

Wednesday, August 17, 2016
12:04 PM

Set path : C:\Program Files\MongoDB\Server\3.0\bin

net start | findstr MongoDB
net stop MongoDB
Mongod --remove

Run mongod

Run mongo

Command:

show dbs
show collections
show tables
db.getCollectionNames()

db
use foo

mongo --host 192.168.0.5 --port 30000

start mongod --port 40000 --replSet "Tec" --dbpath c:\db\path1

start mongod --port 50000 --replSet "Tec" --dbpath c:\db\path2

```
var TecConfig={_id: "Tec", members: [{_id: 0, host: '192.168.0.5:30000', priority: 10},
{ _id: 0, host: '192.168.0.5:40000'},
{ _id: 0, host: '192.168.0.5:50000', arbiterOnly: true}]]};
```

TecConfig

icacls d:\mongoadmin /grant mongodb:(OI) (CI)F

Install service:

mongod --install --config c:\TechGeek\etc\mongodb.cfg --serviceUser <username> --
servicePassword <password>

D:\MONGO~1\config>sc qc mongodb

[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: mongodb
TYPE : 10 WIN32_OWN_PROCESS
START_TYPE : 2 AUTO_START

```

ERROR_CONTROL    : 1  NORMAL
BINARY_PATH_NAME : "C:\Program Files\MongoDB\Server\3.0\bin\mongod.exe"
-f c:\TechGeek\etc\mongodb.cfg --servicex
LOAD_ORDER_GROUP :
TAG              : 0
DISPLAY_NAME     : MongoDB
DEPENDENCIES     :
SERVICE_START_NAME : LocalSystem

```

c:\TechGeek\etc\mongodb.cfg

```

# data place holder
dbpath=c:/TechGeek/db
#log
logpath=c:/TechGeek/log/mongo-server.log
#Debug level
verbose=v
#port
port=30000
#replSet
replSet=Tec

```

Taskkill

```

Tec:PRIMARY> db.collection.find("mycol")
Tec:PRIMARY> db.mycol.find()
{ "_id" : ObjectId("57c619844d5d7d2d640aa38f"), "user_id" : "1" }
Tec:PRIMARY>

```

Drop Database:

```

Kuiper:PRIMARY> show dbs
  local 0.078GB
  myDB 0.078GB
Kuiper:PRIMARY> use myDB
switched to db myDB
Kuiper:PRIMARY> db.getCollectionNames()
[ "mycollection", "system.indexes" ]
Kuiper:PRIMARY> db.dropDatabase()
{ "dropped" : "myDB", "ok" : 1 }
Kuiper:PRIMARY> show dbs
  local 0.078GB
Kuiper:PRIMARY>

```

```

db.mycollection.find()
db.mycollection.find(item_nbr : 10002)

```

```

db.mycollection.insert({item_nbr:10001, item_desc:"new product", price:10.34})
db.mycollection.createIndex({"item_nbr":10001})

```

```
db.mycollection.find().sort({"price" : -1})
```

Operation	Syntax	Example	RDBMS Equivalent
Equality	{<key>:<value> }	db.mycol.find({"by":"tutorial point").pretty()	where by = 'tutorial point'
Less Than	{<key>:{\$lt:<value>}}	db.mycol.find({"likes":{\$lt:50}}).pretty()	where likes < 50
Less Than Equals	{<key>:{\$lte:<value>}}	db.mycol.find({"likes":{\$lte:50}}).pretty()	where likes <= 50
Greater Than	{<key>:{\$gt:<value>}}	db.mycol.find({"likes":{\$gt:50}}).pretty()	where likes > 50
Greater Than Equals	{<key>:{\$gte:<value>}}	db.mycol.find({"likes":{\$gte:50}}).pretty()	where likes >= 50
Not Equals	{<key>:{\$ne:<value>}}	db.mycol.find({"likes":{\$ne:50}}).pretty()	where likes != 50

```
db.item.find({"_id":{$lt:4}}).pretty()
```

```
db.item.find({"_id":{$gt:1}, "_item_nbr":{$ne:1003}}).pretty()
```

```
Kuiper:PRIMARY> db.item.find(
... {
... "_id":{$gt:1},
... "_item_nbr":{$ne:1003}
... }
... ).pretty()
{
  "_id" : 2,
  "_item_nbr" : 1000,
  "item_desc" : "contigo water bottle",
  "price" : 7.78
}
```

Update with set:

```
Kuiper:PRIMARY> db.item.find(
... {
... "_id":{$gt:1},
... "_item_nbr":{$ne:1003}
... }
... ).pretty()
{
  "_id" : 2,
  "_item_nbr" : 1000,
  "item_desc" : "contigo water bottle",
  "price" : 7.78
}
```

```
Remove # db.item.remove({"_id":2})
```

Projection

```
Kuiper:PRIMARY> db.item.find({},{"_item_nbr":1,"price":2}).pretty()
```

```
{ "_id" : 1, "_item_nbr" : 1001, "price" : 1342.43 }
```

```
{ "_id" : 3, "_item_nbr" : 1003, "price" : 72.52 }
```

```
{ "_id" : 4, "_item_nbr" : 1004, "price" : 72.52 }
```

```
db.item.find({},{"_item_nbr":1,"price":2}).pretty().limit(4).sort({"price":-1}).skip(2)
```

Limit the record

Skip the record

Sort to ascend & descending

```
Kuiper:PRIMARY> db.Inventory_Log.aggregate( { $group : { _id:"$action", total :{$sum:1} } } );
```

```
{ "_id" : "CREATED", "total" : 1 }
```

```
{ "_id" : "DELIVERY_ASSIGNED", "total" : 2 }
```

```
{ "_id" : "OPEN_TRAILER", "total" : 5 }
```

```
db.Inventory_Log.aggregate( { $group : { _id:"$item_nbr", total :{$sum:1} } }, { "$project" : { "total" : 1 , "_id" : 0 , "order_status" : "$_id" } } );
```

```
{ "aggregate" : "__collection__", "pipeline" : [ { "$group" : { "_id" : "$order_status", "total" : { "$sum" : 1 } } }, { "$project" : { "total" : 1 , "_id" : 0 , "order_status" : "$_id" } } ] }
```

```
db.Inventory_Log.aggregate(
  { "$group" : { "item_nbr":"$item_nbr", "action" : "$action", "total_qty" : { "$sum" : "$qty" }, },
  { "$project" : { "total_qty" : 1 } }
)
```

```
db.sequence.insert({_id: "Inv_req_id",seq: 0})
```

```
db.Inventory_Log.aggregate(
  {
    $group : {
      _id : { item_nbr: "$item_nbr", status: "$action" },
      total : { $sum : 1 }
    }
  }
);
```

```
{
  "_id" : 10011,
  "_class" : "com.kuiper.loadCustOrd.model.CustomerOrder",
  "store_nbr" : 3442,
  "store_name" : "FARM",
  "order_date" : "Sep 20, 2016 3:51:33 PM",
  "ship_date" : "Sep 20, 2016 3:51:33 PM",
  "create_ts" : ISODate("2016-11-09T02:40:22.373Z"),
  "last_modified_ts" : ISODate("2016-11-09T02:40:22.373Z"),
  "last_modified_prg" : "CustomerOrderLoad",
  "last_modified_user" : "ADMIN",
  "order_status" : "CREATED",
  "customerOrderLine1st" : [
    {
      "order_line_nbr" : 1,
      "unit_UOM" : "EACH",
      "each_cost" : 12.979999542236328,
      "item_nbr" : 10002,
      "qty" : 1
    }
  ]
}
```

```
db.Inventory_Log.aggregate(
{
  $group:
  {
    _id: { item: '$item_nbr', action: '$action' },
    qty: { $sum: '$qty' }
  }
})
```

```
,
totalAmount: {
  $sum: { "$qty" }
}
```

```
Kuiper:PRIMARY> db.Inventory_Log.aggregate({ $group: { _id: "", last: { $max: "$_id" } } });
```