

# NOSQL - Not Only SQL

20 November 2025 09:15

Why can't we use RDBMS in Big Data ?

- Volume, Variety, evolving Schema, price
- Not cluster Friendly --> Doesn't support distributed computing
- Design ( everything should be normalized )
- RDBMS cannot have evolving schema (means we cannot repeatedly add more and more columns with time )
- To solve these big data problem, companies created their own NOSQL databases secretly around 2005-7

Google - Big Table

Amazon - DynamoDB

Facebook - Cassandra

Azure - CosmosDB

LinkedIn - Voldemort

- HBASE is default NOSQL database in Hadoop ( nobody uses HBASE now )
- HBASE is only available in Hadoop.

Properties of NOSQL :

1. Non-relational
2. Cluster Friendly (distributed computing)
3. Schema-less
4. Open-source mostly

NOSQL avoids :

1. Overhead of ACID transactions
2. Complex queries
3. Upfront schema

NOSQL provides :

1. Easy and frequent changes to the DB
2. Horizontal scaling
3. Solution to impedance mismatch
4. Fast development
5. Denormalization

Categories of NOSQL databases

1. Key-value	--> supports aggregation	stored in key-value	ex. DynamoDB
2. Document	--> supports aggregation	stored in JSON docs	ex. MongoDB
3. Column family		group of columns	ex. Cassandra and HBASE
4. Graph		form of graphs	ex. Neo4j

Column family data model

CAP theorem :

- Just like RDBMS has ACID , NOSQL has CAP

1. Consistency - All clients have same view of data.
2. Availability - All clients can read and write anytime.
3. Partition Tolerance - System is functional in-spite of network partition. (Most IMP)
  - Applicable to distributed systems.

Theorem : No distributed system can achieve all the 3 properties at same time.

- Partition tolerance is most important, if we ignore it, the system becomes RDBMS.
- We have to choose CP or AP
- CP ex. BookMyShow
- AP ex. IRCTC tatkal ticket
- Cassandra can give all 3 CAP, it uses consistency tuning

Indexing : assigning indexes to make retrieval faster.

- MongoDB has GeoSpatial Indexing
  - Based on latitude and longitude we can search.  
(ex. Xyz near me, within 5 kms.. etc)

Exercise on MongoDB

Use compass, connect to localhost, start shell.  
OR

Open cmd, type command : mongosh

Open MongoDB\_Basics.txt file, run commands.

Command : show dbs;

Sometimes MongoDB will not show empty databases;

Copy content from persons\_collection\_data.txt, paste into mongo, even if the collection persons is not present, mongodb will create automatically.

Open file CRUD\_operations.txt run commands.

MongoDB query has 2 things :

1. Selection : what is the condition (ex. .... Where "gender" = "F") {gender:"F"}
2. Projection : what columns we want to output ( ex. Select name,age ) {gender : 1} means show only gender column

Ex. Db.persons.find({ "gender": "F"}, { gender:1, yearOfBirth:1});

Ex. Db.persons.find({ "gender": "F"}, { gender:1, yearOfBirth:1, \_id:0});

0 means False, non-zero means True.

Ex. Db.persons.find({ "gender": "F"}, { gender:1,"name.first":1, \_id:0});

Ex. Db.persons.find( {}, { gender:1, yearOfBirth:1}); all records' gender and yearOfBirth

Practice from 03\_mongo\_more\_exercises.txt

Previously mongoDB used to have MapReduce.

Studied about airbnb sample dataset on mongoDb Atlas (cloud database) accessed using mongoDB compass.