Ex no: 6 **210701004**

Import a JSON file from the command line and apply actions with the data present in the JSON file

Aim:

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.

Procedure:

Hive Download and installation:

1. Starting Hadoop Services

Open PowerShell as administrator and go to Hadoop sbin directory and start hadoop services using the following commands: Start-all.cmd

2.Create a .json file with the below content:

{"id": 1, "name": "John Doe", "age": 30, "salary": 50000}
{"id": 2, "name": "Jane Smith", "age": 25, "salary": 60000}
{"id": 3, "name": "Alice Johnson", "age": 28, "salary": 55000}
{"id": 4, "name": "Bob Brown", "age": 35, "salary": 70000}
{"id": 5, "name": "Charlie Davis", "age": 40, "salary": 80000}
{"id": 6, "name": "Eve White", "age": 22, "salary": 48000}
{"id": 7, "name": "Frank Black", "age": 32, "salary": 65000}
{"id": 8, "name": "Grace Green", "age": 27, "salary": 52000}
{"id": 9, "name": "Henry Gold", "age": 29, "salary": 59000}
{"id": 10, "name": "Isabel Blue", "age": 33, "salary": 73000}

Derby Network Server:

Run the following command to open Derby:

StartNetworkServer -h 0.0.0.0

```
C:\Windows\System32>startNetworkServer -h 0.0.0.0
Fri Sep 13 19:17:50 IST 2024 : Security manager installed using the Basic server security policy.
Fri Sep 13 19:17:50 IST 2024 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connectio
ns on port 1527
```

Go to first PowerShell window and check whether NetworkServerControl is running.

```
C:\Windows\System32>jps
2976 DataNode
9392 NodeManager
11640 NetworkServerControl
11116 NameNode
23452 Jps
2668 ResourceManager

C:\Windows\System32>
```

3. Starting Apache Hive:

Go to Apache Hive's bin location with cd command and run the following command:

```
hive --service schematool -dbType derby –initSchema
```

8. Open Hive shell by typing:

```
hive
```

```
C:\hive\bin>hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLogg
erBinder.class]
SLF4J: Found binding in [jar:file:/C:/hive/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.Log4jLoggerFactory]
2024-09-13 19:33:54,247 INFO conf.HiveConf: Found configuration file null
2024-09-13 19:33:55,244 WARN common.LogUtils: hive-site.xml not found on CLASSPATH
```

Create a Database:

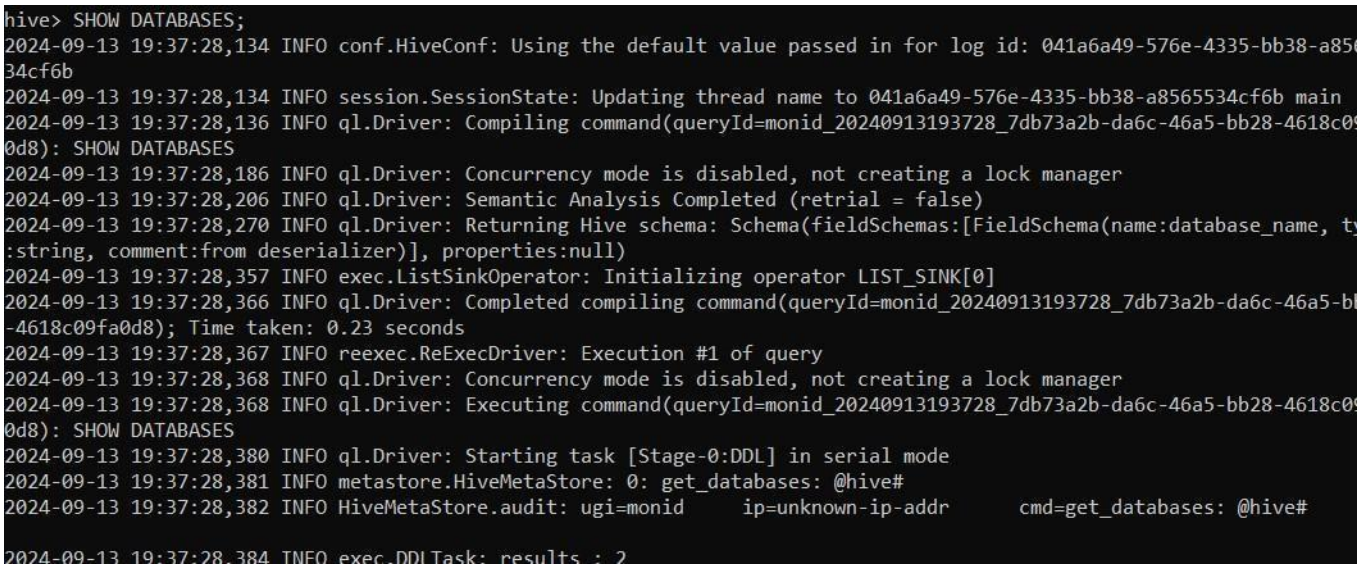
Start by creating a database. Open the Hive CLI and follow the steps below:

1. Use the **CREATE DATABASE** statement to create a new database:

```
CREATE DATABASE IF NOT EXISTS emp_json;
```

2. Verify the database is present:

```
SHOW DATABASES;
```



Create a Table in Hive:

```
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 '/ser/hive/warehouse/emp_json/';
```

```
2024-09-13 18:41:24,255 INFO session.SessionState: Resetting thread name to  main
hive> 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 '/user/hive/warehouse/emp_json/';
2024-09-13 18:42:28,011 INFO conf.HiveConf: Using the default value passed in for log id: 2f3d9a62-b716-4df7-be0c-fc9f57
7d9ed1
2024-09-13 18:42:28,011 INFO session.SessionState: Updating thread name to 2f3d9a62-b716-4df7-be0c-fc9f577d9ed1 main
2024-09-13 18:42:28,013 INFO ql.Driver: Compiling command(queryId=monid_20240913184228_e23cbc73-eaa9-417c-b9f0-f206513f3
b35): 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 '/user/hive/warehouse/emp_json/'
```

Add Data to the TABLE:

Run the **LOAD DATA INPATH** command:

```
LOAD DATA INPATH '/user/hive/warehouse/emp_json/employee.json' INTO TABLE employees_table;
```

```
hive> LOAD DATA INPATH '/user/hive/warehouse/employee.json' INTO TABLE employees_table;
2024-09-13 18:43:03,505 INFO conf.HiveConf: Using the default value passed in for log id: 2f3d9a62-b716-4df7-be0c-fc9f57
7d9ed1
2024-09-13 18:43:03,505 INFO session.SessionState: Updating thread name to 2f3d9a62-b716-4df7-be0c-fc9f577d9ed1 main
2024-09-13 18:43:03,508 INFO ql.Driver: Compiling command(queryId=monid_20240913184303_e143d92f-b894-4b56-855c-d687e518f
311): LOAD DATA INPATH '/user/hive/warehouse/employee.json' INTO TABLE employees_table
2024-09-13 18:43:03,519 INFO metastore.HiveMetaStoreClient: Mestastore configuration metastore.filter.hook changed from
org.apache.hadoop.hive.metastore.DefaultMetaStoreFilterHookImpl to org.apache.hadoop.hive.ql.security.authorization.plug
in.AuthorizationMetaStoreFilterHook
2024-09-13 18:43:03,520 INFO metastore.HiveMetaStore: 0: Cleaning up thread local RawStore...
2024-09-13 18:43:03,523 INFO HiveMetaStore.audit: ugi=monid     ip=unknown-ip-addr      cmd=Cleaning up thread local Raw
```

List Hive Tables and Data:

To show all tables in a selected database, use the following statement:

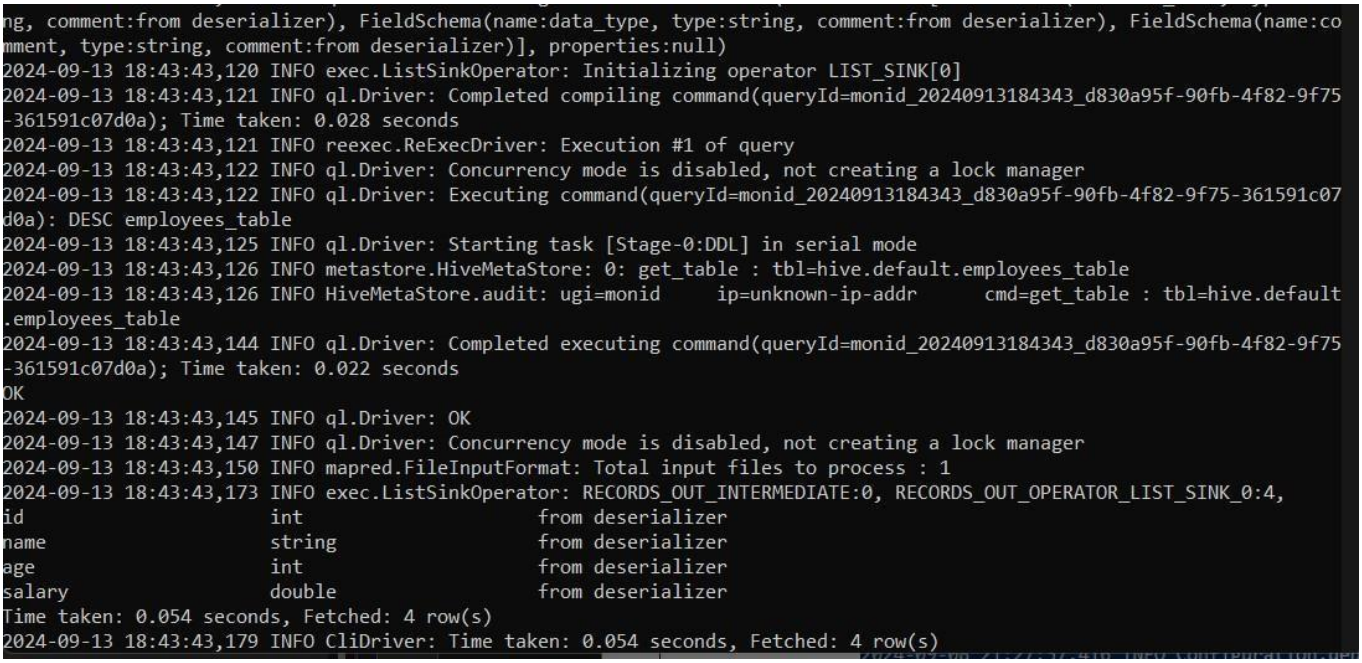
```
SHOW TABLES;
```

```
hive> SHOW TABLES;
2024-09-13 18:43:19,246 INFO conf.HiveConf: Using the default value passed in for log id: 2f3d9a62-b716-4df7-be0c-fc9f57
7d9ed1
2024-09-13 18:43:19,246 INFO session.SessionState: Updating thread name to 2f3d9a62-b716-4df7-be0c-fc9f577d9ed1 main
2024-09-13 18:43:19,249 INFO ql.Driver: Compiling command(queryId=monid_20240913184319_786f1cd8-47e7-4cd5-bc39-48276b96a
9fc): SHOW TABLES
2024-09-13 18:43:19,260 INFO ql.Driver: Concurrency mode is disabled, not creating a lock manager
2024-09-13 18:43:19,262 INFO metastore.HiveMetaStore: 0: get_database: @hive#default
2024-09-13 18:43:19,262 INFO HiveMetaStore.audit: ugi=monid     ip=unknown-ip-addr      cmd=get_database: @hive#default

2024-09-13 18:43:19,264 INFO ql.Driver: Semantic Analysis Completed (retrial = false)
2024-09-13 18:43:19,264 INFO ql.Driver: Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:stri
ng, comment:from deserializer)], properties:null)
2024-09-13 18:43:19,265 INFO exec.ListSinkOperator: Initializing operator LIST_SINK[0]
```

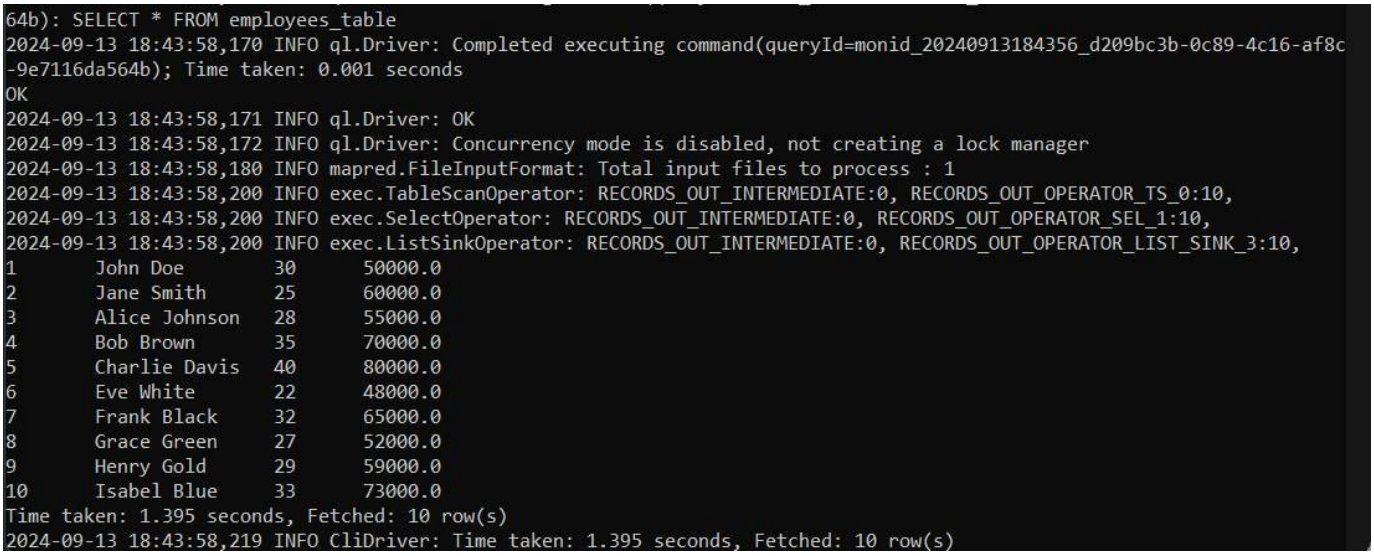
To show table column names and data types, run:

```
DESC employees_table;
```



To display table data, use a **SELECT** statement. For example, to select everything in a table, run:

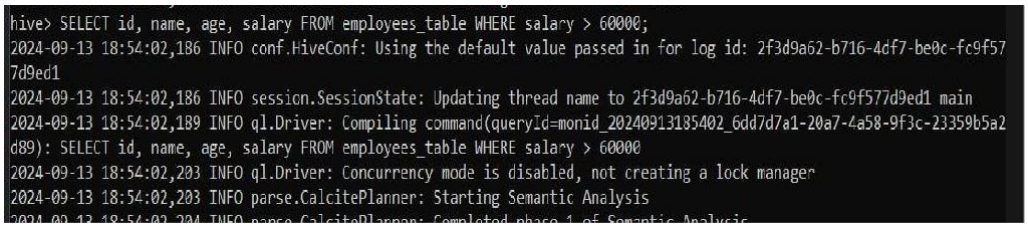
```
SELECT * FROM employees_table;
```

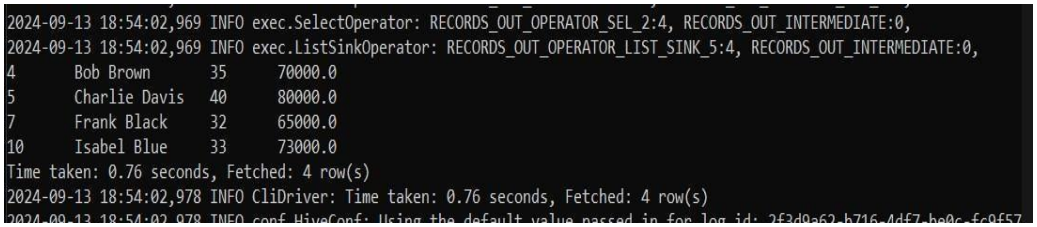


Perform Various Operations on the Data in the table:

WHERE:

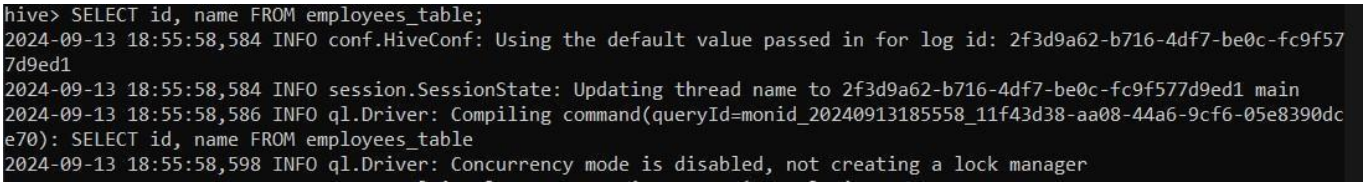
SELECT id, name, age, salary FROM employees_table
WHERE salary > 60000;

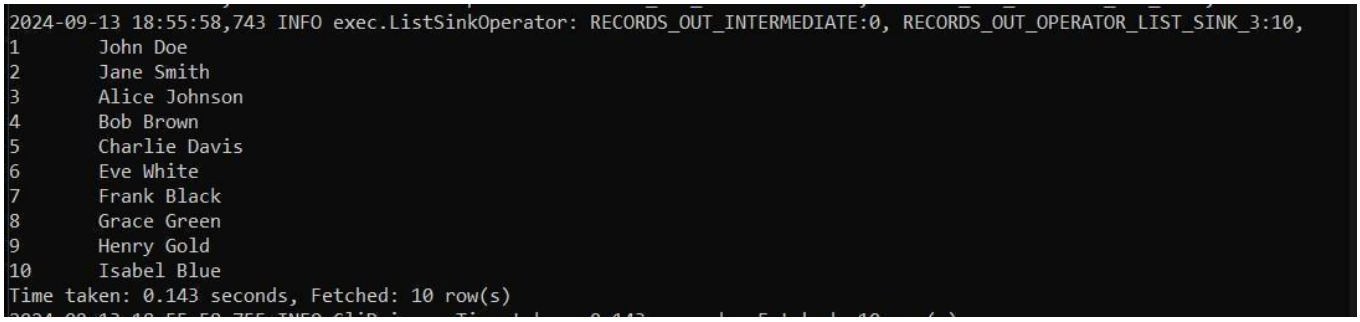




PROJECTION: (Selecting Specific Columns)

SELECT id, name FROM employees_table;





AGGREGATION: (e.g., Summing Salaries by Age Group)

SELECT age, MAX(salary) AS max_salary FROM employees_table GROUP BY age;

```
22    48000.0
25    60000.0
27    52000.0
28    55000.0
29    59000.0
30    50000.0
32
33    73000.0
35    70000.0
40    80000.0
```

REMOVE: (Remove Specific Records)

SELECT * FROM employees_table WHERE salary > 70000;

```
2024-09-13 19:04:58,754 INFO exec.ListSinkOperator: RECORDS_OUT_OPERATOR_LIST_SINK_5:2, RECORDS_O
5       Charlie Davis   40      80000.0
10      Isabel Blue     33      73000.0
Time taken: 0.199 seconds, Fetched: 2 row(s)
```

COUNT: (Counting the Number of Records)

SELECT COUNT(*) FROM employees_table;

```
10
Time taken: 55.015 seconds ,Fetched : 1 row(s)
```

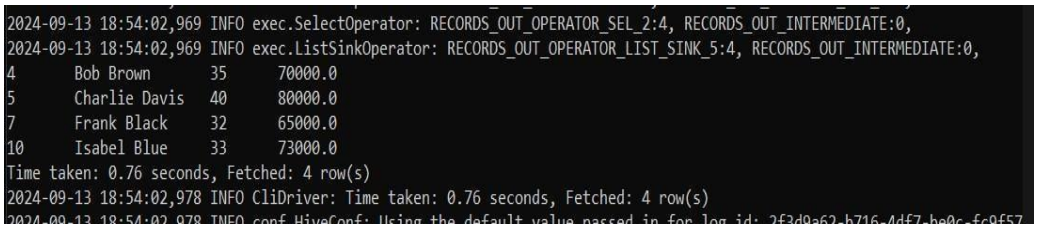
LIMIT: (Restrict the Number of Rows Returned)

SELECT * FROM employees_table LIMIT 5;

```
2024-09-13 19:09:34,014 INFO mapred.FileInputFormat: Total input files to process : 1
1       John Doe        30      50000.0
2       Jane Smith      25      60000.0
3       Alice Johnson   28      55000.0
4       Bob Brown       35      70000.0
5       Charlie Davis   40      80000.0
2024-09-13 19:09:34,031 INFO exec.TableScanOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_TS_0:5,
2024-09-13 19:09:34,032 INFO exec.SelectOperator: RECORDS_OUT_INTERMEDIATE:0, RECORDS_OUT_OPERATOR_SEL_1:5,
```

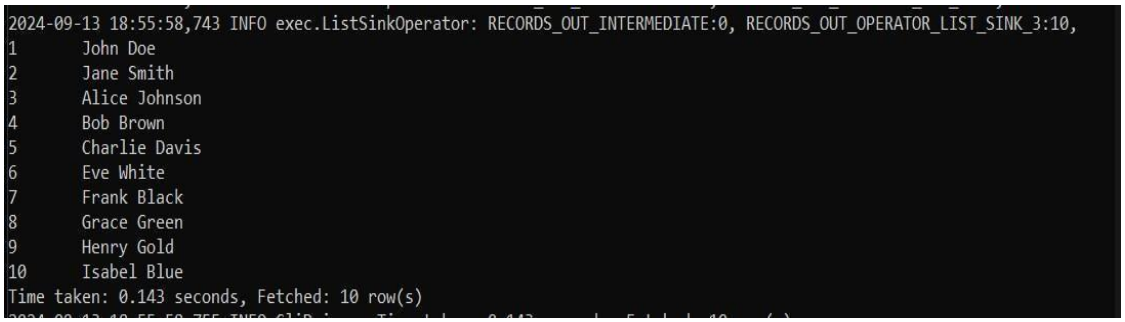
SKIP: (Skipping the First N Rows, using Row Number)

SELECT * FROM (SELECT *, ROW_NUMBER() OVER () AS row_num FROM employees_table) temp WHERE row_num > 3;



SORT: (Order the Data by Salary)

```
SELECT * FROM employees_table ORDER BY salary DESC;
```



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.