1r DAM

BASES DE DADES

UNITAT 1. SISTEMES D'EMMAGATZEMATGE DE LA INFORMACIÓ

Profesor: Manuel Botella

Contingut

1. Introducció

Hem vist que necessitem de l'ordinador quan el volum de dades alt i temps de resposta que necessitem és vital.
Exemples hui dia de Bases de dades :

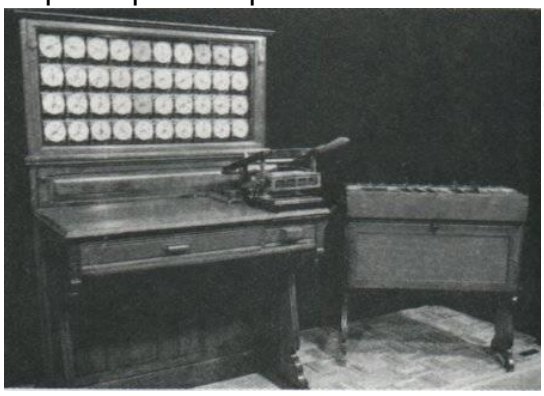
- Comprar en Mercadona
- Ús de la targeta de crèdit
- Reservar Viatges
- Reserva de llibres en una biblioteca
- Contractar una assegurança
- Matricular-se

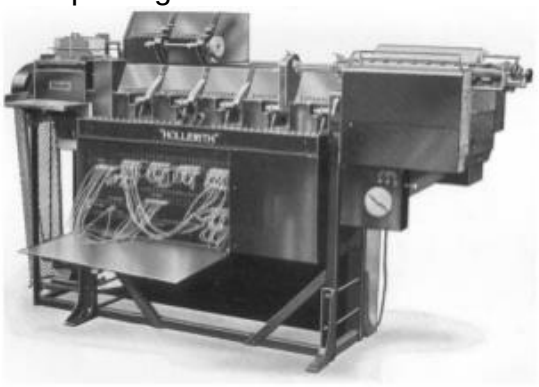
1.1 Història i Evolució Bases de dades

L'ús de sistemes de bases de dades automatitzades, es va desenrotllar a partir de la necessitat d'emmagatzemar grans quantitats de dades, per a la seua posterior consulta, produïdes per les noves indústries que creaven gran quantitat d'informació..

Sistema d'Arxius:.

En 1884 Herman *Hollerit crec la màquina perforadora, amb la qual va aconseguir baixar de 7 anys a 2 anys el temps a censar a la població d'USA. Composta per una perforadora i una lectora que llegia orificis.





Dècada de 1950 es va inventar les targetes magnètiques. Es va començar a automatitzar la informació de les nòmines, comencen a usar-se Bases de dades basades en Sistema d'Arxius seqüencials. Estes cintes només es podien llegir seqüencial i ordenadament

Dècada de 1960 L'ús dels discos en eixe moment va ser un avançament molt efectiu, ja que per mitjà d'este suport es podia consultar la informació directament usant fitxers aleatoris i indexats, on l'accés ja podia ser aleatori, el sistema de fitxers era el sistema més comú d'emmagatzematge de dades , això ajude a estalviar temps. No era necessari saber exactament on estaven les dades en els discos.

SGBD

A la fi dels 60, els programes i les dades eren cada vegada més complexos i requerien exigències complexes apareixen els SGBD de Primera generació c*on estructures de dades com a llistes i arbres.

- Les Bases de dades Jeràrquica
- Les Base de dades de Xarxa

L'aliança d'IBM i *American Airlines per a desenrotllar *SABRE, un sistema operatiu que manejava les reserves de vols, transaccions i informacions sobre els passatgers de la companyia *American Airlines
L'aparició de *CODASYL primer SGBD en Xarxa.

En els 70 Edgar Frank *Codd, científic informàtic engonals conegut per les seues aportacions a la teoria de bases de dades relacionals, va definir el famós model relacional. Que és el que es va dir Segona Generació.

En els 80 es desenrotlla el SQL (*Structured *Query *Language) o cosa que és el mateix un llenguatge de consultes o llenguatge declaratiu d'accés a bases de dades relacionals que permet efectuar consultes amb la finalitat de recuperar informació d'interés d'una base de dades i fer canvis sobre la base de dades de manera senzilla

En els 90 WWW (Internet) facilitarà la consulta a bases de dades . S'agreguen funcionalitats al SQL (agrupació, *order *by….) i aparició del XML. Actualment una tercera generació de Bases de dades postgres basat en l'enfocament relacional estés i els SGBD Objecte Relacional i Orientades a Objectes (amb poc ús comercial encara)

2. Sistema brostit en a*rchivos

Com hem dit tota base de dades és un conjunt de dades estructurades i emmagatzemades en un mitjà. En l'ordinador estes dades al final estan emmagatzemats en fitxers.

Abans d'aparéixer els Sistemes Gestors, els primers Sistemes de Bases de dades van usar sistemes de fitxers dependents del S.O..

Vamos a veure què és un arxiu i que tipus existixen. Els arxius s*on estructures d'informació que creguen els S.O. per a poder emmagatzemar dades. Solen tindre: **<nom>.<extensió>**

2.1 Tipus d'Arxius

2.1.1 Segons el seu contingut:

- ☑ Arxive p*lano: Fitxers de text o fitxer ASCII.
  - Configuració: .*ini .*inf .*conf
  - Codi font: .*sql .c .*java
  - Pàgina Web: .HTML .*php . *asp .*xml
  - Enriquits: .*rtf .*ps .*tex
- ☑ Arxive b*inario: No són de text, i requerixen un format per a interpretar-lo
  - Imatge: *jpg .*gif .*bmp
  - Vídeo: .*mpg .*mov .*avi
  - Comprimits: .*zip .*gz .*rar .*tar
  - *Ejectubles: .*exe .*com .*cgi

- Processadors de text: .*doc .*odt

Exercici 1

a. Comprova quant ocupa el teu nom en on fitxer pla i seguidament prova en un fitxer binari. Per quina hai diferència de grandària?

Sol:

b. El japonés té més de 256 lletres i requerix caràcters *unicode. Veu a la pàgina http://www.unicode.org/charts i descarrega't la t*abla de codis *LATIN (alfabet llatí) i la t*abla de codis *Katakana (Alfabet japonés)

Exercici 2

Descriu de quin tipus són estos fitxers i per a què servixen:

- *Galactica-*lx01.*avi
- com_*jnstalar.txt
- DSCN0776.*JPG
- eclipsi.*exe
- eclipsi.*ini
- *img_*xp_*sp3.*iso
- llicencia_*windows.txt
- microxip.*flv
- *notice.HTML
- *partlogic-O.69-*iso.*zip
- *prac3.2.*sql
- *tem1.pdf

2.1.2 Segons el seu accés, els mètodes d'accés a les dades poden ser:

☑ Arxius seqüencials: són els p*rimeros a aparéixer. El medi físic d'emmagatzematge eren tambors de cinta magnètica de diàmetre comparable als discos de vinil (LP). Les unitats de cinta informàtiques eren com uns cassets gegants on s'emmagatzemaven dades digitals 0 i 1 en orde seqüencial i on c*ada registre tenia una sèrie de camps, que es gravaven en la cinta successivament, de forma ordenada i era m*uy útil per a imprimir etiquetes i enviar circulars (en aquella època, de correu per exemple).

Característiques:

- Lectura ordenada obligatòria: passar per tots els registres anteriors per a llegir un intermedi
- No permet el retrocés: la lectura només es realitza cap avant
- Monousuaris: només accés d'un usuari alhora
- Estructura rígida de camps: tots els registres han d'aparéixer en orde, no podem variar orde en els camps
- Lectures parcials però escriptures totals: l'escriptura comporta l'escriptura total de tota la seqüència completa de registres
- Marca final d'arxiu - *EOF
- Esborrament: cal reescriure tot menys el registre a esborrar (o marcar-lo)
- No deixa buits
- Registres de longitud variable. (Els tipus d'accés aleatoris ho arreglen fent-ho fix)
- Possibilitat d'usar marca de sincronisme
- Contingut llegible per un processador de textos

Exercici
Què vol dir seqüencial? Com es podria crear un arxiu de clients? Posa un exemple

Exercici
Per què hi ha inconvenients amb la impressió?

☑ Arxius d'accés aleatori: asemblen amb l'arribada dels disquets i discos durs. Podem accedir directament a la posició, expressada per un número que indiquem de l'arxiu , ia no fa falta recórrer-se tots els anteriors p*ero cal calcular-lo i traduir-lo.

Posició = *NumRegistro * *LongRegistro

Cada llenguatge de programació disposarà de la seua pròpia funció per al posicionament del cursor de lectura/escriptura per a realitzar l'accés aleatori.
Característiques:

- Posicionament immediat.
- Registres de longitud fixa
- Obertura lectura i/o escriptura
- Ús concurrent (multiusuari)
- Esborrament de registre mitjançant zeros o Marcat
- Problema que deixa molts buits
- Dimensionament màxim en ser creats

Exemple
1. Posem per cas que tenim un registre com este codificat en *ANSI (1 byte per caràcter):

- Nom: 80 caràcters *ANSI.
- Adreça: 100 caràcters *ANSI.
- Població: 50 caràcters *ANSI.

La seua longitud serà de 230 bytes. Això implica que el primer registre es trobarà en la posició 0, el segon registre es trobarà en la posició 230, el tercer estarà en la posició 460 i així successivament.

2. Si en canvi codifiquem els caràcters en *Unicode (*UTF-16) de manera que la seua longitud és de 16 bits per caràcter, això és 2 bytes, el registre quedarà així:

- Nom: 80 caràcters *UTF-16.
- Adreça: 100 caràcters *UTF-16.
- Població: 50 caràcters *UTF-16.

La seua longitud serà de 460 bytes per a un sol registre, ja que cada caràcter ocupa 2 bytes.

☑ Arxius indexats: són bàsicament arxius d'accés aleatori als quals s'afig una utilitat per a accedir al registre desitjat a través d'una clau, és a dir, o*sa un índex de l'arxiu per a suportar els accessos i és el índex el que proveïx una capacitat de busca per a arribar ràpidament a les proximitats d'un registre desitjat. Poden buscar un client per nom, *dni, telèfon,… Manté en la RAM l'estructura d'índexs de forma ordenada. S'utilitzava per

a localitzar registres per l'estructura d'informació per arbre binari o d'índexs on L<date<R

Clau	
Núm. registre	
L	R

Característiques

- Són bàsicament arxius d'accés aleatori amb utilitat (estructura o taula d'índexs, per a no calcular) per a accedir directament al registre buscat.
- L'estructura guarda índexs a les posicions dels registres.
- Arbre de busca de claus

Exercici
En quina codificació estan escrits els codis?
Visual *Basic 6
Java
C#
C++

☑ Arxius *hash: no els analitzarem perquè a penes s'utilitzen a causa del baix rendiment i complexitat.

2.2. Funcionament d'un Sistema d'Arxius

Abans d'aparéixer els SGBD (dècada dels setanta), la informació es tractava i es gestionava utilitzant els típics sistemes de gestió d'arxius que anaven suportats sobre un sistema operatiu i c*onsistían en un conjunt de programes d'aplicació que realitzaven diversos servicis als seus usuaris.

Cada aplicació tenia on conjunt d'arxius de dades i on conjunt de programes (altres arxius) que gestionaven estos arxius de dades, i c*ada programa gestionava els seus propis fitxers de dades.

Es van usar molt en nòmines, control de comandes i manteniment de productes.

Funcionament

Cada funcionalitat (un o diversos programes) utilitzava fitxers de moviments per a actualitzar (creant una còpia nova del fitxer mestre (redundància)), i/o per a consultar un o dos fitxers mestres o, excepcionalment, més de dos.

Cada programa de l'aplicació tractava com a màxim un fitxer mestre, que solia estar sobre cinta magnètica i, en conseqüència, es treballava amb accés seqüencial.

Cada vegada que se li volia afegir una nova funcionalitat que requeria l'ús d'alguns de les dades que ja existien i d'altres nous, es dissenyava un fitxer nou amb totes les dades necessàries (alguna cosa que provocava més redundància) per a evitar que els programes hagueren de llegir molts fitxers.

Característiques dels programes

Si l'estructura de les dades dels arxius canvia, tots els programes que

els manegen s'han de modificar.

Per e*jemplo, un arxiu de dades d'alumnes, amb una estructura o registre ja definit; si s'incorporen elements o camps a l'estructura de l'arxiu, els programes que utilitzen eixe arxiu s'han de modificar per a tractar eixos nous elements.

Això és pel fet que la definició de les dades es troba codificada dins dels programes d'aplicació en lloc d'emmagatzemar-se de manera independent.

El control de l'accés i la manipulació de les dades venen impost pels programes d'aplicació .

Tenien diferents formats i llenguatges, cada programador usava el que volia.

Exercici

1. Avantatges de les *BD sobre sistemes de gestió de dades basades en fitxers
2. Exemple d'un sistema de gestió basat en fitxers i un altre de *BD
3. Esquema entre diferències de seguretat i integritat

Els problemes apareixen a mesura que creixen les necessitats d'informació doncs a*umenta el nombre d'usuaris , augmenten les *nec*esidades dels programes ja que es van creant nous programes per a accedir a les dades i integrar la informació en els existents i hi ha major n*ecesidad d'interconnectar estos programes de diferents departaments.

Exemple:

- Departament de Vendes
  - Productes
  - Clients
  - Venedors
- Departament de Contractes
  - Productes
  - Clients
- Departament de RH
  - Venedors

2.3. Inconvenients d'un sistema de gestió d'arxius

- **Redundància i inconsistència de les dades**, es produïx perquè els arxius són creats per diferents programes i van canviant al llarg del temps, és a dir , poden tindre diferents formats i les dades poden estar duplicats en diversos llocs. Per exemple , el telèfon d'un alumne pot aparéixer en més d'un arxiu. La redundància augmenta els costos d'emmagatzematge i accés, i porta amb si la inconsistència de les dades: les còpies de les mateixes dades no coincidixen per aparéixer en diversos arxius.
- **Dependència de les dades física-lògica**, o cosa que és el mateix , l'estructura física de les dades (definició d'arxius i registres) es troba codificada en els programes d'aplicació. Qualsevol canvi en eixa estructura implica el programador identificar, modificar i provar tots els programes que manipulen eixos arxius.

- **Dificultat per a tindre accés a les dades i ampliacions:**, proliferació de programes, és a dir , cada vegada que es necessite una consulta que no va ser prevista en l'inici implica la necessitat de codificar el programa d'aplicació necessari o buscar-lo manualment. És a dir , no permeten recuperar les dades necessàries d'una forma convenient i eficient.
- **Separació i aïllament de les dades**, és a dir , en estar repartits en diversos arxius, i tindre diferents formats, és difícil escriure nous programes que asseguren la manipulació de les dades correctes. Abans s'haurien de sincronitzar tots els arxius perquè les dades coincidiren.
- **Problemes de *atomicidad.** És crucial assegurar que una vegada una fallada ha ocorregut i s'ha detectat, les dades es restauren a l'estat de consistència anterior a la fallada, o ocorre tot o no ocorre res. És difícil assegurar esta propietat en un sistema d'arxius..
- **Dificultat per a l'accés concurrent**, perquè en un sistema de gestió d'arxius és complicat que els usuaris actualitzen les dades simultàniament. Les actualitzacions concurrents poden donar per resultat dades inconsistents, ja que es pot accedir a les dades per mitjà de diversos programes d'aplicació..
- **Dependència de l'estructura de l'arxiu amb el llenguatge de programació**, perquè l'estructura es definix dins dels programes. Això implica que els formats dels arxius siguen incompatibles. La incompatibilitat entre arxius generats per diferents llenguatges fa que les dades siguen difícils de processar.
- **Problemes en la seguretat de les dades**. Resulta difícil implantar restriccions de seguretat perquè les aplicacions es van afegint al sistema segons es van necessitant.
- **Problemes d'integritat de dades**, és a dir , els valors emmagatzemats en els arxius han de complir amb restriccions de consistència. Per exemple , no es pot inserir una nota d'un alumne en una assignatura si prèviament eixa assignatura no està creada. Un altre exemple, les unitats en magatzem d'un producte determinat no han de ser inferiors a una quantitat. Això implica afegir gran nombre de línies de codi en els programes. El problema es complica quan existixen restriccions que impliquen diverses dades en diferents arxius.

Solució

Sorgix així la idea de separar les dades contingudes en els arxius dels programes que els manipulen, és a dir , que es puga modificar l'estructura de les dades dels arxius sense que per això s'hagen de modificar els programes amb els quals treballen. Es tracta d'estructurar i organitzar les dades de manera que es puga accedir a ells amb independència dels programes que els gestionen i el més important, que TOTES les dades estiguen en un únic repositori.

Tots estos inconvenients fan possible el foment, desenrotllament i aparició dels SGBD.

3. Sistemes Gestors de Bases de dades (SGBD)

L'objectiu primordial d'un gestor és proporcionar eficiència i seguretat a l'hora d'extraure o emmagatzemar informació en les *BD.
Els sistemes gestors de BBDD estan dissenyats per a gestionar grans blocs d'informació, que implica:

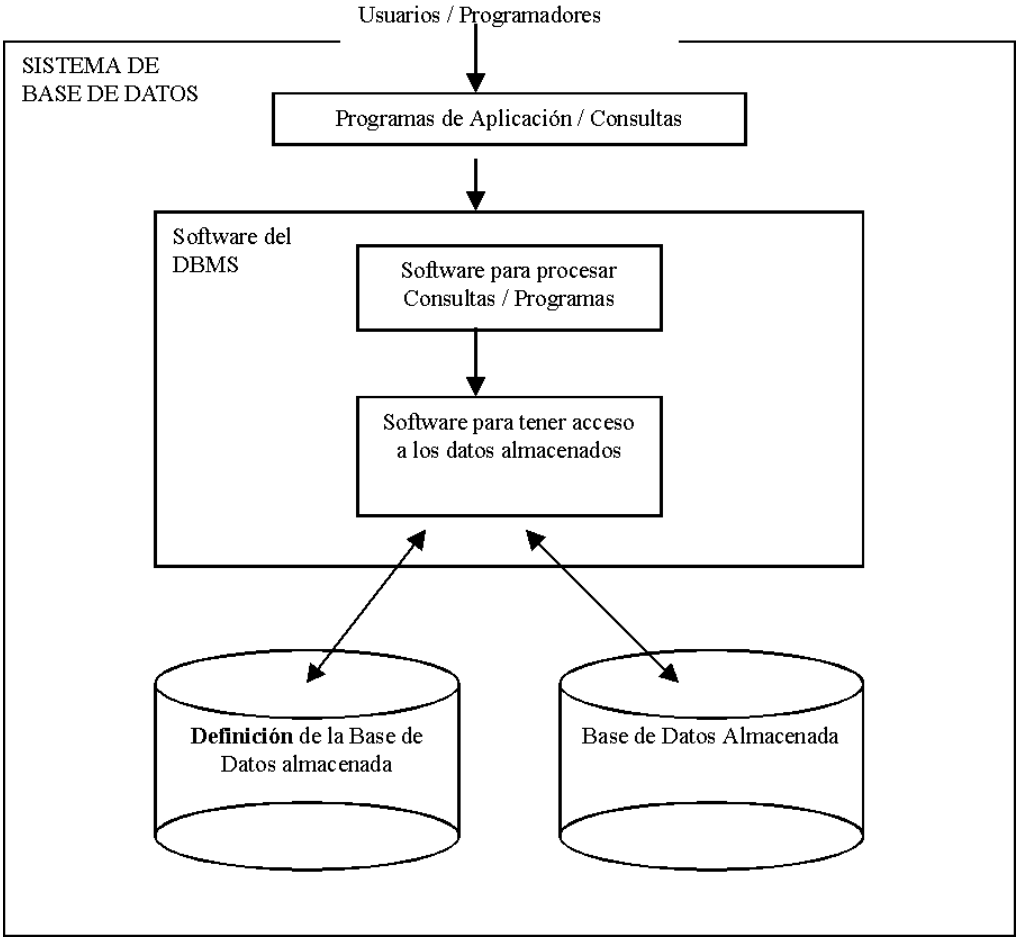
- La definició d'estructures per a l'emmagatzematge..
- Mecanismes per a la manipulació de la informació

*3.1 Definició *BD i SGBD*

***BD**: c*onjunto de dades relacionades i organitzades amb una certa estructura. La informació es referix al mateix tipus de persona o objecte i es mostra estructurat en files i columnes (taula)

SGBD: a*plicación que permet definir, crear i mantindre la *BD. És la i*nterfaz entre l'Usuari i la *BD de manera que Sistema *BD=*BD+SGBD

Un esquema possible de SGBD pot ser el següent:



3.2. Objectius/Característiques d'un SGBD

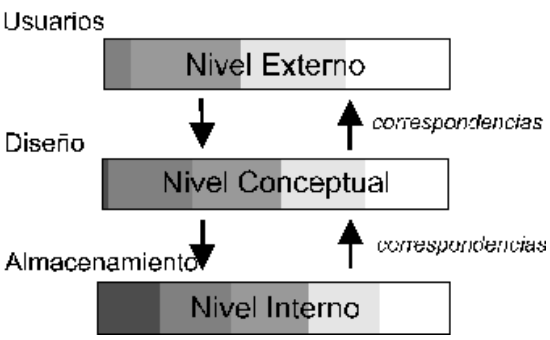
- **Independència física i lògica.** Estructures físiques independents de la lògica i viceversa, accés a dades sense necessitat de conéixer l'estructura interna.
- **Eficaç accés a les dades:** Sense necessitat de conéixer com estan guardats, facilitat de manipulació (si està autoritzat) sense necessitat de ser Informàtic. (Ferramentes gràfiques)
- **Mantindre la integritat i consistència** Garantir que la transacció es faça o es rebutge (*Atomicidad)
- **Permeten la concurrència.** Accés compartit a la *BD, controlant la interacció entre usuaris concurrents. Que no s'assabenten que tots dos estan alhora .
- **Mecanismes de suport i recuperació** per a restablir la informació en cas de fallades en el sistema
- **Coherència de dades: Complir restriccions.** Regles de la realitat
- **Redundància controlada**
- **Administració centralitzada. Ferramentes d'administració..**
- **Seguretat de les dades:** Accés controlat a les dades de la *BD mitjançant mecanismes de seguretat d'accés als usuaris.
- **Ús de transaccions**

Concepte de Transacció
Una transacció és un conjunt d'instruccions que realitzen una acció i té les característiques *ACID (**A***tomicity, **C***onsistency, **I***solation *and **D***urability):

- ***Atomicidad:** és la propietat que assegura que l'operació s'ha realitzat o no, i per tant davant una fallada del sistema no pot quedar a mig fer .
- **Consistència:** *Integritat*. És la propietat que assegura que només es comença allò que es pot acabar. Per tant s'executen aquelles operacions que no trencaran les regles i directrius d'integritat de la base de dades .
- **Aïllament:** és la propietat que assegura que una operació no pot afectar a unes altres. Això assegura que la realització de dos transaccions sobre la mateixa informació siguen independents i no generen cap mena d'error..
- **Durabilitat:** és la propietat que assegura que una vegada realitzada l'operació, esta persistirà i no es podrà desfer encara que falle el sistema.

3.3. Arquitectura SGBD

Es basa en la *ANSI-*SPARC que dividix en 3 nivells:



- **Nivell Intern:** és el més pròxim a l'emmagatzematge físic, és a dir , tal com estan emmagatzemats les

dades en l'ordinador. (Arxius, tipus registre, longitud…). En una *BD les taules s'emmagatzemen en arxius de dades de la *BD. Si hi ha claus, es creen índexs per a accedir a les dades, tot això contingut en el disc dur , en una pista i en un sector, que només el SGBD coneix. Davant una petició, sap a quina pista, a quin sector, a quin arxiu de dades i a quins índexs accedir

- **Nivell Conceptual o lògic**: Esquema conceptual. (Entitats, relacions, atributs…), definix quines dades hi ha emmagatzemats i com es relacionen. Definició de totes les taules, columnes, restriccions, claus i relacions. En este exemple, disposem de tres taules que estan relacionades:
  Taula ALUMNES.
  Columnes: *NMatrícula, Nom, Curs, Direcció, Població. **Clau**: *NMatrícula.
  A més té una relació amb NOTES, perquè un alumne pot tindre notes en diverses assignatures.
  Taula ASSIGNATURES.
  Columnes: Codi, Nom d'assignatura..
  Clau: Codi.
  Està relacionada amb NOTES, perquè per a una assignatura hi ha diverses notes, tantes com alumnes la cursen.
  Taula NOTES.
  Columnes: *NMatrícula, Codi, Nota.
  Està relacionada amb ALUMNES i ASSIGNATURES, perquè un alumne té notes en diverses assignatures, i d'una assignatura existixen diverses notes, tantes com alumnes.

  Podem representar les relacions de les taules en el nivell lògic com es mostra en la Figura:

- **Nivell Extern**: És el més pròxim als usuaris perquè la visió parcial de les taules de la *BD es realitza segons l'usuari . Vistes de parts de la *BD, bé per a restringir o simplificar. Per exemple , la vista que es mostra en la Taula, obté el llistat de notes d'alumnes amb les següents dades: Curs, Nom, Nom d'assignatura i Nota.

Curso	Nombre	Nombre de asignatura	Nota
1	Ana	Programación en lenguajes estructurados	6
1	Ana	Sistemas informáticos multiusuario y en red	8
2	Rosa	Desa. de aplic. en entornos de 4.ª Generación y H. Case	5
2	Juan	Desa. de aplic. en entornos de 4.ª Generación y H. Case	7
1	Alicia	Programación en lenguajes estructurados	5
1	Alicia	Sistemas informáticos multiusuario y en red	4

Passos e*ntre nivells

1. L'Usuari sol·licita unes dades i crea una consulta
2. El SGBD verifica i accepta l'esquema extern per a eixe usuari.
3. Transforma la sol·licitud a l'esquema conceptual
4. Verifica i accepta l'esquema conceptual
5. Transforma la sol·licitud a l'esquema Intern
6. Selecciona l'o les taules implicades en la consulta i l'executa.
7. Transforma de l'esquema intern al conceptual, del conceptual a l'extern.
8. Finalment, l'usuari veu les dades sol·licitades

Amb esta mena d'arquitectura s'aconseguix:

- **Independència lògica**: la capacitat de modificar l'esquema conceptual sense haver d'alterar els esquemes externs ni els programes d'aplicació. Es podrà modificar l'esquema conceptual per a ampliar la *BD o per a reduir-la, per exemple , si s'elimina una entitat, els esquemes externs que no es referisquen a ella no es veuran afectats.
- **Independència física**: la capacitat de modificar l'esquema intern sense haver d'alterar ni l'esquema conceptual, ni els externs. Per exemple , es poden reorganitzar els arxius físics amb la finalitat de millorar el rendiment de les operacions de consulta o d'actualització, o es poden afegir nous arxius de dades perquè els que hi havia s'han omplit. La independència física és més fàcil d'aconseguir que la lògica, perquè es referix a la separació entre les aplicacions i les estructures físiques d'emmagatzematge..

Les correspondències entre nivells requerix ampliar el diccionari de dades

Exercici

Dis-me dos exemples on separes en 3 nivells.

*Ej: Un fitxer:

- Nivell intern: Són bytes (Conjunt d'1 i 0) en posicions de memòria del disc dur .
- Nivell c*onceptual: És una estructura on guardes dades d'algun tipus (imatge, text, vídeo…)
- Nivell e*xterno: Una icona que pots punxar amb el cursor del ratolí..

3.4. Funcions del SGBD.

La funció principal d'un SGBD és p*ermitir als usuaris les 4 operacions fonamentals com són la cregambal i inserció, c*onsulta, a*ctualización i b*orrado.
Té igualment tres objectius, *proporcion*ar als usuaris una visió abstracta de les dades (amagar com s'emmagatzemen i mantenen les dades), *v*elocidad amb bon temps de resposta i seguretat (consistència), és a dir, que siga veritat el que resulta.

3.4.1. Funció de definició

Utilitza el llenguatge *DDL i p*ermite especificar els e*lementos de les dades que integren la *BD, la seua estructura, les relacions entre eixos elements, les r*eglas d'integritat , els *controls d'acte r*ización i les c*aracterísticas físiques

- ☑ Llenguatge SQL de definició
- ☑ Nivell Intern:
  - ➢ Espai físic
  - ➢ Longitud de camps
  - ➢ Mode de representació
  - ➢ Camins d'accés, punters, índexs
- ☑ Nivell Conceptual i Extern
  - ➢ Definició d'entitats, atributs i relacions
  - ➢ Autorització d'accessos.
- ☑ Tot s'emmagatzema en el Diccionari de dades

3.4.2. FunciónManipulació

Utilitza el llenguatge *DML i p*ermite buscar, afegir, suprimir o modificar les dades d'una *BD sempre d'acord amb les especificacions i restriccions de seguretat.

- ☑ Nivell extern: Vistes a usuaris.
- ☑ Nivell Conceptual: Abstracció de la part interna, facilitat d'ús amb ferramentes gràfiques (*Access). Vistes a programadors
- ☑ Nivell Intern: Algorismes eficients d'accés a les dades.

3.5. Components d'un SGBD

- Components humans. USUARIS.
- Components tècnics
  - ➢ Components Funcionals
    - ▪ Components de processament de consultes
    - ▪ Components de Gestió d'emmagatzematge
  - ➢ Implementació Física del sistema.
    - ▪ Arxius de dades
    - ▪ Diccionari de dades

3.5.1. Usuaris

En un SGBD ens podem trobar amb els següents tipus d'usuaris:

- **Administrador**. Disseny físic
- **Dissenyadors**. Disseny lògic
- **Programadors**. Implementa els programes
- **Usuaris finals**. Utilitzen el resultat final

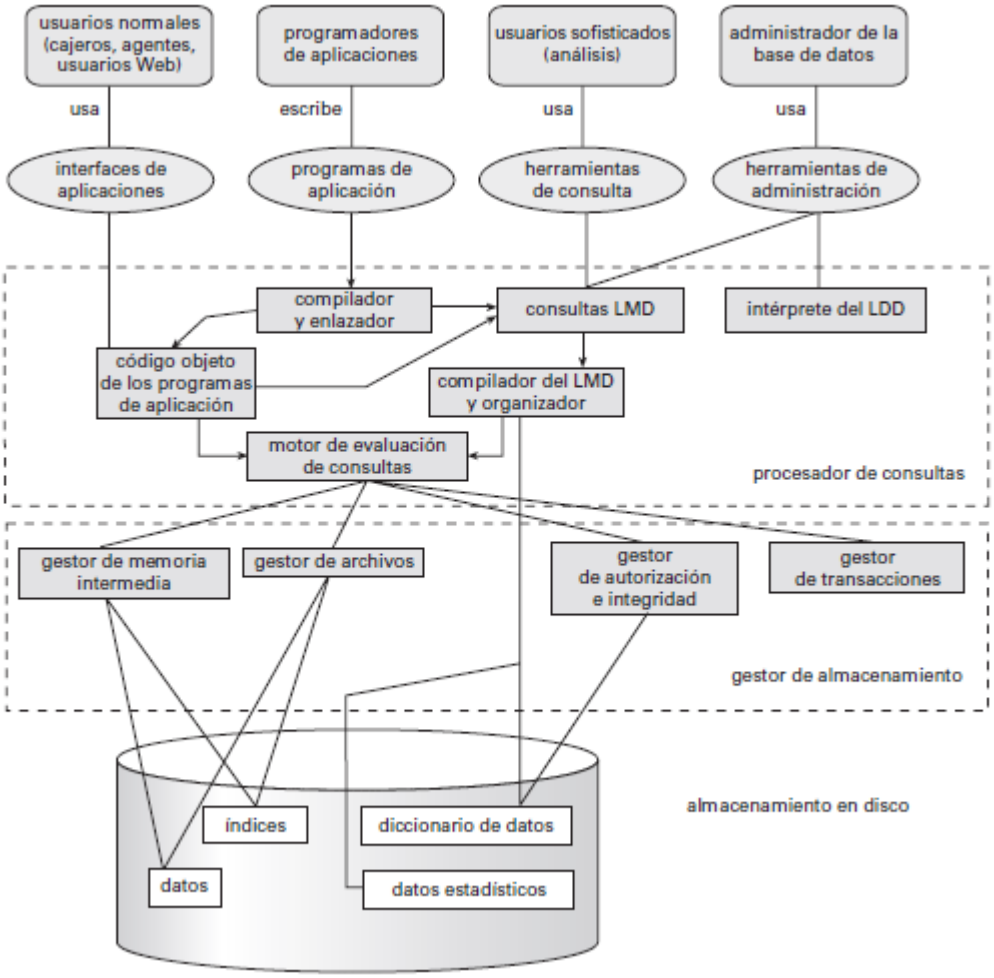
3.5.2. Components tècnics

3.5.2.1. Components f*uncionales

- ☑ Components de processament de consultes
  - Compilador *DML: t*raduce a instruccions que entén el motor
  - precompilador *DML incorporat: t*raduce a llenguatge objecte
  - Intèrpret *DDL: i*nterpreta instruccions *DDL i les introduïx en el diccionari
  - Motor d'avaluació de Consultes: e*jecuta les instruccions generades pel compilador *DML
- ☑ Components de g*estión d'emmagatzematge (Interfície de baix nivell amb els programes d'aplicació).
  - Gestor d'autorització i integritat
  - Gestor de transaccions: asegura la *atomicidad de transaccions.
  - Gestor d'arxius: g*estiona la reserva d'espai.
  - Gestor de memòria intermèdia: trau les dades del disc a memòria principal. Gestiona la caixet.

3.5.2.2. Implementació f*ísica del sistema.

- ☑ Arxius de dades: Emmagatzema la base de dades en si.
- ☑ Diccionari de dades: Emmagatzema metadades sobre l'estructura d'estos .
- ☑ Índexs: Proporciona accés ràpid a les dades.
- ☑ Dades estadístiques: Emmagatzema dades estadístiques sobre les dades. El processador de consultes usa esta informació per a planificar-les.

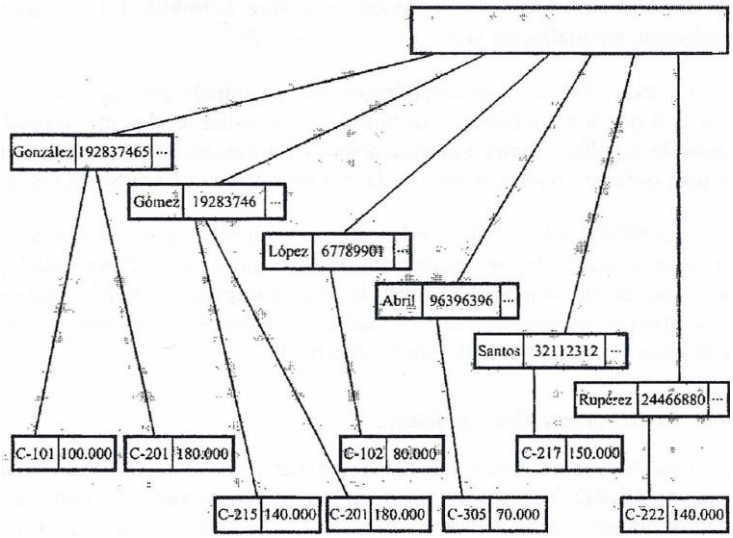


3.6. Tipus de SGBD

Ens podem trobar amb els següents tipus de SGBD s*egún el model (Jeràrquic, xarxa, relacional o *OO) o l'arquitectura (centralitzades o distribuïdes). A continuació descriurem cadascuna d'elles.

3.6.1. Jeràrquic

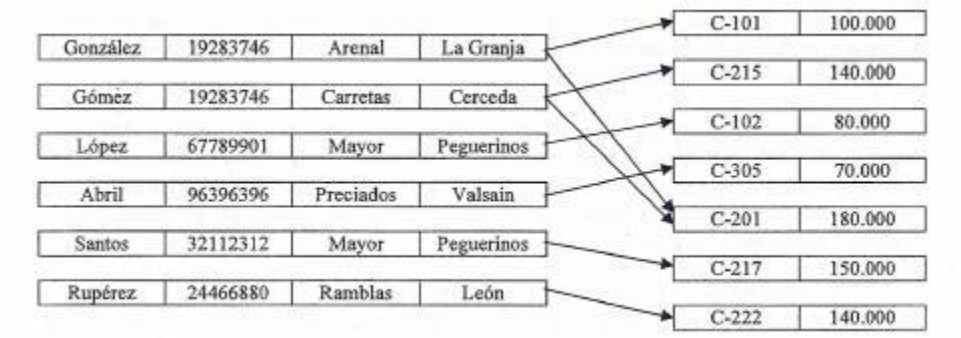
Consistix en un conjunt de registres connectats entre si mitjançant enllaços que són capdavanters.
Té una estructura jeràrquica i és unidireccional de Pare a Fill, mai al revés , la qual cosa dificultava la seua gestió i representació de la realitat.
Tenia els inconvenients principalment que no disposava de relacions M:M i era poc flexible.



3.6.2. Xarxa

Semblança al jeràrquic, conjunt de registres enllaçats mitjançant punter, però esta vegada en forma de graf.

Almenys teòricament sí que disposava de comunicació *bidi*reccional, però en la pràctica era massa costós i ho feia amb diversos punters. Simulava la M:M amb varies 1:M



3.6.3. Relacional

El més estés i comercialment l'únic amb èxit, les dades es descriuen com a relacions que se solen representar com a taules bidimensionals consistents en files i columnes.

Cada fila (*tupla, en terminologia relacional) representa una ocurrència.

Les columnes (atributs) representen propietats de les files.

Cada *tupla s'identifica per una clau primària o identificador.

Els enllaços no són capdavanters són les mateix dades.

Nom	*Dni	Carrer	Ciutat	*Número-compte
G*onzález	19283746	Arenal	La Granja	C-101
Gómez	19283746	Carretes	*Cerceda	C-215
López	67789901	Major	*Peguerinos	C-102
Abril	96396396	Preuats	*Valsain	C-305

González	19283746	Arenal	La Gr*anja	C-201
Sants	32112312	Major	*Peguerinos	C-217.
*Rupérez	24466880	Rambles	León	C-222
Gómez	19283746	Carretes	*Cerceda	C-201

Número-compte	*Saldo
C-101	100.000
C-215	140.000
C-102	80.000
C-305	70.000
C-201	180.000
C-217	150.000
C:Z22--	-

3.6.4. Orientat a Objectes.*UML i XML

És o*na de les novetats més prometedores. Està brostit en una col·lecció d'objectes (com C++) on cada objecte pot tindre els seus propis procediments, única manera d'accedir-los o manipular-los.

Manegen objectes complexos, i són més senzills de mantindre que els relacionals si bé encara no està prou desenrotllat, i comercialment la seua eixida és molt limitada, encara no són molt estables.

3.6.5. Centralitzades. Client/Servidor

L'arquitectura centralitzada és la més clàssica. En ella, el SGBD està implantat en una sola plataforma o ordinador des d'on es gestiona directament, de mode centralitzat, la totalitat dels recursos. És l'arquitectura dels centres de processament de dades tradicionals que se basa en tecnologies senzilles, molt experimentades i de gran robustesa.

3.6.6. Distribuïdes. Homogenis i heterogenis.

En l'arquitectura distribuïda el SGBD i la base de dades no estan associats a un determinat ordinador sinó a una xarxa, els nodes de la qual es repartixen les funcions. És el SGBD distribuït el que s'encarrega de preservar la integritat i coherència de la *BD, externament és vista com centralitzada.

Un sistema és homogeni si el SGBD usat en totes les màquines és el mateix . Si existix més d'un SGBD distint el sistema es denomina heterogeni.

La distribució física, espacial o geogràfica de la informació, pot aconsellar la utilització d'esta arquitectura.

És molt complexa de gestionar i tan sols existixen alguns SGBD que ho suporten, i només per a homogènies.

Fragmentació
Una tècnica dels SGBD Distribuïts
La primera idea que se'ns podria ocórrer seriosa dividir la *BD original en trossos que continguen una o més taules, però es d*escarta per ineficaç i l'augment de trànsit en la xarxa.
En la fragmentació es dividixen les taules reduint trànsit i disponibilitat de dades:

- ☑ Horitzontal: Dividix registres segons unes condicions. (Ex. Els empleats per regions)
- ☑ Vertical: Dividir els camps. (Ex. Els camps més pesats, currículum, fotos, historial etc, en un processador més potent i en un altre menor els més lleugers Nom, cognoms, adreça)
- ☑ Mixta. Ambdues alhora . Molt costós de mantindre

Esta tècnica c*omplica l'accés a dades pel que només és recomanable usar quan siga molt necessari per temes d'eficiència..
Una altra opció és la Replicació de dades que és una b*uena opció si només hi ha consultes encara que a*umenta molt les actualitzacions i l'espai..

3.7. SGBD comercials i lliures

Amb Internet, el programari lliure s'ha consolidat com a alternativa, tècnicament viable i econòmicament sostenible al programari comercial, i cada vegada més barat. Com a exemple, en ofimàtica tenim *Open office o Microsoft Office.
També disposem SGBD a un nivell de propòsit general com *MySQL, SQL *Server i si parlem de més potència i funcionalitat *Oracle.
No convé decantar-se per un en concret, hem d'usar en primer lloc el que millor ens funciona donades les nostres restriccions particulars

Quant a si ha de ser lliure o no, la decisió dependrà de si disposem de personal qualificat, i en este cas un sistema lliure és més barat i potent.
D'altra banda , els sistemes lliures cada vegada proveïxen una millor i més eficient documentació tant a nivell oficial com a través de múltiples fòrums i *blogs,* la qual cosa fa que cada vegada ho use més gent i millore contínuament. No obstant això sempre existix el risc que siga comercialitzat i deixe d'estar disponible de manera oberta

No convé decantar-se per un en concret, hem d'usar en primer lloc el que millor ens funciona donades les nostres restriccions particulars.
Una aplicació d'agenda pot ser perfectament vàlid un *Access o fins i tot un *Openoffice *calc (i en segons quins casos fins i tot un *Bloc de notes).* Si per contra , la meua base de dades requerix una certa quantitat d'accessos d'usuaris diversos, control d'integritat i altres funcionalitats, hauria de plantejar-me alguna cosa com *MySQL.
Finalment, si parlem d'una gran corporació que requerix ferramentes avançades per a grans bases de dades , servici tècnic, operacions crítiques, potser hem de plantejar-nos sistemes més potents com *Oracle.

Altres tipus i usos de *BD poden ser com m*inería de dades, osota de *IA (n*egocios, hàbits, fugides, fraus, internet, terrorisme), en s*istemas Experts, com a*yuda a la decisió, com Data W ar*ehouse (Magatzem de dades), en

s*istemas integrals per a empresa, *SAP,…

***NOSQL**

És una àmplia classe de sistemes de gestió de bases de dades que diferixen del model clàssic del sistema de gestió de bases de dades relacionals.
No usen SQL com el principal llenguatge de consultes.
Les dades emmagatzemades no requerixen estructures fixes com a taules
No suporten operacions *JOIN, ni garantixen completament *ACID
Escalen bé horitzontalment

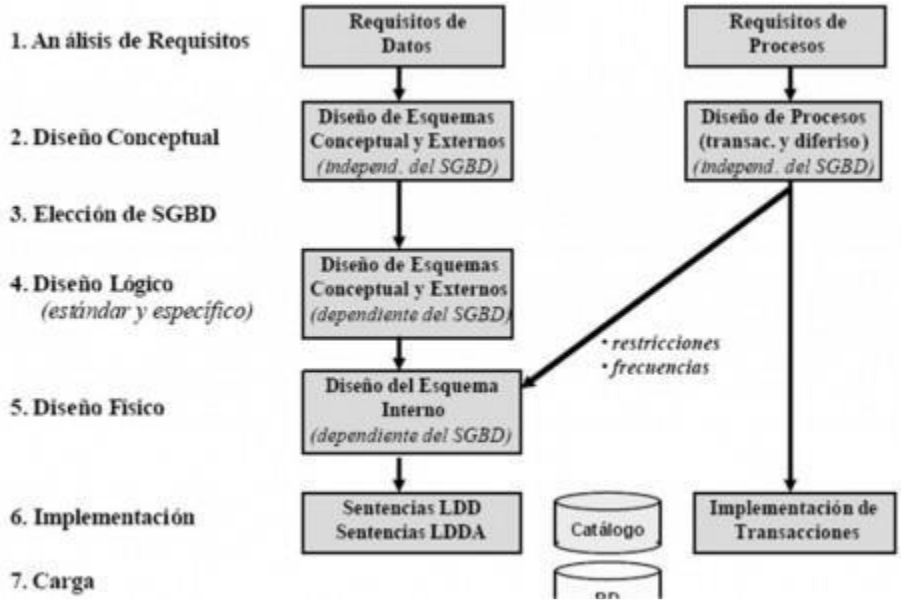
Per què sorgixen

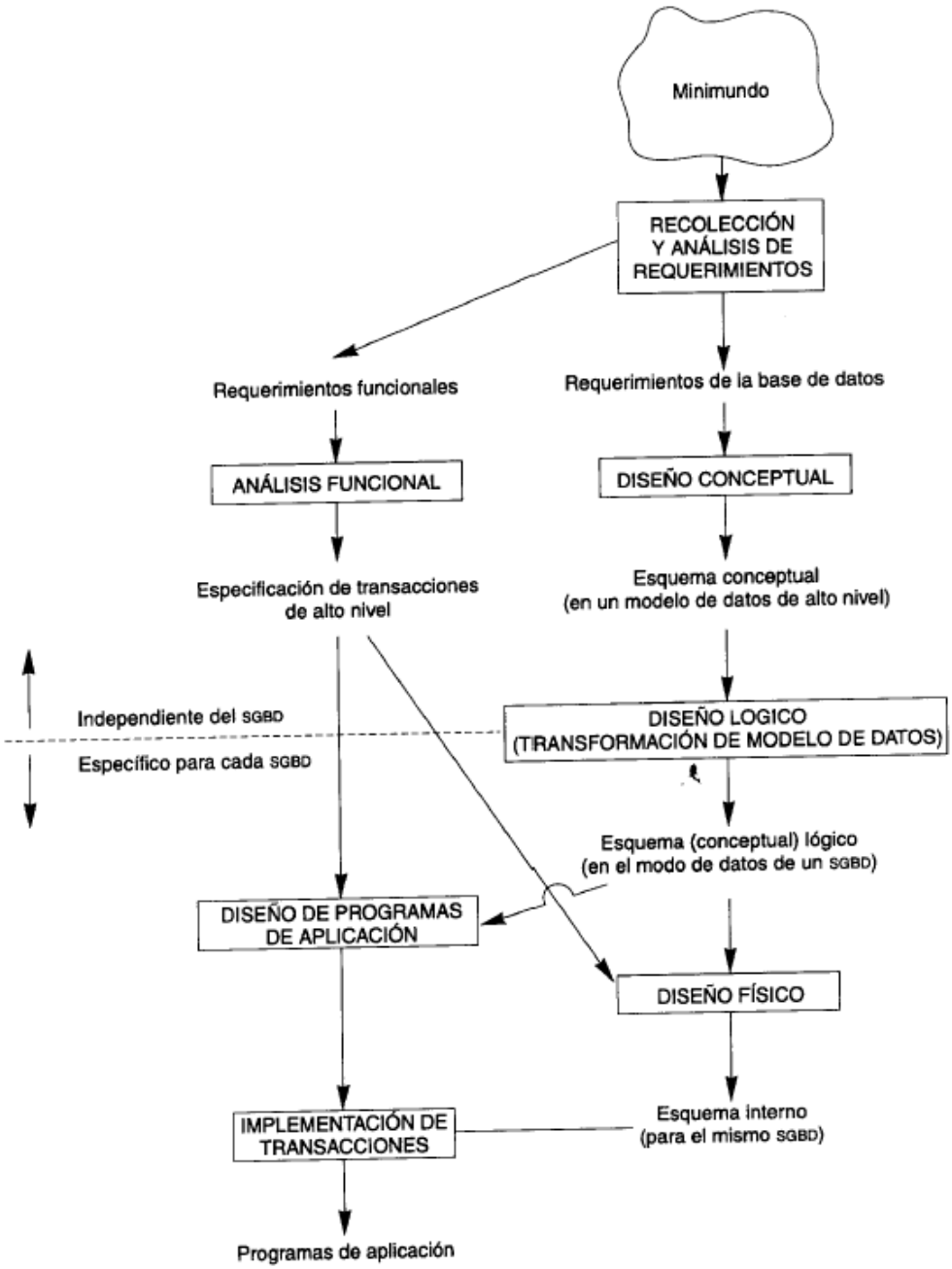
L'aparició de la web, el programari com a servici, els servicis en el núvol i les *startups d'èxit, *facebook, *google…, amb milions d'usuaris, i amb tot això van arribar els problemes d'alta escalabilitat.
Els sistemes *NoSQL intenten atacar este problema proposant una estructura d'emmagatzematge més versàtil encara que siga a costa de consistència i funcionalitat.
Algunes implementacions ben conegudes que podríem com *NoSQL són: *CouchDB, *MongoDB, *RavenDB, *Neo4j, *Cassandra, *BigTable, *Dynamo, *Riak, *Hadoop, i moltes altres

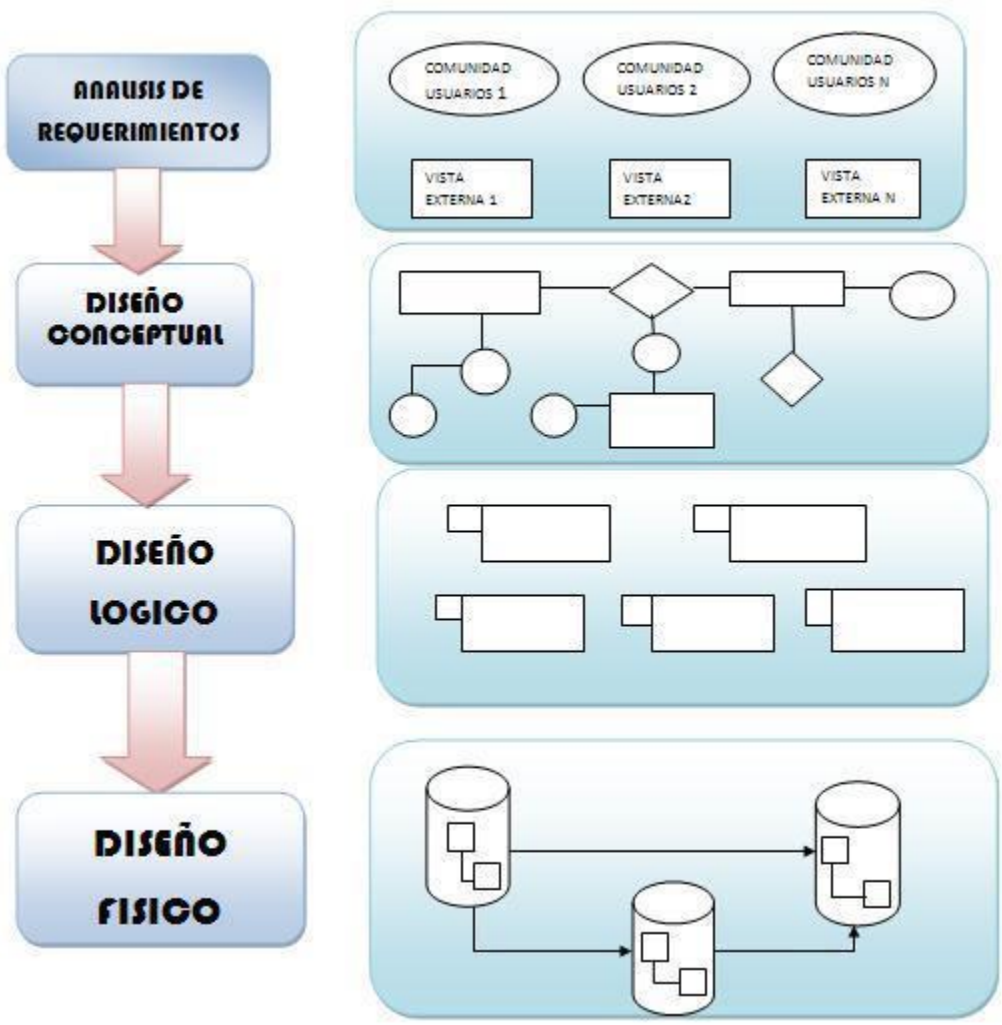
3.8. Cicle de desenrotllament SGBD

Un Sistema de Base de dades és una representació informàtica d'un fet de la realitat.
Hi ha una gran distància entre la realitat i les estructures de dades emmagatzemades en els suporte físics d'un ordinador, després per a arribar a esta representació requerim de diverses fases.
Una vegada realitzat i implantat. Vindria una altra nova fase, a vegades , molt erròniament, no tinguda en compte : EL MANTENIMENT.

FASESDESENROTLLAMENT


1. An álisis de Requisitos
Requisitos de Datos
Requisitos de Procesos
2. Diseño Conceptual
Diseño de Esquemas Conceptual y Externos (Independ. del SGBD)
Diseño de Procesos (transac. y diferiso) (Independ. del SGBD)
3. Elección de SGBD
4. Diseño Lógico (estándar y específico)
Diseño de Esquemas Conceptual y Externos (dependiente del SGBD)
• restricciones
• frecuencias
5. Diseño Físico
Diseño del Esquema Interno (dependiente del SGBD)
6. Implementación
Sentencias LDD Sentencias LDDA
Catálogo
Implementación de Transacciones
7. Carga


Minimundo
RECOLECCIÓN Y ANÁLISIS DE REQUERIMIENTOS
Requerimientos funcionales
Requerimientos de la base de datos
ANÁLISIS FUNCIONAL
DISEÑO CONCEPTUAL
Especificación de transacciones de alto nivel
Esquema conceptual (en un modelo de datos de alto nivel)
Independiente del SGBD
Específico para cada SGBD
DISEÑO LOGICO (TRANSFORMACIÓN DE MODELO DE DATOS)
Esquema (conceptual) lógico (en el modo de datos de un SGBD)
DISEÑO DE PROGRAMAS DE APLICACIÓN
DISEÑO FÍSICO
IMPLEMENTACIÓN DE TRANSACCIONES
Esquema interno (para el mismo SGBD)
Programas de aplicación


ANALISIS DE REQUERIMIENTOS
DISEÑO CONCEPTUAL
DISEÑO LOGICO
DISEÑO FISICO
COMUNIDAD USUARIOS 1
COMUNIDAD USUARIOS 2
COMUNIDAD USUARIOS N
VISTA EXTERNA 1
VISTA EXTERNA2
VISTA EXTERNA N