

RDBMS [Relational Database management system]

- It is a Software that performs data operations on relational databases.
- **Operations** : It stores, manages, query, and retrieve data.
- **Tables** : Data is represented in form of tables.
- **Foreign Keys** : The relationship between the two tables is represented by foreign keys.

Advantages :

- No data redundancy and inconsistency.
- Data Searching : we can run queries for data searching.
- Data Concurrency : A locking system is provided by RDBMS to prevent abnormalities from occurring.
- Data Integrity : Example, To maintain data integrity, numeric columns won't have alphabetic data.

Disadvantage :

- Rigid Schema : Each table has a specific structure
- High Cost

NoSQL Database

It stands for "non-SQL" database or non-relational database

NoSql is the umbrella term comprising of four different types of database

- **Key-Value database** : generally used for caching. Example : redis
- **Document database** : Brings best of both RDBMS and NoSQL. It combines the relationship concept from RDBMS and dynamic schema and horizontal scaling from NoSQL databases. Example : MongoDB
- **Columnar database** : The columns are stored together instead of rows. Because of that, the aggregation in such databases is rapid. It is widely used for data analysis. Example : Cassandra
- **Graph database** : Data is saved using graph DataStructure. It is basically used in social media platforms for creating a network. Example : Google maps, linkedin.

Polyglot Persistence

When an application needed more than a single database, i.e, RDBMS as well as NoSQL Database.

Normalization

If we have a single table , we divide it into different tables to reduce redundancy

Denormalization

If we have multiple tables, we merge them into one corresponding table

Benefits of Denormalization

- Faster data read operations
- Management convenient
- high data availability
- reduces the number of network calls to fetch data from different tables

Challenges of Denormalization

- Redundant data: wastage of memory
- It increases complexity
- Data inconsistency [as the redundancy increases the data starts to be inconsistent]
- slow writes into the tables, since we need to write at different places due to redundancy

How does Indexing works in Database

It helps in ordering the data and help in easy data fetching based on the columns. When we do indexing on a column it creates a lookup table with the columns and the pointers pointing towards the memory location of row, containing that column.

Once a query is hit, it initially check that the columns used in the where clause does have indexing on them or not, if yes, it goes to the index tables and get the pointers of the as per the query that is done on the table.

Before indexing : Time complexity is $O(N)$

After indexing : Time complexity is $O(\log n)$

B-Trees data structure is used to store the indexing as it is a multilevel format of tree base indexing, which has balance searched binary trees.

We do indexing, for the read intensive table/applications.