UNIVERSITY OF MISSOURI-KANSAS CITY

Bigdata Hadoop Programming

 # Lab 2 Assignment

Team Members:

Pragathi Thammaneni

Sridevi Mallipudi

Introduction:

The main core concept for executing the Lab 2 Assignment is to implement the Hive/solr table including the complex data types. And also implemented 10 intuitive queries using various datasets (pattern recognition, topic discussion, most important terms, etc)

Objectives:

To code for the 2 questions the below concepts are implemented.

Hive Table

Solr collections

10 intuitive queries using various datasets (pattern recognition, topic discussion, most important terms.

Time execution for the queries

Approaches /Methods:

Cloudera, Hive and Solr

Workflow &Datasets/Parameters and Evaluation:

The below each question will follow different approaches to solve. Coding is done to perform the evaluation of each individual snippet to execute the datasets which are provided as the input parameters.

Question 1:

1. ***HIVE USECASE***

   a) **Create a Hive Table including Complex Data Types**
   b) **Use built-in functions in your queries**
   c) **Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box.**

Queries implemented:

Below are the queries that has been implemented on the data sets Zomato and episodes.

1. The table creation has been done with the complex data types like map.
2. And the data has been loaded in to the hive for performing the respective queries
3. Built in functions – Table creation and select statement as a basic query, Collect, Count ,explode ,map values, sum , 3 nested queries ,pattern recognition.

Output Screenshot:

1. Table Creation & Loading data in to hive using complex data types:

```
hive> create table episodes_info(Id int,Season int,Episode_title map<int,string>,Airdate date,Writers string,Director string,SEID varchar(255)) row format delimited fields terminated by '\t' collection items terminated by ','  map keys t
erminated by ':' stored as textfile;
OK
Time taken: 31.895 seconds
hive> describe episodes_info;
OK
id                  int
season              int
episode_title       map<int,string>
airdate             date
writers             string
director            string
seid                varchar(255)
Time taken: 1.419 seconds, Fetched: 7 row(s)
hive> load data local inpath '/home/cloudera/Downloads/sample_updated.txt' into table episodes_info;
Loading data to table default.episodes_info
Table default.episodes_info stats: [numFiles=1, totalSize=1809]
OK
Time taken: 2.928 seconds
```

```
hive> load data local inpath '/home/cloudera/Downloads/sample_updated.txt' into table episodes_info;
Loading data to table default.episodes_info
Table default.episodes_info stats: [numFiles=1, totalSize=1809]
OK
Time taken: 2.928 seconds
hive> select *from episodes_info;
OK
1       1       {1:"The Stakeout"}      1990-05-31      "Larry David, Jerry Seinfeld"   Tom Cherones    S01E01
2       1       {2:"The Robbery"}       1990-06-07      Matt Goldman    Tom Cherones    S01E02
3       1       {3:"Male Unbonding"}    1990-06-14      "Larry David, Jerry Seinfeld"   Tom Cherones    S01E03
4       1       {4:"The Stock Tip"}     1990-06-21      "Larry David, Jerry Seinfeld"   Tom Cherones    S01E04
5       2       {1:"The Ex-Girlfriend"} 1991-01-16      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E01
6       2       {2:"The Pony Remark"}   1991-01-30      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E02
7       2       {3:"The Jacket"}        1991-02-06      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E03
8       2       {4:"The Phone Message"} 1991-02-13      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E04
9       2       {5:"The Apartment"}     1991-04-04      Peter Mehlman   Tom Cherones    S02E05
10      2       {6:"The Statue"}        1991-04-11      Larry David     Tom Cherones    S02E06
11      2       {7:"The Revenge"}       1991-04-18      Larry David     Tom Cherones    S02E07
12      2       {8:"The Heart Attack"}  1991-04-25      Larry Charles   Tom Cherones    S02E08
13      2       {9:"The Deal"}  1991-05-02      Larry Charles   Tom Cherones    S02E09
14      2       {10:"The Baby Shower"}  1991-05-16      Larry Charles   Tom Cherones    S02E10
15      2       {11:"The Chinese Restaurant"}   1991-05-23      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E11
16      2       {12:"The Busboy"}       1991-06-26      "Larry David, Jerry Seinfeld"   Tom Cherones    S02E12
17      3       {1:"The Note"}  1991-09-18      Larry David     Tom Cherones    S03E01
18      3       {2:"The Truth"} 1991-09-25      Elaine Pope     David Steinberg S03E02
19      3       {3:"The Pen"}   1991-10-02      Larry David     Tom Cherones    S03E03
20      3       {4:"The Dog"}   1991-10-09      Larry David     Tom Cherones    S03E04
21      3       {5:"The Library"}       1991-10-16      Larry Charles   Joshua White    S03E05
22      3       {6:"The Parking Garage"}        1991-10-30      Larry David     Tom Cherones    S03E06
23      3       {7:"The Cafe"}  1991-11-06      Tom Leopold     Tom Cherones    S03E07
24      3       {8:"The Tape"}  1991-11-13      "Larry David and Bob Shaw, Don McEnery" David  Steinberg   S03E08
25      3       {9:"The Nose Job"}      1991-11-20      Peter Mehlman   Tom Cherones    S03E09
Time taken: 1.492 seconds, Fetched: 25 row(s)
hive>
```

2. Built in function -Collect

```
hive> select collect_set(director) from episodes_info;
Query ID = cloudera_20180630133737_ecbf0a63-fd44-42de-bfac-ff2174f1bf23
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 13:37:54,967 Stage-1 map = 0%,  reduce = 0%
2018-06-30 13:38:08,815 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.66 sec
2018-06-30 13:38:22,275 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.08 sec
MapReduce Total cumulative CPU time: 4 seconds 80 msec
Ended Job = job_1530386334216_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.08 sec   HDFS Read: 11068 HDFS Write: 64 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 80 msec
OK
["Tom Cherones","David Steinberg","Joshua White","David  Steinberg"]
Time taken: 44.856 seconds, Fetched: 1 row(s)
```

3. Built in function – Count

```
hive> select season,count(*) from episodes_info GROUP by season;
Query ID = cloudera_20180630133030_3fc22dea-904a-4888-ba82-bcd1273b7f45
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 13:31:14,110 Stage-1 map = 0%,  reduce = 0%
2018-06-30 13:31:28,226 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.7 sec
2018-06-30 13:31:41,729 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.78 sec
MapReduce Total cumulative CPU time: 3 seconds 780 msec
Ended Job = job_1530386334216_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.78 sec   HDFS Read: 10992 HDFS Write: 13 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 780 msec
OK
1       4
2       12
3       9
Time taken: 43.474 seconds, Fetched: 3 row(s)
```

4. Built in function – Explode

```
hive> select explode(episode_title) from episodes_info;
OK
1       The Stakeout
2       The Robbery
3       Male Unbonding
4       The Stock Tip
1       The Ex-Girlfriend
2       The Pony Remark
3       The Jacket
4       The Phone Message
5       The Apartment
6       The Statue
7       The Revenge
8       The Heart Attack
9       The Deal
10      The Baby Shower
11      The Chinese Restaurant
12      The Busboy
1       The Note
2       The Truth
3       The Pen
4       The Dog
5       The Library
6       The Parking Garage
7       The Cafe
8       The Tape
9       The Nose Job
Time taken: 0.246 seconds, Fetched: 25 row(s)
```

5. Built in function – Map values

```
hive> select map_values(episode_title) from episodes_info;
OK
["The Stakeout"]
["The Robbery"]
["Male Unbonding"]
["The Stock Tip"]
["The Ex-Girlfriend"]
["The Pony Remark"]
["The Jacket"]
["The Phone Message"]
["The Apartment"]
["The Statue"]
["The Revenge"]
["The Heart Attack"]
["The Deal"]
["The Baby Shower"]
["The Chinese Restaurant"]
["The Busboy"]
["The Note"]
["The Truth"]
["The Pen"]
["The Dog"]
["The Library"]
["The Parking Garage"]
["The Cafe"]
["The Tape"]
["The Nose Job"]
Time taken: 0.145 seconds, Fetched: 25 row(s)
```

6. Built in function – Sum

```
hive> select season,SUM(id)from episodes_info GROUP BY season;
Query ID = cloudera_20180630145050_0a2f8648-f891-4e8d-99e0-a1c236c2cc5b
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 14:50:33,804 Stage-1 map = 0%,  reduce = 0%
2018-06-30 14:50:49,479 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.56 sec
2018-06-30 14:51:02,974 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.63 sec
MapReduce Total cumulative CPU time: 3 seconds 630 msec
Ended Job = job_1530386334216_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.63 sec   HDFS Read: 11036 HDFS Write: 17 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 630 msec
OK
1       10
2       126
3       189
Time taken: 63.815 seconds, Fetched: 3 row(s)
```

7. Nested Query 1

```
hive> select country_code,SUM(votes) As Total_votes,COUNT(Z.resturant_name)As Resturants_count FROM zomato_table as Z where Z.resturant_name IN (select Z1.resturant_name FROM zomato_table as Z1 where  Z1.resturant_name=Z.resturant_name)
GROUP BY Z.country_code;
Query ID = cloudera_20180630193131_83779ad4-48a2-423e-8ea4-0b5ce37328cb
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20180630193131_83779ad4-48a2-423e-8ea4-0b5ce37328cb.log
2018-06-30 07:31:26     Starting to launch local task to process map join;      maximum memory = 1013645312
2018-06-30 07:31:30     Dump the side-table for tag: 1 with group count: 7453 into file: file:/tmp/cloudera/c1d2e1ee-cdbc-4356-9ea9-245d27080424/hive_2018-06-30_19-31-10_499_1641235645003585044-1/-local-10004/HashTable-Stage-2/MapJoin-mapfile61--.hashtable
2018-06-30 07:31:31     Uploaded 1 File to: file:/tmp/cloudera/c1d2e1ee-cdbc-4356-9ea9-245d27080424/hive_2018-06-30_19-31-10_499_1641235645003585044-1/-local-10004/HashTable-Stage-2/MapJoin-mapfile61--.hashtable (379710 bytes)
2018-06-30 07:31:31     End of local task; Time Taken: 5.075 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0040, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0040/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0040
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2018-06-30 19:31:50,311 Stage-2 map = 0%,  reduce = 0%
2018-06-30 19:32:09,873 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 4.14 sec
2018-06-30 19:32:24,263 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 6.48 sec
MapReduce Total cumulative CPU time: 6 seconds 480 msec
Ended Job = job_1530386334216_0040
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 6.48 sec   HDFS Read: 2272961 HDFS Write: 170 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 480 msec
OK
country_code    total_votes     resturants_count
NULL    45      24
1       202267  8636
14      38      24
30      29      60
37      7       4
94      400018  21
148     89      40
162     5814    22
166     216     20
184     25      20
189     1890    60
191     NULL    20
208     70      34
214     498     60
215     124     80
```

8. Nested Query 2

```
hive> select COUNT(resturant_name)as count_resturant,price_range as price_range from zomato_table where aggregrate_rating>=3 GROUP BY price_range;
Query ID = cloudera_20180630203535_a9ff8df0-adee-4499-bc03-6c7f77eea4e0
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0046, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0046/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0046
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 20:35:14,626 Stage-1 map = 0%,  reduce = 0%
2018-06-30 20:35:29,923 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.64 sec
2018-06-30 20:35:55,779 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.9 sec
MapReduce Total cumulative CPU time: 4 seconds 900 msec
Ended Job = job_1530386334216_0046
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.9 sec   HDFS Read: 2271101 HDFS Write: 44 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 900 msec
OK
count_resturant price_range
1331    NULL
1       2
1       4
2       72
87      77
1       83
2       120
7       121
Time taken: 57.759 seconds, Fetched: 8 row(s)
```

9. Nested Query 3

```
hive> select YEAR(airdate),count(*)AS C from episodes_info GROUP BY YEAR(airdate);
Query ID = cloudera_20180630153333_ce57b37d-9a45-4031-ae9c-6b1e4d08987e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0010, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 15:34:03,039 Stage-1 map = 0%,  reduce = 0%
2018-06-30 15:34:31,841 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.0 sec
2018-06-30 15:34:50,387 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.0 sec
MapReduce Total cumulative CPU time: 5 seconds 0 msec
Ended Job = job_1530386334216_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.0 sec   HDFS Read: 15042 HDFS Write: 37 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 0 msec
OK
1990    8
1991    42
1992    17
1994    4
1998    4
Time taken: 63.911 seconds, Fetched: 5 row(s)
```

10. Pattern recognition

```
hive> select count(*)AS Tom_C from episodes_info where director LIKE '%Tom Cherones%';
Query ID = cloudera_20180630153636_31a4e2ce-b4e9-4a7f-9c0e-869f0f706ef2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1530386334216_0011, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1530386334216_0011/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1530386334216_0011
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-06-30 15:36:21,388 Stage-1 map = 0%,  reduce = 0%
2018-06-30 15:36:36,260 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.32 sec
2018-06-30 15:36:48,551 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.31 sec
MapReduce Total cumulative CPU time: 4 seconds 310 msec
Ended Job = job_1530386334216_0011
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.31 sec   HDFS Read: 15182 HDFS Write: 3 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 310 msec
OK
66
Time taken: 43.728 seconds, Fetched: 1 row(s)
```

Question 2:

2. *SOLR USECASE*

a. **Create a Solr Collection including our own Field Types**
b. **Perform 10 intuitive questions in Dataset (e.g.: pattern recognition, topic discussion, most important terms, etc.). Use your innovation to think out of box. Implement atleast 5 nested queries among the 10.**
c. **Record the time execution for the queries.**

Queries implemented:

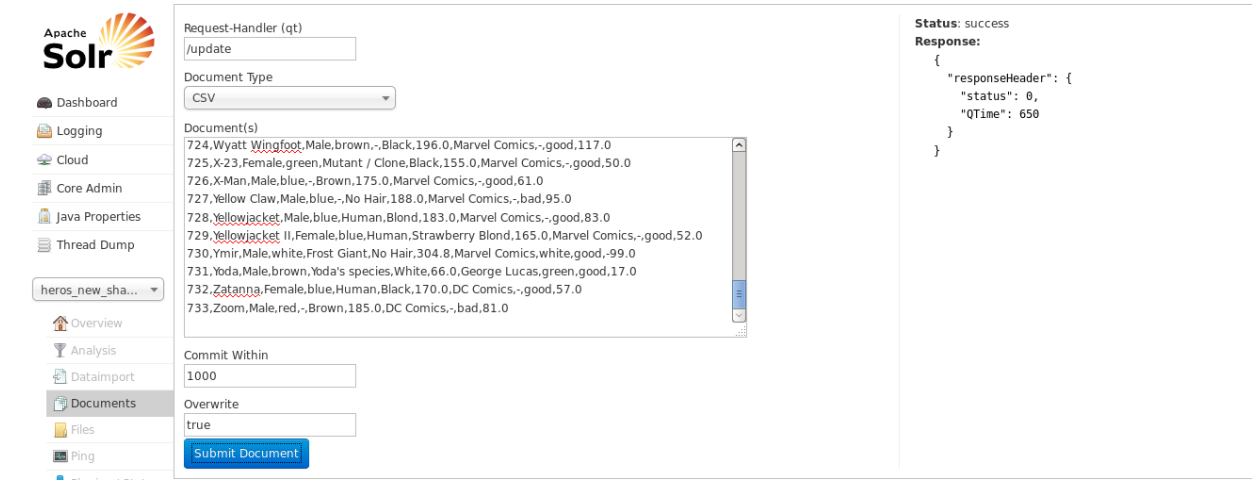
Below are the queries that has been implemented on the data sets heros.

1. The table creation has been done in to the Solr Admin.
2. And the data has been loaded in to the Solr for performing the respective queries
3. Built in functions and nested queries

Output Screenshot:

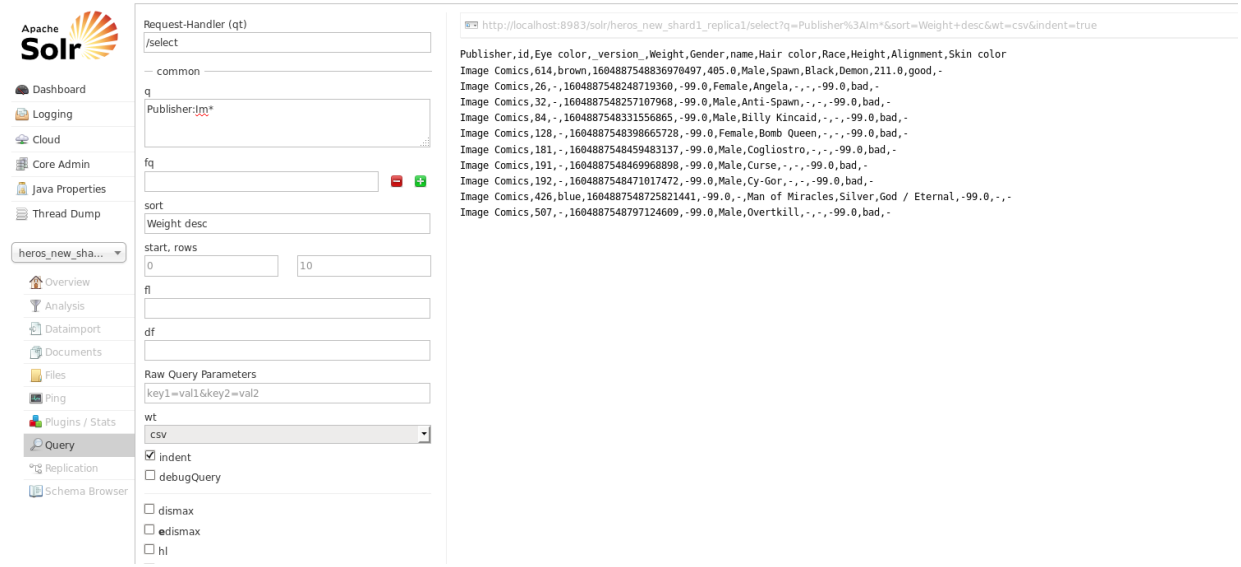
1. Collection Creation & Loading data in to solr using complex data types:

```
[cloudera@quickstart ~]$ ls /tmp/heros_new
conf
[cloudera@quickstart ~]$ ls /tmp/heros_new/conf
admin-extra.html            clustering    lang                        protwords.txt                        _schema_analysis_synonyms_english.json  scripts.conf          spellings.txt  update-script.js
admin-extra.menu-bottom.html  currency.xml  mapping-FoldToASCII.txt     _rest_managed.json                   schema.xml                              solrconfig.xml        stopwords.txt  velocity
admin-extra.menu-top.html     elevate.xml   mapping-ISOLatin1Accent.txt  _schema_analysis_stopwords_english.json  schema.xml~                             solrconfig.xml.secure  synonyms.txt   xslt
[cloudera@quickstart ~]$ gedit /tmp/heros_new/conf/schema.xml
[cloudera@quickstart ~]$ solrctl instancedir --create heros_new /tmp/heros_new
Uploading configs from /tmp/heros_new/conf to quickstart.cloudera:2181/solr. This may take up to a minute.
[cloudera@quickstart ~]$ solrctl collection --create heros_new
[cloudera@quickstart ~]$
```

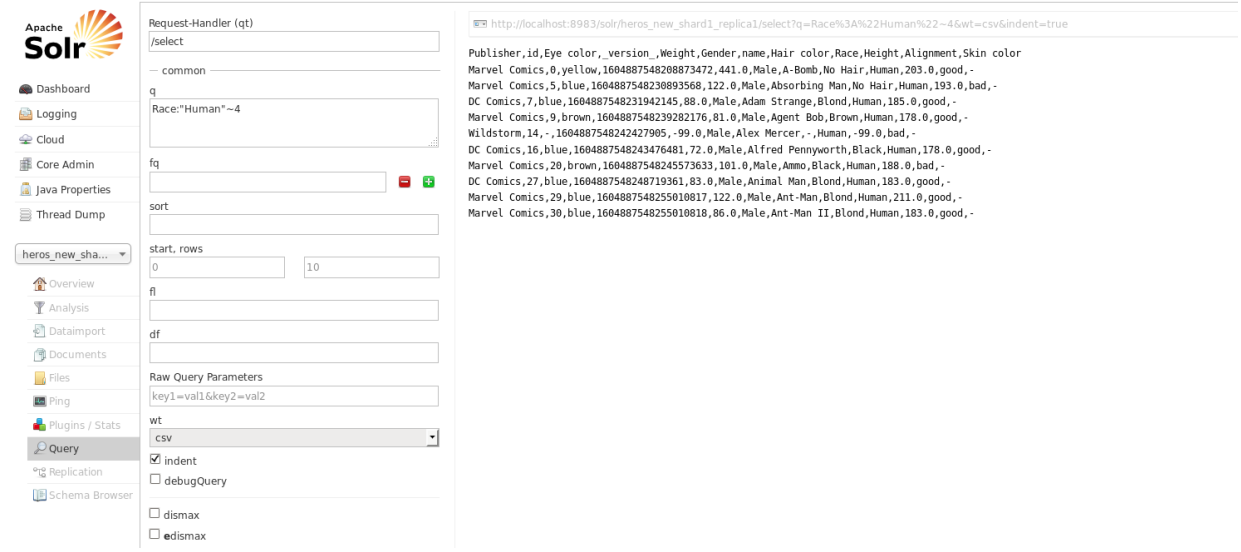


2. Query 1- Range and Sort

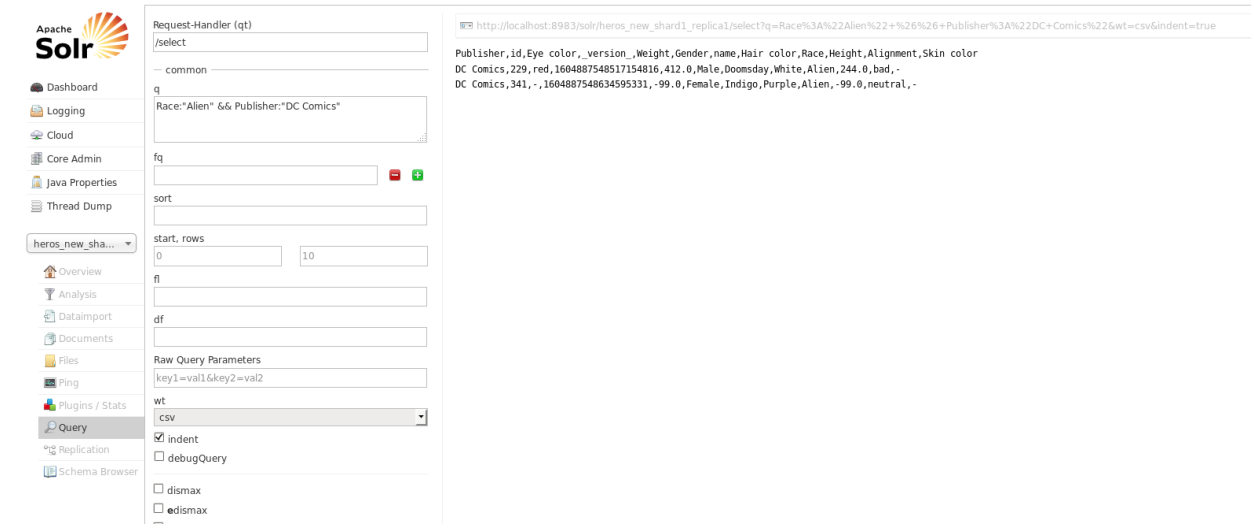
3. Query 2



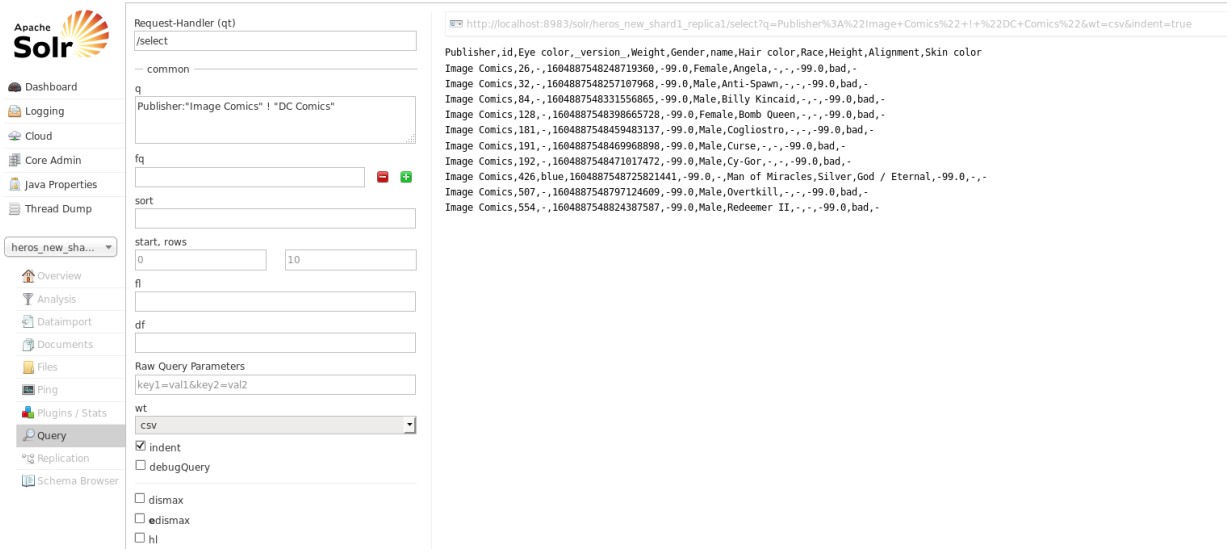
4. Query 3



5. Query 4

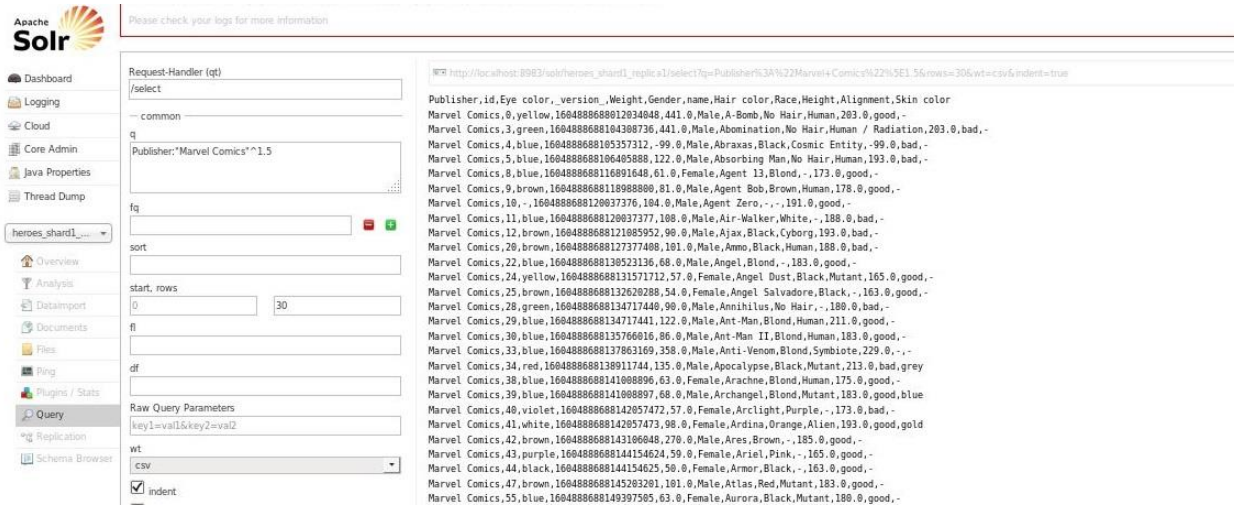


6. Query 5



7. Query 6

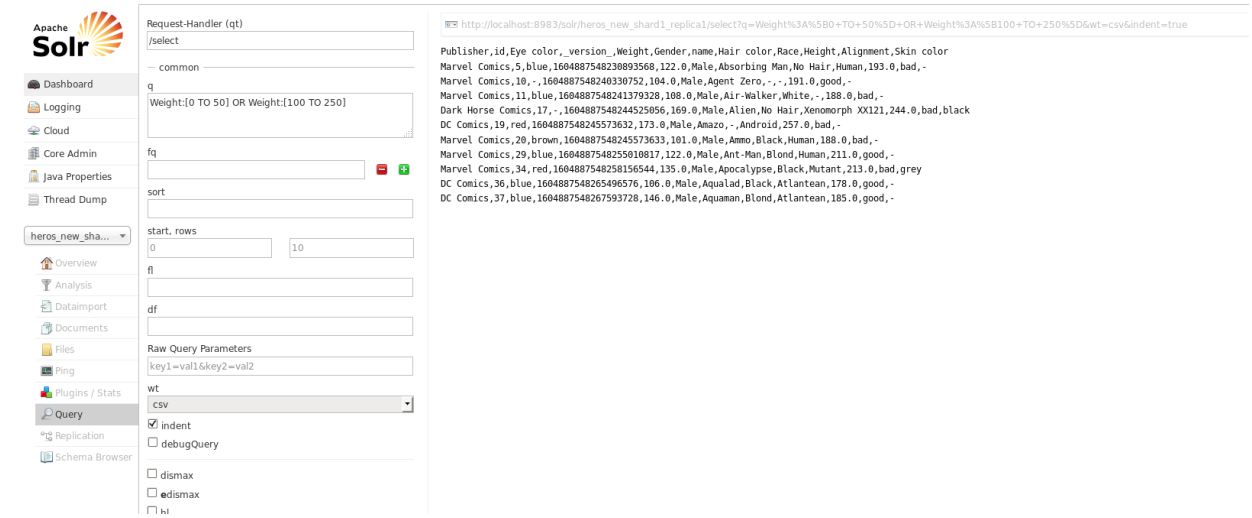
8. Query 7



9. Query 8

10. Query 9

11. **Query 10**



Conclusion:

As stated, the above workflow with certain set of parameters is followed in solving the execution by implementing the core and basic concepts of hive and solr .

Source code link:

https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/tree/master/Lab%202

Video Link: Provided in wiki link https://youtu.be/jUmwqRbMOps

Wiki Link:

https://github.com/PragathiThammaneni/Bigdata-Programming--Hadoop-Spark/wiki/Lab-2-Assignment