SQL vs NoSQL

SQL

- Structured **Query** Language
  - Used to query databases that have structure
  - Could also be used for CSV files, spreadsheets etc.
- Closely tied to RDBMS - relational databases
  - Columns / Fields
  - Tables of data hold relationships
  - All entries in a table **must** have same set of columns
- Tabular databases
  - Efficient indexing possible - use specified columns
  - Storage efficiency: prior knowledge of data size

Problem with tabular databases

- Structure (good? bad?)
- All rows in table must have same set of columns

Example

- Student - hostel => mess
- Student - day-scholar => gate pass for vehicle
- Table? Column for mess, column for gate pass???

Alternate ways to store: Document databases

- Free-form (unstructured) documents
  - Typically JSON encoded
  - Still structured, but each document has own structure
- Examples:
  - MongoDB
  - Amazon DocumentDB


```
[
    {
        "year" : 2013,
        "title" : "Turn It Down, Or Else!",
        "info" : {
            "directors" : [ "Alice Smith", "Bob Jones"],
            "release_date" : "2013-01-18T00:00:00Z",
            "rating" : 6.2,
            "genres" : ["Comedy", "Drama"],
            "image_url" : "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP
            "plot" : "A rock band plays their music at high volumes, annoying the neighb
            "actors" : ["David Matthewman", "Jonathan G. Neff"]
        }
    },
    {
        "year": 2015,
        "title": "The Big New Movie",
        "info": {
            "plot": "Nothing happens at all.",
            "rating": 0
        }
    }
]
```

Alternate ways to store: Key-Value

- Python dictionary, C++ OrderedMap etc.: dictionary/hash table
- Map a key to a value
- Store using search trees or hash tables
- Very efficient key lookup, not good for range type queries
- Examples:
  - Redis
  - BerkeleyDB
  - memcached …
- Often used alongside other databases for “in-memory” fast queries

Alternate ways to store: Column stores

- Traditional relational DBs store all values of a row together on disk
  - Retrieving all entries of a given row very fast
- Instead store all entries in a column together
  - Retrieve all values of a given attribute (age, place of birth, …) very fast
- Examples:
  - Cassandra
  - HBase...

Alternate ways to store: Graphs

- Friend-of-a-friend, social networks, maps: graph oriented relationships
- Different degrees (number of outgoing edges), weights of edges, nodes etc.
- Path-finding more important than just search
  - Connections, knowledge discovery
- Examples:
  - Neo4J
  - Amazon Neptune

Alternative ways to store: Time Series Databases

- Very application specific: store some metric or values as function of time
- Used for log analysis, performance analysis, monitoring
- Queries:
  - How many hits between T1 and T2?
  - Average number of requests per second?
  - Country from where maximum requests came in past 7 days?
- Typical RDBMS completely unsuitable - same for most alternatives
- Examples:
  - RRDTool
  - InfluxDB
  - Prometheus
- Search: elasticsearch, grafana,...

NoSQL?

- Started out as “alternative” to SQL
- But SQL is just a query language - can be adapted for any kind of query, including from a document store or graph!
- “Not-only-SQL”
- Additional query patterns for other types of data stores

A word on ACID

- Transaction: core principle of database
- ACID:
  - Atomic
  - Consistent
  - Isolated
  - Durable
- Many NoSQL databases sacrifice some part of ACID (example: eventual consistency instead of consistency) for performance
- But there can be ACID compliant NoSQL databases as well...

Why not ACID?

- Consistency hard to meet: especially when scaling / distributing
- Eventual consistency easier to meet
- Example:
  - A (located in India) and B (located in the US) both add C as a friend on Facebook
  - Order of adding does not matter!
  - Temporarily seeing C in A's list but not B, or B's list but not A - not a catastrophe (?)
- Financial transactions **absolutely require** ACID
  - Consistency is paramount - even a split second of inconsistent data can cause problems

A word on storage

- In-memory:
  - Fast
  - Doesn’t scale across machines
- Disk
  - Different data structures, organization needed