**NoSQL**

**Ripasso e lezione**

Flat file system: info che contenevano i dati, doc semplice(le info gestite con tab). fare tutto a mano con difficoltà nell’implementare sicurezza. inefficiente

Oggi con db relazionali tutto + semplice(normalizzazione, funzionalità,.... . ma cmq alcune incongruenze da gestire, es join…)

ma perchè **Not Only SQL ?**

fare analisi su dati strutturati e non, oltre agli zettabyte

questione di scalabilità flessibilità affidabilità e costi

**Scale-in:** scalatura verticale,

**Scale-out:** scalatura orizzontale

**Benefici RDBMS.**

basati su proprietà ACID(Atomicità, Coerenza, Isolamento e Durabilità)

Consistenza e recupero

normalizzazione

uso di SQL

e uso di scale-in

**Limiti RDBMS**

NON sono progettati per modalità distribuite(problemi costi di join, difficile scale-out, costi HD e SW )

Crescita big data, non supportano grandi varietà e volumi di dati

**NoSQL:**

sono database non relazionali

creati per uso di Big data e applicazioni web “Real time”

***Vantaggi*:**

gestire data semistrutturati e non

data models non per forza con schema predefiniti

scalabilità: scale-out(orizzontale)

evita principi di ACID(non supportati)

evita query sql complesse

**teorema di CAP**

**consistency:** leggere stesso data**,**

**availability:**sempre disponibili,

**partial tolerance:** possibilità di distribuire dati evitando perdite,

**Classificazione NoSQL:**

*tipo key value pair based* (accesso coppia chiave valore, contenuto nascosto. dati profilazione, ecommerce ecc)

*tipo colonna* (memorizzano dati in famiglie di colonne, contengono righe che contengono a loro volta colonne)

*tipo documento* (uso di documenti che si auto descrivono, es doc json, uso gerarchia)

*tipo graph* (ragionano per relazioni tra entità ecc, raggiungimento dati a seconda del percorso raggiungibile, es data spaziali

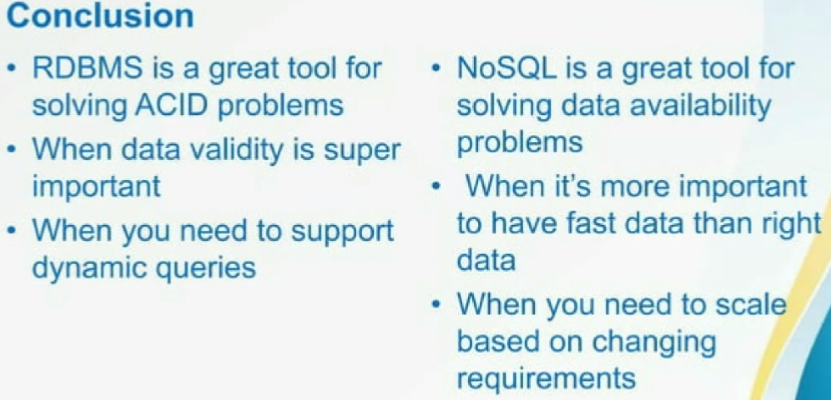


**Possibili criticità**

**processo transazione:** no acid

**processo query:** ogni NoSql ha suo linguaggio

**sicurezza:** basato a posizione struttura



**Modelli di transazione**

<https://www.geeksforgeeks.org/acid-model-vs-base-model-for-database/>

<https://aws.amazon.com/it/compare/the-difference-between-acid-and-base-database/>

ACID è l'acronimo di Atomicity, Consistency, Isolation, Durability (CONSISTENZA)

BASE è l’acronimo di Basically Available, Soft State, Eventually consistent (AFFIDABILITÀ )

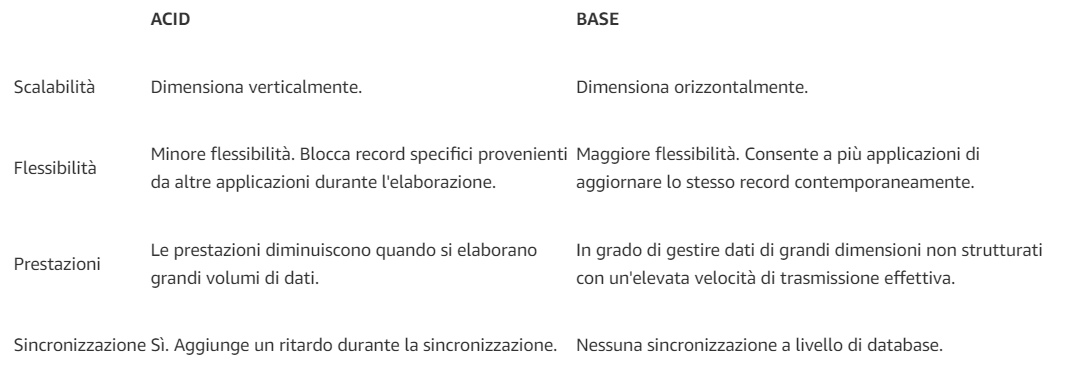


**USO**

**Acid:** per dati prevedibili, garantendone l’integrità dati, analitica

**Base:** per dati meno strutturati e grandi volumi



ram senza controllo ECC per server = + fast del pc standard

mongodb atlas

davidebotta

nosql2024

mongodb+srv://davidebotta:nosql2024@corsonosql.w4p5gz1.mongodb.net/