MapR-DB

MapR-DB is an enterprise-grade, high-performance, in-Hadoop, NoSQL (“Not Only SQL”) database management system. You can use it to add real-time, operational analytics capabilities to Hadoop.

Learn about the logical model of MapR-DB tables

MapR-DB tables are identical conceptually to tables in Apache HBase. If you're familiar with HBase tables, you'll be right at home with MapR-DB tables. In fact, your HBase applications can switch to using MapR-DB tables with no coding changes required.

See "Logical Model of MapR-DB Tables".

Learn about MapR-DB's architecture and its benefits

MapR-DB's architecture gives it a large number of advantages over other NoSQL databases.

See "Architecture of MapR-DB and the Resulting Advantages".

Index table data in Elasticsearch

You can create external indexes for your MapR-DB data by indexing columns, column families, or entire tables in Elasticsearch. When client applications update data in a source table, MapRDB replicates the update to the Elasticsearch type that is associated with it.

Updates to indexes happen in near real-time because individual updates to your MapR-DB source tables are replicated to Elasticsearch. There is no batching of updates, which would lead to recurring times where data is available in MapR-DB but not searchable in your indexes. Therefore, there is minimal latency between the availability of data in MapR-DB and the searchability of that data by end users.

The MapR distribution does not include Elasticsearch, which you can get from https://www.elastic.co/. MapR-DB works with Elasticsearch version 1.4.

See Indexing MapR-DB Data in Elasticsearch.

Get going with table replication

You can replicate changes (puts and deletes) to the data in one table to another table that is in a separate cluster or within the same cluster. Replicate entire tables, specific column families, and specific columns.

See "Replicating MapR-DB Tables".

Get going with the C APIs

`libhbase` is a library of C APIs for creating and accessing Apache HBase tables. The open-source version of this library is published on GitHub at https://github.com/mapr/libhbase.

MapR-DB includes a version of `libhbase` in `libMapRClient` that runs more efficiently and performs faster against MapR-DB tables.

See "Creating MapR-DB Applications with C".

Get going with the supported HBase Java APIs

The API for accessing MapR tables works the same way as the Apache HBase API. Code written for Apache HBase can be easily ported to use MapR-DB tables.

See "Creating MapR-DB Applications with Java".

Current Limitations

- Custom HBase filters are not supported.
- User permissions for column families are not supported. User permissions for tables and columns are supported.
- HBase authentication is not supported.
- HBase replication is handled with Mirror Volumes.
- Bulk loads using the HFiles workaround are not supported and not necessary.
- HBase coprocessors are not supported.
- Filters use a different regular expression library from `java.util.regex.Pattern`. See Supported Regular Expressions for a complete list of supported regular expressions.