



1

# NoSQL

Alpesh Vasant, G Academy

APACHE  
**HBASE**



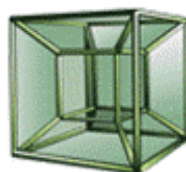
*Cassandra*



**CouchDB**  
relax



**riak**



**mongoDB**

**HYPERTABLE** INC



**Neo4j**



**redis**

# Introduction To NoSQL

- ▶ NO use of SQL or....
- ▶ NO use of ONLY SQL
- ▶ Database specially useful to handle.....
  - ▶ New data with very rapid pace
  - ▶ Large amount of incoming data
  - ▶ Highly unstructured data like data resulting due to ...
    - ▶ Images, GIFs
    - ▶ Videos
    - ▶ Emojis
    - ▶ Chat Session
    - ▶ Text Comments
    - ▶ Likes/Dislikes
    - ▶ User Data (like number of currently logged in user)
    - ▶ Device Data (coming from hardware)

# Introduction To NoSQL Cont..

- Usually non related and de normalized data (row format data)
- Dynamic Schemas , no schema definition required at the start
- Generic data model (sets, maps & arrays)-defining explicit data type not necessary
- Dynamic type discovery and conversion
- Highly distributable across multiple Nodes
- Commodity hardware – Adding more economical hardware at later stages (when needed) is possible
- Sharding - Large databases are partitioned into small, faster and easily manageable databases
- Replication - Auto data replication is also supported in NoSQL databases by default. Hence, if one DB server goes down, data is restored using its copy created on another server in network

# Introduction To NoSQL Cont..

- No Join Support (Means High Scalability)
- No Constraint Support
- No Complex Transaction Support (like Joins)
- No use of SQL

# Difference between a RDBMS and a NoSQL database

RDBMS	NoSQL
1) <b>Upfront Schema Definition</b> needed	1) <b>No upfront Schema Definition</b> needed
2) <b>Explicit Data Type</b> needs to declared for each column of data	2) <b>Explicit data type</b> declaration is <b>not needed</b>
3) Supports the <b>use of JOINS</b>	3) Does <b>not support the use of JOINS</b>
4) Use of <b>Constraints like Primary Key, Foreign Key</b> for data validation	4) There is <b>no use of any type of Constraints</b> for data validation
5) <b>Use of SQL</b> by all RDBMS for handling the data of the table	5) There is <b>no use of SQL</b> as each NoSQL DBMS has its own language for Query



# Understanding the Storage Architecture

➤ Based on the way data is stored in NoSQL, majorly there are four types of databases....

1. Document Database
2. Graph Store
3. Wide Column Store
4. Key Value Stores

# Understanding the Storage Architecture

## 1. Document Database

- Data stored in form of key-value pair further stored in document (Record)
- Each document may stores different key-value pair or key-array pairs, or even nested documents
- Each Document may contains diverse & heterogeneous fields
- Collection = Table (of RDBMS)
- Popular fields in the document can be indexed to provide fast retrieval without knowing the key
- Supports Embedded documents
- MongoDB, Amazon Simple DB, Apache CouchDB, DynamoDB
- **Consumes more space when compared with other NoSQL types**



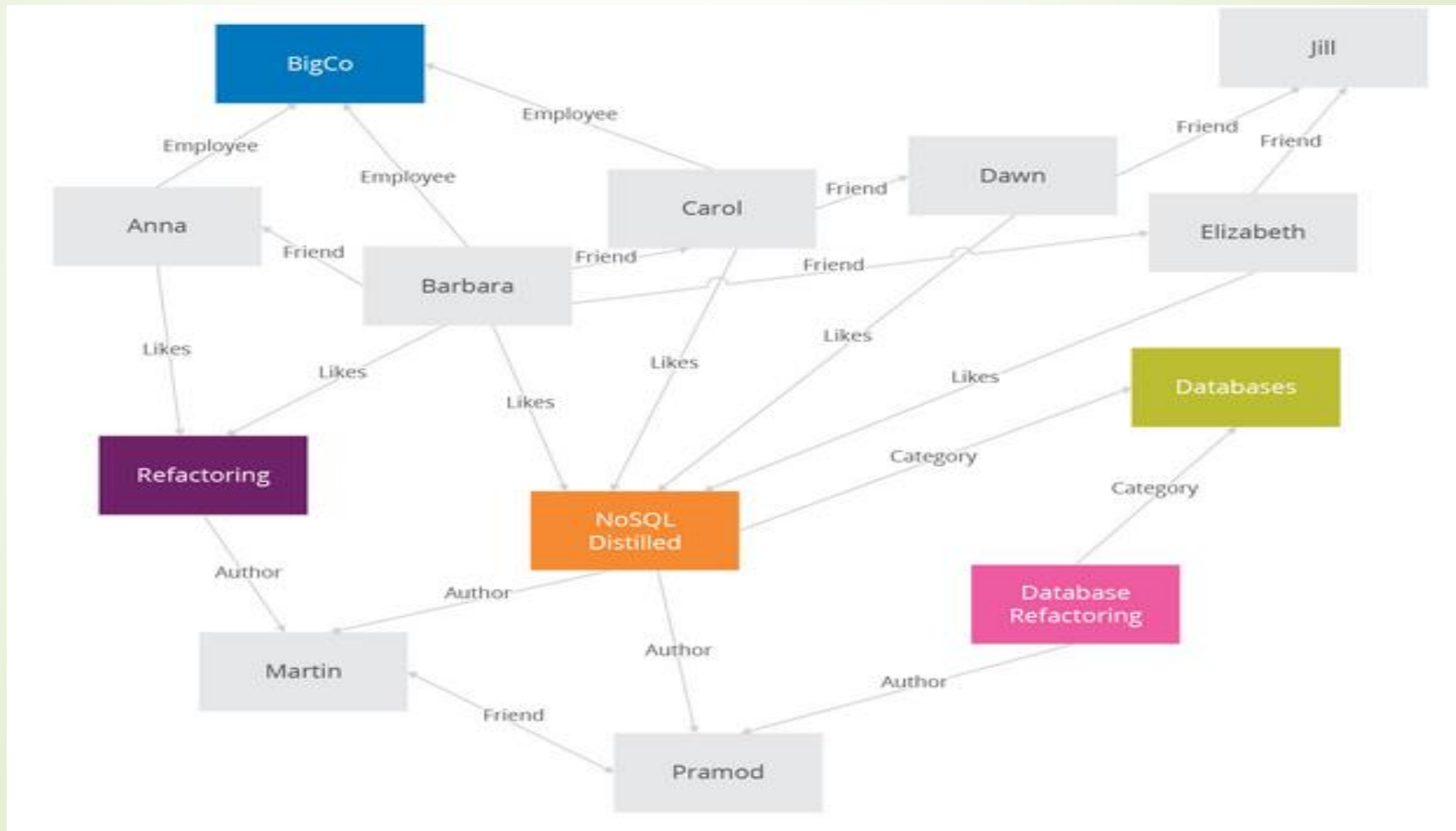
# Understanding the Storage Architecture

## 2. Graph Store

- More suitable for data having lots of relationship (specially social network data)
- Stores highly interconnected data
- Data stored in the form of Node & Edges(relationship)
- Neo4J, Infinite Graph, OrientDB, FlockDB
- ***Not suited with too much heterogeneous data or not for all sort of problem***

# Understanding the Storage Architecture

## 2. Graph Store



# Understanding the Storage Architecture

## 3. Wide Column Store

- Stores columns of data together (instead of rows)
- Often columns are group together logically and called column family
- Excellent for lookup on single field eg. Analytical data
- Cassandra (used in Facebook) ,Hbase, Hypertable
- ***Lookups on other fields not supported***

# Understanding the Storage Architecture

## 4. Key Value Store

- Stores data in the form of Key-Value Pairs in a hash table
- Unique key as a pointer to value
- Logical group of keys => “buckets”
- DynamoDB, Aerospike, Berkeley DB, Couchbase
- **Updating the part of data is challenge**