LIMIT AND SELECTORS

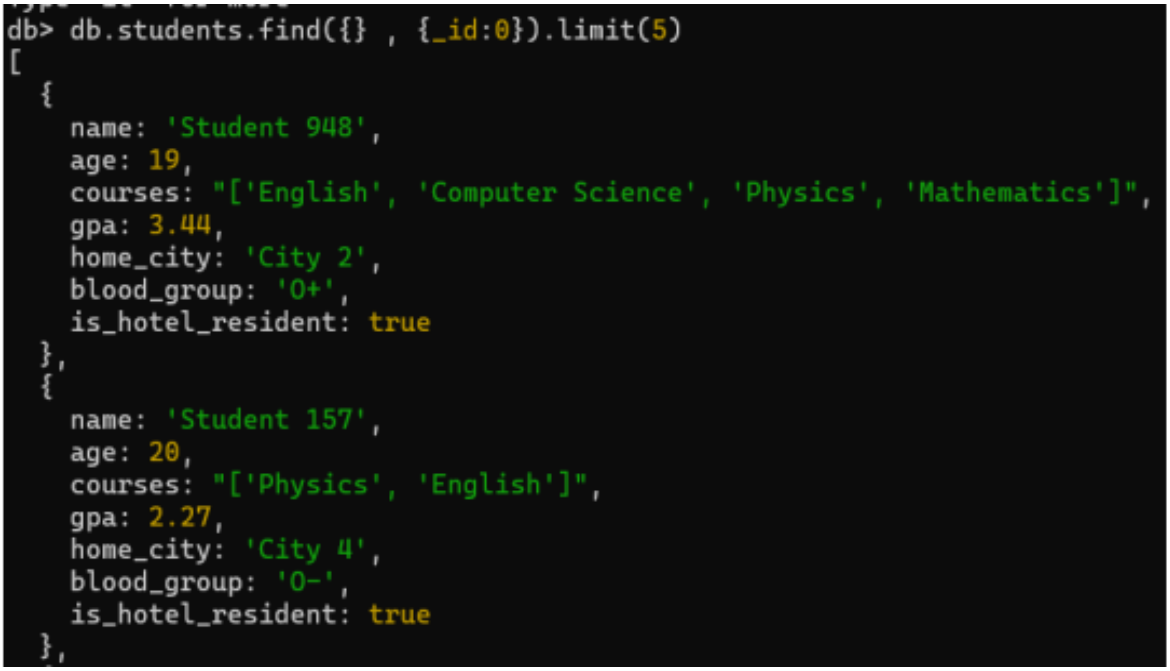
Limit:

The limit operator is used with the find method. It's chained after the filter criteria or any sorting operations.

Syntax:

"db.collection.find ({filter},
{projection}).limit (number)"

Example:



```
db> db.students.find({} , {_id:0}).limit(5)
[
  {
    name: 'Student 948',
    age: 19,
    courses: "['English', 'Computer Science', 'Physics', 'Mathematics']",
    gpa: 3.44,
    home_city: 'City 2',
    blood_group: 'O+',
    is_hotel_resident: true
  },
  {
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
```

Here the limit is used for that this command give the number of records or documents that we want. In the above example to get only first five document we use limit (5).

Selectors:

- Comparison gt and lt.
- AND operator.
- OR operator.

➢ Comparison $gt and $lt:

$gt:

This selects documents where the value of a specified field is greater than the specified value.

$lt:

This selects documents where the value of a specified field is lesser than the specified value. Now, to find all the students with age greater than 20.

```
db> db.stud.find({age:{$gt:20}});
[
  {
    _id: ObjectId('665a89d776fc88153fffc09f'),
    name: 'Student 346',
    age: 25,
    courses: "['Mathematics', 'History', 'English']",
    gpa: 3.31,
    home_city: 'City 8',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('665a89d776fc88153fffc0a0'),
    name: 'Student 930',
    age: 25,
    courses: "['English', 'Computer Science', 'Mathematics', 'History']",
    gpa: 3.63,
    home_city: 'City 3',
    blood_group: 'A-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('665a89d776fc88153fffc0a1'),
    name: 'Student 305',
    age: 24,
    courses: "['History', 'Physics', 'Computer Science', 'Mathematics']",
    gpa: 3.4,
    home_city: 'City 6',
    blood_group: 'O+',
    is_hotel_resident: true
```

➢ AND operator:

A logical `AND` operation on an array of one or more expressions (`<expression1>`, `<expression2>`, and so on) and selects the documents that satisfy all the expressions. To find students from “city 2” with blood group “B+”.

```
db.stud.find({
$and:[
{home_city: "City 2"},
{blood_group: "B+" }
]
});


_id: ObjectId('665a89d776fc88153fffc0b4'),
name: 'Student 504',
age: 21,
courses: "['Physics', 'Computer Science', 'English', 'Mathematics']",
gpa: 2.42,
home_city: 'City 2',
blood_group: 'B+',
is_hotel_resident: true
,

_id: ObjectId('665a89d776fc88153fffc0eb'),
name: 'Student 367',
age: 19,
courses: "['English', 'Physics', 'History', 'Mathematics']",
gpa: 2.81,
home_city: 'City 2',
blood_group: 'B+',
is_hotel_resident: false
```

The above example shows the AND operator.

➢ OR operator:

The OR operation can be explained by using the following example, here we take a example to find the Student who are hostel residents OR have a GPA less than 3.0 It means it takes either the students how are hostel residents or the students whose gpa is less than 3.0.

```
db> db.stud.find({
... $or:[
... {is_hostel_resident:true},
... {gpa:{$lt:3.0}}
... ]
... });
[
  {
    _id: ObjectId('665a89d776fc88153fffc09d'),
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('665a89d776fc88153fffc09e'),
    name: 'Student 316',
    age: 20,
    courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
    gpa: 2.32,
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('665a89d776fc88153fffc0a3'),
    name: 'Student 563',
    age: 18,
    courses: "['Mathematics', 'English']",
```

The above example gives the correct conclusion about the OR operation. Here we take the example that wants to give the output as the students who are hostel residents OR the students whose gpa is less than 3.0 .