

# Emmagatzematge i bases de dades

Ramon Costa i Pujol  
Jaume Raventós Moret

PID\_00153132



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índex

<b>Introducció</b> .....	5
<b>Objectius</b> .....	8
<b>1. Fitxers</b> .....	11
1.1. Procés d'entrada i sortida de dades. Emmagatzematge .....	13
1.2. Operacions amb fitxers .....	15
1.3. Tipus de fitxers .....	16
1.3.1. Segons la longitud dels registres .....	17
1.3.2. Segons l'ús .....	18
1.4. Organització de fitxers .....	19
1.4.1. Organització seqüencial .....	20
1.4.2. Organització seqüencial encadenada .....	20
1.4.3. Organització seqüencial indexada .....	21
1.4.4. Organització directa o aleatòria .....	22
1.5. Limitacions dels fitxers .....	24
1.6. Resum .....	25
<b>2. Bases de dades i sistemes gestors de bases de dades</b> .....	27
2.1. Conceptes bàsics .....	28
2.1.1. Bases de dades .....	28
2.1.2. Fitxers i bases de dades .....	29
2.2. Aplicacions pràctiques de BD .....	30
2.2.1. Característiques i objectius de les BD .....	31
2.3. Sistemes gestors de BD .....	32
2.3.1. Característiques i objectius dels SGBD .....	33
2.3.2. SGBD del mercat .....	34
2.4. Tipus de bases de dades .....	35
2.5. Bases de dades relacionals .....	49
2.5.1. Taules .....	49
2.5.2. Registres .....	49
2.5.3. Camps .....	50
2.5.4. Claus .....	50
2.5.5. Relacions i claus foranes .....	51
2.5.6. Tipus de dades .....	53
2.5.7. Consultes .....	55
2.6. Llenguatges de bases de dades relacionals .....	56
2.6.1. Llenguatge de definició de dades (DDL) .....	57
2.6.2. Llenguatge de manipulació de dades (DML) .....	58
2.6.3. Eines d'interfície .....	59
2.6.4. Programació de bases de dades .....	60

2.7. Resum .....	61
<b>3. Disseny de bases de dades.....</b>	<b>65</b>
3.1. Models de dades: conceptes bàsics .....	66
3.2. Definició conceptual d'una BD .....	68
3.2.1. Model de dades: el model entitat-relació .....	69
3.2.2. Entitats .....	70
3.2.3. Ocurrencia .....	70
3.2.4. Atribut .....	71
3.2.5. Clau .....	71
3.2.6. Relacions .....	71
3.3. Del model conceptual al lògic o relacional .....	73
3.4. Normalització de bases de dades .....	75
3.4.1. Graus de normalització .....	76
3.4.2. Primera forma normal (1FN) .....	76
3.4.3. Segona forma normal (2FN) .....	77
3.4.4. Tercera forma normal (3FN) .....	77
3.4.5. Exemples .....	77
3.4.6. Altres formes normals .....	80
3.5. Casos pràctics "Disseny i creació de bases de dades" .....	80
3.5.1. Cas pràctic 1: Base de dades "Centre de formació" .....	81
3.5.2. Cas pràctic 2: Agència de lloguer de cotxes .....	85
3.6. Resum .....	89
<b>4. Bases de dades documentals.....</b>	<b>90</b>
4.1. Cerca d'informació .....	91
4.2. Categories de BD segons la informació que contenen .....	91
4.3. Tipologia de BD segons el mode d'accés .....	92
4.4. Tipologia de BD segons la cobertura documental .....	92
4.5. Tipologia de BD segons el model de tractament documental .....	93
4.6. Resum .....	94
<b>Exercicis d'autoavaluació.....</b>	<b>95</b>
<b>Solucionari.....</b>	<b>99</b>
<b>Bibliografia.....</b>	<b>103</b>

## Introducció

L'aplicació pràctica dels ordinadors i dels sistemes informàtics és el **tractament de la informació**. Per això, al conjunt de sistemes informàtics i mecanismes de comunicació se l'anomena *tecnologies de la informació i la comunicació* (TIC).

Tots els programes informàtics gestionen dades, que són captades, analitzades, tractades i presentades perquè l'usuari pugui prendre decisions o dur a terme altres accions sobre elles.

Per tant, és necessari poder **emmagatzemar les dades** i informacions que l'ordinador i els sistemes d'informació han de tractar.

En un ordinador hi ha dos tipus de **memòria principal**: RAM i ROM.

- La informació de la **memòria ROM** no s'esborra mai. Emmagatzema el test de fiabilitat de l'ordinador (un conjunt d'instruccions per mitjà de les quals comprova l'estat dels seus components cada vegada que s'encén), les rutines d'iniciació i engegada, i la BIOS.

### Sistema bàsic d'entrada i sortida

La BIOS (*basic input / output system*) és un petit conjunt d'instruccions que utilitza el processador per a trobar el sistema operatiu i carregar-lo en la memòria RAM. També serveix per a gestionar el flux de dades entre el sistema operatiu i els dispositius de l'ordinador.

Aquesta memòria és **permanent**, és a dir, la informació no es perd en desconectar l'ordinador. Les instruccions i les dades de la ROM continuen allà quan s'apaga l'ordinador.

- En la memòria **RAM** es guarda temporalment la informació dels programes i processos que s'executen. S'hi pot llegir i escriure. Perquè un programa es pugui executar s'ha de carregar (emmagatzemar) en la memòria RAM.

Aquesta memòria és volàtil, és a dir, en el moment en què s'apaga l'ordinador es perd tota la informació que hi ha emmagatzemada.

Aquests tipus de memòria, doncs, no aconsegueixen un requisit imprescindible per a emmagatzemar dades que és *l'emmagatzematge permanent de gran quantitat d'informació*, per la qual cosa són necessaris **dispositius d'emmagatzematge** extern com, per exemple, els discos magnètics, els CD-ROM, els DVD o els llapis de memòria (*pen drive* o *USB flash drive*).

### Vegeu també

Per saber què comporta el tractament de la informació en un sistema informàtic podeu consultar el subapartat "Els ordinadors tracten informació" en el mòdul "Ordinadors i sistemes operatius".

### Vegeu també

Per tenir més detalls sobre el funcionament de la memòria RAM i ROM podeu consultar el subapartat "Estructura bàsica d'un ordinador" del mòdul "Computadors i sistemes operatius".

### ROM

ROM (*read only memory*) és memòria de només lectura.

### RAM

RAM (*random access memory*) és memòria d'accés aleatori.

### Vegeu també

Per conèixer els diferents dispositius d'emmagatzematge extern podeu consultar el subapartat "Quines són les unitats d'emmagatzematge més corrents?" de l'apartat "Suports físics d'informació" del mòdul "Ordinadors i sistemes operatius".

D'altra banda, aquesta informació que s'ha d'emmagatzemar i gestionar s'ha d'estructurar de manera que la puguin tractar fàcilment tant els ordinadors i programes informàtics com, sobretot, els seus usuaris.

És per això que hi ha diferents **estructures de dades** que permeten emmagatzemar aquesta informació: els fitxers i les bases de dades. Aquest mòdul tracta els principals aspectes relacionats amb aquestes dues maneres d'organitzar la informació.

- En un **fitxer** s'emmagatzema informació que fa referència al mateix tema d'una manera estructurada per poder treballar amb les dades de manera individual.

Cada fitxer es compon d'estructures més simples anomenades **registres**. Cada registre està format per **camp**s que contenen informació referent a un element o una característica en particular dins del fitxer.

Les operacions principals que es poden dur a terme sobre un fitxer són crear-lo, llegir-lo i actualitzar la informació (registres i camps), ordenar i consultar els registres, eliminar registres i generar informes.

El tipus d'organització d'un fitxer (seqüencial, directa, etc.) condiciona les operacions que s'hi poden fer i, per tant, és determinat per l'aplicació que se li doni.

Ateses les seves limitacions els fitxers han estat substituïts, en la majoria d'aplicacions, per bases de dades.

- Les **bases de dades** van sorgir per a intentar resoldre els problemes que tenien els fitxers (inconsistència de les dades, redundància, rigidesa en les cerques, dependència dels programes, etc.).

Una base de dades és una col·lecció de dades organitzades de manera que es pugui accedir als seus continguts i aquests es puguin administrar i actualitzar amb facilitat.

Un **sistema gestor de bases de dades** (SGBD) és el conjunt de programari destinat a la creació, gestió, control i manipulació de la informació emmagatzemada en una base de dades.

L'SGBD disposa d'un **llenguatge de definició de dades** (DDL) que permet definir l'esquema de la base de dades, un **llenguatge de manipulació de dades** (DML) que facilita la gestió de la informació emmagatzemada, i una **interfície d'usuari** que permet accedir al sistema i treballar amb comoditat.

Hi ha tipus molt diversos de bases de dades. Aquesta assignatura se centra en les relacionals (el model més utilitzat), que permeten gestionar informació estructurada, i les documentals, que gestionen informació no estructurada.

- Una **base de dades relacional** està formada per **taules**, que són estructures de **files** (els registres d'informació) i **columnes** (els **camps** d'informació dels registres).  
El disseny i la creació d'una base de dades relacional consisteix a definir el **model relacional**, és a dir, les taules, els seus camps i les relacions entre taules, principalment.  
Abans d'aquest disseny, però, s'aconsella elaborar el **model conceptual**, que permet plasmar en un esquema entitat-relació les diferents informacions (entitats, atributs i interrelacions) que es volen emmagatzemar i gestionar en la futura base de dades relacional.
- Una **base de dades documental** es caracteritza perquè cada registre es correspon amb un document de qualsevol tipus (publicació, document gràfic o sonor, etc.) o amb la seva referència.  
La informació que conté s'estructura en diferents camps: n'hi ha que es refereixen a la descripció formal del document, d'altres tracten sobre el seu contingut temàtic, i fins i tot, un camp pot guardar el document.  
Segons la informació continguda i la referència al document corresponent, es pot classificar en: base de dades de **text complet** (conté els propis documents), **arxiu electrònic d'imatges** (conté imatges dels documents originals) i base de dades **referencials** (conté referències per a localitzar els documents originals).

## Objectius

Els **objectius generals** que l'estudiant pot assolir són els següents:

1. Identificar la necessitat de les bases de dades (BD), i els seus avantatges en relació amb els fitxers.
2. Conèixer els elements d'un sistema gestor de bases de dades (SGBD) i les seves principals funcionalitats.
3. Adquirir una visió general de les tipologies de BD, especialment les relacionals i les documentals.
4. Conèixer els fonaments del disseny de BD relacionals.

Aquests objectius generals es desglossen en els **objectius específics** següents:

1. Identificar la necessitat que les aplicacions informàtiques emmagatzemin i gestionin la informació.
2. Definir els conceptes de fitxer, registre, camp, registre lògic, contra registre físic i bloc.
3. Identificar el procés d'intercanvi d'informació entre la memòria externa i la interna d'un ordinador.
4. Definir les principals operacions que es poden dur a terme amb un fitxer.
5. Classificar els fitxers segons dos criteris: la longitud dels registres i el seu ús.
6. Descriure les diferents maneres en què es pot organitzar un fitxer.
7. Enumerar les limitacions d'ús i aplicació dels fitxers.
8. Definir les principals funcionalitats de les BD.
9. Enumerar aplicacions pràctiques de les BD.
10. Definir les principals funcionalitats i components o elements d'un SGBD.
11. Identificar diferents SGBD existents al mercat.



- 12.** Enumerar els principals tipus de BD i la seva manera d'organitzar les dades.
- 13.** Descriure els principals conceptes de les BD relacionals i l'estructuració de les seves dades.
- 14.** Distingir entre els llenguatges de definició de les dades (DDL) i els de manipulació d'aquestes dades (DML).
- 15.** Identificar les opcions disponibles per a treballar amb una BD des d'un llenguatge de programació.
- 16.** Identificar els principals elements d'un model conceptual de BD segons el model entitat-relació.
- 17.** Definir una BD relacional a partir d'un model entitat-relació.
- 18.** Enumerar les diferents opcions per a normalitzar el disseny d'una BD.
- 19.** Conèixer els conceptes bàsics de bases de dades documentals (BDD).
- 20.** Enumerar els diferents sistemes de recuperació de la informació en BDD.
- 21.** Distingir entre diverses categories de BDD.
- 22.** Identificar tres tipus diferents de modes d'accés a la informació d'una BDD.
- 23.** Distingir entre les BDD centrades en un sol tipus de document i les que incorporen diversos tipus de documents.
- 24.** Diferenciar les BDD segons el model de tractament documental.



## 1. Fitxers

Un fitxer, també anomenat *arxiu*, és un conjunt ordenat de dades que mantenen entre si una relació lògica i s'emmagatzema en un suport d'informació per a la comunicació amb l'ordinador.

### Exemple d'estructura d'un fitxer

El fitxer que emmagatzema les dades dels abonats a places d'aparcament d'un garatge es pot descriure gràficament de la manera següent:

Nom d'abonat	NIF d'abonat	Matrícula de vehicle	Tipus de vehicle	Plaça d'aparcament	....
Ramon Pujol	46587875	2587-CDD	monovolum	25	
Jaume Moret	46585965	5874-AAF	furgoneta	36	
Marc Costa	45898745	2698-CCE	tot terreny 4 × 4	24	
Marta Pou	58954785	2598-BDA	motocicleta	14	
Albert Raventós	54785478	5698-BEE	motocicleta	05	
Montse Molina	58745854	1985-CCA	monovolum	32	
Núria Camps	47858585	8847-TKM	ciclomotor	18	
....					

En un fitxer s'emmagatzema informació que fa referència al mateix tema d'una manera estructurada per a poder treballar amb les dades de manera individual.

Els fitxers es componen d'estructures més simples anomenades **registres**. Tots els registres d'un fitxer són del mateix tipus.

Cada registre està format per **campos** que contenen informació referent a un element o característica en particular dins del fitxer.

### Fitxer de propietaris de places d'aparcament en un garatge

Gràficament, podríem descriure un fitxer de la manera següent:

Propietari	NIF	Matrícula	Plaça d'aparcament	....
Ramon Pujol	46587875	2587-CDD	25	

Propietari	NIF	Matrícula	Plaça d'aparcament	....
Albert Costa	45898745	2698-CCE	24	
Toni Pou	58954785	2598-BDA	14	
Marc Cartagena	54785478	5698-BEE	05	
Montse Monte	58745854	6985-CCA	36	
Núria Camps	47858585	B-5847-TX	14	
Pere Soler	46585965	5874-AAF	36	
....				
....				

Dins d'un fitxer determinat, els registres s'identificaran per un camp o un conjunt de camps anomenats **clau**. Aquests identificadors serviran per a distingir cada registre dels altres, i també per a facilitar la localització ràpida dels registres dins dels fitxers (per exemple, el NIF). Un fitxer pot tenir una clau, diverses o cap.

La majoria de programari (processadors de text, gestors de fulls de càlcul...) també treballen amb fitxers per tal de poder emmagatzemar la informació que gestionen. Per exemple, qualsevol document elaborat amb un processador de text es guarda com un fitxer, o qualsevol foto o vídeo que emmagatzemem en el nostre disc dur, també es guarda com un fitxer. En aquests casos, però, la informació es visualitza per pantalla, d'una manera diferent a com s'ha guardat. El programa (processador de text, sistema operatiu...) s'encarrega de llegir el contingut d'aquell fitxer i mostrar-lo d'una manera entenedora per a l'usuari.

Els arxius s'emmagatzemen en els dispositius de **memòria massiva**, també anomenats *de memòria auxiliar*, com, per exemple, els discos durs, els disquets, CD, DVD, llapis de memòria (*pen drive* o *flash memory*) i altres dispositius de memòria amb connexió USB.

#### Diverses denominacions per a l'emmagatzematge permanent de dades

Hi ha moltes maneres d'anomenar un suport físic digital d'emmagatzematge permanent de dades segons els termes utilitzats per a descriure'l: dispositiu (unitat o suport) d'emmagatzematge (memòria) auxiliar (secundari/ària o extern/a o massiu/va). Així, és correcte usar, entre d'altres, qualsevol de les denominacions genèriques següents:

- dispositiu d'emmagatzematge secundari
- dispositiu de memòria auxiliar
- dispositiu de memòria massiva
- suport d'emmagatzematge extern
- suport d'emmagatzematge massiu
- unitat de memòria externa
- unitat d'emmagatzematge secundari
- etc.

#### Per a saber-ne més

Per a conèixer les característiques dels diferents tipus de suports d'emmagatzematge podeu consultar el subapartat 1.4.6 del mòdul "Ordinadors i sistemes operatius".

Un element que s'ha de considerar en els fitxers, tal com es comentarà més endavant, és la seva organització, distingint-ne quatre tipus: **seqüencial**, **seqüencial encadenada**, **seqüencial indexada** i **directa o aleatòria**. Aquest aspecte, tal com es comenta més endavant, està relacionat amb el tipus d'accés als fitxers.

En els apartats següents també es presenten les operacions principals que es poden fer amb els fitxers i les diferents tipologies en què es poden classificar.

### 1.1. Procés d'entrada i sortida de dades. Emmagatzematge

Un fitxer està compost per un conjunt de registres, tots del mateix tipus (per exemple, les dades dels propietaris dels cotxes venuts en un concessionari) i cada registre està compost per camps que contenen informació referent a un element o característica del registre (per exemple, el número de la matrícula, model i marca del cotxe, NIF del propietari, data de la venda, etc.).

L'**accés als dispositius** de memòria massiva on s'emmagatzemen els fitxers pot ser de dos tipus: seqüencial i directe.

- En els **suports d'accés seqüencial** (per exemple, una cinta) per a accedir al registre  $n$  s'han de llegir, un a un, els  $n - 1$  registres anteriors en l'ordre en què estan escrits.
- En els **suports d'accés directe** (per exemple, un disc dur o un disquet) es pot accedir directament a un registre físic solament donant-ne l'adreça física.

Es diferencia entre **registre lògic**, entès com la informació corresponent a un dels elements del fitxer, i **registre físic** (o bloc), que és el conjunt d'informació que es pot llegir i escriure alhora en el suport físic.

Els registres físics s'emmagatzemen en el dispositiu corresponent, i el sistema operatiu és l'encarregat d'escriure i llegir les dades que componen el fitxer. El sistema operatiu transporta, cada vegada que s'accedeix al dispositiu (per a llegir o escriure-hi), una quantitat fixa d'informació (bloc o registre físic) que depèn de les característiques físiques o del maquinari.

En general, un bloc pot tenir un nombre variable de registres lògics, és a dir, es poden transferir diversos registres lògics en una sola operació d'escriptura/lectura. Aquest fet rep el nom de **bloqueig**. El número de registres lògics continguts en un bloc (longitud del bloc) rep el nom de **factor de bloqueig**.

El bloqueig de registre aporta dos grans avantatges:

#### Per a saber-ne més

Per a saber com s'organitzen i com és l'accés als fitxers podeu consultar el subapartat 1.4 d'aquest mòdul.

#### Procés d'entrada i sortida de dades

Els processos d'entrada i sortida de dades corresponen a les operacions d'escriptura i lectura, respectivament, mitjançant les quals s'afegeix, es modifica i/o s'accedeix a la informació dels registres.

#### Per a saber-ne més

Per a saber més sobre el procés d'escriptura i lectura de dades en un fitxer podeu consultar el sistema de fitxers i la gestió d'E/S del sistema operatiu en el subapartat 2.2 del mòdul "Ordinadors i sistemes operatius".

- Més velocitat en els processos d'entrada/sortida (E/S), ja que com més gran sigui el factor de bloqueig menys accessos caldrà fer en el dispositiu físic per a tractar la informació.
- Millor aprofitament de la capacitat de suport de l'emmagatzematge.

L'**adreça lògica** (adreça programari) d'un registre és la posició relativa que ocupa en el fitxer, mentre que l'**adreça física** és la posició real on es troba el registre en el suport d'informació (adreça maquinari).

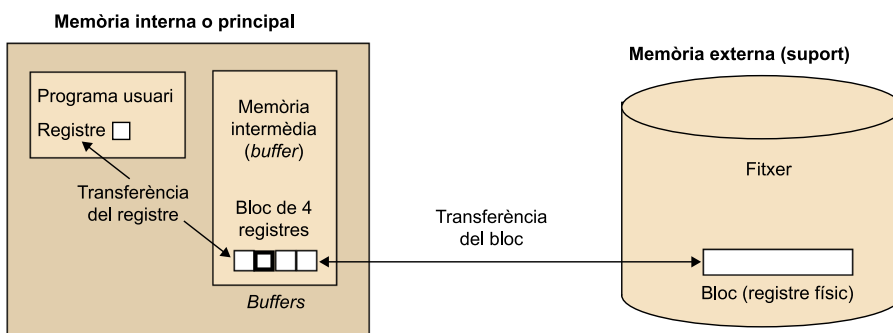
En el fitxer, els registres s'apareixen a l'usuari de manera lògica, és a dir, ordenats segons la seva adreça lògica. En canvi, l'ordre físic dels registres d'un fitxer al disc pot no tenir cap relació amb la informació que conté.

El **sistema operatiu** és el responsable de realitzar la transformació de l'**adreça lògica**, utilitzada en els programes, en l'**adreça física** amb què s'emmagatzema en el dispositiu.

Des d'un programa s'accedeix a un fitxer per a llegir i modificar un dels seus registres o escriure-hi. En llegir, es transfereix de bloc a bloc (registre físic) la informació del fitxer a una àrea de memòria principal anomenada *memòria intermèdia (buffer)*.

Des d'aquesta memòria intermèdia es transfereix la informació que el programa pot programar. I de la mateixa manera, el programa pot transferir informació des d'aquesta zona al fitxer, modificant-ne el contingut.

Esquema bàsic d'entrada i sortida



Aquí el factor de bloqueig (nombre de registres continguts en un bloc) és quatre.

Des de la memòria intermèdia, les dades són processades pel programa.

Així, el programa transfereix informació des d'aquesta memòria al fitxer, amb la qual cosa se n'actualitza el contingut.

## 1.2. Operacions amb fitxers

Els programes, o aplicacions informàtiques, accedeixen als fitxers per dur a terme un conjunt d'**operacions** amb la informació emmagatzemada. A continuació es descriuen les operacions més habituals:

**a) Creació.** La primera operació que s'ha de fer sobre un fitxer. Consisteix a definir les característiques de les dades. A partir d'aquest moment, ja es poden afegir registres al fitxer i treballar-hi.

**b) Lectura.** Consisteix a recuperar la informació del fitxer al nivell del registre.

**c) Manteniment o actualització.** La modificació d'un fitxer inclou tres accions possibles:

- **Inserció** d'un registre nou.
- **Modificació** d'un registre, canviant-ne un o més valors.
- **Eliminació** d'un registre.

**d) Ordenació.** Consisteix a alterar l'ordre dels registres d'un fitxer segons un o més criteris, segons els valors d'un o més camps, i que reben el nom de claus d'ordenació.

**e) Cerca.** Operació que consisteix a localitzar dins d'un fitxer un registre concret. Hi ha diferents procediments per a dur a terme la cerca d'un o més registres en un fitxer:

- **Cerca seqüencial.** Representa examinar cadascun dels registres del fitxer fins que es troba el sol·licitat.
- **Cerca dicotòmica o binària.** Aplicable només en el cas de fitxers ordenats. Consisteix a dividir els fitxers en subfitxers cada vegada més i més petits. Es comença llegint el registre del mig i, si el valor que s'ha de cercar és superior, es descarten els registres de la primera meitat del fitxer. Si el valor que se cerca és inferior, es descarten els de la segona meitat del fitxer. Amb els registres triats es repeteix el procés de cerca, fins a trobar el registre buscat o comprovar que no hi és.
- **Cerca per blocs.** Amb aquest mètode es realitza el mateix recorregut sobre les claus que en un fitxer seqüencialment indexat. En primer lloc, es determinarà el bloc en què es troba el registre. Per a fer-ho, es llegeix l'últim registre de cada bloc, fins a trobar el cercat o un de més gran. En aquest darrer cas, es passarà a buscar el registre en el bloc anterior, fent una cerca seqüencial.

### Per a saber-ne més

Per a tenir més informació sobre l'ordenació de fitxers podeu consultar l'apartat "Organització de fitxers" en aquest mòdul.

f) **Fusió.** En aquesta operació s'uneixen dos o més fitxers per a integrar-los en un de sol.

g) **Divisió.** Operació contrària a la fusió, divideix la informació d'un fitxer en dos o més.

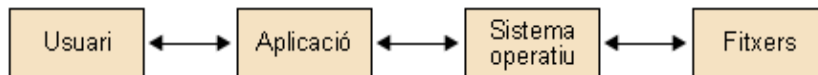
h) **Generació d'informes** a partir de dades contingudes en el fitxer. Els informes poden ser impresos, per pantalla, en format web...

i) **Destrucció.** Eliminació del fitxer del dispositiu corresponent.

Les aplicacions informàtiques i els sistemes operatius inclouen una sèrie de programes útils per a dur a terme les operacions bàsiques amb fitxers: creació, eliminació, lectura, etc. Aquests programes reben el nom de **sistemes de gestió de fitxers**.

En general, un fitxer utilitzat per un usuari des d'un llenguatge d'alt nivell, per exemple, Visual Basic, C++ o Java, no el gestiona directament el mateix programa, sinó el sistema operatiu, responsable de fer els accessos necessaris al dispositiu en què s'arxiva i de transferir la informació sol·licitada del fitxer al programa i a la inversa.

#### Gestió del sistema operatiu



### 1.3. Tipus de fitxers

Els fitxers es poden classificar segons dos criteris principals:

- Segons la longitud dels registres que contenen. En aquest cas, es distingeix entre els fitxers de longitud fixa, els de longitud variable, els delimitats i els indefinits.
- Segons l'ús que se'n farà. Es poden classificar entre fitxers permanents i temporals, i en cadascun d'aquests casos es distingeix entre fitxers mestres o de situació, fitxers constants o fitxers històrics i intermedis, de maniobres o de resultats.

A continuació, es descriuen les característiques principals de cadascun d'aquests tipus de fitxers.



### 1.3.1. Segons la longitud dels registres

Els registres que formen part d'un fitxer poden tenir la mateixa longitud o una de diferent tant si es deu a l'existència de camps de longitud variable com al fet que contenen camps de longitud fixa però que es repeteixen un nombre variable de vegades, com per totes dues causes.

Seguint aquest criteri, els fitxers es poden classificar o distingir entre els tipus següents:

- **Fitxers de longitud fixa.** Són aquells en què la suma dels caràcters de tots els camps dels registres és constant. Tots els registres, per tant, tenen la mateixa longitud.

#### Fitxers de longitud fixa

R	A	M	O	N					C	O	S	T	A					3	6	5	4	-	C	D	A			
P	E	R	E						P	U	J	O	L					B	-	2	5	4	5	-	T	X		
...																												
.																												
...																												
.																												
...																												

- **Fitxers de longitud variable.** Són els fitxers en què cada registre pot tenir una longitud diferent en el fitxer, oscil·lant entre una petita variació entre un mínim i un màxim. En aquests casos, el sistema reserva una paraula a l'inici de cada registre on anota la seva longitud.

#### Fitxers de longitud variable

#18#	R	A	M	O	N	C	O	S	T	A	3	6	5	4	-	C	D	A	#19	P	E	R	E				
P	U	J	O	L	S	B	-	2	5	4	5	-	T	X	#	-	-	-	-	-	-	-	-				
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

El número que acompanya el símbol # indica la longitud de cada registre.

- **Fitxers delimitats.** En aquests tipus de fitxers, la longitud del registre és variable i no es pot saber en quina mesura difereixen els uns dels altres. El sistema inclou un caràcter especial per a indicar el final del registre. En aquests casos, es diu que els fitxers són de tipus text.

#### Fitxers delimitats

R	A	M	O	N	C	O	S	T	A	3	6	5	4	-	C	D	A	;	P	E	R	E	P
U	J	O	L	S	B	-	2	5	4	5	-	T	X	;	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

En aquest cas, el caràcter punt i coma (";") indica on acaba cada registre.

- **Fitxers indefinits.** Són aquells en què la longitud és totalment variable. En aquests casos, el sistema operatiu no realitza cap gestió sobre la longi-

tud dels registres del fitxer. Serà el programa l'encarregat de localitzar el principi i el final de cada registre.

### Fitxers indefinits

R	A	M	O	N	C	O	S	T	A	3	6	5	4	-	C	D	A	P	E	R	E	P	U
J	O	L	S	B	-	2	5	4	5	-	T	X	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

### 1.3.2. Segons l'ús

Segons l'ús que es fa d'un fitxer, els podem classificar en diferents tipus. Per a poder organitzar i dissenyar un fitxer és imprescindible saber quina funció exercirà.

a) **Fitxers permanents.** Contenen la informació necessària per al funcionament d'una aplicació. Tenen una vida llarga i normalment no es poden generar de manera immediata a partir d'altres fitxers. Es distingeixen tres tipus:

- **Fitxers mestres o de situació.** Contenen informació que reflecteix l'estat actual de les dades. S'actualitzen periòdicament per a adaptar-se a cada situació nova. Els registres es modifiquen molt sovint, però l'estructura no varia. En la majoria de casos o aplicacions informàtiques, estan orientades a actualitzar el fitxer mestre o a obtenir resultats a partir d'aquest fitxer.
- **Fitxers constants.** Mantenen dades fixes per a l'aplicació. Aquestes dades són pràcticament inamovibles (tenen poques modificacions) i s'utilitzen com a fitxers de consulta.
- **Fitxers històrics.** Contenen dades que van ser actuals en temps anteriors. Estan constituïts per registres que reuneixen cronològicament les diferents modificacions que ha experimentat el fitxer en el temps. Es conserven per tal de poder reconstruir situacions anteriors. Algunes vegades poden estar formats, simplement, pels registres esborrats del fitxer mestre.

#### Exemple fitxers mestres

Un exemple de fitxer mestre seria les dades dels clients d'una cadena de supermercats que tenen la seva targeta de fidelització.

#### Exemple fitxers constants

Un fitxer que guardi la relació dels codis d'identificació de categories de productes seria un exemple de fitxer amb dades constants.

#### Exemple fitxers històrics

Un exemple d'aquest tipus de fitxers seria la llista de tots els clients que s'han donat de baixa de la cadena de supermercats que tenien la seva targeta de fidelització.

b) **Fitxers temporals.** Aquests fitxers contenen informació necessària per a un procés específic dins d'una aplicació. Es generen a partir de les dades dels fitxers permanents i s'utilitzen per a obtenir resultats o actualitzar-ne la informació. Una vegada feta l'operació, s'eliminen. Es distingeixen tres tipus:

- **Fitxers intermedis.** Generats a partir dels resultats d'un programa i utilitzats com l'entrada a un altre dins de la mateixa tasca o procés. Només es fan servir per a passar informació d'un procés a un altre.

### Exemple fitxers intermedis

Un exemple seria un programa que necessités fer una operació de càlcul amb les comandes de tots els clients d'una àrea geogràfica concreta. Es podria generar el fitxer amb aquestes dades a partir del fitxer mestre, dur a terme el tractament sobre aquest fitxer intermedi i eliminar-lo posteriorment.

- **Fitxers de maniobres.** Aquest tipus de fitxers s'utilitzen per a no perdre la informació generada per un procés que no es pot conservar, per falta d'espai, en la memòria principal.

### Exemple fitxers de maniobres

Un exemple d'aplicació d'un fitxer de maniobres seria el d'un procés d'una entitat financera que volgués calcular tots els interessos que ha cobrat a tots els clients durant els darrers 3 anys.

- **Fitxers de resultats.** Es generen a partir dels resultats finals d'un procés que es transferiran a un dispositiu de sortida.

### Exemple fitxers de resultats

Un exemple seria un fitxer d'impressió amb el llistat de tots els clients del supermercat que han fet compres superiors a un import determinat durant els dos mesos anteriors, i que després es transferirà a la impressora.

## 1.4. Organització de fitxers

L'ordinador ha de tenir accés als fitxers creats pels usuaris, tant per a llegir-hi la informació emmagatzemada com per a gravar-la o modificar-la.

L'accés a un fitxer està íntimament relacionat amb la seva organització. Aquesta organització indica com estan disposats els registres en el **suport material** per a aconseguir-ne una utilització més eficient.

En crear un fitxer, s'especifica quina serà la seva organització, atès que determinarà el tipus d'accés que podrem utilitzar.

Hi ha **dos tipus d'accés** a un fitxer:

- **Accés seqüencial.** Per a accedir al registre  $n$ , primer hem de recórrer als  $n-1$  anteriors en el mateix ordre en què es van escriure fins a trobar el registre cercat.
- **Accés relatiu.** A partir d'una clau (identificador) es pot accedir directament al registre sense haver de recórrer als anteriors.

Els tipus d'organització de fitxers són bàsicament quatre: seqüencial, seqüencial encadenada, seqüencial indexada i directa. En aquest apartat es descriuen cadascun d'aquests quatre tipus d'organització.

#### Per a saber-ne més

Per a tenir més informació sobre els registres en el suport material podeu consultar l'apartat "Suports físics d'informació" en el mòdul "Ordinadors i sistemes operatius".

#### Clau d'un fitxer

La clau (o identificador) d'un fitxer és un camp o un conjunt de camps que identifiquen de manera única cadascun dels registres d'aquest. Un exemple de clau és el NIF, tal com es mostra en l'exemple d'estructura d'un fitxer que podeu veure al principi de l'apartat "Fitxers".

### 1.4.1. Organització seqüencial

En aquest primer tipus d'organització, els registres s'emmagatzemen de manera **contínua**, un a continuació de l'altre, sense espai entre ells i sense cap índex que n'indiqui les adreces.

Aquesta és l'única organització que es pot gestionar en un dispositiu no direccionable com, per exemple, les cintes magnètiques, la qual cosa no implica que no es pugui utilitzar en suports d'accés directe.

Aquesta organització se sol utilitzar amb fitxers en què en cada procés s'ha d'accedir a la major part dels registres. Tenen l'**avantatge** que aprofiten molt bé l'espai, són senzills d'utilitzar i es poder fer servir amb dispositius seqüencials.

El seu principal **inconvenient** és la manca de flexibilitat i la lenta velocitat d'accés, per la qual cosa no es recomanen per a ser utilitzats en processos interactius.

En aquest tipus de fitxers es poden realitzar les operacions d'afegir registres al final del fitxer, i de consultar-los. En canvi, per a inserir, modificar o esborrar registres, necessiten utilitzar fitxers temporals.

#### Exemple d'organització seqüencial

Un exemple d'aplicació d'aquesta organització seria un fitxer que recollís i emmagatzemés, en temps real, les condicions ambientals (temperatura, humitat...) a intervals de 10 segons. Aquesta informació es podria tractar, posteriorment, si es volgués generar alguna mitjana.

### 1.4.2. Organització seqüencial encadenada

Els registres d'un fitxer amb aquesta organització emmagatzemen, a més de la seva informació, un **punter** amb l'adreça del registre següent, segons l'ordre lògic del fitxer. Des d'aquest punt de vista lògic, el fitxer s'ordenarà segons el valor d'una clau, si bé els registres es col·loquen en adreces físiques totalment arbitràries. Els punters garanteixen la seqüència lògica del fitxer.

Aquesta organització és adequada en fitxers que utilitzen processos interactius, amb una actualització freqüent però en què cada operació afecta pocs registres.

Presenten l'**avantatge** d'una gran flexibilitat, ja que s'hi poden realitzar tota mena d'operacions (afegir, consultar, inserir, modificar i esborrar), però en canvi tenen l'**inconvenient**, igual que els fitxers amb organització seqüencial, que per a accedir a un registre concret, cal seguir, de manera seqüencial tots els registres segons els punters.

#### Punter d'un registre

Un punter (o apuntador) és una variable el valor de la qual és l'adreça física on s'emmagatzema una dada. Es diu que la variable  $p$  de tipus punter (que conté l'adreça on s'emmagatzema el valor  $v$ ) *apunta a v*. Aquesta marca interna que té un registre sempre apunta al registre lògic actiu (és a dir, al registre lògic que es processa en l'operació següent del fitxer) i s'incrementa automàticament cada vegada que es processa un registre (es llegeix o s'escriu).

### Exemple d'organització seqüencial encadenada

Un fitxer d'exemple amb una organització seqüencial encadenada seria la llista dels socis d'una entitat esportiva de volum mitjà en què no calgui dur a terme gaires operacions sobre aquestes dades.

### 1.4.3. Organització seqüencial indexada

Un fitxer amb organització seqüencial indexada està compost de **dues zones**:

- La zona de **registres**, que conté tots els registres ordenats segons el valor d'alguna clau (un o més camps del registre que l'identifiquen). Es pot considerar una estructura seqüencial pura.
- La zona d'**índexs** formada per un nombre de registres inferior al total de registres del fitxer.

Els registres de la zona d'índexs tenen una estructura particular que no té res a veure amb els registres reals del fitxer. Tenen el **camp clau** (que conté alguns valors de la clau del fitxer) i el **camp adreça** (que conté l'adreça d'un registre del fitxer).

La zona de registres es considera dividida en una sèrie de **trams lògics** o **segments**, cada un format per registres consecutius. Per a cada tram en la zona de registres, hi ha un registre a la zona d'índexs, que contindrà en el seu camp clau el valor de la clau de l'últim registre del tram, i en el camp adreça, l'adreça del primer registre del tram.

### Exemple de fitxer amb organització seqüencial indexada

Exemple de fitxer amb organització seqüencial indexada

Zona d'índexs		Zona de registres			
Clau	Adreça	Núm. reg.	NIF	Nom	...
34874644	1	1	32564736		
45674635	4	2	34563546		
47657462	7	3	34874644		
59263527	10	4	44756575		
		5	45673653		
		6	45674635		
		7	45674651		
		8	46587689		
		9	47657462		
		10	56734521		
		11	58374632		
		12	59263527		

Segment I (registres 1-4)  
Segment II (registres 5-7)  
Segment III (registres 8-10)  
Segment IV (registres 11-12)

En la zona d'índexs, el camp clau correspon al NIF, que és la clau del fitxer a la zona de registres, i el camp adreça correspon al número de registre del fitxer en la zona de registres.

Una analogia d'aquesta organització és l'índex d'un llibre de lectura.

L'**avantatge** d'aquest tipus d'organització és que resulta molt útil quan cal combinar consultes a registres concrets i el processament seqüencial de tot el fitxer, ja que permet l'accés directe als registres en determinades aplicacions i, seqüencialment, en d'altres.

El seu principal **inconvenient** és la impossibilitat d'introduir nous registres o actualitzacions en els registres que hi ha al fitxer sense haver de dur a terme una reorganització amb les modificacions efectuades.

Una solució a aquest problema és reservar una zona de memòria complementària, on s'emmagatzemin els registres nous i que s'acostuma a anomenar zona de desbordament o cicle de treball.

Una millora per a aquesta estructura consisteix a utilitzar l'organització anomenada *seqüencial indexada encadenada* (una barreja de seqüencial indexada i seqüencial encadenada).

#### **Exemple d'organització seqüencial indexada**

Una aplicació d'aquesta estructura organitzativa seria la d'un fitxer que contingués totes les dades de l'estoc de productes d'un magatzem, consultat de manera molt freqüent, però que s'actualitza un cop cada setmana.

#### **1.4.4. Organització directa o aleatòria**

En aquest tipus d'organització, no hi ha cap relació lògica entre els registres i la seva ubicació física.

Cada registre se situa en una adreça de memòria que es calcula per a cadascun aplicant una fórmula o algorisme matemàtic. Aquests mètodes prenen el valor d'un camp del registre, apliquen una transformació i n'obtenen l'adreça.

Quan es detecta que l'adreça assignada a un registre ja està ocupada per un altre, es pot optar per dues alternatives:

- Cercar seqüencialment una nova adreça lliure en el mateix fitxer fins a trobar una **posició lliure** on emmagatzemar els registres. Aquest procés és lent i provoca una mala ocupació de la memòria (degrada el fitxer), ja que queden espais buits.
- Reservar una **zona de desbordament** on emmagatzemar, de manera consecutiva, els sinònims (registres que obtenen el mateix valor de l'adreça, i per tant, poden ocupar una mateixa posició), a mesura que apareixen.

Els fitxers amb aquesta organització tenen com a principals **avantatges** la gran flexibilitat i la rapidesa de consulta.

Entre els **inconvenients** cal destacar el desaprofitament de l'espai (ja que s'ha de reservar tot l'espai necessari, encara que no hi hagi dades) i la necessitat d'utilitzar suports direccionables.

#### **Exemple d'organització directa o aleatòria**

El fitxer amb les dades dels vehicles que tenen contractada una plaça en un aparcament privat amb 2.000 places. Com que no es pot guardar un registre per a cada matrícula

#### **Nota**

- **Col·lisió:** situació que s'origina quan es vol posar un registre en una posició del fitxer que ja està ocupada per un altre registre.
- **Sinònims:** registres que entren en col·lisió perquè, per transformació, obtenen un mateix valor de la posició o adreça física.
- **Zona de desbordament:** zona on s'emmagatzemen els registres sinònims que no es poden emmagatzemar en la posició que els correspon de la zona de registres perquè està ocupada per un altre registre. També s'anomena àrea d'excedents, zona de saturació o *overflow*.

possible, caldrà "generar" la posició del registre en funció, per exemple, dels números de la matrícula. Una possible fórmula (o algorisme de conversió) és agafar les quatre xifres de la matrícula; si la primera xifra està entre 0 i 4 aquesta es canvia per un 0 i si està entre 5 i 9 es canvia per un 1. Aquesta nova combinació de xifres indica la posició en la qual s'emmagatzemarà el registre en el fitxer. A més, es reserva espai per a 200 registres addicionals, per exemple, en una zona de desbordament al final del registre.

En aquest exemple, una possible distribució seria la següent:

Núm. registre	Matrícula	Propietari	Núm. plaça	...
...				
0233	T-3233-BC			
...				
0323	L-3323-CD			
...				
0333	T-4333-BB			
...				
0367	G-3367-AQ			
...				
0421	T-3421-EF			
...				
0454	B-3454-ZY			
...				
0456	B-3456-SZ			
...				
0543	L-4543-AA			
...				
0565	L-4565-AB			
...				
0567	T-4567-AB			
...				
0576	B-4576-VU			
...				
0587	G-4587-AW			
...				
1458	B-5458-TY			
...				

Núm. registre	Matrícula	Propietari	Núm. plaça	....
1675	G-8675-DD			
1676	G-5676-ED			
...				
1786	T-9786-DX			
1787	B-6787-VB			
...				
Zona de cycle de treball				
2000	T-3454-CC			
2001	L-9786-DD			
...				
2199	...			

Els dos registres emmagatzemats a la zona de desbordament no es poden situar a la zona normal de registres atès que la seva posició natural està ocupada. Són sinònims d'altres registres que ja ocupen l'espai que correspondria als sinònims.

### 1.5. Limitacions dels fitxers

Tot i que en el seu moment els fitxers van ser un element important per a l'emmagatzematge de la informació (i, de fet, encara ho són com a suport de moltes aplicacions informàtiques), el seu ús i treball comporta, però, una sèrie d'inconvenients, i els principals són:

- **Inconsistència de la informació.** En tenir informació parcialment o totalment duplicada en diversos fitxers, o en tenir organitzacions diferents, l'actualització de les dades pot ser costosa. Un error en l'actualització d'aquesta informació pot provocar inconsistències en les dades del fitxer.

#### Exemple d'inconsistència de la informació

Per exemple, si volem emmagatzemar la informació dels pacients d'un hospital amb el seu metge associat, i també les dades dels metges, caldrà disposar de dos fitxers. Part de la informació dels metges estarà repetida en els dos fitxers.

- **Redundància.** Aquest problema s'esdevé en tenir dades que no aporten informació i que es poden calcular a partir d'altres dades.

#### Exemple redundància

Moltes vegades, per tal de facilitar la rapidesa de l'accés a la informació, per exemple, es poden guardar dades en el fitxer que es podrien calcular a partir d'altres camps emmagatzemats. Per exemple, l'import amb IVA de les factures del fitxer de factures de proveïdors de l'empresa.

- **Rigidesa de cerca.** Cada fitxer té una organització determinada, segons el tipus d'accés per al qual s'ha definit.



### Exemple de rigidesa de cerca

Si s'ha creat un fitxer de tipus seqüencial, per exemple, sempre s'hi haurà d'accedir d'aquesta manera, malgrat que el seu contingut pugui evolucionar i es pugui aplicar un altre mètode de cerca.

- **Dependència dels programes.** Les relacions entre les dades no s'emmagatzemen al seu costat. El fet de conèixer i mantenir aquestes relacions és responsabilitat del programa que els gestiona. Qualsevol canvi en l'estructura del fitxer representa modificar els programes que l'utilitzen.

### Exemple de dependència dels programes

Els programes informàtics desenvolupats per a gestionar la informació han de conèixer l'estructura i la disposició dels camps dels fitxers que tracten. Afegir, per exemple, un nou camp al fitxer dels pacients de l'hospital, significaria haver de modificar tots els programes que treballen amb aquest fitxer per tal que puguin consultar la informació d'aquest nou camp.

Per aquest motiu, en la majoria d'aplicacions informàtiques l'ús dels fitxers s'ha substituït pel treball amb bases de dades.

## 1.6. Resum

Un fitxer és un conjunt ordenat de dades que mantenen una relació lògica les unes amb les altres i s'emmagatzema en un suport d'informació perquè els puguin utilitzar un programa o una aplicació.

En un fitxer s'emmagatzema informació que fa referència al mateix tema d'una manera estructurada per a poder treballar amb les dades de manera individual.

Els fitxers estan compostos per estructures més simples anomenades **registres**.

Cada registre està format per **camp**s que contenen informació referent a un element o característica particular dins del fitxer.

Les principals operacions que es poden dur a terme sobre un fitxer són la seva creació, la lectura, el manteniment i l'actualització de la informació (registres i camps), l'ordenació dels registres, la cerca o consulta de dades, la fusió de dos o més fitxers en un de sol, la divisió d'un fitxer en un o més, la seva destrucció o eliminació i la generació d'informes o impresos a partir de la informació continguda.

Els arxius s'emmagatzemen en els **dispositius de memòria massiva**, també anomenats de **memòria auxiliar**.

Aquests dispositius poden ser de dos tipus: seqüencial i d'accés directe.

- En els **suports seqüencials**, per exemple una cinta, per a accedir al registre  $n$  s'han de llegir els  $n-1$  registres anteriors.
- En canvi, en els **suports d'accés directe**, per exemple un disc dur o un disquet, es pot accedir directament a un registre físic solament donant l'adreça física.

Els fitxers es poden classificar segons dos criteris principals:

- Segons la **longitud dels registres** que conté. En aquest cas, es distingeix entre els fitxers de longitud fixa, els de longitud variable, els delimitats i els indefinits.
- Segons l'**ús que se'n farà**. Es poden classificar entre fitxers permanents i temporals. Dins del primer tipus distingim entre fitxers mestres o de situació, fitxers constants o fitxers històrics; el segon tipus poden ser intermedis, de maniobres o de resultats.

Els tipus d'organització de fitxers són bàsicament quatre: seqüencial, seqüencial encadenada, seqüencial indexada i directa. La seva organització condicionarà quins tipus d'operacions es poden fer amb el fitxer. En funció de l'aplicació del fitxer serà més aconsellable una organització o una altra.

Finalment, remarcar que ateses les limitacions dels fitxers (inconsistència, redundància, rigidesa a les cerques, dependència dels programes...), s'han substituït, en la majoria de les aplicacions, per bases de dades.

## 2. Bases de dades i sistemes gestors de bases de dades

Les bases de dades van sorgir per a intentar resoldre els problemes que tenien els fitxers.

Una base de dades és un sistema format per un conjunt de dades organitzades de manera que s'eviten les dades redundants, són independents dels programes que les fan servir, emmagatzemen les dades juntament amb les relacions entre elles i s'hi pot accedir de diverses maneres.

Dit d'una altra manera, una **base de dades (BD)** és una col·lecció de dades organitzades de manera que es pugui accedir, administrar i actualitzar els seus continguts amb facilitat.

El **sistema gestor de bases de dades (SGBD)** és el conjunt de programari destinat a la creació, gestió, control i manipulació de la informació emmagatzemada en una base de dades.

### Acrònims més comuns en bases de dades

En aquest mòdul es fan servir els acrònims següents, àmpliament usats en català (i en castellà):

Concepte	Acrònim	Comentari
Base de dades	<b>BD</b>	En anglès, DB per <i>data base</i>
Sistema gestor de bases de dades	<b>SGBD</b>	En anglès, DBMS per <i>data base management system</i>
Llenguatge de definició de dades	<b>DDL</b>	De l'anglès <i>data definition language</i>
Llenguatge de manipulació de dades	<b>DML</b>	De l'anglès <i>data management language</i>
Llenguatge de consulta estructurat	<b>SQL</b>	De l'anglès <i>structured query language</i>

Els SGBD disposen d'un llenguatge de definició de dades (**DDL**) que permet definir l'esquema de la base de dades, un llenguatge de manipulació de dades (**DML**) que facilita la gestió de la informació emmagatzemada i una interfície d'usuari, que permet l'accés i treball d'una manera còmoda al sistema.

Cronològicament, les bases de dades s'han classificat en tres grups: **jeràrquiques, en xarxa i relacionals**.

Cap als anys setanta van aparèixer les **BD** (bases de dades) relacionals per a obtenir més flexibilitat en el tractament de les dades i són, avui dia, les més esteses.

Una **BD relacional** està formada per taules, que són una estructura de files (els registres d'informació) i columnes (els camps d'informació dels registres).

El disseny i creació d'una base de dades relacional consistirà a definir el **model relacional**, és a dir, les taules, els seus camps, i les relacions entre taules, principalment.

Prèviament al disseny d'aquest model relacional, però, s'aconsella fer el **model conceptual**, que permet "plasmear" en un model d'entitat-relació les diferents informacions que es volen arribar a emmagatzemar i gestionar amb la futura base de dades relacional.

Un tipus de bases de dades especial són les documentals (**BDD**), caracteritzades perquè cada registre es correspon amb un document, de qualsevol tipus, publicació, document gràfic o sonor, etc., o a la seva referència.

Les BDD es poden classificar en diverses categories segons la informació que contenen i la referència al document corresponent: bases de dades de text complet, arxius electrònics d'imatge i bases de dades referencials. També es poden classificar per altres tipologies: segons el mode d'accés a la informació, la cobertura documental o segons el model de tractament documental.

En els propers apartats es detallaran cadascun d'aquests aspectes.

## 2.1. Conceptes bàsics

En els subapartats següents s'exposen alguns conceptes bàsics sobre les bases de dades i els fitxers de les bases de dades.

### 2.1.1. Bases de dades

Una **base de dades** és una col·lecció d'informació emmagatzemada de manera organitzada en format electrònic.

Un **sistema gestor de bases de dades** és un programa que permet organitzar-ne l'emmagatzematge i facilitar la recuperació.

Les bases de dades ofereixen diversos **avantatges**:

- Faciliten l'emmagatzematge de quantitats grans d'informació.
- Faciliten la recuperació ràpida i flexible d'informació.
- Faciliten l'organització i reorganització de la informació.

#### Exemple de gestor BD

Exemples de gestors de BD relacionals serien el programa Microsoft Access, el servidor de BD Oracle, Ms SQL Server, MySQL, etc.

- Faciliten la impressió i distribució d'informació de diverses maneres.

### Exemples de bases de dades

Les bases de dades són una col·lecció d'informació de qualsevol tipus com, per exemple, un directori telefònic, un targeter de receptes, un catàleg de fitxes bibliogràfiques, l'inventari de productes i serveis de la companyia, els registres de qualificacions escolars d'un estudiant, la llista de les assignatures d'un curs, la relació dels treballadors de l'organització, la llista dels clients de la companyia, els pacients d'un centre assistencial, la llista de les lectures de la temperatura d'un termòmetre, etc.

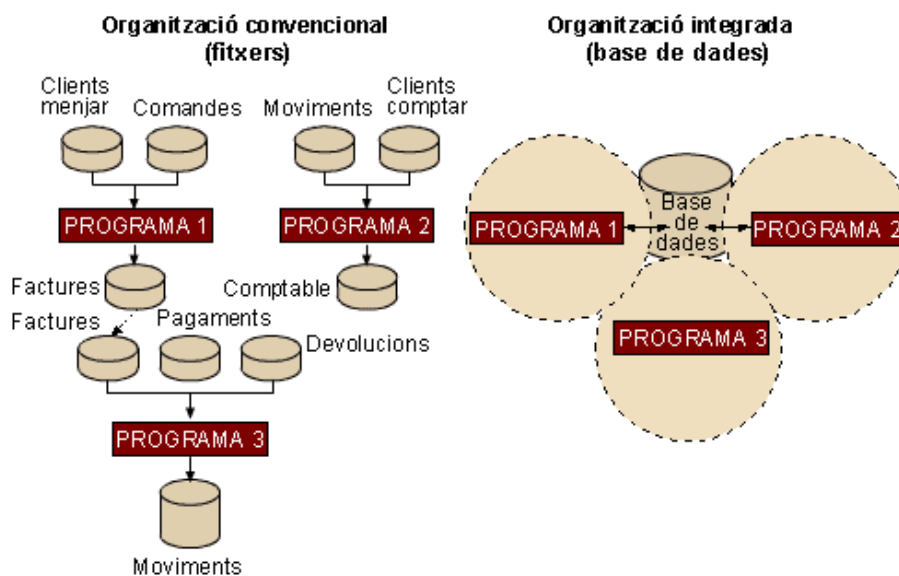
La informació d'alguns d'aquests exemples es pot gestionar en forma de simples fitxers, ja que contenen un únic tipus d'entitat (clients, pacients, temperatures, qualificacions...), mentre que altres integren informació de diversos tipus d'entitat que només pot ser tractada com a bases de dades.

Gairebé qualsevol col·lecció d'informació es pot convertir en una base de dades.

### 2.1.2. Fitxers i bases de dades

Les bases de dades van aparèixer com una alternativa als sistemes de fitxers per a solucionar els inconvenients i desavantatges que presentaven.

La figura següent mostra la diferent manera d'organitzar les dades en un sistema de fitxers i en una base de dades. El primer necessita múltiples fitxers per a organitzar la mateixa informació que la segona integra en un únic arxiu.



#### Per a saber-ne més

Per a tenir més informació sobre els inconvenients i desavantatges que presentaven els fitxers, podeu consultar l'apartat "Limitacions dels fitxers" en aquest mateix mòdul.

Actualment, l'ús de fitxers es limita, bàsicament en aplicacions en què no cal un accés directe a la informació o hi ha pocs tipus de dades relacionades entre diferents fitxers, i com a suport per a emmagatzemar la informació de programes (processadors de textos, etc.) o del sistema operatiu.

Tot i així, físicament, a l'ordinador, les bases de dades s'emmagatzemen i gestionen com a fitxers, per mitjà dels sistemes gestors de bases de dades (SGBD).

Encara que, de manera molt simplificada, es poden identificar les diferències següents entre els fitxers tradicionals i les bases de dades:

- **Entitats tipus.** Els fitxers tenen registres d'una sola entitat tipus (per exemple, pacients), mentre que les BD tenen dades de diverses entitats tipus (pacients, tractaments, metges, etc.).

### El concepte d'entitat

Tant els fitxers com les bases de dades emmagatzemen informació d'*objectes* del món real, anomenats *entitats*, que tenen un paper important en les organitzacions. Es pot distingir dos nivells d'abstracció d'una entitat:

- **Entitat instància:** és un objecte real o abstracte amb existència independent i característiques que el diferencien d'altres objectes, encara que siguin del mateix tipus. Per exemple, un pacient concret, un metge concret, un alumne concret, una malaltia concreta o una assignatura concreta. En aquest mòdul també s'usa el terme *ocurrència* per a referir-se a aquest concepte.
- **Entitat tipus:** és un conjunt d'entitats instància que comparteixen les mateixes característiques. Per exemple, l'abstracció *pacient* formada per tots els pacients d'un hospital o l'abstracció *alumne* formada per tots els alumnes d'una escola. En aquest mòdul també s'usa el terme *entitat* (a seques) per a referir-se a aquest concepte.
- **Interrelacions.** En el cas dels fitxers, el sistema no interrelaciona fitxers, mentre que en el cas de les BD, el sistema té eines previstes que interrelacionen entitats.
- **Redundància.** Mentre es creen fitxers a la mida de cada aplicació, amb totes les dades necessàries, encara que algunes siguin redundants respecte a d'altres fitxers, en les BD totes les aplicacions treballen amb la mateixa BD i la integració de les dades és bàsica, de manera que s'evita la redundància.
- **Usuaris.** Els fitxers serveixen per a un sol usuari o una sola aplicació, mentre que les BD són compartides per molts usuaris de tipus diversos.

## 2.2. Aplicacions pràctiques de BD

Avui dia, la majoria dels sistemes d'informació d'una organització estan basats en bases de dades.

### Exemples d'aplicacions pràctiques de BD

- Per exemple, tota la informació gestionada pel departament comercial (clients, productes, vendes, etc.) estarà en bases de dades.
- També ho estarà tota la informació tractada pel departament de màrqueting i comunicació (campanyes, actuacions, etc.).
- Tota la informació relativa als empleats i gestionat pel departament de recursos humans també estarà en una o més bases de dades, i també tota la informació utilitzada pels departaments de finances i compres.
- Tota les dades d'un sistema de gestió d'alumnes d'un centre de formació, per exemple, també estaran en una base de dades.
- Altres exemples serien, per exemple, tota la informació sobre les pel·lícules d'un videoclub, els seus clients i les seves actuacions (lloguers, retorns, etc.).
- Totes les biblioteques basen la seva gestió sobre bases de dades.

I així trobaríem infinitat d'exemples.

#### 2.2.1. Característiques i objectius de les BD

Tal com hem comentat en l'apartat anterior, les bases de dades sorgeixen per a resoldre els problemes que presenten els fitxers: inconsistència de la informació, redundància, rigidesa de recerca i dependència dels programes.

Una base de dades és un sistema format per un conjunt de dades organitzades de manera que es controla l'emmagatzematge de les que són redundants; les dades resulten independents dels programes que les fan servir; s'emmagatzemen de manera conjunta les dades i les relacions entre si, i es pot accedir a la informació de diverses maneres.

Els requeriments principals que ha de complir un bon sistema de bases de dades (BD) són els següents:

- Diversos usuaris poden accedir simultàniament a la base de dades, cadascun a una informació determinada.
- Es controla l'accés dels usuaris a les dades i es garanteix la confidencialitat i seguretat.
- Les dades s'emmagatzemen sense redundància (excepte en alguns casos en què interessa millorar els accessos i el rendiment).
- Permeten utilitzar diferents mètodes d'accés i assegurar flexibilitat a les recerques.
- Disposen de mecanismes de recuperació d'informació.
- Es pot canviar el suport físic de la base de dades sense que se'n ressentin la base de dades ni els programes que la utilitzen.

- La modificació del contingut i les relacions entre les dades no afecta els programes que les utilitzen.
- Disposen d'una interfície d'usuari que permet utilitzar la base de dades de manera còmoda i flexible.
- Tota base de dades es gestiona per mitjà d'un sistema de gestió de bases de dades (SGBD).

### 2.3. Sistemes gestors de BD

S'anomena *sistema gestor de base de dades* (DBMS o SGBD) el conjunt de programari destinat a la creació, gestió, control i manipulació de la informació sobre una base de dades. Els SGBD tenen com a finalitat registrar i mantenir informació.

Un SGBD ha de permetre dur a terme les operacions següents:

a) **Definició de l'esquema de la base de dades.** Una vegada dissenyat l'esquema de la base de dades, l'hem de descriure mitjançant un conjunt d'instruccions. Això es fa mitjançant un llenguatge específic, anomenat *llenguatge de descripció de dades* (DDL), i que, com qualsevol llenguatge d'alt nivell, necessitarà un traductor per a generar el codi objecte a partir del codi font.

b) **Accés a la informació des d'un llenguatge d'alt nivell.** Això es fa mitjançant un llenguatge específic, anomenat *llenguatge de manipulació de dades* (DML). EL DML es pot utilitzar de dues maneres diferents:

- Com a llenguatge hoste, incloent sentències DML dins d'un programa amfitrió escrit en un llenguatge d'alt nivell (com, per exemple, Cobol, Pascal, Basic, C, etc.).
- De manera interactiva, per mitjà de programes que continguin exclusivament sentències pròpies del DML.

c) **Accés a la informació en mode conversacional.** L'SGBD incorpora una interfície d'usuari per mitjà de la qual, l'usuari pot introduir sentències DDL o DML directament des d'un terminal i obtenir informació interactiva.

d) **Gestió d'arxius.** Funció realitzada per un mòdul anomenat *gestor d'arxius* que s'encarrega de la comunicació amb el sistema operatiu.

#### Per a saber-ne més

Per a tenir més informació sobre l'esquema de la base de dades, podeu consultar l'apartat "Definició conceptual de BD" en aquest mateix mòdul.

#### Per a saber-ne més

Per a saber més sobre els llenguatges de definició (DDL) i de manipulació (DML) de dades podeu consultar l'apartat "Llenguatges de bases de dades relacionals" en aquest mateix mòdul.



e) **Altres funcions:** com controls d'usuaris, recuperació de la informació després d'errades del sistema, organització física de la base de dades, control de seguretat i privadesa d'informació o la gestió d'accessos concurrents.

En un sistema tradicional de fitxers, cadascuna de les aplicacions haurà de realitzar la descripció dels registres i també determinar l'organització de fitxers, els tipus d'accés, etc.

En un entorn de base de dades, aquestes especificacions les fan els SGBD, i és l'administrador de la base de dades l'encarregat de garantir que s'efectuïn aquestes i altres funcions sobre el sistema.

#### Per a saber-ne més

Per a tenir més informació sobre sistemes operatius, podeu consultar la unitat "Sistemes operatius" del mòdul "Ordinadors i sistemes operatius".

#### Per a saber-ne més

Per a tenir més informació sobre fitxers, podeu consultar la unitat "Fitxers" d'aquest mateix mòdul.

### 2.3.1. Característiques i objectius dels SGBD

A continuació es detallen algunes d'aquestes característiques o requisits dels sistemes gestors de bases de dades (SGBD):

- **Consultes no predefinides i complexes.** L'objectiu fonamental dels SGBD és permetre que es facin consultes no predefinides (*ad hoc*) i complexes. L'usuari ha de poder formular la consulta amb un llenguatge senzill, que es quedi en el nivell lògic, i el sistema l'ha d'interpretar directament. Els usuaris podran fer consultes de qualsevol tipus i complexitat directament a l'SGBD, que haurà de respondre immediatament sense que estiguin preestablertes, és a dir, sense que s'hagi d'escriure, compilar i executar un programa específic per a cada consulta. En canvi, en els fitxers tradicionals, cada vegada que es volia fer una consulta s'havia d'escriure un programa a mida.
- **Flexibilitat i independència.** La complexitat de les BD i la necessitat d'anar-les adaptant a l'evolució del sistema d'informació fan que un objectiu bàsic dels SGBD sigui donar flexibilitat als canvis. Interessa obtenir la màxima independència possible entre les dades i els processos i programes perquè es puguin fer tot tipus de canvis tecnològics i variacions en la descripció de la BD, sense que s'hagin de modificar els programes d'aplicació ja escrits ni canviar la manera d'escriure les consultes directes. En el món dels fitxers ja hi havia independència física en un grau determinat, però en el món de les BD acostuma a ser molt més gran.
- **Mínima redundància.** Un dels objectius dels SGBD és facilitar l'eliminació de la redundància, ja que comporta un risc d'inconsistència o incoherència de les dades. Convé fer que una dada només figuri una sola vegada a la BD. En els fitxers tradicionals, cada aplicació utilitzava el seu fitxer. Però com que hi havia molta coincidència de dades entre aplicacions, hi havia molta redundància entre els fitxers.
- **Concurrència d'usuaris.** Un objectiu fonamental dels SGBD és permetre que diversos usuaris puguin accedir concurrentment a la mateixa BD. Per

tal de tractar els accessos concurrents, els SGBD fan servir el concepte de transacció de BD, concepte d'especial utilitat per a tot allò que fa referència a la integritat de les dades. S'anomena *transacció de BD* o simplement *transacció* un conjunt d'operacions simples que s'executen com una unitat. Els SGBD han d'aconseguir que el conjunt d'operacions d'una transacció mai no s'executi parcialment. O s'executen totes o no se n'executa cap.

- **Mecanismes de seguretat.** En el camp dels SGBD, el terme *seguretat* se sol utilitzar per a fer referència als temes relatius a la confidencialitat, les identificacions, les autoritzacions, els drets d'accés, etc. Els SGBD permeten definir autoritzacions o drets d'accés a diferents nivells (globalment a tota la BD, a l'entitat i a l'atribut). Aquests mecanismes de seguretat requereixen que l'usuari es pugui identificar. S'acostumen a utilitzar codis d'usuari (i grups d'usuaris) acompanyats de contrasenyes (*passwords*), però també s'utilitzen targetes magnètiques, identificació per reconeixement de la veu, etc.

### 2.3.2. SGBD del mercat

Avui dia, hi ha molts paquets en el mercat que redueixen costos i esforços i fan de les bases de dades una aplicació pràctica per a la majoria dels usuaris d'ordinador.

Aquests paquets són gestors de bases de dades sofisticats i poderosos que tracten de satisfer les necessitats de dos grups d'usuaris:

- **Usuaris finals**, que són els que volen utilitzar la base de dades per a emmagatzemar, organitzar i recuperar les dades.

#### **Exemple d'SGBD per a usuaris finals**

Entre els SGBD per a l'entorn Windows a nivell d'usuari final destaca Microsoft Access.

- **Programadors**, que la volen utilitzar per a desenvolupar aplicacions específiques per als negocis o organitzacions.

#### **Exemple d'SGBD per a programadors**

Entre els SGBD servidors, pensats per a departaments d'organitzacions i empreses, destaquen els productes comercials Informix, Microsoft SQL Server i el servidor Oracle. Els principals SGBD de programari lliure són MySQL i PostgreSQL.

Entre els paquets líders per a Windows, a nivell d'usuari destaca el Microsoft Access.

Entre els sistemes gestors de bases de dades servidors, és a dir, els pensats per a departaments d'organitzacions i empreses, destaquen els servidors Oracle, Microsoft SQL Server, Informix, MySQL, etc.

## 2.4. Tipus de bases de dades

Atenent al model lògic de dades, hi ha diversos tipus de BD: **jeràrquiques, en xarxa, relacionals i orientades a objectes** o combinacions entre diferents tipus. Per exemple, avui dia, Oracle és un SGBD per a bases de dades relacionals i orientades a objectes.

### Model de dades

Un **model de dades** és, d'una banda, la descripció del contenidor de dades (on es guardarà la informació), i de l'altra, el conjunt de mètodes per a emmagatzemar i recuperar informació dels contenidors.

Tot model de dades proporciona una sèrie d'elements (objectes, associacions, propietats, operacions, restriccions...) que permeten, mitjançant l'abstracció d'una part del món real, obtenir un conjunt estructurat de dades i un conjunt d'operacions definides sobre aquestes dades.

A continuació, es detallen els tipus de BD atenent al seu **model de dades**:

### 1) BD jeràrquica

El primer model de BD que va aparèixer va ser el **model jeràrquic**, a principi dels anys seixanta. Tal com indica el seu nom, emmagatzema la informació en una estructura jeràrquica.

Les dades s'organitzen en registres interrelacionats de manera similar a un arbre invertit. Un arbre es compon d'un conjunt de registres (*nodes*) units per mitjà d'associacions (*arcs*) pare-fill que poden ser d'un a un o d'un a molts. És a dir, un node "pare" pot tenir un o més nodes "fills". El node que no té pares s'anomena *arrel* i els nodes que no tenen fills s'anomenen *fulles*. Aquestes associacions entre nodes s'implementen físicament utilitzant *punters*.

El terme *node*, quan parlem de bases de dades, té les accepcions següents:

- En una BD jeràrquica o en xarxa, **cadascun dels objectes** (entitats o registres) que conformen l'estructura de dades (en arbre o en xarxa, respectivament).
- En una BD distribuïda, **cadascun dels ordinadors** o llocs interconnectats on s'emplacen les dades.

### Exemples d'SGBD jeràrquics

L'IMS (*information management system*) d'IBM dissenyat l'any 1966 i que va sortir l'any 1968 va ser líder dels SGBD jeràrquics durant els anys setanta. Malgrat que va quedar superat per altres models de dades, és l'únic que ha sobreviscut, ja que desenvolupaments posteriors han permès que suporti aplicacions en Java, JDBC, XML i serveis web.

Altres SGBD de l'època, avui dia en desús, són System-2000 (d'MRI, després comercialitzat per SAS Institute), Mark IV (de Control Data Corporation) i TDMS (de System Development Corporation).

#### Per a saber-ne més

Per a saber més sobre models de dades i els seus elements constitutius podeu consultar el subapartat "Models de dades: conceptes bàsics".

#### Per a saber-ne més

Podeu trobar una petita descripció de com funciona el punter d'un registre en el subapartat "Organització seqüencial encadenada".

El model jeràrquic i les estructures en arbre s'utilitzen avui dia en altres sistemes d'emmagatzematge com el registre de Windows, els documents XML o els repositoris LDAP (basats en el protocol d'accés a serveis de directori).

El seu **inconvenient** principal és la incapacitat de representar la redundància de dades (duplicat de registres) i la poca flexibilitat.

## 2) BD en xarxa

A final dels anys seixanta van sortir SGBD basats en un **model en xarxa**. Com en el model jeràrquic, hi ha registres i interrelacions, però un registre (*node*) ja no està limitat a ser "fill" d'un sol "pare". Un node pot tenir diferents pares, la qual cosa permet representar interrelacions més complexes que amb el model jeràrquic.

Té l'avantatge que permet representar qualsevol sistema i disposar de dades redundants, però té una administració molt complexa, per la qual cosa el seu ús s'ha limitat sempre als programadors i no als usuaris finals.

El comitè CODASYL-DBTG va proposar, l'any 1971, un estàndard basat en aquest model que va ser adoptat per molts productors d'SGBD, però va trobar l'oposició d'IBM, l'empresa dominant aleshores.

### Exemples d'SGBD basats en el model en xarxa

El primer que va aparèixer va ser l'IDS (*integrated data store*), desenvolupat per Charles W. Bachman a mitjan anys seixanta per a la companyia General Electric. Versions posteriors, com l'IDS/2, van ser comercialitzades per Honeywell-Bull.

L'SGBD comercial més destacat va ser l>IDMS (*integrated database management system*) de Cullinet (actualment de Computer Associates), malgrat que no compleix plenament l'estàndard CODASYL.

Altres sistemes coneguts inclouen VAX-DBMS (de Digital), IMAGE (de Hewlett-Packard) i DMS\_1100 (d'Univac).

Tant els sistemes del model en xarxa com del jeràrquic (que es pot considerar un cas particular del primer) presentaven llenguatges procedimentals que obligaven el programador a navegar registre a registre per la BD i no disposaven de prou independència físicollògica, cosa que comportava una flexibilitat escassa.

## 3) BD relacional

La introducció de la teoria matemàtica de l'*àlgebra relacional* en el camp de les BD per part d'Edgar F. Codd (d'IBM) l'any 1970 el va dur a proposar el **model relacional**.

Es basa en el concepte matemàtic de *relació* (que té una aparença similar a una taula de valors) i en la *teoria de conjunts*. Una **taula** és una estructura bidimensional on s'emmagatzemen les dades com un conjunt de registres del mateix

### Per a saber-ne més

Per tenir més detalls sobre el model relacional podeu consultar el subapartat següent ("Bases de dades relacionals").

tipus, que es divideixen horitzontalment en files i verticalment en columnes. Una **fila** representa un registre o grup de valors de camps, i una **columna** conté informació referent a un únic camp o atribut de diferents registres.

Aquest model va significar un pas important per al desenvolupament dels SGBD. Durant els anys vuitanta es va estendre l'ús dels sistemes relacionals, la majoria dels quals utilitzen com a llenguatge natiu per a la manipulació de les dades l'SQL (estandarditzat l'any 1986).

#### Per a saber-ne més

Per a saber més sobre l'SQL podeu consultar el subapartat "Llenguatges de bases de dades relacionals".

### Exemples d'SGBDR

El prototip System R (d'IBM) va ser l'origen dels primers SGBD relacionals com Ingres (creat el 1974 per la Universitat de Berkeley, i comercialitzat el 1980 per Ingres Inc. i després per Computer Associates).

Avui dia, hi ha centenars d'SGBD relacionals, tant per a PC (monousuari) com per a sistemes *mainframe* (multiusuari), malgrat que molts no són completament fidels al model relacional.

Els primers SGBD comercials multiusuari van ser Oracle (1979) i DB2 d'IBM (1982). Altres sistemes multiusuari són Informix (després Informix Dynamic Server, d'IBM), Sybase (SQL Server, després Adaptive Server Enterprise), Microsoft SQL Server, Allbase (de Hewlett-Packard), Interbase (de Borland) i els SGBD de codi obert MySQL, Firebird (basat en Interbase) i OpenIngres (de Computer Associates).

Els SGBD relacionals per a microordinadors, inicialment monousuari, van oferir després arquitectura client-servidor i s'han anat adaptant a l'estàndard ODBC (*Open Database Connectivity*) de Microsoft, que permet la utilització d'eines de tipus frontal (*front end*). En són exemples Paradox i dBase (de Borland), FoxPro i R:base (de Microrim) i Access de Microsoft.

La majoria de sistemes informàtics actualment en funcionament utilitzen un SGBD relacional (o extensions d'aquest), tot i que en algunes organitzacions encara s'usen els jeràrquics.

## 4) BD d'objectes o orientada a objectes

Estructura la informació en classes i subclasses d'objectes complets (estat i comportament) i incorpora els conceptes importants de la *programació orientada a objectes* (encapsulació, herència, polimorfisme...). Tots els *objectes* pertanyen a una *classe* d'objectes generalitzats que comparteixen les mateixes funcions. Per mitjà de l'*herència*, els objectes d'una classe poden adquirir les funcions de la classe.

Els conceptes següents de l'enfocament orientat a objectes són importants per als sistemes de BD:

- **Objecte:** entitat discreta que té bàsicament dos components:
  - *Estat* (valor) o estructura amb els seus atributs.
  - *Comportament*, que queda determinat per les **operacions** (procediments, mètodes i funcions) que s'hi poden aplicar externament.

- **Classe:** categoria generalitzada que descriu les característiques i el comportament d'un grup d'objectes que hi pot haver dins seu.
- **Herència:** transferència de les propietats i el comportament d'una classe a una nova subclasse que se'n deriva.
- **Mètode:** implementació d'una operació que s'aplica externament als objectes.
- **Missatge:** codi de programa que s'envia a l'objecte per a cridar o invocar un mètode.
- **Encapsulació:** acció d'ocultar l'estructura i el comportament interns d'un objecte detallant amb rigor el seu comportament extern, cosa que permet impedir conflictes i accessos incorrectes.
- **Polimorfisme:** capacitat segons la qual diferents classes d'objectes tenen un comportament diferent davant d'una mateixa operació.

Correspondència entre els termes d'OO i els de programació tradicional

	OO	Programació
<b>Termes</b>	Objecte	Variable
	Classe	Tipus
	Mètode	Procediment, rutina (seqüència d'instruccions que executen una única funció)
	Missatge	Crida a rutina o procediment

#### Exemples d'SGBDO

Destaquen GemStore (de ServiLogic), ObjectStore (d'Object Design), Ardent (abans O2), Objectivity, Versant, Ontos i Poet.

El model orientat a l'objecte va sorgir a final dels anys vuitanta per la manca de capacitat semàntica del model relacional a l'hora d'atendre determinants tipus d'aplicacions que requereixen modelar objectes i interrelacions complexos, emmagatzemar informació no estructurada, etc.

#### Gestors d'objectes

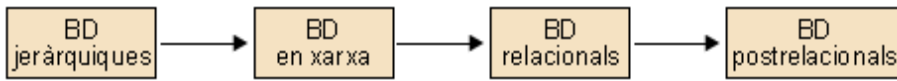
Sistemes amb característiques similars als SGBDO, però amb un model de dades limitat, que són extensions de sistemes de fitxers o de gestió de memòria virtual. En són exemples Mneme (gestor de memòries de traducció) i LOOM (sistema gestor de bases del coneixement).

Això ha fet que els conceptes d'orientació a objectes (OO) s'hagin incorporat als sistemes relacionals.

Tradicionalment, les BD s'han classificat cronològicament en diferents generacions:

- 1a. generació: **BD prerelacionals:** jeràrquiques i en xarxa
- 2a. generació: **BD relacionals**

- 3a. generació: **BD postrelacionals**: orientades a l'objecte, relacionals esteses, distribuïdes, documentals...



Evolució cronològica de les bases de dades

Les **BD postrelacionals** inclouen moltes tipologies i combinacions de BD, algunes de les quals no responen a un model de dades definit. A continuació s'exposen les més destacades:

### 1) BD relacionals esteses

Sistemes apareguts a partir dels anys vuitanta, fruit de l'evolució del model relacional per a superar les seves limitacions i obtenir més capacitat expressiva que representi més fidelment el món real, incorporant funcionalitats d'altres models de dades com orientació a objectes, deductives, amb mecanismes d'activitat, etc.

S'hi poden distingir diversos enfocaments:

a) **BD relacional amb frontal OO**: afegeix un nivell o capa d'OO superposada a un SGBD relacional preexistent.

b) **BD objecte-relacional** (SGBDOR): segueix l'enfocament integrador dels dos models, incorporant noves capacitats:

- Nous tipus de dades (definides per l'usuari, multimèdia, objectes grans BLOB) que permeten gestionar aplicacions més complexes amb una gran riquesa de dominis.
- Operacions que permeten gestionar el comportament dels tipus de dades.
- Més capacitat expressiva per a la semàntica de les dades (disparadors, procediments emmagatzemats i funcions definides per l'usuari).
- Reutilització de llibreries de classes ja existents.
- Més capacitat consultiva (consultes imbricades, recursives, emmagatzemades, prefabricades...).

En realitat, els SGBD objecte-relacionals actuals solen incorporar capacitats deductives, mecanismes d'activitat, integració de documents XML...

c) **BD activa**: permet definir *regles actives* que s'activen automàticament quan es donen determinats *esdeveniments*, que inicien l'execució d'una *acció* (disparador o *trigger*) si es donen unes *condicions* específiques.

Permet modelar millor les regles de funcionament intern d'una organització: mantenir automàticament dades derivades, notificar determinades situacions, garantir l'acompliment de restriccions d'integritat (com les regles de negoci en aplicacions organitzatives complexes).

**d) BD deductiva:** inclou mecanismes d'inferència (basats en *regles deductives* de la lògica matemàtica) que permeten generar informació addicional a partir dels fets emmagatzemats en la BD. Es relaciona amb la intel·ligència artificial i les bases de coneixement.

Permet la comprovació d'hipòtesis, el descobriment de coneixement deductiu (noves relacions entre les dades), la gestió de processos governats per regles, el modelatge de l'estructura i els processos de l'empresa, l'establiment de perfils de clients en comerç electrònic.

Els intents de proporcionar un model de dades que representi més fidelment el món real han donat lloc als **models de dades semàntics**.

### **Extensió del model relacional de Codd**

Per intentar esmenar algunes de les deficiències del seu model relacional, Codd va presentar una versió estesa anomenada RM/T (1979) i, més recentment, RM/V2 (1990).

L'RM/T tracta alguns aspectes similars al model entitat-relació (de Chen, 1976), però de manera més acurada. Per exemple, classifica les entitats en tres categories (nuclis, característiques i associacions), els aspectes estructurals i d'integritat són més amplis i estan definits amb més precisió, inclou operadors especials addicionals, incorpora suport per a la dimensió temps i per a diverses classes d'agregació de dades...

### **Exemples d'SGBDR estesos**

- **BD relacionals amb frontal OO:** evolucions d'SGBDR preexistents són Oracle 8, Informix 9, DB2/UBD (Universal Database, d'IBM) i Sybase. A la dècada del 2000 apareixen sistemes de codi obert com OpenODB (de Hewlett-Packard, basat en l'SGBDR Allbase/SQL) i PostgreSQL (basat en Ingres).
- **BD objecte-relacionals (SGBDOR):** l'any 1990 apareix UniSQL/X, el primer **SGBDO amb SQL** que parteix de zero. A mitjan anys noranta apareixen *Illustra* (després adquirit per Informix i presentat com Informix Universal Server) i *Omniscience*.
- **BD actives:** molts SGBD relacionals actuals (Oracle, DB2, Sybase) disposen d'aquesta funcionalitat que proporcionen en forma de disparadors. *Starburst* és un prototip experimental.
- **BD deductives:** a mitjan anys vuitanta apareix *LDL (Logic Data Language)* i els sistemes experimentals *Coral* i *Nail*!
- Malgrat que no són extensions relacionals, cal esmentar les BDOO amb extensió deductiva (anomenades *BDDOO*) creades a final dels anys vuitanta com *Validity* (de Bull) i *Coral++*.

## **2) BD temporal**

Suporta la variable temps i altres conceptes temporals. Permet emmagatzemar un historial de canvis i consultar l'estat (actual o passat) de la BD. En alguns casos, fins i tot s'emmagatzema informació futura prevista. La informació temporal es pot referir a fets puntuals (que s'esdevenen en un punt del temps) o de durada (en la qual es considera que són vertaders).

### **Aplicacions de les BD temporals**

- **Assistència mèdica,** on cal guardar els historials mèdics dels pacients.



- **Assegurances**, on es necessiten els historials de reclamacions i notificacions d'accidents, i també informació sobre els períodes de vigència de les pòlisses.
- **Reserves** (companyies aèries, serveis de transport, lloguer de cotxes, hostaleria, espectacles, etc.), on cal informació sobre dates i períodes de validesa de les reserves.
- **Recursos humans**, on cal mantenir els historials laborals dels treballadors d'una empresa (referents als salaris, llocs de treball i projectes en què han treballat).
- **Investigació científica**, on cal guardar els valors i períodes de temps en què es mesuren les dades recollides en els experiments.
- **Gestió acadèmica**, on cal incloure el semestre i/o l'any de les qualificacions de les assignatures en expedients acadèmics de l'estudiant i en informació relativa a beques.

### 3) BD espacial

Proporciona conceptes que permeten seguir la pista d'objectes que es troben en un espai multidimensional. Permet interpretar i especificar les característiques espacials dels objectes mitjançant conceptes geomètrics bidimensionals i tridimensionals, i gestionar les relacions espacials entre els objectes.

Els diferents tipus de bases de dades espacials són els següents:

- **BD cartogràfiques**, que emmagatzemen dades de mapes. Inclouen *conceptes geomètrics* bidimensionals (punts, línies, arcs, polígons...) i tenen en compte *operacions espacials* (càlcul de distàncies...) i *operacions booleanes* (verificació de superposicions). S'usen en gestió mediambiental, d'emergències, de conflictes bèl·lics...
- **BD meteorològiques**, que emmagatzemen dades de temperatures i altra informació del temps atmosfèric relacionada amb punts espacials tridimensionals.

### 4) Sistema d'informació geogràfica (SIG)

Permet emmagatzemar, analitzar i representar gràficament informació que descriu diferents propietats geogràfiques i és gestionada per una BD que conté dos tipus de dades:

- Dades espacials: de tipus físic (orografia, topografia i meteorologia), de fronteres polítiques i administratives, de xarxes de transport i comunicacions...
- Dades no espacials: demogràfiques, econòmiques, comercials...

Els sistemes d'informació geogràfica cobreixen disciplines tan diverses com cartografia animada, mapatge interactiu, eines d'ajuda a la decisió espacial, estàndards d'intercanvi de dades geogràfiques, control de qualitat i reenginyeria, BD espaciotemporals, serveis de localització... Es poden agrupar en tres categories:

- **Aplicacions cartogràfiques:** estudis del terreny i el paisatge, anàlisis estructurals de trànsit i xarxes de transport, prospecció marina i petrolera, control i optimització de plantacions agrícoles. Representen les dades mitjançant *atributs espacials* (com la densitat de cultius) i comporten funcions com la *superposició* de diverses capes de mapes per a combinar atributs que permetin la mesura de distàncies en un espai tridimensional.
- **Aplicacions per al modelatge del terreny:** anàlisi del sòl, gestió de recursos de l'aigua, control d'inundacions, estudis de contaminació i impacte ambiental. Representen les dades mitjançant *atributs espacials* (com les característiques del sòl o la qualitat de l'aire) i requereixen la representació digital d'elevacions del terreny en punts de mostreig que s'interconnecten per *interpolació* i donen lloc a un model de la superfície en 3D.
- **Aplicacions d'objectes geogràfics:** anàlisi geogràfica de mercats, anàlisi de pautes de vot, distribució i consum de serveis públics, sistemes de navegació de vehicles, gestió del cadastre i localització de mobiliari urbà. Representen els *objectes* d'interès (com districtes electorals, edificis, semàfors, fanals, centrals elèctriques) i necessiten *funcions espacials* addicionals per a manejar les dades referents a un domini físic concret com xarxes viàries, de telecomunicacions, de subministrament i de recursos hídrics; és a dir, carreteres, cables de fibra òptica i d'alta tensió, conductes de gas i d'aigua, rius...

#### Exemples de SIG

Solen ser sistemes que operen amb SGBD relacionals o d'objectes. ARC/Info, aparegut l'any 1981, integra la funcionalitat d'un SGBD relacional. ARC/Storm (*Arc Store Manager*) maneja BD distribuïdes i s'integra amb SGBD relacionals com Oracle, Informix i Sybase.

## 5) BD multidimensional

Organitza les dades en estructures matricials de diferents dimensions i disposa de llenguatges especials per a fer consultes complexes mitjançant l'accés multidimensional a les dades d'una manera més adequada i eficient que una BD estructurada convencional com la relacional (orientada a transaccions i de naturalesa volàtil).

Aquesta disposició proporciona un tipus més natural d'anàlisi d'informació, ja que facilita l'accés i la visualització de dades en qualsevol combinació de dimensions (les dimensions poden ser, per exemple, els productes, les zones comercials o els períodes impositius d'una empresa) mitjançant diferents criteris i jerarquies (nivells d'agrupament a dins de les dimensions, subtotals prèviament calculats...). Representacions jeràrquiques com l'exploració ascendent i l'exploració descendent permeten agrupar (o resumir) les dades en categories més generals (per exemple, vendes trimestrals per famílies de producte) o disgregar-les en categories més concretes.

S'implementa sobre un motor relacional amb taules on s'emmagatzema un volum de dades molt elevat a les quals s'apliquen operacions de càlcul intensiu (agregació, suma, mitjana...) i tècniques d'indexació especials que permeten consultes ràpides predefinides.

### Processament de dades en línia: tecnologies OLAP i OLTP

De vegades, les BD **multidimensionals** s'anomenen BD de **processament analític en línia** (OLAP, *on line analytical processing*), ja que utilitzen aquesta tecnologia d'anàlisi de dades que considera diferents dimensions i variables al mateix temps.

Els anomenats **magatzems de dades** (*data warehouse*) utilitzen tota la potència de processament analític en línia de les BD multidimensionals.

Les BD convencionals (com les **relacionals**) suporten el **processament de transaccions en línia** (OLTP, *on line transaction processing*), ja que estan optimitzades per a processar un gran volum de consultes i transaccions concurrents (insercions, actualitzacions i supresions d'alguns registres), que poden abraçar una petita part de la BD, amb una freqüència elevada i una alta velocitat de resposta.

Empreses del sector dels serveis (hotels, bancs, línies aèries, companyies d'assegurances, empreses de servei públic i de comunicacions) l'utilitzen com a sistema operacional de consultes i/o reserves durant 24 hores al dia, 7 dies a la setmana.

La informació es pot organitzar en diverses dimensions:

**a) Taula relacional** (una dimensió). Estructura les dades en registres (files) formats per camps (columnes) d'informació relacionada d'una única entitat o objecte del món real; és a dir, d'una sola *dimensió* (i una *variable*). Sovint, però, interessa gestionar més d'una dimensió en una mateixa taula.

#### Exemple de taula relacional

La taula següent mostra informació d'una única dimensió, la corresponent a les zones comercials d'una empresa (zona) amb la variable *vendes*.

ZONA	Distribuidor	Adreça	Telèfon	Vendes
Zona1	Nord	Pl. Catalunya	93 345 89 78	103.600
Zona2	Sud	C/ Marina	93 123 98 78	250.204
Zona3	Centre	Av. Riera	93 333 88 77	303.667
Zona4	Oest	C/ Major	93 123 55 65	238.423
...				

En aquest cas, pot interessar que hi hagi una altra dimensió (per exemple, producte) per a emmagatzemar les vendes de cadascun dels productes de l'empresa, com mostra l'exemple següent de matriu bidimensional.

**b) Matriu de dades bidimensional** (dues dimensions). Permet emmagatzemar conjuntament informació de dues dimensions; és a dir, integra dades de dues entitats.

### Exemple de matriu de dades bidimensional

Un full de càlcul amb les vendes efectuades en cada zona per a cada producte durant un període de temps determinat exemplifica una matriu de dades bidimensional. Les zones es poden mostrar en files i els productes, en columnes; les dades de les *vendes* de cada producte en cada zona estan contingudes a les cel·les. En aquest cas, hi ha dues dimensions (zona i producte) i una variable (*vendes*). Aquesta és la manera multidimensional d'emmagatzemar les dades.

Vendes	Producte				
Zona		Producte1	Producte2	Producte3	...
Zona1		120.000	188.050	123.000	
Zona2		111.000	229.005	128.976	
Zona3		230.400	234.000	90.000	
Zona4		162.050	222.000	89.640	
...					

A continuació, es mostra un exemple de matriu bidimensional amb **dades totalitzades**.

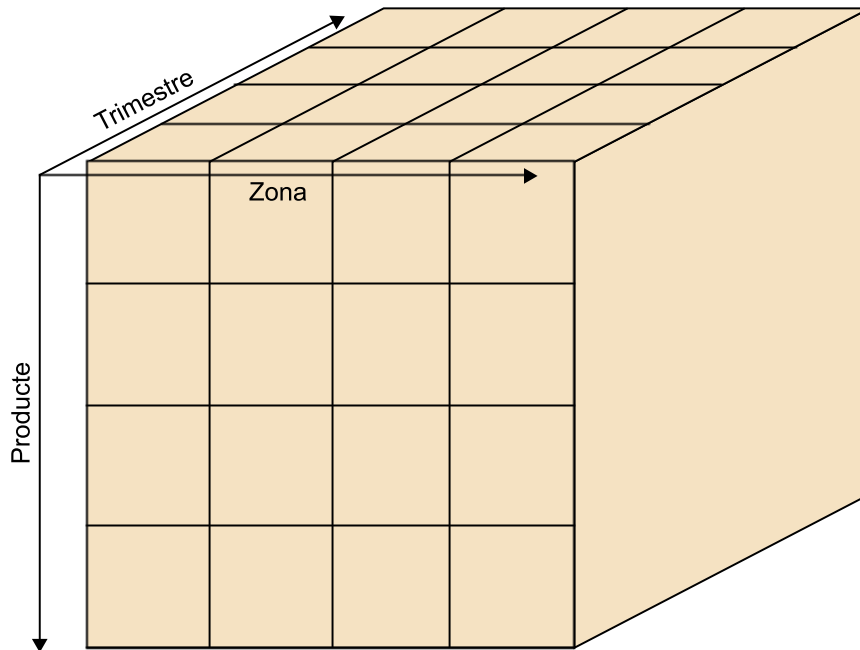
Vendes	Producte					
Zona		Producte1	Producte2	Producte3	...	Total
Zona1		120.000	188.050	123.000		<b>431.050</b>
Zona2		111.000	229.005	128.976		<b>468.981</b>
Zona3		230.400	234.000	90.000		<b>554.400</b>
Zona4		162.050	222.000	89.640		<b>473.690</b>
...						
<b>Total</b>		<b>623.450</b>	<b>873.055</b>	<b>431.616</b>		<b>1.928.121</b>

Així és com es mostrarien aquestes dades en un informe d'un **magatzem de dades** (*data warehouse*).

c) **Cub de dades** (tres dimensions). Correspon a una matriu de dades tridimensional.

### Exemple de cub de dades

Si s'hi afegeix una tercera dimensió temporal com els trimestres impositius, s'obté un cub de dades que organitza les vendes de productes per trimestres impositius i zones comercials. Cada cel·la conté les dades de *vendes* d'un producte concret, un trimestre concret i una zona concreta.



El canvi d'orientació d'un cub de dades s'aconsegueix mitjançant la rotació (pivotació) sobre un dels eixos. Per exemple, la rotació sobre l'eix de zones permetria mostrar les vendes per zona a les files, les vendes per trimestre a les columnes i les vendes per producte a la tercera dimensió.

**d) Hipercubde dades** (més de tres dimensions). Correspon a una matriu de dades  $n$ -dimensional ( $n > 3$ ), malgrat que no es pot representar gràficament ni visualitzar de manera senzilla.

## 6) BD paral·lela

SGBD que aprofita el paral·lisme d'unitats de procés o de memòria per al processament paral·lel d'un nombre significatiu de tasques de consulta i actualització, i de funcions de servei associades (registre de transaccions, maneig d'entrada/sortida i emmagatzematge en una memòria intermèdia – *buffer* – de dades), mitjançant la utilització concurrent de dos o més processadors i/o dispositius d'emmagatzematge. Això proporciona millores significatives en el temps de transacció i de resposta en BD de grans dimensions.

### Processament paral·lel

És la descomposició de les unitats de processament (instruccions individuals, seqüències d'instruccions o blocs de procés) en parts que el sistema operatiu o l'SGBD assignen, per a la seva execució simultània, a processadors diferents que treballen coordinadament en un ordinador amb multiprocessadors.

## 7) BD distribuïda

Les dades estan repartides entre diversos ordinadors (nodes) d'una xarxa informàtica. El gran avantatge d'aquest model és que permet unir informació de diferents localitzacions i accedir-hi sense tenir-ho tot centralitzat en un únic ordinador. Malgrat la descentralització de les dades, suportada en una arquitectura bàsica de tipus client-servidor, té l'inconvenient que la gestió és complicada.

El desenvolupament d'entorns d'Internet ha potenciat l'accés universal a BD distribuïdes a la *web* des de qualsevol ordinador per mitjà de *pàgines web*. Les **BD web** permeten l'accés a BD distribuïdes i heterogènies emmagatzemades

en *servidors web* en arxius compartits. Una capa de programari intermèdia (*middleware*) entre l'usuari i l'SGBD anomenada *passarel·la web* o interfície de passarel·la comuna (CGI, *common gateway interface*) permet executar programes externs per a obtenir la informació que, una vegada processada, torna al servidor en format HTML, la qual és enviada un altre cop al *navegador web*, on es pot visualitzar.

Molts SGBD actuals incorporen prestacions per a facilitar l'accés a BD per Internet. Hi ha dos mètodes principals:

- **Accés mitjançant scripts CGI.** Permet al servidor de la BD relacionar-se amb el servidor web mitjançant l'especificació CGI.
- **Accés mitjançant JDBC** (*Java data base connectivity*). Aquesta interfície de programació d'aplicacions (API) permet als programes en llenguatge Java accedir a BD relacionals mitjançant l'execució de consultes o sentències SQL. Una altra possibilitat és utilitzar l'estàndard de Microsoft ODBC (*open database connectivity*).

#### Per a saber-ne més

Podeu veure una definició d'ODBC quan es parla de BD accessibles per Internet en el subapartat "Tipus de bases de dades".

Els SGBD que donen accés a BD via web proporcionen els **avantatges** següents:

- Permeten un accés senzill, flexible, homogeni, eficient, barat i segur a les dades.
- Permeten l'accés a dades actualitzades contínuament.
- Donen suport a tipus molt diversos de dades difícils i costoses de recollir per altres mitjans. Per exemple, dades multimèdia com ara imatge, àudio i vídeo.
- Eviten els problemes de compatibilitat de formats i sistemes, facilitant l'accés des de pràcticament qualsevol plataforma.
- Proporcionen eines de cerca, indexació i relació que milloren les capacitats i potencialitats de les BD.

## 8) BD XML

Sistema aparegut a principi del segle XXI per la necessitat d'emmagatzemar i recuperar documents XML (que són estructures de dades semiestructurades). Hi ha dos enfocaments d'SGBD que suporten amb eficiència documents XML:

- **Extensions de BD per a XML.** Desglossen un document XML i l'integren en el seu model de base de dades corresponent (relacional o d'objectes).
- **SGBD XML natiu.** Respecten l'estructura del document, permeten fer-hi consultes i recuperen el document tal com es va guardar originalment.

### Exemples d'SGBD XML

#### a) Extensions d'SGBD per a XML:

- SGBDR per a XML: Microsoft SQLXML i Oracle XML DB són productes comercials.
- SGBDOO per a XML: Ozone/XML és de codi obert.

**b) SGBD XML nadius:** Tamino i X-Hive/DB són comercials, mentre que dbXML, eXist i Xindice són de codi obert.

## 9) BD multimèdia

Ofereix característiques que permeten emmagatzemar i consultar informació multimèdia que inclou imatges (fotografies, dibuixos...), vídeo (videoclips, vídeos domèstics, pel·lícules, noticiaris o programes de televisió), àudio (cançons, audiollibres, discursos o missatges telefònics) i documents de text (llibres, articles...).

Per a localitzar fonts multimèdia s'usen consultes de *recuperació basada en contingut*.

### Exemple de recuperació basada en contingut

En una BD de vídeos pot interessar localitzar tots els vídeos que continguin una determinada persona o un tipus d'activitat o esdeveniment, com per exemple riudes, incendis forestals o els gols marcats per un jugador o un equip.

Les BD multimèdia comprenen disciplines molt diverses: gestió de documents i registres, educació i difusió de coneixements, comunicació i entreteniment, control d'activitats en temps real... Es poden classificar en:

- **Aplicacions d'emmagatzematge** de gran quantitat de dades multimèdia (imatges de satèl·lit, fotografies de l'espai...) mitjançant magatzems centrals organitzats en diversos nivells.
- **Aplicacions de presentació** de vídeo o d'àudio en temps real, que es veuen o se senten a mesura que són enviats per l'SGBD.
- **Aplicacions de treball col·laboratiu** per a analitzar informació multimèdia en temps real. S'usen en camps com la telemedicina o l'enginyeria.

## 10) BD documental

Emmagatzema referències i/o text complet de documents de temàtica general (llibres, articles, cartes, contractes...), normalment sense format predefinit, que poden contenir text extens i dades multimèdia. Aquests documents s'indexen identificant paraules clau significatives (que figuren en el text) i les seves freqüències relatives. Les consultes es fan gràcies a aquestes paraules clau o per mitjà a d'un *thesaurus* d'estructura jerarquitzada. De vegades s'anomena BD textual o *sistema de recerca documental*.

Segons si inclouen o no el contingut complet dels documents descrits, les BD documentals es classifiquen en:

**a) BD referencial.** No conté els documents originals, sinó informació descriptiva i referències que permeten localitzar-los en un altre servei. També pot incloure enllaços per a obtenir-los per mitjà d'un altre programa.

### Exemples d'SGBD multimèdia

La majoria són SGBDO amb suport multimèdia: Informix Dynamic Server, DB2 Universal Database UDB) d'IBM, Oracle 8 o superior, Sybase. Altres sistemes permeten la recuperació d'imatges basada en el contingut: QBIC (*Query By Image Content*) d'IBM, Virage, Excalibur...

b) **BD factual.** Emmagatzema els documents font o originals de manera completa o conté tota la informació necessària per a donar resposta a les necessitats de l'usuari.

- **BD de text complet.** Constituïda pels mateixos documents en format digital. També pot incorporar camps amb informació complementària per a facilitar-ne la descripció i l'accés. Permet localitzar termes presents en el text del document. Es pot considerar un cas específic de BD multimèdia on només s'emmagatzemen i manipulen fonts de text.
- **Arxiu electrònic d'imatges.** Constituït per referències que tenen un enllaç a la imatge del document original. No permet localitzar termes presents en el text original.

#### Per a saber-ne més

Aquesta tipologia de BD es desenvolupa en l'apartat "Bases de dades documentals" al final d'aquest mòdul. També s'estudia en altres assignatures de la llicenciatura.

#### Exemples de sistemes de BD documentals

Alguns dels sistemes de BD documentals més usats són Knosys, Verity i Excalibur.

A tall de resum, a continuació us presentem una taula amb tots els tipus de bases de dades:

Tipus de BD	Acrònim	Estructura de dades	Estàndard del model	SGBD comercials
Jeràrquiques	BDJ	Arbre		IMS, System 2000, SABRE
En xarxa	BDX	Xarxa	Codasyl-DBTG	IDS
Relacionals	BDR	Taula	SQL	DB2, dBASE, Ingres, Oracle, Informix, Paradox, Access, Sybase, MySQL
D'objectes	BDO/BDOO	Objecte	ODMG	ObjectStore, Ardent (O2), Objectivity, Versant, Ontos
Objecte-relacionals	BDOR	Taula, objecte		UniSQL/X, Illustra, OpenODB, PostgreSQL, Ominiscience
Actives	BDA	Taula amb disparadors ( <i>triggers</i> )		Oracle, DB2 i Sybase (amb disparadors)
Deductives	BDD*	Taula amb regles lògiques		LDL, Validity
Temporals	BDT	Taula o objecte amb sèries de temps i versions de tuples o d'atributs	TSQL	Informix Universal Servidor amb <i>data-blades</i> de sèries de temps
Espacials	BDE			
Informació geogràfica	SIG			ARC/Info, ARC/Storm
Multidimensionals	BDMD	Hipercub		UniVerse (d'IBM)
Paraleles	BDP			
Distribuïdes	BDD*	Heterogènia	ODBC, JDBC	

L'acrònim BDD s'utilitza per a una tipologia o altra segons l'entorn de què es tracti. Els dos primers tipus són obsolets.



Tipus de BD	Acrònim	Estructura de dades	Estàndard del model	SGBD comercials
Web	BDWeb	Heterogènia	ODBC, JDBC	
XML	BDXML	Documents XML (semiestructurats)		Tamino, X-Hive/DB, dbXML, eXist, Ozone XML DB
Multimèdia	BDMM	Objectes LOB, TDA		Sybase, Oracle, DB2, ODB II, CA-Jasmine, DS Informix, QBIC
Documentals	BDD*	Documents extensos (textuals o no)		Knosys, Verity, Excalibur

L'acrònim BDD s'utilitza per a una tipologia o altra segons l'entorn de què es tracti. Els dos primers tipus són obsolets.

## 2.5. Bases de dades relacionals

Tal com s'ha comentat en els apartats anteriors, les bases de dades més esteses actualment són les relacionals, sobretot pel que fa a l'ús de particulars i departaments d'organitzacions.

Algunes entitats bancàries i hospitalàries encara treballen, però, en l'àmbit corporatiu amb bases de dades prerelacionals.

En aquest apartat es presenten els conceptes principals d'una base de dades relacional i els seus components: taules, registres, camps, etc.

En una base de dades relacional, la informació s'emmagatzema en taules, compostes per registres. Cada registre conté diversos valors, emmagatzemats en camps, que són de diferents tipus de dades. Tots els registres s'identifiquen amb una clau. Les diferents relacions entre els registres de dues o més taules s'implementen per mitjà de claus foranes.

### Per a saber-ne més

Per a tenir més informació sobre els diferents tipus de bases de dades, podeu consultar l'apartat "Tipus de bases de dades" d'aquest mateix mòdul.

### 2.5.1. Taules

Una taula és un conjunt de dades que agrupen totes les ocurrències o elements d'un mateix tipus.

Per a cada entitat d'informació diferent es fa servir una taula diferent, per exemple, tots els alumnes de la universitat, les assignatures de la carrera o la llista de professors.

### 2.5.2. Registres

Cada fila de la taula és un registre d'informació i descriu un element d'aquest conjunt d'informacions.

### Exemples de registres

Exemples de registres serien cada alumne de la universitat, cadascuna de les assignatures o cadascun dels professors, emmagatzemats a les taules d'alumnes, assignatures i professors.

### 2.5.3. Camps

Cada valor o informació dels registres possible és un camp.

#### Exemples de camps

Per exemple, en les taules que hem esmentat abans d'alumnes i professors de la universitat, el nom, el cognom, el NIF, l'adreça, el codi postal, la població o el telèfon serien camps.

O, per exemple, en la taula d'assignatures, el nom de l'assignatura, el seu codi o els crèdits (càrrega lectiva) serien camps.

### 2.5.4. Claus

La clau d'una taula són un o més camps que identifiquen, de manera única, els registres de la taula.

A continuació, es descriuen els diferents **tipus de claus** que pot tenir una taula:

- **Claus candidates** (*candidate key*): són totes les possibles claus que permeten identificar de manera única cadascun dels registres d'una taula.
- **Clau primària** (*primary key*) o principal: és la clau escollida, d'entre les claus candidates d'una taula, per a identificar de manera única els registres. S'acostuma a escollir la clau candidata formada pel menor nombre de camps, i se sol posar al principi de la taula. També rep el nom d'*identificador*.
- **Claus alternatives** (*alternative key*) o secundàries: són les claus candidates que no són escollides com a clau primària d'una taula.
- **Clau forana** (*foreign key*), aliena o externa: és el camp (o conjunt de camps) que pot formar part o no de la clau primària d'una taula (anomenada *taula filla* o referenciant), que, a la vegada, és clau primària o clau alternativa en una altra taula (anomenada *taula mestra* o referenciada) amb la qual es relaciona. Permet establir relacions en cascada entre taules, i pot no existir en una taula. El seu objectiu principal és mantenir la integritat de la informació i és una part important en la normalització de la BD.

D'altra banda, una **clau simple** està formada per un únic camp, i una **clau composta**, per més d'un camp.

Tot i que els registres d'una taula es poden identificar per més d'una combinació possible dels seus camps, és a dir, pot haver-hi diverses *claus candidates*, només una d'aquestes és escollida com a *clau primària*. La resta queden com a *claus alternatives*.

Una taula pot tenir N claus candidates, 1 clau primària, N-1 claus alternatives i M claus foranes, on N és igual o més gran que 1 i M, igual o més gran que 0. Dit d'una altra manera, una taula té, com a mínim, una clau candidata (que, en el pitjor dels casos, estarà formada per tots els camps de la taula i serà escollida com a clau primària). La clau primària, normalment, és un petit subconjunt dels camps de la taula. A més, una taula pot tenir cap o diverses claus foranes procedents de diferents taules referenciades.

### Exemples de claus

Continuant amb els exemples anteriors, els alumnes es poden identificar, de manera única, pel seu NIF o pel codi intern a la universitat. Ambdós camps serien *claus candidates*. Un seria escollit com a *clau primària* i l'altre quedaria com a *clau alternativa*. D'altra banda, el nom i cognoms no poden ser *clau*, ja que no es pot garantir que no hi hagi dos alumnes que es diguin igual.

La clau primària d'una taula permet relacionar els registres d'aquesta amb els d'altres taules per mitjà de les claus foranes, com s'explica a continuació.

### 2.5.5. Relacions i claus foranes

Les relacions entre els registres de les diferents taules s'implementen mitjançant la vinculació de valors. Aquesta vinculació s'estableix entre la clau forana d'una taula i la clau primària d'una altra.

El **tipus de relació** entre dues taules s'indica mitjançant el nombre de registres d'una taula que es poden relacionar amb un registre de l'altra taula i viceversa. Això es pot expressar separant pel signe de dos punts (":") aquests valors. És a dir, *1:1*, *1:n* o *m:n*, segons el cas; on les variables *n* i *m* simbolitzen *molts*.

El tipus de relació entre dues taules entronca amb el concepte de cardinalitat d'una relació entre entitats (model conceptual), que és perfectament aplicable a les relacions entre taules en el model relacional.

Entre dues taules es poden establir tres **tipus de relació**:

**a) Un a un.** A cada registre d'una taula li correspon un i només un registre de l'altra. Es representa per *1:1*.

#### Per a saber-ne més

L'exemple de la relació un a molts exposat en el subapartat següent ("Relacions") mostra la *clau forana* CodiDepartament de la taula PROFESSOR necessària per a implementar la relació amb la *clau primària* de la taula DEPARTAMENT.

#### Per a saber-ne més

En el subapartat "El model entitat-relació" de l'apartat "Disseny de bases de dades" es desenvolupa aquest concepte aplicat a les relacions entre entitats i se'n donen exemples.

b) **Un a molts.** A cada registre de la primera taula (la *taula mestra*) li correspon un o més registres de la segona (la *taula filla*). Es representa per  $1:n$  o  $n:1$  (la simbologia utilitzada per Microsoft Access és  $1:\infty$  i  $\infty:1$ ).

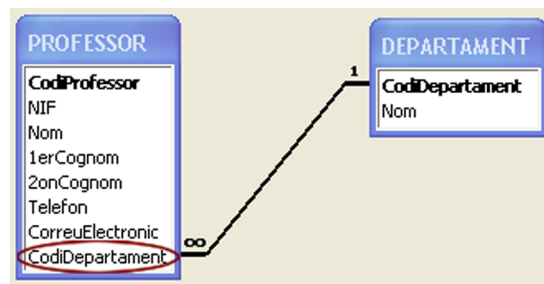
#### Exemple de relació un a molts (i de clau forana)

Per a implementar la relació que hi ha entre els professors d'un centre d'estudis i el departament al qual estan adscrits, es defineix, en la taula PROFESSOR, una *clau forana* que es relaciona amb la *clau primària* de la taula DEPARTAMENT.

D'aquesta manera, en el registre de cada professor hi haurà un camp on s'emmagatzemarà el codi del departament al qual està adscrit, com mostra la taula PROFESSOR creada amb l'SGBD Microsoft Access de la figura següent.

CodiProfessor	NIF	Nom	1erCognom	2onCognom	Telefon	CorreuElectronic	CodiDepartament

Això permet relacionar les taules PROFESSOR i DEPARTAMENT tal com es veu a continuació.

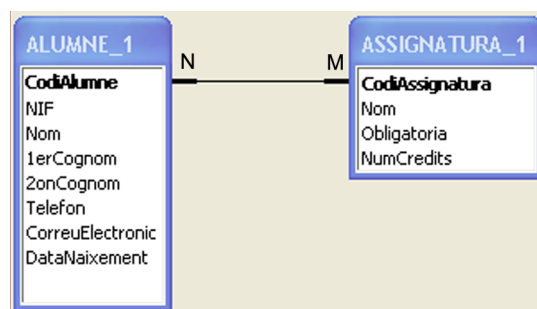


La figura mostra les *claus primàries* en negreta i la relació *un a molts* ( $1:\infty$ ) entre les dues taules mitjançant una línia que uneix la *clau primària* de la *taula mestra* DEPARTAMENT (en aquest extrem es posa el nombre 1) i la *clau forana* de la *taula filla* PROFESSOR (en aquest extrem es posa el signe  $\infty$ ).

c) **Molts a molts.** A cada registre de la primera taula li poden correspondre diversos registres de la segona i viceversa. Es representa per  $m:n$ .

#### Exemple de relació molts a molts

La relació que hi ha entre els alumnes d'un centre d'estudis i les assignatures en què es matriculen (inscriuen) és del tipus *molts a molts*, ja que un alumne es pot inscriure en diverses assignatures i en cada assignatura s'hi poden inscriure diversos alumnes.



La figura mostra aquest tipus de relació  $m:n$ , no permesa en un SGBD relacional.

Els SGBD relacionals no permeten definir aquest tipus de relació entre dues taules. Si es presenta el cas, cal crear una tercera taula, anomenada **taula intermèdia**, que es relacioni amb les altres dues mitjançant relacions  $1:n$  i  $n:1$ .

### Exemple de generació d'una taula intermèdia

Per a implementar la relació *molts a molts* de l'exemple anterior cal elaborar la *taula intermèdia* "Matriculació", que tindrà com a mínim dos camps, l'identificador de l'assignatura i l'identificador de l'alumne, i el conjunt dels dos camps serà la seva *clau primària*.

Ara bé, si per exemple un alumne es matricula més d'una vegada en una mateixa assignatura, ja no n'hi ha prou amb aquesta *clau primària*. Caldrà afegir-hi un tercer camp que ajudi a identificar les diverses matriculacions. Aquest tercer camp pot ser, per exemple, la data de matriculació. Ara, la *clau primària* estarà formada per tres camps.



La figura creada amb Microsoft Access mostra les relacions  $1:n$  i  $n:1$  existents entre les tres taules. La *taula intermèdia* surt encerclada i els camps que formen la *clau primària*, en negreta.

### 2.5.6. Tipus de dades

Els camps d'una taula separen els tipus d'informació que hi conté.

La persona que crea la taula (dissenyador de la base de dades) defineix els camps que tindrà aquesta taula, és a dir, quines informacions s'emmagatzemaran per a cada registre.

Els diferents sistemes de gestió de bases de dades (SGBD) ofereixen una varietat de diferents tipus de dades.

A continuació definim els principals:

- **Text:** cadena o sèrie de caràcters alfanumèrics no gaire llarga. Pot ser informació textual (el nom d'una persona o d'una empresa, una adreça...) que inclogui lletres, dígitos decimals, signes de puntuació, espais i/o altres símbols imprimibles. També pot ser un nombre (telèfon, codi postal...), que serà tractat com una sèrie de dígitos i no com el valor numèric que representa i, per tant, no es podrà utilitzar per a fer càlculs.
- **Numèric:** nombre que requereix càlculs matemàtics (excepte si està relacionat amb un valor monetari; en aquest cas, s'utilitza el tipus Moneda).
- **Moneda:** valor numèric amb un format de presentació configurat per a representar valors monetaris amb decimals. S'utilitza per a evitar

l'arrodoniment de decimals en els càlculs (MS Access té una precisió de fins a 4 decimals).

### **Format de visualització**

És la manera com es presenten (per pantalla o impresos) els valors dels diferents tipus de dades. Els SGBD solen proporcionar formats de visualització predefinits per als tipus Numèric, Data/Hora, Moneda i Sí/No. A més, es poden definir formats personalitzats per a tots els tipus de dades, excepte per al tipus Binari.

- **Data i hora:** informació temporal que representa dates i hores.
- **Lògic:** qualsevol tipus de dada en què només hi hagi dos valors possibles (sí o no, vertader o fals, activat o desactivat, encès o apagat, aprovat o suspès...). S'emmagatzema un dels dos valors possibles com a 1 o 0, respectivament.
- **Memo:** informació textual alfanumèrica més o menys extensa de llargada variable i de tipus memoràndum. Per exemple: descripcions, comentaris, notes, resums...
- **Binari:** objecte o arxiu binari com ara una imatge, àudio, vídeo, document de text amb format (creat amb un processador de text), full de treball (creat amb un gestor de fulls de càlcul) o qualsevol altre tipus d'arxiu binari creat en altres programes que es vincula o s'incrusta com a dada en el camp de la BD mitjançant el protocol OLE (*Object Linking and Embedding*) de vinculació i incrustació d'objectes.
- **Autonumèric:** valor numèric sencer únic que es genera automàticament quan s'agrega cada nou registre. Pot ser seqüencial (amb increments d'una unitat; el primer registre té el valor 1, el segon el valor 2, i així successivament) o aleatori. Un cop generat no es pot actualitzar i si s'elimina, el valor no es pot tornar a usar. El tipus incremental, també anomenat *comptador*, és el més comú i adequat per a utilitzar com a identificador únic o clau primària.
- **Hipervincle:** adreces URL o rutes d'accés segons la convenció universal d'assignació de noms (UNC).

El **tipus de dada** que s'utilitzarà en un camp d'una taula (valors del camp) es pot **decidir en funció de:**

- El tipus de valors que es volen permetre en el camp. Per exemple, no es pot emmagatzemar text en un camp definit per a un tipus de dades Numèric.
- El tipus d'operacions que es vol fer amb els valors del camp. Per exemple, es poden sumar els valors de camps de tipus Numèric i Moneda, però no els de camps de tipus Text o Binari.
- La possibilitat d'ordenar els valors del camp. Per exemple, els valors d'un camp de tipus Binari no es poden ordenar.
- El sistema d'ordenació dels valors del camp. Per exemple, en un camp de tipus Text, els nombres s'ordenen com a cadenes de caràcters (per exemple 1, 10, 100, 2, 20, 200...), no com a valors numèrics (1, 2, 10, 20, 100, 200...). Per a ordenar nombres com a valors numèrics convé utilitzar el tipus Numèric o Moneda. D'altra banda, molts *formats* de data tampoc no

s'ordenen correctament si s'utilitza un camp de tipus Text. Per a garantir un ordre correcte cal usar un camp de tipus Data/Hora.

- L'espai d'emmagatzematge que ocupen els valors del camp. Segons les característiques de les dades textuais es poden usar tres tipus de dada de diferent grandària:
  - Text per a emmagatzemar *text curt*: dades com noms, adreces i qualsevol nombre que no necessiti càlculs, com números de telèfon, números de producte o codis postals (a Access, fins a 255 caràcters).
  - Memo per a emmagatzemar *text extens* (a Access, fins a 64.000 caràcters).
  - Binari per a emmagatzemar *documents de text amb format* més o menys llargs.

En els camps amb dades numèriques es pot especificar la mida del camp en un rang de valors. Per exemple, a Access, per a nombres enters (sense decimals): byte: de 0 a 255, enter: entre -32.768 i 32.767, enter llarg (predeterminat): entre -2.147.483.648 i 2.147.483.647.

### 2.5.7. Consultes

En aquest apartat, s'explica com tracten els SGBD la informació emmagatzemada en una BD, quin tipus d'eines ofereixen per a dur a terme aquesta tasca i les funcionalitats de què disposen per a consultar les dades.

Les consultes tenen funcions per a seleccionar diversos registres de les taules i diversos camps o per a fer accions de càlcul. Una prestació especial és la sentència JOIN (d'SQL), que permet combinar els registres de diferents taules. És a dir, fer consultes en les quals intervenen diverses taules alhora.

#### Exemples de disseny de consultes

Tot i que no és l'objecte d'aquesta assignatura, a continuació s'exemplifica el disseny de consultes (amb Microsoft Access).

Es consulten les dades dels alumnes d'un centre d'estudis i les assignatures en les quals es matriculen.

#### 1) Disseny d'una consulta per a obtenir totes les dades dels alumnes del centre

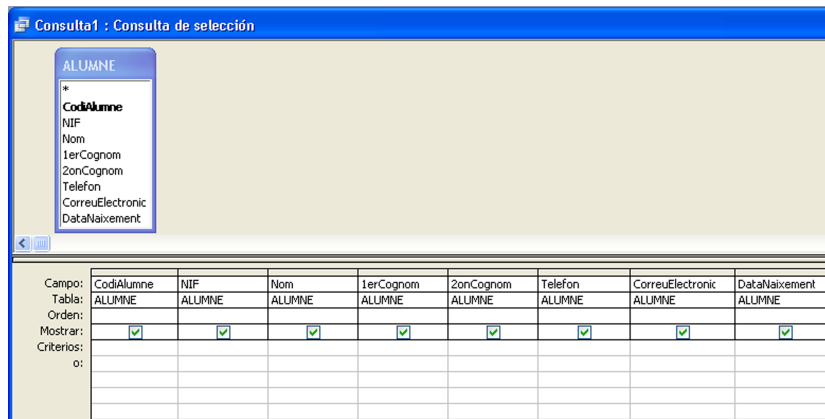
En el disseny d'una consulta en Access cal distingir dues àrees diferenciades. En el panell superior se situen les taules (amb els camps) que intervenen en la consulta i es mostren les relacions que hi ha entre aquestes taules. En el panell inferior es detallen els camps específics que han de figurar en la consulta i la taula a la qual pertanyen, a més d'altres aspectes (com s'ordenaran les dades, si s'ha de mostrar o no la columna, els criteris de filtratge, etc.).

#### La sentència JOIN

És una instrucció d'SQL que permet combinar (concatenar) registres (files) de dues o més taules en una BD relacional, si compleixen una condició d'aparellament. És un component de la instrucció SELECT. Matemàticament, és una funció de composició relacional, l'operació fonamental en l'àlgebra relacional.

#### Per a saber-ne més

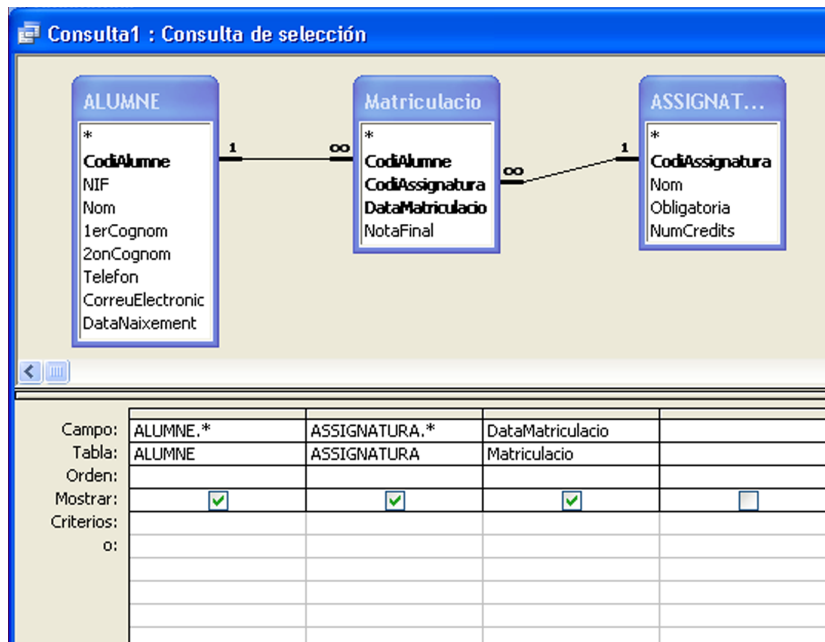
En el subapartat "Taules intermèdies" podeu veure l'exemple de referència de la BD d'un centre d'estudis, on es mostra com es relacionen les taules Alumne, Assignatura i Matriculació.



## 2) Disseny d'una consulta per a obtenir totes les dades de les matriculacions del centre

En la fila de camps al panell inferior d'aquesta consulta, figuren els noms de dues taules acompanyats d'un asterisc. Això vol dir que es mostraran tots els seus camps.

Per tant, el resultat de la consulta mostrarà totes les dades dels alumnes i de les assignatures en què s'han inscrit, a més de la data de matriculació.



## 2.6. Llenguatge de bases de dades relacionals

Qualsevol SGBD disposa d'un llenguatge de BD que té dos mòduls per a fer funcions diferents:

- El **llenguatge de definició de dades** (DDL, *data definition language*), especialitzat en la creació de la BD; és a dir, en la definició de l'esquema de la BD.
- El **llenguatge de manipulació de dades** (DML, *data management language*), especialitzat en la utilització i la gestió de la BD; és a dir, en el manteniment de la informació i la realització de consultes.



Per a comunicar-se amb l'SGBD, l'usuari, tant si és un programa d'aplicació com un usuari final, utilitza aquest llenguatge de BD. Hi ha molts llenguatges diferents segons el **tipus d'usuari** per al qual estan pensats i el que hi han de poder expressar:

- Els **usuaris informàtics** molt experts (programadors...) que volen escriure processos complexos necessitaran llenguatges complexos.
- Els **usuaris finals esporàdics** o ocasionals que només fan consultes necessitaran un llenguatge molt senzill, encara que doni un rendiment baix en temps de resposta.
- Els **usuaris finals quotidians** especialitzats o, fins i tot, dedicats exclusivament a treballar amb la BD necessitaran llenguatges molt eficients i compactes especialitzats en tipus concrets de tasques, encara que no siguin fàcils d'aprendre.

El llenguatge de consulta estructurat **SQL** (*structured query language*) es considera el llenguatge estàndard de BD relacionals. Per això, s'estudiarà el DDL i el DML que utilitza l'SQL.

Les característiques principals d'SQL són la senzillesa, el caràcter estàndard i que és un llenguatge declaratiu o no procedimental (és a dir, perquè SQL executi una acció concreta no s'especifica com ho ha de fer, sinó què es vol obtenir).

El llenguatge SQL té **instruccions** de tres tipus:

- Instruccions de **definició de dades** (tipus DDL). Per exemple, CREATE TABLE per a definir les taules, els camps i les restriccions.
- Instruccions de **gestió de dades** (tipus DML). Per exemple, SELECT per a fer consultes i INSERT, UPDATE i DELETE per al manteniment de les dades (inserir, actualitzar i esborrar registres, respectivament).
- Instruccions de **control de l'entorn**. Per exemple, COMMIT i ROLLBACK per a delimitar transaccions.

A continuació, s'exposen les instruccions més importants de definició i de manipulació que utilitza l'SQL.

### 2.6.1. Llenguatge de definició de dades (DDL)

El llenguatge de definició de dades proporciona ordres per a definir, eliminar i modificar taules, i també per a crear índexs i vistes.

Les ordres o instruccions SQL més importants del DDL són:

- **CREATE TABLE:** crea una taula amb els camps i els seus tipus de dades. La seva sintaxi és la següent:  
**CREATE TABLE n\_taula (n\_camp1 t\_camp1,... n\_campN t\_campN)**, on *n\_taula* és el nom de la taula que es crearà, *n\_camp1* és el nom del primer camp, *t\_camp1* és el tipus de dades del camp1...  
Cada camp té associat un tipus de dades, i els més comuns són: INTEGER per a nombres enters; DECIMAL (x,y) per a nombres reals, on *x* representa la longitud total del camp i *y* el nombre de decimals; CHAR (n) per a cadenes de caràcters de longitud *n* (entre 1 i 255); DATE per a dates, etc.
- **ALTER TABLE:** afegeix nous camps a una taula existent, els modifica o els esborra. No és una ordre estàndard. La seva sintaxi és la següent:  
**ALTER TABLE n\_taula ADD (n\_camp t\_camp)** per a afegir camps a la taula.  
**ALTER TABLE n\_taula MODIFY (n\_camp t\_camp)** per a modificar camps de la taula.  
**ALTER TABLE n\_taula DROP n\_camp** per a esborrar camps de la taula.
- **DROP TABLE:** esborra l'esquema de la BD, és a dir, destrueix tant les dades contingudes a la taula com l'estructura d'aquesta. La seva sintaxi és la següent:  
**DROP TABLE n\_taula.**

### 2.6.2. Llenguatge de manipulació de dades (DML)

El llenguatge de manipulació de dades està basat en l'àlgebra relacional i inclou ordres per a inserir, suprimir i modificar registres (files) de la BD.

Les ordres SQL més importants del DML són:

- **INSERT:** insereix nous registres (files) a una taula. La seva sintaxi és la següent:  
**INSERT INTO n\_taula VALUES (valor\_camp1,... valor\_campN).**
- **UPDATE:** modifica valors de determinats camps (columnes). La seva sintaxi és la següent:  
**UPDATE n\_taula SET camp=valor\_camp.**  
**UPDATE n\_taula SET camp=valor\_camp (WHERE condició).**  
Si s'especifica una condició, únicament s'actualitzen els valors dels camps dels registres que compleixen la condició.
- **DELETE:** esborra registres (files) de la taula especificada. La seva sintaxi és la següent:  
**DELETE FROM n\_taula.**  
Si no s'especifica cap condició, s'eliminaran tots els registres de la taula. És a dir, deixa la taula buida, com si s'acabés de crear, però no en destrueix

l'estructura. No s'ha de confondre amb l'ordre DROP, que, a més a més d'esborrar la informació, també elimina l'estructura.

**DELETE FROM n\_taula (WHERE condició)**

Si s'especifica una condició, únicament s'esborren els registres que satisfan l'esmentada condició. Aquesta és la seva forma més usual.

- **SELECT**: és l'ordre principal d'SQL i s'utilitza per a consultar taules. És molt flexible i admet moltes variacions. Selecciona un conjunt de registres (files) d'una o més taules que compleixen una condició. Si no s'especifica la condició, se seleccionen tots els registres de la taula. També permet crear un filtre seleccionant només alguns camps de la taula. La seva sintaxi és la següent:

**SELECT FROM n\_taula (WHERE condició) (ORDER BY campX)** per a seleccionar tots els camps de la taula. Aquesta és la seva forma més usual.

**SELECT camp1,... campN FROM n\_taula (WHERE condició) (ORDER BY campX)** per a consultar únicament un conjunt de camps.

**SELECT camp1,... campN FROM n\_taula1,... n\_taulaM (WHERE condició) (ORDER BY campX)** per a consultar camps de més d'una taula.

### 2.6.3. Eines d'interfície

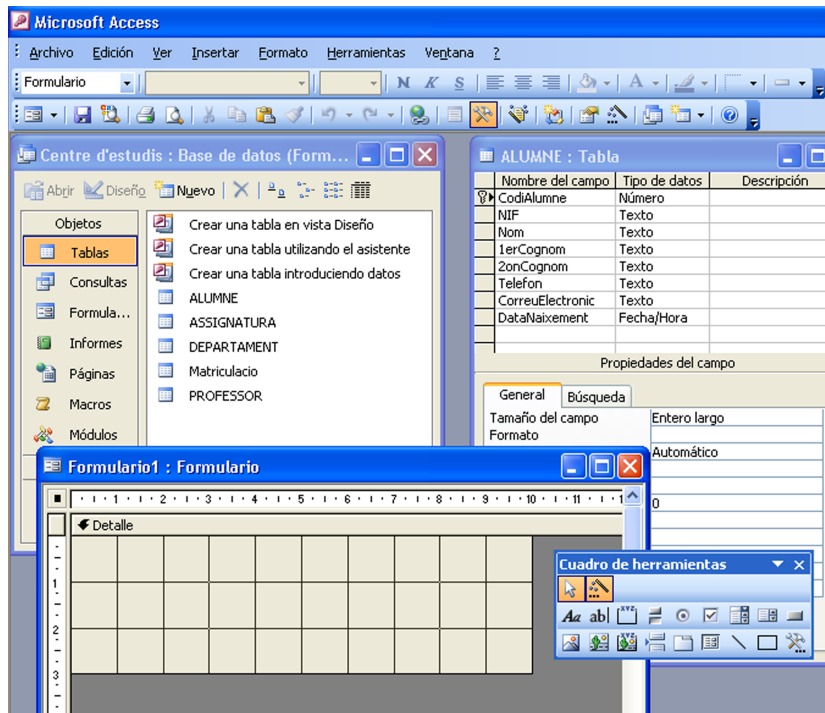
Encara que quasi tots els SGBD del mercat tenen l'SQL com a llenguatge natiu, ofereixen altres possibilitats:

- **Llenguatges de quarta generació (4GL, 4th Generation Languages)** d'un nivell molt alt, que solen combinar elements procedimentals amb elements declaratius. Pretenen fer molt fàcil no tan sols el tractament de la BD, sinó també la definició de menús, pantalles i diàlegs.
- **Interfícies visuals** molt fàcils de fer servir, que permeten utilitzar les BD seguint l'estil de quadres de diàleg amb finestres, icones i ratolí, típic de les aplicacions en l'entorn Windows. No solament són útils als usuaris no informàtics, sinó que faciliten molt la feina als usuaris informàtics: permeten consultar i actualitzar la BD, i també definir-la i actualitzar-ne la definició amb molta facilitat i claredat. Van aparèixer al final dels anys vuitanta i des de llavors han proliferat.

#### Exemple d'interfície visual

Un exemple clar d'eina amb interfície visual és Microsoft Access, que permet crear els diferents elements d'una BD (que aquí s'anomenen *objectes*: taules, consultes, formularis, informes...) per a gestionar-ne i extreure'n la informació per mitjà de finestres (*windows*) sense haver d'utilitzar, si no es vol, el llenguatge SQL.

La imatge següent mostra alguns elements de la interfície d'usuari d'aquest SGBD (el panell d'exploració, una taula i un formulari).



#### 2.6.4. Programació de bases de dades

Si es vol escriure un programa d'aplicació que treballi amb la BD, segurament es farà servir el mateix llenguatge de programació utilitzat per a desenvolupar l'aplicació informàtica (Pascal, Cobol, Basic, C...). Però, generalment, aquests llenguatges no tenen instruccions per a les BD. Llavors, es pot recórrer a una de les **dues opcions** següents:

- **Crides a funcions:** al mercat hi ha llibreries de funcions especialitzades en BD; per exemple, les llibreries ODBC. Només cal incloure crides a les funcions que es vulgui dins del programa escrit amb el llenguatge de programació habitual. Les funcions s'encarregaran d'enviar les instruccions (generalment en SQL) en temps d'execució a l'SGBD.
- **Llenguatge hostatjat:** una altra possibilitat consisteix a incloure directament les instruccions del llenguatge de BD en el programa. Però això exigeix utilitzar un precompilador especialitzat que accepti en el llenguatge de programació habitual les instruccions del llenguatge de BD. Llavors es diu que aquest llenguatge (quasi sempre l'SQL) és el llenguatge hostatjat o incorporat (*embedded*), i el llenguatge de programació (Pascal, C, Cobol, etc.) és el llenguatge amfitrió (*host*).

## 2.7. Resum

Una **base de dades** és un conjunt de dades organitzat als continguts del qual es pot accedir amb facilitat i que es poden administrar i actualitzar fàcilment.

Les BD eviten les dades redundants, garanteixen la independència de les dades respecte als programes que les fan servir, emmagatzemen les dades juntament amb les seves relacions i s'hi pot accedir de diferents maneres. És a dir, resolen la majoria de problemes que presenten els fitxers.

Les **característiques** principals que ha de complir un sistema de BD són:

- És accessible simultàniament per a diversos usuaris, cadascun a una determinada informació.
- Es controla l'accés de diferents usuaris a les dades i es garanteix la confidencialitat i la seguretat.
- Les dades es poden emmagatzemar sense redundància.
- Permet utilitzar diferents mètodes d'accés i assegurar flexibilitat en les cerques.
- Disposa de mecanismes de recuperació d'informació en cas que falli el maquinari.
- El suport físic on s'emmagatzemen les dades es pot canviar sense que se'n ressentin aquestes ni els programes que les fan servir.
- La modificació del seu contingut i de les relacions entre les dades no afecta els programes que les utilitzen.
- Disposa d'una interfície d'usuari que permet d'utilitzar-la de manera còmoda i flexible.

L'**SGBD** (sistema gestor de bases de dades) és el conjunt de programari destinat a la creació, gestió, control i manipulació de la informació emmagatzemada en una BD.

Entre les principals **funcions d'un SGBD** destaquen:

- Definició de l'esquema de la BD, per mitjà d'un **llenguatge de definició de dades** (DDL).
- Accés a la informació des d'un llenguatge d'alt nivell anomenat **llenguatge de manipulació de dades** (DML).
- Accés a la informació en mode conversacional per mitjà d'una **interfície d'usuari**.

- Interacció amb el sistema operatiu per a treballar amb els fitxers de dades que gestiona, mitjançant un mòdul anomenat **gestor de dades**.
- Els **objectius d'un SGBD** són els següents:
- Permetre fer **consultes no predefinides i complexes**.
- Donar **flexibilitat** als canvis i obtenir **independència** entre les dades i els programes.
- Facilitar l'**eliminació de la redundància** per a evitar la inconsistència i incoherència de dades.
- Permetre l'**accés concurrent dels usuaris** a la mateixa BD.
- Permetre definir **autoritzacions i drets d'accés** a diferents nivells per a garantir la **confidencialitat** i la **seguretat** de les dades.

Històricament, s'han diferenciat tres tipus de BD:

- **BD jeràrquiques**. Sistemes, apareguts els anys seixanta, en què la informació es representava en forma d'arbre. El seu inconvenient principal era que no totes les BD s'adaptaven a aquesta estructura.
- **BD en xarxa**. Van aparèixer per a resoldre el problema de les anteriors. Permetien qualsevol tipus de relació i, per tant, representar qualsevol conjunt d'informació. El seu inconvenient principal era la seva falta de flexibilitat.
- **BD relacionals**. Van aparèixer els anys setanta per a obtenir més flexibilitat en el tractament de les dades i són les més utilitzades des dels anys vuitanta.

Altres tipus de BD són les següents:

- **BD d'objectes**. Emmagatzemen la informació en classes i subclasses d'objectes complets (estat i comportament) i incorporen conceptes de la programació orientada a objectes.
- **BD relacionals esteses**. Són evolucions del model relacional per a obtenir més capacitat expressiva, incorporant funcionalitats d'altres models de dades. Es distingeixen:
  - **BD relacionals amb frontal OO**: afegeixen una capa d'OO sobre un SGBD relacional.
  - **BD objecte-relacionals**: integren els dos models, incorporant nous tipus de dades, operacions per a gestionar el comportament, noves capacitats consultiva i expressiva...
  - **BD actives**: permeten definir regles que s'activen quan succeeixen determinats esdeveniments, que inicien l'execució d'una acció si es donen unes condicions.
  - **BD deductives**: inclouen mecanismes d'inferència basats en regles deductives que permeten generar informació addicional a partir dels fets emmagatzemats en la BD.
- **BD temporals**. Suporten la variable temps i altres conceptes temporals. Permeten emmagatzemar un historial de canvis i consultar l'estat (actual

o passat) de la BD. La informació temporal es pot referir a fets puntuals o a fets de durada.

- **BD espacials.** Proporcionen conceptes per a interpretar i especificar les característiques espacials d'objectes que es troben en un espai multidimensional. Poden ser *cartogràfiques* (inclouen conceptes geomètrics bidimensionals) i *meteorològiques* (inclouen informació de punts espacials tridimensionals).
- **Sistemes d'informació geogràfica.** Permeten emmagatzemar, analitzar i representar gràficament informació que descriu propietats geogràfiques, gestionada per una BD que conté dades espacials (físiques, polítiques, administratives...) i dades no espacials (demogràfiques, econòmiques, comercials...).
- **BD multidimensionals.** Organitzen les dades en estructures matricials de diverses dimensions (per exemple, productes, zones comercials, períodes impositius) i disposen de llenguatges especials per a fer consultes complexes. Faciliten la visualització de dades en qualsevol combinació de dimensions per mitjà de diferents criteris, jerarquies (nivells d'agrupament en categories més generals o de disgregació en categories més concretes) i operacions de càlcul intensiu (agregació, suma, mitjana...).
- **BD paral·leles.** Aprofiten el paral·lelisme d'unitats de procés per al processament paral·lel d'un gran nombre de tasques de consulta i actualització mitjançant la utilització concurrent de dos o més processadors i/o dispositius d'emmagatzematge. Això millora significativament el temps de transacció i de resposta en BD de grans dimensions.
- **BD distribuïdes.** Reparteixen les dades entre diversos ordinadors connectats en xarxa i es basen en una arquitectura client-servidor. Tenen l'avantatge que descentralitzen la informació i la fan més disponible localment, però l'inconvenient que la gestió és més complicada.
- **BD accessibles per Internet.** Permeten l'accés flexible, homogeni, eficient i segur, des de qualsevol plataforma, a BD distribuïdes i heterogènies (que contenen tipus de dades molt diversos, com per exemple dades multimèdia) actualitzades permanentment i emmagatzemades en servidors web, amb la qual cosa s'evita problemes de compatibilitat de formats i sistemes.
- **BD XML.** Emmagatzemen documents XML (que són estructures semiestructurades) amb l'eficiència de les BD convencionals. Els SGBD XML poden ser *extensions* del model relacional o d'objectes (que desglossen el document XML) o *natis* (que respecten l'estructura del document i el recuperen tal com es va guardar originalment).

- **BD multimèdia.** Ofereix característiques que permeten emmagatzemar i consultar informació multimèdia que inclou imatges, vídeo, àudio i documents de text. La seva aplicació comprèn disciplines com la gestió documental, la difusió de coneixements, la comunicació, l'entreteniment, el treball col·laboratiu i el control d'activitats en temps real.
- **BD documentals o textuals.** Emmagatzemen les referències i/o el text complet de documents de temàtica general que poden contenir text extens i dades multimèdia. Es consulten per mitjà de paraules clau o de termes d'un *thesaurus* d'estructura jerarquizada.

Es poden classificar en BD *referencials* (que contenen referències per a poder localitzar els documents originals), *BD de text complet* (que emmagatzemen el contingut complet dels documents originals) o *arxius electrònics d'imatges* (constituïts per referències que tenen un enllaç a la imatge del document original).

Una **BD relacional** està formada per **taules**, que són estructures en **files** (els registres d'informació) i **columnes** (els camps d'informació dels registres). Cada taula té una **clau** que és un camp o conjunt de camps (columnes) que identifica de manera única cada registre (fila) i permet relacionar la taula amb altres taules de la BD.

Una **taula** d'una BD relacional compleix les **condicions** següents:

- Totes les files són registres del mateix tipus. Per a emmagatzemar registres de diferents tipus s'utilitzen diferents taules.
- Cada columna s'identifica per un nom de camp.
- No accepta noms de camps (columnes) repetits.
- No accepta registres (files) duplicats.
- L'ordre dels registres és indiferent.
- El seu contingut és independent del suport d'emmagatzematge físic de les dades.
- Es relaciona amb altres taules fent coincidir valors iguals dels registres d'ambdues, mitjançant claus.

Exemples destacats d'SGBD relacionals són Microsoft Access, el servidor de BD Oracle, Microsoft SQL Server o MySQL.



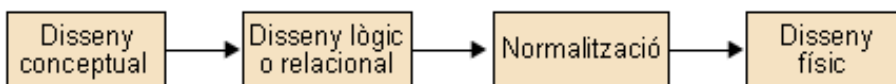
### 3. Disseny de bases de dades

Un error molt freqüent a l'hora de dissenyar una base de dades relacional és generar el model relacional (taules, camps, etc.) directament a partir de la identificació de les necessitats que té l'usuari o organització, utilitzant el sistema gestor de bases de dades.

Aquest mètode pot ser vàlid quan es tracta de crear bases de dades molt senzilles i amb poca complexitat (poques taules i amb poques relacions).

En canvi, quan la complexitat augmenta s'ha de recórrer a unes metodologies que garanteixin la definició correcta del nostre sistema d'informació i les seves estructures, facilitant també que no hi hagi errors i simplificant el treball posterior, el tractament de les dades i l'evolució del sistema.

La metodologia o etapes recomanades són les següents:



#### a) Disseny conceptual

Durant el disseny conceptual es construeix un esquema de la informació que s'utilitza a l'organització (empresa, associació, departament, etc.) independentment de qualsevol consideració física. És a dir, independentment de quin model de bases de dades s'utilitzarà posteriorment i en quin sistema gestor de base de dades s'implementarà.

L'esquema resultant l'anomenem "**esquema conceptual**".

En aquest pas es descriuen les entitats, atributs i relacions de la informació que s'ha d'emmagatzemar posteriorment i que han de gestionar els sistemes informàtics.

La notació més popular per a crear l'esquema conceptual és el model entitat-relació, que es descriu en l'apartat següent.

Un cop que es té el model conceptual, es pot generar el disseny lògic.

#### b) Disseny lògic

El disseny lògic és el procés de construir un esquema de la informació, basant-se en un model de base de dades (en xarxa, jeràrquic, relacional, etc.) independentment, però, de l'SGBD que després s'utilitzarà per a gestionar la informació.

En el marc d'aquest mòdul, entendrem per model lògic la creació del model relacional.

En aquesta etapa, es transformarà l'esquema conceptual en el conjunt de taules, camps i interrelacions entre les taules per tal de poder crear més tard la bases de dades.

### c) Normalització

Després d'aquest pas, i abans de crear l'esquema físic, és convenient normalitzar la base de dades, una tècnica que s'utilitza per a verificar que les taules obtingudes no tenen dades redundants i estan "optimitzades". En els propers apartats, es comentaran aquestes aspectes.

### d) Disseny físic

El disseny físic és el procés de generar la base de dades utilitzant un sistema gestor de base de dades corresponent.

En aquest pas es duen a terme totes les activitats necessàries per a crear les taules, definir els camps, les interrelacions entre taules, índexs, etc.

En el marc d'aquest material, només es tractarà, a tall d'introducció, el disseny conceptual i logicorelacional i s'introduiran conceptes sobre la normalització.

## 3.1. Models de dades: conceptes bàsics

L'objectiu bàsic d'un sistema de BD és obtenir mitjançant l'abstracció del món real, un conjunt estructurat de dades i un conjunt d'operacions definides sobre aquestes dades que permetin satisfer, de manera eficient, les necessitats d'informació d'una organització.

La representació d'una parcel·la de la realitat mitjançant un model de dades dóna lloc a un esquema. Però el món real també comprèn aspectes dinàmics, ja que les dades varien en el temps. Per això, els models de dades es componen de dos submodels que proporcionen elements (o conceptes bàsics) per a representar la naturalesa estàtica i dinàmica del sistema:

- **Naturalesa estàtica:** característiques inalterables que identifiquen el sistema i els seus objectes. Corresponen al que s'entén per estructura.

#### Creació d'índexs

Un índex permet ordenar i trobar registres amb més rapidesa. Es pot indexar un únic camp o una combinació de camps. La clau primària d'una taula s'indexa de manera automàtica. Determinats camps no es poden indexar a causa del tipus de dades que contenen.

- **Naturalesa dinàmica:** accions que admet el sistema fent-lo evolucionar i canviar d'estat al llarg del temps. Descriuen el comportament de l'estructura.

Un **model de dades** és el conjunt de conceptes, regles i convencions que permeten descriure una parcel·la del món real que intervé en un problema determinat (d'univers de discurs).

Els elements que proporcionen els models de dades, malgrat que estan definits amb terminologia i formalisme diferents, tenen significats equivalents. Això permet agrupar-los, genèricament, en:

#### 1) Elements permesos:

- Descriuen les **estructures de dades** (els *objectes*, les seves  *propietats*) i la manera en què es relacionen (*associacions*).
- Defineixen les **operacions** (i *transaccions*) que permeten la manipulació de les dades (addició, eliminació, modificació i recuperació).

#### 2) Elements no permesos:

Són les **restriccions** o limitacions imposades a l'estructura del model o a les dades, que invaliden determinades ocurrences en la BD. Se'n distingeixen dos tipus:

**a) Restriccions inherents:** són limitacions d'utilització a la mateixa naturalesa del model de dades, el qual no admet determinades estructures. Poden ser de dos tipus:

- **Pròpies del model de dades:** la seva definició correspon al dissenyador, però la seva gestió és responsabilitat del model de dades, el qual les reconeix i les recull en l'esquema.
- **Alienes al model de dades:** el model de dades no les reconeix ni proporciona instruments per a manipular-les; són responsabilitat del dissenyador.

#### Exemples de restriccions inherents

##### a) Pròpies del model de dades:

- El model relacional no permet atributs multivalors.
- Utilitzar una regla de validació (restricció CHECK) per a comprovar que l'edat dels votants és més gran de 18 anys.

##### b) Alienes al model de dades: les restriccions de cardinalitat mínima.

**b) Restriccions semàntiques** o d'integritat: són limitacions imposades als valors dels atributs o a les característiques de les interrelacions per a reflectir fidelment la realitat. Permeten captar la semàntica de l'univers de discurs que es vol modelar, i verificar la correcció de les dades emmagatzemades a la BD.

#### **Exemples de restriccions semàntiques**

- L'estat civil d'una persona no pot passar directament de solter/a a vidu/vídua.
- Una persona no pot tenir una professió si és menor de 18 anys.
- El salari d'un treballador no pot ser més alt que el del seu cap o supervisor.

Les restriccions garanteixen la integritat de la BD i la validesa semàntica del seu contingut.

Tot **model de dades** comporta un modelatge o procés d'abstracció que és la tasca intel·lectual mitjançant la qual es representa la realitat per tal d'obtenir una estructura per a emmagatzemar les dades.

Els **tipus d'abstracció** bàsics són els següents: classificació/instanciació, generalització/especialització, agregació/desagregació i associació/dissociació.

### **3.2. Definició conceptual d'una BD**

Una base de dades és cara i difícil de crear.

La inclusió d'una o més bases de dades a les nostres organitzacions implicarà la necessitat de temps de preparació del programari, la necessitat de comptar amb un expert perquè dissenyi la base de dades i un cost de manteniment.

Aquestes tasques es poden dur a terme internament o per mitjà de la contractació de tercers, externs a l'organització.

Per tal de generar una base de dades s'estableix un procés o metodologia que s'inicia amb la visió del món exterior, en concret, de la part que ens interessa representar en dades.

En aquest procés s'ha d'aprendre, comprendre i conceptualitzar aquest món exterior i transformar-lo en un conjunt d'idees i definicions que representin una imatge fidel del comportament del món real.

Aquest procés d'abstracció genera el que coneixem com el **model conceptual**.

La definició correcta del model conceptual de la base de dades és imprescindible per a garantir la generació correcta de la posterior base de dades.

A continuació s'enumeren els components i elements principals d'una metodologia per a definir un model conceptual de dades: **entitats**, **atributs**, **claus** i **relacions**.

### 3.2.1. Model de dades: el model entitat-relació

El model entitat-relació es basa en l'ús d'un conjunt de símbols gràfics que permeten representar "la realitat" de la informació que es vol gestionar.

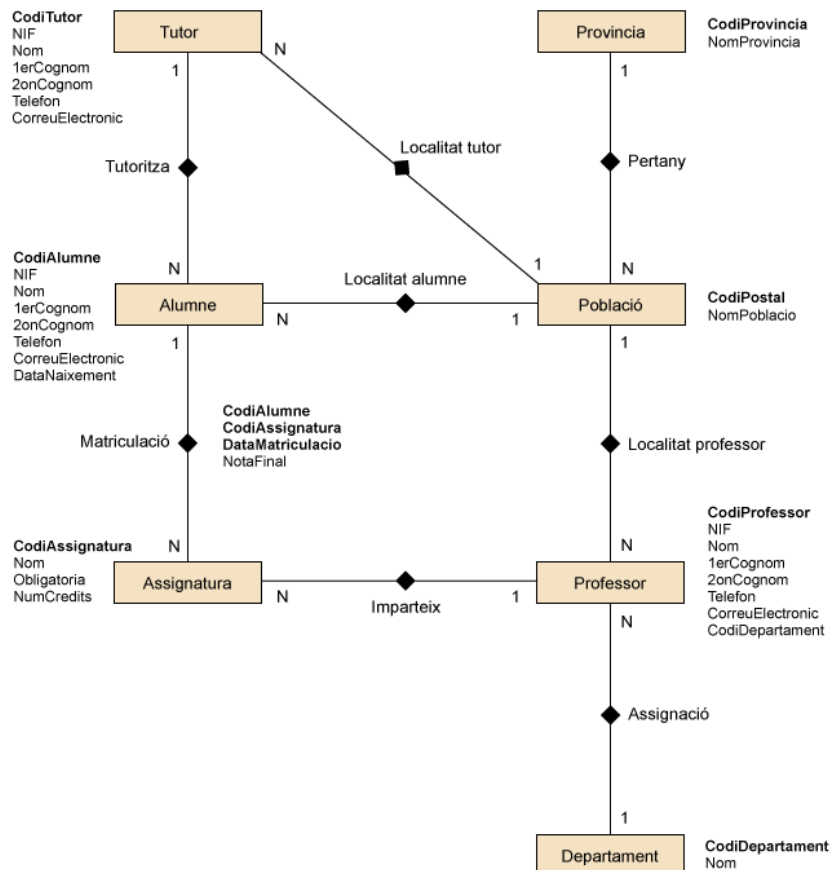
D'aquesta manera, s'identifiquen diferents elements d'informació com són les **entitats** i els seus **atributs**, les **relacions entre entitats** i la **cardinalitat i grau** d'aquestes relacions.

El correcte disseny conceptual de la base de dades, seguint les normes i regles d'aquest model, facilitaran, posteriorment, la definició i creació de la base de dades relacional.

En aquest apartat es defineixen els principals elements d'aquest model i en els casos pràctics del mòdul se n'exemplifica l'ús.

#### Exemple de modelat

El següent és un possible modelat de la BD relacional d'un centre d'estudis (amb departaments, professors, assignatures, alumnes...) exposada en els exemples de "Bases de dades relacionals" de l'apartat anterior.



#### Per a saber-ne més

Altres exemples de modelatge conceptual (generació del diagrama entitat-relació) s'inclouen al final en el subapartat "Casos pràctics de disseny i creació de bases de dades".

Per a simplificar el modelat de l'exemple, es fan els següents supòsits o restriccions semàntiques:

- Una assignatura només pot ser impartida per un professor.
- Els tutors no poden ser professors.

### Equivalència dels elements del model relacional i del model entitat-relació

La taula següent mostra l'equivalència entre els elements bàsics del model relacional i els del model entitat-relació.

	Model de BD		
	Relacional	Entitat-relació	Exemples
Elements	Taula	Entitat	ALUMNE, ASSIGNATURA, PROFESSOR...
	Registre	Ocurrència	Joan Soler Pou..., estadística...
	Camp	Atribut	Nom, cognoms, NIF, adreça...
	Clau	Clau	NIF
	Relació	Relació	Matriculació (relació alumne-assignatura)...

### 3.2.2. Entitats

Una entitat és un objecte real o abstracte amb característiques diferenciadores d'altres objectes i la informació del qual s'emmagatzemarà en la base de dades.

Pren com a significat conceptes o objectes que tenen un paper important a la companyia o organització.

Una entitat està formada per un conjunt d'elements de dades o atributs, cada un dels quals aporta alguna característica a la definició de l'entitat.

### 3.2.3. Ocurrència

Per ocurrència s'entén **cadascun** dels elements que representa una entitat. És a dir, cadascun dels alumnes o professors o cadascuna de les assignatures seria una ocurrència de les entitats "Alumnes", "Professors" i "Assignatures".

#### Per a saber-ne més

Els exemples presentats a continuació per a il·lustrar els elements del model entitat-relació tenen un paral·lelisme directe amb els exposats per a cada element del model relacional en l'apartat "Bases de dades relacionals".

#### Exemples d'entitats

Fent una analogia amb els exemples vistos anteriorment, exemples d'entitats serien Alumnes, Assignatures i Professors.

### 3.2.4. Atribut

Un atribut és una unitat bàsica i indivisible d'informació sobre una entitat que serveix per a identificar-la o descriure-la.

Normalment, s'utilitza la denominació **atribut** per a especificar el *nom d'atribut*, que s'ha de diferenciar del *valor d'atribut* o **valor**.

Els atributs (i els camps) poden ser de diferents tipus:

- atribut compost: adreça (es pot dividir en carrer, número, pis i porta) o data de naixement (dia, mes i any)
- atribut simple (o atòmic): carrer, any de naixement
- atribut derivat (o calculat): edat
- atribut emmagatzemat: data de naixement
- atribut monoavaluat: edat
- atribut multiavaluat: telèfon

Un valor nul pot tenir diverses interpretacions:

- Valor no aplicable: l'atribut no s'aplica en aquesta ocurrència d'entitat (o registre). Per exemple, si una persona no té DNI.
- Valor desconegut: falta (per exemple, si no se'n sap l'estatura); no se sap si existeix (per exemple, correu electrònic); o es coneix, però està absent, encara no s'ha registrat...

### 3.2.5. Clau

S'anomena *clau d'una entitat*, l'atribut o conjunt d'atributs que permet identificar, de manera única, un element d'una entitat.

### 3.2.6. Relacions

Les entitats per si soles no descriuen la realitat d'un sistema d'informació.

No n'hi ha prou d'identificar objectes; a més a més, s'han de definir les associacions que hi ha entre els objectes o entitats.

Aquestes associacions s'anomenen *relacions*.

#### Exemples de relació

- La relació "Pertany a" (o "Pertinença a") que hi ha entre les entitats PROFESSOR i DEPARTAMENT.

#### Exemple atributs

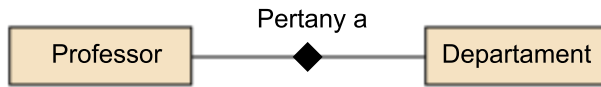
Per exemple, el nom, cognoms, adreça, NIF, etc. serien atributs de l'entitat Alumnes.

#### Exemples de valor

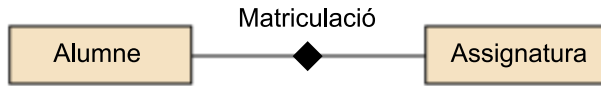
55 123 708-H, Joan, Soler Pou, 36 i Major, 51, 2n.-1a. són, respectivament, els valors dels atributs DNI, Nom, Cognoms, Edat i Adreça.

#### Exemple clau

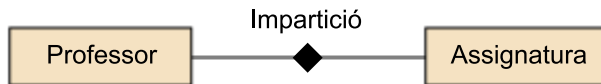
El NIF podria ser, per exemple, la clau de les entitats Alumnes i Professors.



- La relació "Matriculació" (o "Matriculat en") que s'estableix entre les entitats ALUMNE i ASSIGNATURA.



- La relació "Impartició" existent entre les entitats PROFESSOR i ASSIGNATURA.



Tota relació presenta dues característiques que la defineixen: el grau (nombre d'entitats que participen en la relació) i la cardinalitat de cadascuna en la relació. Es descriuen a continuació.

### Grau d'una relació

Per aquest concepte s'identifica el nombre d'entitats que s'interrelacionen.

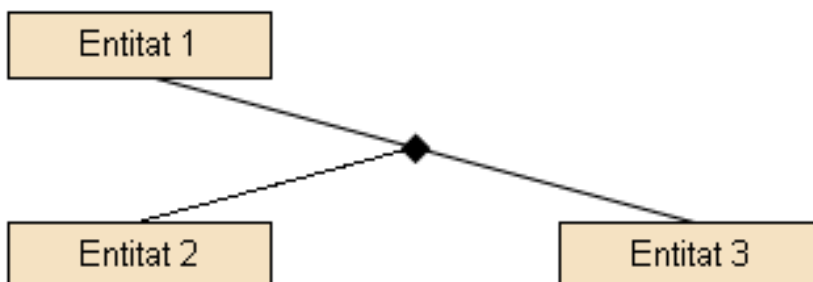
La majoria de vegades el grau d'una relació serà binària (entre dues entitats, com les vistes anteriorment) o ternària (entre tres entitats, tal com es veurà en un dels exemples posteriors del material).

#### Exemple relació binària i ternària

Relació binària



Relació ternària



Es podria donar el cas de relacions quaternàries o superiors, però són casos complexos i poc freqüents.



## Cardinalitat d'una relació

La cardinalitat o grau d'una relació representa la participació en la relació de cadascuna de les entitats afectades.

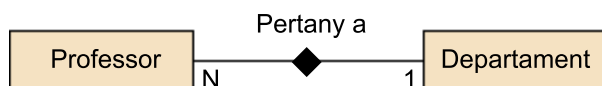
Això, gràficament, s'indica posant en la línia que uneix les entitats el grau de participació ( $1$ ,  $N$  o  $M$ ) segons correspongui a cada entitat.

En una relació binària, hi ha tres tipus possibles:

- **Una a moltes ( $1:n$ ).** A cada ocurrència de la primera entitat li corresponen una o diverses ocurrències de la segona.

### Exemple de relació $1:n$

La relació "Pertany a" té una cardinalitat "1 a n", ja que cada professor pertany a 1 únic departament i a un departament pertanyen diversos professors (N).

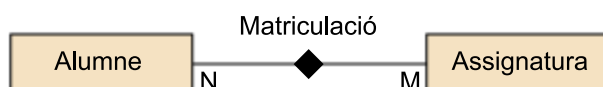


Això s'indica gràficament posant en la línia que uneix les entitats una N al costat de la primera entitat i un 1 al costat de la segona. En aquest cas, la primera entitat és DEPARTAMENT (1) i la segona, PROFESSOR (N).

- **Moltes a moltes ( $m:n$ ).** A cada ocurrència de la primera entitat li poden correspondre diverses ocurrències de la segona i viceversa.

### Exemple de relació $m:n$

La relació entre les entitats ALUMNE i ASSIGNATURA, que es pot anomenar "Matriculació", té una cardinalitat "n a m", ja que cada alumne es pot matricular en més d'una assignatura (M), i en cada assignatura es matricula més d'un alumne (N).



- **Una a una ( $1:1$ ).** A cada instància o ocurrència d'una entitat li correspon una i només una ocurrència de l'altra. Aquest cas es dona poques vegades.

### Exemple de relació $1:1$



Aquest cas es presenta en dividir en dues una entitat per a separar/agrupar les seves dades en dos tipus d'entitat diferents.

## 3.3. Del model conceptual al lògic o relacional

Una vegada dissenyat el model conceptual d'una base de dades, es pot generar el model relacional a partir d'aquest model.

A continuació, s'inclou un resum de les regles de conversió principals:

- Tota entitat del model conceptual es converteix en una taula en el model relacional. Per exemple, alumnes, professors, assignatures, etc.
- Tot atribut d'una entitat es transforma en un camp de la taula. Per exemple, de l'entitat alumnes es generen els camps codi, nom, 1r. cognom, 2n. cognom, telèfon, etc.
- Tota relació 1:n es converteix en una clau forana en la taula corresponent a l'entitat de grau n. Per exemple, la relació adscripció entre professors i departaments.
- Tota relació m:n es transforma en una taula intermèdia. Per exemple, la taula corresponent a la relació matriculació.

#### **Per a saber-ne més**

Podeu veure la conversió de la relació 1:n ("adscripció", entre PROFESSOR i DEPARTAMENT) en clau forana i de la relació m:n ("matriculació", entre ALUMNE i ASSIGNATURA) en taula intermèdia en els exemples exposats en l'ítem "Relacions" del subapartat "Bases de dades relacionals".

### **Etaques de disseny conceptual i lògic de la base de dades**

En aquesta assignatura només s'abordaran, en l'aspecte pràctic, els passos següents del disseny:

**a) Disseny del model conceptual** (model entitat-relació). Identificació de les entitats, els seus atributs i les interrelacions entre aquestes entitats, amb els atributs d'aquestes, si fos el cas.

**b) Generació del model relacional** (model lògic). Partint del model conceptual dissenyat en l'apartat anterior, s'ha de convertir en una BD relacional, indicant les taules corresponents, els camps de cadascuna, el tipus de dada de cada camp, la clau primària de cada taula i les claus foranes corresponents.

### **Pautes per a anomenar entitats, atributs, taules i camps**

Els noms d'entitats i atributs (i, per extensió, de taules i camps) poden incloure qualsevol combinació de lletres, nombres, espais en blanc i caràcters especials excepte el punt (.), el signe d'admiració (!), l'accent greu (`) i els claudàtors ([ ]), entre altres. A més, a Microsoft Access, les taules i camps tampoc no poden començar per un espai en blanc ni incloure cometes dobles (") i, en total, poden tenir fins a 64 caràcters.

**a) Consells:**

- Malgrat que es poden usar espais, és millor evitar-los, ja que hi pot haver conflictes si es fa referència a noms amb espais en expressions o un codi de Visual Basic per a aplicacions.
- Evitar l'ús de noms extremament llargs perquè és difícil recordar-los i referir-s'hi.
- Evitar que el nom coincideixi amb el nom d'una propietat o element que utilitzi el SGBD (Microsoft Access), ja que podria produir un comportament inesperat de la BD.

#### b) Convenis:

- Entitats (i taules) en singular i majúscula.
- Atributs (i camps) en singular, inicials de paraula en majúscula i resta en minúscula.
- Si tenen més d'una paraula, s'ajunten sense deixar espais. Una altra opció, menys recomanable, és usar el guió baix ( \_ ) per a separar-les.

#### Per a saber-ne més

Al final del mòdul hi ha dos casos resolts de disseny de BD en què s'exemplifiquen les etapes de disseny conceptual i lògic.

### 3.4. Normalització de bases de dades

La normalització de bases de dades és el procés mitjançant el qual es transformen dades complexes a un conjunt d'estructures de dades més petites, que a més de ser més simples i més estables, són més fàcils de mantenir.

També es pot entendre la normalització com una sèrie de regles (formes normals) que serveixen per a ajudar els dissenyadors de bases de dades a desenvolupar un model relacional que minimitzi els problemes de lògica.

Cada regla o forma normal està basada en l'anterior.

La normalització es va adoptar ja que l'antiga manera de posar totes les dades en un lloc, tant en un arxiu com en una taula de la base de dades, era ineficient i conduïa a errors quan es volien manipular les dades.

La normalització també fa les coses més fàcils d'entendre entre els desenvolupadors de la base de dades i els usuaris que la requereixen i en plantegen els requisits.

Un altre avantatge de la normalització d'una base de dades és que ajuda a optimitzar espai. Una base de dades normalitzada ocupa menys espai que una no normalitzada, ja que evitem repeticions de valors innecessaris.

### 3.4.1. Graus de normalització

El procés de normalització de les bases de dades té un conjunt de regles per a cada fase.

Bàsicament, hi ha tres nivells de normalització:

- Primera forma normal (1FN).
- Segona forma normal (2FN).
- Tercera forma normal (3FN).

Cadascuna d'aquestes formes normals té les seves regles.

Quan una base es conforma a un nivell, es considera normalitzada segons aquella forma normal.

Algunes vegades, tenir una base de dades normalitzada al més alt nivell la pot fer més complexa que si no ho està.

#### Normalització d'una base de dades

Algunes vegades, la normalització completa d'una base de dades podria esdevenir en un gran nombre de taules. Es podria donar el cas, aleshores, que per a obtenir informacions relacionades, calgués unir moltes d'aquestes taules, per mitjà d'una consulta, alentint el procés. En aquests casos seria més eficient disposar de taules no normalitzades però amb la informació ja relacionada.

Com a introducció, en la taula següent s'introdueix cadascuna d'aquestes tres formes normals.

Regla (forma normal)	Descripció
Primera forma normal (1FN)	Consisteix en l'eliminació de tots els grups repetits
Segona forma normal (2FN)	Assegura que totes les columnes (camps) que no són clau, depenguin completament de la clau primària
Tercera forma normal (3FN)	Elimina les dependències transitives. Una dependència transitiva és aquella en què les columnes (camps) que no són clau depenen d'altres columnes (camps) que tampoc no són clau.

### 3.4.2. Primera forma normal (1FN)

La regla de la primera forma normal estableix que les columnes repetides s'han d'eliminar i col·locar en taules separades.

D'aquesta manera es resolen els problemes de capçaleres de columna múltiples. En lloc de tenir una única taula immensa que té molts aspectes, es disposa de diverses taules més petites i gestionables, cadascuna amb les dades d'un aspecte concret.

En aquest cas, la normalització ajuda a aclarir la base de dades i organitzar-la en parts més petites i més fàcils d'entendre.

### 3.4.3. Segona forma normal (2FN)

La regla de la segona forma normal estableix que totes les dependències parcials s'han d'eliminar i separar de les seves pròpies taules. Una dependència parcial és un terme que descriu les dades que no depenen de la clau primària de la taula per a identificar-les.

### 3.4.4. Tercera forma normal (3FN)

Una taula està en la tercera forma normal si totes les columnes que no són clau primària són totalment i funcionalment dependents de la clau primària i no hi ha dependències transitives. Una dependència transitiva és aquella en què hi ha columnes que no són clau primària i que depenen d'altres columnes que tampoc no són clau primària.

Quan les taules estan en la tercera forma normal s'eviten errors quan s'insereixen o esborren registres.

Cada camp en una taula està identificada de manera única per la clau primària i no hi ha dades repetides.

### 3.4.5. Exemples

L'objectiu d'aquest mòdul és, simplement, introduir el concepte de normalització, però no s'espera que els estudiants dominin les tècniques de disseny de bases de dades i la seva normalització, per la qual cosa, a continuació, s'enumeren dos casos com a exemple, per a fer més fàcil la comprensió dels conceptes.

#### Exemple de normalització en 1FN

Les dades dels socis d'un club de lleure i de les activitats que han dut a terme es podrien recollir en una taula amb l'estructura següent:

Base de dades sense normalitzar (una taula)

NIF	Núm. soci	Nom	Cognoms	...	Activitat	Data activitat	...
1					Globus	01-06-10	
1					Sortida a Cardona	05-07-10	
2					Globus	01-06-10	
2					Sortida a Cardona	05-07-10	
3					Sortida a Cardona	05-07-10	

NIF	Núm. soci	Nom	Cognoms	...	Activitat	Data activitat	...
3					Bàsquet	30-06-10	
4					Bàsquet	30-06-10	
5					Bàsquet	30-06-10	
...							

A la taula, es pot comprovar que els socis amb NIF 1 i NIF 2 van participar en les activitats de globus i sortida a Cardona, i que el soci amb NIF 3 va participar en les activitats de sortida a Cardona i bàsquet.

La taula no està normalitzada, atès que es presenten grups de dades repetits. D'una banda, a cada registre d'activitat d'un soci es repeteixen totes les seves dades personals (núm. soci, nom, cognoms...) i, de l'altra, per a tots els socis que han participat en una mateixa activitat es repeteixen les dades de l'activitat (nom i data de realització).

Base de dades normalitzada (1FN) (dues taules)

Soci				
NIF	Núm. soci	Nom	Cognoms	...
1				
2				
3				
4				
5				
...				

Activitat		
Activitat	Data activitat	...
Globus	01-06-10	
Sortida a Cardona	05-07-10	
Bàsquet	30-06-10	

Per a obtenir la BD normalitzada en 1FN s'han col·locat els grups de columnes que repeteixen dades en taules separades. D'aquesta manera, ja no es repeteixen ni les dades dels socis ni les dades de les activitats.

El pas següent en el procés de normalització, atès que la relació entre les taules resultants és molts a molts (un soci pot participar en més d'una activitat i cada activitat pot tenir diversos socis participants), seria crear una taula intermèdia.

### Exemple de normalització en 2FN

Una situació molt habitual en BD que contenen informació de persones i/o d'organitzacions és la relació entre els codis postals i la població.

Continuant amb l'exemple anterior, en definir les dades de la taula SOCI també s'inclou el codi postal i la població, com mostra la taula.

Base de dades sense normalitzar (una taula)

NIF	Nom	Cognoms	....	Codi postal	Població	...
1				08700	Igualada	
2				08700	Igualada	
3				08202	Sabadell	
4				08206	Sabadell	
5				08225	Terrassa	
6				08760	Martorell	
7				08700	Igualda (*)	
8				08202	Sabadell	
9				08760	Martorel (*)	
...						

Es pot comprovar que quasi tots els camps depenen de la clau primària (que pot ser el NIF). En canvi, el camp Població depèn del Codi postal. Això significa que la BD no està normalitzada segons la segona forma normal.

Per a normalitzar-la, cal crear una segona taula (POBLACIÓ) on hi hagi un registre per a cada codi postal i població, i a la taula SOCI deixar el camp Codi postal (serà clau forana), que es vincularà amb la taula POBLACIÓ.

A la taula, es pot apreciar, d'una banda, que es repeteixen valors (del codi postal i la població) en els registres corresponents al NIF 1 i NIF 2, o al NIF 3 i NIF 8.

A més, en una situació real, com que la BD no està normalitzada segons la 2FN, hi pot haver dades equivocades. En l'exemple, això s'evidencia en els registres dels NIF 7 i NIF 9, en què el nom de la població és incorrecte ("Igualda" i "Martorel", respectivament, que s'han marcat amb asterisc), tot i que el codi postal està bé. En aquest cas, si es fa una consulta, pel camp Població, dels socis que viuen a Igualada no es trobarà el registre del NIF 7. El mateix passarà amb Martorell i el registre del NIF 9.

Base de dades normalitzada (2FN) (dues taules)

Soci					
NIF	Nom	Cognoms	....	Codi postal	....
1				08700	
2				08700	
3				08202	
4				08206	
5				08225	
6				08760	
7				08700	

Soci					
8				08202	
9				08760	
...					

Població	
Codi postal	Població
08700	Igualada
08202	Sabadell
08206	Sabadell
08225	Terrassa
08760	Martorell
...	

Un cop normalitzada la BD, queda resolt el problema de les dades repetides a la taula SOCI i també possibles errors com els detectats en la situació anterior.

### 3.4.6. Altres formes normals

Després de la tercera forma normal, n'hi ha algunes més (forma normal Boyce-Codd, quarta forma normal, cinquena forma normal o forma normal de projecció-unió, etc.), però si es garanteix tenir les bases de dades en tercera forma normal, es resolen la majoria de problemes i dificultats.

### 3.5. Casos pràctics "Disseny i creació de bases de dades"

A continuació, es plantegen i es resolen dos casos pràctics que ajuden a exemplificar els conceptes exposats en aquest apartat de disseny de BD. Un correspon a un centre de formació i l'altre, a una empresa de lloguer de cotxes. En el segon hi ha un exemple de relació ternària.



El disseny d'una base de dades no és únic, ja que depèn dels supòsits i/o restriccions que calgui tenir en compte.

En una situació real, és habitual que el dissenyador de la BD sol·liciti diferents aclariments al client (responsable i/o usuaris de la BD) per anar solucionant possibles dubtes que es presentin durant el procés de disseny.

Per tant, els exercicis proposats no tenen una única solució, ja que segons el disseny del model conceptual que es faci, la BD relacional obtinguda serà una o altra.

### 3.5.1. Cas pràctic 1: Base de dades "Centre de formació"

FormProf SA, Formació Professional en Informàtica, és un centre de formació en l'àmbit de les tecnologies d'informació, de creació recent.

Els seus serveis es basen en la planificació i l'execució de programes de formació en l'àrea d'informàtica, tant a nivell d'usuari (introducció a Internet, ús de MSFT Outlook, treball amb processadors de textos, creació d'animacions amb Flash, etc.) i també com a "programadors especialistes" (desenvolupament d'aplicacions amb Visual Studio .NET, instal·lació i configuració de xarxes amb Unix, etc.).

L'equip humà del centre i la seva organització és la següent:

- Un/a director/a, responsable màxim del centre i de les seves operacions.
- Departament Comercial i de Màrqueting. Dependent del director del centre, hi haurà un equip de força de vendes, compost per dos comercials assessors que cobriran tot el territori, més una persona responsable de les campanyes de màrqueting i promoció.
- Un/a responsable de l'àrea de RH i Finances.
- Un/a auxiliar administratiu/va que donarà suport a les tasques de RH i Finances i Comptabilitat.
- Un/a responsable tant de la recepció i atenció al client com de donar suport al director.
- Finalment, hi ha un responsable de tecnologies, que sou vosaltres.

Com a equip docent, tenim l'organització següent:

- Un cap d'estudis que també exerceix com a professor, puntualment.
- Dos professors, pertanyents a la plantilla del centre, responsables tant de la direcció de programes de formació com d'impartir part dels seus mòduls.
- Un conjunt de professors, aliens al centre, contractats puntualment per a impartir alguns dels mòduls i cursos oferts per FormProf.

A continuació, s'especifiquen els requisits d'una base de dades per a emmagatzemar i gestionar tota la informació de les accions formatives que es duen a terme en aquest centre, i també el seguiment dels clients, alumnes, sessions de formació, etc.

- D'una banda, caldrà emmagatzemar la informació de tots els **cursos** que es planifiquen i executen en la nostra companyia. Per a cada curs, caldrà guardar informació del seu identificador (serà un codi de 5 lletres), el títol del curs, la data d'inici i la data de finalització i un valor (del tipus Sí/No), que indicarà si el curs és de calendari (valor Sí) o a mida (valor No) per a una companyia.
- Així, cal guardar la informació de les **companyies** client, indicant-hi el seu CIF, raó social, adreça, codi postal, població, telèfon, correu electrònic i nom de la persona de contacte.
- Un curs el pot organitzar una companyia (en el cas que sigui a mida) o de calendari. En aquest darrer cas, no té cap companyia associada. Una companyia ens pot contractar més d'un curs.
- Per a cada curs, també s'emmagatzemarà la informació dels **assistents** o matriculats. Caldrà disposar de la informació següent: NIF, nom, cognoms, adreça, telèfon i correu electrònic. Cal tenir en compte que un alumne pot participar en més d'un curs.
- D'altra banda, també hem de poder guardar la informació de les **sessions** d'un curs. Tot curs tindrà un mínim d'una sessió (en cas que sigui d'un sol dia). Per a cada sessió hem de guardar la informació següent: codi de la sessió (identificador numèric), hora d'inici, hora de fi i data de realització.
- Per a cada sessió caldrà guardar la informació del **professor** responsable. De cada professor guardarem el NIF, el nom i els cognoms. Només hi haurà un professor per sessió, però, per descomptat, un professor participarà en diverses sessions de formació.
- I, finalment, cal considerar que hem de guardar l'**aula** on es durà a terme la sessió (Aula 1, Aula 2 o "In Company").

El disseny correcte de la BD ha de permetre fer consultes per a obtenir, a partir de la informació emmagatzemada, els resultats següents:

- Determinar quins cursos són de calendari.
- Especificar l'empresa per a la qual s'ha organitzat cadascun dels cursos fets a mida.
- Obtenir la llista de tots els alumnes assistents a un curs.
- Saber quins professors impartiran alguna sessió d'un curs.
- Saber quines sessions ha d'impartir un professor entre unes dates concretes.
- Identificar les sessions que s'han de dur a terme o s'han de fer en una data concreta.

S'han de fer les activitats següents:

1) **Disseny del model conceptual de la BD.** Identificació de les entitats, els seus atributs i les interrelacions entre aquestes entitats, amb els atributs d'aquestes (si fos el cas) en un esquema entitat-relació.

2) **Generació del model relacional de la BD.** Partint de l'esquema conceptual dissenyat en l'apartat anterior, convertir-lo en una BD relacional, indicant les taules corresponents, els camps de cadascuna, el tipus de dada de cada camp, la clau primària de cada taula i les claus foranes corresponents.

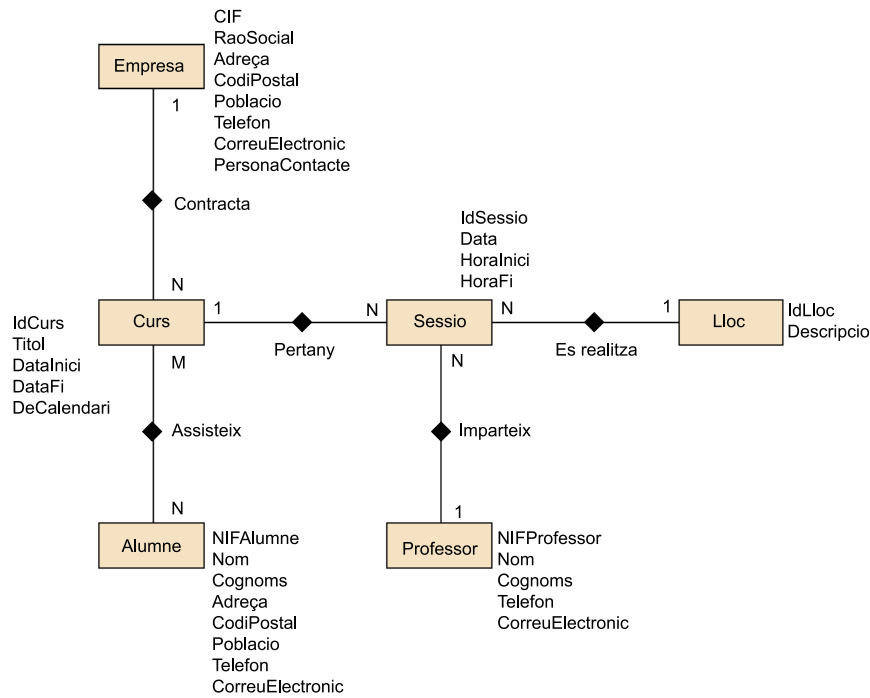
### **Cas pràctic 1: Base de dades "Centre de formació". Resolució**

Aquest exercici no té una solució única, ja que en funció del disseny del model conceptual, la base de dades relacional final serà una o altra.

En una situació real, durant el disseny de la base de dades, el responsable d'aquesta base de dades sol·licita diferents requeriments al "client" per anar determinant ambigüitats i dubtes possibles.

#### 1) Model conceptual de la base de dades

El model conceptual es representa gràficament mitjançant el diagrama següent entitat-relació (DER), en què s'identifiquen les entitats, els seus atributs i les relacions entre entitats:



Notes:

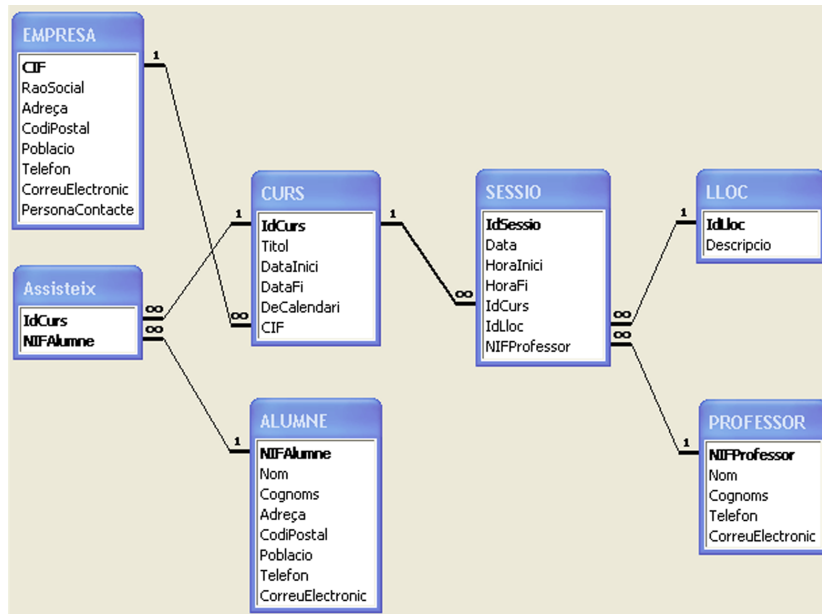
- Cal destacar la relació  $m:n$  entre CURS i ALUMNE, ja que normalment un curs té diversos alumnes i un alumne pot estar matriculat en diversos cursos.
- En un model real, seria convenient considerar la població com una entitat identificada pel codi postal. D'aquesta manera, no caldria guardar el nom de la població a les entitats EMPRESA i ALUMNE, sinó que l'existència de dues interrelacions (EmpresaPoblació i AlumnePoblació) permetria guardar-hi només el codi postal. Aquest aspecte, però, no s'ha considerat en el cas enunciat per tal de simplificar-ne el disseny.

## 2) Generació model de base de dades relacional

Per a generar aquest model, s'han aplicat les regles de conversió següents:

- La relació original  $m:n$  entre les entitats CURS i ALUMNE es transforma en tres taules. Una per a CURS, una altra per a ALUMNE i una d'intermèdia que permet emmagatzemar la informació de la relació Assisteix (o AlumneCurs) que hi ha entre aquestes.
- La resta d'entitats es transformen en taules, els atributs de cada entitat en camps de cada taula i les relacions  $1:n$  s'estableixen entre la clau primària d'una taula (extrem 1) i la clau forana d'una altra (extrem  $n$ ).

La imatge següent mostra les taules, els seus camps i les relacions entre les taules de l'esquema relacional de la BD creat amb MS Access. Els camps que formen clau primària figuren en negreta.



### 3.5.2. Cas pràctic 2: Agència de lloguer de cotxes

Es vol dissenyar una base de dades per a emmagatzemar i gestionar la informació utilitzada en una empresa dedicada al lloguer de vehicles, tenint en compte els requisits següents:

- L'empresa disposa d'un conjunt de vehicles per llogar. Es necessita saber la matrícula, marca, model, color, tipus de combustible i tarifa diària de lloguer de cada cotxe.
- Cada vehicle està assignat a un determinat garatge, que no pot canviar, del qual interessa saber el codi, la denominació, les dades d'ubicació (adreça, població, codi postal) i el telèfon.
- Les dades que s'han d'emmagatzemar de cada client són el NIF, el nom, les dades d'ubicació i les de contacte (telèfon i correu electrònic).
- Quan un client vol llogar un cotxe, es posa en contacte amb alguna de les agències o delegacions de l'empresa de lloguer per sol·licitar una reserva.
- De cada agència es guardarà el codi, la denominació, les dades d'ubicació i les de contacte.
- Un mateix client pot fer o haver fet unes quantes reserves al llarg del temps, que es poden encavalcar en el temps.
- Un client pot fer reserves a diferents agències. En cada reserva hi participa una única agència.
- Una reserva pot incloure més d'un vehicle. De cada reserva es registrarà la data d'inici (data de lliurament del vehicle o vehicles al client), la data de finalització (data prevista per retornar el vehicle), el preu total de la reserva i un indicador de si el vehicle s'ha retornat.
- Per a cada cotxe, interessa recollir el nivell de combustible que conté el dipòsit en el moment de lliurar-lo al client.
- El preu final de la reserva de cada cotxe s'obté multiplicant la tarifa diària de lloguer pel nombre de dies que el client el vol reservar.

- El preu total d'una reserva s'obté sumant els preus finals de lloguer de tots els cotxes inclosos en la reserva.

El correcte disseny de la base de dades haurà de permetre que l'empresa pugui realitzar consultes del tipus següent:

- Llista de totes les agències, cotxes i clients.
- Historial de totes les reserves realitzades indicant el client, les dates i l'agència que l'ha tramitat.
- Nombre de reserves realitzades pels clients d'una població concreta.
- Resum de les reserves (número, data inici, data final i preu total) pendents de devolució, ordenades per data de finalització.
- Llista dels cotxes guardats a cada garatge.
- Agències (codi i nom) que participen en reserves encara no retornades.
- Llista de totes les reserves que ha tingut un cotxe determinat.
- Client amb més de tres reserves, alguna de les quals l'ha realitzat l'agència amb codi "4".
- Per a cada client de la companyia, llista de totes les reserves realitzades, i també el preu mitjà de totes.
- Les agències que no participen en cap reserva amb data d'inici posterior a una concreta.
- Tots els cotxes (matrícula, marca i model) reservats més de tres vegades.
- La marca i model del cotxe més reservat.
- Cotxes que no s'han reservat mai.
- Agències que participen en més de tres reserves d'una setmana o més de durada.
- Preu de la reserva més cara realitzada per a cada marca i model de cotxe.

### **Cas pràctic 2: Agència de lloguer de cotxes. Resolució**

En aquest apartat es presenta una solució possible per a aquest enunciat, donant per fet que n'hi pot haver d'altres.

#### **1) Disseny del model conceptual**

#### **Per a saber-ne més**

Per a tenir més informació sobre la normalització de les bases de dades, podeu consultar l'apartat "Normalització de base de dades" d'aquest mateix mòdul.

A continuació s'indiquen les entitats i atributs identificats en el model conceptual (s'ha subratllat l'atribut que identifica de manera única les instàncies de les entitats):

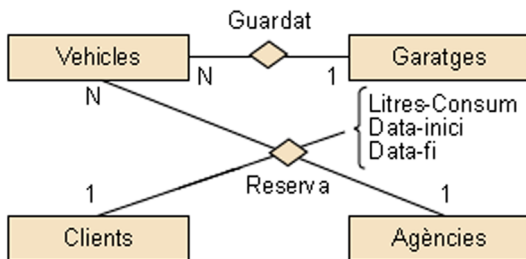
- vehicles: matrícula, marca, model, color, tipus de combustible i tarifa diària
- garatges: codi, denominació, adreça, codi postal, població, telèfon
- clients: NIE, nom, cognoms, adreça, codi postal, població, telèfon, correu electrònic
- agències: codi, nom, adreça, codi postal, població, telèfon, correu electrònic

D'altra banda, s'han identificat les relacions següents entre les entitats:

- Guardat: Vehicles-Garatges
- Reserva: Vehicles-Client-Agència

En el cas de la relació reserva, també caldrà guardar una sèrie d'atributs: litres-consum, data-inici, data-fi.

De manera gràfica, aquest model es representaria de la manera següent:

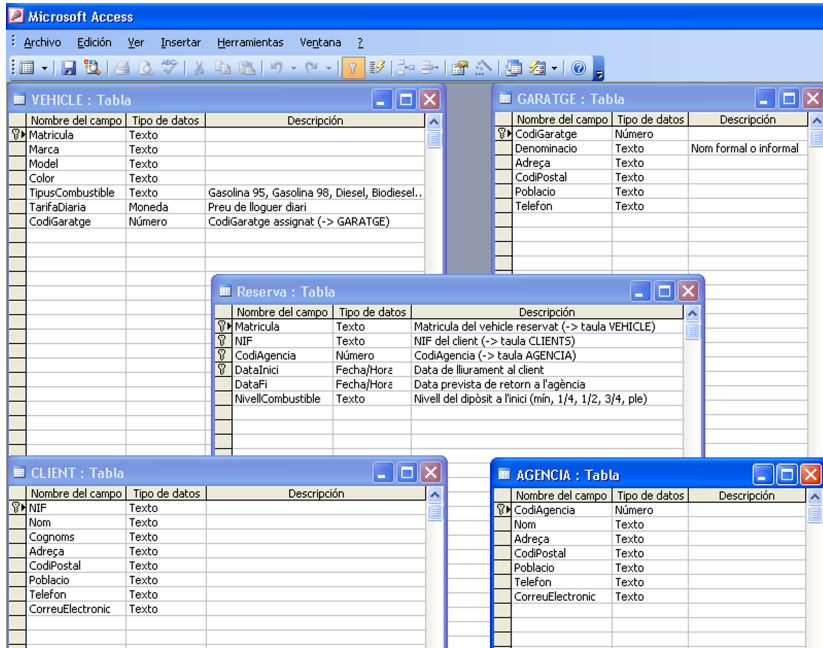


### Model relacional

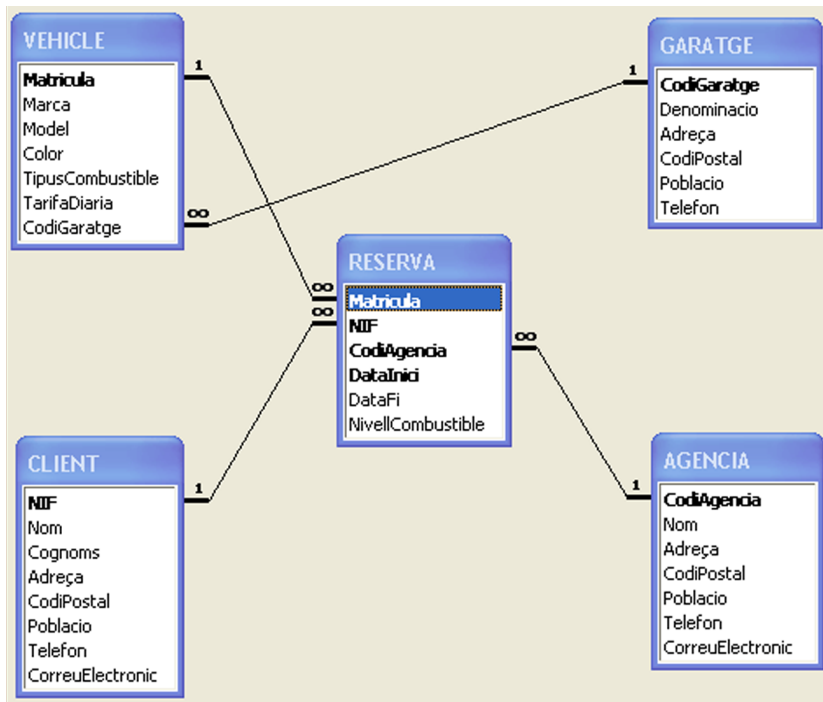
En crear el model relacional, caldrà tenir en compte les consideracions següents:

- Cada entitat serà una taula.
- L'atribut de cada entitat és un camp d'aquella taula.
- L'atribut que identifica de manera única les instàncies de l'entitat és la clau primària de la taula.
- La relació binària Guardat, entre vehicles i garatges (1:n) s'implementa per mitjà d'una clau forana a la taula vehicles que "apunta" al codi de garatge on es troba aquell cotxe.
- La relació "ternària" reserves, entre vehicles, client i agència, es converteix en una taula intermèdia que contindrà les claus primàries d'aquestes tres taules, a més dels litres consumits pel vehicle concret, la data d'inici de la reserva i la data de fi.
- La clau primària d'aquesta taula serà la combinació de les tres claus (vehicle, client i agència) juntament amb la data d'inici de la reserva.

A continuació es presenten les diferents taules (creades utilitzant l'MS Access):



I en la imatge següent, s'identifiquen les relacions entre les diferents taules de la base de dades en MS Access:



Notes:

- També s'hauria pogut fer el disseny de la base de dades considerant "Reserves" com una entitat. S'ha preferit fer-ho de l'altra manera, per a introduir un cas amb una relació ternària.
- Si es volgués normalitzar la base de dades caldria crear una taula de Poblacions, on hi hauria emmagatzemats el codi postal i la població a què correspon. En les taules Clients, Agències i Garatges no hi hauria el camp corresponent a la població, sinó el camp corresponent al Codi postal que "apuntaria" (seria clau forana) a la taula de Poblacions.



### 3.6. Resum

Un **model de dades** és el conjunt de conceptes, regles i convencions que permeten descriure una parcel·la del món real que intervé en un problema determinat (d'univers de discurs).

El disseny i la creació d'una base de dades es caracteritza pel següent:

- El disseny d'una BD consisteix a definir el model lògic; habitualment el **model relacional**. És a dir, construir un esquema de la informació (principalment, crear les taules, els seus camps i les relacions entre taules), independentment del SGBD que s'utilitzarà.
- Abans del disseny d'aquest model relacional, s'aconsella fer el **model conceptual**, que permet plasmar en un model entitat-relació les diferents informacions que es volen emmagatzemar (entitats, atributs i interrelacions) amb independència de consideracions físiques
- Abans de crear l'esquema físic, és convenient aplicar una tècnica de **normalització** de les taules per a verificar que no tenen dades redundants i estan "optimitzades".
- Finalment, en el **disseny físic** s'implementen tots els elements de la BD (taules, camps, interrelacions, índexs...) mitjançant un SGBD concret.

Esquema de la conversió d'elements del model conceptual al model relacional

Model conceptual		Model relacional
Entitat	<b>es converteix en</b>	Taula
Atribut		Camp
Relació 1:n		Clau forana en la taula corresponent a l'entitat de grau <i>n</i>
Relació m:n		Taula intermèdia

## 4. Bases de dades documentals

Tal com ja hem vist en els apartats anteriors, una base de dades es compon de diferents registres amb el seu corresponent número d'identificació.

Les bases de dades documentals (**BDD**) es caracteritzen perquè cada registre es correspon amb un document, de qualsevol tipus, publicació, document gràfic o sonor, etc. o a la seva referència.

La informació continguda en una base de dades documental s'estructura en diferents camps per tal de facilitar-ne el control i l'accés individualitzat.

Alguns camps es referiran a la descripció formal del document i, en d'altres, se'n tractarà el contingut temàtic. Fins i tot, en alguns casos, podran registrar el mateix document.

Per a facilitar la rapidesa en la recuperació de la informació, les BDD faciliten l'elaboració de diccionaris o índexs alfabètics. Alguns d'aquests sistemes treballen amb un índex únic format per paraules procedents de diferents camps de cada registre, mentre que en altres sistemes, es gestionen índexs per a cada camp.

Les BDD es poden classificar en diferents categories segons la informació que contenen i la referència al document corresponent: BD de text complet (que contenen els mateixos documents), arxius electrònics d'imatges (que contenen imatges dels documents originals) i BD referencials (que contenen referències per a localitzar els documents originals).

També es poden classificar per altres tipologies: segons el **mode d'accés a la informació**, la **cobertura documental** o segons el **model de tractament documental**.

Per tant, per a utilitzar eficaçment una BDD es recomana adaptar-se tant com es pugui a les seves característiques particulars.

Cal estar informat del contingut i de com s'han de fer les cerques, si cobreix una o diverses bases de dades, quina és la cobertura temàtica, si és un catàleg o disposa d'índexs i resums...

#### 4.1. Cerca d'informació

Tots els sistemes de recuperació d'informació permeten realitzar diferents modalitats de cerca:

- **Directa.** Es teclegen directament una o diverses paraules, podent-se interrograr en text lliure o sobre camps individuals.
- **Per mitjà d'índexs.** En aquests casos, l'usuari visualitza un diccionari o índex alfabètic de les entrades de tots els camps i selecciona els més adequats. Es pot cercar per índexs de paraules o per frase.
- **Jerarquitzada.** La consulta es realitza mitjançant una estructura jeràrquica. A partir d'un concepte jeràrquic, es poden localitzar no solament els registres en què apareix aquell terme, sinó tots aquells en què figuri algun concepte més específic del seu camp semàntic.
- **A través de codi.** Numèrics o alfanumèrics que codifiquen la classificació, l'idioma o la tipologia documental, entre d'altres.

Així doncs, per a fer una cerca cal utilitzar un nombre elevat de conceptes, per la qual cosa hi ha diverses **eines que** construeixen una estratègia i relacionen, de manera clara, els diferents termes utilitzats en la cerca:

- **Operadors lògics o booleans.** Permeten la combinació de conceptes en una mateixa cerca, tant si és la unió (OR) com la resta (AND NOT), la intersecció (AND) o l'operació contrària (XOR).
- **Operadors sintàctics de proximitat.** Presència en la mateixa frase, el mateix paràgraf, aparició en un ordre determinat o amb una separació mínima... Per exemple, NEAR.
- **Refinament.** Fins i tot, en alguns casos, el sistema permet de refinar les cerques, aplicant una cerca nova al conjunt de registres trobats en la cerca anterior.

#### 4.2. Categories de BD segons la informació que contenen

Els registres d'aquestes bases de dades poden incloure o no el contingut complet dels documents que descriuen, segons el qual es descriuen tres categories:

- **Bases de dades de text complet.** Constituïdes pels mateixos documents en format electrònic (digital). Poden incorporar, a més, camps amb informació complementària per a facilitar-ne la descripció i l'accés. Permeten localitzar termes presents en el text del document.

- **Arxius electrònics d'imatges.** Constituïdes per referències que permeten un enllaç directe amb la imatge del document original. En aquests casos, la cerca està limitada als camps de la referència bibliogràfica, però no es poden localitzar termes presents en el text del document original.
- **Bases de dades referencials.** Són aquelles en què els registres no contenen el text original sinó tan sols la informació fonamental per tal de descriure i permetre la localització de documents impresos, sonors, iconogràfics, etc. En aquests casos només es poden obtenir referències sobre documents, que després caldrà localitzar en un altre servei (arxiu, biblioteca, etc.). També poden incloure camps que facilitin la localització del document i/o enllaços directes per a obtenir l'original directament per mitjà d'un altre programa.

### 4.3. Tipologia de BD segons el mode d'accés

En funció del tipus d'accés permès a la base de dades, també es poden classificar en tres categories: d'accés local, en dispositiu físic (CD-ROM, DVD, etc.) o en línia.

- **Bases de dades d'accés local.** Per tal de consultar-les, cal anar al centre productor, tant a la seva biblioteca com al centre de documentació on es troben emmagatzemades.
- **Bases de dades en dispositius d'emmagatzematge.** S'adquireixen, habitualment, per compra o subscripció i estan emmagatzemades, físicament, en dispositius com els CD-ROM o DVD.
- **Bases de dades "en línia".** Es poden consultar des de qualsevol ordinador, ja que són accessibles "per Internet".

Una mateixa base de dades pot tenir diferents tipus d'accés.

En canvi, però, pot ser que l'actualització d'aquesta sigui diferent en cada cas. Per exemple, una base de dades en CD-ROM s'actualitzarà cada X mesos o anys, per a tornar a ser distribuïda, mentre que una base de dades "en línia" pot estar actualitzada en temps real.

### 4.4. Tipologia de BD segons la cobertura documental

Una altra classificació de les bases de dades documentals és en funció dels tipus de documents que emmagatzemen, tant centrades en un únic tipus de document com sobre diversos tipus.

#### Per a saber-ne més

En l'ítem sobre BD distribuïdes del subapartat "Tipus de bases de dades" podeu veure els avantatges de l'accés a BD en línia per Internet i una descripció de la tecnologia que s'utilitza.

- **Centrades en un únic tipus de document.** Recopilen i permeten la localització d'un tipus de document en concret com, per exemple, patents, tesis doctorals, informes, articles de revista, etc.
- **En diversos tipus de documents.** El seu objectiu és donar informació sobre una disciplina, incorporant, per tant, diferents tipologies documentals. Se centren en una temàtica específica.

#### 4.5. Tipologia de BD segons el model de tractament documental

Finalment, una darrera classificació de les BDD és segons el model de tractament documental, és a dir, segons les "operacions" sobre la informació del document que el productor ha dut a terme.

- **Bases de dades de sumaris** o sense anàlisi de contingut. Es componen de referències bibliogràfiques senzilles en què el productor es limita a gravar les dades de la mateixa font i no realitza cap anàlisi del document. Només incorporen les dades descriptives per tal de localitzar el document com, per exemple, l'autor, el títol i les dades de la font. En aquestes bases de dades, la cerca per matèries només es pot dur a terme per mitjà de les paraules del títol del document (normalment articles).
- **Catàlegs de biblioteques.** Aquestes bases de dades corresponen a fons continguts en una biblioteca o en una xarxa de biblioteques. Tenen una gran homogeneïtat.
- **BD amb anàlisi documental completa.** Sistemes d'informació que incorporen un nombre més gran de punts d'accés per tal de facilitar la localització per matèries. Cada registre bibliogràfic inclou un resum del contingut del document original i/o un conjunt de conceptes o termes representatius dels temes tractats. En aquest grup de bases de dades es pot distingir entre les que contenen classificació i resums, les que tenen una classificació i indexació per descriptors o paraules claus o les que disposen d'una classificació, una indexació i resums.
- **Índexs de cites.** Sistemes d'informació, en què a més d'extreure dades de descripció dels documents, també es tracten les referències bibliogràfiques citades en els articles de les revistes científiques.

#### Descriptor

Cadascun dels termes (o expressions) escollits entre un conjunt per a representar o etiquetar (en qualitat de terme preferit) un concepte susceptible de figurar amb certa freqüència en els documents indexables, i de ser utilitzat en les consultes. És la unitat mínima de significació que integra un tesaurus. És el terme pel qual seran indexats i recuperats els documents referits a la seva temàtica.

### **Tesaurus (del llatí *thesaurus*)**

Llista alfabètica de termes (mots i expressions) utilitzats per a representar els conceptes o temes dels continguts dels documents amb l'objecte d'efectuar una normalització terminològica que permeti accedir-hi millor.

És un intermediari entre el llenguatge natural dels documents i el llenguatge controlat usat pels especialistes d'un determinat camp de coneixement.

Els termes que conformen un tesaurus s'interrelacionen en tres modalitats de relacions:

- **Jeràrquiques:** estableixen subdivisions que generalment reflecteixen estructures del tipus tot/part.
- **D'equivalència:** controlen la sinonímia, homonímia, antonímia i polinímia entre els termes.
- **Associatives:** milloren les estratègies de recuperació i ajuden a reduir la multiplicitat de jerarquies entre termes.

## **4.6. Resum**

Les **bases de dades documentals** (BDD) es caracteritzen perquè cada registre es correspon amb un document de qualsevol tipus: publicació, document gràfic o sonor, o la seva referència.

Emmagatzema les referències i/o el text complet de documents de temàtica general que poden contenir text extens i dades multimèdia que es consulten per mitjà de paraules clau (que figuren en el text) o d'un *tesaurus* d'estructura jerarquizada.

La informació continguda en una BDD s'estructura en diferents camps per a facilitar-ne el control i accés individualitzat. Alguns camps es refereixen a la descripció formal del document, altres tracten del seu contingut temàtic; fins i tot poden guardar el mateix document.

Es poden classificar en diferents categories:

- **Segons la informació que contenen:** BD referencials (contenen informació descriptiva i referències que permeten localitzar els documents originals), BD de text complet (guarden el contingut complet dels documents originals) o arxius electrònics d'imatges (contenen referències que tenen un enllaç a la imatge del document original).
- **Segons el mode d'accés a la informació:** BD d'accés local, BD en dispositius físics d'emmagatzematge i BD en línia.
- **Segons la cobertura documental:** BD centrades en un únic tipus de document i BD amb diversos tipus de documents (centrades en una disciplina).
- **Segons el model de tractament documental:** BD de sumaris (o sense anàlisi de contingut), BD amb anàlisi documental completa, catàlegs de biblioteques i índexs de citacions.

## Exercicis d'autoavaluació

Seleccioneu per a cada pregunta la resposta adequada.

1. Si hi ha les memòries (RAM i ROM), per què són necessaris els dispositius per a emmagatzemar dades?

- a) Perquè la memòria RAM és volàtil.
- b) Perquè la memòria ROM és permanent, no es pot modificar (almenys no es pot modificar d'una manera senzilla i ràpida).
- c) Perquè la RAM és molt cara econòmicament i de capacitat limitada. Cal tenir dispositius amb gran capacitat d'emmagatzemar dades, encara que sigui pagant el preu d'una velocitat menor.
- d) Totes són certes.

2. Assenyaleu l'única de les quatre afirmacions que és certa:

- a) La memòria ROM només s'esborra quan es desendolla l'ordinador del corrent elèctric.
- b) En la memòria RAM es pot llegir i escriure.
- c) Els dispositius d'arxivament de dades no són necessaris si l'ordinador compta amb memòria RAM.
- d) Totes les afirmacions anteriors són falses.

3. Indiqueu quina de les afirmacions següents és vertadera:

- a) La memòria ROM és només de lectura i d'escriptura bipolar.
- b) La informació emmagatzemada en la memòria ROM es perd si després d'apagar l'ordinador es desendolla del corrent elèctric.
- c) En la memòria RAM es pot llegir i escriure informació i perdura fins i tot si apaguem l'ordinador.
- d) Cap de les anteriors és certa.

4. Indiqueu quina de les afirmacions següents sobre els dispositius d'emmagatzematge de dades és certa:

- a) Permet un volum més gran d'emmagatzematge de dades i amb més rapidesa que l'ordinador.
- b) Els diferents dispositius d'emmagatzematge existents ofereixen diferents prestacions i capacitats d'emmagatzematge.
- c) Perden la seva informació quan es desendollen del corrent elèctric.
- d) Permeten únicament l'escriptura de dades, ja que la lectura s'ha de fer per mitjà de la CPU.

5. Indiqueu quines de les afirmacions següents és falsa.

- a) En els suports seqüencials per a accedir al registre  $n$  s'ha de llegir necessàriament els  $n-1$  anteriors.
- b) Als suports d'accés directe no es pot accedir directament a menys que es conegui l'adreça física.
- c) Dins d'un fitxer els registres s'identifiquen per un únic camp que és la clau.
- d) La memòria intermèdia és un espai de la memòria principal que s'utilitza en les operacions de lectura i escriptura en els fitxers.

6. Indiqueu quina de les afirmacions següents és certa:

- a) Els fitxers permanents són els que, independentment del tipus de suport en què estiguin, no desapareixen en apagar l'ordinador.
- b) Els fitxers permanents són els que no pateixen modificacions en el seu contingut al llarg del temps.
- c) Els fitxers temporals només són necessaris per a operar amb la memòria principal.
- d) Els fitxers de maniobres permeten executar aplicacions en què els requeriments de dades superen la capacitat de la memòria principal.

7. Indiqueu quina és l'única afirmació vertadera entre les quatre següents:

- a) En l'operació de fusió de fitxers s'uneixen dos o més fitxers per a integrar-los en un de sol.
- b) L'operació de cerca sobre un fitxer consisteix a dur a terme els passos necessaris per a trobar l'arxiu dins del disc dur del PC.
- c) L'eliminació d'un registre és una operació de destrucció de fitxers.
- d) Totes les afirmacions anteriors són falses.

8. Seleccioneu l'única afirmació vertadera d'entre les quatre propostes següents:

- a) Hi ha dos tipus d'accés als fitxers: seqüencial directe i directe automàtic.
- b) Exclusivament, hi ha dos tipus d'organització de fitxers: relatiu i seqüencial.
- c) Hi ha quatre tipus d'organització de fitxers: seqüencial, seqüencial indexada, seqüencial encadenada i rotativa.
- d) Totes les afirmacions anteriors són falses.

9. Indiqueu quina de les afirmacions següents sobre fitxers és vertadera:

- a) Un fitxer es compon de camps que al seu torn estan compostos de registres.
- b) La informació continguda en un fitxer està desestructurada (d'aquí la necessitat de les bases de dades).
- c) El registre físic es correspon amb la quantitat d'informació que es pot llegir i escriure al mateix temps en suport físic.
- d) Totes les anteriors són certes.

10. Indiqueu quina afirmació de les quatre següents és vertadera:

- a) Les bases de dades són una col·lecció d'informació de tipus exclusivament informàtic.
- b) Les bases de dades eviten sense excepcions les redundàncies d'informació.
- c) Un sistema gestor de bases de dades (SGBD) accedeix a la informació des d'un llenguatge d'alt nivell anomenat "Llenguatge de manipulació de dades" (DML).
- d) Totes les afirmacions anteriors són falses.

11. Indiqueu quina afirmació de les quatre següents és vertadera:

- a) Les bases de dades asseguren que no hi haurà redundància en les dades que emmagatzemen en cap cas.
- b) En alguns casos les bases de dades permeten redundàncies a fi de millorar l'espai que necessiten per a emmagatzemar-se.
- c) Algunes vegades les bases de dades presenten redundàncies a fi de millorar els accessos i el rendiment.
- d) Totes les afirmacions anteriors són falses.

12. Seleccioneu l'afirmació falsa entre les quatre següents:

- a) Una base de dades és informació emmagatzemada de manera organitzada en un ordinador.
- b) La informació continguda en una base de dades pot ser de qualsevol tipus.
- c) La utilització d'una base de dades assegura l'eliminació de les inconsistències de dades.
- d) La utilització d'una base de dades presenta més avantatges que la utilització d'un sistema de fitxers.

13. Indiqueu quina de les afirmacions següents és vertadera:

- a) Les dades emmagatzemades en una base de dades no presenten redundàncies.
- b) Un bon sistema de base de dades ha de permetre fer canvis sobre el suport físic de la base de dades sense que per això es vegin afectats els programes que la utilitzen.
- c) Les bases de dades textuals són més modernes i més utilitzades que les jeràrquiques i que les relacionals.
- d) Totes les anteriors són falses.

14. Indiqueu quina de les afirmacions següents és certa:

- a) Les relacions entre diferents taules d'una base de dades relacional s'implementen mitjançant les claus internacionals.
- b) Una taula emmagatzema tota la informació de qualsevol tipus que ha de contenir la base de dades.
- c) Per a cada taula hi ha d'haver una clau primària que és un i només un dels seus camps que no repeteix un valor a la taula.
- d) Totes les anteriors són falses.

15. Quina de les afirmacions següents és certa?

- a) Quan la relació entre dues taules és d'un a molts és necessari crear una tercera taula per tal de poder representar el model físic.
- b) En el model conceptual una entitat està formada per una sèrie d'atributs.
- c) En la generació del model físic a partir del conceptual tota relació 1:n es converteix en una clau forània en la taula corresponent de grau 1.
- d) Totes les anteriors són falses.



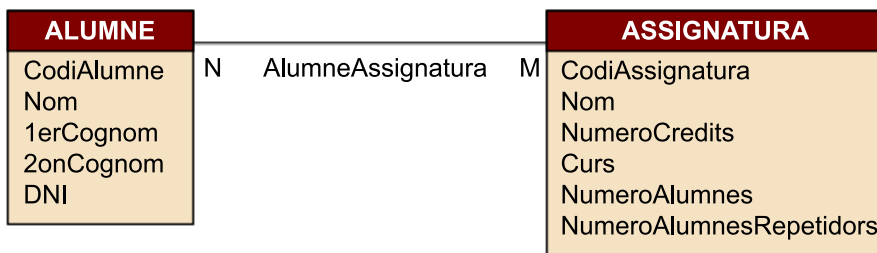
16. Indiqueu l'única afirmació que és vertadera entre les quatre següents:

- El grau d'una relació pot ser d'una a una (1:1) o d'una a dues (1:2).
- El grau d'una relació pot ser d'una a moltes (1:n) o de dues a moltes (2:n).
- El grau d'una relació ternària és sempre d'una a tres (1:3).
- Totes les afirmacions anteriors són falses.

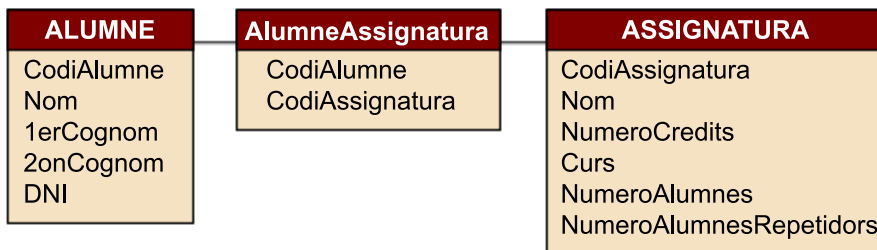
17. Quina afirmació és vertadera?

- Un atribut d'una entitat en el model conceptual es converteix en un camp d'una taula en el model físic.
- La relació n:1 indica que per a un determinat registre de la primera taula hi poden haver  $n$  registres a la segona entitat que continguin el mateix atribut que és clau primària en la primera taula.
- Les respostes *a* i *b* són certes.
- Totes són falses.

18. En el model conceptual d'una base de dades s'observen dues entitats amb una relació  $n:m$ . Segons la figura, cada alumne pot cursar una o més assignatures, i a cada assignatura hi ha un o més alumnes. Els atributs que formen claus primàries figuren subratllats.



Si es passa al model relacional...



... indiqueu quina de les afirmacions següents referida a la taula AlumneAssignatura és falsa.

- Es pot evitar que aparegui la taula de relació AlumneAssignatura que no existia en el model conceptual.
- La relació entre ALUMNE i AlumneAssignatura serà 1:n.
- La clau primària de la taula ALUMNE juntament amb la clau primària de la taula ASSIGNATURA formen la clau primària de la taula de relació AlumneAssignatura.
- Totes les afirmacions anteriors són falses.

19. Disseny d'una base de dades

Us demanem que dissenyeu una petita base de dades que permeti gestionar les activitats que s'organitzen en els diferents espais compartits (sala de conferències i aules de formació) i les associacions que participen en un casal d'entitats.

L'espai disposa de tres espais compartits:

- la sala de conferències,
- l'aula 1, destinada a accions de formació general,
- l'aula 2, equipada amb equips informàtics, per a poder-hi dur actuacions de formació en l'àmbit de les tecnologies de la informació.

Tot i així, aquests espais es poden ampliar amb el temps.

Es preveu disposar, per exemple, a mitjà termini, d'una tercera aula de formació i d'una sala d'exposicions.

Aquests espais els poden utilitzar per a les diferents activitats que s'organitzen, tant directament pel casal com per associacions de la comarca.

Així doncs, cal que la base de dades permeti emmagatzemar el següent:

- a) Llista d'associacions, amb les seves principals dades (NIF, nom entitat, responsable, adreça, telèfon de contacte, correu electrònic, web, etc.).
- b) Relació d'activitats organitzades (nom activitat, responsable, data d'inici i data de finalització, associació organitzadora, etc.).
- c) Espais disponibles (identificador espai, aforament màxim, etc.).
- d) L'ús dels espais, per part de les activitats, indicant els dies i horaris.
- e) I la cessió, d'altra banda, dels espais a les associacions.

En el cas *d*, cal considerar que una activitat pot utilitzar més d'un espai, en diferents dies. Per exemple, un curs de gestió d'entitats, que dura tres dissabtes i es du a terme durant els matins, utilitza el primer i el segon dia l'aula 1 i el tercer dia, l'aula 2, ja que s'expliquen eines de suport a la gestió d'entitats. En aquest cas, a més a més, aquesta activitat no l'organitza cap entitat, sinó el mateix casal.

Sempre cal considerar l'horari d'ús, matí o tarda.

Finalment, en el cas *e* hi ha situacions en què una associació demana l'ús d'un espai per a un dia (horari de matí o de tarda) per tal de dur-hi a terme alguna actuació interna o activitat que no es registra com a tal.

En aquest cas, es parla d'una cessió de l'espai.

Per exemple, l'Associació Jove Cambra pot haver sol·licitat la cessió de la sala de conferències per a un divendres a la tarda, per tal de fer-hi la seva assemblea general.

En aquests casos, només ens interessa saber quin espai ha estat cedit a l'Associació i per a quin dia i quin horari, però no enregistrem l'activitat.

Es demana, doncs, que realitzeu les activitats següents:

- a) Disseny del **model conceptual de la base de dades** (model entitat-relació) Identificació de les entitats, els seus atributs i les interrelacions entre elles, amb els atributs, si fos el cas.
- b) Generació del **model relacional de la base de dades**. Partint del model conceptual dissenyat en l'apartat 1, s'ha de convertir en una base de dades relacional, indicant les taules corresponents, els camps de cadascuna i el seu tipus, la clau primària de cada tipus i les claus foranes corresponents.

## Solucionari

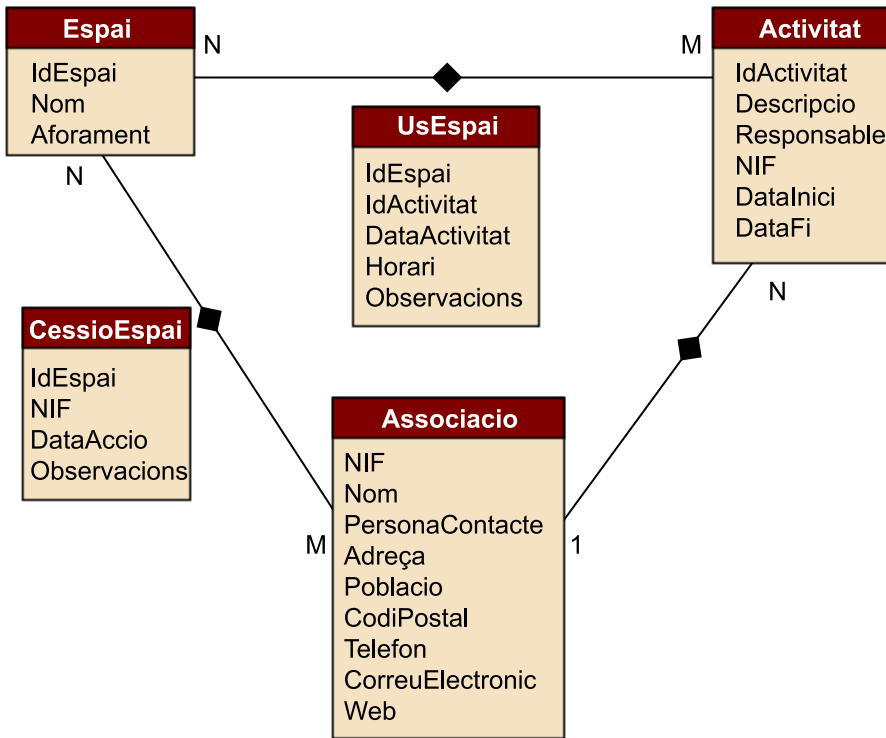
### Exercicis d'autoavaluació

- 1) La resposta correcta és *d*. Totes les opcions són certes.
- 2) La resposta correcta és *b*. En la memòria RAM es pot llegir i escriure.
- 3) La resposta correcta és *d*. Cap de les opcions anteriors és certa.
- 4) La resposta correcta és *b*. Els diferents dispositius d'emmagatzematge existents ofereixen diferents prestacions i capacitats d'emmagatzematge.
- 5) La resposta correcta és la *c*. Dins d'un fitxer, els registres s'identifiquen per un únic camp que és la clau.
- 6) La resposta correcta és la *d*. Els fitxers de maniobres permeten executar aplicacions en què els requeriments de dades superen la capacitat de la memòria principal.
- 7) La resposta correcta és la *a*. En l'operació de fusió de fitxers s'uneixen dos o més fitxers per a integrar-los en un de sol.
- 8) La resposta correcta és la *d*. Totes les afirmacions anteriors són falses.
- 9) La resposta correcta és la *c*. El registre físic es correspon amb la quantitat d'informació que es pot llegir i escriure al mateix temps en suport físic.
- 10) La resposta correcta és la *c*. Un sistema gestor de bases de dades (SGBD) accedeix a la informació des d'un llenguatge d'alt nivell anomenat *llenguatge de manipulació de dades* (DML).
- 11) La resposta correcta és la *c*. Algunes vegades les bases de dades presenten redundàncies a fi de millorar els accessos i el rendiment.
- 12) La resposta correcta és la *c*. La utilització d'una base de dades assegura l'eliminació de les inconsistències de dades.
- 13) La resposta correcta és la *b*. Un bon sistema de base de dades ha de permetre fer canvis sobre el suport físic de la base de dades sense que per això es vegin afectats els programes que la utilitzen.
- 14) La resposta correcta és la *d*. Totes les anteriors són falses.
- 15) La resposta correcta és la *b*. En el model conceptual, una entitat està formada per una sèrie d'atributs.
- 16) La resposta correcta és la *d*. Totes les afirmacions anteriors són falses.
- 17) La resposta correcta és la *c*. Les opcions *a* i *b* són certes.
- 18) La resposta correcta és la *a*. Es pot evitar que aparegui la taula de relació "AlumneAssignatura" que no teníem en el model conceptual.
- 19) Disseny d'una base de dades

Aquest exercici no té una única solució, ja que depenent del disseny del model conceptual, la base de dades relacional final serà una o altra.

En una situació real, durant el disseny de la base de dades, el responsable sol·licita diferents requeriments al "client" per anar determinant possibles ambigüitats i dubtes. En el vostre cas, el consultor de l'assignatura actua com a client.

**a) Disseny del model conceptual** (entitats i interrelacions, amb els seus atributs)



Les entitats es mostren en rectangles amb el nom a la capçalera i els atributs a continuació. Les relacions entre entitats es mostren amb un rombe al costat del qual figura el seu nom seguit dels seus atributs. La cardinalitat de les relacions s'indica mitjançant 1, N o M al costat de les entitats que hi participen. Els atributs que formen clau primària estan subratllats.

**b) Generació del model relacional (amb MS Access)**

- Disseny de les taules corresponents a les entitats del model conceptual: ASSOCIACIÓ, ACTIVITAT i ESPAI. Cada taula inclou la definició dels camps, els tipus de dades de cadascun i la identificació de la seva clau primària (a Access els camps que la formen s'assenyalen amb el símbol d'una clau).

Nombre del camp	Tipo de datos	Descripción
<u>NIF</u>	Texto	
Nom	Texto	Nom de l'entitat o associació
PersonaContacte	Texto	
Adreça	Texto	
Poblacio	Texto	
CodiPostal	Texto	
Telefon	Texto	
CorreuElectronic	Texto	E-mail de la persona de contacte
Web	Texto	Página web de l'entitat o associació

Nombre del campo	Tipo de datos	Descripción
<u>IdActivitat</u>	Texto	
Descripció	Texto	Descripció de l'activitat
Responsable	Texto	Persona que coordina l'activitat
NIF	Texto	NIF de l'entitat organitzadora (-> taula ASSOCIACIÓ)
DataInici	Fecha/Hora	
DataFi	Fecha/Hora	

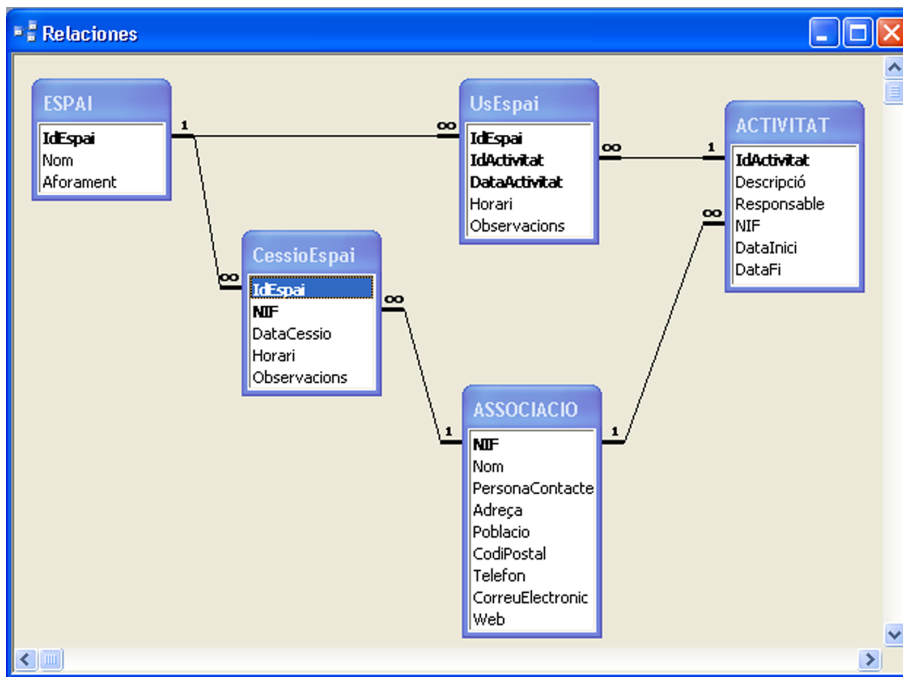
ESPAI : Taula			
Nombre del campo	Tipo de datos	Descripción	
IdEspai	Texto		
Nom	Texto	Nom de l'espai (sala, aula...)	
Aforament	Número	Capacitat màxima de persones	

- Disseny de les taules intermèdies generades a partir de les relacions M:N del model conceptual: ÚEspai i CessióEspai.

UsEspai : Taula			
Nombre del campo	Tipo de datos	Descripción	
IdEspai	Texto	IdEspai utilitzat (-> taula ESPAI)	
IdActivitat	Texto	IdActivitat realitzada (-> taula ACTIVITAT)	
DataActivitat	Fecha/Hora	Data d'utilització de l'espai	
Horari	Texto	(matí o tarda)	
Observacions	Texto		

CessióEspai : Taula			
Nombre del campo	Tipo de datos	Descripción	
IdEspai	Texto	IdEspai cedit (-> taula ESPAI)	
NIF	Texto	NIF de l'entitat (-> taula ASSOCIACIO)	
DataCessio	Fecha/Hora	Data de cessió	
Horari	Texto	(matí o tarda)	
Observacions	Texto		

- La relació entre les entitats ASSOCIACIÓ i ACTIVITAT no requereix una taula intermèdia, ja que és de tipus un a molts (1:n). És a dir, una associació pot organitzar diverses activitats, però una activitat només l'organitza una associació.
- Implementació de les relacions entre les taules de la BD.



Els conjunts de camps que formen les claus primàries s'assenyalen en negreta.

Els camps que són clau forana figuren al costat del símbol d'infinít ( $\infty$ ).

## **Bibliografia**

**Silberschatz, A.; Korth, H. L.; Sudarsahn, S.** (1998). *Fonaments de bases de dades (3)*. Madrid: McGraw-Hill.

**Ullman, J.** (1991). *Principles of database and knowledge - Base systems*. Computer Science Press.

**De Miguel, A.; Piattini, M.** (1999). *Fonaments i models de bases de dades (2)*. Madrid: RA-DT.

**Pressman, Roger S.** (1997). *Ingeniería de Software. Un enfoque práctico*. Madrid: McGraw-Hill

