

# Lecture #13: Databases

*CS106E Spring 2018, Young*

---

*In this lecture we study Databases. We take a close look at SQLite, which we will be using to build websites.*

*In addition, we take a quick look at NoSQL databases, a new class of databases that have some, but not all the properties of traditional databases.*

---

**See the SQL handout. It contains the bulk of information for this lecture.**

## **NoSQL Databases**

- The vast majority of databases in use today are Relational Databases using SQL.
- Recently a new class of databases called NoSQL databases has arisen.

## **The Need for NoSQL Databases**

- There are some problems with traditional relational databases, particularly when it comes to their use with websites.
  - o Traditional relational databases are not designed for distribution.
  - o One important technique for speeding up websites is the use of a **Content Delivery Network (CDN)**
    - A CDN distributes identical copies of our files around the world.
    - Visitors get copies of our files from the closest server on the CDN, instead of having to travel halfway around the world to a central location. This greatly reduces their access time.
    - If a visitor comes to our website from India, we have a copy of our files waiting for them in South Asia.
    - If a visitor comes to our website from Europe, there is a copy of our files in Europe.
  - o Using a traditional relational database forces all visitors to access the database in a central location, reducing the speed of our responses.

## **NoSQL Examples**

NoSQL Databases generally fall into one of two categories

- Most store aggregate chunks of information.
  - o The chunks may be considered as objects, key-value pairs, or documents.
  - o examples include:
    - MongoDB
    - Google's BigTable
    - Amazon's Dynamo
    - Facebook's RocksDB
- However, some NoSQL databases are based on storing information as a graph.
  - o Facebook's TAO is an example of this approach

### Advantages of NoSQL Databases

- NoSQL Databases allow information to be distributed.
- NoSQL Databases, which store information as aggregates, have a straightforward mapping between the database and the programming model.
  - o In contrast, Relational Databases may have information we think of as being closely tied together actually scattered across multiple relationships.
  - o This makes using NoSQL databases easier for programmers with no database experience.

### Disadvantages of NoSQL Databases

- NoSQL Databases may have difficulty getting relational information from across a wide swath of the data in your database.
  - o This makes it difficult to do business analysis with NoSQL databases.
- NoSQL Databases lose some of the key qualities that are important for databases.
  - o NoSQL databases for example, may tradeoff the ability to accurately get the same information consistently across all users.
    - While this may seem critical, it may not be, and may be worth the tradeoff in some situations.
    - If I'm Amazon and someone in India is trying to buy an item, I may have a local copy of the inventory state.
      - This allows me to quickly check if an item is available.
      - However, it also means I have multiple copies of the inventory state, which may briefly become inconsistent with each other.
    - If someone in the US simultaneously buys that same item, I now have a problem.
    - The cost of having to disappoint one of my customers by not having an item I marked as available may not be that high in comparison to the ability to get all customers information quickly.
  - o NoSQL databases may also lose the ability to handle complex transactions as single atomic operations.
    - In a traditional database, if I'm processing a check, and I take the money out of one account and start to put it in another account, and the power goes out before I finish, the database will rollback to the previous state.
      - We think of processing a check as a single **atomic** operation, even though it consists of multiple steps.
    - Some NoSQL databases do not fully support the ability to define atomic operations.

### MongoDB

MongoDB is a widely used NoSQL database which uses JSON (JavaScript Object Notation).

- We'll be studying JavaScript next week.
- JavaScript objects do not necessarily have predefined types or instance variables. For example, I could define the three following objects:

```
{
  name: "Patrick",
  units: 12
}
```

```
{
  name: "Flo",
  units: 16,
  dorm: "Gavilan"
}

{
  name: "Molly",
  units: 4,
  species: "Canine"
}
```

- Note that there is no consistency across these objects. They each have different properties.
  - o This is in contrast with traditional class systems, where the properties are fixed when I create the class.
  - o This is also in contrast with relational databases, where I would define a table with pre-defined columns to store this data.