

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

**Aim:**

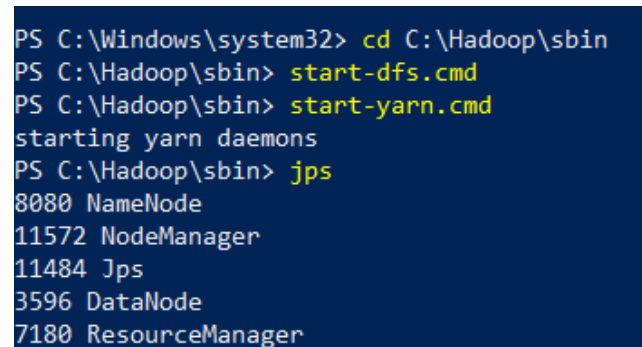
To import a JSON file from the command line and apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort.

**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-dfs.cmd
```

```
start-yarn.cmd
```



```
PS C:\Windows\system32> cd C:\Hadoop\sbin
PS C:\Hadoop\sbin> start-dfs.cmd
PS C:\Hadoop\sbin> start-yarn.cmd
starting yarn daemons
PS C:\Hadoop\sbin> jps
8080 NameNode
11572 NodeManager
11484 Jps
3596 DataNode
7180 ResourceManager
```

**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 }
```

### 3. Open a new PowerShell window and add the json file to Hadoop using –put command:

```
PS C:\Windows\system32> hdfs dfs -put /C:/Users/Admin/employee.json /user/hive/warehouse/emp_json/
PS C:\Windows\system32> hdfs dfs -ls /user/hive/warehouse/emp_json/
Found 1 items
-rw-r--r--  1 Admin supergroup          657 2024-09-08 20:22 /user/hive/warehouse/emp_json/employee.json
```

```
PS C:\Windows\system32> hdfs dfs -cat /user/hive/warehouse/emp_json/employee.json
{"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}
PS C:\Windows\system32>
```

Derby Network Server:

Run the following command to open Derby:

```
StartNetworkServer -h 0.0.0.0
```

```
PS C:\Windows\system32> StartNetworkServer -h 0.0.0.0
Sat Aug 31 20:11:02 IST 2024 : Security manager installed using the Basic server security policy.
Sat Aug 31 20:11:07 IST 2024 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connections on port 1527
```

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

```
PS C:\Hadoop\sbin> jps
12480 NetworkServerControl
8080 NameNode
11572 NodeManager
12180 Jps
3596 DataNode
7180 ResourceManager
```

### 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
```

```

PS C:\Hadoop\sbin> cd C:\apache-hive-3.1.3-bin\bin
PS C:\apache-hive-3.1.3-bin\bin> hive --service schematool -dbType derby -initSchema
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/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]
2024-08-31 20:12:45,641 INFO conf.HiveConf: Found configuration file null
2024-08-31 20:12:46,492 INFO tools.HiveSchemaHelper: Metastore connection URL: jdbc:derby;;databaseName=metastore_db;create=true
Metastore connection URL: jdbc:derby;;databaseName=metastore_db;create=true
2024-08-31 20:12:46,494 INFO tools.HiveSchemaHelper: Metastore Connection Driver : org.apache.derby.jdbc.EmbeddedDriver
Metastore Connection Driver : org.apache.derby.jdbc.EmbeddedDriver
2024-08-31 20:12:46,495 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

```

```

Initialization script completed
schemaTool completed

```

8. Open Hive shell by typing:

```
hive
```

```

PS C:\apache-hive-3.1.3-bin\bin> hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/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]
2024-08-31 20:13:15,204 INFO conf.HiveConf: Found configuration file null
2024-08-31 20:13:18,554 WARN common.LogUtils: hive-site.xml not found on CLASSPATH
Hive Session ID = 272282ae-ff6f-4567-bab6-f339170eaaea
2024-08-31 20:13:18,670 INFO SessionState: Hive Session ID = 272282ae-ff6f-4567-bab6-f339170eaaea

```

## 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;
```

```

hive> CREATE DATABASE IF NOT EXISTS emp_json;
2024-09-08 20:05:17,842 INFO conf.HiveConf: Using the default value passed in for lc
2024-09-08 20:05:18,057 INFO ql.Driver: Compiling command(queryId=Admin_202409082005
2024-09-08 20:05:19,093 INFO ql.Driver: Concurrency mode is disabled, not creating a
2024-09-08 20:05:19,129 INFO ql.Driver: Semantic Analysis Completed (retrial = false
2024-09-08 20:05:19,139 INFO ql.Driver: Returning Hive schema: Schema(fieldSchemas:n
2024-09-08 20:05:19,155 INFO ql.Driver: Completed compiling command(queryId=Admin_20
2024-09-08 20:05:19,155 INFO reexec.ReExecDriver: Execution #1 of query
2024-09-08 20:05:19,157 INFO ql.Driver: Concurrency mode is disabled, not creating a
2024-09-08 20:05:19,157 INFO ql.Driver: Executing command(queryId=Admin_202409082005
2024-09-08 20:05:19,185 INFO ql.Driver: Starting task [Stage-0:DDL] in serial mode
2024-09-08 20:05:19,260 INFO sqlstd.SQLStdHiveAccessController: Created SQLStdHiveAc
2024-09-08 20:05:19,266 WARN session.SessionState: METASTORE_FILTER_HOOK will be ign
2024-09-08 20:05:19,267 INFO metastore.HiveMetaStoreClient: Mestastore configuration

```

2. Verify the database is present:

```
SHOW DATABASES;
```

```

hive> SHOW DATABASES;
2024-09-08 20:05:27,600 INFO conf.HiveConf: Using the default
2024-09-08 20:05:27,600 INFO session.SessionState: Updating
2024-09-08 20:05:27,603 INFO ql.Driver: Compiling command(q
2024-09-08 20:05:27,624 INFO ql.Driver: Concurrency mode is
2024-09-08 20:05:27,650 INFO ql.Driver: Semantic Analysis C
2024-09-08 20:05:27,774 INFO ql.Driver: Returning Hive sche
2024-09-08 20:05:27,937 INFO exec.ListSinkOperator: Initial
2024-09-08 20:05:27,951 INFO ql.Driver: Completed compiling
2024-09-08 20:05:27,951 INFO reexec.ReExecDriver: Execution
2024-09-08 20:05:27,952 INFO ql.Driver: Concurrency mode is
2024-09-08 20:05:27,952 INFO ql.Driver: Executing command(q
2024-09-08 20:05:27,953 INFO ql.Driver: Starting task [Stag
2024-09-08 20:05:27,954 INFO metastore.HiveMetaStore: 0: ge
2024-09-08 20:05:27,954 INFO HiveMetaStore.audit: ugi=Admini

2024-09-08 20:05:27,965 INFO exec.DDLTask: results : 2
2024-09-08 20:05:28,073 INFO ql.Driver: Completed executing
OK
2024-09-08 20:05:28,075 INFO ql.Driver: OK
2024-09-08 20:05:28,076 INFO ql.Driver: Concurrency mode is
2024-09-08 20:05:28,105 INFO Configuration.deprecation: map
2024-09-08 20:05:28,191 INFO mapred.FileInputFormat: Total
2024-09-08 20:05:28,319 INFO exec.ListSinkOperator: RECORDS
default
emp_json
Time taken: 0.478 seconds, Fetched: 2 row(s)

```

3. Switch to the new database:

```
USE emp_json;
```

```

hive> USE emp_json;
2024-09-08 20:05:36,394 INFO conf.HiveConf: Using the default value passed in for log id: 6c6
2024-09-08 20:05:36,394 INFO session.SessionState: Updating thread name to 6c64d964-49eb-4a09
2024-09-08 20:05:36,398 INFO ql.Driver: Compiling command(queryId=Admin_20240908200536_2f966b
2024-09-08 20:05:36,595 INFO ql.Driver: Concurrency mode is disabled, not creating a lock man

```

**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 '/user/hive/warehouse/emp_json/';

```

```
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/';
```

### 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/emp_json/employee.json' INTO TABLE employees_table;
2024-09-08 20:24:19,751 INFO conf.HiveConf: Using the default value passed in for log id: 6c64d964-49eb-4a09-803d-ba07
2024-09-08 20:24:19,751 INFO session.SessionState: Updating thread name to 6c64d964-49eb-4a09-803d-ba07eeb9ef77 main
2024-09-08 20:24:19,772 INFO ql.Driver: Compiling command(queryId=Admin_20240908202419_7c2853c4-b3da-4d84-8129-3d3d471
2024-09-08 20:24:19,840 INFO ql.Driver: Concurrency mode is disabled, not creating a lock manager
2024-09-08 20:24:19,840 INFO metastore.HiveMetaStore: 0: get_table : tbl=hive.emp_json.employees_table
```

### List Hive Tables and Data:

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

```
SHOW TABLES;
```

```
hive> SHOW TABLES;
2024-09-08 20:15:44,793 INFO conf.HiveConf: Using
2024-09-08 20:15:44,794 INFO session.SessionState
2024-09-08 20:15:44,811 INFO ql.Driver: Compiling
2024-09-08 20:15:44,961 INFO ql.Driver: Concurr
2024-09-08 20:15:45,226 INFO metastore.HiveMetaSt
2024-09-08 20:15:45,227 INFO HiveMetaStore.audit:
2024-09-08 20:15:45,285 INFO ql.Driver: Semantic
2024-09-08 20:15:45,441 INFO ql.Driver: Returnin
2024-09-08 20:15:45,474 INFO exec.ListSinkOperato
2024-09-08 20:15:45,512 INFO ql.Driver: Completed
2024-09-08 20:15:45,513 INFO reexec.ReExecDriver:
2024-09-08 20:15:45,514 INFO ql.Driver: Concurr
2024-09-08 20:15:45,515 INFO ql.Driver: Executing
2024-09-08 20:15:45,517 INFO ql.Driver: Starting
2024-09-08 20:15:45,542 INFO metastore.HiveMetaSt
2024-09-08 20:15:45,543 INFO HiveMetaStore.audit:
2024-09-08 20:15:45,597 INFO metastore.HiveMetaSt
2024-09-08 20:15:45,597 INFO HiveMetaStore.audit:
2024-09-08 20:15:46,006 INFO ql.Driver: Completed
OK
2024-09-08 20:15:46,009 INFO ql.Driver: OK
2024-09-08 20:15:46,011 INFO ql.Driver: Concurr
2024-09-08 20:15:46,059 INFO mapred.FileInputForm
2024-09-08 20:15:46,166 INFO exec.ListSinkOperato
employees
employees2
employees_table
```

To show table column names and data types, run:

```
DESC employees_table;
```

```
hive> DESC employees_table;
2024-09-08 21:27:35,982 INFO conf.HiveConf: Using the default value
2024-09-08 21:27:35,984 INFO session.SessionState: Updating thread n
2024-09-08 21:27:35,986 INFO ql.Driver: Compiling command(queryId=Ad
2024-09-08 21:27:36,014 INFO ql.Driver: Concurrency mode is disabled
2024-09-08 21:27:36,017 INFO metastore.HiveMetaStore: 0: get_table :
2024-09-08 21:27:36,018 INFO HiveMetaStore.audit: ugi=Admin ip=u
2024-09-08 21:27:36,864 INFO parse.DDLSemanticAnalyzer: analyzeDescr
2024-09-08 21:27:36,865 INFO ql.Driver: Semantic Analysis Completed
2024-09-08 21:27:37,033 INFO ql.Driver: Returning Hive schema: Schem
:comment, type:string, comment:from deserializer)], properties:null)
2024-09-08 21:27:37,223 INFO exec.ListSinkOperator: Initializing ope
2024-09-08 21:27:37,244 INFO ql.Driver: Completed compiling command(
2024-09-08 21:27:37,244 INFO reexec.ReExecDriver: Execution #1 of qu
2024-09-08 21:27:37,244 INFO ql.Driver: Concurrency mode is disabled
2024-09-08 21:27:37,244 INFO ql.Driver: Executing command(queryId=Ad
2024-09-08 21:27:37,245 INFO ql.Driver: Starting task [Stage-0:DDL]
2024-09-08 21:27:37,245 INFO metastore.HiveMetaStore: 0: get_table :
2024-09-08 21:27:37,246 INFO HiveMetaStore.audit: ugi=Admin ip=u
2024-09-08 21:27:37,384 INFO ql.Driver: Completed executing command(
OK
2024-09-08 21:27:37,387 INFO ql.Driver: OK
2024-09-08 21:27:37,388 INFO ql.Driver: Concurrency mode is disabled
2024-09-08 21:27:37,416 INFO Configuration.deprecation: mapred.input
2024-09-08 21:27:37,505 INFO mapred.FileInputFormat: Total input fil
2024-09-08 21:27:37,607 INFO exec.ListSinkOperator: RECORDS_OUT_INTE
id          int          from deserializer
name        string        from deserializer
age          int          from deserializer
salary      double        from deserializer
Time taken: 1.402 seconds, Fetched: 4 row(s)
```

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

```
SELECT * FROM employees_table;
```

```
hive> SELECT * FROM employees_table;
2024-09-08 21:28:47,514 INFO conf.HiveConf: Using the default value passed in for log id:
2024-09-08 21:28:47,516 INFO session.SessionState: Updating thread name to e521c02a-db38-
2024-09-08 21:28:47,520 INFO ql.Driver: Compiling command(queryId=Admin_20240908212847_32
2024-09-08 21:28:47,602 INFO ql.Driver: Concurrency mode is disabled, not creating a lock
2024-09-08 21:28:47,614 INFO parse.CalcitePlanner: Starting Semantic Analysis
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
6      Eve White     22      48000.0
7      Frank Black   32      65000.0
8      Grace Green   27      52000.0
9      Henry Gold     29      59000.0
10     Isabel Blue   33      73000.0
Time taken: 5.181 seconds, Fetched: 10 row(s)
```



## Perform Various Operations on the Data in the table:

### WHERE:

```
SELECT id, name, age, salary
```

```
FROM employees_table
```

```
WHERE salary > 60000;
```

```
hive> SELECT id, name, age, salary
> FROM employees_table
> WHERE salary > 60000;
2024-09-08 21:51:20,336 INFO conf.HiveConf: Using the
2024-09-08 21:51:20,338 INFO session.SessionState: Upd
2024-09-08 21:51:20,340 INFO ql.Driver: Compiling comm
FROM employees_table
WHERE salary > 60000
```

```
4      Bob Brown      35      70000.0
5      Charlie Davis  40      80000.0
7      Frank Black    32      65000.0
10     Isabel Blue    33      73000.0
Time taken: 2.675 seconds, Fetched: 4 row(s)
```

### PROJECTION: (Selecting Specific Columns)

```
SELECT id, name FROM employees_table;
```

```
hive> select id,name from employees_table;
2024-09-08 20:29:49,171 INFO conf.HiveConf: Using the default value passed in for log id: 6c
2024-09-08 20:29:49,171 INFO session.SessionState: Updating thread name to 6c64d964-49eb-4a0
2024-09-08 20:29:49,191 INFO ql.Driver: Compiling command(queryId=Admin_20240908202949_42665
2024-09-08 20:29:49,254 INFO ql.Driver: Concurrency mode is disabled, not creating a lock ma
```

```
1      John Doe
2      Jane Smith
3      Alice Johnson
4      Bob Brown
5      Charlie Davis
6      Eve White
7      Frank Black
8      Grace Green
9      Henry Gold
10     Isabel Blue
Time taken: 0.537 seconds, Fetched: 10 row(s)
```

### 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      65000.0
33      73000.0
35      70000.0
40      80000.0
Time taken: 57.669 seconds, Fetched: 10 row(s)
```

### REMOVE: (Remove Specific Records)

```
SELECT *
```

```
FROM employees_table
```

```
WHERE salary > 70000;
```

```
5      Charlie Davis    40      80000.0
10     Isabel Blue     33      73000.0
Time taken: 0.404 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;
```

```
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
```



### 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;
```

```
7      Frank Black      32      65000.0 4  
6      Eve White       22      48000.0 5  
5      Charlie Davis   40      80000.0 6  
4      Bob Brown       35      70000.0 7  
3      Alice Johnson   28      55000.0 8  
2      Jane Smith      25      60000.0 9  
1      John Doe        30      50000.0 10  
Time taken: 62.45 seconds, Fetched: 7 row(s)
```

### SORT: (Order the Data by Salary)

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

```
5      Charlie Davis   40      80000.0  
10     Isabel Blue     33      73000.0  
4      Bob Brown       35      70000.0  
7      Frank Black     32      65000.0  
2      Jane Smith      25      60000.0  
9      Henry Gold      29      59000.0  
3      Alice Johnson   28      55000.0  
8      Grace Green     27      52000.0  
1      John Doe        30      50000.0  
6      Eve White       22      48000.0  
Time taken: 75.296 seconds, Fetched: 10 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.