Savitribai Phule Pune University, Pune

Department of Technology

B.Sc. Data Science

SEMESTER - VI

Subject: Big Data Acquisition and Analysis - Lab

Full Name:

Roll No:

Submission Date:

Internal Examiner Signature	External Examiner Signature	Course Coordinator Signature

LAB File of Big Data Acquisition and Analysis

INDEX PAGE

Sr. No	Title of Lab	Lab Date	Submission Date	Signature / Remark
1	HDFS Tutorial	04/03/2025	11/03/2025	
2	Apache Pig Tutorial	11/03/2025	18/03/2025	
3	Apache Hive Tutorial	18/03/2025	25/03/2025	
4	HBase Tutorial	25/03/2025	02/04/2025	

LAB 1

HDFS (Hadoop Distributed File System) Basic Operations.

To be able to work with HDFS (Hadoop Distributed File System) using basic commands.

Navigating, managing, and manipulating files and directories in HDFS.

1. Listing Files and Directories in HDFS

To view files and directories in the current working directory or from a specific directory:

List files and directories in the current directory hdfs

dfs -ls

List files and directories in a specific directory

hdfs dfs -ls <directory_path> **Examples:**

```
[cloudera@quickstart ~]$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2025-04-23 10:37 new
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2025-04-23 10:37 /user/cloudera/new
[cloudera@quickstart ~]$
```

2. Creating a Directory in HDFS To

create a new directory in HDFS:

Create a directory named 'test' in the current directory

hdfs dfs -mkdir test

Create a directory named 'test' in a specific path (e.g., /user/cloudera) hdfs

dfs -mkdir /user/cloudera/test

Examples:

```
[cloudera@quickstart ~]$ hdfs dfs -mkdir test
[cloudera@quickstart ~]$ hdfs dfs -mkdir /user/cloudera/test
mkdir: `/user/cloudera/test': File exists
```

3. Uploading Files from Local to HDFS

To upload files from your local file system to HDFS:

Upload a local file to HDFS hdfs dfs -put

<local_file_path> <hdfs_destination_path> **Example:**

```
[cloudera@quickstart ~]$ echo "This is a sample text file for HDFS." > ~/sample.txt
[cloudera@quickstart ~]$ hdfs dfs -put ~/sample.txt /user/cloudera/test/
[cloudera@quickstart ~]$
```

4. Downloading Files from HDFS to Local

Download a file from HDFS to your local directory

hdfs dfs -get <hdfs_file_path> <local_destination_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -get /user/cloudera/test/sample.txt ~/downloa
ds/
[cloudera@quickstart ~]$
```

5. Creating an Empty File in HDFS

Create an empty file in the specified directory

hdfs dfs -touchz <file_path> **Example:**

```
[cloudera@quickstart ~]$ hdfs dfs -touchz /user/cloudera/test/empty_file.txt
[cloudera@quickstart ~]$
```

6. Viewing File Contents in HDFS

Show the contents of a file

hdfs dfs -cat <hdfs_file_path> **Example:**

```
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/test/sample.txt
This is a sample text file for HDFS.
[cloudera@quickstart ~]$
```

7. Deleting Files in HDFS # Remove a file from HDFS hdfs dfs -rm <hdfs_file_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -rm /user/cloudera/test/empty_file.txt
25/05/01 20:05:28 INFO fs.TrashPolicyDefault: Namenode trash configuration: Del
etion interval = 1440 minutes, Emptier interval = 0 minutes.
Moved: 'hdfs://quickstart.cloudera:8020/user/cloudera/test/empty_file.txt' to t
rash at: hdfs://quickstart.cloudera:8020/user/cloudera/.Trash/Current
[cloudera@quickstart ~]$
```

8. Additional Commands

8.1. Copying Files Within HDFS

Copy a file from one directory to another in HDFS

hdfs dfs -cp <source_path> <destination_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -cp /user/cloudera/test/sample.txt /user/clou
dera/test_backup/
[cloudera@quickstart ~]$
```

8.2. Moving/Renaming Files in HDFS

Move or rename a file in HDFS hdfs dfs -mv

<source_path> <destination_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -mv /user/cloudera/test/sample.txt /user/clou
dera/test/sample_renamed.txt
[cloudera@quickstart ~]$
```

8.3. Checking File Size and Disk Usage

Check disk usage of a directory hdfs

dfs -du <directory_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -du /user/cloudera/test
37  37  /user/cloudera/test/sample_renamed.txt
[cloudera@quickstart ~]$
```

8.4. Recursive Directory Deletion

Delete a directory and all files inside it

hdfs dfs -rm -r <directory_path>

Example:

```
[cloudera@quickstart ~]$ hdfs dfs -rm -r /user/cloudera/test_backup
25/05/01 20:15:38 INFO fs.TrashPolicyDefault: Namenode trash configuration: Del
etion interval = 1440 minutes, Emptier interval = 0 minutes.
Moved: 'hdfs://quickstart.cloudera:8020/user/cloudera/test_backup' to trash at:
 hdfs://quickstart.cloudera:8020/user/cloudera/.Trash/Current
[cloudera@quickstart ~]$
```

LAB 2

Pig Tutorial.

Getting Started:

- **Start the Cloudera VM** using VirtualBox
- **Open Cloudera Terminal**
- Start services if not already running
- **Pig Startup Commands:**
  - MapReduce Mode: pig
  - Local Mode: pig -x local
- **Exit Pig**: exit or Ctrl+C

1. Load and Store commands

```
data = LOAD 'data/data-bag.txt' USING PigStorage(',');
DUMP data;
STORE data INTO 'data/output/load-store' USING PigStorage('|');
```

```
grunt> data = LOAD 'data-bag.txt' USING PigStorage(',');
grunt> DUMP data;
til - Total input paths to process : 1
(1,2,3)
(1,2,3)
(1,2,4)
(2,3,4)
(3,4,5)
(4,5,6)
(4,5,6)
grunt>
```

```
grunt> STORE data INTO 'data/output/load-store' USING PigStorage('|');
Counters:
Total records written : 7
Total bytes written : 42
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1746200053197_0002
```

2. **Load with Multiple Delimiters** outerbag = LOAD 'data/multi-delimiter.txt' USING PigStorage('\t') AS (f1, f2); innerbag = FOREACH outerbag GENERATE f1, STRSPLIT(f2, ','); DUMP innerbag;

```
grunt> outerbag = LOAD 'multi-delimiter.txt' USING PigStorage('\t') AS (f1, f2);
2025-05-02 09:07:39,320 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 1 time(s).

grunt> innerbag = FOREACH outerbag GENERATE f1, STRSPLIT(f2, ',');
2025-05-02 09:08:28,951 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 2 time(s).

til - Total input paths to process : 1
(key1      value11,value12,value13,)
(key2      value21,value22,value23,)
(key3      value31,value32,value33,)
(key4      value41,value42,value43,)
grunt>
```

3. **Access Specific Columns** data = LOAD 'data/nested-schema.txt' AS (f1:int, f2:bag{t:tuple(n1:int, n2:int)}, f3:map[]); by_pos = FOREACH data GENERATE $0; DUMP by_pos;

by_field = FOREACH data GENERATE f2; DUMP by_field;

by_map = FOREACH data GENERATE f3#'name'; DUMP

by_map;

```
grunt> data = LOAD 'nested-schema.txt' AS (f1:int, f2:bag{t:tuple(n1:int,n2:int)},f3:map[]);
2025-05-02 09:20:19,749 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:20:19,750 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt>
grunt> by_pos = FOREACH data GENERATE $0;
2025-05-02 09:21:32,064 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:21:32,064 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt>
til - Total input paths to process : 1
()
()
()
()
grunt>
grunt> by_field = FOREACH data GENRATE f2;
2025-05-02 09:25:39,092 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1200: <line 11, c
olumn 24>  Syntax error, unexpected symbol at or near 'GENRATE'
Details at logfile: /home/cloudera/pig_1746200195561.log
grunt>
grunt> by_field = FOREACH data GENERATE f2;
2025-05-02 09:27:00,367 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:27:00,367 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt> DUMP by field;

grunt> by_map = FOREACH data GENERATE f3#'name';
2025-05-02 09:28:58,532 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:28:58,532 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt>
```

4. Group By Operator

outerbag = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int); DUMP outerbag;

innerbag = GROUP outerbag BY f1; DUMP innerbag;

```
grunt> outerbag = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int,f3:int);
2025-05-02 09:40:58,976 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:40:58,977 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt>
```

```
til - Total input paths to process : 1
(1,2,3)
(1,2,3)
(1,2,4)
(2,3,4)
(3,4,5)
(4,5,6)
(4,5,6)
grunt>
```

```
grunt> innerbag = GROUP outerbag BY f1;
2025-05-02 09:41:34,793 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:41:34,793 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt>
```

```
til - Total input paths to process : 1
(1,{(1,2,4),(1,2,3),(1,2,3)})
(2,{(2,3,4)})
(3,{(3,4,5)})
(4,{(4,5,6),(4,5,6)})
grunt>
```

5. Filter Operator

data = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int); filtered = FILTER data BY f1 == 1; DUMP filtered;

```
grunt> data = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:
int, f3:int);
grunt> filtered = FILTER data BY f1 == 1;
grunt> DUMP filtered;
```

```
til - Total input paths to process : 1
(1,2,3)
(1,2,3)
(1,2,4)
grunt>
```

6. Count Operator

data = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int);
grouped = GROUP data BY f2; counted = FOREACH grouped GENERATE group, COUNT(data); DUMP counted;

```
grunt> data = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int,f3:int);
2025-05-02 09:56:38,234 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:56:38,234 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt> DUMP data;
```

```
grunt> grouped = GROUP data BY f2;
2025-05-02 09:59:13,242 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:59:13,242 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt> counted = FOREACH grouped GENERATE group, COUNT(data);
2025-05-02 09:59:25,523 [main] WARN  org.apache.pig.PigServer - Encountered Warning IMPLICIT_CA
ST_TO_CHARARRAY 3 time(s).
2025-05-02 09:59:25,523 [main] WARN  org.apache.pig.PigServer - Encountered Warning USING_OVERL
OADED_FUNCTION 3 time(s).
grunt> DUMP counted;
2025-05-02 09:59:34,839 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features u
sed in the script: GROUP BY
```

```
til - Total input paths to process : 1
(2,3)
(3,1)
(4,1)
(5,2)
grunt>
```

7. Order By Operator

data = LOAD 'data/nested-sample.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int);

ordera = ORDER data BY f1 ASC; DUMP ordera;

```
grunt> data = LOAD 'nested-sample.txt' USING PigStorage(',') AS (f1:int, f2:int
, f3:int);
grunt> ordera = ORDER data BY f1 ASC;
grunt> DUMP ordera;
```

orderd = ORDER data BY f1 DESC; DUMP orderd;

```
grunt> orderd = ORDER data BY f1 DESC;
grunt> DUMP orderd;
```

8. Distinct Operator

data = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int);

unique = DISTINCT data; DUMP unique;

```
grunt> data = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int,f3:int);
grunt> unique = DISTINCT data;
grunt> DUMP unique;
```

```
til - Total input paths to process : 1
(1,2,3)
(1,2,4)
(2,3,4)
(3,4,5)
(4,5,6)
grunt>
```

9. **Limit Operator** data = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int);

limited = LIMIT data 3; DUMP limited;

```
grunt> data = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int,f3:int);
grunt> limited = LIMIT data 3;
grunt> DUMP limited;
```

```
til - Total input paths to process : 1
(1,2,3)
(1,2,3)
(1,2,4)
grunt>
```

10. **Join Operator** a = LOAD 'data/data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int, f3:int); b = LOAD 'data/simple-tuples.txt' USING PigStorage(',') AS (t1:int, t2:int); joined = JOIN a BY f1, b BY t1; DUMP joined;

```
grunt> a = LOAD 'data-bag.txt' USING PigStorage(',') AS (f1:int, f2:int,f3:int);
grunt> b = LOAD 'simple-tuples.txt' USING PigStorage(',') AS (f1:int, f2:int);
grunt> joined = JOIN a BY f1, b BY f1;
grunt> DUMP joined;
```

```
til - Total input paths to process : 1
(1,2,4,1,4)
(1,2,4,1,3)
(1,2,4,1,2)
(1,2,3,1,4)
(1,2,3,1,3)
(1,2,3,1,2)
(1,2,3,1,4)
(1,2,3,1,3)
(1,2,3,1,2)
(2,3,4,2,4)
(2,3,4,2,3)
(3,4,5,3,4)
(4,5,6,4,6)
(4,5,6,4,5)
(4,5,6,4,6)
(4,5,6,4,5)
grunt>
```

LAB 3

Apache Hive Tutorial Basic Operations.

Getting Started:

- **Start the Cloudera VM**
- **Open Cloudera Terminal**
- Start services if not already running

1. Start Hive

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in jar:file:/usr/jars/hive-common-1.1.0-
cdh5.4.2.jar!/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
```

2. Show Existing Tables

```
hive> SHOW TABLES;
OK
Time taken: 4.331 seconds
hive>
```

3. Exit Hive #Exit Hive

exit;

4. Create Hive Table

CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]hiking
(
  col_name data_type [COMMENT col_comment], ...
)
[COMMENT table_comment] [ROW
FORMAT row_format]
[STORED AS file_format];

```
hive> CREATE TABLE IF NOT EXISTS hiking(
    > id INT,
    > name STRING,
    > region STRING,
    > distance FLOAT,
    > altitude INT,
    > suiteHiking INT
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > LINES TERMINATED BY '\n';
OK
Time taken: 3.04 seconds
hive>
```

5. Show Table Structure

#Show structure of the hiking table

```
hive> DESCRIBE hiking;
OK
id                      int
name                    string
region                  string
distance                float
altitude                int
suitehiking             int
Time taken: 0.834 seconds, Fetched: 6 row(s)
hive>
```

6. Insert Data into Table

```
hive> INSERT INTO TABLE hiking VALUES
    > (1, 'AniketWalunj','Pune', 35, 1000, NULL),
    > (2, 'YashrajBhosale', 'Solarpur', 25, 514, NULL),
    >
    >
    >
    > (3, 'RevatiJadhav','Nagpur',  21, 1100, NULL),
    > (4, 'SampadaSwami', 'Pune', 18, 890, 3),
    > (5, 'SarthakPatil', 'Pune', 19,900,NULL),
    > (6, 'AtharvaKadam', 'UK', 8, 165, 8),
    > (7, 'PrathmeshGadsing', 'Nashik', 10, 1900, NULL),
    > (8, 'YuvrajPondkule', 'Mumbai', 14.18, 454, NULL),
    > (9, 'VaishnaviBhosale','Delhi', 6.25, 1548, 1);
Query ID = cloudera_20250501212525_e47a888d-166f-4857-af4f-4f2d860c4963
Total jobs = 3
```

✞ Run Hive Queries on hiking

To show all data from hiking table

1. Hikes More Than 20 KM

```
hive> SELECT * FROM hiking;
OK
1       AniketWalunj    Pune    35.0    1000    NULL
2       YashrajBhosale  Solarpur        25.0    514     NULL
3       RevatiJadhav    Nagpur  21.0    1100    NULL
4       SampadaSwami    Pune    18.0    890     3
5       SarthakPatil    Pune    19.0    900     NULL
6       AtharvaKadam    UK      8.0     165     8
7       PrathmeshGadsing        Nashik  10.0    1900    NULL
8       YuvrajPondkule  Mumbai  14.18   454     NULL
9       VaishnaviBhosale        Delhi   6.25    1548    1
Time taken: 0.419 seconds, Fetched: 9 row(s)
hive>
```

2. Hikes That Have a Suite

```
hive> SELECT * FROM hiking WHERE distance >=20;
OK
1       AniketWalunj    Pune    35.0    1000    NULL
2       YashrajBhosale  Solarpur        25.0    514     NULL
3       RevatiJadhav    Nagpur  21.0    1100    NULL
Time taken: 0.346 seconds, Fetched: 3 row(s)
hive>
```

3. Maximum Distance by Region

SELECT region, MAX(distance) AS max FROM hiking GROUP BY region;

```
hive> SELECT * FROM hiking WHERE suiteHiking IS NOT NULL;
OK
4       SampadaSwami    Pune    18.0    890     3
6       AtharvaKadam    UK      8.0     165     8
9       VaishnaviBhosale        Delhi   6.25    1548    1
Time taken: 0.259 seconds, Fetched: 3 row(s)
hive>
```

4. Average Distance by Region

SELECT region, AVG(distance) AS avg FROM hiking GROUP BY region;

```
Total MapReduce CPU Time Spent: 5 seconds 300 msec
OK
Delhi   6.25
Mumbai  14.18
Nagpur  21.0
Nashik  10.0
Pune    35.0
Solarpur        25.0
UK      8.0
Time taken: 78.726 seconds, Fetched: 7 row(s)

Total MapReduce CPU Time Spent: 5 seconds 230 msec
OK
Delhi   6.25
Mumbai  14.180000305175781
Nagpur  21.0
Nashik  10.0
Pune    24.0
Solarpur        25.0
UK      8.0
Time taken: 161.147 seconds, Fetched: 7 row(s)
hive>
```

LAB 4

Hbase Tutorial Basic Operations.

Getting Started:

- **Start the Cloudera VM**
- **Open Cloudera Terminal**
- Start services if not already running

1. Start and Stop HBase Shell

Start HBase Shell hbase shell

```
[cloudera@quickstart ~]$ hbase shell
25/05/01 21:52:38 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use
 io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.0.0-cdh5.4.2, rUnknown, Tue May 19 17:07:29 PDT 2015

hbase(main):001:0>
```

2. get Hbase Shell command list

table_help

3. Show all tables

```
hbase(main):002:0> list
TABLE
0 row(s) in 0.7240 seconds

=> []
hbase(main):003:0>
```

4. Check if a table exists:

exists 'table_name'

5. Exit HBase shell:

exit

1. Create HBase Table: hiking

```
hbase(main):003:0> create 'table_name', 'column_family1', 'column_family2', ...
hbase(main):004:0*
```

2. Describe table structure

```
hbase(main):004:0* discribe 'table_name'
SyntaxError: (hbase):3: syntax error, unexpected tDOT3

create 'table_name', 'column_family1', 'column_family2', ...
                                                        ^


hbase(main):005:0>
```

3. Command to create hiking table

```
hbase(main):003:0> create 'hiking', 'InfoHiking', 'InfoTechnique'
0 row(s) in 4.5370 seconds

=> Hbase::Table - hiking
hbase(main):004:0>
```

4. Insert Data (PUT Command)

```
hbase(main):006:0' put 'table_name', 'row_id', 'column_family:column', 'value'
hbase(main):007:0'
```

Example Inserts

```
hbase(main):010:0> put 'hiking', '1','InfoHiking:name','Monts du Djurdjura'
0 row(s) in 0.5810 seconds

hbase(main):011:0> put 'hiking', '1', 'InfoHiking:region' , 'Tizi Quzou'
0 row(s) in 0.0100 seconds

hbase(main):012:0> put 'hiking', '1', 'InfoHiking:distance', 35
0 row(s) in 0.0580 seconds

hbase(main):013:0> put 'hiking', '1', 'InfoHiking:altitude', 1000
0 row(s) in 0.0170 seconds



hbase(main):015:0> put 'hiking','2', 'InfoHiking:name','Ciruit de Misserghin'
0 row(s) in 0.0430 seconds

hbase(main):016:0> put 'hiking','2', 'InfoHiking:region', 'Oran'
0 row(s) in 0.0160 seconds

hbase(main):017:0> put 'hiking','2', 'InfoHiking:distance', 25
0 row(s) in 0.0490 seconds

hbase(main):018:0> put 'hiking','2', 'InfoHiking:altitude', 514
0 row(s) in 0.0330 seconds
```

```
hbase(main):019:0> put 'hiking','3', 'InfoHiking:name', 'Montagne de Murdjadju'
0 row(s) in 0.0370 seconds

hbase(main):020:0> put 'hiking','3', 'InfoHiking:region', 'Oran'
0 row(s) in 0.0410 seconds

hbase(main):021:0> put 'hiking','3', 'InfoHiking:distance', 31
0 row(s) in 0.0310 seconds

hbase(main):022:0> put 'hiking','3', 'InfoHiking:altitude', 1100
0 row(s) in 0.0250 seconds


hbase(main):023:0> put 'hiking','4', 'InfoHiking:name', 'Canastel'
0 row(s) in 0.0320 seconds

hbase(main):024:0> put 'hiking','4', 'InfoHiking:region', 'Oran'
0 row(s) in 0.0130 seconds

hbase(main):025:0> put 'hiking','4', 'InfoHiking:distance', 18
0 row(s) in 0.0080 seconds

hbase(main):026:0> put 'hiking','4', 'InfoHiking:altitude', 890
0 row(s) in 0.0630 seconds

hbase(main):027:0> put 'hiking','4', 'InfoHiking:suiteHiking', 3
0 row(s) in 0.0080 seconds


hbase(main):028:0> put 'hiking','5', 'InfoHiking:name', 'Yama Gourya'
0 row(s) in 0.0130 seconds

hbase(main):029:0> put 'hiking','5', 'InfoHiking:region', 'Bejaia'
0 row(s) in 0.0170 seconds

hbase(main):030:0> put 'hiking','5', 'InfoHiking:distance', 19
0 row(s) in 0.0300 seconds

hbase(main):031:0> put 'hiking','5', 'InfoHiking:altitude', 900
0 row(s) in 0.0290 seconds
```

```
hbase(main):032:0> put 'hiking','6', 'InfoHiking:name', 'Sidi Makhlouf'
0 row(s) in 0.0420 seconds

hbase(main):033:0> put 'hiking','6', 'InfoHiking:region', 'Blida'
0 row(s) in 0.0270 seconds

hbase(main):034:0> put 'hiking','6', 'InfoHiking:distance', 8
0 row(s) in 0.0070 seconds

hbase(main):035:0> put 'hiking','6', 'InfoHiking:altitude', 165
0 row(s) in 0.0590 seconds

hbase(main):036:0> put 'hiking','6', 'InfoHiking:suiteHiking', 8
0 row(s) in 0.0180 seconds


hbase(main):037:0> put 'hiking','7', 'InfoHiking:name', 'Tikjda'
0 row(s) in 0.0120 seconds

hbase(main):038:0> put 'hiking','7', 'InfoHiking:region', 'Tizi Ouzou'
0 row(s) in 0.0320 seconds

hbase(main):039:0> put 'hiking','7', 'InfoHiking:distance', 10
0 row(s) in 0.0390 seconds

hbase(main):040:0> put 'hiking','7', 'InfoHiking:altitude', 1900
0 row(s) in 0.0270 seconds


hbase(main):041:0> put 'hiking','8', 'InfoHiking:name', 'Feroukha'
0 row(s) in 0.0150 seconds

hbase(main):042:0> put 'hiking','8', 'InfoHiking:region', 'Blida'
0 row(s) in 0.0290 seconds

hbase(main):043:0> put 'hiking','8', 'InfoHiking:distance', 14.18
0 row(s) in 0.0240 seconds

hbase(main):044:0> put 'hiking','8', 'InfoHiking:altitude', 454
0 row(s) in 0.0130 seconds


hbase(main):045:0> put 'hiking','9', 'InfoHiking:name','Chrea Azzazga'
0 row(s) in 0.0140 seconds

hbase(main):046:0> put 'hiking','9', 'InfoHiking:region', 'Tizi Ouzou'
0 row(s) in 0.0110 seconds

hbase(main):047:0> put 'hiking','9', 'InfoHiking:distance', 6.23
0 row(s) in 0.0120 seconds

hbase(main):048:0> put 'hiking','9', 'InfoHiking:altitude', 1548
0 row(s) in 0.0110 seconds

hbase(main):049:0> put 'hiking','9', 'InfoHiking:suiteHiking', 11
0 row(s) in 0.0130 seconds
```

✞ Update Data in HBase

Update distance of hiking with ID 1:

```
hbase(main):050:0> put 'hiking','1', 'InfoHiking:distance', 55
0 row(s) in 0.0120 seconds
```

Get Data from Table get 'table_name',

'row_id' get 'table_name', 'row_id',

'column_family' get

'table_name', 'row_id', 'column_family:column'

Examples:

Get all data for ID 4 get

'hiking', '4'

```
hbase(main):051:0> get 'hiking', '4'
COLUMN                  CELL
 InfoHiking:altitude    timestamp=1746168488732, value=890
 InfoHiking:distance    timestamp=1746168477680, value=18
 InfoHiking:name        timestamp=1746168446249, value=Canastel
 InfoHiking:region      timestamp=1746168460892, value=Oran
 InfoHiking:suiteHiking  timestamp=1746168501182, value=3
5 row(s) in 0.0160 seconds
```

Get InfoHiking data of ID 5 get

'hiking', '5', 'InfoHiking'

```
hbase(main):052:0> get 'hiking', '5', 'InfoHiking'
COLUMN                  CELL
 InfoHiking:altitude    timestamp=1746168654533, value=900
 InfoHiking:distance    timestamp=1746168641922, value=19
 InfoHiking:name        timestamp=1746168612530, value=Yama Gourya
 InfoHiking:region      timestamp=1746168633092, value=Bejaia
4 row(s) in 0.0130 seconds
```

Filters in HBase Show

available filters:

show_filters

Get distance of 'Montagne de Murdjadju':

```
hbase(main):005:0> scan 'hiking', { FILTER => "SingleColumnValueFilter('InfoHiking','name', = 'binary:Montagne
ROW                                                     COLUMN+CELL
 3                                                      column=InfoHiking:distance, timestamp=174616831971
1 row(s) in 0.1870 seconds

hbase(main):006:0>
```

Get distance of all hiking in region 'Tizi Ouzou':

```
hbase(main):006:0> scan 'hiking', { FILTER => "SingleColumnValueFilter('InfoHiking','region', = 'binary:Tizi Qu
zou') AND QualifierFilter(=,'binary:distance')"}
ROW                           COLUMN+CELL
 1                            column=InfoHiking:distance, timestamp=1746169157067, value=55
1 row(s) in 0.0240 seconds

hbase(main):007:0>
```