Welcome!

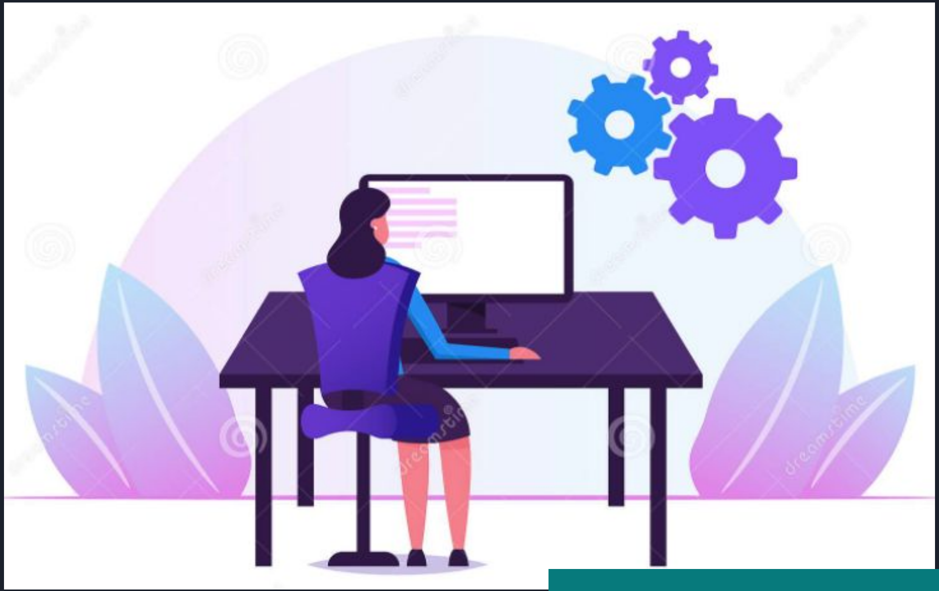
- We'll start in a moment :)
- We are NOT recording tonight's event. We may plan to take screenshots for social media.
    - ***If you want to remain anonymous***, change your name & keep video off.
- We'll introduce the hosts and break in-between for Q/A.
- We will make some time for Q&A at the end of the presentation as well.
- You can come prepared with questions. And, feel free to take notes.
- Online event best practices:
    - Don't multitask. Distractions reduce your ability to remember concepts.
    - Mute yourself when you aren't talking.
    - We want the session to be interactive.
    - Feel free to unmute and ask questions in the middle of the presentation.
    - Turn on your video if you feel comfortable.
    - Disclaimer: Speaker doesn't knows everything!

Check out:

- Technical Tracks and Digital Events
- Get updates – join the Digital mailing list
- Give us your feedback – take the Survey

Copyright (c) 2022, Harini Rajendran

WOMEN WHO CODE



WWCode Digital + **Backend Study Group**

February 3, 2022

Copyright (c) 2022, Harini Rajendran

WOMEN WHO CODE

Backend Study Group

- Welcome from WWCode!
- Our mission: Inspiring women to excel in technology careers.
- Our vision: A world where women are representative as technical executives, founders, VCs, board members and software engineers.

Copyright (c) 2022, Harini Rajendran

Harini Rajendran
Software Engineer, Confluent

https://www.linkedin.com/in/hrajendran/

Prachi Shah
Director, Women Who Code San Francisco

https://www.linkedin.com/in/prachisshah/

Copyright (c) 2022, Harini Rajendran

WOMEN WHO CODE

SQL vs NoSQL Databases

The war of databases!!

Copyright (c) 2022, Harini Rajendran

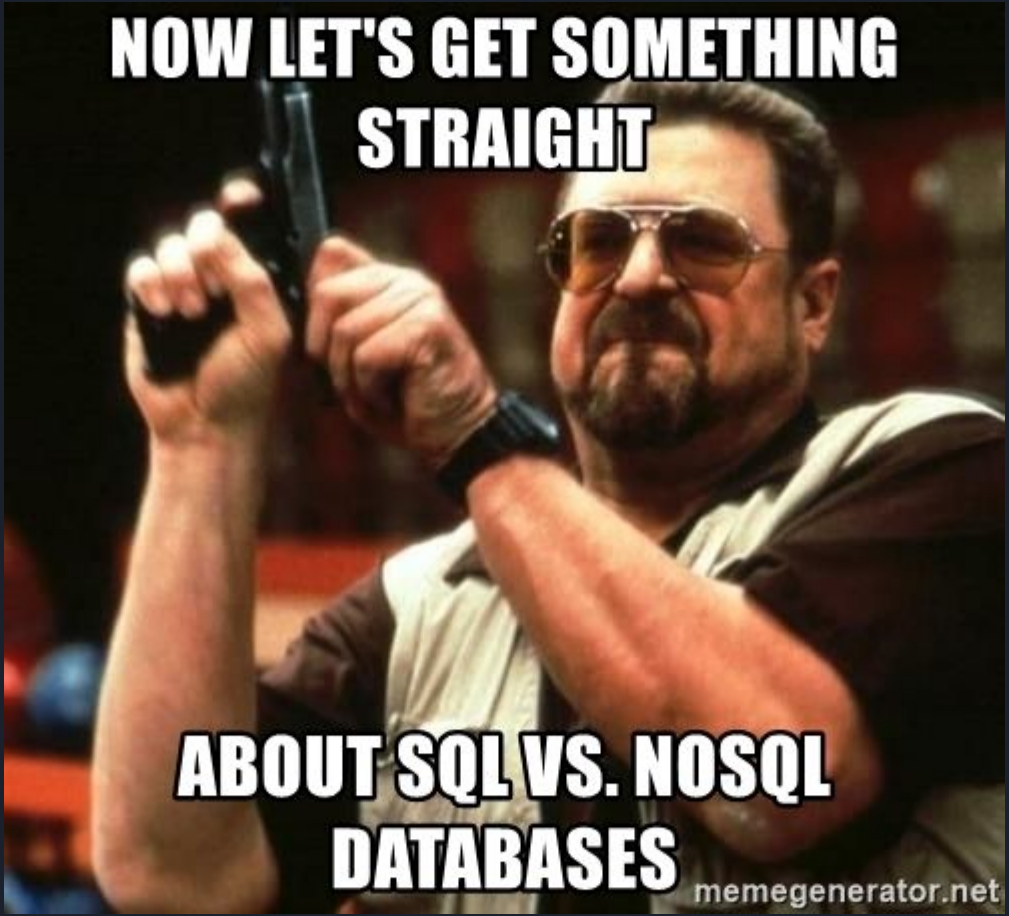

NOW LET'S GET SOMETHING STRAIGHT
ABOUT SQL VS. NOSQL DATABASES
memegenerator.net

Image Credit

Agenda

What is SQL Database

What is NoSQL Database

Pros and cons of both

Types of NoSQL databases

When to use what

Q & A

Copyright (c) 2022, Harini Rajendran

What is SQL database

- Relational Database Management System
- Data is stored in tabular format organized as rows and columns
- Tables have relationship between them
- Data model is rigid
- Supports data normalization
- Supports JOIN between tables
- Supports ACID properties

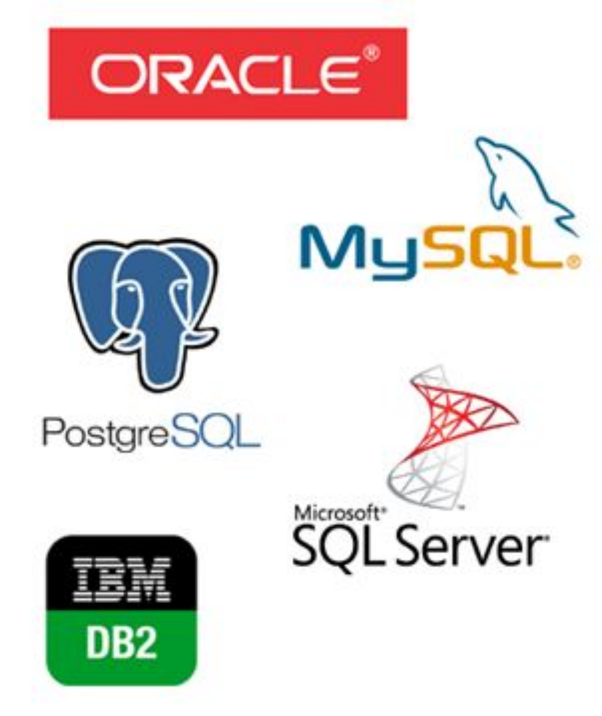


Image Credit

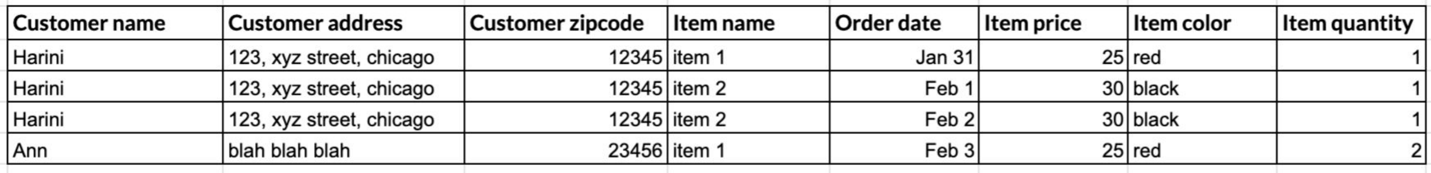
Copyright (c) 2022, Harini Rajendran

Explaining some terms

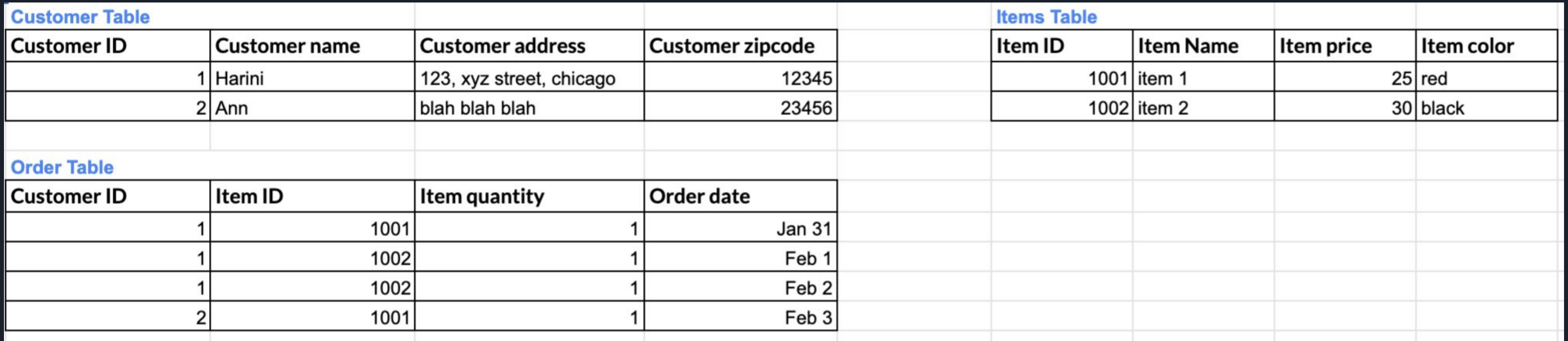
Rigid data model

CREATE TABLE People (PersonID int, LastName varchar(255), FirstName varchar(255), Address varchar(255), City varchar(255));

Data normalization and JOINS



Customer name	Customer address	Customer zipcode	Item name	Order date	Item price	Item color	Item quantity
Harini	123, xyz street, chicago	12345	item 1	Jan 31	25	red	1
Harini	123, xyz street, chicago	12345	item 2	Feb 1	30	black	1
Harini	123, xyz street, chicago	12345	item 2	Feb 2	30	black	1
Ann	blah blah blah	23456	item 1	Feb 3	25	red	2



Customer Table

Customer ID	Customer name	Customer address	Customer zipcode
1	Harini	123, xyz street, chicago	12345
2	Ann	blah blah blah	23456

Items Table

Item ID	Item Name	Item price	Item color
1001	item 1	25	red
1002	item 2	30	black

Order Table

Customer ID	Item ID	Item quantity	Order date
1	1001	1	Jan 31
1	1002	1	Feb 1
1	1002	1	Feb 2
2	1001	1	Feb 3

Copyright (c) 2022, Harini

WOMEN WHO CODE

ACID properties

Atomicity

All changes to data and transactions are executed completely and as a single operation. If that isn’t possible, none of the changes are performed. It’s all or nothing.

Consistency

The data must be valid and consistent at the start and end of a transaction.

Isolation

Multiple transactions can occur without stepping on each other.

Durability

When a transaction is completed, its associated data is permanent and cannot be changed.

Copyright (c) 2022, Harini

What is NoSQL database

- Data is stored in various formats: JSON document, key-value pairs, graphs (nodes and edges), etc
- Data has no relationship between them
- No data normalization
- Data model is flexible
- Cannot JOIN between data
- Suitable for huge volume of data

mongoDB
APACHE HBASE
redis
cassandra
amazon DynamoDB
neo4j
CouchDB relax

Image Credit

Copyright (c) 2022, Harini Rajendran

Pros and Cons of SQL

Pros

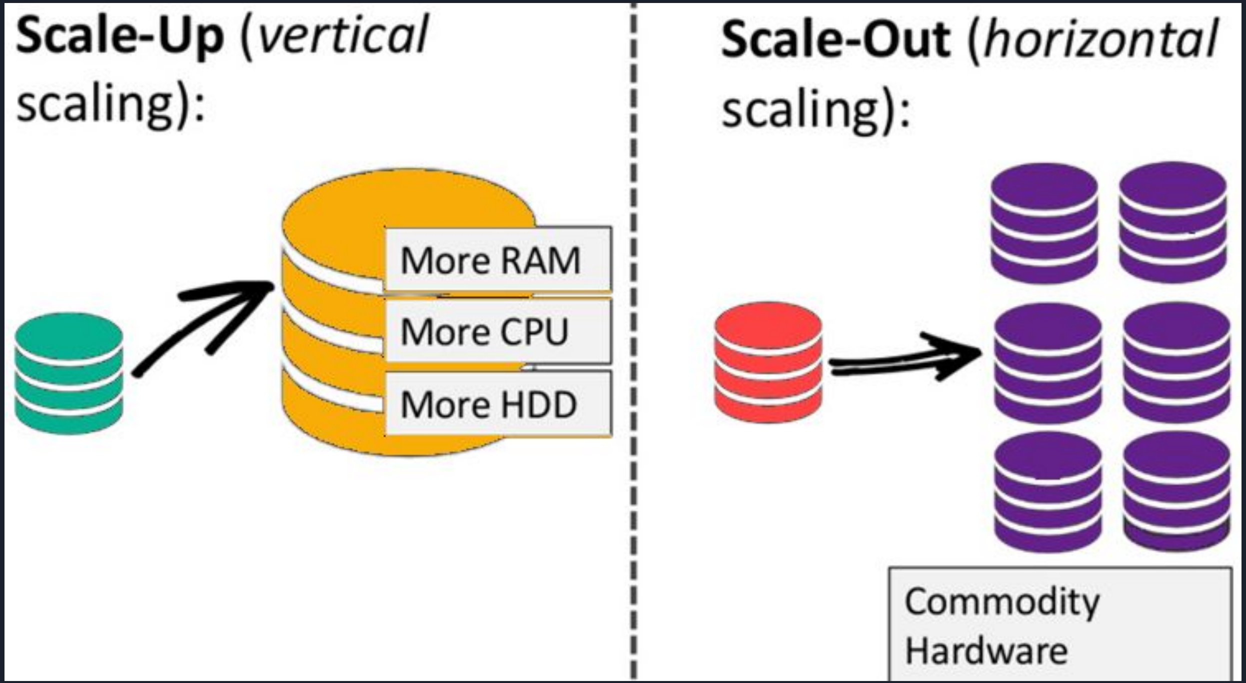
- No extensive coding is needed
- Almost everyone is familiar with basic SQL commands
- Standardized data schema (Can be a con too)
- High demand in the industry. Large user community
- Supports ACID properties

Cons

- Rigid data schema. No way to store unstructured data
- Data normalization can make querying slow
- Cannot scale horizontally. Can only be scaled vertically.
- Hardware maintenance is expensive

Copyright (c) 2022, Harini Rajendran

Scaling SQL vs NoSQL databases



Scale-Up (*vertical* scaling):
More RAM
More CPU
More HDD
Scale-Out (*horizontal* scaling):
Commodity Hardware

Image Credit

Copyright (c) 2022, Harini Rajendran

Sql Database scaling challenges

Image Credit

Pros and Cons of NoSQL

Pros

- Scale out architecture. Just keep adding more nodes. No single point of failure
- Less hardware management. Usually cheap hardware is used
- Flexibility of storing unstructured data
- Can store massive amounts of data and is typically used in the big data world
- Data is denormalized. No JOINs needed. So faster queries
- Very agile - great flexibility and adaptability

Cons

- No ACID properties support - only eventual consistency of data (Speed and availability takes preference)
- Smaller user community
- Each one is different. Different data models, ways of access, etc
- So many different options are available in the market. You need to know when to use what
- Learning curve is stiff for new developers

Copyright (c) 2022, Harini Rajendran

The CAP Theorem

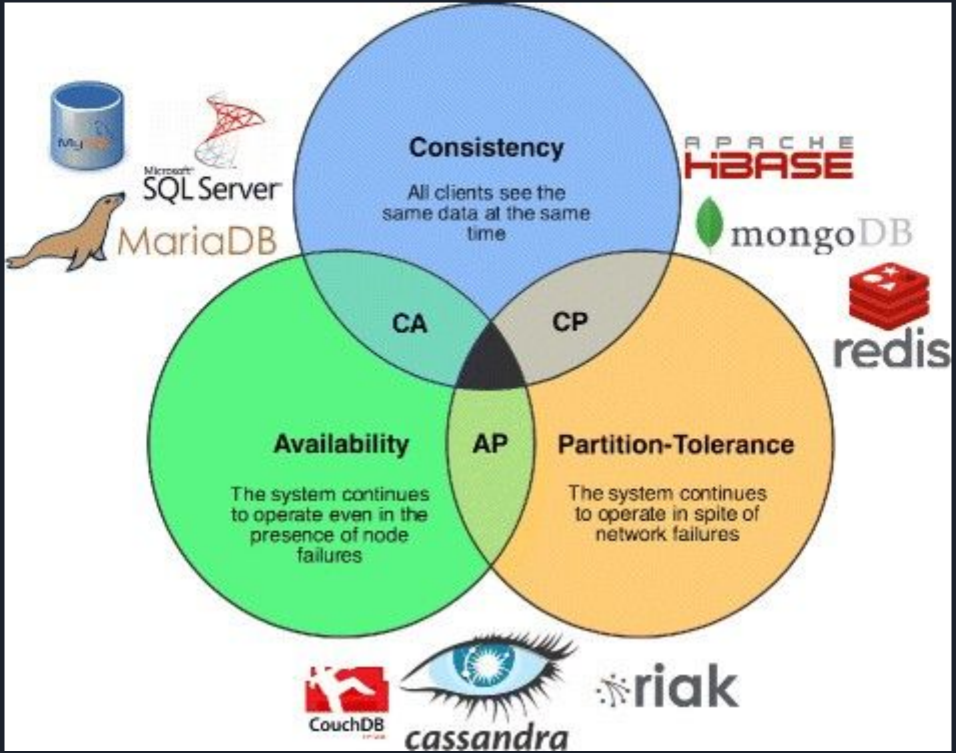

Consistency
All clients see the same data at the same time
CA
CP
AP
Availability
The system continues to operate even in the presence of node failures
Partition-Tolerance
The system continues to operate in spite of network failures
MySQL
Microsoft SQL Server
MariaDB
APACHE HBASE
mongoDB
redis
CouchDB
cassandra
riak

Image Credit

Copyright (c) 2022, Harini

Types of NoSQL databases

Key-value based

Data is stored in the form of key value pairs. Like a hash table. Keys are usually String. Values can be string, json, xml, etc. Example: DynamoDB, Redis

Graph based

Data is stored in the form of graphs. Each entity forms the nodes in the graph and the relationship between the entities forms the edges in the graph. Example: Neo4j, Orient DB

Column oriented

These databases work on columns known as column families. Each column is stored separately in a contiguous fashion. Example: HBase, Cassandra

Document oriented

Data is stored as key value pairs where the value is stored as document. Values are usually of JSON or XML format which the database understands and are queryable. Example: MongoDB, CouchDB, ElasticSearch

Copyright (c) 2022, Harini Rajendran

SQL vs NoSQL

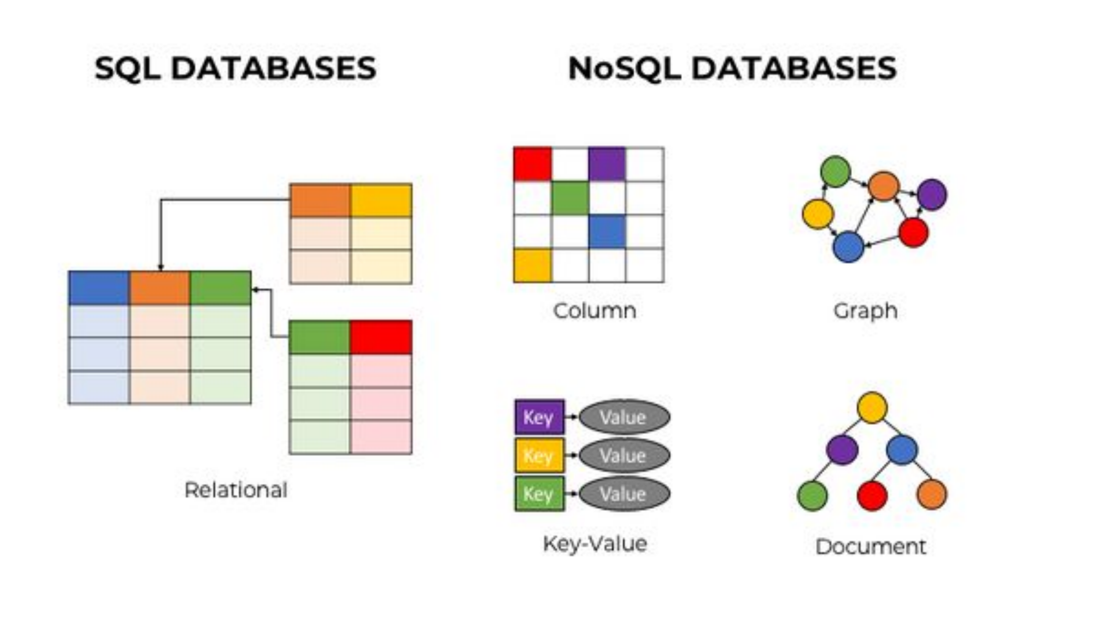
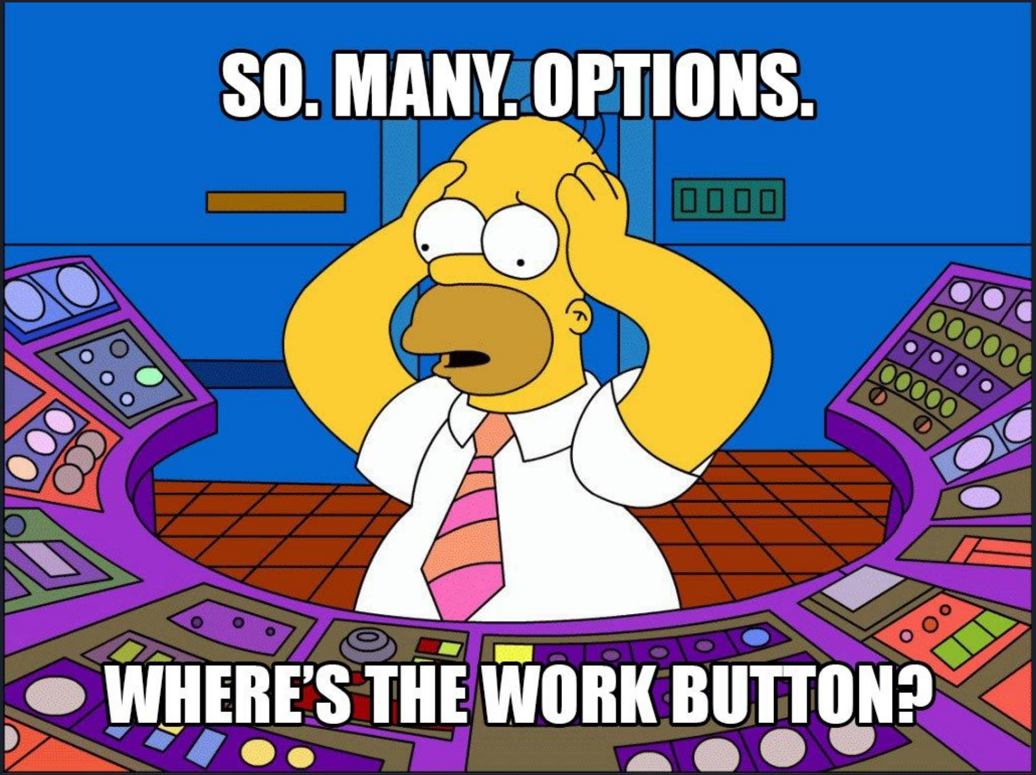


Image Credit

Copyright (c) 2022, Harini Rajendran

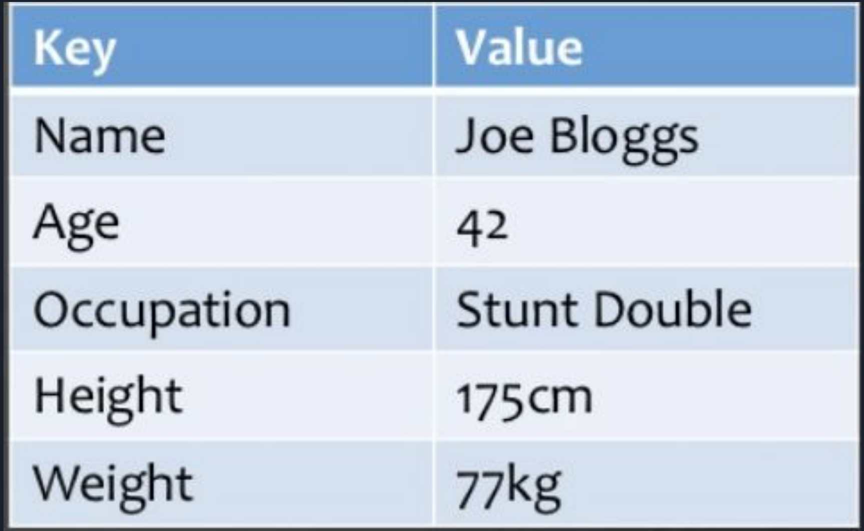
When to use what?!?!



Copyright (c) 2022, Harini Rajendran

Key Value based

- One of the basic NoSQL databases
- Data is stored in key/value pairs
- Designed to handle lots of data and heavy load
- Fast query performance owing to its simplicity
- Huge volume of simple queries
- Best fit where data is not highly related
- Based on Amazon's Dynamo paper



Key	Value
Name	Joe Bloggs
Age	42
Occupation	Stunt Double
Height	175cm
Weight	77kg

Sample use cases

- Shopping cart details - holiday season of online giants
- Storing session information

Image Credit

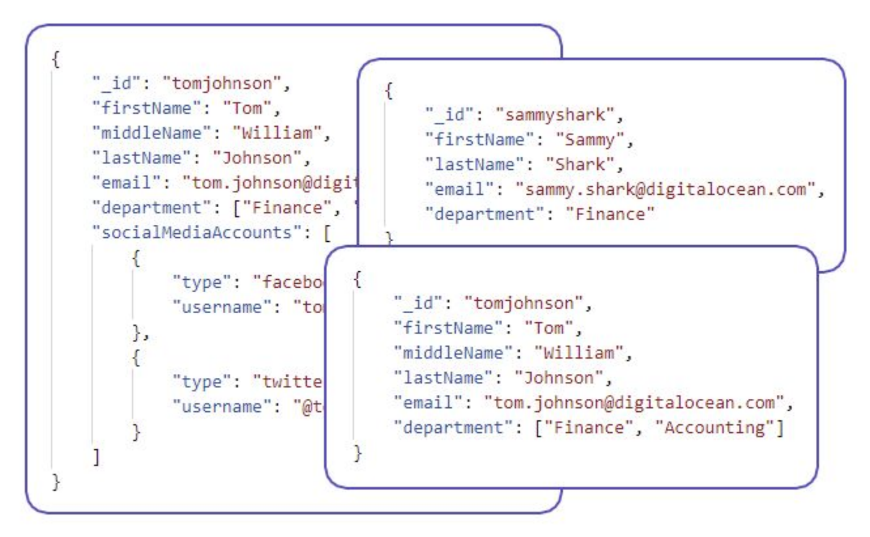
Copyright (c) 2022, Harini Rajendran

Document based

- Extension of kv based where value is stored as document
- Schema free
- Doesn't support relation
- Can query fields in the document
- Supports all uses case of kv store and more

Sample use cases

- Storing user profiles. Each document can store different attributes and values.
- Content Management systems like blogging platforms



```
{
    "_id": "tomjohnson",
    "firstName": "Tom",
    "middleName": "William",
    "lastName": "Johnson",
    "email": "tom.johnson@digi
    "department": ["Finance", 
    "socialMediaAccounts": [
        {
            "type": "facebo
            "username": "to
        },
        {
            "type": "twitte
            "username": "@t
        }
    ]
}
```

```
{
    "_id": "sammyshark",
    "firstName": "Sammy",
    "lastName": "Shark",
    "email": "sammy.shark@digitalocean.com",
    "department": "Finance"
}
```

```
{
    "_id": "tomjohnson",
    "firstName": "Tom",
    "middleName": "William",
    "lastName": "Johnson",
    "email": "tom.johnson@digitalocean.com",
    "department": ["Finance", "Accounting"]
}
```

Image Credit

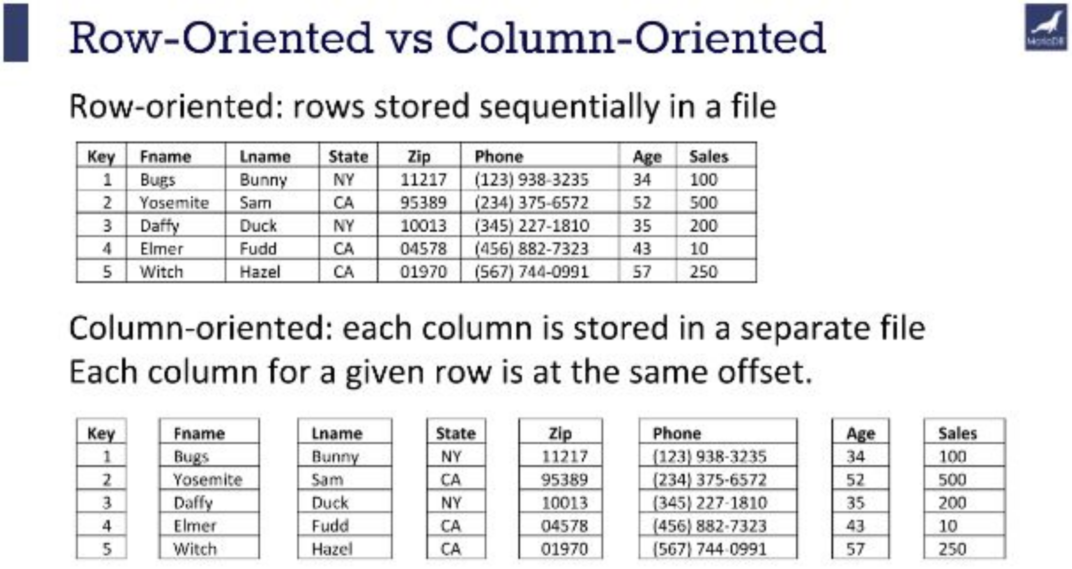
Copyright (c) 2022, Harini Rajendran

Column based

- Data is stored column by column
- Aggregation like SUM, COUNT, etc on columns are blazing fast
- Used in large scale applications where query pattern is on few columns at a time

Sample use cases

- Reporting system - For eg, sales over 2 yrs across all stores
- Business Intelligence
- Manage data warehouses
- Big data applications



Row-Oriented vs Column-Oriented

Row-oriented: rows stored sequentially in a file

Key	Fname	Lname	State	Zip	Phone	Age	Sales
1	Bugs	Bunny	NY	11217	(123) 938-3235	34	100
2	Yosemite	Sam	CA	95389	(234) 375-6572	52	500
3	Daffy	Duck	NY	10013	(345) 227-1810	35	200
4	Elmer	Fudd	CA	04578	(456) 882-7323	43	10
5	Witch	Hazel	CA	01970	(567) 744-0991	57	250

Column-oriented: each column is stored in a separate file
Each column for a given row is at the same offset.

Key
1
2
3
4
5

Fname
Bugs
Yosemite
Daffy
Elmer
Witch

Lname
Bunny
Sam
Duck
Fudd
Hazel

State
NY
CA
NY
CA
CA

Zip
11217
95389
10013
04578
01970

Phone
(123) 938-3235
(234) 375-6572
(345) 227-1810
(456) 882-7323
(567) 744-0991

Age
34
52
35
43
57

Sales
100
500
200
10
250

Image Credit

Copyright (c) 2022, Harini Rajendran

Graph based

- Data is stored as nodes and relationship between data as edges
- Every node is connected to its adjacent nodes
- Optimized for traversing through connected data.
- Mostly used to build and model networks

Sample use cases

- Social networking sites
- Logistics
- Supply chain management
- Recommendation Engine
- Smart home networks

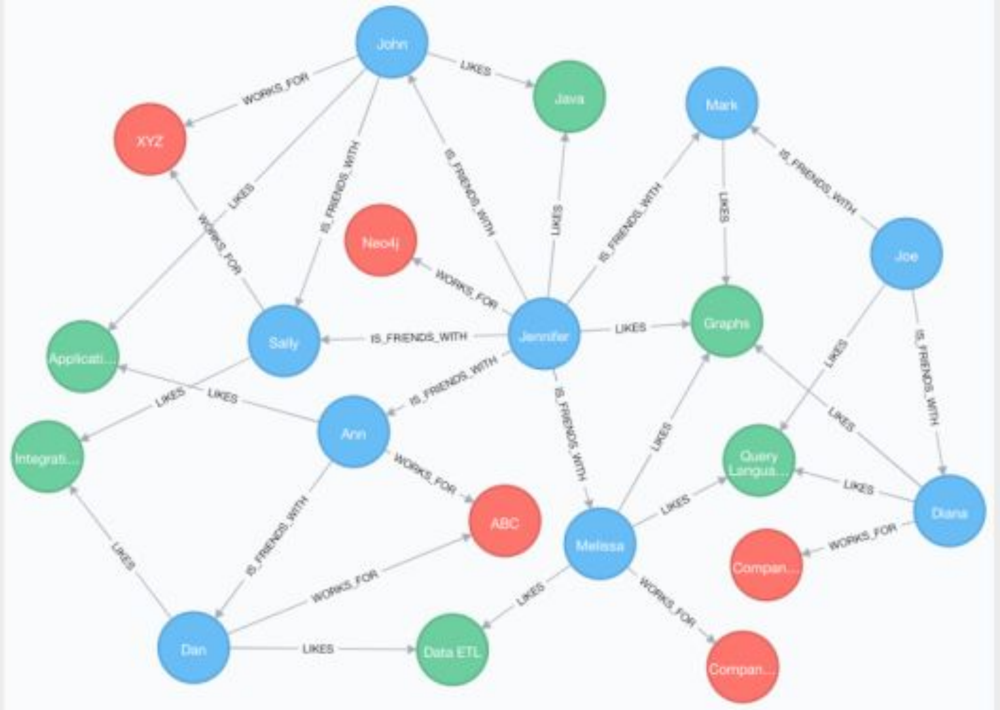


Image Credit

Copyright (c) 2022, Harini Rajendran

Backend Study Group

WWCode Slack Handle: Harini Rajendran

https://www.linkedin.com/in/hrajendran/

Resources and References:

- https://www.datastax.com/blog/sql-vs-nosql-pros-cons
- https://www.interviewbit.com/blog/sql-vs-nosql/
- https://www.mongodb.com/nosql-explained/nosql-vs-sql
- https://www.guru99.com/nosql-tutorial.html
- https://medium.com/system-design-blog/cap-theorem-1455ce5fc0a0
- https://hub.packtpub.com/different-types-of-nosql-databases-and-when-to-use-them/
- https://www.educba.com/types-of-nosql-databases/

Backend Study Group:

• Presentations and session recordings found here: WWCode YouTube channel

You can unmute and talk or use the chat.

Copyright (c) 2022, Harini Rajendran