

Week 6 Assignment

Guruprasad Velikadu Krishnamoorthy

College of Science and Technology, Bellevue University

DSC650-T301: Big Data

Professor. Nasheb Ismaily

December 22, 2023

1. Table Creation and Management

Creating students table with 'details' column family

```
create 'students', 'details'
```

list

```
hadoop-3.0.4> ps
1203 Run-J?
1204 Run-J?
2100 hbase@h
h99 hbase@hadoopNameNode
693 NameNode
1179 JobHistoryServer
2145 Hadoop
2145 Jps
2146 HistoryServer
403 ResourceManager
hadoop-3.0.4 hbase shell
2021-12-21 19:06:33,616 WARN [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
hbase shell
hbase>
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For reference, please visit: http://hbase.apache.org/2.0/quickstart.html
Version 2.3.6, r714579f2c20ca6b75146c29ab272ef0e43ac9, Wed Jul 28 22:24:42 UTC 2021
hbase(main):001:0> list
TABLES
1 row(s)
Took 0.7086 seconds
=> []
hbase(main):002:0> create 'students', 'details'
Created table students
Took 0.7219 seconds
hbase(main):003:0> list
hbase(main):003:0> list
TABLES
1 row(s)
Took 0.1187 seconds
=> ["students"]
hbase(main):004:0> []
```

2. Data Manipulation in HBase:

```
# Inserting records into the students table and running a scan on the table:
```

```
put 'students', '1', 'details:firstName', 'John'
put 'students', '1', 'details:lastName', 'Doe'
get 'students', '1'
scan 'students'
```

```
hbase(main):004>
hbase(main):006>
hbase(main):006> put 'students','1','details:firstName','John'
Took 0.2863 seconds
hbase(main):007> put 'students','1','details:lastName','Doe'
Took 0.0313 seconds
hbase(main):008> get 'students','1'
COLUMN=CELL
detail:firstName timestamp=2023-12-21T19:10:39.777 value=John
detail:lastName timestamp=2023-12-21T19:10:46.497 value=Doe
1 row(s)
Took 0.0841 seconds
hbase(main):009> scan 'students'
ROW
1
1 ROW(s)
Took 0.0047 seconds
hbase(main):010>
```

3. Advanced HBase Features: Composite Row Key

```
# Creating a table named 'orders' and inserting records into the table.
```

```
create 'orders', 'orderDetails'
put 'orders', '101:20230806', 'orderDetails:item', 'Laptop'
put 'orders', '102:20230806', 'orderDetails:item', 'Smartphone'
scan 'orders', {STARTROW => '101:', ENDROW => '101:~'}
```

```
hbase(main):010:0>
hbase(main):011:0>
hbase(main):012:0>
hbase(main):013:0>
hbase(main):014:0> create 'orders', 'orderDetails'
Created table orders
Took 1.1772 seconds
> Hbase::Table.new('orders').put('orders', '101:20230806', 'orderDetails:item', 'Laptop')
hbase(main):015:0>
Took 0.0385 seconds
hbase(main):016:0>
hbase(main):017:0> put 'orders', '102:20230806', 'orderDetails:item', 'Smartphone'
Took 0.0565 seconds
hbase(main):018:0>
hbase(main):019:0> scan 'orders'
ROW
101:20230806 column=orderDetails:item, timestamp=2023-12-21T19:14:59.085, value=Laptop
102:20230806 column=orderDetails:item, timestamp=2023-12-21T19:15:05.633, value=Smartphone
2 ROW(s)
Took 0.0059 seconds
hbase(main):020:0>
hbase(main):021:0>
hbase(main):022:0> scan 'orders', {STARTROW => '101:', ENDROW => '101:-'}
ROW
101:20230806 column=orderDetails:item, timestamp=2023-12-21T19:14:59.085, value=Laptop
1 ROW(s)
Took 0.0029 seconds
hbase(main):023:0>
hbase(main):024:0> scan 'orders', {STARTROW => '101:', ENDROW => '102:-'}
ROW
101:20230806 column=orderDetails:item, timestamp=2023-12-21T19:14:59.085, value=Laptop
102:20230806 column=orderDetails:item, timestamp=2023-12-21T19:15:05.633, value=Smartphone
2 ROW(s)
Took 0.0020 seconds
hbase(main):025:0>
```

4. Data Generation for HBase

Using loop to insert records into the students table

(2..100).each do |i|

first_name = "Student#{i}"

last_name = "LastName#{i}"

put 'students', "#{i}", 'details:firstName', first_name

put 'students', "#{i}", 'details:lastName', last_name

end

```
hbase(main):008:0> (2..100).each do |i|
hbase(main):008:1> first_name = "Student#{i}"
hbase(main):008:1b> last_name = "LastName#{i}"
hbase(main):008:1b> put 'students', "#{i}", 'details:firstName', first_name
hbase(main):009:1b> put 'students', "#{i}", 'details:lastName', last_name
hbase(main):010:1b> end
Took 0.0000 seconds
Took 0.0093 seconds
Took 0.0112 seconds
Took 0.0089 seconds
Took 0.0106 seconds
Took 0.0072 seconds
Took 0.0171 seconds
Took 0.0173 seconds
Took 0.0157 seconds
Took 0.0071 seconds
Took 0.0064 seconds
Took 0.0074 seconds
Took 0.0074 seconds
Took 0.0113 seconds
Took 0.0127 seconds
Took 0.0085 seconds
Took 0.0126 seconds
Took 0.0017 seconds
Took 0.0056 seconds
Took 0.0055 seconds
Took 0.0050 seconds
Took 0.0050 seconds
Took 0.0052 seconds
Took 0.0050 seconds
Took 0.0044 seconds
Took 0.0045 seconds
Took 0.0035 seconds
Took 0.0034 seconds
Took 0.0032 seconds
Took 0.0033 seconds
Took 0.0033 seconds
Took 0.0039 seconds
Took 0.0038 seconds
Took 0.0038 seconds
Took 0.0032 seconds
Took 0.0041 seconds
Took 0.0036 seconds
Took 0.0045 seconds
Took 0.0034 seconds
Took 0.0034 seconds
Took 0.0040 seconds
Took 0.0034 seconds
Took 0.0048 seconds
Took 0.0049 seconds
Took 0.0060 seconds
Took 0.0041 seconds
Took 0.0045 seconds
Took 0.0048 seconds
Took 0.0042 seconds
Took 0.0055 seconds
Took 0.0036 seconds
Took 0.0042 seconds
Took 0.0046 seconds
```

scan 'students'

5. HBase Data Manipulation

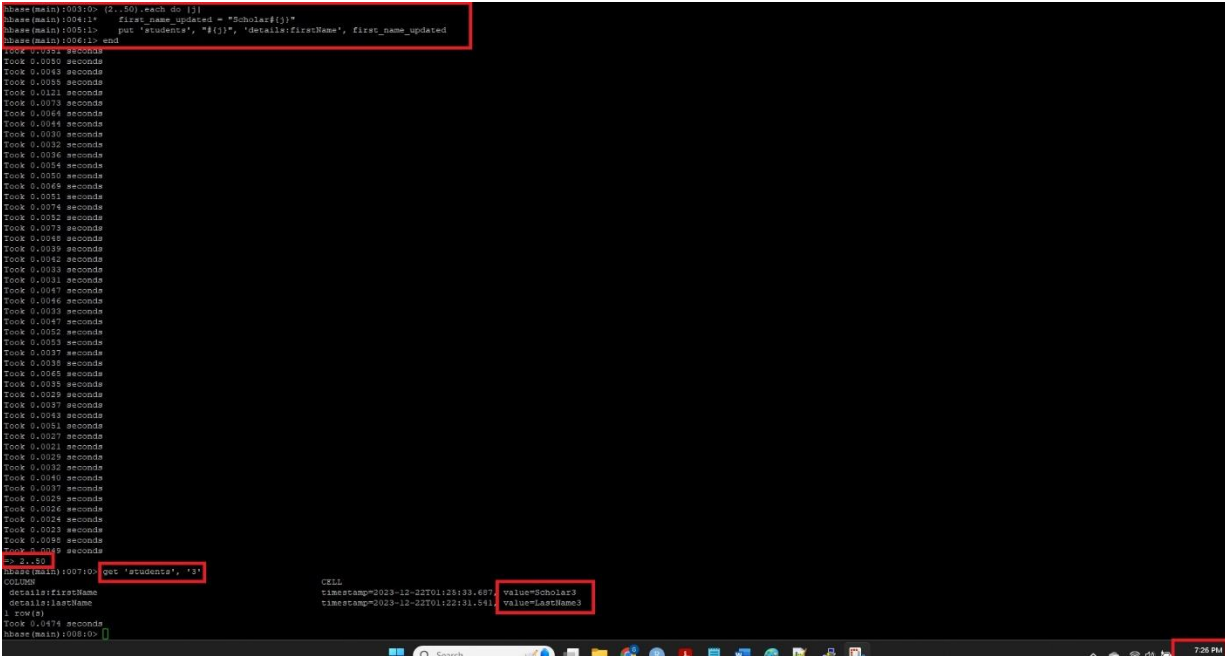
5.1. Update First Names:

For students with IDs from 2 to 50, change the first name prefix from Student to Scholar. For instance, Student3 should become Scholar3.

Solution:

Using loop to update the firstName using the put method:

```
(2..50).each do |j|  
  first_name_updated = "Scholar#{j}"  
  put 'students', "#{j}", 'details:firstName', first_name_updated  
end
```



```
hbase(main):003> (2..50).each do |j|  
hbase(main):004> first_name_updated = "Scholar#{j}"  
hbase(main):005> put 'students', "#{j}", 'details:firstName', first_name_updated  
hbase(main):006> end  
Took 0.0049 seconds  
Took 0.0030 seconds  
Took 0.0048 seconds  
Took 0.0055 seconds  
Took 0.0121 seconds  
Took 0.0073 seconds  
Took 0.0064 seconds  
Took 0.0044 seconds  
Took 0.0030 seconds  
Took 0.0032 seconds  
Took 0.0036 seconds  
Took 0.0014 seconds  
Took 0.0050 seconds  
Took 0.0069 seconds  
Took 0.0051 seconds  
Took 0.0074 seconds  
Took 0.0032 seconds  
Took 0.0073 seconds  
Took 0.0048 seconds  
Took 0.0039 seconds  
Took 0.0042 seconds  
Took 0.0033 seconds  
Took 0.0021 seconds  
Took 0.0047 seconds  
Took 0.0046 seconds  
Took 0.0033 seconds  
Took 0.0047 seconds  
Took 0.0052 seconds  
Took 0.0033 seconds  
Took 0.0037 seconds  
Took 0.0035 seconds  
Took 0.0045 seconds  
Took 0.0029 seconds  
Took 0.0037 seconds  
Took 0.0043 seconds  
Took 0.0031 seconds  
Took 0.0027 seconds  
Took 0.0021 seconds  
Took 0.0032 seconds  
Took 0.0040 seconds  
Took 0.0037 seconds  
Took 0.0029 seconds  
Took 0.0026 seconds  
Took 0.0024 seconds  
Took 0.0023 seconds  
Took 0.0036 seconds  
Took 0.0049 seconds  
=> 2..50  
hbase(main):007> get 'students', '3'  
COLUMN  
details:firstName  
details:lastName  
i rowid  
Took 0.0474 seconds  
hbase(main):008> |  
CELL  
timestamp=2023-12-22T01:28:33.687 value=Scholar3  
timestamp=2023-12-22T01:22:31.541 value=LastName3
```

5.2. Add a Middle Name:

For students with IDs from 51 to 75, add a middle name column under the details column family. The middle name should follow the pattern MidName#{i}.

Solution:

Using loop to add the Middle Name using the put method:

```
(51..75).each do |k|  
  
  middle_name = "MidName#{k}"
```

```
put 'students', "#{k}", 'details:middleName', middle_name
```

```
end
```

```
hbase(main):010:0> (51..75).each do |k|
hbase(main):011:1>   middle_name = "MidName#{k}"
hbase(main):012:1>   put 'students', "#{k}", 'details:middleName', middle_name
hbase(main):013:1> end
Took 0.0049 seconds
Took 0.0037 seconds
Took 0.0034 seconds
Took 0.0042 seconds
Took 0.0030 seconds
Took 0.0026 seconds
Took 0.0072 seconds
Took 0.0034 seconds
Took 0.0036 seconds
Took 0.0025 seconds
Took 0.0024 seconds
Took 0.0022 seconds
Took 0.0026 seconds
Took 0.0020 seconds
Took 0.0021 seconds
Took 0.0019 seconds
Took 0.0022 seconds
Took 0.0086 seconds
Took 0.0080 seconds
Took 0.0086 seconds
Took 0.0044 seconds
Took 0.0034 seconds
Took 0.0033 seconds
Took 0.0030 seconds
Took 0.0032 seconds
=> 51..75
hbase(main):014:0> get 'students', '53'
COLUMN                                CELL
details:firstName                      timestamp=2023-12-22T01:22:32.096 value=Student53
details:lastName                       timestamp=2023-12-22T01:22:32.101 value=LastName53
details:middleName                     timestamp=2023-12-22T01:27:36.464 value=MidName53
1 row(s)
Took 0.0115 seconds
hbase(main):015:0> get 'students', '74'
COLUMN                                CELL
details:firstName                      timestamp=2023-12-22T01:22:32.276 value=Student74
details:lastName                       timestamp=2023-12-22T01:22:32.282 value=LastName74
details:middleName                     timestamp=2023-12-22T01:27:36.247 value=MidName74
1 row(s)
Took 0.0144 seconds
hbase(main):016:0>
```

Another way of doing using "Append" command:

```
(51..75).each do |k|
```

```
  middle_name = "MidName#{k}"
```

```
  append 'students', "#{k}", 'details:middleName', middle_name
```

```
end
```

```

hbase(main):034:0*
hbase(main):035:0 (51..75).each do |k|
hbase(main):036:1   middle_name = "MidName#{k}"
hbase(main):037:1   append 'students', "#{k}", 'details:middleName', middle_name
hbase(main):038:1 end
CURRENT VALUE = MidName51
Took 0.0090 seconds
CURRENT VALUE = MidName52
Took 0.0046 seconds
CURRENT VALUE = MidName53
Took 0.0038 seconds
CURRENT VALUE = MidName54
Took 0.0068 seconds
CURRENT VALUE = MidName55
Took 0.0051 seconds
CURRENT VALUE = MidName56
Took 0.0047 seconds
CURRENT VALUE = MidName57
Took 0.0040 seconds
CURRENT VALUE = MidName58
Took 0.0037 seconds
CURRENT VALUE = MidName59
Took 0.0036 seconds
CURRENT VALUE = MidName60
Took 0.0038 seconds
CURRENT VALUE = MidName61
Took 0.0042 seconds
CURRENT VALUE = MidName62
Took 0.0044 seconds
CURRENT VALUE = MidName63
Took 0.0052 seconds
CURRENT VALUE = MidName64
Took 0.0073 seconds
CURRENT VALUE = MidName65
Took 0.0038 seconds
CURRENT VALUE = MidName66
Took 0.0031 seconds
CURRENT VALUE = MidName67
Took 0.0038 seconds
CURRENT VALUE = MidName68
Took 0.0042 seconds
CURRENT VALUE = MidName69
Took 0.0029 seconds
CURRENT VALUE = MidName70
Took 0.0029 seconds
CURRENT VALUE = MidName71
Took 0.0030 seconds
CURRENT VALUE = MidName72
Took 0.0036 seconds
CURRENT VALUE = MidName73
Took 0.0067 seconds
CURRENT VALUE = MidName74
Took 0.0049 seconds
CURRENT VALUE = MidName75
Took 0.0043 seconds
=> 51..75
hbase(main):039:0> get 'students','53'
COLUMN
  details:firstName
  details:lastName
  details:middleName
1 row(s)
Took 0.0272 seconds

```

```

CELL
timestamp=2024-01-18T02:49:37.935, value=Student53
timestamp=2024-01-18T02:49:37.938, value=LastName53
timestamp=2024-01-18T03:03:48.036, value=MidName53

```

5.3. Modify Last Names:

For students with IDs from 76 to 100, append `_Modified` to the last name. So, `LastName76` should be updated to `LastName76_Modified`.

Solution:

```
# Using loop to modify the Last Name using the put method:
```

```
(76..100).each do ||
```

```
  last_name_mod = "LastName#{l}_Modified"
```

```

        put 'students', "#{i}", 'details:lastName', last_name_mod
    end
end

```

```

hbase(main):020:0> (76..100).each do |i|
  hbase(main):021:1> last_name_mod = "LastName#{i}_Modified"
  hbase(main):022:1> put 'students', "#{i}", 'details:lastName', last_name_mod
hbase(main):023:1> end
Took 0.0081 seconds
Took 0.0046 seconds
Took 0.0046 seconds
Took 0.0046 seconds
Took 0.0049 seconds
Took 0.0047 seconds
Took 0.0038 seconds
Took 0.0031 seconds
Took 0.0037 seconds
Took 0.0030 seconds
Took 0.0031 seconds
Took 0.0040 seconds
Took 0.0037 seconds
Took 0.0036 seconds
Took 0.0045 seconds
Took 0.0072 seconds
Took 0.0036 seconds
Took 0.0026 seconds
Took 0.0040 seconds
Took 0.0027 seconds
Took 0.0039 seconds
Took 0.0027 seconds
Took 0.0024 seconds
Took 0.0031 seconds
=> 76..100
hbase(main):024:0> get 'students', '76'
COLUMN                                CELL
details:firstName                      timestamp=2023-12-22T01:22:32.299, value=Student76
details:lastName                       timestamp=2023-12-22T01:29:49.244, value=LastName76_Modified
1 row(s)
Took 0.0126 seconds
hbase(main):025:0> get 'students', '98'
COLUMN                                CELL
details:firstName                      timestamp=2023-12-22T01:22:32.495, value=Student98
details:lastName                       timestamp=2023-12-22T01:29:49.397, value=LastName98_Modified
1 row(s)
Took 0.0105 seconds
hbase(main):026:0>

```

5.4. Bulk Delete:

Delete all the details for students with IDs from 90 to 100.

Solution:

Using loop to delete the rows with rowid 90 through 100:

```

(90..100).each do |m|
    deleteall 'students', "#{m}"
end

```

```

hbase(main):031:0> (90..100).each do |m|
  hbase(main):032:1> deleteall 'students', "#{m}"
hbase(main):033:1> end
Took 0.0004 seconds
Took 0.0055 seconds
Took 0.0042 seconds
Took 0.0039 seconds
Took 0.0037 seconds
Took 0.0033 seconds
Took 0.0027 seconds
Took 0.0046 seconds
Took 0.0038 seconds
Took 0.0030 seconds
Took 0.0036 seconds
=> 90..100
hbase(main):034:0> get 'students', '90'
COLUMN                                CELL
0 row(s)
Took 0.0048 seconds
hbase(main):035:0> get 'students', '100'
COLUMN                                CELL
0 row(s)
Took 0.0034 seconds
hbase(main):036:0>

```

5.5. Data Retrieval:

After all modifications, retrieve and display the details for students with IDs 40, 60, 80, and 90 to verify changes.

Solution:

Using get method to print the contents of RowId 40,60,80 and 90:

```
arr=[40,60,80,90]
for n in arr do
    get 'students', "#{n}"
end
```

```
hbase(main):041:0> arr=[40,60,80,90]
hbase(main):042:0> for n in arr do
hbase(main):043:1> get 'students', "#{n}"
hbase(main):044:1> end

COLUMN=CELL
timestamp=2023-12-27T01:25:33.918, value=Scholar40
timestamp=2023-12-27T01:25:33.932, value=Scholar40

COLUMN=CELL
timestamp=2023-12-27T01:22:32.165, value=Student60
timestamp=2023-12-27T01:22:32.171, value=Scholar60
timestamp=2023-12-27T01:25:33.932, value=Scholar60

COLUMN=CELL
timestamp=2023-12-27T01:22:32.332, value=Student80
timestamp=2023-12-27T01:25:33.932, value=Scholar80_Modified

COLUMN=CELL
timestamp=2023-12-27T01:22:32.332, value=Student90
timestamp=2023-12-27T01:25:33.932, value=Scholar90_Modified

hbase(main):045:0>
hbase(main):045:0>
```

Final scan of the table 'students':

```
hbase(main):046:0> scan 'students'
hbase(main):047:0>
hbase(main):048:0> scan 'students'

COLUMN=CELL
timestamp=2023-12-27T01:22:04.057, value=John
timestamp=2023-12-27T01:22:04.185, value=Doe
timestamp=2023-12-27T01:25:33.742, value=Scholar10
timestamp=2023-12-27T01:22:31.423, value=Scholar10
timestamp=2023-12-27T01:25:33.743, value=Scholar11
timestamp=2023-12-27T01:22:31.440, value=Scholar11
timestamp=2023-12-27T01:25:33.743, value=Scholar12
timestamp=2023-12-27T01:22:31.444, value=Scholar12
timestamp=2023-12-27T01:25:33.743, value=Scholar13
timestamp=2023-12-27T01:22:31.444, value=Scholar13
timestamp=2023-12-27T01:25:33.743, value=Scholar14
timestamp=2023-12-27T01:22:31.444, value=Scholar14
timestamp=2023-12-27T01:25:33.743, value=Scholar15
timestamp=2023-12-27T01:22:31.444, value=Scholar15
timestamp=2023-12-27T01:25:33.743, value=Scholar16
timestamp=2023-12-27T01:22:31.444, value=Scholar16
timestamp=2023-12-27T01:25:33.743, value=Scholar17
timestamp=2023-12-27T01:22:31.444, value=Scholar17
timestamp=2023-12-27T01:25:33.743, value=Scholar18
timestamp=2023-12-27T01:22:31.444, value=Scholar18
timestamp=2023-12-27T01:25:33.743, value=Scholar19
timestamp=2023-12-27T01:22:31.444, value=Scholar19
timestamp=2023-12-27T01:25:33.743, value=Scholar20
timestamp=2023-12-27T01:22:31.444, value=Scholar20
timestamp=2023-12-27T01:25:33.743, value=Scholar21
timestamp=2023-12-27T01:22:31.444, value=Scholar21
timestamp=2023-12-27T01:25:33.743, value=Scholar22
timestamp=2023-12-27T01:22:31.444, value=Scholar22
timestamp=2023-12-27T01:25:33.743, value=Scholar23
timestamp=2023-12-27T01:22:31.444, value=Scholar23
timestamp=2023-12-27T01:25:33.743, value=Scholar24
timestamp=2023-12-27T01:22:31.444, value=Scholar24
timestamp=2023-12-27T01:25:33.743, value=Scholar25
timestamp=2023-12-27T01:22:31.444, value=Scholar25
timestamp=2023-12-27T01:25:33.743, value=Scholar26
timestamp=2023-12-27T01:22:31.444, value=Scholar26
timestamp=2023-12-27T01:25:33.743, value=Scholar27
timestamp=2023-12-27T01:22:31.444, value=Scholar27
timestamp=2023-12-27T01:25:33.743, value=Scholar28
timestamp=2023-12-27T01:22:31.444, value=Scholar28
timestamp=2023-12-27T01:25:33.743, value=Scholar29
timestamp=2023-12-27T01:22:31.444, value=Scholar29
timestamp=2023-12-27T01:25:33.743, value=Scholar30
timestamp=2023-12-27T01:22:31.444, value=Scholar30
timestamp=2023-12-27T01:25:33.743, value=Scholar31
timestamp=2023-12-27T01:22:31.444, value=Scholar31
timestamp=2023-12-27T01:25:33.743, value=Scholar32
timestamp=2023-12-27T01:22:31.444, value=Scholar32
timestamp=2023-12-27T01:25:33.743, value=Scholar33
timestamp=2023-12-27T01:22:31.444, value=Scholar33
timestamp=2023-12-27T01:25:33.743, value=Scholar34
timestamp=2023-12-27T01:22:31.444, value=Scholar34
timestamp=2023-12-27T01:25:33.743, value=Scholar35
timestamp=2023-12-27T01:22:31.444, value=Scholar35
```

[illegible]