Sistemes de base de dades

Jaume Raventós Moret

PID_00168107

Universitat Oberta de Catalunya
www.uoc.edu

Índex

Introducció

Les bases de dades són elements imprescindibles en moltes activitats de la vida quotidiana. Per exemple, en ingressar o retirar diners del compte bancari, en reservar un vol o una habitació d'hotel, en accedir al catàleg d'una biblioteca o en comprar productes en un supermercat es requereix que algú interactuï amb una base de dades.

Des que es van començar a introduir els ordinadors per a automatitzar la gestió de les empreses utilitzant un llenguatge de programació, l'evolució dels sistemes d'informació ha tingut una repercussió considerable en la gestió de les dades. Els **sistemes informatitzats de gestió de dades** es van concebre per la necessitat d'emmagatzemar una gran quantitat d'informació de manera ràpida, senzilla i fiable per a la consulta i la utilització posteriors en qualsevol moment sense necessitat de desplaçar-se a estances dedicades a arxivar documentació.

Remissions a altres parts del contingut

En les remissions a altres continguts (referències creuades), si no s'especifica el contrari, els apartats referenciats es troben en el mateix mòdul didàctic.

Les **bases de dades** són una de les eines més àmpliament difoses en la societat de la informació i un producte estratègic de primer ordre atès que constitueixen el **fonament dels sistemes d'informació**. El seu ràpid creixement ha donat lloc a una industria específica de productivitat excepcional i impacte econòmic impressionant.

La societat de la informació es caracteritza pel maneig d'un volum desmesurat i acceleradament creixent d'informació de tot tipus que comporta mètodes i tècniques d'emmagatzematge i accés a les dades radicalment diferents dels utilitzats tradicionalment. En les últimes dècades han aparegut aplicacions noves de sistemes de base de dades que permeten emmagatzemar informació audiovisual, extreure i analitzar informació útil de bases de dades grans per a la presa de decisions, controlar processos industrials, localitzar informació a través d'Internet.

Noves aplicacions de sistemes de base de dades

Estem parlant de:

- BD multimèdia.
- BD *data warehouse*.
- BD web.

Per tal de poder desenvolupar els continguts exposats en aquest mòdul i entendre els fonaments de la tecnologia de bases de dades, convé tenir clars uns **conceptes bàsics** que definirem breument en el primer apartat.

Farem un repàs de l'**evolució de la gestió de dades**, des dels sistemes manuals de fitxes en targetes de cartolina, passant per la gestió informatitzada amb fitxers de dades, fins als sistemes de base de dades. A continuació, descriurem les **estructures de dades** més utilitzades en els sistemes d'informació, les quals permeten especificar la manera d'organitzar les dades per tal d'obtenir un rendiment raonable en el seu emmagatzematge, el seu tractament i la seva recuperació.

Seguidament, assenyalarem els **problemes** que presenta la **utilització dels sistemes de gestió de dades basats en fitxers**, que justifiquen l'ús de bases de dades, de les quals enumerarem els **objectius** i les **característiques** que han de complir per a ser considerades com a tals. A partir d'aquestes característiques extraurem els **avantatges** que presenten les bases de dades enfront dels fitxers de dades tradicionals. També comentarem els **inconvenients** que comporta la utilització de bases de dades i els **casos en què és millor utilitzar un sistema de fitxers** en comptes d'un sistema de base de dades.

Identificarem els principals **elements que componen un sistema de base de dades** en la seva concepció més amplia. Això és, el contingut (la base de dades), el programari (bàsicament, el sistema de gestió de bases de dades o SGBD), el maquinari i el personal humà. Donada la seva importància per al funcionament adequat de l'SGBD estudiarem les diverses **tipologies d'usuari** que interactuen amb la base de dades. Veurem que els relacionats més directament amb la base de dades són l'usuari final, el programador i l'administrador de la base de dades.

Més endavant, tractarem els conceptes de **model, esquema i estat de la base de dades.** Una característica fonamental d'un sistema de base de dades és que proporciona un cert nivell d'abstracció de les dades perquè oculta detalls del seu emmagatzematge. Veurem que un model de dades permet aconseguir el procés d'abstracció que condueix del món real al món de les dades, distingint entre la descripció de la base de dades i la mateixa base de dades; és a dir, entre l'esquema i l'estat de la base de dades.

L'**abstracció** permet considerar l'essència d'una realitat amagant els detalls o destriar els seus elements per a considerar-los aïlladament, de manera que se'n redueix la complexitat i se'n facilita la comprensió. Introduirem els diferents **tipus d'abstracció** que ofereixen els models de dades per a facilitar la representació de les dades en el disseny de bases de dades i que permeten establir vincles entre els elements d'un model.

Considerarem l'**arquitectura dels sistemes de gestió de bases de dades** des de diferents enfocaments per a entendre les seves característiques fonamentals. Tractarem, en especial, l'**arquitectura de tres nivells**, que estableix una divisió de la base de dades segons la perspectiva des de la qual aquesta és vista en nivells d'abstracció, que es corresponen amb els tres grups principals d'usuaris. A més, analitzarem les correspondències entre nivells i el concepte de lligadura.

Donarem una visió integrada dels elements i les característiques d'un sistema de base de dades entès com el sistema que integra l'SGBD i la base de dades en un sentit ampli en el que anomenem **estructura global del sistema de base de dades**.

En l'apartat següent, tractarem l'**emmagatzematge de bases de dades**, és a dir, com s'organitzen les dades en suports d'emmagatzematge secundari, els quals permeten desar grans quantitats d'informació de manera persistent, de manera que l'SGBD pugui recuperar, actualitzar i processar les dades quan sigui necessari. Per tal de comprendre l'**accés de l'SGBD a les dades**, veurem l'esquema general del flux de dades i de control que segueix el procés d'execució d'una consulta feta per un programa d'usuari a l'SGBD amb l'ajuda del sistema operatiu subjacent.

A partir dels objectius de les bases de dades, deduirem les **funcions** que ha d'executar l'SGBD, l'anàlisi de les quals ens ajudarà a determinar els **components** que ha de tenir l'SGBD per a complir-les.

A continuació, classificarem segons diversos criteris els **llenguatges de base de dades** que permeten manejar la base de dades. Presentarem els diferents tipus d'**interfícies de l'SGBD**, moltes de les quals permeten ometre l'ús del llenguatge de base de dades. Estudiarem els mòduls de programari que componen el **nucli de l'SGBD**, els quals s'executen com a processos transparents per a l'usuari. Finalment, veurem com el nucli, que es pot considerar l'intèrpret entre l'usuari i les dades, s'ocupa de les tasques bàsiques de la base de dades, del treball amb el llenguatge de base de dades, del diàleg amb el sistema operatiu, etc.

Objectius

En el material didàctic d'aquest mòdul hi trobarem les indicacions per a assolir els objectius següents:

1. Aprendre els conceptes bàsics sobre els sistemes de base de dades.

2. Saber quins són els usos més importants de les bases de dades.

3. Entendre quina és l'estructura dels sistemes de gestió de bases de dades.

1. Alguns conceptes bàsics

Des de l'aparició dels sistemes informàtics, una de les seves principals aplicacions ha estat l'emmagatzematge i el tractament de grans quantitats de dades per a permetre'n la consulta i la utilització posteriors. Una base de dades, formada per una col·lecció de dades entre les quals s'estableix una relació, és una de les eines més utilitzades a l'hora de treballar amb grans volums d'informació.

L'objectiu d'aquest apartat és aclarir una sèrie de conceptes fonamentals en l'entorn de les bases de dades, definint-los breument. Això permetrà desenvolupar, més endavant, els aspectes següents:

- La problemàtica que presenta la utilització de fitxers de dades, que justifica l'ús de sistemes de base de dades.
- Els objectius, els avantatges i els inconvenients de les bases de dades.
- Els requisits que han de complir les bases de dades per a ser considerades com a tals.
- Els elements que componen un sistema de base de dades en la seva concepció més amplia; és a dir, incloent, juntament amb les dades i el programari que les gestiona, els equips necessaris i les persones que interactuen amb la base de dades.

A continuació, s'exposen alguns conceptes bàsics sobre bases de dades.

Informació

És l'agregació de diverses dades juntament amb les seves relacions o dependències per a accedir a un nivell de comprensió i enteniment més alt d'un fenomen determinat.

La informació és el **recurs bàsic per a la presa de decisions** i pot ser definida des de múltiples camps de coneixement que l'associen a problemàtiques i àmbits d'estudi molt diversos.

Dada

És la **unitat mínima bàsica d'informació** (o propietat que caracteritza un objecte, fenomen o esdeveniment) que, representada en un format adequat, es pot transmetre i/o processar per mitjans manuals o automàtics.

Una dada pot contenir caràcters alfabètics, magnituds numèriques, símbols especials, colors, imatges o, fins i tot, grups de símbols no aleatoris que representen quantitats, accions, objectes.

El concepte de dada en informàtica (més ampli que l'usat en ciències pures com la física o les matemàtiques) és un conjunt de símbols o caràcters emprats per a expressar o representar un valor numèric, un fet, un objecte o una idea de la manera adequada per al seu tractament per l'ordinador. Les dades es poden escriure en un teclat, trobar en la memòria principal, emmagatzemar en un disc.

Jerarquia de les dades

La jerarquia que segueixen les dades emmagatzemades en una base de dades s'exposa en el subapartat "El contingut: les dades" dins l'apartat "Elements d'un sistema de base de dades".

El procés informacional: dades, informació, coneixement

És útil recordar la gradació jeràrquica que habitualment s'estableix entre els conceptes de dada, informació, coneixement i saviesa o intel·ligència en l'anomenat procés informacional:

Dades → Informació → Coneixement → Saviesa

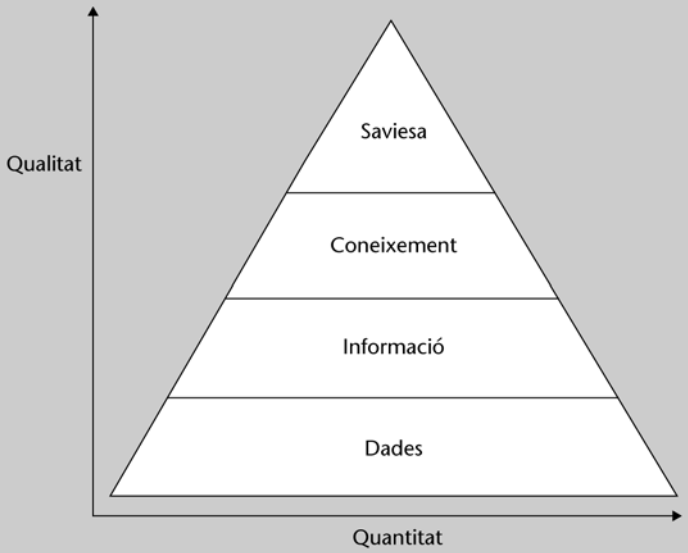
Coneixement: és el resultat de relacionar informacions per tal de construir teories.

Saviesa: implica un grau més alt de consciència de l'abast i dels límits del coneixement propi i una capacitat de contextualització més àmplia.

El contingut de la informació és el coneixement que aporta la informació que es té d'un fenomen determinat a través de les dades que se n'han obtingut i és utilitzat en la presa de decisions.

Aquest procés es pot expressar en l'anomenada *piràmide informacional,* formada per quatre nivells, els quals impliquen una jerarquia de les variables qualitat i quantitat:

Figura 1. Piràmide informacional



Des del punt de vista de la gestió d'informació, Iraset Páez Urdaneta (1992) planteja la reflexió següent:

"En l'antiguitat, l'home occidental volia ser savi; després l'home modern va voler ser coneixedor; l'home contemporani sembla acontentar-se amb estar informat (i possiblement l'home del futur no estigui interessat en altra cosa que a tenir dades)".

Font: Páez Urdaneta, Iraset (1992). *Gestión de la inteligencia, aprendizaje tecnológico y modernización del trabajo informacional. Retos y oportunidades*. Caracas: Instituto de Estudios del Conocimiento de la Universidad Simón Bolívar/CONICIT.

Interpretació de dades i conversió en informació

Les dades han de ser interpretades (incorporant significat) per tal que es converteixin en informació útil. Per tant, una dada és qualsevol element d'informació que no té significat fora del seu context.

La relació entre aquests conceptes es decriu amb aquests exemples:

Dades:	1985 (any de naixement); X (nom de l'alumne)
Interpretació:	any de naixement d'un alumne determinat
Informació:	l'alumne X va néixer l'any 1985
Dades:	65 (pes en quilograms); Y (nom del pacient)
Interpretació:	pes en quilograms d'un pacient determinat
Informació:	el pacient Y pesa 65 quilograms

Objecte

És la part del món real que ens interessa i que percebem amb els nostres sentits. Un objecte concret pot ser físic o abstracte.

Exemples d'objecte

Una persona, un objecte físic, una empresa, un producte, un compost químic, un concurs, una ètnia, una ideologia, un partit polític.

Entitat

És la conceptualització d'un objecte del món real.

***Tipus d'entitat* (tipus d'objecte)**

Aquest concepte pel qual una entitat és instància (ocurrència) el podeu consultar en el punt "Ocurrència i tipus" del subapartat "Model de dades" dins l'apartat "Model, esquema i estat de la base de dades".

Registre

És un conjunt de dades corresponents a un únic objecte del món real.

Atribut

És la propietat d'una entitat.

Camp

És la representació del valor d'un atribut.

Clau

És l'atribut o el conjunt d'atributs que permeten identificar els objectes (distingir-los dels altres).

Fitxer

Un fitxer de dades és un conjunt de dades relatives a un tipus d'objecte i estructurades en registres, configurat com una unitat d'emmagatzematge per a la cerca d'una o més dades individuals.

Dit d'una altra manera, és un **conjunt de registres** relatius a un mateix tipus d'entitat.

Distinció entre els termes *fitxer* i *arxiu*

Tot i que aquí els termes *arxiu* i *fitxer* es fan servir indistintament i, habitualment, s'usen com a sinònims, a vegades és convenient considerar-ne algunes diferencies:

1) Es pot emprar **arxiu** en un sentit general, com a *fitxer de fitxers*, o en un sentit més limitat o específic, com a *fitxer arxivat* (fitxer que ha deixat de tenir operativitat directa però que es conserva per diverses raons –legals, històriques, etc.– i que, normalment, implica el seu traspàs a un suport més econòmic al qual s'accedeix molt poques vegades).

2) Es considera més apropiat el terme **fitxer** per a fer referència al *conjunt de dades* relacionades entre si ubicades en un suport material accessible, que pot ser objecte de diverses operacions (desar, recuperar, eliminar…).

Entre altres, és comú distingir els següents tipus de fitxer:

- **Fitxer de dades** o de document: consisteix en dades en forma de text, números o gràfics. Normalment, és creat i usat per l'usuari.
- **Fitxer de programa** o d'aplicació: conté la part executable d'un programa. És utilitzat per l'ordinador i creat pel programador. Un programa pot estar format per més d'un fitxer, cadascun dels quals conté instruccions per a executar parts determinades del funcionament global del programa.

Base de dades

De moment, anomenem *base de dades* el **conjunt estructurat de fitxers de dades** que representen objectes del món real entre els quals s'estableixen relacions.

Segons la definició establerta per C. J. Date:

> "Una base de dades és un sistema computacional amb el propòsit general d'albergar informació i fer que sigui accessible cada vegada que sigui necessària entenent per dada qualsevol cosa, real o abstracta, que tingui interès per a un usuari o una organització".

Sistema de gestió de bases de dades (SGBD)

És el conjunt del programari que permet la gestió automàtica d'una base de dades; generalment, la creació, l'emmagatzematge, la manipulació i el control de les dades.

Sistema de base de dades

És un sistema d'informació basat en una base de dades.

Sistema d'informació

És un conjunt d'elements relacionats entre si ordenadament d'acord amb unes regles determinades, que aporten a l'organització a la qual serveixen la informació necessària per al compliment dels seus fins.

Normalment, es fa referència a un *sistema d'informació informatitzat* (suportat per un sistema informàtic) simplement com a sistema d'informació, malgrat que també hi pot haver *sistemes d'informació manuals*.

Elements i funcions d'un sistema d'informació

Les funcions bàsiques d'un sistema d'informació són la recollida, el processament i l'emmagatzematge de dades, així com l'elaboració i la presentació d'aquestes dades. Per al compliment d'aquestes funcions, un sistema d'informació es compon dels elements següents:

- **Contingut**: les dades i la seva descripció.
- **Equip lògic**: el sistema de comunicacions, el sistema de gestió de bases de dades, altres programes que les manipulen, el sistema operatiu, etc.
- **Equip físic**: l'ordinador o els ordinadors que suporten tot el sistema.
- **Equip humà**: l'administrador, l'usuari final, etc.

El nucli de tot sistema d'informació actual és una base de dades.

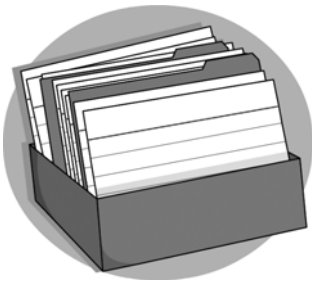
2. Evolució de la gestió de dades

En aquest apartat es fa una repassada de com han evolucionat els sistemes de gestió de dades, des dels antics sistemes manuals de fitxes de cartolina que es desaven en arxivadors, passant per la gestió informatitzada amb fitxers de dades, fins als actuals sistemes de base de dades.

A més, es constata que els sistemes de fitxers posen l'èmfasi en el tractament de les dades disperses en fitxers creats per a una aplicació concreta, mentre que les bases de dades són conjunts estructurats de dades no creats per a una aplicació específica.

2.1. Sistema manual de fitxes de cartolina

Antigament, les dades extretes de fonts bibliogràfiques (llibres, revistes, diaris) o no (conceptes, idees...) es registraven i consultaven per mitjà de **fitxes o targetes** rectangulars, habitualment **de cartolina** o paper consistent. Les fitxes s'ordenaven alfabèticament i es reunien en **arxivadors, catàlegs o fitxers manuals**..



Conjunt de fitxes o targetes

Com a criteris de cerca i recuperació dels objectes registrats s'utilitzaven **molt pocs descriptors** (normalment, tres). Per exemple, cada llibre es registrava en tres fitxes diferents: la fitxa d'autor, la fitxa de matèria i la fitxa de títol. El contingut de les tres fitxes era el mateix, només canviava la posició dels descriptors: en la primera, el nom d'autor apareixia en primer lloc; en la segona, la matèria apareixia primer, i en la tercera, el títol era el que apareixia en primer lloc. D'aquesta manera, per a cada llibre es creaven tres catàlegs o fitxers manuals.

2.2. Gestió de dades informatitzada

L'aparició dels ordinadors, els suports electrònics i, especialment, els fitxers de dades en format digital van canviar radicalment la situació. A més, gràcies a les xarxes d'ordinadors i als sistemes de comunicacions, les dades no necessiten estar allotjades físicament en el lloc on són requerides i poden ser accessibles des d'ubicacions remotes.

A més, és possible assignar a cada objecte registrat un nombre elevat de descriptors per a millorar-ne la representació, i facilitar i ampliar les possibilitats de cerca i recuperació.

Els **avantatges** d'un sistema de gestió de dades *informatitzat* monousuari respecte al sistema manual clàssic de fitxes de cartolina o paper són els següents:

- **Més consistència:** agrupació de la informació i gran reducció de l'espai físic, ja que s'evita la utilització d'arxivadors voluminosos de paper.
- **Més actualització**: disponibilitat immediata d'informació actual i precisa.
- **Més velocitat:** obtenció i modificació de les dades per part de l'ordinador amb molta més rapidesa que l'ésser humà.
- **Menys laboriositat:** eliminació del manteniment manual feixuc d'arxivadors i fitxes, gràcies a la millor i més eficient realització de les tasques mecàniques per l'ordinador.
- **Menys cost:** reducció dels costos associats al menor consum d'espai físic gràcies a la utilització de suports electrònics.

2.3. Sistemes de fitxers: orientats al procés

Els primers sistemes informatitzats de gestió de dades van ser els fitxers.

Els sistemes de gestió de dades basats en fitxers posen l'èmfasi en els tractaments que reben les dades, que s'emmagatzemen disperses en diversos fitxers, generalment plans, dissenyats per a una aplicació determinada.

A mesura que creixen les necessitats d'informació, s'implementen aplicacions noves independents entre si i es creen fitxers de dades nous que no se solen transferir entre programes d'aplicació, sinó que es dupliquen, si cal, com a part d'aquests.

Els principals **inconvenients** que presenten aquests sistemes són els següents:

- **Ocupació inútil de memòria** secundària i augment dels temps de processament perquè es repeteixen els mateixos controls i operacions en els diferents fitxers (redundància).
- **Inconsistència** a causa del fet que l'actualització de les dades no es pot fer de manera simultània en tots els fitxers on es troben.
- **Aïllament** a causa del fet que les dades i/o els programes s'emmagatzemen en diversos formats, o que programadors diferents poden haver usat llenguatges de programació diferents.
- **Dependència de les dades respecte al suport físic i als programes**, cosa que provoca falta de flexibilitat i d'adaptabilitat enfront dels canvis i repercuteix negativament en el rendiment del sistema.

Integració de dades de fitxers de departaments diferents

Si es volen integrar les dades de diferents departaments d'una organització i mantenir-les actualitzades, cal crear un programa nou. Això exigeix un gran esforç per a conèixer com s'han emmagatzemat físicament en cadascun dels programes existents i quin ha de ser el nou sistema d'emmagatzematge. Aquest sistema nou se solaparà amb l'existent, la qual cosa dificultarà la introducció de canvis, ja que **les dades depenen físicament del fitxer i del llenguatge** amb què van ser creades.

- **Incapacitat per a respondre a demandes inesperades d'informació** fora del context per al qual van ser concebuts.
- **Poca fiabilitat**, falta d'adequació a la realitat, mal control de la confidencialitat o privadesa...

Els inconvenients dels fitxers de dades

Es tracten àmpliament en l'apartat "Problemes de la gestió de fitxers de dades" d'aquest mateix mòdul.

2.4. Del processament de fitxers a la tecnologia de bases de dades

Els programes dels **sistemes tradicionals de processament de fitxers** tractaven únicament les dades dels seus fitxers. Com a molt, permetien que altres programes llegissin aquesta informació de la qual eren "propietaris", però sense poder modificar-la. D'altra banda, els ordinadors que els controlaven tenien poca potència, de manera que les dades es tractaven en un únic ordinador (sistemes monousuari).

En ampliar-se el nombre de programes d'aplicació que podien accedir a les dades i amb l'aparició de sistemes més potents que permetien el treball multiusuari, es feia necessari:

- **Independitzar les dades** de les seves aplicacions respectives.
- Crear sistemes que mantinguessin la **coherència** de les dades per a resoldre els possibles conflictes originats per l'actuació simultània de diversos usuaris.

En el procés per a solucionar aquests problemes (de dependència i de concurrència) a mitjan anys seixanta va néixer la **tecnologia de bases de dades**.

Les característiques de les bases de dades

S'enumeren en l'apartat "Objectius i característiques de les bases de dades" d'aquest mòdul.

2.5. Sistemes de base de dades: orientats a les dades

Les dades s'organitzen i es mantenen en un conjunt estructurat que no està dissenyat per a una aplicació concreta, sinó que pretén satisfer les necessitats d'informació, canviants i molt diverses, dels usuaris i de tota l'organització.

El seu **avantatge** principal és la **independència de les dades** respecte al suport físic i respecte als programes.

Els avantatges de les bases de dades

Es detallen en l'apartat "Avantatges i inconvenients de les bases de dades" d'aquest mòdul.

3. Estructures de dades

La informació s'organitza o s'estructura de manera adequada per a obtenir un rendiment raonable en el seu emmagatzematge, el seu tractament i la seva recuperació. L'estructura de dades és l'esquema organitzatiu que s'aplica a les dades per tal de facilitar-ne la interpretació o fer-hi operacions. Per especificar com s'organitzen les dades de la informació s'utilitza el tipus de dada.

El **tipus de dada** és la definició d'un conjunt de dades que especifica l'interval de valors possibles, les operacions i funcions que es poden executar sobre els valors i la manera d'emmagatzemar-los en memòria. Hi ha dues categories de tipus de dada:

1) *Tipus de dada elemental* o, simplement, **tipus de dada**

Es pot utilitzar per a construir tipus de dades més elaborats:

2) *Tipus de dada estructurat* o **estructura de dades**

Es compon d'una sèrie de dades de tipus elemental amb alguna relació existent entre elles. Normalment, sol ser una relació d'ordre, però també n'hi ha d'altres tipus.

Exemple de tipus de dada estructurat

Una dada de tipus número complex (que representa el conjunt de números complexos) és un parell ordenat de dades elementals acompanyat d'un operador. Té la forma a+b*i*, on "a" i "b" són números reals i "***i***" és el valor constant *arrel quadrada de –1* (anomenat unitat imaginària).

Espai de memòria segons el tipus de dada

Sempre que s'utilitza una dada en un programa, el seu tipus de dada ha d'estar determinat per tal que el programa traductor sàpiga com l'ha de tractar i emmagatzemar de manera apropiada.

1) En les *dades de tipus elemental*, el tipus de dada determina l'espai que s'usa en memòria.

2) En les *dades de tipus estructurat*, es presenten dos casos:

- **Estructura de dades dinàmica.** El nombre de components que té pot anar creixent. L'espai de memòria que necessita s'assigna dinàmicament. Són exemples d'aquest cas les llistes i els arbres, que veurem més endavant.
- **Estructura de dades estàtica**. La dada sempre ocupa el mateix espai de memòria. Un exemple d'aquest cas són les matrius.

Les **estructures de dades** poden ser:

- **Contigües:** els seus elements s'emmagatzemen en un conjunt consecutiu d'ubicacions de memòria. Exemples: matriu, cadena de caràcters, registre, llista lineal.
- **No contigües:** els seus elements s'emmagatzemen en ubicacions de memòria no contigües i utilitzen punters per a connectar-se. Exemples: arbre, llista vinculada.

Els elements de dades s'organitzen de manera que es puguin recuperar en un ordre particular que no és necessàriament el mateix en què estan emmagatzemats. Per a cada element de dades s'emmagatzemen dues parts: la dada del mateix element i un **punter** amb l'adreça de l'element següent. Això permet que es puguin inserir o eliminar elements sense necessitat de moure els altres elements per a deixar espai.

El **punter** (apuntador o *locator*) conté una variable auxiliar que especifica l'**adreça de memòria** (és a dir, la posició o ubicació en memòria en què s'emmagatzema físicament un element de dades determinat).

Els **tipus de dades estructurats** més utilitzats són els següents:

1) Matrius

Una matriu (també anomenada *formació* o *array*) consisteix en una quantitat fixa d'elements de dades del mateix tipus, cadascun dels quals té associat almenys un índex que determina de manera unívoca la posició de l'element en la matriu. Cada combinació de valors d'índex (a vegades anomenats *subíndex*s) identifica un i només un element de la matriu.

Una matriu es pot imaginar com una estructura de cel·les en les quals s'emmagatzemen valors. És una estructura de dades estàtica, ja que en definir-la s'especifica el nombre d'elements de què es compon.

És l'estructura de dades més usual. El nombre d'índexs de la matriu s'anomena *nombre de dimensions*. Una matriu d'una dimensió s'anomena *vector* o *matriu lineal*. Les matrius de dues dimensions són útils per a emmagatzemar taules.

2) Cadenes de caràcters

Una cadena de caràcters (també anomenada *string*) està formada per una seqüència de caràcters. Permet emmagatzemar una paraula, una frase, una temperatura.

Exemple de cadena de caràcters

El nom d'una persona: "Miquel Martí".

3) Registres

Un registre (també anomenat *file*) es compon d'elements, emmagatzemats junts, que contenen informació relativa a una mateixa entitat. Aquests elements (anomenats *camps*) poden ser de tipus diferents, apareixen en un ordre determinat i s'identifiquen per mitjà d'un nom. Per definir un registre cal especificar el nom i el tipus de cada camp.

Exemple de registre i matriu

El registre d'un alumne individual pot constar del nom de l'alumne (caràcters), el nombre d'absències (nombre sencer) i la mitjana de les seves qualificacions (nombre amb coma flotant).

D'altra banda, el conjunt de registres dels alumnes d'una aula forma una matriu de dues dimensions de registres individuals (una taula).

4) Llistes

Una llista està formada per un nombre variable de dades (o elements) d'un mateix tipus, ordenades segons una seqüència lineal (primer, segon, tercer…).

Formalment, es pot definir com una estructura de dades formada per registres d'almenys dos camps, en què un d'ells conté informació que permet localitzar el registre següent en la llista segons una seqüència determinada.

Exemples de llista

El conjunt de lletres de l'abecedari, les poblacions del trajecte d'un mitjà de transport interurbà, la llista d'espera per a una operació de menisc.

A diferència de la matriu lineal (o vector), és una estructura dinàmica (ocupa en memòria l'espai necessari per a albergar els seus elements) i no és direccionable sinó seqüencial (només es pot processar un element accedint prèviament als que el precedeixen).

Hi ha dos tipus principals de llista:

a) Llista lineal: és una llista simple i ordenada d'elements en la qual cada element, excepte el primer, va a continuació d'un altre element, i cada element, excepte l'últim, precedeix un altre element.

b) Llista vinculada (o enllaçada): és com una matriu, excepte pel fet que les posicions de la memòria física en què s'emmagatzemen els elements no són necessàriament consecutives. Els seus elements tenen enllaços (per mitjà de punters) als elements que els segueixen lògicament (la posició de l'element següent s'emmagatzema juntament amb cada element).

5) Arbres

Un arbre és una estructura de dades similar a l'estructura de llista vinculada, però més complexa, ja que cada element incorpora (emmagatzema) les adreces de dos o més elements, en lloc de l'adreça de només un element.

Es pot definir, recursivament, com un conjunt de nodes format per un node principal anomenat *arrel* i una sèrie finita de subconjunts de nodes (disjunts) que també són arbres. El *node* és una localització en l'arbre que pot tenir enllaços a un o més nodes per sota d'ell. Cada element (node) té un únic *pare*. Per representar un arbre, normalment s'utilitzen tres punters per element: un per a dirigir-se al pare, un altre per al primer fill, i el tercer per al germà següent.

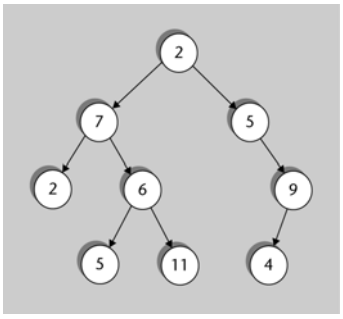


Figura 2. Exemple d'estructura en arbre

L'arbre és una estructura dinàmica i una manera eficaç d'emmagatzemar elements que han de ser buscats i recuperats ràpidament.

4. Problemes de la gestió de fitxers de dades

L'anàlisi dels problemes que comporta el *processament de fitxers tradicionals* permet justificar la utilització de *bases de dades* com a alternativa per a eliminar (o, al menys, reduir) aquests inconvenients.

En un *sistema de gestió de dades basat en fitxers* la informació està dispersa en diversos fitxers de dades i diferents programes que les agrupen i les recuperen. A mesura que creixen les necessitats d'informació, es creen programes d'aplicació i fitxers de dades nous (com a part d'aquests programes d'aplicació).

Els inconvenients que presenta la gestió de fitxers de dades es poden agrupar en dos blocs: els que afecten els fitxers i els que afecten les dades tal com es descriu a continuació:

4.1. Problemes respecte als fitxers

1) Necessitat de controlar la **integritat semàntica**

La *integritat semàntica* és un conjunt de restriccions (també anomenades *regles de validació*, *normes d'introducció* o *limitants de consistència*) que controla l'emmagatzematge de determinats valors a l'hora d'introduir-los en el sistema.

Els diferents programes que permeten la introducció de dades poden tenir exigències diferents respecte a la integritat semàntica. Si cada fitxer disposa de les seves pròpies regles de validació per a l'addició o modificació de dades és difícil que els diferents programes tinguin en compte alhora totes les regles de validació, per la qual cosa es produeix **inconsistència** de la informació a nivell de tot el sistema, encara que les dades introduïdes compleixin les regles de validació creades per als subsistemes o fitxers individuals.

Exemple de regla de validació

El valor de l'atribut o camp *NIF (número d'identificació fiscal)* està format per vuit dígits numèrics i una lletra, la qual s'obté mitjançant un algorisme en què intervenen els vuit dígits. Aquest algorisme (que calcula si una lletra determinada compleix la condició de ser correcta per a un NIF) és una regla de validació.

2) Dificultat per a gestionar el **control d'autoritzacions** (seguretat)

És habitual establir mecanismes que controlin l'accés a les dades dels diferents usuaris en funció de la seva categoria o perfil, per tal d'evitar que es produeixin accessos indeguts, per la qual cosa s'atorguen un identificador i una clau (o contrasenya) a cada usuari. L'identificador permet discriminar els usuaris reals del sistema dels que intenten accedir-hi de manera no autoritzada; la clau permet autenticar l'usuari.

Per implementar aquest control d'autoritzacions no hi sol haver cap problema. Però, pel fet que els elements (fitxers i programes) estan distribuïts en el sistema

sense organitzar, no és possible establir un mètode eficaç que controli l'accés a cada element. Això provoca que un usuari pugui accedir a programes o dades determinats sobre els quals no té permís d'accés, ja que, en modificar el programa perquè tingui aquests permisos o no, és difícil preveure i crear els controls de seguretat adequats. En aquest cas pot succeir que, de manera no desitjada, es permeti a usuaris no autoritzats accedir a dades sobre les quals no haurien de tenir permís d'accés.

(Des)control d'autoritzacions a tercers

Un problema afegit es produeix quan un usuari que pot atorgar permisos per a accedir a uns elements determinats, els ha atorgat a terceres persones i més tard se li retiren. Pot quedar el dubte de si s'han eliminat o no els permisos per a aquestes terceres persones.

3) Falta de **control de concurrència**

La concurrència és l'accés simultani de diversos usuaris a les mateixes dades. Quan no hi ha un programa que controli l'accés dels diferents usuaris que accedeixen alhora al mateix fitxer per a modificar-ne les dades que conté, no es pot saber quina de les dades modificades es desarà ni en quin ordre s'han produït aquestes modificacions. Això fa que la informació obtinguda sigui inconsistent respecte a les operacions dutes a terme.

4.2. Problemes respecte a les dades

Els principals problemes causats per l'estructura física de les dades (és a dir, per la manera en què aquestes estan emmagatzemades en els diferents fitxers), són els següents:

1) Redundància (repetició d'informació)

És la repetició innecessària d'informació en diversos fitxers. També es produeix quan hi ha dades que no aporten informació addicional, ja que es poden obtenir o calcular a partir d'altres.

La redundància es deu a l'absència d'integració de dades en fitxers procedents de departaments diferents o a l'ús de llenguatges diferents.

Integritat de les dades

És l'exactitud i consistència interna de la informació emmagatzemada en un sistema de gestió de dades.

Problemes de la redundància

Associats als problemes de redundància estan els derivats de la **inconsistència**, la **falta d'integritat** de les dades, el malbaratament de l'espai d'emmagatzematge (**cost de l'espai** ocupat inútilment), la dificultat per a **mantenir actualitzades** les dades comunes i la **recuperació lenta** de les dades.

2) Inconsistència (difícil actualització)

Com a conseqüència de l'existència d'informació redundant, l'actualització de les dades duplicades en diversos fitxers es fa molt difícil, ja que quan es modifica una dada en algun fitxer pot succeir que no es modifiqui en els altres en què també es troba duplicada; això fa que la informació dels diferents fitxers no concordi. El resultat és que la base de dades es torna inconsistent i proporciona informació incorrecta o contradictòria.

Exemple d'inconsistència

Si no s'actualitza alhora la informació d'un alumne repetida en diferents fitxers, aquesta pot ser diferent segons el departament en què es localitza, la qual cosa provocaria, per exemple, que es mantinguessin dos comptes corrents diferents, en comptes d'un únic compte actualitzat.

3) Aïllament (dades repartides)

És la fragmentació de la informació que es produeix quan les dades referents a un objecte s'emmagatzemen en fitxers diferents i és difícil obtenir alhora tota la informació relativa a un mateix objecte.

Els usuaris de departaments o projectes diferents interessats en dades de la mateixa entitat (o objecte) del món real mantenen per separat fitxers independents i programes diferents per a manipular les dades, perquè cadascun requereix algunes dades que no es troben en els fitxers dels altres. Això, a més, complica l'escriptura de programes.

Gestió de fitxers separats amb programes diferents

Per exemple, un usuari del departament de gestió acadèmica pot mantenir un fitxer d'alumnes amb les seves qualificacions (per introduir-les en el fitxer i imprimir-les es crea el programa pertinent). Un usuari del departament de comptabilitat porta el control de taxes dels alumnes i el seu pagament en un altre fitxer. Malgrat que tots dos gestionen dades dels alumnes, mantenen fitxers separats (i programes diferents), perquè cada un necessita altres dades que no pot obtenir del fitxer de l'altre.

4) Dificultat d'accés a les dades

És un problema derivat de l'aïllament de la informació que consisteix en la mancança d'un catàleg actualitzat que inclogui la relació de fitxers de dades i els seus continguts i que permeti obtenir les dades desitjades en un temps adequat. La carència d'aquest catàleg comporta un problema organitzatiu (és necessari que l'administrador del sistema obtingui les dades sol·licitades), i un altre de temporal (la demora pot fer que la informació sol·licitada es torni obsoleta abans de ser rebuda).

D'altra banda, les funcions de recuperació i actualització de dades de què disposen els diferents programes existents són limitades (només contemplen una sèrie d'opcions específiques) i poden obligar a reprogramar, amb la consegüent pèrdua de temps.

5) Dependència dels programes (i diferència de formats)

Les dades s'emmagatzemen en diversos fitxers (en formats diferents o no) que estan gestionats per aplicacions diferents.

La definició de dades forma part dels mateixos programes d'aplicació. Per tant, aquests programes només poden treballar amb un fitxer concret, l'estructura del qual està codificada en el mateix programa (utilitzant terminologia de programació, es diu que l'estructura està *declarada* mitjançant instruccions no executables).

Qualsevol canvi en l'estructura dels fitxers (és a dir, l'organització segons el tipus d'accés per al qual es van definir: seqüencial, relatiu, direccional, aleatori…) pot afectar els programes d'aplicació que els gestionen.

5. Objectius i característiques de les bases de dades

L'objectiu principal d'una base de dades per a superar els inconvenients de la gestió de dades mitjançant fitxers és **centralitzar la informació**. Això permet que els programes **integrin les dades** (que abans es trobaven disperses en fitxers de diferents departaments) i **mantinguin la informació actualitzada**.

Els inconvenients de la gestió de fitxers

Es detallen en l'apartat "Problemes de la gestió de fitxers de dades" d'aquest mòdul.

Un altre objectiu és **estandarditzar un mode de gestió de dades** més eficaç, universal i independent del maquinari i del sistema operatiu sobre el qual s'implementi, de manera que es generi un sistema d'emmagatzematge amb una **interfície d'usuari fàcil** de consultar i de modificar.

No menys important és que permeten **optimitzar el cost** del suport d'emmagatzematge de les dades, la qual cosa depèn de la configuració de l'SGBD implementat (monousuari, multiusuari, centralitzat, distribuït...) i del volum de la informació emmagatzemada.

Amb independència de com s'organitza la informació en una base de dades, per a ser considerada com a tal, ha de complir una sèrie de requisits o requeriments que permeten definir-la de la manera següent:

> Una base de dades està formada per un conjunt estructurat de dades gestionat per un programari de tal manera que es controla l'emmagatzematge de dades redundants, les dades són independents dels programes que les utilitzen, les relacions entre les dades s'emmagatzemen juntament amb aquestes i és possible accedir a les dades de diverses maneres.

Les **característiques de les bases de dades** es poden enumerar resumidament tal com segueix:

- **Integració de tota la informació** de l'organització (absència de redundància i d'inconsistència).
- **Persistència de les dades** (disponibilitat immediata per a emmagatzematge en memòria secundària).
- **Accessibilitat simultània i concurrent** per a diferents usuaris (compartició de la informació).
- **Descripció unificada** de les dades (emmagatzematge conjunt de les dades i de la definició de les estructures) **i independent** de les aplicacions.

- **Independència** de les aplicacions respecte a la representació **física de les dades**.

Abstracció de dades i separació entre programes i dades

La característica que permet la independència de les aplicacions i les dades és l'**abstracció de dades**. Un SGBD ofereix a l'usuari una representació conceptual de les dades que no inclou gaires detalls sobre el seu *emmagatzematge* ni sobre com *s'implementen* les operacions.

El **model de dades** és un tipus d'abstracció que permet ocultar aquests detalls que no interessen a la majoria d'usuaris de la base de dades i que se serveix de conceptes lògics (els objectes i les seves propietats i interrelacions) que són més fàcils de comprendre per a l'usuari.

Hi ha molts models de dades que permeten oferir a l'usuari aquesta abstracció de les dades.

Els models de dades i els tipus d'abstracció

Aquests models utilitzats per a la representació de les dades es tracten en els apartats "Model, esquema i estat de la base de dades" i "Tipus d'abstracció en el disseny de bases de dades".

- **Definició de vistes** parcials de les dades per a diferents usuaris.

- **Gestió de les dades** (manipulació de les dades i manteniment de la base de dades).

- **Manteniment de la integritat** (qualitat i correcció) **i la seguretat** (accessos autoritzats) de les dades.

- **Flexibilitat** (utilització de diferents mètodes d'accés amb temps de resposta petits).

Per complir aquests objectius i requisits, els SGBD, independentment del tipus al qual pertanyin i del seu fabricant, disposen de components amb funcions ben definides i tenen una arquitectura estàndard anomenada *arquitectura de nivells*.

Arquitectura, funcions i components dels SGBD

Es presenten en els apartats "Arquitectura dels SGBD" i "Funcions i components de l'SGBD".

Les característiques definitòries de les bases de dades les diferencien de tècniques anteriors de gestió de dades (com el processament de fitxers tradicionals) i aporten una sèrie d'avantatges per bé que també presenten algun inconvenient. Tots ells es tracten a continuació.

6. Avantatges i inconvenients de les bases de dades

En aquest apartat, s'exposen els avantatges que presenten les bases de dades (respecte als fitxers tradicionals), agrupats segons si afecten les dades, els usuaris o l'organització. Ara bé, la seva utilització pot comportar una sèrie d'inconvenients pel que fa a la seva inversió inicial, implantació, integració, normalització de sistemes o rendibilitat així com també per als seus usuaris. En casos molt específics és millor utilitzar un sistema de fitxers en comptes d'una base de dades, tal com es comentarà.

6.1. Avantatges

Els avantatges que les bases de dades presenten enfront dels sistemes de gestió de dades basats en fitxers són conseqüència de les seves característiques.

La integració de tots els elements del sistema de base de dades (dades, maquinari, programari i personal humà) i el seu control centralitzat facilita la disponibilitat de les dades als usuaris sense dependre dels propietaris de porcions físiques de l'aplicació i comporta l'augment de la seguretat del sistema pels nivells de seguretat diferents que s'implementen en la base de dades.

Els avantatges es poden referir a les dades, als usuaris i a l'organització.

6.1.1. Avantatges respecte a les dades

1) Disminució de la redundància

Reduir la redundància és un dels objectius principals de les bases de dades, sempre que no comporti l'augment de la seva complexitat ni la disminució del temps de resposta (o eficiència).

Si es fa un bon disseny, la redundància es veu notablement disminuïda, encara que no s'evita totalment. Sovint, però, interessa un cert nivell de redundància controlada.

Redundància controlada

Els motius per repetir determinades dades són bàsicament dos:

- La **redundància lògica** s'utilitza per a assegurar la integritat de les dades. Per a representar les interdependències existents entre els objectes del món real (que formen part del domini del problema) cal repetir dades que permetin establir associacions entre elles.

- La **redundància física** s'utilitza per motius d'eficiència i rendiment. La fragmentació i replicació d'informació utilitzada en bases de dades distribuïdes comporta una redundància desitjada que permet millorar la disponibilitat de les dades, el temps de resposta i el cost de les comunicacions.
- En qualsevol cas, la gestió de la informació redundant es fa de manera interna en el mateix sistema de base de dades.

2) Prevenció de la inconsistència

La disminució de la redundància evita la inconsistència i permet més eficiència en la validació i la introducció de dades en el sistema..

Propagació d'actualitzacions

Si hi ha un cert grau de redundància, cal constatar que les actualitzacions s'enregistrin en tots els llocs on apareixen les dades. Aquest procés és possible gràcies a la propagació d'actualitzacions que el sistema pot fer automàticament.

3) Representació d'associacions entre les dades

Es poden representar diversos tipus de vincles (més o menys complexos) existents entre les dades, la qual cosa permet obtenir i actualitzar amb rapidesa i eficiència dades que estan mútuament interrelacionades..

Simplicitat del sistema

La naturalesa del problema a tractar informàticament és un factor de complexitat de partida que la base de dades ajuda a minimitzar mitjançant representacions lògiques senzilles que permeten la modificació dels seus requisits, de manera que la inclusió i/o la modificació de nous elements de dades i interrelacions no comporta una complexitat excessiva.

4) Manteniment de la integritat

L'eliminació de la redundància minimitza la falta d'integritat de les dades. Encara que es comparteixin dades entre diferents usuaris i ubicacions o es modifiquin els objectes que formen la base de dades, el sistema permet definir restriccions d'integritat i garantir el seu acompliment.

Exemples de restriccions d'integritat

Algunes restriccions d'integritat que deriven de la semàntica o el significat de les dades són, per exemple, especificar el tipus de dada per als atributs (camps), obligar que els registres d'un fitxer estiguin relacionats amb registres de fitxers determinats, imposar que els valors dels registres siguin únics.

5) Subministrament de còpies de seguretat i recuperació

Si el maquinari o el programari fallen durant l'execució d'una actualització, el *subsistema de còpies de seguretat i recuperació* de l'SGBD assegura la restauració de la base de dades a l'estat anterior a l'execució o permet que es reprengui l'execució del procés en el punt on fou interromput.

6) Emmagatzematge de les especificacions de la base de dades (naturalesa autodescriptiva del sistema)

El sistema no conté únicament la mateixa base de dades, sinó també una *definició* o *descripció* completa de *l'estructura i els elements* de la base de dades mitjançant metadades (dades que descriuen les dades) que inclouen, per exemple: tipus i format d'emmagatzematge de cada element de dades, restriccions sobre les dades. Aquesta definició s'emmagatzema en el *diccionari de dades* (o *catàleg*) de l'SGBD separadament dels programes d'accés.

Donat que un SGBD de propòsit general no es crea per a una única base de dades, el programari de l'SGBD (i també l'usuari) pot accedir a qualsevol base de dades mitjançant l'extracció de les seves definicions del diccionari de dades, mentre que el programari per al *processament de fitxers tradicionals* només pot accedir al fitxer de dades específic per al qual s'ha creat (i la definició de dades està declarada en el mateix programa).

7) Independència entre els programes i les dades

La independència

Cal destacar que aquest aspecte (no tan evident, tot i estar implícit en altres avantatges) és un objectiu important de les bases de dades.

Els programes d'aplicació desenvolupats per a manipular les dades són independents de la implementació escollida per a les estructures de la base de dades. Hi ha dos tipus d'independència:

- **Independència lògica**, per la qual la modificació de la representació lògica general del domini del problema (els objectes de la base de dades i les seves associacions) no afecta els programes d'aplicació que manipulen les dades.
- **Independència física**, per la qual la modificació de l'estructura física i/o la distribució de les dades en les unitats d'emmagatzematge no afecta els programes que manegen les dades ni l'estructura lògica general de la informació.

Flexibilitat en la modificació de l'organització de les dades

1) Per canvi del suport físic

En canviar els objectes d'un esquema de base de dades d'un equip a un altre no cal fer canvis en els procediments d'accés a les dades pel canvi d'equip o de suport físic. Un exemple és el d'usuaris que utilitzen ordinadors portàtils o amb mobilitat dins d'una xarxa.

2) Per afinació de l'organització física

La modificació de l'organització física de les dades per la seva evolució (per exemple, per a millorar el temps de resposta) no afecta les representacions de les dades ni els procediments que operen sobre elles.

En tots dos casos, una modificació del disseny (a nivell físic o conceptual) no implica modificacions en les aplicacions, i viceversa.

En canvi, en el *processament de fitxers tradicionals*, l'estructura dels fitxers de dades està integrada en els programes d'accés. Per tant, la modificació de l'estructura d'un fitxer pot requerir la modificació dels programes que hi accedeixen.

8) Compatibilitat entre formats (importació i exportació)

Permet reconèixer informació organitzada físicament per un altre programari de manera diferent de la que utilitza la base de dades. Això es refereix tant a la informació existent abans de la implantació de la base de dades com a possibles canvis posteriors.

6.1.2. Avantatges respecte als usuaris

La tecnologia de base de dades s'ha desenvolupat per a donar resposta a les exigències creixents de funcionalitat i eficiència que planteja l'usuari. Els avantatges descrits a continuació, si no s'indica el contrari, es consideren referits a sistemes multiusuari.

1) Possibilitat d'aplicació eficient de restriccions de seguretat

El *subsistema de seguretat i autorització* de l'SGBD permet garantir automàticament l'acompliment de restriccions d'accés establertes per l'administrador de la base de dades, segons el perfil de cada usuari. Sense elles, la informació d'una base de dades (centralitzada) està en més perill que la d'un sistema de fitxers tradicionals (dispersos).

Diverses restriccions d'accés

Els controls de seguretat permeten restringir:

- L'**accés a les dades** (limiten els usuaris que poden accedir a informació delicada o confidencial).
- Les **operacions d'accés** (permeten a alguns usuaris només la recuperació de dades determinades i a uns altres, la seva actualització).
- **L'accés al programari** de l'sgbd (permeten a l'administrador utilitzar determinat *programari privilegiat* –com el que serveix per a crear comptes i perfils d'usuari– i a l'operador de consola només fer determinades *transaccions programades*).

2) Subministrament de múltiples interfícies d'usuari

L'SGBD ofereix diferents interfícies per a usuaris de diversos nivells de coneixement tècnic i amb diferents necessitats d'utilització.

Exemples d'interfícies d'usuari

Hi ha interfícies de llenguatge de programació per a programadors d'aplicacions, llenguatges de consulta per a usuaris ocasionals, interfícies de llenguatge natural per a usuaris autònoms... Com a ***interfícies gràfiques d'usuari*** (GUI) cal esmentar les interfícies controlades per menús per a usuaris autònoms, els formularis i les interfícies de transaccions programades per a usuaris paramètrics i les interfícies web per a l'accés a través d'Internet.

3) Suport de múltiples vistes de dades

Les interfícies d'usuari

Les interfícies d'usuari que ofereix un SGBD es poden consultar en el subapartat "Interfícies de l'SGBD", dins l'apartat "Funcions i components de l'SGBD".

Una *vista* és una representació externa de la base de dades que només considera un subconjunt de les dades emmagatzemades. Malgrat la informació que forma part del domini d'un problema o sistema és única, un SGBD multiusuari proporciona mecanismes per a definir múltiples vistes globals i parcials per als diversos programes d'aplicació i grups d'usuaris sense perdre el caràcter integrador de la base de dades.

Una vista pot contenir dades virtuals que no estan explícitament emmagatzemades, sinó que són derivades de dades emmagatzemades (com, per exemple, el valor calculat d'una mitjana o l'edat d'un alumne calculada a partir de la data de naixement). De fet, la majoria d'usuaris no necessita saber si utilitza dades emmagatzemades o derivades.

Diferents vistes de les dades

Hi ha usuaris que necessiten vistes particulars que els permetin actualitzar determinades dades rellevants per a ells.

Tindran diferents vistes l'usuari que només gestiona els expedients dels alumnes d'un centre i l'usuari que comprova que els alumnes han cursat tots el prerequisits de cada curs en què estan matriculats.

Com a avantatge addicional, la utilització de les vistes constitueix en si mateixa un mecanisme de seguretat important, ja que proporciona un control discrecional d'autoritzacions.

4) Disponibilitat d'informació actualitzada

El *subsistema de control de concurrència i de recuperació* de l'SGBD fa possible que tots els usuaris puguin disposar, de manera immediata i transparent, de les dades actualitzades per altres usuaris.

5) Rapidesa i senzillesa d'accés a la informació

La capacitat de processament del sistema permet, en cada moment, una recuperació ràpida i senzilla de la informació requerida per l'usuari. Això es concreta en dos aspectes:

- **Eficiència**

  La base de dades assegura un *baix temps de resposta* adequat a l'usuari i permet l'accés simultani als conjunts d'elements de dades pel mateix procediment, la mateixa aplicació o el mateix usuari o per uns de diferents.

- **Capacitat d'accés**

  La base de dades respon, en un temps acceptable, a qualsevol consulta sobre la seva informació, *sense restriccions* importants pel que fa als elements de dades, les seves relacions, l'agrupació (o el format sol·licitats).

6) Accessibilitat múltiple i simultània

Diversos usuaris poden compartir la base de dades i accedir-hi simultàniament. Això es tradueix en els avantatges següents:

- **Compartició de dades**

  Les diferents parts de la base de dades es poden compartir entre diversos grups d'usuaris sense que es produeixin conflictes i mantenint la integritat de les dades.

  La Independència facilita la compartició

  La independència entre dades i aplicacions facilita la compartició i la disponibilitat de les dades tant per als usuaris com per a les aplicacions, sense necessitat de modificar la base de dades.

- **Processament de transaccions multiusuari**

  L'SGBD inclou un *gestor de control de concurrència* que assegura que quan diversos usuaris intenten actualitzar les mateixes dades, ho facin de manera controlada per tal que el resultat sigui correcte. Perquè les *transaccions concurrents* operin adequadament, els accessos successius se situen en cua de manera que, fins que un usuari no acabi de fer modificacions, la resta no en pot fer de noves.

  Processament de transaccions en línia (*online transaction processing, OLTP*)

  Un exemple es presenta quan diversos encarregats de reserves proven d'assignar una localitat d'un espectacle (o mitjà de transport) a un client. L'SGBD garanteix que, en un moment determinat, només un usuari tingui accés a la reserva d'una localitat concreta per a assignar-la a l'espectador (o passatger) corresponent.

7) Flexibilitat per a atendre nous requeriments d'informació

Quan canvien els requeriments d'informació, és possible modificar l'estructura de la base de dades sense afectar les dades emmagatzemades ni els programes d'aplicació existents gràcies a la independència entre la base de dades i les aplicacions que les manipulen..

Adaptació a requeriments nous

Aquest cas es dóna quan cal l'addició d'un fitxer nou o l'ampliació dels elements de dades (camps) d'un fitxer existent per a atendre les necessitats d'informació d'un nou grup d'usuaris.

8) Coherència dels resultats

El fet que en tots els tractaments s'utilitzin les mateixes dades (i que es previngui la inconsistència) fa que els resultats siguin coherents.

6.1.3. Avantatges respecte a l'organització

A part dels aspectes esmentats anteriorment, hi ha altres implicacions de la utilització de bases de dades (moltes de les quals tenen a veure amb el cost) que poden resultar beneficioses a l'organització. De fet, les bases de dades són un instrument d'influència decisiva en les organitzacions.

1) Integració de tota la informació de l'organització

La base de dades dóna servei a tota l'organització, o a una part important d'aquesta, i no únicament a alguns departaments o usuaris individuals, de manera que s'evita la redundància de dades i els problemes d'inconsistència.

2) Eficiència de l'emmagatzematge físic

La disminució de la redundància i les tècniques de compactació comporten la reducció de l'*espai d'emmagatzematge* i dels costos associats al consum d'espai físic.

3) Reducció del cost d'emmagatzematge i de manteniment

Aquest avantatge és conseqüència de l'eficiència de l'emmagatzematge físic i de la independència lògica i física de les dades.

4) Reducció del cost de formació del personal

El fet que la majoria d'SGBD del mercat disposin d'eines que ofereixen un ús fàcil repercuteix favorablement, a llarg termini, en el cost de formació del personal.

5) Economia d'escala

La consolidació de les dades i les aplicacions redueix el malbaratament produït pel solapament d'activitats del personal de processament de dades en diferents departaments. Això comporta la reducció del cost total d'operació i gestió, ja que l'organització pot invertir en sistemes globals de processament, emmagatzematge i/o comunicació més potents, en lloc de fer-ho en equipament de capacitat inferior per a cada departament.

6) Capacitat per a establir normes

L'administrador de la base de dades pot garantir l'acompliment de normes per a la representació de les dades. Ell és responsable de definir i imposar als usuaris les normes de l'organització per a facilitar la comunicació i la cooperació entre diversos departaments o projectes.

Exemples de definició de normes

La normalització permet unificar, per exemple: els noms i els formats dels elements de dades (i així facilita l'escriptura de programes, l'intercanvi d'informació i la migració de dades entre sistemes); la terminologia (per a documentar les dades); les estructures de formularis i informes.

7) Reducció del temps de creació d'aplicacions

Per bé que es tarda més a dissenyar i implementar una base de dades des de zero que a crear una aplicació de fitxers especialitzada, quan la base de dades és operativa es requereix molt menys temps (entorn d'una cinquena part) per a la creació d'una nova aplicació (com la que permet obtenir una informació determinada en un informe) amb els recursos de l'SGBD que amb un *sistema tradicional de fitxers*.

6.2. Inconvenients

Malgrat els seus avantatges, un sistema de base de dades pot comportar una sèrie d'inconvenients que suposin barreres a la seva implantació. Abans de prendre aquesta decisió, cal avaluar si la base de dades soluciona els problemes que planteja la informació en l'organització.

Els costos addicionals que genera la implantació d'una base de dades es deuen als fets següents:

1) Inversió inicial elevada

Les primeres fases d'implantació d'una base de dades impliquen els inconvenients següents:

a) Instal·lació costosa de nous equips i eines que s'adeqüin a les utilitats del sistema. Això inclou l'actualització o ampliació del **maquinari** existent sobre el qual funcioni el nou SGBD, la conversió del **programari** (sistemes operatius, compiladors…) i l'adquisició i el manteniment del mateix SGBD.

b) Cost de capacitació de personal especialitzat (final i informàtic) generat en les primeres fases de disseny, engegada, administració i utilització.

2) Implantació llarga i difícil

Les dificultats que van apareixent al llarg de la implantació de la base de dades poden fer que se superin àmpliament els terminis previstos inicialment per a estar plenament operativa.

3) Rendibilitat a mig (o llarg) termini

A pesar que els beneficis pel que fa a velocitat de procés i obtenció de resultats són immediats, la rendibilitat no s'obté fins que no s'ha afinat el sistema. Això significa que cal considerar el sistema de base de dades globalment com a instrument per a l'obtenció de beneficis en altres àrees de l'empresa.

4) Absència de normalització dels SGBD comercials

Els SGBD comercials difícilment acompleixen totes les especificacions definides pels estàndards de bases de dades i a més tenen característiques afegides que els fan incompatibles entre ells.

Actualment, però, aquest inconvenient es pot obviar, ja que les tècniques de compartició permeten utilitzar les dades amb qualsevol SGBD amb independència de on s'emmagatzemin.

5) Problemes d'integració de bases de dades

Quan es volen obtenir bases de dades distribuïdes per integració d'altres, creades per dissenyadors diferents o no, es poden produir problemes de redundància o d'obtenció d'objectes diferents incoherents perquè no existeix un disseny únic per a les bases de dades.

6) Risc de frustració dels usuaris

Existeix un risc evident de decepció per part de l'usuari (especialment directiu) que pot fer passar per alt el potencial real d'un sistema de base de dades.

Les expectatives frustrades pel fracàs de projectes de bases de dades es deuen a factors diversos: disseny inadequat de la base de dades, implementació incorrecta de les aplicacions del sistema, assumpció de funcionalitats o prestacions que només són estudis teòrics (de fet, la divergència amb la implantació pràctica d'innovacions en SGBD comercials és considerable).

7) Altres inconvenients

Altres inconvenients dels sistemes de base de dades es deuen al propòsit generalista que presenta l'SGBD per a la definició i el processament de les dades, als costos que implica la necessitat d'oferir funcions de seguretat, control de concurrència, recuperació i integritat.

6.3. Casos en què no s'aconsella usar un sistema de base de dades

Convé utilitzar un sistema de fitxers tradicionals en comptes d'un sistema de base de dades en les situacions següents:

- El sistema i les aplicacions existents estan ben definits, són senzills i, previsiblement, no han de canviar.

- No cal l'accés multiusuari i simultani a les dades.

- El temps de resposta rigorós requerit pels programes és difícil de complir amb un SGBD.

7. Elements d'un sistema de base de dades

Un sistema de base de dades està format per quatre recursos o elements principals: el recurs principal és la mateixa base de dades i el secundari, el programari relacionat amb ella; els altres són el maquinari i les persones que intervenen en el sistema.

Es pot afegir un cinquè element: els procediments (que són les instruccions i regles que dirigeixen el disseny i la utilització dels components de programari).

7.1. El contingut: les dades

El contingut de la base de dades el conformen les dades, que constitueixen la part essencial del sistema de base de dades i en justifiquen l'existència. Les dades d'una base de dades segueixen una jerarquia establerta i han de presentar una sèrie de propietats o característiques.

Propietats de les dades d'una base de dades

Les dades emmagatzemades en una base de dades han de ser:

- **Perdurables**: han de romandre invariables en el temps, no han de ser efímeres.
- **Estructurades**: han de tenir una organització que faciliti la compartició entre diferents usuaris.
- **Operacionals i transaccionals**: han de ser manejables aplicant operadors per a obtenir els resultats desitjats.
- **Significants**: han de tenir un significat o sentit semàntic concret.
- **Íntegres**: han de reflectir, sense contradiccions, una realitat existent.

Jerarquia de les dades

Les dades d'una base de dades s'emmagatzemen seguint la jerarquia següent:

BIT → CARÀCTER → DADA → REGISTRE → FITXER → BASE DE DADES

Això es mostra en l'esquema següent:

Concepte	Descripció	Exemple	Comentari
BIT	Dígit binari.	0	Pot ser 0 o 1.
BYTE = CARÀCTER	Conjunt ordenat de 8 bits (octet) que representa un caràcter.	010001101	Representa el caràcter "m".
CAMP = DADA	Conjunt ordenat de bytes que representa un únic element (o ítem) de dades. També s'utilitza el terme **ATRIBUT**.	Mercè	Nom de l'alumne.
REGISTRE	Conjunt de camps relacionats que caracteritzen un únic objecte, persona… També s'utilitza el terme **ENTITAT**.	Dades de l'alumne	Inclou nom, cognoms, data de naixement, DNI, adreça, telèfon, etc.
FITXER	Col·lecció de registres relacionats referits a un conjunt d'objectes d'un mateix tipus.	Alumnes	Inclou totes les dades personals relatives a tots els alumnes de la universitat.
BASE DE DADES	Conjunt de fitxers relacionats que representa el domini d'un sistema.	Dades de la universitat	Inclou tots els fitxers relatius a alumnes, professors, estudis, assignatures, aules, etc.

Font: adaptat de Parker i Case (1993)

7.2. L'equip lògic: el programari

Bàsicament, fa referència al programa de gestió de dades instal·lat, que permet als usuaris manipular les dades sense necessitat de conèixer com s'han emmagatzemat físicament. Aquest programa rep el nom de *sistema gestor de bases de dades* (SGBD) i es pot considerar l'entorn de la base de dades.

En sentit ampli, també engloba el sistema operatiu, el programari de comunicacions, les utilitats, els programes d'aplicació que processen les dades, els compiladors de llenguatges de programació…

7.3. L'equip físic: el maquinari

Engloba l'ordinador sobre el qual s'instal·la l'SGBD i on s'emmagatzemen físicament les dades (i altres objectes) de la base de dades i els dispositius que en permeten la gestió. És a dir, inclou el disc dur, la CPU, les unitats d'entrada/sortida, etc.

En bases de dades distribuïdes, el maquinari també comprèn els equips connectats en xarxa a l'equip servidor de dades.

7.4. El personal humà: els usuaris

El personal humà que interacciona amb un sistema de base de dades s'ha considerat tradicionalment un component fonamental per al funcionament correcte i adequat de l'SGBD.

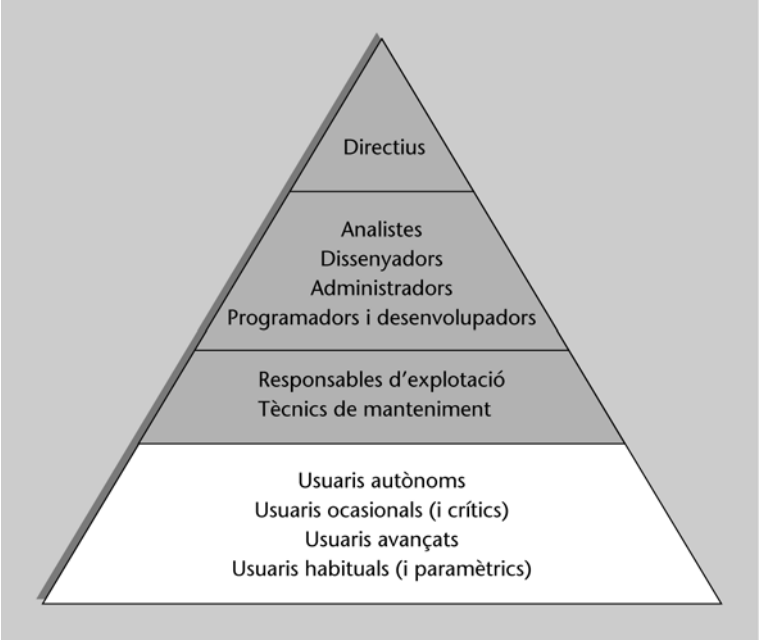
El nombre i el tipus d'usuaris depenen de la magnitud de la base de dades

En la definició, la construcció i la manipulació d'una base de dades personal petita (com una llista d'adreces i telèfons) intervé una única persona. En canvi, en el disseny, la utilització i el manteniment d'una base de dades gran o corporativa (com pot ser el conjunt de declaracions de la renda que gestiona l'agència tributaria) participen moltes persones.

8. Usuaris de les bases de dades

En organitzacions grans els usuaris de les bases de dades es poden dividir en dos tipus principals: el personal informàtic i els usuaris finals.

Figura 3. Piràmide d'usuaris de les bases de dades



8.1. Personal informàtic

El personal humà que intervé en el desenvolupament i el manteniment de bases de dades es pot classificar en les categories següents:

1) Directius

Les seves tasques principals són l'**organització i coordinació** del departament d'informàtica, la definició de les bases de dades a desenvolupar i l'organització de cursos de formació.

2) Analistes de sistemes

Els analistes de sistemes planifiquen i controlen el desenvolupament de les bases de dades aprovades per la direcció i fan els estudis de viabilitat necessaris. A partir de l'**anàlisi de necessitats** de tractament d'informació dels usuaris finals (especialment els paramètrics) desenvolupen especificacions per a *transaccions programades* que satisfacin aquests requeriments i generen la documentació analítica necessària per al treball dels programadors.

Els usuaris paramètrics

Els que usen transaccions programades es descriuen en el subapartat "Usuaris finals" d'aquest mateix apartat.

Els analistes tenen funcionalitat d'administradors de bases de dades quan aquestes ja estan creades.

3) Dissenyadors de bases de dades

S'encarreguen de tasques prèvies a la implementació de la base de dades com ara identificar les dades que s'emmagatzemaran en la base de dades i escollir les **estructures** apropiades per a presentar-les i emmagatzemar-les.

Els dissenyadors es comuniquen amb els futurs grups d'usuaris de la base de dades a fi de comprendre les seves necessitats i d'implementar un disseny que satisfaci els seus requeriments de dades i de processament, i desenvolupar les vistes corresponents.

Sovint formen part del personal d'administració de la base de dades i, una vegada acabat el seu disseny, poden assumir altres responsabilitats.

4) Programadors d'aplicacions

Mitjançant un llenguatge de programació, implementen les especificacions determinades pels analistes de sistemes en forma de **programes**, provant, depurant i mantenint les transaccions programades que els usuaris paramètrics utilitzaran per a interactuar amb la base de dades.

Per dur a terme les seves tasques, aquests usuaris, també anomenats enginyers de programari, han de conèixer perfectament totes les capacitats de l'SGBD (igual que els analistes de sistemes).

5) Administradors de bases de dades

Com a principals usuaris informàtics de la base de dades, els administradors (o gestors) de bases de dades (ABD) són responsables de supervisar i gestionar els **recursos** del sistema (la mateixa base de dades i el programari relacionat amb aquesta) i d'adquirir els recursos de programari i maquinari necessaris. S'encarreguen de dissenyar i modificar l'estructura d'emmagatzematge de la base de dades, determinar les estratègies d'accés a les dades, coordinar i vigilar la seva utilització, comprovar el rendiment del sistema i resoldre els problemes de falta d'eficiència a fi de garantir la **qualitat** i la **disponibilitat** de les dades.

Per tant, han de gestionar la **seguretat** de la base de dades respecte als accessos i la **integritat** de les dades. Defineixen els esquemes externs de cada usuari, creen proteccions d'accés per a evitar violacions de la seguretat i asseguren una gestió correcta de les transaccions que eviti la pèrdua o corrupció de les dades. També són responsables de definir l'estratègia de **recuperació**, amb la creació de les còpies de seguretat necessàries.

Responsabilitats de l'administrador de bases de dades

Les tasques més importants de l'ABD tenen a veure amb els aspectes següents:

- **Recuperabilitat**: crear i provar còpies de seguretat de la base de dades.
- **Integritat**: verificar o ajudar a verificar la integritat de les dades.
- **Seguretat**: implementar i gestionar controls d'accés a les dades.
- **Disponibilitat**: assegurar l'accés a les dades en tot moment (24 hores ×365 dies).
- **Rendiment**: assegurar el mínim temps de resposta.
- **Suport a proves**: proporcionar dades de funcionament i rendiment a programadors i dissenyadors per a millorar l'eficiència de la base de dades.

6) Responsables d'explotació

S'ocupen de **posar en marxa** les bases de dades i els sistemes que hi treballen diàriament.

Hi ha altres usuaris informàtics que, malgrat són crucials perquè l'usuari final pugui utilitzar el sistema, no solen tenir un interès clar en el contingut de la base de dades. Es tracta de personal relacionat amb el disseny, la creació i el funcionament del programari de l'SGBD i el seu entorn. Són els següents:

7) Desenvolupadors de l'SGBD

Dissenyen i implementen els **mòduls** i les **interfícies** de l'SGBD en forma de paquets de programari.

8) Desenvolupadors d'eines

Dissenyen i implementen **eines i utilitats** de l'SGBD, que faciliten el disseny i la utilització del sistema i ajuden a millorar-ne el rendiment.

9) Tècnics de manteniment

Són professionals d'administració del sistema i operadors responsables de donar solució als problemes de maquinari i de programari produïts durant el **funcionament** del sistema de base de dades.

8.2. Usuaris finals

La base de dades existeix principalment perquè l'usuari final aconsegueixi els objectius personals o empresarials proposats. La seva feina requereix accedir a la base de dades per a fer consultes i actualitzacions, generar informes, i interactuar-hi directament per mitjà d'una interfície gràfica. Els usuaris finals es poden classificar en les categories següents:

1) Usuaris habituals

Tot i no ser usuaris especialitzats, tenen certs coneixements que els permeten usar sistemes guiats de comunicació amb les dades (com assistents i eines visuals) o *aplicacions verticals* per a consultar o modificar dades. Cada grup d'usuaris disposa de vistes externes segons els seus permisos d'accés i d'execució específics.

Les aplicacions verticals

Es descriuen en el punt sobre llenguatges i aplicacions del subapartat "Llenguatges de base de dades" dins l'apartat "Funcions i components de l'SGBD".

Un tipus específic d'usuari habitual és l'*usuari paramètric*.

- **Usuaris paramètrics**

Es tracta d'usuaris quotidians dedicats gairebé exclusivament a treballar amb la base de dades, que no necessiten conèixer gaire sobre els recursos de l'SGBD. N'hi ha prou si aprenen a realitzar un tipus de consultes i actualitzacions estàndard, anomenades *transaccions programades*, que s'han dissenyat, implementat i provat acuradament per a ells. La seva funció principal pot consistir, per exemple, a introduir dades de manera massiva i constant.

Les transaccions programades

Són procediments o conjunts d'operacions (que poden ser d'inserció, d'eliminació, de modificació i/o de recuperació) invocats pels usuaris paramètrics des d'una interfície d'usuari (com un formulari) per a l'execució correcta de les operacions d'emmagatzematge i recuperació d'informació.

Aquests usuaris, també anomenats *terminals*, solen formar part de l'estructura informàtica de l'empresa a nivell d'operadors de consola o administratius i representen una part important de la totalitat d'usuaris finals.

Els llenguatges i interfícies

Els llenguatges i interfícies que permeten als usuaris paramètrics usar transaccions programades es tracten en l'apartat "Funcions i components de l'SGBD".

Exemples d'usuaris paramètrics

Els usuaris paramètrics poden ser de tipus molt diversos:

- *Caixers* d'entitats bancàries, que revisen saldos i fan dipòsits i reintegraments de diners.
- *Encarregats de reserves* de línies aèries, hotels, companyies de lloguer de cotxes o espectacles que revisen la disponibilitat i fan reserves.
- *Personal de correus*, que introdueix la identificació dels paquets (mitjançant codis de barres) i informació descriptiva (a través de botons) amb l'objectiu d'actualitzar la base de dades centralitzada dels paquets rebuts i dels que estan en camí.

2) Usuaris avançats

Estan prou familiaritzats amb la major part de recursos de l'SGBD (utilitats, LMD, etc.) com per a executar processos propis i implementar aplicacions que compleixin els seus requeriments més o menys complexos. Necessiten una bona gestió de la informació que és processada per un altre sistema (comercial o desenvolupat per ells) i, per tant, utilitzen l'SGBD com a submòdul o eina per al desenvolupament d'altres sistemes particulars.

Funcionalment, els usuaris avançats es corresponen amb usuaris experts que utilitzen les dades per a fer anàlisis de suport a la presa de decisions en l'organització.

Exemples d'usuaris avançats

Poden ser enginyers, científics, analistes de negoci o, fins i tot, dissenyadors de programes no relacionats amb els processos habituals que es duen a terme en la base de dades.

Un exemple són els especialistes en desenvolupament de *sistemes experts*, que utilitzen l'SGBD per a la gestió de la *base de fets* i la *metabase de coneixements* del sistema expert.

3) Usuaris ocasionals

Accedeixen a la base de dades esporàdicament i solen requerir informació diferent en cada ocasió. Per a especificar les seves peticions utilitzen interfícies amb un llenguatge de consulta d'alt nivell, però sense emprar ordres estructurades. Amb independència dels seus coneixements en tecnologia de bases de dades, per recuperar i manipular la informació solen utilitzar pocs recursos de l'SGBD.

Un tipus d'usuari ocasional molt concret és l'*usuari crític*.

- **Usuaris crítics**

Normalment, es tracta de **personal de gerència** o pertanyent a l'*staff* de l'organització que requereix, en un temps mínim, informació de la base de dades en un format, amb un detall i amb uns requisits no previstos d'antuvi.

Donat que la majoria d'SGBD disposen de *llenguatges de quarta generació* integrats (els quals permeten resultats eficaços en el desenvolupament de processos específics com, per exemple, la generació d'informes), l'usuari no necessita l'assistència de personal tècnic (programadors) per a fer consultes no predefinides. No obstant això, el temps de resposta és alt i, sovint, els resultats no són els desitjats, la qual cosa genera la **crítica generalitzada** d'aquest usuari que, d'altra banda, sol tenir poder decisori sobre aspectes d'inversió i contractació dins l'organització.

Els llenguatges de quarta generació (4GL, *4th generation languages*)...

Són llenguatges de molt alt nivell que solen combinar elements procedimentals amb elements declaratius. Bàsicament, faciliten el tractament de la base de dades, però també la definició de menús, quadres de diàleg, formularis de pantalla i informes.

4) Usuaris autònoms

Mantenen bases de dades personals utilitzant paquets de programari específics que tenen interfícies d'ús fàcil (de tipus gràfic o basades en menús), per als quals solen adquirir una gran habilitat.

Un exemple és l'usuari d'un paquet fiscal que emmagatzema dades financeres personals.

9. Model, esquema i estat de la base de dades

Una característica fonamental d'un sistema de base de dades és que proporciona un cert nivell d'abstracció de les dades perquè oculta detalls d'emmagatzematge que la majoria d'usuaris no necessita conèixer.

L'abstracció és l'acció i l'efecte d'abstreure, és a dir, el procés mental innat en l'ésser humà d'amagar els detalls i centrar-se en allò que és essencial. El seu objectiu és buscar les propietats comunes d'una realitat o d'un conjunt d'objectes del món real reduint la complexitat i ajudant a comprendre-la. Mitjançant aquest procés, els elements o les qualitats d'una realitat són destriats dels altres per a considerar-los aïlladament, o bé aquesta realitat és considerada en la seva pròpia essència.

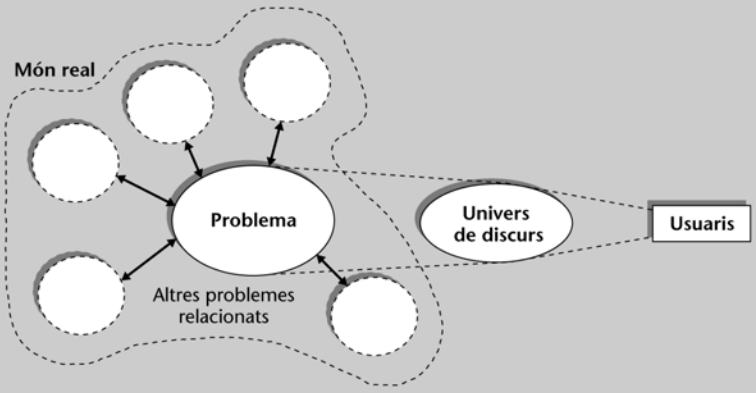
9.1. Model de dades

Un **model de dades** és un instrument que proporciona els mitjans necessaris per a aconseguir el procés d'abstracció que condueix de la part del món real que interessa estudiar, anomenada *univers del discurs*, al món de les dades.

Univers de discurs d'un problema del món real

Un model de dades permet representar qualsevol fenomen real o abstracte, és a dir, qualsevol **problema** sobre el qual es vol obtenir informació per al seu coneixement. Per a la representació del coneixement sobre un problema és necessària la seva caracterització.

Figura 4. Univers de discurs i límits d'un problema del món real



Definició i exemples de model

Un model és la simulació o representació (gràfica, matemàtica, conceptual, etc.) del comportament d'un sistema material o immaterial del món real. És la representació simplificada de part de la realitat que hom pren com a objecte d'estudi a fi de reproduir-la.

En són exemples un model matemàtic de la distribució de la matèria a l'univers, un model gràfic d'una molècula, un model numèric d'un full de càlcul o sobre operacions financeres, un model meteorològic.

El problema a representar es pot considerar un sistema en què intervenen una sèrie de propietats que es poden interrelacionar amb la resta de propietats del sistema. Un **sistema** és un conjunt d'elements, el comportament dels quals queda determinat per les seves propietats i interdependències. Per a l'estudi d'un sistema és necessària la simplificació (basada en l'abstracció) del problema que representa, mitjançant la determinació dels seus límits respecte a la resta de problemes o fenòmens existents i les propietats d'interès (que representen les dades que intervenen en el sistema).

L'univers de discurs (a vegades anomenat *minimón*) és la visió concreta que tenen del problema del món real els usuaris de la base de dades (en especial, el dissenyador).

Un model de dades és una eina intel·lectual (un conjunt de conceptes, regles i convenis definits explícitament) que permet descriure l'estructura d'una base de dades en nivells d'abstracció diferents. El concepte d'estructura d'una base de dades fa referència a elements com els tipus d'objectes, les seves associacions (o interrelacions) i les restriccions que han de complir.

L'ús del terme *model* en l'entorn de bases de dades

- No s'ha de confondre la **base de dades** considerada com a *model de la realitat* amb el concepte **model de base de dades** que aquí es tracta.
- La denominació correcta és **model de base de dades**, malgrat que s'acostuma a utilitzar l'expressió **model de dades**, ja que permet modelitzar les dades (a pesar que alguns autors consideren que l'expressió no està ben escollida perquè *és menys un model en si mateix que un marc per a concebre models del món real*).

Els models de dades es basen en l'abstracció per a definir categories o tipus d'objectes, ja que la utilització de l'ordinador requereix la tipificació de les dades tractades.

9.1.1. Ocurrència i tipus

Determinats elements d'una base de dades poden tenir dues accepcions: tipus i ocurrència.

- El **tipus** és una categoria generalitzada que descriu un conjunt d'elements més específics que tenen les mateixes propietats. Els tipus equivalen a les categories que s'utilitzen per a organitzar informació del món real. Per exemple, els éssers vius s'organitzen en animals i vegetals.

  En l'entorn dels models de dades, els conceptes *categoria*, *classe* o *tipus* són sinònims. El terme més utilitzat és *tipus*, mentre que *classe* se sol utilitzar en els models orientats a l'objecte.

- Una **ocurrència** és un cas concret d'un tipus. Les ocurrències comparteixen propietats similars, però cada element té el seu propi valor (o conjunt de valors) per a cada propietat. Per exemple, un gerani i un roure són ocurrències de vegetals que, malgrat que comparteixen determinades propietats, tenen característiques diferencials.

  El terme més utilitzat és *ocurrència*. No obstant això, també s'utilitza sovint el terme *instància*, encara que el seu significat no respon al concepte exposat. Altres sinònims d'ocurrència més apropiats però menys utilitzats són *exemplar*, *realització* o *extensió*.

Una ocurrència és el resultat d'una instanciació i un tipus és el resultat d'una classificació.

Les dues accepcions descrites s'apliquen als conceptes:

Instanciació i classificació

Es relacionen amb ocurrència i tipus tal com es descriu en l'apartat "Tipus d'abstracció en el disseny de bases de dades".

- **Tipus d'objecte**: és la conceptualització d'un objecte genèric del món real com a classe.

  Per exemple, no un estudiant concret ni el conjunt d'estudiants, sinó l'abstracció ESTUDIANT.

  Generalment, els diferents tipus d'objectes de l'univers de discurs s'indiquen amb majúscules. Per exemple, COTXE, VEHICLE, DOCUMENT, PROFESSOR, ESTUDIANT, PERSONA.

- **Ocurrència d'objecte**: és la conceptualització d'un objecte concret del món real com a individu. Per exemple, l'estudiant Miquel Nebot Jover.

  Cada ocurrència d'objecte pertany a un tipus d'objecte genèric. Totes les ocurrències d'un mateix tipus es caracteritzen per tenir les mateixes propietats (o atributs). Per exemple, tots els estudiants tenen nom, data de naixement, número de matrícula, DNI, etc.

Utilització del termes *objecte* i *ocurrència*

Per tal de simplificar, amb freqüència s'utilitza simplement el terme **objecte** per a referir-se a un "tipus d'objecte", i l'expressió **ocurrència d'objecte** per a referir-se a un objecte concret d'aquest tipus.

Aquesta extralimitació en l'ús del llenguatge no comporta cap error d'interpretació, atès que habitualment pel context es pot deduir a quina de les dues accepcions es fa referència (ocurrència o tipus). Si no queda prou clar, però, cal especificar-ho.

9.2. Esquema de la base de dades

En qualsevol model de dades és important la distinció entre la descripció de la base de dades i la mateixa base de dades; és a dir, entre l'esquema i l'estat de la base de dades.

L'**esquema** és la descripció de l'estructura de la base de dades, que s'especifica durant el procés de disseny de la base de dades i, en principi, és invariable en el temps.

Els **elements de l'esquema** són cadascun dels objectes de l'esquema, les seves associacions, les propietats i les restriccions.

La major part de models de dades utilitzen certes convencions per a representar els esquemes. La representació d'un esquema s'anomena **diagrama de l'esquema** i presenta l'estructura de tots els tipus d'objectes però no les ocurrències d'objectes. Aquest diagrama visualitza únicament alguns aspectes de l'esquema com, per exemple: els noms dels tipus d'objecte, els noms de les propietats i alguns tipus de restriccions. En canvi, no apareixen reflectits els tipus de dades de cada propietat, o alguns tipus de restriccions que són molt difícils de representar.

Exemple de diagrama de l'esquema

El diagrama següent representa l'esquema d'una base de dades que emmagatzema informació de les assignatures, els estudiants i les qualificacions dels diferents estudis o facultats d'una universitat.

ASSIGNATURA

Codi assignatura	Nom assignatura	Nombre crèdits	Codi estudis

ESTUDIS

Codi estudis	Nom estudis

ESTUDIANT

Codi estudiant	Nom i cognoms estudiant	DNI estudiant	Telèfon estudiant	Data naixement

PROFESSOR

Codi professor	Nom i cognoms professor	DNI professor	Telèfon professor	Adreça professor

CURS

Codi curs	Codi assignatura	Codi professor	Semestre-any

QUALIFICACIÓ

Codi estudiant	**Codi curs**	Qualificació

Amb majúscula s'indiquen els noms dels tipus d'objecte i enquadrats, els noms de les seves propietats (atributs). En negreta apareixen els conjunts d'atributs que formen clau primària.

Atribut* i *clau primària

Són conceptes que es desenvolupen amb detall en el mòdul "El model relacional i l'àlgebra relacional".

9.3. Estat de la base de dades

Les dades emmagatzemades en la base de dades poden canviar molt sovint; per exemple, quan s'afegeix una nova ocurrència d'un element de dades.

El conjunt de valors que prenen els objectes d'un esquema en un moment donat rep el nom d'**estat de la base de dades**.

També s'anomena *conjunt actual d'ocurrències* de la base de dades, ja que en un estat concret cada element de l'esquema té el seu propi conjunt d'ocurrències. Per exemple, el conjunt d'estudiants individuals (els registres) són les ocurrències de l'element ESTUDIANT. També es poden utilitzar les denominacions *ocurrència de l'esquema* o *extensió de l'esquema*.

A un esquema concret li poden correspondre molts estats de la base de dades. Cada vegada que es modifica el valor d'un element de dades o que s'afegeix o s'elimina un registre, la base de dades passa d'un estat a un altre.

Els estats en què es pot trobar una base de dades són:

- **Estat buit** (o sense dades): quan s'acaba de definir una base de dades nova, és a dir, quan se n'ha especificat l'esquema.

- **Estat inicial**: després de carregar-hi les dades per primer cop.

- **Estat actual**: en qualsevol moment (en general, després d'executar qualsevol operació d'actualització).

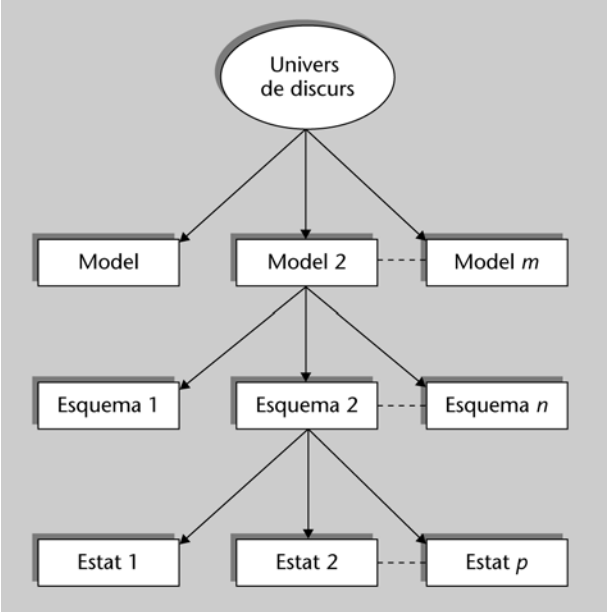
L'SGBD ha d'assegurar que tots els estats de la base de dades siguin *estats vàlids*, és a dir, que satisfacin l'estructura i les restriccions especificades en l'esquema. L'SGBD emmagatzema les descripcions d'aquests elements de l'esquema en el seu diccionari de dades perquè, quan calgui, siguin consultats pel programari de l'SGBD.

9.4. Relació entre model, esquema i estat

Com s'ha vist, un **model** és el conjunt de regles per a estructurar les dades de l'univers de discurs; un **esquema** és la percepció d'un univers de discurs determinat interpretat segons un model concret, i l'**estat** és el valor que pren l'esquema en un moment determinat.

La descripció de l'univers de discurs del món real mitjançant un model de dades dóna com a resultat un esquema de la base de dades, que pot tenir diversos estats.

Figura 5. Model, esquema i estat



Els estats representats en la figura corresponen a l'esquema 2, que s'ha obtingut aplicant el model de dades 2 a l'univers de discurs d'un problema del món real.

Un **model de dades** és un conjunt d'eines conceptuals que, fonamentant-se en l'abstracció, permet descriure els objectes que intervenen en un sistema, així com també les seves relacions, els limitants d'integritat que els afecten i la terminologia a utilitzar.

Un **esquema** és la representació abstracta de les propietats estàtiques i dinàmiques d'una part del món real utilitzant un model de dades i atenent els seus aspectes fonamentals més significatius.

L'**estat d'una base de dades** és un conjunt de dades estructurades segons un esquema determinat en un moment donat.

9.5. Evolució de l'esquema

En principi, l'**esquema és invariable** en el temps, ja que descriu l'estructura de la base de dades, mentre que l'**estat canvia molt sovint** perquè depèn de les ocurrències de la base de dades en un moment donat.

Malgrat que se suposa que l'esquema no varia, molt esporàdicament pot ser necessària la seva modificació si canvien els requeriments de l'univers de discurs. Els SGBD més moderns inclouen operacions per a tenir en compte l'evolució de l'esquema, que es poden aplicar mentre la base de dades està operativa, tot i que aquest procés és més complicat que l'execució de simples actualitzacions en la base de dades.

9.6. Intensió i extensió de l'esquema

L'esquema de la base de dades es pot descriure en termes d'intensió i extensió. Així, s'anomena *intensió* el mateix esquema i *extensió* de l'esquema l'estat de la base de dades.

- La **intensió** és la representació genèrica de les ocurrències de la base de dades. Es correspon amb el **tipus**, el qual s'obté per *classificació* d'un conjunt d'objectes en un ens de nivell d'abstracció superior. La intensió és la part definitòria i estàtica (invariable en el temps) de l'esquema.

- L'**extensió** (efecte de fer extensiu o aplicable) és la concreció del **conjunt d'ocurrències** del tipus (s'obté per *instanciació*) que, en un moment determinat, satisfan l'esquema i estan emmagatzemades en la base de dades. L'extensió varia en el transcurs del temps.

Intensió i extensió

Es relacionen amb els tipus d'abstracció, classificació i instanciació, tal com es veu en l'apartat següent, "Tipus d'abstracció en el disseny de bases de dades".

10. Tipus d'abstracció en el disseny de bases de dades

Els models de dades (especialment els semàntics) ofereixen tipus d'abstracció diferents per a facilitar la representació de les dades en el disseny de bases de dades. Els tipus d'abstracció bàsics són la **classificació**, la **generalització**, l'**agregació** i l'**associació**. Combinant-los s'obtenen mecanismes semàntics eficients per a estructurar les dades.

Les abstraccions permeten establir vincles entre els elements d'un model. En la *classificació* s'estableix un vincle entre una categoria (*tipus*) d'objectes i cada objecte concret (*ocurrència*) d'aquesta categoria, mentre que en la *generalització,* l'*agregació* i l'*associació* el vincle s'estableix entre categories d'objectes (i, per tant, també entre les seves ocurrències corresponents). D'altra banda, la classificació, la generalització, l'agregació i l'associació són síntesis conceptuals, mentre que els processos inversos respectius (la *instanciació*, l'*especialització*, la *desagregació* i la *dissociació*) són refinaments conceptuals.

Ocurrència i tipus es defineixen en el subapartat "Model de dades" de l'apartat "Model, esquema i estat de la base de dades".

Un altre tipus d'abstracció (que no es considera bàsica) és la *identificació*, la qual permet identificar de manera única els tipus d'objecte i les seves ocurrències mitjançant un identificador.

A continuació, s'exposen aquests tipus d'abstracció.

10.1. Classificació (i instanciació)

La **classificació** és el procés d'abstreure les característiques comunes a un conjunt d'ocurrències per a crear una categoria a la qual pertanyin aquestes ocurrències.

Permet categoritzar un conjunt d'objectes (que comparteixen les mateixes propietats, associacions i restriccions) en una nova categoria de nivell més alt, anomenada **classe** o **tipus** d'objecte. La classificació correspon a la funció de **pertinença** a un conjunt. Una **instància** o **ocurrència** està relacionada amb el tipus mitjançant les relacions "pertany a", "és membre de" o "és una ocurrència de" un tipus. Per exemple, *gener* és una ocurrència del tipus MES.

Segons la teoria de conjunts, el tipus es pot considerar la **intensió** (part definitòria) de totes les ocurrències possibles d'aquest tipus. El conjunt d'aquestes ocurrències en un moment donat constitueix una **extensió** del tipus corresponent.

La **instanciació** (o particularització) és el procés invers a la classificació. Consisteix en la generació i l'examen dels diferents objectes particulars (ocurrències) a partir del tipus que els descriu.

Figura 6. Classificació i instanciació

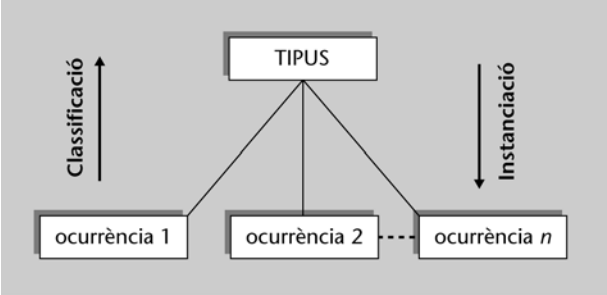
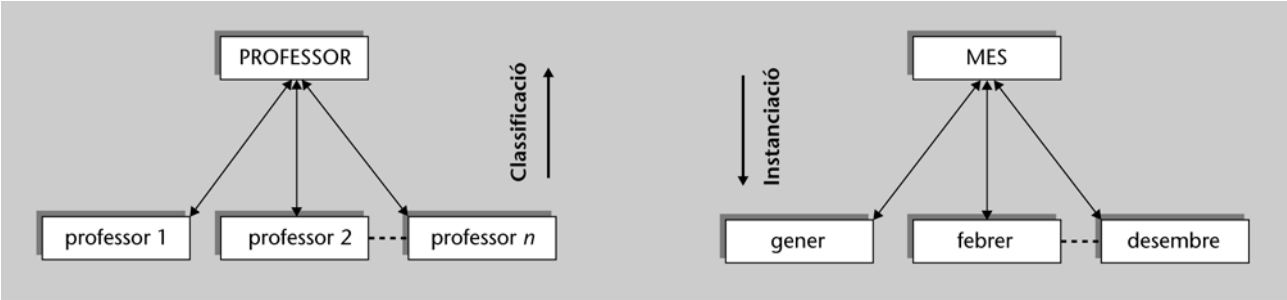


Figura 7. Exemples de classificació/instanciació



Cada un dels professors d'una universitat *es categoritza* ← *Clasificació* → Cada un dels mesos de l'any *es categoritza* en el tipus PROFESSOR. en el tipus MES.

Cada un dels professors d'una universitat *és una ocurrència* del tipus PROFESSOR. ← *Instanciació* → Cada un dels mesos de l'any *és una ocurrència* del tipus MES.

La classificació es pot considerar una mena de **generalització**, ja que permet agrupar cada una de les ocurrències d'objecte en un tipus d'objecte. Cal advertir, però, que la generalització estableix únicament vincles entre tipus d'objecte.

10.2. Generalització (i especialització)

La **generalització** és el procés d'agrupar diversos tipus d'objecte per a obtenir un tipus més general (de nivell d'abstracció més alt) que inclou els objectes de tots aquests tipus.

Per agrupació d'objectes més simples (els **subtipus**) s'obté un nou element, el **supertipus**. Per a relacionar un subtipus amb un supertipus s'utilitzen les expressions "és un subtipus de" o simplement "és un". Per exemple, un *llibre* és un subtipus de DOCUMENT.

L'**especialització** és el procés invers pel qual es categoritza un tipus d'objecte en subtipus més especialitzats. En el món real és habitual la descomposició (o

especialització) d'un tipus d'objecte creant una jerarquia en què es distingeix un supertipus del qual depenen diversos subtipus.

Figura 8. Generalització i especialització

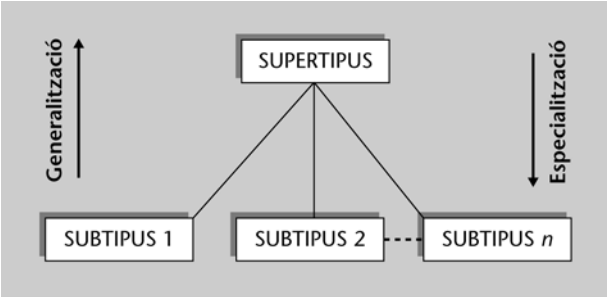
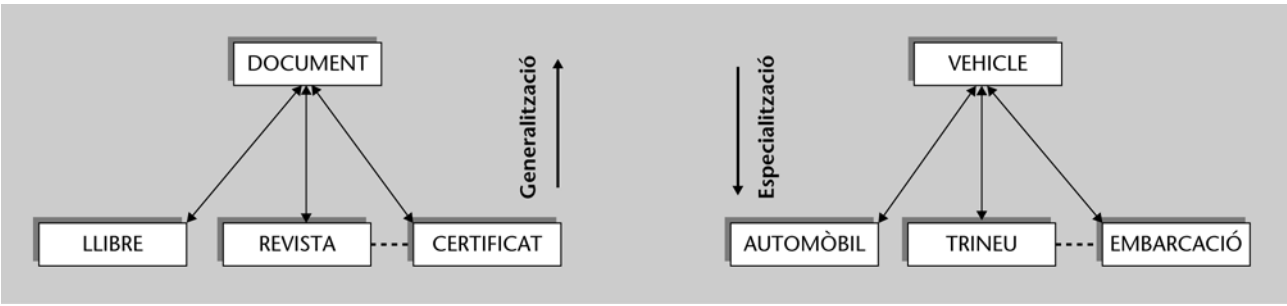


Figura 9. Exemples de generalització/especialització



Els tipus d'objecte LLIBRE, REVISTA i CERTIFICAT *s'agrupen* en el nou tipus d'objecte DOCUMENT, que és el supertipus de la jerarquia. ← *Generalització* → Els tipus d'objecte AUTOMÒBIL, TRINEU i EMBARCACIÓ *s'agrupen* en el nou tipus d'objecte VEHICLE.

Els tipus d'objecte LLIBRE, REVISTA i CERTIFICAT *són subtipus* del supertipus DOCUMENT. ← *Especialització* → Els tipus d'objecte AUTOMÒBIL, TRINEU i EMBARCACIÓ *són subtipus* del supertipus VEHICLE.

La generalització no s'utilitza en tots els models de dades, però s'està introduint en les extensions dels diferents models, com per exemple en el model entitat-interrelació E/R.

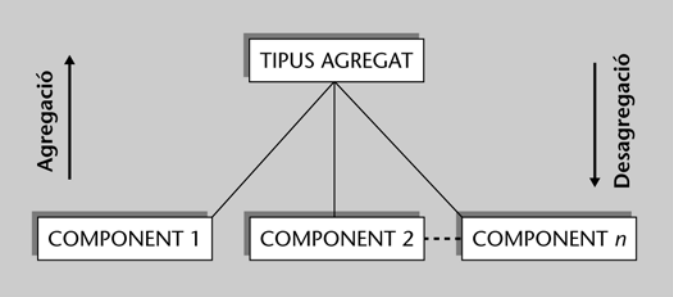
10.3. Agregació (i desagregació)

L'**agregació** és el procés de construir objectes compostos a partir dels seus objectes components.

Permet considerar un objecte segons els elements que el constitueixen. Bàsicament, es poden agregar tipus (per a obtenir un tipus compost, de nivell més alt), però també propietats (per a obtenir una propietat composta o un tipus d'objecte). Per a relacionar els objectes components amb l'objecte agregat s'utilitzen les nocions "és part de" o "és un component de".

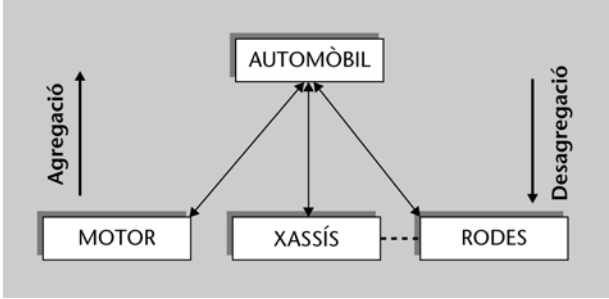
La **desagregació** (o descomposició) és el procés invers a l'agregació. Consisteix a passar de l'element compost als seus components. Permet representar tant els objectes components com les propietats que caracteritzen un tipus d'objecte.

Figura 10. Agregació i desagregació



Els components també són tipus.

Figura 11. Exemple d'agregació/desagregació



Els tipus d'objecte MOTOR, XASSÍS i RODES *són components* del tipus d'objecte AUTOMÒBIL.

Agregació: El tipus AUTOMÒBIL *és l'agregació* dels tipus MOTOR, XASSÍS i RODES.

Desagregació: Els tipus d'objecte MOTOR, XASSÍS i RODES *s'obtenen per descomposició* del tipus AUTOMÒBIL.

10.4. Associació (i dissociació)

L'**associació** és el procés de vincular dos objectes o més de diversos tipus independents (permet vincular tant els tipus com les seves ocurrències).

S'obté un nou element, *l'associació*, amb característiques distintives pròpies que la distingeixen dels participants. Per tal de relacionar un tipus amb un altre s'utilitza la noció "està associat a".

La **dissociació** és el procés invers a l'associació. Consisteix en l'eliminació de l'associació.

Figura 12. Associació i dissociació

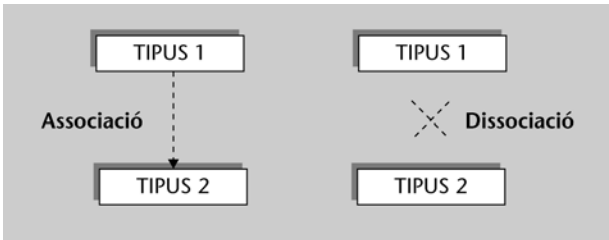
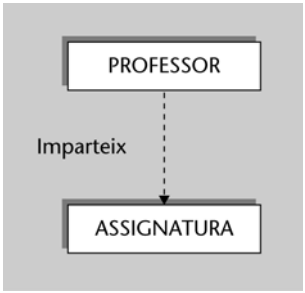
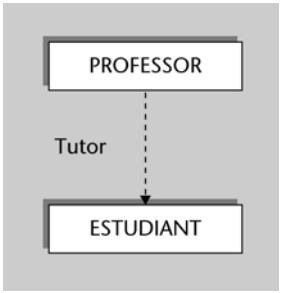


Figura 13. Exemples d'associació



El tipus PROFESSOR *està associa*t al tipus ASSIGNATURA mitjançant l'associació IMPARTEIX.



El tipus PROFESSOR *està associat* al tipus ESTUDIANT mitjançant l'associació TUTOR.

L'associació és un tipus d'agregació, malgrat que en alguns models de dades no existeix cap concepte especial per a representar-la com a abstracció.

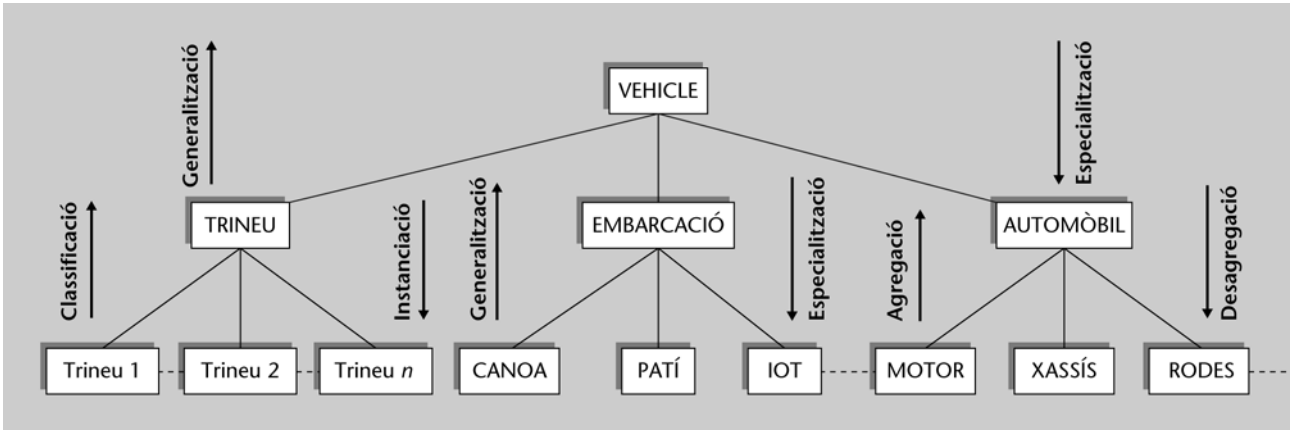
L'associació en els models de dades: relació i interrelació

En el *model relacional*, tant els tipus com les associacions entre tipus es representen mitjançant **relacions**, i no es fa distinció entre els uns i les altres. En canvi, l'associació és un concepte fonamental en el *model entitat-interrelació* en què el tipus nou s'anomena d'**interrelació**.

Hi ha diferències importants entre agregació i associació. La principal diferència estructural és que, si s'elimina una ocurrència d'associació, els objectes participants poden continuar existint, mentre que l'eliminació de l'objecte agregat implica l'eliminació dels seus objectes components. Per exemple, si s'elimina l'objecte COTXE, s'eliminaran els seus components MOTOR, XASSÍS i RODES.

10.5. Utilització de l'abstracció

Figura 14. Exemple d'aplicació de diversos tipus d'abstracció



L'**especialització** permet l'abstracció del tipus d'objecte *vehicle* als subtipus *trineu*, *embarcació* i *automòbil*. El procés invers d'abstracció d'aquests últims a vehicle es fa en la **generalització**.
El mateix procés d'especialització permet abstraure del tipus d'objecte *embarcació* a les seves tipologies específiques (*canoa*, *patí*, *iot*…).
D'altra banda, la **instanciació** és el tipus d'abstracció que permet especificar diverses ocurrències del tipus d'objecte *trineu* (que, per exemple, poden pertànyer a propietaris o fabricants diferents). El procés d'abstracció invers permet la **classificació** d'aquestes ocurrències en el tipus *trineu*.
La **desagregació** és el procés de descomposició del tipus d'objecte *automòbil* en els seus components (*motor*, *xassís*, *rodes*…). El procés d'abstracció invers permet l'**agregació** d'aquests components en el tipus d'objecte automòbil.

11. Arquitectura dels SGBD

En aquest apartat es presenta l'arquitectura dels SGBD segons criteris diferents: la distribució del seu processament, els nivells d'abstracció de les dades i la seva composició interna i externa.

L'arquitectura estàndard dels SGBD actuals divideix la base de dades en tres nivells d'abstracció, segons la perspectiva des de la qual és vista. S'examinen les correspondències entre aquests nivells i les limitacions que comporta la lligadura per a la independència de les dades.

11.1. Les diferents arquitectures dels SGBD

L'arquitectura d'un SGBD es pot considerar des d'enfocaments o punts de vista diferents, els presentem a continuació.

Arquitectura

Una arquitectura (com qualsevol altra manera d'esquematitzar la realitat) és una eina senzilla i potent per a abstreure i entendre les característiques fonamentals d'un sistema.

11.1.1. Arquitectura operacional

L'arquitectura operacional especifica com s'organitza la *distribució del processament* de l'SGBD en el sistema informàtic.

1) Arquitectura centralitzada (utilitzada en els primers sistemes)

Consta d'un ordinador central únic en què es fa tot el processament de l'SGBD i en què resideixen les dades de la base de dades. Els usuaris accedeixen al sistema de manera remota (per mitjà de **xarxes de comunicacions**) a través de terminals sense poder de processament, els quals només ofereixen capacitat de visualització. L'ordinador central envia als terminals la informació i els controls de pantalla.

2) Arquitectura client-servidor (predomina en els sistemes actuals)

El processament de l'SGBD es reparteix entre dos tipus de mòduls: els clients (la interfície amb l'usuari, que se sol executar sobre un ordinador personal) i el servidor o els servidors (en què s'executa el cor de l'SGBD, anomenat *motor*).

La funcionalitat del sistema se sol distribuir de la manera següent:

a) El **mòdul client** maneja la interacció amb l'usuari. Normalment, proporciona els programes d'aplicació i les interfícies d'usuari (GUI basades en formularis i menús) que accedeixen a la base de dades.

b) El **mòdul servidor** s'encarrega de fer les tasques pròpies de la base de dades. Normalment, maneja l'emmagatzematge, l'accés i la cerca de dades.

Els dos mòduls es poden executar en el mateix ordinador o, habitualment, en ordinadors diferents si aquests s'interconnecten a través d'un **sistema de comunicacions**.

Atès que l'execució de la base de dades es pot fer sobre un únic ordinador, es pot dir que el mòdul servidor és, en si, l'SGBD i el client és el consumidor de recursos de la base de dades.

Llenguatge per a la definició i consulta de bases de dades

Per a la comunicació entre el servidor i la base de dades s'utilitzen un llenguatge de consulta i protocols de xarxa. SQL (sigles de *structured query language,* llenguatge de consulta estructurat en anglès) és el llenguatge estàndard per a la definició i l'accés a bases de dades relacionals en sistemes client-servidor.

11.1.2. Arquitectura de referència

L'arquitectura de referència determina els *nivells d'abstracció de les dades* en l'SGBD. Pot ser de dos o de tres nivells:

1) Arquitectura de dos nivells (utilitzada en els sistemes clàssics, avui en desús)**.**

a) El **nivell lògic** oculta els detalls físics d'emmagatzematge i accés a les dades. Inclou entitats, atributs, interrelacions i regles d'integritat.

b) El **nivell físic** inclou elements d'emmagatzematge físic com índexs, espai físic en què s'agrupen els registres, magnitud de les pàgines o blocs, etc.

2) Arquitectura de tres nivells (predomina en els sistemes actuals, encara que no en tots)**.**

a) El **nivell extern** gestiona la informació des del punt de vista individual de cada grup d'usuaris.

L'enfocament de tres nivells

És objecte dels continguts del subapartat "Arquitectura de nivells dels SGBD" dins d'aquest mateix apartat.

b) El **nivell conceptual** descriu l'estructura de la base de dades per a tots els usuaris amb independència de l'ordinador en què aquesta s'implementi.

c) El **nivell intern** descriu la informació en funció del sistema en què s'implementarà la base de dades.

11.1.3. Arquitectura externa

L'arquitectura externa (o aparent) es refereix a les aplicacions que s'executen en el programari de l'SGBD i que l'usuari veu com un conjunt de parts més o menys ben estructurat.

Malgrat que aquesta arquitectura sol estar determinada per criteris comercials, no hi ha gaires diferències externes entre els SGBD del mercat. Pot canviar l'empaquetat, la presentació o el nom de les diferents utilitats per raons comercials. Destaquen les utilitats següents:

- Processadors de consultes (o generadors de vistes).
- Generadors de formularis (o pantalles).
- Generadors d'informes i gràfics.
- Generadors de menús.
- Processadors de llenguatge natural.

11.1.4. Arquitectura interna

L'arquitectura interna (o funcional) és la manera com l'SGBD s'estructura internament en parts components amb les seves funcions.

Les funcionalitats que proporciona un SGBD s'agrupen en blocs o mòduls de programari anomenats **gestors**. L'arquitectura funcional és una simplificació i, per tant, no coincideix necessàriament amb l'arquitectura real dels components que constitueixen un SGBD específic.

Els components interns de l'SGBD

Són objecte dels continguts del subapartat "El nucli de l'SGBD" dins l'apartat "Funcions i components dels SGBD".

Cada SGBD estructura internament el conjunt de la seva funcionalitat de manera diferent. Fins i tot, un mateix SGBD pot anar canviant d'estructura interna en versions successives per tal de millorar el rendiment i facilitar l'addició de funcionalitats noves.

Funcionalitat dels mòduls de l'SGBD

El motor de l'SGBD té mòduls per a executar funcions específiques com la interpretació d'SQL, l'optimització de consultes, la gestió de vistes, la gestió de memòries intermèdies, la gestió de l'espai de la memòria externa, la gestió de processos i de transaccions, el control de l'accés concurrent, la creació de còpies de seguretat, el control de la privadesa…

Existeix un alt grau d'**encavalcament** (execució simultània concurrent de fases d'instruccions consecutives) entre algunes de les funcions d'un SGBD (les més físiques o bàsiques) i les del **sistema operatiu** (SO). Per a aspectes determinats com, per exemple, la gestió de memòries intermèdies, la gestió de l'espai del 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. Algunes raons per a no aprofitar l'SO són que l'SGBD ho pot fer de manera més eficient o per a augmentar la confidencialitat.

11.2. Arquitectura de nivells dels SGBD

11.2.1. Nivells d'abstracció i esquemes

Perquè una base de dades pugui satisfer els requisits que se li exigeixen, és necessari que els seus usuaris tinguin una visió tan abstracta com sigui possible de les dades emmagatzemades.

Hi ha tres característiques inherents a l'enfocament de bases de dades que és important destacar. Són la separació entre els programes i les dades (anomenada **independència de dades**), el suport de múltiples **vistes d'usuari** i la utilització d'un **catàleg** per a emmagatzemar la descripció de la base de dades.

Les característiques de les bases de dades

Les característiques s'enumeren en l'apartat "Objectius i característiques de les bases de dades" d'aquest mòdul.

La **independència de dades** assegura que els programes d'aplicació són independents dels canvis introduïts en les dades que no s'utilitzen o en els detalls d'implementació de les dades a què s'accedeix. Aquesta propietat és una concreció del principi d'**abstracció de dades** dels llenguatges de programació, entès com la independència entre l'especificació de les estructures de dades i la seva implementació.

Segons qui accedeixi a la base de dades, aquesta ha de presentar una **vista** de les dades que l'usuari sigui capaç de reconèixer, interpretar i manejar. Per exemple, és convenient ocultar la complexitat de la base de dades a l'usuari final, ja que no té la necessitat de conèixer com s'organitzen físicament les dades.

Per tal d'aconseguir aquest objectiu, els SGBD permeten la definició de la base de dades en **nivells d'abstracció** diferents. La definició en cadascun d'aquests nivells s'anomena *esquema*.

Els conceptes *abstracció* i *esquema*

Es tracten en l'apartat "Model, esquema i estat de la base de dades" d'aquest mateix mòdul.

Els SGBD necessiten una descripció o definició de la base de dades, que rep el nom d'**esquema**.

L'esquema de la base de dades és un element fonamental de l'arquitectura d'un SGBD. Permet independitzar l'SGBD de la base de dades i permet que l'SGBD sàpiga com és l'estructura de la base de dades amb la qual ha de treballar. D'aquesta manera, es pot canviar el disseny de la base de dades (el seu esquema) sense haver de fer cap canvi en l'SGBD.

11.2.2. L'arquitectura de tres nivells ANSI/SPARC

L'arquitectura de dos nivells dels SGBD s'amplia a tres

La distinció clàssica entre dos nivells d'abstracció (lògic i físic) en els SGBD és ampliada a tres nivells l'any 1975 per l'arquitectura recomanada per l'ANSI/SPARC (*American National Standard Institute - Standards Planning and Requirements Committee*). La idea consisteix a descompondre el nivell lògic en dos: el nivell extern i el nivell conceptual. El nivell físic correspon al que s'anomenarà *nivell intern*.

Els nivells d'abstracció de l'arquitectura ANSI/SPARC

Es descriuen amb detall més endavant en aquest mateix apartat.

L'objectiu principal d'aquesta arquitectura estandarditzada per l'ANSI/SPARC és establir una divisió de la base de dades en tres nivells d'esquemes segons la perspectiva des de la qual aquesta és vista. És a dir, tres **nivells d'abstracció** que es corresponen amb els tres principals grups d'**usuaris** de la base de dades (usuaris finals, programadors i administradors). Aquests nivells s'introdueixen a continuació:

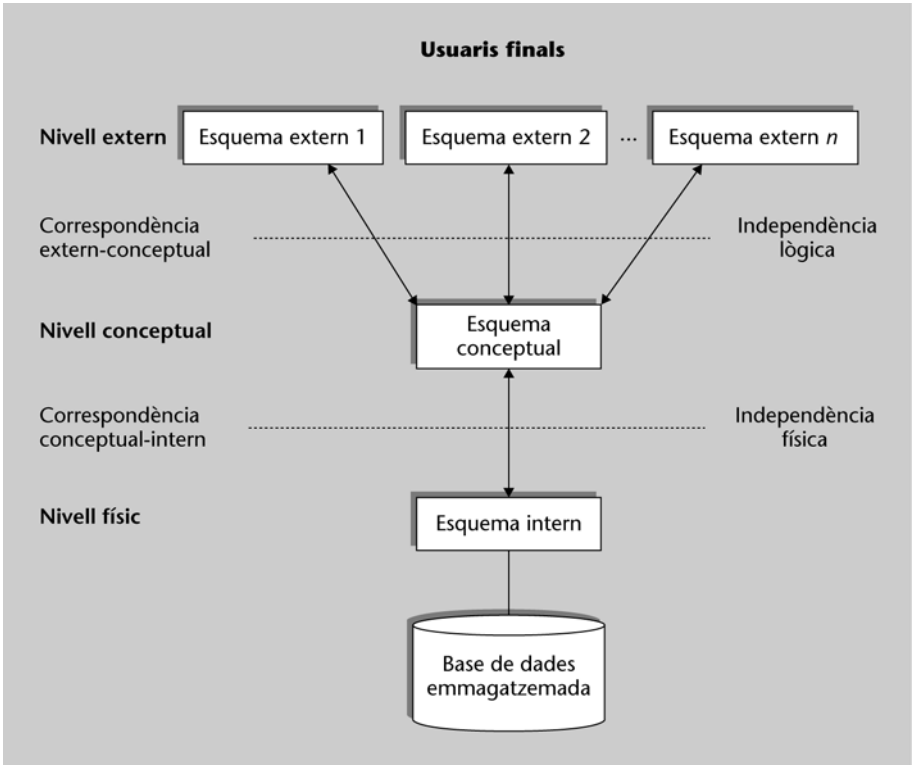
- El **nivell extern**, que és el nivell més abstracte, permet gestionar la informació des del punt de vista individual de cada usuari o grup d'usuaris. Es tracta de visions lògiques que els processos usuaris (programes d'aplicació i usuaris finals) tenen de les parts de la base de dades que utilitzen. Aquestes visions s'anomenen **esquemes externs**.

- El **nivell conceptual**, que es troba per sota de l'anterior, representa a alt nivell tota la informació de la base de dades independentment de l'ordinador en què s'implementi. En aquest nivell intermedi hi ha una descripció lògica bàsica, única i global, anomenada **esquema conceptual**, que descriu l'estructura de la base de dades completa per a una comunitat d'usuaris.

- El **nivell intern** és el nivell d'abstracció més baix, en què es descriu la informació en funció del sistema en el qual s'implementarà la base de dades. Aquesta descripció física, que és única, s'anomena **esquema intern**.

La figura següent de l'arquitectura de tres nivells d'un SGBD mostra els esquemes, els nivells, la correspondència entre aquests i la independència.

Figura 15. Arquitectura ANSI/SPARC de tres nivells d'un SGBD



Cada nivell representa una visió diferent de les dades de la bases de dades.

Nivells, esquemes i models de dades dels SGBD del mercat

La majoria d'SGBD no separen clarament els tres nivells de descripció. Però alguns, en certa manera, suporten l'arquitectura de tres esquemes:

- Alguns SGBD permeten incloure detalls del nivell físic en l'esquema conceptual.
- Quasi tots els SGBD que utilitzen vistes d'usuari especifiquen els esquemes externs en el mateix model de dades que descriu la informació a nivell conceptual.
- Alguns SGBD permeten utilitzar diferents models de dades en els nivells conceptual i extern.

Per exemple, en el **model de dades relacional**, que es limita al nivell lògic i no fa referència al nivell físic, els SGBD relacionals fan referència a un únic esquema (anomenat ***schema***), però aquest inclou descripcions dels tres nivells. En l'***schema*** es descriuen elements com entitats, atributs i restriccions (que corresponen a l'esquema conceptual en l'arquitectura ANSI/SPARC) i vistes (que equivalen a l'esquema extern). A més, els SGBD possibiliten incloure-hi descripcions d'estructures i característiques físiques com índex, espai de taula (*tablespace)*, unitat d'assignació d'espai de memòria (*cluster)*, etc. (que corresponen a l'esquema intern).

Es defineixen **esquemes** en els tres nivells: l'esquema intern (descripció de les característiques físiques), l'esquema conceptual (visió centralitzada) i els esquemes externs (visions dels usuaris).

Una base de dades específica té un únic esquema intern i un únic esquema conceptual, però pot tenir diversos esquemes externs, cadascun definit per a un usuari o més d'un.

L'arquitectura de tres esquemes és una eina adequada perquè l'usuari visualitzi els nivells de l'**esquema d'un sistema de base de dades**.

Per tal de crear una base de dades d'acord amb l'arquitectura ANSI/SPARC de tres nivells, cal definir-ne els esquemes. Seguint l'ordre d'execució de les fases del disseny d'una base de dades, primerament es defineixen els esquemes del nivell lògic (un esquema conceptual i un esquema extern, com a mínim) i, a continuació, es defineix un esquema intern. El procés seria el següent:

1) Definició de l'esquema **conceptual**. Aquest serveix de referència per als altres dos tipus d'esquemes i actua d'intermediari entre ells. Els tipus d'estructures que es poden utilitzar depenen del model de dades subjacent a l'SGBD.

2) Definició d'un esquema **extern**, com a mínim. Per a cada grup d'usuaris es pot definir una vista parcial de la base de dades, que consisteix en un conjunt d'estructures derivades a partir de les estructures de *l'esquema conceptual*.

3) Definició de l'esquema **intern**. Tot i que, per qüestions d'eficiència i rendiment, pot interessar definir els detalls d'organització física, si això no es fa, el

mateix SGBD s'encarrega de decidir una implementació per a cadascuna de les estructures definides en *l'esquema conceptual*.

A continuació, es desenvolupen en aquest mateix ordre els tres nivells de descripció de la base de dades.

11.2.3. El nivell conceptual

La descripció d'aquest nivell es fa mitjançant un **esquema conceptual** (també conegut simplement com a *esquema de la base de dades*.

1) S'obté a partir dels requeriments dels usuaris potencials del sistema per implantar i de les necessitats de l'empresa, per la qual cosa es crea de manera centralitzada durant l'anomenat *disseny lògic* de la base de dades.

2) Oculta els detalls de les estructures físiques d'emmagatzematge de les dades (forma i lloc).

3) Permet definir l'organització de les dades i descriu les interrelacions que s'estableixen entre elles, juntament amb altres característiques com la seguretat. Concretament, es defineixen:

a) Tots els elements de dades que intervenen en el sistema: els objectes (**entitats**), les seves propietats o característiques (**atributs**) i les dependències que existeixen entre ells (**interrelacions**).

b) Les ordres de control d'integritat, les restriccions i regles que regeixen el funcionament de l'empresa, les regles de validació, els valors permesos per als atributs, els tipus de dades, etc.

L'esquema conceptual és la representació abstracta d'un problema tal com aquest es presenta en el món real, independentment de com serà tractada la informació, de quins esquemes externs pugui tenir i de com aquesta informació pugui ser emmagatzemada físicament (esquema intern). L'esquema conceptual de la base de dades no canvia si no canvia la naturalesa del problema.

En aquest nivell es pot usar un model de dades d'alt nivell o un d'implementació.

Aquest nivell és el que utilitzen els programadors i els dissenyadors de la base de dades, que són responsables de considerar la informació prevista pels analistes de sistemes i crear adequadament les estructures lògiques de la base de dades.

Nota

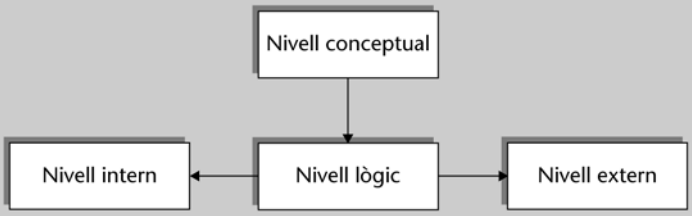
En aquest nivell, des del punt de vista funcional del programador, s'executen els processos següents:

- **Accés a l'esquema de la base de dades** mitjançant el llenguatge de definició de dades (DDL, *data definition language*) i el llenguatge de control de dades (DCL, *data control language*).
- **Crides al sistema** mitjançant el llenguatge de manipulació de dades (DML, *data management language*).
- **Utilització d'eines de desenvolupament visual** implementades en el gestor de base de dades.

El nivell lògic (independència del nivell conceptual)

Es pot considerar que hi ha un quart nivell d'abstracció en la representació de la informació en una base de dades: el nivell lògic. Per tal de centrar el tema, cal analitzar el nivell de descripció conceptual del qual deriva.

Figura 16. Relació del nivell lògic amb els altres nivells



El nivell conceptual és el més important, ja que s'hi sostenen els altres nivells i permet garantir la descripció correcta dels elements d'informació que intervenen en el sistema o en aquella part del sistema (el domini del problema) que es vol tractar (les entitats, les seves propietats i atributs, així com també les característiques de les relacions existents entre elles).

Hi ha moltes formes vàlides de descriure o representar, de manera abstracta, un sistema (un problema del món real):

- Es pot fer una **descripció estricta** d'un problema utilitzant diverses tècniques (diagrames d'estructures, diagrames d'entitats i relacions, arbres, taules, xarxes, etc.). Però **no tots els procediments són capaços d'interpretar qualsevol representació conceptual**.

- Una **representació abstracta** (conceptual) d'un problema suposa l'aplicació d'una sèrie de regles que restringeixen la manera en què aquest és representat. Però, en realitat, **el problema del món real és independent de la manera en què sigui representat** (per l'ésser humà).

D'aquí es conclou que l'esquema **conceptual** és independent dels esquemes extern i intern, ja que només depèn del problema objecte de la representació. Si el problema no canvia, el seu esquema conceptual no canvia, encara que canviïn els mecanismes per a la seva representació.

En canvi, la **representació del nivell lògic** sí que depèn de la manera, els mecanismes o els procediments pels quals es manipula la informació. L'esquema **lògic** deriva de l'esquema **conceptual** segons l'aplicació de regles i restriccions que tenen en consideració com la informació representada pot ser tractada pels procediments que manipulen i defineixen la informació segons les representacions.

Així, hi pot haver moltes representacions lògiques o canòniques d'una mateixa representació conceptual, de la mateixa manera que d'una representació lògica es deriven moltes representacions externes.

La inclusió d'aquest nivell nou de representació del domini del problema, que depèn del programari que manipula la informació, permet garantir la independència de la informació en una base de dades.

El nivell conceptual, de vegades, rep el nom de *nivell lògic global*.

11.2.4. El nivell extern

El nivell extern, també anomenat *nivell de visió* o d'usuari, defineix com la base de dades és percebuda per l'usuari final. Inclou un conjunt d'esquemes externs, que alguns sistemes anomenen *subesquemes*.

L'usuari veu la base de dades mitjançant un **esquema extern** apropiat a les seves necessitats. Cada **esquema extern** o **vista** individual descriu la part de l'esquema conceptual de la base de dades que interessa a un grup d'usuaris determinat i oculta a aquest grup la resta de la base de dades (sense preocupar-se per la correspondència amb l'esquema conceptual).

La visió externa oculta els detalls físics

De la mateixa manera que el nivell conceptual, el **nivell extern** oculta la complexitat interna de les bases de dades i ofereix a l'usuari una visió molt més senzilla, ja que aquest no necessita conèixer els detalls físics sobre com s'emmagatzemen les dades o com se'n fa el manteniment.

Cadascun dels esquemes externs es pot considerar com una *redefinició de l'esquema conceptual*, amb els elements que convinguin a les necessitats de cada usuari (o aplicació d'usuari). Cada usuari pot tenir la seva visió particular i parcial de l'esquema conceptual.

Per tant, l'usuari no percep la base de dades en la seva totalitat. La seguretat del sistema permet que, segons el nivell d'accés que se li hagi concedit, tingui accés a una part de la base de dades (les dades que necessita, les relacions que utilitza i les restriccions d'ús que se li hagin definit).

Hi ha tants esquemes externs com exigeixen les diferents aplicacions o els diferents usuaris. El mateix esquema extern (un subesquema) pot ser utilitzat per diverses aplicacions i compartit per diversos usuaris (i un usuari té la possibilitat d'accedir a diferents subesquemes). D'aquestes possibilitats sorgeix el concepte de **rol** o grup d'usuaris (conjunt de permisos i restriccions d'accés que poden atorgar-se a diversos usuaris).

Usuaris de l'esquema extern

Es poden distingir dos tipus d'usuaris pel que fa a l'esquema extern:

- Els que accedeixen a un esquema extern molt limitat a través de **sistemes de menús** i **transaccions programades** o predefinides. Habitualment, són usuaris finals paramètrics que només fan tasques de lectura o addició de dades (operadors de terminal o d'estació de treball).
- Els que accedeixen al seu **propi esquema extern**, des del qual fan totes les consultes i accions que desitgen. Són altres tipus d'usuaris com administradors, programadors o, fins i tot, usuaris finals més o menys experimentats (usuaris autònoms i avançats).

Aquests *subesquemes* o visions particulars dels usuaris són proporcionats pels procediments o programes d'aplicació que només manegen part de la informació de la base de dades.

Des del punt de vista funcional de l'usuari, el seu nivell extern consisteix en el conjunt d'objectes a què pot accedir. Per a ell el que té existència real són les taules i els seus registres, les vistes, els formularis, els informes, etc. Disposa d'un conjunt d'eines visuals de generació de consultes, formularis i informes, i un llenguatge de consulta per a la creació de programes que permeten accedir a les dades des de la base de dades.

Tasques que permet l'esquema extern

Algunes possibilitats que facilita la definició d'un esquema extern són les següents:

- Mostrar només els atributs i entitats que interessen i ometre'n d'altres
- Redefinir una entitat perquè sembli que són dues
- Definir combinacions d'entitats perquè sembli que siguin una de sola (combinant atributs d'entitats diferents)
- Definir dades derivades a partir d'atributs descrits en l'esquema conceptual (camps calculats...)
- Canviar el nom d'entitats i atributs
- Reordenar atributs

En aquest nivell es pot usar un model de dades d'alt nivell o un d'implementació.

El **nivell extern** és el conjunt de percepcions individuals que tenen els usuaris finals de la base de dades. Hi pot haver tants esquemes externs diferents com usuaris té la base de dades.

11.2.5. El nivell intern

És el nivell més baix i complex de descripció de la base de dades, també anomenat *nivell físic*.

Aquest nivell té associat un **esquema intern** que descriu els detalls d'emmagatzematge de les dades en els suports físics, així com també els camins d'accés i les especificacions de les estructures físiques que representen el domini del problema d'una manera comprensible per a l'ordinador (dispositius, volums, arxius, tipus de dades, apuntadors, etc.). L'esquema intern és la representació més propera a l'emmagatzematge físic de les dades.

Organització física de les dades

A l'hora d'especificar l'organització física de les dades en els dispositius d'emmagatzematge cal considerar els aspectes següents:

1) Estratègies d'emmagatzematge

Corresponen a l'assignació d'espais per al conjunt de dades.

a) *Gestió i reserva d'espai* per a emmagatzemar les dades (volum de la pàgina...).

b) *Descripció d'arxius* de la base de dades que contenen la informació (nom, tipus d'organització, unitat física en què s'emmagatzemen...).

c) *Descripció de taules* de dades que contenen els registres.

d) *Descripció de registres* (longitud i tipus de dades de cada camp).

2) Estratègies d'emplaçament

Corresponents a la designació dels llocs que han d'ocupar els elements de dades per tal d'optimitzar els temps de resposta i els espais de la memòria secundària.

a) *Mètodes o camins d'accés* als registres (especificacions de claus, índexs, apuntadors...)

b) *Compactació de dades* (desplaçament físic i concatenació de dades a fi de reduir la quantitat d'espai utilitzat i concentrar tot l'espai que no s'utilitza en una única àrea indivisa). Té relació amb la compressió o l'empaquetatge de dades (en castellà, *desfragmentación*).

c) *Xifratge de dades* (encriptació o codificació de les dades, mitjançant una clau adequada, a fi que resultin críptiques o secretes).

d) *Descripció del diccionari de dades* (*i el directori de dades*), si el sistema no fa aquesta tasca automàticament.

Els usuaris que intervenen en aquest nivell són els administradors de la base de dades.

L'esquema intern respon a les qüestions de **rendiment** (espai i temps de resposta) plantejades en elaborar el *disseny físic* de la base de dades i l'*ajust* d'aquesta a les necessitats canviants.

L'esquema intern utilitza un model de dades físic.

En el **nivell físic** es descriuen les dades des del punt de vista de l'ordinador que les emmagatzema.

11.2.6. Usuaris i nivells

En cada nivell es descriuen els objectes d'interès que poden ser entesos pels usuaris d'aquell nivell. Això es pot concretar de la manera següent:

- **L'usuari final** només entén de registres o formes de comunicació similars als documents externs manipulats en l'organització (**nivell extern**).
- El **programador**, a més del dissenyador i l'analista de sistemes, entén de tipus d'entitats o classes d'objectes, relacions existents entre ells i procediments executats per a la solució del problema que l'organització desitja tractar (**nivell conceptual**).
- **L'administrador** de la base de dades s'encarrega de determinar l'organització física que garanteixi el funcionament òptim del sistema (**nivell físic**).

11.2.7. Correspondència entre nivells

Cal assenyalar que els tres esquemes no són més que *descripcions* de les dades. De fet, les úniques dades que existeixen *realment* es troben en el nivell intern.

Però, en un SGBD basat en l'arquitectura de tres esquemes, cada grup d'usuaris fa referència exclusivament al seu propi esquema extern.

Lògicament, hi ha comunicació entre tots tres nivells d'esquemes, que l'SGBD s'encarrega de dur a terme mitjançant dos procediments de transformació de les dades.

L'SGBD ha de transformar una sol·licitud expressada en termes d'un esquema extern en una sol·licitud expressada en termes d'un esquema conceptual i després en una sol·licitud en l'esquema intern que, finalment, es processarà sobre la base de dades emmagatzemada.

El procés de transformar sol·licituds i resultats d'un nivell a un altre s'anomena **correspondència o transformació** *(mapping)*. Aquest procés es pot definir, de manera elemental, com l'assignació o conversió dels registres de dades reals des d'un suport físic en un format útil per a ser presentats a l'usuari (i el procés invers).

La integració entre els tres nivells s'estableix de la manera següent:

- **Correspondència extern-conceptual**: *transforma les dades des de la seva representació externa a la conceptual*.

  Quan un usuari, amb un nivell extern determinat, sol·licita mitjançant una consulta l'accés a unes dades determinades, l'SGBD interpreta la sol·licitud, verifica l'esquema extern definit per a aquest usuari i transforma la sol·licitud des del nivell extern al nivell conceptual.

- **Correspondència conceptual-intern**: *transforma les dades des de la seva representació conceptual a la interna*.

  Una vegada verificat l'esquema conceptual es passa la sol·licitud des del nivell conceptual a l'intern. En el nivell intern se selecciona l'estructura d'emmagatzematge de dades sobre la qual s'ha fet la petició i es duen a terme les operacions necessàries per a donar una resposta a la sol·licitud de l'usuari.

  Qualsevol modificació a nivell intern implicarà un canvi en aquesta correspondència sense que es modifiqui el nivell conceptual.

Aquest procés es repeteix a la inversa per a poder accedir als registres objecte de la consulta. En l'exemple exposat més endavant es mostren detalladament les diferents fases d'un procés sencer de transformació. El conjunt de procediments que transforma un nivell en un altre rep el nom de *regles de correspondència*.

Aquestes correspondències poden requerir força temps, per la qual cosa alguns SGBD no suporten esquemes externs (tenen una arquitectura de dos nivells).

De qualsevol manera, sempre cal fer la transformació de sol·licituds entre els nivells conceptual i intern.

En els SGBD de tres nivells és necessari ampliar el *diccionari de dades* o catàleg de manera que, a més de contenir les definicions dels esquemes de la base de dades, inclogui informació sobre com s'ha d'establir la correspondència entre els tres nivells. L'SGBD utilitza programari addicional per a fer aquestes correspondències fent referència a la informació de correspondència que es troba en el *diccionari*, que també pot contenir algun control d'estadístiques d'ús i d'accessos.

Transformació de dades entre esquemes

Les fases de transformació de dades d'un esquema a un altre, en el cas de fer una consulta en una base de dades relacional, són les següents:

1) Sol·licitud de l'usuari que vol veure unes dades determinades d'una taula i creació d'una consulta.

2) Verificació de l'esquema extern per a aquest usuari.

3) Acceptació de l'esquema extern.

4) Transformació de la sol·licitud al nivell conceptual.

5) Verificació de l'esquema conceptual.

6) Acceptació de l'esquema conceptual. Es manipulen les taules (entitats) que contenen les dades sol·licitades per la consulta.

7) Transformació de la sol·licitud al nivell intern. Es busca en els espais de taula (objectes lògics formats per un arxiu o més d'un) que contenen la informació sobre l'emmagatzematge físic de les dades sobre els registres seleccionats en cada taula.

8) Selecció de la taula objecte de la consulta.

9) Execució de la consulta.

10) Transformació del nivell intern al nivell conceptual.

11) Transformació del nivell conceptual al nivell extern. Cal modificar el format de la informació extreta per tal que coincideixi amb la vista externa de l'usuari.

12) Es mostren a l'usuari els registres corresponents.

En l'explicació no es fa referència al *diccionari de dades*, per bé que és l'encarregat de les transformacions que s'han de produir entre els diferents esquemes.

L'SGBD ha d'assegurar que els tres **nivells siguin independents entre si**, és a dir, que els canvis introduïts en qualsevol d'ells no afecti els nivells superiors.

Amb l'arquitectura de nivells, els programes amb què l'usuari accedeix a la base de dades a través d'un esquema extern seran totalment independents dels canvis següents:

- Els canvis introduïts en l'esquema **lògic** relatius a les dades no incloses en el seu esquema **extern**.

- Els canvis introduïts en l'esquema **físic** relatius a la implementació de les estructures de dades de l'esquema **lògic**.

11.2.8. Independència de les dades

La descripció de les dades en tres nivells d'abstracció permet explicar la independència de les dades, que és un dels objectius principals de les bases de dades.

El concepte *independència de dades* es pot definir com la capacitat per a modificar l'esquema en un nivell del sistema de base de dades sense haver de modificar l'esquema del nivell immediatament superior.

Es poden definir dos tipus d'independència de dades: la independència física i la independència lògica de les dades.

1) Independència física de les dades

És la capacitat de modificar l'esquema intern sense haver d'alterar l'esquema conceptual (ni els esquemes externs).

Hi ha independència física quan els canvis en l'organització física de la base de dades no afecten l'esquema conceptual (ni els programes d'aplicació ni l'usuari).

Pot ser necessari modificar l'esquema intern per tal de millorar el rendiment de les operacions de recuperació i actualització de dades.

Modificació de l'esquema intern

La major part de canvis en l'esquema intern comporten refer la base de dades real emmagatzemada. Això pot requerir accions de **reorganització física** com les següents:

- Crear estructures d'accés addicionals canviant el mètode d'accés a uns registres determinats.
- Modificar el format o la codificació de dades concretes.
- Reorganitzar fitxers físics determinats movent dades d'un suport a un altre.

Si hi ha independència física, l'única cosa que varia en modificar l'esquema intern són les *correspondències* entre l'esquema conceptual i l'intern.

2) Independència lògica de les dades

És la capacitat de modificar l'esquema conceptual sense haver de modificar els esquemes externs ni els programes d'aplicació, sempre que no s'eliminin de l'esquema conceptual objectes requerits en el nivell extern.

Efectes dels canvis en els esquemes conceptual i extern

En un sentit més ampli, es considera que hi ha independència lògica quan els usuaris finals i els programes d'aplicació no es veuen afectats pels canvis en els nivells conceptual i extern (que corresponen al nivell lògic en l'arquitectura clàssica de dos nivells de bases de dades). Això vol dir el següent:

1) Els **canvis en l'esquema conceptual**...

a) poden afectar els **esquemes externs** que utilitzin els elements modificats (entitats o atributs).

b) no afecten els esquemes externs que no facin referència o no utilitzin els elements modificats.

2) Els **canvis en un esquema extern**...

a) poden afectar els **usuaris** que utilitzen els elements modificats.

b) no afecten la resta d'usuaris (ni l'esquema conceptual ni, en conseqüència, l'esquema intern).

Exemple

Si s'elimina l'atribut *cognom* en l'esquema conceptual, caldrà modificar l'esquema extern en què s'hagi definit *nom complet* com la concatenació de *nom* i *cognom*.

No es veuen afectats els esquemes externs (ni els usuaris) que no facin referència a aquest atribut.

Pot ser necessari modificar l'esquema conceptual per tal d'ampliar o reduir la base de dades, afegint o eliminant entitats o atributs o, fins i tot, canviant-ne les característiques. Això rep el nom de **reorganització lògica** de la base de dades.

Si l'SGBD disposa d'independència lògica, només serà necessari modificar la definició dels esquemes externs pertinents i les **correspondències**. Després de la reorganització lògica de l'esquema conceptual, els programes d'aplicació que facin referència a elements de l'esquema extern hauran de funcionar com ho feien abans. A més, les restriccions es podran modificar sense que afectin els esquemes externs ni els programes d'aplicació.

Els esquemes externs poden canviar a causa de la incorporació al domini del problema de noves necessitats funcionals o operatives per a l'organització. De fet, els SGBD actuals donen força independència lògica, però no prou, ja que les exigències de canvis constants en els sistemes informàtics demanen graus de flexibilitat molt elevats (és clar que els sistemes de fitxers tradicionals, en canvi, no presenten independència lògica).

Com s'ha dit, la independència de dades s'aconsegueix quan, en modificar l'esquema en algun nivell, l'esquema del nivell immediatament superior roman sense canvis; només es modifica la **correspondència** entre els dos nivells. Per tant, no cal modificar els programes d'aplicació que fan referència a l'esquema del nivell superior.

Independència de dades en els SGBD del mercat

L'arquitectura de tres nivells pot facilitar la construcció de la veritable independència de dades, tant física com lògica. Això no obstant, les **correspondències** poden requerir força temps, ja que impliquen una despesa addicional durant la compilació i l'execució d'una consulta o d'un programa, cosa que redueix l'eficiència de l'SGBD. Per aquest motiu són pocs els SGBD que han implementat completament l'arquitectura de tres nivells. Alguns, com els que gestionen bases de dades petites, no suporten vistes externes.

Perquè hi hagi una independència de les dades, els tres nivells d'abstracció han de ser completament independents.

Aquest fet, que no és del tot possible en la major part de bases de dades, es pot aconseguir si:

- L'esquema intern no és una *traducció* que depèn de l'esquema conceptual. Un mateix esquema conceptual es pot representar físicament de maneres diferents. Els requisits funcionals i de rendiment determinaran aquesta representació.

- Encara que els esquemes externs depenen de l'esquema conceptual pel fet que els objectes del nivell extern que manipula l'usuari (entitats, atributs, registres…) han d'estar formats per elements de dades representats en el nivell conceptual, l'estructura d'aquests objectes (nombre d'elements, disposició…) ha de ser independent de com aquests elements s'han representat en el nivell conceptual i de les relacions que hi mantenen. El mateix es pot dir entre el nivell intern i el conceptual.

3) Independència de dades i lligadura

Per tal de garantir la integritat de la base de dades cal que, en algun moment, el procés de manipulació d'un nivell d'esquema determinat tingui en compte com es representa la informació en els altres nivells (o esquemes). En aquest moment els diferents esquemes es vinculen entre si i, per tant, es perd la independència entre ells. Aquest procés de vinculació o establiment de les correspondències entre esquemes executat per l'SGBD es coneix amb el nom de *lligadura*.

La **lligadura** és la transformació d'una operació descrita en termes d'un esquema extern en una altra operació descrita en termes de l'esquema intern.

Aquesta transformació és necessària perquè l'SGBD necessita convertir una dada de l'esquema extern en un agregat o element de dades d'un registre de l'esquema intern. Donat que aquesta transformació passa per l'esquema conceptual, la lligadura es pot desglossar en dues:

- **Lligadura conceptual** (transformació entre l'esquema extern i el conceptual)

- **Lligadura física** (transformació entre l'esquema conceptual i l'intern)

Quan es produeix la lligadura, es vinculen els diferents esquemes. Això provoca la pèrdua de la independència de dades, ja que l'esquema extern ha estat traduït en termes de l'esquema intern. Així, resulta que, si la lligadura s'ha produït en compilar el programa d'aplicació, quan es produeixi un canvi en l'esquema conceptual o intern caldrà tornar a compilar-lo. Per tant, és convenient retardar tant com es pugui el procés de vinculació.

Compilar

És l'acció de traduir el codi font d'un programa (escrit en un llenguatge de programació d'alt nivell) a llenguatge màquina (codi objecte) executable per l'ordinador.

El procés de vinculació o lligadura entre els diferents esquemes es pot produir en qualsevol de les fases d'un programa d'aplicació (ordenades per ordre d'execució en el procés):

– En la **compilació** o en un pas de precompilació.

– En el **muntatge** per a generar el mòdul executable del programa.

– En l'**arrencada** d'un programa (aquesta fase correspon a l'inici de l'**execució** del programa o, més concretament, abans que el programa comenci a sol·licitar accessos a les dades).

– En cada **accés** a les dades.

Com més tard es produeixi la lligadura menys modificacions caldrà fer en els programes i, per tant, més gran serà la independència de dades (malgrat que la implementació de l'SGBD serà més complexa).

D'altra banda, donat que la lligadura pot produir una despesa notable de temps, el funcionament de les aplicacions que utilitzen la base de dades serà menys eficient com més freqüent sigui la lligadura.

Lligadures estàtica i dinàmica

Els sistemes que fan la lligadura en la fase **d'accés a les dades** mantenen la independència fins a l'últim moment i només vinculen les dades a les quals es pot accedir i que estan implicades en cadascun dels accessos. En canvi, els sistemes que fan la lligadura en la fase de **compilació** perden la independència de les dades en el moment en què els programes d'aplicació són traduïts a codi objecte.

- **Lligadura estàtica**: si la lligadura es produeix en una **fase primerenca** (de compilació o de muntatge), això implica que els programes d'aplicació hauran de ser tornats a compilar cada vegada que es produeixi una modificació dels esquemes conceptual i físic, tot i que, d'altra banda, el rendiment d'aquests serà alt perquè el temps que comporta la lligadura es consumeix una única vegada (i no en l'arrencada del programa o en cada accés a les dades).

- **Lligadura dinàmica**: si la lligadura es produeix en una **fase tardana** (d'execució o en cada accés a les dades) es podran modificar els esquemes conceptual i físic sense que els programes d'aplicació s'hagin de traduir de nou a codi màquina (compilar). Malgrat tot, el rendiment d'aquests programes serà menor perquè necessiten una despesa de recursos addicional per a dur a terme el procés de lligadura (o bé en l'arracada del programa o bé en cada accés a les dades).

La solució de compromís adoptada per molts SGBD és que el procés de lligadura es produeixi en la fase d'execució. Això suposa una despesa de recursos addicional en l'arrencada del programa d'aplicació, però evita una compilació reiterada i contínua de les aplicacions cada vegada que s'alteren les representacions de les dades.

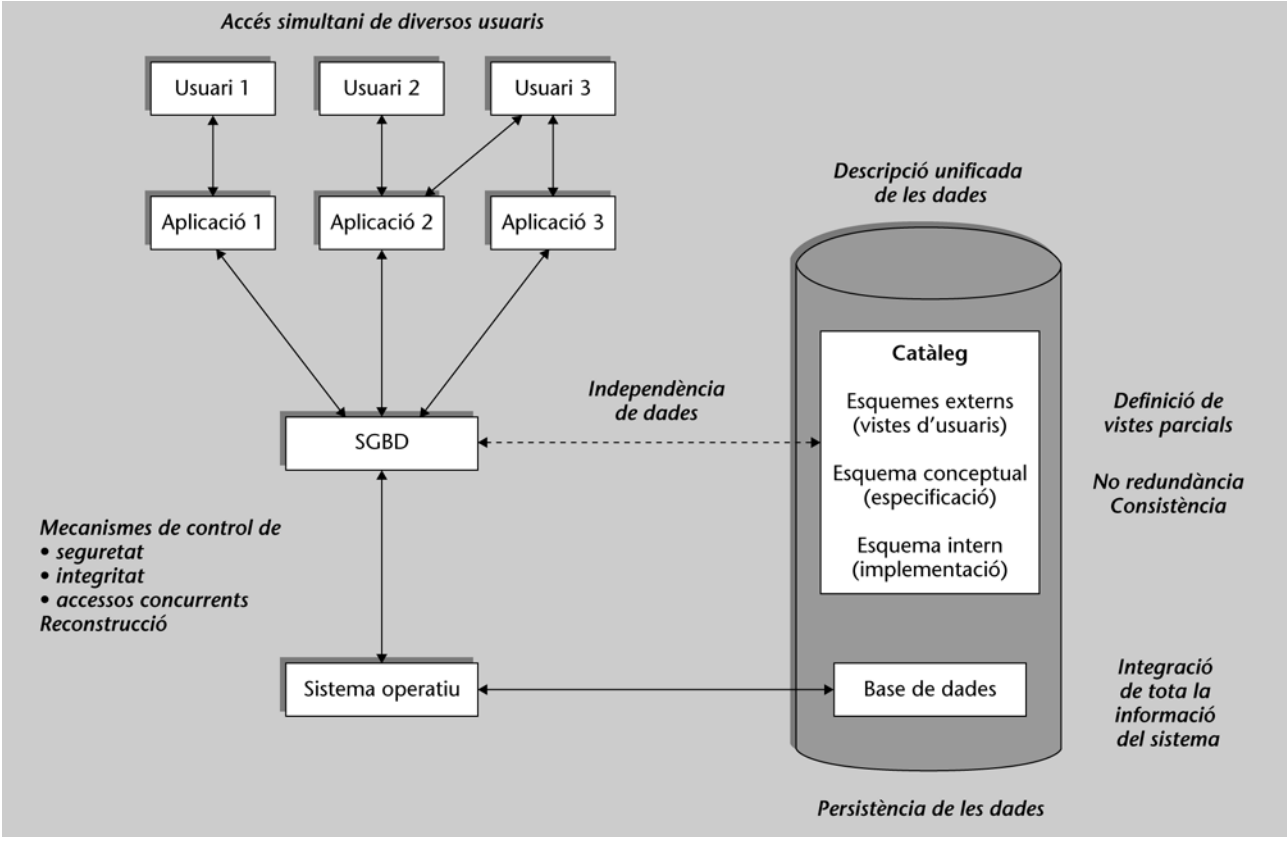
12. Estructura global d'un sistema de base de dades

Sintetitzant la matèria exposada prèviament, es dóna una visió gràfica integrada de les característiques i els elements d'un sistema de base de dades entès com el sistema que integra l'SGBD i la base de dades en un sentit ampli: les **dades** (base de dades), les seves **definicions** (esquemes), el **programari** de manipulació (SGBD i aplicacions), els **usuaris**, etc.

Els elements i les característiques

Els elements i les característiques d'un sistema de base de dades s'exposen respectivament en els apartats "Elements d'un sistema de base de dades" i "Objectius i característiques de les bases de dades" d'aquest mòdul.

Figura 17. Sistema de base de dades



Elements i característiques d'un sistema de base de dades. En aquest mapa conceptual, els **elements** del sistema es mostren enquadrats i les *característiques*, en cursiva. Les fletxes contínues representen ordres i flux de dades. La línia discontínua simbolitza consultes de l'SGBD al catàleg en què es troben les definicions del sistema.

L'SGBD, amb l'ajuda del sistema operatiu, maneja les sol·licituds de l'usuari per a dur a terme les accions de base de dades i permet complir els requisits de control de seguretat, integritat, accés concurrent a les dades.

Es pot observar que l'SGBD introdueix un nivell d'independència nou entre l'usuari i les dades i constitueix una interfície o capa sobre el sistema operatiu subjacent (que no existia en els sistemes de gestió de dades basats en fitxers).

La separació entre programes i dades

Es troba en el contingut complementari referit a la independència respecte a la representació física de les dades del subapartat "Objectius i característiques de les bases de dades".

En el processament de fitxers tradicionals, els programes d'usuari accedien a les dades a través dels mètodes d'accés a fitxers del sistema operatiu, la qual cosa feia que hi hagués una estreta dependència entre els programes i les dades.

13. Emmagatzematge de bases de dades

Aquest apartat tracta sobre l'organització de les bases de dades en el suport d'emmagatzematge.

Les bases de dades solen emmagatzemar grans volums de dades que han de *perdurar* durant períodes de temps llargs en què es processen les dades i s'hi accedeix molt sovint (a diferència de les estructures de dades *transitòries* que persisteixen un temps limitat durant l'execució d'un programa).

Per aquest motiu, les dades d'una base de dades s'han d'emmagatzemar en un mitjà d'emmagatzematge informàtic, de manera que l'SGBD pugui recuperar, actualitzar i processar les dades quan sigui necessari.

Els mitjans d'emmagatzematge de l'ordinador poden ser de dos tipus:

1) Emmagatzematge primari: inclou la memòria principal i la memòria cau sobre les quals el processador o unitat central de processament (PCU) de l'ordinador pot operar directament. Tenen un accés ràpid a les dades però la seva capacitat d'emmagatzematge és limitada i són més cares.

2) Emmagatzematge secundari: són dispositius d'emmagatzematge extern amb més capacitat i més barats, però amb un accés més lent a les dades. El processador no pot processar directament les dades en emmagatzematge secundari; cal copiar-les en un mitjà d'emmagatzematge primari.

Les raons per les quals la major part de bases de dades s'emmagatzemen de manera *persistent* en discos magnètics (com el disc dur) o altres **mitjans d'emmagatzematge secundari** (i no en mitjans primaris) són les següents:

a) Pel seu elevat **volum de dades**, les bases de dades grans no caben senceres en la memòria principal.

b) Les **pèrdues permanents** de dades es presenten menys sovint que en l'emmagatzematge primari, que és volàtil.

c) El **cost d'emmagatzematge** per unitat de dades és menor.

Les aplicacions de bases de dades normalment requereixen localitzar, carregar en memòria principal i processar només una *part petita de la base de dades* en un moment donat. Després, si hi ha hagut alguna modificació, l'escriuen una altra vegada en el disc.

Les bases de dades s'emmagatzemen físicament en el disc com a **fitxers de registres**; així, els registres es poden localitzar de manera eficient quan cal.

Cada **registre** consta d'un conjunt de **valors** o elements de dades relacionats, en què cada valor està format per almenys un byte i correspon a un **camp** determinat del registre. En general, els registres solen ser homogenis (del mateix tipus) i descriuen entitats i els seus atributs. Una col·lecció de noms de camps i els seus tipus de dades corresponents constitueix una definició de **tipus de registre** o **format de registre**. El **tipus de dades** associat a cada camp especifica el tipus de valors que aquest pot prendre.

Un fitxer pot estar constituït per registres dels tipus següents:

- **Registres de longitud fixa:** si tots els registres del fitxer tenen exactament la mateixa mida (en bytes).

- **Registres de longitud variable:** si alguns registres del fitxer tenen mides diferents. Això pot ser pel fet que hi hagi camps de longitud variable, camps repetitius (que es repeteixen un nombre variable de vegades) o camps opcionals, o que es tracti d'un fitxer mixt amb registres de tipus diferents.

  Hi ha diverses maneres d'especificar la longitud del registre:

  – El sistema reserva un **camp de control** a l'inici de cada registre per a indicar-ne la longitud.

  – El sistema inclou un **caràcter especial de separació** (o caràcter delimitador de control) al final de cada registre. S'anomenen **registres delimitats.**

  – El **programa d'aplicació** s'encarrega de localitzar el principi i el final de cada registre. S'anomenen **registres indefinits.**

Camps de longitud variable

Hi ha dues maneres d'especificar la longitud d'un camp:

- Al final del camp s'inclou un **caràcter separador especial** que no aparegui en cap valor de camp (per exemple, els caràcters ?, % o $).
- Davant del valor del camp es desa la seva **longitud en bytes**.

Avui dia, se solen utilitzar **discos magnètics** per a emmagatzemar les bases de dades de manera persistent. En un disc magnètic, la memòria es divideix en blocs de longitud fixa, que es defineixen quan es dóna format al disc.

El **bloc** és la unitat de transferència de dades entre el disc i la memòria principal. És a dir, el bloc és la quantitat de dades (en bytes) que el sistema escriu o llegeix d'una vegada en una única operació física d'entrada o sortida.

Concretament, la transferència la fa l'**administrador d'entrada i sortida** (E/S) del sistema operatiu, que transfereix el bloc llegit en el disc a la memòria intermèdia de la memòria principal (àrees contigües de memòria amb capacitat per a emmagatzemar temporalment diversos blocs de dades, que en anglès s'anomenen *buffers*).

El **registre** és la unitat de transferència entre la memòria intermèdia i el programa d'usuari.

Els registres s'estructuren en un fitxer seguint alguna de les organitzacions de fitxers disponibles en el sistema. Cada tipus d'organització implica una distribució física particular dels registres en el disc i un tipus d'accés (seqüencial o directe) a aquests. Aquestes organitzacions de fitxers són les que s'utilitzen per a implementar les **estructures de dades** de la base de dades, segons com descriu l'**esquema físic**.

Les estructures de dades

Les estructures de dades utilitzades en els sistemes d'informació es tracten en l'apartat "Estructures de dades" d'aquest mateix mòdul.

Lectura i escriptura de registres i blocs

En un sistema monousuari és freqüent que s'executin simultàniament molts processos. Cada procés pot treballar amb més d'un fitxer de dades i li pot convenir tenir uns quants blocs en la memòria intermèdia. Però la mida dels blocs està condicionada per l'espai disponible per al conjunt de memòries intermèdies en la memòria principal.

Un bloc pot tenir diversos registres i un registre pot ocupar diversos blocs. Com que la mida d'un registre sol ser molt menor que la d'un bloc, els registres s'agrupen en blocs. El bloc mínim és un sector, però s'acostuma a llegir de cop tota una sèrie de sectors.

En algunes ocasions, el bloc s'anomena **registre físic**, i **registre lògic** el que aquí s'anomena *registre*. D'altra banda, en l'entorn de bases de dades habitualment es fa servir el terme **pàgina** per a referir-se al bloc.

blocs de dades → fitxers de registres → estructures de base de dades

Aquesta jerarquia d'abstracció de dades (blocs de dades – fitxers de registres – estructures de base de dades) existent en un sistema de base de dades permet explicar l'**esquema d'accés de l'SGBD a les dades** per a satisfer un requisit de consulta que es tracta en l'apartat següent.

L'accés de l'SGBD a les dades

L'accés de l'SGBD a les dades per a satisfer una consulta s'esquematitza en la figura de l'apartat següent, "Accés de l'SGBD a les dades".

14. Accés de l'SGBD a les dades

L'SGBD és el conjunt de **mòduls de programari** per a accedir a les dades que permet localitzar un element d'informació específic (un registre) en la base de dades i presentar-lo a l'usuari.

Per tal de comprendre el seu funcionament, tot seguit s'examinen els passos implicats en el procés d'execució d'una consulta de les dades d'un registre feta per un programa d'usuari a l'SGBD, i que aquest duu a terme amb l'ajuda del sistema operatiu.

Llenguatges de consulta

La consulta també es pot fer mitjançant un llenguatge de base de dades (per exemple, SQL) tal com es descriu en el subapartat "Llenguatges de base de dades" dins l'apartat "Funcions i components de l'SGBD".

Malgrat que hi pot haver diferències importants en els detalls de funcionament (i terminologia) entre els diferents SGBD, tots ells solen seguir unes línies generals bastant comunes que es poden enunciar conceptualment de la manera següent:

Els usuaris accedeixen a les dades a través de peticions a l'SGBD que consisteixen en operacions sobre les estructures de dades d'un esquema extern determinat.

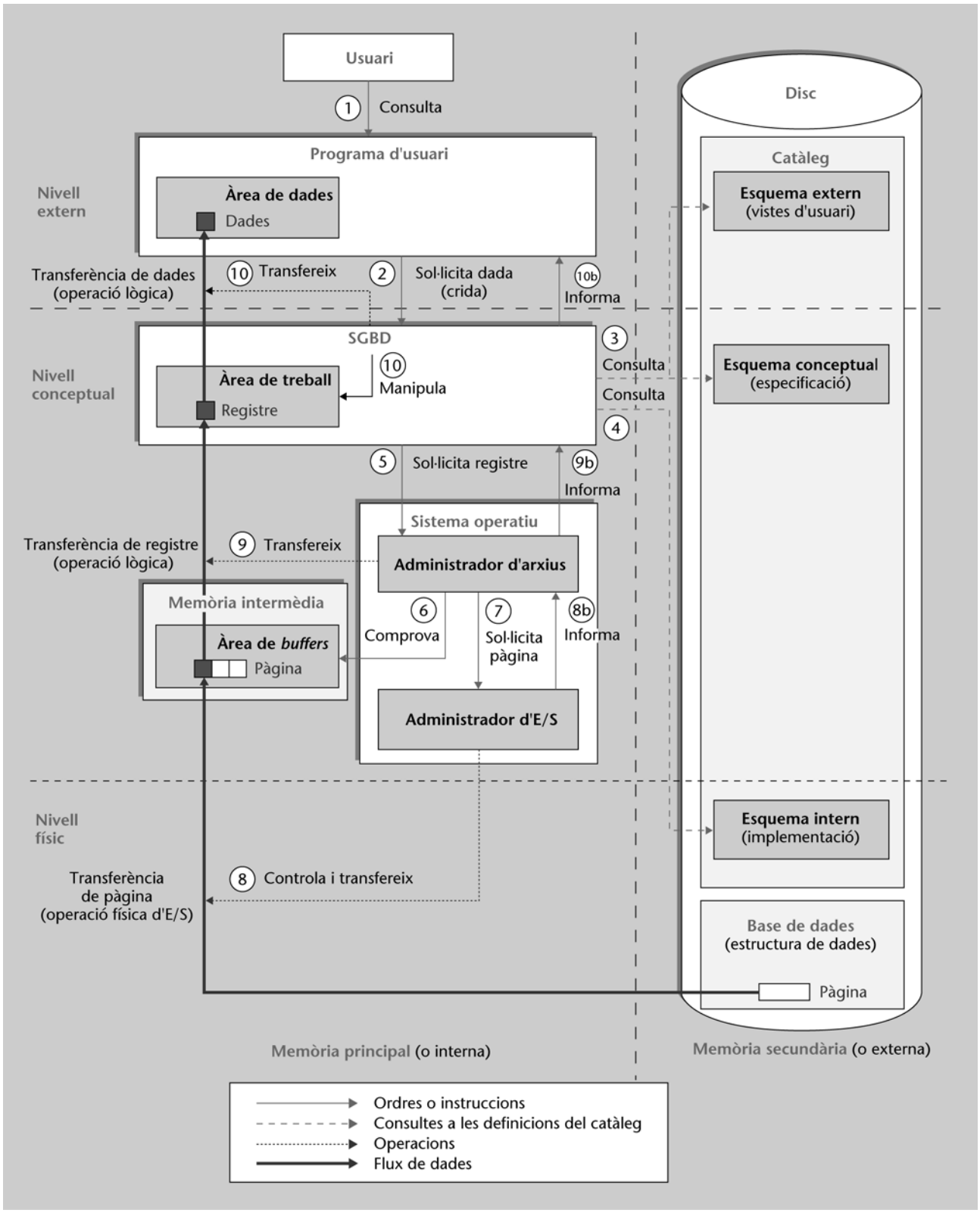
L'SGBD, amb el coneixement que té dels esquemes extern, conceptual i físic, tradueix aquestes peticions en operacions sobre els fitxers en els quals s'implementa la base de dades i n'encarrega l'execució al sistema operatiu.

Aquest llegeix pàgines del suport físic en què s'emmagatzema la base de dades i les transfereix a l'àrea de memòria intermèdia en la memòria principal.

Finalment, l'SGBD passa registres des de la memòria intermèdia a l'àrea de treball del programa d'usuari. (Cada etapa pot requerir conversions de tipus de dades o altres.)

La figura següent representa l'esquema general del **flux de dades i de control** que segueix el procés entre el programa d'usuari que fa la consulta, l'SGBD, el sistema operatiu subjacent i la base de dades.

Figura 18. Esquema de la gestió d'accés a les dades feta per l'SGBD



El flux de dades es pot produir en sentit contrari al que es mostra en la figura. Aquest és el cas de l'execució d'una operació d'escriptura de dades en el disc (en comptes d'una lectura o consulta com es mostra en l'esquema).

En la figura es mostren els **tres nivells** d'abstracció de l'arquitectura ANSI/SPARC en què es desenvolupa el procés: **físic**, **conceptual i extern**. (Per exemple, el programa d'usuari intervé a nivell lògic –més concretament, a nivell extern– i les operacions que executa són lectures de registres lògics). Tanmateix, s'aprecia que la base de dades i el catàleg s'emmagatzemen en la **memòria ex-**

terna (disc) i els diferents mòduls de programari es carreguen en la **memòria principal**, en la qual també resideixen els magatzems de memòria intermèdia.

Els passos indicats en la figura són els següents:

1) L'usuari fa una **consulta** utilitzant un programa d'usuari o aplicació.

2) Això genera una **crida** a l'SGBD en la qual s'envia l'operació de consulta de dades.

3) L'SGBD interpreta la sol·licitud rebuda i l'analitza (verifica si la sintaxi de l'operació és correcta, si l'usuari està autoritzat a fer-la, etc.). Per a fer-ho, es basa en l'**esquema extern** amb què treballa el programa d'usuari i en l'**esquema conceptual** (fent la correspondència extern-conceptual associada). Si la consulta no és vàlida es passa al pas 10b, que posa fi al procés.

4) Si la consulta és vàlida, l'SGBD, després de fer la correspondència conceptual-intern, examina l'**esquema intern** i tradueix la consulta de l'usuari en una **operació de lectura** sobre un fitxer (i determina quin mecanisme físic s'ha de seguir).

5) L'SGBD, una vegada determinat quin **registre** es necessita, indica el fitxer en el qual s'emmagatzema i sol·licita a l'administrador d'arxius que extregui aquest registre.

L'administrador d'arxius

En alguns sistemes l'administrador d'arxius és un component del sistema operatiu subjacent; en altres, es troba inclòs en l'SGBD.

Les operacions que pot dur a terme l'administrador d'arxius són llegir, modificar, afegir o eliminar registres d'un arxiu, i crear o destruir arxius. Aquestes operacions primitives permeten que l'SGBD construeixi i manipuli les estructures d'emmagatzematge de dades.

En la pràctica, la consulta de l'usuari (pas 1) pot exigir la recuperació d'un **conjunt de registres**. En aquest cas, es repeteixen els passos 5 a 9 per a cadascun dels registres, que, una vegada recuperats, són examinats per l'SGBD en la memòria principal per a trobar el registre requerit. Tot seguit, s'executa el pas 10.

6) **L'administrador d'arxius** comprova si la **pàgina** en què es troba el registre requerit és, casualment en aquell moment, en l'àrea de memòria intermèdia de la memòria principal com a resultat d'una lectura anterior. En aquest cas, no cal accedir al disc per a llegir la pàgina de nou (i, per tant, es poden obviar els passos 7 i 8 i passar directament al pas 9).

La memòria intermèdia o àrea de memòria intermèdia (*buffers*)

Aquesta àrea de memòria és molt important per a augmentar la velocitat de procés de l'SGBD. S'hi desen temporalment les dades a les quals l'SGBD ha accedit recentment, ja que hi ha una alta probabilitat que calgui tornar a accedir-hi. Això permet estalviar un gran nombre d'operacions físiques d'entrada/sortida que comporten més despesa de temps que els accessos a la memòria principal, en la qual resideix la memòria intermèdia.

7) Si la **pàgina** sol·licitada no és en l'àrea de memòria intermèdia, l'administrador d'arxius en demana la lectura a l'administrador d'E/S del disc del sistema operatiu subjacent (aquest mòdul, en alguns sistemes, rep el nom de component de serveis bàsics d'E/S).

En definitiva, l'administrador d'arxius (i, finalment, l'SGBD) sol·licita operacions d'E/S de pàgines i gestiona l'espai dedicat a la memòria intermèdia.

8) L'administrador d'E/S determina la localització física en el disc de la pàgina desitjada, executa l'operació física d'E/S necessària (en aquest cas, la seva recuperació) i transfereix la pàgina des del disc a l'àrea de memòria intermèdia.

9) L'SGBD (amb l'ajuda de l'administrador d'arxius) extreu el **registre** sol·licitat entre els diferents registres que la pàgina pot contenir i el diposita en l'**àrea de treball**. L'SGBD interpreta la codificació de registre segons el que indica l'esquema intern.

10) L'SGBD, comparant l'esquema conceptual i l'esquema extern, duu a terme les transformacions eventuals que implica aquest últim (inverses a les fetes en el pas 3) per a determinar les **dades** requerides, que transfereix a l'**àrea de dades** del programa d'usuari que ha fet la consulta original. Finalment, l'SGBD retorna el control (10b) al programa i dóna per acabada l'execució de la consulta.

Els passos 8b i 9b, indicats en la figura permeten que els mòduls corresponents informin de les accions executades per tal de tornar el control als mòduls que els han precedit en el procés. Els passos 3, 4 i 10 impliquen processos interns de manipulació de registres per part de l'SGBD que no queden reflectits en la figura. D'altra banda, cal remarcar que en la descripció s'ha suposat que la lligadura es produeix en cada accés a la base de dades.

Els llenguatges de bases de dades

Els llenguatges de bases de dades i l'especificació dels processos interns de l'SGBD es discuteixen amb detall en el proper apartat, "Funcions i components de l'SGBD".

El concepte *lligadura*

Es descriu en tractar la independència de dades en el subapartat "Arquitectura de nivells" dins l'apartat "Arquitectura dels SGBD".

Compilació de les sol·licituds d'accés a les dades

Òbviament, l'esquema d'accés a les dades mostrat en la figura és una descripció simplificada i pot induir a pensar que tot el procés és interpretatiu, ja que suggereix que els processos d'analitzar la sol·licitud, inspeccionar els diversos esquemes, etc. es fan en el moment de l'execució.

La **interpretació** implica gairebé sempre un baix rendiment (a causa de l'augment del temps d'execució). En la pràctica, però, és possible **compilar** les sol·licituds d'accés abans del moment de l'execució (fent servir un llenguatge de base de dades hostatjat).

En termes generals, es pot concloure el següent:

- El **programa d'usuari** percep la base de dades com un **conjunt de dades**.
- L'**SGBD** percep la base de dades com un **conjunt de registres** emmagatzemats.
- L'**administrador d'arxius** percep la base de dades com un **conjunt de pàgines** (blocs).
- L'**administrador d'E/S** percep la base de dades tal com és físicament **en realitat**.

15. Funcions i components de l'SGBD

Els SGBD es caracteritzen per permetre la **descripció unificada de les dades** i la **definició de vistes parcials** d'aquestes per a diferents usuaris. Per tal de satisfer els objectius de les bases de dades cal exigir a l'SGBD que asseguri el manteniment de les **propietats** de la base de dades: la **independència**, la **integritat** i la **seguretat** de les dades.

A partir dels **objectius** (i de les característiques) de les bases de dades, es poden deduir les **funcions** d'un SGBD. Basant-se en l'anàlisi d'aquestes funcions, es pot saber quins han de ser els seus **components** bàsics.

Els objectius i les característiques

Els objectius i les característiques de les bases de dades s'exposen en l'apartat "Objectius i característiques de les bases de dades" d'aquest mòdul.

15.1. Funcions de l'SGBD

L'SGBD ha de proporcionar els mitjans necessaris per a dur a terme les tasques següents:

- **Definició** de l'esquema de la base de dades.
- Gestió de l'**organització física** de les dades: ubicació, organització, agrupació i estructures de recuperació de dades per fallides del sistema.
- Gestió d'**accés a les dades** des dels dispositius d'emmagatzematge. Aquestes tasques d'emmagatzematge i recuperació de la informació es poden executar des del sistema operatiu, des de les utilitats de la base de dades o mitjançant algun llenguatge d'alt nivell (de quarta generació, 4GL) com SQL.
- **Interrogació i resposta a les peticions de l'usuari**: el sistema interpreta, analitza i envia les peticions de l'usuari per a la seva execució; finalment, presenta la informació de resposta recuperada.

Si es relacionen aquestes tasques amb les operacions que afecten els arxius i els seus registres, es pot deduir que, d'acord amb la seva definició, l'SGBD està format per **components** que li permeten complir tres tipus de **funcions**: descripció, manipulació i control de la base de dades. Aquestes funcions, però, no es corresponen de manera lineal amb els diferents nivells d'abstracció.

15.1.1. Funció de definició de dades

Permet especificar l'esquema o disseny d'una base de dades mitjançant la definició dels objectes que la constitueixen, la seva estructura, les interrelacions

existents entre ells, les regles d'integritat semàntica, així com també les característiques físiques.

Els tres nivells d'abstracció intervenen en aquesta funció de la manera següent:

- A nivell físic es defineixen les condicions físiques de les estructures de dades: espai en disc, mida de les pàgines (blocs), mida de l'espai de taules, etc. Això es fa a través del *llenguatge de definició d'emmagatzematge* (*LDE*).

- A nivell conceptual es creen els objectes per als quals s'ha reservat l'espai i s'han definit les característiques físiques. Això es fa mitjançant interfícies gràfiques o a través del *llenguatge de definició de dades* (*LDD*).

- A nivell extern és possible crear i eliminar perfils d'usuari, definir rols i assignar els seus permisos.

Altres tasques incloses en aquesta funció són les relatives al catàleg (o diccionari de dades), que és un component especial de l'SGBD en què s'emmagatzemen els esquemes de definició de dades i que permet la transformació de les instruccions que comuniquen uns nivells d'abstracció amb uns altres.

15.1.2. Funció de manipulació de dades

Permet executar accions sobre les dades emmagatzemades o afegir-ne de noves, treballant sobre la totalitat dels registres d'un arxiu o més d'un o sobre cada registre de manera individual, d'acord amb les normes de seguretat. Les accions bàsiques són:

- La **recuperació** (o consulta), que comporta la lectura de dades.

- L'**actualització** (o manteniment), que comporta l'escriptura de dades. Per exemple, la inserció de dades noves, la modificació de dades existents o la seva eliminació.

Per tal d'executar aquesta funció, l'administrador i el programador utilitzen el *llenguatge de manipulació de dades (LMD)*, encara que, a nivell extern, es poden utilitzar menús, interfícies gràfiques i transaccions prèviament definides que faciliten la utilització a l'usuari.

15.1.3. Funció de control de dades

No és una funció ben definida i sovint es considera inclosa en les dues funcions anteriors.

Si s'accepta la seva identitat pròpia, es pot desdoblar en dos tipus de funcions:

- Funcions referents a la **gestió d'usuaris**: formades per utilitats i **interfícies** que els permeten accedir, segons els seus permisos, als diferents elements del sistema. Aquestes funcions de definició s'inclouen aquí perquè l'administrador pot crear perfils, modificar i eliminar permisos.

- Funcions relatives a l'**administració de la base de dades**: monitorització del **funcionament** (estadístiques, capacitats, espais utilitzats, fragmentació); control de la **seguretat** (gestió d'accessos –privadesa–, gestió de connexió a xarxes, gestió de transaccions, gestió de còpies de recuperació, etc.); manteniment de la **integritat** (respecte a les dades en si mateixes, els seus valors i les seves relacions), etc.

Aquestes funcions se solen dur a terme mitjançant la utilització d'interfícies gràfiques implementades en els SGBD i el *llenguatge de control de dades* (LCD).

Objectius de les bases de dades i funcions i components de l'SGBD

En la taula següent es presenta un resum de la correlació entre els objectius de les bases de dades les funcions de l'SGBD i els seus components associats.

Objectius de les BD	Funcions de l'SGBD	Components de l'SGBD
Descriure les dades de manera unificada i independentment de les aplicacions	**Definició** de la base de dades en diversos nivells **d'esquemes**: • Conceptual (definició de les estructures de la base de dades) • Intern (implementació de les estructures de l'esquema conceptual) • Externs (definició d'estructures derivades) **Establiment de correspondències** entre esquemes*	**Llenguatges de definició** d'esquemes de la base de dades i traductors
Independitzar les aplicacions respecte a la representació física de les dades		
Definir vistes parcials de les dades per a diferents usuaris		
Gestionar les dades	**Manipulació** de les dades: consulta i actualització	**Llenguatges de manipulació** de les dades i traductors associats
	Administració de la base de dades	**Eines per a l'administració:** • Reestructuració • Simulació • Obtenció d'estadístiques • Impressió de dades
Mantenir la integritat i la seguretat de les dades	**Funcions de control** de: • Integritat semàntica • Seguretat • Reconstrucció de la base de dades en cas de fallides • Accessos concurrents	**Eines per al control**: • Control de la integritat • Control de seguretat • Reconstrucció • Control de concurrència

*Les funcions de definició d'esquemes i d'establiment de correspondències entre ells es tracten amb detall en l'apartat "Arquitectura dels SGBD".

15.2. Components de l'SGBD

L'SGBD és un sistema molt complex que es pot considerar format per un conjunt de programari dividit en dos grups diferenciats: el motor de base de dades i les interfícies, utilitats i eines.

15.2.1. Motor de base de dades

S'encarrega de les tasques de gestió de les dades: el procés de consultes, la gestió de transaccions, l'accés a les dades i la forma d'emmagatzematge. Per tant, accepta les consultes dels clients, les processa i retorna els resultats. Està format pels elements següents:

- **Llenguatges de base de dades.** L'usuari interpel·la l'SGBD a través d'un llenguatge de maneig de la base de dades (un exemple és SQL).

- **Diccionari de dades o catàleg.** Conté informació com els noms dels fitxers i els elements de dades, els detalls d'emmagatzematge de cada fitxer, la informació de correspondència entre els esquemes i les restriccions, a més d'altres tipus d'informació que necessiten consultar els mòduls de l'SGBD.

- **Nucli de l'SGBD**. És el conjunt complex de mòduls de programari que s'executen com a processos transparents per a l'usuari (que no es demanen explícitament). S'ocupa de les tasques bàsiques de la base de dades: control d'emmagatzematge, diàleg amb el sistema operatiu, maneig del diccionari de dades (o catàleg) i tasques necessàries per al treball amb el llenguatge de base de dades.

15.2.2. Interfícies, utilitats i eines

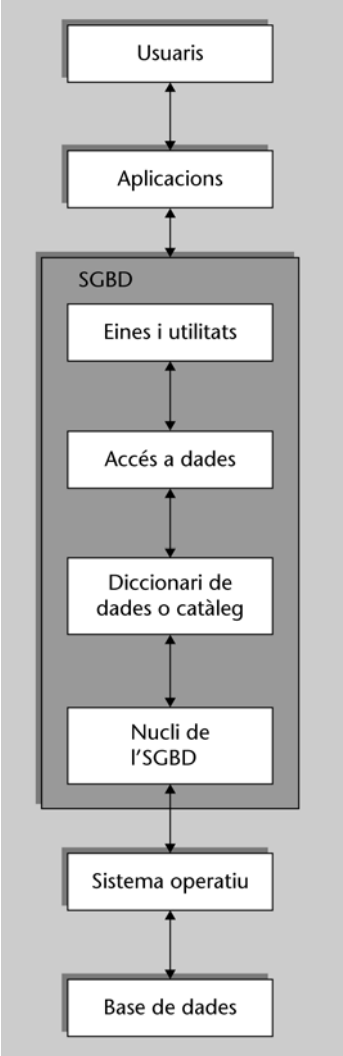
- **Interfícies**. L'SGBD ofereix interfícies apropiades a cada categoria d'usuari (ocasional, paramètric, administrador o programador d'aplicacions) com a ajuda perquè aquest especifiqui les seves sol·licituds (ja siguin, respectivament, consultes interactives, transaccions programades, sentències en LDD o instruccions d'un programa d'aplicació). Les interfícies gràfiques d'usuari (GUI) dels actuals SGBD permeten ometre l'ús del llenguatge de base de dades (la major part del temps) per part de l'usuari.

- **Utilitats i eines**. Un grup important de funcionalitats de l'SGBD és constituït per les eines i utilitats de valor afegit, que són aplicacions de programari integrades en el sistema per a simplificar les tasques de l'administrador, el programador i l'usuari final.

Si només s'instal·la el nucli, bàsicament es pot prestar el mateix servei d'emmagatzematge i recuperació d'informació que amb l'SGBD complet. La diferència rau en el fet que els diferents usuaris (administrador, programador i usuari final) han d'utilitzar eines més complexes i de nivell més baix.

En la figura següent es mostren els elements de l'SGBD integrat en un sistema de base de dades.

Figura 19. Elements d'un SGBD en un sistema de base de dades



El catàleg de l'SGBD i la base de dades se solen emmagatzemar en disc. D'altra banda, el sistema operatiu sol controlar l'accés al disc planificant l'entrada i sortida (E/S) del disc.

L'emmagatzematge en disc

L'emmagatzematge en disc de la base de dades i el catàleg de l'SGBD s'expliquen en els apartats "Emmagatzematge de bases de dades" i "Accés de l'SGBD a les dades" d'aquest mòdul.

15.3. Llenguatges de base de dades

Els llenguatges de base de dades permeten manejar la informació continguda en la base de dades mitjançant conjunts d'ordres o instruccions amb una sintaxi pròpia per a cadascun dels llenguatges (igual que un llenguatge de programació habitual).

Es poden analitzar i classificar segons criteris diversos:

- La seva **funcionalitat**.
- La **modalitat** d'ús o mode de treball.
- Les **aplicacions** que suportin.
- El **model de dades** en què es fonamenta la base de dades.
- Els tipus d'**usuaris** que accedeixen a la base de dades.

A continuació, es tracten aquests aspectes amb detall.

15.3.1. Llenguatges i funció

Els tipus de llenguatge es classifiquen en tres grups (o subllenguatges) pel que fa a la seva funció: llenguatges de definició, llenguatges de manipulació i llenguatges de control. Aquests llenguatges es poden subdividir en parts més petites per a diverses funcions especialitzades. Ara bé, el més freqüent és que el mateix llenguatge disposi de construccions per als tres tipus de funcions. Generalment, és un llenguatge simple basat en una gramàtica senzilla que disposa d'un conjunt limitat de morfemes (unitats mínimes de significació).

1) Definició d'esquemes

Després de completar el disseny d'una base de dades i d'escollir un SGBD per a la seva implementació, el dissenyador i l'administrador (o l'usuari que executa tasques d'administrador amb les dades de la seva propietat) han d'especificar els esquemes conceptual i intern de la base de dades i la correspondència existent entre ells.

En un SGBD amb arquitectura de tres nivells hi ha un **llenguatge de definició de dades** (LDD) que pot tenir dos subcomponents:

a) L'LDE, **llenguatge de definició d'emmagatzematge** (esquema intern).
b) L'LDV, **llenguatge de definició de vistes** (esquemes externs).

Seguidament, s'exposen els usos que tenen en els SGBD:

a) En els SGBD *que no mantenen una separació estricta de nivells* s'utilitza el **llenguatge de definició de dades** (LDD) per a definir, de manera no ambigua, els esquemes conceptual i intern.

Aquesta definició ha de ser compilada (pel **compilador d'LDD** que té l'SGBD) per a donar lloc a una representació orientada a l'ordinador que és el que utilitza l'SGBD en temps de processament. La funció del compilador és processar les sentències d'aquest llenguatge per a identificar les descripcions dels elements dels esquemes. La representació de les dades obtinguda en aquest procés de compilació s'emmagatzema en el *diccionari de dades* o *catàleg* de l'SGBD.

Les ordres principals que cal destacar permeten crear o eliminar objectes de la base de dades, definir restriccions i regles d'integritat semàntica.

Exemples d'instruccions de definició

En el llenguatge SQL, les ordres per a crear objectes de la bases de dades, eliminar-los, definir restriccions i definir regles d'integritat semàntica corresponen, respectivament, a CREATE, DROP, CHECK i CONSTRAIN.

b) En els SGBD *que mantenen una separació clara entre els nivells conceptual i intern*, l'LDD serveix per a especificar únicament l'esquema conceptual. Per a especificar l'esquema intern s'utilitza el **llenguatge de definició d'emmagatzematge (LDE)** que sovint s'incorpora en el mateix LDD, de manera que se'l considera un sub-

component d'aquest. La correspondència entre els dos esquemes es pot especificar en qualsevol dels dos llenguatges.

c) En una veritable arquitectura de tres nivells seria necessari un tercer llenguatge, el **llenguatge de definició de vistes (LDV)** de l'usuari, per a especificar l'esquema extern i les seves correspondències amb l'esquema conceptual.

Ara bé, la realitat és que en la major part d'SGBD l'LDD s'utilitza tant per a descriure l'esquema conceptual (crear, modificar i eliminar taules, índexs, etc.) com per a definir i modificar l'esquema extern (vistes) i els permisos d'accés per als usuaris.

2) Control de dades

El **llenguatge de control de dades** (**LCD**) assegura un bon ús de la base de dades: permet el control de l'accés a la informació emmagatzemada en el diccionari de dades (definició de privilegis i tipus d'accés) així com també el control de la seguretat de les dades. Sovint, també es considera un subcomponent de l'LDD.

Exemples d'instruccions de control

Ordres SQL del llenguatge de control són definir o concedir permisos (GRANT) i denegar o revocar permisos (REVOKE).

3) Manipulació de dades

Una vegada compilats els esquemes de la base de dades i introduïdes les dades en aquesta, en la fase d'explotació de la base de dades l'usuari requereix algun mecanisme per a manipular-la. L'SGBD ofereix un **llenguatge de manipulació de dades (LMD)** per a aquesta finalitat.

Les quatre instruccions bàsiques corresponen a les quatre funcions bàsiques de manipulació de dades (recuperació, inserció, eliminació i modificació).

Exemples d'instruccions de manipulació

Les principals ordres en SQL que intervenen en la funció de manipulació de dades (recuperació, inserció, eliminació i modificació) són, respectivament: SELECT, INSERT, DELETE i UPDATE.

El **llenguatge de consulta** (*query language, QL*) és un subconjunt de l'LMD, especialment la part relacionada amb la recuperació i visualització de les dades. Ocasionalment, s'utilitza per a referir-se al conjunt complet de l'LMD.

Per a recuperar les dades, exigeix que es defineixin criteris de selecció i cerca. A l'hora de modificar un registre o més d'un de la base de dades, s'utilitza per a especificar els criteris d'actualització (per exemple, per a buscar els productes d'un proveïdor concret per tal d'incrementar el valor del seu preu en un percentatge determinat).

En els SGBD actuals, els tipus de llenguatge esmentats no es consideren llenguatges diferents, sinó diferents funcions integrades en un mateix llenguatge més ampli, anomenat **llenguatge de dades** o de base de dades (que disposa d'elements per a definir esquemes conceptuals, definir vistes, manipular dades i definir el seu emmagatzematge). La definició de l'emmagatzematge, normalment, es manté separada, atès que s'utilitza per a definir les estructures físiques d'emmagatzematge per tal d'harmonitzar l'execució del sistema de base de dades i normalment és utilitzada per l'administrador de la base de dades.

Exemple de llenguatge de base de dades

El llenguatge de base de dades relacionals **SQL** és un exemple representatiu que combina LDD, LDV i LMD, i també sentències per a l'especificació de restriccions i d'evolució de l'esquema. L'LDE, que fou un component en les primeres versions d'SQL, s'ha retirat per a mantenir aquest llenguatge únicament a nivell conceptual i extern.

Funcions i llenguatges associats

Com a resum del que s'ha exposat, en el quadre següent s'especifica la funció que fa cada llenguatge.

Funcions	Components associats
Definició de les dades	**Llenguatges de definició** d'esquemes de la base de dades
Manipulació de les dades	**Llenguatges de manipulació** de les dades
Control de les dades	**Llenguatges de control** de la base de dades

Els principals llenguatges de base de dades es poden definir de la manera següent:

El llenguatge de definició de dades (LDD) és un llenguatge especialitzat en la descripció (escriptura d'esquemes) de la base de dades (és a dir, en la creació i manteniment de l'estructura). Hi ha llenguatges específics per a esquemes conceptuals, per a esquemes interns i per a esquemes externs.

El llenguatge de manipulació de dades (LMD) és un llenguatge especialitzat en la utilització de la base de dades (consultes i manteniment).

15.3.2. Modalitats d'ús del llenguatge

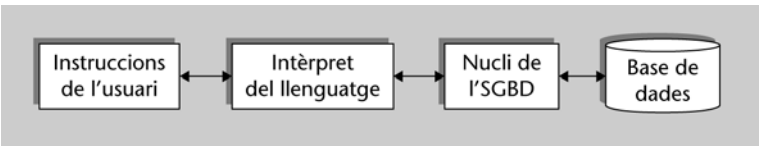
Bàsicament, hi ha dues maneres de treballar amb el llenguatge de base de dades, les definim a continuació.

1) Mode interactiu

En el mode interactiu (o directe) s'estableix una conversa entre l'usuari i el programari de l'SGBD similar a la manera en què es manté un diàleg amb el sistema operatiu. Qualsevol instrucció del llenguatge de base de dades (LDD, LMD...) s'introdueix sense restriccions directament des del teclat d'un terminal. L'SGBD disposa d'un *intèrpret* que analitza, verifica i executa les instruccions.

Aquesta modalitat d'ús, no gaire habitual, respon a l'esquema descrit en la figura següent.

Figura 20. Execució del llenguatge des d'un terminal



2) Mode programat

En el mode programat (des d'un programa d'aplicació), l'usuari executa una aplicació sobre el sistema operatiu.

Atès que, en general, els llenguatges de programació convencionals no tenen instruccions pròpies per a bases de dades, cal usar el llenguatge de base de dades des de dintre d'un programa. En aquest cas, es poden presentar dues possibilitats:

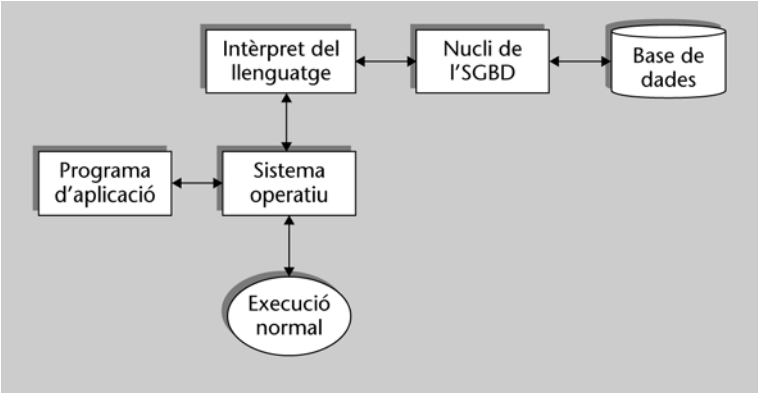
a) Un **programa escrit íntegrament en el llenguatge de base de dades** (SQL). Hi ha extensions de SQL que el doten de les estructures de programació imperativa usuals (bucles, selecció). Un exemple és el llenguatge PL/SQL de l'entorn Oracle i PostgreSQL. Es tracta d'un llenguatge procedimental que permet desenvolupar programes imperatius que finalment accedeixen a la base de dades a través d'SQL. Aquests programes són com guions que l'*intèrpret* de SQL va seguint.

Dins del programa escrit en el llenguatge habitual s'inclouen **crides a funcions** (*call level interface, CLI*), que són llibreries de funcions especialitzades a sol·licitar al servidor de la base de dades l'execució d'instruccions del llenguatge de base de dades (per exemple, llibreries ODBC o JDBC). Aquestes funcions s'encarreguen d'enviar les instruccions en el llenguatge de base de dades a l'SGBD en temps d'execució.

b) Llenguatge hostatjat o immers (*embedded*). Consisteix a incorporar les instruccions del llenguatge de base de dades (gairebé sempre SQL) directament al programa escrit en un llenguatge de programació de propòsit general (Cobol, PL/I, C, Pascal, Java). Això implica disposar d'un *precompilador* especialitzat que accepti, dins del llenguatge de programació, les instruccions del llenguatge de base de dades. El llenguatge de programació que conté les sentències de base de dades s'anomena **llenguatge amfitrió** (*host*).

Les instruccions del llenguatge de programació s'executen pel procediment usual i les d'SQL es transfereixen a un mòdul especial d'execució de l'SGBD. En la figura següent es mostra un esquema d'aquesta modalitat d'ús:

Figura 21. Execució del llenguatge des d'un programa



Una implementació de SQL hostatjat estableix les relacions que han de mantenir els objectes de la base de dades amb els objectes del programa amfitrió, així com també certes restriccions de funcionament.

La modalitat de llenguatge hostatjat admet dues variants:

- **Llenguatge estàtic**: el programador escriu explícitament les instruccions que vol que s'executin en el programa, que no admet canvis durant l'execució. Té l'avantatge d'estalviar temps d'execució, ja que la verificació i la traducció es fan en temps de compilació. Aquest és el mètode usat en la major part d'aplicacions.

- **Llenguatge dinàmic**: el programador escriu parcialment les instruccions que s'han d'executar i en deixa algunes perquè siguin completades en temps d'execució del programa. És necessari usar-lo si durant l'execució varia algun paràmetre o part del programa. Permet escriure programes genèrics (com les eines d'accés visual a bases de dades) que dialoguen amb l'usuari final per a determinar què han de fer. En escriure el programa no se sap exactament quines instruccions s'hauran d'enviar a l'SGBD (que es concretaran en temps d'execució), ja que això depèn del que l'usuari demani. Aquesta variant és menys eficient que un programa en llenguatge estàtic i més difícil per al programador, ja que utilitza tècniques dinàmiques de maneig de variables.

  La modalitat CLI del llenguatge és dinàmica, ja que el programa passa les instruccions del llenguatge de base de dades com el valor d'un paràmetre i, per tant, les pot construir en temps d'execució.

15.3.3. Llenguatges i aplicacions

Els tipus de llenguatge, pel que fa les aplicacions que suporten, es classifiquen en dos grups (principalment, referits a LMD):

1) Llenguatges d'alt nivell o no procedimentals

Només requereixen que en les sentències s'especifiqui *quines dades* es volen manipular i *què* es vol obtenir, però no com s'ha de fer (el mateix llenguatge és l'encarregat de determinar els procediments més efectius per a fer-ho). Per això, aquests llenguatges també s'anomenen **declaratius o implícits**.

Es poden utilitzar de manera independent per a especificar operacions complexes de base de dades de manera concisa. En molts SGBD és possible introduir instruccions d'LMD d'alt nivell de dues maneres diferents:

a) Interactivament des d'un terminal.

b) Hostatjades en un llenguatge de programació de propòsit general. En aquest cas, cal identificar les sentències del llenguatge de dades dins del programa per tal que el **precompilador** les pugui extreure i l'SGBD les pugui processar.

Permeten extreure informació agrupada especificant i recuperant molts registres amb una sola instrucció, i per aquest motiu s'anomenen **llenguatges orientats a conjunt**.

També s'anomenen **llenguatges orientats a la consulta**, atès que permeten que l'usuari faci ús de les eines del sistema per a accedir a les dades mitjançant instruccions aïllades (emprant les eines de disseny visual com a generadors de formularis, informes...) o bé mitjançant grups d'instruccions.

Atès que proporcionen un cert nivell d'abstracció del codi màquina subjacent (cada sentència es tradueix en més d'una instrucció de llenguatge màquina), són fàcils d'entendre i manejar per part d'usuaris no experts.

Els LDD són declaratius

Per la seva pròpia naturalesa, els llenguatges de definició de dades (LDD) són declaratius o no procedimentals.

Quan els requisits de temps de resposta de l'SGBD en els processos de manipulació de les dades són importants, és necessària la modificació del codi que generen els llenguatges declaratius per a assegurar un rendiment adequat del sistema. Això s'aconsegueix amb els llenguatges procedimentals.

2) Llenguatges de baix nivell o procedimentals (o explícits)

Requereixen que en les sentències s'especifiqui *quines dades* es volen manipular, *què* es vol obtenir i *amb quins procediments* (s'han de conèixer més qüestions del funcionament de l'SGBD per a detallar pas a pas com s'han d'executar les operacions). Per això s'anomenen **llenguatges orientats a procediments**. (Cada instrucció escrita se sol correspondre amb una instrucció de màquina; és, per tant, un llenguatge que depèn de l'ordinador que l'utilitza i comprensible per al llenguatge màquina). També s'anomenen *explícits* o *no declaratius*.

Han d'estar incorporats o hostatjats en un llenguatge de programació de propòsit general.

Les aplicacions verticals

Són programes d'aplicació dissenyats per a satisfer les necessitats d'un sector empresarial molt específic (per exemple, restaurants, comerços al detall, consultes mèdiques…). Solen proporcionar funcions completes d'administració, com ara planificació, facturació, control d'inventaris i compres.

En contraposició, les *aplicacions horitzontals* donen suport a organitzacions i àrees de negoci molt diverses. Són programes d'aplicació tranversals com ara eines ofimàtiques (per exemple, processadors de text o fulls de càlcul), sistemes CRM, etc.

S'utilitzen per a crear programes, mòduls o procediments (seqüència, amb nom, d'instruccions) que s'executen des d'un **gestor del llenguatge** o s'incrusten en un **llenguatge amfitrió**. La seva utilitat consisteix a permetre el desenvolupament d'aplicacions verticals, interactives o per lots per al seu ús destinades a l'usuari final amb pocs coneixements informàtics, mantenir per a aquest usuari un esquema extern tan limitat com sigui possible i ocultar-li la complexitat de la base de dades, que utilitza bàsicament per a emmagatzemar i recuperar informació sense cap tractament més complex.

En general, aquest tipus de llenguatge recupera registres (o objectes) individuals de la base de dades i els processa per separat. Per tant, necessita utilitzar elements de llenguatges de programació, com els bucles (subrutines), per a recuperar i processar cada registre individual d'un conjunt de registres. Per aquesta raó, es coneixen com **llenguatges orientats a registre**.

Els llenguatges utilitzats en els SGBD prerelacionals eren procedimentals.

SGBD prerelacionals

SQL relacional és bàsicament declaratiu, però té alguns detalls procedimentals.

Els llenguatges procedimentals s'acostumen a aprendre i utilitzar amb més dificultat que els declaratius (per aquest motiu només són utilitzats per usuaris informàtics).

Resumint:

> Els **llenguatges declaratius** només requereixen que s'especifiqui *quines dades* es volen manipular i *què* es vol obtenir, però no com. En canvi els **llenguatges procedimentals** requereixen que, a més, s'especifiqui *amb quins procediments* s'han de fer.

Els llenguatges procedimentals permeten formular processos més eficients que els declaratius.

D'altra banda, els LMD d'alt nivell utilitzats de manera interactiva i independent, reben el nom de **llenguatges de consulta.** En general, tant les instruccions de recuperació com les d'actualització de dades d'un LMD d'alt nivell es poden utilitzar de manera interactiva; per tant, es consideren part del llenguatge de consulta. (Tot i que, d'acord amb el significat de la paraula *consulta*, realment només s'hauria d'utilitzar per a descriure la recuperació de dades, no l'actualització).

15.3.4. Llenguatges i models de dades

Segons el model de dades, els llenguatges de bases de dades es poden classificar de cinc maneres diferents. Les presentem a continuació.

1) Llenguatges relacionals

a) Basats en l'àlgebra relacional

Són llenguatges procedimentals en què l'usuari o el programador han de conèixer la sintaxi del llenguatge que utilitzen.

- **SQL**. És el **llenguatge** més utilitzat en bases de dades relacionals.

  Apareix en 1979 de la mà d'Oracle, que s'avança per poc temps a IBM (1982). És l'evolució de **SEQUEL** (*structured English query language*), llenguatge proposat l'any 1975 per Boyce i que procedeix de **SQUARE** (*specifying queries as relational expressions*). Té una sintaxi en anglès gairebé natural. Disposa d'instruccions de tres tipus diferents:

  – **De manipulació**: per exemple, SELECT per a fer consultes i INSERT, UPDATE i DELETE per al manteniment de les dades.

  – **De definició**: per exemple, CREATE TABLE per a definir les taules, les columnes i les restriccions.

  – **De control de l'entorn**: per exemple, COMMIT i ROLLBACK per a delimitar transaccions.

Algunes instruccions d'SQL es comenten en el subapartat "Llenguatges i funcions" d'aquest mateix apartat.

b) Basats en el càlcul relacional

Són llenguatges no procedimentals en què l'usuari o el programador han de conèixer el tipus de resultat que esperen aconseguir, les dades originals a partir de les quals han d'obtenir el resultat i els criteris de selecció o filtratge imposats a les dades d'entrada. Destaquen els següents:

- **QUEL** (*query language* creat en 1974 per IBM). És el DML de l'SGBD Ingres (que s'executava en màquines UNIX) i per això moltes de les seves característiques procedeixen d'aquest llenguatge. Es basa en càlcul relacional de tuples. La seva sintaxi és en anglès i se sembla a la de SQL.

- **QBE** (*query by example*, creat en 1977 també per IBM). Per mitjà d'una interfície gràfica interactiva, similar als filtres implementats en els fulls de càlcul, mostra taules i permet generar expressions sobre els seus camps utilitzant un exemple de resultat. Es basa en càlcul relacional de dominis. Access i dBase incorporen SQL i QBE.

2) Llenguatges de sistemes en xarxa

A partir del model en xarxa Codasyl apareixen llenguatges per als SGBD següents:

a) TOTAL. El seu llenguatge s'ha d'incrustar en un llenguatge amfitrió per a la seva execució.

b) IDMS (*integrated database management system*). Disposa de DML, DDL i DCL, que compleixen parcialment l'estàndard Codasyl.

3) Llenguatges de sistemes jeràrquics

El llenguatge principal per a accedir a bases de dades jeràrquiques és:

a) DL/1 (d*ata language one*) d'IBM. És el llenguatge de dades de l'SGBD jeràrquic DMI (*information management system*) d'IBM aparegut els anys seixanta. Es tracta d'un llenguatge hoste i procedimental que s'ha d'incrustar en altres llenguatges. Malgrat que ha estat superat per altres alternatives tecnològiques, ha sobreviscut, ja que suporta aplicacions en Java, JDBC, XML i serveis web.

4) Llenguatges orientats a la programació

Un llenguatge de base de dades definit expressament per a la programació és el següent:

a) NATURAL. És el llenguatge de quarta generació de l'SGBD ADABAS (de Software AG). Processa la informació de bases de dades que emmagatzemen les dades en fitxers seqüencials o seqüencials indexats. Es pot executar com a mòduls de programari i pot efectuar crides a altres llenguatges.

5) Llenguatges orientats a l'objecte

Els llenguatges de consulta d'SGBD orientats a l'objecte més destacats són els següents:

a) OSQL. És el llenguatge de consulta de l'SGBD orientat a l'objecte IRIS, que és una extensió dialectal OO de SQL.

b) OQL. És el llenguatge de consulta de l'SGBD orientat a l'objecte O2 (actualment, Ardent).

15.3.5. Llenguatges i usuaris

Per tal de comunicar-se amb l'SGBD, cada usuari (o programa d'aplicació) fa servir un llenguatge. Segons el tipus d'usuari a qui van destinats, els llenguatges es poden classificar en tres grups:

Comunicació entre l'usuari i l'SGBD

L'usuari necessita comunicar-se amb l'SGBD per a formular (mitjançant un llenguatge) accions com, per exemple:

- Descriure la base de dades (dissenyada pel dissenyador).
- Actualitzar la base de dades (amb les dades facilitades).
- Sol·licitar la recuperació d'informació (consultes).

1) Llenguatges per a usuaris informàtics experts

Administradors i programadors requereixen **llenguatges potents i flexibles** per a escriure processos complexos que els permetin definir, administrar, extreure i manipular dades o, fins i tot, crear i ajustar les aplicacions verticals que hi accedeixen.

a) En molts casos utilitzen DDL i DML en la seva **forma hostatjada en un llenguatge amfitrió**, per la qual cosa aquest haurà de permetre **crides** a l'SGBD **des d'un programa** d'aplicació.

b) En altres casos el llenguatge utilitzat serveix per a parametritzar la base de dades abans de l'inici dels treballs de l'usuari final.

2) Llenguatges per a usuaris ocasionals

Aquest usuaris, com a simples gestors de les aplicacions que utilitzen, normalment només solen fer consultes. Per tal d'especificar-les, necessiten un **llenguatge de consulta d'alt nivell** que sigui prou senzill (sense utilitzar ordres estructurades), encara que doni un rendiment baix en temps de resposta.

3) Llenguatges per a usuaris simples i paramètrics

Els usuaris paramètrics (dedicats, per exemple, a introduir dades massivament) necessiten **llenguatges molt eficients i compactes**, potser especialitzats en tipus concrets de tasques (transaccions programades), encara que no siguin fàcils d'aprendre.

Els usuaris paramètrics i les transaccions programades

Es tracten en el subapartat "Usuaris finals" dins l'apartat "Usuaris de les bases de dades".

Per als usuaris paramètrics, els ocasionals i altres que no volen aprendre els detalls d'un llenguatge de consulta d'alt nivell, l'SGBD proporciona **interfícies amigables per a l'usuari** que permeten interactuar amb la base de dades. Aquestes interfícies s'analitzen a continuació.

15.4. Interfícies de l'SGBD

L'SGBD pot oferir diferents tipus d'interfícies segons el mètode utilitzat per a realitzar les sol·licituds i la categoria d'usuari a la qual van dirigides.

15.4.1. Interfícies i mètode

Segons el mètode utilitzat per a fer les sol·licituds al sistema (teclejant instruccions, escollint opcions de menú, fent clic en els botons amb el ratolí...), les interfícies poden ser de quatre tipus:

1) Interfícies de llenguatge natural

Accepten sol·licituds escrites (en una línia d'ordres o de comando) utilitzant un subconjunt de paraules d'algun idioma, habitualment l'anglès. La interfície consulta les paraules estàndard del seu propi esquema (l'*intèrpret d'ordres*) per a interpretar la sol·licitud. Si la interpretació té èxit, la interfície **genera una consulta** d'alt nivell (que correspon a la sol·licitud feta per l'usuari en llenguatge natural) i l'envia a l'SGBD per al seu processament. En cas contrari, s'inicia un diàleg amb l'usuari per tal d'aclarir la sol·licitud. També es pot anomenar *interfície d'ordres*.

2) Interfícies de navegació basades en menús

Són interfícies que presenten llistes d'opcions denominades *menús*, que guien l'usuari per a formular una sol·licitud.

Menú

Un **menú** és una llista que es presenta en pantalla i que mostra diverses opcions juntament amb un mecanisme que recull l'opció escollida i, seguidament, l'executa. Un menú fa innecessari memoritzar les instruccions i la sintaxi específica d'un llenguatge de consulta, ja que permet elaborar la sol·licitud pas a pas.

Actualment, els **menús desplegables** (molt utilitzats en interfícies basades en finestres) permeten que l'usuari examini els continguts de la base de dades utilitzant una forma d'exploració no estructurada.

Aquest ha estat un tipus d'interfície molt popular i còmode fins a l'arribada al mercat dels SGBD basats en finestres (*Windows*). Tot i així, l'existència de programes d'aplicació que no utilitzen interfícies gràfiques d'usuari (GUI) fa que se segueixi utilitzant.

3) Interfícies basades en formularis

Aquest tipus d'interfícies presenten a cada usuari un conjunt de *formularis* que permet atacar les dades emmagatzemades (bàsicament, modificar o afegir) d'acord amb les especificacions de seguretat i accés definides prèviament.

Formulari

Un **formulari**, que també rep els noms de *finestra d'usuari*, *format de pantalla* o, simplement, *pantalla*, forma part de l'esquema extern de l'usuari. És la manera més còmoda i natural per a comunicar l'usuari amb la base de dades i ofereix una bona presentació i organització de les dades. Internament, fa el mateix que es pot fer treballant amb comandaments directes del llenguatge de consulta.

Un formulari és una finestra o quadre estructurat de presentació en pantalla amb àrees predefinides de blocs de dades (contenidors lògics de quadres de text, llistes...) i altres elements de control (caselles de verificació, botons...) que permeten fer tasques que presentem a continuació.

1) Establiment d'un ordre predeterminat per a la gestió de les dades.

2) Visualització de les dades **en un format** determinat, segons el tipus de dada emmagatzemada. Molt esporàdicament, un formulari s'usa per a mostrar dades, ja que per a això és millor elaborar un informe.

3) Recuperació de dades de registres que coincideixen amb valors especificats. La consulta de dades no és tampoc l'ús més habitual dels formularis.

4) Filtratge de la informació. Aquesta tasca es pot dur a terme en dos nivells:

a) **Filtratge en la introducció** mitjançant una sèrie de **controls**.
Per exemple, impedint la introducció de lletres en un camp numèric o limitant les opcions d'una llista desplegable en funció d'una dada introduïda prèviament.

b) **Filtratge visual en la recuperació** de les dades.
Per exemple, mostrant un registre únic o diversos registres alhora, i veient alguns o tots els atributs de les dades subjacents.

5) Gestió de gran part de la coherència de la informació per tal d'obtenir resultats de qualitat durant qualsevol acció sobre les dades.
Per exemple, en una base de dades bibliogràfica, impedint que s'esborri el registre corresponent a un autor si abans no s'han eliminat totes les seves obres.

6) Modificació i eliminació de dades existents.

7) Introducció de dades noves.

Es tracta d'aplicacions construïdes sota el concepte *client-servidor*, ja que estan a l'espera d'una acció per part de l'usuari per a respondre de manera

programada. És a dir, resten inactives mentre l'usuari llegeix dades, escriu text o no interactua, però, a una petició d'aquest (per exemple, un clic de ratolí sobre un botó), responen, com a servidor, amb l'execució de l'acció o transacció sol·licitada. Per aquest motiu, també s'anomenen **interfícies de transaccions programades**.

Solen estar destinades a usuaris paramètrics que habitualment fan un tipus de transaccions estàndard (consultes i actualitzacions).

4) Interfícies gràfiques d'usuari

Les interfícies gràfiques d'usuari (GUI, *graphic user interface*, en anglès) solen presentar els esquemes de la base de dades en forma de diagrama i permeten que l'usuari especifiqui una consulta manipulant el diagrama. Solen utilitzar un *dispositiu apuntador*, com el ratolí, per a escollir les parts del diagrama de l'esquema mostrat. Habitualment, els elements gràfics (finestres, icones, botons...) també es combinen amb menús i formularis , com és el cas del sistema Windows de Microsoft.

15.4.2. Interfícies i usuaris

Segons els tipus d'usuaris a qui van destinades, destaquen les següents interfícies d'usuari:

1) Interfícies per a usuaris paramètrics

Disposen d'un conjunt restringit d'**operacions abreviades** que aquest tipus d'usuaris, com ara caixers d'un banc, han de fer repetidament. L'objectiu és que per a fer cada tasca o sol·licitud concreta, només calgui introduir una quantitat mínima de paràmetres o efectuar un nombre reduït de pulsacions de teclat. Per exemple, es poden programar les tecles de funció perquè s'iniciïn instruccions determinades.

2) Interfícies per a l'administrador de la base de dades

La majoria d'SGBD contenen **instruccions privilegiades** que només pot utilitzar l'administrador.

En són exemples les instruccions per a establir els paràmetres del sistema, crear perfils d'usuari, atorgar rols o autoritzacions als perfils, crear connexions a bases de dades remotes, modificar els esquemes i reorganitzar l'estructura d'emmagatzematge d'una base de dades.

15.5. El nucli de l'SGBD

El nucli és el component encarregat de garantir que l'emmagatzematge de les dades i l'accés a aquestes siguin correctes, segurs, íntegres i eficients. Els meca-

nismes d'emmagatzematge de les dades, transparents per a l'usuari, són tant importants com els dedicats a protegir la base de dades contra possibles riscos i perills (intencionats o accidentals).

Per això, un SGBD multiusuari ha d'oferir un conjunt de controls de seguretat d'accés i de seguretat interna enfront de pèrdues d'integritat; controls de concurrència i recuperació enfront de fallides del sistema, etc. Un altre aspecte important per a obtenir un bon rendiment és l'optimització de consultes.

Recuperació i concurrència en sistemes monousuari

Els SGBD monousuari, normalment, no ofereixen sistemes de **recuperació** automàtica per si es presenta una fallida. Es considera que l'usuari és responsable de preparar còpies de seguretat de la base de dades i recuperar les seves dades manualment.

D'altra banda, la **concurrència** no és pertinent en sistemes monousuari.

Interfície amb el sistema operatiu

En la figura 19 del subapartat "Components de l'SGB" d'aquest mateix apartat es veu com el nucli actua com a interfície entre l'SGB i el SO.

Correspon al nucli, com a interfície amb el sistema operatiu, transformar les peticions de dades que rep de l'usuari en instruccions que el sistema operatiu pugui entendre i gestionar. El nucli proporciona una interfície entre les dades (a nivell d'emmagatzematge físic) i els programes d'aplicació dissenyats per a la seva manipulació. Es pot considerar que actua com a intèrpret entre l'usuari i les dades. Cada operació que s'ha de fer "contra" la base de dades ha de ser permesa prèviament pel nucli, que una vegada interpretada i validada la instrucció, o bé executa l'operació retornant el resultat d'aquesta al programa (o procediment) que l'ha sol·licitat o bé la rebutja.

Els components del nucli de l'SGBD

Els components del nucli de l'SGBD intervenen en l'execució d'una consulta feta per un programa d'usuari tal com es mostra en l'esquema de la gestió d'accés a les dades feta per l'SGBD de l'apartat "Accés de l'SGBD a les dades".

El motor de l'SGBD (que sempre s'executa en el servidor) sol tenir mòduls que es concentren en el nucli del sistema, per a executar, com a mínim, les funcions següents:

- Control de la seguretat i la privadesa.
- Gestió de vistes.
- Interpretació del llenguatge de base de dades (SQL).
- Optimització de consultes.
- Control de l'accés concurrent (per exemple, la gestió de bloquejos).
- Gestió de memòries intermèdies (*buffers*).
- Gestió de l'espai de la memòria externa.
- Mètode d'accés (per exemple, els índexs).
- Gestió de processos (multitasca, paral·lelisme).
- Gestió de transaccions, disparadors i procediments emmagatzemats.
- Manteniment de diaris d'actualitzacions (*log*).

Arquitectura funcional i processament d'una sentència

Per a descriure i comprendre l'arquitectura funcional d'un SGBD es pot seguir el processament d'una sentència LMD del llenguatge de base de dades (habitualment SQL).

L'estructura interna d'un SGBD modern inclou una gran quantitat de components que són mòduls de programari. Des d'un punt de vista d'alt nivell, el nucli de l'SGBD es pot considerar constituït, únicament, per tres blocs anomenats gestors: el gestor de sentències, el gestor de concurrència i el gestor de dades.

15.5.1. Gestor de sentències

És la part de l'SGBD que interactua directament amb l'usuari. Per aquest motiu, rep el nom de **part del davant** (*front-end*) o secció frontal de l'SGBD. Es correspon amb el processador **client** que inclou la interfície d'usuari i els processos o serveis en primer pla. Engloba quatre components amb funcionalitats ben diferenciades:

1) Component de control de seguretat d'accés

Abans de processar una sentència cal comprovar si l'usuari que en demana l'execució està autoritzat per a executar el conjunt d'operacions que implica la sentència (per a determinar-ho es fa servir un conjunt de **tècniques de seguretat** relatives a la privadesa de les dades). Si la sentència no supera aquests filtres de seguretat, és rebutjada pel sistema i la seva execució no continua endavant.

2) Component de transformació d'esquemes

Si la sentència és autoritzada, l'esquema extern es transforma en l'esquema conceptual. Aquest procés, anomenat **processament de vistes**, permet eliminar les referències a vistes i obtenir una sentència d'operacions més homogènia referida únicament a entitats o objectes de l'esquema conceptual (taules, en el cas de bases de dades relacionals).

3) Component de control d'integritat

Cal comprovar que les dades emmagatzemades satisfan els **requeriments d'integritat** definits en l'esquema conceptual i, així, assegurar-ne la consistència (claus primàries i foranes, i altres restriccions predefinides).

Un cop superats tots aquests controls, la sentència s'ha d'executar. Però si el llenguatge de la sentencia és declaratiu s'ha de **transformar** (traduir + descompondre) en un conjunt d'accions expressades en un llenguatge procedimental que pugui ser entès pel sistema operatiu.

4) Component de processament de sentències

Primer s'ha de *traduir* la sentència a operacions, que posteriorment s'han de *descompondre* en operacions més elementals sobre els elements físics de la base de dades. (En una base de dades relacional, una sentència SQL es tradueix a operacions de l'àlgebra relacional.)

Les sentències dels usuaris de la base de dades

Els usuaris ocasionals executen consultes interactives, els usuaris paramètrics executen transaccions programades, els programadors d'aplicacions creen programes d'aplicació que executen sentències de manipulació de dades (LMD), i els administradors de la base de dades executen sentències de definició de dades (LDD) i instruccions privilegiades.

Les sol·licituds dels clients (usuaris i programes d'aplicació) es transformen en codi d'accés a la base de dades per a ser executades pel processador de base de dades en temps d'execució.

El component de processament de sentències inclou els mòduls següents:

- **Compilador de consultes**. Maneja les consultes d'alt nivell que l'usuari introdueix de manera interactiva. Analitza la seva sintaxi, les compila o interpreta i crea el codi d'accés a la base de dades (per a executar aquest codi, genera crides al **processador en temps d'execució**).

- **Precompilador**. Extreu les instruccions escrites en LMD hostatjat en un programa d'aplicació escrit en un llenguatge de programació amfitrió (i les agrupa en un **mòdul de sol·licituds a la base de dades**). Aquestes instruccions s'envien al **compilador d'LMD**. La resta del programa s'envia al **compilador del llenguatge amfitrió**.

- **Compilador d'LMD**. Converteix les instruccions rebudes (de part del precompilador) en codi objecte per a l'accés a les dades.

- **Lligador**. Enllaça (munta, *linkedit*) el codi objecte de les instruccions en LMD i el de la resta del programa i forma una **transacció programada** (que rep el nom de **pla d'accés**) el codi executable de la qual inclou crides al processador de la base de dades durant el temps d'execució.

- **Optimitzador**. És un dels subcomponents principals del lligador. En cadascuna de les dues transformacions de la sentència (traducció + descomposició), s'encarrega d'escollir la millor combinació possible d'operacions procedimentals que satisfaci l'execució de la sentència LMD processada pel lligador.

  Pel fet que les sentències es poden executar de formes diferents (no totes les operacions tenen el mateix cost en temps de resposta), s'ha d'establir el mètode d'accés més idoni per a augmentar la rapidesa sense que afecti negativament la resta del sistema. És a dir, s'han de reduir al mínim els accessos al disc per a l'obtenció de les dades, la utilització del processador (CPU), etc.

  El procés d'optimització acaba reflectint aquest resultat en un procediment anomenat **pla d'accés o d'execució**, que és l'estratègia d'execució d'una sentència que té associat un consum mínim de recursos (bàsicament, temps de resposta).

Els SGBD relacionals en l'optimització de sentències

Un SGBD es considera eficient quan la resposta a una sentència es fa en el menor temps possible. Per a això, ha de disposar d'un mètode que optimitzi aquesta resposta. Aquesta optimització només és aplicable en SGBD relacionals, que, tot aplicant l'àlgebra relacional, han de trobar la manera més curta d'executar una sentència.

Processament (i optimització) de consultes

Una consulta expressada en un llenguatge d'alt nivell (com SQL) ha de passar primer per una anàlisi lèxica, una anàlisi sintàctica i una validació. El procés en una base de dades relacional inclou:

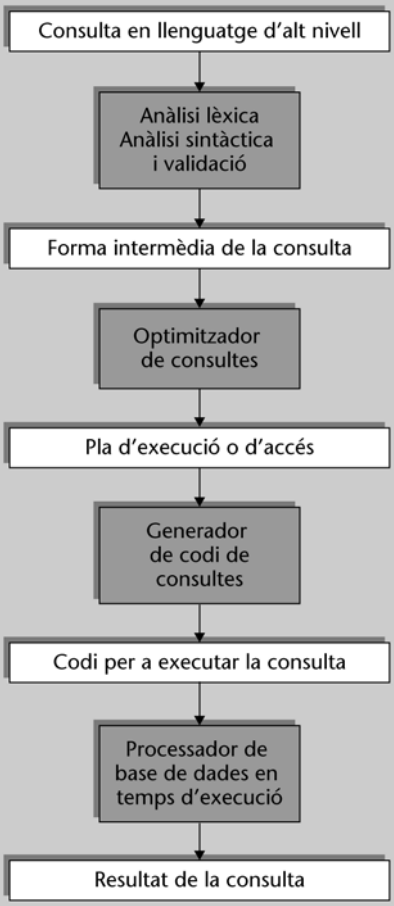
- **Analitzador lèxic**. Identifica els símbols del llenguatge (com les paraules clau del llenguatge SQL, els noms dels camps i els noms de les taules) en el text de la consulta.
- **Analitzador sintàctic**. Revisa la sintaxi de la consulta per a determinar si està formulada d'acord amb les regles sintàctiques del llenguatge de consulta.
- **Validació de la consulta**. Es comprova que tots els noms de camps i de taules siguin vàlids i tinguin sentit des del punt de vista semàntic en l'esquema de la base de dades.

A continuació, es crea una **representació interna** de la consulta, normalment en forma d'estructura de dades d'arbre (anomenada *estructura d'arbre de consulta*).

Seguidament, l'SGBD ha de crear l'estratègia d'execució o **pla d'execució** per a obtenir el resultat de la consulta a partir dels fitxers implicats de la base de dades. El procés d'escollir l'estratègia d'execució més adequada per a processar una consulta s'anomena **optimització de consultes**.

En la figura es mostren els diferents passos per a processar una consulta d'alt nivell.

Figura 22. Processament d'una consulta d'alt nivell



- El mòdul **optimitzador de consultes** és responsable d'escollir el mètode d'accés o pla d'execució més eficient per a portar a terme la consulta.

- El **generador de codi** genera el codi necessari per a executar la consulta. El codi es pot executar directament (mode interpretat) o emmagatzemar i executar quan sigui necessari (mode compilat).
- El **processador de base de dades en temps d'execució** s'encarrega d'executar el codi de la consulta per a produir el resultat de la consulta. Si es presenta un error durant l'execució, aquest processador genera un missatge d'error.

L'SGBD porta a terme tres tipus d'optimització: sintàctica, física i semàntica. Normalment, l'optimització semàntica no es troba implementada en els SGBD comercials.

Quan la sentència LMD s'ha transformat en el **pla d'accés**, cadascuna de les operacions elementals que el componen segueix altres tractaments a través de la resta de l'SGBD.

En primer lloc passa pel **gestor de concurrència** i, finalment, pel **gestor de dades**.

15.5.2. Gestor de concurrència

Aquest gestor (també anomenat **component de serveis de bloqueig**) subministra els controls necessaris per a gestionar l'accés concurrent a bases de dades multiusuari evitant la pèrdua de consistència o integritat de les dades. És a dir, controla l'execució concurrent de múltiples operacions elementals d'actualització corresponents a diferents sentències que afecten la mateixa dada i comprova que l'acció que intenta fer cada operació elemental no entri en conflicte amb l'acció d'una altra operació que també s'estigui executant.

15.5.3. Gestor de dades

Finalment, si no hi ha cap problema de concurrència, cada operació elemental s'executa "contra" la base de dades. D'això s'encarrega el gestor de dades, que representa la part de l'SGBD més allunyada de l'usuari. Per això, se sol anomenar **part del darrere** (*back-end*) o secció posterior de l'SGBD. Correspon al servidor on es produeixen els processos en segon pla (a vegades s'anomena **component de serveis de base de dades**).

El gestor de dades és responsable de la **interacció amb el sistema operatiu** (en particular, amb l'administrador d'arxius d'aquest). L'estructura utilitzada per a l'emmagatzematge dels arxius varia en funció del sistema operatiu implantat en l'ordinador.

El gestor de dades és un dels components de més complexitat de l'SGBD. Depèn del mateix SGBD, de les característiques del sistema operatiu i del maquinari en el qual s'implanti.

Factors que afecten les funcions del gestor de dades

La correcta execució de les funcions assignades al gestor de la base de dades depèn de factors molt diversos. Destaquem els factors següents:

- El volum de la base de dades.
- Les estructures físiques definides per a l'emmagatzematge de les dades.
- Els procediments desenvolupats per a la manipulació de les dades.
- Les característiques del maquinari.
- La qualitat del mateix gestor de dades.

El gestor de dades es divideix en diversos subcomponents principals, les funcions dels quals es detallen a continuació:

1) Processador de base de dades en temps d'execució

Aquest component s'encarrega dels accessos a la base de dades durant l'execució. Rep operacions de recuperació o d'actualització i les executa sobre la base de dades segons les instruccions del **pla d'accés** (que supervisa l'accés al disc, mitjançant el gestor de dades emmagatzemades).

2) Gestor de recuperació

El gestor de recuperació s'encarrega de prendre nota de les operacions elementals (lectura o escriptura de dades) abans que no es produeixin, en un **diari d'operacions** que permetrà recuperar-les en cas de fallides posteriors del sistema.

3) Gestor de dades emmagatzemades

Aquest gestor controla l'accés a la informació emmagatzemada en el disc (tant de la **base de dades** com del **catàleg**).

Pot utilitzar serveis bàsics del sistema operatiu per a transferir físicament les dades entre el disc i la memòria principal. Concretament, és responsable de les funcions atribuïdes a l'**administrador d'arxius**. Ara bé, durant l'execució de la seva tasca bàsica, controla altres aspectes de la transferència de dades invocant altres components quan cal portar a terme funcions de detall (com el bloqueig, la connexió d'usuaris, el maneig de memòries intermèdies, etc.)

Administrador d'arxius

Les funcions de l'administrador d'arxius que permeten manipular les estructures d'emmagatzematge de dades es comenten en el pas 5 de l'esquema de la gestió d'accés a les dades feta per l'SGBD de l'apartat "Accés de l'SGBD a les dades".

4) Gestor de memòria intermèdia

El gestor de memòria intermèdia resol totes les operacions elementals que pot en la memòria intermèdia i només accedeix al disc quan és necessari. Quan les dades són en aquesta àrea de memòria intermèdia, poden ser processades per altres mòduls de l'SGBD o per programes d'aplicació.

Altres mòduls del nucli

El nucli de l'SGBD inclou altres components:

- Processadors d'interfícies.
- Mòdul per a implementar el catàleg.
- Mòduls de comunicació amb altres programes com el sistema operatiu o els compiladors de llenguatges de programació.

Interacció amb el sistema operatiu

Quan es requereix accés al disc (a la base de dades o al catàleg), l'SGBD interactua amb el sistema operatiu per mitjà del mòdul gestor de dades.

Quan s'executen programes amfitrió que accedeixen a les dades des de l'LMD hostatjat en aquests, per cada programa es crea una unitat d'execució en la qual hi ha una àrea de treball d'usuari, amb les seves àrees d'entrada, sortida i de comunicació amb el gestor de dades. L'àrea de comunicació és la responsable de rebre els missatges i la informació de control procedents del gestor.

L'execució de les **ordres de l'LMD** segueix els passos següents:

- Una **unitat d'execució** crida el gestor de dades. Aquesta crida té informació sobre l'esquema extern del client (usuari final) que vol accedir a la base de dades.

- El gestor de dades completa la informació de l'esquema extern amb l'estructura conceptual i interna. La informació d'aquestes estructures està emmagatzemada en forma d'esquemes conceptual i intern en el **catàleg**.

- El gestor de dades tradueix la petició, la converteix en ordres per als **mètodes d'accés** del sistema operatiu i aquests accedeixen al disc en què està emmagatzemada la base de dades.

- Es recuperen les dades sol·licitades i es desen en una **àrea de memòria intermèdia**, des de la qual són transferides a l'**àrea de treball** (d'E/S) de l'usuari.

- El gestor de dades, quan ha completat la manipulació de les dades, passa a l'**àrea de comunicació** els indicadors d'estat en els quals s'indica si l'operació ha finalitzat correctament. Si en revisar l'estat dels indicadors no n'hi ha cap que hagi tingut problemes, les dades són utilitzades pel programa d'aplicació.

Resum

La definició d'una sèrie de **conceptes bàsics** a l'inici del mòdul ens ha ajudat a entendre els continguts sobre sistemes de base de dades desenvolupats al llarg d'aquest mòdul. Hem definit *base de dades* com el conjunt estructurat de fitxers de dades entre les quals s'estableixen relacions i sistema gestor de bases de dades (SGBD), com el conjunt de programari que permet la gestió automàtica d'una base de dades.

Hem fet un repàs de l'**evolució de la gestió de dades** començant pels sistemes manuals de fitxes de cartolina, passant per la gestió informatitzada amb fitxers de dades, fins als sistemes de base de dades. Hem vist que els fitxers de dades estan orientats al procés (posen l'èmfasi en el tractament de les dades, que estan disperses en fitxers dissenyats per a una aplicació determinada), mentre que les bases de dades estan orientades a les dades (les quals s'organitzen i es mantenen en un conjunt estructurat no dissenyat per a una aplicació concreta).

Per a especificar com s'organitzen les dades hem descrit les **estructures de dades** més utilitzades en els sistemes d'informació: matrius, cadenes de caràcters, registres, arbres, llistes i llistes vinculades.

Per a justificar l'ús de les bases de dades, hem analitzat els **problemes** que comporta la utilització de **sistemes de gestió de fitxers** de dades, que hem agrupat en:

- els que afecten els *fitxers*: necessitat de controlar la integritat semàntica, dificultat per a gestionar el control d'autoritzacions i falta de control de concurrència;
- i els que afecten les *dades*: redundància, inconsistència, aïllament, dificultat d'accés a les dades i dependència dels programes.

Hem esmentat els principals **objectius de les bases de dades**:

- Centralitzar la informació i integrar les dades i els programes.
- Estandarditzar una gestió de dades eficaç, universal i independent del maquinari i del sistema operatiu.
- Proporcionar un sistema d'emmagatzematge amb una interfície d'usuari fàcil de consultar i de modificar.
- Optimitzar el cost del suport d'emmagatzematge de les dades.

Les **característiques** que han de complir ens permeten manifestar que "les bases de dades permeten la integració de la informació del sistema per a evitar

redundància, sense que per això es perdin les diferents perspectives que en tenen els usuaris (vistes), i les eines de programari (SGBD) usades per a gestionar-les asseguren la independència (dels programes respecte a la representació física de les dades), la integritat i la seguretat de les dades".

Hem agrupat els **avantatges de les bases de dades** en tres blocs:

1) Els avantatges *referits a les dades*

- Disminució de la redundància.
- Prevenció de la inconsistència.
- Representació d'associacions entre les dades.
- Manteniment de la integritat.
- Subministrament de còpies de seguretat i recuperació.
- Emmagatzematge de les especificacions de la base de dades.
- Independència entre els programes i les dades.
- Compatibilitat entre formats.

2) Els avantatges *referits als usuaris* (en sistemes multiusuari)

- Aplicació eficient de restriccions de seguretat.
- Subministrament de múltiples interfícies d'usuari.
- Suport de múltiples vistes de dades.
- Disponibilitat d'informació actualitzada.
- Accés ràpid i senzill a la informació.
- Accessibilitat múltiple i simultània.
- Flexibilitat per a atendre nous requeriments.
- Coherència dels resultats.

3) Els avantatges *referits a l'organització*

- Integració de tota la informació de l'organització.
- Eficiència de l'emmagatzematge físic.
- Reducció del cost d'emmagatzematge i de manteniment.
- Reducció del cost de formació del personal.
- Economia d'escala.
- Capacitat per a establir normes.
- Reducció del temps de creació d'aplicacions.

Hem vist que la implantació d'una base de dades pot comportar certs **inconvenients**: inversió inicial elevada, implantació llarga i difícil, rendibilitat a llarg termini, absència de normalització dels SGBD comercials, problemes d'integració de bases de dades, risc de frustració dels usuaris.

Així mateix, hem comentat que hi ha **situacions en què és millor utilitzar un sistema de fitxers** en comptes d'un sistema de base de dades: si el sistema

existent està ben definit, és senzill i no ha de variar; si no cal l'accés multiusuari i simultani a les dades, o si el temps de resposta requerit per les aplicacions no es pot complir amb l'SGBD.

Hem après que els principals **elements d'un sistema de base de dades** són el contingut (les dades i la seva descripció), l'equip lògic (l'SGBD, el sistema operatiu, el programari de comunicacions, etc.), l'equip físic (l'ordinador o els ordinadors en què s'instal·la l'SGBD i en què s'emmagatzema la base de dades, i els dispositius que en permeten la gestió) i el personal humà (els usuaris que interactuen amb la base de dades).

Hem classificat els **usuaris d'una base de dades** en dues grans categories:

1) El *personal informàtic* que intervé en el desenvolupament i manteniment de la base de dades: directius, analistes, dissenyadors de bases de dades, programadors, administradors, responsables d'explotació. Aquí també hi hem inclòs usuaris sense interès en el contingut de la base de dades relacionats amb el disseny, la creació i el funcionament del programari de l'SGBD: desenvolupadors de l'SGBD i d'eines i tècnics de manteniment.

2) Els *usuaris finals*, entre els quals hem distingit els tipus següents:

a) *Usuaris habituals*, que usen assistents i sistemes visuals guiats. Un tipus específic, els *usuaris paramètrics*, que introdueixen dades de manera massiva mitjançant transaccions programades, són operadors de consola com caixers de banc o encarregats de reserves.

b) *Usuaris avançats o experts*, familiaritzats amb la major part de recursos de l'SGBD, que l'utilitzen per a implementar sistemes que executen processos propis i compleixen els seus requeriments específics. Solen ser enginyers, científics i analistes de negoci que utilitzen les dades com a suport a la presa de decisions.

c) *Usuaris ocasionals o esporàdics* que requereixen informació diferent en cada ocasió i utilitzen pocs recursos de l'SGBD. Un tipus concret és l'*usuari crític*, per l'elevat temps de resposta i la dificultat d'obtenir resultats en el format i el nivell de detall requerits, sol ser personal de gerència o *staff* amb poder decisori en l'organització.

d) *Usuaris autònoms* que mantenen bases de dades personals i tenen gran habilitat en l'ús de paquets de programari específics amb interfícies gràfiques molt amigables.

Més endavant hem definit els conceptes de **model**, **esquema** i **estat de la base de dades** d'aquesta manera:

- Un *model de dades* és una eina intel·lectual que permet aconseguir el procés d'abstracció que condueix de la part del món real que interessa estudiar (anomenada univers del discurs) al món de les dades.

- L'*esquema* és la descripció de l'estructura de la base de dades.

- L'*estat* de la base de dades és el conjunt de valors que prenen els objectes d'un esquema en un moment donat i que pot ser buit (sense dades), inicial o actual.

Per tal d'assenyalar la **relació entre aquests conceptes** hem apuntat que "la descripció de l'univers del discurs del món real mitjançant un model de dades dóna com a resultat un esquema, que pot prendre diversos estats". A més, hem definit conceptes importants per al disseny de bases de dades com *ocurrència*, *tipus*, *intensió* i *extensió*.

Els models de dades ofereixen diferents **tipus d'abstracció** per a facilitar la representació de les dades en el disseny de bases de dades i establir vincles entre els elements del model. Hem vist que la *classificació* estableix un vincle entre un *tipus* d'objecte i cada *ocurrència*, mentre que la *generalització,* l'*agregació* i *l'associació* estableixen el vincle entre *tipus* d'objectes (i, per tant, també entre les seves *ocurrències*). A més, hem assenyalat que aquestes abstraccions són síntesis conceptuals i els respectius processos inversos (la *instanciació*, *l'especialització*, la *desagregació* i la *dissociació*) són refinaments conceptuals.

Per a abstreure, esquematitzar i entendre les característiques fonamentals dels SGBD, hem analitzat l'**arquitectura dels SGBD** des de diferents enfocaments:

1) *Arquitectura operacional*. Segons com es distribueix el processament de l'SGBD en el sistema informàtic, pot ser:

a) *Arquitectura centralitzada*: un ordinador central fa tot el processament i emmagatzema les dades a les quals accedeixen els usuaris a través de terminals remots.

b) *Arquitectura client-servidor*: el processament es reparteix entre el client, que proporciona les aplicacions i interfícies d'usuari en un ordinador personal, i el *servidor*, que fa les tasques pròpies de la base de dades (emmagatzematge i accés a les dades).

2) *Arquitectura de* referència: segons els nivells d'abstracció de les dades en l'SGBD, pot ser de dos nivells o de tres nivells.

3) *Arquitectura externa* o aparent: mostra les utilitats de l'SGBD que l'usuari veu com un conjunt estructurat (processadors de consultes, generadors de formularis, d'informes, de menús…).

4) *Arquitectura interna* o funcional: mostra com l'SGBD estructura internament la seva funcionalitat en mòduls de programari per a executar funcions específiques (interpretació d'SQL, optimització de consultes, gestió de vistes, gestió de transaccions, control de concurrència, etc.).

Hem vist que els SGBD actuals han de respectar l'**arquitectura de tres nivells** (proposada per ANSI/SPARC), la qual aporta una gran flexibilitat per als canvis i estableix una divisió de la base de dades segons la perspectiva des de la qual aquesta és vista. Hem descrit aquests **nivells d'abstracció** de l'SGBD, que es corresponen amb els seus tres principals grups d'usuaris (finals, programadors i administradors), de la manera següent:

- El *nivell extern* gestiona la informació des del punt de vista individual de cada grup d'usuaris.

- El *nivell conceptual* descriu l'estructura de la base de dades per a tots els usuaris independentment de l'ordinador en què aquesta s'implementi.

- El *nivell intern* descriu la informació en funció del sistema en què s'implementarà la base de dades.

A més, hem estudiat les **correspondències** entre aquests nivells i hem assenyalat les limitacions que comporta la lligadura per a la independència de les dades.

Com a síntesi de la matèria exposada, hem presentat una visió integrada dels elements i les característiques d'un sistema de base de dades entès com el sistema que integra l'SGBD i la base de dades en un sentit ampli, que hem anomenat **estructura global del sistema de base de dades**.

Després, hem vist com s'organitzen les dades en suports d'**emmagatzematge** secundari, els quals permeten desar de manera persistent grans volums de dades que poden ser recuperades, actualitzades i processades per l'SGBD quan sigui necessari. Normalment, s'utilitzen discos magnètics en els quals la memòria es divideix en blocs. Hem definit *registre físic* (bloc o pàgina) com la unitat de transferència de dades entre el disc i la memòria principal, i *registre lògic* com la unitat de transferència entre la memòria intermèdia i el programa d'usuari.

La jerarquia d'abstracció de dades existent en un sistema de base de dades (blocs de dades – fitxers de registres – estructures de base de dades) ens ha permès explicar l'accés de l'SGBD a les dades per a satisfer una consulta. Per a això, hem mostrat l'esquema general del flux de dades i de control que segueix el procés d'execució d'una consulta feta per un programa d'usuari a l'SGBD. Hem vist que el programa d'usuari percep la base de dades com un conjunt de dades, l'SGBD la veu com un conjunt de registres emmagatzemats, l'administrador d'arxius la veu com un conjunt de blocs i l'administrador d'E/S la percep tal com és físicament en realitat.

A partir dels objectius de les bases de dades, hem deduït que les **funcions principals de l'SGBD** són la definició, la manipulació i el control de les dades.

L'anàlisi d'aquestes funcions ens ha permès determinar els **components** de programari bàsics de l'SGBD, que hem dividit en dos grups:

- El *motor*, que s'encarrega de les tasques de gestió de les dades (accepta les consultes dels clients, les processa i retorna els resultats), està format per: *llenguatges de base de dades, diccionari de dades i nucli de l'SGBD*.

- Les *interfícies*, que permeten ometre l'ús del llenguatge de base de dades, i les *utilitats i eines*, que simplifiquen les tasques dels usuaris principals.

Com a síntesi, hem presentat la correlació entre els objectius de les bases de dades, les funcions de l'SGBD i els seus components associats.

Hem classificat els **llenguatges de base de dades** que permeten manejar la base de dades segons diversos criteris:

- Segons la *funció*, tenim llenguatge de definició de dades (LDD) especialitzat en la creació de l'estructura de la base de dades i un llenguatge de manipulació de dades (LMD) especialitzat en la consulta i manteniment de les dades.

- Segons la *modalitat d'ús* del llenguatge distingim el mode interactiu (en què l'usuari introdueix a través del teclat les instruccions que són analitzades, verificades i executades pel programari) i el mode programat (en què l'usuari executa un programa d'aplicació sobre el sistema operatiu).

- Segons les *aplicacions que suporten* poden ser llenguatges d'alt nivell (declaratius o no procedimentals) o llenguatges de baix nivell (procedimentals).

- Segons el *model de dades* en què es fonamenta la base de dades. Hem destacat SQL, llenguatge relacional de consulta basat en l'àlgebra relacional.

- Segons el *tipus d'usuari* a qui van adreçats, es poden classificar en llenguatges per a usuaris informàtics experts, per a usuaris ocasionals i per a usuaris paramètrics.

Després hem presentat els tipus d'**interfícies de l'SGBD** segons el mètode utilitzat per a fer les sol·licituds (llenguatge natural, menús, formularis o gràfic) i segons la categoria principal d'usuari a qui s'adrecen (administrador o usuari paramètric).

Finalment, hem vist que el **nucli de l'SGBD** proporciona una interfície entre les dades i els programes d'aplicació dissenyats per a la seva manipulació, i fa una sèrie de tasques que permeten garantir que l'emmagatzematge de les da-

des i l'accés a aquestes siguin correctes, segurs, íntegres i eficients. També hem analitzat els mòduls de programari que incorpora el nucli:

- *Gestor de sentències*, amb components per al control de seguretat, la transformació d'esquemes, el control d'integritat i el processament de sentències (aquest últim inclou compilador de consultes, precompilador, compilador de l'LMD, lligador i optimitzador).

- *Gestor de concurrència*. Gestiona l'accés concurrent a bases de dades multiusuari i evita la pèrdua de consistència o integritat de les dades.

- *Gestor de dades*. Executa cada operació elemental contra la base de dades i interactua amb l'administrador d'arxius del sistema operatiu. Inclou processador en temps d'execució, gestor de recuperació, gestor de dades emmagatzemades i gestor de memòria intermèdia.

Bibliografia

Connolly, T. M.; Begg, C. E. (2005). *Sistemas de bases de datos* (4a. ed.). Madrid: Addison-Wesley (Pearson Educación).

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

Elmasri, R.; Navathe, S. B. (2007). *Fundamentos de sistemas de bases de datos* (5a. ed.). Madrid: Addison-Wesley (Pearson Educación).

García-Molina, H.; Ullman, J.D.; Widom, J. (2009). *Database Systems: The Complete Book* (2a. ed.). Prentice Hall.

Ramakrishnan, R.; Gehrke, J. (2002). *Database Management Systems* (3a. ed.). Boston, Massachusetts: McGraw-Hill.

Silberschatz, A.; Korth, H. F.; Sudarshan, S. (2006). *Fundamentos de bases de datos* (5a. ed.). Madrid: McGraw-Hill.

Smith, P.D.; Barnes, G.M. (1987). *Files and Databases: An Introduction*. Reading, Massachusetts: Addison-Wesley.

Ullman, J.D.; Widom, J. (1999). *Introducción a los sistemas de bases de datos* (1a. ed.). Prentice Hall.