hdoop@admin1-HP-280-G4-MT-Business-PC:~$ mongo -version

MongoDB shell version v5.0.1

Build Info: {

"version": "5.0.1",

"gitVersion": "318fd9cabc59dc9651f3189b622af6e06ab6cd33",

"openSSLVersion": "OpenSSL 1.1.1f 31 Mar 2020",

"modules": [],

"allocator": "tcmalloc",

"environment": {

"distmod": "ubuntu2004",

"distarch": "x86\_64",

"target\_arch": "x86\_64"

}

}

hdoop@admin1-HP-280-G4-MT-Business-PC:~$ mongo

MongoDB shell version v5.0.1

connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb

Implicit session: session { "id" : UUID("c24bdcc9-e1b2-4e67-bbd1-2bff28d5aaa3") }

MongoDB server version: 5.0.1

================

Warning: the "mongo" shell has been superseded by "mongosh",

which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in

an upcoming release.

We recommend you begin using "mongosh".

For installation instructions, see

https://docs.mongodb.com/mongodb-shell/install/

================

---

The server generated these startup warnings when booting:

2022-04-05T08:51:47.033+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem

2022-04-05T08:51:50.114+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

> use adamya

switched to db adamya

> db.student.insertOne({Name:"Tom",Age:26,Section: C})

uncaught exception: ReferenceError: C is not defined :

@(shell):1:41

> db.student.insertOne({Name:"Tom",Age:26,Roll no:134})

uncaught exception: SyntaxError: missing : after property id :

@(shell):1:45

> db.student.insertOne({Name:"Tom",Age:26,Rollno:134})

{

"acknowledged" : true,

"insertedId" : ObjectId("624bbc5d8da19f669dbc0b1d")

}

> show collections

student

> db.student.insertMany([{Name:"Zendaya",Age:23,Rollno:136},{

... Name:"Timothe",Age:25,Rollno:125},{Name:"Josephine",Age:26,Rollno:167}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("624bbe1d8da19f669dbc0b1e"),

ObjectId("624bbe1d8da19f669dbc0b1f"),

ObjectId("624bbe1d8da19f669dbc0b20")

]

}

> show dbs

adamya 0.000GB

admin 0.000GB

config 0.000GB

gaurav 0.000GB

local 0.000GB

practise 0.000GB

punith 0.000GB

rishik 0.000GB

vaishnavi195 0.000GB

> db.student.insertMany([{Name:"Avani",Age:28,Rollno:005},{Name:"Naidile",Age:20,Rollno:165},{Name:"Vidisha",Age:21,Rollno:158}])

{

"acknowledged" : true,

"insertedIds" : [

ObjectId("624bc0d88da19f669dbc0b21"),

ObjectId("624bc0d88da19f669dbc0b22"),

ObjectId("624bc0d88da19f669dbc0b23")

]

}

> db.student.find()

{ "\_id" : ObjectId("624bbc5d8da19f669dbc0b1d"), "Name" : "Tom", "Age" : 26, "Rollno" : 134 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b1e"), "Name" : "Zendaya", "Age" : 23, "Rollno" : 136 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b1f"), "Name" : "Timothe", "Age" : 25, "Rollno" : 125 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b20"), "Name" : "Josephine", "Age" : 26, "Rollno" : 167 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b21"), "Name" : "Avani", "Age" : 28, "Rollno" : 5 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b22"), "Name" : "Naidile", "Age" : 20, "Rollno" : 165 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b23"), "Name" : "Vidisha", "Age" : 21, "Rollno" : 158 }

> db.student.updateOne({Name:"Avani"},{$set:{Age:24}})

{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }

> db.student.updateMany({Age:26},{$set:{Age:22}})

{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }

> db.student.find()

{ "\_id" : ObjectId("624bbc5d8da19f669dbc0b1d"), "Name" : "Tom", "Age" : 22, "Rollno" : 134 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b1e"), "Name" : "Zendaya", "Age" : 23, "Rollno" : 136 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b1f"), "Name" : "Timothe", "Age" : 25, "Rollno" : 125 }

{ "\_id" : ObjectId("624bbe1d8da19f669dbc0b20"), "Name" : "Josephine", "Age" : 22, "Rollno" : 167 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b21"), "Name" : "Avani", "Age" : 24, "Rollno" : 5 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b22"), "Name" : "Naidile", "Age" : 20, "Rollno" : 165 }

{ "\_id" : ObjectId("624bc0d88da19f669dbc0b23"), "Name" : "Vidisha", "Age" : 21, "Rollno" : 158 }

>