

Unitat 1 Introducció a les bases de dades.

- 1. Introducció
- 2. Fitxers tradicionals
- 3. Concepte de Base de Dades (BD)
 - 3.1. Definició de Base de Dades
 - 3.2. Comparació amb fitxers tradicionals.
 - 3.3. Conceptes bàsics d'una base de dades.
- 4. Models de dades
 - 4.1. Arquitectura a 3 nivells
- 5. Sistemes Gestor de Bases de Dades.
 - 5.1. Característiques desitjables d'un SGBD.
 - 5.2. Llenguatges del SGBD
 - 5.3. Funcions o serveis dels SGBD
 - 5.4. Usuaris de una Base de Dades
 - 5.5. Classificacions dels SGBD



Fecha	Versión	Descripción
31/08/2022	1.0.0	Versión inicial

Unitat 1 Introducció a les bases de dades.

1. Introducció

A l'entorn del mercat actual, la competitivitat i la rapidesa de maniobra d'una empresa són imprescindibles per al seu èxit. Per a aconseguir-ho existeix cada vegada una major demanda de dades i, per tant, més necessitat de gestionar-los. Aquesta demanda sempre ha estat palesa en empreses i societats, però en aquests

anys s'ha disparat a causa de l'accés multitudinari a les xarxes integrades en Internet i a l'aparició dels dispositius mòbils que també requereixen eixa informació.

En informàtica es coneix com a **dada** a qualsevol element informatiu que tinga rellevància per a un usuari. Des del seu naixement, la informàtica s'ha encarregat de proporcionar eines que faciliten la manipulació de les dades. Abans de l'aparició de les aplicacions informàtiques, les empreses tenien com a úniques eines de gestió de dades els *fitxers* amb calaixos, carpetes i fitxes de cartó. En aquest procés manual, el temps requerit per a manipular aquestes dades era enorme. Però la pròpia informàtica ha adaptat les seues eines perquè els elements que l'usuari utilitza quant a maneig de dades se semblen als manuals. Per això se segueix parlat de fitxers, formularis, carpetes, directoris,....

La clientela fonamental del professional informàtic és l'empresa. L'empresa es pot entendre com un sistema d'informació format per diversos objectes: el capital, els recursos humans, els immobles, els serveis que presta, etc.

Els sistemes d'informació actuals es basen en **bases de dades** (BD) i **sistemes de bases de dades** (SGBD) que s'han convertit en elements imprescindibles de la vida quotidiana de la societat moderna.

2. Fitxers tradicionals

Els fitxers tradicionals, que són molt útils per a aplicacions menudes i concretes, comencen a donar problemes quan el sistema d'informació va creixent.

Anem a veure un exemple que il·lustre l'explicació.

Suposem una empresa menuda que ha informatitzat la informació dels seus empleats. En un principi només volia una informació general d'ells, per a poder tenir uns llistats, enviar cartes a tots, etc. Així es podria plantejar un fitxer on guardar la informació dels empleats: nom, cognoms, DNI, adreça, telèfon i data de naixement. Tots els programes que utilitzen el fitxer (en un principi molt pocs) han de tenir la declaració dels camps del fitxer. Per exemple, si són en llenguatge C, la definició haurà de ser com la següent:

```
...
```

```
struct TEmpleat
```

```
{
```

```
    char dni[10];
```

```

char nom[30];

char adreca[30];

char telefon[10];

char data_n[10]

} v_empleat;

FILE *pFEmpleats;

...

```

I un exemple del contingut d'aquest fitxer podria ser

18876543	Llopis Bernat, Jaume	C/ Artana, 3	964213243	07/12/1955
18900111	Garrido Vidal, Rosa	C/ Herrero, 54	964253545	25/01/1958
18922222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	964216191	08/06/1959
18932165	Folch Mestre, Pilar	C/ Palància, 22	964234567	08/06/1960
18933333	Peris Andreu, Joan	C/ Balmes, 3	964223344	15/03/1960
18934567	Sebastià Broch, Ferran	C/ Magallanes, 38	964281706	14/07/1962
18944444	Garcia Tomàs, Àlicia	C/ Amunt, 15	964205080	10/05/1964

Suposem ara que l'empresa es planteja ampliar el seu sistema informàtic, i incloure el sou de cada empleat per a poder enviar també les nòmines. Aleshores decideix ampliar el fitxer, incloent el nou camp (sou). El primer que s'hauria de fer és canviar el fitxer, ja que l'estructura no és la mateixa. S'haurà de crear un fitxer amb la nova estructura, fer un programeta per a passar les dades del fitxer vell al fitxer nou, esborrar el vell i en tot cas canviar el nom del nou. Després s'hauran de modificar tots els programes que utilitzaven el fitxer per a que l'estructura del fitxer siga la correcta, tornar a compilar-los, ...

```

struct TEmpleat
{
char dni[10];

char nom[30];

char adreca[30];

char telefon[10];

char data_n[10];

int sou;

} v_empleat;

FILE *pFEmpleats;

```

I un exemple del contingut d'aquest fitxer podria ser

18876543	Llopis Bernat, Jaume	C/ Artana, 3	964213243	07/12/1955	2100
18900111	Garrido Vidal, Rosa	C/ Herrero, 54	964253545	25/01/1958	2000
18922222	Nebot Aliaga, Carme	C/ Sant Vicent, 5	964216191	08/06/1959	1500
18932165	Folch Mestre, Pilar	C/ Palància, 22	964234567	08/06/1960	3000
18933333	Peris Andreu, Joan	C/ Balmes, 3	964223344	15/03/1960	2100
18934567	Sebastià Broch, Ferran	C/ Magallanes, 38	964281706	14/07/1962	2500
18944444	Garcia Tomàs, Àlicia	C/ Amunt, 15	964205080	10/05/1964	2500

Suposem que l'empresa decideix continuar ampliant la informació, i ara vol mantenir informació dels diferents departaments, i els empleats que pertanyen. Potser per a no haver de repetir el procés d'adequació, es construeix un fitxer nou d'empleats on està el DNI de l'empleat, el nom, el telèfon i el departament al qual pertany, ja que aquesta és la informació que li interessa.

```
struct TEmpleat2
```

```
{ char dni[10];
```

```
char nom[30];
```

```
char telefon[10];
```

```
char departament[20];
```

```
} v_empleat2;
```

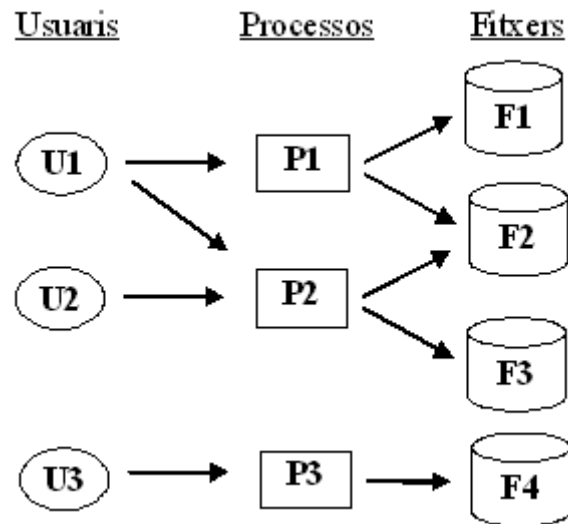
```
FILE *pFEmpleats2;
```

I un exemple del contingut d'aquest fitxer podria ser ara

18876543	Llopis Bernat, Jaume	964213243	Comptabilitat
18900111	Garrido Vidal, Rosa	964253545	Administració
18922222	Nebot Aliaga, Carme	964216191	Comercial
18932165	Folch Mestre, Pilar	964234567	Direcció
18933333	Peris Andreu, Joan	964223344	Comptabilitat
18934567	Sebastià Broch, Ferran	964281706	Informàtica
18944444	Garcia Tomàs, Àlicia	964205080	Comercial

En definitiva, com el sistema està orientat als processos, es tendeix, a mida que creix el sistema, a tenir molts fitxers, a més amb molta redundància. El problema de la redundància és, a més de desaprofitar espai, la possible inconsistència de les dades. Per exemple, suposem que ara un determinat empleat canvia de telèfon. No s'haurà de canviar aquest en un fitxer, sinó en dos. Si només es canviara en un d'ells hi hauria inconsistència.

Aquest seria l'esquema d'un sistema basat en fitxers tradicionals.



on recordeu que en cada programa ha d'estar definida l'estructura dels fitxers que utilitza.

Els problemes més importants que porta aquest sistema que ha anat creixent serien:

- **Redundància de les dades.** Les mateixes dades estan en múltiples fitxers, la qual cosa comporta ocupar més lloc del necessari, i a més una dificultat de manteniment per a que les dades siguin consistents.
- **Dificultat per a modificar l'estructura dels fitxers.** Ja s'ha comentat abans que s'haurien de modificar tant el fitxer (amb l'actualització del fitxer antic al nou) com els programes que l'utilitzen.
- **Problemes de seguretat.** Quan hi ha molts usuaris que accedeixen als fitxers s'ha de vigilar molt que no es puguin fer actualitzacions no autoritzades

Tornarem a comentar els problemes quan vegem la comparació entre fitxers tradicionals i Bases de Dades.

3. Concepte de Base de Dades (BD)

Cada dia, la majoria de nosaltres ens trobem amb activitats que requereixen algun tipus d'interacció amb una base de dades (ingrés en un banc, reserva d'una entrada per al teatre, sol·licitud d'una subscripció a una revista, compra de productes, ...).

Aquestes interaccions són exemples del que es diu aplicacions tradicionals de bases de dades (bàsicament informació numèrica o de text), encara que els avanços tecnològics han permès que també existisquen: bases de dades multimèdia, sistemes d'informació geogràfica (GIS), magatzems de dades, sistemes de procés analític en línia,

3.1. Definició de Base de Dades

Una **base de dades** s'entendrà com una col·lecció o conjunt de dades relacionades entre si, independent dels programes que la utilitzen i sense redundàncies innecessàries.

Per **dades** volem dir fets coneguts que poden registrar-se i que tenen un significat implícit. Es a dir, no intentarem agafar totes les dades del món, sinó únicament sobre les coses que volem estudiar, i d'elles únicament les dades interessants, per aconseguir el propòsit perseguit. Per tant, una Base de Dades té una font de la qual es deriven les dades i uns usuaris als quals interessin les dades.

Exemple

Una agenda amb els noms i telèfons d'un conjunt de persones conegudes és una base de dades, ja que és una col·lecció de dades relacionades amb un significat implícit.

La definició presentada anteriorment fa referència a dos elements perquè un conjunt de dades constituïska una Base de dades:

1. **Relacions entre dades**, tema que es tractarà en les seccions següents.
2. **Significat implícit** de les dades que s'atribueix depenent del context en què s'utilitzen els mateixos. Per exemple, la dada data en una base de dades de VENDES pot referir-se a la data d'emissió de les factures, mentre que si la base de dades és de MÚSICA potser correspon a la data en què es va gravar un tema musical. És a dir, el significat d'una dada, depén de la BD que la continga.

Per a manipular i gestionar les bases de dades van sorgir ferramentes software denominades: **sistemes gestors de bases de dades** (SGBD en endavant) sobre els quals s'aprofundirà mes en davant.

3.2. Comparació amb fitxers tradicionals.

Una vegada hem vist la definició de Base de Dades i alguna de les seues característiques anem a comparar amb els fitxers tradicionals per veure els **avantatges**.

- Una Base de Dades és autodescriptiva, és a dir la definició de l'estructura, tipus de dades, etc. forma part de la pròpia base de dades (concretament en el **diccionari de dades**). En els fitxers tradicionals l'estructura d'aquestos havia d'anar en cada programa.
- Independència de les dades dels programes. És conseqüència de l'anterior. Així, en una BD, la modificació de l'estructura (afegint algun camp o modificant-ne

algun) no afecta als programes que puguin accedir a ella. Ja havíem vist que no era així en els fitxers tradicionals.

- No hi ha redundància, o molt poca. Potser convenient un poc de redundància per un millor rendiment, però aquesta pot ser controlada (per a evitar inconsistències, ...)
- Múltiples vistes d'usuari. Podríem pensar que al concentrar en una única BD totes les dades es perd la visió particular d'un usuari (que en fitxers tradicionals són els fitxers individuals de cadascun d'ells). Però no és així, ja que en les BD es poden definir vistes, que són la part de les dades que li interessin a un usuari determinat.
- Compartiment de les dades per part de molts usuaris, amb accés simultani sobre elles. Haurà d'haver en el software un control de concurrència, per assegurar que l'actualització simultània d'uns quants usuaris siga correcta.

Els **inconvenients** que suposa la utilització de BD són els següents:

- El Hardware i el software són més costosos.
- Necessitat de personal especialitzat.
- La implantació és més llarga i difícil, i per tant és una solució bona a mig i llarg termini.

És a dir, que si és una aplicació senzilla, potser siga convenient utilitzar fitxers tradicionals, però si és d'una certa envergadura o s'ha d'anar ampliant al llarg del temps, convé una BD.

3.3. Conceptes bàsics d'una base de dades.

Un dels grans problemes al qual s'enfronten els informàtics quan comencen el seu aprenentatge, és el gran nombre de termes desconeguts que ha d'assimilar, incloent l'enorme nombre de sinònims i sigles que s'utilitzen per a nomenar la mateixa cosa. A continuació vam mostrar, a manera de resum, alguns dels components que es poden trobar en una base de dades:

- **Dada:** La dada és un fet conegut que pot registrar-se i que té un significat implícit. Per exemple, els noms, números telefònics i direccions de persones que coneixem. Les dades es caracteritzen per pertànyer a un tipus.
- **Tipus de dada:** El tipus de dada indica la naturalesa del camp. Així, es pot tindre dades numèriques, que són aquells amb els quals es poden realitzar càlculs aritmètics (sumes, restes, multiplicacions...), les dades alfanumèriques, que són els que contenen caràcters alfabètics i números...
- **Taula o Entitat.** És un conjunt de files i columnes sota un mateix nom que representa el conjunt de valors emmagatzemats per a una sèrie de dades. Per

exemple, la informació de tots els clients d'una BD s'emmagatzemaran en una taula anomenada CLIENTS.

- **Camp.** Cadascuna de les columnes d'una taula. Cada camp pertany a un tipus de dades. Per exemple, el camp "*DataNaiximent*" representa les dates de naixement de tots els clients que conté una taula CLIENTS. Aquest camp pertany al tipus de dada *Data*. Al camp també se l'anomena *columna* o *atribut*.
- **Registre.** És el conjunt de dades referents a un mateix concepte o succés. Per exemple, les dades d'un client poden ser, el seu NIF, nom, any de naixement, direcció, ... Als registres també se'ls diuen *files* o *tuplas*.
- **Clau Primària.** Cada entitat té una clau primària o *camp clau* o *clau* que identifica unívocament al conjunt de dades. Així, el NIF és únic per a cada persona, per tant és un camp clau.
- **Clau forana.** Quan en una entitat figura la clau primària d'una altra entitat, aquesta es denomina **clau forana** o **clau aliena**. Les entitats es **relacionen** entre si a través de les claus foranes.
- **Consulta:** És una instrucció per a fer peticions a una BD.
- **Índex:** És una estructura que emmagatzema els camps clau d'una taula, organitzant-los per a fer més fàcil trobar i ordenar els registres.
- **Vista:** S'obtenen en guardar una consulta d'una o diverses taules. D'aquesta manera s'obté una taula virtual, és a dir, no està emmagatzemada en els dispositius d'emmagatzematge de l'ordinador, encara que sí que s'emmagatzema la seua definició.
- **Informe:** És un llistat ordenat dels camps i registres seleccionats en un format fàcil de llegir. Per exemple, un informe de les factures impagades del mes de gener ordenades per nom de client.
- **Guions o scripts:** Són un conjunt d'instruccions, que executades de forma ordenada, realitzen operacions avançades o manteniment de les dades emmagatzemades en la BD.
- **Procediments:** Són un tipus especial de *script que estan emmagatzemats en la *BD i formen part del seu esquema.

4. Models de dades

Per a poder definir l'estructura de la BD ens farà falta alguna ferramenta, alguna manera d'expressar el que veiem en el món real.

Un **Model de Dades** serà un conjunt de conceptes que poden servir per a descriure, a distints nivells d'abstracció, l'estructura d'una BD. A aquesta estructura expressada per mig del model de dades l'anomenarem **esquema**. Caldrà diferenciar entre esquema i les dades concretes que pot tenir la BD en un moment determinat

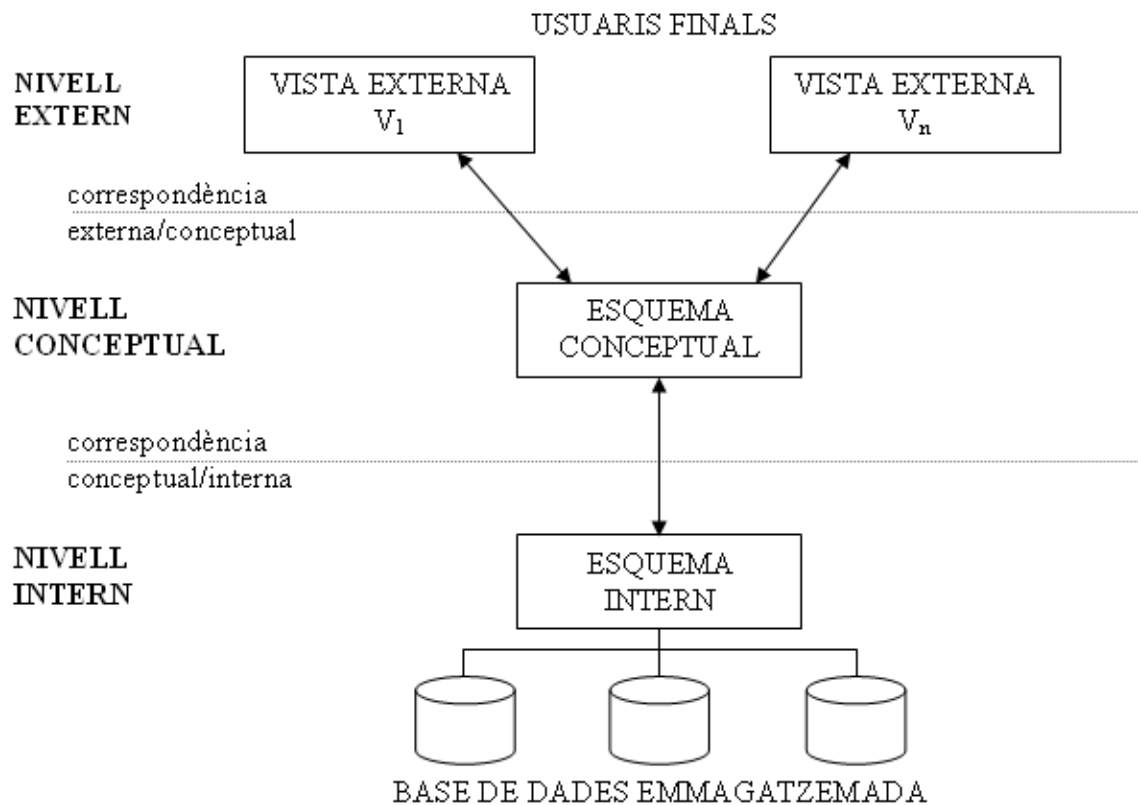
(a la qual cosa alguns autors anomenen **estat de la BD** o conjunt d'**ocurrències** o **exemplars**).

Al llarg de la història s'han proposat molts models de dades, tants que s'hauran de classificar un poc.

- **Models d'alt nivell** o **conceptuals**: disposen de conceptes molt propers a la manera de percebre les dades les persones. El més conegut és el **Model Entitat-Relació**. Uns altres serien els **orientats a objectes**. Serviran per a poder dissenyar la BD a partir de les percepcions del món real.
- **Models de baix nivell** o **físics**: proporcionen detalls de com es guardaran les dades a l'ordinador.
- Models de **representació** o **implementació**: estan a mig camí dels dos anteriors, ja que els conceptes els poden entendre els usuaris finals, encara que no estan massa allunyats de la manera com es guarden les dades a l'ordinador. Són els més utilitzats en els SGBD actuals. Es poden destacar el **Jeràrquic**, el **de Xarxa** (o **CodasyI**) i el **Relacional**, que és el que nosaltres veurem en profunditat.

4.1. Arquitectura a 3 nivells

Per a estandarditzar l'arquitectura dels SGBD, el comitè *ANSI/X3/SPARC* va proposar una arquitectura a 3 nivells, per assegurar algunes de les característiques que hem vist com a desitjables, en concret la de la separació entre programes i dades, les múltiples vistes i la utilització d'un catàleg per a la descripció de la B.D. Aquest són els 3 nivells:



1. Al **NIVELL INTERN** o **FÍSIC** hi ha un **esquema intern** que descriu l'estructura física d'emmagatzematge de la BD. S'utilitza un model físic, i descriu tots els detalls per al seu emmagatzematge (on estan els fitxers, quants, estratègies d'accés a les dades, ...).
2. Al **NIVELL CONCEPTUAL** o **LÒGIC** hi ha un **esquema conceptual**, que descriu l'estructura de tota la BD per al conjunt dels usuaris. Oculta els detalls de les estructures físiques d'emmagatzematge. S'utilitza un model de dades conceptual o d'implementació.
3. Al **NIVELL EXTERN** o **DE VISTES** s'inclouen diversos **esquemes externs** o **vistes d'usuari**. Cada esquema extern descriu la part de la B.D. que interessa a un grup d'usuaris determinat, i oculta la resta de la BD. S'utilitza també un model conceptual o d'implementació.

Per tant els tres esquemes són distintes maneres de descriure les dades, encara que aquestes només existeixen realment en el nivell físic. Però l'esquema intern ha de ser totalment transparent als usuaris, i ells han de "veure" el seu esquema extern. Qualsevol referència a aquest esquema s'haurà de traduir, per part del SGBD, a referències a les dades oportunes de l'esquema lògic. I posteriorment s'haurà de traduir en una sol·licitud a l'esquema físic. Per exemple, suposem una vista on tenim el nom i l'edat dels empleats. Una sol·licitud de tota la vista externa s'haurà de traduir en una sol·licitud de les dades correctes de l'esquema lògic, el nom i la data de naixement. I per la seua banda s'haurà de traduir en una sol·licitud a l'esquema físic, on se sabran on estan exactament les dades, si hi ha índex per fer més ràpid

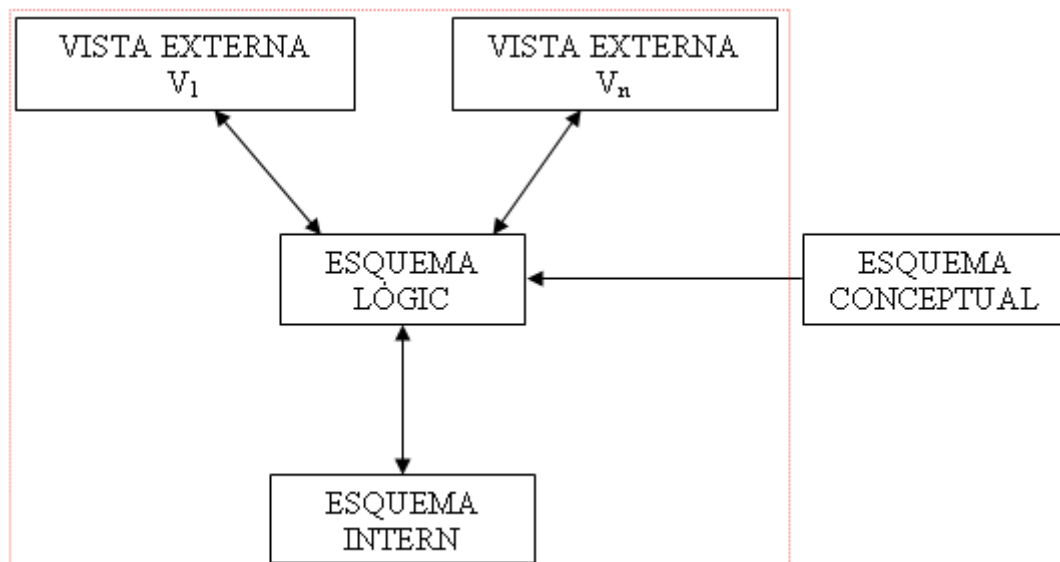
l'accés, ... Posteriorment quan ja estan les dades, s'hauran de passar al nivell superior, i després de fer el càlcul oportú passar-les al nivell extern.

El procés de transformar sol·licituds i resultats d'un nivell a un altre s'anomena correspondència o transformació (*mapping*). Evidentment consumiran temps, però facilitaran el que es pretenia:

- **Independència lògica respecte a les dades:** es pot modificar l'esquema lògic sense haver de modificar els esquemes externs. Per exemple, en l'esquema lògic puc incorporar un camp per als empleats que siga la data d'ingrés en l'empresa. Això no afectarà per a res a la vista externa d'empleats i edats.
- **Independència física respecte a les dades:** es pot modificar l'esquema físic sense haver de modificar l'esquema lògic, i molt menys els esquemes externs. Així per exemple es pot afegir més espai per a la BD incorporant un nou fitxer on es guardaran les coses (Oracle), o s'inclou un nou índex per accedir més ràpidament a les dades per un determinat ordre.

També hem de dir que no sempre els SGBD comercials compleixen estrictament aquestos 3 nivells en la seua arquitectura. Sobretot en els més menuts, ja que obligatòriament la correspondència entre els nivells tarda temps, i retarda els resultats.

Anem a ampliar l'anterior arquitectura a tres nivells per incorporar tot el procés que farem en la construcció d'una BD.



On hem incorporat, fora de l'arquitectura a 3 nivells, és a dir fora del SGBD, l'esquema conceptual, que seria la concepció de la B.D. en un model d'alt nivell com el Model E/R. A partir d'ell realitzaríem l'esquema lògic (amb un model d'implantació). Després ja es passaria a l'esquema intern, i també als esquemes externs.

5. Sistemes Gestor de Bases de Dades.

Com ja hem comentat al principi, per a manipular i gestionar les bases de dades van sorgir ferramentes software denominades **sistemes gestors de bases de dades** (SGBD).

Així podem definir un sistema gestor de bases de dades (SGBD) com el *conjunt de ferramentes que permeten als usuaris definir, crear i mantindre una base de dades, i proporcionen accés controlat a aquesta*.

5.1. Característiques desitjables d'un SGBD.

Ja hem vist algunes característiques que han de tenir els SGBD. Anem a completar aquest llista.

- **Control de redundància.** Ja hem comentat que potser siga convenient en algunes ocasions un poc de redundància. Però en cas que hi haja, aquesta ha d'estar controlada. És a dir, el SGBD hauria de proporcionar mecanismes per a controlar les dades que estan duplicades, i que quan una informació s'actualitze en un lloc, automàticament s'actualitze en els altres.
- **Restricció d'accessos no autoritzats.** Quan molts usuaris utilitzen la B.D., és normal que no tots tinguin autorització per a accedir a tota la informació. Pot haver informació confidencial a la qual pocs usuaris tenen accés, i pot haver informació a la qual un usuari pot tenir accés però que no pot modificar. El SGBD ha de tenir, per tant, un subsistema de seguretat i autorització per a crear usuaris (que s'identifiquen amb una contrasenya) i donar distints permisos d'accés a cadascun.
- **Subministrament de múltiples interfícies per als usuaris.** Com que hi ha molts tipus d'usuaris, amb nivells de coneixements tècnics distints, el SGBD ha d'oferir diferents interfícies: llenguatges de consulta per a usuaris esporàdics, llenguatges de definició i control per a administradors, formularis per a usuaris habituals, ...
- **Compliment de restriccions.** Hi haurà restriccions que han de complir les dades. Per exemple la nota d'un alumne ha d'estar entre 0 i 10; la data de naixement no pot ser posterior a l'actual; el curs on està matriculat un alumne ha de ser un curs existent; un alumne que estiga de baixa ha de tenir contingut en la data de baixa, o no se li poden posar notes; ... Algunes d'aquestes restriccions les suportarà directament el SGBD. Altres requeriran verificació per mig de programa quan s'introdueixi o es modifiqui la dada.
- **Recolzament i recuperació.** Tot SGBD ha de comptar amb recursos per a recuperar-se d'errades del hardware o del software i deixar les dades com

estaven abans de la fallida.

5.2. Llenguatges del SGBD

La principal eina d'un SGBD és la interfície de programació amb l'usuari. Aquesta interfície consisteix en un llenguatge molt senzill mitjançant el qual l'usuari interactua amb el servidor. Aquest llenguatge comunament es denomina **SQL**, *Structure Query Language*, està estandarditzat per l'ISO, és a dir, totes les BD que suporten SQL han de tindre la mateixa sintaxi a l'hora d'aplicar el llenguatge.

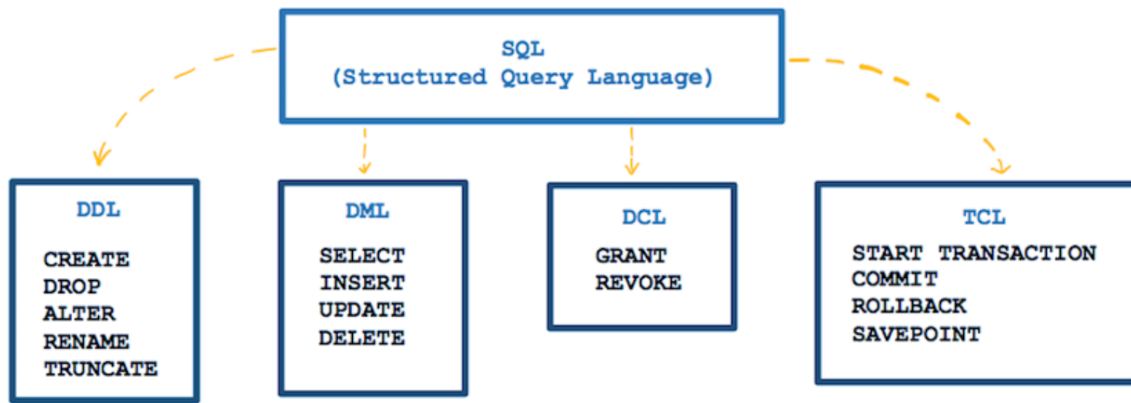
El llenguatge SQL es divideix en 4 subllenguatges, i entre tots permeten complir amb les funcions o serveis de tot SGBD:

Llenguatge de definició de dades (DDL: *data definition language*). Aquest llenguatge permet crear tota l'estructura d'una base de dades (des de taules fins a usuaris). Les seues clàusules són del tipus DROP (eliminar objectes) i CREATE (crear objectes). Detallarem la seua sintaxi i ús en la UD4.

Llenguatge de manipulació de dades (DML: *data manipulation language*). Aquest llenguatge amb 4 sentències senzilles seleccionar determinades dades (SELECT), inserir dades (INSERT), modificar-los (UPDATE) o fins i tot esborrar-los (DELETE). Detallarem la seua sintaxi i ús en les UD5, UD6 i UD7.

Llenguatge de control de dades (DCL: *data control language*). Inclou comandos (GRANT, REVOKE) que permeten a l'administrador gestionar l'accés a les dades contingudes en la base de dades.

Llenguatge de control de transaccions (TCL). El propòsit d'aquest llenguatge és permetre executar diversos comandos de manera simultània com si fora un comando atòmic o indivisible. Si és possible executar tots els comandos, s'aplica la transacció (COMMIT), i si, en algun pas de l'execució, succeeix una cosa inesperada, es poden desfer tots els passos donats (ROLLBACK).



<http://josejuansanchez.org/bd>

5.3. Funcions o serveis dels SGBD

En general, un SGBD proporciona els següents serveis:

- Permet la **definició** de la base de dades mitjançant el llenguatge de definició de dades (*DDL – Data Description Language*). Aquest llenguatge permet especificar l'estructura i el tipus de les dades, així com les restriccions sobre les dades. Tot això s'emmagatzemarà en la base de dades.
- Permet la **inserció, actualització, eliminació i consulta** de dades mitjançant el llenguatge de maneig o manipulació de dades (*DML - Data Manipulation Language*).
- Proporciona un accés controlat a la base de dades mitjançant:
 - Un sistema de seguretat, de manera que els usuaris no autoritzats no puguin accedir a la base de dades, mitjançant el llenguatge de control de dades (*DCL - Data Control Language*);
 - Un sistema d'integritat que manté la integritat i la consistència de les dades;
 - Un sistema de control de concurrència que permet l'accés compartit a la base de dades;
 - Un sistema de control de recuperació que restableix la base de dades després que es produïska una fallada del maquinari o del programari;
 - Un **diccionari de dades** o catàleg accessible per l'usuari que conté la descripció de les dades de la base de dades.

5.4. Usuaris de una Base de Dades

Els múltiples usuaris que poden utilitzar la BD són de distints tipus. Anem a fer una classificació d'aquests. Aquesta classificació agrupa dos grans blocs: els usuaris amb coneixements informàtics i els usuaris que no en tenen (o no tenen per què tenir-ne).

En el primer bloc, usuaris informàtics podem trobar:

- **Disenyador** de la BD: Encarregats de dissenyar la BD, triar el tipus d'informació necessària, organitzar-la en estructures adequades.
- **Administrador** de la BD: És l'encarregat de vetllar pel bon funcionament del sistema: administrar els usuaris i permisos, protegir la BD d'errades (fent còpies de seguretat, controlant la concurrència, ..), optimitzar el sistema, etc.
- **Programador** de la BD: Estudien els requeriments dels usuaris finals per a fer programes, formularis, ... que possibiliten la feina d'aquests.

En l'altre bloc, usuaris sense coneixements informàtics, també anomenats, "Usuaris finals", podem classificar-los entre:

- **Usuaris Habituals**: També anomenats **paramètrics**, solen fer consultes i actualitzacions constants (sempre les mateixes). Normalment els analistes i programadors els faran els programes o formularis per a fer la seua feina.
- **Usuaris esporàdics**: Utilitzen la BD de tant en tant, i cada vegada per a obtenir una informació diferent. El més normal és que utilitzen un llenguatge de consulta de la BD avançat però senzill d'utilitzar.

5.5. Classificacions dels SGBD

Veurem la classificació respecte uns quants criteris.

- Segons el model de dades utilitzat, tindrem els SGBD **relacionals**, jeràrquics i de xarxa. Els dos últims, encara que fa temps van tenir molta importància i aplicació comercial, han quedat antiquats. Nosaltres ens centrarem en els SGBD Relacionals. Últimament han aparegut els **orientats a objectes**. Intentarem veure alguns aspectes en l'últim tema. Actualment, hi ha algunes noves tendències incipients, gràcies al fenomen Internet (models NoSQL), tot i que la informació en les empreses segueix utilitzant els models clàssics de BD.
- Segons el número d'usuaris, tindrem sistemes **monousuari**, que només atenen un usuari al mateix temps, i els sistemes **multiusuari**, que atenen molts usuaris al mateix temps.
- Segons el nombre de llocs on està ubicat el sistema, tindrem els **centralitzats**, en els quals tant el SGBD com les dades estan en un únic ordinador central

(encara que es puga accedir a la BD de forma remota). Per contra estaran els SGBD **distribuïts (SGBDD)**, en els quals el SGBD i les dades estan distribuïts en uns quants llocs connectats per una xarxa. Dins d'aquestos podríem fer una distinció entre els **homogenis**, que utilitzen tots els llocs el mateix software, i els **heterogenis** o **federats** que utilitzen distint software i té cada SGBD particular un cert grau d'autonomia local.