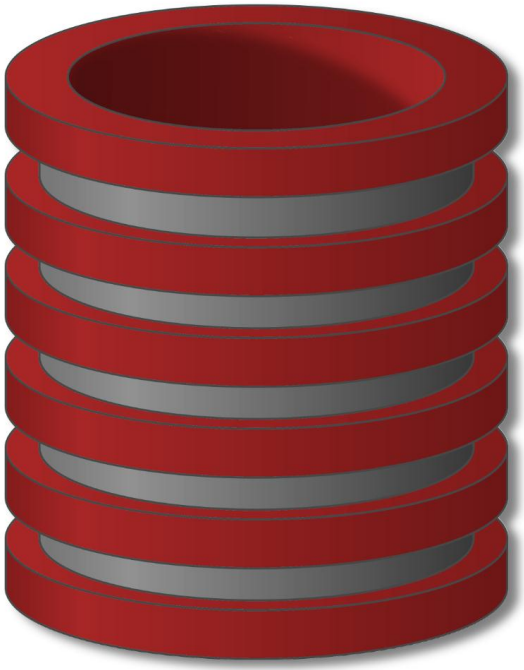


NoSQL Systems

Motivation

NoSQL: The Name

- "SQL" = Traditional relational DBMS
- Recognition over past decade or so:
  Not every data management/analysis problem is best solved using a traditional relational DBMS
- "NoSQL" = "No SQL" =
  Not using traditional relational DBMS
- "No SQL" ≠ Don't use SQL language

NoSQL: The Name

- "SQL" = Traditional relational DBMS
- Recognition over past decade or so:
  Not every data management/analysis problem is best solved using a traditional relational DBMS
  (exclusively)
- "NoSQL" = "No SQL" =
  Not using traditional relational DBMS
- "No SQL" ≠ Don't use SQL language

* "NoSQL" = "Not Only SQL"

Not every data management/analysis problem is best solved using a traditional DBMS

Database Management System (DBMS) provides….

… efficient, reliable, convenient, and safe multi-user storage of and access to massive amounts of persistent data.

Not every data management/analysis problem is best solved using a traditional DBMS

- Convenient
  - simple data model ✓
  - declarative query language ✓
  - transaction guarantees ✓
- Multi-user
- Safe
- Persistent — files OK
- Reliable — redo OK
- Massive +++
- Efficient ++++

NoSQL Systems

Alternative to traditional relational DBMS

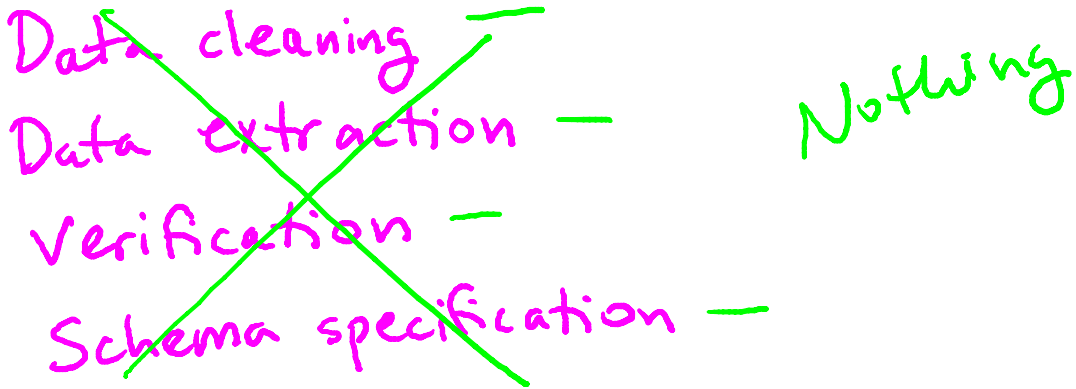
+ Flexible schema
+ Quicker/cheaper to set up
+ Massive scalability
+ Relaxed consistency → higher performance & availability

– No declarative query language → more programming
– Relaxed consistency → fewer guarantees

Example #1: Web log analysis

Each record: UserID, URL, timestamp, additional-info

Task: Load into database system



Example #1: Web log analysis

Each record: UserID, URL, timestamp, additional-info

Task: Find all records for…

- Given UserID ✓
- Given URL ✓
- Given timestamp ✓
- Certain construct appearing in additional-info

No SQL!

Highly Parallelizable

Example #1: Web log analysis

Each record: UserID, URL, timestamp, additional-info

Task: Find all pairs of UserIDs accessing same URL

⋈ Self-Join

Example #1: Web log analysis

Each record: UserID, URL, timestamp, additional-info

Separate records: UserID, name, age, gender, …

Task: Find average age of user accessing given URL

SQL-like

Consistency

Example #2: Social-network graph

Each record: UserID_1, UserID_2

Separate records: UserID, name, age, gender, …

Task: Find all friends of given user

Example #2: Social-network graph

Each record: UserID_1, UserID_2

Separate records: UserID, name, age, gender, …

Task: Find all friends of friends given user

Example #2: Social-network graph

Each record: $UserID_1$, $UserID_2$

Separate records: UserID, name, age, gender, …

Task: Find all women friends of men friends of given user

Example #2: Social-network graph

Each record: $UserID_1$, $UserID_2$

Separate records: UserID, name, age, gender, …

Task: Find all friends of friends of friends of … friends of given user

Not suitable for SQL

× Consistency

Example #3: Wikipedia pages

Large collection of documents ✓

Combination of structured and unstructured data

Task: Retrieve introductory paragraph of all pages about U.S. presidents before 1900

Consistency

NoSQL Systems

Alternative to traditional relational DBMS

+ Flexible schema
+ Quicker/cheaper to set up
+ Massive scalability
+ Relaxed consistency → higher performance & availability

– No declarative query language → more programming
– Relaxed consistency → fewer guarantees