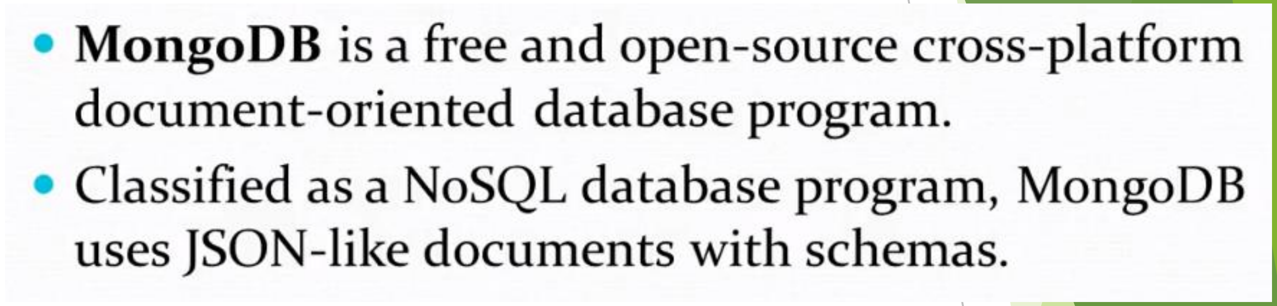
Mongo DB

Made by : Abdul Hafeez

www.codehafeez.com

- MongoDB is NoSQL language.
- MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.

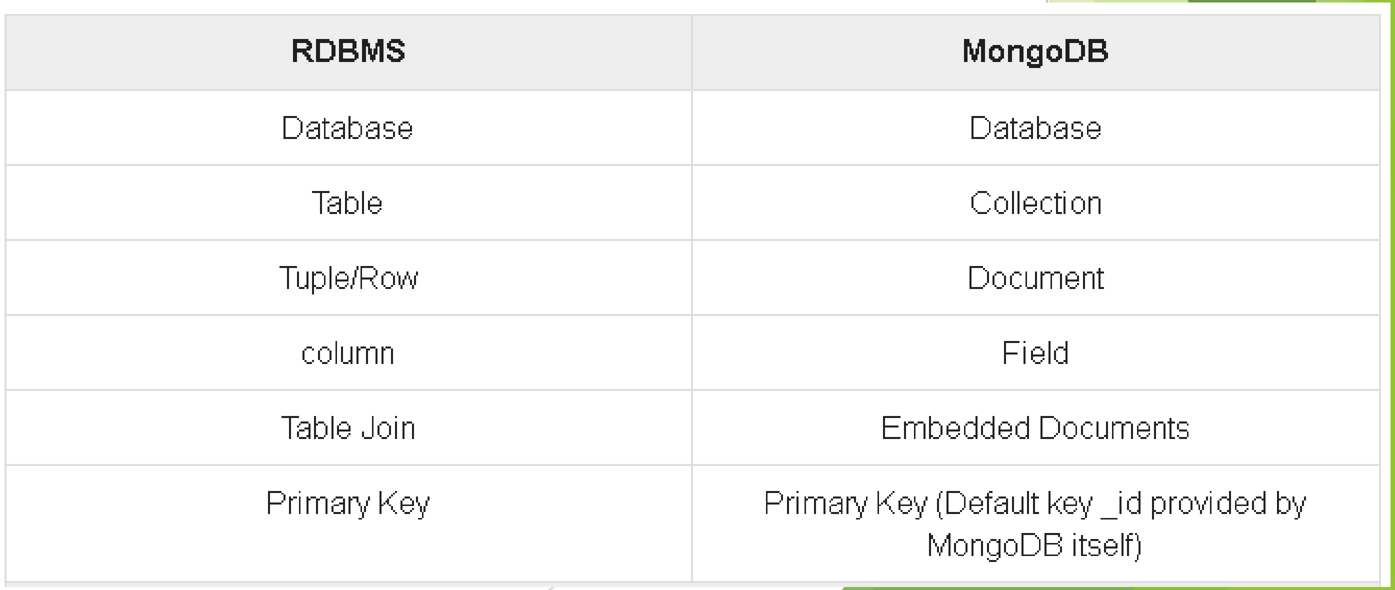


- **MongoDB** is a free and open-source cross-platform document-oriented database program.
- Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas.

- Download IDE MongoDB

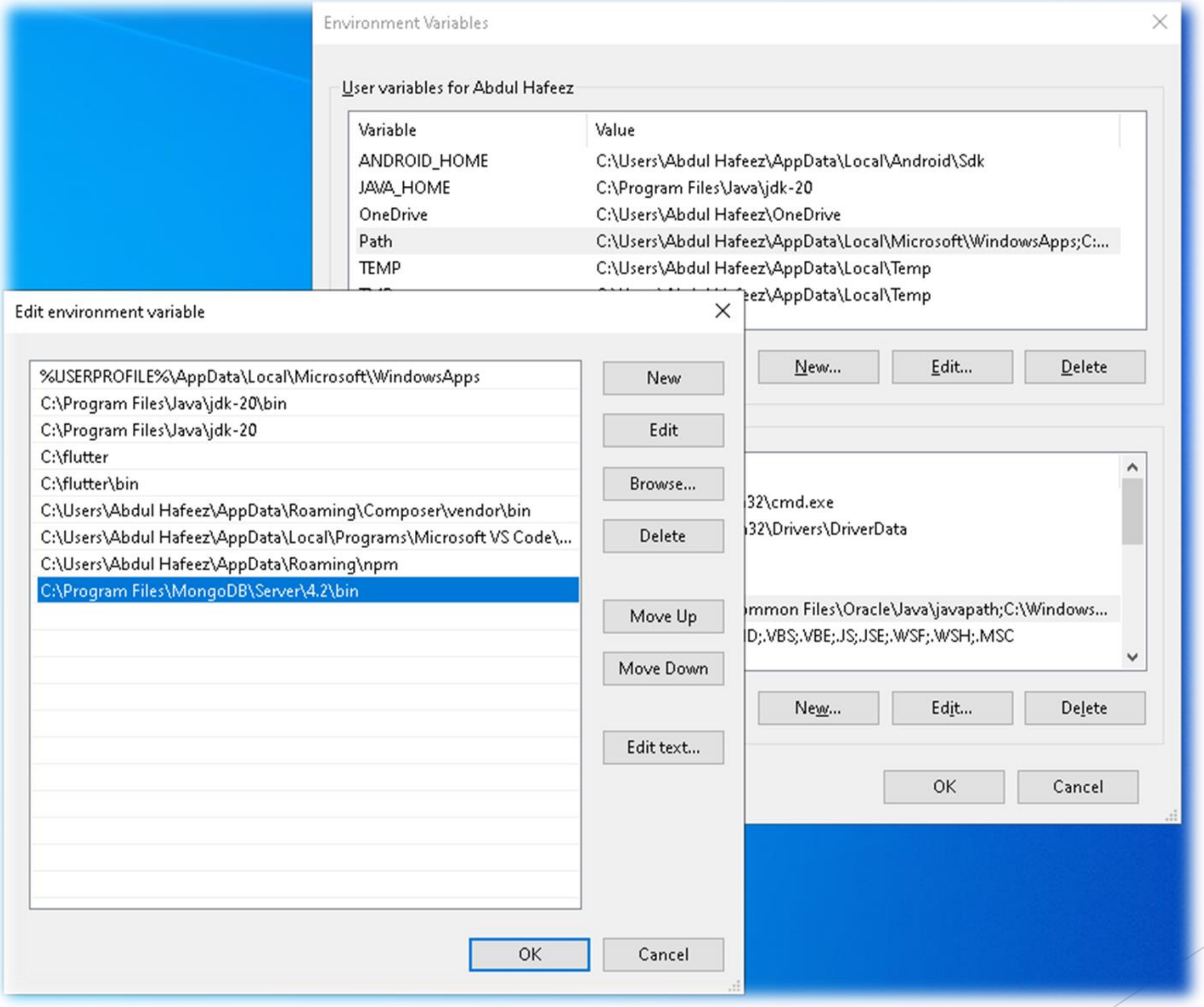
https://www.mongodb.com/try/download/community

- mongo --version



RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by MongoDB itself)

Install & Set Environment Variables



Environment Variables

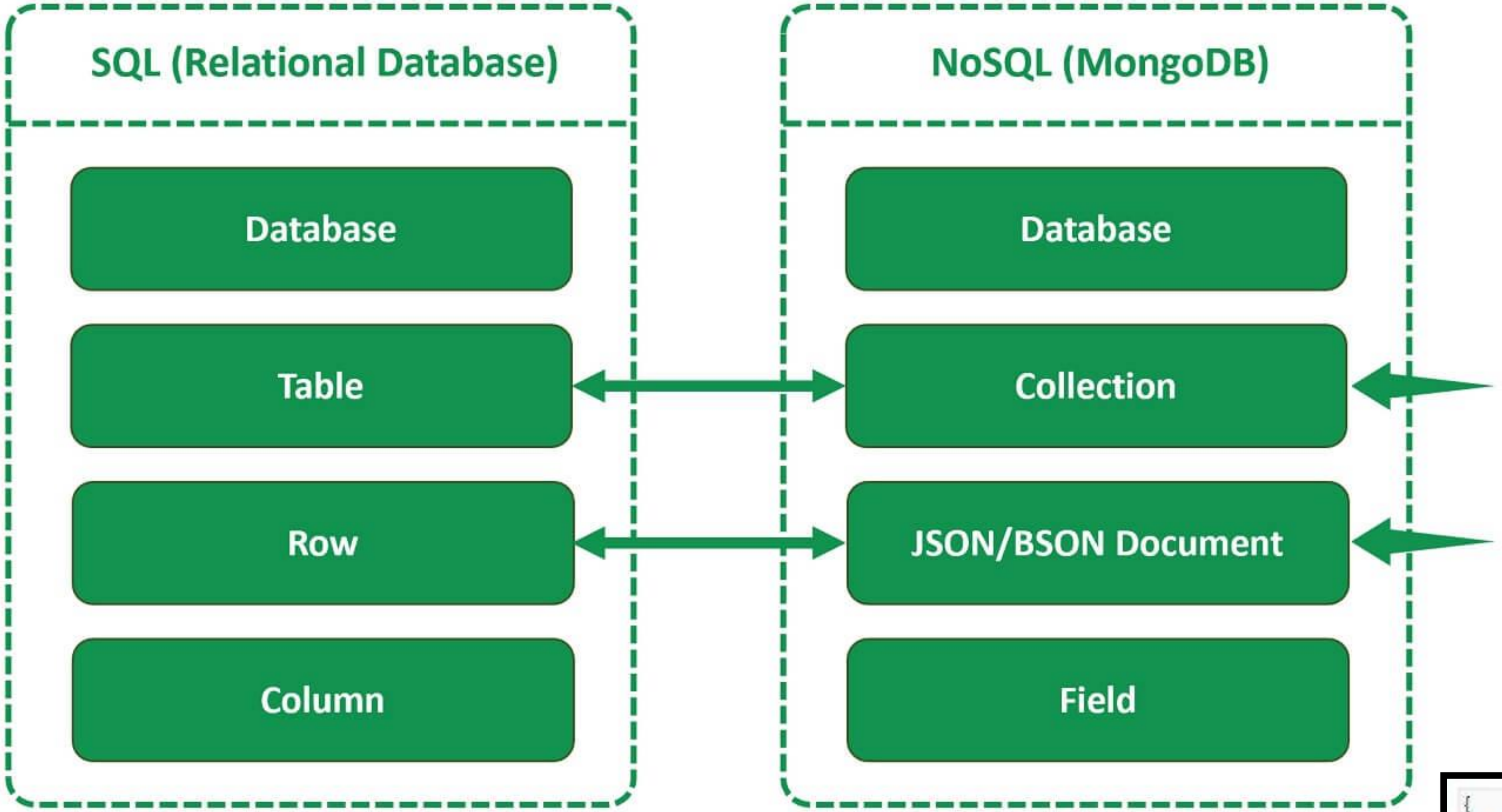
User variables for Abdul Hafeez

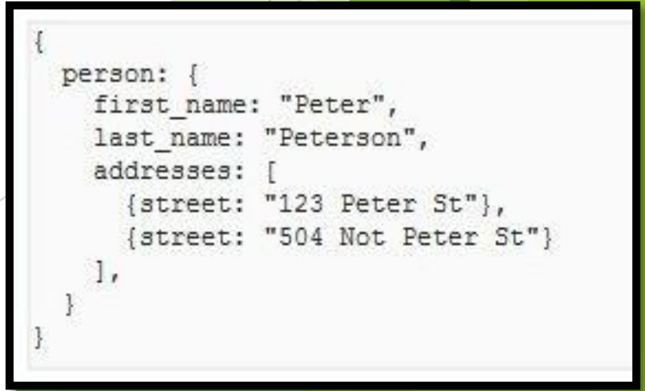
Variable	Value
ANDROID_HOME	C:\Users\Abdul Hafeez\AppData\Local\Android\Sdk
JAVA_HOME	C:\Program Files\Java\jdk-20
OneDrive	C:\Users\Abdul Hafeez\OneDrive
Path	C:\Users\Abdul Hafeez\AppData\Local\Microsoft\WindowsApps;C:...
TEMP	C:\Users\Abdul Hafeez\AppData\Local\Temp
	...ez\AppData\Local\Temp

Edit environment variable

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\Java\jdk-20\bin
C:\Program Files\Java\jdk-20
C:\flutter
C:\flutter\bin
C:\Users\Abdul Hafeez\AppData\Roaming\Composer\vendor\bin
C:\Users\Abdul Hafeez\AppData\Local\Programs\Microsoft VS Code\...
C:\Users\Abdul Hafeez\AppData\Roaming\npm
C:\Program Files\MongoDB\Server\4.2\bin
```

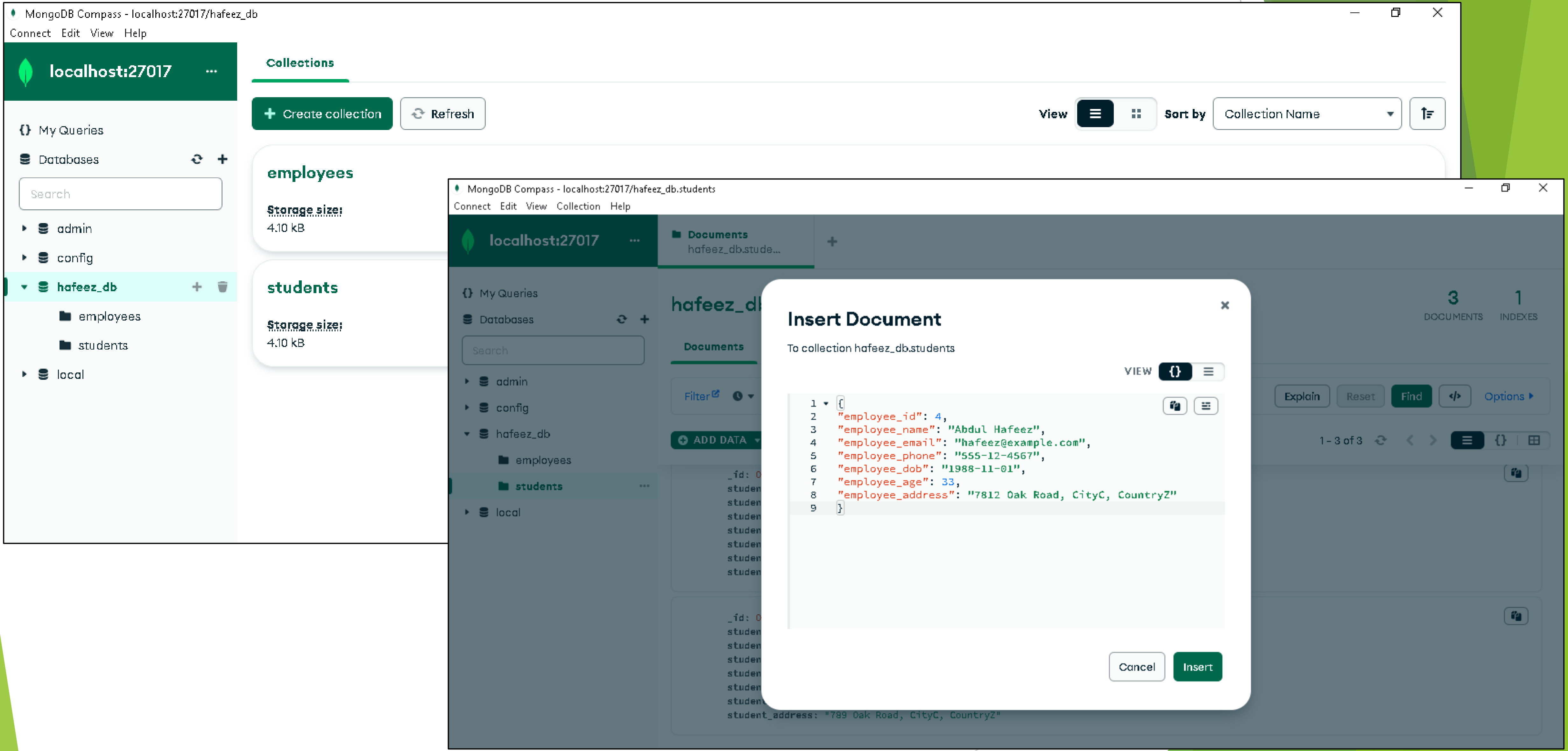
```
...32\cmd.exe
...32\Drivers\DriverData
...mmon Files\Oracle\Java\javapath;C:\Windows...
...D;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
```





```
{
  person: {
    first_name: "Peter",
    last_name: "Peterson",
    addresses: [
      {street: "123 Peter St"},
      {street: "504 Not Peter St"}
    ],
  }
}
```

Insert Single Data


MongoDB Compass - localhost:27017/hafeez_db

Connect Edit View Help

localhost:27017

Collections

Create collection | Refresh

View | Sort by: Collection Name

My Queries

Databases

- admin
- config
- hafeez_db
  - employees
  - students
- local

employees
Storage size: 4.10 kB

students
Storage size: 4.10 kB

MongoDB Compass - localhost:27017/hafeez_db.students

Connect Edit View Collection Help

Documents hafeez_db.stude...

3 DOCUMENTS | 1 INDEXES

Filter | ADD DATA | Explain | Reset | Find | Options

1 - 3 of 3

student_address: "789 Oak Road, CityC, CountryZ"

Insert Document

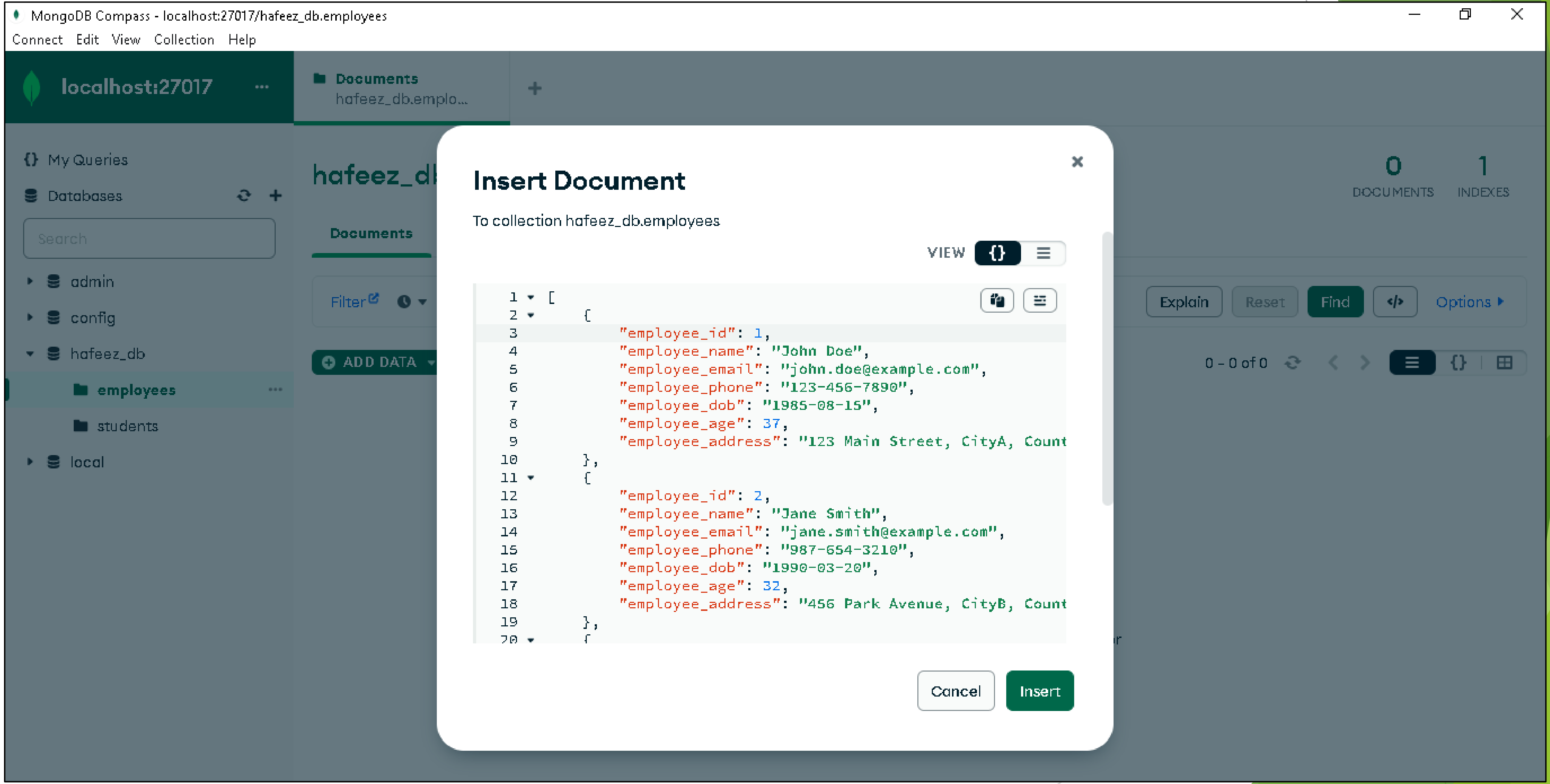
To collection hafeez_db.students

VIEW

```
{
"employee_id": 4,
"employee_name": "Abdul Hafeez",
"employee_email": "hafeez@example.com",
"employee_phone": "555-12-4567",
"employee_dob": "1988-11-01",
"employee_age": 33,
"employee_address": "7812 Oak Road, CityC, CountryZ"
}
```

Cancel | Insert

Insert Multiple Data

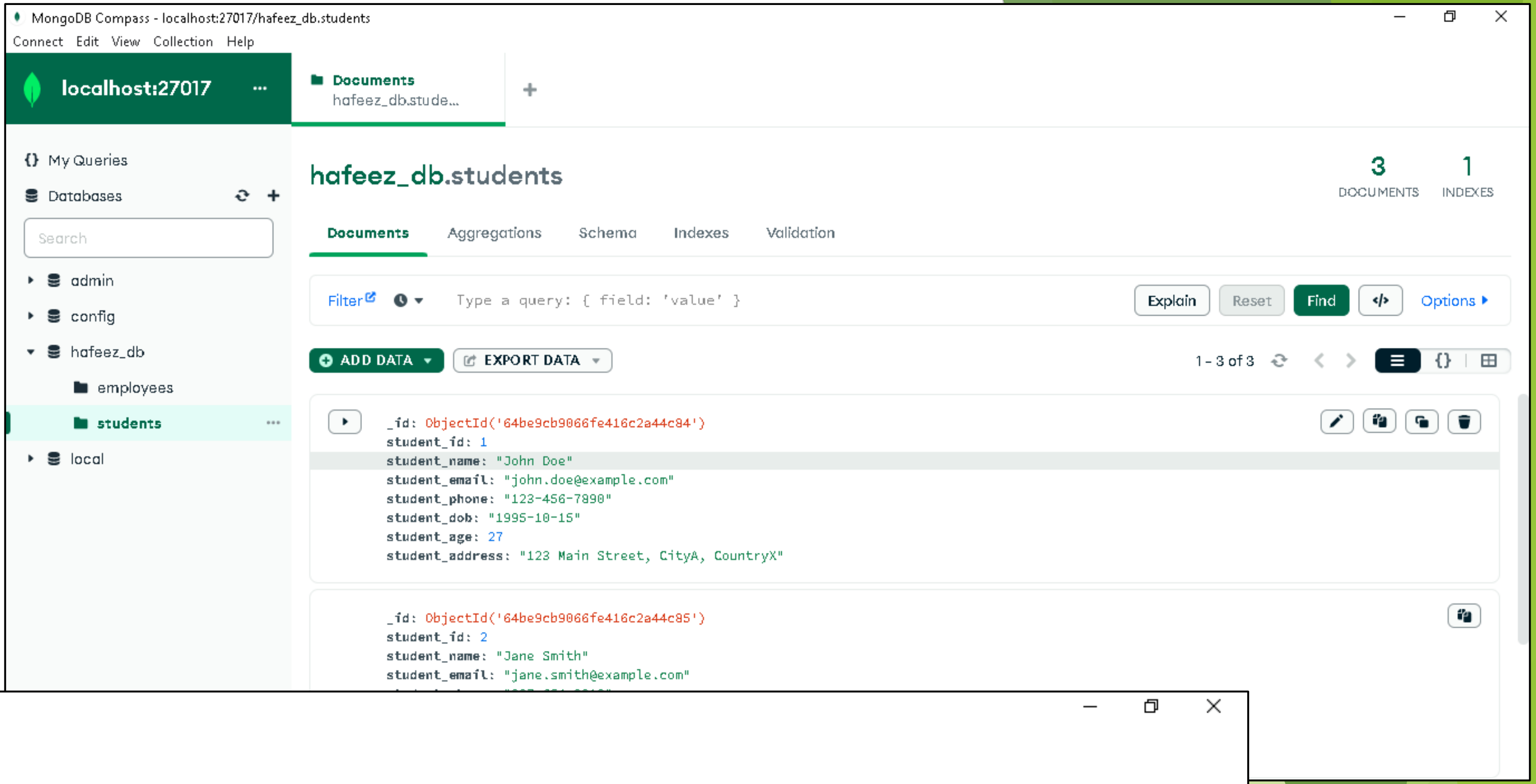


Insert Document

To collection hafeez_db.employees

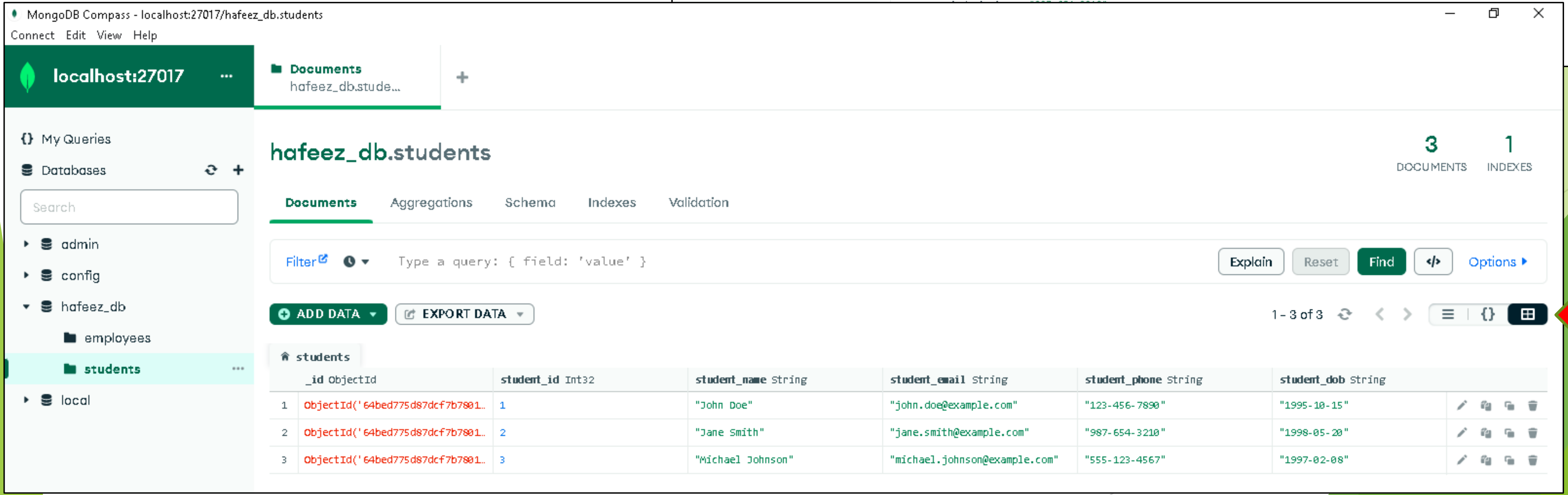
```
[
    {
        "employee_id": 1,
        "employee_name": "John Doe",
        "employee_email": "john.doe@example.com",
        "employee_phone": "123-456-7890",
        "employee_dob": "1985-08-15",
        "employee_age": 37,
        "employee_address": "123 Main Street, CityA, Count
    },
    {
        "employee_id": 2,
        "employee_name": "Jane Smith",
        "employee_email": "jane.smith@example.com",
        "employee_phone": "987-654-3210",
        "employee_dob": "1990-03-20",
        "employee_age": 32,
        "employee_address": "456 Park Avenue, CityB, Count
    },
    {
```

View Data



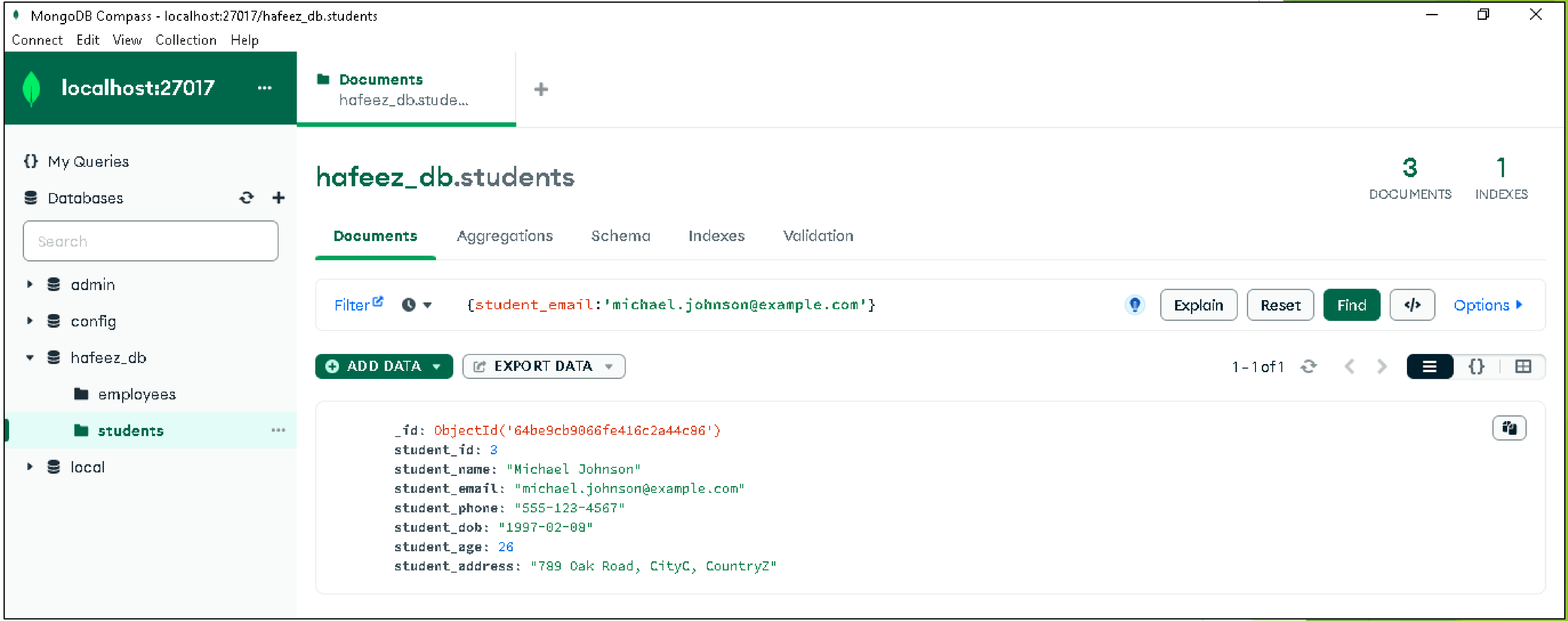
```
_id: ObjectId('64be9cb9066fe416c2a44c84')
student_id: 1
student_name: "John Doe"
student_email: "john.doe@example.com"
student_phone: "123-456-7890"
student_dob: "1995-10-15"
student_age: 27
student_address: "123 Main Street, CityA, CountryX"

_id: ObjectId('64be9cb9066fe416c2a44c85')
student_id: 2
student_name: "Jane Smith"
student_email: "jane.smith@example.com"
```



	_id ObjectId	student_id Int32	student_name String	student_email String	student_phone String	student_dob String
1	ObjectId('64bed775d87dcf7b7801…	1	"John Doe"	"john.doe@example.com"	"123-456-7890"	"1995-10-15"
2	ObjectId('64bed775d87dcf7b7801…	2	"Jane Smith"	"jane.smith@example.com"	"987-654-3210"	"1998-05-20"
3	ObjectId('64bed775d87dcf7b7801…	3	"Michael Johnson"	"michael.johnson@example.com"	"555-123-4567"	"1997-02-08"

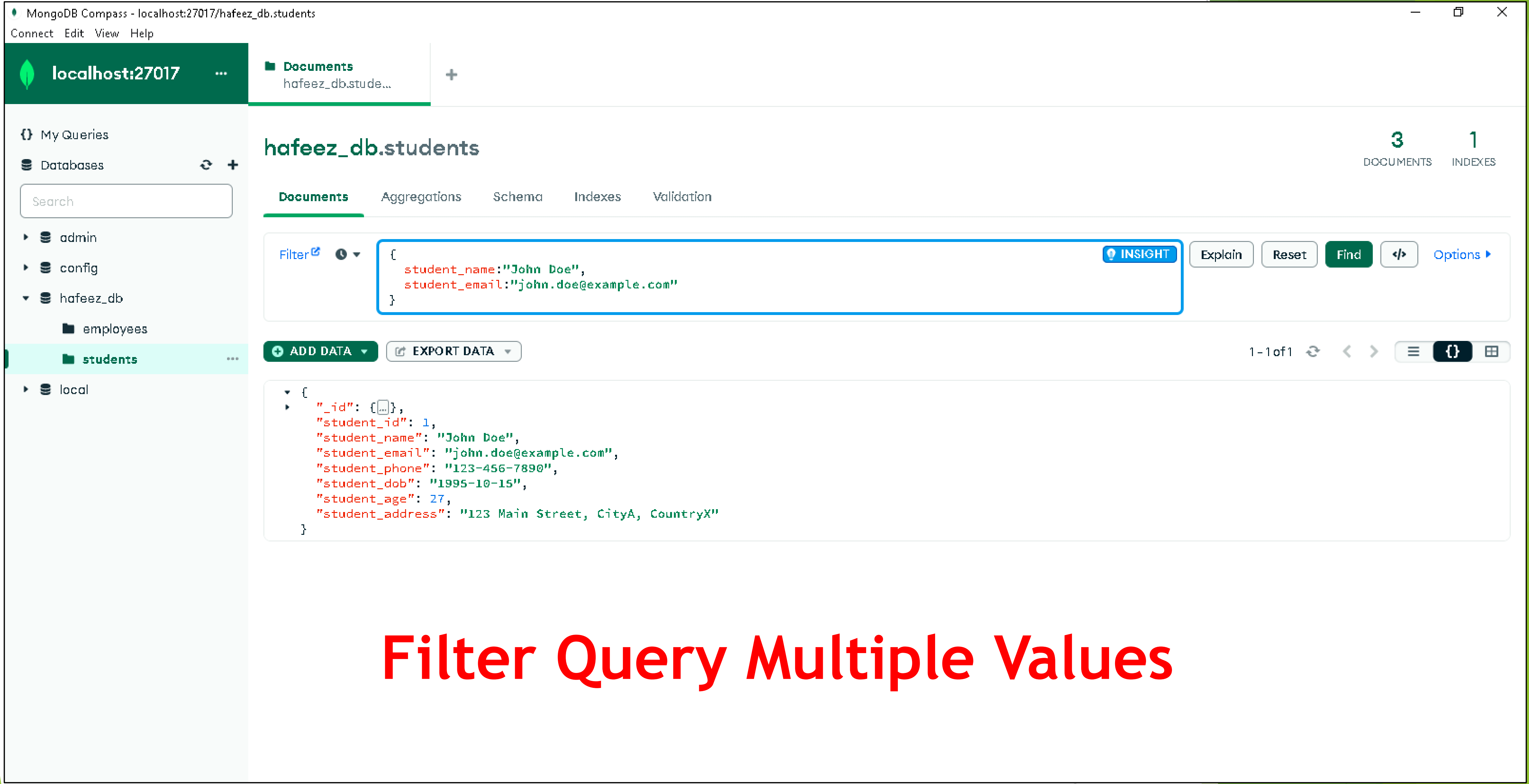
{student_email:'michael.johnson@example.com'}



```
{student_email:'michael.johnson@example.com'}
```

```
_id: ObjectId('64be9cb9066fe416c2a44c86')
student_id: 3
student_name: "Michael Johnson"
student_email: "michael.johnson@example.com"
student_phone: "555-123-4567"
student_dob: "1997-02-08"
student_age: 26
student_address: "789 Oak Road, CityC, CountryZ"
```

Filter Query Single Value

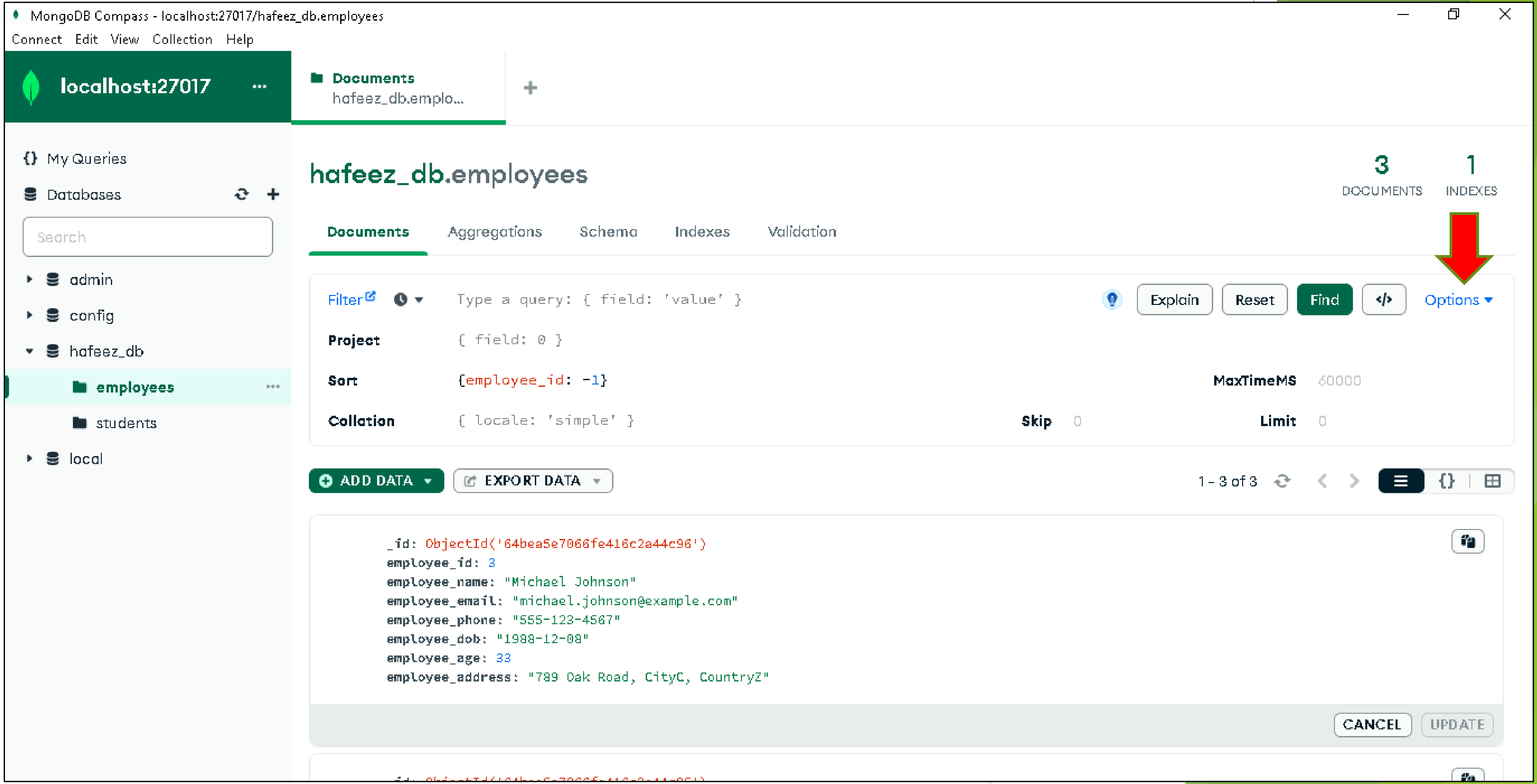


```
{
  student_name:"John Doe",
  student_email:"john.doe@example.com"
}
```

```
{
  "_id": {…},
  "student_id": 1,
  "student_name": "John Doe",
  "student_email": "john.doe@example.com",
  "student_phone": "123-456-7890",
  "student_dob": "1995-10-15",
  "student_age": 27,
  "student_address": "123 Main Street, CityA, CountryX"
}
```

Filter Query Multiple Values

View Data Sort

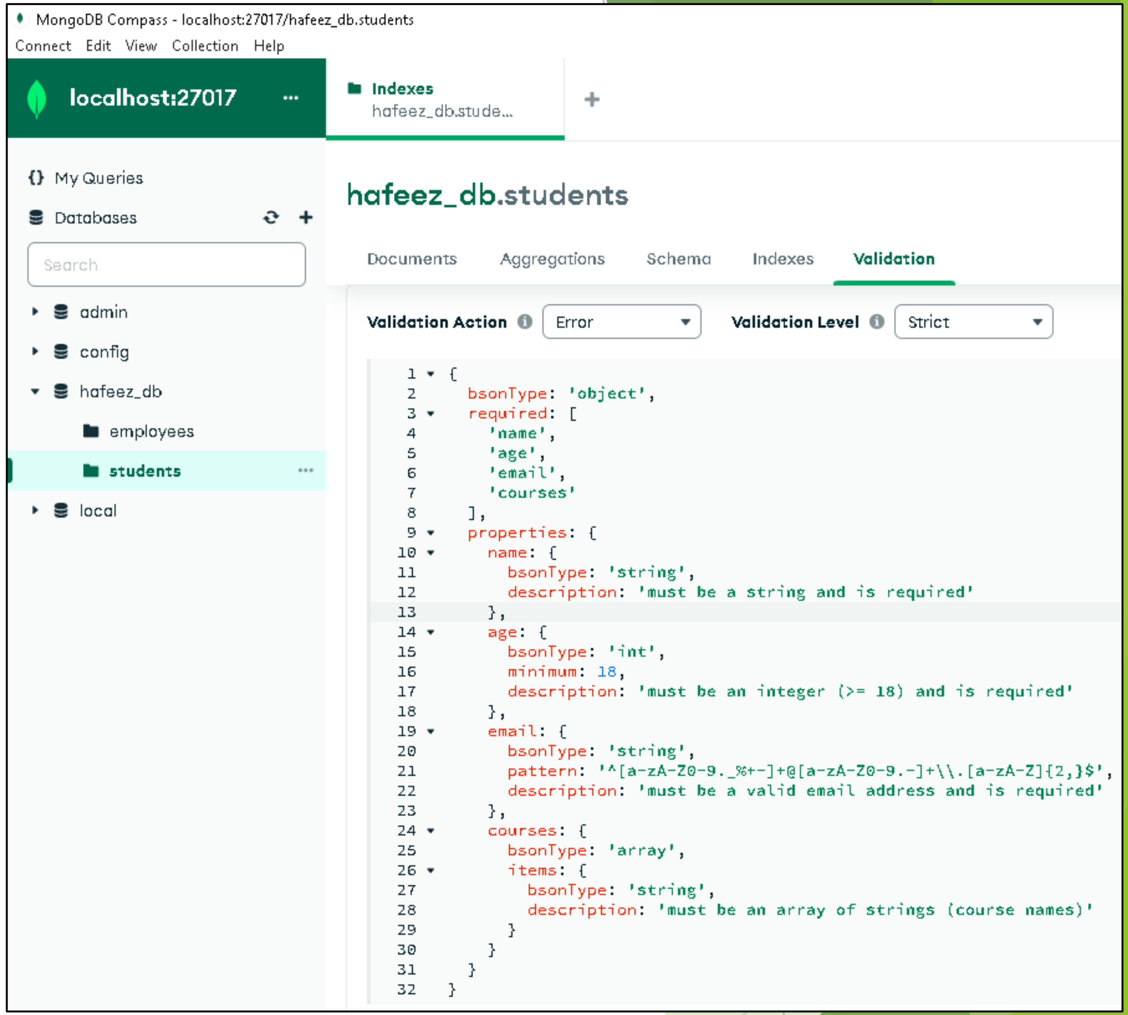


```
Filter     Type a query: { field: 'value' }
Project    { field: 0 }
Sort       {employee_id: -1}                          MaxTimeMS  60000
Collation  { locale: 'simple' }          Skip  0      Limit      0
```

```
_id: ObjectId('64bea5e7066fe416c2a44c96')
employee_id: 3
employee_name: "Michael Johnson"
employee_email: "michael.johnson@example.com"
employee_phone: "555-123-4567"
employee_dob: "1988-12-08"
employee_age: 33
employee_address: "789 Oak Road, CityC, CountryZ"
```

Insert Data - Validation Rules

```
{
  "bsonType": "object",
  "required": ["name", "age", "email", "courses"],
  "properties": {
    "name": {
      "bsonType": "string",
      "description": "must be a string and is required",
    },
    "age": {
      "bsonType": "int",
      "minimum": 18,
      "description": "must be an integer (>= 18) and is required",
    },
    "email": {
      "bsonType": "string",
      "pattern": "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$",
      "description": "must be a valid email address and is required",
    },
    "courses": {
      "bsonType": "array",
      "items": {
        "bsonType": "string",
        "description": "must be an array of strings (course names)",
      },
    }
  }
}
```


MongoDB Compass - localhost:27017/hafeez_db.students

Connect Edit View Collection Help

localhost:27017

Indexes hafeez_db.stude...

My Queries

Databases

Search

admin
config
hafeez_db
employees
students
local

hafeez_db.students

Documents Aggregations Schema Indexes Validation

Validation Action: Error

Validation Level: Strict

```
{
  bsonType: 'object',
  required: [
    'name',
    'age',
    'email',
    'courses'
  ],
  properties: {
    name: {
      bsonType: 'string',
      description: 'must be a string and is required'
    },
    age: {
      bsonType: 'int',
      minimum: 18,
      description: 'must be an integer (>= 18) and is required'
    },
    email: {
      bsonType: 'string',
      pattern: '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$',
      description: 'must be a valid email address and is required'
    },
    courses: {
      bsonType: 'array',
      items: {
        bsonType: 'string',
        description: 'must be an array of strings (course names)'
      }
    }
  }
}
```

```
[
 {
  "student_id": 1,
  "student_name": "John Doe",
  "student_email": "john.doe@example.com",
  "student_phone": "123-456-7890",
  "student_dob": "1995-10-15",
  "student_age": 27,
  "student_address": "123 Main Street, CityA, CountryX"
 },
 {
  "student_id": 2,
  "student_name": "Jane Smith",
  "student_email": "jane.smith@example.com",
  "student_phone": "987-654-3210",
  "student_dob": "1998-05-20",
  "student_age": 24,
  "student_address": "456 Park Avenue, CityB, CountryY"
 },
 {
  "student_id": 3,
  "student_name": "Michael Johnson",
  "student_email": "michael.johnson@example.com",
  "student_phone": "555-123-4567",
  "student_dob": "1997-02-08",
  "student_age": 26,
  "student_address": "789 Oak Road, CityC, CountryZ"
 }
]
```

students.json

```
[
    {
        "employee_id": 1,
        "employee_name": "John Doe",
        "employee_email": "john.doe@example.com",
        "employee_phone": "123-456-7890",
        "employee_dob": "1985-08-15",
        "employee_age": 37,
        "employee_address": "123 Main Street, CityA, CountryX"
    },
    {
        "employee_id": 2,
        "employee_name": "Jane Smith",
        "employee_email": "jane.smith@example.com",
        "employee_phone": "987-654-3210",
        "employee_dob": "1990-03-20",
        "employee_age": 32,
        "employee_address": "456 Park Avenue, CityB, CountryY"
    },
    {
        "employee_id": 3,
        "employee_name": "Michael Johnson",
        "employee_email": "michael.johnson@example.com",
        "employee_phone": "555-123-4567",
        "employee_dob": "1988-12-08",
        "employee_age": 33,
        "employee_address": "789 Oak Road, CityC, CountryZ"
    }
]
```

JSON

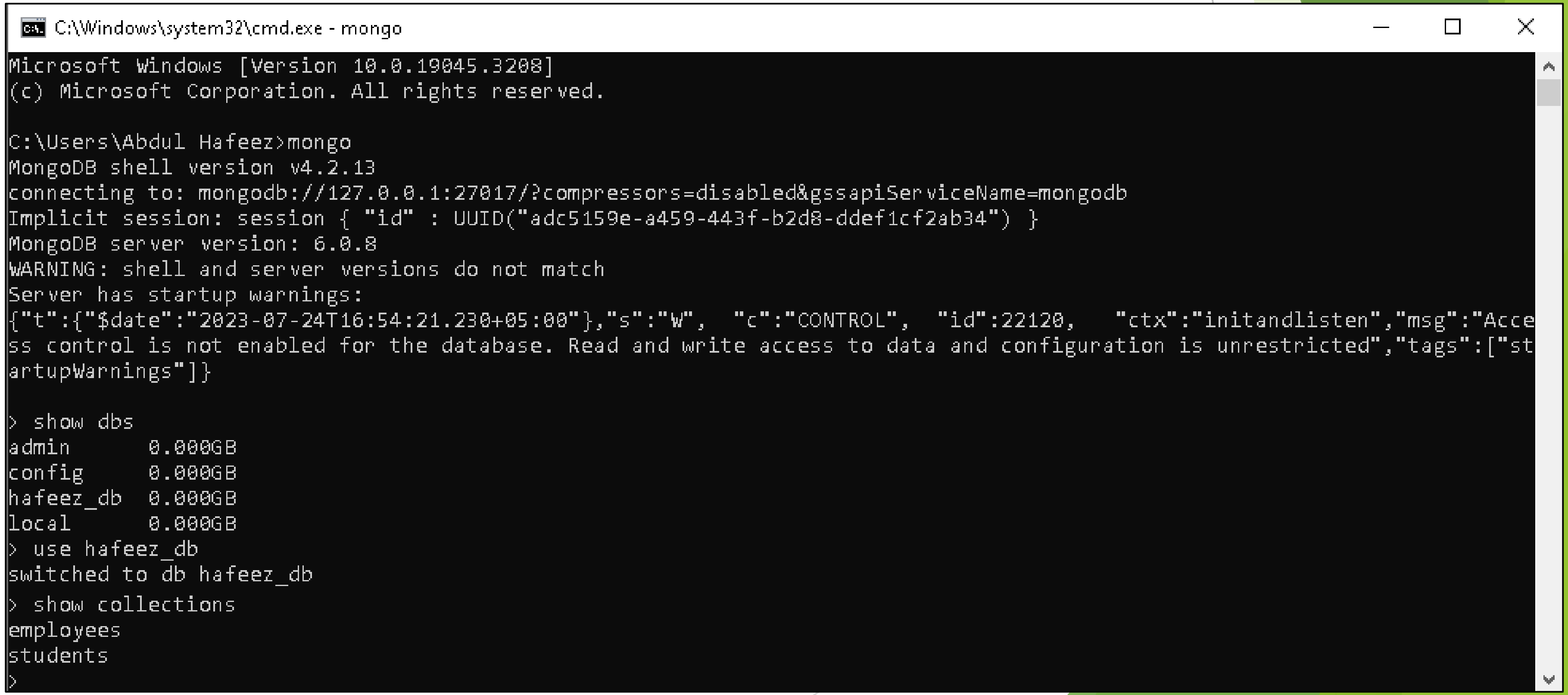
employees.json

```
{
 bsonType: 'object',
 required: [
  'name',
  'age',
  'email',
  'courses'
 ],
 properties: {
  name: {
   bsonType: 'string',
   description: 'must be a string and is required'
  },
  age: {
   bsonType: 'int',
   minimum: 18,
   description: 'must be an integer (>= 18) and is required'
  },
  email: {
   bsonType: 'string',
   pattern: '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$',
   description: 'must be a valid email address and is required'
  },
  courses: {
   bsonType: 'array',
   items: {
    bsonType: 'string',
    description: 'must be an array of strings (course names)'
   }
  }
 }
}
```

students-validation-rules.json

Using CMD use MongoDB

- mongo
- show dbs
- use mydatabase
- show collections



```
Microsoft Windows [Version 10.0.19045.3208]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abdul Hafeez>mongo
MongoDB shell version v4.2.13
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("adc5159e-a459-443f-b2d8-ddef1cf2ab34") }
MongoDB server version: 6.0.8
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2023-07-24T16:54:21.230+05:00"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Acce
ss control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}

> show dbs
admin      0.000GB
config     0.000GB
hafeez_db  0.000GB
local      0.000GB
> use hafeez_db
switched to db hafeez_db
> show collections
employees
students
>
```

Using CMD use MongoDB

```
mongo
use hafeez_db
db.createCollection("users")
db.users.insertOne({ name: "John Doe", age: 30, email: "john@example.com" })
db.users.insertMany([
 { name: "Alice", age: 25, email: "alice@example.com" },
 { name: "Bob", age: 28, email: "bob@example.com" },
 { name: "Eve", age: 22, email: "eve@example.com" }
])

db.users.find()
db.users.find().limit(2)
db.users.find({ name: "John Doe" })
db.users.updateOne({ name: "John Doe" }, { $set: { age: 31 } })
db.users.updateMany({ name: "John Doe" }, { $set: { age: 31 } })
db.users.deleteOne({ age: 30 })
db.users.deleteMany({ age: 30 })
db.users.find().sort({ age: 1 }) // 1=> ASC & -1 => DESC
db.users.find().sort({ age: 1, name: -1 })
```

```
mongo
use hafeez_db;

db.createCollection("students", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "age", "email", "courses"],
      properties: {
        name: {
          bsonType: "string",
          description: "must be a string and is required",
        },
        age: {
          bsonType: "int",
          minimum: 18,
          description: "must be an integer (>= 18) and is required",
        },
        email: {
          bsonType: "string",
          pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$",
          description: "must be a valid email address and is required",
        },
        courses: {
          bsonType: "array",
          items: {
            bsonType: "string",
            description: "must be an array of strings (course names)",
          },
        },
      },
    },
  },
});
```

MongoDB Validation Rules Using CMD

Example 01

```
db.runCommand({
  collMod: "students",
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["name", "age", "email", "courses"],
      properties: {
        name: {
          bsonType: "string",
          description: "must be a string and is required",
        },
        age: {
          bsonType: "int",
          minimum: 18,
          description: "must be an integer (>= 18) and is required",
        },
        email: {
          bsonType: "string",
          pattern: "^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\\.[a-zA-Z]{2,}$",
          description: "must be a valid email address and is required",
        },
        courses: {
          bsonType: "array",
          items: {
            bsonType: "string",
            description: "must be an array of strings (course names)",
          },
        },
      },
    },
  },
  validationLevel: "strict",
  validationAction: "error",
});
```

MongoDB Validation Rules Using CMD

Example 02