UNIT - IV

HBase

Hadoop Database

Column-Oriented NoSQL Database That Runs On Top Of HDFS

Data Model and Implementation

- ✓ It is designed for Storing and Retrieving large amounts of data with low latency (Measures Delay).
- ✓ Column Oriented Distributed Database –
  - Data is stored in Individual Columns and indexed by a unique Row Key
  - The Intersection of Rows and Columns is known as cell
  - Each cell has Time Stamp Identifier called Version

The HBase data model is based on the following concepts:-

Table: logical grouping of data. It is made up of rows and columns,

Rows: A row is a single record in a table. Each row is identified by a **Unique Row Key**.

Column Families: A column family is a collection of **Related Columns**.

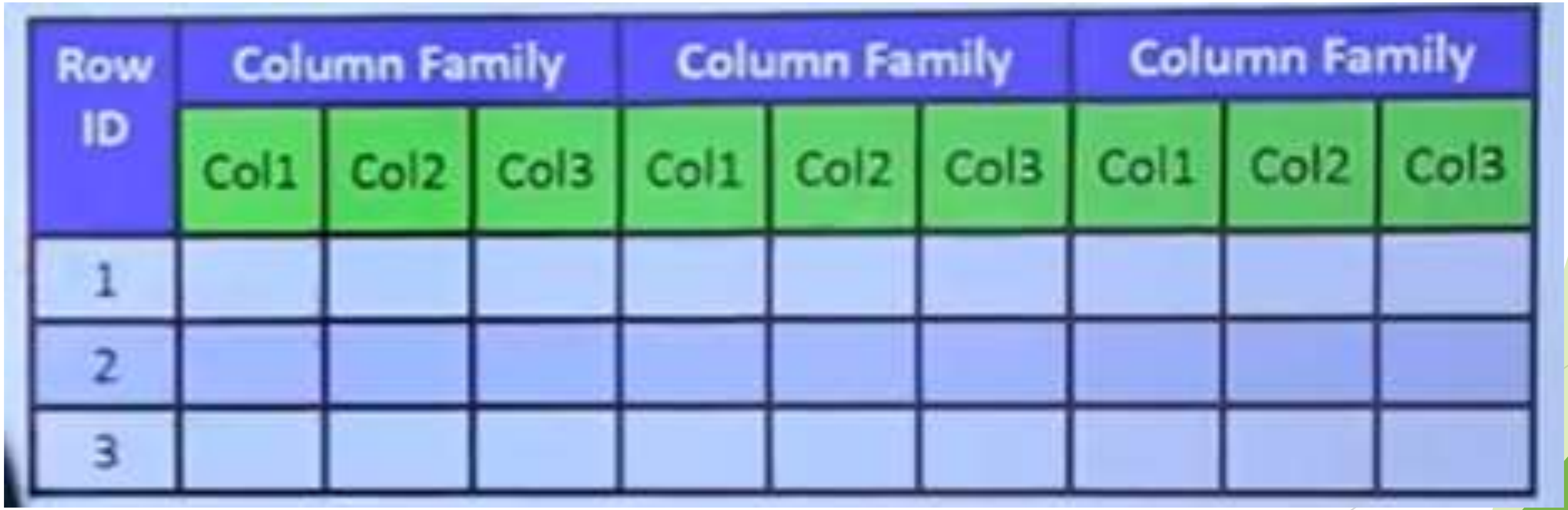
Columns: A column is a single piece of **Data In A Row**.

Cells: A cell is the intersection of a row, column family, and column.

.

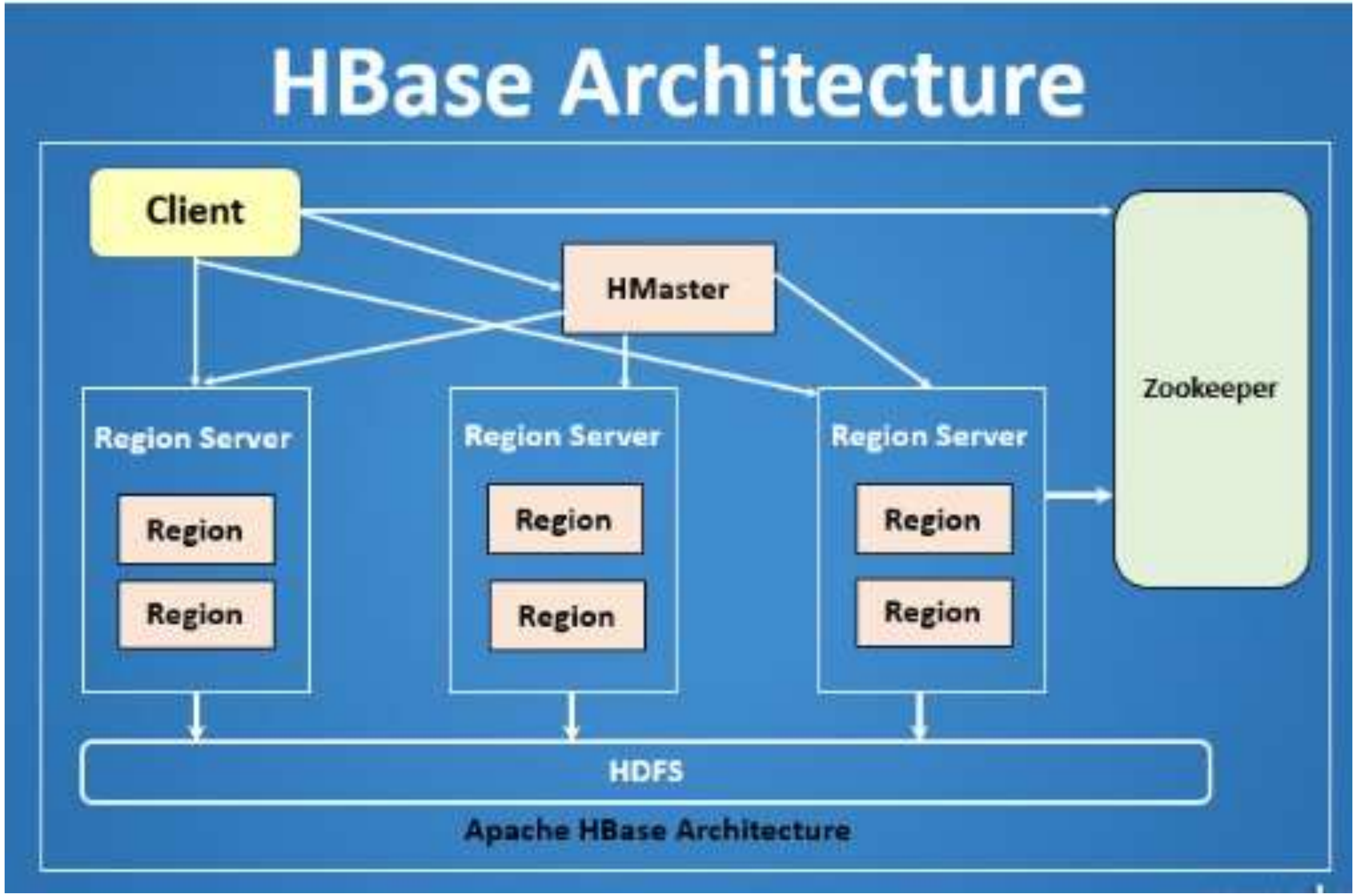
In Short HBase:

1. A Table is a collection of rows.
2. The Row is a collection of Column Families.
3. Column Family is a collection of Columns.
4. The column is a collection of Key-Value pairs.



Row ID	Column Family			Column Family			Column Family		
	Col1	Col2	Col3	Col1	Col2	Col3	Col1	Col2	Col3
1									
2									
3									

- ✓ HBase tables are horizontally partitioned into regions.
- ✓ Each region is served by a single region server.
- ✓ HBase automatically splits and merges regions as needed to maintain load balance.



Implementations of HBase

HBase is an open-source project, and there are a number of different implementations available:-

- ✓ The most popular implementation is the **Apache HBase** project

Other Implementations Of HBase Include:-

Cloudera HBase: A commercial implementation of HBase from **Cloudera**.

Hortonworks HBase: A commercial implementation of HBase from **Horton works**.

Google Dataproc HBase: A managed HBase service from **Google Cloud Platform**.

Amazon DynamoDB: A NoSQL database service from **Amazon Web Services** that is similar to HBase.

Use Cases for HBase:-

HBase is a good choice for a variety of use cases, including:-

Real-time Analytics: HBase can be used to **Store And Analyze Real-time Data**, such as sensor data or financial data.

Log Aggregation: HBase can be used to **Aggregate And Store Log Data** from a variety of sources.

Social Media Applications: HBase can be used to **Store And Retrieve Data** for social media applications, such as **User Profiles, Posts, And Messages**.

E-commerce Applications: HBase can be used to **Store And Retrieve Data** for e-commerce applications, such as **Product Catalogs, Customer Orders, And Inventory Levels**.

Conclusion:-

- HBase is a powerful tool for managing and analyzing big data. It is a good choice for applications that need to perform real-time queries on large amounts of data.

HBase Clients - Programs

- ✓ HBase clients are programs that allow you to interact with an HBase cluster.
- ✓ There are a number of different HBase clients available, each with its own strengths and weaknesses.
- ✓ The most popular HBase client is the **Java client API.** .
- ✓ However, it can be complex to use, especially for beginners.

Other HBase Clients Include:

HBase shell: **A Command-line Interface** for interacting with HBase.

- ✓ The HBase shell is easy to use, but it does not provide the same level of functionality as the Java client API.

Thrift Client: A Thrift client is a **Language-agnostic Client** for interacting with HBase.

- ✓ Thrift clients are available for a variety of programming languages, including **Java, Python, And Ruby.**

REST Client: A REST client is a **Web-based Client** for interacting with HBase.

- ✓ REST clients can be used to access HBase from any programming language that supports **HTTP Requests.**

- ✓ In addition to the above clients, there are a number of **Third-party HBase Clients** available.
- ✓ These clients often provide additional features, such as support for specific programming languages or frameworks.

Choosing an HBase client

- ✓ The best HBase client for you will depend on your **Specific Needs**.
- ✓ If you are **new to HBase**, I recommend starting with the **HBase Shell**.
- ✓ It is easy to use and provides a good introduction to the basic concepts of HBase.
- ✓ If you need more **Advanced Functionality**, you may want to consider the **Java client API or a third-party HBase client**.

When choosing a client, be sure to consider the following factors:-

Programming Language: Choose a client that supports the programming language you are using.

Features: Consider the features that are important to you, such as support for specific data types or transactions.

Ease Of Use: Choose a client that is easy to use and learn.

Community Support: Choose a client with a large and active community, so that you can get help if you need it.

HBase Examples

✓Here are some examples of how HBase can be used:

✓ **Real-time Analytics**: HBase can be used to store and analyze real-time data, such as sensor data or financial data.

For Example, a company could use HBase to store and analyze sensor data from its manufacturing plant to identify potential problems before they occur.

✓ **Log Aggregation**: HBase can be used to aggregate and store log data from a variety of sources.

For Example, a company could use HBase to store and analyze log data from its web servers, application servers, and database servers to identify trends and troubleshoot problems.

- ✓ **Social Media Applications**: HBase can be used to store and retrieve data for social media applications, such as user profiles, posts, and messages.

For Example, a social media company could use HBase to store and retrieve user profiles, posts, and messages so that users can quickly and easily find the information they are looking for.

- ✓ **E-commerce Applications**: HBase can be used to store and retrieve data for e-commerce applications, such as product catalogs, customer orders, and inventory levels.

For Example, an e-commerce company could use HBase to store and retrieve product catalogs so that customers can quickly and easily find the products they are looking for.

Praxis – Managing HBase Cluster

- Praxis is a **Web-based User Interface** for managing HBase clusters.

It provides a variety of features, including:

HDFS Integration

Praxis provides a **Visual Representation Of The HDFS Filesystem**, making it easy to see where HBase data is stored and how it is distributed.

UI

Praxis has a **User-friendly Interface** that makes it easy to perform common HBase tasks, such as

- ✓ Creating And Managing Tables.
- ✓ Loading And Querying Data.
- ✓ Monitoring Cluster Performance.

Metrics

Praxis provides a variety of metrics that can be used to **Monitor The Health And Performance Of HBase Clusters**.

These metrics include things like the

- ✓ Number Of Active Regions,
- ✓ The Number Of Requests Being Processed
- ✓ The Average Latency Of Requests.

Counters:

Praxis also provides a variety of counters that can be used to **Track The Activity Of HBase Clusters**.

These counters include things like

- ✓ The Number Of Reads And Writes Performed,
- ✓ The Number Of Errors Encountered, And
- ✓ The Number Of Regions That Have Been Split Or Merged.

Conclusion

- ✓ Praxis can be used to manage HBase clusters of all sizes, from small development clusters to large production clusters.
- ✓ It is a valuable tool for anyone who needs to manage or monitor HBase clusters.

1. Data Model -
2. Implementation of HBase-
3. HBase Clients -
4. Examples of HBase-
5. Praxis -