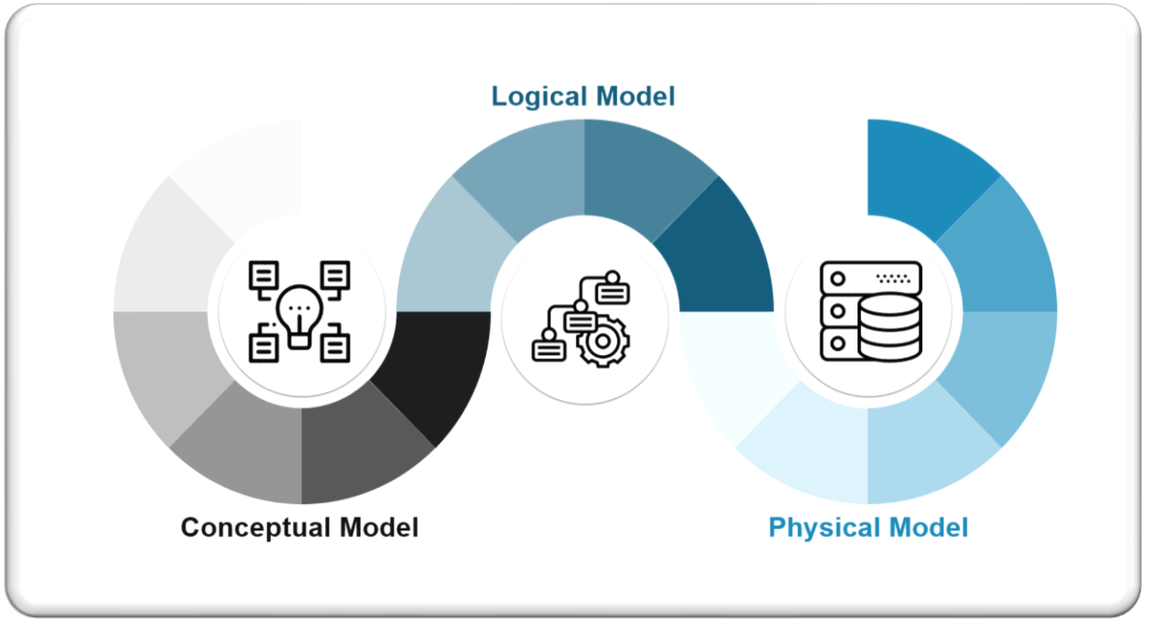
Models de dades



institut sa palomera

Cicle: ASIX-DAW-DAM
Curs: 2023/2024
Mòdul: 02 Bases de Dades

Objectius

- Definició dels models de dades
- Models de dades més utilitzats
- Eines per treballar amb models de dades
- Separació entre el model conceptual, d'implementació o lògic i el físic

Video

Data models

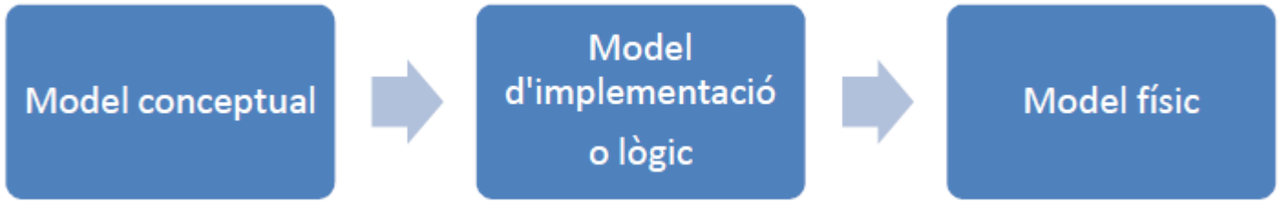
https://www.youtube.com/watch?v=zTcUj4xTnnA

Models de dades

- Una BD és una representació de la realitat. És a dir, un model de la realitat. Nosaltres volem transformar la realitat i plasmar-la en un entorn el qual el puguem tractar manipular i gestionar.

  - **Model de dades conceptual**: model de representació d'alt nivell capaç de representar part de la realitat i sense dependre de cap tipus de llenguatge. Per exemple ER/ERE, UML, etc...
  - **Model de dades d'implementació / lògic**: Conceptes que els usuaris poden entendre, però no estan massa allunyats de l'organització de dades físiques. S'amaguen alguns detalls d'emmagatzematge. Per exemple: model jeràrquic, model de xarxa, model relacional, model relacional amb objectes,..
    - Per exemple, el component fonamental per a modelar en un SGBD relacional són les taules, però en altres tipus de SGBD s'utilitzen altres components.
  - **Model de dades físic**: Conceptes a baix nivell que descriuen en detall de quina manera s'emmagatzemen les dades.

Models de dades



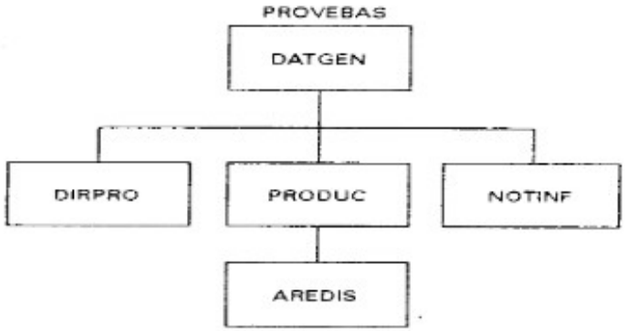
Eines models de dades

- Estructures de dades amb les quals es pot construir la BD: taules, arbres, etc..
- Diferents tipus de restriccions (o regles) d'integritat que el SGBD haurà de fer complir a les dades: dominis, claus, etc..
- Tot un seguit d'operacions per a treballar amb les dades. Per exemple, en el model relacional l'operació SELECT per seleccionar o llegir files que compleixen una condició. Un exemple d'operació dels models jeràrquic i en xarxa podria ser la que ens diu si un determinat registre té fills o no.

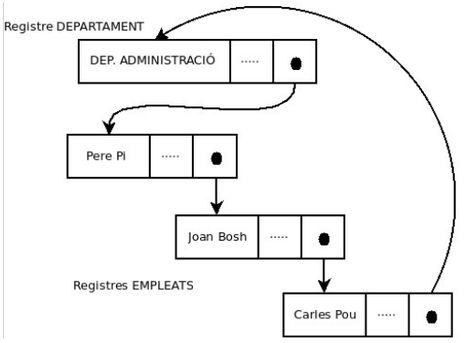
Models de Base de dades

Els models de dades més utilitzats al llarg del temps han estat els següents, exposats en ordre d'aparició:

- **Model Jeràrquic**: Tal i com hem explicat en l'apartat d'història el primer model de BD utilitzat va aparèixer a principis dels anys seixanta i va ser el model jeràrquic. Les seves estructures són registres interrelacionats en forma d'arbres. El SGBD clàssic d'aquest model és l'IMS/DL1 d'IBM. El model jeràrquic va ser desenvolupat per permetre la representació de situacions de la vida real a on predominen les relacions de tipus 1:N (Un curs(1) hi ha diferent alumnes(N))

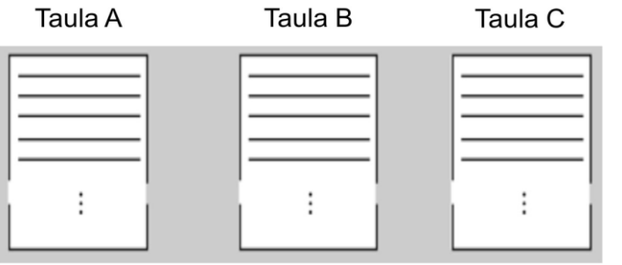


- **Model En xarxa**: El model de dades en xarxa representa les entitats en forma de nodes d'un graf i les relacions entre aquestes mitjançant arcs que uneixen els diferents nodes. Amb aquest model es millorava l'anterior de tal manera que es donava una solució més eficient al problema de redundància de dades, però tot i així la dificultat d'administrar la informació d'una base de dades en xarxa ha significa que els seus usuaris siguin purament tècnics.



Models de Base de dades

- **Model Relacional** :els models pre-relacionals i el model relacional és que aquest es limita al nivell lògic, no fa absolutament cap consideració sobre les representacions físiques i d'aquesta forma aconsegueix una independència física de dades total. Tal i com hem dit anteriorment aquest model va ser introduït per Codd (1970). Així com els models pre-relacionals (jeràrquic i en xarxa) les estructures de dades consten de dos elements bàsics, els registres i les interrelacions, en el model relacional la forma informal d'un sol element, la taula, formada per files I columnes. Les interrelacions s'han de modelitzar utilitzant les taules.



- **Model Orientat a Objectes** : Els últims anys s'han estès el model de BD relacional amb objectes. Es tracta d'ampliar el model relacional, afegint-hi la possibilitat que els tipus de dades siguin tipus abstractes de dades, TAD. Des de fa poc, han entrat en joc els SGBD anomenats noSQL considerades la pròxima generació de SGBD amb els següents punts característics: no relacionals, distribuïdes, no tenir un esquema fix, open-source (algunes), fàcilment escalables de forma horitzontal.

Models de Base de dades - NoSQL

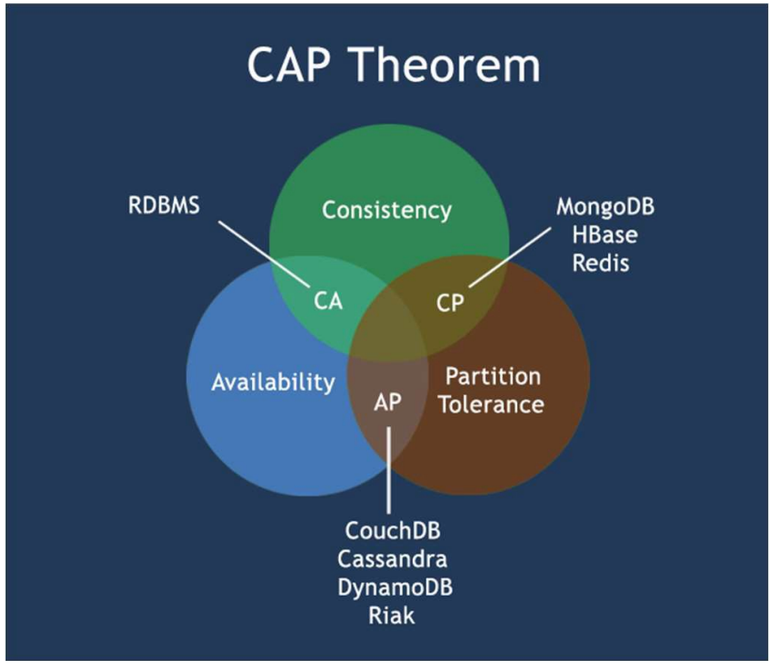
- Des de fa poc, han entrat en joc els SGBD anomenats noSQL considerades la pròxima generació de SGBD amb els següents punts característics: no relacionals, distribuïdes, no tenir un esquema fix, open-source (algunes), fàcilment escalables de forma horitzontal.

Per què NoSQL?

- La majoria dels SGBD clàssics es basen en transaccions per tal de garantir la integritat de les dades. Això assegura la consistència de les dades en totes les situacions de gestió de les dades de forma concurrent. Aquestes característiques transaccionals també conegudes com ACID (Atomicity, Consistency, Isolation, Durability) fan que no sigui fàcil ampliar-los i escalar-los tal com es demostra en el teorema CAP.

- **Teorema de CAP**

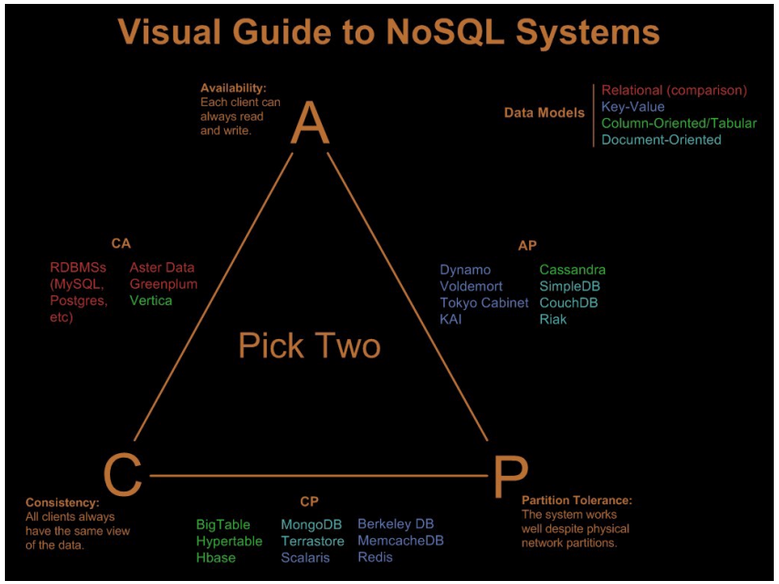
https://es.wikipedia.org/wiki/Teorema_CAP
https://www.youtube.com/watch?v=Jw1iFr4v58M



Models de Base de dades - NoSQL

En informàtica teòrica, el teorema CAP (Consistency, Availability, Partition Tolerance), també conegut com a teorema de Brewer (Eric Brewer), formula que és impossible garantir simultàniament les tres característiques següents en una aplicació distribuïda:

- **Consistència**: Tots els nodes veuen la mateixa dada al mateix temps.
- **Disponibilitat**: La garantia que cada petició a un node rep una resposta de si ha tingut èxit o a fallat.
- **Tolerància a la partició**: El sistema continua operant malgrat la partició arbitrària a causa d'errors en la xarxa i això és totalment transparent pels clients

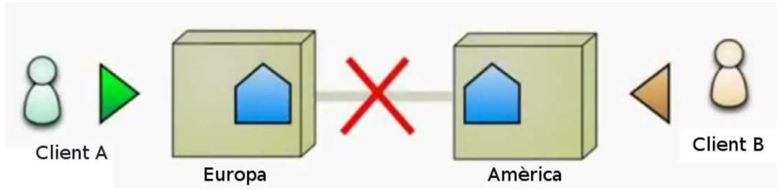


Molts del SGBD NoSQL, s'han preocupat més per tenir una alta disponibilitat i partionat disminuint les seves característiques de Consistència/Coherència.

Un dels principals objectius dels sistemes NoSQL és l'escalabilitat horitzontal (augmentar el número de màquines) i per fer això es necessita d'una forta tolerància de partició de xarxa que requereix renuncia a qualsevol consistència o disponibilitat.

Aplicació distribuïda de reserves

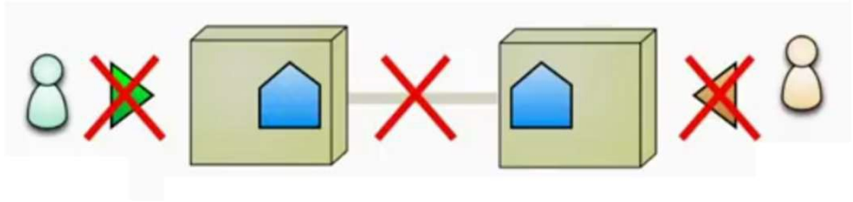
- Per exemple imaginem que tenim una aplicació distribuïda de reserves d'apartaments .
- Per millorar el rendiment tenim la base de dades distribuïda en dos continents: Europa i Amèrica.



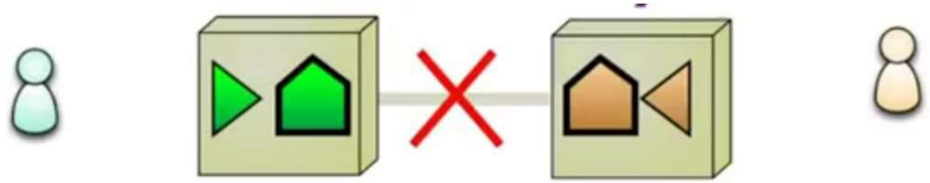
- Imaginem que la comunicació entre Europa i Amèrica falla. Què fem amb els clients que volen fer una reserva en el mateix apartament? Els deixem fer reserves o no? Què premiem?

Aplicació distribuïda de reserves

- Consistència: No deixem fer la reserva, perquè no sabem si Europa o Amèrica tenen una reserva per aquell apartament concret.



- Disponibilitat: Deixem fer la reserva i ja arreglarem els possibles problemes de consistència més endavant.



Important: ÉS EL NEGOCI/CLIENT QUE DECIDIRÀ QUÈ VOL FER!!!

Relacional vs NoSQL

Quan utilitzar Relacional?

- Estem al davant d'una mitjana o gran escala de base de dades (10-100GB) amb poca concurrència (cents d'usuaris com a màxim);
- Quan necessitem de ACID, especialment l'Atomicitat i la Consistència; Les dades tenen moltes relacions entre elles (jerarquies, mestre-esclau,etc..);
- Quan volem emmagatzemar una àmplia varietat de dades en cents o milers de taules normalitzades.

Els sistemes relacions estan dissenyats per ser escalats verticalment i de bon inici necessiten màquines amb molta quantitat de memòria, ràpid I/O throughput mitjançant SANs i SSDs. S'escalen ocasionalment a través de clustering i partició de les dades

Quan utilitzar NoSQL?

- Volem implementar a gran escala una base de dades d'alta concurrència (cents de GB, milers d'usuaris).
- No requerim de les regles ACID
- No requerim de relacions i restriccions a l'esquema de dades
- Quan utilitzem poques taules (5-10 taules en el model relacional).
- Quan volem implementar alta disponibilitat amb costos baixos.

Aquests sistemes són bons en molts dels llocs web (web sites). Per exemple, Google, Twitter Creieu que és realment important si es retrassen uns segons alguns tweets o fins i tot que es perdin? I si Google no mostra la informació de resultats de l'última pàgina web que s'ha creat o ell ha escanejat? O per exemple si els resultats de Google perd algun enllaç pel camí?

Classificació bases de dades NoSQL

Si considerem el Teorema CAP trobem 4 tipus de SGBD:

Orientats a documents

- Són SGBD que gestionen dades semi estructurades. És a dir documents. Aquest documents normalment són formats estàndard (XML, JSON, ...)
  - Exemples: MongoDB, CouchDB. (JSON document)

Orientats a columnes

- Aquest tipus de base de dades estan pensades per realitzar consultes i agregacions sobre grans quantitats de dades. El seu model de dades és similar al de les BD relacionals, però emmagatzemen columnes de dades en comptes de registres.
  - Exemples: HBase, Cassandra (aquesta és un híbrid entre orientada a columnes i clau-valor).

Orientats a Clau-Valor

- Aquestes són les més senzilles d'entendre. Simplement guarden una dada (string, enter, JSON) mitjançant una clau única. Cada vegada que volem fer referència aquella dada ho hem de fer mitjanç ant aquesta clau.
  - Exemples: DynamoDB, Redis, Memcaché (només a memòriaprincipal i no es consideraria SGBD).

Graf

- Basades en la teoria de grafs utilitzant nodes i arestes per representar les dades emmagatzemades. Són molt útils per guarda informació en models amb moltes relacions, com xarxes, I connexions socials. Fan pensar amb els models de xarxa.
  - Exemples: Infinite Graph, Neo4j, MongoDB

WEBGRAFIA

- Batini, C.; Ceri, S.; Navathe, S.B. (1992). Conceptual Database Design: An Entity-Relationship Approach. Reading, Massachusetts: Addison Wesley.
- Teorey, T.J. (1999). Database Modeling & Design. The Fundamental Principles (3a ed.). San Francisco: Morgan Kaufmann Publishers, Inc.
- Teorema de CAP https://es.wikipedia.org/wiki/Teorema_CAP
- Video funcionament teorema CAP, Setembre 2023, https://www.youtube.com/watch?v=Jw1iFr4v58M