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.

PROCEDURE:

1. Open command prompt and run jq to start the jq server.
2. Create a database using use <database_name>.
3. After creating the database importing the JSON file perform specific commands for projection, aggregation, remove, count, limit and sort.

OUTPUT:

```
C:\ex_6 da>jq "[.[] | select(.age > 22)] | length" data.json
2
```

```
C:\ex_6 da>jq "[.[] | .score] | add" data.json
265
```

```
C:\ex_6 da>jq "[.[] | .score] | add / length" data.json
88.33333333333333
```

```
C:\ex_6 da>jq "sort_by(.score) | reverse" data.json
[
  {
    "name": "Bob",
    "age": 20,
    "score": 92
  },
  {
    "name": "Alice",
    "age": 25,
    "score": 88
  },
  {
    "name": "Charlie",
    "age": 23,
    "score": 85
  }
]
```

```
C:\ex_6 da>jq ".[1:]" data.json
[
  {
    "name": "Bob",
    "age": 20,
    "score": 92
  },
  {
    "name": "Charlie",
    "age": 23,
    "score": 85
  }
]
```

```
C:\ex_6 da>jq ".[0:2]" data.json
[
  {
    "name": "Alice",
    "age": 25,
    "score": 88
  },
  {
    "name": "Bob",
    "age": 20,
    "score": 92
  }
]
```

```
C:\ex_6 da>jq "del(.[] | select(.age < 23))" data.json
[
  {
    "name": "Alice",
    "age": 25,
    "score": 88
  },
  {
    "name": "Charlie",
    "age": 23,
    "score": 85
  }
]
```

```
C:\ex_6 da>jq ".[] | {name, age}" data.json
{
  "name": "Alice",
  "age": 25
}
{
  "name": "Bob",
  "age": 20
}
{
  "name": "Charlie",
  "age": 23
}

C:\ex_6 da>jq ".[] | {name, age}" data.json
{
  "name": "Alice",
  "age": 25
}
{
  "name": "Bob",
  "age": 20
}
{
  "name": "Charlie",
  "age": 23
}
```

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 is completed successfully.