Ex no: 6 Import a JSON file from the command line and apply actions with the data present in the JSON file

Aim: To create tables in Hive and write queries to access the data in the table.

Procedure:

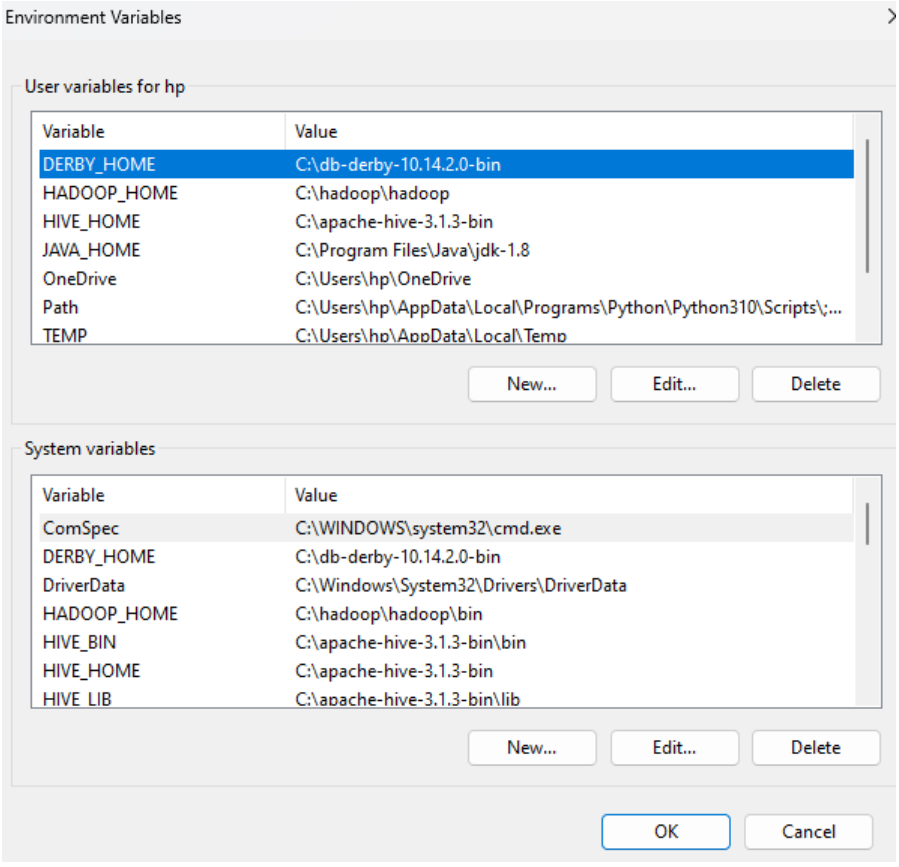
Hive Download and installation:

1. Hive Installation setup: - Download and install Apache Derby version 10.14.2.0: https://db.apache.org/derby/derby_downloads.html#For+Java+8+and+Higher –

2. Download and install Apache Hive version 3.1.3:

https://downloads.apache.org/hive/hive-3.1.3/ 2.

Add environment variables: Environment variables > System variables > Add the below paths -> (Inside Path)



PATH

3. Copy Derby libraries: Go to the Derby libraries directory (db-derby-10.14.2.0\lib) and copy all *.jar files. Then, paste them within the Hive libraries directory.

4. Configuring hive-site.xml and Hive's Bin folder: Refer following link to download the file. Also download the guava file. Put hive-site.xml file to hive's conf location and replace hive's current guava

file with this one in lib location. Also download the bin folder from link and replace the existing hive's bin folder. https://1drv.ms/f/s!ArSg3Xpur4Grmw0SDqW0g44T7HYU?e=wDsoB

5. Start the hadoop services.

start-all.cmd

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>java -version
java version "1.8.0_421"
Java(TM) SE Runtime Environment (build 1.8.0_421-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.421-b09, mixed mode)

C:\Windows\System32>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\Windows\System32>jps
15904 DataNode
3296 Jps
4976 NodeManager
8532 ResourceManager
15560 NameNode
12540 NetworkServerControl

C:\Windows\System32>hive --service schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
2024-09-17 14:06:03,908 INFO conf.HiveConf: Found configuration file file:/C:/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-09-17 14:06:04,720 INFO tools.HiveSchemaHelper: Metastore connection URL:   jdbc:derby://localhost:1527/metastore_db;create=true
Metastore connection URL:        jdbc:derby://localhost:1527/metastore_db;create=true
2024-09-17 14:06:04,721 INFO tools.HiveSchemaHelper: Metastore Connection Driver :       org.apache.derby.jdbc.ClientDriver
Metastore Connection Driver :    org.apache.derby.jdbc.ClientDriver
2024-09-17 14:06:04,722 INFO tools.HiveSchemaHelper: Metastore connection User:  APP
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

6. Start the derby host in 0.0.0.0

StartNetworkServer –h 0.0.0.0

```
C:\Windows\System32>StartNetworkServer -h 0.0.0.0
Tue Sep 17 14:04:03 IST 2024 : Security manager installed using the Basic server security policy.
Tue Sep 17 14:04:03 IST 2024 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connectio
ns on port 1527
```

```
Initialization script completed
schemaTool completed

C:\Windows\System32>
```

7. Start the hive services:

hive --service schematool -dbType derby –initSchema

```
C:\Windows\System32>hive --service schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
2024-09-17 14:06:03,908 INFO conf.HiveConf: Found configuration file file:/C:/apache-hive-3.1.3-bin/conf/hive-site.xml
2024-09-17 14:06:04,720 INFO tools.HiveSchemaHelper: Metastore connection URL:   jdbc:derby://localhost:1527/metastore_db;create=true
Metastore connection URL:        jdbc:derby://localhost:1527/metastore_db;create=true
2024-09-17 14:06:04,721 INFO tools.HiveSchemaHelper: Metastore Connection Driver :       org.apache.derby.jdbc.ClientDriver
Metastore Connection Driver :    org.apache.derby.jdbc.ClientDriver
2024-09-17 14:06:04,722 INFO tools.HiveSchemaHelper: Metastore connection User:  APP
Metastore connection User:       APP
Starting metastore schema initialization to 3.1.0
Initialization script hive-schema-3.1.0.derby.sql
```

8. To execute the SQL query open hive shell

hive

```
C:\Windows\System32>cd C:\apache-hive-3.1.3-bin\bin

C:\apache-hive-3.1.3-bin\bin>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hadoop/hadoop/share/hadoop/common/lib/slf4j-reload4j-1.7.36.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/apache-hive-3.1.3-bin/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Reload4jLoggerFactory]
2024-09-17 14:08:05,973 INFO conf.HiveConf: Found configuration file file:/C:/apache-hive-3.1.3-bin/conf/hive-site.xml
Hive Session ID = 3c26b629-b286-465a-a0fc-03f0264456e5
2024-09-17 14:08:09,202 INFO SessionState: Hive Session ID = 3c26b629-b286-465a-a0fc-03f0264456e5
```

Create a database:

CREATE DATABASE empl;

```
hive> CREATE DATABASE empl;
2024-09-17 15:34:13,478 INFO con
2024-09-17 15:34:13,479 INFO ses
2024-09-17 15:34:13,487 INFO ql.
2024-09-17 15:34:13,525 INFO ql.
2024-09-17 15:34:13,528 INFO ql.
2024-09-17 15:34:13,528 INFO ql.
```

SHOW DATABASES;

```
hive> SHOW DATABASES;
2024-09-17 15:34:24,844 INFO conf.HiveConf: Usi
2024-09-17 15:34:24,845 INFO session.SessionSta
2024-09-17 15:34:24,857 INFO ql.Driver: Compili
2024-09-17 15:34:24,894 INFO ql.Driver: Concurr
2024-09-17 15:34:24,896 INFO ql.Driver: Semanti
2024-09-17 15:34:24,897 INFO ql.Driver: Returni
:null)
2024-09-17 15:34:24,901 INFO exec.ListSinkOpera
2024-09-17 15:34:24,901 INFO ql.Driver: Complet
2024-09-17 15:34:24,902 INFO reexec.ReExecDrive
2024-09-17 15:34:24,902 INFO ql.Driver: Concurr
2024-09-17 15:34:24,902 INFO ql.Driver: Executi
2024-09-17 15:34:24,905 INFO ql.Driver: Startin
2024-09-17 15:34:24,908 INFO metastore.HiveMeta
2024-09-17 15:34:24,910 INFO HiveMetaStore.audi
2024-09-17 15:34:24,919 INFO exec.DDLTask: resu
2024-09-17 15:34:24,929 INFO ql.Driver: Complet
OK
2024-09-17 15:34:24,930 INFO ql.Driver: OK
2024-09-17 15:34:24,935 INFO ql.Driver: Concurr
2024-09-17 15:34:24,946 INFO mapred.FileInputFo
2024-09-17 15:34:25,016 INFO exec.ListSinkOpera
default
empl
mydb
Time taken: 0.082 seconds, Fetched: 3 row(s)
```

```
hive> USE empl;
2024-09-17 15:34:30,35
2024-09-17 15:34:30,36
2024-09-17 15:34:30,36
2024-09-17 15:34:30,39
2024-09-17 15:34:30,39
2024-09-17 15:34:30,39
```

CREATE TABLE:

CREATE TABLE employees_table (id INT, name STRING, age INT, salary DOUBLE)

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

STORED AS TEXTFILE

LOCATION '/hivee';

```
hive> CREATE TABLE employee(id INT,name STRING,age INT,salary DOUBLE) ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe' STORED AS TEXTFILE LOCATION '/hivee/emp
json.json/';
2024-09-17 15:38:39,678 INFO conf.HiveConf: Using the default value passed in for log id: 3c26b629-b286-465a-a0fc-03f0264456e5
2024-09-17 15:38:39,678 INFO session.SessionState: Updating thread name to 3c26b629-b286-465a-a0fc-03f0264456e5 main
```

Create a json file to load the data in table.

empjson.json

C: > text > empjson.json

```
{"id": 1, "name": "Michael Scott", "age": 45, "salary": 85000}
{"id": 2, "name": "Pam Beesly", "age": 26, "salary": 62000}
{"id": 3, "name": "Jim Halpert", "age": 30, "salary": 68000}
{"id": 4, "name": "Dwight Schrute", "age": 35, "salary": 75000}
{"id": 5, "name": "Stanley Hudson", "age": 50, "salary": 90000}
{"id": 6, "name": "Ryan Howard", "age": 24, "salary": 50000}
{"id": 7, "name": "Angela Martin", "age": 33, "salary": 67000}
{"id": 8, "name": "Kevin Malone", "age": 36, "salary": 58000}
{"id": 9, "name": "Oscar Martinez", "age": 31, "salary": 64000}
{"id": 10, "name": "Toby Flenderson", "age": 40, "salary": 72000}
```

Load the data :

```
hive> LOAD DATA INPATH '/hivee/empjson.json' INTO TABLE employee;
2024-09-17 15:41:19,552 INFO conf.HiveConf: Using the default value
2024-09-17 15:41:19,553 INFO session.SessionState: Updating thread n
2024-09-17 15:41:19,560 INFO ql.Driver: Compiling command(queryId=hp
```

Put the json file inside the hadoop directory:

```
C:\Windows\System32>hdfs dfs -put C:\text\empjson.json /hivee

C:\Windows\System32>hdfs dfs -cat /hivee/empjson.json
{"id": 1, "name": "Michael Scott", "age": 45, "salary": 85000}
{"id": 2, "name": "Pam Beesly", "age": 26, "salary": 62000}
{"id": 3, "name": "Jim Halpert", "age": 30, "salary": 68000}
{"id": 4, "name": "Dwight Schrute", "age": 35, "salary": 75000}
{"id": 5, "name": "Stanley Hudson", "age": 50, "salary": 90000}
{"id": 6, "name": "Ryan Howard", "age": 24, "salary": 50000}
{"id": 7, "name": "Angela Martin", "age": 33, "salary": 67000}
{"id": 8, "name": "Kevin Malone", "age": 36, "salary": 58000}
{"id": 9, "name": "Oscar Martinez", "age": 31, "salary": 64000}
{"id": 10, "name": "Toby Flenderson", "age": 40, "salary": 72000}
```

DESC employee; -Display the structure of the table.

```
hive> DESC employee;
2024-09-17 15:41:42,857 INFO c
2024-09-17 15:41:42,858 INFO s
2024-09-17 15:41:42,864 INFO q
2024-09-17 15:41:42,898 INFO q
2024-09-17 15:41:42,900 INFO m
2024-09-17 15:41:42,901 INFO H
```

```
id                      int                     from deserializer
name                    string                  from deserializer
age                     int                     from deserializer
salary                  double                  from deserializer
Time taken: 0.174 seconds, Fetched: 4 row(s)
```

Performing various operations on the table:

Query:

```
hive> SELECT * FROM employee;
2024-09-17 15:41:58,221 INFO conf
2024-09-17 15:41:58,223 INFO sess
2024-09-17 15:41:58,227 INFO ql.D
2024-09-17 15:41:58,262 INFO ql.D
```

Output:

```
1       Michael Scott   45      85000.0
2       Pam Beesly      26      62000.0
3       Jim Halpert     30      68000.0
4       Dwight Schrute  35      75000.0
5       Stanley Hudson  50      90000.0
6       Ryan Howard     24      50000.0
7       Angela Martin   33      67000.0
8       Kevin Malone    36      58000.0
9       Oscar Martinez  31      64000.0
10      Toby Flenderson 40      72000.0
Time taken: 0.517 seconds, Fetched: 10 row(s)
```

Query:

```
hive> SELECT id,name FROM employee where age=45;
2024-09-17 15:42:40,142 INFO conf.HiveConf: Using
2024-09-17 15:42:40,143 INFO session.SessionState
2024-09-17 15:42:40,145 INFO ql.Driver: Compiling
```

Output:

```
2024-09-17 15:42:42,316 INFO
1       Michael Scott
Time taken: 2.13 seconds, Fet
2024-09-17 15:42:42,321 INFO
```

Query:

```
hive> SELECT SUM(salary) as TOTAL FROM employee;
2024-09-17 15:43:26,453 INFO conf.HiveConf: Using t
2024-09-17 15:43:26,453 INFO session.SessionState:
```

Output:

```
691000.0
Time taken: 8.
```

Query:

```
hive> SELECT * FROM employee LIMIT 2;
2024-09-17 15:45:12,610 INFO conf.HiveCon
2024-09-17 15:45:12,610 INFO session.Sess
```

Output:

```
1       Michael Scott   45      85000.0
2       Pam Beesly      26      62000.0
2024-09-17 15:45:12,920 INFO exec.TableSca
```

Query:

```
hive> SELECT * FROM employee ORDER BY salary;
2024-09-17 15:45:40,409 INFO conf.HiveConf: Usin
```

Output:

```
6       Ryan Howard     24      50000.0
8       Kevin Malone    36      58000.0
2       Pam Beesly      26      62000.0
9       Oscar Martinez  31      64000.0
7       Angela Martin   33      67000.0
3       Jim Halpert     30      68000.0
10      Toby Flenderson 40      72000.0
4       Dwight Schrute  35      75000.0
1       Michael Scott   45      85000.0
5       Stanley Hudson  50      90000.0
Time taken: 2.059 seconds, Fetched: 10 row(s)
```

Query:

```
hive> SELECT COUNT(*) FROM employee;
```

Output:

```
2024-09-17 16:03:51,822 INFO mapred.FileInputFor
2024-09-17 16:03:51,882 INFO exec.ListSinkOperat
10
Time taken: 3.809 seconds, Fetched: 1 row(s)
```

Result: Thus, to import a JSON file from the command line and apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort was completed successfully.