COURSE OUTCOME 3

Analyse and compare relational and non-relational (NoSQL) databases and configure NoSQL databases.

PROGRAM 1

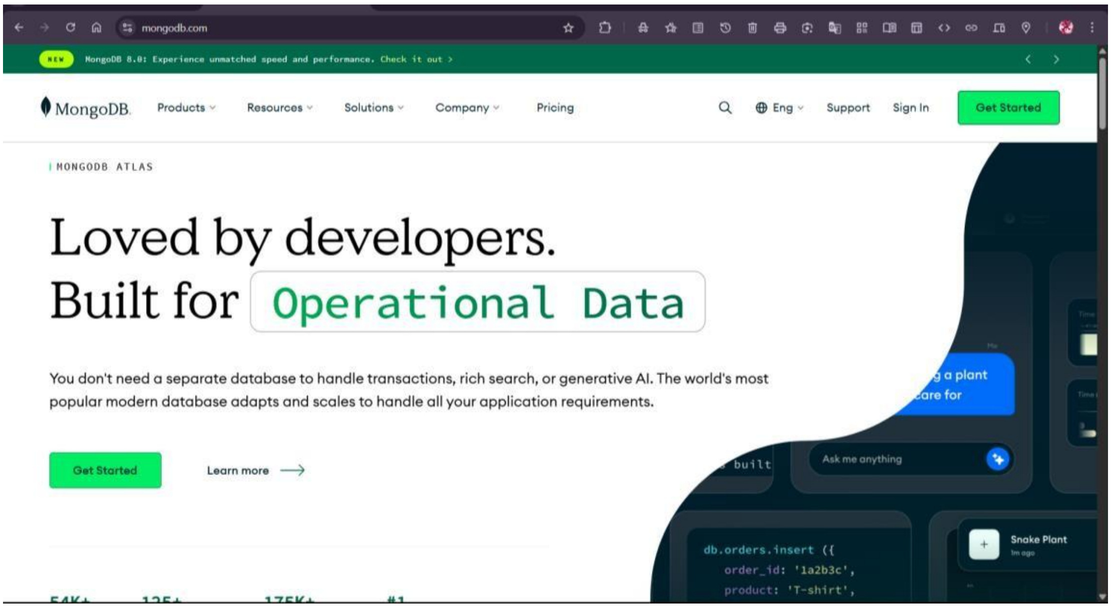
AIM: Download, install and configure Mongo DB server and Mongo Shell.

PROCEDURE

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

STEP 1:

Go to the website : https://www.mongodb.com/

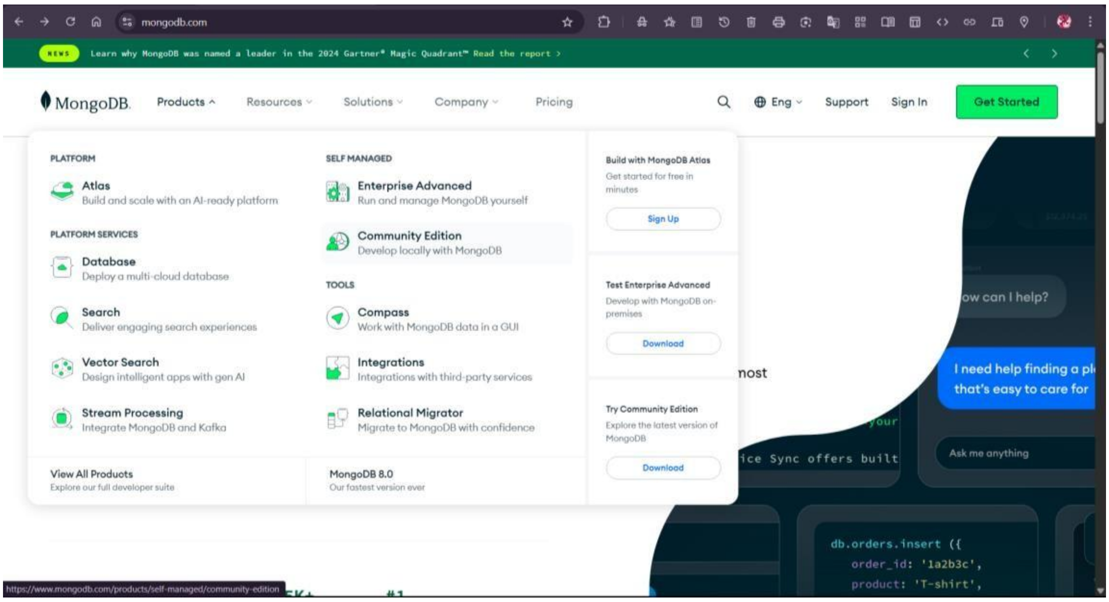


Loved by developers.
Built for Operational Data

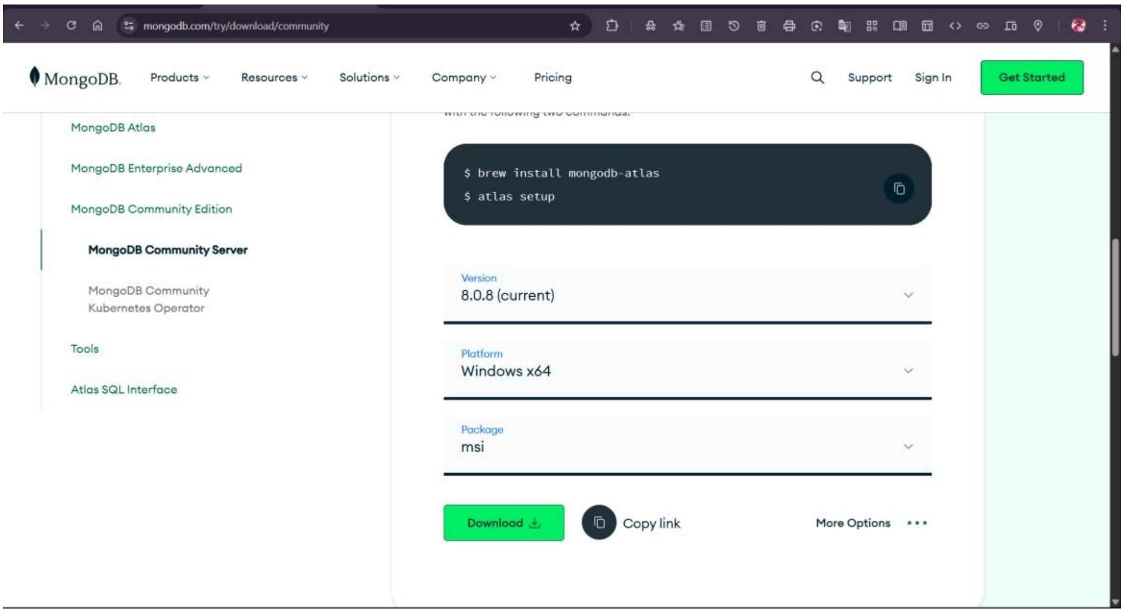
You don't need a separate database to handle transactions, rich search, or generative AI. The world's most popular modern database adapts and scales to handle all your application requirements.

```
db.orders.insert ({
  order_id: '1a2b3c',
  product: 'T-shirt',
```

STEP 2:
Under the products section, select the Community Edition and click Download Community


mongodb.com
NEWS Learn why MongoDB was named a leader in the 2024 Gartner® Magic Quadrant™ Read the report >
MongoDB
Products
Resources
Solutions
Company
Pricing
Eng
Support
Sign In
Get Started
PLATFORM
Atlas
Build and scale with an AI-ready platform
PLATFORM SERVICES
Database
Deploy a multi-cloud database
Search
Deliver engaging search experiences
Vector Search
Design intelligent apps with gen AI
Stream Processing
Integrate MongoDB and Kafka
SELF MANAGED
Enterprise Advanced
Run and manage MongoDB yourself
Community Edition
Develop locally with MongoDB
TOOLS
Compass
Work with MongoDB data in a GUI
Integrations
Integrations with third-party services
Relational Migrator
Migrate to MongoDB with confidence
View All Products
Explore our full developer suite
MongoDB 8.0
Our fastest version ever
Build with MongoDB Atlas
Get started for free in minutes
Sign Up
Test Enterprise Advanced
Develop with MongoDB on-premises
Download
Try Community Edition
Explore the latest version of MongoDB
Download
ow can I help?
I need help finding a pl that's easy to care for
Ask me anything
https://www.mongodb.com/products/self-managed/community-edition

STEP 3:
Check the specifications to the right of the screen are correct. At the time of writing, the latest version is 8.0.8. Ensure that the platform is Windows, and the package is MSI. Go ahead and click on download.



```
$ brew install mongodb-atlas
$ atlas setup
```

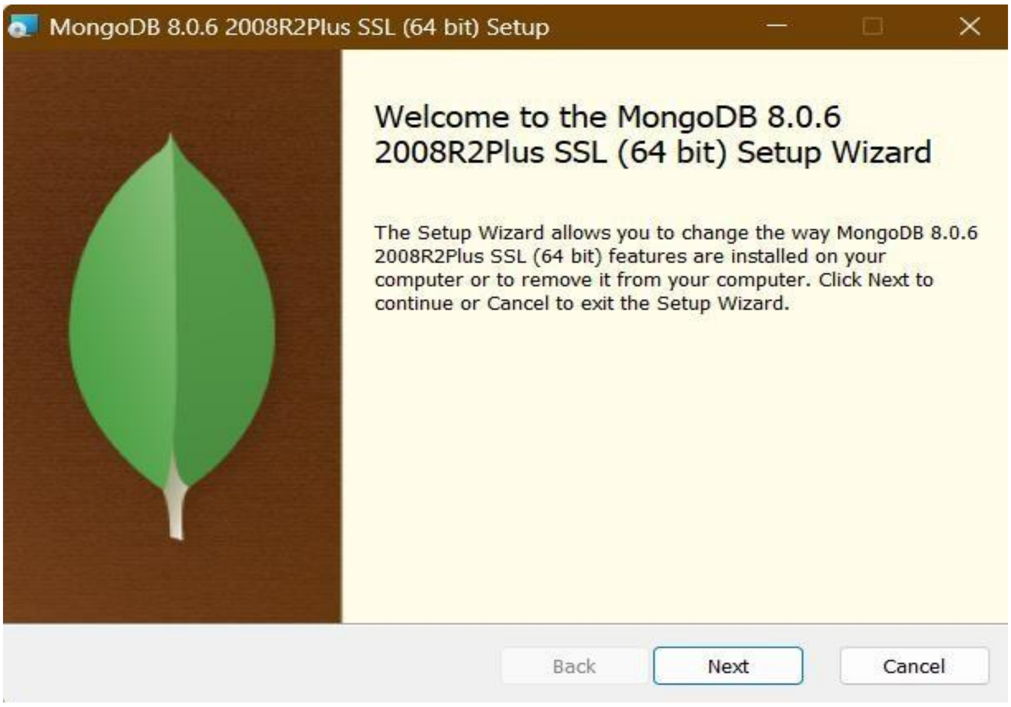
Version
8.0.8 (current)

Platform
Windows x64

Package
msi

STEP 4:

Start the installation process by opening the downloaded file.

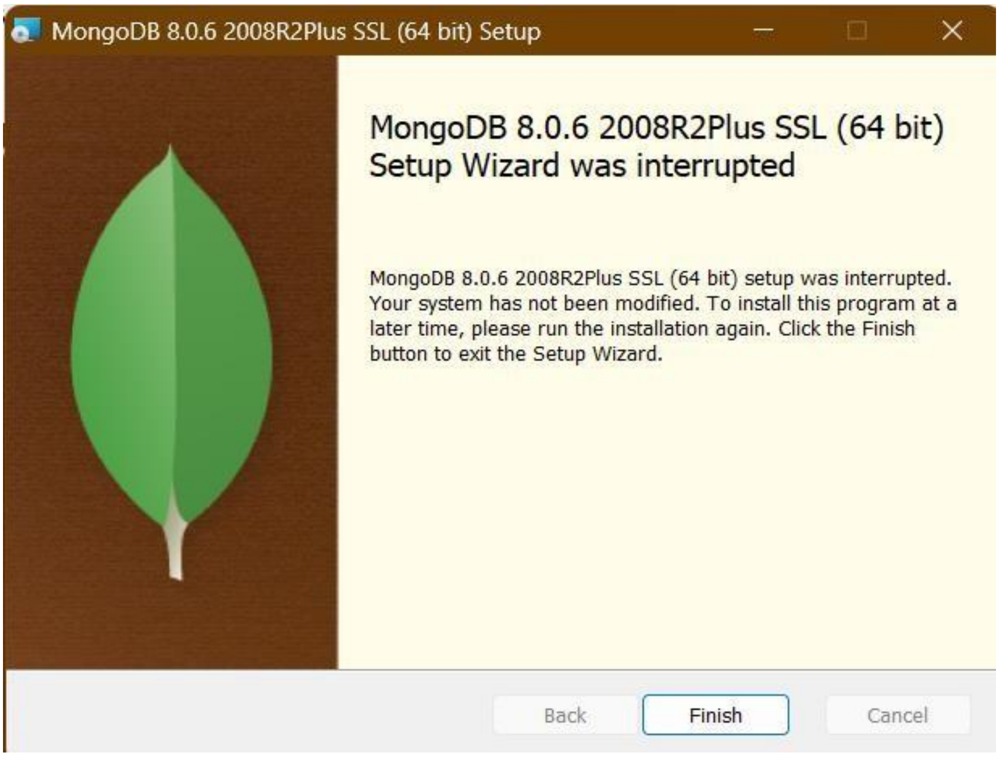


MongoDB 8.0.6 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 8.0.6 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard allows you to change the way MongoDB 8.0.6 2008R2Plus SSL (64 bit) features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Complete the installation procedure based on individual preferences



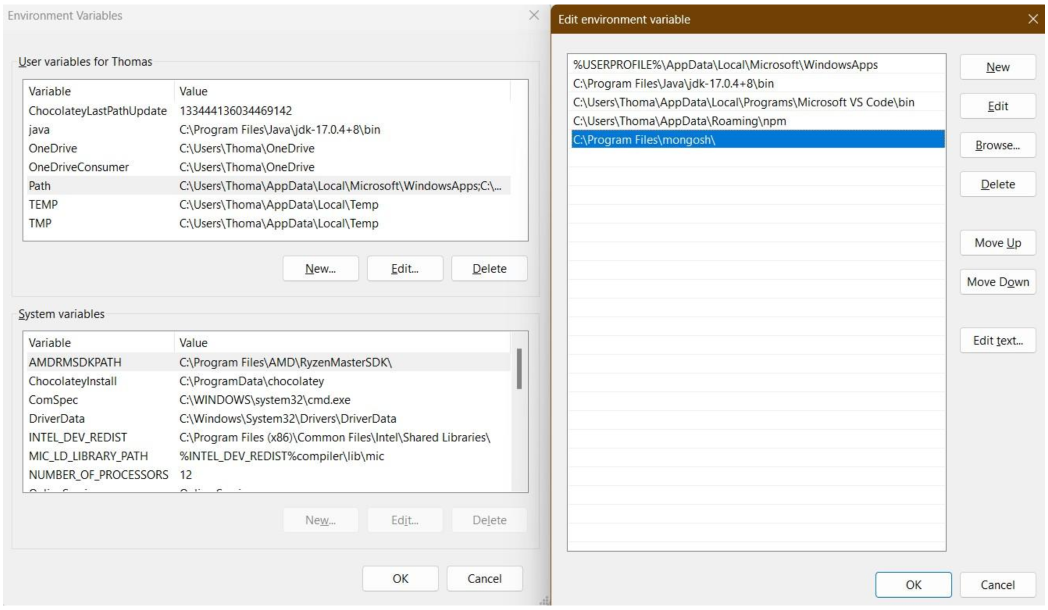
MongoDB 8.0.6 2008R2Plus SSL (64 bit) Setup

MongoDB 8.0.6 2008R2Plus SSL (64 bit) Setup Wizard was interrupted

MongoDB 8.0.6 2008R2Plus SSL (64 bit) setup was interrupted. Your system has not been modified. To install this program at a later time, please run the installation again. Click the Finish button to exit the Setup Wizard.

STEP 5:

Start the installation process by opening the downloaded file.



Environment Variables

User variables for Thomas

Variable	Value
ChocolateyLastPathUpdate	133444136034469142
java	C:\Program Files\Java\jdk-17.0.4+8\bin
OneDrive	C:\Users\Thoma\OneDrive
OneDriveConsumer	C:\Users\Thoma\OneDrive
Path	C:\Users\Thoma\AppData\Local\Microsoft\WindowsApps;C:\...
TEMP	C:\Users\Thoma\AppData\Local\Temp
TMP	C:\Users\Thoma\AppData\Local\Temp

System variables

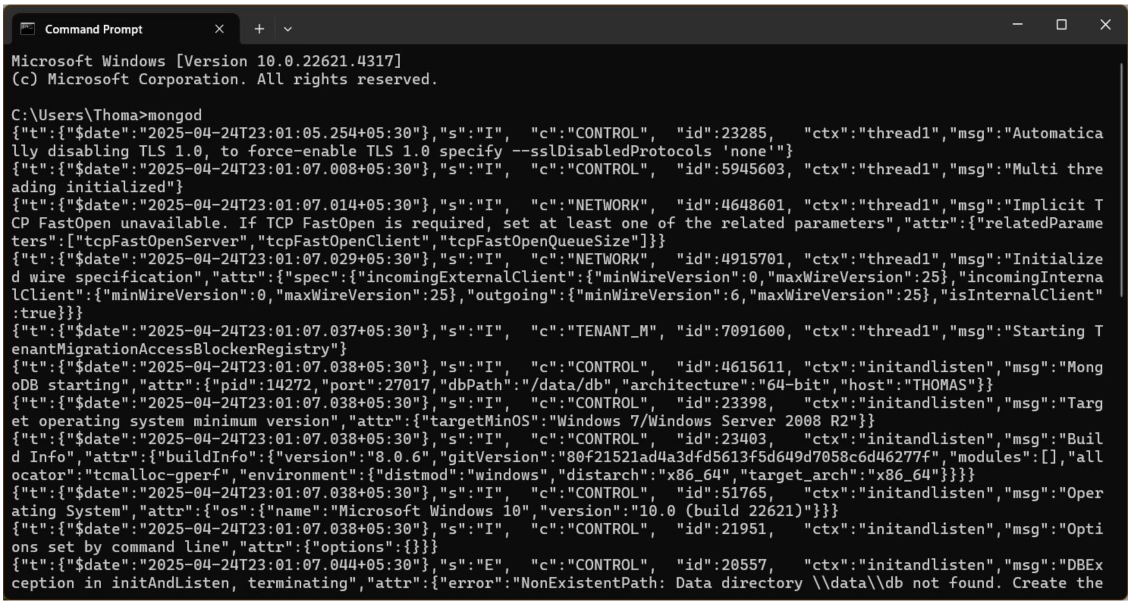
Variable	Value
AMDRMSDKPATH	C:\Program Files\AMD\RyzenMasterSDK\
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\WINDOWS\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
INTEL_DEV_REDIST	C:\Program Files (x86)\Common Files\Intel\Shared Libraries\
MIC_LD_LIBRARY_PATH	%INTEL_DEV_REDIST%compiler\lib\mic
NUMBER_OF_PROCESSORS	12

Edit environment variable

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Program Files\Java\jdk-17.0.4+8\bin
C:\Users\Thoma\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\Thoma\AppData\Roaming\npm
C:\Program Files\mongosh\
```

STEP 6:

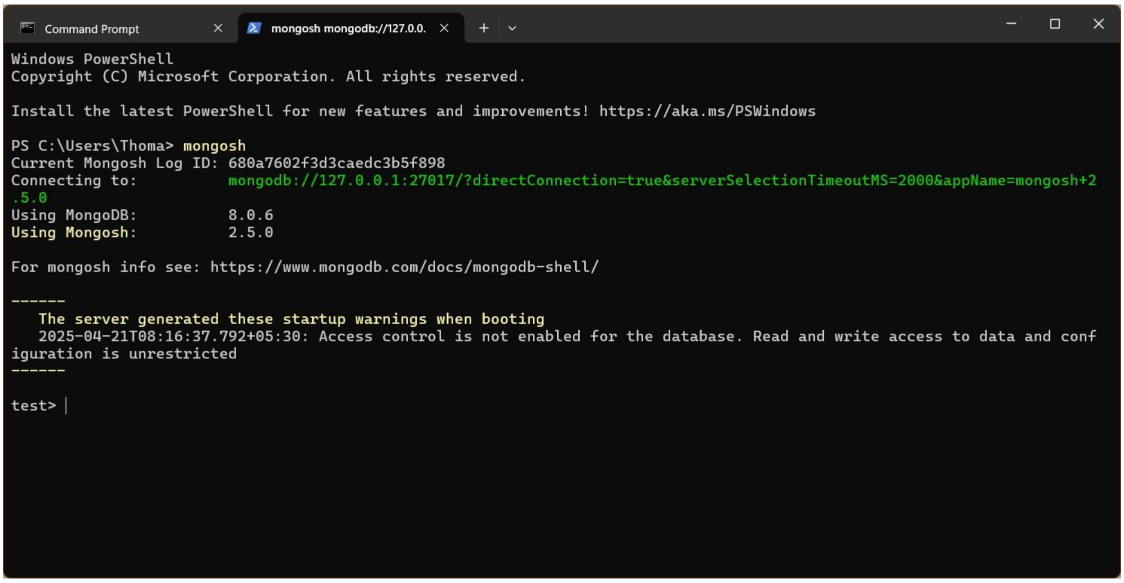
After creating an environment path, download mongosh and install. we can open the command prompt and type mongod. An instance of mongodb server is started.



```
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Thoma>mongod
{"t":{"$date":"2025-04-24T23:01:05.254+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatica
lly disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2025-04-24T23:01:07.008+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"thread1","msg":"Multi thre
ading initialized"}
{"t":{"$date":"2025-04-24T23:01:07.014+05:30"},"s":"I",  "c":"NETWORK",  "id":4648601, "ctx":"thread1","msg":"Implicit T
CP FastOpen unavailable. If TCP FastOpen is required, set at least one of the related parameters","attr":{"relatedParame
ters":["tcpFastOpenServer","tcpFastOpenClient","tcpFastOpenQueueSize"]}}
{"t":{"$date":"2025-04-24T23:01:07.029+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"thread1","msg":"Initialize
d wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":25},"incomingInterna
lClient":{"minWireVersion":0,"maxWireVersion":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient"
:true}}}
{"t":{"$date":"2025-04-24T23:01:07.037+05:30"},"s":"I",  "c":"TENANT_M", "id":7091600, "ctx":"thread1","msg":"Starting T
enantMigrationAccessBlockerRegistry"}
{"t":{"$date":"2025-04-24T23:01:07.038+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"Mong
oDB starting","attr":{"pid":14272,"port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"THOMAS"}}
{"t":{"$date":"2025-04-24T23:01:07.038+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Targ
et operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2025-04-24T23:01:07.038+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Buil
d Info","attr":{"buildInfo":{"version":"8.0.6","gitVersion":"80f21521ad4a3dfd5613f5d649d7058c6d46277f","modules":[],"all
ocator":"tcmalloc-gperf","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2025-04-24T23:01:07.038+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Oper
ating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2025-04-24T23:01:07.038+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Opti
ons set by command line","attr":{"options":{}}}
{"t":{"$date":"2025-04-24T23:01:07.044+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "ctx":"initandlisten","msg":"DBEx
ception in initAndListen, terminating","attr":{"error":"NonExistentPath: Data directory \\data\\db not found. Create the
```

STEP 7:

Now take another terminal and type mongosh. This creates a client instance of mongodb in your local system.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Thoma> mongosh
Current Mongosh Log ID: 680a7602f3d3caedc3b5f898
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.5.0
Using MongoDB:          8.0.6
Using Mongosh:          2.5.0

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-04-21T08:16:37.792+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
------

test> |
```

STEP 8:

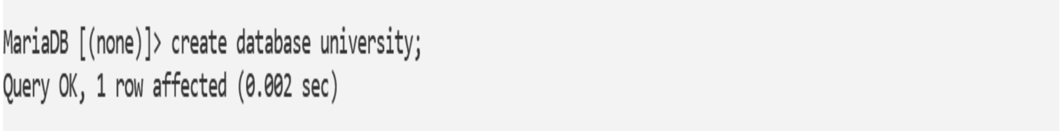
Now you can start creating new databases and use them.

PROGRAM 2

AIM: Compare Traditional MySQL/SQL Plus database and NoSQL database such as MongoDB.

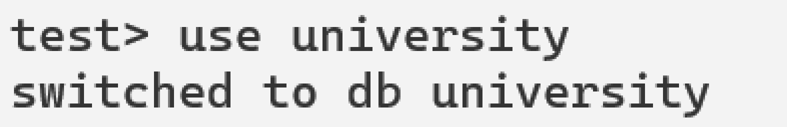
1. Create database university

SQL



```
MariaDB [(none)]> create database university;
Query OK, 1 row affected (0.002 sec)
```

MongoDB

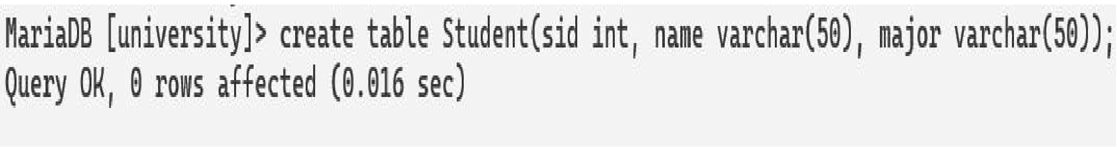


```
test> use university
switched to db university
```

2. Create table and collection

SQL

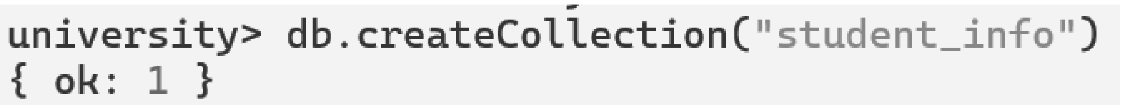
Create a table named Student



```
MariaDB [university]> create table Student(sid int, name varchar(50), major varchar(50));
Query OK, 0 rows affected (0.016 sec)
```

MongoDB

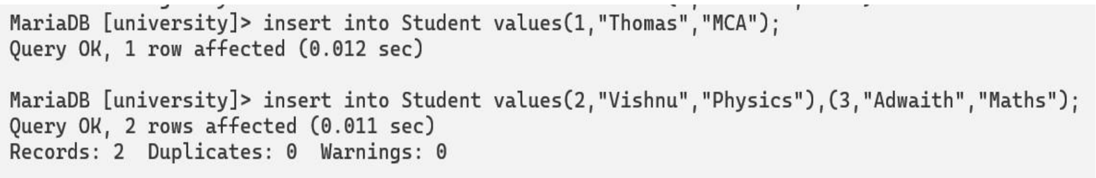
Create a collection named student_info



```
university> db.createCollection("student_info")
{ ok: 1 }
```

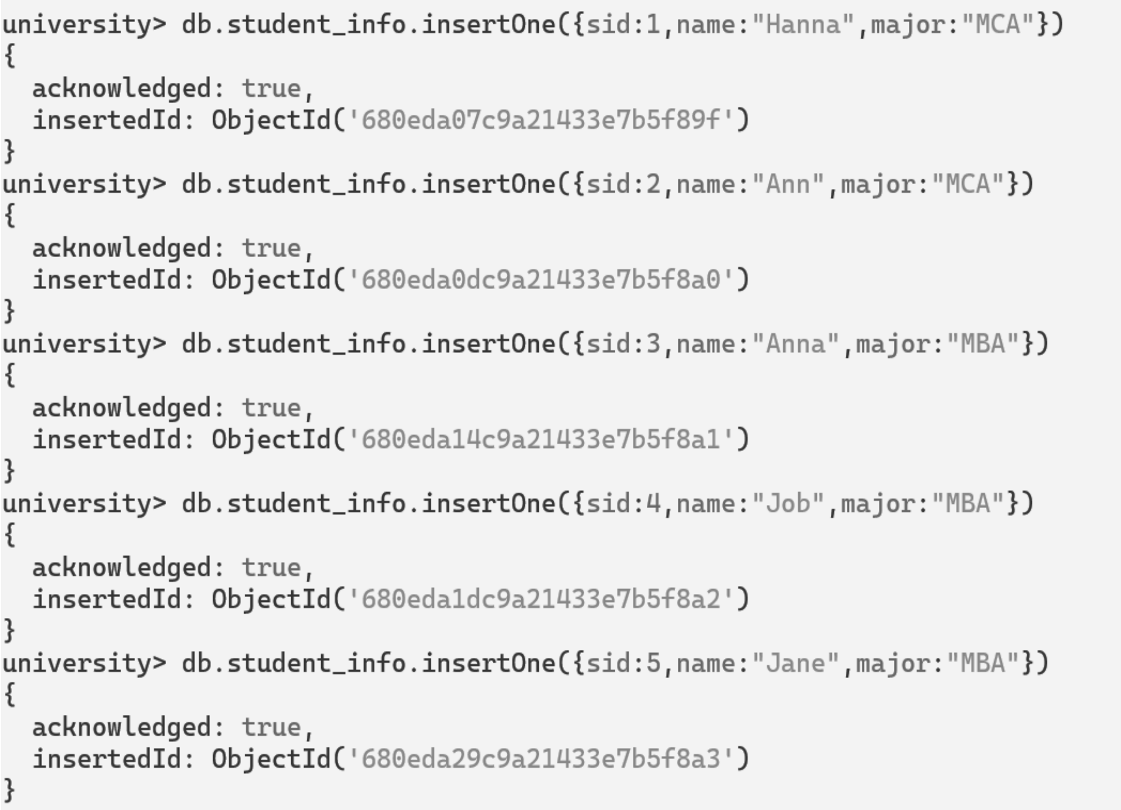
3. Insert data

SQL

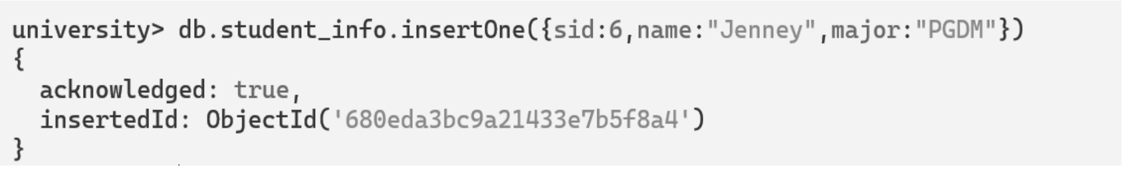


```
MariaDB [university]> insert into Student values(1,"Thomas","MCA");
Query OK, 1 row affected (0.012 sec)

MariaDB [university]> insert into Student values(2,"Vishnu","Physics"),(3,"Adwaith","Maths");
Query OK, 2 rows affected (0.011 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

MongoDB



```
university> db.student_info.insertOne({sid:1,name:"Hanna",major:"MCA"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda07c9a21433e7b5f89f')
}
university> db.student_info.insertOne({sid:2,name:"Ann",major:"MCA"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda0dc9a21433e7b5f8a0')
}
university> db.student_info.insertOne({sid:3,name:"Anna",major:"MBA"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda14c9a21433e7b5f8a1')
}
university> db.student_info.insertOne({sid:4,name:"Job",major:"MBA"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda1dc9a21433e7b5f8a2')
}
university> db.student_info.insertOne({sid:5,name:"Jane",major:"MBA"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda29c9a21433e7b5f8a3')
}
```

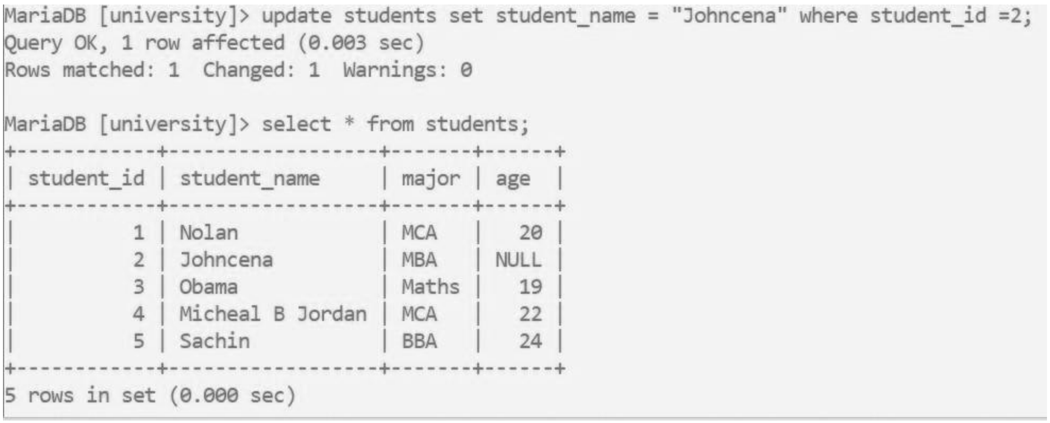


```
university> db.student_info.insertOne({sid:6,name:"Jenney",major:"PGDM"})
{
  acknowledged: true,
  insertedId: ObjectId('680eda3bc9a21433e7b5f8a4')
}
```

4. Perform update operation

SQL

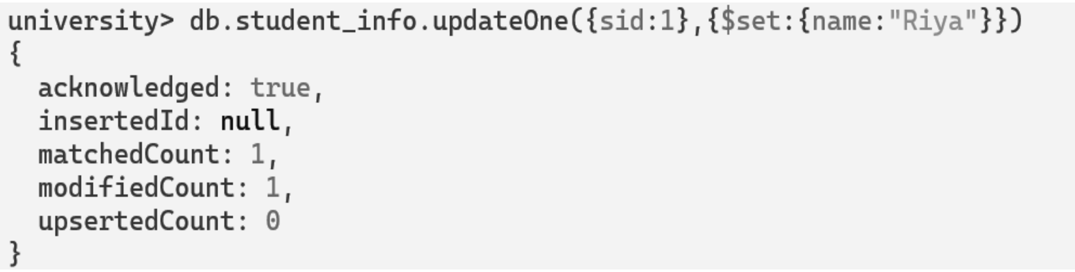
Update the name of student whose id is 2



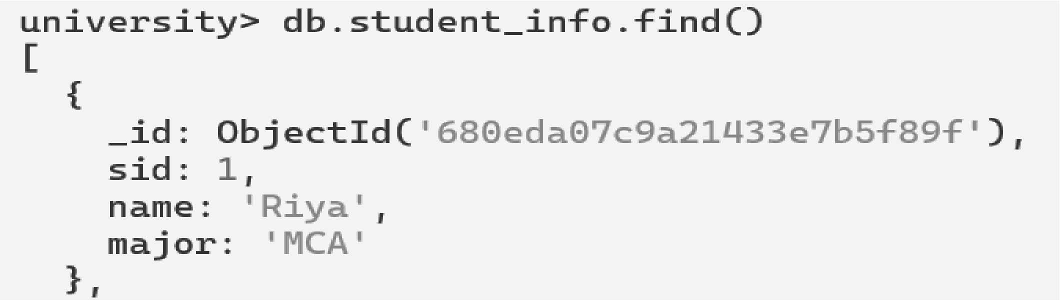
```
MariaDB [university]> update students set student_name = "Johncena" where student_id =2;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [university]> select * from students;
+------------+-----------------+-------+------+
| student_id | student_name    | major | age  |
+------------+-----------------+-------+------+
|          1 | Nolan           | MCA   |   20 |
|          2 | Johncena        | MBA   | NULL |
|          3 | Obama           | Maths |   19 |
|          4 | Micheal B Jordan | MCA   |   22 |
|          5 | Sachin          | BBA   |   24 |
+------------+-----------------+-------+------+
5 rows in set (0.000 sec)
```

MongoDB

Update the name of student whose sid is 1



```
university> db.student_info.updateOne({sid:1},{$set:{name:"Riya"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

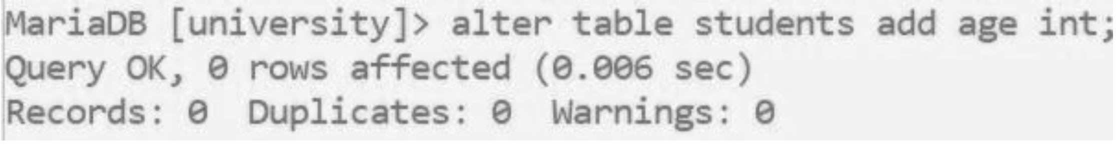


```
university> db.student_info.find()
[
  {
    _id: ObjectId('680eda07c9a21433e7b5f89f'),
    sid: 1,
    name: 'Riya',
    major: 'MCA'
  },
```

5. Alter by adding new field

SQL

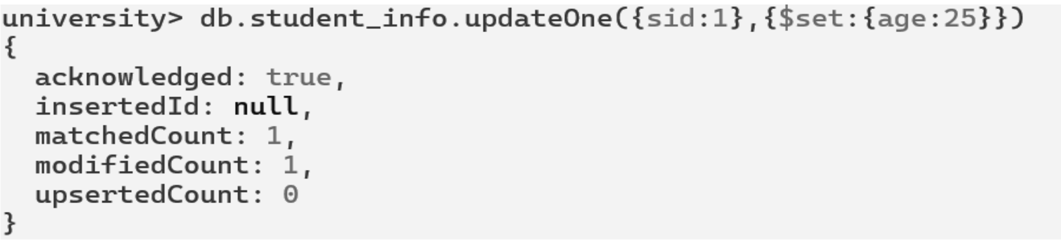
Add a new column with name age



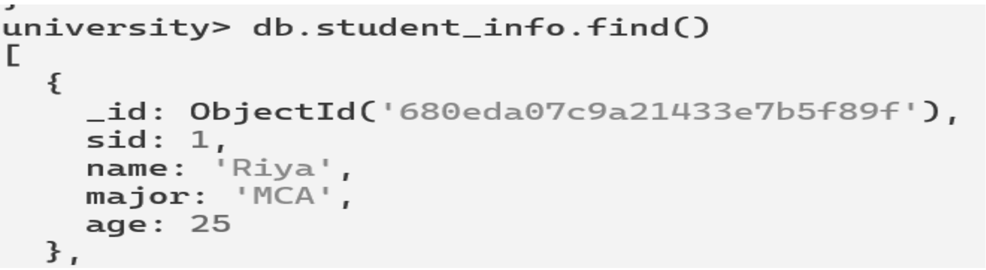
```
MariaDB [university]> alter table students add age int;
Query OK, 0 rows affected (0.006 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

MongoDB

Add a new column with name age



```
university> db.student_info.updateOne({sid:1},{$set:{age:25}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

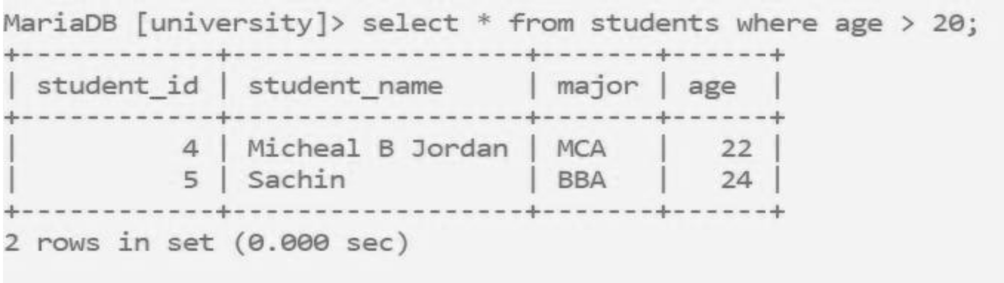


```
university> db.student_info.find()
[
  {
    _id: ObjectId('680eda07c9a21433e7b5f89f'),
    sid: 1,
    name: 'Riya',
    major: 'MCA',
    age: 25
  },
```

6. Display details of students based on a condition

SQL

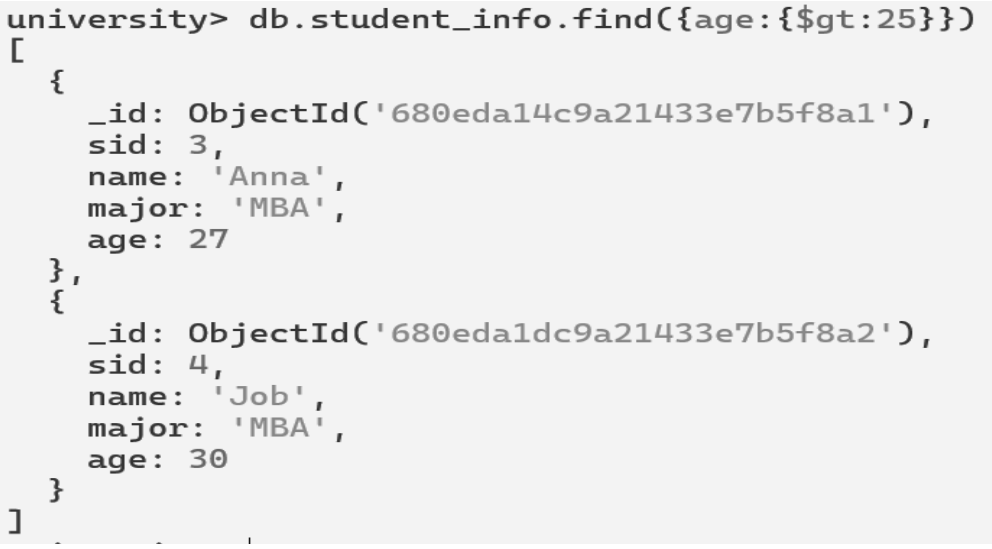
Display the details of student whose age is greater than 20



```
MariaDB [university]> select * from students where age > 20;
+------------+-----------------+-------+------+
| student_id | student_name    | major | age  |
+------------+-----------------+-------+------+
|          4 | Micheal B Jordan | MCA   |   22 |
|          5 | Sachin          | BBA   |   24 |
+------------+-----------------+-------+------+
2 rows in set (0.000 sec)
```

MongoDB

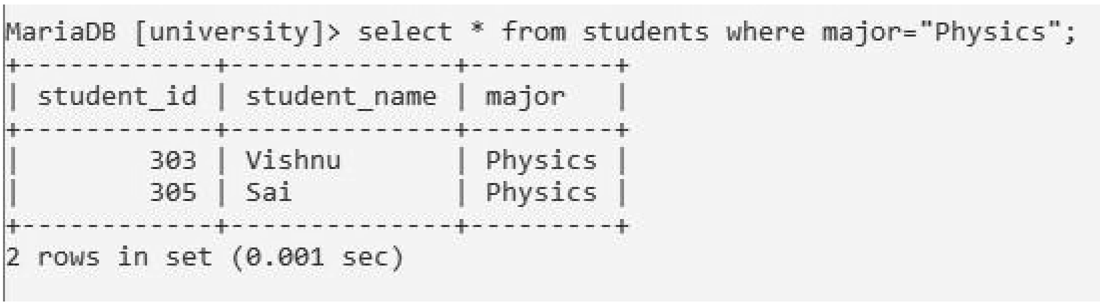
Display the details of student whose age is greater than 25



```
university> db.student_info.find({age:{$gt:25}})
[
  {
    _id: ObjectId('680eda14c9a21433e7b5f8a1'),
    sid: 3,
    name: 'Anna',
    major: 'MBA',
    age: 27
  },
  {
    _id: ObjectId('680eda1dc9a21433e7b5f8a2'),
    sid: 4,
    name: 'Job',
    major: 'MBA',
    age: 30
  }
]
```

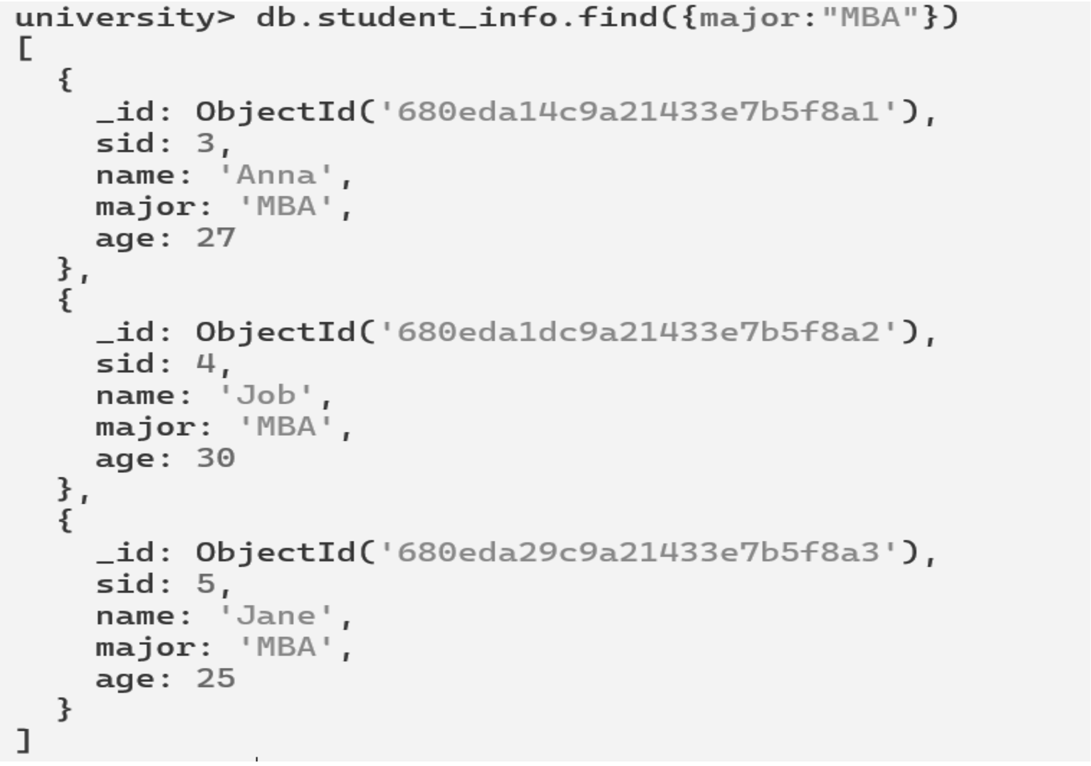
7. Select students with respect to their major

SQL



```
MariaDB [university]> select * from students where major="Physics";
+------------+--------------+---------+
| student_id | student_name | major   |
+------------+--------------+---------+
|        303 | Vishnu       | Physics |
|        305 | Sai          | Physics |
+------------+--------------+---------+
2 rows in set (0.001 sec)
```

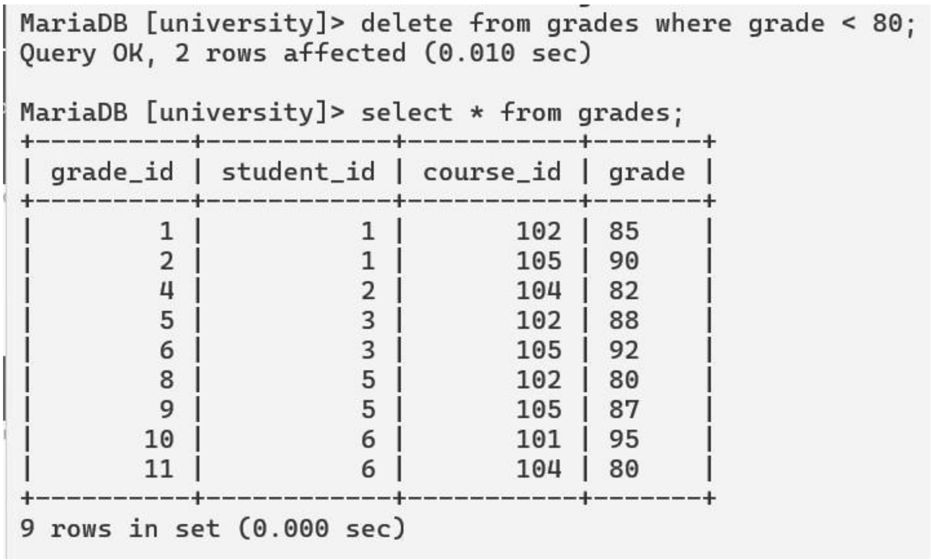
MongoDB



```
university> db.student_info.find({major:"MBA"})
[
  {
    _id: ObjectId('680eda14c9a21433e7b5f8a1'),
    sid: 3,
    name: 'Anna',
    major: 'MBA',
    age: 27
  },
  {
    _id: ObjectId('680eda1dc9a21433e7b5f8a2'),
    sid: 4,
    name: 'Job',
    major: 'MBA',
    age: 30
  },
  {
    _id: ObjectId('680eda29c9a21433e7b5f8a3'),
    sid: 5,
    name: 'Jane',
    major: 'MBA',
    age: 25
  }
]
```

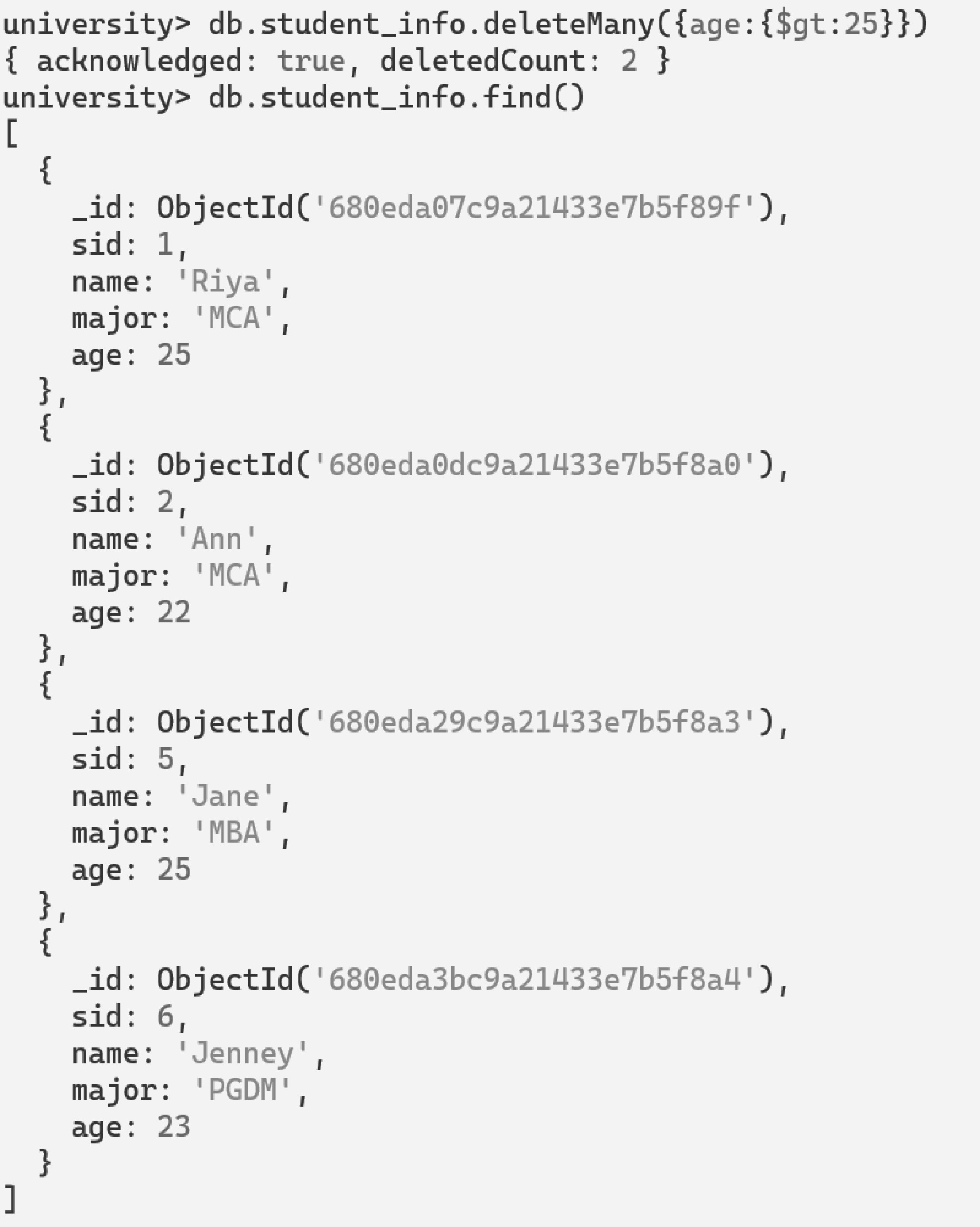
8. Deletion in MongoDB and MySQL

SQL



```
MariaDB [university]> delete from grades where grade < 80;
Query OK, 2 rows affected (0.010 sec)

MariaDB [university]> select * from grades;
+----------+------------+-----------+-------+
| grade_id | student_id | course_id | grade |
+----------+------------+-----------+-------+
|        1 |          1 |       102 | 85    |
|        2 |          1 |       105 | 90    |
|        4 |          2 |       104 | 82    |
|        5 |          3 |       102 | 88    |
|        6 |          3 |       105 | 92    |
|        8 |          5 |       102 | 80    |
|        9 |          5 |       105 | 87    |
|       10 |          6 |       101 | 95    |
|       11 |          6 |       104 | 80    |
+----------+------------+-----------+-------+
9 rows in set (0.000 sec)
```

MongoDB



```
university> db.student_info.deleteMany({age:{$gt:25}})
{ acknowledged: true, deletedCount: 2 }
university> db.student_info.find()
[
  {
    _id: ObjectId('680eda07c9a21433e7b5f89f'),
    sid: 1,
    name: 'Riya',
    major: 'MCA',
    age: 25
  },
  {
    _id: ObjectId('680eda0dc9a21433e7b5f8a0'),
    sid: 2,
    name: 'Ann',
    major: 'MCA',
    age: 22
  },
  {
    _id: ObjectId('680eda29c9a21433e7b5f8a3'),
    sid: 5,
    name: 'Jane',
    major: 'MBA',
    age: 25
  },
  {
    _id: ObjectId('680eda3bc9a21433e7b5f8a4'),
    sid: 6,
    name: 'Jenney',
    major: 'PGDM',
    age: 23
  }
]
```