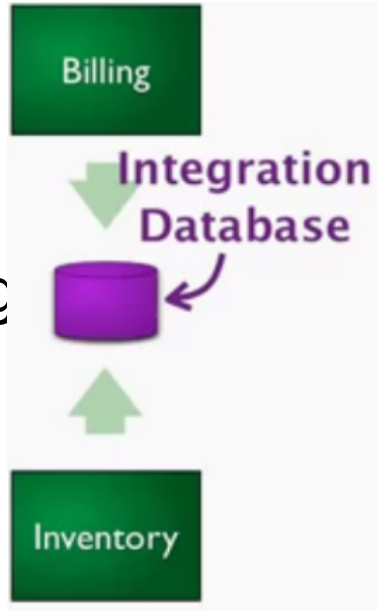
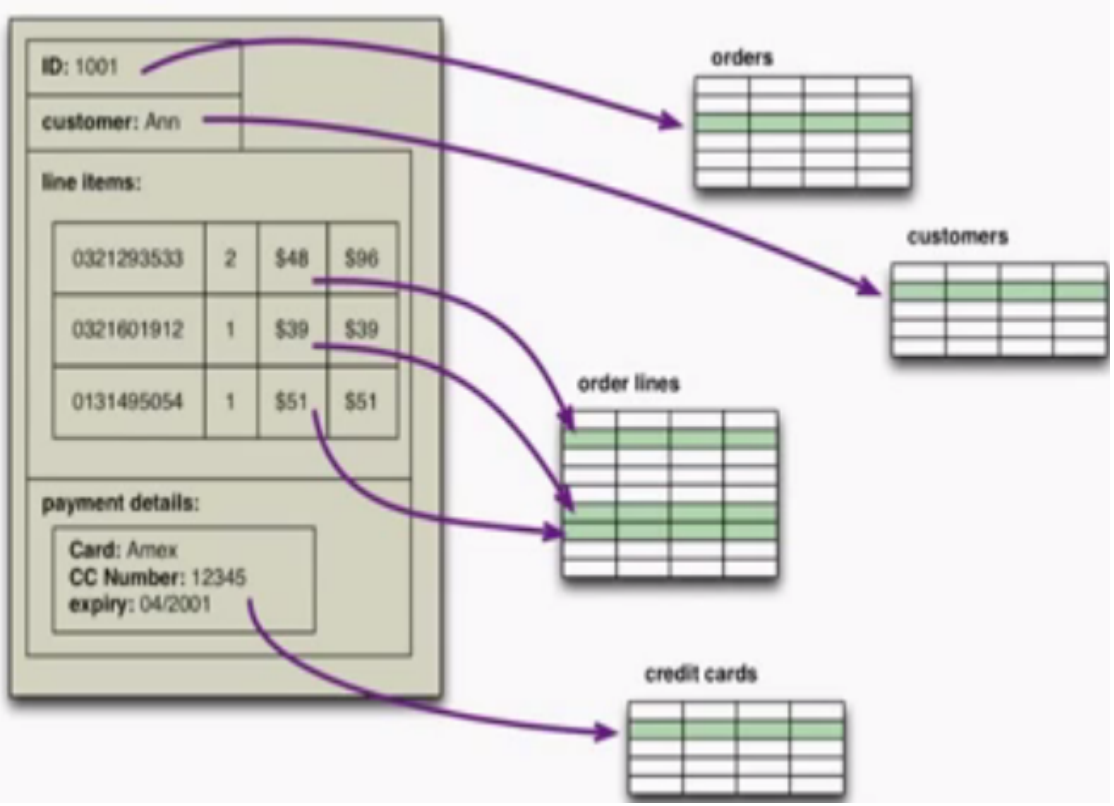
INTRODUCTION TO NOSQL DATABASES

Background

- Rise of RDBMS Why
  - Persistence, Integration, SQL, Tx Mgt, Reporting
  - Rise of OO Database
    - Don’t really picked…
    - RDBMS still dominant technolog
  - Now What?



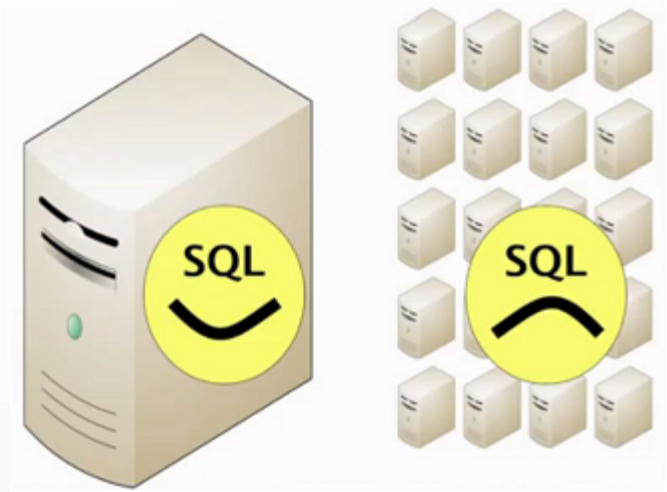
Problem with RDBMS?



- We have to manually map Requirement to different tables..
- It would be great if order can be stored as one Unit
- Impedance Mismatch=> ORM tools

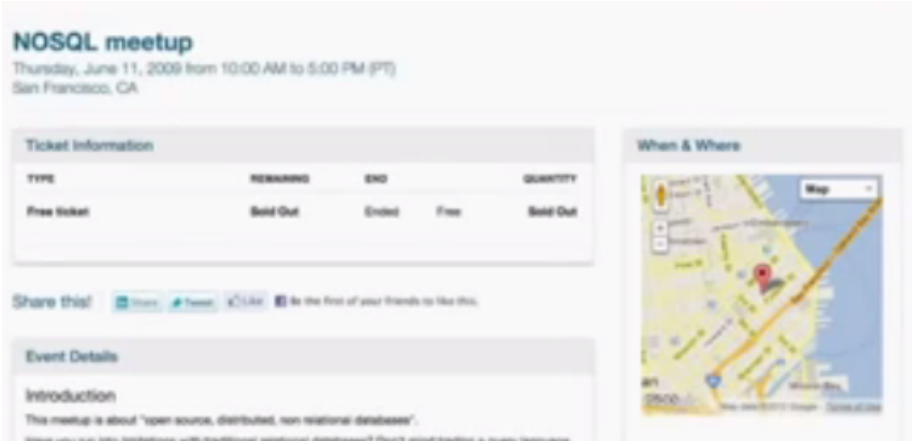
Why No SQL

- Modern web application/ cloud create huge traffic
- Vertical Scale vs Horizontal scaling
- Horizontal scaling/ Hadoop is the solution=> RDBMS can not scale
- RDBMS is designed to run over big server rather then community H/W
- Running Relational DBMS over cluster is not easy!
- **Industry comes with new solutions**



Why NoSQL named NoSQL?

- John an big data guys propose meet up for alternative solution to RDBMS
- Need twitter hashtag #nosql



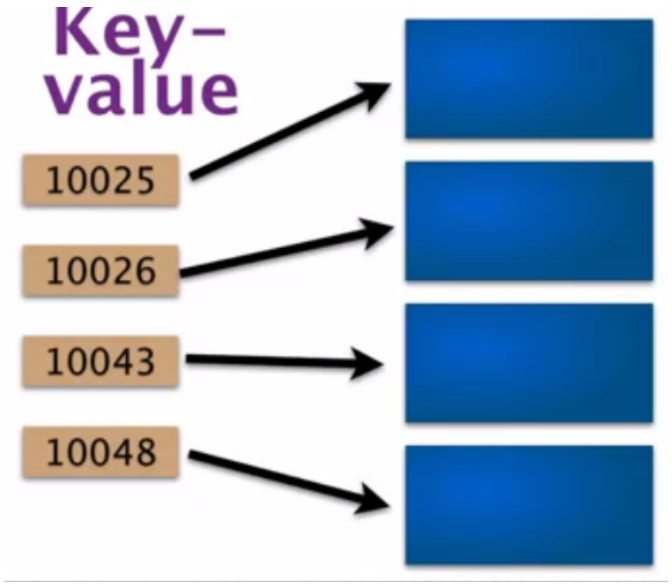
Categories of NoSQL

- Document
  - MongoDB, Couch DB
- Key value
  - Riak, Redis
- Column family
  - Cassandra, Hbase
- Graph
  - Neo4j

Characteristics of NoSQL

- Non Relational model
- Open Source
- Cluster friendly
- 21st century web
- Schema less design

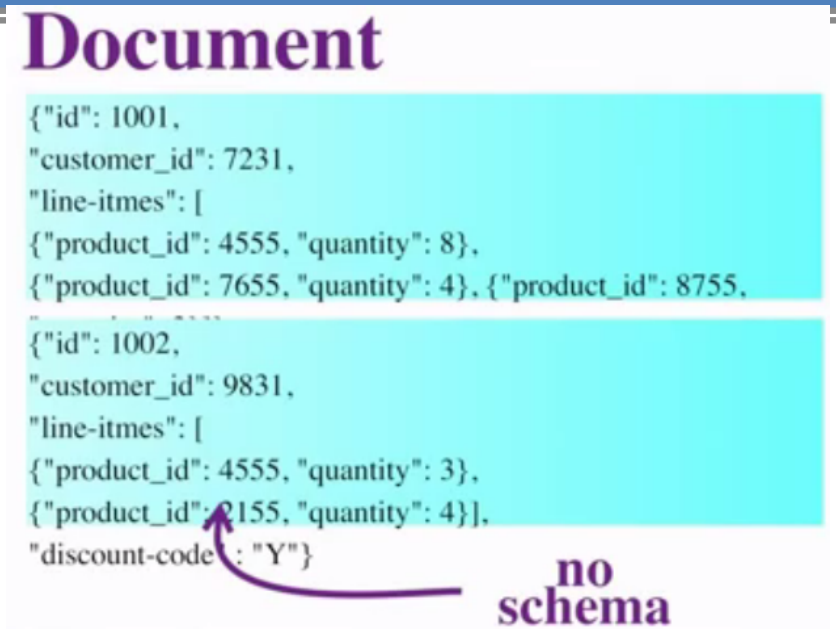
Key-value store



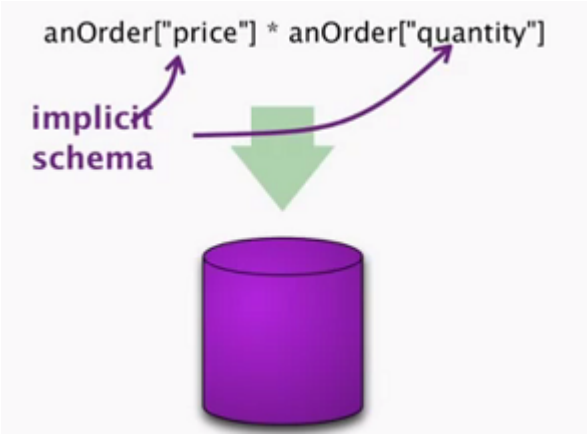
- Simplest model
- Key have no idea what is actually stored against a key, it could be doc/image/Blob data
- Aka hashMap
- Table==Bucket

- Where should not be used?
  - Need relationship
  - Need Tx
  - ACID
  - Quaries based on value eg: price>=200
- Use cases
  - Storing session information
  - User profile /Perference
  - Shopping cart

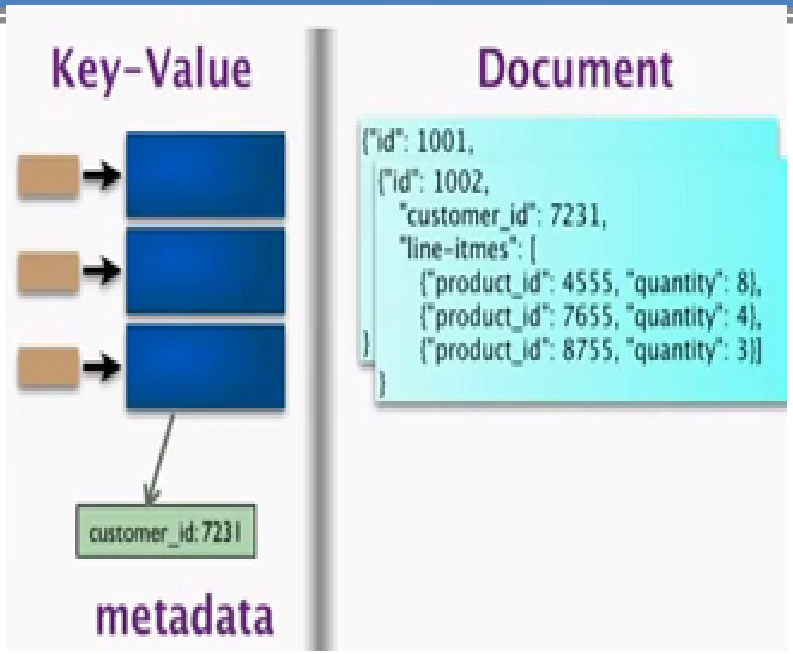
Document Data Model

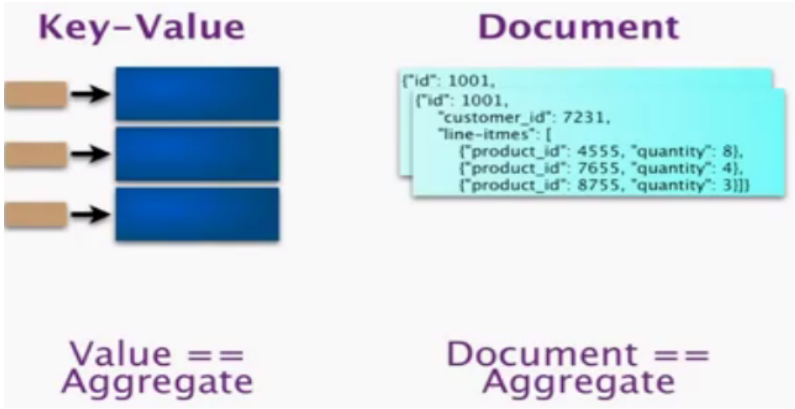


- Key -> Document (JSON)
- Flexible
- Can quarries to doc
- More transparent then key-value
- Provide implicit schema



Key value vs. Document





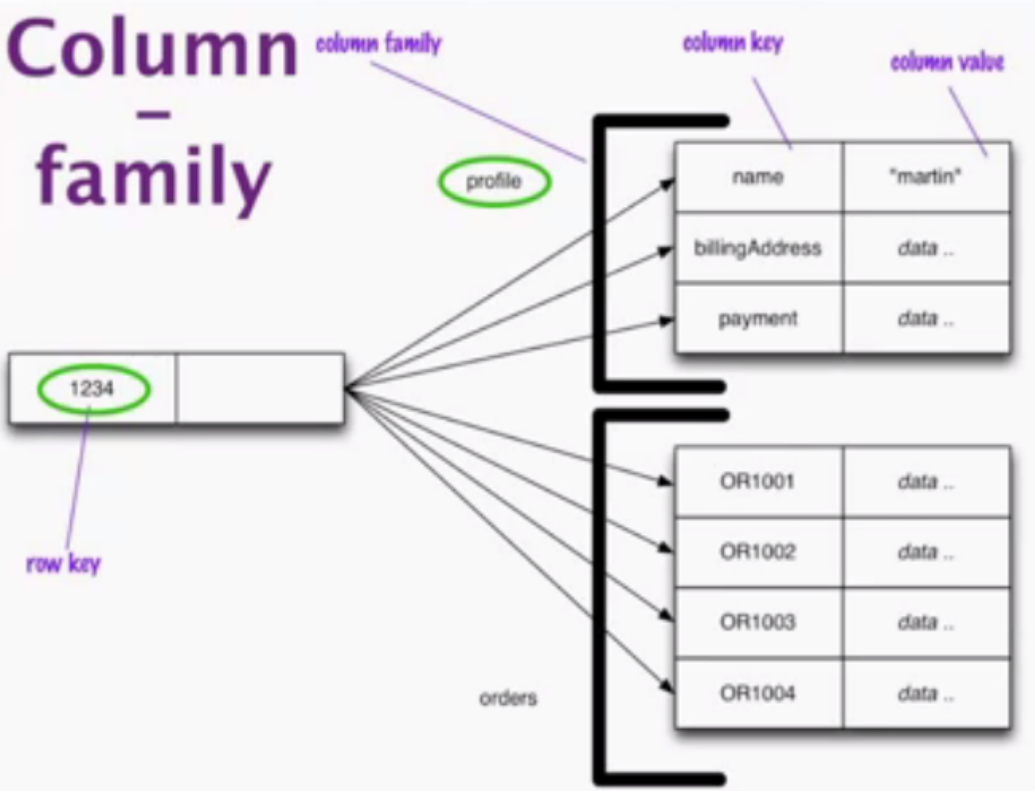
- Difference is very small
- Many key value database allowed use to have meta data like customer_id=7321
- Index is allowed

Collectively called Aggregate oriented database

Aggregate database

- Order is consider as a single unit that should be store in DB
- Very useful in case of distributed cluster: Each aggregate can be stored in an node without join etc, Only one node need to refer to access one order details.

Column family DB

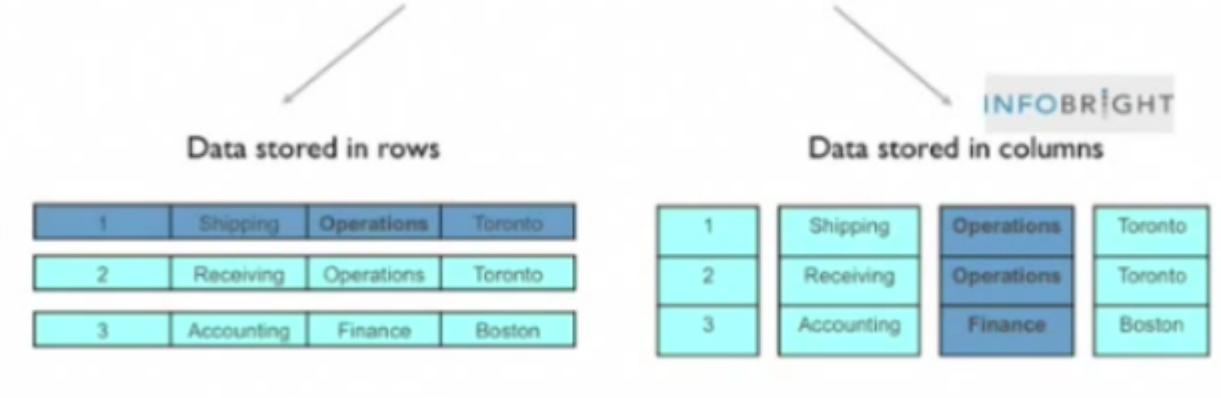


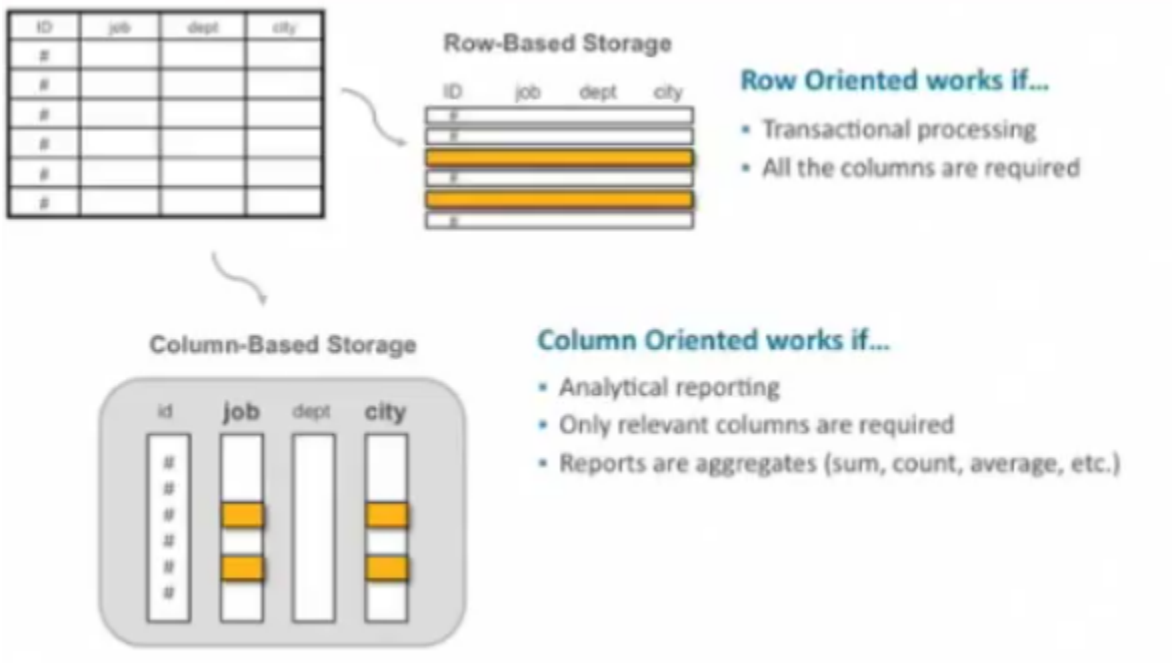
- Aka Aggregate oriented DB
- More Complicated then earlier model
- Within one “Row key” we can store multiple column family, where each column family is combination of column that fit togather
- Row Key+ column family name aggregate
- Easily retrieval of infidel elements

RDBMS vs Column db

Employee_ID	Job	Dept	City
1	Shipping	Operations	Toronto
2	Receiving	Operations	Toronto
3	Accounting	Finance	Boston

Employee_ID	*Job*	*Dept*	*City*
1	Shipping	Operations	Toronto
2	Receiving	Operations	Toronto
3	Accounting	Finance	Boston




ID
job
dept
city
Row-Based Storage
ID
job
dept
city
Row Oriented works if...
• Transactional processing
• All the columns are required
Column-Based Storage
id
job
dept
city
Column Oriented works if...
• Analytical reporting
• Only relevant columns are required
• Reports are aggregates (sum, count, average, etc.)