# **IST769 Homework Submission**

Name: **Sathish Kumar Rajendiran**   
SUID: **666555028**  
Email: srajendi@syr.edu  
Due Date: **08/24/2021**

Task: **Impala, HBase and HCatalog**

Homework #:**7**

**Exercise(s):**

1. From Impala, use the two external tables weblogs created from clickstream/logs\_noheader and iplookup created from clickstream/iplookup\_noheader you created in the previous assignment to complete this question. Use the impala shell to answer the following questions, making sure to include the SELECT query you used to answer it.
   1. How many GET and POST requests are there in the weblogs?
   2. How many requests have Mac in the user agent?
   3. How many hosts (ip addresses) have Mac in the user agent?

**Solution:**

-- Query to find number of GET and POST requests

select method, count(\*) as requests

from weblogs

where method in (‘GET’,’POST’)

group by method;

--Query to find number of requests have MAC in User agent

select count(\*)

from weblogs

where lower(useragent) like ‘%mac%’;

--Query to find number of hosts have MAC in User agent

select count(distinct ipaddress) as ipaddresses

from weblogs

where lower(useragent) like ‘%mac%’;

select count(distinct i.ip) as ipaddresses

from iplookup i join weblogs w on i.ip = w.ipaddress

where

where lower(w.useragent) like ‘%mac%’;

**Evidence:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

1. From the HBase shell, include the commands required to complete the following.
   1. Create a table named **computers** with column family **info**.
   2. Issue HBase commands to write the following data to the table in the column family:

|  |  |  |  |
| --- | --- | --- | --- |
| Computer ID | Model | GB\_Ram | TB\_Disk |
| 1 | Dell | 16 | 1 |
| 2 | IBM | 32 | 1.5 |
| 3 | HP | 8 | 1 |
| 4 | Acer | 16 | 2 |

**Solution:**

-- create a Hbase table as computers with column family info

t = create ‘computers’, ‘info’

-- verify the table existence

scan ‘computers’

-- Add records to computers table

put ‘computers’, ‘1’, ‘info:Model’, ‘Dell’

put ‘computers’, ‘1’, ‘info:GB\_RAM’, ‘16’

put ‘computers’, ‘1’, ‘info:TB\_Disk, ‘1’

put ‘computers’, ‘2’, ‘info:Model’, ‘IBM

put ‘computers’, ‘2’, ‘info:GB\_RAM’, ‘32’

put ‘computers’, ‘2’, ‘info:TB\_Disk, ‘1.5’

put ‘computers’, ‘3’, ‘info:Model’, ‘HP

put ‘computers’, ‘3’, ‘info:GB\_RAM’, ‘8’

put ‘computers’, ‘3’, ‘info:TB\_Disk, ‘1’

put ‘computers’, ‘4’, ‘info:Model’, ‘Acer

put ‘computers’, ‘4’, ‘info:GB\_RAM’, ‘16’

put ‘computers’, ‘4’, ‘info:TB\_Disk, ‘2’

-- verify the data existence

scan ‘computers’

**Evidence:**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. From the Hive shell, write an HQL statement to create an external Hive table from the HBase **computers** table. Then write a hive query to add up the total ram and disk across all computers. Your answer should include all HQL statements.

**Solution:**

-- create an external table in Hive from Hbase (computers)

create external table computers

(id int, Model string, GB\_RAM int, TB\_Disk float)

stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH

SERDEPROPERTIES (‘hbase.columns.mapping’=’:key,info:Model,info:GB\_RAM,info:TB\_Disk’)

TBLPROPERTIES (‘hbase.table.name’=’computers’);

-- verify the data existence

describe computers;

select \* from computers;

-- HQL statement to add total ram and disk across computers

select

sum(GB\_RAM) as Total\_Memory

, sum(TB\_Disk) as Total\_Storage

from computers;

**Evidence:**

**Text

Description automatically generated**

Text

Description automatically generated

Text

Description automatically generated

1. Use Hive to load the **iplookup** table you created from **clickstream/iplookup\_noheader** into and HBase table, with IP address as key. Include the HQL Queries you wrote to make the table and load the data as the answer to your question.

**Solution:**

-- create a table in Hbase from Hive (iplookup)

create table iplookup\_hbase

(ip string, country string, state string, city string, lat float, lan float)

stored by 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'

WITH

SERDEPROPERTIES (‘hbase.columns.mapping’=’:key, ip:country, ip:state, ip:city, ip:lat, ip:lan’)

TBLPROPERTIES (‘hbase.table.name’=’iplookup\_hbase’, ‘hbase.mapred.output.outputtable’ = ‘iplookup\_hbase’);

-- HQL to load the data

INSERT OVERWRITE TABLE iplookup\_hbase SELECT \* FROM iplookup;

-- Hbase shell to query the data

scan ‘iplookup\_hbase’

**Evidence:**

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

1. From the HBase shell, write an HBase query to retrieve the city and state columns for all rows in the iplookup table.

**Solution:**

-- Hbase shell to retrieve city and state info from iplookup table

scan ‘iplookup\_hbase’, {COLUMNS => [‘ip:city’, ‘ip:state’]}

**Evidence:**

A screen shot of a computer screen

Description automatically generated with low confidence

Graphical user interface, text

Description automatically generated

# **Appendix**

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

**Text

Description automatically generated**

Graphical user interface, text

Description automatically generated

**Text

Description automatically generated**

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

**Graphical user interface

Description automatically generated**

**Text

Description automatically generated**

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A screen shot of a computer screen

Description automatically generated with low confidence

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Word

Description automatically generated with medium confidence

Text

Description automatically generated

Graphical user interface, application

Description automatically generated