

Índex

Índex.....	3
UD1. Introducció a les bases de dades	5
1. Base de dades i sistema gestor de bases de dades.....	5
2. Objectius del sistema gestor de bases de dades.....	7
2.1. Possibilitat de consultes complexes	7
2.2. Elasticitat a canvis	7
2.3. Evitar redundància de les dades	8
2.4. Assegurar la consistència de les dades	8
2.5. Prevenir accessos no autoritzats.....	13
2.6. Capacitat d'auditoria	14
2.7. Potents eines d'administració.....	14
3. Arquitectura d'un sistema gestor de bases de dades.....	14
4. Independència lògica i física de les dades	17
5. Llenguatges en sistemes gestors de bases de dades	18
6. Usuaris de bases de dades	20
7. Diccionari de dades	22
8. Arquitectura client-servidor.	23
9. Bases de dades distribuïdes i replicades.....	26
9.1. Possibilitats de distribuir i/o replicar les dades.....	28
9.2. Fragmentació.....	28
9.3. Avantatges i inconvenients.....	30
9.4. Gestió de bases de dades distribuïdes	31

UD1. Introducció a les bases de dades

1. Base de dades i sistema gestor de bases de dades

En el present apartat intentarem definir els conceptes de base de dades i de sistema gestor de base de dades, establint la distinció entre aquests conceptes i els conceptes d'arxiu i de sistema gestor d'arxius¹.

Què és una base de dades? Vegem-ne algunes definicions.

Una **base de dades** és un conjunt de fitxers **interrelacionats**, amb **integració i compartició** de dades.

En aquesta definició apareixen tres conceptes importants. Anem a explicitar-los.

- Arxius **interrelacionats**, que indica que els registres d'un arxiu estan relacionats amb els registres dels altres arxius.

Les relacions entre els registres dels diferents arxius poden ser de tres tipus:

- Un a molts, com el cas que es produeix quan per un client hi pot haver moltes comandes de venda.
 - Molts a un, com el cas que es produeix quan moltes comandes de venda poden correspondre a un mateix client
 - Molts a molts, com el cas que es produeix quan un article es ven a molts clients i un client compra molts articles.
- **Integració** de dades, que indica que no hi ha d'haver repeticions ni redundàncies en el conjunt de dades de tots els arxius.

Això s'aconsegueix considerant els diversos conjunts de dades a gestionar (diferents arxius) com un únic conjunt.

- **Compartició** de dades, que indica que molts usuaris, de manera simultània, han de poder accedir a les dades, actualitzant-les fins i tot.

L'anterior definició presenta una base de dades com un conjunt d'arxius. Com a primera definició, no està malament, però no és la realitat. Una base de dades ha d'aparèixer com el resultat d'un anàlisi de la situació a informatitzar i en l'etapa d'anàlisi no es parla mai d'arxius, sinó de les entitats a gestionar, amb els seus atributs i els valors que poden prendre.

¹ **Arxius i sistema gestor d'arxius.**

La comprensió de les bases de dades i dels sistemes gestors de bases de dades és més fàcil si l'alumne ha treballat amb arxius i, per tant, amb sistemes gestors d'arxius. En el *CFS Desenvolupament d'aplicacions informàtiques* aquests conceptes es presenten amb deteniment en el crèdit *Programació estructurada i modular*.

Pel cas en que aquest crèdit es cursi paral·lelament a l'anterior, comentem que arxius i bases de dades tenen per objectiu l'emmagatzematge de les dades de manera que la seva gestió sigui simple i eficaç. Les tècniques informàtiques inicials per assolir aquest objectiu consistiren en la gestió dels arxius seqüencials, la qual va evolucionar cap la gestió d'arxius relatius, calculats i indexats. La continua evolució en l'emmagatzematge de les dades va portar a la gestió de les bases de dades, objectiu del present crèdit.

En determinats punts d'aquesta unitat didàctica es compara la gestió de bases de dades amb la gestió d'arxius, pressuposant que l'alumne és coneixedor de la gestió d'arxius. No cal preocupar-se si no es dona aquesta situació ja que en tal cas l'alumne estaria cursant simultàniament aquest crèdit i el crèdit *Programació estructurada i modular* que inclou l'estudi dels sistemes gestors d'arxius.

Així, seria millor presentar una nova definició de base de dades.

Una **base de dades** és un conjunt d'entitats **interrelacionades**, amb **integració i compartició** de coneixements.

Aquesta definició s'emmarca en el món dels coneixements. Però la informàtica tracta amb dades i, per tant, haurem de tractar la base de dades en el món de les representacions.

Ens cal, per tant, establir com representar les bases de dades. Tenim, d'entrada, dues possibilitats:

- Utilitzar tècniques d'arxius
- Utilitzar tècniques basades en estructures de dades, algunes possiblement conegudes si som coneixedors de la metodologia de la programació (arbres, objectes) i altres potser desconegudes (xarxes, taules -no confondre amb les taules com a tipus de dada en memòria interna-)

La decisió que es va prendre en el seu moment, va ser representar les bases de dades amb tècniques basades en estructures de dades, i així han aparegut diferents models de bases de dades:

- **Model jeràrquic**, basat en l'estructura de dada *arbre*, coneguda per nosaltres. No aprofundirem, però, en aquest model de base de dades.
- **Model en xarxa**, que es basa en l'estructura de dada *xarxa*, desconeguda en aquest cicle formatiu. Tampoc aprofundirem en aquest model de base de dades.
- **Model relacional**, que es basa en l'estructura de dada *taula* -no confondre amb el tipus de dada estàtica *taula*-, encara desconeguda per nosaltres, però que veurem en profunditat en properes unitats didàctiques. Aquest model serà el que posarem en pràctica en aquest crèdit.
- **Models orientats a objectes**, que tenen en compte els conceptes introduïts en l'anàlisi i disseny orientats a objectes².

Així, la definició més correcta de base de dades seria:

Una **base de dades** és un conjunt estructurat de dades amb **integració i compartició**.

I, com gestionarem les bases de dades? Doncs de manera similar a com es gestionen els arxius. Necessitarem d'un conjunt de programes que ens facilitin la seva gestió, de manera similar al conjunt de programes que formen el sistema gestor de fitxers i que faciliten la gestió dels arxius. En el món de les bases de dades, aquest conjunt de programes conformen el que s'anomena **sistema gestor de bases de dades**³. De manera semblant a la utilització de les sigles SGF per referenciar els sistemes gestors de fitxers, utilitzarem les sigles SGBD per referenciar els sistemes gestors de bases de dades.

² Orientació a Objectes

És una metodologia d'anàlisi i programació de nova implantació en comparació amb la metodologia estructurada i modular.

En el món informàtic s'ha demostrat que l'arrelament de les noves tecnologies es produeix bastant més enllà del moment en que comencen a sorgir. I això és el que està passant amb els models de bases de dades orientats a objectes. N'hi ha alguns prototipus, però poc o gens s'utilitzen en el món real.

En l'actualitat, el model de base de dades més arrelat és el relacional, pels que alguns fabricants han incorporat, a títol personal -sense seguir cap estàndard-, conceptes de l'orientació a objectes. Ens movem encara, però, en el model relacional.

³ Termes anglesos

Oi que un SGF permet gestionar diferents arxius? De la mateixa manera, un SGBD permet gestionar diferents bases de dades. No s'ha de confondre el concepte de SGBD (conjunt de programes que permeten gestionar bases de dades) amb el concepte de base de dades (conjunt estructurat de dades amb integració i compartició).

2. Objectius del sistema gestor de bases de dades

En aquest apartat establirem els principals objectius d'un sistema gestor de bases de dades.

2.1. Possibilitat de consultes complexes

Es disposa de mecanismes senzills per efectuar consultes complexes.

Així, imaginem-nos la necessitat de conèixer les principals dades (*codi, raó social, telèfon i NIF*) dels clients que han adquirit algun producte de la família "*Aliments*" al llarg d'un determinat exercici econòmic (per exemple, l'any 2000).

En una gestió d'arxius, possiblement tindríem un arxiu de clients (on hi hauria les dades dels clients), un arxiu de productes (on hi hauria les dades dels productes i entre elles el codi de la família a la que pertany cada producte), un arxiu de famílies de productes (on hi hauria la codificació de les famílies de productes) i un arxiu de vendes (on hi hauria la informació de les vendes de productes efectuades a cada client: quantitat, preu, data, descomptes,...).

Us imagineu la quantitat de feina, amb les tècniques que ens possibiliten els SGF, per tal d'aconseguir la informació demanada? Caldria efectuar un programa amb força codi. En un SGBD relacional (que encara no coneixem), es resoluria amb una instrucció semblant a:

```
select distinct cl.codi, cl.rao, cl.telefon, cl.nif
from client cl, productes pr, familia fm, vendes vn
where cl.codi = vn.codcli
  and pr.codi = vn.codpro
  and fm.codi = pr.codfam
  and upper(fm.desc) = "ALIMENTS"
  and vn.data >= "01-01-2000"
  and vn.data <= "31-12-2000";
```

Oi que val la pena? Compareu les anteriors línies amb el programa que hauríeu d'haver fet en un SGF per aconseguir la informació demanada.

Les anteriors línies corresponen a una instrucció desenvolupada amb el llenguatge SQL, que coneixerem al llarg d'aquest crèdit. Fixeu-vos que no ha calgut obrir ni tancar cap arxiu, ni efectuar cap recorregut ni situar-se en cap registre concret. Simplement hem demanat al SGBD que ens doni una certa informació explicitant on l'ha d'anar a cercar i ell fa la feina per nosaltres. Interessant, oi?

2.2. Elasticitat a canvis

Es disposa de la possibilitat d'efectuar canvis a la base de dades impossibles d'efectuar de manera immediata en SGF.

El terme anglès per un *sistema gestor de base de dades* és *Database Management System* i s'acostuma a abreujar amb DBMS. En canvi el terme per un *sistema gestor de fitxers* és *File Management System* i s'abreuja amb FMS.

Així, imagineu la necessitat d'afegir o suprimir atributs en una entitat o de modificar-ne el seu domini de valors. Imagineu també la necessitat d'afegir noves entitats i d'afegir o modificar relacions entre les entitats. Els SGBD permeten aquests canvis de manera senzilla i transparent a les aplicacions ja existents i no afectades de manera directa pels canvis efectuats.

Aquest objectiu dels SGBD té un nom molt més concret: **independència física i lògica de les dades**. És tal la importància d'aquest objectiu que li donarem, més endavant, un apartat per a tractar-lo amb la profunditat que es mereix.

2.3. Evitar redundància de les dades

Es facilita evitar:

- Repeticions de dades en diferents entitats, amb la implementació de les relacions entre entitats.

Per exemple, no ens caldrà tenir, en les *comandes de venda*, la raó social del client si hi tenim el codi de client, doncs amb aquesta informació es té la raó social de manera ràpida a partir de l'entitat *clients* en tenir relacionades les entitats *clients* i *comandes de venda*.

En SGF no estàvem obligats a tenir en l'arxiu *comandes de venda* la raó social del client, però moltes vegades s'hi tenia per tal de tenir accés de manera ràpida a aquesta informació en gestionar la comanda. Aquesta forma de treballar portava lloc a nombrosos problemes motivats per l'actualització de les dades repetides.

En SGBD no es té la necessitat de tenir la informació repetida ja que el SGBD ens sap facilitar de manera fàcil la informació relacionada existent en altres entitats.

- Emmagatzemar dades calculables

Per exemple, no ens caldrà tenir a l'entitat *clients*, l'import facturat i pendent de cobrar per cada client, ja que aquesta informació és calculable a partir dels cobraments pendents per les factures dels clients i aquesta informació resideix en altres entitats de la base de dades. Quan es necessiti saber aquesta informació, el SGBD ens permetrà calcular-la a l'instant.

D'aquesta manera s'estalvia espai i s'estalvia actualitzar la dada calculada quan hi ha modificacions en les dades a partir de la que es calcula.

A vegades, però, és recomanable renunciar als anteriors estalvis i mantenir la redundància de les dades calculables. Això es justifica quan:

- La consulta de les dades calculables s'efectua moltes vegades, ja que cada vegada cal efectuar els corresponents càlculs. Per exemple, el risc en curs d'un client.
- Els càlculs a efectuar són de cost considerable. Per exemple, les vendes resumides per mes a nivell de client.

Caldrà, per tant, decidir en temps d'anàlisi, el tipus de gestió a efectuar per cadascuna de les dades calculables.

2.4. Assegurar la consistència de les dades

El SGBD contempla mecanismes per a conservar correctes les dades dins la base de dades.

Hi ha diferents causes per les que les dades poden passar a ser incorrectes:

- Desastre en el suport físic on hi ha enregistrades les dades
- Errors de disseny en programes que no efectuen el tractament de dades de manera correcta.
- Actualització incompleta de dades per caiguda del sistema informàtic.
- Introducció de dades no vàlides a través de les eines que puguin existir per interactuar amb la base de dades.
- Actualitzacions concurrents que provoquen la pèrdua d'operacions d'actualització.

Com pot el SGBD fer front a les causes enumerades? Amb diferents mecanismes que explicitem a continuació.

2.4.1. Mecanismes de recuperació

Hi ha diferents situacions en les que pot ser necessari efectuar una recuperació de les dades. Per tant, també hi haurà diferents formes d'efectuar-la.

Es produeix una **recuperació en calent** quan després d'una caiguda del SGBD (per qualsevol motiu) han quedat operacions d'actualització a mig efectuar.

Les operacions d'actualització a mig efectuar poden correspondre a una única operació sobre la base de dades o a un conjunt d'instruccions agrupades sota una transacció.

La recuperació en calent consisteix en finalitzar o en desfer les operacions que havien quedat a mig efectuar. El SGBD decideix de manera automàtica, en tornar a estar operatiu, si finalitza les operacions o les desfà.

Per a poder dur a terme la recuperació en calent, el SGBD en execució normal, va enregistrant en un arxiu especial, anomenat **dietari** o **log**, la informació corresponent a les actualitzacions a efectuar sobre la base de dades. En finalitzar correctament les actualitzacions corresponents a una operació o transacció, el SGBD deixa una marca de verificació (**checkpoint**) en el diari i en un arxiu de rearrencada. D'aquesta manera, quan és necessària la recuperació en calent, el SGBD obté la marca de verificació més recent de l'arxiu de rearrencada i efectua un recorregut de l'arxiu dietari des d'aquest punt fins el final, per obtenir la informació de quines operacions ha de desfer o finalitzar.

Es produeix una **recuperació en fred** quan es detecta un desastre a la base de dades i s'ha de refer el seu contingut actual a partir de la darrera **còpia de seguretat** i dels arxius **dietaris** enregistrats des de la còpia.

Cal observar que per a poder efectuar aquesta recuperació és tan important disposar d'una còpia de seguretat recent com dels arxius dietaris. Per aquest motiu aquests arxius no es destrueixen.

Hi ha encara una tercera situació en la que pot ser necessari haver d'utilitzar els arxius dietaris: la recuperació semàntica.

Es produeix una **recuperació semàntica** quan es detecta que hi ha hagut, en un instant temporal passat, una actualització incorrecta segons el punt de vista semàntic i s'ha de refer dita actualització i tots els moviments posteriors fins el moment present.

Aquesta situació és la que es pot produir, per exemple, quan s'ha enregistrat un càlcul incorrecte que ha provocat uns posteriors moviments en cascada. Imaginem, per exemple, els problemes que pot ocasionar a un client d'una entitat bancària l'enregistrament d'un ingrés en metàl·lic efectuat en un caixer automàtic, en una data posterior de la realment efectuada (per error de l'operador que enregistra l'ingrés en el sistema). Evidentment, caldrà restaurar l'actualització a la data correcta i refer els moviments posteriors que s'haguessin desencadenat.

A vegades, aquesta recuperació es pot efectuar directament a través d'un històric de moviments enregistrats a la pròpia base de dades, però en altres ocasions és necessari restaurar el contingut de la base de dades a partir dels arxius dietaris. En tal situació, s'acostuma a partir de la base de dades actual i a desfer els moviments enregistrats en els dietaris (**recuperació cap enrera**), fins arribar al moment en que es va produir l'actualització semànticament incorrecta; s'enregistra l'actualització de manera correcta i s'inicia el procés de **recuperació cap endavant** a partir dels mateixos dietaris enregistrant els moviments pertinents a partir de la nova situació.

2.4.2. Definició de la integritat

Sota aquest mecanisme s'agrupen:

a) Regles d'integritat del model

Són les regles de funcionament que porta associat el model de base de dades que correspongui. Així, en el model jeràrquic, basat en l'estructura de dades *arbre*, es verifica, obligatòriament les regles d'integritat dels arbres:

- un node no arrel sempre té un i només un pare; l'arrel és l'únic node que no té pare
- si l'arbre és binari, cap node pot tenir més de dos fills
- si l'arbre és balancejat, s'ha de mantenir el factor d'equilibri

El SGBD controlarà sempre les regles d'integritat del model en el que es basa.

b) Regles d'integritat referencial

Són les regles que el SGBD obligarà a complir en funció de les relacions que s'estableixin entre les entitats de la base de dades.

Així, si es disposa d'una entitat *país* i d'una entitat *departament* (província) i s'estableix que entre *país* i *departament* hi ha una relació de *1 a molts*, de manera que tot *departament* ha de pertànyer obligatòriament a un *país* existent a l'entitat *país*, el SGBD controlarà aquest fet i no permetrà:

- donar d'alta un departament i no assignar-li país o assignar-li un país inexistent a la base de dades
- modificar el país assignat correctament a un departament per un país inexistent o deixar-lo sense país assignat

- eliminar un país quan hi hagi departaments que el tenen assignat
- modificar la clau primària d'un país que tingui departaments assignats a través d'aquest valor de clau

Alguns SGBD permeten definir el comportament a seguir davant la tercera situació presentada, que pot establir-se en no permetre l'eliminació o en efectuar una eliminació en cascada (esborrant país i departaments associats) o en efectuar l'eliminació i deixar els corresponents departaments sense assignar (per a poder fer això no hi hauria d'haver l'obligatorietat d'assignar cada departament a un país, és a dir, caldria l'existència d'un valor nul en aquesta assignació).

Alguns SGBD també permeten definir el comportament a seguir davant la quarta situació presentada, que pot establir-se en no permetre la modificació o en modificar la clau primària del país i el valor assignat als corresponents departaments, de manera que el lligam país-departament continuï mantenint-se.

c) Restriccions d'usuari

Són restriccions que es defineixen pels diferents atributs de les entitats, de manera que el SGBD les comprova en les operacions d'actualització. Poden ser:

- estàtiques, com per exemple, "10 < edat < 99"
- dinàmiques, com per exemple, "els preus mai baixaran"

2.4.3. Control de concurrència

Els SGBD han de controlar de forma efectiva els accessos concurrents a les dades. Del contrari es pot produir el problema conegut com a **operació perduda**, exemplificat en la següent situació.

Considerem diferents llocs de treball en un magatzem, ocupats per operaris que executen sortides de material. Considerem un article A del que en consten en existència 100 unitats. Suposem que cada operari ha d'efectuar una sortida de 20 unitats.

Imaginem-nos que els esdeveniments tenen lloc, en el temps, en el següent ordre:

Operari 1	Operari 2
Consulta l'existència de la base de dades i en queden 100 unitats	
	Consulta l'existència de la bases de dades i en queden 100 unitats
Efectua la sortida de 20 unitats, actualitzant l'existència a la lectura anterior (100) menys la sortida efectuada. És a dir, actualitza el registre a 80 unitats	
	Efectua la sortida de 20 unitats, actualitzant l'existència a la lectura anterior (100) menys la sortida efectuada. És a dir, actualitza el registre a 80 unitats

És clar que el valor final existent a la base de dades és incorrecte. S'ha perdut l'efecte d'una de les operacions d'actualització.

Per a solucionar aquest problema, el SGBD està programat per a controlar, mitjançant el **mecanisme de bloqueig**, les actualitzacions de dades a un determinat nivell (entitat, instància, atribut o, fins i tot, interval de valors) de manera que en

iniciar-se una actualització es bloqueja la dada al nivell que correspongui i ningú més pot accedir a la dada bloquejada fins que el propi SGBD l'allibera. Les sol·licituds d'actualització poden quedar en espera de l'alliberament o ser rebutjades, segons s'hagi definit l'actuació en el SGBD.

El nivell al què s'aplica el bloqueig acostuma a anomenar-se **grànul**. Així hi pot haver grànuls a nivell d'entitat, a nivell d'instància de l'entitat, a nivell d'atribut o, fins i tot, a nivell d'interval de valors per un atribut. Exemplificant-ho, pot donar-se el cas que es bloquegi tota l'entitat *clients* (el grànul és l'entitat) o un *client* concret (el grànul és la instància) o l'atribut *domicili* d'un client concret (el grànul és l'atribut) o l'atribut *codi postal* del client si està entre un determinat interval de valors (el grànul és un interval de valors de l'atribut).

L'elecció del nivell d'actualització a controlar pel SGBD és un tema polèmic. Si s'escull un nivell petit s'afavoreix un major paral·lelisme de les transaccions, però augmenta la quantitat de temps utilitzada en l'accés a les dades.

El mecanisme de bloqueigs soluciona el control concurrent, però provoca les conegudes **abraçades mortals**, consistents en una espera circular entre dues o més transaccions de manera que cadascuna sol·licita una actualització sobre el mateix element. Exemplifiquem-ho.

Transacció 1	SGBD	Transacció 2
Inicia actualització d'un element A		
	Bloqueja l'element A	
		Inicia actualització d'un element B
	Bloqueja l'element B	
Necessita accedir a l'element B, que està bloquejat		
	Deixa la transacció 1 en espera de desbloqueig de l'element B	
		Necessita accedir a l'element A, que està bloquejat
	Deixa la transacció 2 en espera de desbloqueig de l'element A	

Per resoldre aquest problema es poden seguir dues tècniques:

- Controlar que no succeeixi, cosa que no sempre és fàcil

Hi ha diferents mecanismes per a prevenir l'abraçada mortal (amb un d'ells n'hi ha prou). Suposem que la transacció T_1 té bloquejat el grànul i que la transacció T_2 intenta bloquejar el mateix grànul. Podem actuar amb una de les tres formes següents:

- "Wait-Die": Si T_2 és més vella, s'espera; en cas contrari, és cancel·la.
- "Kill-Wait": Si T_2 és més vella, T_1 és cancel·la; en cas contrari, T_2 s'espera.
- "Wound-Wait": Si T_2 és més vella, T_1 es cancel·la a no ser que hagi començat a confirmar actualitzacions; en cas contrari, T_2 s'espera.

Una transacció podria ser cancel·lada moltes vegades si no s'adopta alguna mesura per detectar tal situació i donar prioritat a dita transacció. L'*edat* de les transaccions es calcula a través d'un valor que té un comptador quan s'inicien.

- Detectar la seva aparició i trencar-la cancel·lant algunes transaccions (les víctimes).

Hi ha diferents maneres de detectar l'abraçada mortal. Una d'elles consisteix en mesurar l'interval de temps que dura una transacció, de manera que en superar uns determinats límits es pugui considerar que està en espera permanent.

2.5. Prevenir accessos no autoritzats

El SGBD ha de protegir les dades dels intents d'accés dels usuaris que no tinguin els privilegis corresponents. Aquesta prevenció s'efectua a nivell del SGBD i no és incompatible amb els mecanismes que pugui tenir el propi sistema informàtic (control d'accés a les instal·lacions, control d'entrada al sistema per part del sistema operatiu,)

Pel que fa referència al SGBD (tema que ens ocupa), els sistemes moderns permeten una o ambdues de les estratègies següents:

- **Control discrecional**, que permet definir privilegis d'accés a cada usuari (o grup d'usuaris) sobre cada objecte (o grup d'objectes).
- **Control obligatori**, en el que cada objecte (o grup d'objectes) és etiquetat amb un cert nivell de classificació i cada usuari (o grup d'usuaris) és etiquetat amb un cert nivell d'acreditació, de manera que cada objecte només pot ser accedit pels usuaris amb un determinat nivell d'acreditació.

Els esquemes obligatoris són més rígids que els esquemes discrecionals.

Per a que el SGBD pugui decidir quines restriccions ha d'aplicar ha de ser capaç de reconèixer l'origen de la sol·licitud. Per aquest motiu, els SGBD han de demanar a l'usuari la identificació en iniciar una sessió de treball en el SGBD. Normalment, aquesta identificació va acompanyada d'una contrasenya⁴.

Normalment, en iniciar una sessió de treball en un sistema informàtic, aquest, via sistema operatiu, exigeix la identificació de l'usuari. L'usuari, una vegada identificat, entra en el sistema informàtic i pot demanar d'iniciar una sessió de treball amb el SGBD. Aquest pot exigir altra vegada la identificació a l'usuari, la qual pot ser totalment diferent de l'exigida pel sistema operatiu per accedir al sistema informàtic. Alguns SGBD permeten utilitzar el mateix sistema d'accés que l'utilitzat per accedir al sistema informàtic, de manera que l'usuari no hagi d'anar identificant-se en les diferents capes que ha de travessar fins arribar al SGBD.

A més de totes aquestes tècniques, alguns SGBD permeten utilitzar la criptografia per a emmagatzemar les dades dins la base de dades. És a dir, les dades s'enregistren prèviament criptografiades (xifrades), de manera que en recuperar-les el SGBD ha de procedir a descriptografiar-les (desxifrar-les). D'aquesta manera es prevé el fet que s'accedeixi a les dades sense passar pel SGBD, és a dir, accedint directament al suport físic on estan enregistrades.

És clar que tota tècnica per prevenir els accessos no desitjats implica un esforç de temps i càlcul per part del SGBD. Per tant, caldrà establir els criteris d'accés en funció de les reals necessitats de control dins cada organització.

⁴ **Mecanismes d'identificació**

La forma més normal d'accedir al SGBD és via identificació escrita de l'usuari acompanyada d'una contrasenya. Aquests sistemes possiblement seran modificats en un futur, tal i com es veu en les pel·lícules actuals, passant a utilitzar altres mètodes basats en característiques físiques, com les empremtes dactilars, la veu, la pupil·la dels ulls,....

2.6. Capacitat d'auditoria

L'objectiu que es pretén amb l'auditoria és el de poder determinar quins usuaris van interactuar amb la base de dades i en quin moment.

Hi ha diferents nivells d'auditoria:

- Control dels usuaris que inicien sessió de treball amb una base de dades.

Aquest nivell d'auditoria precisa pocs recursos.

- Control de les modificacions efectuades en una base de dades (o en uns determinats objectes de la base de dades) per cada usuari.

Aquest nivell d'auditoria precisa més recursos que el nivell anterior i normalment s'activa per aquells objectes crítics.

- Control de les lectures efectuades en una base de dades per cada usuari.

Aquest nivell d'auditoria precisa molts més recursos que els nivells anteriors i normalment s'activa quan hi ha indicis de que s'està extraient informació confidencial (espionatge).

2.7. Potents eines d'administració

Per poder fer front a tots els objectius enumerats prèviament, el SGBD necessita d'un administrador (el presentarem posteriorment) i aquest necessita d'unes potents eines per fer front als requeriments que l'organització pretén obtenir del SGBD.

Algunes de les eines del SGBD poden formar part del propi SGBD. Altres poden ser programes d'aplicació que interactuen amb el SGBD.

3. Arquitectura d'un sistema gestor de bases de dades

En els apartats anteriors hem presentat els conceptes de *base de dades* i de *sistema gestor de bases de dades*, establint les diferències entre ells i els objectius que persegueix el darrer.

L'evolució dels SGF cap als SGBD ha estat motivada per noves prestacions que les organitzacions demanaven als SGF i que aquests no podien facilitar. En especial, una de les grans demandes era el fet que els arxius que fins el moment eren pensats per unes aplicacions concretes, poguessin passar a formar part d'una visió més global.

Amb el concepte de BD s'aconsegueix la integració de totes les dades (dispersades fins el moment en diferents arxius i fins i tot duplicades) en un sol objecte. Ara, però, apareix una necessitat. Cal que la BD es pugui veure a trossos simulant, d'alguna manera, els arxius als que tenien accés les diferents aplicacions en el SGF.

Aquesta necessitat és motivada pel fet que els usuaris d'un determinat departament de l'organització han de poder veure les dades des del seu punt de vista, des del seu enfocament. Exemplem-ho.

Imaginem-nos l'entitat *clients*. El departament comercial necessita gestionar les dades *codi, raó social, domicili fiscal, domicili de lliurament, venedor assignat, polítiques de descomptes i facturació*. En canvi, el departament comptable necessita gestionar les dades *codi, raó social, domicili fiscal, facturació, compte comptable, compte d'explotació i cobraments*. És clar que l'entitat *clients* és única i que englobarà la totalitat dels atributs descrits. Però l'organització exigeix que la visió que en tingui el departament comercial sigui diferent de la visió que en tingui el departament comptable, de manera semblant a les visions que tenien antigament en treballar amb SGF. I aquesta exigència és lògica ja que es vol centrar als usuaris en la gestió de les dades que els pertoca.

Entenc que ara mateix podeu tenir una mica d'embolic en el cap. Pot ser us esteu preguntant: quin problema hi ha en tenir tots els atributs en una única entitat si les aplicacions desenvolupades pels departaments comercial i comptable ja s'encarregaran de gestionar les dades que els pertocin? per què cal tenir diferents visions de la base de dades?

Hi ha diferents motius que donen resposta a la necessitat plantejada:

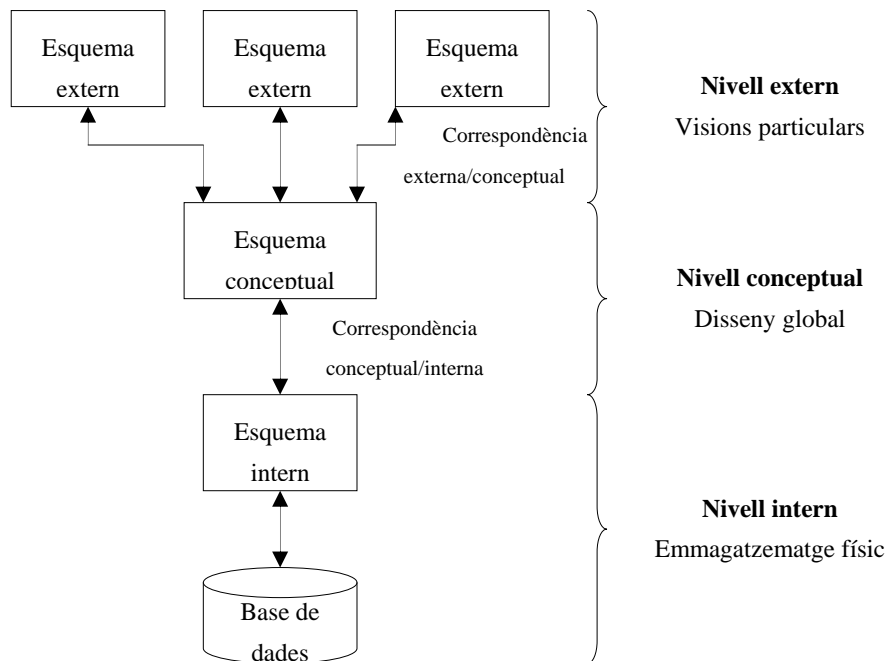
- El SGBD possibilita la realització de consultes complexes de manera senzilla. Recordeu l'exemple de l'apartat 2.1.? Un usuari que conegui la sintaxi del llenguatge (cosa no gens difícil com bé veurem més endavant), pot executar operacions per obtenir i modificar informació de la base de dades. Si l'usuari és del departament comercial hauria de veure els *clients* dels del punt de vista comercial, mentre que si l'usuari és del departament comptable els hauria de veure dels de punt de vista comptable. Cap d'ells n'ha de fer absolutament res amb les dades necessàries per l'altre departament.
- Si les aplicacions desenvolupades pels diferents departaments no es basen en l'entitat *clients* sinó en la particular visió de l'entitat que correspon al seu departament, qualsevol canvi en l'entitat *clients* que afecti a l'altre departament, li serà transparent, és a dir, no es veurà afectada doncs la particular visió de l'entitat continuarà invariable.

Així doncs, es vol treballar amb bases de dades per a integrar i compartir les dades però, a la vegada, es vol tenir diferents visions d'ella.

Com aconseguir aquesta dualitat?

La situació es va plantejar en els inicis dels SGBD i evidentment va sorgir als Estats Units d'Amèrica. Allí, l'*American National Standard Institute (ANSI)*, responsable de la definició de normes per la indústria del país, estructurat en diferents divisions, una de les quals és l'anomenada *X3* que s'ocupa de les normes relatives als ordinadors i a les màquines d'oficina, va crear el *System Planning And Requirements Committee (SPARC)*, del qual va sorgir l'informe que posa les bases per solucionar la situació plantejada i que proposa per les bases de dades una arquitectura de tres nivells que s'ha conegut com **arquitectura ANSI/X3/SPARC**.

L'esmentat informe planteja que l'arquitectura d'una base de dades ha de tenir tres nivells: **intern, conceptual i extern**, cadascun dels quals correspon a un punt de vista diferent de la base de dades: emmagatzematge físic de la BD, disseny global de la BD i visions particulars de la BD tal i com es veu a la figura:



Anem a veure amb una mica de profunditat cadascun dels tres nivells.

1) Nivell intern

Descriu l'estructura d'emmagatzematge físic que conté les dades, és a dir, descriu, entre altres elements, els arxius físics que contenen les dades (noms, tipus d'accés, ubicació,...), els registres físics d'aquests arxius (longitud, camps,...) i les rutes d'accés (índexs, encadenaments,...)

El nivell intern està descrit en el SGBD pel que s'anomena **esquema intern**. És el SGBD qui juntament amb el SO gestionarà, en temps d'execució, les entrades i sortides físiques, en base a la informació existent en l'esquema intern. En aquesta gestió el SGBD transforma dades de registre físic a registre lògic, operació anomenada de diferents formes: transformació conceptual/interna, correspondència conceptual/interna, mapeig conceptual/intern.

2) Nivell conceptual

Descriu el disseny global de tota la base de dades. En aquest nivell s'inclou la descripció de totes les entitats, amb els seus atributs i dominis de valors, així com les interrelacions entre les entitats, les regles d'integritat referencial, les restriccions, les autoritzacions,... En definitiva, representa la visió integrada de tots els usuaris. El nivell conceptual està descrit en el SGBD pel que s'anomena **esquema conceptual**.

3) Nivell extern

Descriu les diferents visions definides a la base de dades, cadascuna de les quals representa la percepció que de la base de dades en tenen els diferents departaments o grups d'usuaris o aplicacions.

Cadascuna de les visions de la base de dades definides en el nivell extern ha d'estar descrita en el que s'anomena **esquema extern**. Els anteriors nivells intern i conceptual estan descrits sempre en un únic esquema intern i conceptual. En canvi, en el nivell extern hi ha d'haver tants esquemes externes com visions diferents de la base de dades es vulgui mantenir.

Cada esquema extern correspon a la visió que es donarà als usuaris i aplicacions que hi tinguin accés. Les entitats aquí definides poden provenir directament d'entitats existents en l'esquema conceptual o ser el resultat d'un muntatge a partir de les entitats existents en l'esquema conceptual. Així, en el cas presentat a l'inici d'aquest apartat, en l'esquema extern corresponent al departament comercial hi hauria una entitat *clients* aconseguida com a subconjunt de la vertadera entitat *clients* existent en l'esquema conceptual. Les entitats de l'esquema extern poden ser muntades a partir de subconjunts, interseccions i unions d'altres entitats de l'esquema conceptual o del propi esquema extern. Els noms dels atributs també es poden redefinir i els dominis de valors es poden fer més restrictius que els existents en l'esquema conceptual.

És el SGBD qui gestionarà, en temps d'execució, les entrades i sortides sol·licitades pels usuaris en els esquemes externs, en base a la informació existent en el propi esquema. En aquesta gestió el SGBD transforma les sol·licituds de l'esquema extern a l'esquema conceptual, operació anomenada de diferents formes: transformació externa/conceptual, correspondència externa/conceptual, mapeig extern/conceptual.

4. Independència lògica i física de les dades⁵

En l'apartat destinat als objectius principals a assolir amb els SGBD apareixia l'*elasticitat a canvis* i es comentava que aquesta s'assolia gràcies a la independència física i lògica de les dades, fet molt important que es mereixia un apartat específic.

S'aconsegueix **independència física de les dades** quan l'accés a les dades no depèn del suport extern, ni del sistema operatiu, ni del maquinari, ni de l'estructura física en que estan distribuïdes les dades.

S'aconsegueix **independència lògica de les dades** quan el disseny lògic de les dades es pot modificar sense haver de modificar les aplicacions ja desenvolupades i que no estan directament implicades en els canvis efectuats.

Els SGBD asseguren els dos tipus d'independència.

La independència física s'assoleix amb la distinció entre nivells conceptual i intern, ja que qualsevol canvi en el suport físic, sistema operatiu, maquinari o estructura física afectarà a com es troben les dades en el suport físic i, per tant, es reflectiran en l'esquema intern, però l'accés a les dades per part dels usuaris i de les aplicacions no es veurà afectat ja que serà el SGBD qui en la transformació conceptual/interna reconduirà l'accés a la nova situació.

La independència lògica s'assoleix amb la distinció entre nivells externs i conceptual, ja que qualsevol canvi en el disseny de les dades afectarà l'esquema conceptual i, de retruc, algun(s) esquema(es) extern(s). Les aplicacions ja desenvolupades sobre un esquema extern que es manté intacte no han de patir cap reestructuració. Fins i tot, es pot donar el cas de modificacions en l'esquema conceptual que afectarien directament a un esquema extern però aquest no es veu afectat per què la transformació externa/conceptual ha anul·lat la modificació.

En efecte. Suposem que en l'esquema conceptual teníem les entitats:

⁵ **Independència física de les dades en sistemes gestors de fitxers**

En els sistemes gestors de fitxers hi ha independència física de les dades que s'assoleix amb la distinció entre arxiu extern (visió de l'arxiu des del sistema operatiu) i arxiu intern (visió de l'arxiu des dels programes), fet possibilitat per l'arquitectura de dos nivells existent en sistemes gestors d'arxius.

país (codi, nom, habitants)
província (codi, nom, codi_país)
hospital (codi, nom, num_llits)

Suposem que tenim dos esquemes externs:

Esquema extern 1:

país, mapejada a partir de *país* de l'esquema conceptual
província, mapejada a partir de *província* de l'esquema conceptual

Esquema extern 2:

hospital, mapejada a partir d'*hospital* de l'esquema conceptual

Suposem que s'arriba a la conclusió que l'atribut *habitants* de l'entitat *país* ha de passar a estar a nivell de l'entitat *província*, doncs interessa tenir aquest valor desglossat entre les diferents províncies que formen un país. Això implica una modificació en l'esquema conceptual que pot mantenir-se transparent pels dos esquemes externs, tal i com es veu a continuació.

Esquema conceptual:

país (codi, nom)
província (codi, nom, codi_país, habitants)
hospital (codi, nom, num_llits)

Esquema extern 1:

país, amb els atributs *codi* i *nom* mapejats de manera directa de *país* de l'esquema conceptual i l'atribut *habitants* mapejat com a suma dels valors de l'atribut *habitants* de *província* per les províncies assignades al corresponent país
província, amb els atributs *codi*, *nom* i *codi_país* de manera directa de província de l'esquema conceptual

Esquema extern 2:

hospital, mapejada a partir d'*hospital* de l'esquema conceptual

Observeu que les modificacions efectuades en l'esquema s'han pogut mantenir transparents en els dos esquemes externs. Això no acostumarà a passar en tots els esquemes externs, ja que si s'efectua alguna modificació de disseny en l'esquema conceptual és molt probable que algun esquema extern s'hagi de veure afectat. En el cas que hem presentat, és molt possible que la modificació efectuada es vulgui fer aparèixer en l'esquema extern 1. Si és així, les aplicacions desenvolupades sobre aquest esquema extern hauran de modificar-se en allò que es vegin afectades.

5. Llenguatges en sistemes gestors de bases de dades

Els SGBD han de proporcionar llenguatges per a poder descriure els diferents nivells de la base de dades i per a poder accedir a les dades i així efectuar consultes i actualitzacions. Aquests llenguatges es classifiquen en⁶:

⁶ Termes en anglès

En l'argot informàtic s'acostuma a utilitzar els termes:

- *Data Definition Language*, abreujat per DDL
- *Data Control Language*, abreujat per DCL
- *Query Language*, abreujat per QL
- *Data Manipulation Language*, abreujat per DML

a) **Llenguatges destinats a la definició de les dades (LDD)**, que permeten definir els objectes (entitats, atributs, dominis, regles d'integritat referencial, restriccions, rutes d'accés,...) en els diferents esquemes.

Cada SGBD disposa d'un o més llenguatges LDD específics. En uns casos existeix un únic llenguatge, amb sentències que permeten descriure els elements dels diferents esquemes i, en altres, hi ha un llenguatge diferent per cadascun dels nivells.

b) **Llenguatges destinats al control sobre les dades (LCD)**, que permeten concedir i retirar permisos sobre els diferents objectes de la base de dades.

En alguns SGBD no hi ha distinció entre LDD i LCD i es parla únicament de LDD per a les tasques de definició de dades i de concessió i revocació de privilegis.

c) **Llenguatges destinats a la consulta de les dades (LC)**, que permeten efectuar lectures o sortides lògiques en el nivell extern.

d) **Llenguatges destinats a la manipulació de les dades (LMD)**, que permeten actualitzar la base de dades (altes, baixes i modificacions).

En alguns SGBD no hi ha distinció entre LC i LMD i es parla únicament de LMD per a les consultes i actualitzacions.

Tots aquests llenguatges acostumen a tenir una sintaxi senzilla, semblant a les ordres de consola per un sistema operatiu, anomenada **sintaxi autosuficient**.

Els llenguatges LC i LMD poden presentar, però, una segona sintaxi, **sintaxi hostatjada**, consistent en un conjunt de sentències que són admeses dins d'un llenguatge, anomenat **llenguatge amfitrió**, de propòsit general.

Així ens podem trobar LC i LMD que es poden hostatjar en llenguatges de tercera generació com C, Cobol, Fortran,... i en llenguatges de quarta generació.

Per hostatjar en llenguatges de tercera generació compilats es necessari disposar d'un precompilador específic pel llenguatge, subministrat pel fabricant del SGBD, amb el que es precompila el codi font que conté les sentències LC i LMD i s'obté, en cas d'absència d'errors, un nou font on les sentències LC i LMD han estat tractades i encapsulades de manera que es pot procedir a la compilació normal per tal d'obtenir el codi objecte i passar a la fase d'enllaç per obtenir el codi executable.

Els llenguatges de quarta generació que permeten utilitzar sentències LC i LMD acostumen a ser llenguatges incorporats al propi SGBD proporcionats pel fabricant del SGBD. Aquest és el cas dels productes comercials que utilitzarem en el present crèdit. Els productes *Access '97* i *Personal Oracle 8* incorporen eines visuals de quarta generació per a desenvolupar pantalles (altes, baixes, consultes i modificacions), informes, gràfics,... que interactuen amb la base de dades.

Per finalitzar, distingir que podem trobar dos tipologies de llenguatges LC i LMD en funció de la manera que l'usuari haurà d'accedir a la base de dades:

- **implícits**, on l'usuari demana les dades a accedir però no diu la forma d'accedir-hi; també s'anomenen **no procedurals** ja que no s'indica el procediment per accedir a les dades,
- **explícits**, on l'usuari indica la forma d'accedir a les dades desitjades; també s'anomenen **procedurals** ja que s'indica el procediment a seguir per accedir a les dades.

6. Usuaris de bases de dades

Un SGBD permet emmagatzemar dades, les quals han de ser accedides pels usuaris de l'organització i, com és d'imaginar, no tots els usuaris tenen les mateixes necessitats ni les mateixes capacitats d'execució. Per aquest motiu es distingeixen diferents tipus d'usuaris.

1) Usuaris informàtics

És el grup més reduït d'usuaris de bases de dades. Han de tenir forts coneixements informàtics. Podem distingir-ne:

a) **Dissenyadors d'aplicacions** (analistes, dissenyadors i programadors), que són els qui desenvolupen les aplicacions informàtiques que interactuen amb bases de dades.

Per a desenvolupar la seva feina utilitzen els llenguatges LC i LMD hostatjats en llenguatges de tercera i quarta generació textuals o gràfics. En l'actualitat, però, es treballa quasi exclusivament en entorns gràfics.

Aquests usuaris utilitzen també, en les fases de disseny, codificació i verificació de les aplicacions, els llenguatges LC i LMD autosuficients per a tenir accés immediat a la base de dades i així comprovar els seus continguts.

Per últim, tot i que no és una funció que els pertoqui directament, ja que és tasca d'administració, també acostumen a conèixer els llenguatges LDD i LCD doncs en les fases de desenvolupament de les aplicacions hauran de ser, en ocasions, els que defineixin els diferents esquemes de la base de dades.

Així doncs, els programadors d'aplicacions són usuaris dels llenguatges LC i LMD autosuficients i hostatjats en 3GL i 4GL, procedurals i no procedurals. També són usuaris, en menor grau, dels llenguatges LDD i LCD.

b) **Administradors de bases de dades**, que com el seu nom indica, són els que s'encarreguen d'*administrar* la base de dades, tasca que pot ser desenvolupada per una o varies persones. I què s'entén per *administrar*?

Les responsabilitats de l'administrador de la base de dades (*Database Administrator - DBA*) són, entre d'altres:

- Creació i manteniment dels diferents esquemes (externs, conceptual i intern) de la base de dades.

Aquesta tasca inclou la definició de les entitats, amb els seus atributs, dominis de valors, regles d'integritat, restriccions i procediments de validació, junt amb les correspondències externa/conceptual i conceptual/interna. Per l'esquema intern s'inclou l'estratègia d'emmagatzematge a seguir pels SGBD junt amb els mètodes d'accés a utilitzar.

Per a poder fer front a aquesta responsabilitat, l'administrador utilitza el llenguatge LDD i les eines que el SGBD pugui incorporar. En situacions de manteniment (canvis en el disseny conceptual) li caldrà dissenyar i executar processos de reorganització de les entitats existents i, per aquest motiu, precisarà dels llenguatges LMD i LC autosuficients i, fins i tot, de les corresponents versions hostatjades en un llenguatge de tercera generació.

- Control de l'accés a les dades.

S'inclou en aquest apartat tota la gestió d'usuaris (altes, baixes i modificacions) amb els diferents privilegis d'accés (consulta i/o actualització) sobre els diferents objectes de la base de dades.

Hi ha SGBD que contemplen la possibilitat de treballar amb grups d'usuaris i grups de privilegis, de manera que faciliten la tasca repetitiva de concedir i/o revocar un determinat conjunt de privilegis a un conjunt d'usuaris.

Per a poder fer front a aquesta responsabilitat, l'administrador utilitza el llenguatge LCD i les eines que el SGBD pugui incorporar.

- Gestió dels mecanismes de recuperació, definint les estratègies oportunes en funció de l'organització i de la utilització que se'n fa de la base de dades.
- Execució del *tunning* o de la *posta a punt*

Aquest és un punt fonamental en l'administració de les bases de dades ja que consisteix en cercar el millor funcionament possible de la base de dades en funció de la utilització que se'n fa. És a dir, s'estudia el rendiment (*performance*).

Així, de forma periòdica, l'administrador ha d'analitzar, mitjançant estadístiques i eines que proporciona el propi SGBD, la utilització que s'efectua de la base de dades i la resposta que el SGBD està donant, per tal d'efectuar les correccions oportunes i, en conseqüència, millorar el funcionament.

La posta a punt s'ha de repetir periòdicament ja que la utilització de la base de dades pot canviar per les necessitats de funcionament de l'organització. També és possible que l'organització no hagi modificat el seu funcionament però que el volum de dades hagi tingut una important modificació de manera que l'estructura d'emmagatzematge i accés (nivell intern) precisi d'algun retoc.

Dels diferents punts anomenats, se'n dedueix que aquest l'administrador ha de conèixer els llenguatges LDD i LCD i les versions autosuficients de LMD i LC. En certs SGBD necessitarà també els llenguatges LMD i LC hostatjats en llenguatges de tercera generació.

2) Usuaris no informàtics

És el grup més ampli d'usuaris de bases de dades. No estan obligats a tenir coneixements informàtics i, si en tenen, són escassos. Podem distingir-ne:

a) **Usuaris paramètrics**, en els que la interacció amb la base de dades s'efectua en base a l'execució d'aplicacions dissenyades per atendre específicament les seves necessitats. L'usuari, com a molt, ha de donar resposta a les preguntes que l'aplicació li formula.

Així, per exemple, quan estem en un caixer automàtic d'una entitat bancària, som usuaris paramètrics, ja que responem a les opcions i preguntes que una aplicació informàtica ens efectua i és aquesta la que interactua directament amb la base de dades de l'entitat bancària.

Aquest usuari no té per què conèixer cap llenguatge de base de dades ni cap eina de generació de codi.

b) **Usuaris finals**, que poden efectuar consultes no previstes, diguem-ne espontànies, a la base de dades, generant petits programes.

Aquest és el cas, per exemple, del personal d'administració d'una empresa, que té al seu abast la possibilitat d'efectuar consultes i/o actualitzacions sobre la base de dades sense utilitzar cap de les aplicacions desenvolupades a l'efecte. Per a tenir aquesta possibilitat és menester dues condicions:

- disposar d'una eina que ho possibiliti (eines de quarta generació i llenguatges LC-LMD autosuficients i implícits)
- conèixer el contingut de l'esquema extern de la base de dades sobre el que té accés (entitats, atributs, dominis de valors, relacions entre les entitats, regles d'integritat, restriccions,...)
- tenir les adequades capacitats per a executar la consulta i/o actualització sense provocar cap desgavell

És molt important disposar en una organització d'usuaris d'aquesta tipologia ja que permet interactuar, en fets puntuals, amb la base de dades, sense necessitat de tenir una aplicació informàtica que ens ho faciliti.

7. Diccionari de dades

Les dades són part fonamental d'una base de dades de manera semblant als passatgers en un tren o avió. És a dir, de res ens servirà una base de dades que es mantingui buida. Concloem, per tant, en que les dades són part fonamental d'una base de dades.

Ara bé, el tren i l'avió no únicament són els passatgers. També hi ha la maquinària que els fa córrer i volar. En les bases de dades, la maquinària és el SGBD. Encara més, en el tren i a l'avió, hi ha uns conductors o pilots que condueixen la maquinària. En les bases de dades, el SGBD és gestionat per l'administrador.

La maquinària del tren i de l'avió segueix un conjunt de regles de funcionament, algunes mecàniques i la majoria, avui en dia, electròniques. Aquestes regles de funcionament estan enregistrades en els circuits electrònics i en les memòries disposades a aquestes finalitats. Els dissenyadors dels aparells (enginyers) són els responsables d'aquestes regles de funcionament.

En els SGBD passa quelcom semblant. El SGBD ha de regir-se per unes *regles* a l'hora de gestionar una base de dades. Aquestes *regles* han estat dissenyades pels analistes i comunicades al SGBD per l'administrador. Les *regles* a què ens referim són el conjunt de definicions dels esquemes (entitats, atributs, dominis, regles d'integritat, restriccions, procediments de validació,...), correspondències entre els esquemes, estratègies d'emmagatzematge físic i mètodes d'accés,... Tota aquesta informació ha d'estar emmagatzemada en algun lloc de manera que el SGBD hi tingui accés.

El **diccionari de dades** és l'espai on resideix el conjunt d'informació necessària per la gestió de la base de dades a càrrec del SGBD.

Cada BD té el seu diccionari de dades. Un SGBD tindrà accés a tants diccionaris de dades com BD gestioni. Cal observar que el diccionari de dades conté la informació que l'administrador de la BD ha hagut de comunicar al SGBD per la gestió de la BD. Recordem quina és aquesta informació a partir de les responsabilitats que tenia l'administrador de la BD:

- Definicions de les entitats, amb els seus atributs, dominis de valors, regles d'integritat, restriccions i procediments de validació.
- Definicions de les correspondències externa/conceptual i conceptual/interna.
- Definicions de l'estratègia d'emmagatzematge i dels mètodes d'accés a utilitzar.

- Definicions d'usuaris, grups d'usuaris, privilegis i grups de privilegis.

Observem que el diccionari de dades és el conjunt de dades necessàries per gestionar les dades de la base de dades. Per aquest motiu, en molts àmbits, s'acostuma a dir que és una **meta-base**⁷.

- **Qui té accés al diccionari de dades?**

La informació del diccionari usualment és introduïda per l'administrador de la BD i es genera a partir de les instruccions executades amb els llenguatges LDD i LCD i amb eines que aporta el mateix SGBD o que subministra, en paral·lel al SGBD, el fabricant del SGBD.

El propi diccionari contempla la possibilitat de permetre diferents tipus d'accés als usuaris que ell mateix té definits per l'accés a la base de dades. Així, es pot permetre als usuaris finals la consulta dels noms de les entitats i dels seus atributs per a facilitar la utilització dels llenguatges LMD i LC. També es pot permetre accés amb més privilegis als analistes i dissenyadors de la base de dades i d'aplicacions sobre la base de dades, ja que potser ells mateixos executaran alguns canvis en l'estructura de la base de dades utilitzant els llenguatges LDD i LCD i les eines existents.

- **Tipus de diccionaris de dades**

Segons la localització del diccionari de dades, classifiquem:

- *Diccionaris de dades independents o aïllats*, consistents en un conjunt d'arxius que es situen independentment de la base de dades de la qual en formen el diccionari i, fins i tot, es gestionen amb eines externes al propi SGBD. En l'actualitat se'n troben pocs.
- *Diccionaris de dades totalment integrats*, que són gestionats pel propi SGBD i formen part de la pròpia base de dades. És el tipus de diccionari emprat en la majoria de SGBD actuals.

8. Arquitectura client-servidor.

En la present unitat didàctica hem introduït un munt de conceptes involucrats en la gestió de bases de dades i que aniran apareixent directament o indirecta al llarg del present crèdit. El món de les bases de dades és, però, molt ampli i en continua expansió motivada per la creixent necessitat d'accedir a la informació i per l'evolució de les telecomunicacions. No hem tractat, fins el moment, els punts d'accés a les bases de dades.

Anomenem **client d'un SGBD** a qualsevol aplicació que permet establir un diàleg amb el SGBD per a accedir a la base de dades. El SGBD es constitueix en **servidor** que dona serveis als clients que ho precisin.

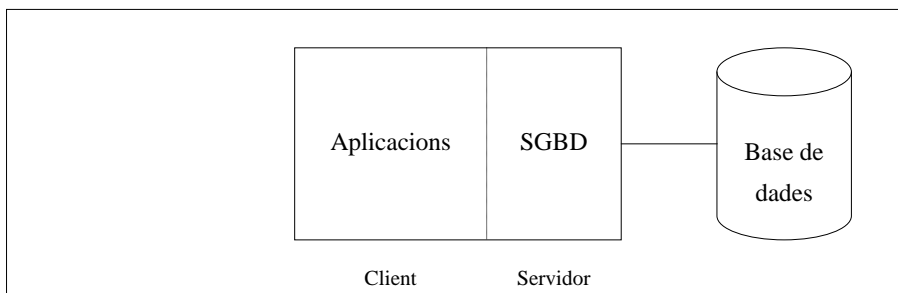
El propi SGBD incorpora clients per a facilitar les tasques de l'administrador i possibilitar la utilització dels llenguatges LMD i LC per part dels dissenyadors d'aplicacions i dels usuaris finals.

⁷ **Prefix meta**

El prefix meta indica posterioritat o més enllà de... o per damunt de... Per aquest motiu s'utilitza per a definir el diccionari de dades, ja que es tracta d'una base de dades utilitzada per gestionar la base de dades. És a dir, és una base de dades que està per damunt de la base de dades que veritablement es vol gestionar.

Així, el producte *Access XX* de la família *Microsoft Office* és un conjunt d'aplicacions clients del SGBD *Microsoft Jet*, tot i que s'acostuma a anomenar *SGBD Access XX*. El SGBD anomenat *Oracle* incorpora el veritable SGBD i moltes eines que en són clients, com *SQL*Plus*.

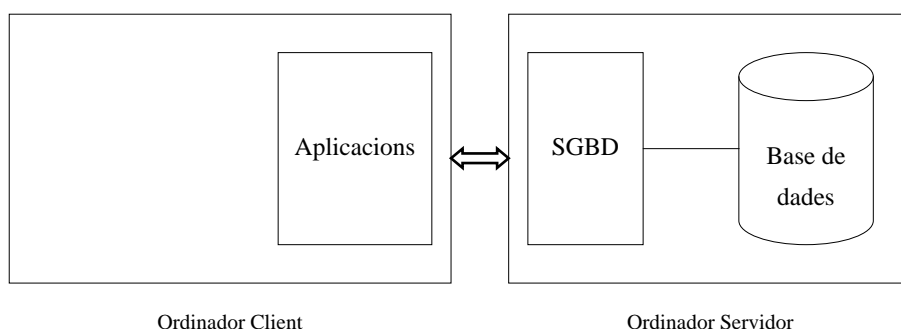
Podem veure el funcionament d'un SGBD i de les aplicacions que en requereixen els seus serveis, sota l'esquema de la dreta.



Aquesta arquitectura client-servidor a nivell de gestió de bases de dades es pot donar en una única màquina (SGBD i aplicacions s'executen en el mateix ordinador) o en diferents màquines. En aquest darrer cas, hi ha diferents possibilitats:

- Un ordinador actuant de servidor de dades (conté la base de dades i el SGBD) i diferents ordinadors actuant de clients (contenen les aplicacions que interactuen amb el SGBD).

Tot i que en l'esquema següent s'observa una única màquina client, la majoria de les vegades hi ha moltes màquines clients i una màquina servidora. En l'actualitat, el concepte d'arquitectura client/servidor s'identifica amb aquesta darrera situació, ja que en tenir separats els ordinadors que contenen les aplicacions -clients- de l'ordinador que conté les dades -servidor-, la distinció és clara i tothom reconeix una *arquitectura client-servidor*.



Cal remarcar, però, que el concepte d'arquitectura client-servidor no s'ha de vincular a la distinció entre màquines, sinó a la distinció entre aplicacions -clients- que utilitzen els serveis d'un SGBD -servidor- per accedir a la(es) base(s) de dades.

En l'esquema anterior s'observa una màquina client que conté aplicacions clients i una màquina servidora que conté el SGBD. Aquesta podria contenir, també, les aplicacions clients (com en l'esquema presentat inicialment) de manera que ella mateixa actués com a client i servidora. De fet, aquesta situació és molt probable, ja que l'administrador de la base de dades utilitza aplicacions clients per a desenvolupar la seva feina les quals segurament resideixen en la mateixa màquina on hi ha el SGBD.

- Variis ordinadors actuant de servidors de dades i diferents ordinadors actuant de clients (contenen les aplicacions que interactuen amb el SGBD).

Aquesta és la situació que es planteja quan la base de dades ha d'estar geogràficament distribuïda entre diferents punts. En tal situació es parla de bases de dades distribuïdes i/o replicades. Aprofundirem el concepte en el següent apartat d'aquesta mateixa unitat didàctica.

Així doncs, el concepte d'*arquitectura client-servidor*, àmpliament difós en la dècada dels noranta, distingeix les aplicacions i el(s) servidor(s) de dades. Com s'arriba a aquesta arquitectura?

Aquesta arquitectura va substituir l'arquitectura existent des dels anys setanta, consistent en grans ordinadors que actuaven de servidors de dades i que contenien les aplicacions a executar. Els usuaris executaven aquestes aplicacions des de terminals (conjunt format per pantalla i teclat connectats a l'ordinador central directament o per línies telefòniques punt a punt). Amb l'evolució, en els anys vuitanta, dels microordinadors, va iniciar-se la substitució dels *terminals no intel·ligents* per microordinadors que possibilitaven la mateixa gestió dels terminals (via programes d'emulació) i, a més, permetien a l'usuari l'execució d'altres tipus de programari de millors prestacions que les que oferien les aplicacions dels grans ordinadors, com per exemple, en el tema ofimàtic.

En generalitzar-se la utilització dels microordinadors en els sistemes informàtics i en assolir aquells unes prestacions cada cop més importants, els fabricants de programari arriben a la conclusió que poden utilitzar la seva potència per a executar les aplicacions clients del SGBD i així alliberar recursos en l'ordinador que conté el SGBD i la base de dades. En aquest moment el concepte d'*arquitectura client-servidor* més amunt detallada deixa d'utilitzar-se a nivell de programari (aplicacions clients del SGBD -aplicació servidora-) per passar a utilitzar-se a nivell de maquinari (màquines clients de màquina servidora). En el fons continua sent vàlida la utilització inicial del concepte, ja que les màquines clients ho són pel fet d'executar les aplicacions clients de l'aplicació servidora que resideix en la màquina servidora.

Amb aquest nou funcionament, les aplicacions clients s'instal·len, en part o en la seva totalitat, en les màquines clients (microordinadors), fet que provoca noves situacions:

- Cal instal·lar i mantenir cada aplicació client en cada màquina client des de la que es vulgui executar, enlloc de tenir-la instal·lada i mantinguda una única vegada en la màquina que conté el SGBD.
- L'usuari de microordinadors té accés a la configuració de la seva màquina i hi instal·la altres aplicacions de propòsit general i específic dins l'organització i, fins i tot, per propòsits personals, les quals poden interferir en el correcte funcionament de les aplicacions clients.

Les dues situacions provoquen l'augment de personal dedicat a mantenir en correcte funcionament el sistema informàtic. Per tot això, l'*arquitectura client-servidor* més amunt detallada i que anomenarem *de dos nivells*, ha evolucionat cap l'estai actual: *arquitectura client-servidor de tres nivells*.

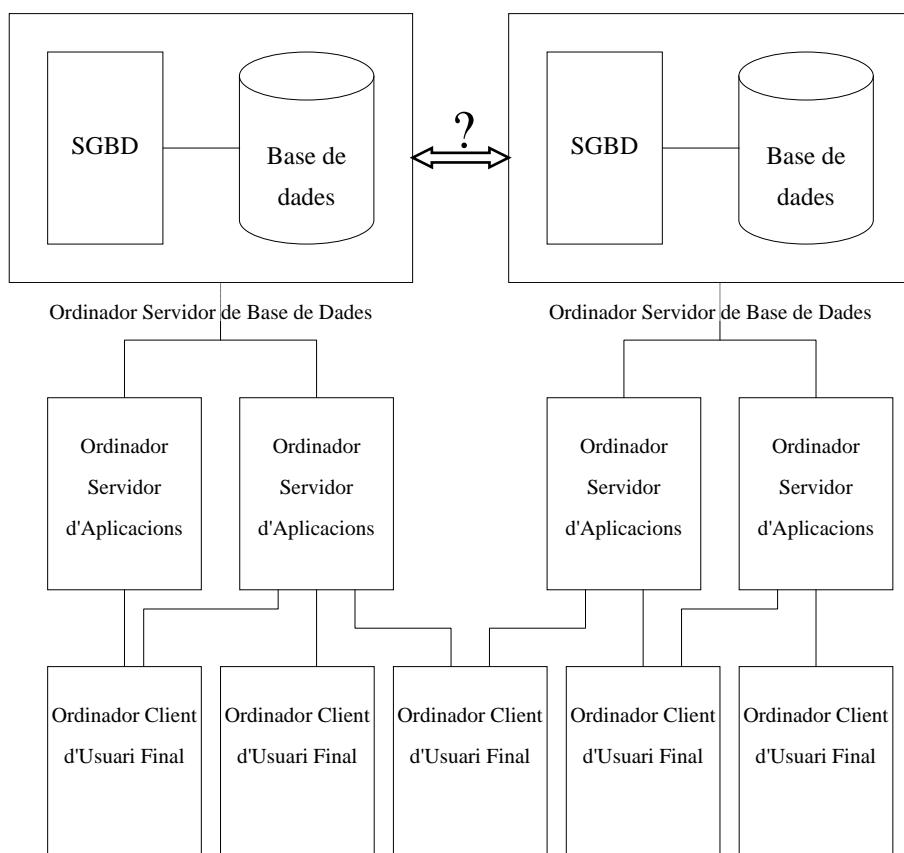
En aquesta nova arquitectura es manté el concepte de servidor(s) de base(s) de dades (una o varies màquines, segons es tracti de base de dades centralitzada o distribuïda o replicada, tal i com veurem en el proper apartat) però les aplicacions clients es distribueixen en dos tipus de màquines: màquines servidores d'aplicacions i màquines d'accés a les aplicacions.

Les màquines servidores d'aplicacions són ordinadors que contenen les aplicacions clients dels SGBD. Una determinada aplicació client pot estar en una o varies màquines servidores d'aplicacions. De manera anàloga al cas anterior, cal fer el manteniment de les diferents màquines servidores d'aplicacions, però aquestes són en número molt inferior al que ens trobàvem en l'altre cas i no hi ha els problemes referents a que els usuaris en modifiquin la configuració o instal·lin altre programari que interactui negativament amb les aplicacions.

I llavors, què tenen els usuaris? Des d'on executen les aplicacions? Doncs des de les seves màquines, amb molta o poca capacitat de procés, les quals utilitzen per accedir als servidors d'aplicacions. És a dir, d'alguna manera es torna al concepte que l'usuari accedeix a les aplicacions per medi de terminals poc intel·ligents. Avui en dia, amb l'explosió de l'accés a internet, interessa que els usuaris tinguin màquines amb qualsevol navegador instal·lat (segons preferències de l'usuari) el qual

permet l'accés -via internet o intranet- a les aplicacions web que es troben en execució en els corresponents servidors d'aplicacions.

En l'esquema següent es pot observar una típica situació d'aquesta arquitectura. En un primer nivell es disposa d'un parell de SGBD que poden estar enllaçats o no (d'aquí l'interrogant en l'enllaç entre elles). En un segon nivell es té quatre servidors d'aplicacions, cadascun dels quals té aplicacions clients d'un dels SGBD. Un servidor d'aplicacions podria tenir aplicacions clients de diferents SGBD situats en diferents servidors de bases de dades. En el tercer nivell estan els ordinadors dels usuaris, els quals accedeixen als servidors d'aplicacions que interessen. L'execució de les aplicacions la duen a terme els servidors d'aplicacions. Les màquines dels usuaris finals únicament funcionen com a interfície entre l'usuari i l'aplicació en execució. Aquest és el típic funcionament actual que l'usuari es troba en accedir via internet a pàgines web des de les que pot comprar, efectuar reserves,... Tot i que aquest funcionament ja existeix en el món internet, també es pot utilitzar en intranets. Una mateixa aplicació es pot estar executant en diferents servidors d'aplicacions. L'usuari final establirà connexió amb el servidor d'aplicacions pel que tingui un accés més ràpid i directe.



9. Bases de dades distribuïdes i replicades

Aquest concepte ja ha aparegut en l'anterior apartat. En moltes ocasions serà necessari i/o convenient tenir les dades d'una mateixa base de dades repartides i/o repetides en diferents punts. En tal situació, cada punt ha de tenir el seu SGBD per a gestionar les corresponents dades i els diferents SGBD han d'estar connectats i treballar de forma conjunta.

Una **base de dades distribuïda i/o replicada** és un conjunt de dades localitzades en diferents màquines, cadascuna de les quals pot participar en l'execució de les transaccions que gestionen dades d'un o varis conjunts.

Cadascuna de les màquines on hi ha part de la base de dades és un **node** i ha de tenir un SGBD per a la gestió de les seves dades. Els diferents nodes han de formar part d'una xarxa d'àrea local o d'una xarxa d'àrea estesa o d'una barreja d'ambdues situacions.

No es pot confondre el concepte **node** amb el concepte **punt d'accés** que s'utilitza per fer referència al punt des del que es sol·liciten transaccions sobre la base de dades i que, com molt bé podeu deduir, tan pot estar situat en el propi node com en estacions de treball remotes.

Quan es vol remarcar que una base de dades no és distribuïda ni replicada, sinó que tota ella es troba emmagatzemada en una màquina sota la supervisió d'un únic SGBD es pot dir que es tracta d'una base de dades **centralitzada**.

Notarem **distribució de dades** quan les dades es troben repartides en diferents nodes i notarem **rèplica de dades** quan les dades es troben repetides en diferents nodes, provocant redundància.

Una base de dades pot ser distribuïda i replicada a la vegada, és a dir, pot tenir dades repartides entre diferents nodes (distribució) i dades repetides en diferents nodes -alguns o tots- (rèpliques). Tot i que ja deveu estar intuïnt els problemes que pot suposar tenir una base de dades replicada pel fet de les actualitzacions de les dades repetides, la utilització de rèpliques pot estar justificada per aquelles dades que s'utilitzen molt en nodes diferents.

Les bases de dades distribuïdes i/o replicades poden tenir el mateix o diferent SGBD en els diferents nodes. Es necessita, però, un SGBD global (SGBDG) encarregat d'organitzar les comunicacions entre els diferents SGBD. La problemàtica del SGBDG varia molt segons els diferents SGBD parlin o no el mateix llenguatge. Per tant, l'interès radica no en el fet que els diferents nodes es regeixin pel mateix SGBD sinó en el fet que els diferents SGBD tinguin uns llenguatges semblants o, si més no, semblants en les parts on s'ha d'establir comunicació. Així doncs, distingirem:

- **SGBDG homogènis**, on els SGBD de tots els nodes disposen de llenguatges semblants.
- **SGBDG heterogènis**, on els nodes utilitzen SGBD amb llenguatges força diferents. En aquesta situació, la gestió del SGBDG és força complicada.

Degut a que la informació es troba distribuïda en diferents nodes i que des de cadascun d'ells es pot gestionar dades del seu node o dades d'altres nodes, apareixen els següents tipus de transaccions:

- **Transaccions locals**, que gestionen dades situades en el propi node.
- **Transaccions globals**, que gestionen dades situades en nodes diferents.

Per finalitzar aquesta breu introducció a la temàtica de les bases de dades distribuïdes i/o replicades, fem esment a com es produeix el naixement d'una tal base de dades, el qual pot ser fruit de:

- Disseny inicial que preveu la distribució i/o rèpliques de dades entre diferents nodes: **base de dades autònoma**.
- Agrupament de diferents bases de dades independents, existents amb anterioritat, majoritàriament centralitzades però amb possibilitat inicial de distribució i/o rèpliques en alguna d'elles, que s'han de continuar gestionant de manera relativament independent: **base de dades federal**.

La distinció entre bases de dades autonòmiques i federals és independent, en principi, del fet que el SGBDG sigui homogeni o heterogeni. La realitat és, però, que una base de dades federal té moltes possibilitats de tractar-se amb un SGBDG heterogeni, mentre que en una base de dades autonòmica hi ha més marge per escollir el SGBDG a utilitzar i per simplicitat sembla lògic escollir un SGBDG homogeni.

9.1. Possibilitats de distribuir i/o replicar les dades

En treballar amb bases de dades distribuïdes i/o replicades es pot tenir diferents tipologies segons sigui el repartiment que de les dades se'n fa en els diferents nodes. Les possibilitats existents són:

- **Amb multiplicació.** Totes les dades estan duplicades en cada node amb la finalitat de reduir els costos de comunicació i millorar el temps d'execució, ja que cada operació de consulta es pot executar en cada node. Presenta l'inconvenient de la quantitat d'espai necessari per tenir les dades duplicades, la redundància extrema entre diferents nodes i els problemes d'actualització de les dades en els diferents nodes. Aquesta tipologia no s'acostuma a utilitzar.
- **Amb partició.** Consisteix en que les dades es troben repartides entre diferents nodes sense cap tipus de rèplica. D'aquesta manera, qualsevol actualització sobre l'entitat s'ha d'executar únicament en el node que la conté. Presenta els inconvenients del cost de temps en efectuar operacions que afecten a dades de diferents nodes i de la caiguda total de la base de dades quan un node es queda sense servei.
- **Amb un node que conté tota la base de dades i altres nodes que en contenen parts replicades.** Una organització d'aquest estil es justifica per reduir al màxim els costos de comunicació en les transaccions globals. Presenta l'inconvenient dels processos d'actualització en les parts replicades. La base de dades pot quedar sense servei si cau el node que conté tota la base de dades.
- **Amb informació replicada en nodes seleccionats.** Consisteix en tenir la informació emmagatzemada, majoritàriament, en els nodes des dels que s'hi accedeix, és a dir, la informació accedida des de varis nodes estarà ubicada en tots ells (rèpliques), mentre que la informació accedida pràcticament des d'un únic node, estarà emmagatzemada únicament en ell. Aquest tipus d'organització continua tenint els inconvenients de l'actualització de les dades replicades i de l'accés a dades que es troben en diferents indrets.

9.2. Fragmentació

En l'apartat anterior hem estat presentant diferents possibilitats de combinar els conceptes de *repartir* i *replicar*. En totes elles hem parlat genèricament de *dades*, és a dir, *repartim i/o replicàvem dades*.

Ja sabem que en una base de dades hi representarem entitats amb els seus atributs (com en els arxius) i a més les integrarem i compartirem. Ara, però, estem repartint i/o replicant les dades entre diferents nodes. Ens apareixen les preguntes: *què entenem per dada repartida i/o replicada?*, *"dada" és igual a "entitat"?*, *"dada" és igual a "conjunt d'entitats"?*, *"dada" és igual a "part d'una entitat"?*,... La resposta no és única. Hi ha diferents possibilitats.

Anomenem **fragment** d'una entitat distribuïda i/o replicada a cada subconjunt d'informació que es distribueix i/o replica.

Hi ha diferents possibilitats d'escollir els fragments d'una entitat. Per exemplificar-les, considerem l'entitat *persones* amb les següents instàncies:

Dni	Nom	Data Naixement	Altura	Pes	Localitat	Comarca
450	Marius	09-12-1980	169	65	Igualada	Anoia
870	Empar	19-03-1936	165	70	Barcelona	Barcelonès
920	Manel	25-04-1967	168	74	Jorba	Anoia
225	Guifré	07-02-1995	100	16	Igualada	Anoia
342	Josep	09-12-1960	175	85	Olesa de Montserrat	Baix Llobregat
425	Regi	05-12-1992	105	18	Olesa de Montserrat	Baix Llobregat
972	Manela	24-11-1932	155	65	Igualada	Anoia
782	Elvira	09-01-1965	165	60	Barcelona	Barcelonès
645	Joan	07-01-1924	174	75	Barcelona	Barcelonès
123	Benet	03-08-1931	158	50	Igualada	Anoia
500	Maite	27-01-1958	165	48	Igualada	Anoia

1) **Fragmentació horitzontal**, on els fragments són conjunts d'instàncies de l'entitat.

Situant-nos en l'exemple, si suposem que cada comarca té un node de la base de dades, podríem arribar a la conclusió de fragmentar l'entitat *persones* de manera que cada comarca emmagatzemi les "seves" persones. Així, tindríem tres fragments:

Dni	Nom	Data Naixement	Altura	Pes	Localitat	Comarca
450	Marius	09-12-1980	169	65	Igualada	Anoia
920	Manel	25-04-1967	168	74	Jorba	Anoia
225	Guifré	07-02-1995	100	16	Igualada	Anoia
972	Manela	24-11-1932	155	65	Igualada	Anoia
123	Benet	03-08-1931	158	50	Igualada	Anoia
500	Maite	27-01-1958	165	48	Igualada	Anoia

Dni	Nom	Data Naixement	Altura	Pes	Localitat	Comarca
870	Empar	19-03-1936	165	70	Barcelona	Barcelonès
782	Elvis	09-01-1965	165	60	Barcelona	Barcelonès
645	Joan	07-01-1924	174	75	Barcelona	Barcelonès

Dni	Nom	Data Naixement	Altura	Pes	Localitat	Comarca
342	Josep	09-12-1960	175	85	Olesa de Montserrat	Baix Llobregat
425	Regi	05-12-1992	105	18	Olesa de Montserrat	Baix Llobregat

2) Fragmentació vertical, on els fragments són conjunts de valors d'atributs de l'entitat.

Situant-nos en l'exemple, podríem considerar que en un cert node hi ha les dades corresponents a la localització de les persones (dni, nom, localitat., comarca) i en un altre node hi ha dades mèdiques de les persones (dni, nom, data de naixement, pes, altura). Així, tindríem dos fragments:

Dni	Nom	Data N.	Alt	Pes	Dni	Nom	Localitat	Comarca
450	Marius	09-12-1980	169	65	450	Marius	Igualada	Anoia
870	Empar	19-03-1936	165	70	870	Empar	Barcelona	Barcelonès
920	Manel	25-04-1967	168	74	920	Manel	Jorba	Anoia
225	Guifré	07-02-1995	100	16	225	Guifré	Igualada	Anoia
342	Josep	09-12-1960	175	85	342	Josep	Olesa de Montserrat	Baix Llobregat
425	Regi	05-12-1992	105	18	425	Regi	Olesa de Montserrat	Baix Llobregat
972	Manela	24-11-1932	155	65	972	Manela	Igualada	Anoia
782	Elvis	09-01-1965	165	60	782	Elvis	Barcelona	Barcelonès
645	Joan	07-01-1924	174	75	645	Joan	Barcelona	Barcelonès
123	Benet	03-08-1931	158	50	123	Benet	Igualada	Anoia
500	Maite	27-01-1958	165	48	500	Maite	Igualada	Anoia

Evidentment hi ha d'haver algun atribut identificador que permeti enllaçar les parts separades de cada instància. En el nostre exemple suposem que és l'atribut *dni*. Observem, també, que un mateix atribut pot estar repetit en diferents fragments, com succeeix amb l'atribut *nom*.

3) Fragmentació mixta, que consisteix en considerar fragments verticals, algun dels quals poden, a la vegada, estar fragmentats horitzontalment.

Per finalitzar, notem que els tres tipus de fragmentació presentats són combinables amb les possibilitats de distribuir i replicar una base de dades.

9.3. Avantatges i inconvenients

En els apartats anteriors ja s'han anat intuït les avantatges i inconvenients de la gestió d'una base de dades distribuïda. Anem a resumir-les.

1) Entre les **avantatges**, cal considerar:

a) Reduir el cost en les comunicacions, ja que la majoria d'operacions es suposa que es realitzaran, si la base de dades està ben distribuïda, accedint a les dades del node local.

b) Certa autonomia en el node local, tot i que no es pot obviar que es forma part d'una organització superior.

c) Possibilitat de continuar el treball amb les dades emmagatzemades en el node local quan es produeix una caiguda d'algun altre node. En restablir-se el funcionament d'aquest, caldrà reintegrar-se a la xarxa amb el mínim de complicacions possible.

2) Entre els **inconvenients**, cal considerar:

a) Cost del programari, ja que el SGBDG ha de ser més potent

b) Actualització de les dades duplicades

c) Major possibilitat d'errors com a conseqüència d'una major concurrència des de diferents nodes a una mateixa dada.

d) Aconseguir la transparència d'ubicació de cara als diferents usuaris (aplicacions, dissenyadors i programadors d'aplicacions, usuaris finals).

e) Increment del temps de procés en les transaccions globals.

9.4. Gestió de bases de dades distribuïdes

En els següents apartats volem presentar una aproximació del funcionament intern que es produeix en el treball en bases de dades distribuïdes, des del moment en que l'usuari provoca una consulta i/o actualització a la base de dades (ja sigui directament via LMD-LC autosuficient o des d'una aplicació que utilitza LMD-LC hostatjat) fins el moment en que l'usuari rep la resposta proporcionada pel sistema gestor de base de dades.

9.4.1. Transaccions globals i locals

En un apartat anterior hem introduït la distinció entre transacció global i local, diferenciant-les en funció dels nodes als que han d'accedir per donar resposta a la sol·licitud de l'usuari.

Davant una transacció sobre una base de dades distribuïda (provocada per una sol·licitud d'usuari en un determinat punt d'accés), el SGBDG reconeix si es tracta d'una transacció global (que ha d'accedir a diferents nodes) o d'una transacció local (per un node en concret). En aquest darrer cas passarà la transacció al SGBD del node afectat per a que actuï en conseqüència i n'esperarà el corresponent resultat. Si es tracta, però, d'una transacció global, el SGBDG la descompon en subtransaccions de les que n'esperarà els corresponents resultats per tal de poder confeccionar el resultat final per la transacció inicial.

Cadascuna de les subtransaccions pot ser, a la vegada, global (encara intervenen diferents nodes) o local per un node en concret. El SGBDG torna a iniciar la gestió de les globals i traspasa les locals als corresponents SGBD. Els resultats corresponents a cada subtransacció es generen amb les dades a les que s'accedeix directament i amb els resultats d'altres subtransaccions que la pròpia subtransacció havia generat. D'aquesta manera s'arriba a confeccionar el resultat de la transacció inicial.

Si la transacció executa una actualització de dades existents en fragments replicats en altres nodes, es genera una subtransacció per actualitzar les rèpliques afectades.

En cas de transacció d'actualització que afecti a diferents nodes caldrà assegurar que l'actualització es produeixi totalment o s'anul·li en la seva totalitat, de manera que si s'avorta una subtransacció que actualitzava, també s'avortin la resta de

subtransaccions. Una excepció a aquest fet pot permetre's en les subtransaccions d'actualització a rèpliques, les quals es permet, en certes situacions que s'efectuïn en temps diferit⁸.

9.4.2. Arquitectura d'un SGBD distribuït

En un apartat anterior hem presentat l'arquitectura de tres nivells (intern, conceptual i extern) com l'arquitectura adoptada pels sistemes gestors de bases de dades. Hem vist, també, que ens proporcionava la independència física i lògica de les dades.

Com s'ha de modificar aquesta arquitectura en un SGBD distribuït?

La descripció del contingut de tota la base de dades distribuïda i l'especificació de quins fragments hi ha a cada node forma el que s'acostuma a anomenar **esquema global**, el qual haurà d'estar emmagatzemat en cada punt d'accés, ja que cal tenir-hi accés ràpid per poder descompondre cada transacció en subtransaccions.

L'arquitectura és sensiblement diferent segons es tracti d'una base de dades autonòmica o federal.

1) Base de dades autonòmica

En una base de dades autonòmica el disseny de la base de dades s'efectua preveient que les dades estaran distribuïdes i/o replicades en diferents punts d'emmagatzematge. Per tant, en la fase de disseny es defineixen:

- L'esquema conceptual, corresponent a tota la base de dades.
- L'esquema global, que conté l'esquema conceptual més els criteris de distribució i replicació.

Aquesta informació determina els esquemes conceptuals dels diferents nodes, els quals es podran generar de manera quasi automàtica.

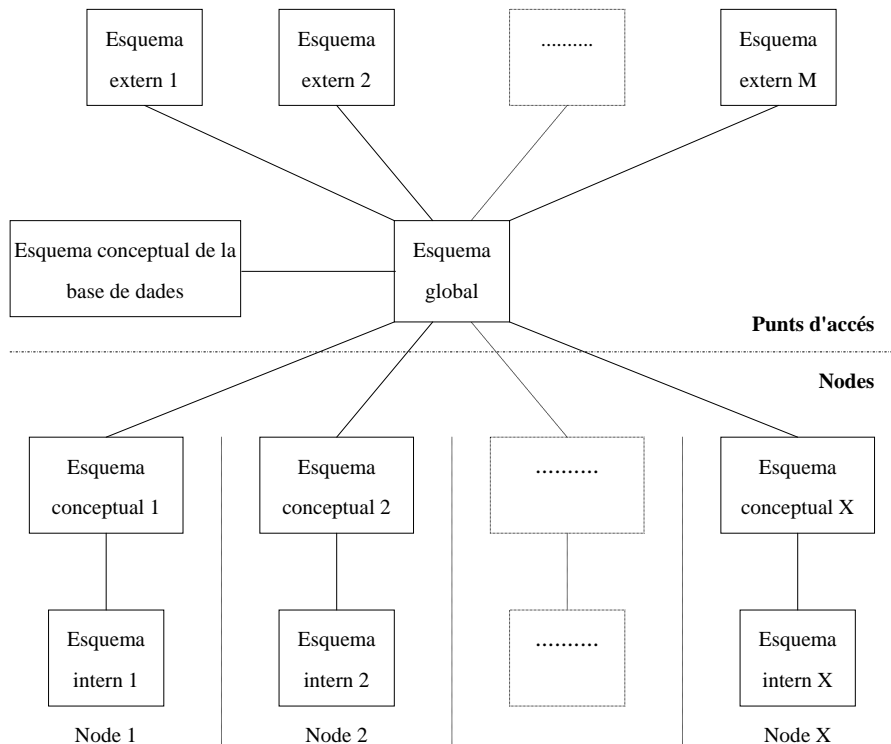
Els esquemes interns es defineixen per cada node en funció, també, dels criteris establerts en l'esquema global.

Els esquemes externs es defineixen a partir de l'esquema global, a diferència d'una base de dades centralitzada en la que es defineixen a partir de l'esquema conceptual.

En la figura següent es pot observar un cas de X nodes en una base de dades autonòmica, gestionada sota un esquema global que es correspon amb l'esquema conceptual de la base de dades. En tal situació, els M esquemes externs són dissenyats sobre l'esquema global.

⁸ **Rèpliques discordants**

En certes situacions de bases de dades distribuïdes es permet que fragments replicats no estiguin actualitzats. En dites situacions, hi ha subtransaccions pendents d'execució les quals s'executen en temps diferit.



Observem que els punts d'accés han de tenir la informació referent a l'esquema global i als esquemes externs que es pot accedir. Una transacció sol·licitada en un punt d'accés serà tractada dins l'esquema extern que correspongui i descomposta en el propi punt d'accés en diverses subtransaccions, cadascuna de les quals serà adreçada al node que pertorqui. Cada node crearà un esquema extern per la subtransacció rebuda, de manera que el SGBD del node actuarà com si es tractés d'una transacció local.

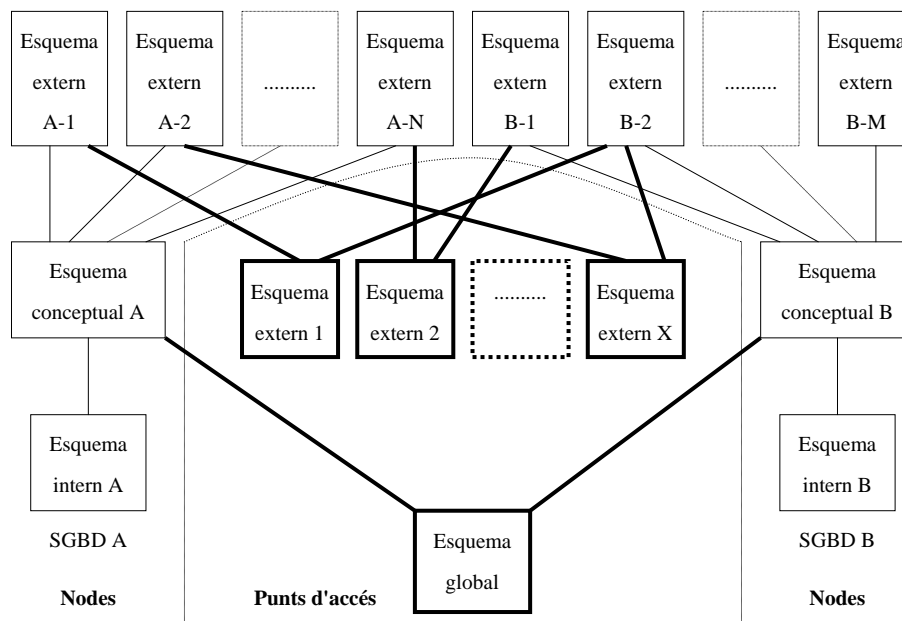
2) Base de dades federal

En una base de dades federal es parteix de diferents bases de dades ja existents. Per tant, cadascuna ja té els seus esquemes conceptual, intern i externs. En tal situació, l'arquitectura del SGBD distribuït és més complicada.

L'esquema global es construeix a partir dels esquemes conceptuals de les diferents bases de dades que es federen. Apareixen, segurament, conflictes relatius a sinònims i homònims, que es s'acostumen a resoldre afegint un prefix que indiqui la base de dades d'origen.

Els esquemes externs es podrien definir a partir dels esquemes externs existents a les diferents bases de dades que es federen.

Realitzar un dibuix que esquematitzi la situació plantejada no és tan simple com en el cas de la base de dades autonòmic. Mostrem, a la pàgina següent, un exemple, pel cas de una federació de dues bases de dades.



9.4.3. Components d'un SGBD distribuït

És de calaix que en un SGBD distribuït es necessita un gestor de comunicacions, és a dir, un sistema en xarxa, que garanteixi que tots els missatges arriben al seu destí en l'ordre en què s'han anat produint i sense duplicació. Considerant assolida aquesta situació, tenim com a components propis del SGBD:

- En els **punts d'accés**:
 - Els **catàlegs**, on s'emmagatzema l'esquema global i els esquemes externs vinculats al punt d'accés.
 - El **processador de transaccions font**, que comprova que només accedeixin a la xarxa els usuaris autoritzats i obtinguin la informació requerida, així com en quins punts es troben les dades demanades. A més, descompon cada transacció en subtransaccions tenint en compte la màxima reducció d'utilització de les línies de comunicacions, compila cada subtransacció totalment o parcial i li genera un pla d'execució.
 - El **gestor de transaccions**, que controla a nivell global el pla d'execució de cada transacció.
- En els **nodes**:
 - El SGBD que correspongui
 - El **gestor de subtransaccions**, que rep les transaccions, les acaba de compilar si és menester, i les executa. En finalitzar la seva execució, ho comunica al gestor de transaccions del punt d'accés que va enviar la transacció i envia els resultats obtinguts als nodes explicitats en el pla d'execució.