UNIT-6

Big data technologies and Databases: Introduction to NoSQL, Uses, Features and Types, Need, Advantages, Disadvantages and Application of NoSQL, Overview of NewSQL, Comparing SQL, NoSQL and NewSQL, Introduction to MongoDB and its needs, Characteristics of MongoDB, Introduction of apache cassandra and its needs, Characteristics of Cassandra.

Introduction to NoSQL

NoSQL is a type of database management system (DBMS) that is designed to handle and store large volumes of unstructured and semi-structured data. Unlike traditional relational databases that use tables with pre-defined schemas to store data, NoSQL databases use flexible data models that can adapt to changes in data structures and are capable of scaling horizontally to handle growing amounts of data.

The term NoSQL originally referred to "non-SQL" or "non-relational" databases, but the term has since evolved to mean "not only SQL," as NoSQL databases have expanded to include a wide range of different database architectures and data models.

NoSQL databases are generally classified into four main categories:

1. **Document databases:** These databases store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages.
2. **Key-value stores:** These databases store data as key-value pairs, and are optimized for simple and fast read/write operations.
3. **Column-family stores:** These databases store data as column families, which are sets of columns that are treated as a single entity. They are optimized for fast and efficient querying of large amounts of data.
4. **Graph databases:** These databases store data as nodes and edges, and are designed to handle complex relationships between data.

NoSQL databases are often used in applications where there is a high volume of data that needs to be processed and analyzed in real-time, such as social media analytics, e-commerce, and gaming. They can also be used for other applications, such as content management systems, document management, and customer relationship management.

However, NoSQL databases may not be suitable for all applications, as they may not provide the same level of data consistency and transactional guarantees as traditional relational databases. It is important to carefully evaluate the specific needs of an application when choosing a database management system.

Use of NoSQL:

1. **Session Store**
   - Managing session data using relational database is very difficult, especially in case where applications are grown very much.
   - In such cases the right approach is to use a global session store, which manages session information for every user who visits the site.
   - NOSQL is suitable for storing such web application session information very is large in size.
   - Since the session data is unstructured in form, so it is easy to store it in schema less documents rather than in relation database record.
2. **User Profile Store**
   - To enable online transactions, user preferences, authentication of user and more, it is required to store the user profile by web and mobile application.
   - In recent time users of web and mobile application are grown very rapidly. The relational database could not handle such large volume of user profile data which growing rapidly, as it is limited to single server.
   - Using NOSQL capacity can be easily increased by adding server, which makes scaling cost effective
3. **Content and Metadata Store**
   - Many companies like publication houses require a place where they can store large amount of data, which include articles, digital content and e-books, in order to merge various tools for learning in single platform
   - The applications which are content based, for such application metadata is very frequently accessed data which need less response times.
   - For building applications based on content, use of NoSQL provide flexibility in faster access to data and to store different types of contents
4. **Mobile Applications**
   - Since the smartphone users are increasing very rapidly, mobile applications face problems related to growth and volume.
   - Using NoSQL database mobile application development can be started with small size and can be easily expanded as the number of user increases, which is very difficult if you consider relational databases.
   - Since NoSQL database store the data in schema-less for the application developer can update the apps without having to do major modification in database.
   - The mobile app companies like Kobo and Playtika, uses NOSQL and serving millions of users across the world.
5. **Third-Party Data Aggregation**
   - Frequently a business require to access data produced by third party. For instance, a consumer packaged goods company may require to get sales data from stores as well as shopper's purchase history.
   - In such scenarios, NoSQL databases are suitable, since NoSQL databases can manage huge amount of data which is generating at high speed from various data sources.
6. **Internet of Things**
   - Today, billions of devices are connected to internet, such as smartphones, tablets, home appliances, systems installed in hospitals, cars and warehouses. For such devices large volume and variety of data is generated and keep on generating.

- Relational databases are unable to store such data. The NOSQL permits organizations to expand concurrent access to data from billions of devices and systems which are connected, store huge amount of data and meet the required performance.

7. **E-Commerce**
   - E-commerce companies use NoSQL for store huge volume of data and large amount of request from user.
8. **Social Gaming**
   - Data-intensive applications such as social games which can grow users to millions. Such a growth in number of users as well as amount of data requires a database system which can store such data and can be scaled to incorporate number of growing users NOSQL is suitable for such applications.
   - NOSQL has been used by some of the mobile gaming companies like, electronic arts, zynga and tencent.
9. **Ad Targeting**
   - Displaying ads or offers on the current web page is a decision with direct income To determine what group of users to target, on web page where to display ads, the platforms gathers behavioral and demographic characteristics of users.
   - A NoSQL database enables ad companies to track user details and also place the very quickly and increases the probability of clicks.
   - AOL, Mediamind and PayPal are some of the ad targeting companies which uses NoSQL

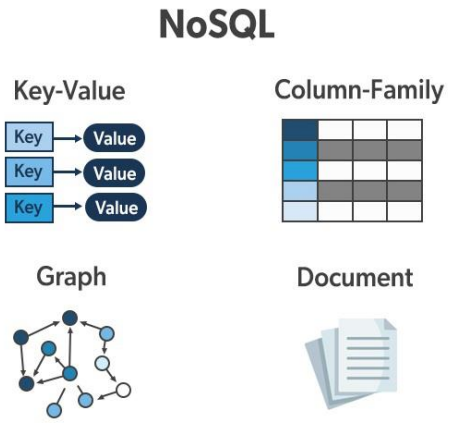
Key Features of NoSQL:

1. **Dynamic schema:** NoSQL databases do not have a fixed schema and can accommodate changing data structures without the need for migrations or schema alterations.
2. **Horizontal scalability:** NoSQL databases are designed to scale out by adding more nodes to a database cluster, making them well-suited for handling large amounts of data and high levels of traffic.
3. **Document-based:** Some NoSQL databases, such as MongoDB, use a document-based data model, where data is stored in semi-structured format, such as JSON or BSON.
4. **Key-value-based:** Other NoSQL databases, such as Redis, use a key-value data model, where data is stored as a collection of key-value pairs.
5. **Column-based:** Some NoSQL databases, such as Cassandra, use a column-based data model, where data is organized into columns instead of rows.
6. **Distributed and high availability:** NoSQL databases are often designed to be highly available and to automatically handle node failures and data replication across multiple nodes in a database cluster.
7. **Flexibility:** NoSQL databases allow developers to store and retrieve data in a flexible and dynamic manner, with support for multiple data types and changing data structures.
8. **Performance:** NoSQL databases are optimized for high performance and can handle a high volume of reads and writes, making them suitable for big data and real-time applications.

Types of NoSQL

A database is a collection of structured data or information which is stored in a computer system and can be accessed easily. A database is usually managed by a Database Management System (DBMS).

NoSQL is a non-relational database that is used to store the data in the nontabular form. NoSQL stands for not only SQL. The main types are documents, key-value, wide-column, and graphs.



Types of NoSQL Database:

- Document-based databases
- Key-value stores
- Column-oriented databases
- Graph-based databases

Document-Based Database:

The document-based database is a non-relational database. Instead of storing the data in rows and columns (tables), it uses the documents to store the data in the database. A document database stores data in JSON, BSON, or XML documents.

Documents can be stored and retrieved in a form that is much closer to the data objects used in applications which means less translation is required to use these data in the applications. In the Document database, the particular elements can be accessed by using the index value that is assigned for faster querying.

Collections are the group of documents that store documents that have similar contents. Not all the documents are in any collection as they require a similar schema because document databases have a flexible schema.

Key features of documents database:

- Flexible schema: Documents in the database has a flexible schema. It means the documents in the database need not be the same schema.
- Faster creation and maintenance: the creation of documents is easy and minimal maintenance is required once we create the document.
- No foreign keys: There is no dynamic relationship between two documents so documents can be independent of one another. So, there is no requirement for a foreign key in a document database.
- Open formats: To build a document we use XML, JSON, and others.

Key-Value Stores:

A key-value store is a nonrelational database. The simplest form of a NoSQL database is a key-value store. Every data element in the database is stored in key-value pairs. The data can be retrieved by using a unique key allotted to each element in the database. The values can be simple data types like strings and numbers or complex objects.

A key-value store is like a relational database with only two columns which is the key and the value.

Key features of the key-value store:

- Simplicity.
- Scalability.
- Speed.

Column Oriented Databases:

A column-oriented database is a non-relational database that stores the data in columns instead of rows. That means when we want to run analytics on a small number of columns, you can read those columns directly without consuming memory with the unwanted data.

Columnar databases are designed to read data more efficiently and retrieve the data with greater speed. A columnar database is used to store a large amount of data. Key features of columnar oriented database:

- Scalability.
- Compression.
- Very responsive.

Graph-Based databases:

Graph-based databases focus on the relationship between the elements. It stores the data in the form of nodes in the database. The connections between the nodes are called links or relationships.

Key features of graph database:

- In a graph-based database, it is easy to identify the relationship between the data by using the links.
- The Query's output is real-time results.
- The speed depends upon the number of relationships among the database elements.

Advantages of NoSQL:

There are many advantages of working with NoSQL databases such as MongoDB and Cassandra. The main advantages are high scalability and high availability.

1. **High scalability :** NoSQL databases use sharding for horizontal scaling. Partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved is sharding. Vertical scaling means adding more resources to the existing machine whereas horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement but horizontal scaling is easy to implement. Examples of horizontal scaling databases are MongoDB, Cassandra, etc. NoSQL can handle a huge amount of data because of scalability, as the data grows NoSQL scale itself to handle that data in an efficient manner.
2. **Flexibility:** NoSQL databases are designed to handle unstructured or semi-structured data, which means that they can accommodate dynamic changes to the data model.

This makes NoSQL databases a good fit for applications that need to handle changing data requirements.

3. **High availability :** Auto replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.
4. **Scalability:** NoSQL databases are highly scalable, which means that they can handle large amounts of data and traffic with ease. This makes them a good fit for applications that need to handle large amounts of data or traffic
5. **Performance:** NoSQL databases are designed to handle large amounts of data and traffic, which means that they can offer improved performance compared to traditional relational databases.
6. **Cost-effectiveness:** NoSQL databases are often more cost-effective than traditional relational databases, as they are typically less complex and do not require expensive hardware or software.

Disadvantages of NoSQL:

NoSQL has the following disadvantages.

1. **Lack of standardization** : There are many different types of NoSQL databases, each with its own unique strengths and weaknesses. This lack of standardization can make it difficult to choose the right database for a specific application
2. **Lack of ACID compliance :** NoSQL databases are not fully ACID-compliant, which means that they do not guarantee the consistency, integrity, and durability of data. This can be a drawback for applications that require strong data consistency guarantees.
3. **Narrow focus :** NoSQL databases have a very narrow focus as it is mainly designed for storage but it provides very little functionality. Relational databases are a better choice in the field of Transaction Management than NoSQL.
4. **Open-source :** NoSQL is open-source database. There is no reliable standard for NoSQL yet. In other words, two database systems are likely to be unequal.
5. **Lack of support for complex queries :** NoSQL databases are not designed to handle complex queries, which means that they are not a good fit for applications that require complex data analysis or reporting.
6. **Lack of maturity :** NoSQL databases are relatively new and lack the maturity of traditional relational databases. This can make them less reliable and less secure than traditional databases.
7. **Management challenge :** The purpose of big data tools is to make the management of a large amount of data as simple as possible. But it is not so easy. Data management in NoSQL is much more complex than in a relational database. NoSQL, in particular, has a reputation for being challenging to install and even more hectic to manage on a daily basis.
8. **GUI is not available :** GUI mode tools to access the database are not flexibly available in the market.

9. **Backup :** Backup is a great weak point for some NoSQL databases like MongoDB. MongoDB has no approach for the backup of data in a consistent manner.
10. **Large document size :** Some database systems like MongoDB and CouchDB store data in JSON format. This means that documents are quite large (BigData, network bandwidth, speed), and having descriptive key names actually hurts since they increase the document size.

Applications of NoSQL Databases

Data Mining

- When it comes to data mining, NoSQL databases are useful in retrieving information for data mining uses. Particularly when it's about large amounts of data, NoSQL databases store data points in both structured and unstructured formats leading to efficient storage of big data.
- Perhaps when a user wishes to mine a particular dataset from large amounts of data, one can make use of NoSQL databases, to begin with. Data is the building block of technology that has led mankind to such great heights.
- Therefore, one of the most essential fields where NoSQL databases can be put to use is data mining and data storage.

Social Media Networking Sites

- Social media is full of data, both structured and unstructured. A field that is loaded with tons of data to be discovered, social media is one of the most effective applications of NoSQL databases.
- From comments to posts, user-related information to advertising, social media marketing requires NoSQL databases to be implemented in certain ways to retrieve useful information that can be helpful in certain ways.
- Social media sites like Facebook and Instagram often approach open-source NoSQL databases to extract data that helps them keep track of their users and the activities going on around their platforms.

Software Development

- The third application that we will be looking at is software development. Software development requires extensive research on users and the needs of the masses that are met through software development.
- However, a developer must be able to scan through data that is available.
- Perhaps NoSQL databases are always useful in helping software developers keep a tab on their users, their details, and other user-related data that is important to be noted. That said, NoSQL databases are surely helpful in software development.

Overview of NewSQL

- NewSQL database is developed by integrating the speed and performance of NoSQL with the reliability of SQL.

- This database emphasizes on the features, which are not available in NoSQL Database and offers a strong dependability.
- The term NewSQL was coined in 2011 for online transaction processing (OLTP) systems, while maintaining atomicity, consistency, isolation and durability (ACID) guarantees. It was devised to work around the limitations of traditional SQL based systems.
- It aims to revamp the flaws in NoSQL by reincorporating some related database features.
- NewSQL databases resolve the problems concerned with the traditional online transaction processing. These databases run on SQL but differ in terms of their internal design. They can assimilate new information and perform many transaction at the same time.
- The main categories of NewSQL system are SQL engines, new architecture, transparent sharding middleware and Database-as-a-service.

NewSQL Database Features

Following are some best features of NewSQL Databases:

- **Currency Control:** This feature allows performing simultaneous transactions while maintaining the data integrity. It tackles the problem that may occur when multiple users are accessing or modifying the data simultaneously.
- **Replication:** With this feature, user can create copies of database and store them in a remote site next to the main site. User can update this database replica simultaneously.
- **Crash Recovery:** This mechanism enables the system to retrieve the data and move to a consistent state whenever system crashes.
- **Secondary Indexes:** With the secondary index feature, database user can approach databases information by using a different value other than the primary key.
- **Partitioning/Sharding:** NewSQL system divides the database into different subsets known as partitions or shards. The tables are bifurcated into various fragments with the boundaries based on Column values.

NewSQL Database Advantages and Disadvanatges

As we know, every new system or technology comes with some advantages over its previous version. But, on the other hand, there are some limitations too. So, let's check out the advantages and disadvantages of NewSQL Databases in this section.

NewSQL Database Advantages

- Benefits traditional ones with the currency control feature
- It preserves the ACID properties of databases
- It brings the advantages of SQL and NoSQL together
- Provide synchronous updates of data over the WAN
- Easy to switch between the users need and the type
- High availability and strong data durability
- Faster query processing time

NewSQL Database Disadvantages

- Not standardized
- In-memory architecture may be unsuitable for handling larger volumes
- Not fit for general purpose
- Provides limited access to traditional SQL system

Comparing SQL, NoSQL and NewSQL

Feature	SQL	NoSQL	NewSQL
Relational Property	Yes, it follows relational modeling to a large extent.	No, it doesn't follow a relational model. It was designed to be entirely different from that.	Yes, since the relational model is equally essential for real-time analytics.
ACID	Yes, ACID properties are fundamental to their application	No, rather provides for CAP support	Yes, Acid properties are taken care of.
SQL	Support for SQL	No support for old SQL	Yes, proper support and even enhanced functionalities for Old SQL
OLTP	Inefficient for OLTP databases.	It supports such databases, but it is not the best suited.	Fully functionally supports OLTP databases and is highly efficient
Scaling	Vertical scaling	Only Vertical scaling	Vertical + Horizontal scaling
Query Handling	Can handle simple queries with ease and fails when they get complex in nature	Better than SQL for processing complex queries	Highly efficient in processing complex queries and smaller queries.
Distributed Databases	No	Yes	Yes

Introduction to MongoDB:

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables.

- **MongoDB**, the most popular NoSQL database, is an open-source document-oriented database.
- The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different

mechanism for storage and retrieval of data. This format of storage is called BSON (similar to JSON format).

A simple MongoDB document Structure:

```
{
 title: 'Geeksforgeeks',
 by: 'Harshit Gupta',
 url: 'https://www.geeksforgeeks.org',
 type: 'NoSQL'
}
```

- SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today's real-world highly growing applications. **Modern applications are more networked, social and interactive than ever**. Applications are storing more and more data and are accessing it at higher rates.
- Relational Database Management System(RDBMS) i**s not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable**. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.
- example of a document database in MongoDB

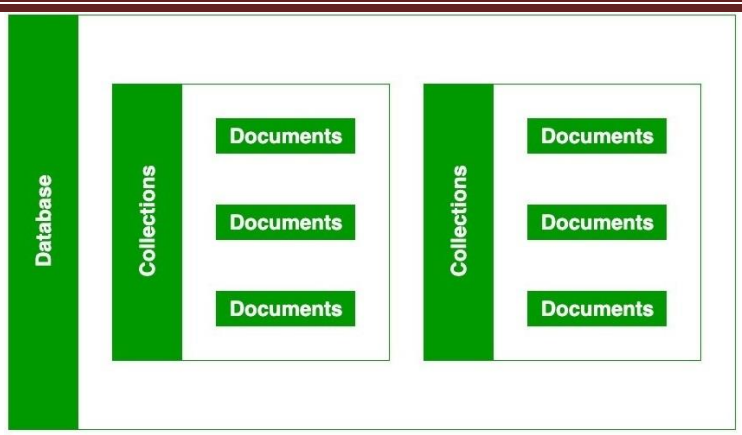
```
{_id:
name: "Thomson",
Age: 22,
Address: {{street: "124 church street",
        city: "brooklyn",
        state: "NY",
        zip: "13400",
        country: "US"}}
}
```

How it works?

Now, we will see how actually thing happens behind the scene. As we know that MongoDB is a database server and the data is stored in these databases. Or in other words, MongoDB environment gives you a server that you can start and then create multiple databases on it using MongoDB.

Because of its NoSQL database, the data is stored in the collections and documents. Hence the database, collection, and documents are related to each other as shown below:

- The MongoDB database contains collections just like the MYSQL database contains tables. You are allowed to create multiple databases and multiple collections.
- Now inside of the collection we have documents. These documents contain the data we want to store in the MongoDB database and a single collection can contain multiple documents and you are schema-less means it is not necessary that one document is similar to another.



- The documents are created using the fields. Fields are key-value pairs in the documents; it is just like columns in the relation database. The value of the fields can be of any BSON data types like double, string, boolean, etc.
- The data stored in the MongoDB is in the format of BSON documents. Here, BSON stands for Binary representation of JSON documents. Or in other words, in the backend, the MongoDB server converts the JSON data into a binary form that is known as BSON and this BSON is stored and queried more efficiently.
- In MongoDB documents, you are allowed to store nested data. This nesting of data allows you to create complex relations between data and store them in the same document which makes the working and fetching of data extremely efficient as compared to SQL. In SQL, you need to write complex joins to get the data from table 1 and table 2. The maximum size of the BSON document is 16MB.

How mongoDB is different from RDBMS?

MongoDB	RDBMS
It is a non-relational and document-oriented database.	It is a relational database.
It is suitable for hierarchical data storage.	It is not suitable for hierarchical data storage.
It has a dynamic schema.	It has a predefined schema.
It centers around the CAP theorem (Consistency, Availability, and Partition tolerance).	It centers around ACID properties (Atomicity, Consistency, Isolation, and Durability).
In terms of performance, it is much faster than RDBMS.	In terms of performance, it is slower than MongoDB.

Features of MongoDB

- **Schema-less Database:** It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it. Or in other words, in the MongoDB database, a single collection can hold multiple documents and these documents may consist of the different numbers of fields, content, and size. It is not necessary that the one document is similar to

another document like in the relational databases. Due to this cool feature, MongoDB provides great flexibility to databases.

- **Document Oriented:** In MongoDB, all the data stored in the documents instead of tables like in RDBMS. In these documents, the data is stored in fields(key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.
- **Indexing:** In MongoDB database, every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data. If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.
- **Scalability:** MongoDB provides horizontal scalability with the help of sharding. Sharding means to distribute data on multiple servers, here a large amount of data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. It will also add new machines to a running database.
- **Replication:** MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.
- **Aggregation:** It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause. It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods
- **High Performance:** The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

Advantages of MongoDB :

- It is a schema-less NoSQL database. You need not to design the schema of the database when you are working with MongoDB.
- It does not support join operation.
- It provides great flexibility to the fields in the documents.
- It contains heterogeneous data.
- It provides high performance, availability, scalability.
- It supports Geospatial efficiently.
- It is a document oriented database and the data is stored in BSON documents.
- It also supports multiple document ACID transition(string from MongoDB 4.0).
- It does not require any SQL injection.
- It is easily integrated with Big Data Hadoop

Disadvantages of MongoDB :

- It uses high memory for data storage.
- You are not allowed to store more than 16MB data in the documents.
- The nesting of data in BSON is also limited you are not allowed to nest data more than 100 levels.

Introduction of Apache Cassandra:

Apache Cassandra is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL database. Let us first understand what a NoSQL database does.

Apache Cassandra is an open source, distributed and decentralized/distributed storage system (database), for managing very large amounts of structured data spread out across the world. It provides highly available service with no single point of failure.

Listed below are some of the notable points of Apache Cassandra −

- It is scalable, fault-tolerant, and consistent.
- It is a column-oriented database.
- Its distribution design is based on Amazon's Dynamo and its data model on Google's Bigtable.
- Created at Facebook, it differs sharply from relational database management systems.
- Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful "column family" data model.
- Cassandra is being used by some of the biggest companies such as Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix, and more.

Features of Cassandra

Cassandra has become so popular because of its outstanding technical features. Given below are some of the features of Cassandra:

- **Elastic scalability** − Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.
- **Always on architecture** − Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure.
- **Fast linear-scale performance** − Cassandra is linearly scalable, i.e., it increases your throughput as you increase the number of nodes in the cluster. Therefore it maintains a quick response time.
- **Flexible data storage** − Cassandra accommodates all possible data formats including: structured, semi-structured, and unstructured. It can dynamically accommodate changes to your data structures according to your need.
- **Easy data distribution** − Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.
- **Transaction support** − Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).

- **Fast writes** − Cassandra was designed to run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency.

History of Cassandra

- Cassandra was developed at Facebook for inbox search.
- It was open-sourced by Facebook in July 2008.
- Cassandra was accepted into Apache Incubator in March 2009.
- It was made an Apache top-level project since February 2010.