Course name

Trivedi Maharshi Mayankbhai

Student ID

General Theoretical Questions

Question 1: List all big data-specific formats that you know.

Answer 1: There are many file formats, which are being used to deal big-data. Plain text file, Sequence file, Avro, Parquet and ORC. Below a short description is provided for each big-data type:

Plain text file: Data is saved in *txt* or *csv* formats, they are used in the non-Hadoop environment as well. Heavy disk space/storage can be occupied by these plain text file formats. Thus, a robust compression is required.

Sequence file: This file type was mainly developed for MapReduce. Each entry of the data is encoded with a key and a value. Hence the data is stored in a binary format and occupies lesser storage space on the disk than the file formats which are text-based. It is also compatible with block-level compression, which is one of the advantages of the Sequence file.

Avro file: While dealing with the big-data files, Avro format is a good choice. On top of a normal file-format, it also supports serialization-deserialization and blocks compression. It is a row based and a splittable format. Avro is only a machine-readable file format.

Parquet and *ORC file*: ORC stands for Optimized Row Columnar. Unlike the Avro file, it also stores the data in columnar format. Hence, the horizontal and vertical partition of the data is possible. However, these file formats are also machine-readable only. Because these file formats are columnar, they outperform in terms of storage optimization than any other file formats.

Question 2: Why the compression of data matters for Hadoop?

Answer 2: Necessity of compression in a typical Hadoop eco-system is because of the huge volumes of data, it must deal with. Reduction in- (1) space needed for storing data and (2) data transfer speed to or from the disk, could be two significant advantages while dealing with big-data. These two advantages could be introduced with data compression. For example, MapReduce job for a compressed-large volume results into a low-latency task.

However, the data compression rate must be tuned with a trade-off between compression and speed of computation. Compressed data first gets decompressed before any other operation on the cluster. Thus, it increases CPU utilization along with the compression. The more compression is set for data, the more resources are used to first decompress it.

YARN application/commands

Question 3: What is YARN? What are YARN's two most important functions?

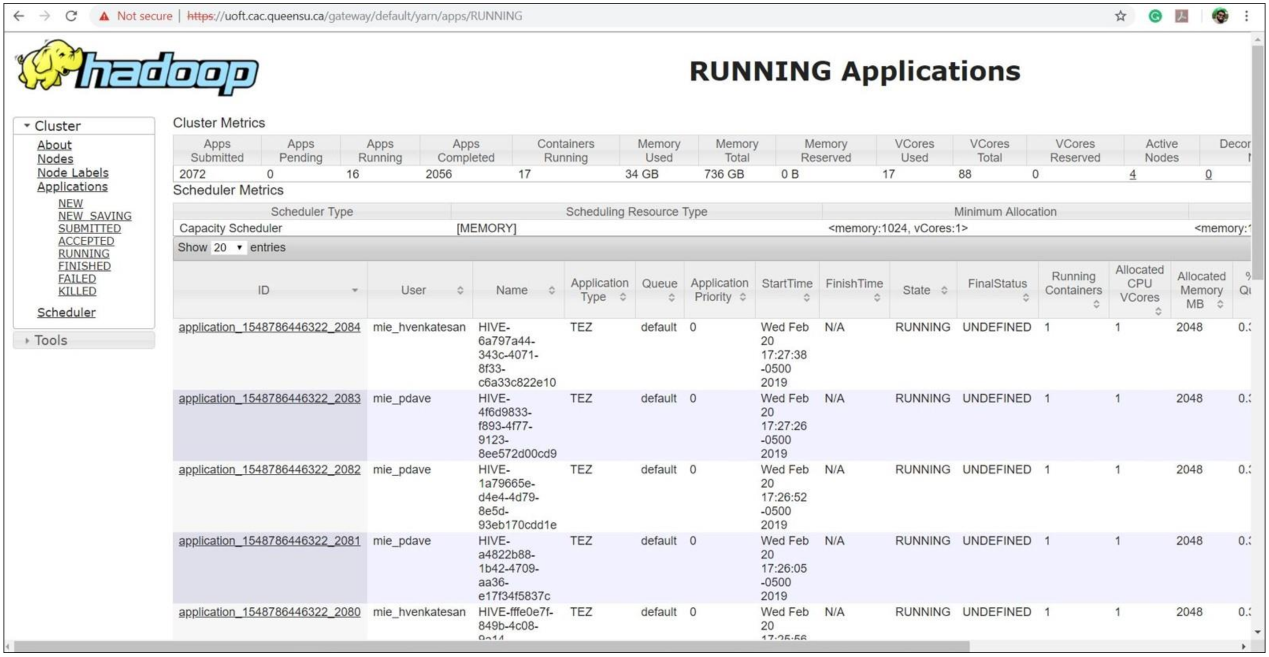
Answer 3: YARN stands for Yet Another Resource Negotiator. YARN sits in between of Hadoop File System (HDFS) and processing layer. It is essentially used for the two most important function, mentioned below:

(1) Resource Management: YARN allocates resources such as memory, to the application.

(2) Job Scheduling: It supports multiple scheduling methods for submitting the job to be processed in sequence. Some of the state-of-the-art scheduling methods are FIFO, Capacity Scheduler and Fair Scheduler.

Question 4: List all running applications

Answer 4: The Running application can be visualized with both: YARN UI and HDFS terminal. Figure 1 shows a screenshot of all the running application with YARN UI. Figure 2 lists all the running application with HDFS terminal.



RUNNING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decor
2072	0	16	2056	17	34 GB	736 GB	0 B	17	88	0	4	0

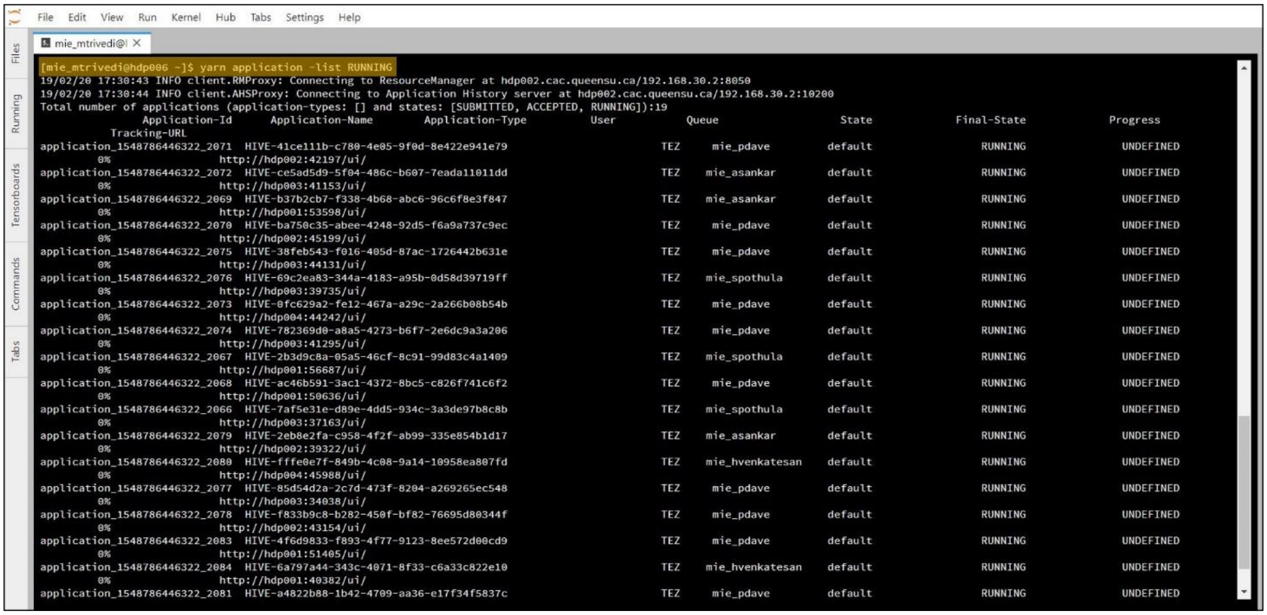
Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% Qu
application_1548786446322_2084	mie_hvenkatesan	HIVE-6a797a44-343c-4071-8f33-c6a33c822e10	TEZ	default	0	Wed Feb 20 17:27:38 -0500 2019	N/A	RUNNING	UNDEFINED	1	1	2048	0.:
application_1548786446322_2083	mie_pdave	HIVE-4f6d9833-f893-4f77-9123-8ee572d00cd9	TEZ	default	0	Wed Feb 20 17:27:26 -0500 2019	N/A	RUNNING	UNDEFINED	1	1	2048	0.:
application_1548786446322_2082	mie_pdave	HIVE-1a79665e-d4e4-4d79-8e5d-93eb170cdd1e	TEZ	default	0	Wed Feb 20 17:26:52 -0500 2019	N/A	RUNNING	UNDEFINED	1	1	2048	0.:
application_1548786446322_2081	mie_pdave	HIVE-a4822b88-1b42-4709-aa36-e17f34f5837c	TEZ	default	0	Wed Feb 20 17:26:05 -0500 2019	N/A	RUNNING	UNDEFINED	1	1	2048	0.:
application_1548786446322_2080	mie_hvenkatesan	HIVE-fffe0e7f-849b-4c08-	TEZ	default	0	Wed Feb 20	N/A	RUNNING	UNDEFINED	1	1	2048	0.:

Figure 1. Running the application with YARN UI.

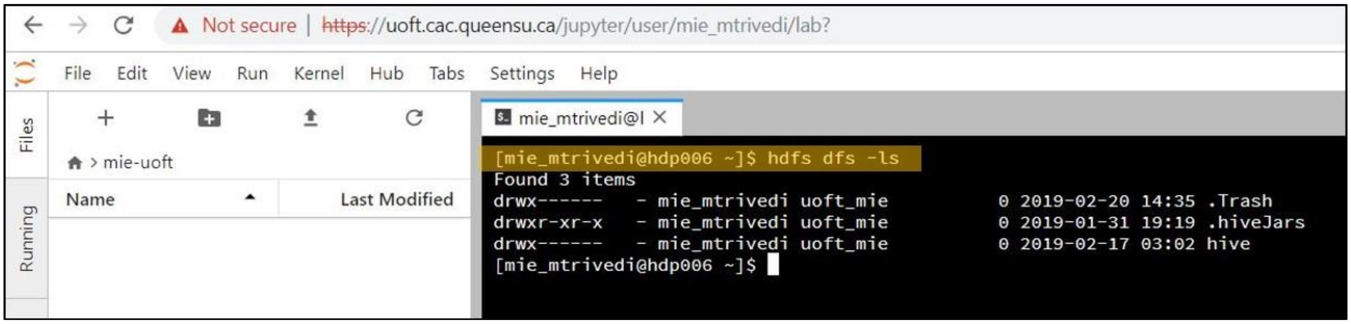


```
[mie_mtrivedi@hdp006 ~]$ yarn application -list RUNNING
19/02/20 17:30:43 INFO client.RMProxy: Connecting to ResourceManager at hdp002.cac.queensu.ca/192.168.30.2:8050
19/02/20 17:30:44 INFO client.AHSProxy: Connecting to Application History server at hdp002.cac.queensu.ca/192.168.30.2:10200
Total number of applications (application-types: [] and states: [SUBMITTED, ACCEPTED, RUNNING]):19
                Application-Id      Application-Name        Application-Type          User           Queue                   State             Final-State             Progress
           Tracking-URL
application_1548786446322_2071  HIVE-41ce111b-c780-4e05-9f0d-8e422e941e79                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp002:42197/ui/
application_1548786446322_2072  HIVE-ce5ad5d9-5f04-486c-b607-7eada11011dd                         TEZ     mie_asankar       default              RUNNING             UNDEFINED
         0%                 http://hdp003:41153/ui/
application_1548786446322_2069  HIVE-b37b2cb7-f338-4b68-abc6-96c6f8e3f847                         TEZ     mie_asankar       default              RUNNING             UNDEFINED
         0%                 http://hdp001:53598/ui/
application_1548786446322_2070  HIVE-ba750c35-abee-4248-92d5-f6a9a737c9ec                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp002:45199/ui/
application_1548786446322_2075  HIVE-38feb543-f016-405d-87ac-1726442b631e                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp003:44131/ui/
application_1548786446322_2076  HIVE-69c2ea83-344a-4183-a95b-0d58d39719ff                         TEZ     mie_spothula      default              RUNNING             UNDEFINED
         0%                 http://hdp003:39735/ui/
application_1548786446322_2073  HIVE-0fc629a2-fe12-467a-a29c-2a266b08b54b                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp004:44242/ui/
application_1548786446322_2074  HIVE-782369d0-a8a5-4273-b6f7-2e6dc9a3a206                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp003:41295/ui/
application_1548786446322_2067  HIVE-2b3d9c8a-05a5-46cf-8c91-99d83c4a1409                         TEZ     mie_spothula      default              RUNNING             UNDEFINED
         0%                 http://hdp001:56687/ui/
application_1548786446322_2068  HIVE-ac46b591-3ac1-4372-8bc5-c826f741c6f2                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp001:50636/ui/
application_1548786446322_2066  HIVE-7af5e31e-d89e-4dd5-934c-3a3de97b8c8b                         TEZ     mie_spothula      default              RUNNING             UNDEFINED
         0%                 http://hdp003:37163/ui/
application_1548786446322_2079  HIVE-2eb8e2fa-c958-4f2f-ab99-335e854b1d17                         TEZ     mie_asankar       default              RUNNING             UNDEFINED
         0%                 http://hdp002:39322/ui/
application_1548786446322_2080  HIVE-fffe0e7f-849b-4c08-9a14-10958ea807fd                         TEZ     mie_hvenkatesan   default              RUNNING             UNDEFINED
         0%                 http://hdp004:45988/ui/
application_1548786446322_2077  HIVE-85d54d2a-2c7d-473f-8204-a269265ec548                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp003:34038/ui/
application_1548786446322_2078  HIVE-f833b9c8-b282-450f-bf82-76695d80344f                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp002:43154/ui/
application_1548786446322_2083  HIVE-4f6d9833-f893-4f77-9123-8ee572d00cd9                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
         0%                 http://hdp001:51405/ui/
application_1548786446322_2084  HIVE-6a797a44-343c-4071-8f33-c6a33c822e10                         TEZ     mie_hvenkatesan   default              RUNNING             UNDEFINED
         0%                 http://hdp001:40382/ui/
application_1548786446322_2081  HIVE-a4822b88-1b42-4709-aa36-e17f34f5837c                         TEZ      mie_pdave        default              RUNNING             UNDEFINED
```

Figure 2. List of all the running application with HDFS terminal (Highlighted yellow is the command).

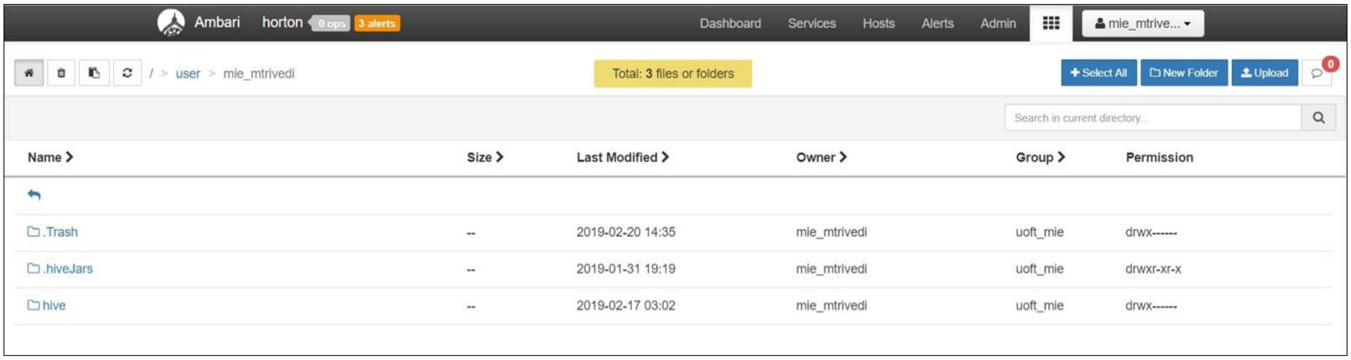
HDFS commands

Question 5: Create a folder/directory 'Lab1_results' in your own HDFS directory.
Answer 5: Command used to list the directories/files in a home folder, before generating *Lab1_results* directory: *hdfs dfs -ls*. Figure 3 and Figure 4 shows the listing of all the directories before generating Lab1_results directory. All the commands are implemented in HDFS terminal only.



```
[mie_mtrivedi@hdp006 ~]$ hdfs dfs -ls
Found 3 items
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-20 14:35 .Trash
drwxr-xr-x   - mie_mtrivedi uoft_mie          0 2019-01-31 19:19 .hiveJars
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-17 03:02 hive
[mie_mtrivedi@hdp006 ~]$
```

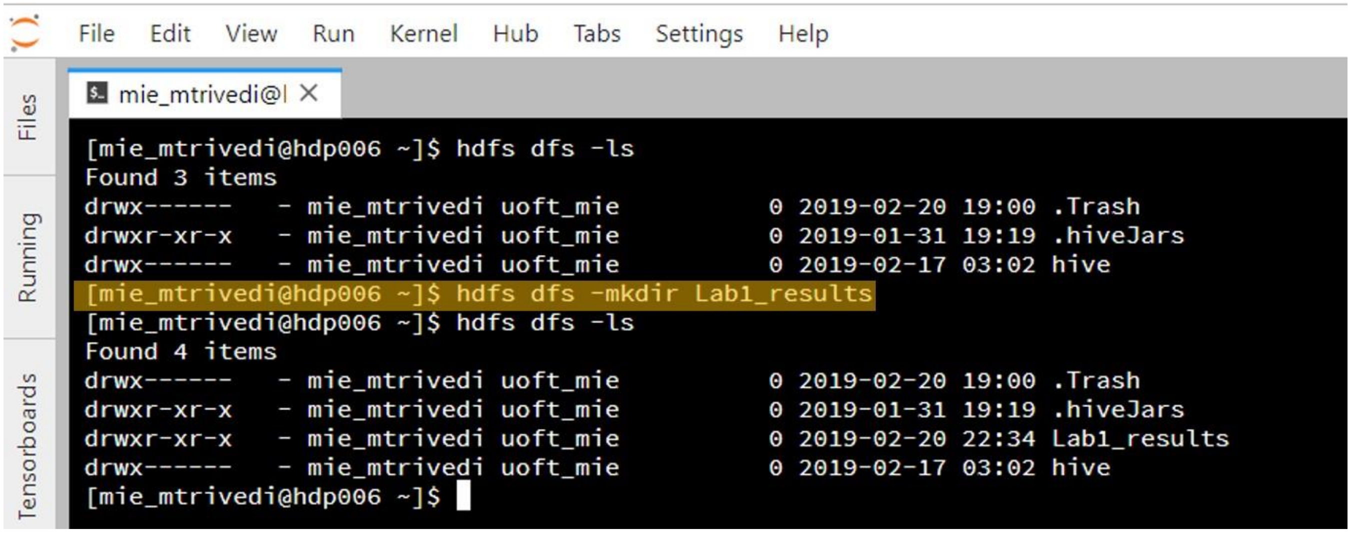
Figure 3. Listing of all the directories/files in present HDFS directory with HDFS terminal.



Name	Size	Last Modified	Owner	Group	Permission
.Trash	--	2019-02-20 14:35	mie_mtrivedi	uoft_mie	drwx------
.hiveJars	--	2019-01-31 19:19	mie_mtrivedi	uoft_mie	drwxr-xr-x
hive	--	2019-02-17 03:02	mie_mtrivedi	uoft_mie	drwx------

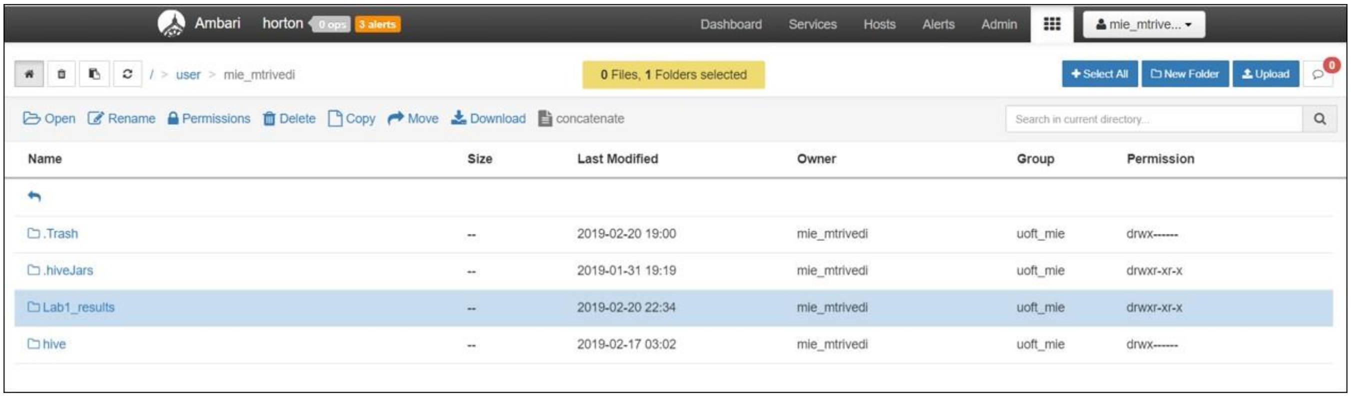
Figure 4. Listing of all the directories/files with HDFS UI.

Directory *Lab1_results* is generated with *hdfs dfs -mkdir Lab1_results* in the present working directory. Figure 5 shows the usage of *mkdir* command to generate *Lab1_results* along with the listing of all the directories after generating *Lab1_results*.



```
[mie_mtrivedi@hdp006 ~]$ hdfs dfs -ls
Found 3 items
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-20 19:00 .Trash
drwxr-xr-x   - mie_mtrivedi uoft_mie          0 2019-01-31 19:19 .hiveJars
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-17 03:02 hive
[mie_mtrivedi@hdp006 ~]$ hdfs dfs -mkdir Lab1_results
[mie_mtrivedi@hdp006 ~]$ hdfs dfs -ls
Found 4 items
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-20 19:00 .Trash
drwxr-xr-x   - mie_mtrivedi uoft_mie          0 2019-01-31 19:19 .hiveJars
drwxr-xr-x   - mie_mtrivedi uoft_mie          0 2019-02-20 22:34 Lab1_results
drwx------   - mie_mtrivedi uoft_mie          0 2019-02-17 03:02 hive
[mie_mtrivedi@hdp006 ~]$
```

Figure 5. Command to generate a directory *Lab1_results*, with *mkdir* command in HDFS.



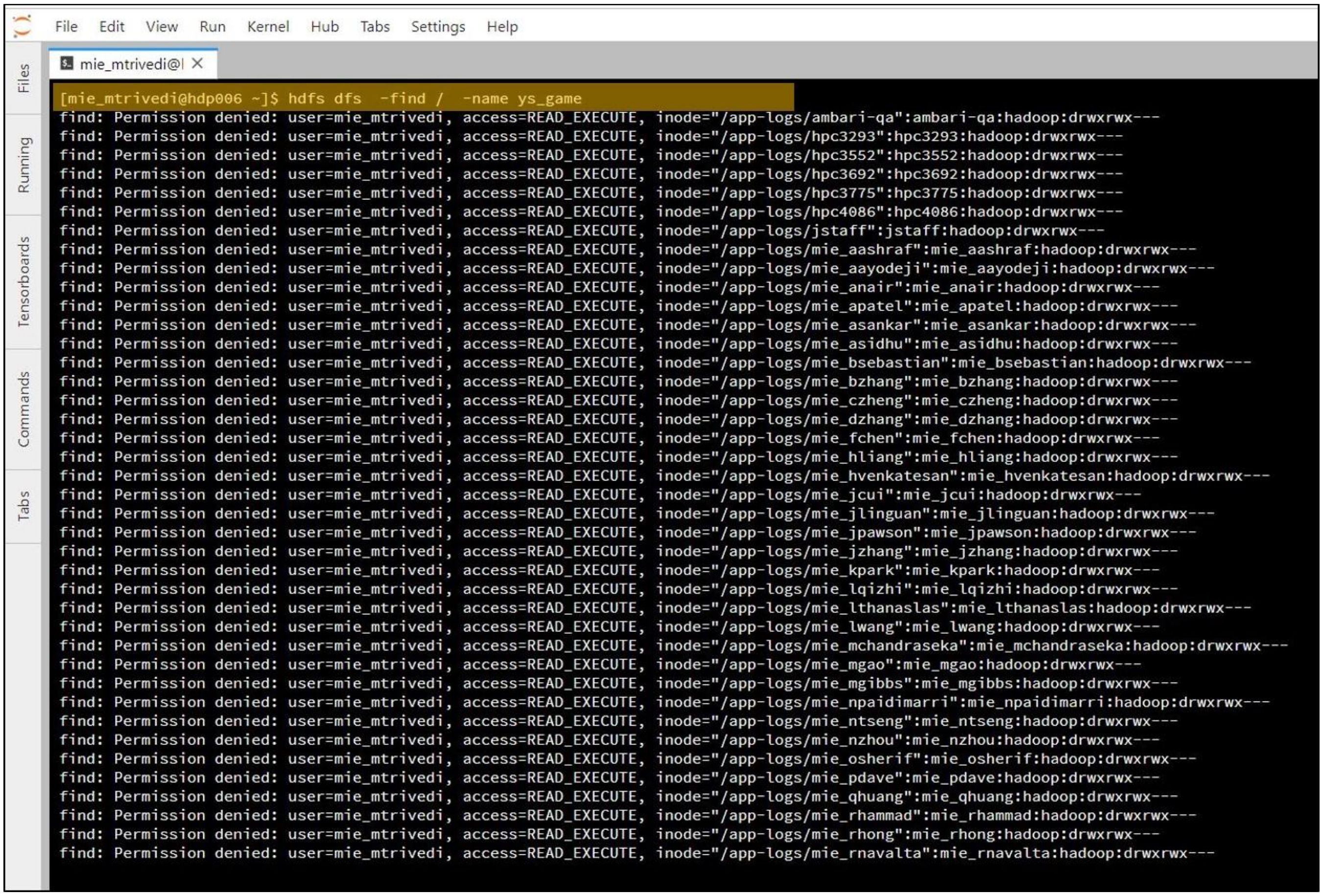
Name	Size	Last Modified	Owner	Group	Permission
.Trash	--	2019-02-20 19:00	mie_mtrivedi	uoft_mie	drwx------
.hiveJars	--	2019-01-31 19:19	mie_mtrivedi	uoft_mie	drwxr-xr-x
Lab1_results	--	2019-02-20 22:34	mie_mtrivedi	uoft_mie	drwxr-xr-x
hive	--	2019-02-17 03:02	mie_mtrivedi	uoft_mie	drwx------

Figure 6. Checking HDFS UI for Lab1_results after generating it with HDFS command.

Question 6: Where in HDFS is MIE_Lecture4.ys_game file located? Provide path of the file.

Answer 6: Command *hdfs dfs -find / -name ys_game* could be used to find the location of this file with respect to the root directory. Figure 7 shows the usage of *find* command in HDFS terminal.

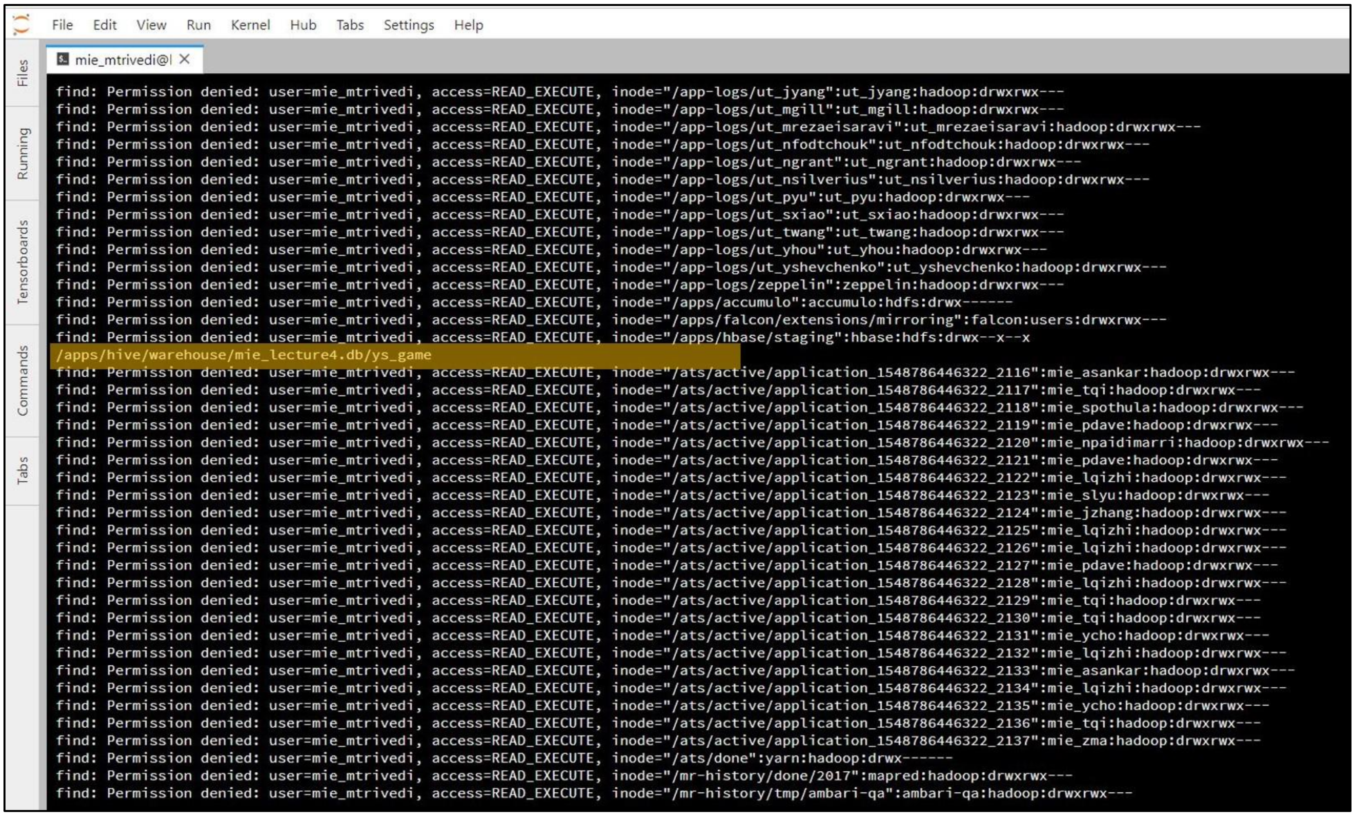
Path of the file ys_game: /apps/hive/warehouse/mie_lecture4.db/ys_game



```
[mie_mtrivedi@hdp006 ~]$ hdfs dfs  -find /  -name ys_game
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ambari-qa":ambari-qa:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/hpc3293":hpc3293:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/hpc3552":hpc3552:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/hpc3692":hpc3692:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/hpc3775":hpc3775:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/hpc4086":hpc4086:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/jstaff":jstaff:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_aashraf":mie_aashraf:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_aayodeji":mie_aayodeji:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_anair":mie_anair:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_apatel":mie_apatel:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_asankar":mie_asankar:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_asidhu":mie_asidhu:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_bsebastian":mie_bsebastian:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_bzhang":mie_bzhang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_czheng":mie_czheng:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_dzhang":mie_dzhang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_fchen":mie_fchen:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_hliang":mie_hliang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_hvenkatesan":mie_hvenkatesan:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_jcui":mie_jcui:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_jlinguan":mie_jlinguan:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_jpawson":mie_jpawson:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_jzhang":mie_jzhang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_kpark":mie_kpark:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_lqizhi":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_lthanaslas":mie_lthanaslas:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_lwang":mie_lwang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_mchandraseka":mie_mchandraseka:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_mgao":mie_mgao:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_mgibbs":mie_mgibbs:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_npaidimarri":mie_npaidimarri:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_ntseng":mie_ntseng:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_nzhou":mie_nzhou:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_osherif":mie_osherif:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_pdave":mie_pdave:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_qhuang":mie_qhuang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_rhammad":mie_rhammad:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_rhong":mie_rhong:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/mie_rnavalta":mie_rnavalta:hadoop:drwxrwx---
```

Figure 7. Command to locate *ys_game* file in HDFS terminal.

The output of *find* command lists the location of all the files with respect to its root directory. Scrolling down the output terminal window, *ys_game* could be located. Figure 8 shows the location of *ys_game* with the highlighted field.

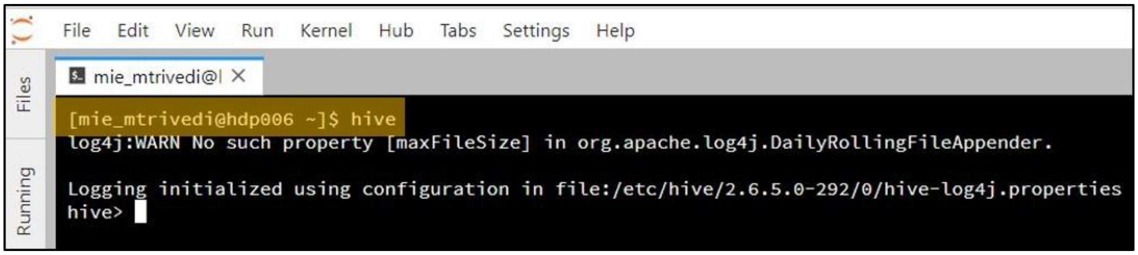


```
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_jyang":ut_jyang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_mgill":ut_mgill:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_mrezaeisaravi":ut_mrezaeisaravi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_nfodtchouk":ut_nfodtchouk:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_ngrant":ut_ngrant:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_nsilverius":ut_nsilverius:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_pyu":ut_pyu:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_sxiao":ut_sxiao:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_twang":ut_twang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_yhou":ut_yhou:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/ut_yshevchenko":ut_yshevchenko:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/app-logs/zeppelin":zeppelin:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/apps/accumulo":accumulo:hdfs:drwx------
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/apps/falcon/extensions/mirroring":falcon:users:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/apps/hbase/staging":hbase:hdfs:drwx--x--x
/apps/hive/warehouse/mie_lecture4.db/ys_game
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2116":mie_asankar:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2117":mie_tqi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2118":mie_spothula:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2119":mie_pdave:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2120":mie_npaidimarri:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2121":mie_pdave:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2122":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2123":mie_slyu:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2124":mie_jzhang:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2125":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2126":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2127":mie_pdave:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2128":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2129":mie_tqi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2130":mie_tqi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2131":mie_ycho:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2132":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2133":mie_asankar:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2134":mie_lqizhi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2135":mie_ycho:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2136":mie_tqi:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/active/application_1548786446322_2137":mie_zma:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/ats/done":yarn:hadoop:drwx------
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/mr-history/done/2017":mapred:hadoop:drwxrwx---
find: Permission denied: user=mie_mtrivedi, access=READ_EXECUTE, inode="/mr-history/tmp/ambari-qa":ambari-qa:hadoop:drwxrwx---
```

Figure 8. Global path of *ys_game* is highlighted with yellow color.

Question 7: What format is underlying Hive tables saved in? How can you find the format?

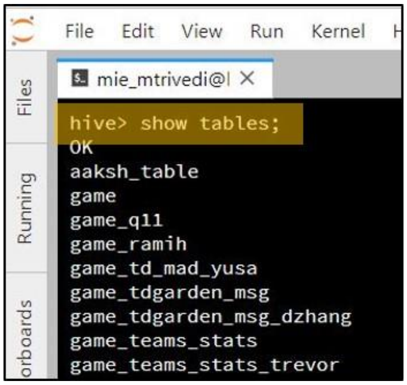
Answer 7: Through HDFS, entry to the Hive environment is made.



```
[mie_mtrivedi@hdp006 ~]$ hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.5.0-292/0/hive-log4j.properties
hive> 
```

Figure 9. Entering the Hive with HDFS.

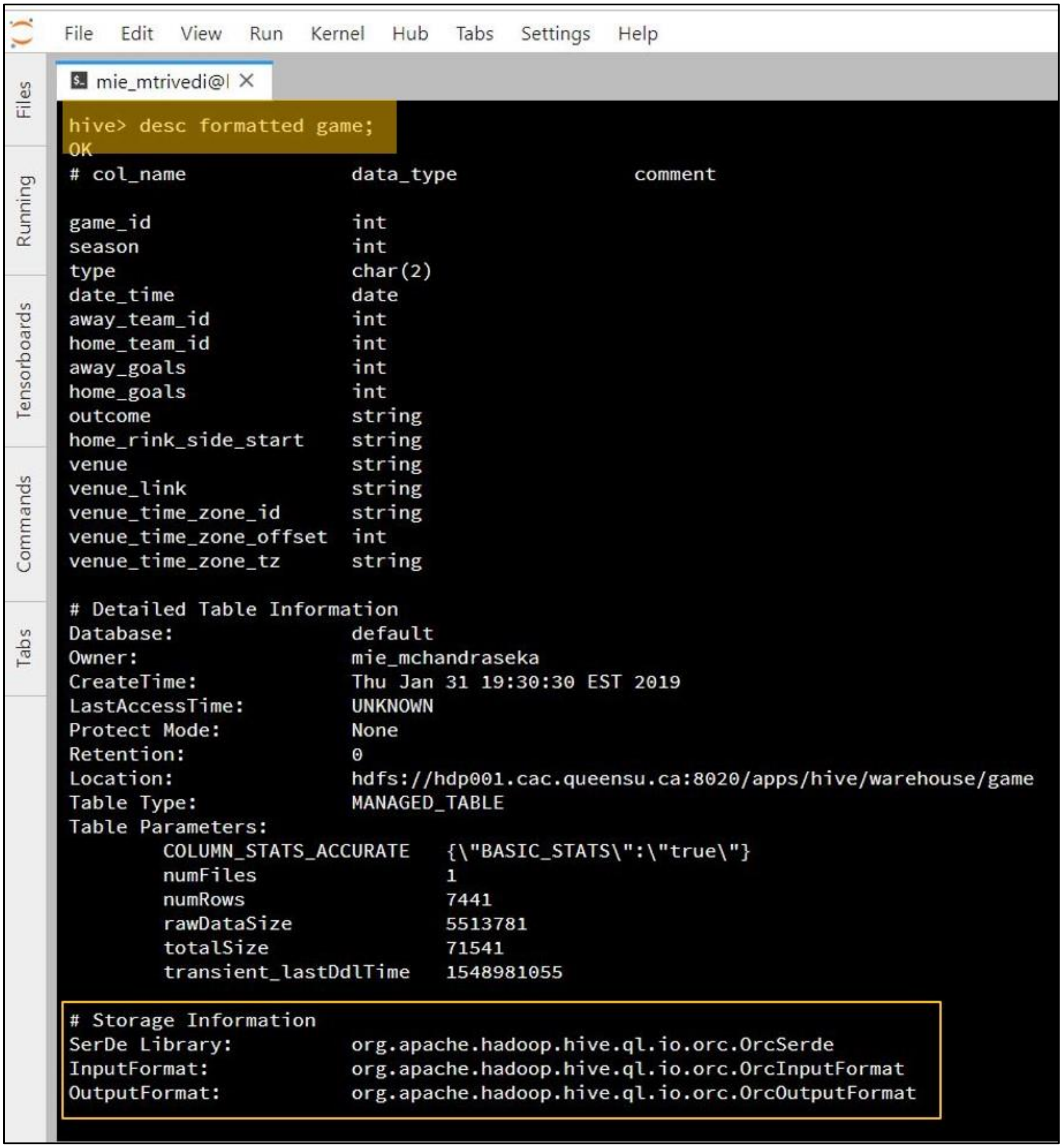
After entering the Hive, all the tables are listed with *show tables*.



```
hive> show tables;
OK
aaksh_table
game
game_q11
game_ramih
game_td_mad_yusa
game_tdgarden_msg
game_tdgarden_msg_dzhang
game_teams_stats
game_teams_stats_trevor
```

Figure 10. Listing of all the tables after entering in Hive environment with HDFS.

Once all the tables are listed, any of them is picked up and file format is checked with *desc formatted Table_name*. Here, table game is chosen for checking the file format and thus the command *desc formatted game* is used.



```
hive> desc formatted game;
OK
# col_name              data_type               comment

game_id                 int
season                  int
type                    char(2)
date_time               date
away_team_id            int
home_team_id            int
away_goals              int
home_goals              int
outcome                 string
home_rink_side_start    string
venue                   string
venue_link              string
venue_time_zone_id      string
venue_time_zone_offset  int
venue_time_zone_tz      string

# Detailed Table Information
Database:               default
Owner:                  mie_mchandraseka
CreateTime:             Thu Jan 31 19:30:30 EST 2019
LastAccessTime:         UNKNOWN
Protect Mode:           None
Retention:              0
Location:               hdfs://hdp001.cac.queensu.ca:8020/apps/hive/warehouse/game
Table Type:             MANAGED_TABLE
Table Parameters:
        COLUMN_STATS_ACCURATE   {\"BASIC_STATS\":\"true\"}
        numFiles                1
        numRows                 7441
        rawDataSize             5513781
        totalSize               71541
        transient_lastDdlTime   1548981055

# Storage Information
SerDe Library:          org.apache.hadoop.hive.ql.io.orc.OrcSerde
InputFormat:            org.apache.hadoop.hive.ql.io.orc.OrcInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.orc.OrcOutputFormat
```

Figure 11. The file format is checked for the *game* table. The file format is bounded with a rectangle which is shown with yellow color.

Above described commands can be used to check the underlying format of all the files. As it can be seen from Figure 11, the file format of the *game* is ORC (Optimized Row Columnar).

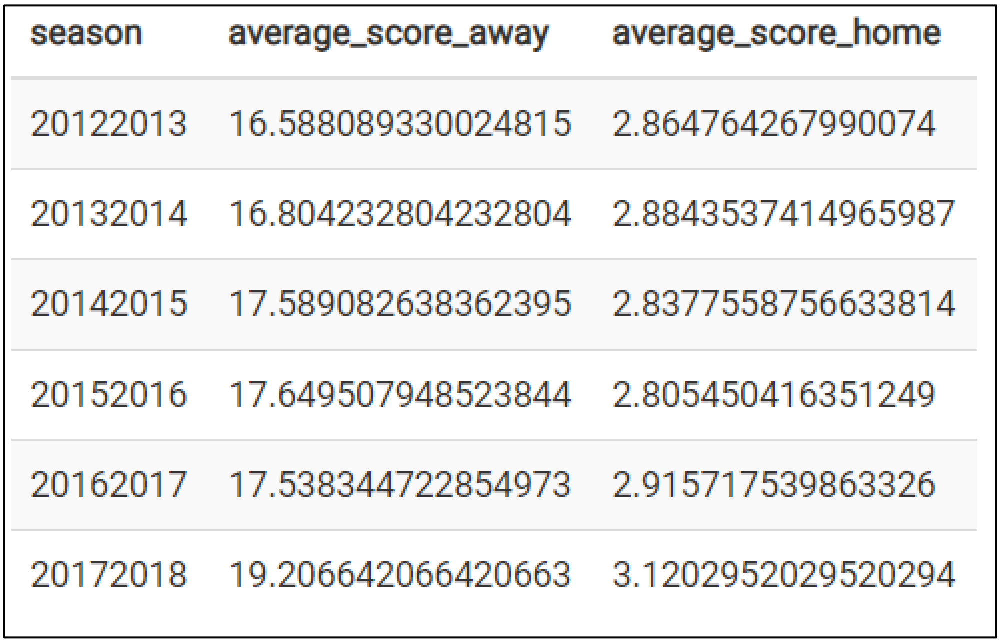
Hive functionality, file manipulation in Hive via Ambari

Question 8: Calculate the average score of the away team, and the home team for every season. Show results by ordering by the season.

Hive Query

```
-- Seasonwise average score for both, home and away teams

select season,

avg(home_team_id) Average_score_away,

avg(home_goals) Average_score_home

from game

group by season

sort by season;

-- End of the query
```

Result



season	average_score_away	average_score_home
20122013	16.588089330024815	2.864764267990074
20132014	16.804232804232804	2.8843537414965987
20142015	17.589082638362395	2.8377558756633814
20152016	17.649507948523844	2.805450416351249
20162017	17.538344722854973	2.915717539863326
20172018	19.206642066420663	3.1202952029520294

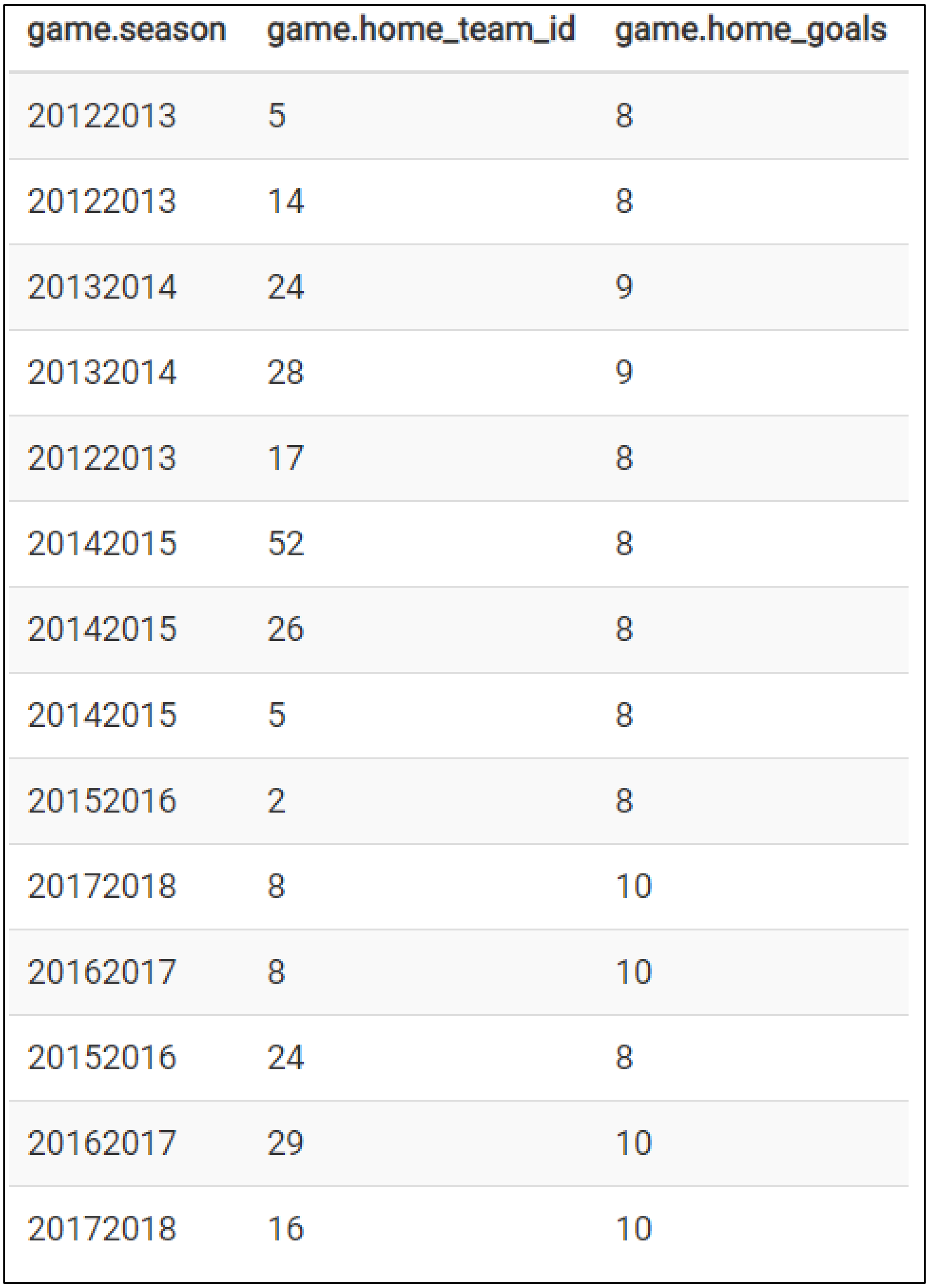
Question 9: For every season, find the highest scoring home team. Include the score in your answer.

Hive Query

```
-- List all the team ids which scored highest goals in a season

create temporary table if not exists season_go_max as

select season,

max(home_goals) goals_max

from game

group by season;
```

```
select game.season,game.home_team_id,game.home_goals
from game
join season_go_max
on game.season=season_go_max.season and game.home_goals=season_go_max.goals_max
-- End of the query
```

Result



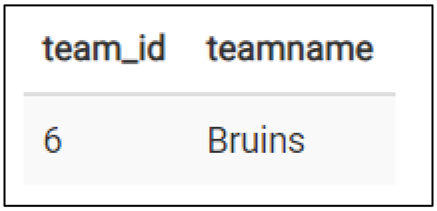
game.season	game.home_team_id	game.home_goals
20122013	5	8
20122013	14	8
20132014	24	9
20132014	28	9
20122013	17	8
20142015	52	8
20142015	26	8
20142015	5	8
20152016	2	8
20172018	8	10
20162017	8	10
20152016	24	8
20162017	29	10
20172018	16	10

Question 10: Provide team_id and name for the team that played as home team at TD Garden.

Hive Query

```
-- Team id and Team name which played at TD Garden location

select team_id,teamname from team_info

where team_info.team_id in (select distinct home_team_id from game

                                                  where venue='TD Garden')

-- End of the query
```

Result



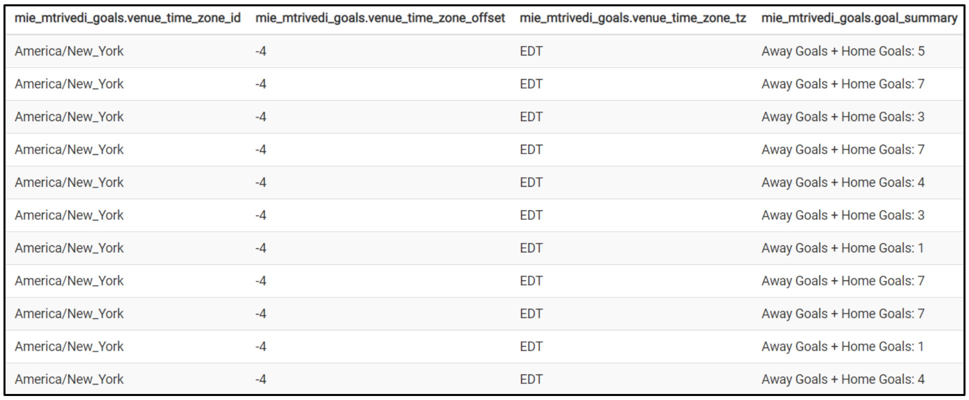
team_id	teamname
6	Bruins

Question 11: Create a new table that lists all games that happened at TD Garden or Madison Square Garden. Add a column that summarizes away and home goals.

Hive Query

```
-- Summarize goals of home and away team which played at TD Garden or Medison Square Garden

create table if not exists mie_mtrivedi_goals as

select *, concat('Away Goals + Home Goals: ',away_goals+home_goals) as goal_summary

from game

where venue='TD Garden' or venue='Madison Square Garden';


select * from mie_mtrivedi_goals

-- End of the query
```

Result



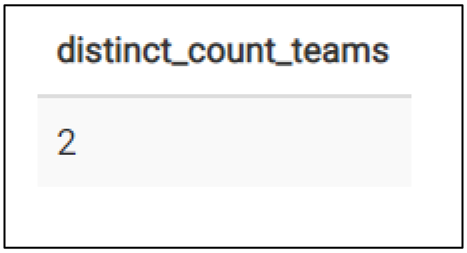
mie_mtrivedi_goals.venue_time_zone_id	mie_mtrivedi_goals.venue_time_zone_offset	mie_mtrivedi_goals.venue_time_zone_tz	mie_mtrivedi_goals.goal_summary
America/New_York	-4	EDT	Away Goals + Home Goals: 5
America/New_York	-4	EDT	Away Goals + Home Goals: 7
America/New_York	-4	EDT	Away Goals + Home Goals: 3
America/New_York	-4	EDT	Away Goals + Home Goals: 7
America/New_York	-4	EDT	Away Goals + Home Goals: 4
America/New_York	-4	EDT	Away Goals + Home Goals: 3
America/New_York	-4	EDT	Away Goals + Home Goals: 1
America/New_York	-4	EDT	Away Goals + Home Goals: 7
America/New_York	-4	EDT	Away Goals + Home Goals: 7
America/New_York	-4	EDT	Away Goals + Home Goals: 1
America/New_York	-4	EDT	Away Goals + Home Goals: 4

Question 12: Use subquery and count unique team names of teams that played as away team at TD Garden center and scored > 6. Provide the code and the answer.

Hive Query

```
-- Use of subquery and counting unique team names of teams that played as away teams at TD Garden center and scored >6

select count(distinct(team_id)) as Distinct_count_teams from team_info

where team_id in (select away_team_id from game

                                        where away_goals>6 and venue='TD Garden')

-- End of the query
```

Result



distinct_count_teams
2