APACHE HBASE

Introduction

What is HBase?

HBase is an open-source, column-oriented distributed database system in a Hadoop environment.

Initially, it was Google Big Table, afterward; it was renamed as HBase and is primarily written in Java.

Apache HBase is needed for real-time Big Data applications.

What is HBase?

HBase can store massive amounts of data from terabytes to petabytes.

The tables present in HBase consist of billions of rows having millions of columns.

HBase is built for low latency operations, which is having some specific features compared to traditional relational models

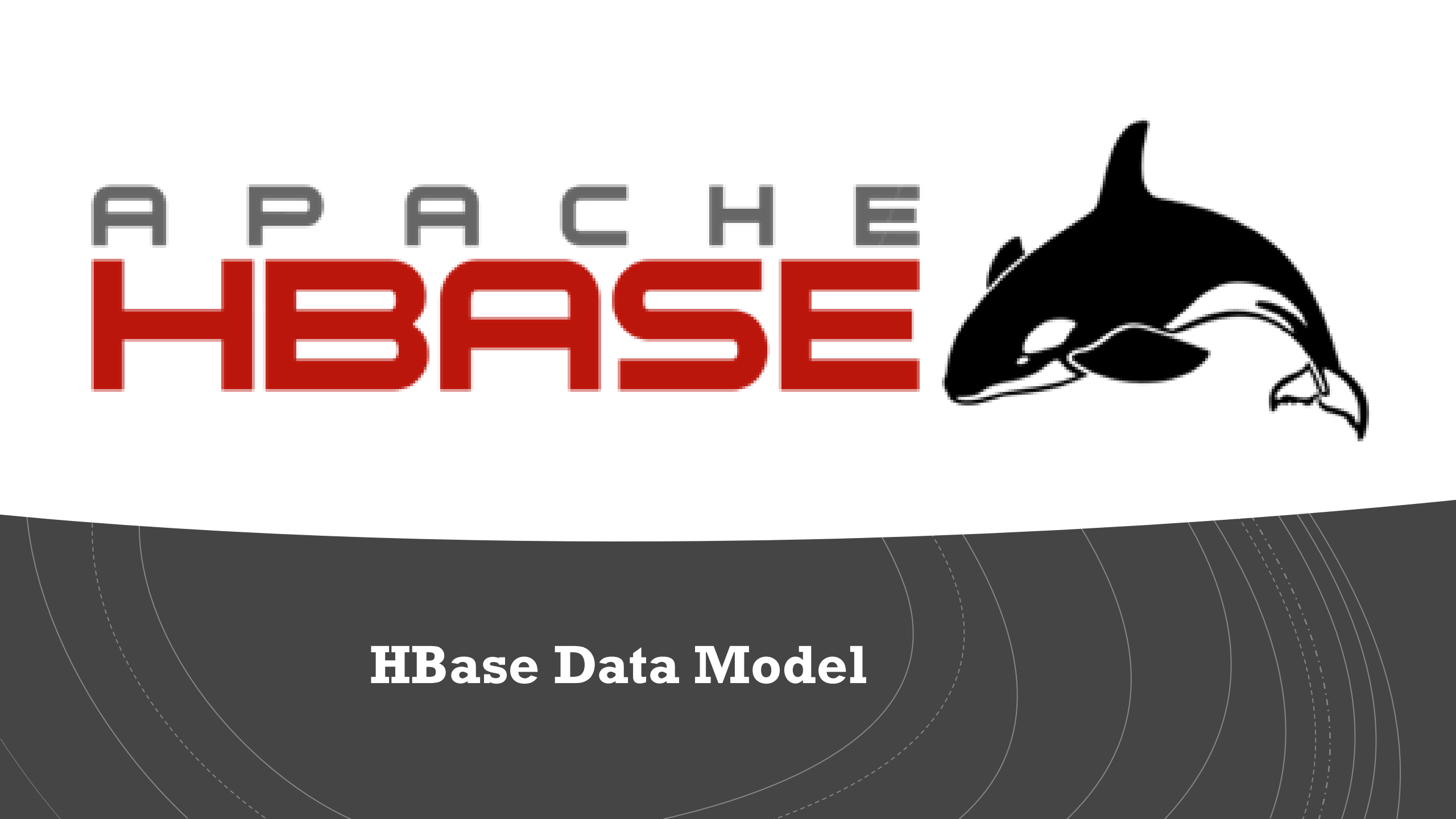
Why Choose HBase?

- A table for a popular web application may consist of billions of rows. If we want to search a particular row from such a huge amount of data, HBase is the ideal choice as query fetch time is less. Most of the online analytics applications use HBase.

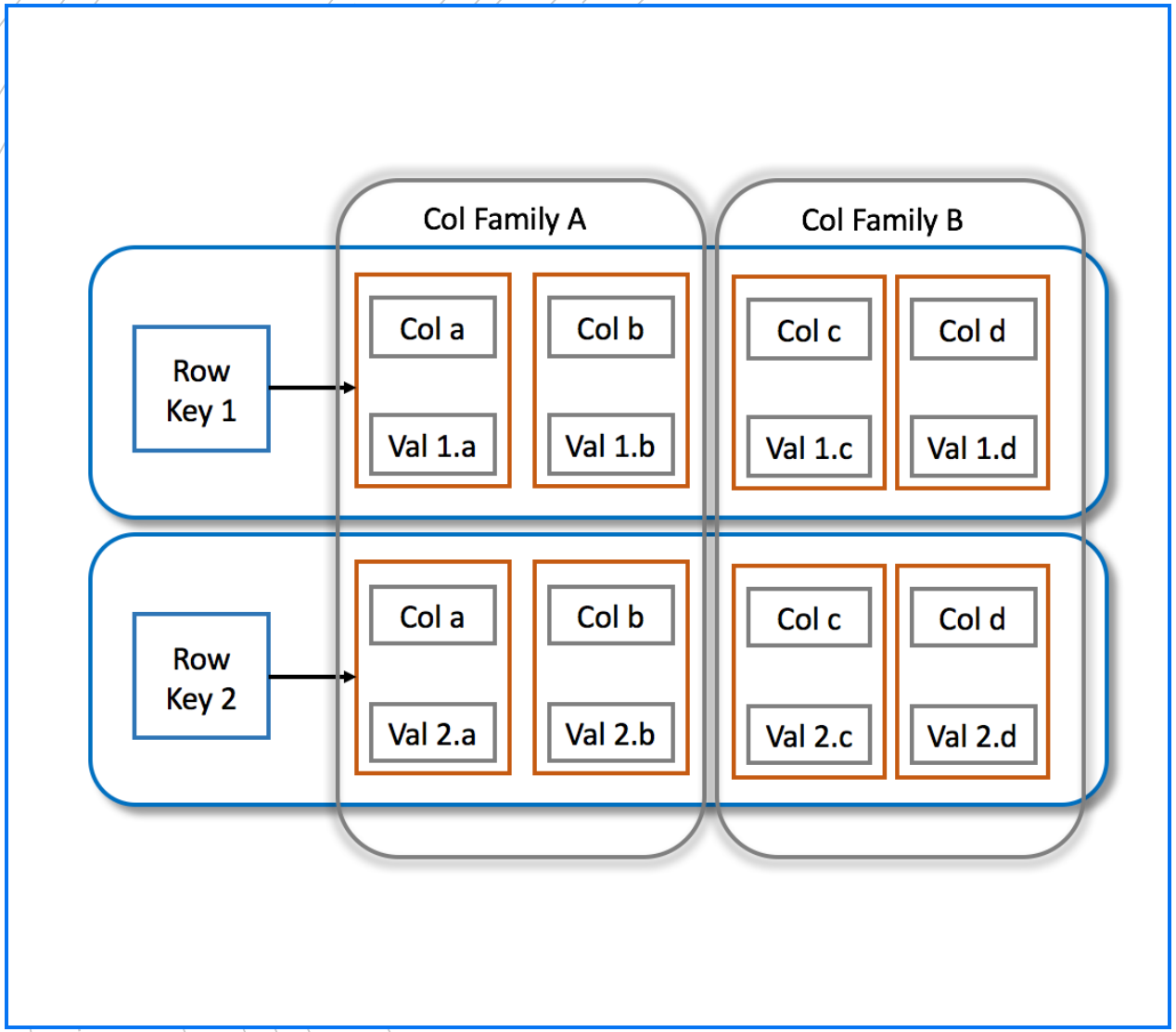
- Traditional relational data models fail to meet the performance requirements of very big databases. These performance and processing limitations can be overcome by Apache HBase.

Apache HBase Features

- HBase is built for low latency operations
- HBase is used extensively for random read and write operations
- HBase stores a large amount of data in terms of tables
- Provides linear and modular scalability over cluster environment
- Strictly consistent to read and write operations
- Automatic and configurable sharding of tables
- Automatic failover supports between Region Servers
- Convenient base classes for backing Hadoop MapReduce jobs in HBase tables
- Easy to use Java API for client access
- Block cache and Bloom Filters for real-time queries
- Query predicate pushes down via server-side filters.

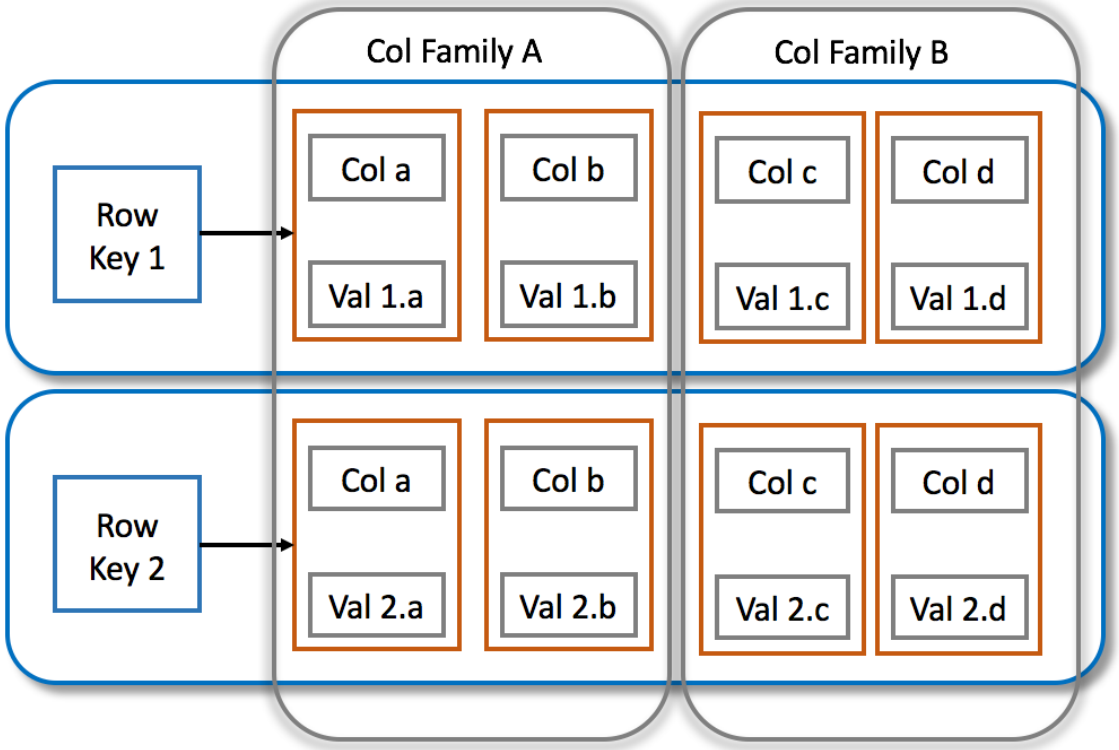


HBase Data Model

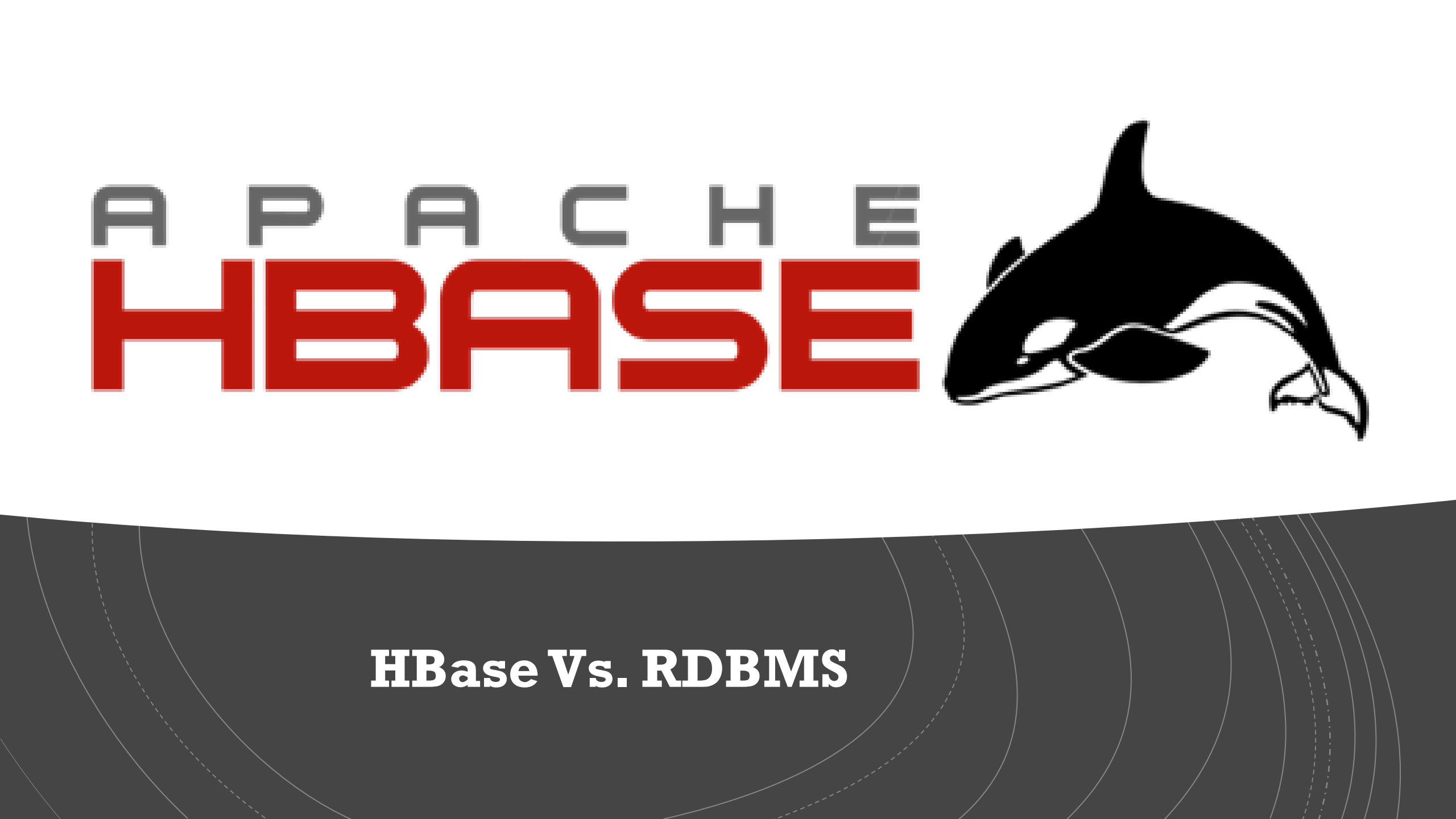


HBase Data Model is a set of components that consists of

- Tables,
- Rows,
- Column families,
- Cells,
- Columns
- Versions.



- **Table**: Collection of rows present.
- **Row**: Collection of column families.
- **Column Family**: Collection of columns.
- **Column**: Collection of key-value pairs.
- **Namespace**: Logical grouping of tables.
- **Cell**: A {row, column, version} tuple exactly specifies a cell definition in HBase.



HBase Vs. RDBMS

While comparing HBase with Traditional Relational databases, we have to take three key areas into consideration:

- data model
- data storage
- data diversity

APACHE HBASE

HBASE	RDBMS
• Schema-less in database	• Having fixed schema in database
• Column-oriented databases	• Row oriented datastore
• Designed to store De-normalized data	• Designed to store Normalized data
• Wide and sparsely populated tables present in HBase	• Contains thin tables in database
• Supports automatic partitioning	• Has no built in support for partitioning
• Well suited for OLAP systems	• Well suited for OLTP systems
• Read only relevant data from database	• Retrieve one row at a time and hence could read unnecessary data if only some of the data in a row is required
• Structured and semi-structure data can be stored and processed using HBase	• Structured data can be stored and processed using RDBMS
• Enables aggregation over many rows and columns	• Aggregation is an expensive operation

Summary

HBase provides unique features and will solve typical industrial use cases.

As column-oriented storage, it provides fast querying, fetching of results, and a high amount of data storage.