Non-relational Data Management Systems

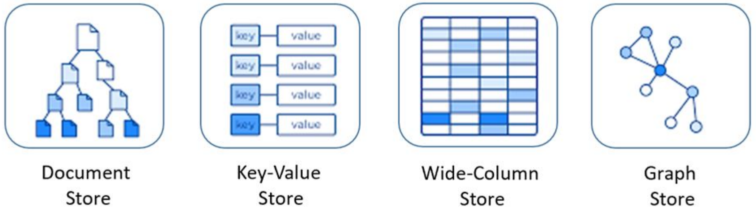
CS4440 Technology Presentation

Dhruv Sharma, Karen Sun, Joanita Nandaula, Joo Lim

Georgia Tech

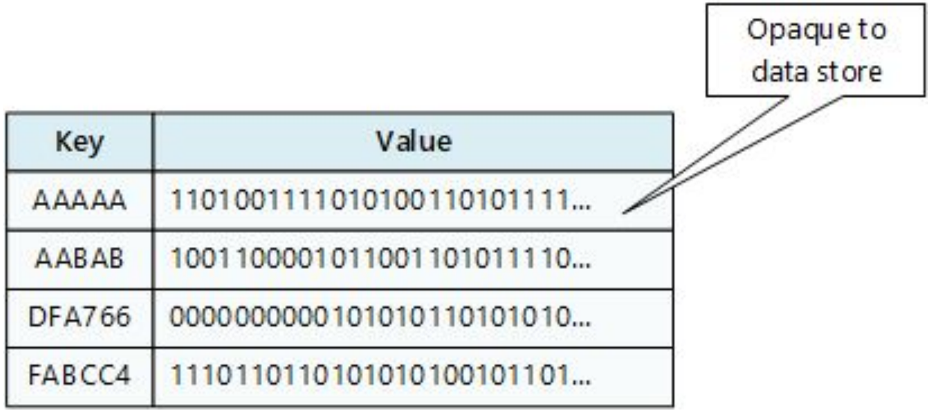
What is non-relational database?

- Database that does not use the relational models
- NoSQL ("Not only SQL") refers to data stores that do not use SQL for queries
- Four major categories of NoSQL:
    - Document Store
    - Key-Value Store
    - Wide-Column Store
    - Graph Store



Key-Value Store

- Large hash table structure
- Associates each data value with an unique key
- Optimized for simple lookups, less suitable for querying across multiple tables
- Requires overwriting entire value for updates
- Use Cases: Storing session information, user profiles, preferences, shopping cart data

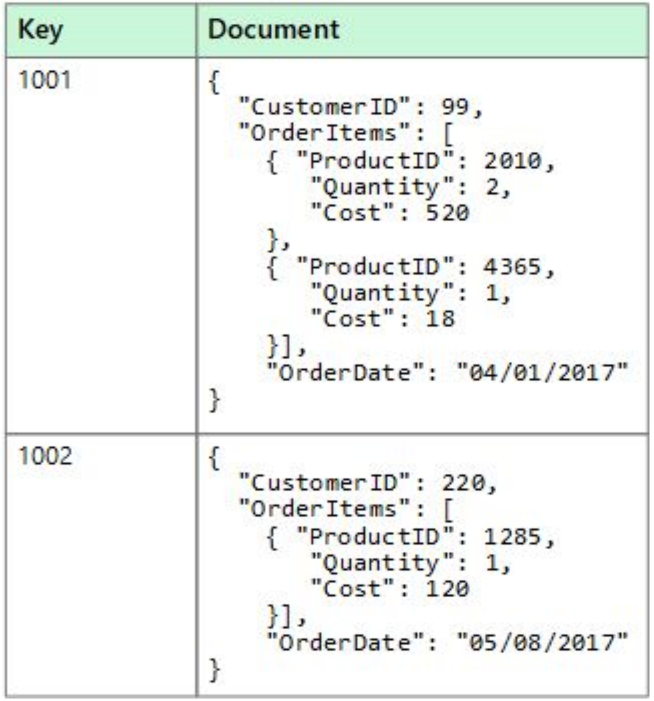


Key	Value
AAAAA	1101001111010100110101111...
AABAB	1001100001011001101011110...
DFA766	0000000000101010110101010...
FABCC4	1110110110101010100101101...

https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data

Document Store

- Extension of Key-Value, but Value as Document
- Each field value = scalar item
  - Number
  - Compound element
  - Parent-child collection
- Encoded in various ways
  - XML, YAML, JSON, BSON, plain text, etc..
- Contains entire data for an entity
- Supports in-place updates
- Keys are often hashed
  - Let DB create VS use an unique attribute as key
- Use Case: Content management systems, blogging platforms, web analytics, real-time analytics, e-commerce applications

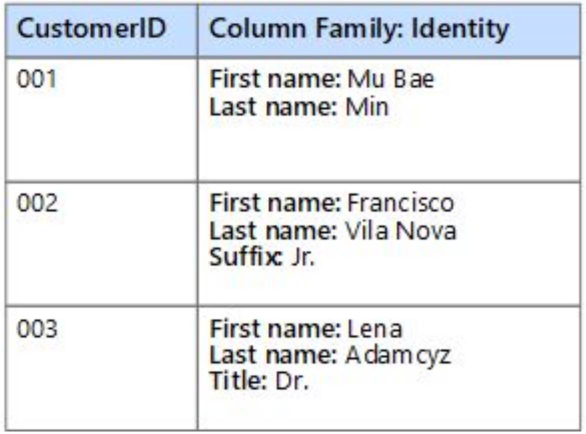


Key	Document
1001	`{ "CustomerID": 99, "OrderItems": [ { "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }`
1002	`{ "CustomerID": 220, "OrderItems": [ { "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }`

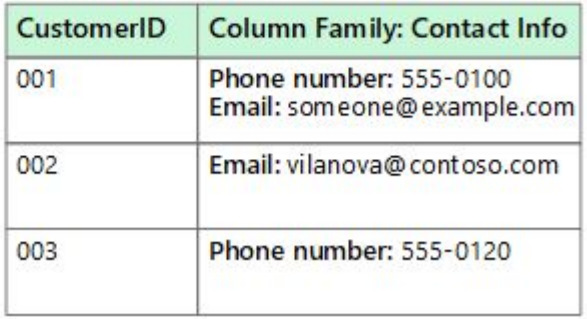
https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data

Wide-Column Store

- Stores data as column families
- Optimized for fast retrieval of columns of data
- Reduces overall disk I/O and the amount of data need to load from disk
- Use Cases: Content management systems, blogging platforms, log aggregation



CustomerID	Column Family: Identity
001	**First name:** Mu Bae **Last name:** Min
002	**First name:** Francisco **Last name:** Vila Nova **Suffix:** Jr.
003	**First name:** Lena **Last name:** Adamcyz **Title:** Dr.

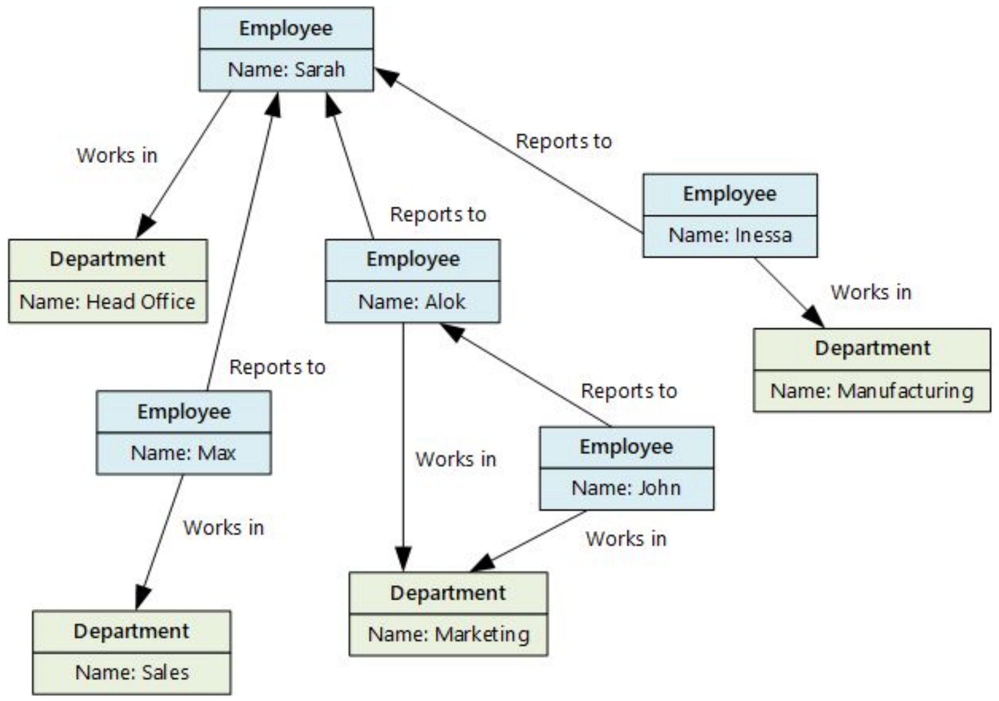


CustomerID	Column Family: Contact Info
001	**Phone number:** 555-0100 **Email:** someone@example.com
002	**Email:** vilanova@contoso.com
003	**Phone number:** 555-0120

https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data

Graph Store

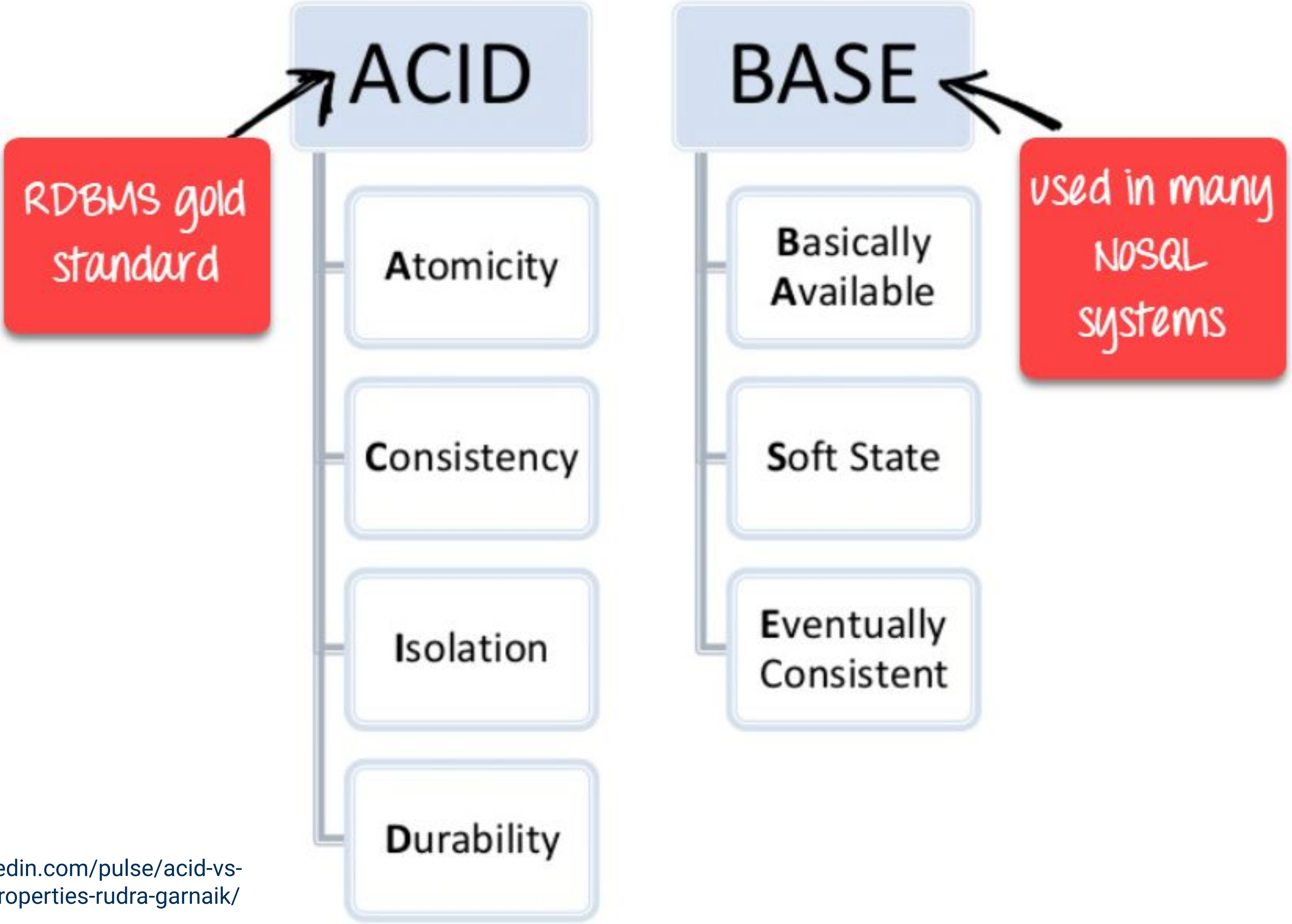
- Nodes and edges: entities and relationships
- Perform queries by traversing the network of nodes and edges
- Use Cases: Fraud detection, recommendation engines, route optimization, pattern discovery



https://learn.microsoft.com/en-us/azure/architecture/data-guide/big-data/non-relational-data

Why NoSQL?

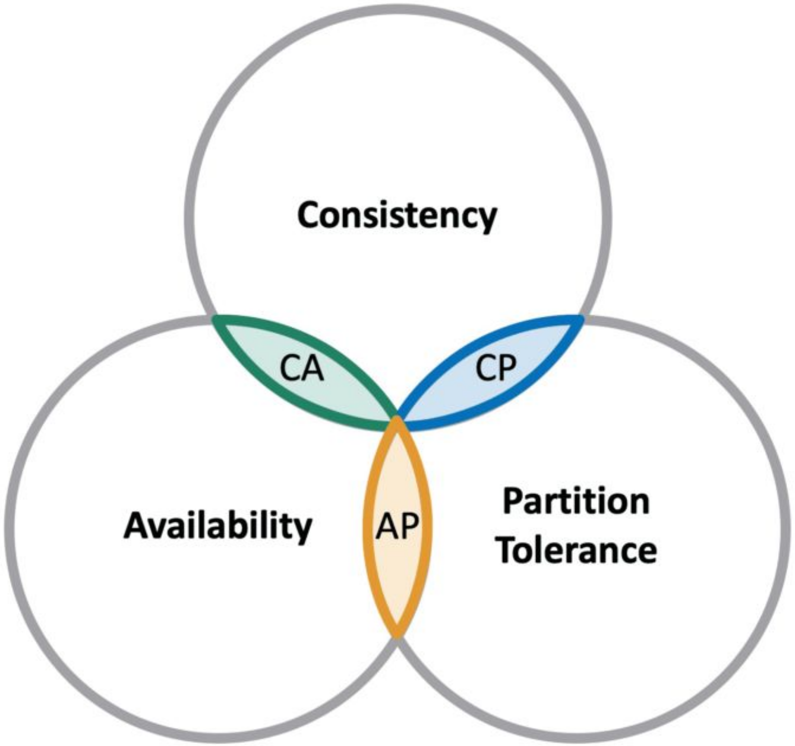
- Internet paved the way for Big Data
  - Social networks, high traffics
- Let’s upgrade the RDBMS server
  - Scale up / Vertical scaling
    - (memory, CPU, disk) ↑
    - $$$$$
  - Scale out / Horizontal scaling / Sharding
    - High latency / Low throughput
    - Updating columns / Schema changes
    - Expensive JOIN operations
    - Network congestions
- Let’s shift the priority
  - ACID vs BASE
  - CAP Theorem
  - Low latency / high throughput



https://www.linkedin.com/pulse/acid-vs-base-database-properties-rudra-garnaik/

Georgia Tech

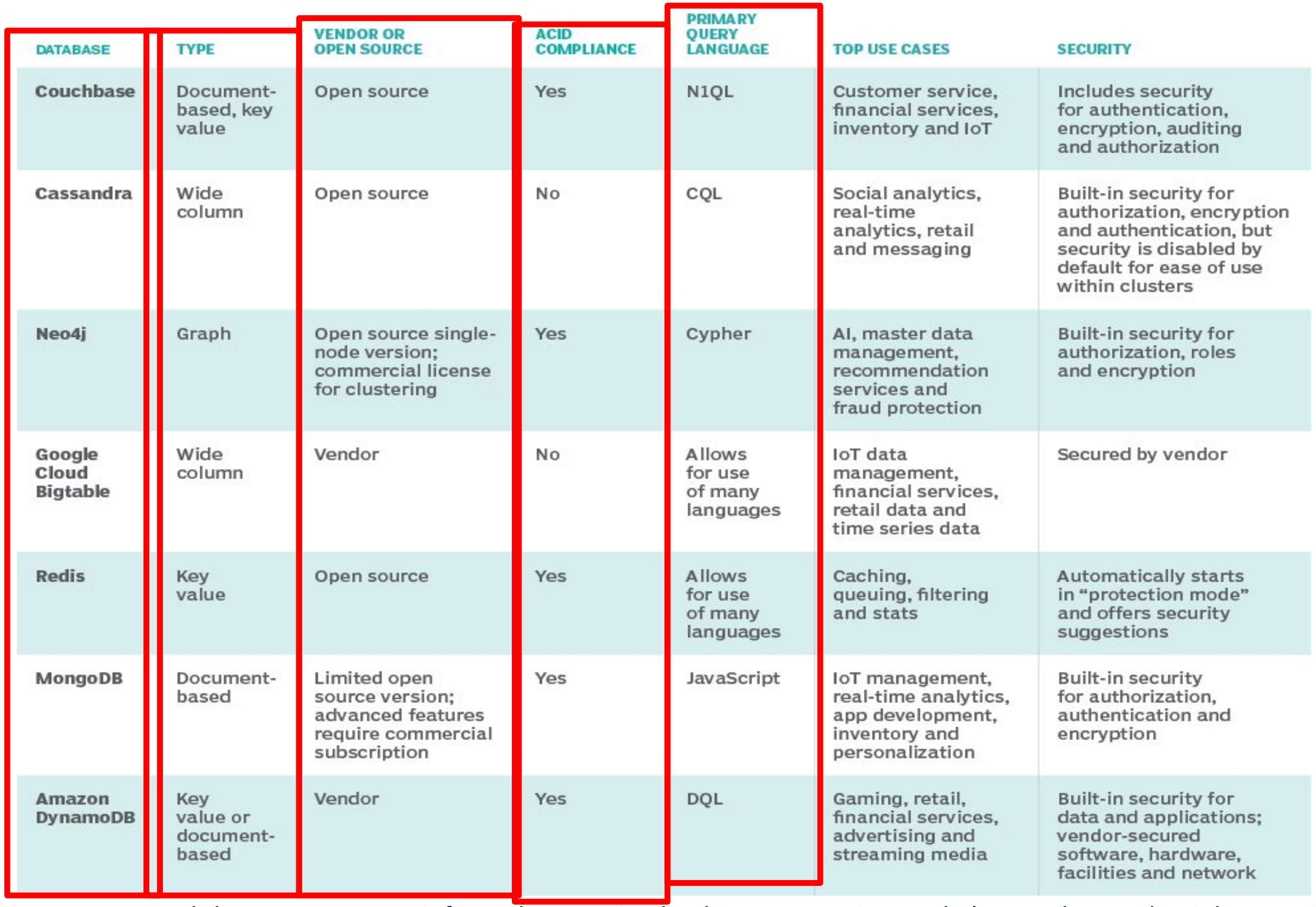
CAP Theorem



- Introduced in 2000 by Eric Brewer
- At most 2 of 3 properties:
  - Consistency
  - Availability
  - Partition Tolerance
- Cannot achieve all 3
- RDBMS falls under CA database
- MongoDB fall under CP database
- Cassandra falls under AP database

https://www.ibm.com/topics/cap-theorem

Comparison of NoSQL databases

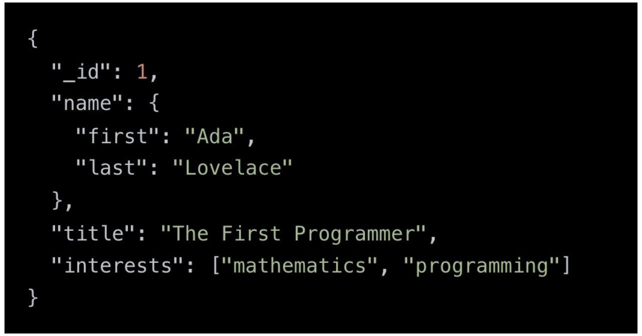


DATABASE	TYPE	VENDOR OR OPEN SOURCE	ACID COMPLIANCE	PRIMARY QUERY LANGUAGE	TOP USE CASES	SECURITY
Couchbase	Document-based, key value	Open source	Yes	N1QL	Customer service, financial services, inventory and IoT	Includes security for authentication, encryption, auditing and authorization
Cassandra	Wide column	Open source	No	CQL	Social analytics, real-time analytics, retail and messaging	Built-in security for authorization, encryption and authentication, but security is disabled by default for ease of use within clusters
Neo4j	Graph	Open source single-node version; commercial license for clustering	Yes	Cypher	AI, master data management, recommendation services and fraud protection	Built-in security for authorization, roles and encryption
Google Cloud Bigtable	Wide column	Vendor	No	Allows for use of many languages	IoT data management, financial services, retail data and time series data	Secured by vendor
Redis	Key value	Open source	Yes	Allows for use of many languages	Caching, queuing, filtering and stats	Automatically starts in "protection mode" and offers security suggestions
MongoDB	Document-based	Limited open source version; advanced features require commercial subscription	Yes	JavaScript	IoT management, real-time analytics, app development, inventory and personalization	Built-in security for authorization, authentication and encryption
Amazon DynamoDB	Key value or document-based	Vendor	Yes	DQL	Gaming, retail, financial services, advertising and streaming media	Built-in security for data and applications; vendor-secured software, hardware, facilities and network

https://www.techtarget.com/searchdatamanagement/infographic/NoSQL-database-comparison-to-help-you-choose-the-right-store

MongoDB

- Created by MongoDB, In
- Open-source document database
- Classified as a NoSQL database product
- Stores data in a type of JSON format called BSON



```
{
  "_id": 1,
  "name": {
    "first": "Ada",
    "last": "Lovelace"
  },
  "title": "The First Programmer",
  "interests": ["mathematics", "programming"]
}
```

MongoDB: Architecture

Application and Driver: Applications interact with the database using a Driver. Driver is the bridge between your application and the MongoDB database.

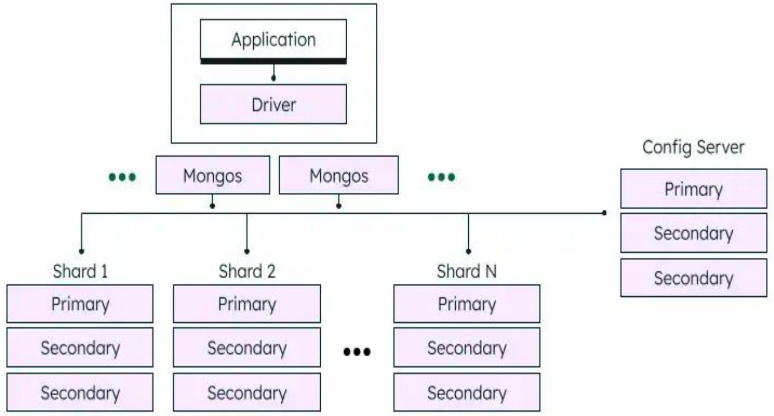
Query router (Mongo): It acts as an interface between the application and the sharded cluster.

Config Server: Config servers store metadata and configuration settings for the cluster

Shard: Each shard contains a subset of the sharded data.It is the approach to horizontal scaling.

The Primary Nodes: The primary node is the one that handles read and write operations under normal situations.

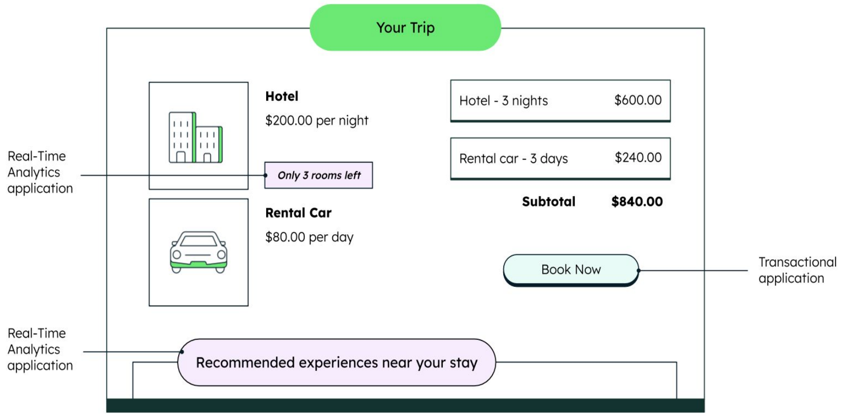
The Secondary Nodes: The Secondary nodes replicate the primary data, providing redundancy and enabling high availability.



https://www.geeksforgeeks.org/mongodb-architecture/

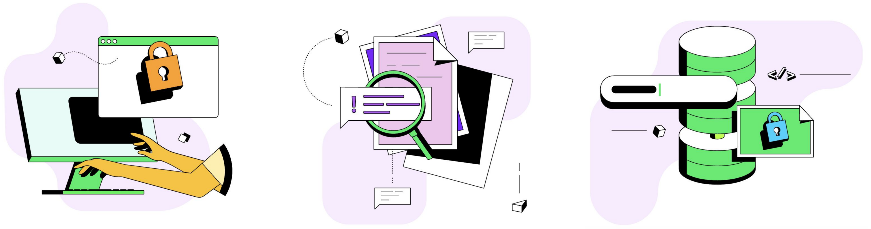
MongoDB: Features

- **Integrating large amounts of diverse data**
  - It is ideal for projects that require a unified view from multiple data sources.
- **Supporting agile development and collaboration**
  - Document databases enable different teams to manage distinct parts of a document.
  - It allows developers to store data in the database immediately and change it whenever necessary.
- **Providing real-time analytics**
  - With MongoDB, businesses can analyze any data in place and deliver insights in real time.


Your Trip
Real-Time Analytics application
Hotel
$200.00 per night
Only 3 rooms left
Rental Car
$80.00 per day
Hotel - 3 nights $600.00
Rental car - 3 days $240.00
Subtotal $840.00
Book Now
Transactional application
Real-Time Analytics application
Recommended experiences near your stay

Why use mongodb and when to use it?. MongoDB. (n.d.-c). https://www.mongodb.com/why-use-mongodb#:~:text=MongoDB%20is%20built%20o

MongoDB: Data Security



- **Authentication**
  - We need to create an "authentication database" that holds the authentication information of users.
  - SCRAM and Kerberos
- **Database Monitoring and Upgrading**
  - It helps detect and fix potential flaws before they negatively impact the system's performance.
  - Mongostat and mongotop
- **Network Security**
  - isolate your data and prevent inbound network access from the internet.
  - Allow just a one-way connection from your AWS, Azure, or Google Cloud to Atlas Clusters via Private Endpoints.

11 mongodb security features and best practices. Satori. (2023, April 22). https://satoricyber.com/mongodb-security/11-mongodb-security-features-and-best-practices/

Company using MongoDB

- **Ulta Beauty**: Ulta Beauty used MongoDB to better manage their expansive data and to scale offerings quickly and successfully.

- **Lyft**: Lyft adopted MongoDB to handle large volumes of data with varying structures, support agile development practices, and ensure scalability for their growing service offerings.

- **Cisco**: With their existing relational database in WebEx Social, complex SQL queries were time consuming. MongoDB now serves as the primary real-time data store for features in the platform that are write-heavy.

Cisco. MongoDB. (n.d.-a). https://www.mongodb.com/customers/cisco

Jurczak, S. (2023, June 15). *Ulta beauty solves seasonal shopping: Mongodb blog*. MongoDB. https://www.mongodb.com/blog/post/ulta-beauty-solves-seasonal-shopping

MongoDB: Case Study

- **Shutterfly**
  - a major Digital Picture Exchange and Private Publishing Company with over 6 billion photographs

- **Problem**: Shutterfly will generate a massive volume of new content that needs to be stored and available 24/7 for editing in real time. However, the relational database is inflexible to operate and costly to scale.

- **Solution**: Shutterfly chosed MongoDB as its document database platform running on AWS.

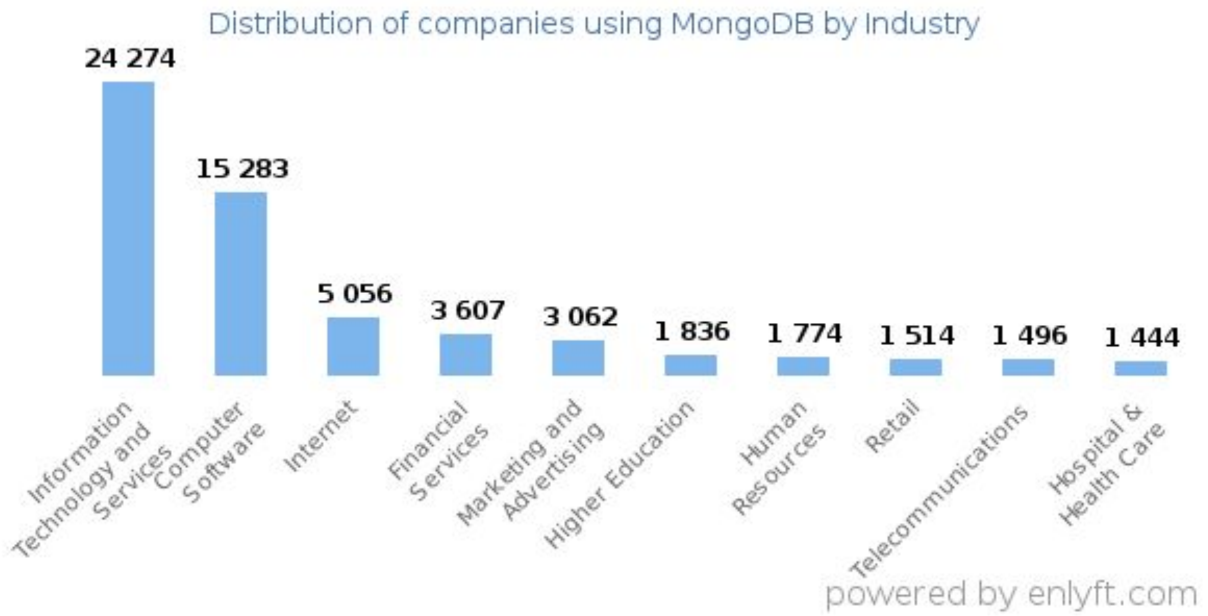
Shutterfly brings scalability and user experience into focus with mongodb atlas on AWS. MongoDB. (n.d.-c). https://www.mongodb.com/customers/shutterfly#:~:text=Shutterfly%20used%20MongoDB’s%20Mongomirror%20tool,applications%20continued%20to%20function%20normally.

MongoDB: Case Study continue

Enhancing Scalability and Flexibility: MongoDB increased the capacity of Atlas clusters during peak season and then reduce it back when usage slows down again, resulting in up to a 20% cost savings compared to their previous data setup.

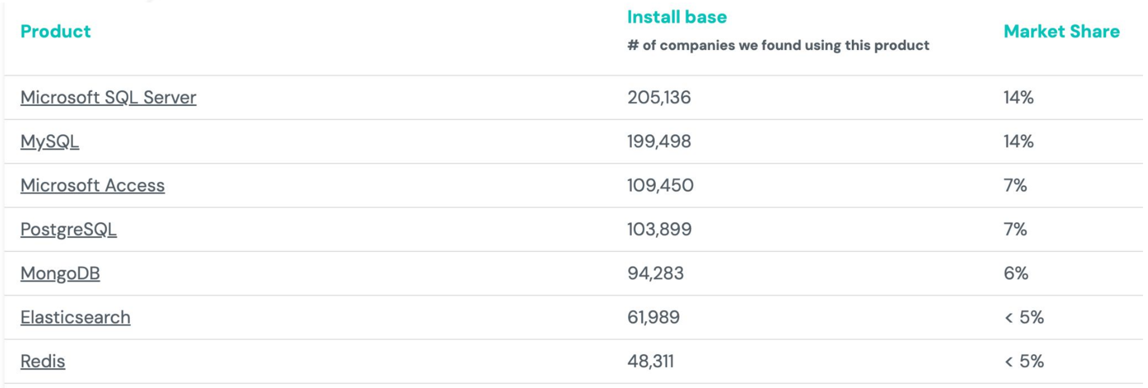
Rapid Service Deployment: shutterfly now can launch new services much more quickly and thus easily manage resources for new projects, significantly saving time and effort previously spent on manual process.

MongoDB: Market Data and Analysis

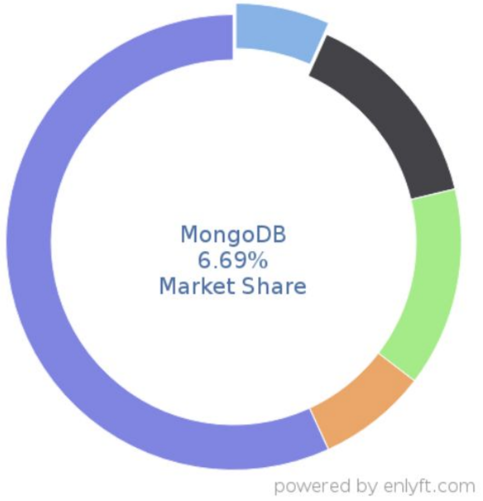


MongoDB commands 6.69% market share in Database Management System. and its marketshare. (n.d.). https://enlyft.com/tech/products/mongodb

MongoDB: Market Data and Analysis



Product	Install base # of companies we found using this product	Market Share
Microsoft SQL Server	205,136	14%
MySQL	199,498	14%
Microsoft Access	109,450	7%
PostgreSQL	103,899	7%
MongoDB	94,283	6%
Elasticsearch	61,989	< 5%
Redis	48,311	< 5%



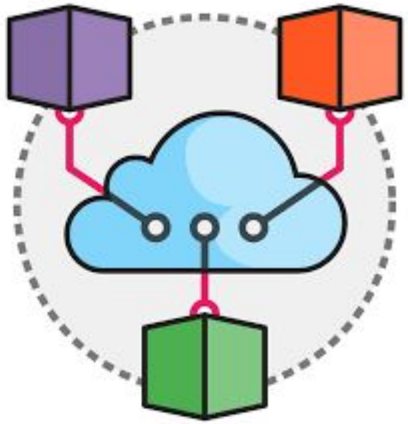
MongoDB commands 6.69% market share in Database Management System. and its marketshare. (n.d.). https://enlyft.com/tech/products/mongodb

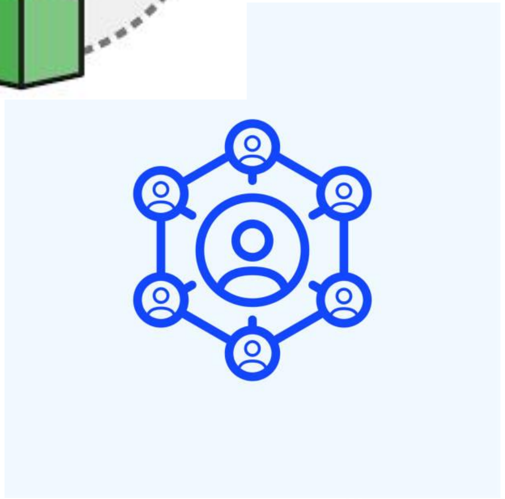
Apache Cassandra

- Apache Cassandra: Distributed, wide-column store NoSQL database management system.
- Inception
    - Originating from Facebook in 2007, Cassandra addressed the exponential data growth challenge faced by platforms like Messenger
- Designed for scalability and high availability in distributed environments.
- Widely used for managing large volumes of structured data across multiple nodes.

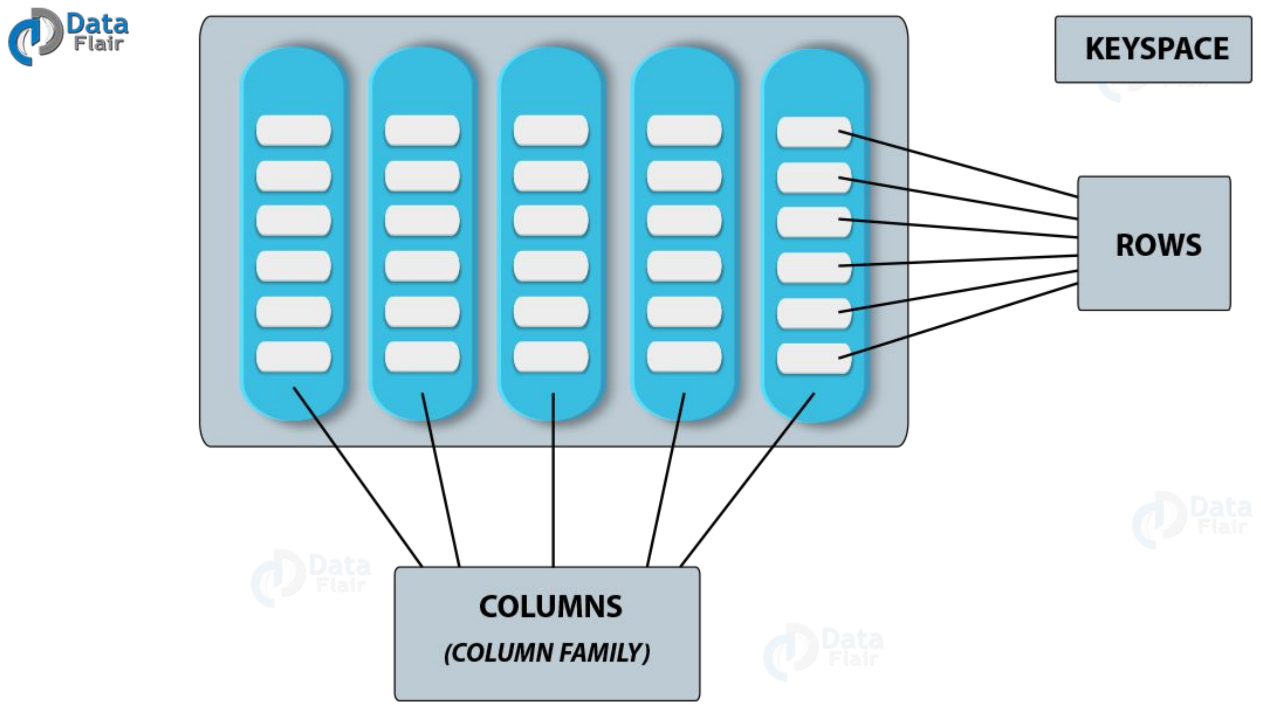
Cassandra Technical Details

- Distributed Architecture
  - Peer-to-peer architecture with no single point of failure.
  - Ring topology for distributing data across nodes.
- Consistency Levels
  - Tunable consistency allowing trade-offs between consistency and availability.
- Data Model
  - Column-family based data model supporting wide rows and flexible schemas.





Data Model

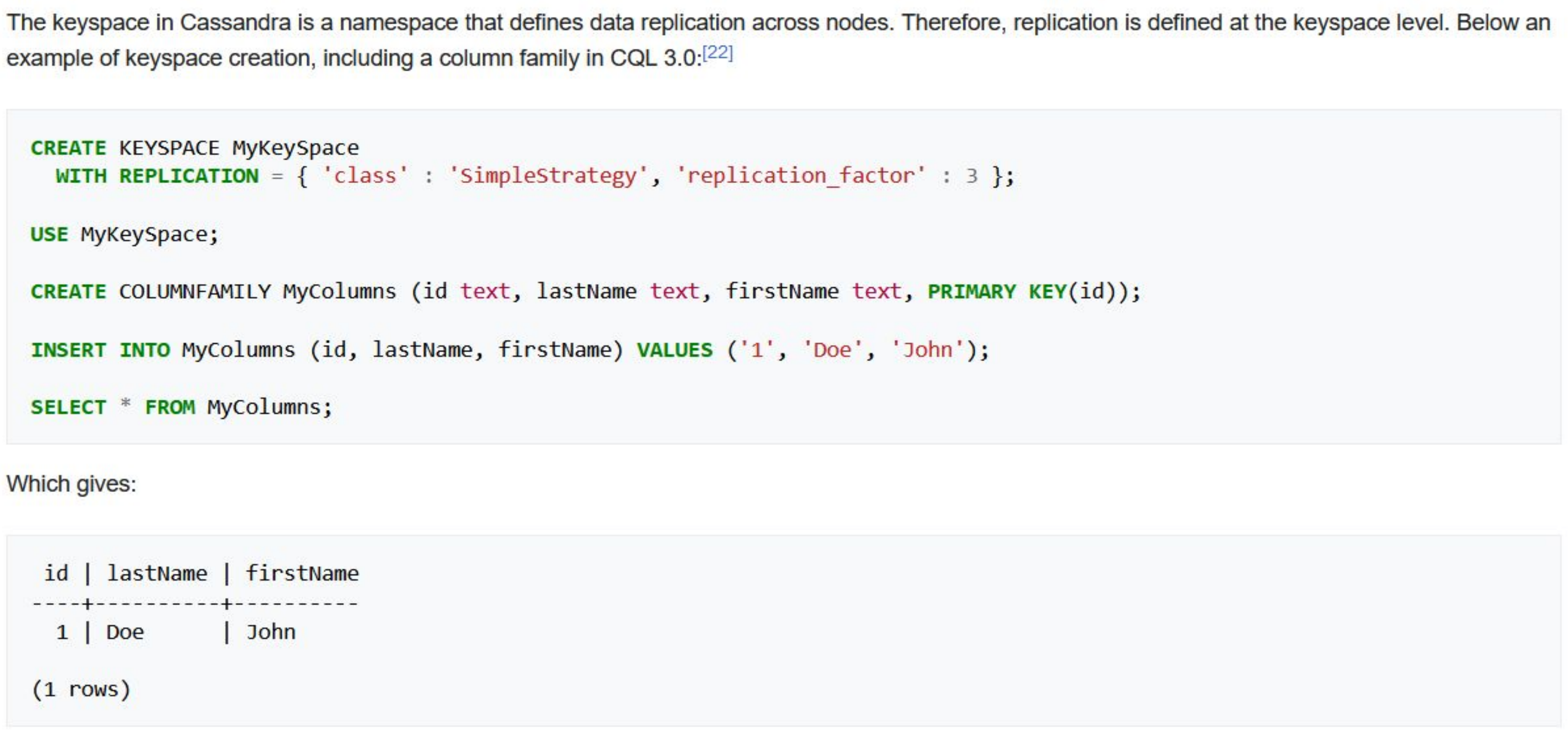


https://data-flair.training/blogs/cassandra-data-model/

Cassandra Query Language

- The Cassandra Query Language (CQL) is the primary interface for interacting with Cassandra databases.
- Syntax - CQL syntax is similar to SQL, making it familiar and easy to learn for developers accustomed to relational databases.
- Features:
  - Data Definition Language (DDL): Create, alter, and drop keyspaces, tables, and indexes.
  - Data Manipulation Language (DML): Insert, update, delete, and select data from tables.
  - Data Control Language (DCL): Grant and revoke permissions on keyspaces and tables.
- Consistency: CQL supports tunable consistency levels, allowing developers to balance between consistency and availability based on application requirements.

Sample Keyspace Creation in CQL



The keyspace in Cassandra is a namespace that defines data replication across nodes. Therefore, replication is defined at the keyspace level. Below an example of keyspace creation, including a column family in CQL 3.0:[22]

```
CREATE KEYSPACE MyKeySpace
  WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 3 };

USE MyKeySpace;

CREATE COLUMNFAMILY MyColumns (id text, lastName text, firstName text, PRIMARY KEY(id));

INSERT INTO MyColumns (id, lastName, firstName) VALUES ('1', 'Doe', 'John');

SELECT * FROM MyColumns;
```

Which gives:

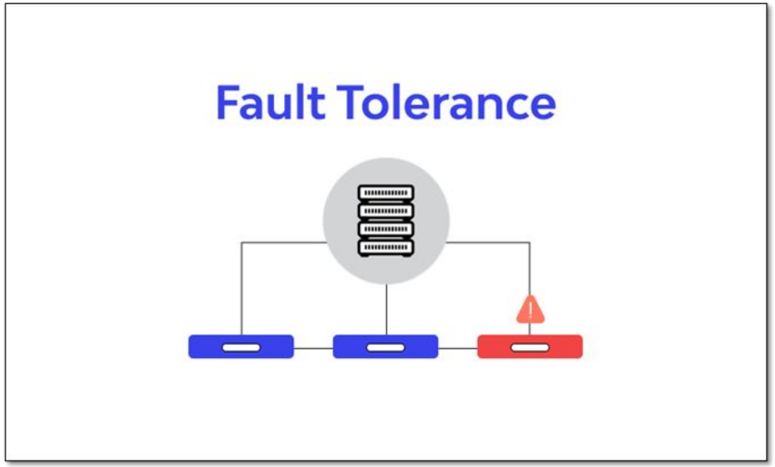
```
 id | lastName | firstName
----+----------+----------
  1 | Doe      | John

(1 rows)
```

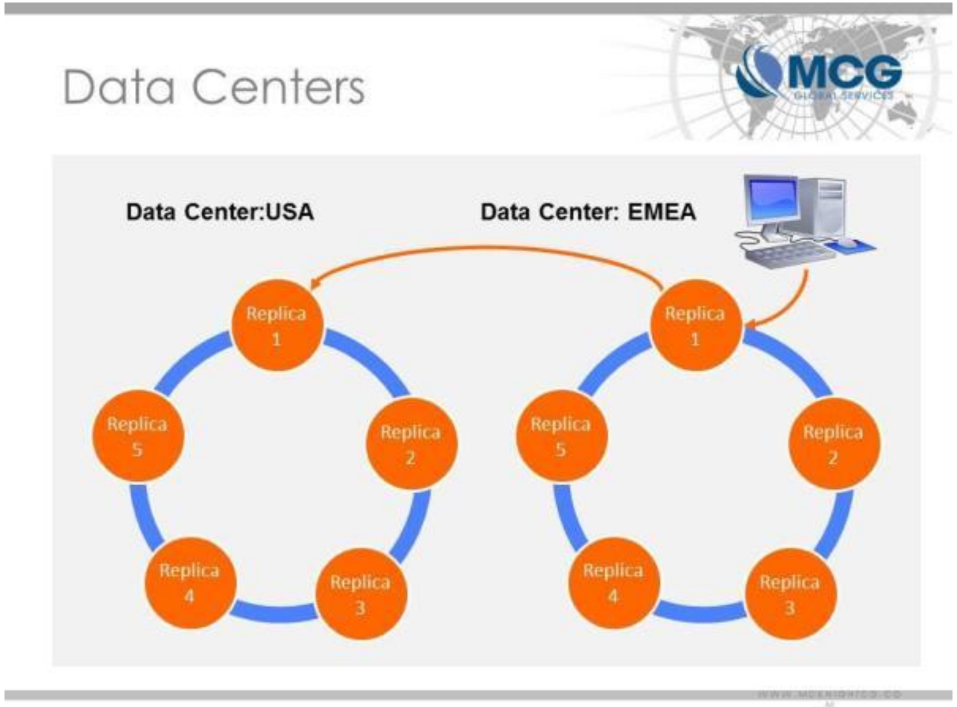
https://en.wikipedia.org/wiki/Apache_Cassandra

Cassandra Features and Functionalities

- Linear Scalability
  - Additions of nodes lead to linear scaling of performance and capacity.
- Tunable Consistency
  - Ability to adjust consistency levels dynamically based on application requirements.
- Fault Tolerance
  - Built-in replication and data distribution ensure fault tolerance and high availability.



Tunable Consistency



https://tdwi.org/articles/2016/07/18/tunable-consistency-in-cassandra-nosql.aspx

Market Positioning of Cassandra

- Endurance
  - Despite there being many NoSQL contemporaries, Cassandra has solidified its position as a robust database solution.
- Adoption
  - Cassandra is utilized by approximately 90% of Fortune 100 companies, reflecting its growing appeal in managing vast amounts of data.
- Growth
  - Driven by global distribution and constant connectivity demands, Cassandra remains a crucial component for data-intensive applications.
- Integrations
  - Cassandra's commitment to operational simplicity at scale and integration with Apache Spark and Apache Kafka reflects its adaptability.

Companies that Use Cassandra

NETFLIX

Netflix manages petabytes of data in Apache Cassandra which must be reliably accessible to users in mere milliseconds. They built sophisticated control planes that turn their persistence layer based on Apache Cassandra into a truly self-driving system.

Spotify

Spotify uses Cassandra as a solution for all personalization needs and are confident to scale it up to serve personalized experience to their ever growing size of engaged user base.

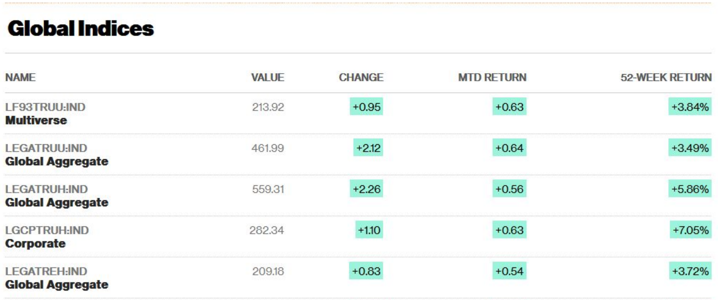
DISCORD

Cassandra was the only database that fulfilled all of Discord's requirements, as they can add nodes to scale it and it can tolerate a loss of nodes without any impact on the application. Related data is stored contiguously on disk providing minimum seeks and easy distribution around the cluster.

Testimony from : https://cassandra.apache.org/_/case-studies.html

Case Study: Bloomberg Barclays using Cassandra

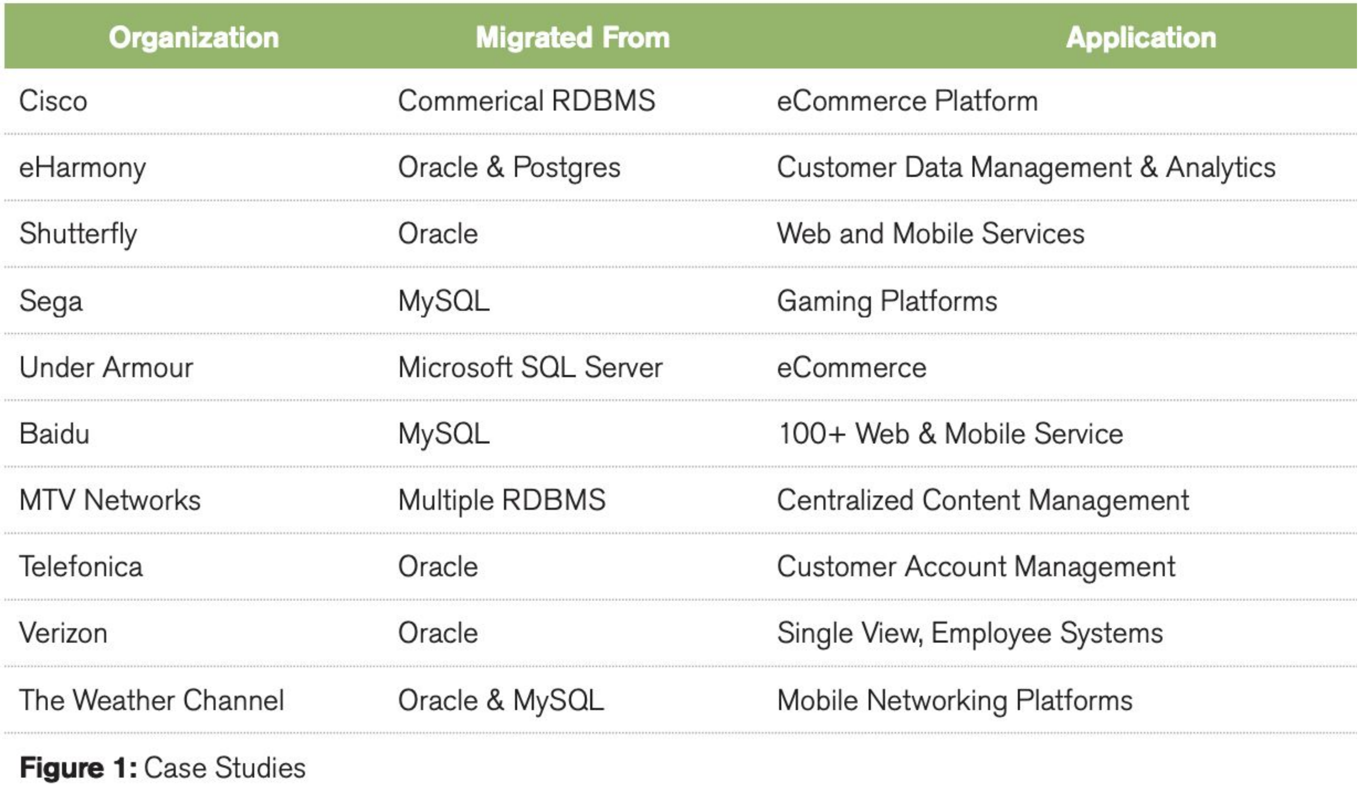
- Bloomberg undertook a multi-year project to develop an Index Construction Platform to manage the daily production of the Bloomberg Barclays fixed income indices.
- Technical Components
  - Apache Cassandra Backend Database: Deployed to store millions of data points generated daily, ensuring scalability and reliability for data management.
  - Apache Solr-backed Search Platform: Implemented to handle thousands of searches per minute, supporting efficient retrieval of index-related information.
  - Distributed Computational Engine: Engineered to process millions of computations daily, facilitating complex calculations required for index construction.
- Apache Cassandra offered a distributed, fault-tolerant database solution capable of handling the massive influx of data points while ensuring high availability.



Global Indices

NAME	VALUE	CHANGE	MTD RETURN	52-WEEK RETURN
LF93TRUU:IND **Multiverse**	213.92	+0.95	+0.63	+3.84%
LEGATRUU:IND **Global Aggregate**	461.99	+2.12	+0.64	+3.49%
LEGATRUH:IND **Global Aggregate**	559.31	+2.26	+0.56	+5.86%
LGCPTRUH:IND **Corporate**	282.34	+1.10	+0.63	+7.05%
LEGATREH:IND **Global Aggregate**	209.18	+0.83	+0.54	+3.72%

Relational → Non-Relational Databases



Organization	Migrated From	Application
Cisco	Commerical RDBMS	eCommerce Platform
eHarmony	Oracle & Postgres	Customer Data Management & Analytics
Shutterfly	Oracle	Web and Mobile Services
Sega	MySQL	Gaming Platforms
Under Armour	Microsoft SQL Server	eCommerce
Baidu	MySQL	100+ Web & Mobile Service
MTV Networks	Multiple RDBMS	Centralized Content Management
Telefonica	Oracle	Customer Account Management
Verizon	Oracle	Single View, Employee Systems
The Weather Channel	Oracle & MySQL	Mobile Networking Platforms

Figure 1: Case Studies

Image from MongoDB, 2015

Current Problems with NoSQL databases

Apache Cassandra

- Unencrypted Data Files
  - Cassandra doesn’t automatically encrypt data in storage.
  - Attackers with file-system access can extract information directly.
  - Mitigation: Encrypt sensitive information before writing to the database and use OS-level mechanisms to prevent unauthorized file access.
- Complexity
  - The clustered node system can be complex
  - Challenges with maintenance and troubleshooting
- Intercluster Communication
  - Nodes on a cluster communicate freely without encryption or authentication.
  - Suggested Mitigation: The current stable branch can support encryption on intercluster communication, which should be enabled.

https://ieeexplore.ieee.org/abstract/document/6120863?casa_token=l-AsxMwSMLcAAAAA:H5bC-BIHLvrfCt6tj_pW6qKifrhjE1Qh2gCcI55El87iVEWDBvlP4YjI_3kJwEVSlNfq_OyjTQ

Current Problems with NoSQL databases

Apache Cassandra

- Injection Attacks
    - Despite being NoSQL, Cassandra is vulnerable to injection attacks like SQL.

      unsafe CQL query: SELECT * FROM users WHERE username='[user_input]' AND password='[user_input]' ALLOW FILTERING;

      unsanitized input: SELECT * FROM users WHERE username='admin'/*' AND password='*/ and password >'' ALLOW FILTERING;
- Limited Query Support
    - Does not support joins
- Requires a lot of storage

https://medium.com/@harsh.b26/drawbacks-of-apache-cassandra-fecaa4704d14
https://www.invicti.com/blog/web-security/investigating-cql-injection-apache-cassandra/

Georgia Tech

Current Problems with NoSQL databases

MongoDB

- Injection Attacks
  - MongoDB is susceptible to injection attacks, similar to Cassandra.
  - It uses JavaScript, an interpreted language, increasing its vulnerability.
- Document Size Limit
  - Size limit of 16MB
  - Has GridFS API, but can be inefficient
  - Nest only upto 100 levels
- Data Redundancy, Inefficient Memory Usage
  - Stores information in <key,value> pairs, multiple similar keys → more storage
  - Even worse if keys are long strings
  - $lookup operator exists for left outer joins but has some tradeoffs

https://ieeexplore.ieee.org/abstract/document/6120863?casa_token=l-AsxMwSMLcAAAAA:H5bC-BIHLvrfCt6tj_pW6qKifrhjE1Q
h2gCcI55El87iVEWDBvlP4YjI_3kJwEVSlNfq_OyjTQ
https://www.geeksforgeeks.org/mongodb-advantages-disadvantages/

Georgia Tech

Future Trends & Ongoing Research

The benefits of non-relational databases have led to majority migration from relational to non relational database management system.

Most of the ongoing research is in data migration from relational to non relational databases.

Before Migration:

- Planning
- Number of records
- Mapping the data types (Data types must match)
- Character Encoding (to prevent automatic replacement of characters and loss)
- Tests (carry out tests on small subsets first)
- Implementation (identify how long it may take to migrate, provision for losses)
- Partial and Final Monitoring

http://archive.sciendo.com/MJSS/mjss.2018.9.issue-2/mjss-2018-0042/mjss-2018-0042.pdf

Framework for Migration from SQL →NoSQL

NoSQLayer framework is divided into:

Data Migration Module

- Identify Elements of the original Database (metadata required)
- Create equivalent structure using NoSQL model
- Completely migrate the data

Data Mapping Module

- Intercepts any queries issued to the application to NoSQL DBMS
- extracts information from SQL operations such as tables and “WHERE”clauses
- Translates the SQL operations to their equivalent NoSQL ones

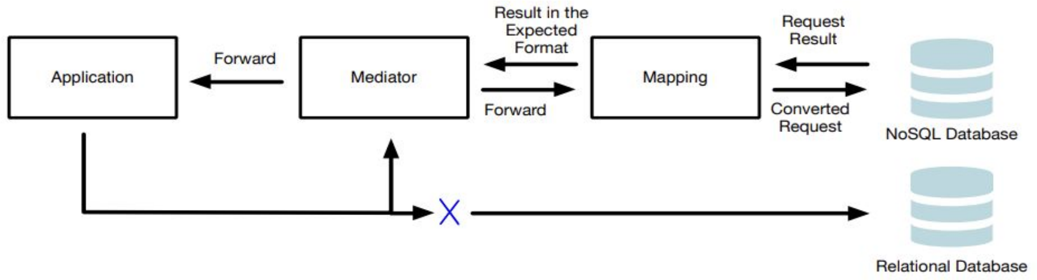


Figure 2: Data Mapping Module Working Diagram

https://www.researchgate.net/publication/277931056_A_Framework_for_Migrating_Relational_Datasets_to_NoSQL1

Questions?