

HBase Vs RDBMS:

HBase

An open source and sorted map data built on Hadoop is what we call HBase. Basically, it is column-oriented and horizontally scalable. Moreover, it offers APIs enabling development in practically any programming language. Also, it offers random real-time read/write access to data in the Hadoop File System, as it is part of the Hadoop ecosystem.

RDBMS

RDBMS refers to Relational Database Management Systems. Basically, systems like SQL, MS SQL Server, IBM DB2, Oracle, MySQL and Microsoft Access are based on RDBMS. Since it is based on the relational model introduced by E.F. Codd so it is called Relational Database Management System (RDBMS).

Feature Wise Comparison of HBase Vs RDBMS

Below we are discussing the feature wise difference of HBase Vs RDBMS, let's explore this in detail:

i. Database Type

- HBase

HBase is the column-oriented database. On defining Column-oriented, each column is a continuous unit of page.

- RDBMS

Whereas, RDBMS is row-oriented that means here each row is a contiguous unit of page.

ii. Schema-type

- HBase

Schema of HBase is less restrictive, adding on the fly is possible.

- RDBMS

However, Schema of RDBMS is more restrictive.

iii. Sparse Tables

- HBase

HBase is good with the Sparse table.

- RDBMS

Whereas, RDBMS is not optimized for sparse tables.

iv. Scale up/Scale out.

- HBase

HBase supports scale out. It means while we need memory processing power and more disk, we need to add new servers to the cluster rather than upgrading the present one.

- RDBMS

However, RDBMS supports scale up. That means while we need memory processing power and more disk, we need upgrade same server to a more powerful server, rather than adding new servers.

v. Amount of data

- HBase

While here it does not depend on the particular machine but the number of machines.

- RDBMS

In RDBMS, on the configuration of the server, amount of data depends.

vi. Support of

- HBase

For HBase, there is no built-in support.

- RDBMS

And, RDBMS has ACID support.

vii. Data type

- HBase

HBase supports both structure and nonstructure data.

- RDBMS

RDBMS is suited for structure data.

viii. Transaction Integrity

- HBase

In HBase, there is no transaction guaranty.

- RDBMS

Whereas, RDBMS mostly guarantee transaction integrity.

ix. JOINS

- HBase

HBase supports JOINS.

- RDBMS

RDBMS does not support JOINS.

x. Referential Integrity

- HBase

While it comes to referential integrity, there is no in-built support.

- RDBMS

And, RDBMS supports referential integrity.

Features of HBase and RDBMS.

- Why HBase?

1. HBase is horizontally scalable.
2. Integrations with Map/Reduce framework.
3. Moreover, it is possible to refer HBase as a key-value store or column family oriented database.

- Why RDBMS?

1. Here, in form of rows and columns, data stores.
2. By using SQL queries, it also supports virtual tables from where we can retrieve data.
3. For the purpose of data uniqueness, RDBMS provides a primary key.
4. Also, it offers referential integrity..

HBase and other column-oriented DATABASE are often compared to more traditional and popular relational database or RDBMS.

HBASE	RDBMS
1. Column-oriented	1. Row-oriented (mostly)
2. Flexible schema, add columns on the Fly.	2. Fixed schema.
3. Good with sparse tables.	3. Not optimized for sparse tables.
4. No query language.	4. SQL.
5. Wide tables.	5. Narrow tables.
6. Joins using MR-not optimized.	6. Optimized for Joins (small, fast ones)
7. Tight-Integration with MR.	7. Not really.
8. De-normalize your data.	8. Normalize as you can.
9. Horizontal scalability-just add hard war.	9. Hard to share and scale.
10. Consistent.	10. Consistent.
11. No transactions.	11. Transactional.
12. Good for semi-structured data as well as structured data.	12. Good for structured data.

Basis CRUD Operations in HBase:

If you want any CRUD Operations in HBase, HBase should be up and running otherwise the operations will not be successful.

Running the child instance, but not running the master instance is not same as the running master instance as creating the child instance.

The initial sets of basic operations are often referred to as CRUD which stands for Create, Read, Update and Delete.