cor de l'SGBD, anomenat **motor**, s'executa al servidor. Hi poden haver molts servidors diferents en una mateixa xarxa.

2) La majoria d'SGBD actuals permeten que una transacció tracti dades distribuïdes, és a dir, que pugui treballar de forma coordinada amb les BD locals de diferents servidors veient el conjunt com si fos una única BD.

3) El motor pot estar més o menys preparat per a aprofitar el paral·lelisme de certes plataformes. Pràcticament tots els SGBD del mercat són capaços d'executar amb un sol processador diversos processos (transaccions i processos del motor). Això es fa utilitzant tècniques multitasca, repartint l'ús del processador. Bastants SGBD poden aprofitar les plataformes multiprocessador per a executar uns quants processos del motor i unes quantes transaccions en paral·lelisme real. Però hi ha pocs SGBD capaços d'anar més lluny, de descompondre una consulta en un conjunt de processos executables en paral·lel. Això últim és el que fan els anomenats **SGBD paral·lels**.

L'informàtic usuari de l'SGBD el veu com un conjunt de parts més o menys ben estructurat. L'**arquitectura externa** o arquitectura aparent de l'SGBD acostuma a estar bastant determinada per raons comercials. Els fabricants modifiquen l'empaquetat o la presentació dels seus productes* d'acord amb la política de vendes, les modes, etc., però de fet no hi ha grans diferències externes entre diferents productes.

En canvi, l'**arquitectura interna**; és a dir, la manera com els SGBD s'estructuren en parts realment és molt diversa. Cada SGBD estructura internament el





Vegeu l'evolució dels SGBD a l'apartat 2 del mòdul "Introducció a les bases de dades" de l'assignatura Bases de dades I.

* Per exemple, canvien els noms dels diferents components.

conjunt de la seva funcionalitat de manera diferent. És més, un mateix SGBD va canviant l'estructura interna en versions successives per a millorarne el rendiment i facilitar-hi l'addició de noves funcionalitats.

El motor de l'SGBD, allò que sempre s'executa al servidor, sol tenir mòduls, com a mínim, per a les funcions següents: la interpretació de l'SQL, l'optimització de consultes, la gestió de vistes, la gestió de memòries intermèdies (*buffers*), la gestió de l'espai de la memòria externa, els mètodes d'accés (per exemple, els índexs), la gestió de processos (multitasca, paral·lelisme), la gestió de transaccions, disparadors i procediments emmagatzemats, el control de l'accés concurrent (per exemple, la gestió de bloquejos), el manteniment de diaris d'actualitzacions (*log*), la creació de còpies de seguretat (*backup*) i restauració/recuperació, i el control de la seguretat/privacitat.

La majoria de les funcions que acabem d'exposar s'estudien amb detall al llarg d'aquesta assignatura. 

Hi ha molt encavalcament entre algunes de les funcions d'un SGBD (les més físiques o bàsiques) i les del sistema operatiu (SO). Per a determinats aspectes com, per exemple, la gestió de memòries intermèdies, la de l'espai de disc, els controls de seguretat o la gestió de processos, els SGBD poden aprofitar (o complementar) la funcionalitat de l'SO o bé resoldre'ls directament. Una raó per a no aprofitar l'SO és perquè l'SGBD ho pot fer més eficientment o per augmentar la confidencialitat. 

Sense SO

Un cas extrem d'invasió del terreny dels SO és l'Oracle Appliance. Es tracta d'un SGBD que es ven incorporat a un maquinari proveït d'un petit subconjunt del nucli d'un SO. Es ven, doncs, un equip amb un SGBD i SO.

El motor de l'SGBD està envoltat de tot un seguit de programes complementaris que se sol considerar part integrant de l'SGBD. Alguns d'aquests programes són eines que s'executen a les màquines client. Els més freqüents són els que enumerem a continuació:

- Precompiladors per a SQL hostatjat en diferents llenguatges de programació*.
- Eines per a participar en xarxes de servidors heterogenis*.
- Eines bàsiques d'administració i eines de control i ajust del rendiment.
- Llibreries de funcions especialitzades per a l'accès a BD*.
- Eines visuals per a accedir fàcilment a la BD sense saber gran cosa d'SQL.

* Com ara Cobol, C, PL/I, Java, etc.

* Com poden ser SGBD de marques diferents.

* Per exemple, ODBC o JDBC.

Però a més de l'SGBD pròpiament dit, els proveïdors ofereixen moltes eines complementàries, generalment amb preu separat. Algunes d'aquestes eines necessiten que el motor incorpori una funcionalitat específica. A part dels proveïdors dels SGBD, hi ha un gran nombre de proveïdors d'eines inde-

pendents. Vegem una llista típica, però no exhaustiva, de tipus d'aquestes eines:

- Eines complementàries per a dissenyar la BD*.
- Llenguatges 4GL, visuals, per a desenvolupar aplicacions amb rapidesa.
- Eines per a administrar la BD*.
- Eines per a interoperar amb altres sistemes del mercat.
- Eines per a aplicacions especials. Per exemple, el processament de text lliure i la recuperació d'informació documental, dades espacials o l'anàlisi i prospecció de dades*.
- Gestor de reproduccions.
- Eines específiques per a la informàtica mòbil*.
- Programari intermediari (middleware) entre clients i servidors per a entorns transaccionals de gran volum i criticitat*.
- Eines per a utilitzar els SGBD com a servidors de pàgines web.

* Per exemple, eines que ajuden a fer el diagrama ER i la seva traducció a un esquem SQL.

* Com ara la supervisió del rendiment, la gestió de còpies de seguretat, la gestió d'usuaris, etc.

* Com ara les aplicacions OLAP, data warehouse o mineria de dades (data mining).

* Com ara ordinadors portàtils de connexió esporàdica a la xarxa.

* Per exemple, monitors de transaccions.

6. Els SGBD comercials

Ja coneixeu una mica la història de l'evolució dels SGBD al llarg dels anys. Recordeu que la majoria dels SGBD relacionals actuals van aparèixer abans de 1985. Actualment hi ha al mercat més d'un centenar de marques diferents d'SGBD relacionals i unes quantes més d'altres tipus d'SGBD (orientats a l'objecte, especialitzats en documents, especialitzats en multimèdia, especialitzats en gestió geogràfica, etc.).

Encara que la funcionalitat bàsica de tots els productes relacionals és molt semblant i que tots es basen en l'SQL, cada producte té les seves particularitats i una arquitectura (interna i externa) diferent, que s'adapta als tipus d'entorns a què va enfocat.

Actualment tots els SGBD del mercat estan pensats perquè es puguin integrar en entorns client/servidor i han de donar servei en situacions d'operació molt diverses.


Hi ha SGBD específics per a ser utilitzats com a eina monousuari sobre un petit ordinador personal. Aquest ordinador pot estar aïllat o bé connectat com a client a una xarxa C/S. En aquest darrer cas, a l'equip client només tindriem la BD local i l'SGBD local, amb el qual també podríem accedir als SGBD remots.

Hi ha servidors de BD molt potents per a donar servei a milers d'usuaris simultàniament. Aquests SGBD, que sovint són especialitzats en paral·lelisme, funcionen generalment sobre SO propietaris*.

Per raons de disponibilitat i preu ens pot interessar tenir, en lloc d'un servidor molt potent, tot un conjunt de servidors de menys potència, potser distribuïts per la xarxa*.

Molts dels SGBD del mercat estan pensats per a poder funcionar en un gran ventall d'entorns i potències.

Aquests tipus d'SGBD es venen amb diferents presentacions o versions. És freqüent disposar de tres versions: una versió monousuari d'ús personal, una multiusuari per a entorns C/S i una altra per a grans servidors multiprocessadors paral·lels.

L'SGBD que els estudiants utilitzeu per a les pràctiques d'aquesta assignatura és una versió monousuari del sistema Informix, sistema que té versions ideades per a funcionar en equips de gran potència amb paral·lelisme massiu. 

Els SGBD més coneguts

Els SGBD relacionals més coneguts són, sense cap mena de dubte, els següents:

- Oracle.
- DB2 d'IBM.
- Informix.
- SQL-Server de Microsoft.

Però n'hi ha molts d'altres prou coneguts com, per exemple, Adaptive Server de Sybase (abans es deia SQL Server i va ser l'origen del de Microsoft) o l'OpenIngres de CA (abans es deia Ingres, i se'l considera el pare de la majoria dels sistemes actuals).

Exemple

Access de Microsoft (encara que seria més correcte dir-ne Jet, que és el nom del motor). És una eina d'accés a bases de dades que pot treballar amb qualsevol gestor de dades que tingui previst l'ODBC (és a dir, amb pràcticament tots).

* Per exemple, DB2/MVS o Teradata de NCR.

* Un exemple de producte per a aquest tipus d'entorns seria l'SQL-Server de Microsoft.

Per exemple,...

... els sistemes Oracle i Informix van ser concebuts inicialment per a plataformes de tipus mitjà, però posteriorment han anat ampliant l'abast cap amunt i cap avall en la gamma de potències.

Òbviament, hi ha una certa relació entre el preu d'un SGBD i la seva potència. Els preus poden anar des de centenars d'euros fins a centenars de milers.

Les **tendències actuals dels SGBD relacionals del mercat** es poden resumir en els punts següents:

- Facilitat de creixement mitjançant el funcionament amb plataformes de paral·lelisme massiu.
- Funcionalitat pensada per a dur a terme anàlisis massives de dades*.
- Ampliació dels tipus de dades i possibilitat que l'usuari informàtic en pugui definir segons les seves necessitats.
- Incorporació al món de l'orientació a l'objecte.
- Incorporació al món d'Internet.
- Incorporació a les arquitectures d'objectes distribuïts*.

* Per exemple, *data warehousing* i prospecció de dades.

* Com ara CORBA o COM++.

Resum

En aquest mòdul introductori hem recordat breument alguns conceptes fonamentals dels SGBD:

- 1) Hem revisat els objectius dels SGBD actuals i algunes de les funcionalitats que ens donen per a aconseguir-los.
- 2) Hem recordat el concepte de *transacció* i hem vist com és utilitzat per avetllar per la integritat de les dades.
- 3) També hem recordat que l'arquitectura de tres nivells ens aporta una gran flexibilitat als canvis, tant físics com lògics, i que els SGBD relacionals ens donen més independència de les dades que els prerelacionals.
- 4) Per acabar, hem vist que l'evolució actual dels SGBD relacionals els apropa al món de l'orientació a l'objecte.

Hem vist que el llenguatge SQL es pot utilitzar de forma directa, interactiva, però que també el podem fer servir des d'un programa si disposem d'un pre-compiler que ens permeti considerar l'SQL com incorporat o hostatjat al nostre llenguatge de programació. A més, podem treballar amb l'SQL des dels programes mitjançant crides a llibreries de funcions especialitzades. Hem distingit també la modalitat d'SQL estàtica de la dinàmica.

Hem parlat de l'arquitectura dels SGBD, és a dir, de la manera com estan estructurats en components, i hem vist que aquesta arquitectura té un aspecte intern i un d'extern. Hem citat els components típics d'un SGBD i les eines que habitualment l'acompanyen.

Finalment, hem donat algunes idees bàsiques sobre els SGBD relacionals que es poden trobar ara al mercat i sobre les tendències que s'observen en la seva evolució.

Activitats

1. Obteniu informació tecnicocomercial d'algun fabricant d'SGBD sobre les versions noves dels seus productes i mireu de reconèixer-hi els conceptes de què hem tractat en aquest mòdul.
2. Redacteu un informe breu (no més de cinc pàgines) sobre l'estat actual dels SGBD comercials a partir dels articles i les notícies que es publiquen sobre els productes del mercat a les revistes impreses, com ara *Datamation*, *Byte*, *ComputerWorld*, *Computing*, i a les revistes digitals, com *Intelligent Enterprise*.

Trobareu la revista digital *Intelligent Enterprise* a l'adreça d'internet següent:
<http://intelligententerprise.com>

WEB

Exercicis d'autoavaluació

1. Quina és la diferència essencial entre els SGBD relacionals i els prerelacionals?
2. Quins són els avantatges principals de l'SQL hostatjat estàtic respecte al basat en llibreries de funcions?
3. Què caracteritza els SGBD autènticament paral·lels?
4. Supposeu que compreu un SGBD, però en la versió personal. Sabeu que ocupa molt menys memòria que la versió multiusuari. És probable que de les dotze funcionalitats que hem citat com a més freqüents dels motors dels SGBD n'hi faltin algunes. Quines us sembla que podrien ser?

Solucionari

Exercicis d'autoavaluació

1. En els sistemes prerelacionals l'usuari està molt més lligat al món físic. Hi ha molt poca independència (tant física com lògica) de les dades. Així, per exemple, pràcticament qualsevol canvi que es faci en l'estructura de les dades obliga a retocar els programes o a escriure les consultes de manera diferent.

2. A l'SQL hostatjat estàtic l'anàlisi i la traducció de les instruccions es fa en temps de compilació; en canvi, si usen llibreries de funcions es fa en temps d'execució. Així, doncs, les crides a funcions en principi són menys eficients que les instruccions incorporades al programa. D'altra banda, la segona tècnica obliga a saber, a més de la sintaxi de l'SQL, la sintaxi de les crides a les funcions de la llibreria que, per cert, generalment no són gaire senzilles.

3. En un SGBD paral·lel cada ordre SQL rebuda és dividida en parts, de manera que es puguin executar independentment les unes de les altres. Cada part s'executa en un processador separat (que potser treballa amb discs separats) i es posen en comú els resultats parcials per a formar el resultat final. El guany de temps pot arribar a ser espectacular.

4. Una versió monousuari no necessita la funcionalitat de l'accés concurrent i és possible que no necessiti un rendiment molt elevat. A més, si ocupa poca memòria, pot ser que les funcionalitats que pugui fer el sistema operatiu s'hagin eliminat de l'SGBD. Per tant, podria ser que les funcionalitats desaparegudes (algunes potser només parcialment) fossin les següents:

- Gestió de memòries intermèdies.
- Gestió de l'espai de la memòria externa.
- Mètodes d'accés.
- Gestió de processos.
- Control de l'accés concurrent.
- Control de la seguretat/privacitat.

Glossari

base de dades *f* Conjunt estructurat de dades que representa, entre d'altres, entitats i les seves interrelacions, amb integració i compartició de dades.

sigla: BD

BD *Vegeu* base de dades.

client/servidor *m* Tecnologia habitual per a distribuir processament. Dos processos diferents, que es poden executar en un mateix sistema o en sistemes separats, actuen de manera que l'un fa de client, o peticionari d'un servei, i l'altre fa de servidor.

Un procés client pot demanar serveis a diversos servidors, i un servidor pot rebre peticions de molts clients. En general, un procés A que demana un servei (fa de client) a un procés B, també pot fer de servidor d'un servei que li demani un altre procés C.

Sigla: C/S

C/S *Vegeu* Client/servidor.

esquema *m* Descripció o definició de la base de dades. Aquesta descripció està separada dels programes i és utilitzada per l'SGBD per a saber com és la base de dades amb la qual ha de treballar. L'arquitectura ANSI/SPARC recomana tres nivells d'esquemes: els esquemes externs (visions dels usuaris), l'esquema conceptual (visió global) i l'esquema físic (descripció de les característiques físiques).

modalitats d'SQL *f* Modalitat directa o interactiva, o crides a llibreries de funcions, i modalitat d'SQL hostatjat.

sistema de gestió de bases de dades *f* Programari que gestiona i controla bases de dades. Les seves principals funcions són les de facilitar-ne la utilització simultània a molts usuaris de tipus diferents, independitzar l'usuari del món físic i mantenir la integritat de les dades.

Sigla: SGBD

SGBD *Vegeu* sistema de gestió de base de dades.

SQL *m* *Vegeu Structured Query Language.*

Structured Query Language *m* Llenguatge pensat per a descriure, crear, actualitzar i consultar bases de dades. Fou creat per IBM al final dels anys setanta i estandarditzat per ANSI i ISO l'any 1986 (l'últim estàndard de l'SQL és de 1999). Actualment l'utilitzen gairebé tots els SGBD del mercat (fins i tot alguns SGBD no relacionals i alguns sistemes de fitxers).

Sigla: SQL

transacció *f* Conjunt d'operacions (sobre la base de dades) que s'executen com una unitat (o tot o res) i aïlladament (sense interferències) d'altres conjunts d'operacions que s'executen concurrentment.

Bibliografia

Bibliografia bàsica

Date, C.J. (2001). *Introducción a los sistemas de bases de datos* (7. ed.). Prentice-Hall.

És el llibre més popular sobre bases de dades en el món de l'ensenyament universitari. La introducció dóna una bona visió global del tema.

Silberschatz, A.; Korth, H.F.; Sudarshan, S. (1998). *Fundamentos de bases de datos* (3a. ed.). Madrid: McGraw-Hill.

Per a millorar la comprensió d'aquest mòdul recomanem llegir els capítols 1 i 16 d'aquesta obra.

Bibliografia complementària

Per a actualitzar la vostra visió global dels SGBD del mercat, podeu consultar a Internet les pàgines dels fabricants d'SGBD (Informix, Oracle, IBM, CA, Microsoft, NCR, etc.), especialment els documents de tipus *white papers*, *product overview*, *technical briefs*, etc. També trobareu informació d'interès sobre els SGBD del mercat a les adreces d'Internet següents:

Intelligent Enterprise. www.intelligententerprise.com

Oracle view. www.oreview.com

DB2 magazine. www.db2mag.com

A les adreces següents trobareu informació del món de la recerca sobre bases de dades:

ACM Sigmod. www.acm.org/sigmod

Computer Science Bibliography (Universitat de Trier). www.informatic.uni-trier.de/~ley/db

Teradata. www.teradata.com

Recordeu que...

... si x és el nom de l'empresa, la seva adreça d'Internet acostuma a ser www.x.com.

