

Ex: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 perform operations such as projection, aggregation, removal, counting, limiting, skipping, and sorting using jq on Windows.

PROCEDURE:

1. Install `jq`:
 - Download `jq` from its official website.
 - Extract the `.zip` file and place the `jq.exe` in a folder.
 - Add the folder path to the system's environment variables to make `jq` accessible from any command prompt window.
2. Open Command Prompt:
 - Open the Command Prompt by pressing `Win + R`, typing `cmd`, and pressing Enter.
3. Navigate to the Directory with the JSON File:
 - Use the Command Prompt to navigate to the folder where your JSON file is located.
4. Projection:
 - Use `jq` to select specific fields from the JSON file, displaying only the required data.
5. Aggregation:
 - Perform basic aggregations such as summing up values or calculating the average for numerical fields.
6. Remove Fields:
 - Remove unwanted fields from the JSON data, ensuring only the necessary information is kept.
7. Count Elements:
 - Count the number of elements in the JSON file to understand the data size.
8. Limit the Data:
 - Limit the number of records displayed to focus on a subset of the data.
9. Skip Records:
 - Skip the first few records to process or view a specific part of the data.
10. Sort the Data:
 - Sort the JSON data based on a specific field, either in ascending or descending order.
11. View the Results:
 - After performing each operation, the filtered or modified results will be displayed in the Command Prompt.

EX 6

OUTPUT:

PROJECTION:

```
C:\jq>jq-windows-amd64.exe "[.[] | {id: .id, name: .name}]" sample_json.json
[
  {
    "id": 1,
    "name": "Alice"
  },
  {
    "id": 2,
    "name": "Bob"
  },
  {
    "id": 3,
    "name": "Charlie"
  }
]
```

AGGREGATION(AVERAGE):

```
C:\jq>jq-windows-amd64.exe "[.[] | .age] | add / length" sample_json.json
29
```

COUNT:

```
C:\jq>jq-windows-amd64.exe "length" sample_json.json
3
```

LIMIT:

```
C:\jq>jq-windows-amd64.exe ".[0:2]" sample_json.json
[
  {
    "id": 1,
    "name": "Alice",
    "age": 28,
    "isEmployed": true,
    "address": {
      "street": "123 Maple Street",
      "city": "Springfield"
    },
    "skills": [
      "Python",
      "Data Analysis"
    ]
  },
  {
    "id": 2,
    "name": "Bob",
    "age": 34,
    "isEmployed": false,
    "address": {
      "street": "456 Oak Avenue",
      "city": "Shelbyville"
    },
    "skills": [
      "Java",
      "Project Management"
    ]
  }
]
```

EX 6

SKIP:

```
C:\jq>jq-windows-amd64.exe ".[1:]" sample_json.json
[
  {
    "id": 2,
    "name": "Bob",
    "age": 34,
    "isEmployed": false,
    "address": {
      "street": "456 Oak Avenue",
      "city": "Shelbyville"
    },
    "skills": [
      "Java",
      "Project Management"
    ]
  },
  {
    "id": 3,
    "name": "Charlie",
    "age": 25,
    "isEmployed": true,
    "address": {
      "street": "789 Pine Road",
      "city": "Ogdenville"
    },
    "skills": [
      "JavaScript",
      "Web Development"
    ]
  }
]
```

EX 6

SORT:

```
C:\jq>jq-windows-amd64.exe "sort_by(.age)" sample_json.json
[
  {
    "id": 3,
    "name": "Charlie",
    "age": 25,
    "isEmployed": true,
    "address": {
      "street": "789 Pine Road",
      "city": "Ogdenville"
    },
    "skills": [
      "JavaScript",
      "Web Development"
    ]
  },
  {
    "id": 1,
    "name": "Alice",
    "age": 28,
    "isEmployed": true,
    "address": {
      "street": "123 Maple Street",
      "city": "Springfield"
    },
    "skills": [
      "Python",
      "Data Analysis"
    ]
  },
  {
    "id": 2,
    "name": "Bob",
    "age": 34,
    "isEmployed": false,
    "address": {
      "street": "456 Oak Avenue",
      "city": "Shelbyville"
    },
    "skills": [
      "Java",
      "Project Management"
    ]
  }
]
```

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 MongoDB is completed successfully.