mongoDB®

INTRODUCTION

What is **MongoDB?**

MongoDB is a source-available, cross-platform, document-oriented database program.
Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas .

What is database?

- A database is an organized collection of data stored in a computer system and usually controlled by a database management system (DBMS).
- The data in common databases is modeled in tables, making querying and processing efficient.

Structured Data:

- Structured data refers to data that is organized and formatted in a specific way to make it easily readable and understandable by both humans and machines.
- This is typically achieved through the use of a well-defined schema or data model, which provides a structure for the data.

Database Management System:

- A database management system (DBMS) is system software for creating and managing databases.
- A DBMS makes it possible for end users to create, protect, read, update and delete data in a database.

SET UP:

https://www.geeksforgeeks.org/how-to-install-mongodb-on-windows/?ref=ml_lbp

Few Commands to test after connections

Command	Expected Output	Notes
show dbs	admin 40.00 KiB config 72.00 KiB db 128.00 KiB local 40.00 KiB	All Databases are shown
use db	switched to db db	Connect and use db
show collections	Students	Show all tables
db.foo.insert({"bar" : "baz"})		Insert a record to collection. Create Collection if not exists

Command	Notes
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows
db.foo.remove()	Remove foo table

Documents, Collections And Datatypes

Document:

A **document** is a fundamental unit of data storage. It’s a record that contains field-and-value pairs, similar to a JSON object.

The representation of a document varies by programming language, but most languages have a data structure that is a natural fit, such as a map, hash, or dictionary.

{"greeting" : "Hello, world!"}

Collections:

A collection is a grouping of MongoDB documents. Each document within a collection can have different fields. They are analogous to tables in relational databases.

Database:

MongoDB groups collections into databases.

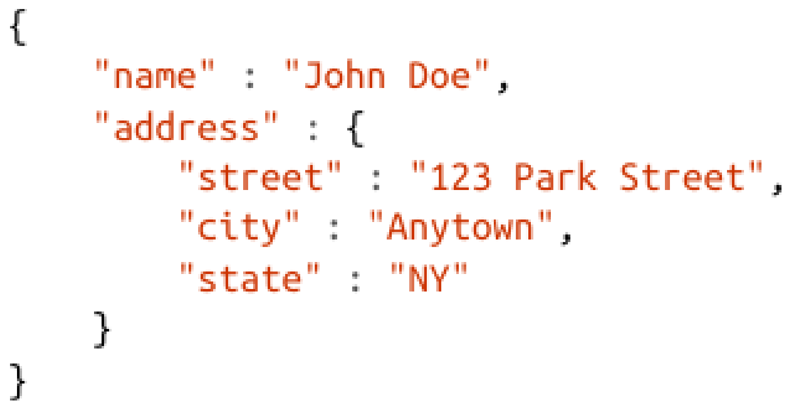
A single instance of MongoDB can host several databases, each grouping together zero or more collections.

A database has its own permissions, and each database is stored in separate files on disk.

A good rule of thumb is to store all data for a single application in the same database.

Datatype:

Basically each document will be in JSON format which will be as follows. Where each attributes inside can be of multiple data types

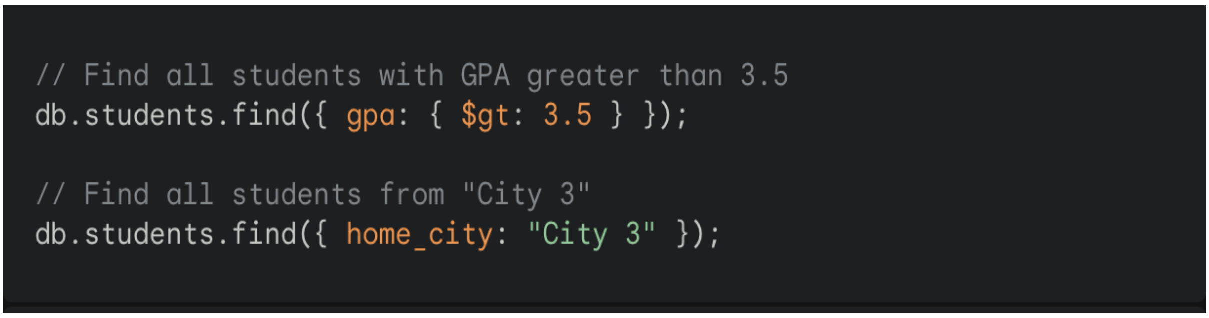


```
{
    "name" : "John Doe",
    "address" : {
        "street" : "123 Park Street",
        "city" : "Anytown",
        "state" : "NY"
    }
}
```

WHERE, AND, OR & CRUD

WHERE

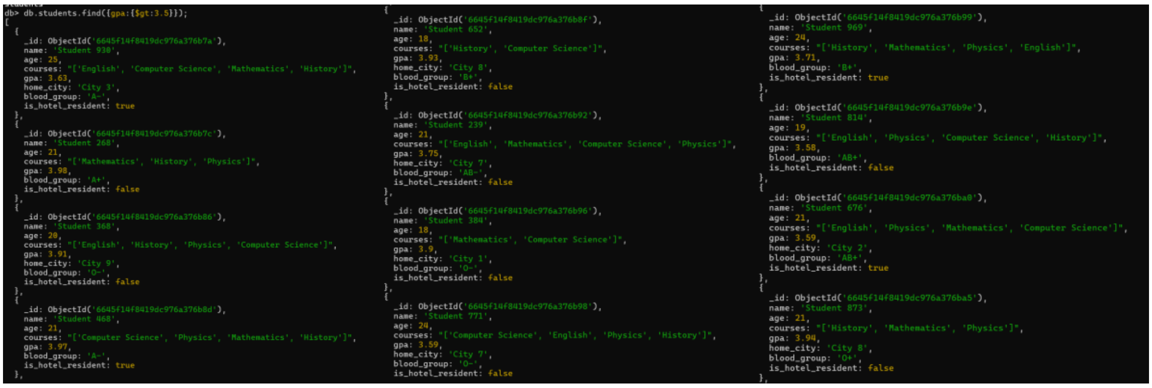
Given a Collection you want to FILTER a subset based on a condition. That is the place WHERE is used.



```
// 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" });
```

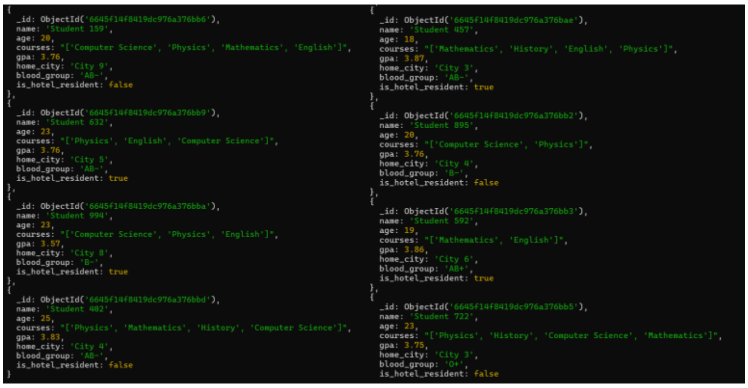
OUTPUT



```
students
db> db.students.find({gpa:{$gt:3.5}});
[
  {
    _id: ObjectId('6645f14f8419dc976a376b7a'),
    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('6645f14f8419dc976a376b7c'),
    name: 'Student 268',
    age: 21,
    courses: "['Mathematics', 'History', 'Physics']",
    gpa: 3.98,
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b86'),
    name: 'Student 368',
    age: 20,
    courses: "['English', 'History', 'Physics', 'Computer Science']",
    gpa: 3.91,
    home_city: 'City 9',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b8d'),
    name: 'Student 468',
    age: 21,
    courses: "['Computer Science', 'Physics', 'Mathematics', 'History']",
    gpa: 3.97,
    blood_group: 'A-',
    is_hotel_resident: true
  },
```

```
  {
    _id: ObjectId('6645f14f8419dc976a376b8f'),
    name: 'Student 652',
    age: 18,
    courses: "['History', 'Computer Science']",
    gpa: 3.93,
    home_city: 'City 8',
    blood_group: 'B+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b92'),
    name: 'Student 239',
    age: 21,
    courses: "['English', 'Mathematics', 'Computer Science', 'Physics']",
    gpa: 3.75,
    home_city: 'City 7',
    blood_group: 'AB-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b96'),
    name: 'Student 384',
    age: 18,
    courses: "['Mathematics', 'Computer Science']",
    gpa: 3.9,
    home_city: 'City 1',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b98'),
    name: 'Student 771',
    age: 24,
    courses: "['Computer Science', 'English', 'Physics', 'History']",
    gpa: 3.59,
    home_city: 'City 7',
    blood_group: 'O-',
    is_hotel_resident: false
```

```
  {
    _id: ObjectId('6645f14f8419dc976a376b99'),
    name: 'Student 969',
    age: 24,
    courses: "['History', 'Mathematics', 'Physics', 'English']",
    gpa: 3.71,
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b9e'),
    name: 'Student 814',
    age: 19,
    courses: "['English', 'Physics', 'Computer Science', 'History']",
    gpa: 3.58,
    blood_group: 'AB+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376ba0'),
    name: 'Student 676',
    age: 21,
    courses: "['English', 'Physics', 'Mathematics', 'Computer Science']",
    gpa: 3.59,
    home_city: 'City 2',
    blood_group: 'AB+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376ba5'),
    name: 'Student 873',
    age: 21,
    courses: "['History', 'Mathematics', 'Physics']",
    gpa: 3.94,
    home_city: 'City 8',
    blood_group: 'O+',
    is_hotel_resident: false
  },
```



```
{
  _id: ObjectId('6645f14f8419dc976a376bb6'),
  name: 'Student 159',
  age: 20,
  courses: "['Computer Science', 'Physics', 'Mathematics', 'English']",
  gpa: 3.76,
  home_city: 'City 9',
  blood_group: 'AB-',
  is_hotel_resident: false
},
{
  _id: ObjectId('6645f14f8419dc976a376bb9'),
  name: 'Student 632',
  age: 23,
  courses: "['Physics', 'English', 'Computer Science']",
  gpa: 3.76,
  home_city: 'City 5',
  blood_group: 'AB-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376bba'),
  name: 'Student 994',
  age: 23,
  courses: "['Computer Science', 'Physics', 'English']",
  gpa: 3.57,
  home_city: 'City 8',
  blood_group: 'B-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376bbd'),
  name: 'Student 402',
  age: 25,
  courses: "['Physics', 'Mathematics', 'History', 'Computer Science']",
  gpa: 3.83,
  home_city: 'City 4',
  blood_group: 'AB-',
  is_hotel_resident: false
}
```

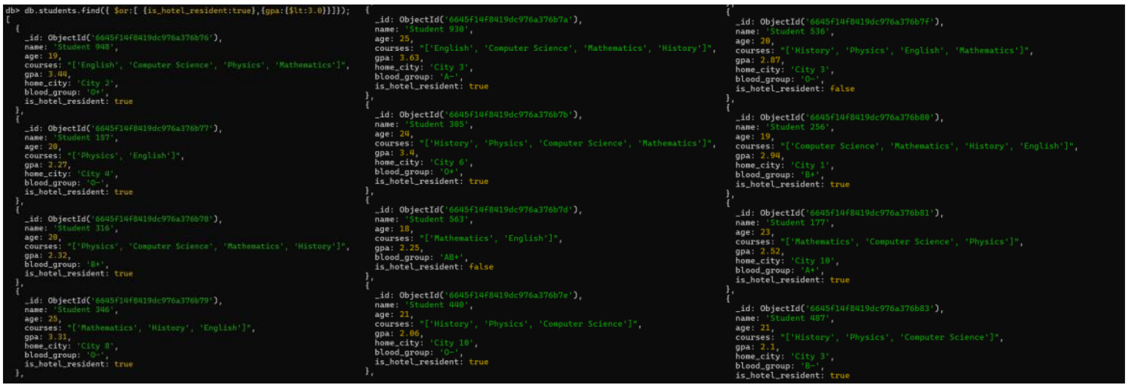
```
  {
    _id: ObjectId('6645f14f8419dc976a376bae'),
    name: 'Student 457',
    age: 18,
    courses: "['Mathematics', 'History', 'English', 'Physics']",
    gpa: 3.87,
    home_city: 'City 3',
    blood_group: 'AB-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376bb2'),
    name: 'Student 895',
    age: 20,
    courses: "['Computer Science', 'Physics']",
    gpa: 3.76,
    home_city: 'City 4',
    blood_group: 'B-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376bb3'),
    name: 'Student 592',
    age: 19,
    courses: "['Mathematics', 'English']",
    gpa: 3.86,
    home_city: 'City 6',
    blood_group: 'AB+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376bb5'),
    name: 'Student 722',
    age: 23,
    courses: "['Physics', 'History', 'Computer Science', 'Mathematics']",
    gpa: 3.75,
    home_city: 'City 3',
    blood_group: 'O+',
    is_hotel_resident: false
```

OR

The `$or` operator is used to specify a compound query with multiple conditions, where at least one condition must be satisfied for a document to match.

```
// 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 } }
  ]
});
```

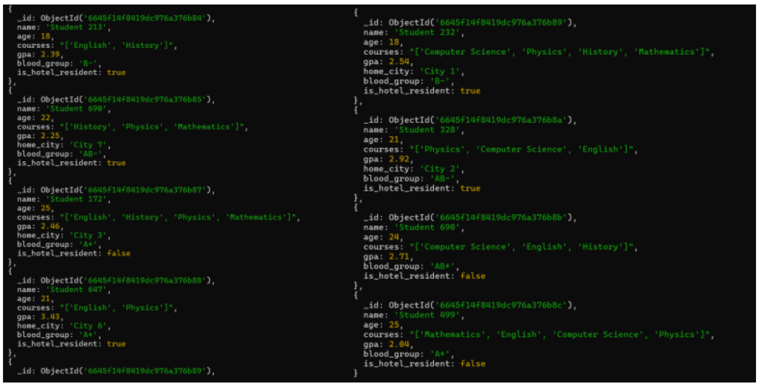
OUTPUT



```
db> db.students.find({ $or:[ {is_hotel_resident:true},{gpa:{$lt:3.0}}]});
[
  {
    _id: ObjectId('6645f14f8419dc976a376b76'),
    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('6645f14f8419dc976a376b77'),
    name: 'Student 157',
    age: 20,
    courses: "['Physics', 'English']",
    gpa: 2.27,
    home_city: 'City 4',
    blood_group: 'O-',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b78'),
    name: 'Student 316',
    age: 20,
    courses: "['Physics', 'Computer Science', 'Mathematics', 'History']",
    gpa: 2.32,
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b79'),
    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('6645f14f8419dc976a376b7a'),
    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('6645f14f8419dc976a376b7b'),
    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('6645f14f8419dc976a376b7d'),
    name: 'Student 563',
    age: 18,
    courses: "['Mathematics', 'English']",
    gpa: 2.25,
    blood_group: 'AB+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b7e'),
    name: 'Student 440',
    age: 21,
    courses: "['History', 'Physics', 'Computer Science']",
    gpa: 2.06,
    home_city: 'City 10',
    blood_group: 'O-',
    is_hotel_resident: true
  },
```

```
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b7f'),
    name: 'Student 536',
    age: 20,
    courses: "['History', 'Physics', 'English', 'Mathematics']",
    gpa: 2.87,
    home_city: 'City 3',
    blood_group: 'O-',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b80'),
    name: 'Student 256',
    age: 19,
    courses: "['Computer Science', 'Mathematics', 'History', 'English']",
    gpa: 2.94,
    home_city: 'City 1',
    blood_group: 'B+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b81'),
    name: 'Student 177',
    age: 23,
    courses: "['Mathematics', 'Computer Science', 'Physics']",
    gpa: 2.52,
    home_city: 'City 10',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376b83'),
    name: 'Student 487',
    age: 21,
    courses: "['History', 'Physics', 'Computer Science']",
    gpa: 2.1,
    home_city: 'City 3',
    blood_group: 'B-',
    is_hotel_resident: true
```

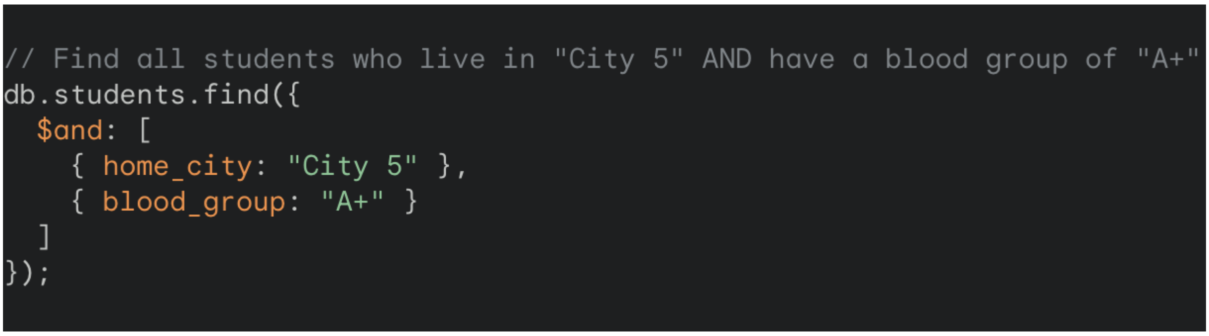


```
{
  _id: ObjectId('6645f14f8419dc976a376b84'),
  name: 'Student 213',
  age: 18,
  courses: "['English', 'History']",
  gpa: 2.39,
  blood_group: 'B-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376b85'),
  name: 'Student 690',
  age: 22,
  courses: "['History', 'Physics', 'Mathematics']",
  gpa: 2.25,
  home_city: 'City 7',
  blood_group: 'AB-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376b87'),
  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('6645f14f8419dc976a376b88'),
  name: 'Student 647',
  age: 21,
  courses: "['English', 'Physics']",
  gpa: 3.43,
  home_city: 'City 6',
  blood_group: 'A+',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376b89'),
```

```
{
  _id: ObjectId('6645f14f8419dc976a376b89'),
  name: 'Student 232',
  age: 18,
  courses: "['Computer Science', 'Physics', 'History', 'Mathematics']",
  gpa: 2.54,
  home_city: 'City 1',
  blood_group: 'B-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376b8a'),
  name: 'Student 328',
  age: 21,
  courses: "['Physics', 'Computer Science', 'English']",
  gpa: 2.92,
  home_city: 'City 2',
  blood_group: 'AB-',
  is_hotel_resident: true
},
{
  _id: ObjectId('6645f14f8419dc976a376b8b'),
  name: 'Student 690',
  age: 24,
  courses: "['Computer Science', 'English', 'History']",
  gpa: 2.71,
  blood_group: 'AB+',
  is_hotel_resident: false
},
{
  _id: ObjectId('6645f14f8419dc976a376b8c'),
  name: 'Student 499',
  age: 25,
  courses: "['Mathematics', 'English', 'Computer Science', 'Physics']",
  gpa: 2.04,
  blood_group: 'A+',
  is_hotel_resident: false
}
```

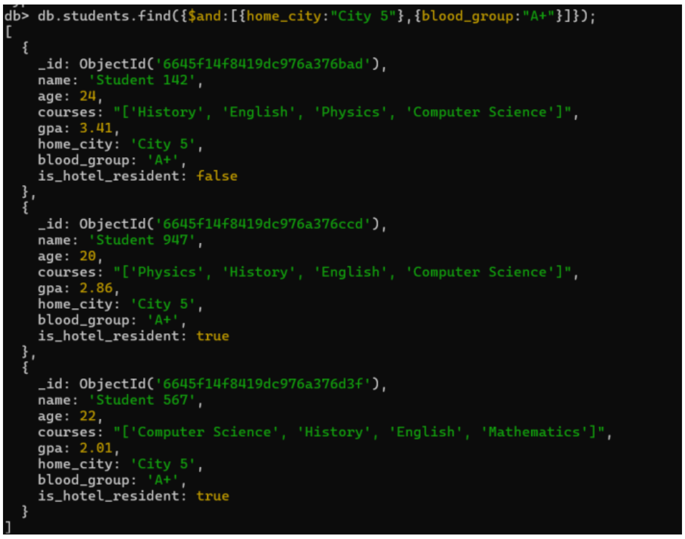
AND

The $and operator allows you to specify multiple conditions that documents must satisfy to match the query.



```
// 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



```
db> db.students.find({$and:[{home_city:"City 5"},{blood_group:"A+"}]});
[
  {
    _id: ObjectId('6645f14f8419dc976a376bad'),
    name: 'Student 142',
    age: 24,
    courses: "['History', 'English', 'Physics', 'Computer Science']",
    gpa: 3.41,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: false
  },
  {
    _id: ObjectId('6645f14f8419dc976a376ccd'),
    name: 'Student 947',
    age: 20,
    courses: "['Physics', 'History', 'English', 'Computer Science']",
    gpa: 2.86,
    home_city: 'City 5',
    blood_group: 'A+',
    is_hotel_resident: true
  },
  {
    _id: ObjectId('6645f14f8419dc976a376d3f'),
    name: 'Student 567',
    age: 22,
    courses: "['Computer Science', 'History', 'English', 'Mathematics']",
    gpa: 2.01,
    home_city: 'City 5',
    blood_group: 'A+',
    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)