Database 2

NoSQL

- Does not mean No SQL!
- NoSQL - Not Only SQL!

Motivation

- Speed.
  - Flattened data.
  - NO FK (i.e. NO JOINS).
  - De-normalize.
- Flexible DB
  - Unstructured data (kinda).
  - Data can have multiple forms.
- Distributed.
  - Many nodes!

Types

- Document
  - Data is stored in JSON type of documents.
  - Data can be anything.
  - A unique ID to identify this data on the whole machine.
- Column.
  - Rather than store data in rows, data is stored in columns.
  - For example: Given a 'table' with id, fname and lname.
    - Row-oriented systems will store id, fname and lname together.
    - Column-oriented systems will store id-fname separate from id-lname.
- Key-Value.
  - Like a dictionary.
  - key: how you find the data.
  - value: Data to be stored. Can usually be anything!

RDBMS vs. NoSQL

Advantages of RDBMS

- Better for relational Data.

- Normalized.
  - No replication (reduces redundancy).
  - Single point of update***
- Structured (Organized).
  - Specific.
  - Predictable.
- ACID compliant [https://www.thoughtco.com/the-acid-model-1019731]
  - Atomic: All or nothing. If ALL operations don't complete successfully, then revert everything.
  - Consistency: All data written must be valid (Enforcing FK, Constraints, etc.).
    - Deleting***
    - DataTypes
  - Isolation: Multiple transactions cannot happen at the same time on the same data.
    - Row locking.
  - Durability: Ensures that any transaction committed to database will not be lost. Data immediately written to DB.
- Very Stable.
  - Hardened. What does this mean?

Advantages of NoSQL

- Handles Big-Data well.
  - Speed (no-joins).
  - Flattened data.
  - Data replication.
    - Good for speed too.
- No need to specify schema (kinda).
  - Unstructured Data (non-uniform).
- Cheaper to manage.
  - Just add another node!
- Scaling
  - Just add another node!

Examples

- Document (MongoDB, CouchDB, Elastic, Solr)
- Full-Text
  - (Document types) Elastic Search.
  - Solr.
- Column (Apache Cassandra)
- Key-Value (Redis, Couchbase)
- Cache (Redis, Memcached)
- Graph Databases (Neo4J)
  - Data as nodes w/ relationships as edges.

Full-Text searching.

- https://en.wikipedia.org/wiki/Search_engine_(computing)
- https://en.wikipedia.org/wiki/Elasticsearch
- A kinda database.
  - File based.
- It provides a distributed (Scalable), full-text search engine.
- FAST (near real-time search).
- Real time analytics.
- Static (not often changing)

Cache

Why do we need cache?

Motivation

- Speed
  - Memory is faster than most* Hard Disks (and SSDs).
- Large amounts of memory.
- When we have more reads than writes.
- Frequently accessed information. Why?

Persistent

- Data in cache goes away when machine is shut down. Why?

Process

- Make a request for a resource.
- Check if resource is found in cache.
  - If found, return it.
- If not, fetch it from the database.
- Save that copy to the cache.
- Return resource.

Software

- Memcached
  - Can store objects.
  - Cannot manipulate objects.

- Redis
  - Cannot store objects.
  - Can store integers, strings, arrays.
  - Can perform array manipulation
    - push, pop, etc.

Real world Examples

- Files
- Database
- Wikipedia
- Facebook