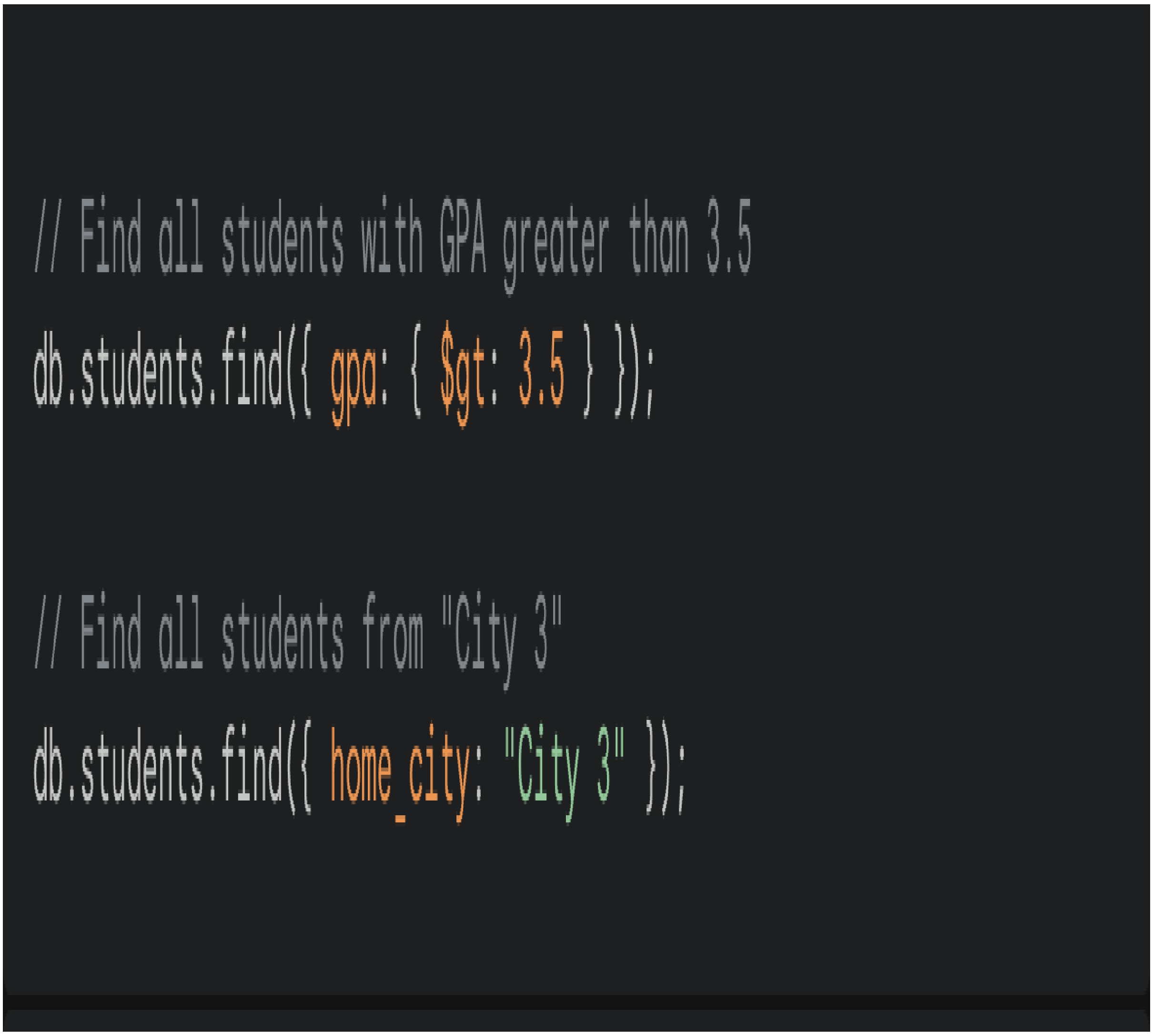
Where,AND,OR&CRUD

EXPERIMENT-1

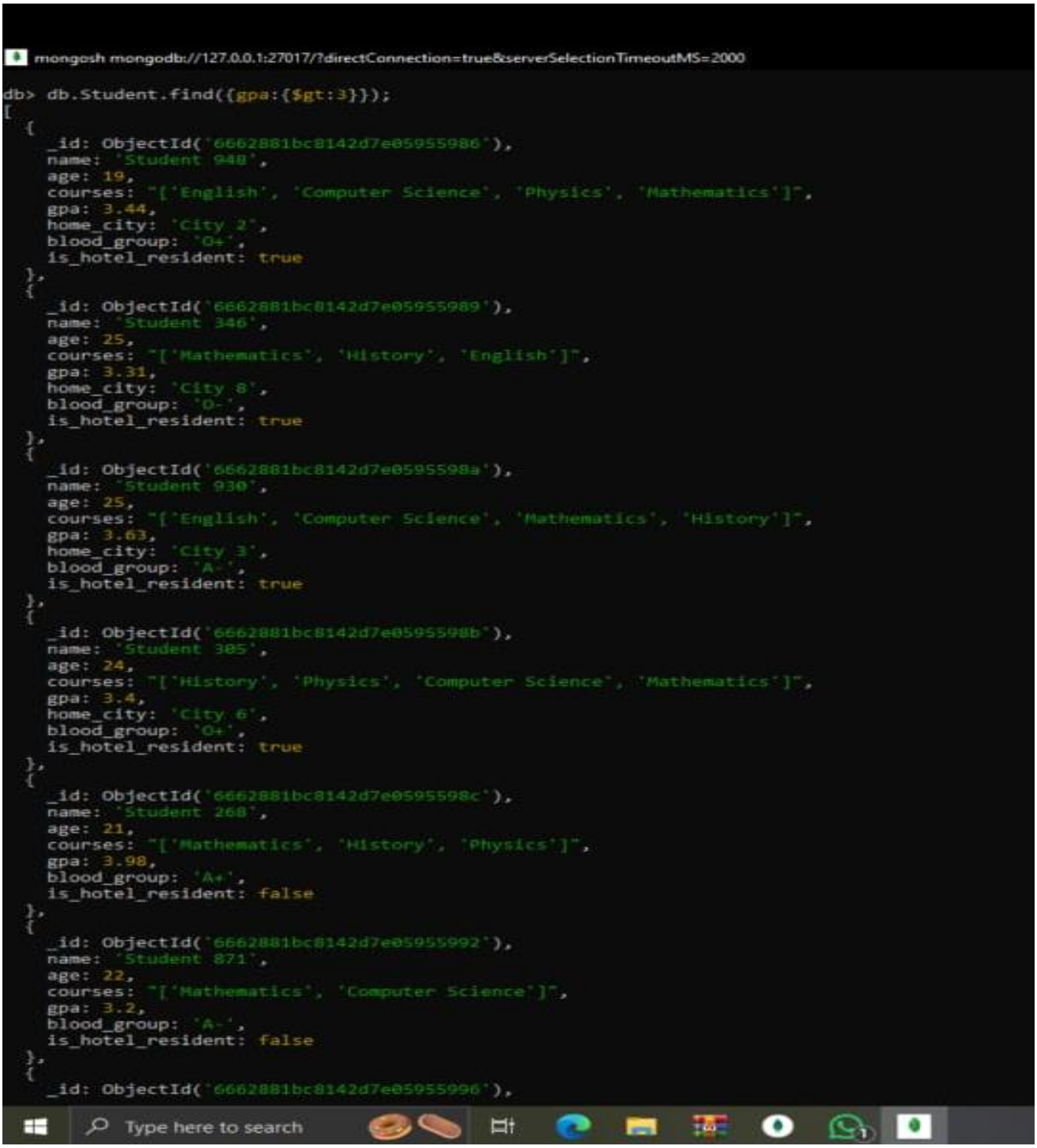
WHERE

It is a collection that you want to filter a subset based on condition .That is the place we are going to use the WHERE



```
// Find all students with GPA greater than 3.5
db.students.find({ gpa: { $gt: 3.5 } });

// Find all students from "City 3"
db.students.find({ home_city: "City 3" });
```

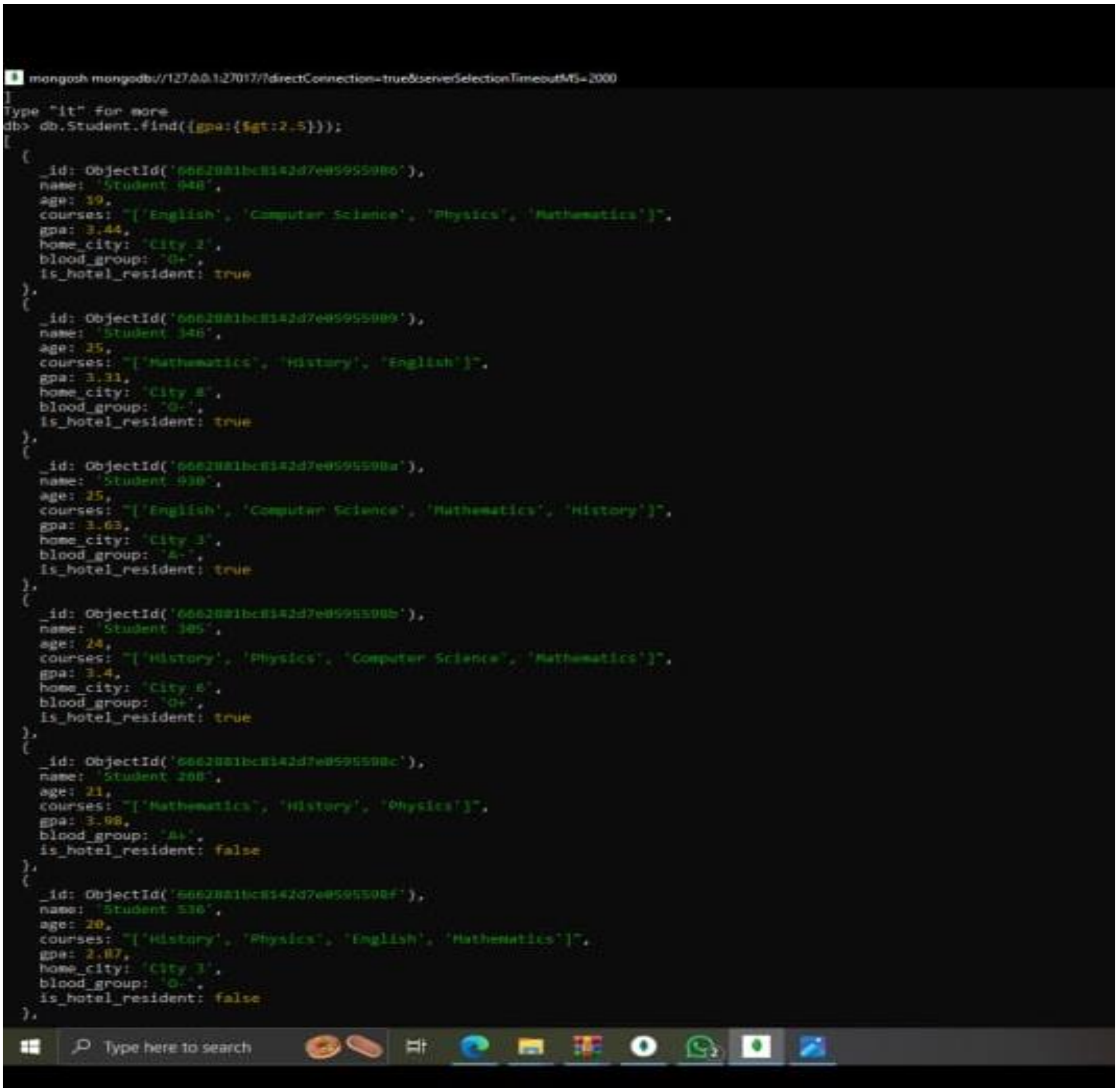
Output:



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
db> db.Student.find({gpa:{$gt:3}});
[
  {
    _id: ObjectId('6662881bc8142d7e05955986'),
    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
  },
  {
    _id: ObjectId('6662881bc8142d7e05955989'),
    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('6662881bc8142d7e0595598a'),
    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('6662881bc8142d7e0595598b'),
    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
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598c'),
    name: 'Student 268',
    age: 21,
    courses: "['Mathematics', 'History', 'Physics']",
    gpa: 3.98,
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6662881bc8142d7e05955992'),
    name: 'Student 871',
    age: 22,
    courses: "['Mathematics', 'Computer Science']",
    gpa: 3.2,
    blood_group: 'A-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6662881bc8142d7e05955996'),
```

OUTPUT:



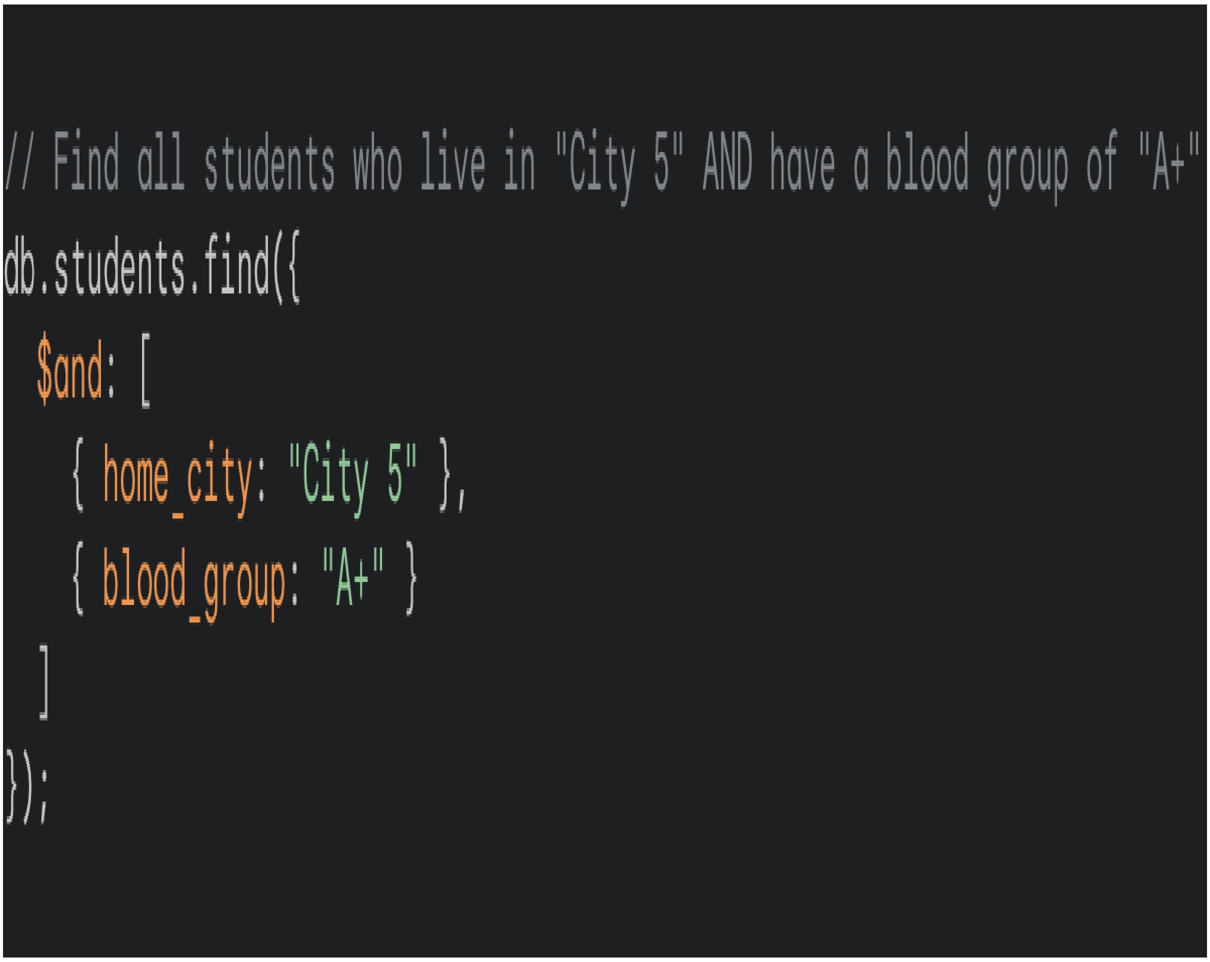
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
]
Type "it" for more
db> db.Student.find({gpa:{$gt:2.5}});
[
  {
    _id: ObjectId('6662881bc8142d7e05955986'),
    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
  },
  {
    _id: ObjectId('6662881bc8142d7e05955989'),
    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('6662881bc8142d7e0595598a'),
    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('6662881bc8142d7e0595598b'),
    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
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598c'),
    name: 'Student 268',
    age: 21,
    courses: "['Mathematics', 'History', 'Physics']",
    gpa: 3.98,
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598f'),
    name: 'Student 536',
    age: 20,
    courses: "['History', 'Physics', 'English', 'Mathematics']",
    gpa: 2.87,
    home_city: 'City 3',
    blood_group: 'O-',
    is_hotel_resident: false
  },
```

AND

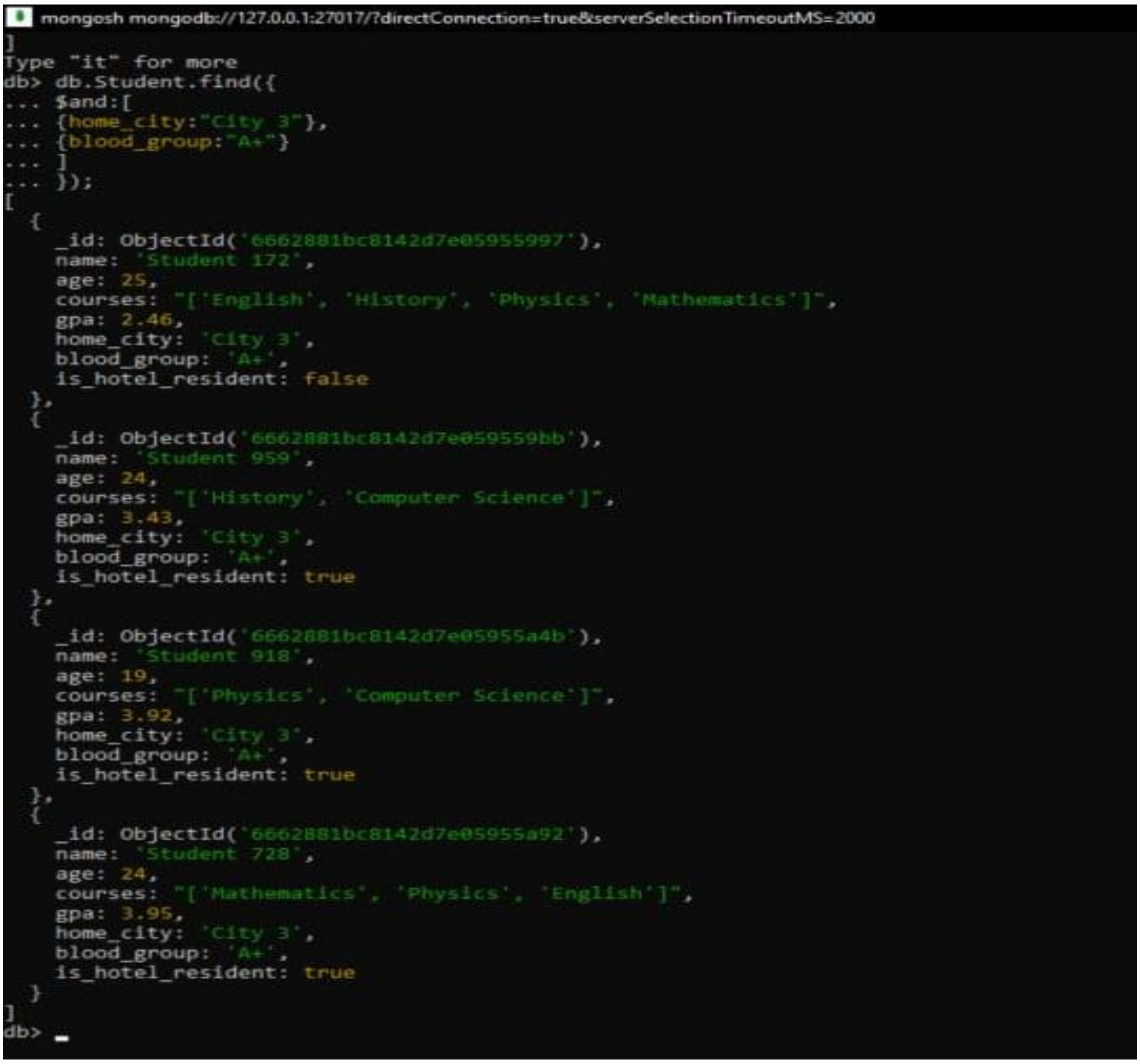
TO FILTER A SUBSET BASED ON MULTIPLE CONDITION A COLLECTION THAT WE WANT

INPUT:



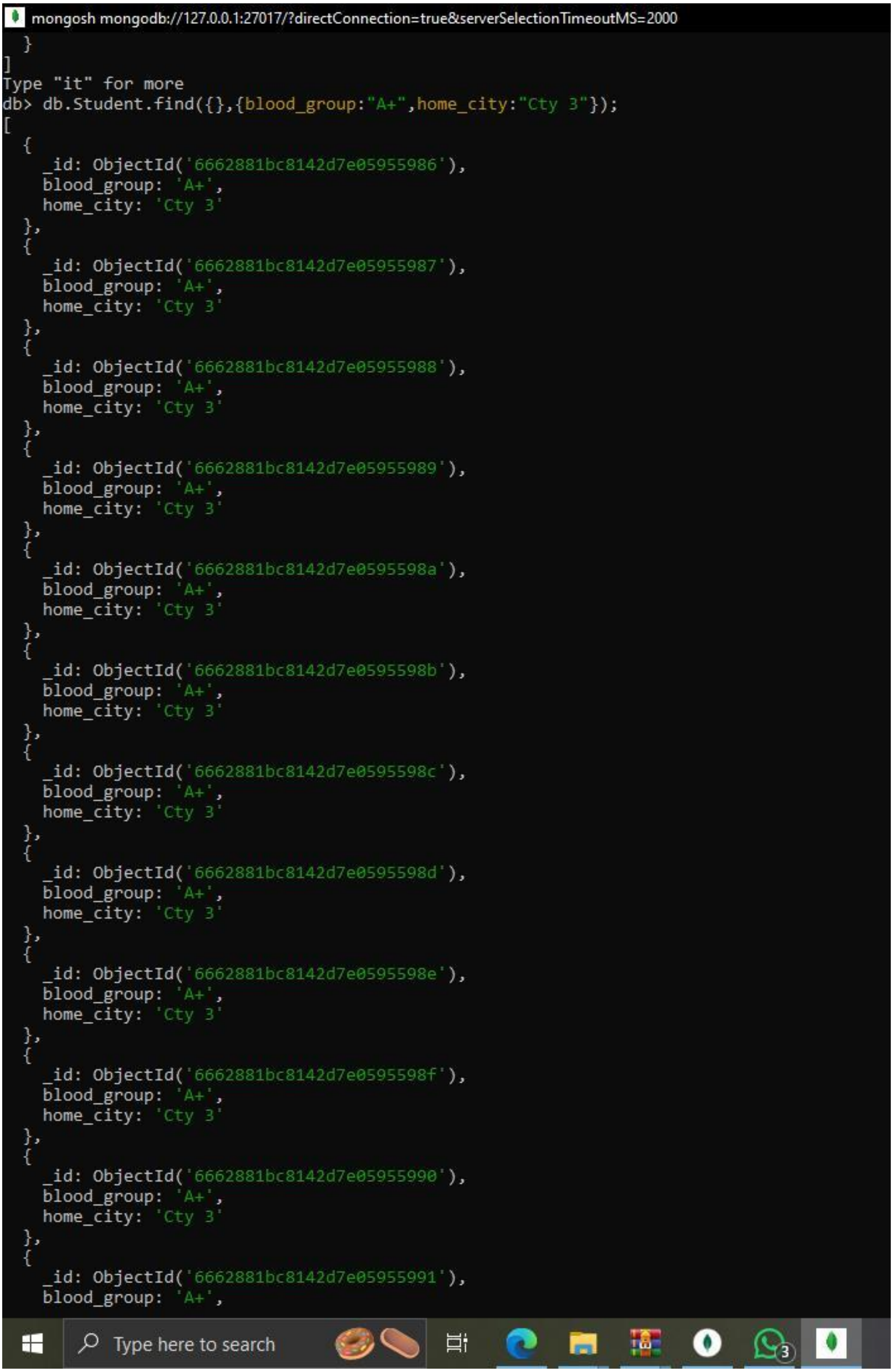
```
// Find all students who live in "City 5" AND have a blood group of "A+"
db.students.find({
  $and: [
    { home_city: "City 5" },
    { blood_group: "A+" }
  ]
});
```

OUTPUT:



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
]
Type "it" for more
db> db.Student.find({
... $and:[
... {home_city:"City 3"},
... {blood_group:"A+"}
... ]
... });
[
  {
    _id: ObjectId('6662881bc8142d7e05955997'),
    name: 'Student 172',
    age: 25,
    courses: "['English', 'History', 'Physics', 'Mathematics']",
    gpa: 2.46,
    home_city: 'City 3',
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6662881bc8142d7e059559bb'),
    name: 'Student 959',
    age: 24,
    courses: "['History', 'Computer Science']",
    gpa: 3.43,
    home_city: 'City 3',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6662881bc8142d7e05955a4b'),
    name: 'Student 918',
    age: 19,
    courses: "['Physics', 'Computer Science']",
    gpa: 3.92,
    home_city: 'City 3',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6662881bc8142d7e05955a92'),
    name: 'Student 728',
    age: 24,
    courses: "['Mathematics', 'Physics', 'English']",
    gpa: 3.95,
    home_city: 'City 3',
    blood_group: 'A+',
    is_hotel_resident: true
  }
]
db> _
```



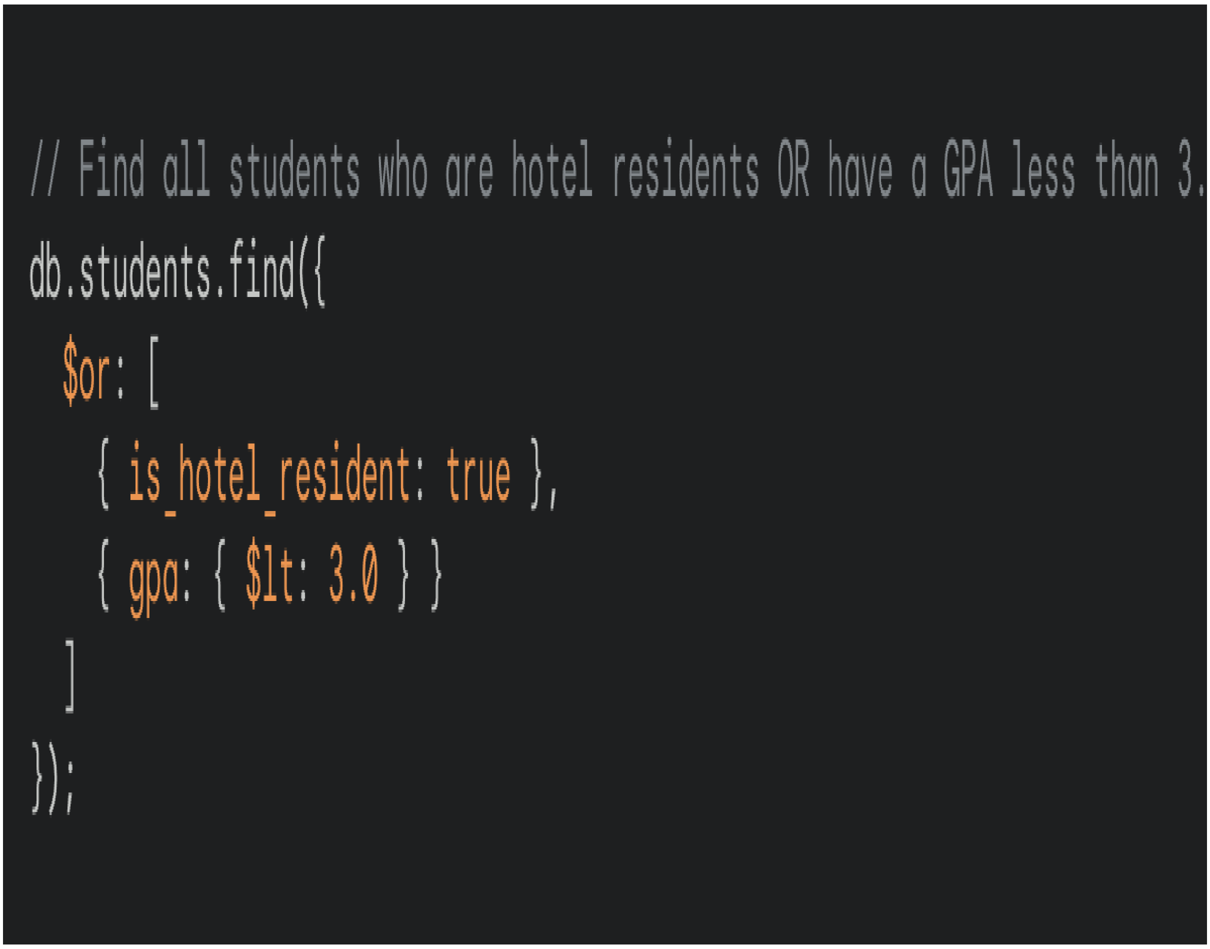
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Type here to search
3

```
  }
]
Type "it" for more
db> db.Student.find({},{blood_group:"A+",home_city:"Cty 3"});
[
  {
    _id: ObjectId('6662881bc8142d7e05955986'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e05955987'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e05955988'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e05955989'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598a'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598b'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598c'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598d'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598e'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e0595598f'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e05955990'),
    blood_group: 'A+',
    home_city: 'Cty 3'
  },
  {
    _id: ObjectId('6662881bc8142d7e05955991'),
    blood_group: 'A+',
```

OR

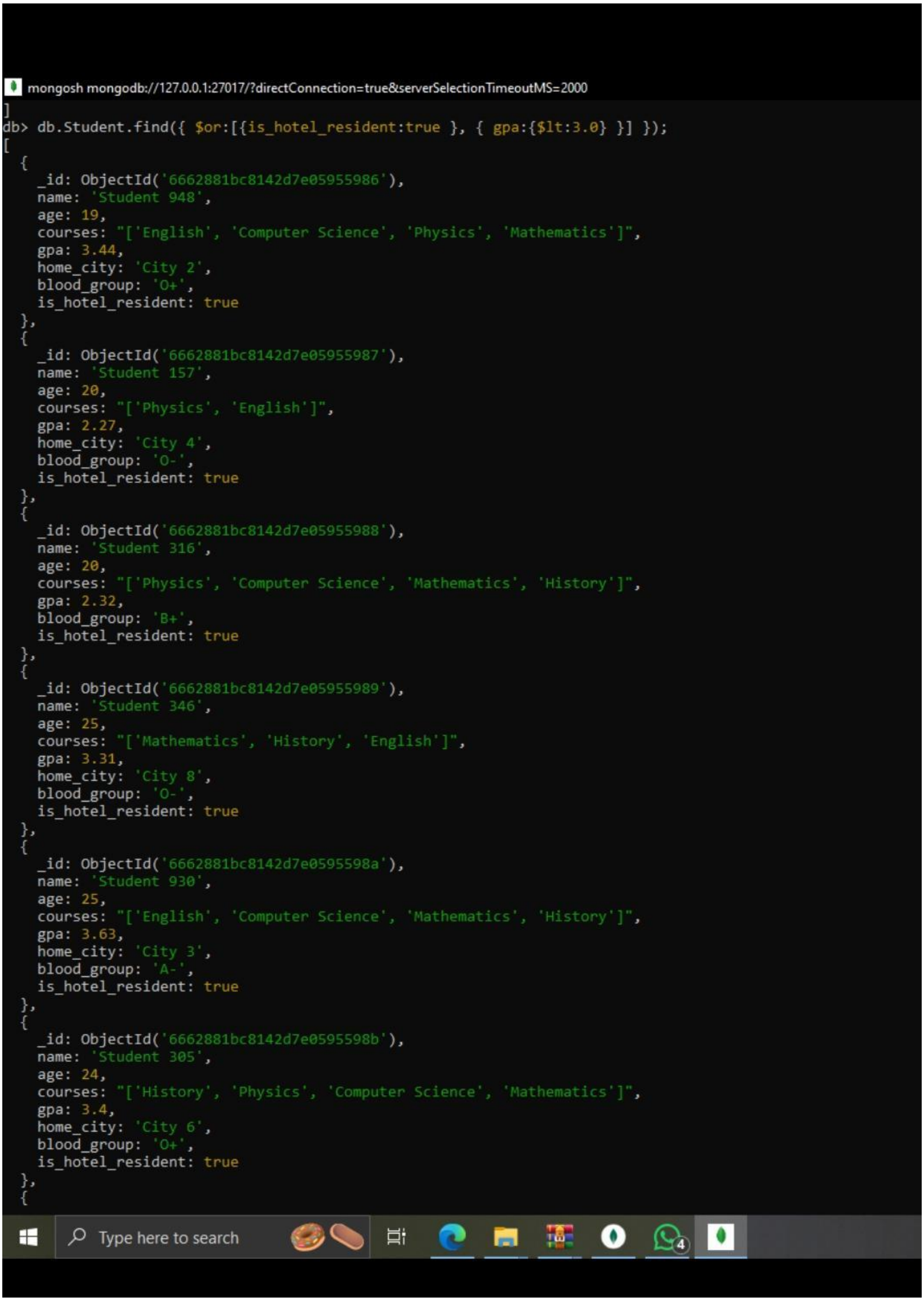
A COLLECTION WE WANT TO FILTER SUBSET BASED ON MULTIPLE CONDITION BUT ANY ONE IS SUFFICIENT

INPUT:



```
// Find all students who are hotel residents OR have a GPA less than 3.
db.students.find({
  $or: [
    { is_hotel_resident: true },
    { gpa: { $lt: 3.0 } }
  ]
});
```

OUTPUT:



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Type here to search

```
]
db> db.Student.find({ $or:[{is_hotel_resident:true }, { gpa:{$lt:3.0} }] });
[
  {
    _id: ObjectId('6662881bc8142d7e05955986'),
    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
  },
  {
    _id: ObjectId('6662881bc8142d7e05955987'),
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6662881bc8142d7e05955988'),
    name: 'Student 316',
    age: 20,
    courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
    gpa: 2.32,
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6662881bc8142d7e05955989'),
    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('6662881bc8142d7e0595598a'),
    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('6662881bc8142d7e0595598b'),
    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
  },
  {
```

CRUD

C - Create / Insert

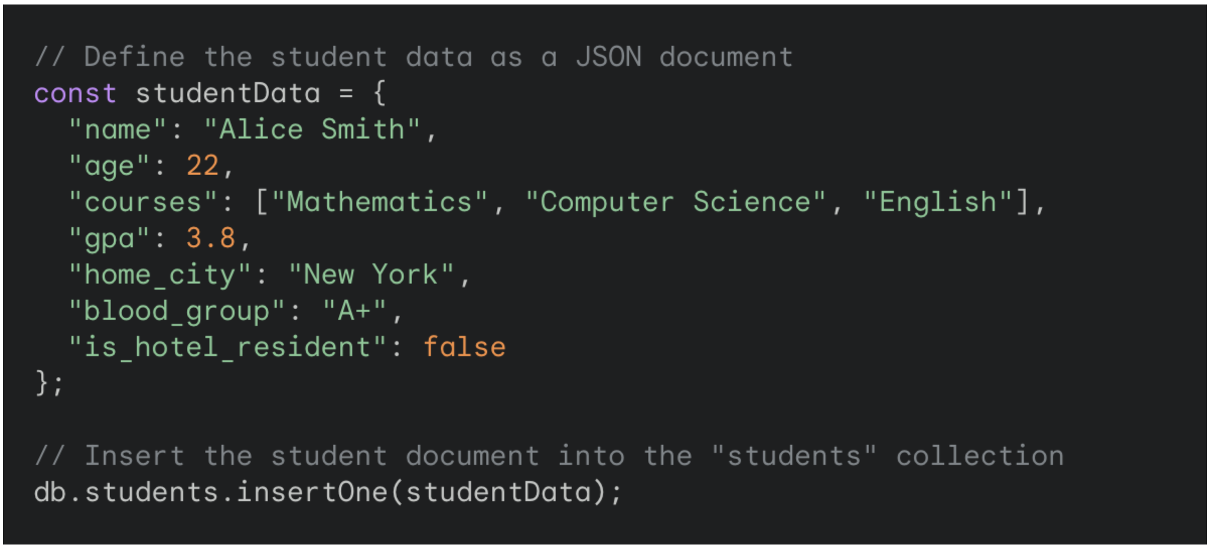
- R - Remove
- U - update
- D - Delete

This is applicable for a Collection (Table) or a Document (Row)

Insert

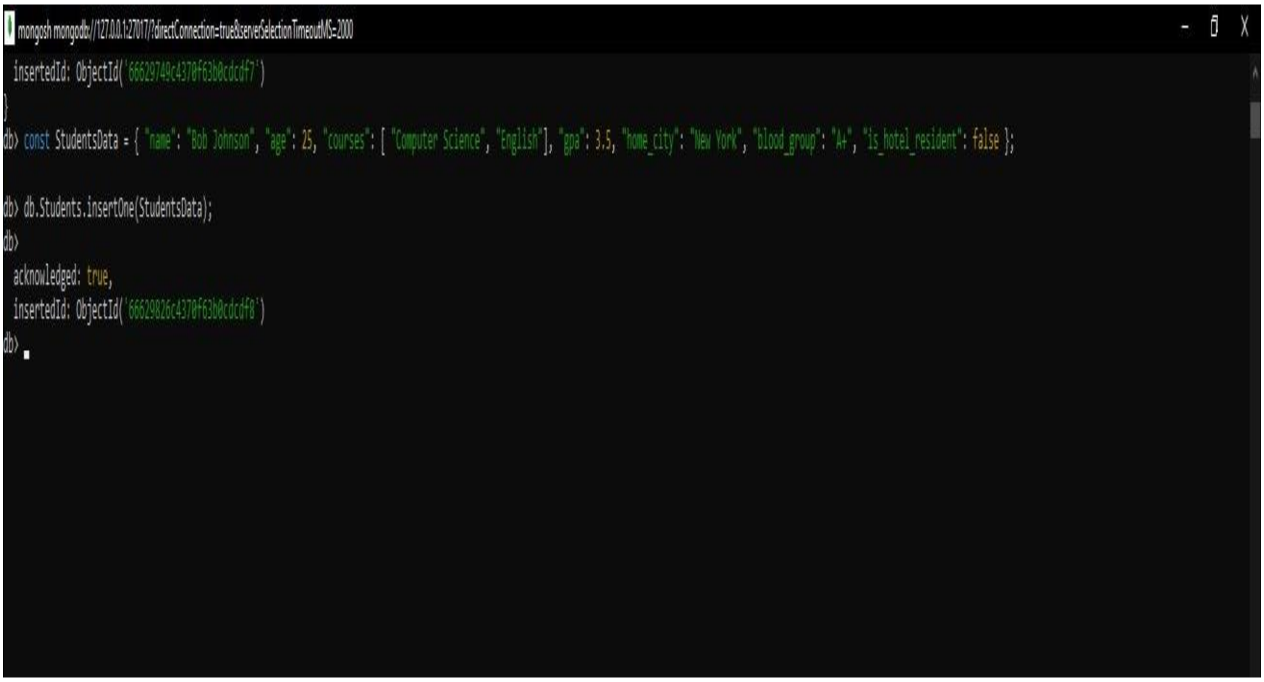
Insert is used to insert one or more document and returns a document containing the status of all inserts

Input:



```
// Define the student data as a JSON document
const studentData = {
  "name": "Alice Smith",
  "age": 22,
  "courses": ["Mathematics", "Computer Science", "English"],
  "gpa": 3.8,
  "home_city": "New York",
  "blood_group": "A+",
  "is_hotel_resident": false
};

// Insert the student document into the "students" collection
db.students.insertOne(studentData);
```

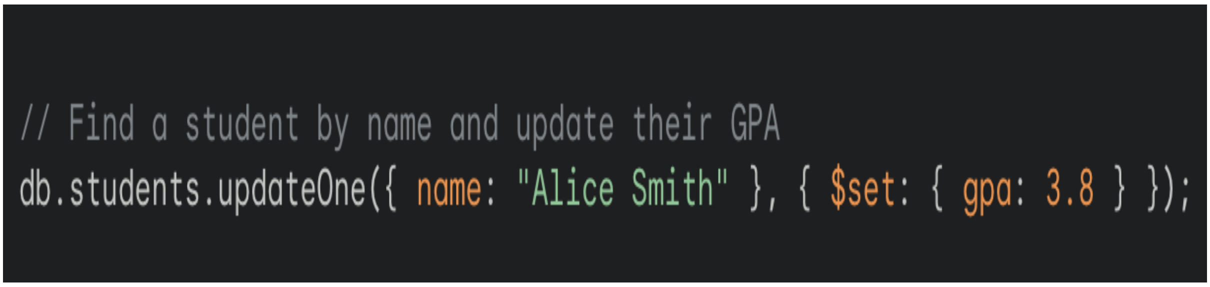
Output:


mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
 insertedId: ObjectId('66629749c4370f63b0cdcdf7')
}
db> const StudentsData = { "name": "Bob Johnson", "age": 25, "courses": [ "Computer Science", "English"], "gpa": 3.5, "home_city": "New York", "blood_group": "A+", "is_hotel_resident": false };

db> db.Students.insertOne(StudentsData);
db>
 acknowledged: true,
 insertedId: ObjectId('66629826c4370f63b0cdcdf8')
db> 
```

UPDATE:

Method used to update document into a collection update(),and save()



```
// Find a student by name and update their GPA
db.students.updateOne({ name: "Alice Smith" }, { $set: { gpa: 3.8 } });
```

Output:

db> db.Students.updateOne({name:"Alice Smith"},{$set:{gpa:3.5}});

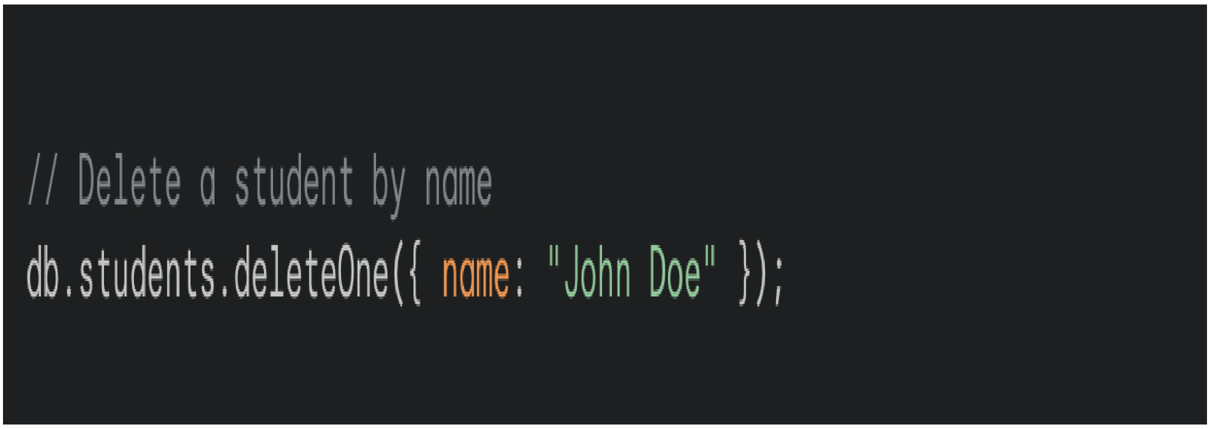
{

 acknowledged: true,

```
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Delete

Input:



```
// Delete a student by name
db.students.deleteOne({ name: "John Doe" });
```

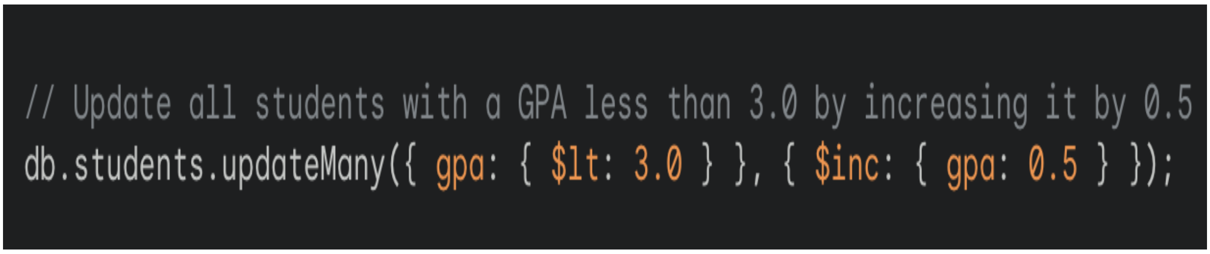
Output:

```
db> db.Students.deleteOne({name:"John Doe"});
{ acknowledged: true, deletedCount: 0 }
db>
```

UPDATE MANY:

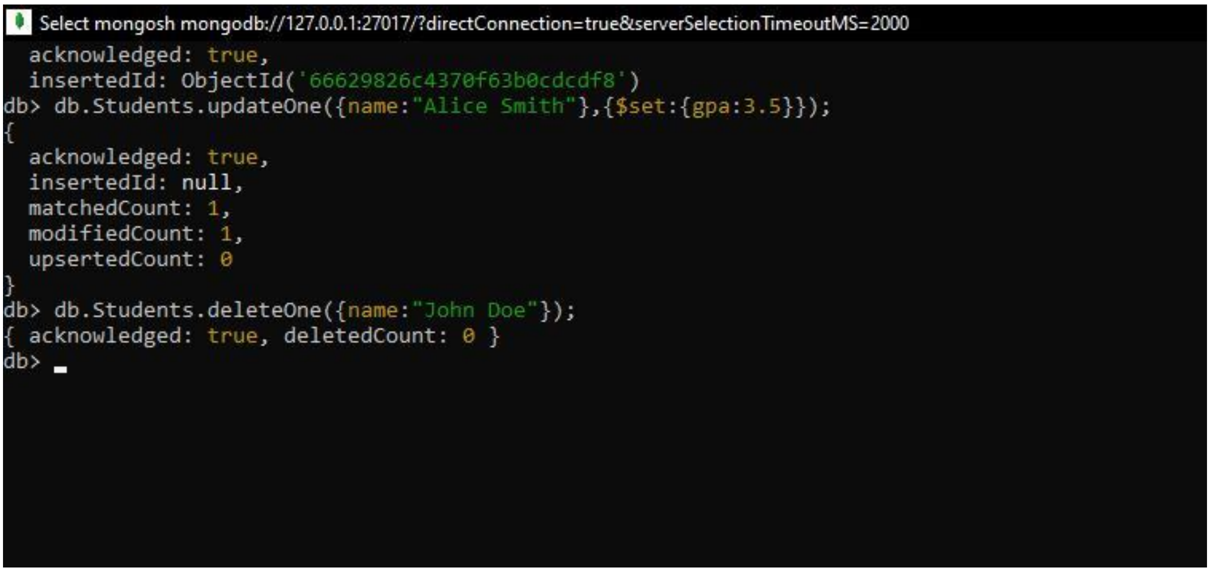
Update all the documents in mongo db collections that match the given query

Input:



```
// Update all students with a GPA less than 3.0 by increasing it by 0.5
db.students.updateMany({ gpa: { $lt: 3.0 } }, { $inc: { gpa: 0.5 } });
```

Output:



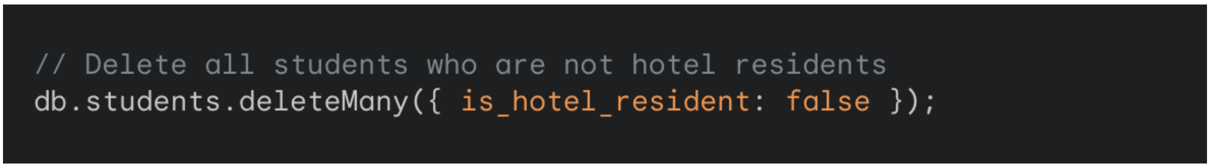
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
  acknowledged: true,
  insertedId: ObjectId('66629826c4370f63b0cdcdf8')
db> db.Students.updateOne({name:"Alice Smith"},{$set:{gpa:3.5}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
db> db.Students.deleteOne({name:"John Doe"});
{ acknowledged: true, deletedCount: 0 }
db> _
```

Delete Many

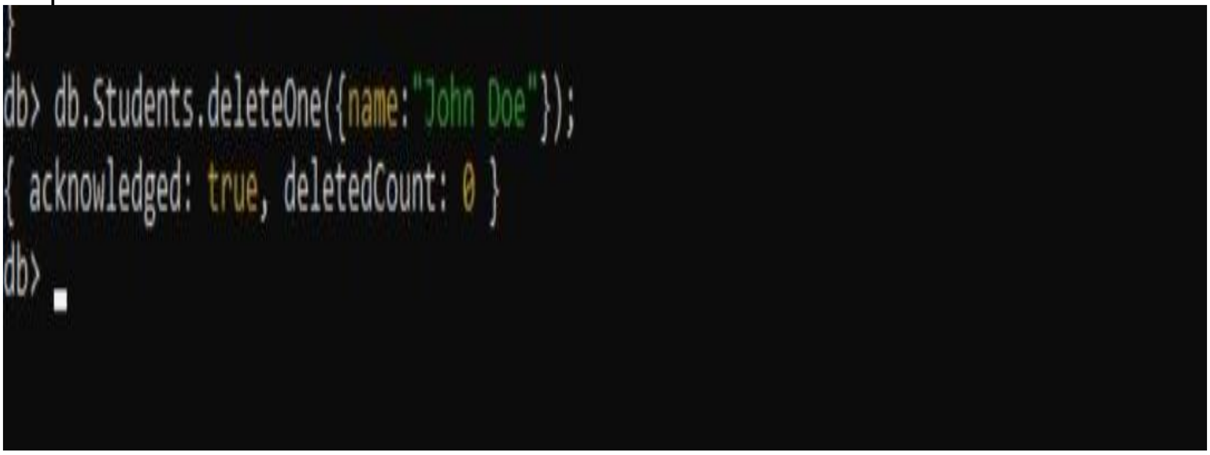
Method used to delete many multiple documents from a collection in mongo db

Input



```
// Delete all students who are not hotel residents
db.students.deleteMany({ is_hotel_resident: false });
```

Output:

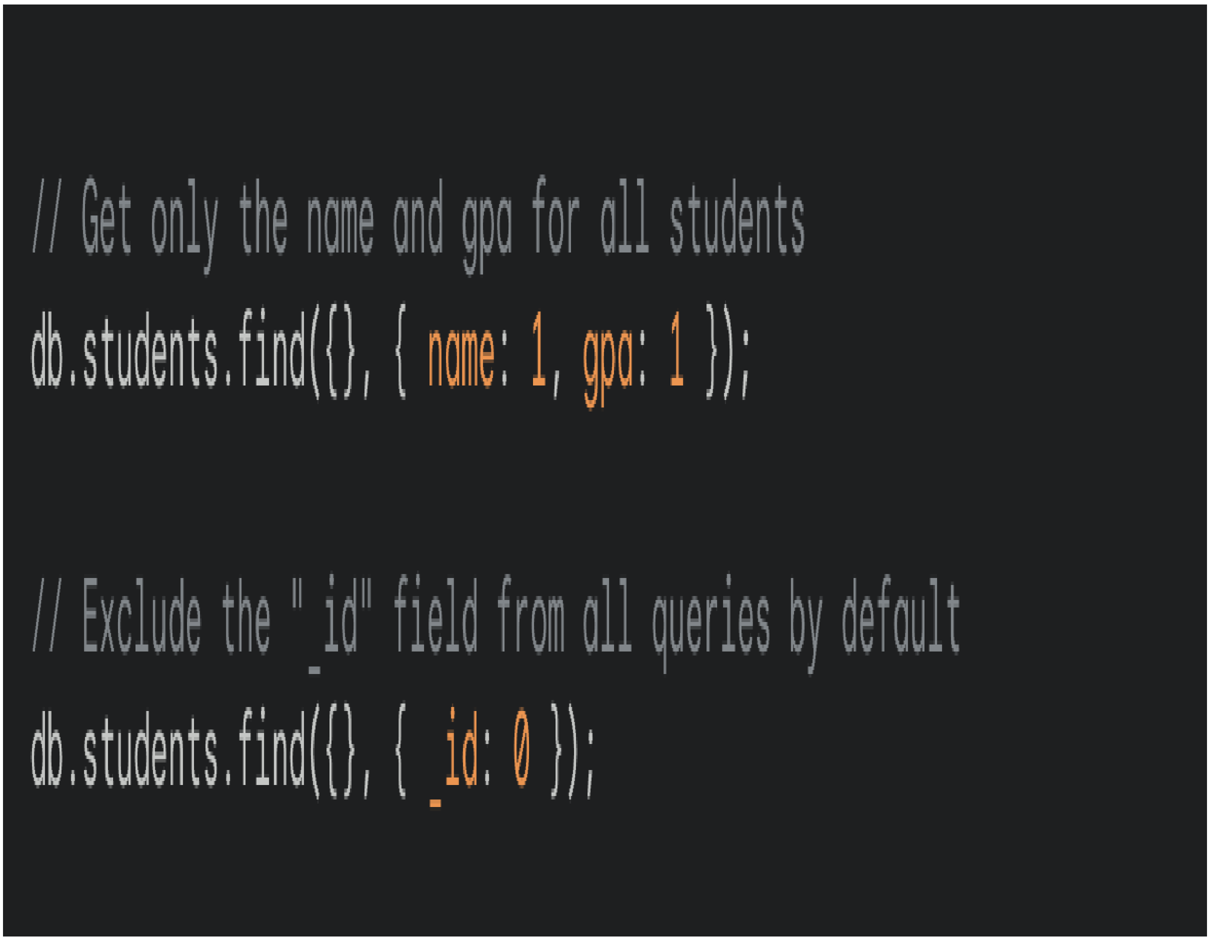


```
}
db> db.Students.deleteOne({name:"John Doe"});
{ acknowledged: true, deletedCount: 0 }
db> 
```

Projection

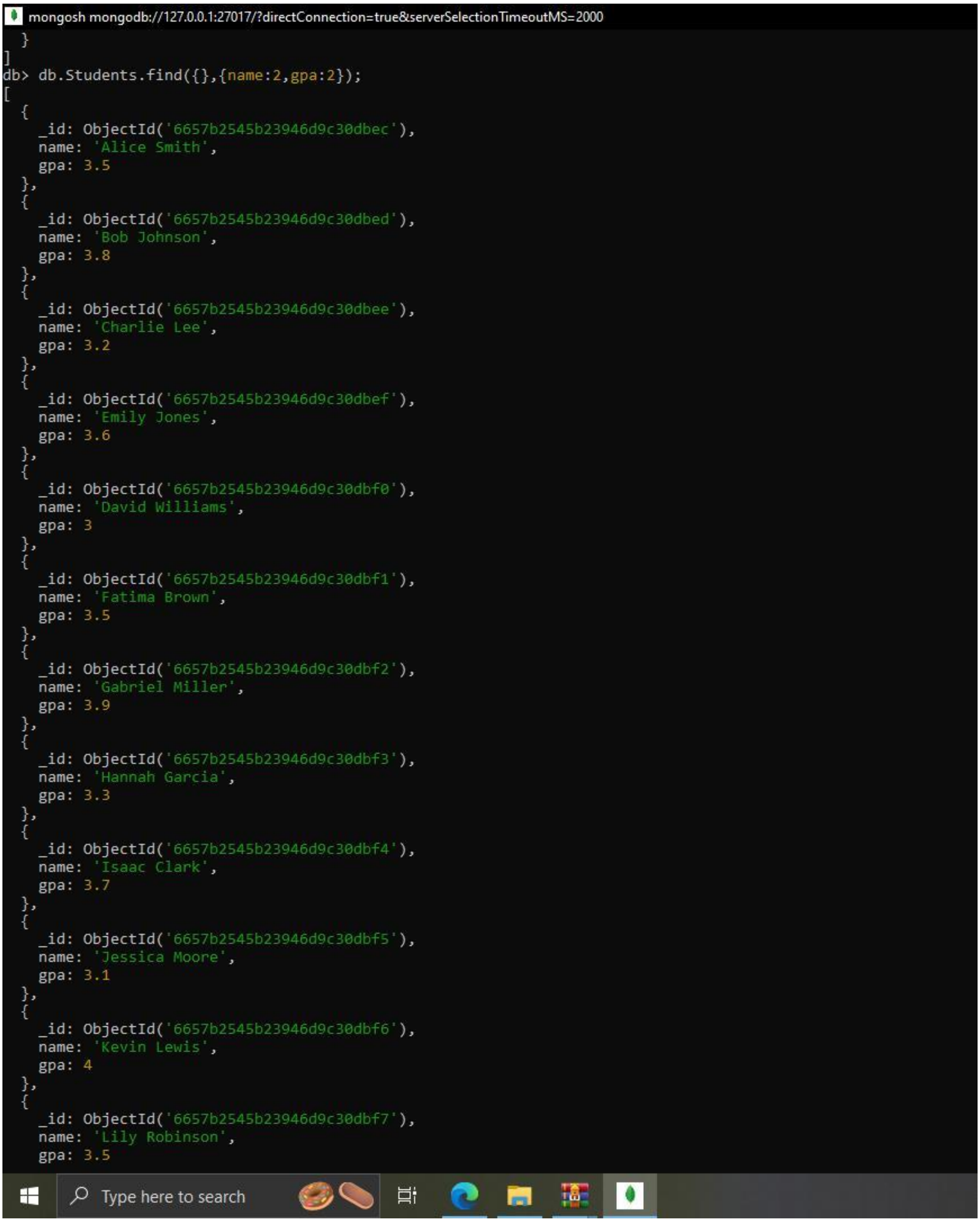
This is used when we don’t need all columns/attributes

Input



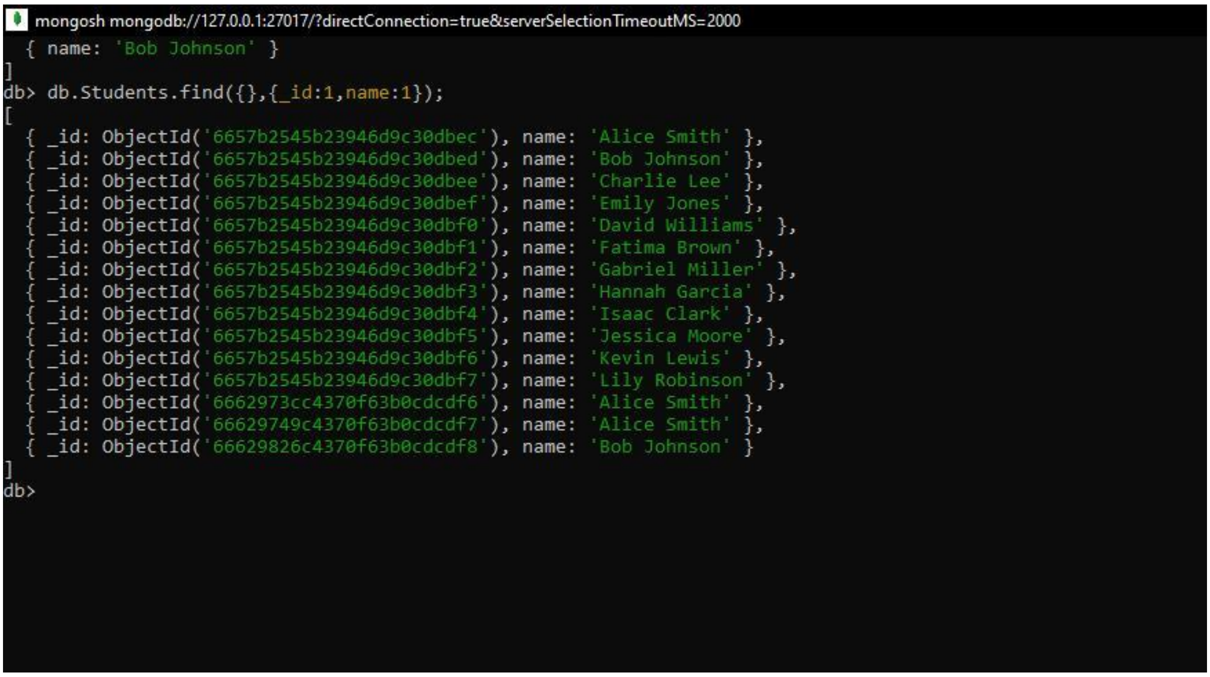
```
// Get only the name and gpa for all students
db.students.find({}, { name: 1, gpa: 1 });

// Exclude the "_id" field from all queries by default
db.students.find({}, { _id: 0 });
```

Output



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Type here to search

```
  }
]
db> db.Students.find({},{name:2,gpa:2});
[
  {
    _id: ObjectId('6657b2545b23946d9c30dbec'),
    name: 'Alice Smith',
    gpa: 3.5
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbed'),
    name: 'Bob Johnson',
    gpa: 3.8
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbee'),
    name: 'Charlie Lee',
    gpa: 3.2
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbef'),
    name: 'Emily Jones',
    gpa: 3.6
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf0'),
    name: 'David Williams',
    gpa: 3
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf1'),
    name: 'Fatima Brown',
    gpa: 3.5
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf2'),
    name: 'Gabriel Miller',
    gpa: 3.9
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf3'),
    name: 'Hannah Garcia',
    gpa: 3.3
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf4'),
    name: 'Isaac Clark',
    gpa: 3.7
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf5'),
    name: 'Jessica Moore',
    gpa: 3.1
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf6'),
    name: 'Kevin Lewis',
    gpa: 4
  },
  {
    _id: ObjectId('6657b2545b23946d9c30dbf7'),
    name: 'Lily Robinson',
    gpa: 3.5
```



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
  { name: 'Bob Johnson' }
]
db> db.Students.find({},{_id:1,name:1});
[
  { _id: ObjectId('6657b2545b23946d9c30dbec'), name: 'Alice Smith' },
  { _id: ObjectId('6657b2545b23946d9c30dbed'), name: 'Bob Johnson' },
  { _id: ObjectId('6657b2545b23946d9c30dbee'), name: 'Charlie Lee' },
  { _id: ObjectId('6657b2545b23946d9c30dbef'), name: 'Emily Jones' },
  { _id: ObjectId('6657b2545b23946d9c30dbf0'), name: 'David Williams' },
  { _id: ObjectId('6657b2545b23946d9c30dbf1'), name: 'Fatima Brown' },
  { _id: ObjectId('6657b2545b23946d9c30dbf2'), name: 'Gabriel Miller' },
  { _id: ObjectId('6657b2545b23946d9c30dbf3'), name: 'Hannah Garcia' },
  { _id: ObjectId('6657b2545b23946d9c30dbf4'), name: 'Isaac Clark' },
  { _id: ObjectId('6657b2545b23946d9c30dbf5'), name: 'Jessica Moore' },
  { _id: ObjectId('6657b2545b23946d9c30dbf6'), name: 'Kevin Lewis' },
  { _id: ObjectId('6657b2545b23946d9c30dbf7'), name: 'Lily Robinson' },
  { _id: ObjectId('6662973cc4370f63b0cdcdf6'), name: 'Alice Smith' },
  { _id: ObjectId('66629749c4370f63b0cdcdf7'), name: 'Alice Smith' },
  { _id: ObjectId('66629826c4370f63b0cdcdf8'), name: 'Bob Johnson' }
]
db>
```