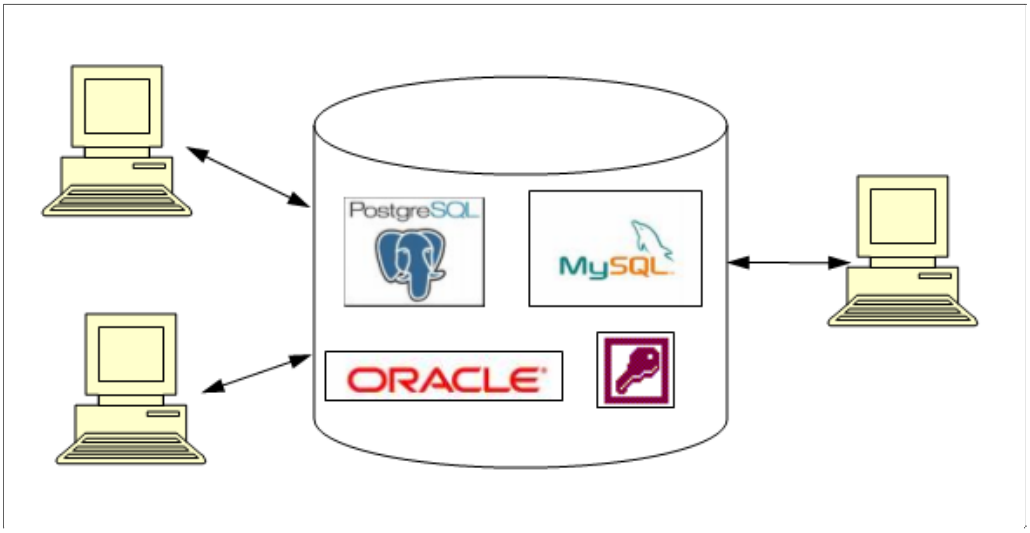
Tema 1: Introducció



Objectius

Objectius

texto

Aquest és el tema d'introducció, i per tant de primer contacte amb la matèria.

Els objectius són:

- Veure la necessitat de millorar els fitxers tradicionals
- Tenir clar els conceptes de Base de Dades i de Sistema Gestor de Base de Dades.
- Veure els avantatges i també els inconvenients
- Entendre el concepte de Model de Dades i conèixer els distints tipus.
- Conèixer l'estandarització de l'arquitectura a 3 nivells.
- Saber de l'existència de distints llenguatges

---

1. 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, Alícia          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, Alícia          C/ Amunt,15                 964205080 10/05/1964 2500
```

Suposem que l'empresa decideix continuar ampliant la informació, i ara vol mantenir informació dels distints 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.

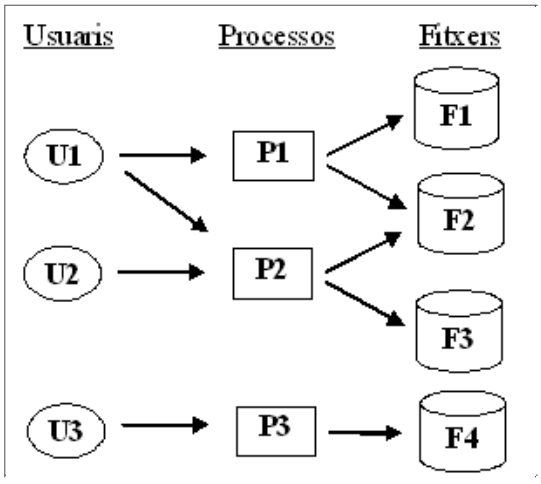
```
...
struct TEmpleat2
{ char dni[10];
    char nom[30];
    char telefon[10];
    char departament[20]
} v_empleat2;

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, Alícia          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 siguen

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 puguen fer actualitzacions no autoritzades

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

2. Concepte de Base de Dades

Per a millorar els problemes que se'ns presentaven amb els fitxers tradicionals introduirem el concepte de Base de Dades.

2.1 Definició

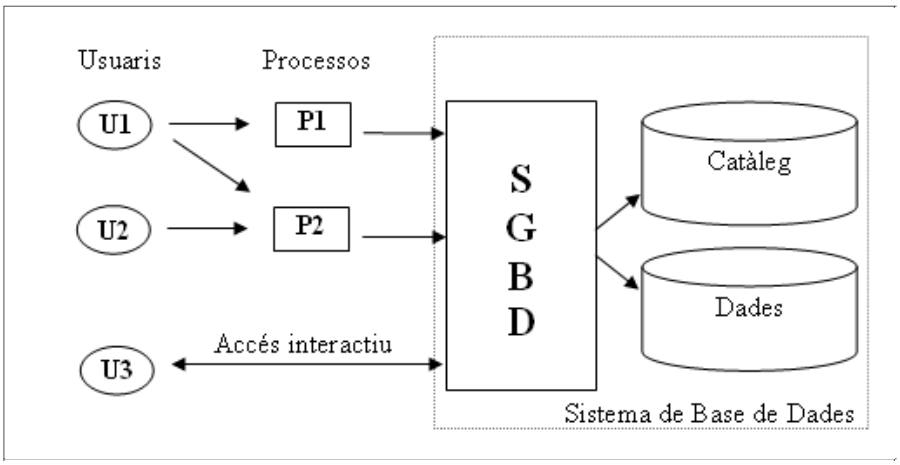
Hi ha moltes definicions de Bases de Dades. Intentarem donar una que no sigue massa pesada.

Una **Base de Dades** és un conjunt de dades relacionades entre elles, integrat en una única estructura global que és independent dels programes que la utilitzen i sense redundàncies innecessàries.

Comentem un poc aquesta definició. El primer comentari és sobre les dades. 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 interessen les dades.

Per la pròpia definició, la Base de Dades es podria crear i mantenir manualment (pensem en les fitxes tradicionals d'una biblioteca). Es podria crear i mantenir també per mig d'un grup de programes d'aplicació realitzats específicament per aquesta feina, però el més normal, i el que ens interessa a nosaltres és que açò es faça per mig d'un Sistema Gestor de Bases de Dades.

Un **Sistema Gestor de Bases de Dades** (**SGBD**; en anglès *DataBase Management System: DBMS*) és un conjunt de programes que permet als usuaris crear i mantenir Bases de Dades. Així la Base de Dades són les nostres dades (organitzades) i el SGBD serien productes com ACCESS o ORACLE que ens permeten crear i mantenir aquesta Base de Dades.



Per la figura ja veiem que la Base de Dades s'organitza en dos grans blocs, un és el **catàleg** o **diccionari de dades**, on està l'estructura i tipus de les dades. En l'altre estan les pròpies dades. Per tant veiem que l'estructura de les dades va amb la pròpia Base de Dades.

Totes les peticions d'accés a les dades no es fan directament, sinó que es demanen al SGBD, que és qui s'encarrega d'accedir a elles. L'accés es pot fer per mig de processos (programes, formularis, ...) o bé de forma interactiva (per exemple amb consultes SQL). Alguns autors anomenen al conjunt de la Base de Dades i el SGBD com **Sistema de Base de Dades**.

2.2 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 B.D. (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 B.D., la modificació de l'estructura (afegint algun camp o modificant-ne algun) no afecta als programes que puguen 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 B.D. totes les dades es perd la visió particular d'un usuari (que en fitxers tradicionals són el fitxers individuals de cadascun d'ells). Però no és així, ja que en les B.D. es poden definir vistes, que són la part de les dades que li interessen 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 B.D. 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 B.D.

2.3 Usuaris de la BD

Els múltiples usuaris que poden utilitzar la B.D. són de distints tipus. Anem a fer una classificació d'aquestos. 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).

A. Usuaris informàtics.

A.1 Dissenyadors.

Encarregats de dissenyar la B.D.: triar el tipus d'informació necessària, organitzar-la en estructures adequades.

A.2 Administradors.

És l'encarregat de vetllar pel bon funcionament del sistema: administrar els usuaris i permisos, protegir la B.D. d'errades (fent còpies de seguretat, ..), optimitzar el sistema, etc. És a dir molta feina.

A.3 Analistes i programadors.

Estudien els requeriments dels usuaris finals per a fer programes, formularis, ... que possibiliten la feina d'aquestos.

B. Usuaris finals.

Són les persones que necessiten l'accés a la B.D. per a introduir dades, actualitzar-les, consultar-les, generar informes, ... Entre ells distingim:

B.1 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.

B.2 Esporàdics.

Utilitzen la B.D. 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 B.D. avançat però senzill d'utilitzar.

3. 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 tinguen 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'introduesca o es modifique 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.

4. Models de dades

Per a poder definir l'estructura de la B.D. 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 B.D. 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 B.D. en un moment determinat (a la qual cosa alguns autors anomenen **estat de la B.D.** 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.

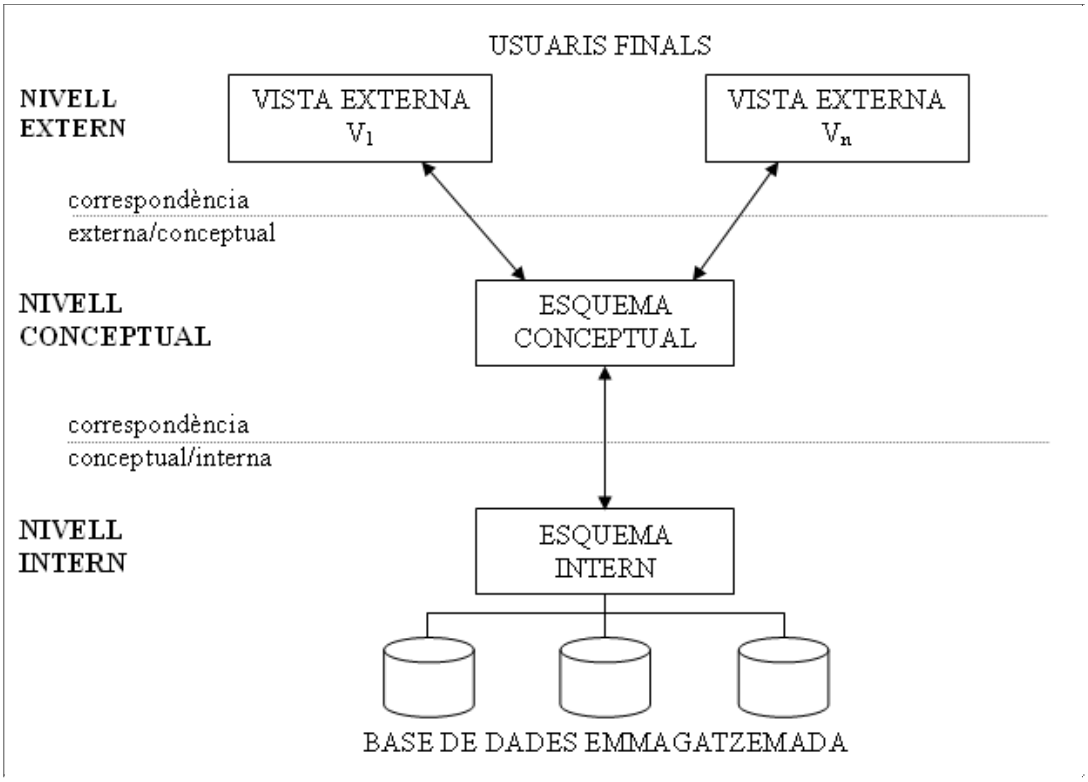
- **Models d'alt nivell** o **conceptuals**: disposen de conceptes molt propers a la manera de percebre les dades les persones. El més conegut és el **Model Entitat-Relació**. Uns altres serien els **orientats a objectes**. Serviran per a poder dissenyar la B.D. a partir de les percepcions del món real. Nosaltres veurem el Model Entitat-Relació (M. E/R). El model orientat a objectes està utilitzant-se prou en l'actualitat, bé de forma pura o bé barrejat amb el Model Relacional. Al final de curs veurem un exemple.

- **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 **Codasyl**) i el **Relacional**, que és el que nosaltres veurem en profunditat.

5. 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 B.D. 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 B.D. 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 B.D. 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.

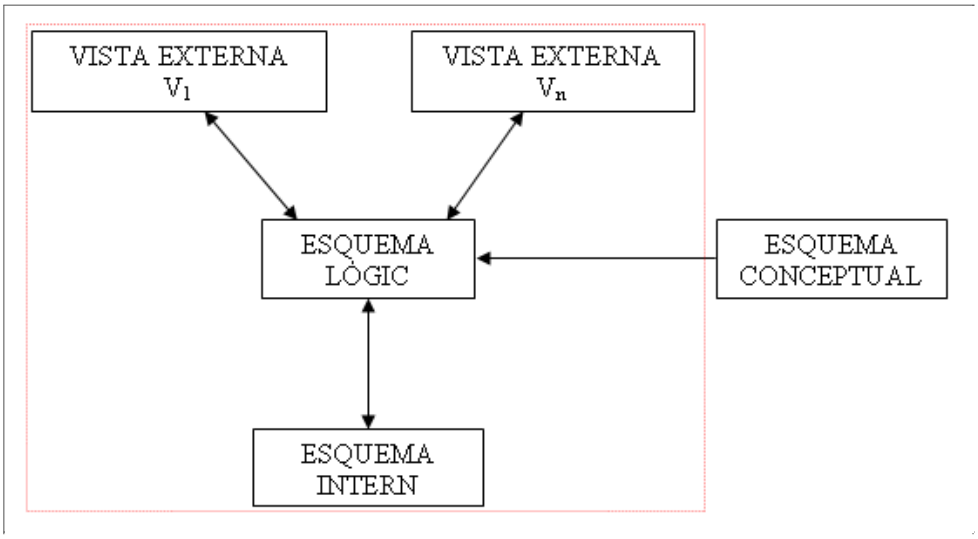
El procés de transformar sol·licituds i resultats d'un nivell a un altre s'anomena correspondència o transformació (*mapping*). Evidentment consumiran temps, però facilitaran el que es pretenia:

- **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 B.D. 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 B.D.



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.

6. Llenguatges del SGBD

Tant per a la construcció dels distints esquemes (administradors) com per a la posterior utilització per a introduir, modificar, eliminar i consultar dades (tots els usuaris) se li hauran de donar ordres al SGBD. Per tant el SGBD ha de proporcionar uns llenguatges als usuaris per a poder realitzar aquestes accions. Dos seran els grups de llenguatges:

- Llenguatges de definició. Serviran per a construir els distints esquemes. Podrem distingir:

  - **Llenguatge de definició de dades** (***DDL**: data definition language*). Serviran per a construir l'esquema lògic.

  - **Llenguatge de definició d'emmagatzematge** (***SDL**: storage definition language*), per a l'esquema físic.

  - **Llenguatge de definició de vistes** (***VDL**: view definition language*), per a les vistes

  En la pràctica a tots aquestos (que com es comentarà més avant solen anar junts) se'ls anomena ***DDL***

- **Llenguatges de manipulació de dades** (***DML**: data manipulation language*). Serviran per a manipular les dades, és a dir fer consultes, insercions, modificacions i eliminacions.

En els SGBD actuals no s'acostuma a separar els llenguatges, sinó que s'utilitza un ampli llenguatge que combina tots els aspectes i permet fer-ho tot. Així per exemple, el llenguatge més utilitzat, el SQL, compta amb característiques de DML (SELECT), i de DDL en tots els nivells (CREATE TABLE esquema lògic, CREATE INDEX esquema físic, CREATE VIEW esquema extern).

Hi ha dues maneres d'aplicar les sentències dels llenguatges.

- De forma interactiva: s'escriu una sentència, s'executa, després una altra, ... Obtenen conjunts de resultats, per això s'anomenen **de conjunt en conjunt**.
- Immergits en un altre llenguatge de propòsit general, anomenat llenguatge ***amfitrió***, on entre els procediment s'inclouen les sentències del DML, anomenat llenguatge ***hoste*** o ***sublenguatge***. Normalment s'obtenen registres individuals de la B.D., per la qual cosa s'hauran d'utilitzar bucles per fer tota una consulta. Per això s'anomenen **de registre a registre**.

7. 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.
- 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 B.D. 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.

Autoavaluació

Omplir els espais en blanc

Una [ ] [ ] [ ] és un conjunt de dades relacionades entre elles, integrat en una única estructura global que és independent dels programes que la utilitzen i sense redundàncies innecessàries.

Al conjunt de programes que permet als usuaris crear i mantenir l'anterior s'anomena [ ] [ ] [ ] [ ] [ ] [ ]

Envia

Tria la correcta

L'ususuari encarregat de gestionar la Base de Dades, donar d'alta usuaris, fer còpies de seguretat, rep el nom de:

- [ ] Programador
- [ ] Paramètric
- [ ] Administrador
- [ ] Analista

Mostra Realimentació

Preguntes Veritat/Fals

Amb les eines que ens proporciona un **Model de Dades** es pot descriure l'estructura d'una Base de Dades

Veritat ○ Fals ○

Un **Model de Dades** és la part d'un SGBD que permet definir l'estructura de la Base de Dades

Veritat ○ Fals ○

Un exemple de Model d'Implementació és el Model Entitat-Relació

Veritat ○ Fals ○

Un exemple de Model d'alt nivell és el Model Jeràrquic

Veritat ○ Fals ○

El Model Relacional, el més estés en l'actualitat, es pot considerar tant d'alt nivell com d'implementació.

Veritat ○ Fals ○

Elecció múltiple

L'arquitectura a 3 nivells vol dir que:

- ○ la Base de Dades s'ha de construir únicament en un dels tres nivells "estàndar"

- es poden descriure les dades en tres nivells diferents d'abstracció, encara que les dades només existeixen realment en el nivell físic
- tota Base de Dades ha de tenir 3 arquitectes o "constructors"
- cap de les opcions anteriors és correcta

Selecció múltiple

Indica quines afirmacions són correctes.

- [ ] Els SGBD no poden controlar la redundància, per això és convenient que no n'hi haja gens
- [ ] Els SGBD poden oferir diferents interrfícies als usuaris
- [ ] Els SGBD poden "imposar" restriccions a les dades que s'han d'introduir, com per exemple que una nota estiga entre 0 i 10
- [ ] Tots els usuaris de la Base de Dades tenen els mateixos permisos
- [ ] Els SGBD es basen en altres sistemes (com per exemple el Sistema Operatiu) per a fer les còpies de seguretat

Mostra Realimentació

Elecció múltiple

El fet que la definició de les dades estiga incorporada en la mateixa Base de Dades, s'anomena tècnicament que la Base de Dades és

- automàtica
- autòmata
- automòbil
- autodescriptiva
