VINSYS

DELiVERiNGSKiLLS
DRiViNGSUCCESS

www.VINSYS.COM

Need of NoSQL

VINSYS

Background of Relational Database

- For more than four decades people are using relational databases as a primary data storage mechanism. Structured Query Language (SQL) happens to be the more **structured, rigid way** of storing data, like a phone book. They are designed for reliable transactions and follow a proper structure to store data in a very **organized manner**.

Limitations of RDBMS

- Relational databases can handle thousands of queries in just a fraction of seconds, but this is possible at the small-scale applications. When the application grows relational databases start facing the **scalability issue**.
- Big gigantic website (such as Facebook, Google, Amazon) that throws billions or trillions of queries within a small amount of time then relational databases get failed in handling the queries.
- To get rid of this limitation in relational databases NoSQL comes in the picture that mainly focuses on two things…**high operations speed and flexibility** in storing the data. These two are the main common things that gave birth to the NoSQL database.

Features of NoSQL

1) About not requiring any "Structure"!!

- Relational databases store data in a **fixed and predefined structure**. It means when you start development you will have to define your data schema in terms of tables and columns. You have to change the schema every time the requirements change. This will lead to creating new columns, defining new relations, reflecting the changes in your application, discussing with your database administrators, etc.
- NoSQL database provides much more flexibility when it comes to handling data. There is **no requirement to specify the schema** to start working with the application. Also, the NoSQL database doesn't put a restriction on the types of data you can store together. It allows you to add more new types as your needs change. These all are the reasons NoSQL is best suited for **agile development** which requires **fast implementation**. Developers and architects choose NoSQL to handle data easily for various kinds of agile development application requirements.

2. Easily Scalable

- The primary reason to choose a NoSQL database is easy scalability. Well, relational databases can also be scaled but not easily and at a lower cost. Relational databases are built on the concept of traditional master-slave architecture. Scaling up means upgrading your servers by adding more processors, RAM, and hard-disks to your machine to handle more load and increase capacity. **You will have to divide the databases into smaller chunks across multiple hardware servers instead of a single large server. This is called sharding which is very complicated in relational databases.** Replacing and upgrading your database server machines to accommodate more throughput results in downtime as well. These things become a headache for developers and architects.

- NoSQL database built with a masterless, peer-to-peer architecture. Data is partitioned and balanced across multiple nodes in a cluster, and aggregate queries are distributed by default. This allows easy scaling in no time. Just executing a few commands will add the new server to the cluster. This scalability also improves performance, allowing for continuous availability and very high read/write speeds.

3. Distributed

- Relational databases use a **centralized application** that is **location-dependent** (e.g., single location), especially for write operations.

- On the other hand, the NoSQL database is designed to distribute data on a global scale. It uses **multiple locations involving multiple data centers** and/or **Cloud** regions for write and read operations.

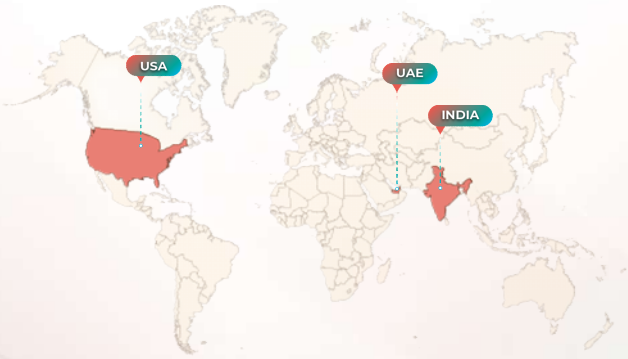
When to proceed with RDBMS

- When **Data is Structured.**
- When **complex Queries & Reports** are required to get output.
- When **Transaction Volume is relatively low per second**

When to proceed with NoSQL

- If your organization is dealing with **massive amounts of unstructured data** and your data requirements aren't clear at the outset, you probably don't have the luxury of developing a relational database with a clearly defined schema. In these cases, use NoSQL databases, you will get much more flexibility than its traditional counterparts.
- When **De-Normalization** of Data Model needs to be done for getting faster access to data.
- When Data is **changing frequently.**
- When Data is **dynamic**

Physical Presence



www.vinsys.com

enquiry@vinsys.com

Shivaji Niketan, Tejas Society, Behind Kothrud Bus Stand, Near Mantri Park, Kothrud, Pune - 411029.

304, City Tower 2, Near Crown Plaza, Sheikh Zayed Road. Dubai, UAE.
PO.Box - 213279

132 West 31st Street, First Floor, New York, 10001, USA