MAI172: Advance Database Technologies

Register Number: 2448513

Name: Deshmukh Pratik Bhushanrao

Experiment Number and Name 8: Demonstration MongoDB aggregate functions.

Date: 05/09/2024 Time: 9.45 to 11.45

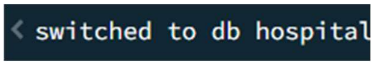
---

1. Create a Database “Demo_Database”

Query:

Use hospital

Output:



```
switched to db hospital
```

Inference:

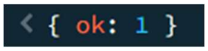
Created a database for our task.

2. Create a Collection “patient”

Query:

db.createCollection(“patient”)

Output:



```
{ ok: 1 }
```

Inference:

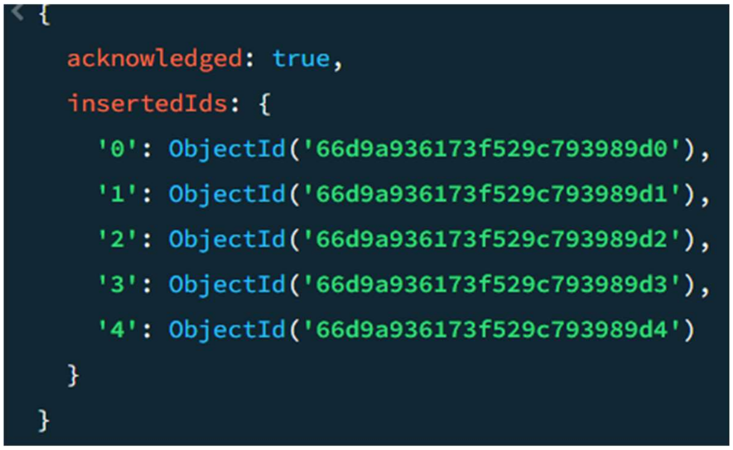
Created a collection for our task.

3. Insert records into the collection.

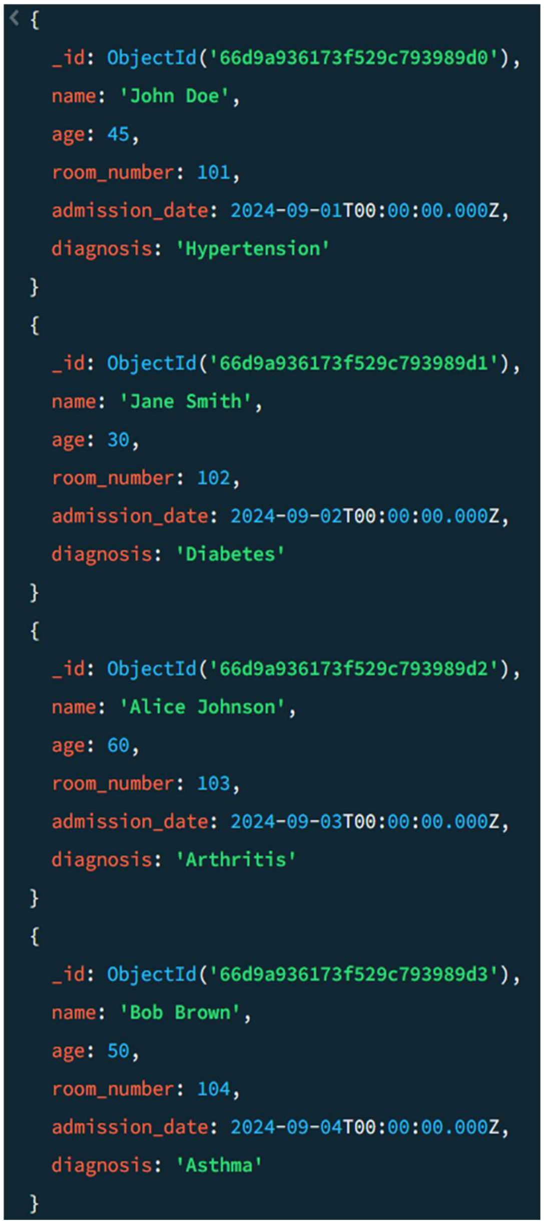
Query:

```
db.patients.insertMany([
{
"name": "John Doe",
"age": 45,
"room_number": 101,
"admission_date": new Date("2024-09-01"),
"diagnosis": "Hypertension"
},
{
"name": "Jane Smith",
"age": 30,
"room_number": 102,
"admission_date": new Date("2024-09-02"),
"diagnosis": "Diabetes"
},
{
"name": "Alice Johnson",
"age": 60,
"room_number": 103,
"admission_date": new Date("2024-09-03"),
"diagnosis": "Arthritis"
},
{
"name": "Bob Brown",
"age": 50,
"room_number": 104,
"admission_date": new Date("2024-09-04"),
"diagnosis": "Asthma"
},
{
"name": "Charlie Davis",
"age": 35,
"room_number": 105,
"admission_date": new Date("2024-09-05"),
"diagnosis": "Migraine"
}
])
```

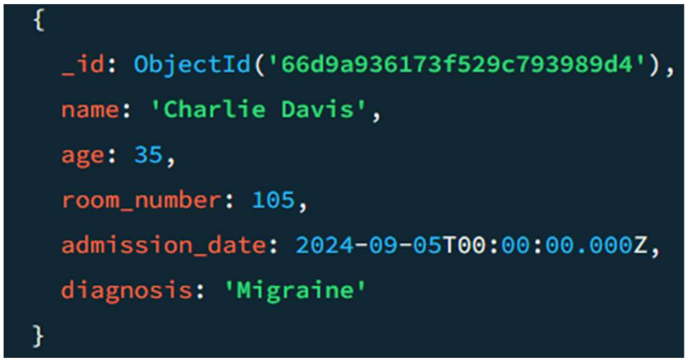
Output:



```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66d9a936173f529c793989d0'),
    '1': ObjectId('66d9a936173f529c793989d1'),
    '2': ObjectId('66d9a936173f529c793989d2'),
    '3': ObjectId('66d9a936173f529c793989d3'),
    '4': ObjectId('66d9a936173f529c793989d4')
  }
}
```



```
{
  _id: ObjectId('66d9a936173f529c793989d0'),
  name: 'John Doe',
  age: 45,
  room_number: 101,
  admission_date: 2024-09-01T00:00:00.000Z,
  diagnosis: 'Hypertension'
}
{
  _id: ObjectId('66d9a936173f529c793989d1'),
  name: 'Jane Smith',
  age: 30,
  room_number: 102,
  admission_date: 2024-09-02T00:00:00.000Z,
  diagnosis: 'Diabetes'
}
{
  _id: ObjectId('66d9a936173f529c793989d2'),
  name: 'Alice Johnson',
  age: 60,
  room_number: 103,
  admission_date: 2024-09-03T00:00:00.000Z,
  diagnosis: 'Arthritis'
}
{
  _id: ObjectId('66d9a936173f529c793989d3'),
  name: 'Bob Brown',
  age: 50,
  room_number: 104,
  admission_date: 2024-09-04T00:00:00.000Z,
  diagnosis: 'Asthma'
}
```



```
{
  _id: ObjectId('66d9a936173f529c793989d4'),
  name: 'Charlie Davis',
  age: 35,
  room_number: 105,
  admission_date: 2024-09-05T00:00:00.000Z,
  diagnosis: 'Migraine'
}
```

Inference:

Inserting data into our collection.

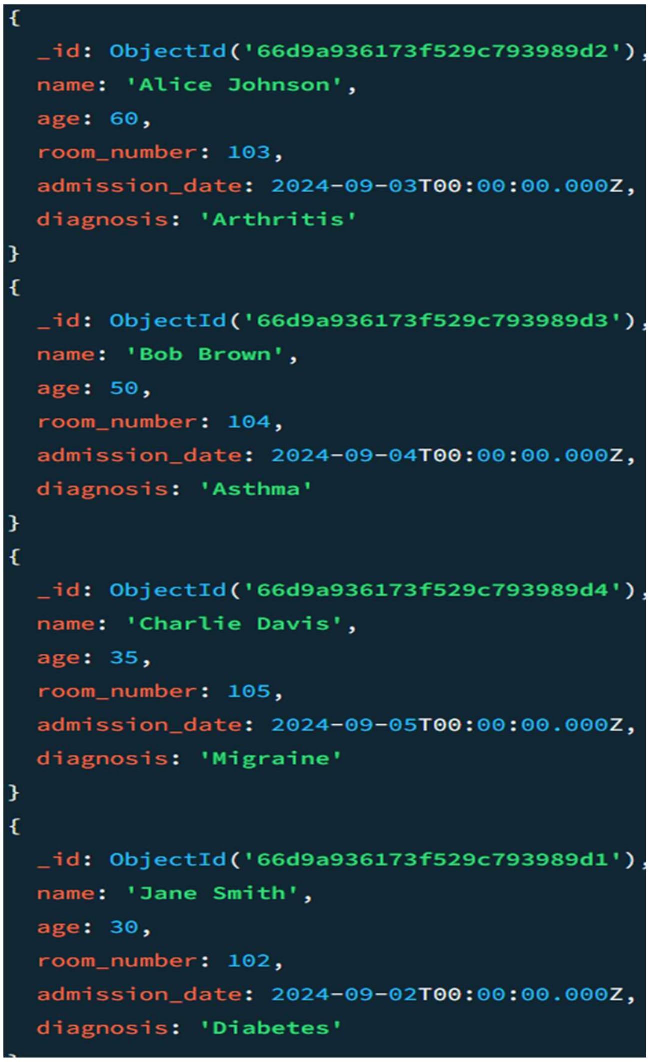
4. Apply aggregate function (Minimum 6) with logical sense.

A. Sort the patients based on the name of the patients.

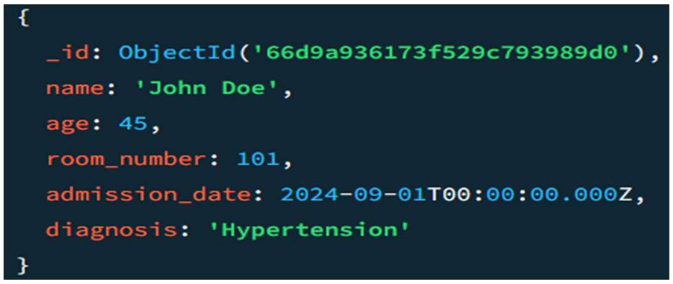
Query:

```
db.patient.aggregate([{$sort: {name: 1}}])
```

Output:



```
{
  _id: ObjectId('66d9a936173f529c793989d2'),
  name: 'Alice Johnson',
  age: 60,
  room_number: 103,
  admission_date: 2024-09-03T00:00:00.000Z,
  diagnosis: 'Arthritis'
}
{
  _id: ObjectId('66d9a936173f529c793989d3'),
  name: 'Bob Brown',
  age: 50,
  room_number: 104,
  admission_date: 2024-09-04T00:00:00.000Z,
  diagnosis: 'Asthma'
}
{
  _id: ObjectId('66d9a936173f529c793989d4'),
  name: 'Charlie Davis',
  age: 35,
  room_number: 105,
  admission_date: 2024-09-05T00:00:00.000Z,
  diagnosis: 'Migraine'
}
{
  _id: ObjectId('66d9a936173f529c793989d1'),
  name: 'Jane Smith',
  age: 30,
  room_number: 102,
  admission_date: 2024-09-02T00:00:00.000Z,
  diagnosis: 'Diabetes'
```



```
{
  _id: ObjectId('66d9a936173f529c793989d0'),
  name: 'John Doe',
  age: 45,
  room_number: 101,
  admission_date: 2024-09-01T00:00:00.000Z,
  diagnosis: 'Hypertension'
}
```

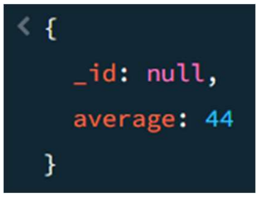
Inference: This query sorts all the records in our collection (patients) based on the patients' names.

B. Find the average age of the patients in the hospital.

Query:

```
db.patients.aggregate ([{$group: { _id: null, average: {$avg: "$age"} } }])
```

Output:



```
{
  _id: null,
  average: 44
}
```

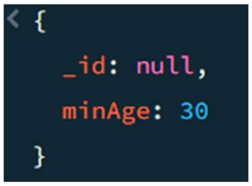
Inference: The query displays the average age of patients in our collection (patients).

C. Find the minimum age of the patient admitted into the hospital

Query:

```
db.patients.aggregate ({$group: { _id null, minAge: {$min: “$age” }}})
```

Output:



```
{
  _id: null,
  minAge: 30
}
```

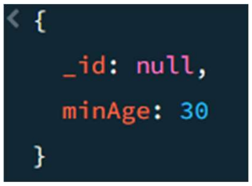
Inference: The query displays the minimum age of the patients in our collection (patients).

D. Find the maximum age of the patient admitted into the hospital.

Query:

```
db.patients.aggregate( {$group: { _id: null, MaxAge: {$max: “$age”}}})
```

Output:



```
{
  _id: null,
  minAge: 30
}
```

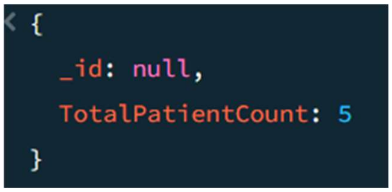
Inference: The query displays the maximum age of the patients in our collection (patients).

E. Count the total number of patients admitted into the hospital.

Query:

```
db.patients.aggregate ( {$group: { _id: null, TotalPatientCount: {$sum: 1 } } })
```

Output:



```
< {
    _id: null,
    TotalPatientCount: 5
  }
```

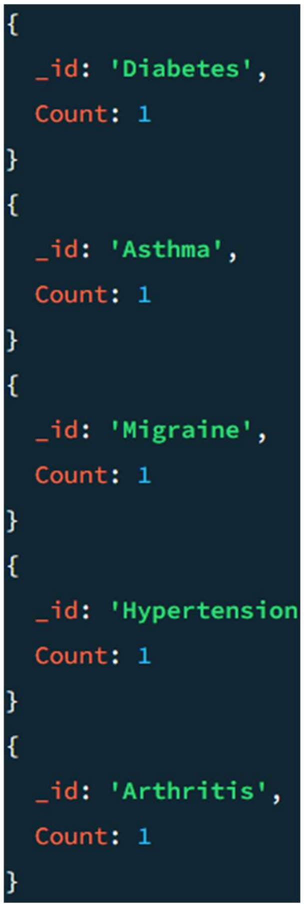
Inference: The query displays the total number of items in our collection (patients).

F. Group patients by diagnosis and count the number of patients in each group.

Query:

db.patients.aggregate ({ $group: { _id: "$**diagnosis**", Count: { $sum : 1} } })

Output:



```
{
  _id: 'Diabetes',
  Count: 1
}
{
  _id: 'Asthma',
  Count: 1
}
{
  _id: 'Migraine',
  Count: 1
}
{
  _id: 'Hypertension
  Count: 1
}
{
  _id: 'Arthritis',
  Count: 1
}
```

Inference: The query displays the number of patients with specific diagnosis criteria.