TP Hbase

**HBase** is a distributed non-relational database management system, written in Java, with structured storage for large tables.

HBase is a column-oriented database. Based on a master/slave architecture, databases of this type are capable of managing huge quantities of information.

HBase is a sub-project of Hadoop, a distributed architecture framework. The HBase database is generally installed on Hadoop's HDFS file system, to facilitate distribution.

**Example**: Consider the Hbase table below:

COLUMN FAMILIES



 Column qualifiers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **row key** | **address** | | | **order** | |
| customer\_ID | city | state | street | number | amount |
| 1 | Paris | France | Bailly | ORD-15 | 15 |
| 2 | Nancy | France | Belfort | ORD-16 | 15 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Start shell command

|  |  |
| --- | --- |
| MAC | Windows |
| * start-all.sh * start-hbase.sh * hbase shell | * cmd 1 (launch hadoop): hdfs namenode -format   cd../..  cd hadoop-3.2.1 cd sbin  start-all |
| * cmd 2 (launch server): start-hbase |
| * cmd 3 (launch shell): hbase shell |

# List the number of hbase servers used

Command: hbase(main):001:0> status

**Result:**

**hbase(main):001:0> status**

**1 active master, 0 backup masters, 1 servers, 1 dead, 5.0000 average load**

1. **The Hbase version used**

Order: version

**Result:**

hbase(main):002:0> version

1.2.6, rUnknown, Mon May 29 02:25:32 CDT 2017

# Information about the current user

Ordering: whoami

**Result:**

**hbase(main):003:0> whoami Nouha Othman (auth:SIMPLE)**

**groups: Administrators, Users**

1. **Help command in hbase for all available commands and their explanations**

Command: hbase(main):001:0> table\_help

**Result:**

Hbase displays all commands available in Hbase

# Handling tables in HBASE

1. **List tables created in Hbase**

Command: list

**Result:**

**hbase(main):005:0> list TABLE**

**Customers Student cars**

**3 row(s) in 0.0320 seconds**

**=> ["Customers", "Student", "cars"]**

1. **Create an Hbase table: to do this, you need to enter the table name and the** column **family names**. **In our example, the table name is** *customer* **and the column family names are address and order.**

Command: create 'tableName','Column1Name','Column2Name'

hbase(main):029:0> create 'customer','address','order'

0 row(s) in 1.2990 seconds

=> Hbase:Table - customer

# Check the creation of the customer table :

Command: list

# Display the contents of the customer table :

Order: scan 'customer

# Deactivate customer table :

Command: disable 'customer

1. **Display table contents disabled :** Command: scan 'customer' Result: error message

# Check if the customer table is deactivated:

Command: is\_disabled 'customer

# Check if the customer table exists:

Order: exists 'customer

# Description of the customer table :

Command: describe 'customer

# Activate customer table :

Command: enable 'customer

# Description of the customer table :

Command: describe 'customer

# Add a family column to the customer table :

Command: alter 'customer', {NAME=>'personal data'}

# Delete a family column from the customer table :

Command: alter 'customer', 'delete'=>'personal data'

# Add a customer to the customer table :

Ordering:

put 'customer', '1', 'address:city','Paris' put 'customer', '1', 'address:state','France' put 'customer', '1', 'address:street','Bailly' put 'customer', '1', 'order:number','ORD-15'

put 'customer', '1', 'order:amount','15

# Display the contents of the customer table :

Ordering:

hbase(main):025:0> scan 'customer'

ROW COLUMN+CELL

1 column=address:city,

timestamp=1647735037503, value=Paris

1 column=address:state,

timestamp=1647735096356, value=France

1 column=address:street,

timestamp=1647735096363, value=Bailly

1 column=order:amount,

timestamp=1647735096384, value=15

1 column=order:number,

timestamp=1647735096376, value=ORD-16

# Add a second customer to the customer table :

Ordering:

put 'customer', '2', 'address:city','Nancy' put 'customer', '2', 'address:state','France' put 'customer', '2', 'address:street','Belfort' put 'customer', '2', 'order:number','ORD-16'

put 'customer', '2', 'order:amount','15'

# Reading data from the customer table :

Orders :

get 'customer', '1'

get 'customer', '1', 'address'

get 'customer', '1', 'address:city'

get 'customer', '1', {COLUMN=>'address:city'}

get 'customer', '1', {COLUMN=>['address:city','address:street']}

scan 'customer',{COLUMNS=>['address:city']}

scan 'customer',{COLUMNS=>['address:city'], LIMIT=>1}

**Question**: What's the difference between get and scan results?

# Return the number of customers in the customer table:

Command: count 'customer

2 row(s) in 0.0210 seconds

=> 2

# Delete the city column for customer 1:

Command: delete 'customer' , '1', 'address:city'.

2 row(s) in 0.0210 seconds

=> 2

# Delete all customer 2 boxes:

deleteall 'customer', '2'

# version history to a column :

alter 'customer',{NAME=>'address',VERSIONS=>5}

To test the result of VERSIONS, we insert 3 different values for a client 3 and display the second version.

put 'customer', '3', 'address:city','Paris' put 'customer', '3', 'address:city','Lyon' put 'customer', '3', 'address:city','Nantes'

scan 'customer',{COLUMN=>'address:city',VERSIONS=>2}

# Delete customer table (it must be deactivated before being deleted)

Command: disable 'customer

drop 'customer