NoSQL adatbázis-kezelők

Szárnyas Gábor (BME) diáinak felhasználásával

https://www.db.bme.hu/targyak/adatbazisok-haladoknak

Relációs adatmodell, SQL

- Codd: A Relational Model of Data for Large Shared Data Banks, 1970
- 1970-es évek eleje: SEQUEL (Structured English QUEry Language)
- 1986: az SQL ANSI szabvány (Structured Query Language)

Relációs adatbázisok ma

- Kevés, nagy szereplő
- Zárt forráskódú
  - Oracle Database
  - Microsoft SQL Server
  - IBM DB2
- Nyílt forráskódú
  - MySQL ($\rightarrow$ Oracle)
  - PostgreSQL

Az SQL erősségei

- Kiforrott elmélet és technológia
- Sok szakember
- Sok szoftveres eszköz
- Bevált módszerek
- Robusztus rendszerek
- Ad hoc lekérdezések
- Tranzakciók

Konkurrens hozzáférés

- Ha egy adatbázist egyszerre több alkalmazás használhat.
- Problémák:
  - Elveszett update: egyszerre ketten ugyanazt akarják felülírni
  - Inkonzisztens állapot olvasása: egymás utáni lekérdezésekkel, ha közben valaki módosította ez adatot
  - Versenyhelyzet: például ketten egyszerre ugyanazt a helyet akarják lefoglalni

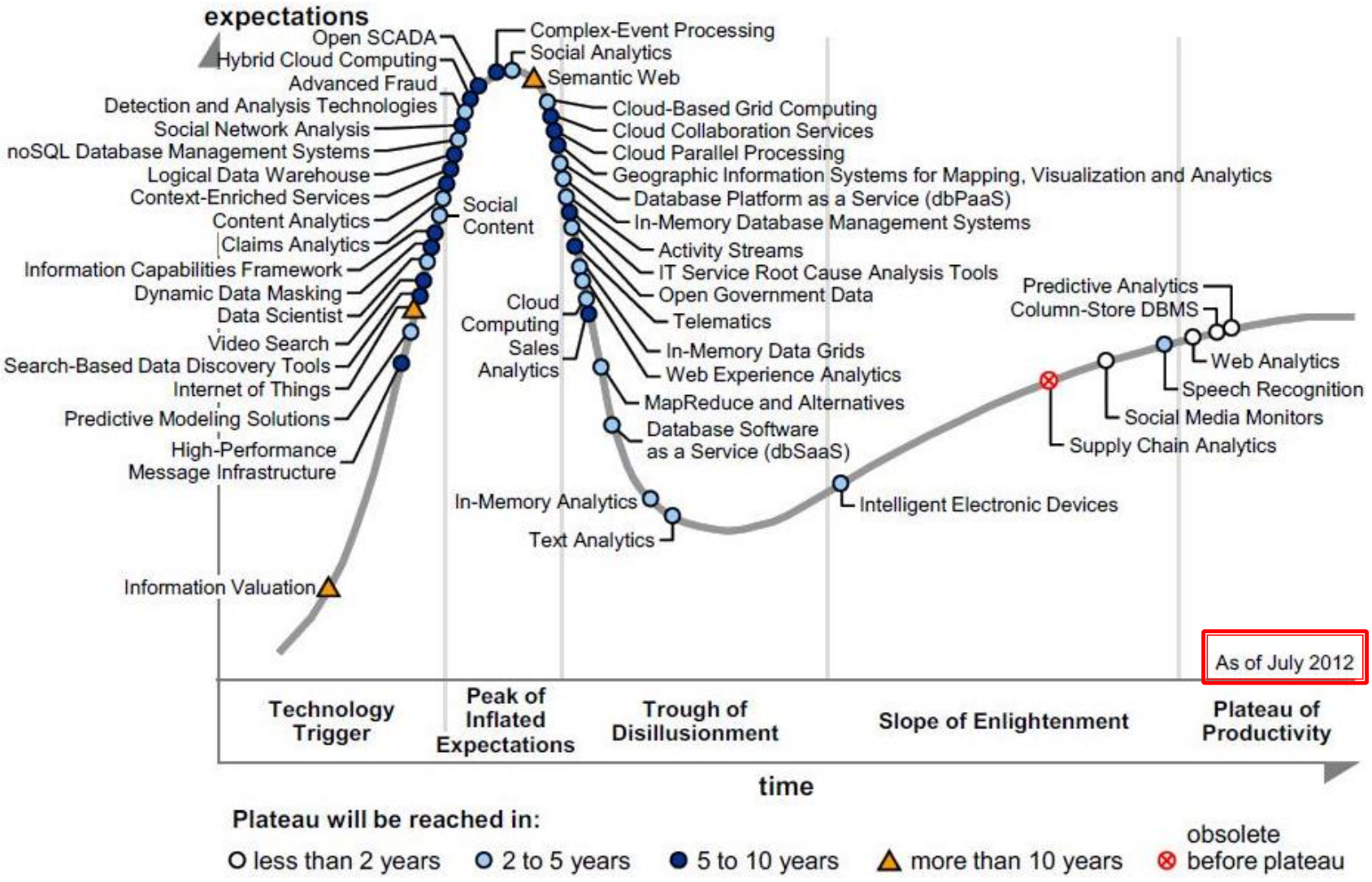
Tranzakciók – ACID garanciák

- Tranzakció: az adatbázison elvégzett műveletek logikai egysége.
- ACID garanciák:
  - Atomicity (atomikusság): egy tranzakció vagy teljesen lefut (commit), vagy teljesen visszagörgetésre kerül (rollback), nincs félig lefutott tranzakció (pl.: banki átutalás)
  - Consistency (konzisztencia): nincsenek félig kiírt rekordok, az integritási feltételek (pl.: külső kulcsok) nem sérülnek
  - Isolation (elszigeteltség): a tranzakciók nincsenek egymásra hatással, az egyszerre futó tranzakciók egymás módosításait nem zavarják (akár nem is látják). Mértékét az izolációs szint határozza meg.
  - Durability (tartósság): ha egy tranzakció eljutott a commitig, akkor eredménye benne van az adatbázisban, akkor is, ha a következő pillanatban rendszerösszeomlás következik be

Big Data

- Az IBM szerint **napi** 2,5 exabájt ($2{,}5 \times 10^{18}$ bájt = 2,5 milliárd GB) adat keletkezik
- Az elmúlt két évben jött létre a ma tárolt adatok 90%-a.
- Gyorsan, nagy mennyiségű, változatos adat keletkezik.
- *Mennyiség* (volume): az eddig megszokottnál nagyságrendekkel megnövekedett adatmennyiség.
- *Sebesség* (velocity): az adatok nagy sebességgel generálódnak, a rendszertől minél gyorsabb feldolgozást és visszacsatolást várnak el.
- *Változatosság* (variety): az adatok többféle forrásrendszerből érkeznek, lazán struktúráltak és gyakran kérdéses minőségűek.

Gartner: Big Data életciklusgörbe

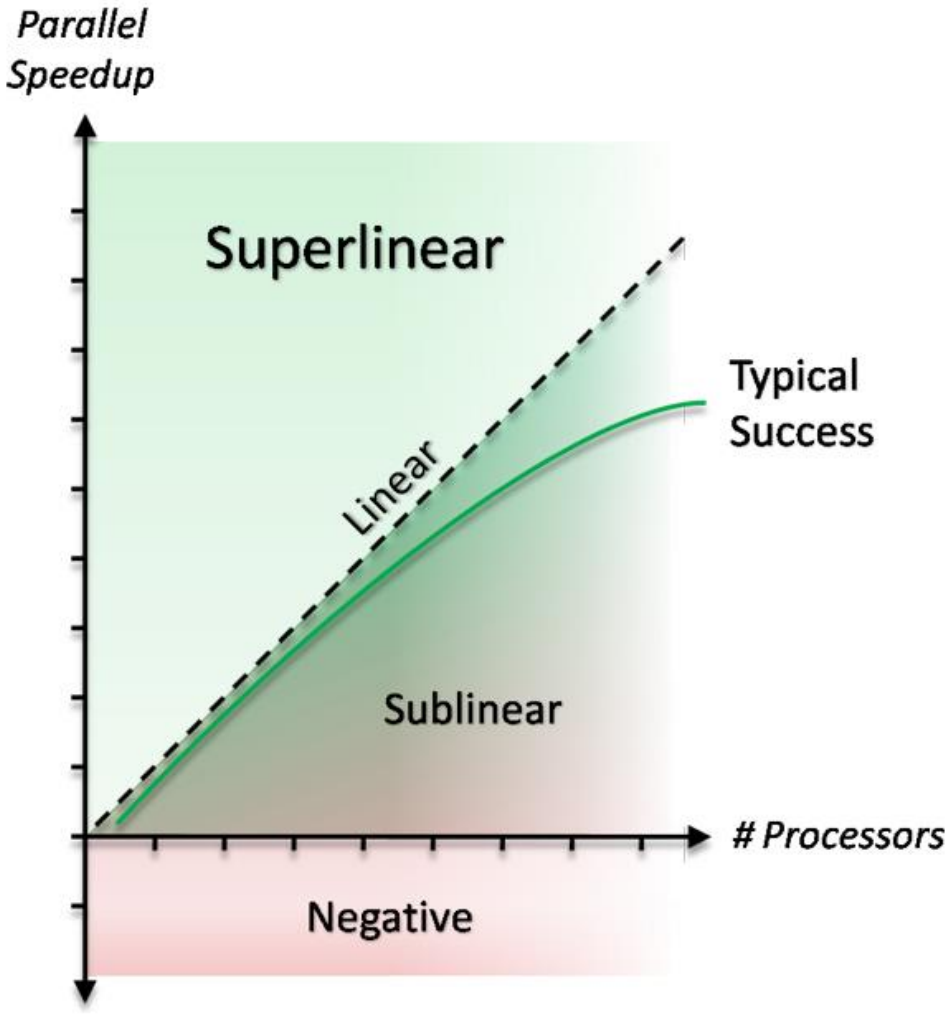


Skálázhatóság

- Skálázhatóság: a rendszer erőforrásait növelve a rendszer teljesítménye is arányosan növekszik
- Egyszerre feldolgozható lekérdezések száma, késleltetés, feldolgozott adatok mennyisége

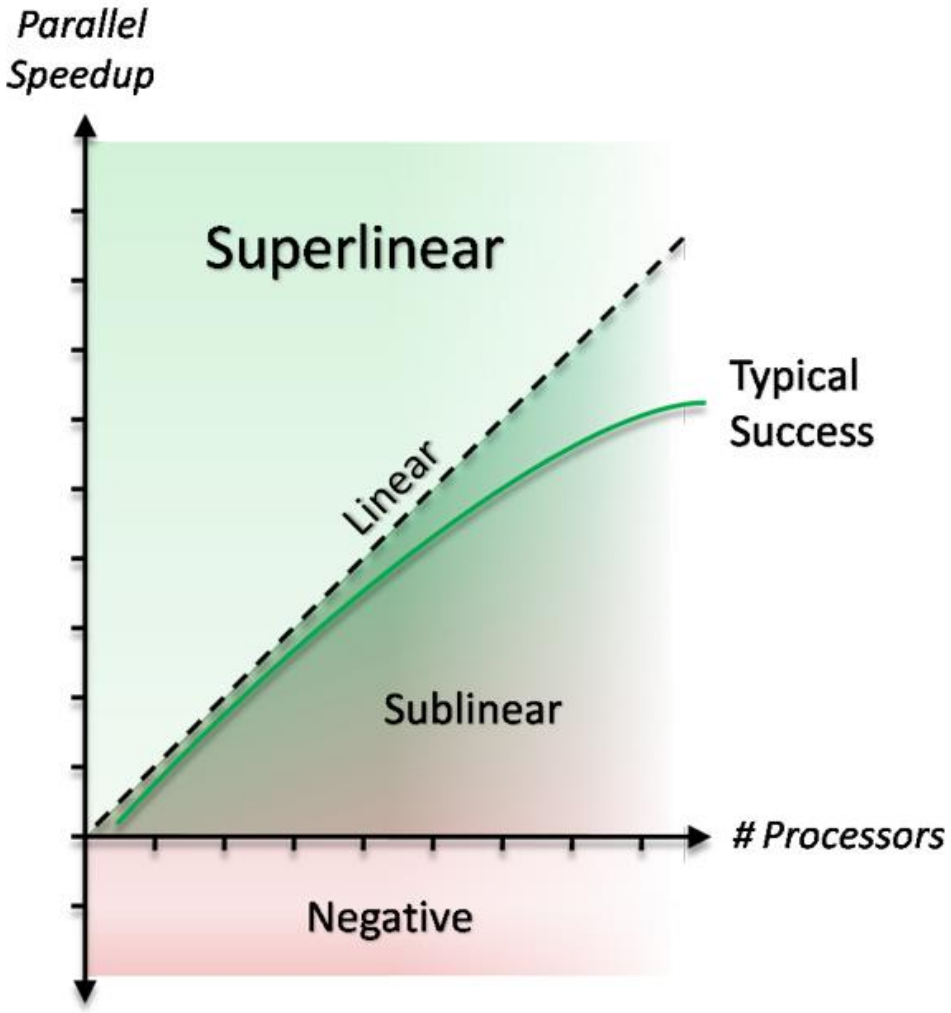
Vertikális skálázhatóság

- a rendszer kiválasztott elemét új erőforrással (processzorral, memóriával) bővítjük
- egyszerű megvalósítás



Horizontális skálázhatóság

- a rendszert új géppel bővítjük
- sok, olcsó gépből nagy teljesítmény érhető el
- elosztottsága miatt többféle meghibásodás is felléphet
- bonyolult szoftvert igényel
- legtöbb NoSQL rendszer ezt támogatja



Elosztott architektúrák

- Megosztott memória (shared memory)
- Megosztott lemez (shared disk)
- Megosztás néküli (shared nothing): a gépek nem osztanak meg CPU/diszk/memóriát, csak hálózaton beszélnek
- Sorban egyre nehezebben programozhatók, de egyre jobban skálázhatók

NoSQL – Not Only SQL

- 70-es évek: kulcs-érték tárolók, hálós adatbázismodell
- Google cikkek 2004-2006 (Google File System, Chubby, BigTable, MapReduce, Paxos Made Live) – NoSQL rendszerek elméleti alapja
- Azóta 100+ nemrelációs adatbázis-kezelő

NoSQL jellemzői

- Főbb jellemzők
  - nem-relációs adatmodell,
  - elosztott működés,
  - nyílt forráskód,
  - horizontális skálázhatóság.

- További jellemzők:
  - sémamentesség vagy gyenge séma kényszerek,
  - replikáció támogatása,
  - egyszerű alkalmazásprogramozási interfész (API),
  - fokozatos konzisztencia (eventual consistency).

A CAP tétel (2002)

- Sejtés: Eric Brewer, 2000
- Tétel: Nancy Lynch, Seth Gilbert, 2002

- Consistency (nem az ACID konzisztenciája, hanem a rendszeré): bármely időpillanatban egy adategység értékét bármely csomóponttól lekérdezve ugyanazt az értéket kapjuk
- Availability (rendelkezésre állás): minden működő csomóponthoz érkező kérésre válaszol (egy behatárolt alacsony időtartamon belül)
- Partition tolerance (partíció tolerancia): a rendszer akkor is tovább működni, ha meghibásodás miatt csomópontok esnek ki, vagy hálózati hiba miatt köztük megszakad az összeköttetés

- Elosztott rendszerben egy időben nem garantálható mindhárom tulajdonság → CAP háromszög

Triviális esetek

- Konzisztens és rendelkezésre álló - CA:
  pl.: a rendszert egy gépen futtatjuk
- Konzisztens és partíció toleráns – CP:
  pl.: hálózati partíció esetén elérhetetlenné válnak bizonyos adategységek
  pl.: ha egyáltalán nem dolgozza fel a beérkező üzeneteket
- Rendelkezésre álló és partíció toleráns – AP:
  pl.: konzisztencia gyengítésével elérhető
  pl.: bármit visszaadhat

Teljesítménymetrikák

- Áteresztőképesség [adategység/s]
- Késleltetés [s]

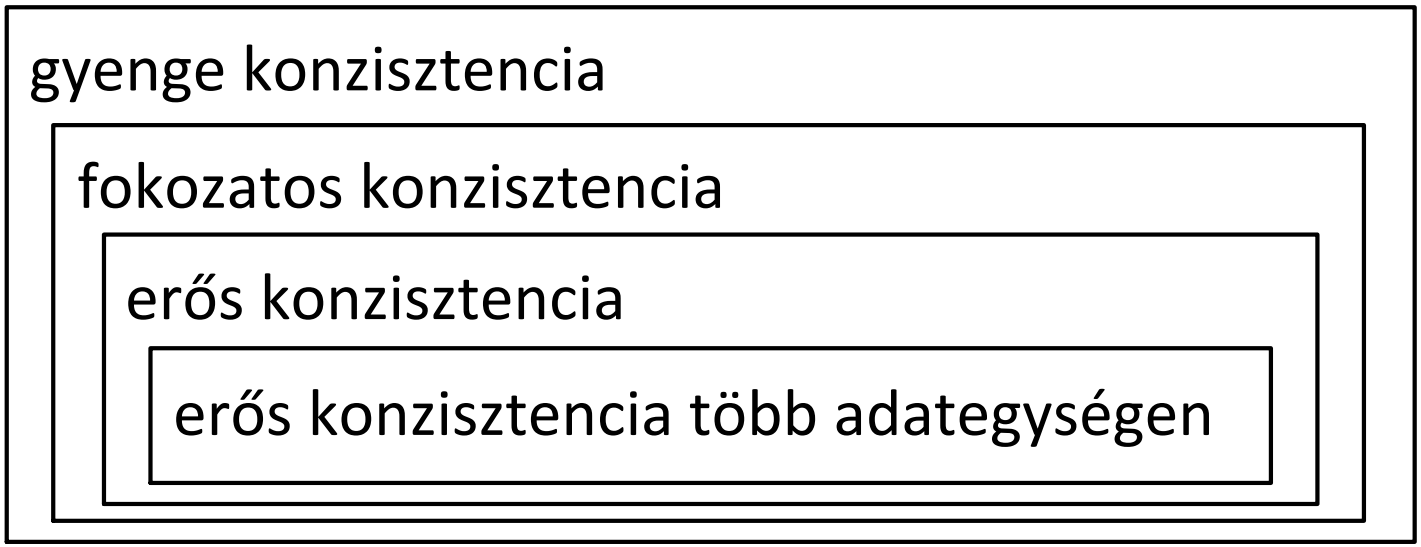
- A CAP tétel nem beszél teljesítményről

A késleltetés ára

- Amazon
  - +100 ms késleltetés
  - 1% csökkenés az eladásokban
- Google
  - +500 ms késleltetés
  - 20% bevételcsökkenés
- A konzisztencia gyengítésével a késleltetés csökkenthető

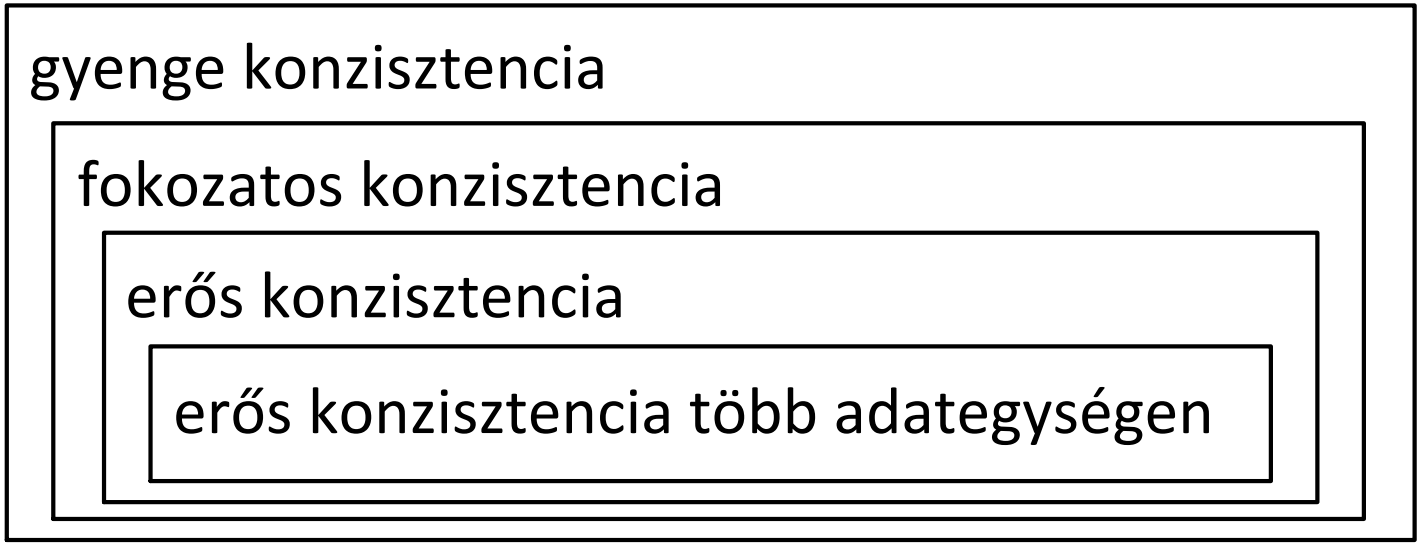
Konzisztenciamodellek

- A CAP tétel következménye
- A fejlesztők erős konzisztenciát szeretnének
- Gyenge konzisztencia: hibás működés?
- Fokozatos konzisztencia: kompromisszum



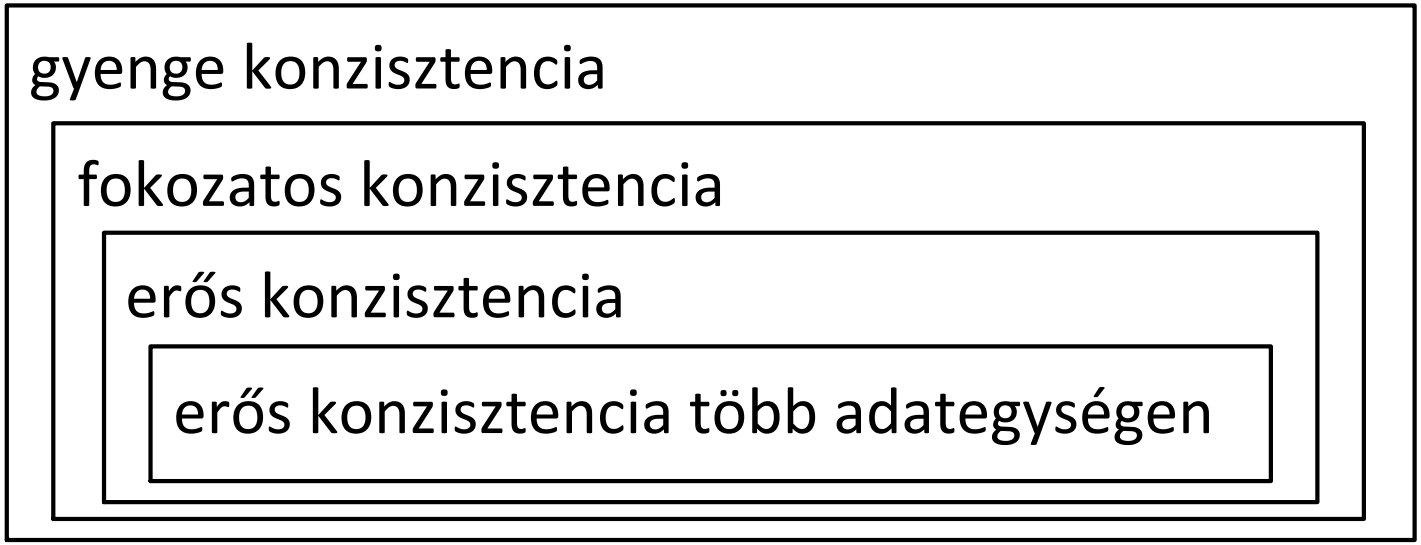
Gyenge konzisztencia (weak consistency)

- A rendszer nem garantálja, hogy az írást követő olvasások a legutoljára beírt adatot érik el. Az írás és azon pillanat között eltelt időt, amíg nem garantálható, hogy minden megfigyelő a frissített adatot látja, inkonzisztenciaablaknak (inconsistency window) nevezzük.



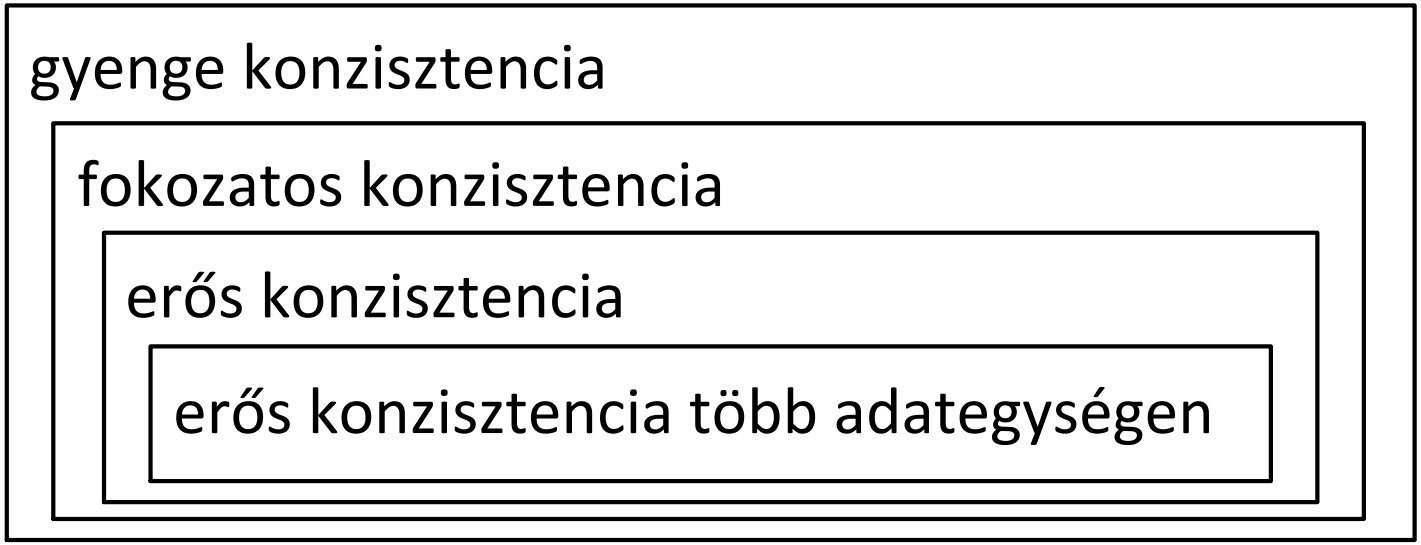
Fokozatos konzisztencia (evantual consistency)

- A gyenge konzisztencia egyik típusa. A rendszer garantálja, hogy ha nincsenek további frissítések, előbb-utóbb minden olvasás a legutóbbi írás értékét éri el.
- DNS (Domain Name System): a gyorsítótárak előre meghatározott időközönként frissülnek.



Erős konzisztencia (strong consistency)

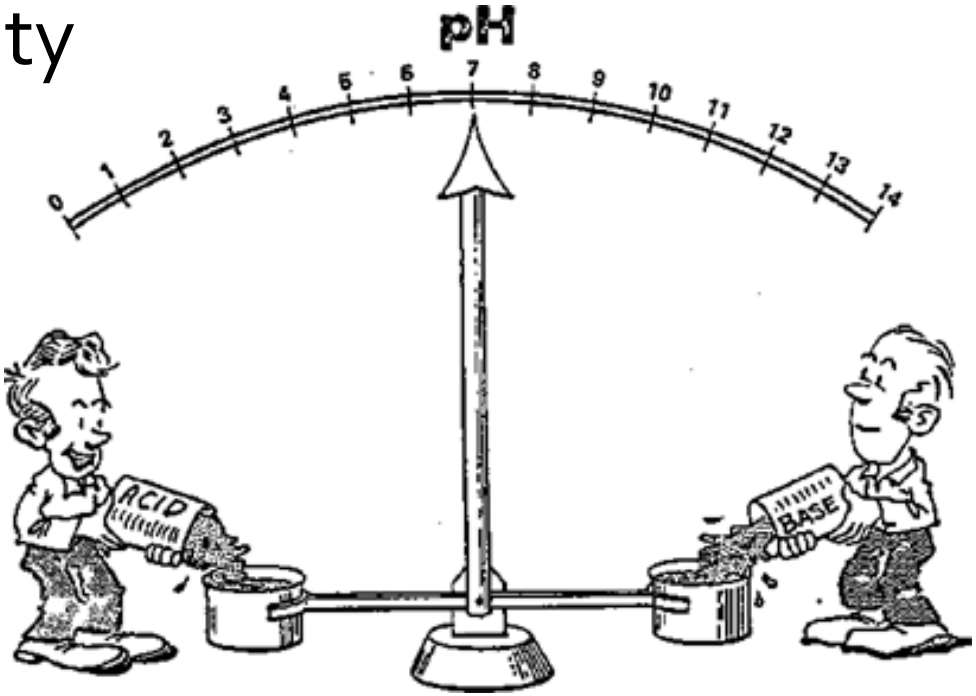
- Minden olvasás művelet az adategységen legutóbb befejezett írás művelet eredményével tér vissza, függetlenül attól, hogy az adategységet melyik csomóponton éri el.
- Könnyen érthető a rendszer felhasználói számára
- Kliensalkalmazások fejlesztése egyszerűbb



ACID–BASE

- Atomicity
- Consistency
- Isolation
- Durability

- Basically Available
- Soft state
- Eventually consistent



Replikáció

- Azonos adategység többszörözése
- ha N példányban tároljuk az adategységet
- R: olvasáshoz szükséges szerverek száma
- W: íráshoz szükséges szerverek száma
- W>N/2 (két egymást követő írásnak van közös szervere, az egymás utáni írások sorrendje megfigyelhető)
- W+R>N (az írási és olvasási területek mindig átlapolódnak → erős konzisztencia)
- W+R≤N esetén az erős konzisztencia nem garantálható, tipikusan R=1 és az inkonzisztencia ablak mérete akkora, amíg a frissítés minden replikát el nem ér

NoSQL adatbázisok típusai

- Kulcs-érték tárolók
- Dokumentumtárolók
- Oszlopcsaládok
- Gráfadatbázisok

Kulcs-érték tárolók

- Nagyon egyszerű API:
  - get(key)
  - put(key, value)
- Felhasználásuk:
  - Munkamenetek tárolása
  - Egyszerű felhasználói profilok
  - Vásárlói kosár

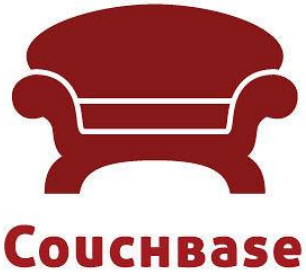
ORACLE
BERKELEY DB

ORACLE
NOSQL DATABASE

redis

Dokumentumtárolók

- Szemistrukturált adatok
- Nem szövegfile-okat kezelnek, hanem adatok laza módon struktúrált halmazát
- Nincs előre definiált séma
- JSON (JavaScript Object Notation), XML nyelvű leírások
- Felhasználásuk:
  - Naplózás
  - Tartalomkezelő rendszerek (CMS)
  - Valós idejű adatelemzés



Dokumentumtárolók – JSON

```
{"document": [
  {
    "firstname": "Klemens",
    "city": "Stuttgart",
    "age": "42"
  },
  {
    "firstname": "Rajesh",
    "city": "Delhi",
    "age": "29"
  },
  {
    "firstname": "Colin",
    "company": "Oracle"
  },
  {
    "cars": ["BMW 320d", "Jaguar XF"]
  }
]}
```

firstname	city	age	company
Klemens	Stuttgart	42	NULL
Rajesh	Delhi	29	NULL
Colin	NULL	NULL	Oracle

Oszlopcsaládok

- Sorok = kulcs-érték párok

kulcs	oszlopkulcs0	oszlopkulcs1	...	oszlopkulcsN
	érték0	érték1	...	értékN

12100	user_ID	text	datetime
	bmestudent	just decomposed a schema to 3NF #db #exam	2011-01-03 07:30:11

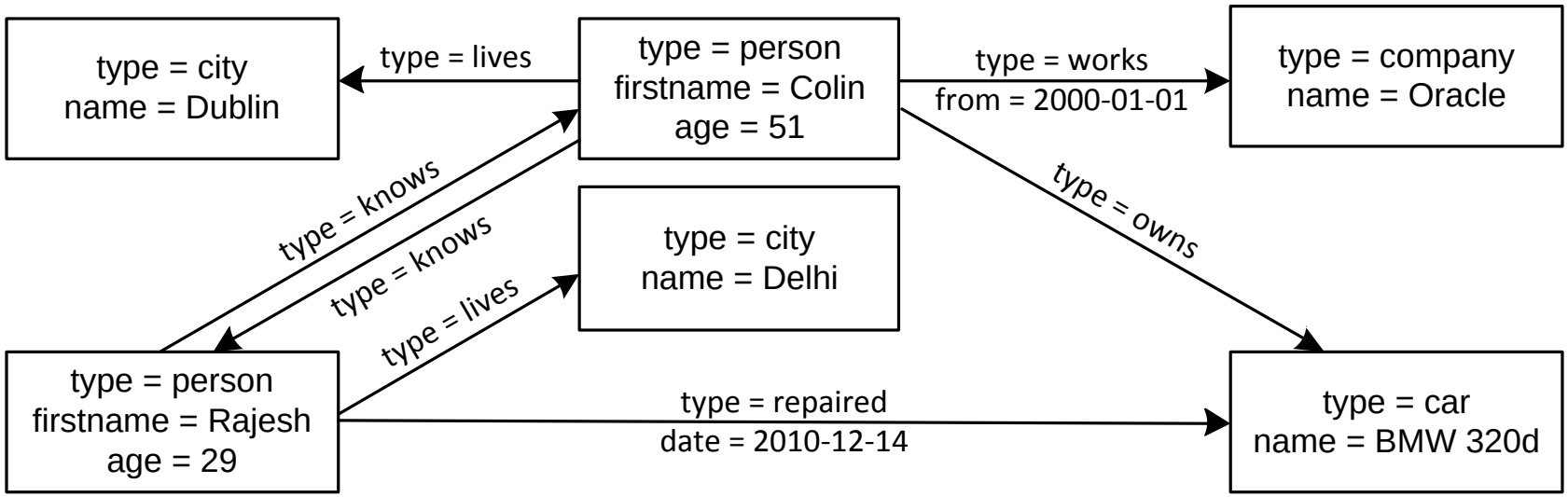
12187	user_ID	text	datetime
	bmestudent	just decomposed a schema to BCNF #db #exam	2011-01-03 07:41:36

Oszloptárolók használata

- Dokumentumtárolókhoz hasonló
  - Naplózás
  - CMS
  - Analitika: Hadoop

Gráfadatbázisok

- G = (V, E) helyett tulajdonsággráfok
- Gráffal kényelmesen reprezentálható adatok



InfiniteGraph
The Distributed Graph Database

Neo4j
the graph database