**Ex no: 6 210701132**

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

**Aim:**

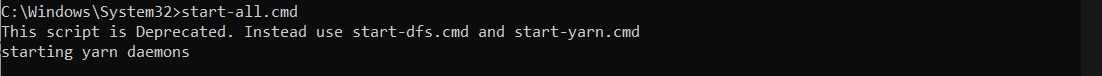
To import a JSON file from the command line and apply the following actions with the data present in the JSON file where, projection, aggregation, remove, count, limit, skip and sort.

**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}

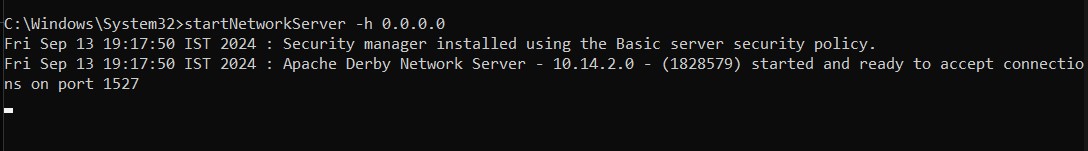
{"id": 9, "name": "Henry Gold", "age": 29, "salary": 59000}

{"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



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



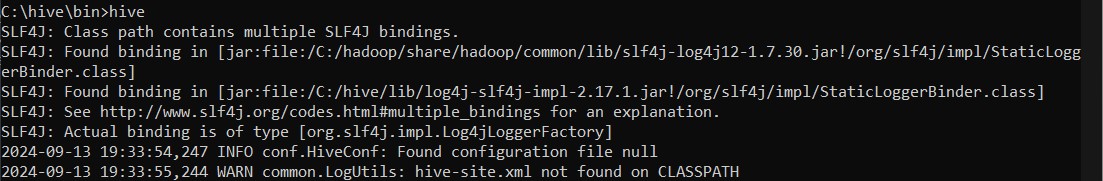
3. Starting Apache Hive:

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



**Create a Database:**

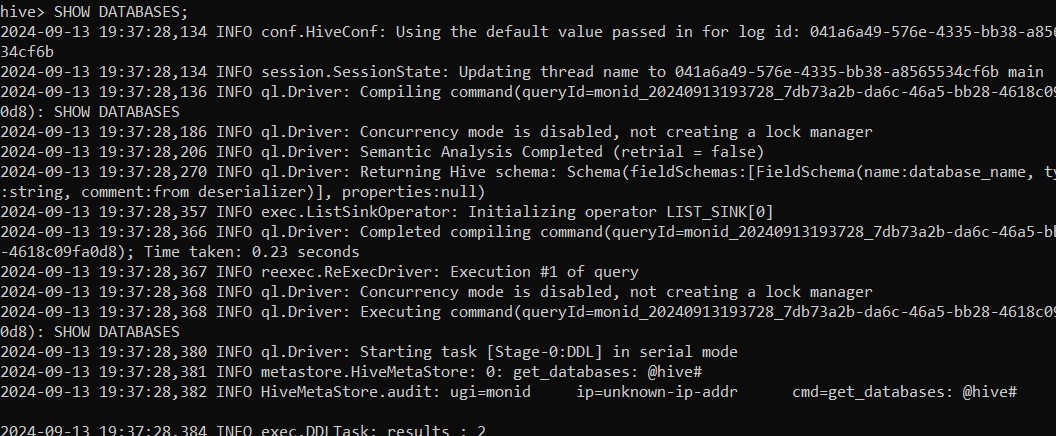
Start by creating a database. Open the Hive CLI and follow the steps below:

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

CREATE DATABASE IF NOT EXISTS emp\_json;

1. Verify the database is present:

# SHOW DATABASES;



**Create a Table in Hive:**

CREATE TABLE employees\_table (

id INT,

name STRING,

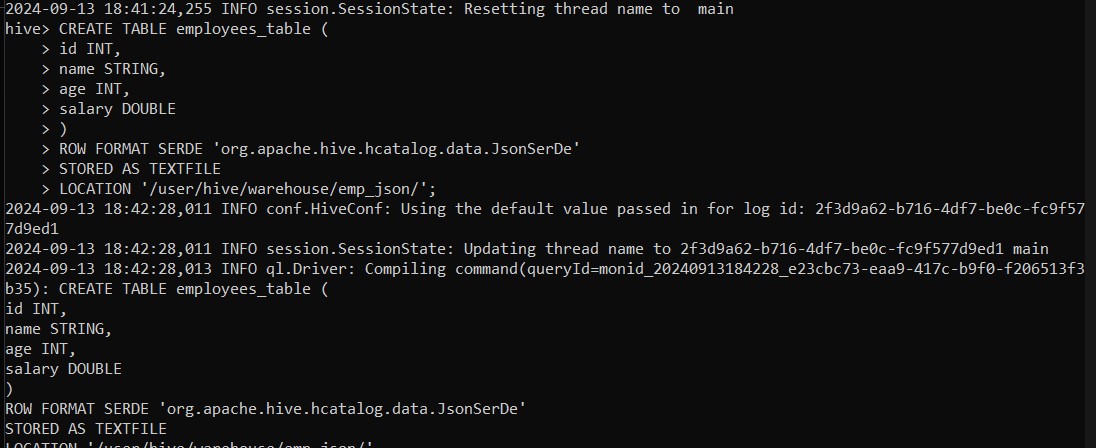
age INT, salary DOUBLE

)

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

STORED AS TEXTFILE

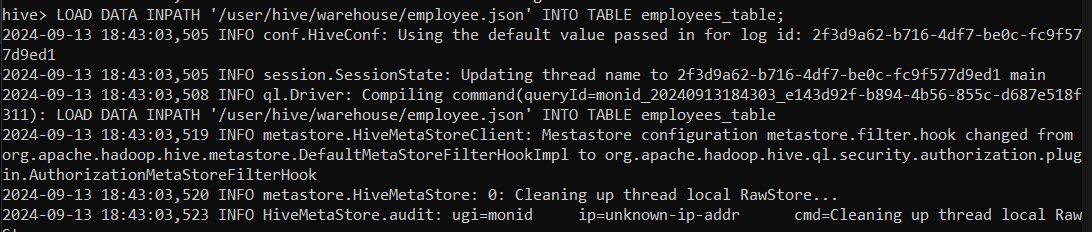
LOCATION '/ser/hive/warehouse/emp\_json/';



**Add Data to the TABLE:**

Run the **LOAD DATA INPATH** command:

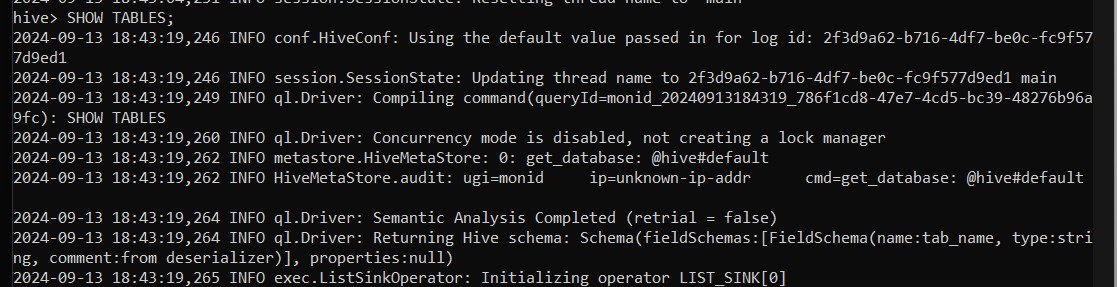
LOAD DATA INPATH '/user/hive/warehouse/emp\_json/employee.json' INTO TABLE employees\_table;



**List Hive Tables and Data:**

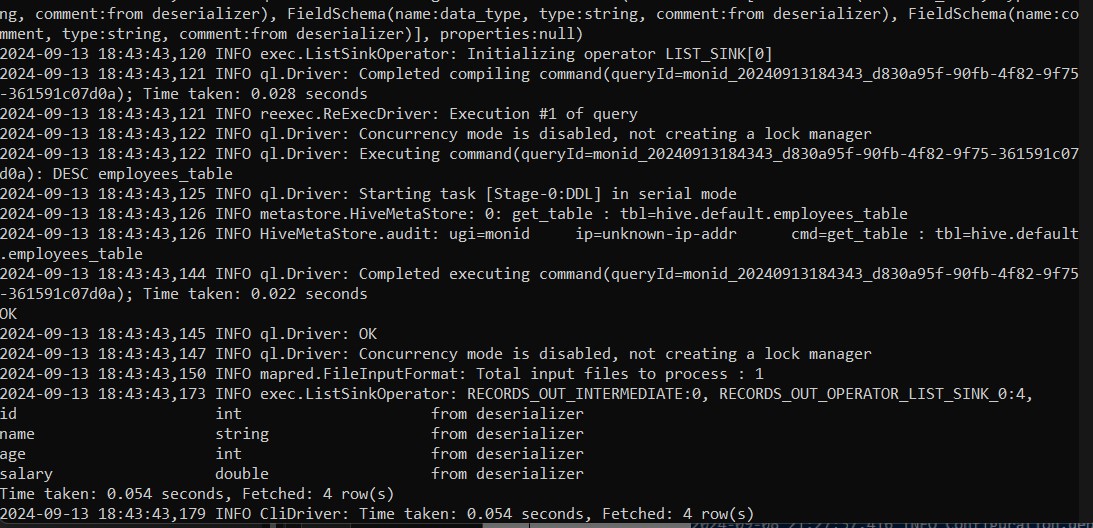
To show all tables in a selected database, use the following statement:

# SHOW TABLES;



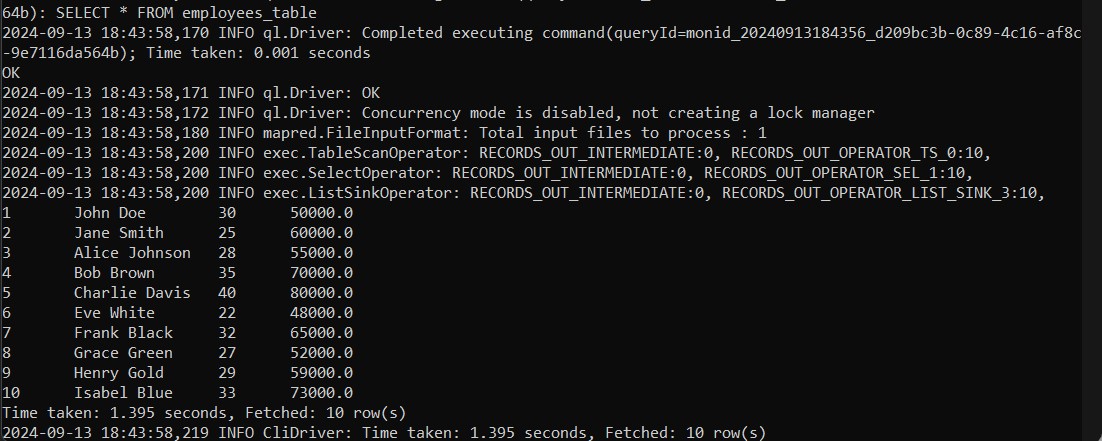
To show table column names and data types, run:

# DESC employees\_table;



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

# SELECT \* FROM employees\_table;

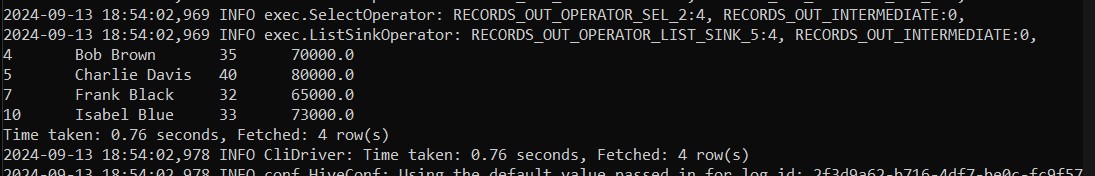
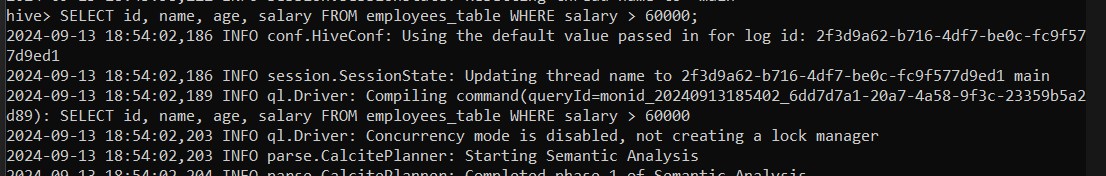


**Perform Various Operations on the Data in the table:**

**WHERE:**

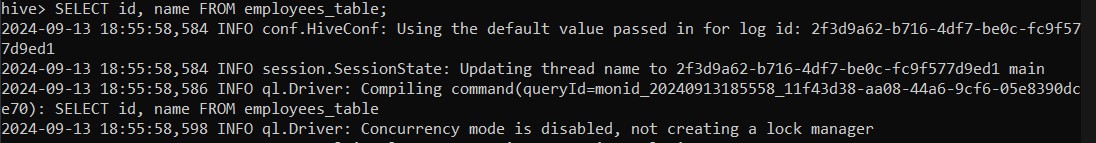
# SELECT id, name, age, salary FROM employees\_table

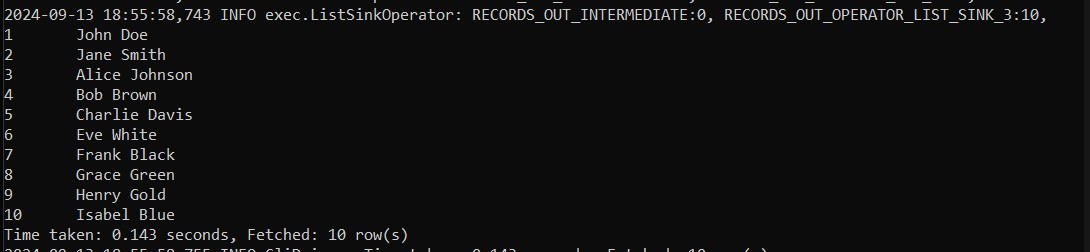
WHERE salary > 60000;



**PROJECTION: (Selecting Specific Columns)**

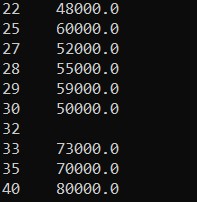
# SELECT id, name FROM employees\_table;





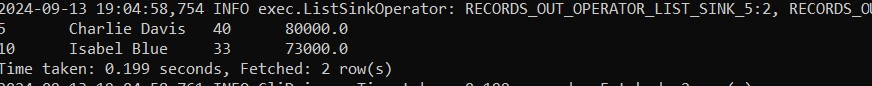
**AGGREGATION: (e.g., Summing Salaries by Age Group)**

# SELECT age, MAX(salary) AS max\_salary FROM employees\_table GROUP BY age;



**REMOVE: (Remove Specific Records)**

# SELECT \* FROM employees\_table WHERE salary > 70000;



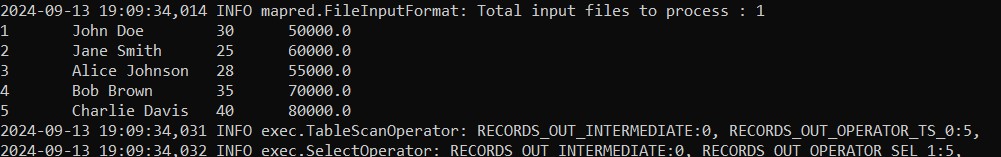
**COUNT: (Counting the Number of Records)**

# SELECT COUNT(\*) FROM employees\_table;



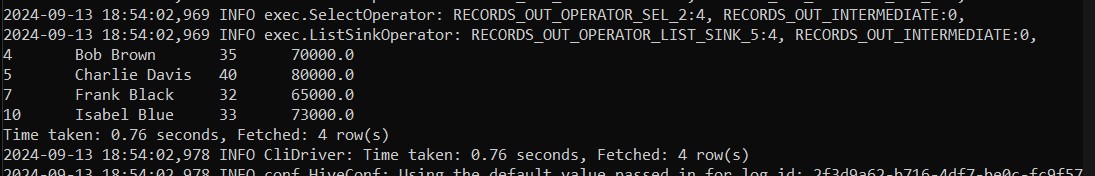
**LIMIT: (Restrict the Number of Rows Returned)**

# SELECT \* FROM employees\_table LIMIT 5;



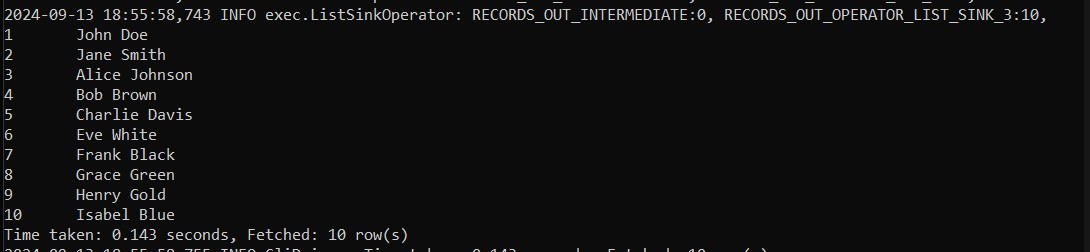
**SKIP: (Skipping the First N Rows, using Row Number)**

# 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.