

8. Complements

- ✓ 1. SGBD
- ✓ 2. Escalabilitat
- ☞ 3. NO SQL
- 4. Altres "models"

8.3 NoSQL

a. Conceptes i principis

- Big Data
- NoSQL
- ACID vs BASE
- Teorema CAP

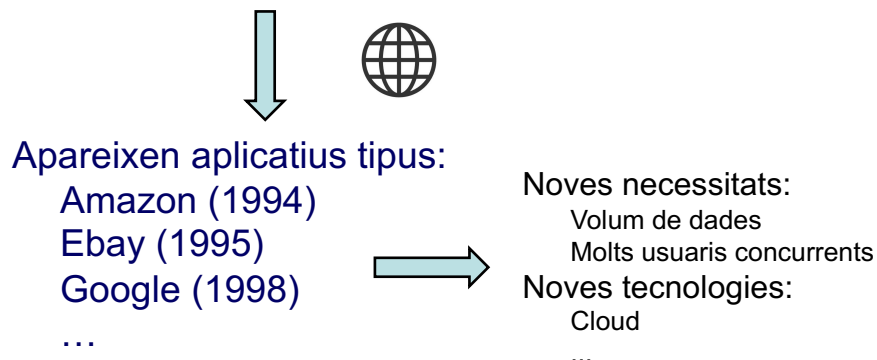
b. Tipus

c. Pros/Contres

8.3 NoSQL

a. Conceptes i principis

Les BD relacionals dominen el mercat des de mitjans del anys 80 del passat segle, però ...



3

8.3 NoSQL a. Conceptes i principis

Big Data

Sorgeix el terme de BIG DATA

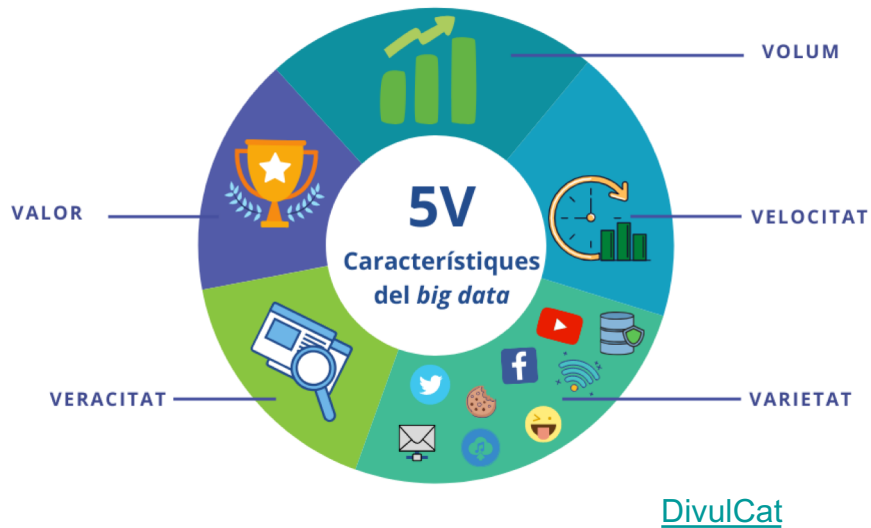
Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.

Gartner

I les 3, 5 o 7 V's

Volum
Velocitat
Varietat

4



Els RDBMS no són adequats per a tractar Big Data, no escalen prou bé.

Solució: Implementar models alternatius que s'ajustin a les necessitats

Google, Facebook, Amazon ... creen el seu propi SBBD



NoSQL: terme presentat al 1998 per Carlo Strozzi, descriu un conjunt de SGBDs que es diferencien dels relacionals bàsicament en:

- No tenen un model fix (taules , relacions...)
- Desnormalitzats
- No empren JOINS
- Arquitectura distribuïda
- Gran capacitat per replicar i escalar
- No garanteixen ACID



Mantenen el concepte de clau!!

Beu història dels inicis:

1998- Carlo Strozzi emprà el terme NoSQL per a la seva BD de codi obert "relacional" sense SQL

2000- Apareix la *graph-database* Neo4j



2004- Google publica BigTable

2005- Apareix CouchDB



2007- Es publica el primer article de Amazon sobre Dynamo

2008- Facebook allibera el projecte Cassandra

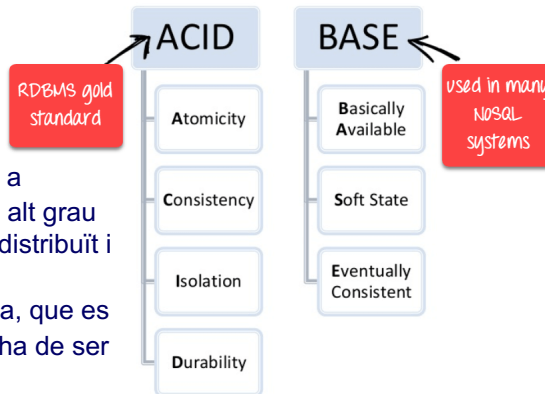
2009- Es re-introdueix el terme NoSQL

Principis **ACID vs BASE****BASE:**

Disponibilitat bàsica. Enfocat a disponibilitat malgrat hi hagi un alt grau d'errors. Enfocament altament distribuït i replicat

Estat suau. No cal consistència, que es problema del programador, no ha de ser controlada pel SGBD

Consistència eventual L'únic requisit quan a consistència es què, en algun moment del futur, les dades convergeixen a un estat consistent, sense garantir quan es produirà



- STRONG CONSISTENCY
- ISOLATION
- TRANSACTIONS
- SCALE-UP (LIMITED)
- PRECISE ANSWERS
- CONSISTENCY FIRST



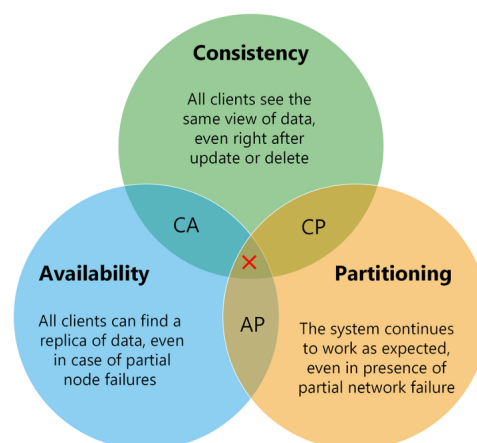
- WEAK CONSISTENCY
- LAST WRITE WINS
- DEVELOPER MANAGED
- SCALE-OUT (UNLIMITED)
- APPROXIMATE ANSWERS
- AVAILABILITY FIRST

Teorema CAP: o de Brewer (2000), estableix que es impossible que una BD distribuïda proporcioni simultàniament més de dues de les tres garanties:

- **Consistència:** cada lectura obté la dada actualitzada o error
- **Disponibilitat:** Cada lectura obté una resposta, sense error i sense la garantia de que es correspongui amb l'escriptura més recent. Conflicte amb l'Atomicitat de **ACID**
- **Tolerància de partició:** el sistema continua funcionant malgrat caiguin (o es retardin) un nombre arbitrari de missatges entre nodes. Conflicte amb Aïllament de **ACID**

Teorema CAP: Ofereix tres possibilitats:

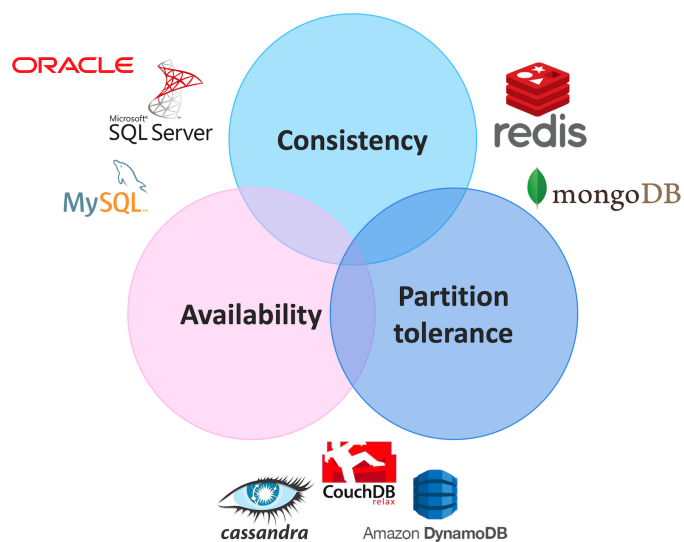
CP
AP
CA



Teorema CAP: Ofereix tres possibilitats:

- **CP** (Consistència i Tolerància al particionament): no garanteix la disponibilitat, enfocat a aplicar els canvis de forma consistent malgrat es perdi comunicació amb alguns nodes
- **AP** (Disponibilitat i Tolerància al particionament): no garanteix que les dades siguin iguals a tots els nodes durant tot el temps, però el sistema sempre estarà disponible
- **CA** (Consistència i disponibilitat): no admet particionament, ja que garanteix que les dades sempre son consistents i accessibles per a totes les peticions

13



14

En cas d'una caiguda de la xarxa, cal decidir:

- **Cancel·lar l'operació:** disminueix disponibilitat garantint consistència: ORACLE, MySQL, Postgres, SQL Server ...
- **Continua l'operació:** Manté disponibilitat però amb possible inconsistència de dades: Redis, MongoDB ...
- **Continua l'operació sense importar gens ni mica la consistència:** Cassandra, CouchDB, DynamoDB ...

- a. **Conceptes i principis**
- b. Tipus
- c. Pros/Contres