**What is HBase?**

**HBase = Hadoop Database**

HBase is a **NoSQL database** that runs on top of **HDFS (Hadoop Distributed File System)**. It is modeled after Google's BigTable and is designed to **store and manage large volumes of sparse data**.

**Key Features of HBase:**

| **Feature** | **Meaning** |
| --- | --- |
| **Column-oriented** | Data is stored by **columns**, not by rows (like in RDBMS). |
| **Schema-less** | You only define **column families**, not fixed schemas like SQL tables. |
| **Massive Scalability** | Designed for billions of rows and millions of columns. |
| **Real-time read/write** | Unlike Hive (batch processing), HBase supports quick access. |
| **Built on HDFS** | Inherits HDFS’s fault-tolerance and scalability. |

**Real-world Use Cases of HBase:**

* Facebook’s messaging platform
* Search engines for indexing millions of URLs
* IoT data storage (sensor readings over time)
* Time-series data (logs, stock prices, etc.)

**How is HBase different from Hive?**

| **Feature** | **Hive** | **HBase** |
| --- | --- | --- |
| **Type** | SQL-on-Hadoop (for analytics) | NoSQL database (real-time access) |
| **Schema** | Rigid, table-based schema | Flexible, column family based |
| **Use Case** | Batch processing | Fast lookups and updates |
| **Data Access** | Slow, non-transactional | Fast, random read/write |

**What Are We Doing in Task 7?**

We’ll create a **simple HBase table** named employee\_details with two column families:

1. PersonalInfo — contains: name, age, gender
2. ProfessionalInfo — contains: designation, salary, department

Then we will:

1. **Create the table**
2. **Insert 10 records** into it
3. **Query** a specific employee by name
4. **Update** the salary of an employee
5. **Delete** a specific employee’s record

All this will be done using the **HBase shell**, not SQL or HiveQL.

## Step-by-Step Procedure

### Step 1: Open Cloudera Terminal

* Start your **Cloudera Quickstart VM**.
* Open the **Terminal** window inside the VM.

**Step 2: Launch the HBase Shell**

In the terminal, type:

hbase shell

**🔍 Explanation:**

* hbase: invokes the HBase environment
* shell: starts the HBase command-line shell interface

You’ll now see a prompt like this:

hbase(main):001:0>

**Step 3: Create the HBase Table**

At the HBase shell prompt, type:

create 'employee\_details', 'PersonalInfo', 'ProfessionalInfo'

**Explanation:**

* create – the command to create a table
* 'employee\_details' – the table name (must be in quotes)
* 'PersonalInfo', 'ProfessionalInfo' – the **column families**

If successful, you’ll see:

0 row(s) in X.X seconds

Table employee\_details is now created with 2 column families.

**Step 4: Insert 10 Employee Records**

We will insert one row per employee.

Each record will include:

* A **row key** (e.g., emp101, emp102, etc.)
* Fields under **PersonalInfo** column family: name, age, gender
* Fields under **ProfessionalInfo** column family: designation, salary, department

**Syntax to insert data in HBase:**

put 'table\_name', 'row\_key', 'column\_family:column\_name', 'value'

**Let’s insert 10 sample employee records:**

Paste the following commands **one by one** in the HBase shell:

put 'employee\_details', 'emp101', 'PersonalInfo:name', 'Alice'

put 'employee\_details', 'emp101', 'PersonalInfo:age', '29'

put 'employee\_details', 'emp101', 'PersonalInfo:gender', 'Female'

put 'employee\_details', 'emp101', 'ProfessionalInfo:designation', 'Developer'

put 'employee\_details', 'emp101', 'ProfessionalInfo:salary', '70000'

put 'employee\_details', 'emp101', 'ProfessionalInfo:department', 'IT'

put 'employee\_details', 'emp102', 'PersonalInfo:name', 'Bob'

put 'employee\_details', 'emp102', 'PersonalInfo:age', '34'

put 'employee\_details', 'emp102', 'PersonalInfo:gender', 'Male'

put 'employee\_details', 'emp102', 'ProfessionalInfo:designation', 'Manager'

put 'employee\_details', 'emp102', 'ProfessionalInfo:salary', '90000'

put 'employee\_details', 'emp102', 'ProfessionalInfo:department', 'HR'

put 'employee\_details', 'emp103', 'PersonalInfo:name', 'Charlie'

put 'employee\_details', 'emp103', 'PersonalInfo:age', '26'

put 'employee\_details', 'emp103', 'PersonalInfo:gender', 'Male'

put 'employee\_details', 'emp103', 'ProfessionalInfo:designation', 'Analyst'

put 'employee\_details', 'emp103', 'ProfessionalInfo:salary', '60000'

put 'employee\_details', 'emp103', 'ProfessionalInfo:department', 'Finance'

put 'employee\_details', 'emp104', 'PersonalInfo:name', 'Diana'

put 'employee\_details', 'emp104', 'PersonalInfo:age', '30'

put 'employee\_details', 'emp104', 'PersonalInfo:gender', 'Female'

put 'employee\_details', 'emp104', 'ProfessionalInfo:designation', 'Team Lead'

put 'employee\_details', 'emp104', 'ProfessionalInfo:salary', '80000'

put 'employee\_details', 'emp104', 'ProfessionalInfo:department', 'IT'

put 'employee\_details', 'emp105', 'PersonalInfo:name', 'Edward'

put 'employee\_details', 'emp105', 'PersonalInfo:age', '28'

put 'employee\_details', 'emp105', 'PersonalInfo:gender', 'Male'

put 'employee\_details', 'emp105', 'ProfessionalInfo:designation', 'Tester'

put 'employee\_details', 'emp105', 'ProfessionalInfo:salary', '55000'

put 'employee\_details', 'emp105', 'ProfessionalInfo:department', 'QA'

put 'employee\_details', 'emp106', 'PersonalInfo:name', 'Fatima'

put 'employee\_details', 'emp106', 'PersonalInfo:age', '32'

put 'employee\_details', 'emp106', 'PersonalInfo:gender', 'Female'

put 'employee\_details', 'emp106', 'ProfessionalInfo:designation', 'HR Executive'

put 'employee\_details', 'emp106', 'ProfessionalInfo:salary', '58000'

put 'employee\_details', 'emp106', 'ProfessionalInfo:department', 'HR'

put 'employee\_details', 'emp107', 'PersonalInfo:name', 'George'

put 'employee\_details', 'emp107', 'PersonalInfo:age', '40'

put 'employee\_details', 'emp107', 'PersonalInfo:gender', 'Male'

put 'employee\_details', 'emp107', 'ProfessionalInfo:designation', 'Director'

put 'employee\_details', 'emp107', 'ProfessionalInfo:salary', '150000'

put 'employee\_details', 'emp107', 'ProfessionalInfo:department', 'Management'

put 'employee\_details', 'emp108', 'PersonalInfo:name', 'Hina'

put 'employee\_details', 'emp108', 'PersonalInfo:age', '25'

put 'employee\_details', 'emp108', 'PersonalInfo:gender', 'Female'

put 'employee\_details', 'emp108', 'ProfessionalInfo:designation', 'Intern'

put 'employee\_details', 'emp108', 'ProfessionalInfo:salary', '25000'

put 'employee\_details', 'emp108', 'ProfessionalInfo:department', 'IT'

put 'employee\_details', 'emp109', 'PersonalInfo:name', 'Imran'

put 'employee\_details', 'emp109', 'PersonalInfo:age', '36'

put 'employee\_details', 'emp109', 'PersonalInfo:gender', 'Male'

put 'employee\_details', 'emp109', 'ProfessionalInfo:designation', 'Architect'

put 'employee\_details', 'emp109', 'ProfessionalInfo:salary', '100000'

put 'employee\_details', 'emp109', 'ProfessionalInfo:department', 'Engineering'

put 'employee\_details', 'emp110', 'PersonalInfo:name', 'Jaya'

put 'employee\_details', 'emp110', 'PersonalInfo:age', '31'

put 'employee\_details', 'emp110', 'PersonalInfo:gender', 'Female'

put 'employee\_details', 'emp110', 'ProfessionalInfo:designation', 'Consultant'

put 'employee\_details', 'emp110', 'ProfessionalInfo:salary', '75000'

put 'employee\_details', 'emp110', 'ProfessionalInfo:department', 'Consulting'

**Explanation of one line:**

put 'employee\_details', 'emp101', 'PersonalInfo:name', 'Alice'

| **Part** | **Meaning** |
| --- | --- |
| put | Insert data into table |
| 'employee\_details' | Table name |
| 'emp101' | Row key (unique identifier for the employee) |
| 'PersonalInfo:name' | Column family PersonalInfo, column name |
| 'Alice' | The actual value being inserted |

**Step 5: Display details of an employee with a specific name**

Since HBase is **row-key-based**, we fetch data by row key (not by name).

Let’s assume you want to display all details of employee emp105:

get 'employee\_details', 'emp105'

This will print all fields for emp105 — both from PersonalInfo and ProfessionalInfo.

**Step 6: Update the salary of an employee with a specific name**

Again, we identify by row key.  
Let’s say you want to **update emp105 salary to 72000**:

put 'employee\_details', 'emp105', 'ProfessionalInfo:salary', '72000'

This overwrites the existing salary value.

**Step 7: Delete the record of an employee with a specific name**

To **delete the entire record of employee emp105**:

deleteall 'employee\_details', 'emp105'

This will remove **all columns** under all column families for that row.

### Final Optional Step (to verify):

scan 'employee\_details'

This shows the current data after updates and deletions.