### 24/04/2024

### Day02-MongoDB

### 1) Insert One Student Record

**Explanation:**  
This command adds one student (Amina) with details like registration number, name, age, degree, GPA, gender, and skills into the database.

db.students.insertOne(

{

"regno": "2021ict127",

"name": "Amina",

"age": 24,

"degree": "Computer Science",

"gpa": 3.9,

"gender": "female",

"skills": ["Python", "MongoDB", "HTML"]

}

)

### 2) Insert Many Student Records

**Explanation:**  
This command adds multiple students (Ravi, Meera, Zayan, Fatima) at once into the database with all their information.

db.students.insertMany([

{

"regno": "2021ict128",

"name": "Ravi",

"age": 26,

"degree": "Software Engineering",

"gpa": 3.5,

"gender": "male",

"skills": ["C++", "Java", "Git"]

},

{

"regno": "2021ict129",

"name": "Meera",

"age": 23,

"degree": "Information Systems",

"gpa": 3.8,

"gender": "female",

"skills": ["SQL", "Excel", "PowerBI"]

},

{

"regno": "2021ict130",

"name": "Zayan",

"age": 25,

"degree": "IT",

"gpa": 3.6,

"gender": "male",

"skills": ["React", "Node.js", "Docker"]

},

{

"regno": "2021ict131",

"name": "Fatima",

"age": 22,

"degree": "Computer Engineering",

"gpa": 3.85,

"gender": "female",

"skills": ["C#", "Unity", "CSS"]

}

])

3)Console queries and outputs:

#### 3.1 Find All Female Students

**Explanation:**  
This command shows all students whose gender is "female" from the database.

db.students.find({gender:"female"});

{

\_id: ObjectId('680bc7729f9f08d0a9e9d6a1'),

regno: '2021ict127',

name: 'Amina',

age: 24,

degree: 'Computer Science',

gpa: 3.9,

gender: 'female',

skills: [

'Python',

'MongoDB',

'HTML'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a3'),

regno: '2021ict129',

name: 'Meera',

age: 23,

degree: 'Information Systems',

gpa: 3.8,

gender: 'female',

skills: [

'SQL',

'Excel',

'PowerBI'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a5'),

regno: '2021ict131',

name: 'Fatima',

age: 22,

degree: 'Computer Engineering',

gpa: 3.85,

gender: 'female',

skills: [

'C#',

'Unity',

'CSS'

]

}

#### 3.2 Find One Female Student

**Explanation:**  
This command shows only **one** female student from the database.

db.students.findOne({gender:"female"});

{

\_id: ObjectId('680bc7729f9f08d0a9e9d6a1'),

regno: '2021ict127',

name: 'Amina',

age: 24,

degree: 'Computer Science',

gpa: 3.9,

gender: 'female',

skills: [

'Python',

'MongoDB',

'HTML'

]

}

#### 3.3 Find Students Older than 25

**Explanation:**  
This command displays students who are **older than 25 years**.

db.students.find({age:{$gt:25}});

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a2'),

regno: '2021ict128',

name: 'Ravi',

age: 26,

degree: 'Software Engineering',

gpa: 3.5,

gender: 'male',

skills: [

'C++',

'Java',

'Git'

]

}

#### 3.4 Sort Students by GPA (High to Low)

**Explanation:**  
This command shows all students, arranged by their GPA in **descending order** (highest GPA first).

db.students.find().sort({gpa:-1});

\_id: ObjectId('680bc7729f9f08d0a9e9d6a1'),

regno: '2021ict127',

name: 'Amina',

age: 24,

degree: 'Computer Science',

gpa: 3.9,

gender: 'female',

skills: [

'Python',

'MongoDB',

'HTML'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a5'),

regno: '2021ict131',

name: 'Fatima',

age: 22,

degree: 'Computer Engineering',

gpa: 3.85,

gender: 'female',

skills: [

'C#',

'Unity',

'CSS'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a3'),

regno: '2021ict129',

name: 'Meera',

age: 23,

degree: 'Information Systems',

gpa: 3.8,

gender: 'female',

skills: [

'SQL',

'Excel',

'PowerBI'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a3'),

regno: '2021ict129',

name: 'Meera',

age: 23,

degree: 'Information Systems',

gpa: 3.8,

gender: 'female',

skills: [

'SQL',

'Excel',

'PowerBI'

]

}

{

\_id: ObjectId('680bbd1386c2c061683da842'),

regno: '2021ict126',

name: 'Shahla',

age: 25,

degree: 'IT',

gpa: 3.7,

skills: [

'JS',

'MathLab',

'NoSql'

]

}

{

\_id: ObjectId('680bbe1b86c2c061683da843'),

regno: '2021ict126',

name: 'Shahla',

age: 25,

degree: 'IT',

gpa: 3.7,

gender: 'male',

skills: [

'JS',

'MathLab',

'NoSql'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a4'),

regno: '2021ict130',

name: 'Zayan',

age: 25,

degree: 'IT',

gpa: 3.6,

gender: 'male',

skills: [

'React',

'Node.js',

'Docker'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a2'),

regno: '2021ict128',

name: 'Ravi',

age: 26,

degree: 'Software Engineering',

gpa: 3.5,

gender: 'male',

skills: [

'C++',

'Java',

'Git'

]

}

#### 3.5 Find IT Degree Students Sorted by GPA

**Explanation:**  
This command finds students who are studying "IT" and arranges them by GPA from highest to lowest.

db.students.find({'degree':"IT"}).sort({gpa:-1});

{

\_id: ObjectId('680bbd1386c2c061683da842'),

regno: '2021ict126',

name: 'Shahla',

age: 25,

degree: 'IT',

gpa: 3.7,

skills: [

'JS',

'MathLab',

'NoSql'

]

}

{

\_id: ObjectId('680bbe1b86c2c061683da843'),

regno: '2021ict126',

name: 'Shahla',

age: 25,

degree: 'IT',

gpa: 3.7,

gender: 'male',

skills: [

'JS',

'MathLab',

'NoSql'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a4'),

regno: '2021ict130',

name: 'Zayan',

age: 25,

degree: 'IT',

gpa: 3.6,

gender: 'male',

skills: [

'React',

'Node.js',

'Docker'

]

}

#### 3.6 Find Male Students Sorted by Age (High to Low)

**Explanation:**  
This command shows only male students and sorts them by age in **descending order** (oldest first).

db.students.find({'gender':"male"}).sort({age:-1});

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a2'),

regno: '2021ict128',

name: 'Ravi',

age: 26,

degree: 'Software Engineering',

gpa: 3.5,

gender: 'male',

skills: [

'C++',

'Java',

'Git'

]

}

{

\_id: ObjectId('680bbe1b86c2c061683da843'),

regno: '2021ict126',

name: 'Shahla',

age: 25,

degree: 'IT',

gpa: 3.7,

gender: 'male',

skills: [

'JS',

'MathLab',

'NoSql'

]

}

{

\_id: ObjectId('680bc7839f9f08d0a9e9d6a4'),

regno: '2021ict130',

name: 'Zayan',

age: 25,

degree: 'IT',

gpa: 3.6,

gender: 'male',

skills: [

'React',

'Node.js',

'Docker'

]

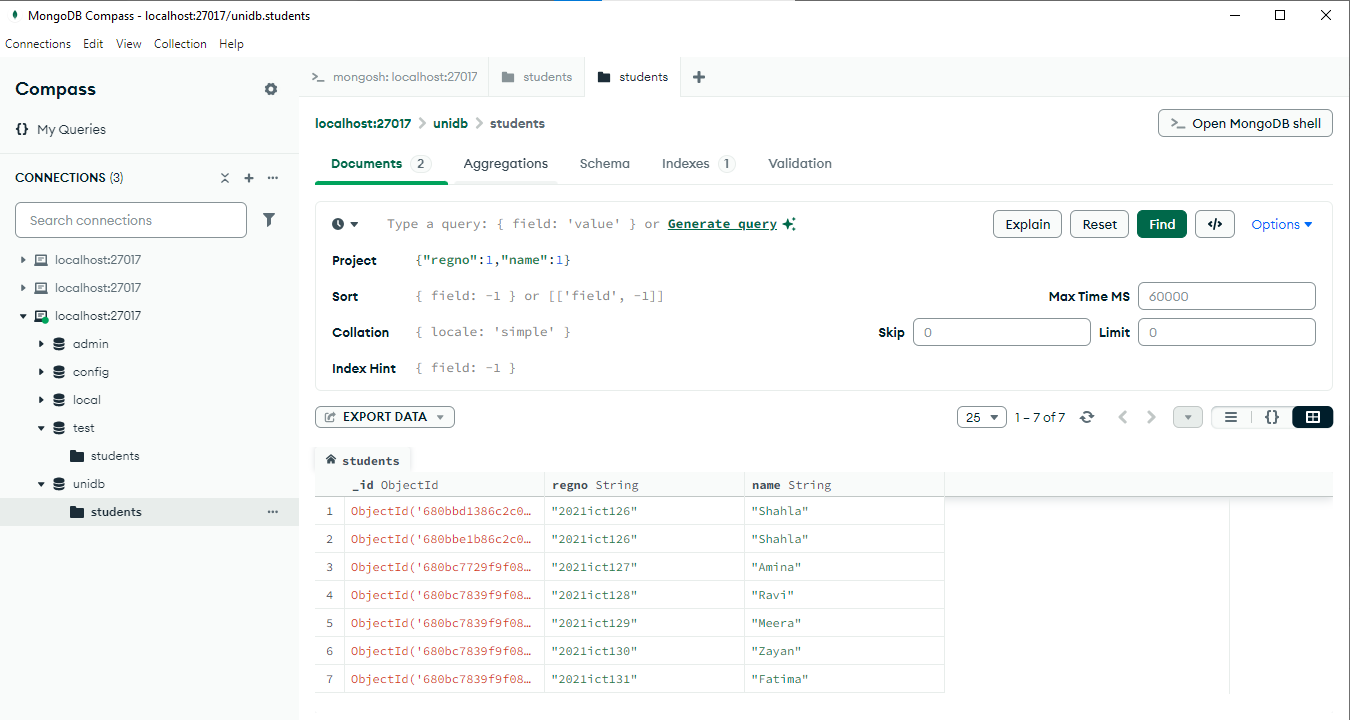
}

4)Normal queries

#### 4.1 Project Name and Age

**Explanation:**  
This selects and shows only the name and age of each student.

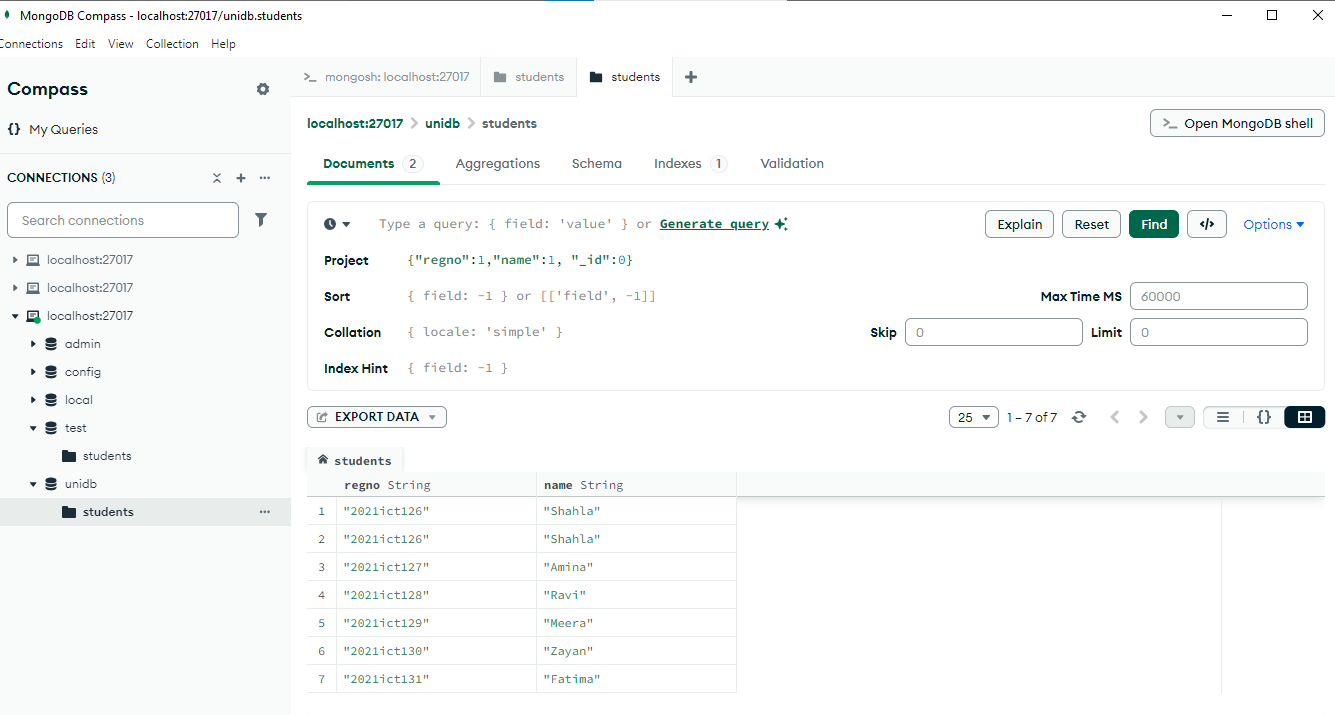
{project: {name: 1,age: 1}}



#### 4.2 Project Name and Age Without ID

**Explanation:**  
This shows the name and age of students but hides their \_id field.

{project: {name: 1,age: 1,\_id: 0}}

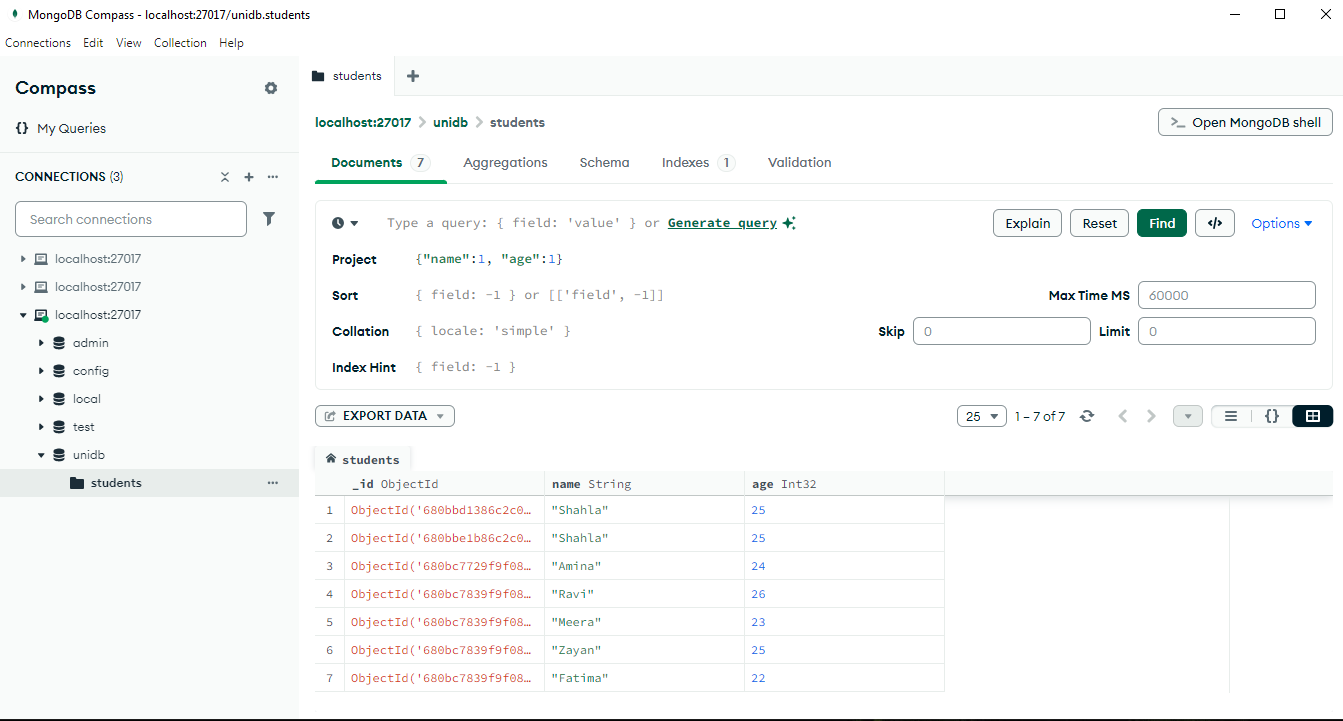


#### 4.3 Select Name and Age Only

**Explanation:**  
Another way to show just the name and age fields.

{“name”:1, “age”:1}

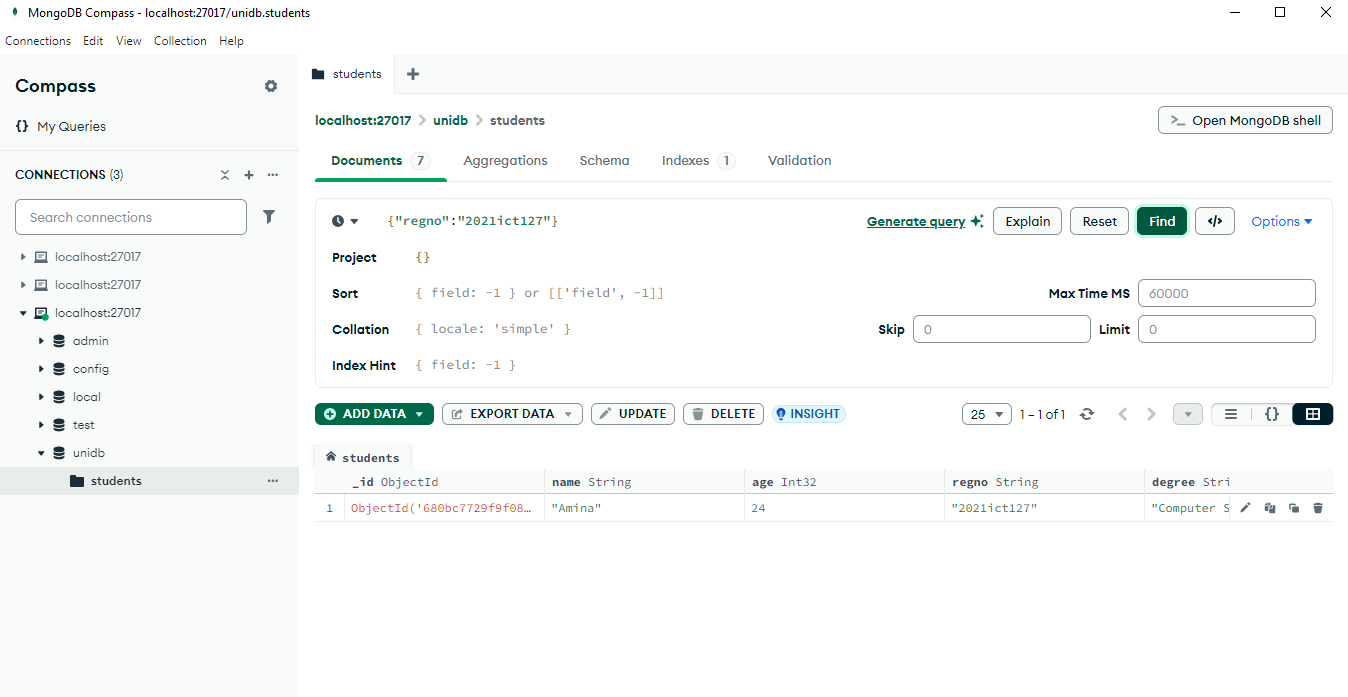
z



#### 4.4 Find Student by Registration Number

**Explanation:**  
This finds the student with registration number 2021ict127.

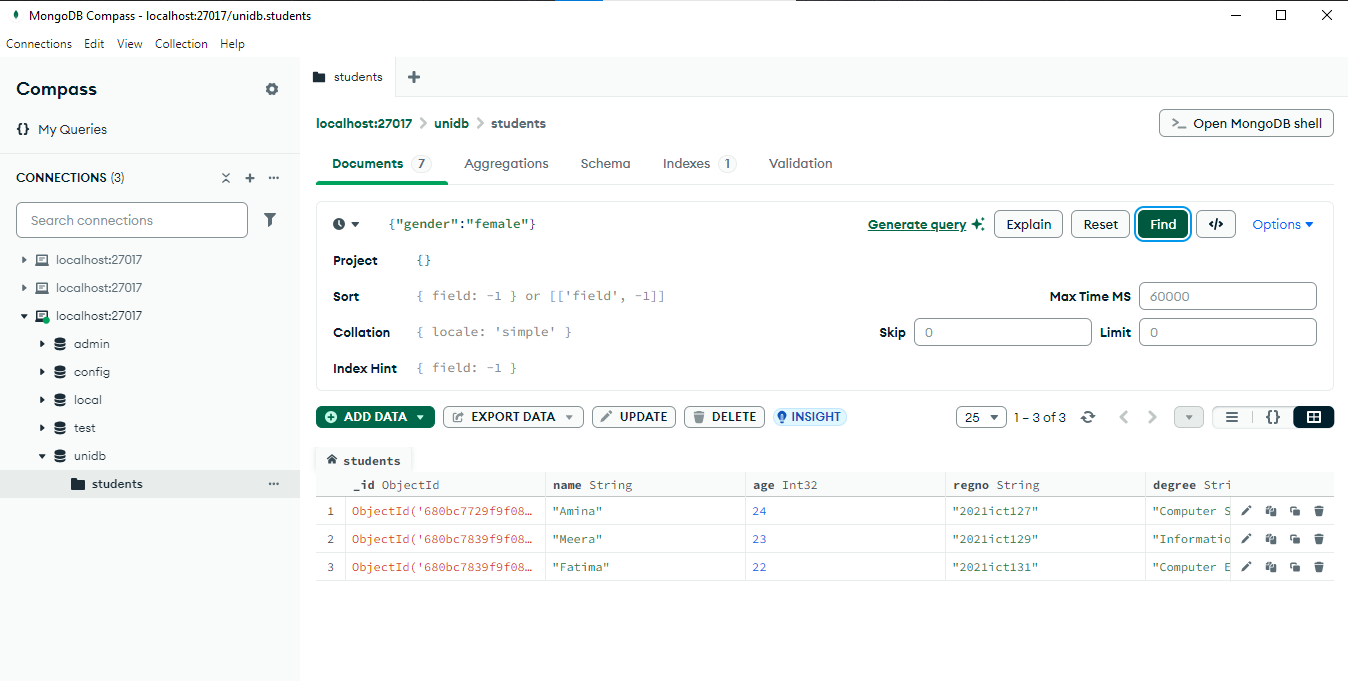
{regno:”2021ict127”}



#### 4.5 Find Students by Gender

**Explanation:**  
This shows all students with a specific gender (e.g., female).

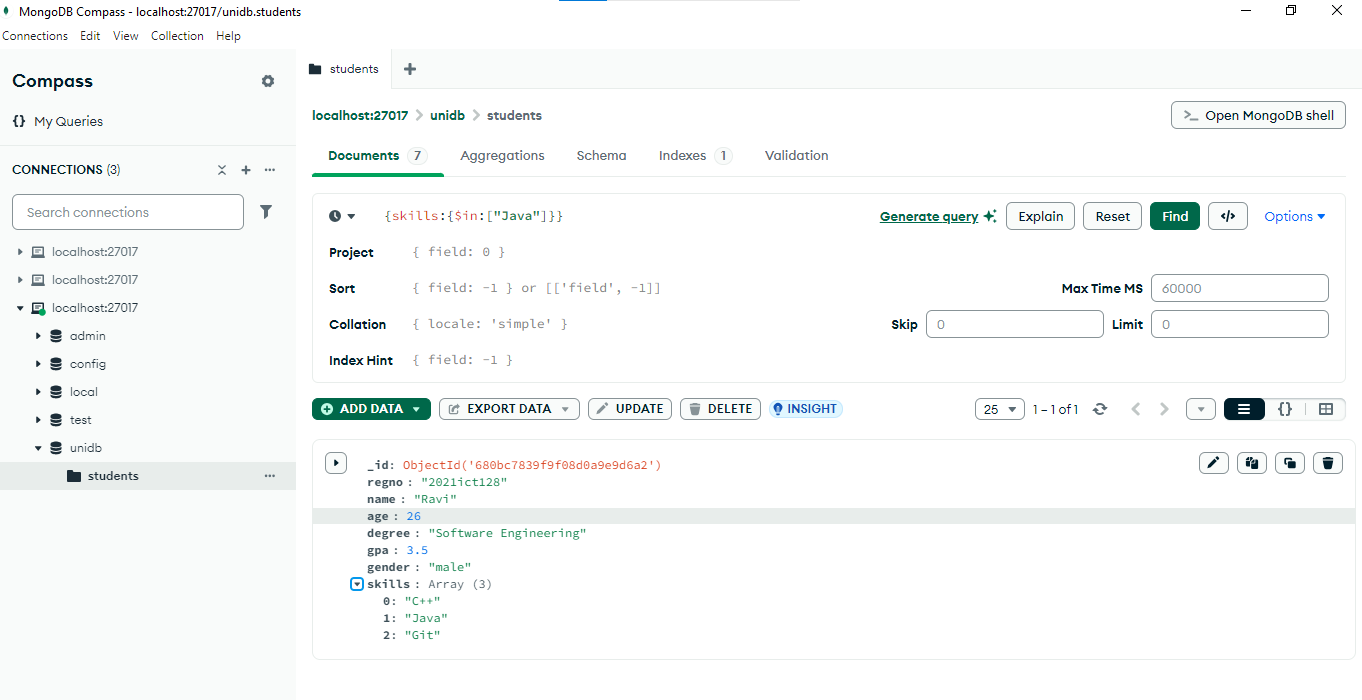
{gender:”female”}



#### 4.6 Find Students with a Skill (Java)

**Explanation:**  
This finds students who have **Java** as one of their skills.

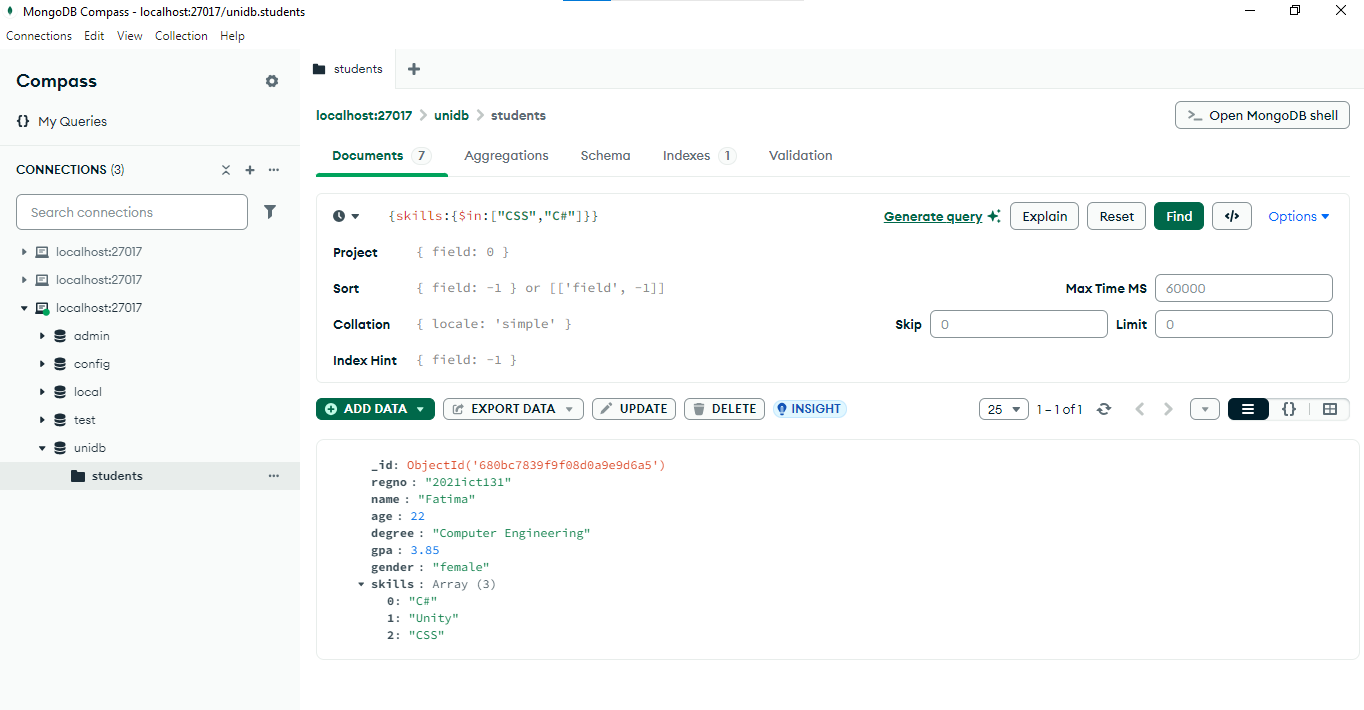
{skills:{$in:[“Java”]}



#### 4.7 Find Students Skilled in CSS or C#

**Explanation:**  
This shows students who know either **CSS** or **C#**.

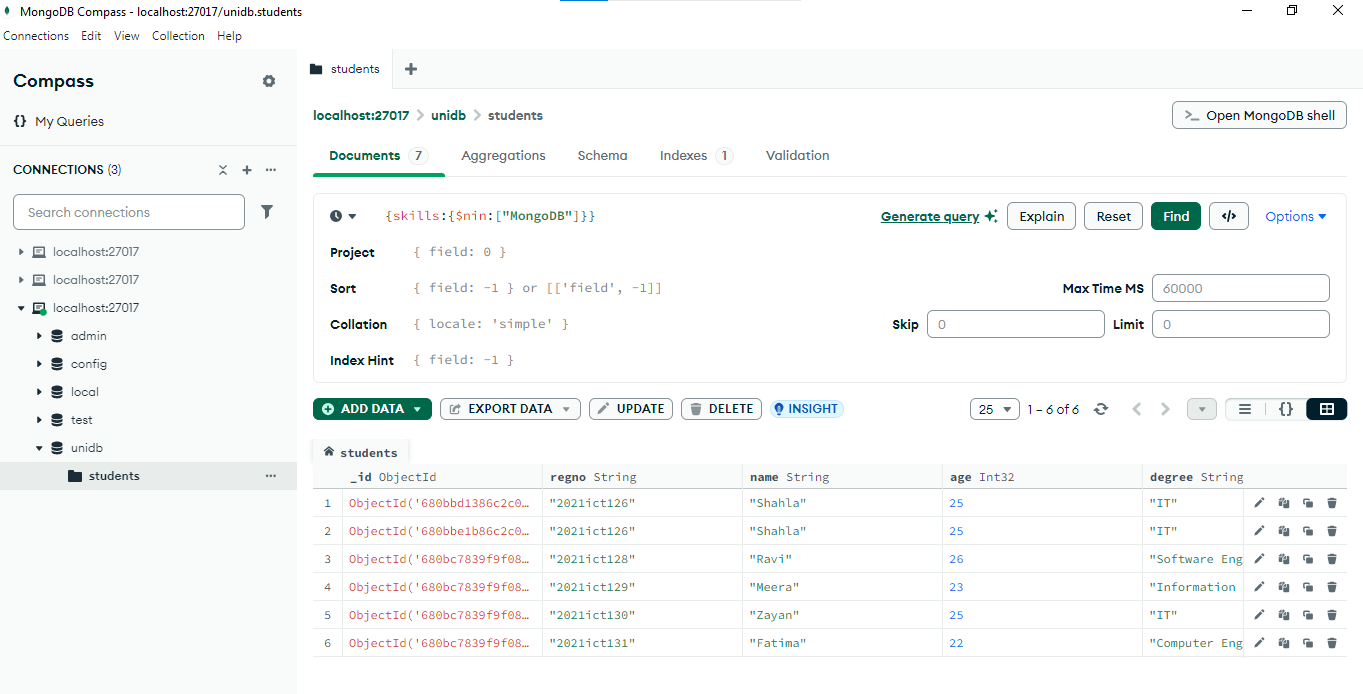
{skills:{$in:[“CSS”,”C#”]}



#### 4.8 Find Students Without MongoDB Skill

**Explanation:**  
This finds students who **do not** have **MongoDB** as a skill.

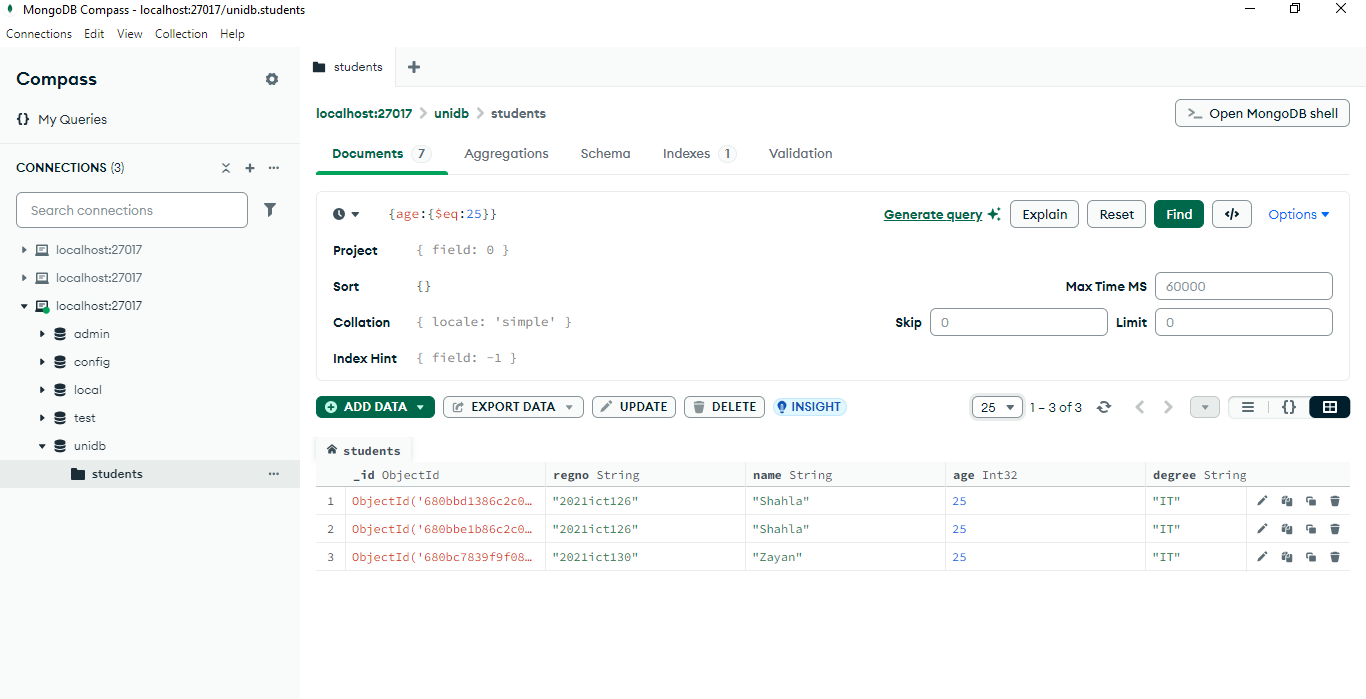
{skills:{$nin:[“MongoDB”]}



#### 4.9 Find Students Aged Exactly 25

**Explanation:**  
This finds students who are exactly **25 years old**.

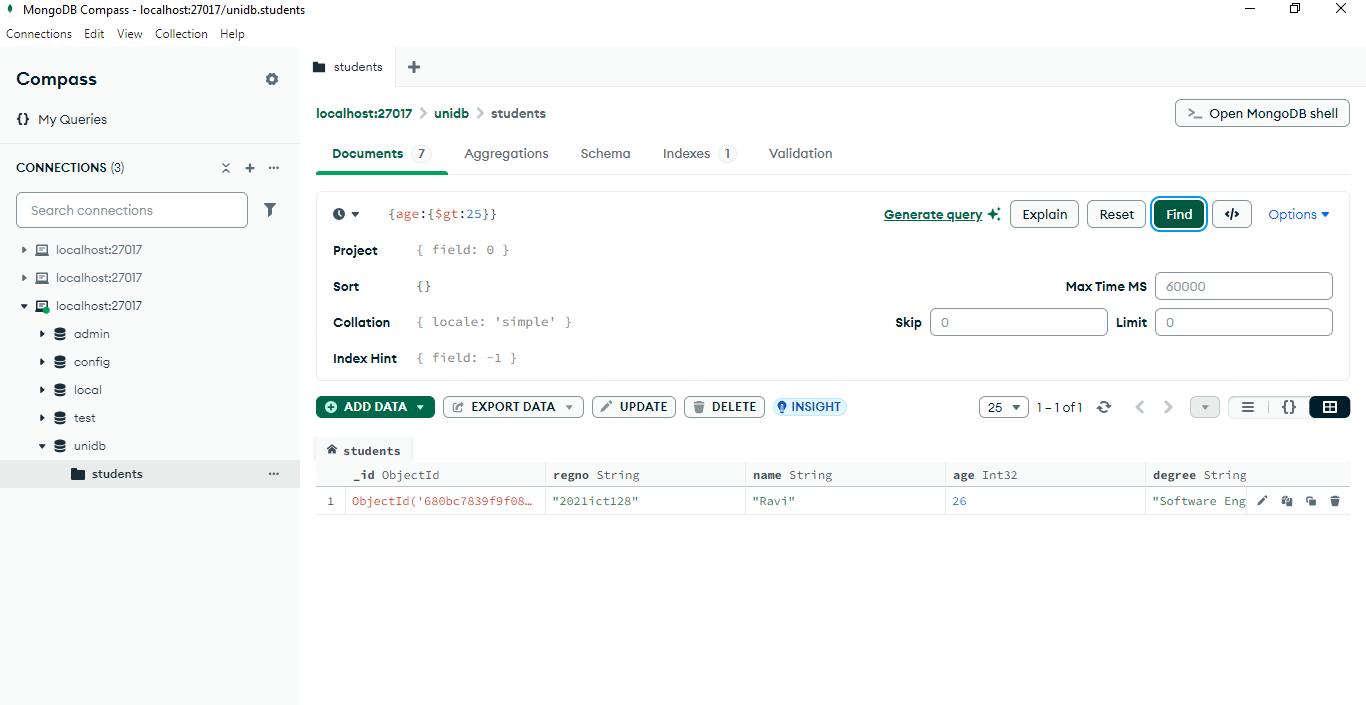
{age:{$eq:25}}



#### 4.10 Find Students Older than 25

**Explanation:**  
This finds students whose age is **greater than 25**.

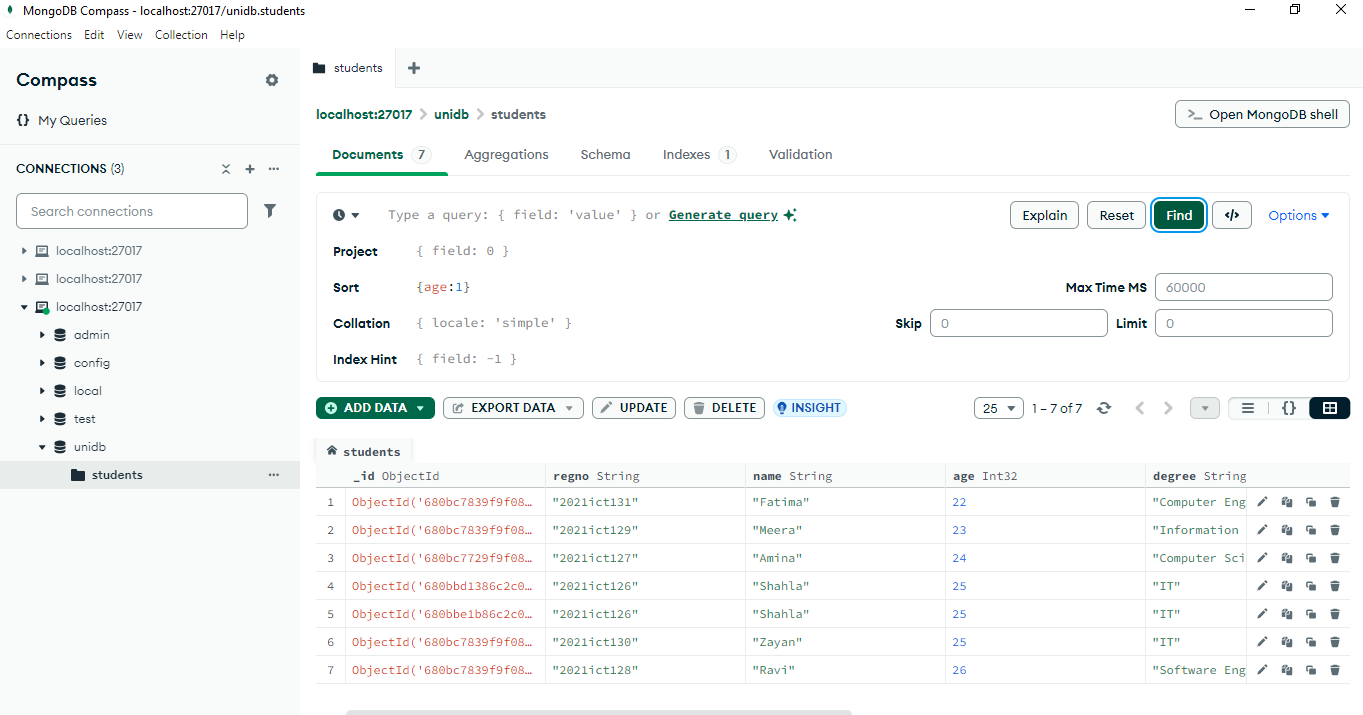
{age:{$gt:25}}



#### 4.11 Sort Students by Age (Ascending)

**Explanation:**  
This arranges students by age from **youngest to oldest**.

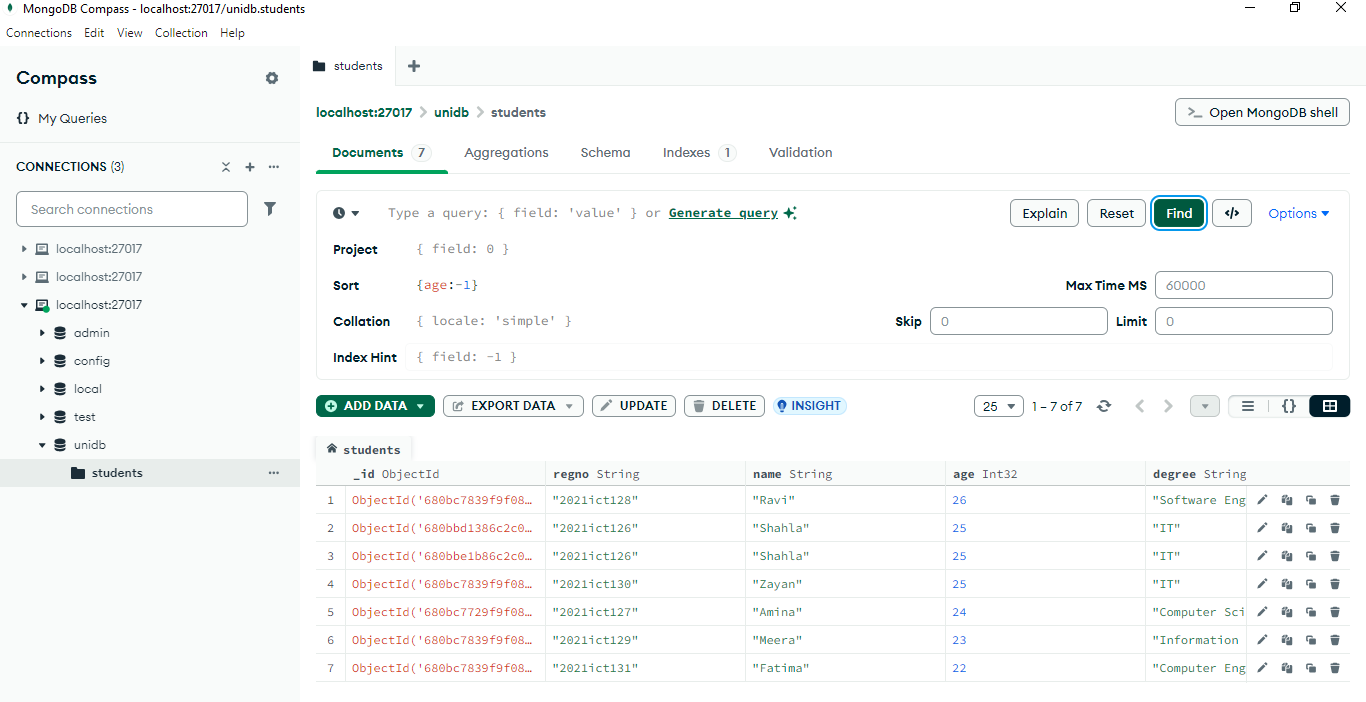
sort{age:1}



#### 4.12 Sort Students by Age (Descending)

**Explanation:**  
This arranges students by age from **oldest to youngest**.

sort{age:-1}



#### 4.13 Find IT Students and Sort by GPA

**Explanation:**  
This finds students studying **IT** and sorts them by their GPA in descending order.

{degree:”IT”}

Sort{gpa:-1}

