

```

mongo
MongoDB shell version v3.6.8
connecting to: mongoddb://127.0.0.1:27017
Implicit session: session { "id" : UUID("f7f307ee-ea44-4d9f-bc1f-c49f34e45861") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-13T14:10:17.384+0530 I STORAGE [initandlisten]
2022-04-13T14:10:17.384+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem
is strongly recommended with the WiredTiger storage engine
2022-04-13T14:10:17.384+0530 I STORAGE [initandlisten] **          See
http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-13T14:10:21.311+0530 I CONTROL [initandlisten]
20use Premsaidb;
switched to db Premsaidb
> db
Premsaidb
> db.Student.insert({_id:1,StudName:"Prem Sai",Grade:"V",Hobbies:"Reading books"});
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: Premsaidb.Student index: _id_ dup key:
{ : 1.0 }"
  }
})
> db.Student.insert({_id:2,StudName:"Prem Sai",Grade:"V",Hobbies:"Reading books"});
WriteResult({ "nInserted" : 1 })
> db.Student.update({_id:2,StudName:"Prem Sai",Grade:"V"},{$set:{Hobbies:"Playing"}},
{upsert:true});
2022-04-13T14:49:10.362+0530 E QUERY [thread1] SyntaxError: unterminated string literal
@(shell):1:72
> db.Student.update({_id:2,StudName:"Prem Sai",Grade:"V"},{$set:{Hobbies:"Playing"}},
{upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.update({_id:2},{ $set:{StudName:"Nithish"}},Grade:"V",Hobbies:"Playing"},
{upsert:true});
2022-04-13T14:52:51.610+0530 E QUERY [thread1] SyntaxError: missing ) after argument list
@(shell):1:60
> db.Student.update({_id:2},{ $set:{StudName:"Nithish"},Grade:"V",Hobbies:"Playing"},{upsert:true});
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 0,
  "nModified" : 0,
  "writeError" : {
    "code" : 9,
    "errmsg" : "Unknown modifier: Grade"
  }
})
> db.Student.update({_id:2},{ $set:{StudName:"Nithish"},Grade:"V",Hobbies:"Playing"},{upsert:true});
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 0,
  "nModified" : 0,
  "writeError" : {
    "code" : 9,
    "errmsg" : "Unknown modifier: Grade"
  }
})

```

```

    }
  })
> db.Student.find({StudName:"Prem Sai"},{_id:1});
{ "_id" : 1 }
{ "_id" : 2 }
> db.Student.insert({_id:3,StudName:"Tharun",Grade:"V",Hobbies:"Sleeping"});
WriteResult({ "nInserted" : 1 })
> db.Student.find({StudName:"Tharun"},{_id:1,Hobbies:1,Grade:0});
Error: error: {
  "ok" : 0,
  "errmsg" : "Projection cannot have a mix of inclusion and exclusion.",
}

> db.Student.find({StudName:"Tharun"},{_id:1,Hobbies:1,Grade:0});
Error: error: {
  "ok" : 0,
  "errmsg" : "Projection cannot have a mix of inclusion and exclusion.",
  "code" : 2,
  "codeName" : "BadValue"
}
> db.Student.find({id:3},{StudName:1,Grade:1});
> db.Student.find({id:3});
>
> db.Student.find({_id:3});
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({_id:3},{StudName:1,Grade:1,Hobbies:0});
... db.Student.find({id:3});
...
... ^C

> db.Student.find({_id:3},{StudName:1,Grade:1});
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V" }
> db.Student.find({_id:3},{StudName:1,Grade:1,Hobbies:0});
Error: error: {
  "ok" : 0,
  "errmsg" : "Projection cannot have a mix of inclusion and exclusion.",
  "code" : 2,
  "codeName" : "BadValue"
}
> db.Student.find({StudName:{$eq:"Prem Sai"}}).pretty();
{
  "_id" : 1,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Reading books"
}
{
  "_id" : 2,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Playing"
}
> db.Student.find({Hobbies:{$in:['Sleeping','Playing']}}).pretty();
2022-04-13T15:06:37.670+0530 E QUERY [thread1] SyntaxError: missing ) in parenthetical
@ (shell):1:29
> db.Student.find({Hobbies:{$in:['Sleeping','Playing']}}).pretty();
{
  "_id" : 2,

```

```

    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Playing"
}
{ db.Student.find({Hobbies:{$in:['Sleeping','Playing']}}).pretty();
{
    "_id" : 2,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Playing"
}
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({Hobbies:{$in:['Sleeping','Playing']}}).pretdb.Student.find({Hobbies:{$in:
['Sleeping','Playing']}}).pretdb.Student.find({Hobbies:{$in:['Sleeping','Playing']}}).pretty()
{
    "_id" : 2,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Playing"
}
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:/^T/}).pretty();
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:/^T/}).pretty();
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:/^T/}).pretty();
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:Tharun}).pretty();
2022-04-13T15:11:54.389+0530 E QUERY [thread1] ReferenceError: Tharun is not defined :
@($shell):1:18
> db.Student.find({StudName:"Tharun"}).pretty();
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:"Tharun"});
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.find({StudName:"Tharun"}).pretty();
{ "_id" : 3, "StudName" : "Tharun", "Grade" : "V", "Hobbies" : "Sleeping" }
> db.Student.save({StudName:"Nithish",Grade:"VII"});
WriteResult({ "nInserted" : 1 })
> db.Student.update({_id:4},{ $set: {Location:"Hyderabad" }});
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.Student.update({_id:3},{ $set: {Location:"Hyderabad" }});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find({StudName:i$/})
{ "_id" : 1, "StudName" : "Prem Sai", "Grade" : "V", "Hobbies" : "Reading books" }
{ "_id" : 2, "StudName" : "Prem Sai", "Grade" : "V", "Hobbies" : "Playing" }
> db.Student.find({StudName:i$/}).pretty();
{
    "_id" : 1,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Reading books"
}
{
    "_id" : 2,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Playing"
}

```

```

}
> db.Student.update({_id:3},{ $set: {Hobbies:null}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.count();
4
> db.Student.count({StudName:"Prem Sai"});
2
> db.Student.find({Grade:"V"}).limit(3).pretty();
{
  "_id" : 1,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Reading books"
}
{
  "_id" : 2,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Playing"
}
{
  "_id" : 3,
  "StudName" : "Tharun",
  "Grade" : "V",
  "Hobbies" : null,
  "Location" : "Hyderabad"
}
> db.Student.find.sort({StudName:1}).pretty();
2022-04-13T15:22:34.358+0530 E QUERY [thread1] TypeError: db.Student.find.sort is not a function :
@(@shell):1:1
> db.Student.find().sort({StudName:1}).pretty();
{
  "_id" : ObjectId("62569b7845e4d1901db32cd8"),
  "StudName" : "Nithish",
  "Grade" : "VII"
}
{
  "_id" : 1,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Reading books"
}
{
  "_id" : 2,
  "StudName" : "Prem Sai",
  "Grade" : "V",
  "Hobbies" : "Playing"
}
{
  "_id" : 3,
  "StudName" : "Tharun",
  "Grade" : "V",
  "Hobbies" : null,
  "Location" : "Hyderabad"
}
> db.Student.find().sort({StudName:-1}).pretty();
{

```

```

    "_id" : 3,
    "StudName" : "Tharun",
    "Grade" : "V",
    "Hobbies" : null,
    "Location" : "Hyderabad"
  }
  {
    "_id" : 1,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Reading books"
  }
  {
    "_id" : 2,
    "StudName" : "Prem Sai",
    "Grade" : "V",
    "Hobbies" : "Playing"
  }
  {
    "_id" : ObjectId("62569b7845e4d1901db32cd8"),
    "StudName" : "Nithish",
    "Grade" : "VII"
  }
> db.Student.find().skip(3).pretty();
{
  "_id" : ObjectId("62569b7845e4d1901db32cd8"),
  "StudName" : "Nithish",
  "Grade" : "VII"
}
> db.createCollection("food");
{ "ok" : 1 }
> db
Premsaidb
> db.food.insert({_id:1,fruits:['grapes','mango','apple']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({_id:2,fruits:['grapes','mango','cherry']});
WriteResult({ "nInserted" : 1 })
> db.food.insert({_id:3,fruits:['banana','mango']});
WriteResult({ "nInserted" : 1 })
> db.food.find({fruits:['grapes','mango','apple']}).pretty();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> db.food.find({fruits:['grapes','mango','apple']});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
>
> db.food.find({fruits:['grapes','mango','apple']}).pretty();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> db.food.find({'fruits.1':'grapes'});
> db.food.find({fruits:['grapes','mango','apple']}).pretty()
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> db.food.find({'fruits.1':'grapes'});
> db.food.find({'fruits.1':'mango'});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
> db.food.find({"fruits":{"$size:2}})
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
> db.food.find({_id:1},{"fruits":{"$slice:2}})

```

```
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
> db.food.find({fruits:{$all:["mango","grapes"]}})
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
> db.food.update({_id:3},{ $set:{'fruits.1':'apple'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.update({_id:2},{ $push:{price:{grapes:80,mango:200,cherry:100}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```