

INTRODUCTION TO NOSQL DATABASES

(ABSTRACT)

Name: Abhishek De

Roll no: 001710501027

BSCE 4th year 1st semester

Seminar 1

Relational databases rose in popularity in 1970s. In those times, memory was expensive, hence minimising data redundancy was an important issue. Also, it was the era of waterfall model of development. Hence database design was frozen before development started. Hence the concept of fixed schema came up.

With memory becoming cheaper and the rise in big data companies in the 21st century, data redundancy became a less concerning issue, while the major requirements included speed of database access, scalability, flexibility of schema and ease of development. This gave rise to new kinds of databases, which have different data models compared to traditional relational databases. NoSQL is an umbrella term used to describe all these databases. They are divided into 4 categories:

Document: Each entity is a JSON document, and a document may contain nested documents. (Most widely used among the four.)

Key-value: Each entity is a key-value pair, where value contains opaque unstructured data, unlike relational DBs.

Column-width: Tabular structure, where each row has a key and its own set of columns, ie columns don't span across all rows.

Graph: Entities are represented as nodes and relations between them as edges. Query languages are based on graph algorithms.

Some examples of popularly used NoSQL databases include DynamoDB, MongoDB, CouchDB, Neo4j, etc.

NoSQL databases have several advantages over SQL, such as

- They have flexible schema, hence developers have more control.
- Data is modelled in a self-contained/aggregated manner, hence expensive JOIN operations are not required.
- They are horizontally scalable, i.e databases can be distributed across multiple servers, to handle increasing load.
- Their data models are closer to the way data is used in code, hence easy for developers.

However they suffer from a few drawbacks:

- They don't follow ACID properties, hence any application which require transactions should not use NoSQL database, for ex – any financial application.
- There is no standardised query language as in case of SQL.
- They are not compatible with relational databases, hence migration of older relational databases is a challenge.