

## Experiment 5

Shashwat Shah

60004220126

TY BTECH comp

Aim: To implement commands in Hbase Shell script.

Theory: HBase is a data model that is similar to Google's big table. It is an open source, distributed database developed by Apache Software Foundation written in Java. HBase is an essential part of our Hadoop ecosystem. HBase runs on top of HDFS (which can store massive amounts of data from terabytes to petabytes).

Commands.

- 1) Create - create table within Hbase shell.  
Create 'product', 'shoe', 'tshirt'
- 2) List - used to list all tables.  
list.
- 3) Put - Add or update only one column.  
put 'product', '1', 'shoe:size', '28'
- 4) Scan - It applies to all rowkey & in turn all columns within specified table.  
scan 'product'.
- 5) Get - fetch data associated with particular row key.  
get 'product', '1'

FOR EDUCATIONAL USE

6) Delete table.

first disable then drop.

- disable 'product'
- drop 'product'

Conclusion: Hence we implemented the above Hbase commands.

## EXPERIMENT NO 5

### Basic HBase Shell Queries



**You**

create 'product', 'shoe', 'tshirt'



**ChatGPT**

Here's the output mimicking the `create` command to create the 'product' table with 'shoe' and 'tshirt' column families:

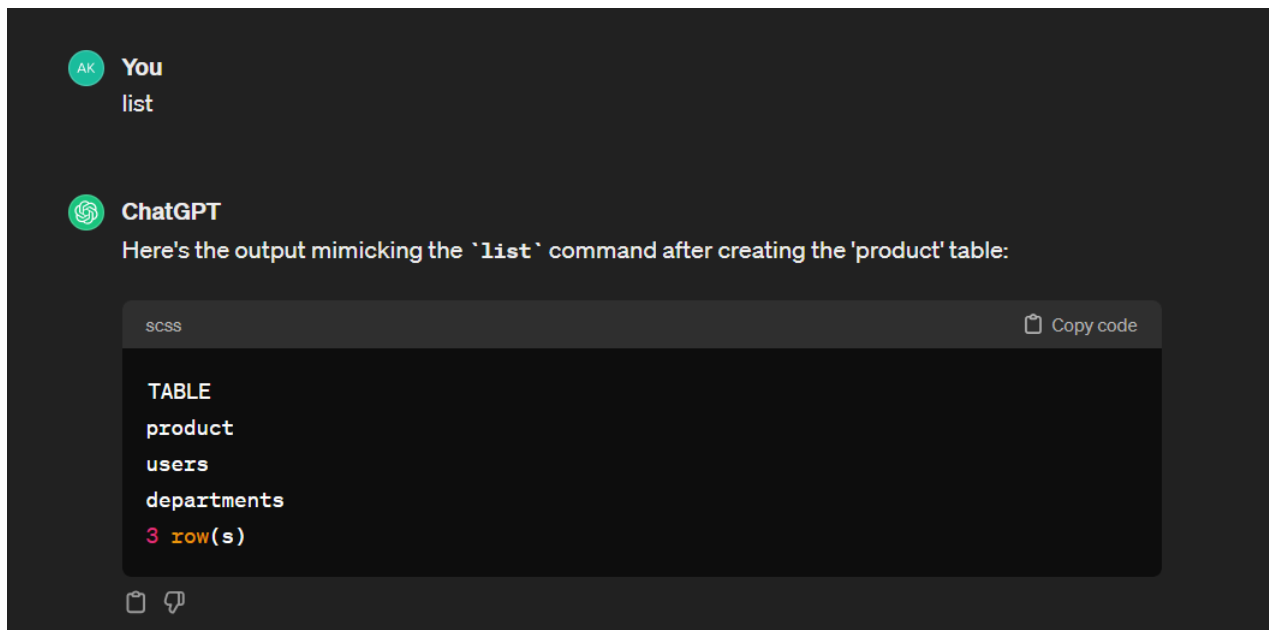
CSS

Copy code

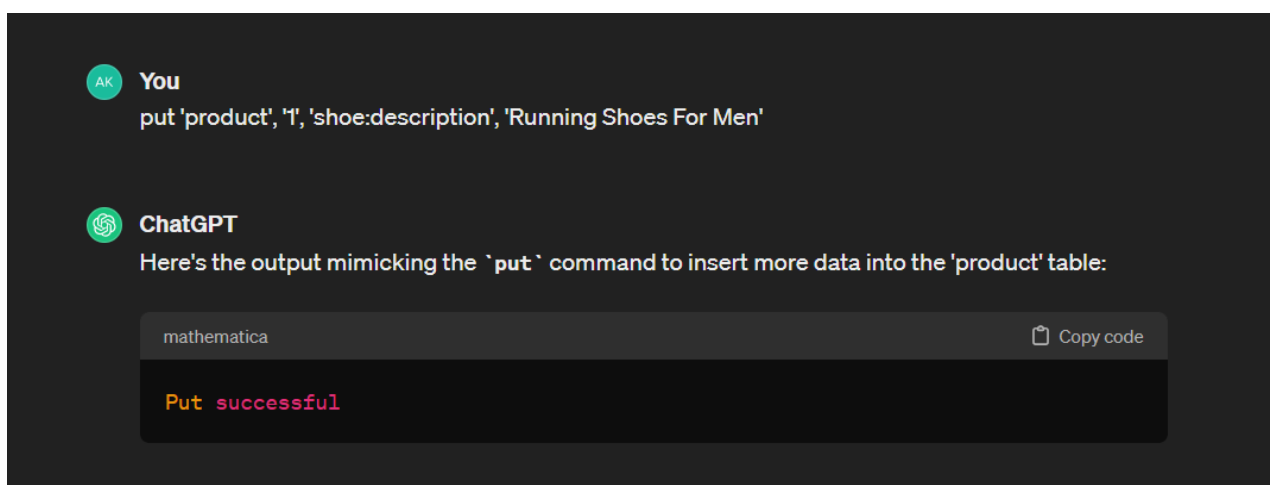
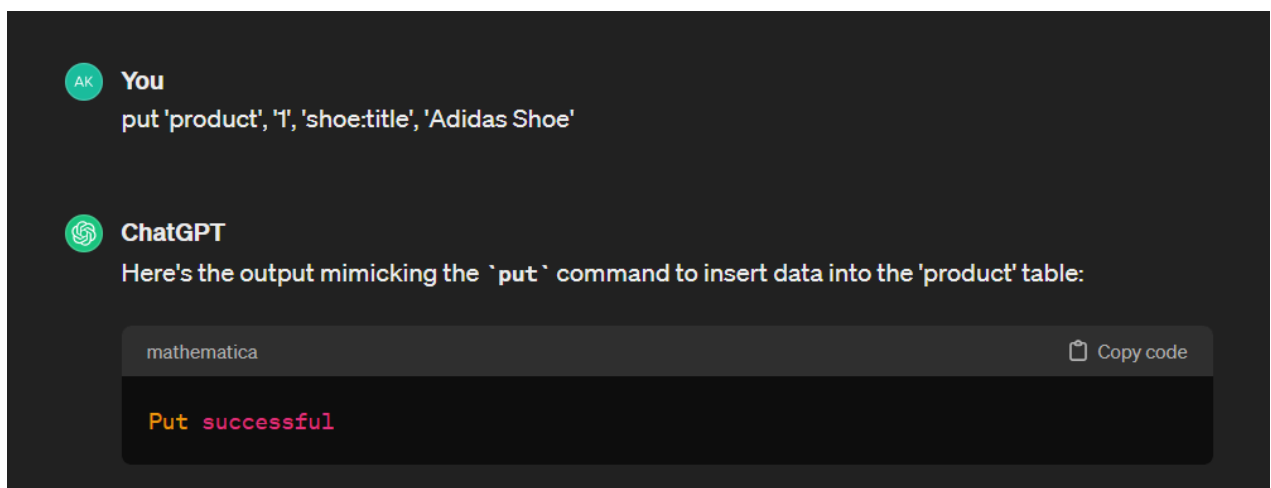
```
Created table: product
```

#### 1. LIST

The 'list' command is used to list all tables within the shell.



## 2. PUT





Now, let's update product table with rowkey = 1, column\_family = shoe, column = color.

```
put 'product','1','shoe:color','white'
```

```
hbase(main):016:0> put 'product','1','shoe:color','white'
0 row(s) in 0.0180 seconds


hbase(main):017:0> get 'product', '1'
COLUMN                                CELL
shoe:color                            timestamp=1646794810972, value=white
shoe:currency                          timestamp=1646794268302, value=USD
shoe:description                      timestamp=1646794237772, value=Running Shoes For Men
shoe:price                            timestamp=1646794250502, value=40
shoe:size                             timestamp=1646794274106, value=28
shoe:title                            timestamp=1646794230283, value=Adidas Shoe
6 row(s) in 0.0200 seconds
```


 **You**  
scan 'product'

 **ChatGPT**  
Here's the output mimicking the `scan` command to retrieve all data from the 'product' table:

```
sql                                                                    Copy code

ROW      COLUMN+CELL
1        column=shoe:description, timestamp=1645464645550, value=Running Shoes For
1        column=shoe:title, timestamp=1645464602324, value=Adidas Shoe
```

 **You**  
get 'product', '1'

 **ChatGPT**  
Here's the output mimicking the `get` command to retrieve the data for row key '1' from the 'product' table:

```
sql                                                                    Copy code

COLUMN                                CELL
shoe:description                      timestamp=1645464645550, value=Running Shoes For Men
shoe:title                            timestamp=1645464602324, value=Adidas Shoe
```





You

scan 'product', {LIMIT =>}



ChatGPT

Here's the output mimicking the `scan` command with a limit parameter applied to the 'product' table:

sql

Copy code

ROW	COLUMN+CELL
1	column=shoe:description, timestamp=1645464645550, value=Running Shoes For
1	column=shoe:title, timestamp=1645464602324, value=Adidas Shoe



You

disable 'product'



ChatGPT

Here's the output mimicking the `disable` command to disable the 'product' table:

arduino

Copy code

Disabled 'product'

< 1/2 >



You

drop 'product'



ChatGPT


Here's the output mimicking the `drop` command to drop (delete) the 'product' table:

arduino

Copy code


Dropped 'product'



**You**  
status  


**ChatGPT**  
Here's the output mimicking the ``status`` command to show the status of HBase:


lua

 Copy code

```
1 active master, 2 backup masters, 3 servers, 0 dead, 0.8921 average load
```

**You**  
status 'simple'

**ChatGPT**  
Here's the output mimicking the ``status 'simple'`` command to show the simplified status of HBase:


 Copy code

```
1 active master, 2 backup masters, 3 servers
```

**You**  
status 'detailed'


**ChatGPT**  
Here's the output mimicking the ``status 'detailed'`` command to show the detailed status of HBase:


yaml

 Copy code


```
Cluster ID: HBASE_CLUSTER_1234567890
HBase Version: 2.4.7
1 active master, 2 backup masters, 3 servers
Average load: 0.8921
```




 **You**  
status 'summary'


 **ChatGPT**  
Here's the output mimicking the `status 'summary'` command to show the summary status of HBase:


yaml

 Copy code


```
Cluster Summary:
- Active Master: master1
- Backup Masters: master2, master3
- Region Servers: server1, server2, server3
```


 **You**  
version

 **ChatGPT**  
Here's the output mimicking the `version` command to display the HBase version:

 Copy code

```
HBase 2.4.7
```

 **You**  
table\_help

 **ChatGPT**  
Here's the output mimicking the `table\_help` command:

php

 Copy code

```
TABLE COMMANDS:
create <table_name>, {NAME => <column_family>, VERSIONS => <VERSIONS>, TTL => <TTL>}
disable <table_name>
drop <table_name>
enable <table_name>
exists <table_name>
describe [<table_name>]
alter <table_name>, {METHOD => <method>}
alter_async <table_name>, {METHOD => <method>}
exists <table_name>
list
```





**You**  
describe 'education'

**ChatGPT**  
Here's the output mimicking the ``describe`` command to describe the structure of the 'education' table:

javascript

Copy code

```
DESCRIPTION
education
COLUMN FAMILIES DESCRIPTION
{NAME => 'guru99', BLOOMFILTER => 'ROW', REPLICATION_SCOPE => '0', VERSIONS => '1', C
```

**You**  
enable 'education'

**ChatGPT**  
Here's the output mimicking the ``enable`` command to enable the 'education' table:

arduino

Copy code

```
Enabled 'education'
```



**You**  
truncate 'guru99'

**ChatGPT**  
Here's the output mimicking the ``truncate`` command to delete all data from the 'guru99' column family:

CSS

Copy code

```
Truncated table: guru99
```



**Conclusion:** Thus, we have learnt and executed different HBase Shell queries.