

Exp.No: 6

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

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 using jq tool.

PROCEDURE:

- Create a json file 'emp.json' and provide data in it.

```
[
  {
    "name" : "Anu",
    "age":12,
    "dept": "Computer",
    "salary":10000
  },
  {
    "name" : "Bob",
    "age" :14,
    "dept" : "HR",
    "salary":15000
  },
  {
    "name": "Jane Smith",
    "age": 25,
    "department": "IT",
    "salary": 60000
  },
  {
    "name": "Alice Johnson",
    "age": 35,
    "department": "Finance",
```

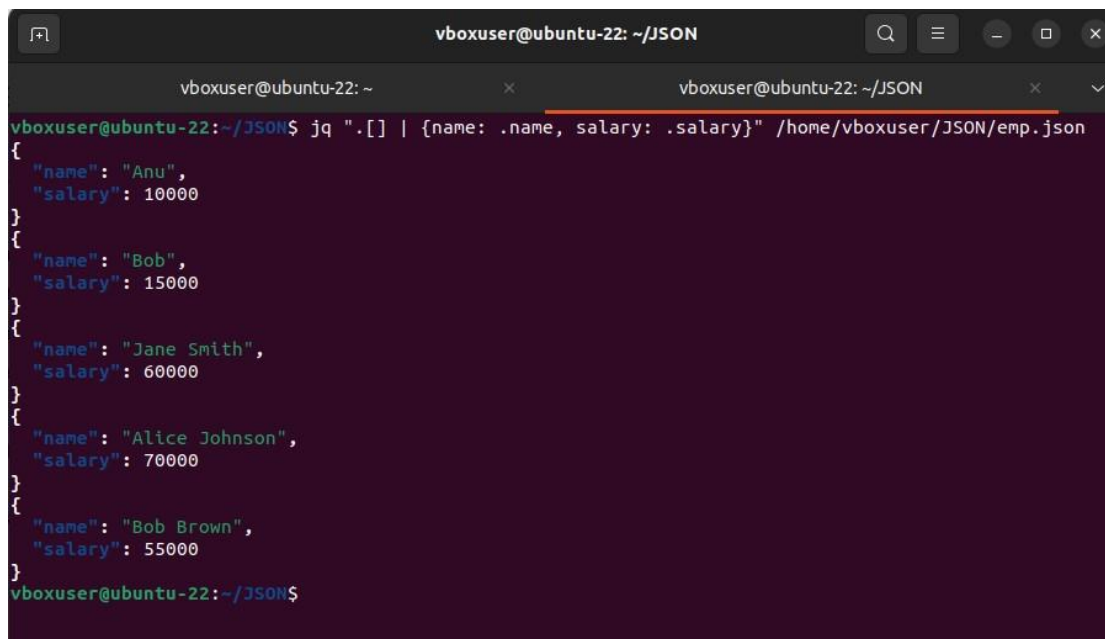
```
    "salary": 70000
  },
  {
    "name": "Bob Brown",
    "age": 28,
    "department": "Marketing",
    "salary": 55000
  }
]
```

- Open the command prompt.
- Navigate to the folder where emp.json is stored.
- Load and view the JSON data with jq.
- Use the jq commands for projection, aggregation, removal, counting, limiting, and sorting operations.

OUTPUT:

Running jq queries:

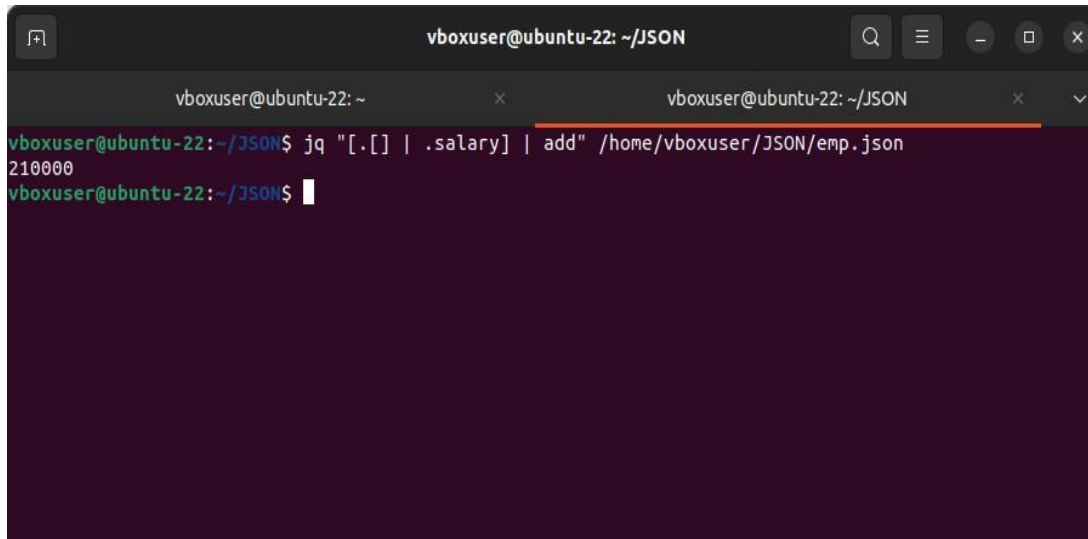
1. Projection:



A terminal window titled 'vboxuser@ubuntu-22: ~/JSON' showing a jq command and its output. The command is `jq ".[]" | {name: .name, salary: .salary}" /home/vboxuser/JSON/emp.json`. The output is a JSON array of objects, each containing 'name' and 'salary' fields.

```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22: ~/JSON$ jq ".[]" | {name: .name, salary: .salary}" /home/vboxuser/JSON/emp.json
[
  {
    "name": "Anu",
    "salary": 10000
  },
  {
    "name": "Bob",
    "salary": 15000
  },
  {
    "name": "Jane Smith",
    "salary": 60000
  },
  {
    "name": "Alice Johnson",
    "salary": 70000
  },
  {
    "name": "Bob Brown",
    "salary": 55000
  }
]
vboxuser@ubuntu-22: ~/JSON$
```

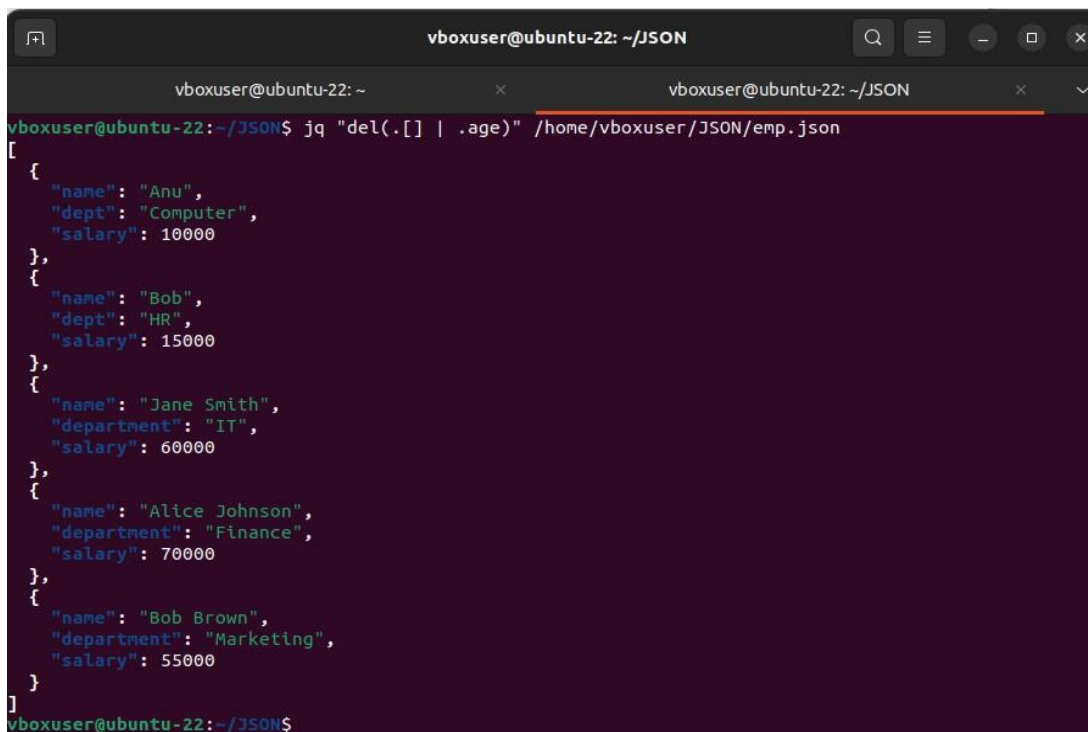
2. Aggregation:



A terminal window titled 'vboxuser@ubuntu-22: ~/JSON' with two tabs. The active tab shows the command `jq "[.[] | .salary] | add" /home/vboxuser/JSON/emp.json` being executed. The output is `210000`.

```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22:~/JSON$ jq "[.[] | .salary] | add" /home/vboxuser/JSON/emp.json
210000
vboxuser@ubuntu-22:~/JSON$
```

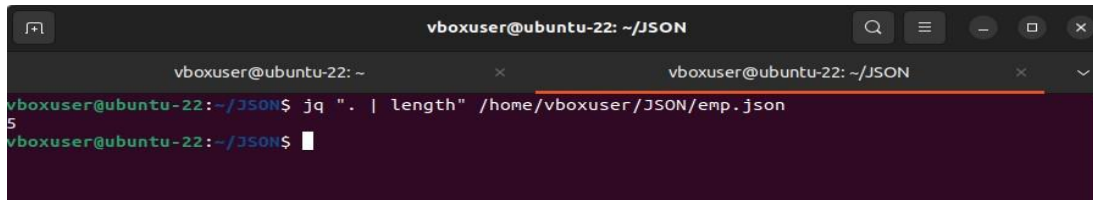
3. Remove:



A terminal window titled 'vboxuser@ubuntu-22: ~/JSON' with two tabs. The active tab shows the command `jq "del(.[] | .age)" /home/vboxuser/JSON/emp.json` being executed. The output is a JSON array of employee objects.

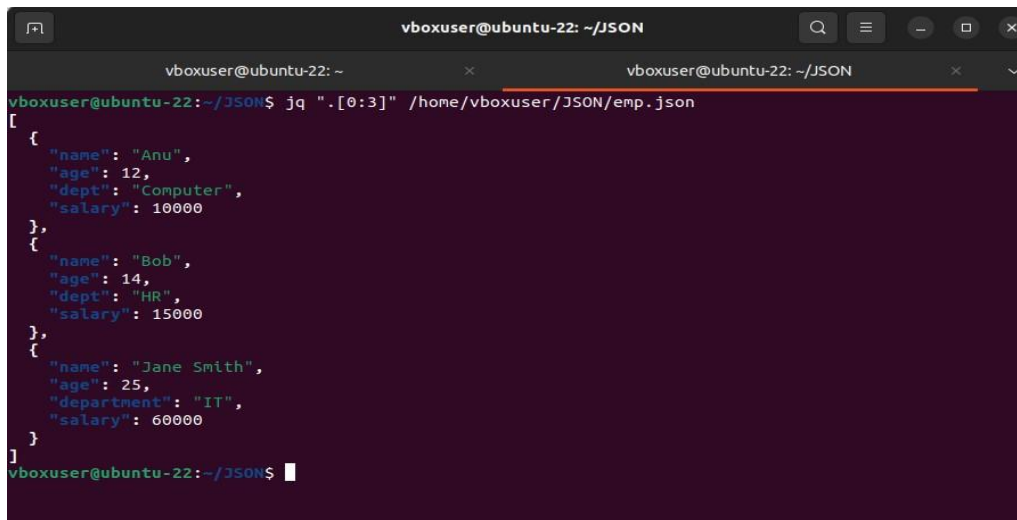
```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22:~/JSON$ jq "del(.[] | .age)" /home/vboxuser/JSON/emp.json
[
  {
    "name": "Anu",
    "dept": "Computer",
    "salary": 10000
  },
  {
    "name": "Bob",
    "dept": "HR",
    "salary": 15000
  },
  {
    "name": "Jane Smith",
    "department": "IT",
    "salary": 60000
  },
  {
    "name": "Alice Johnson",
    "department": "Finance",
    "salary": 70000
  },
  {
    "name": "Bob Brown",
    "department": "Marketing",
    "salary": 55000
  }
]
vboxuser@ubuntu-22:~/JSON$
```

4. Count:



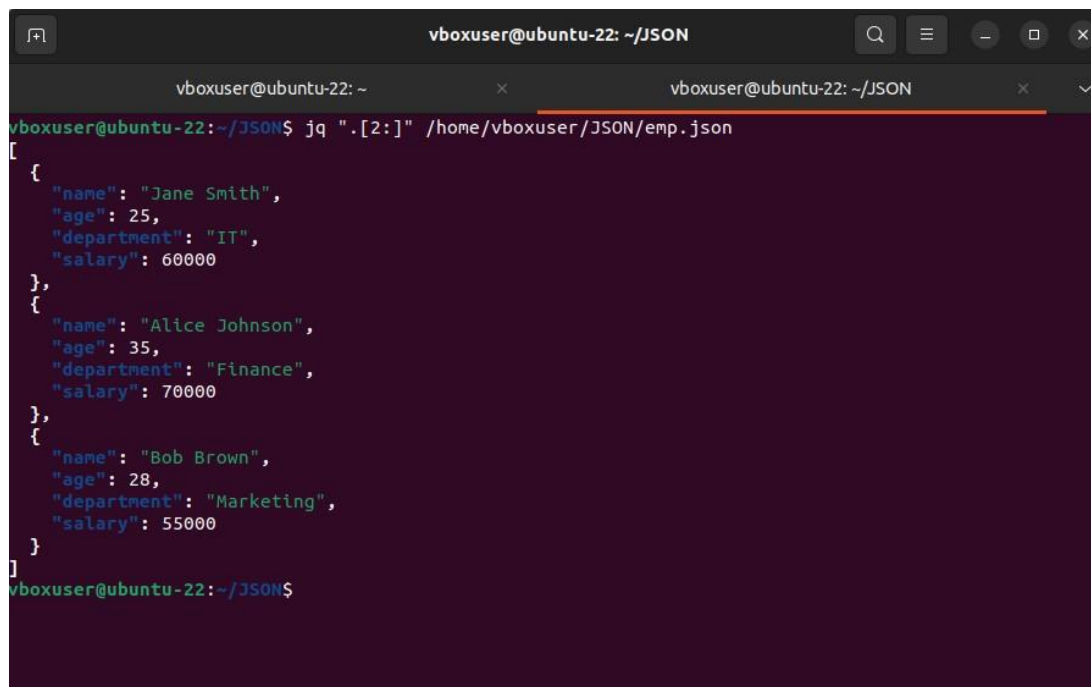
```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22:~/JSON$ jq ". | length" /home/vboxuser/JSON/emp.json
5
vboxuser@ubuntu-22:~/JSON$
```

5. Limit:



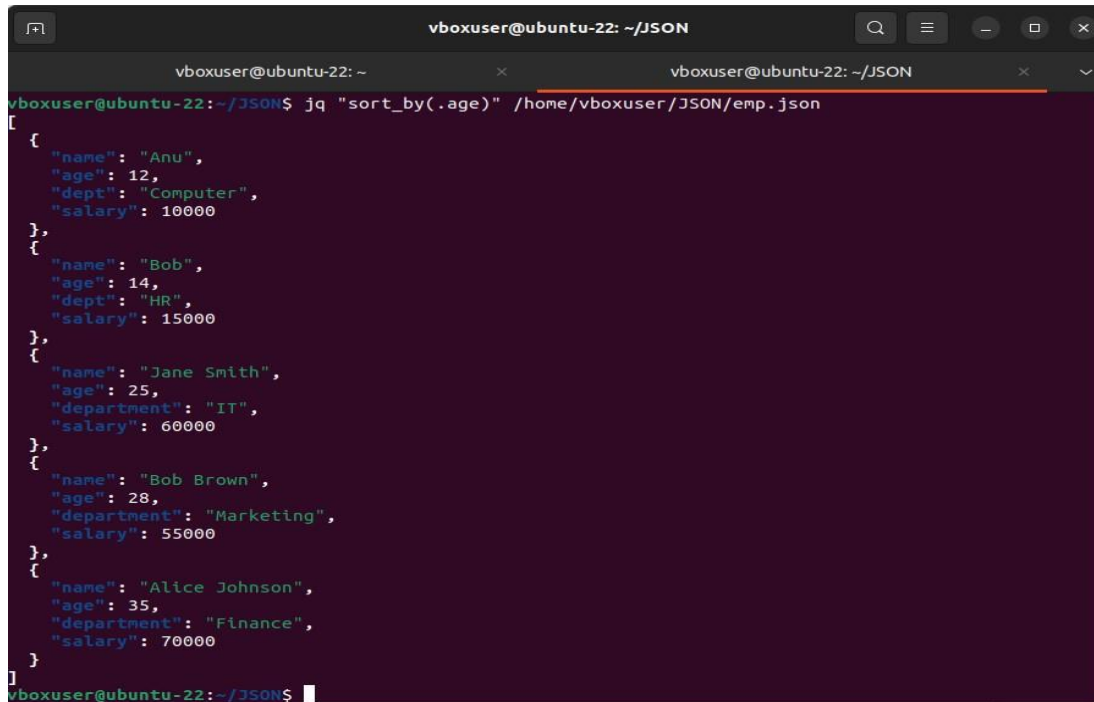
```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22:~/JSON$ jq ".[0:3]" /home/vboxuser/JSON/emp.json
[
  {
    "name": "Anu",
    "age": 12,
    "dept": "Computer",
    "salary": 10000
  },
  {
    "name": "Bob",
    "age": 14,
    "dept": "HR",
    "salary": 15000
  },
  {
    "name": "Jane Smith",
    "age": 25,
    "department": "IT",
    "salary": 60000
  }
]
vboxuser@ubuntu-22:~/JSON$
```

6. Skip:



```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22:~/JSON$ jq ".[2:]" /home/vboxuser/JSON/emp.json
[
  {
    "name": "Jane Smith",
    "age": 25,
    "department": "IT",
    "salary": 60000
  },
  {
    "name": "Alice Johnson",
    "age": 35,
    "department": "Finance",
    "salary": 70000
  },
  {
    "name": "Bob Brown",
    "age": 28,
    "department": "Marketing",
    "salary": 55000
  }
]
vboxuser@ubuntu-22:~/JSON$
```

7. Sort:

A terminal window titled 'vboxuser@ubuntu-22: ~/JSON' shows a jq command being executed. The command is 'jq "sort_by(.age)" /home/vboxuser/JSON/emp.json'. The output is a JSON array of five employee objects, sorted by age in ascending order. The objects are: Anu (age 12, Computer dept, salary 10000), Bob (age 14, HR dept, salary 15000), Jane Smith (age 25, IT dept, salary 60000), Bob Brown (age 28, Marketing dept, salary 55000), and Alice Johnson (age 35, Finance dept, salary 70000).

```
vboxuser@ubuntu-22: ~/JSON
vboxuser@ubuntu-22: ~
vboxuser@ubuntu-22: ~/JSON$ jq "sort_by(.age)" /home/vboxuser/JSON/emp.json
[
  {
    "name": "Anu",
    "age": 12,
    "dept": "Computer",
    "salary": 10000
  },
  {
    "name": "Bob",
    "age": 14,
    "dept": "HR",
    "salary": 15000
  },
  {
    "name": "Jane Smith",
    "age": 25,
    "department": "IT",
    "salary": 60000
  },
  {
    "name": "Bob Brown",
    "age": 28,
    "department": "Marketing",
    "salary": 55000
  },
  {
    "name": "Alice Johnson",
    "age": 35,
    "department": "Finance",
    "salary": 70000
  }
]
vboxuser@ubuntu-22: ~/JSON$
```

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 using jq tool is completed successfully