MASTER IN DATA SCIENCE

INDIVIDUAL ASSIGNMENT

COURSE CODE : WQD7009

COURSE TITLE : Big Data Application and Analytics

NAME : Nur Hidayah binti Ahmad Shafii

MATRIC NUMBER : 22120931

LECTURER : Dr Riyaz Ahamed Ariyaluran Habeeb Mohamed

1 Introduction

According to the United Nations' Sustainable Development Goals (SDGs), agricultural sustainability was one of the key components to reduce climate change and improve food security. This project takes a dataset from Kaggle (https://www.kaggle.com/datasets/bhadramohit/agriculture-and-farming-dataset) to get an insight into productivity, resource management, and sustainability in agricultural operations. The dataset contains 10 columns and 50 rows. It consists of farm area, crop types, irrigation, fertilizer and pesticide usage, yield, soil type, season, and water consumption of farming activity in India. In India, the farming seasons are categorised into Kharif, Rabi and Zaid. Kharif crops are known as monsoon crops and harvested from June to September. Rabi crops are known as winter crops and harvested from October to March. Zaid crops are summer season crops and are harvested from March to June (Bates, 2020).

The dataset presented in CSV format will be analysed using the HBase NoSQL platform to understand farming practices and their impact on sustainability. In the project, , HBase shell commands will be utilised to check the status of the HBase operations and ensure smooth execution. Additionally, we will execute a data manipulation query using DDL and DML commands to get insight into factors such as techniques, soil types, and resource management practices. A detailed description of the dataset parameters was provided in Table 1.1

Table 1.1 Overview of Dataset Parameters

Parameters	Description
Farm ID	Unique identifier for each farm
Crop Type	Type of plant that was grown in the farm
Farm Area (acres)	Size of farm area in acres
Irrigation Type	Type of irrigation used
Fertilizer Used (tons)	Amount of fertilizer used in tons
Pesticide Used (kg)	Amount of pesticide used in kilograms
Yield (tons)	Total crop yield in tons
Soil Type	Type of soil used on the farm
Season	Farming season (Kharif, Rabi and Zaid)
Water Usage (cubic meters)	Total water usage in cubic meters

The dataset was pre-processed before it was imported to HBase to improve the effectiveness of data storage and querying. In the season column, we will need to replace Kharif, Rabi, and Zaid with Monsoon, Winter, and Summer, respectively. Next, the dataset was checked for missing values, and any detected outliers are removed. After checking that there are no missing values or outliers, the dataset was then converted into a text-delimited file to be imported into Hbase.

2 Query Result

The following command in the Windows Command Prompt was used to import the file:

```
scp -o HostKeyAlgorithms=+ssh-rsa C:\Users\User\Downloads\agriculture.txt cloudera@192.168.56.104:/home/cloudera/
```

The file will be transferred to the /home/cloudera/ directory as shown in Figure 2.1. Then, the file was transferred to “hbase_data” folder in HDFS before ingesting to HBase for optimisation.

```
File  Edit  View  Search  Terminal  Help
[cloudera@quickstart Desktop]$ ls /home/cloudera/
agriculture.txt  Documents                  lib              Templates
cloudera-manager Downloads                  Music            Videos
cm_api.py        eclipse                    parcels          WordCount.jar
datasetlab4.txt  enterprise-deployment.json Pictures         workspace
dataset.txt      express-deployment.json    Processfile.txt
Desktop          kerberos                   Public
[cloudera@quickstart Desktop]$ hdfs dfs -ls
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2024-11-08 23:04 my_spark_output1
[cloudera@quickstart Desktop]$ hdfs dfs -ls/
-ls/: Unknown command
[cloudera@quickstart Desktop]$ hdfs dfs -ls /
Found 11 items
drwxrwxrwx   - hdfs     supergroup          0 2017-04-05 04:27 /benchmarks
drwxr-xr-x   - hbase    supergroup          0 2024-10-27 02:43 /hbase
drwxr-xr-x   - cloudera supergroup          0 2024-10-20 08:33 /inputfolder1
drwxr-xr-x   - cloudera supergroup          0 2024-10-23 21:58 /inputfolder2
drwxr-xr-x   - cloudera supergroup          0 2024-10-20 08:36 /out1
drwxr-xr-x   - cloudera supergroup          0 2024-10-23 22:46 /out2
drwxr-xr-x   - solr     solr                0 2017-04-05 04:29 /solr
drwxr-xr-x   - cloudera supergroup          0 2024-11-08 22:43 /sparkdata
drwxrwxrwt   - hdfs     supergroup          0 2024-10-27 03:06 /tmp
drwxr-xr-x   - hdfs     supergroup          0 2024-10-27 03:06 /user
drwxr-xr-x   - hdfs     supergroup          0 2017-04-05 04:29 /var
[cloudera@quickstart Desktop]$ hdfs dfs -mkdir /hbase_data
[cloudera@quickstart Desktop]$ hdfs dfs -put /home/cloudera/agriculture.txt /hbase_data/
```

Figure 2.1 File Transfer Output

Next, `sudo jps` will ensure we have all the permwassion to check the master server. Then, start the HBase Master service using the `sudo service hbase-master start` command and use `hbase shell` to interact with HBase through its command-line interface.

The HBase shell commands to check the status of HBase operations are version, whoami, status, status ‘detailed’ and table_help as shown in Figure 2.2. The version used was 1.2.0-cdh5.10.0. The root user name was cloudera and default namespace was being used. The status showed that there was one active master, one server and 3000 average load for Hbase. For more information on the Hbase, we use the `status 'detailed'` command.

```
[cloudera@quickstart Desktop]$ sudo jps
6815 RunJar
5794 JobHistoryServer
1341 HMaster
6707 ThriftServer
8133 Bootstrap
6167 ResourceManager
5463 NameNode
5597 SecondaryNameNode
7325 Bootstrap
7357 HistoryServer
6901 RunJar
5874 NodeManager
8208
12817 Jps
5204 QuorumPeerMain
7882 Bootstrap
5275 DataNode
5376 JournalNode
5745 Bootstrap
6578 RESTServer
8159
[cloudera@quickstart Desktop]$ sudo service hbase-master start
HBase master daemon is running                             [  OK  ]
[cloudera@quickstart Desktop]$ hbase shell
2024-11-27 11:47:04,127 INFO  [main] Configuration.deprecation: hadoop.native.li
b 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 1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

hbase(main):001:0> version
1.2.0-cdh5.10.0, rUnknown, Fri Jan 20 12:13:18 PST 2017

hbase(main):002:0> whoami
cloudera (auth:SIMPLE)
    groups: cloudera, default

hbase(main):003:0>
```

```
hbase(main):004:0> status
1 active master, 0 backup masters, 1 servers, 0 dead, 3.0000 average load

hbase(main):005:0> status 'detailed'
version 1.2.0-cdh5.10.0
0 regionsInTransition
active master:  quickstart.cloudera:60000 1732738612166
0 backup masters
master coprocessors: []
1 live servers
    quickstart.cloudera:60020 1732738623731
        requestsPerSecond=0.0, numberOfOnlineRegions=3, usedHeapMB=32, maxHeapMB=941, numberOfStores=3, numberOfStore
files=2, storefileUncompressedSizeMB=0, storefileSizeMB=0, memstoreSizeMB=0, storefileIndexSizeMB=0, readRequestsCoun
t=33, writeRequestsCount=6, rootIndexSizeKB=0, totalStaticIndexSizeKB=0, totalStaticBloomSizeKB=0, totalCompactingKVs
=0, currentCompactedKVs=0, compactionProgressPct=NaN, coprocessors=[MultiRowMutationEndpoint]
        "agriculture,,1732738684241.0ad3bbab997ba90a101bcb8360ecc3b4."
            numberOfStores=1, numberOfStorefiles=0, storefileUncompressedSizeMB=0, lastMajorCompactionTimestamp=0, st
orefileSizeMB=0, memstoreSizeMB=0, storefileIndexSizeMB=0, readRequestsCount=0, writeRequestsCount=0, rootIndexSizeKB
=0, totalStaticIndexSizeKB=0, totalStaticBloomSizeKB=0, totalCompactingKVs=0, currentCompactedKVs=0, compactionProgre
ssPct=NaN, completeSequenceId=-1, dataLocality=0.0
        "hbase:meta,,1"
```

```
hbase(main):007:0> table_help
Help for table-reference commands.

You can either create a table via 'create' and then manipulate the table via commands like 'put', 'get', etc.
See the standard help information for how to use each of these commands.

However, as of 0.96, you can also get a reference to a table, on which you can invoke commands.
For instance, you can get create a table and keep around a reference to it via:

   hbase> t = create 't', 'cf'

Or, if you have already created the table, you can get a reference to it:

   hbase> t = get_table 't'

You can do things like call 'put' on the table:
```

Figure 2.2 HBase shell commands

Firstly, we created agriculture table with one column family named ‘cf’ as shown in Figure 2.3. Then, we will import the data from the hdfs to Hbase using ImportTsv DML command:

```
hbase org.apache.hadoop.hbase.mapreduce.ImportTsv \
  -
Dimporttsv.columns=HBASE_ROW_KEY,cf:Crop_Type,cf:Farm_Area,cf:Irriga
tion_Type,cf:Fertilizer_Used,cf:Pesticide_Used,cf:Yield,cf:Soil_Type
,cf:Season,cf:Water_Usage \
  agriculture \
  /hbase_data/agriculture.txt
```

```
hbase(main):001:0> create 'agriculture','cf'
0 row(s) in 1.9700 seconds

=> Hbase::Table - agriculture
```

```
[cloudera@quickstart Desktop]$ hbase org.apache.hadoop.hbase.mapreduce.ImportTsv
 \
>   -Dimporttsv.columns=HBASE_ROW_KEY,cf:Crop_Type,cf:Farm_Area,cf:Irrigation_Ty
pe,cf:Fertilizer_Used,cf:Pesticide_Used,cf:Yield,cf:Soil_Type,cf:Season,cf:Water
_Usage \
>   agriculture \
>   /hbase_data/agriculture.txt
2024-11-27 14:14:03,532 INFO  [main] zookeeper.RecoverableZooKeeper: Process ide
ntifier=hconnection-0x503fcce8 connecting to ZooKeeper ensemble=localhost:2181
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:zoo
keeper.version=3.4.5-cdh5.10.0--1, built on 01/20/2017 20:10 GMT
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:hos
t.name=quickstart.cloudera
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.version=1.7.0_67
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.vendor=Oracle Corporation
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.home=/usr/java/jdk1.7.0_67-cloudera/jre
2024-11-27 14:14:03,568 INFO  [main] zookeeper.ZooKeeper: Client environment:jav
a.class.path=/usr/lib/hbase/bin/../conf:/usr/java/jdk1.7.0_67-cloudera/lib/tools
.jar:/usr/lib/hbase/bin/..:/usr/lib/hbase/bin/../lib/activation-1.1.jar:/usr/lib
/hbase/bin/../lib/apacheds-i18n-2.0.0-M15.jar:/usr/lib/hbase/bin/../lib/apacheds
-kerberos-codec-2.0.0-M15.jar:/usr/lib/hbase/bin/../lib/api-asn1-api-1.0.0-M20.j
ar:/usr/lib/hbase/bin/../lib/api-util-1.0.0-M20.jar:/usr/lib/hbase/bin/../lib/as
m-3.2.jar:/usr/lib/hbase/bin/../lib/avro.jar:/usr/lib/hbase/bin/../lib/aws-java-
sdk-core-1.10.6.jar:/usr/lib/hbase/bin/../lib/aws-java-sdk-kms-1.10.6.jar:/usr/l
ib/hbase/bin/../lib/aws-java-sdk-s3-1.10.6.jar:/usr/lib/hbase/bin/../lib/aws-jav
a-sdk-sts-1.10.6.jar:/usr/lib/hbase/bin/../lib/commons-beanutils-1.9.2.jar:/usr/
lib/hbase/bin/../lib/commons-beanutils-core-1.8.0.jar:/usr/lib/hbase/bin/../lib/
commons-cli-1.2.jar:/usr/lib/hbase/bin/../lib/commons-codec-1.9.jar:/usr/lib/hba
se/bin/../lib/commons-collections-3.2.2.jar:/usr/lib/hbase/bin/../lib/commons-co
mpress-1.4.1.jar:/usr/lib/hbase/bin/../lib/commons-configuration-1.6.jar:/usr/li
b/hbase/bin/../lib/commons-daemon-1.0.13.jar:/usr/lib/hbase/bin/../lib/commons-d
igester-1.8.jar:/usr/lib/hbase/bin/../lib/commons-el-1.0.jar:/usr/lib/hbase/bin/
../lib/commons-httpclient-3.1.jar:/usr/lib/hbase/bin/../lib/commons-io-2.4.jar:/
usr/lib/hbase/bin/../lib/commons-lang-2.6.jar:/usr/lib/hbase/bin/../lib/commons-
logging-1.2.jar:/usr/lib/hbase/bin/../lib/commons-math-2.1.jar:/usr/lib/hbase/bi
n/../lib/commons-math3-3.1.1.jar:/usr/lib/hbase/bin/../lib/commons-net-3.1.jar:/
usr/lib/hbase/bin/../lib/core-3.1.1.jar:/usr/lib/hbase/bin/../lib/curator-client
-2.7.1.jar:/usr/lib/hbase/bin/../lib/curator-framework-2.7.1.jar:/usr/lib/hbase/
bin/../lib/curator-recipes-2.7.1.jar:/usr/lib/hbase/bin/../lib/disruptor-3.3.0.j
ar:/usr/lib/hbase/bin/../lib/findbugs-annotations-1.3.9-1.jar:/usr/lib/hbase/bin
/../lib/gson-2.2.4.jar:/usr/lib/hbase/bin/../lib/guava-12.0.1.jar:/usr/lib/hbase
/bin/../lib/hamcrest-core-1.3.jar:/usr/lib/hbase/bin/../lib/hbase-annotations-1.
2.0-cdh5.10.0.jar:/usr/lib/hbase/bin/../lib/hbase-annotations-1.2.0-cdh5.10.0-te
sts.jar:/usr/lib/hbase/bin/../lib/hbase-client-1.2.0-cdh5.10.0.jar:/usr/lib/hbas
```

Figure 2.3 Importing Data to HBase

The output in Figure 2.4 shows that the data was successfully imported in the HBase. It shows information such the number of records and CPU time.

```
or job: job_1729436882035_0004
2024-11-27 14:14:11,316 INFO  [main] impl.YarnClientImpl: Submitted application
application_1729436882035_0004
2024-11-27 14:14:11,429 INFO  [main] mapreduce.Job: The url to track the job: ht
tp://quickstart.cloudera:8088/proxy/application_1729436882035_0004/
2024-11-27 14:14:11,430 INFO  [main] mapreduce.Job: Running job: job_17294368820
35_0004
2024-11-27 14:14:27,315 INFO  [main] mapreduce.Job: Job job_1729436882035_0004 running in uber mode : false
2024-11-27 14:14:27,322 INFO  [main] mapreduce.Job:  map 0% reduce 0%
2024-11-27 14:14:43,700 INFO  [main] mapreduce.Job:  map 100% reduce 0%
2024-11-27 14:14:43,736 INFO  [main] mapreduce.Job: Job job_1729436882035_0004 completed successfully
2024-11-27 14:14:44,060 INFO  [main] mapreduce.Job: Counters: 31
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=154724
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3480
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=2
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=13293
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=13293
                Total vcore-seconds taken by all map tasks=13293
                Total megabyte-seconds taken by all map tasks=13612032
        Map-Reduce Framework
                Map input records=51
                Map output records=51
                Input split bytes=123
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=173
                CPU time spent (ms)=2340
                Physical memory (bytes) snapshot=246661120
                Virtual memory (bytes) snapshot=1572061184
                Total committed heap usage (bytes)=175636480
        ImportTsv
                Bad Lines=0
        File Input Format Counters
                Bytes Read=3357
        File Output Format Counters
                Bytes Written=0
[cloudera@quickstart Desktop]$
```

Figure 2.4 ImportTsv Output

From the Figure 2.5, the `describe 'agriculture'` DDL command showed that agriculture table to have a single column family (cf) and only the latest version of data was saved. Since the TTL for data was set to 'FOREVER', the data will not expire automatically. The `lwast` command was used to dwasplay all the tables that in the HBase. Since the agriculture table we created was exist, we use the `count 'agriculture'` DDL command to check the number of rows in the agriculture table. It was showed that there are 51 rows as counted in 0.3270 seconds. Next, the `scan 'agriculture'` DML command was to check if the data was correct.

```
hbase(main):003:0> describe 'agriculture'
Table agriculture is ENABLED
agriculture
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf', DATA_BLOCK_ENCODING => 'NONE', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', COMPRE
SSION => 'NONE', MIN_VERSIONS => '0', TTL => 'FOREVER', KEEP_DELETED_CELLS => 'FALSE', BLOCKSIZE => '65536', IN_MEMOR
Y => 'false', BLOCKCACHE => 'true'}
1 row(s) in 0.0680 seconds

hbase(main):001:0> list
TABLE
agriculture
1 row(s) in 0.4310 seconds

=> ["agriculture"]
hbase(main):002:0> count 'agriculture'
51 row(s) in 0.3270 seconds

=> 51
hbase(main):003:0> scan 'agriculture'
ROW                             COLUMN+CELL
 F001                           column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F001                           column=cf:Farm_Area, timestamp=1732745642435, value=329.4
 F001                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=8.14
 F001                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Sprinkler
 F001                           column=cf:Pesticide_Used, timestamp=1732745642435, value=2.21
 F001                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F001                           column=cf:Soil_Type, timestamp=1732745642435, value=Loamy
 F001                           column=cf:Water_Usage, timestamp=1732745642435, value=76648.2
 F001                           column=cf:Yield, timestamp=1732745642435, value=14.44
 F002                           column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F002                           column=cf:Farm_Area, timestamp=1732745642435, value=18.67
 F002                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=4.77
 F002                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Manual
 F002                           column=cf:Pesticide_Used, timestamp=1732745642435, value=4.36
 F002                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F002                           column=cf:Soil_Type, timestamp=1732745642435, value=Peaty
 F002                           column=cf:Water_Usage, timestamp=1732745642435, value=68725.54
 F002                           column=cf:Yield, timestamp=1732745642435, value=42.91
 F003                           column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F003                           column=cf:Farm_Area, timestamp=1732745642435, value=306.03
 F003                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=2.91
 F003                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Flood
 F003                           column=cf:Pesticide_Used, timestamp=1732745642435, value=0.56
 F003                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F003                           column=cf:Soil_Type, timestamp=1732745642435, value=Silty
 F003                           column=cf:Water_Usage, timestamp=1732745642435, value=75538.56
 F003                           column=cf:Yield, timestamp=1732745642435, value=33.44
 F004                           column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F004                           column=cf:Farm_Area, timestamp=1732745642435, value=380.21
 F004                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=3.32
 F004                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Rain-fed
 F004                           column=cf:Pesticide_Used, timestamp=1732745642435, value=4.35
 F004                           column=cf:Season, timestamp=1732745642435, value=Summer
 F004                           column=cf:Soil_Type, timestamp=1732745642435, value=Silty
```

Figure 2.5 HBase Describe, List, Count and Scan Output

From the output of the `scan 'agriculture'` DML command, it shows that the Row key of Farm_ID was the column headers values. To address this issue, we will remove the row using the `deleteall 'agriculture', 'Farm_ID'` DDL command as shown in Figure 2.6. Finally, the `count 'agriculture'` command was executed again to verify if there was any reduction in the number of rows. The `show_filters` showed all the available filter such as RowFilter and ValueFilter in HBase to retrieve specific data. The filters can be applied to narrow down the results based on certain conditions.

```
F047                        column=cf:Season, timestamp=1732745642435, value=Monsoon
F047                        column=cf:Soil_Type, timestamp=1732745642435, value=Sandy
F047                        column=cf:Water_Usage, timestamp=1732745642435, value=86989.88
F047                        column=cf:Yield, timestamp=1732745642435, value=31.47
F048                        column=cf:Crop_Type, timestamp=1732745642435, value=Potato
F048                        column=cf:Farm_Area, timestamp=1732745642435, value=77.39
F048                        column=cf:Fertilizer_Used, timestamp=1732745642435, value=9.34
F048                        column=cf:Irrigation_Type, timestamp=1732745642435, value=Sprinkler
F048                        column=cf:Pesticide_Used, timestamp=1732745642435, value=3
F048                        column=cf:Season, timestamp=1732745642435, value=Summer
F048                        column=cf:Soil_Type, timestamp=1732745642435, value=Silty
F048                        column=cf:Water_Usage, timestamp=1732745642435, value=5874.17
F048                        column=cf:Yield, timestamp=1732745642435, value=20.53
F049                        column=cf:Crop_Type, timestamp=1732745642435, value=Barley
F049                        column=cf:Farm_Area, timestamp=1732745642435, value=462.37
F049                        column=cf:Fertilizer_Used, timestamp=1732745642435, value=2.3
F049                        column=cf:Irrigation_Type, timestamp=1732745642435, value=Sprinkler
F049                        column=cf:Pesticide_Used, timestamp=1732745642435, value=0.14
F049                        column=cf:Season, timestamp=1732745642435, value=Monsoon
F049                        column=cf:Soil_Type, timestamp=1732745642435, value=Clay
F049                        column=cf:Water_Usage, timestamp=1732745642435, value=53879.87
F049                        column=cf:Yield, timestamp=1732745642435, value=39.51
F050                        column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
F050                        column=cf:Farm_Area, timestamp=1732745642435, value=292.25
F050                        column=cf:Fertilizer_Used, timestamp=1732745642435, value=4.08
F050                        column=cf:Irrigation_Type, timestamp=1732745642435, value=Rain-fed
F050                        column=cf:Pesticide_Used, timestamp=1732745642435, value=0.76
F050                        column=cf:Season, timestamp=1732745642435, value=Monsoon
F050                        column=cf:Soil_Type, timestamp=1732745642435, value=Silty
F050                        column=cf:Water_Usage, timestamp=1732745642435, value=90232.08
F050                        column=cf:Yield, timestamp=1732745642435, value=45.14
Farm_ID                     column=cf:Crop_Type, timestamp=1732745642435, value=Crop_Type
Farm_ID                     column=cf:Farm_Area, timestamp=1732745642435, value=Farm_Area(acres)
Farm_ID                     column=cf:Fertilizer_Used, timestamp=1732745642435, value=Fertilizer_Used(tons)
Farm_ID                     column=cf:Irrigation_Type, timestamp=1732745642435, value=Irrigation_Type
Farm_ID                     column=cf:Pesticide_Used, timestamp=1732745642435, value=Pesticide_Used(kg)
Farm_ID                     column=cf:Season, timestamp=1732745642435, value=Season
Farm_ID                     column=cf:Soil_Type, timestamp=1732745642435, value=Soil_Type
Farm_ID                     column=cf:Water_Usage, timestamp=1732745642435, value=Water_Usage(cubic meters)
Farm_ID                     column=cf:Yield, timestamp=1732745642435, value=Yield(tons)
51 row(s) in 1.1790 seconds

hbase(main):004:0> deleteall 'agriculture', 'Farm_ID'
0 row(s) in 0.2720 seconds

hbase(main):005:0> count 'agriculture'
50 row(s) in 0.1200 seconds

=> 50
hbase(main):006:0>
```

```
hbase(main):007:0> show_filters
ColumnPrefixFilter
TimestampsFilter
PageFilter
MultipleColumnPrefixFilter
FamilyFilter
ColumnPaginationFilter
SingleColumnValueFilter
RowFilter
QualifierFilter
ColumnRangeFilter
ValueFilter
PrefixFilter
SingleColumnValueExcludeFilter
ColumnCountGetFilter
InclusiveStopFilter
DependentColumnFilter
FirstKeyOnlyFilter
KeyOnlyFilter
```

Figure 2.6 Deleteall, Count and Show_filter Output

Instead of showing all the data in the agriculture table, we can filter the row keys with digits less than 10 using `RowFilter` DML command as shown in Figure 2.7. It only showed the row keys start with F00 followed by any digit between 0 and 9. The command used:

```
scan 'agriculture', {FILTER => "RowFilter(=, 'regexstring:F00[0-9]{1}')"}
```

```
hbase(main):006:0> scan 'agriculture', {FILTER => "RowFilter(=, 'regexstring:F00[0-9]$')"}
ROW                             COLUMN+CELL
 F001                           column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F001                           column=cf:Farm_Area, timestamp=1732745642435, value=329.4
 F001                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=8.14
 F001                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Sprinkler
 F001                           column=cf:Pesticide_Used, timestamp=1732745642435, value=2.21
 F001                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F001                           column=cf:Soil_Type, timestamp=1732745642435, value=Loamy
 F001                           column=cf:Water_Usage, timestamp=1732745642435, value=76648.2
 F001                           column=cf:Yield, timestamp=1732745642435, value=14.44
 F002                           column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F002                           column=cf:Farm_Area, timestamp=1732745642435, value=18.67
 F002                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=4.77
 F002                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Manual
 F002                           column=cf:Pesticide_Used, timestamp=1732745642435, value=4.36
 F002                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F002                           column=cf:Soil_Type, timestamp=1732745642435, value=Peaty
 F002                           column=cf:Water_Usage, timestamp=1732745642435, value=68725.54
 F002                           column=cf:Yield, timestamp=1732745642435, value=42.91
 F003                           column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F003                           column=cf:Farm_Area, timestamp=1732745642435, value=306.03
 F003                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=2.91
 F003                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Flood
 F003                           column=cf:Pesticide_Used, timestamp=1732745642435, value=0.56
 F003                           column=cf:Season, timestamp=1732745642435, value=Monsoon
 F003                           column=cf:Soil_Type, timestamp=1732745642435, value=Silty
 F003                           column=cf:Water_Usage, timestamp=1732745642435, value=75538.56
 F003                           column=cf:Yield, timestamp=1732745642435, value=33.44
 F004                           column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F004                           column=cf:Farm_Area, timestamp=1732745642435, value=380.21
 F004                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=3.32
 F004                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Rain-fed
 F004                           column=cf:Pesticide_Used, timestamp=1732745642435, value=4.35
 F004                           column=cf:Season, timestamp=1732745642435, value=Summer
 F004                           column=cf:Soil_Type, timestamp=1732745642435, value=Silty
 F004                           column=cf:Water_Usage, timestamp=1732745642435, value=45401.23
 F004                           column=cf:Yield, timestamp=1732745642435, value=34.08
 F005                           column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F005                           column=cf:Farm_Area, timestamp=1732745642435, value=135.56
 F005                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=8.33
 F005                           column=cf:Irrigation_Type, timestamp=1732745642435, value=Sprinkler
 F005                           column=cf:Pesticide_Used, timestamp=1732745642435, value=4.48
 F005                           column=cf:Season, timestamp=1732745642435, value=Summer
 F005                           column=cf:Soil_Type, timestamp=1732745642435, value=Clay
 F005                           column=cf:Water_Usage, timestamp=1732745642435, value=93718.69
 F005                           column=cf:Yield, timestamp=1732745642435, value=43.28
 F006                           column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F006                           column=cf:Farm_Area, timestamp=1732745642435, value=12.5
 F006                           column=cf:Fertilizer_Used, timestamp=1732745642435, value=6.42
```

Figure 2.7 Filter Data Less than 10

By using `scan 'agriculture', {COLUMNS => ['cf:Crop_Type', 'cf:Season']}` DML command, we retrieved specific column of Crop_Type and Season to focus our analysis. The output presented in Figure 2.8 showed various of crops were farmed in a particular season.

```
=> ["agriculture"]
hbase(main):002:0> scan 'agriculture', {COLUMNS => ['cf:Crop_Type', 'cf:Season']
}
ROW                   COLUMN+CELL
 F001                 column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F001                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F002                 column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F002                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F003                 column=cf:Crop_Type, timestamp=1732745642435, value=Sugarc
                      ane
 F003                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F004                 column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F004                 column=cf:Season, timestamp=1732745642435, value=Summer
 F005                 column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F005                 column=cf:Season, timestamp=1732745642435, value=Summer
 F006                 column=cf:Crop_Type, timestamp=1732745642435, value=Sugarc
                      ane
 F006                 column=cf:Season, timestamp=1732745642435, value=Summer
 F007                 column=cf:Crop_Type, timestamp=1732745642435, value=Soybea
                      n
 F007                 column=cf:Season, timestamp=1732745642435, value=Winter
 F008                 column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F008                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F009                 column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F009                 column=cf:Season, timestamp=1732745642435, value=Winter
 F010                 column=cf:Crop_Type, timestamp=1732745642435, value=Soybea
                      n
 F010                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F011                 column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F011                 column=cf:Season, timestamp=1732745642435, value=Summer
 F012                 column=cf:Crop_Type, timestamp=1732745642435, value=Sugarc
                      ane
 F012                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F013                 column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F013                 column=cf:Season, timestamp=1732745642435, value=Summer
 F014                 column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F014                 column=cf:Season, timestamp=1732745642435, value=Winter
 F015                 column=cf:Crop_Type, timestamp=1732745642435, value=Sugarc
                      ane
 F015                 column=cf:Season, timestamp=1732745642435, value=Monsoon
 F016                 column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F016                 column=cf:Season, timestamp=1732745642435, value=Summer
 F017                 column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F017                 column=cf:Season, timestamp=1732745642435, value=Summer
 F018                 column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F018                 column=cf:Season, timestamp=1732745642435, value=Winter
 F019                 column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F019                 column=cf:Season, timestamp=1732745642435, value=Summer
 F020                 column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F020                 column=cf:Season, timestamp=1732745642435, value=Monsoon
```

Figure 2.8 Filter Crop Type and Season Column

The `scan` and `SingleColumnValueFilter` DML command was used to filter the crops specifically grown in the Monsoon season. The following command was executed:

```
scan 'agriculture', {COLUMNS => ['cf:Crop_Type', 'cf:Season'], FILTER
=> "SingleColumnValueFilter('cf', 'Season', =, 'binary:Monsoon')"}
```

The output in Figure 2.9 showed that there are 16 rows corresponding to crops planted during the Monsoon season. These crops are Cotton, Carrot, Sugarcane, Rice, Soybean, Barley, Tomato and Potato. It shows diverse crops ranging from staple food crops like Rice and Potato to cash crops like Cotton and Sugarcane. The variety of crops during the Monsoon showed that there is a need for crop rotation to maintain soil health and ensure food security.

```
hbase(main):003:0> scan 'agriculture', {COLUMNS => ['cf:Crop_Type', 'cf:Season'], FILTER => "SingleColumnValueFilter(
'cf', 'Season', =, 'binary:Monsoon')"}
ROW                           COLUMN+CELL
 F001                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F001                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F002                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F002                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F003                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F003                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F008                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F008                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F010                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
 F010                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F012                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F012                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F015                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F015                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F020                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F020                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F027                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F027                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F038                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F038                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F041                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F041                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F045                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F045                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F046                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F046                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F047                         column=cf:Crop_Type, timestamp=1732745642435, value=Potato
 F047                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F049                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F049                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F050                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F050                         column=cf:Season, timestamp=1732745642435, value=Monsoon
16 row(s) in 0.2410 seconds
```

Figure 2.9 Crops in Monsoon Season

We used the `get` DML command to retrieve a specific record in detail. For example, we used the command as shown in Figure 2.10 to retrieved detail data for Farm_ID = F045. It showed that Tomato was cultivated during Monsoon season. The irrigation method used for the farm was Sprinkler, and the resulting yield of tomatoes was 18.34 tons. The policymakers can develop crop-specific strategies that target the most productive farming methods by analyzing specific crop yields in relation to season, irrigation, and soil type. This will support better planning and reduced resources for optimal crop cultivation.

```
hbase(main):002:0> get 'agriculture', 'F045'
COLUMN                        CELL
 cf:Crop_Type                 timestamp=1732745642435, value=Tomato
 cf:Farm_Area                 timestamp=1732745642435, value=326.69
 cf:Fertilizer_Used           timestamp=1732745642435, value=5.24
 cf:Irrigation_Type           timestamp=1732745642435, value=Sprinkler
 cf:Pesticide_Used            timestamp=1732745642435, value=0.55
 cf:Season                    timestamp=1732745642435, value=Monsoon
 cf:Soil_Type                 timestamp=1732745642435, value=Peaty
 cf:Water_Usage               timestamp=1732745642435, value=37466.11
 cf:Yield                     timestamp=1732745642435, value=18.34
9 row(s) in 0.2240 seconds
```

Figure 2.10 Get Command Output

Besides that, we used the `scan` with `ValueFilter` DML command to check the numbers of farms planted Rice. From the output in Figure 2.11, it showed that that Barley and Cotton are the highest farm in India with value of 7. This suggest that Barley and Cotton may be suitable to the local climate or there is high market demand for it. On the other hand, Maize was the least farm in India with value of 3. This could suggest that less market demand make Maize to be less popular choice for farmers. The policymakers could focus on promoting the cultivation of Maize through subsidies and training to improve the production.

```
hbase(main):003:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Rice')"}
ROW                           COLUMN+CELL
 F008                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F011                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F014                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F024                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
 F041                         column=cf:Crop_Type, timestamp=1732745642435, value=Rice
5 row(s) in 0.1290 seconds


hbase(main):005:0>
hbase(main):006:0* scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Barley')"}
ROW                           COLUMN+CELL
 F016                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F020                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F025                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F031                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F033                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F038                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
 F049                         column=cf:Crop_Type, timestamp=1732745642435, value=Barley
7 row(s) in 0.0840 seconds

hbase(main):007:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Carrot')"}
ROW                           COLUMN+CELL
 F002                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F017                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F032                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
 F046                         column=cf:Crop_Type, timestamp=1732745642435, value=Carrot
4 row(s) in 0.0370 seconds

hbase(main):008:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Cotton')"}
ROW                           COLUMN+CELL
 F001                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F021                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F027                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F036                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F039                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F040                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
 F043                         column=cf:Crop_Type, timestamp=1732745642435, value=Cotton
7 row(s) in 0.0520 seconds

hbase(main):009:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Maize')"}
ROW                           COLUMN+CELL
 F009                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F018                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F019                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
3 row(s) in 0.0430 seconds

hbase(main):010:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Potato')"}
ROW                           COLUMN+CELL
 F023                         column=cf:Crop_Type, timestamp=1732745642435, value=Potato
 F030                         column=cf:Crop_Type, timestamp=1732745642435, value=Potato
 F047                         column=cf:Crop_Type, timestamp=1732745642435, value=Potato
 F048                         column=cf:Crop_Type, timestamp=1732745642435, value=Potato
4 row(s) in 0.0510 seconds


hbase(main):011:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Soybean')"}
ROW                           COLUMN+CELL
 F007                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
 F010                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
 F035                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
 F037                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
 F044                         column=cf:Crop_Type, timestamp=1732745642435, value=Soybean
5 row(s) in 0.0710 seconds

hbase(main):012:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Sugarcane')"}
ROW                           COLUMN+CELL
 F003                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F006                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F012                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F015                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
 F042                         column=cf:Crop_Type, timestamp=1732745642435, value=Sugarcane
5 row(s) in 0.1370 seconds

hbase(main):013:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Tomato')"}
ROW                           COLUMN+CELL
 F004                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F005                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F028                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F034                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F045                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
 F050                         column=cf:Crop_Type, timestamp=1732745642435, value=Tomato
6 row(s) in 0.0560 seconds

hbase(main):014:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Wheat')"}
ROW                           COLUMN+CELL
 F013                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F022                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F026                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F029                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
4 row(s) in 0.1310 seconds

hbase(main):015:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Wheat')"}
ROW                           COLUMN+CELL
 F013                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F022                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F026                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
 F029                         column=cf:Crop_Type, timestamp=1732745642435, value=Wheat
4 row(s) in 0.1100 seconds
```

Figure 2.11 Number of Crops Farm

To see all the Row key in detail, we use the `scan` and `SingleColumnValueFilter` DML command as shown in Figure 2.12. For Maize farm, three types of soils are Peaty, Loamy and Sandy. Drip irrigation and Rain fed irrigation are used but the most common was drip irrigation. Maize was primarily cultivated during the Winter season, with occasional cultivation in Summer. The data reveals the amount of fertilizer and pesticide used for F009 Farm ID were 0.57 and 4.93 respectively. The excessive use of fertilizers and pesticides will cause various environmental effects such as water pollution and soil degradation. Hence, government should implement policies such as introducing pesticide tax, educating consumer to balance the use of fertilizer and promote the alternative pest management to the reduce risk from pesticide residues (Sapbamrer et al., 2023).

```
hbase(main):018:0> scan 'agriculture', {FILTER => "SingleColumnValueFilter('cf', 'Crop_Type', =, 'binary:Maize')"}
ROW                           COLUMN+CELL
 F009                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F009                         column=cf:Farm_Area, timestamp=1732745642435, value=389.37
 F009                         column=cf:Fertilizer_Used, timestamp=1732745642435, value=0.57
 F009                         column=cf:Irrigation_Type, timestamp=1732745642435, value=Drip
 F009                         column=cf:Pesticide_Used, timestamp=173274[redacted]2435, value=4.93
 F009                         column=cf:Season, timestamp=1732745642435, value=Winter
 F009                         column=cf:Soil_Type, timestamp=1732745642435, value=Peaty
 F009                         column=cf:Water_Usage, timestamp=1732745642435, value=60202.14
 F009                         column=cf:Yield, timestamp=1732745642435, value=3.86
 F018                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F018                         column=cf:Farm_Area, timestamp=1732745642435, value=128.23
 F018                         column=cf:Fertilizer_Used, timestamp=1732745642435, value=4.91
 F018                         column=cf:Irrigation_Type, timestamp=1732745642435, value=Rain-fed
 F018                         column=cf:Pesticide_Used, timestamp=1732745642435, value=0.77
 F018                         column=cf:Season, timestamp=1732745642435, value=Winter
 F018                         column=cf:Soil_Type, timestamp=1732745642435, value=Loamy
 F018                         column=cf:Water_Usage, timestamp=1732745642435, value=18660.03
 F018                         column=cf:Yield, timestamp=1732745642435, value=16.67
 F019                         column=cf:Crop_Type, timestamp=1732745642435, value=Maize
 F019                         column=cf:Farm_Area, timestamp=1732745642435, value=460.93
 F019                         column=cf:Fertilizer_Used, timestamp=1732745642435, value=1.09
 F019                         column=cf:Irrigation_Type, timestamp=1732745642435, value=Drip
 F019                         column=cf:Pesticide_Used, timestamp=1732745642435, value=1.31
 F019                         column=cf:Season, timestamp=1732745642435, value=Summer
 F019                         column=cf:Soil_Type, timestamp=1732745642435, value=Sandy
 F019                         column=cf:Water_Usage, timestamp=1732745642435, value=54314.28
 F019                         column=cf:Yield, timestamp=1732745642435, value=39.96
3 row(s) in 0.1110 seconds
```

Figure 2.12 Detail of Maize Farms

We used the `scan` DML command with a `ValueFilter` to analyze the distribution of crops across different seasons. The output in Figure 2.13 revealed that most crops are cultivated during the Summer season, followed by the Monsoon and Winter seasons, with respective values of 23, 16, and 11. This highlight that Summer season is likely a key agricultural period for farrmers. The seasonal distribution of crops helps in developing policies to encourage crop diversification to reduce effects of climate change (Blackmore et al., 2021). For example, promoting crops that can tolerate seasonal shift will increase food security. In Summer season, water availability is important for agricultural activities. Hence, incentives from government in water-efficient irrigation technologies like drip irrigation can help the farmers.

```
hbase(main):020:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Winter')"}
ROW                           COLUMN+CELL
 F007                         column=cf:Season, timestamp=1732745642435, value=Winter
 F009                         column=cf:Season, timestamp=1732745642435, value=Winter
 F014                         column=cf:Season, timestamp=1732745642435, value=Winter
 F018                         column=cf:Season, timestamp=1732745642435, value=Winter
 F021                         column=cf:Season, timestamp=1732745642435, value=Winter
 F022                         column=cf:Season, timestamp=1732745642435, value=Winter
 F026                         column=cf:Season, timestamp=1732745642435, value=Winter
 F028                         column=cf:Season, timestamp=1732745642435, value=Winter
 F032                         column=cf:Season, timestamp=1732745642435, value=Winter
 F034                         column=cf:Season, timestamp=1732745642435, value=Winter
 F044                         column=cf:Season, timestamp=1732745642435, value=Winter
11 row(s) in 0.0810 seconds

hbase(main):021:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Summer')"}
ROW                           COLUMN+CELL
 F004                         column=cf:Season, timestamp=1732745642435, value=Summer
 F005                         column=cf:Season, timestamp=1732745642435, value=Summer
 F006                         column=cf:Season, timestamp=1732745642435, value=Summer
 F011                         column=cf:Season, timestamp=1732745642435, value=Summer
 F013                         column=cf:Season, timestamp=1732745642435, value=Summer
 F016                         column=cf:Season, timestamp=1732745642435, value=Summer
 F017                         column=cf:Season, timestamp=1732745642435, value=Summer
 F019                         column=cf:Season, timestamp=1732745642435, value=Summer
 F023                         column=cf:Season, timestamp=1732745642435, value=Summer
 F024                         column=cf:Season, timestamp=1732745642435, value=Summer
 F025                         column=cf:Season, timestamp=1732745642435, value=Summer
 F029                         column=cf:Season, timestamp=1732745642435, value=Summer
 F030                         column=cf:Season, timestamp=1732745642435, value=Summer
 F031                         column=cf:Season, timestamp=1732745642435, value=Summer
 F033                         column=cf:Season, timestamp=1732745642435, value=Summer
 F035                         column=cf:Season, timestamp=1732745642435, value=Summer
 F036                         column=cf:Season, timestamp=1732745642435, value=Summer
 F037                         column=cf:Season, timestamp=1732745642435, value=Summer
 F039                         column=cf:Season, timestamp=1732745642435, value=Summer
 F040                         column=cf:Season, timestamp=1732745642435, value=Summer
 F042                         column=cf:Season, timestamp=1732745642435, value=Summer
 F043                         column=cf:Season, timestamp=1732745642435, value=Summer
 F048                         column=cf:Season, timestamp=1732745642435, value=Summer
23 row(s) in 0.1760 seconds

hbase(main):022:0> scan 'agriculture', {FILTER => "ValueFilter(=, 'binary:Monsoon')"}
ROW                           COLUMN+CELL
 F001                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F002                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F003                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F008                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F010                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F012                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F015                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F020                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F027                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F038                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F041                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F045                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F046                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F047                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F049                         column=cf:Season, timestamp=1732745642435, value=Monsoon
 F050                         column=cf:Season, timestamp=1732745642435, value=Monsoon
16 row(s) in 0.0890 seconds
```

Figure 2.13 Distribution of Crops across Different Seasons

The `disable 'agriculture'` DDL command was used to disable the 'agriculture' table to prevent any modification in the table. Then, the `was_enabled 'agriculture'` DDL command was executed to verify if the table was disabled. The false output means that the 'agriculture' table was successfully disabled and no longer active for read or write operations.

```
hbase(main):024:0> disable 'agriculture'
0 row(s) in 2.4000 seconds

hbase(main):025:0> is_enabled 'agriculture'
false
0 row(s) in 0.0570 seconds
```

3 Reference

Bates, J. (2020). Kitchen gardens, wild forage and tree fruits: A hypothesis on the role of the Zaid season in the indus civilisation (c.3200-1300 BCE). *Archaeological Research in Asia*, *21*, 100175. https://doi.org/https://doi.org/10.1016/j.ara.2019.100175

Blackmore, I., Rivera, C., Waters, W. F., Iannotti, L., & Lesorogol, C. (2021). The impact of seasonality and climate variability on livelihood security in the Ecuadorian Andes. *Climate Risk Management*, *32*, 100279. https://doi.org/https://doi.org/10.1016/j.crm.2021.100279

Sapbamrer, R., Kitro, A., Panumasvivat, J., & Assavanopakun, P. (2023). Important role of the government in reducing pesticide use and risk sustainably in Thailand: Current situation and recommendations. *Frontiers in Public Health*, *11*. https://doi.org/https://doi.org/10.3389/fpubh.2023.1141142