MongoDB Server avdavanced 2:

Please enter a MongoDB connection string (Default: mongodb://localhost/): localhost:27017

localhost:27017

Current Mongosh Log ID: 635a6f17b487b8990768eb4f

Connecting to: mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0

Using MongoDB: 6.0.2

Using Mongosh: 1.6.0

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).

You can opt-out by running the disableTelemetry() command.

------

The server generated these startup warnings when booting

2022-10-27T11:06:13.820+05:00: 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()

------

test> show dbs

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

mydatabase 172.00 KiB

test> db

test

test> "Hello".length

5

test> mongosh --help

Uncaught:

SyntaxError: Missing semicolon. (1:10)

> 1 | mongosh --help

| ^

2 |

test> --help

NaN

test> mongosh --nodb

Uncaught:

SyntaxError: Missing semicolon. (1:10)

> 1 | mongosh --nodb

| ^

2 |

test> mongosh -- nodb

Uncaught:

SyntaxError: Missing semicolon. (1:10)

> 1 | mongosh -- nodb

| ^

2 |

test> mongosh --nodb;

Uncaught:

SyntaxError: Missing semicolon. (1:10)

> 1 | mongosh --nodb;

| ^

2 |

test> show.help

'show databases'/'show dbs': Print a list of all available databases.

'show collections'/'show tables': Print a list of all collections for current database.

'show profile': Prints system.profile information.

'show users': Print a list of all users for current database.

'show roles': Print a list of all roles for current database.

'show log <type>': log for current connection, if type is not set uses 'global'

'show logs': Print all logs.

test> show databases

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

mydatabase 172.00 KiB

test> show collections

test> show log

[

'{"t":{"$date":"2022-10-27T11:06:08.014+05:00"},"s":"I", "c":"CONTROL", "id":20698, "ctx":"-","msg":"\*\*\*\*\* SERVER RESTARTED \*\*\*\*\*"}\n',

'{"t":{"$date":"2022-10-27T11:06:12.349+05:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}\n',

`{"t":{"$date":"2022-10-27T11:06:12.375+05:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}\n`,

'{"t":{"$date":"2022-10-27T11:06:12.417+05:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}\n',

'{"t":{"$date":"2022-10-27T11:06:12.621+05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.621+05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.622+05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.622+05:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1","msg":"Multi threading initialized"}\n',

'{"t":{"$date":"2022-10-27T11:06:12.622+05:00"},"s":"I", "c":"CONTROL", "id":23316, "ctx":"thread1","msg":"Trying to start Windows service","attr":{"serviceName":"MongoDB"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.623+05:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":4268,"port":27017,"dbPath":"C:/Program Files/MongoDB/Server/6.0/data","architecture":"64-bit","host":"ajman"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.623+05:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.625+05:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"6.0.2","gitVersion":"94fb7dfc8b974f1f5343e7ea394d0d9deedba50e","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86\_64","target\_arch":"x86\_64"}}}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.632+05:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 19041)"}}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.632+05:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{"config":"C:\\\\Program Files\\\\MongoDB\\\\Server\\\\6.0\\\\bin\\\\mongod.cfg","net":{"bindIp":"127.0.0.1","port":27017},"service":true,"storage":{"dbPath":"C:\\\\Program Files\\\\MongoDB\\\\Server\\\\6.0\\\\data","journal":{"enabled":true}},"systemLog":{"destination":"file","logAppend":true,"path":"C:\\\\Program Files\\\\MongoDB\\\\Server\\\\6.0\\\\log\\\\mongod.log"}}}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.779+05:00"},"s":"I", "c":"STORAGE", "id":22270, "ctx":"initandlisten","msg":"Storage engine to use detected by data files","attr":{"dbpath":"C:/Program Files/MongoDB/Server/6.0/data","storageEngine":"wiredTiger"}}\n',

'{"t":{"$date":"2022-10-27T11:06:12.862+05:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache\_size=7497M,session\_max=33000,eviction=(threads\_min=4,threads\_max=4),config\_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy),builtin\_extension\_config=(zstd=(compression\_level=6)),file\_manager=(close\_idle\_time=600,close\_scan\_interval=10,close\_handle\_minimum=2000),statistics\_log=(wait=0),json\_output=(error,message),verbose=[recovery\_progress:1,checkpoint\_progress:1,compact\_progress:1,backup:0,checkpoint:0,compact:0,evict:0,history\_store:0,recovery:0,rts:0,salvage:0,tiered:0,timestamp:0,transaction:0,verify:0,log:0],"}}\n',

'{"t":{"$date":"2022-10-27T11:06:13.722+05:00"},"s":"I", "c":"STORAGE", "id":4795906, "ctx":"initandlisten","msg":"WiredTiger opened","attr":{"durationMillis":860}}\n',

'{"t":{"$date":"2022-10-27T11:06:13.724+05:00"},"s":"I", "c":"RECOVERY", "id":23987, "ctx":"initandlisten","msg":"WiredTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}\n',

'{"t":{"$date":"2022-10-27T11:06:13.820+05:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}\n',

'{"t":{"$date":"2022-10-27T11:06:13.895+05:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":17,"maxWireVersion":17},"outgoing":{"minWireVersion":17,"maxWireVersion":17},"isInternalClient":true}}}\n',

'{"t":{"$date":"2022-10-27T11:06:13.895+05:00"},"s":"I", "c":"REPL", "id":5853300, "ctx":"initandlisten","msg":"current featureCompatibilityVersion value","attr":{"featureCompatibilityVersion":"6.0","context":"startup"}}\n',

'{"t":{"$date":"2022-10-27T11:06:13.895+05:00"},"s":"I", "c":"STORAGE", "id":5071100, "ctx":"initandlisten","msg":"Clearing temp directory"}\n',

'{"t":{"$date":"2022-10-27T11:06:13.903+05:00"},"s":"I", "c":"CONTROL", "id":20536, "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}\n',

'{"t":{"$date":"2022-10-27T11:06:14.713+05:00"},"s":"I", "c":"FTDC", "id":20625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/Program Files/MongoDB/Server/6.0/data/diagnostic.data"}}\n',

'{"t":{"$date":"2022-10-27T11:06:14.867+05:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}\n',

'{"t":{"$date":"2022-10-27T11:06:14.867+05:00"},"s":"I", "c":"STORAGE", "id":22262, "ctx":"initandlisten","msg":"Timestamp monitor starting"}\n',

'{"t":{"$date":"2022-10-27T11:06:15.029+05:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}\n',

'{"t":{"$date":"2022-10-27T11:06:15.029+05:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections","attr":{"port":27017,"ssl":"off"}}\n',

'{"t":{"$date":"2022-10-27T11:06:15.029+05:00"},"s":"I", "c":"CONTROL", "id":20555, "ctx":"initandlisten","msg":"Service running"}\n',

'{"t":{"$date":"2022-10-27T11:12:36.445+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50110","uuid":"3e6cb6c6-ddc7-4c81-801f-ad4d444ced56","connectionId":1,"connectionCount":1}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.453+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"client metadata","attr":{"remote":"127.0.0.1:50110","client":"conn1","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.470+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50111","uuid":"2bdfbf67-c613-4e35-830c-81cfaa9fec61","connectionId":2,"connectionCount":2}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.472+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client metadata","attr":{"remote":"127.0.0.1:50111","client":"conn2","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.642+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50112","uuid":"03100a19-0522-49c1-a3b0-e5cdb15a214c","connectionId":3,"connectionCount":3}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.642+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50113","uuid":"240a1be2-023b-4930-b619-1f78f1498dd3","connectionId":4,"connectionCount":4}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.665+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn3","msg":"client metadata","attr":{"remote":"127.0.0.1:50112","client":"conn3","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.665+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn4","msg":"client metadata","attr":{"remote":"127.0.0.1:50113","client":"conn4","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.701+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50114","uuid":"f8f6c0fd-d87b-4964-9483-f000778dbfb8","connectionId":5,"connectionCount":5}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.755+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50115","uuid":"aafd388f-ed10-4092-b063-83f1282281f0","connectionId":6,"connectionCount":6}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.759+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn5","msg":"client metadata","attr":{"remote":"127.0.0.1:50114","client":"conn5","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:36.796+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn6","msg":"client metadata","attr":{"remote":"127.0.0.1:50115","client":"conn6","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.455+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50116","uuid":"4af31e91-71b4-47f5-8bb5-217cf23ae2f9","connectionId":7,"connectionCount":7}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.463+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn7","msg":"client metadata","attr":{"remote":"127.0.0.1:50116","client":"conn7","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.860+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50117","uuid":"2a9a4a35-90a8-490f-93b5-ea22e5cfdb28","connectionId":8,"connectionCount":8}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.861+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50118","uuid":"d78fbb9d-0331-465c-b72d-5c7dffd3d01b","connectionId":9,"connectionCount":9}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.862+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn8","msg":"client metadata","attr":{"remote":"127.0.0.1:50117","client":"conn8","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.863+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn9","msg":"client metadata","attr":{"remote":"127.0.0.1:50118","client":"conn9","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.873+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50119","uuid":"4a4e560f-bd8e-44d3-a3db-701e8a904007","connectionId":10,"connectionCount":10}}\n',

'{"t":{"$date":"2022-10-27T11:12:37.875+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn10","msg":"client metadata","attr":{"remote":"127.0.0.1:50119","client":"conn10","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:47.529+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50123","uuid":"31337491-dc4e-41d6-88f4-1ecbfbbf561b","connectionId":11,"connectionCount":11}}\n',

'{"t":{"$date":"2022-10-27T11:12:47.530+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn11","msg":"client metadata","attr":{"remote":"127.0.0.1:50123","client":"conn11","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:12:48.000+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50124","uuid":"661cc6e0-ae7d-4da2-a273-a2baa2073960","connectionId":12,"connectionCount":12}}\n',

'{"t":{"$date":"2022-10-27T11:12:48.002+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn12","msg":"client metadata","attr":{"remote":"127.0.0.1:50124","client":"conn12","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:20:04.210+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50213","uuid":"fd327561-03a5-4ba9-bd88-d43a4526880f","connectionId":13,"connectionCount":13}}\n',

'{"t":{"$date":"2022-10-27T11:20:04.210+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:50214","uuid":"3e663aa0-b690-4978-81b4-f7758749adab","connectionId":14,"connectionCount":14}}\n',

'{"t":{"$date":"2022-10-27T11:20:04.331+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn13","msg":"client metadata","attr":{"remote":"127.0.0.1:50213","client":"conn13","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:20:04.332+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn14","msg":"client metadata","attr":{"remote":"127.0.0.1:50214","client":"conn14","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T11:21:19.568+05:00"},"s":"I", "c":"STORAGE", "id":20320, "ctx":"conn2","msg":"createCollection","attr":{"namespace":"mydatabase.Student","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"93971e6c-f84d-4ca4-aa70-ebabebd7580f"}},"options":{}}}\n',

'{"t":{"$date":"2022-10-27T11:21:19.600+05:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"conn2","msg":"Index build: done building","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"93971e6c-f84d-4ca4-aa70-ebabebd7580f"}},"namespace":"mydatabase.Student","index":"\_id\_","ident":"index-1-5116716859605844652","collectionIdent":"collection-0-5116716859605844652","commitTimestamp":null}}\n',

'{"t":{"$date":"2022-10-27T11:37:50.292+05:00"},"s":"I", "c":"STORAGE", "id":20320, "ctx":"conn13","msg":"createCollection","attr":{"namespace":"mydatabase.colorarr","uuidDisposition":"generated","uuid":{"uuid":{"$uuid":"fbb1e63e-906e-4539-b559-bfd044e8a0bc"}},"options":{}}}\n',

'{"t":{"$date":"2022-10-27T11:37:50.317+05:00"},"s":"I", "c":"INDEX", "id":20345, "ctx":"conn13","msg":"Index build: done building","attr":{"buildUUID":null,"collectionUUID":{"uuid":{"$uuid":"fbb1e63e-906e-4539-b559-bfd044e8a0bc"}},"namespace":"mydatabase.colorarr","index":"\_id\_","ident":"index-3-5116716859605844652","collectionIdent":"collection-2-5116716859605844652","commitTimestamp":null}}\n',

'{"t":{"$date":"2022-10-27T13:49:01.987+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1977","uuid":"b153c066-aa0a-4e78-8af6-6f964aaf2a27","connectionId":15,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T13:49:01.988+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1978","uuid":"80856816-2881-4c23-8027-8252cd4ba539","connectionId":16,"connectionCount":16}}\n',

'{"t":{"$date":"2022-10-27T13:49:01.991+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn15","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:1977","connectionId":15}}\n',

'{"t":{"$date":"2022-10-27T13:49:02.002+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn15","msg":"Connection ended","attr":{"remote":"127.0.0.1:1977","uuid":"b153c066-aa0a-4e78-8af6-6f964aaf2a27","connectionId":15,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T13:49:02.255+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1979","uuid":"3318147c-1833-4b11-9734-76cce8a63aad","connectionId":17,"connectionCount":16}}\n',

'{"t":{"$date":"2022-10-27T13:49:02.263+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn16","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:1978","connectionId":16}}\n',

'{"t":{"$date":"2022-10-27T13:49:02.263+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn16","msg":"Connection ended","attr":{"remote":"127.0.0.1:1978","uuid":"80856816-2881-4c23-8027-8252cd4ba539","connectionId":16,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T13:50:07.850+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn17","msg":"Connection ended","attr":{"remote":"127.0.0.1:1979","uuid":"3318147c-1833-4b11-9734-76cce8a63aad","connectionId":17,"connectionCount":14}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.830+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:2576","uuid":"acf6f0a7-0e1d-4e2e-93d3-ea4e28821bb4","connectionId":18,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.830+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:2577","uuid":"c93b1e39-ef59-44f6-8a36-9dc5b1b66d9c","connectionId":19,"connectionCount":16}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.830+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn18","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:2576","connectionId":18}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.831+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn18","msg":"Connection ended","attr":{"remote":"127.0.0.1:2576","uuid":"acf6f0a7-0e1d-4e2e-93d3-ea4e28821bb4","connectionId":18,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.974+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn19","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:2577","connectionId":19}}\n',

'{"t":{"$date":"2022-10-27T14:05:16.974+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn19","msg":"Connection ended","attr":{"remote":"127.0.0.1:2577","uuid":"c93b1e39-ef59-44f6-8a36-9dc5b1b66d9c","connectionId":19,"connectionCount":14}}\n',

'{"t":{"$date":"2022-10-27T14:05:17.097+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:2578","uuid":"7895c382-d8c9-45d0-8262-c62fb7986448","connectionId":20,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T14:05:20.531+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn20","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:2578","connectionId":20}}\n',

'{"t":{"$date":"2022-10-27T14:05:20.531+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn20","msg":"Connection ended","attr":{"remote":"127.0.0.1:2578","uuid":"7895c382-d8c9-45d0-8262-c62fb7986448","connectionId":20,"connectionCount":14}}\n',

'{"t":{"$date":"2022-10-27T14:05:20.832+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:2584","uuid":"d6df428c-93bd-4c6d-b2b8-6a9eb231552e","connectionId":21,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T14:05:20.832+05:00"},"s":"I", "c":"NETWORK", "id":22988, "ctx":"conn21","msg":"Error receiving request from client. Ending connection from remote","attr":{"error":{"code":17,"codeName":"ProtocolError","errmsg":"Client sent an HTTP request over a native MongoDB connection"},"remote":"127.0.0.1:2584","connectionId":21}}\n',

'{"t":{"$date":"2022-10-27T14:05:20.833+05:00"},"s":"I", "c":"NETWORK", "id":22944, "ctx":"conn21","msg":"Connection ended","attr":{"remote":"127.0.0.1:2584","uuid":"d6df428c-93bd-4c6d-b2b8-6a9eb231552e","connectionId":21,"connectionCount":14}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.094+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1121","uuid":"d9cecada-8eb9-40c6-9e71-855667b33fa9","connectionId":22,"connectionCount":15}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.103+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn22","msg":"client metadata","attr":{"remote":"127.0.0.1:1121","client":"conn22","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.124+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1122","uuid":"9a535d41-9fbb-4d75-8a6a-813c3f5cff7b","connectionId":23,"connectionCount":16}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.125+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn23","msg":"client metadata","attr":{"remote":"127.0.0.1:1122","client":"conn23","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.295+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1125","uuid":"4edaea6d-e355-4c9f-bbb8-5a007e6f6fb8","connectionId":24,"connectionCount":17}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.296+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1126","uuid":"9835ba59-ded7-43af-a3cd-9a0c95cb7697","connectionId":25,"connectionCount":18}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.322+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn24","msg":"client metadata","attr":{"remote":"127.0.0.1:1125","client":"conn24","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.324+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn25","msg":"client metadata","attr":{"remote":"127.0.0.1:1126","client":"conn25","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.357+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1127","uuid":"ea31e600-b41d-46c3-997c-b5b0fb677359","connectionId":26,"connectionCount":19}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.407+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1128","uuid":"afa5098b-3826-40bb-8d50-c61faa209ffe","connectionId":27,"connectionCount":20}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.411+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn26","msg":"client metadata","attr":{"remote":"127.0.0.1:1127","client":"conn26","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.440+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn27","msg":"client metadata","attr":{"remote":"127.0.0.1:1128","client":"conn27","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.468+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1129","uuid":"6b44eff5-bd4f-4f87-8acd-3f92f3eb68d6","connectionId":28,"connectionCount":21}}\n',

'{"t":{"$date":"2022-10-27T14:07:13.482+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn28","msg":"client metadata","attr":{"remote":"127.0.0.1:1129","client":"conn28","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:14.146+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1131","uuid":"bc135100-7996-401e-b1f3-ce4de6069170","connectionId":29,"connectionCount":22}}\n',

'{"t":{"$date":"2022-10-27T14:07:14.153+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn29","msg":"client metadata","attr":{"remote":"127.0.0.1:1131","client":"conn29","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

'{"t":{"$date":"2022-10-27T14:07:14.598+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1133","uuid":"a52b2a03-f42d-4969-a441-96d4aac17b31","connectionId":30,"connectionCount":23}}\n',

'{"t":{"$date":"2022-10-27T14:07:14.598+05:00"},"s":"I", "c":"NETWORK", "id":22943, "ctx":"listener","msg":"Connection accepted","attr":{"remote":"127.0.0.1:1134","uuid":"264b04b0-3382-49a5-a463-b771f6cf07dd","connectionId":31,"connectionCount":24}}\n',

'{"t":{"$date":"2022-10-27T14:07:14.601+05:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn30","msg":"client metadata","attr":{"remote":"127.0.0.1:1133","client":"conn30","doc":{"driver":{"name":"nodejs","version":"4.8.1"},"os":{"type":"Windows\_NT","name":"win32","architecture":"x64","version":"10.0.19041"},"platform":"Node.js v16.5.0, LE (unified)|Node.js v16.5.0, LE (unified)","application":{"name":"MongoDB Compass"}}}}\n',

... 85 more items

]

test> db.help()

ReferenceError: dbo is not defined

Database Class:

getMongo Returns the current database connection

getName Returns the name of the DB

getCollectionNames Returns an array containing the names of all collections in the current database.

getCollectionInfos Returns an array of documents with collection information, i.e. collection name and options, for the current database.

runCommand Runs an arbitrary command on the database.

adminCommand Runs an arbitrary command against the admin database.

aggregate Runs a specified admin/diagnostic pipeline which does not require an underlying collection.

getSiblingDB Returns another database without modifying the db variable in the shell environment.

getCollection Returns a collection or a view object that is functionally equivalent to using the db.<collectionName>.

dropDatabase Removes the current database, deleting the associated data files.

createUser Creates a new user for the database on which the method is run. db.createUser() returns a duplicate user error if the user already exists on the database.

updateUser Updates the user’s profile on the database on which you run the method. An update to a field completely replaces the previous field’s values. This includes updates to the user’s roles array.

changeUserPassword Updates a user’s password. Run the method in the database where the user is defined, i.e. the database you created the user.

logout Ends the current authentication session. This function has no effect if the current session is not authenticated.

dropUser Removes the user from the current database.

dropAllUsers Removes all users from the current database.

auth Allows a user to authenticate to the database from within the shell.

grantRolesToUser Grants additional roles to a user.

revokeRolesFromUser Removes a one or more roles from a user on the current database.

getUser Returns user information for a specified user. Run this method on the user’s database. The user must exist on the database on which the method runs.

getUsers Returns information for all the users in the database.

createCollection Create new collection

createView Create new view

createRole Creates a new role.

updateRole Updates the role’s profile on the database on which you run the method. An update to a field completely replaces the previous field’s values.

dropRole Removes the role from the current database.

dropAllRoles Removes all roles from the current database.

grantRolesToRole Grants additional roles to a role.

revokeRolesFromRole Removes a one or more roles from a role on the current database.

grantPrivilegesToRole Grants additional privileges to a role.

revokePrivilegesFromRole Removes a one or more privileges from a role on the current database.

getRole Returns role information for a specified role. Run this method on the role’s database. The role must exist on the database on which the method runs.

getRoles Returns information for all the roles in the database.

currentOp Calls the currentOp command. Returns a document that contains information on in-progress operations for the database instance. The db.currentOp() method wraps the database command currentOp.

killOp Calls the killOp command. Terminates an operation as specified by the operation ID. To find operations and their corresponding IDs, see $currentOp or db.currentOp().

shutdownServer Calls the shutdown command. Shuts down the current mongod or mongos process cleanly and safely. You must issue the db.shutdownServer() operation against the admin database.

fsyncLock Calls the fsync command. Forces the mongod to flush all pending write operations to disk and locks the entire mongod instance to prevent additional writes until the user releases the lock with a corresponding db.fsyncUnlock() command.

fsyncUnlock Calls the fsyncUnlock command. Reduces the lock taken by db.fsyncLock() on a mongod instance by 1.

version returns the db version. uses the buildinfo command

serverBits returns the db serverBits. uses the buildInfo command

isMaster Calls the isMaster command

hello Calls the hello command

serverBuildInfo returns the db serverBuildInfo. uses the buildInfo command

serverStatus returns the server stats. uses the serverStatus command

stats returns the db stats. uses the dbStats command

hostInfo Calls the hostInfo command

serverCmdLineOpts returns the db serverCmdLineOpts. uses the getCmdLineOpts command

rotateCertificates Calls the rotateCertificates command

printCollectionStats Prints the collection.stats for each collection in the db.

getFreeMonitoringStatus Calls the getFreeMonitoringStatus command

disableFreeMonitoring returns the db disableFreeMonitoring. uses the setFreeMonitoring command

enableFreeMonitoring returns the db enableFreeMonitoring. uses the setFreeMonitoring command

getProfilingStatus returns the db getProfilingStatus. uses the profile command

setProfilingLevel returns the db setProfilingLevel. uses the profile command

setLogLevel returns the db setLogLevel. uses the setParameter command

getLogComponents returns the db getLogComponents. uses the getParameter command

cloneDatabase deprecated, non-functional

cloneCollection deprecated, non-functional

copyDatabase deprecated, non-functional

commandHelp returns the db commandHelp. uses the passed in command with help: true

listCommands Calls the listCommands command

getLastErrorObj Calls the getLastError command

getLastError Calls the getLastError command

printShardingStatus Calls sh.status(verbose)

printSecondaryReplicationInfo Prints secondary replicaset information

getReplicationInfo Returns replication information

printReplicationInfo Formats sh.getReplicationInfo

printSlaveReplicationInfo DEPRECATED. Use db.printSecondaryReplicationInfo

setSecondaryOk This method is deprecated. Use db.getMongo().setReadPref() instead

watch Opens a change stream cursor on the database

sql (Experimental) Runs a SQL query against Atlas Data Lake. Note: this is an experimental feature that may be subject to change in future releases.

test> db.test.help()

ReferenceError: getmongo is not defined

Collection Class:

aggregate Calculates aggregate values for the data in a collection or a view.

bulkWrite Performs multiple write operations with controls for order of execution.

count Returns the count of documents that would match a find() query for the collection or view.

countDocuments Returns the count of documents that match the query for a collection or view.

deleteMany Removes all documents that match the filter from a collection.

deleteOne Removes a single document from a collection.

distinct Finds the distinct values for a specified field across a single collection or view and returns the results in an array.

estimatedDocumentCount Returns the count of all documents in a collection or view.

find Selects documents in a collection or view.

findAndModify Modifies and returns a single document.

findOne Selects documents in a collection or view.

renameCollection Renames a collection.

findOneAndDelete Deletes a single document based on the filter and sort criteria, returning the deleted document.

findOneAndReplace Modifies and replaces a single document based on the filter and sort criteria.

findOneAndUpdate Updates a single document based on the filter and sort criteria.

insert Inserts a document or documents into a collection.

insertMany Inserts multiple documents into a collection.

insertOne Inserts a document into a collection.

isCapped Checks if a collection is capped

remove Removes documents from a collection.

replaceOne Replaces a single document within the collection based on the filter.

update Modifies an existing document or documents in a collection.

updateMany Updates all documents that match the specified filter for a collection.

updateOne Updates a single document within the collection based on the filter.

compactStructuredEncryptionData Compacts structured encryption data

convertToCapped calls {convertToCapped:'coll', size:maxBytes}} command

createIndexes Creates one or more indexes on a collection

createIndex Creates one index on a collection

ensureIndex Creates one index on a collection

getIndexes Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.

getIndexSpecs Alias for getIndexes. Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.

getIndices Alias for getIndexes. Returns an array that holds a list of documents that identify and describe the existing indexes on the collection.

getIndexKeys Return an array of key patterns for indexes defined on collection

dropIndexes Drops the specified index or indexes (except the index on the \_id field) from a collection.

dropIndex Drops or removes the specified index from a collection.

totalIndexSize Reports the total size used by the indexes on a collection.

reIndex Rebuilds all existing indexes on a collection.

getDB Get current database.

getMongo Returns the Mongo object.

dataSize This method provides a wrapper around the size output of the collStats (i.e. db.collection.stats()) command.

storageSize The total amount of storage allocated to this collection for document storage.

totalSize The total size in bytes of the data in the collection plus the size of every index on the collection.

drop Removes a collection or view from the database.

exists Returns collection infos if the collection exists or null otherwise.

getFullName Returns the name of the collection prefixed with the database name.

getName Returns the name of the collection.

runCommand Runs a db command with the given name where the first param is the collection name.

explain Returns information on the query plan.

stats Returns statistics about the collection.

latencyStats returns the $latencyStats aggregation for the collection. Takes an options document with an optional boolean 'histograms' field.

initializeOrderedBulkOp Initializes an ordered bulk command. Returns an instance of Bulk

initializeUnorderedBulkOp Initializes an unordered bulk command. Returns an instance of Bulk

getPlanCache Returns an interface to access the query plan cache for a collection. The interface provides methods to view and clear the query plan cache.

mapReduce Calls the mapReduce command

validate Calls the validate command. Default full value is false

getShardVersion Calls the getShardVersion command

getShardDistribution Prints the data distribution statistics for a sharded collection.

watch Opens a change stream cursor on the collection

hideIndex Hides an existing index from the query planner.

unhideIndex Unhides an existing index from the query planner.

test> use humanresouredb

switched to db humanresouredb

humanresouredb> show dbs

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

mydatabase 172.00 KiB

humanresouredb> db.dropDatabase(humanresouredb)

ReferenceError: humanresouredb is not defined

mydatabase> db.createcollection("Marks")

TypeError: db.createcollection is not a function

mydatabase> db.createCollection("Marks")

{ ok: 1 }

mydatabase> db.createCollection("Employee")

{ ok: 1 }

mydatabase> show Collections

MongoshInvalidInputError: [COMMON-10001] 'Collections' is not a valid argument for "show".

mydatabase> show collections

colorarr

Employee

Marks

Student

student\_view [view]

system.views

mydatabase> db.createCollection("address")

{ ok: 1 }

mydatabase> db.createCollection("Country")

{ ok: 1 }

mydatabase> show collections

address

colorarr

Country

Employee

Marks

Student

student\_view [view]

system.views

mydatabase> db.address.drop()

true

mydatabase> show collections

colorarr

Country

Employee

Marks

Student

student\_view [view]

system.views

mydatabase> show Employee

MongoshInvalidInputError: [COMMON-10001] 'Employee' is not a valid argument for "show".

mydatabase> db.createCollection("Address")

{ ok: 1 }

mydatabase> db.createCollection("Department")

{ ok: 1 }

mydatabase> show collections

Address

colorarr

Country

Department

Employee

Marks

Student

student\_view [view]

Studentdepartmet

system.views

mydatabase> db.Marks.insertOne({"MarksObtained":987,"Total marks":1100,"CGPA":3.71})

{

acknowledged: true,

insertedId: ObjectId("635b30ca0ac0ff7a169facfe")

}

mydatabase> db.Marks.find().pretty()

[

{

\_id: ObjectId("635abd911bc43a2866caf5da"),

MarksObtained: 945,

'Total marks': 1100,

CGPA: 3.8

},

{

\_id: ObjectId("635b30ca0ac0ff7a169facfe"),

MarksObtained: 987,

'Total marks': 1100,

CGPA: 3.71

}

]

mydatabase> db.Marks.insertOne({"MarksObtained":887,"Total marks":1100,"CGPA":2.71})

{

acknowledged: true,

insertedId: ObjectId("635b325b0ac0ff7a169facff")

}

mydatabase> db.Marks.insertOne({"MarksObtained":899,"Total marks":1100,"CGPA":2.91})

{

acknowledged: true,

insertedId: ObjectId("635b32790ac0ff7a169fad00")

}

mydatabase> db.Marks.insertOne({"MarksObtained":859,"Total marks":1100,"CGPA":2.82})

{

acknowledged: true,

insertedId: ObjectId("635b32a80ac0ff7a169fad01")

}

mydatabase> db.Marks.insertOne({"MarksObtained":959,"Total marks":1100,"CGPA":2.81})

{

acknowledged: true,

insertedId: ObjectId("635b32bf0ac0ff7a169fad02")

}

mydatabase> db.Marks.insertOne({"MarksObtained":859,"Total marks":1100,"CGPA":2.64})

{

acknowledged: true,

insertedId: ObjectId("635b32d70ac0ff7a169fad03")

}

mydatabase> db.Marks.insertOne({"MarksObtained":959,"Total marks":1100,"CGPA":3.64})

{

acknowledged: true,

insertedId: ObjectId("635b33740ac0ff7a169fad04")

}

mydatabase> db.Marks.insertOne({"MarksObtained":1050,"Total marks":1100,"CGPA":3.83})

{

acknowledged: true,

insertedId: ObjectId("635b339e0ac0ff7a169fad05")

}

mydatabase> db.Marks.insertOne({"MarksObtained":1078,"Total marks":1100,"CGPA":3.87})

{

acknowledged: true,

insertedId: ObjectId("635b33b70ac0ff7a169fad06")

}

mydatabase> db.Marks.insertOne({"MarksObtained":1096,"Total marks":1100,"CGPA":4})

{

acknowledged: true,

insertedId: ObjectId("635b33cd0ac0ff7a169fad07")

}

mydatabase> db.Marks.insertOne({"MarksObtained":1097,"Total marks":1100,"CGPA":4})

{

acknowledged: true,

insertedId: ObjectId("635b33db0ac0ff7a169fad08")

}

mydatabase> db.Marks.insertOne({"MarksObtained":1093,"Total marks":1100,"CGPA":4})

{

acknowledged: true,

insertedId: ObjectId("635b33e90ac0ff7a169fad09")

}

mydatabase> db.Marks.find().pretty()

[

{

\_id: ObjectId("635abd911bc43a2866caf5da"),

MarksObtained: 945,

'Total marks': 1100,

CGPA: 3.8

},

{

\_id: ObjectId("635b30ca0ac0ff7a169facfe"),

MarksObtained: 987,

'Total marks': 1100,

CGPA: 3.71

},

{

\_id: ObjectId("635b325b0ac0ff7a169facff"),

MarksObtained: 887,

'Total marks': 1100,

CGPA: 2.71

},

{

\_id: ObjectId("635b32790ac0ff7a169fad00"),

MarksObtained: 899,

'Total marks': 1100,

CGPA: 2.91

},

{

\_id: ObjectId("635b32a80ac0ff7a169fad01"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.82

},

{

\_id: ObjectId("635b32bf0ac0ff7a169fad02"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 2.81

},

{

\_id: ObjectId("635b32d70ac0ff7a169fad03"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.64

},

{

\_id: ObjectId("635b33740ac0ff7a169fad04"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 3.64

},

{

\_id: ObjectId("635b339e0ac0ff7a169fad05"),

MarksObtained: 1050,

'Total marks': 1100,

CGPA: 3.83

},

{

\_id: ObjectId("635b33b70ac0ff7a169fad06"),

MarksObtained: 1078,

'Total marks': 1100,

CGPA: 3.87

},

{

\_id: ObjectId("635b33cd0ac0ff7a169fad07"),

MarksObtained: 1096,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33db0ac0ff7a169fad08"),

MarksObtained: 1097,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33e90ac0ff7a169fad09"),

MarksObtained: 1093,

'Total marks': 1100,

CGPA: 4

}

]

mydatabase> db.Marks.insertOne({"Fname":"Rahan ","Lname":"Sohail","Contact":03357856445,"Email":"Rahan564@gmail.com"})

{

acknowledged: true,

insertedId: ObjectId("635b35de0ac0ff7a169fad0a")

}

mydatabase> db.Marks.insertOne({"Fname":"Zeeshan ","Lname":"khan","Contact":03659856445,"Email":"Ranara964@gmail.com"})

{

acknowledged: true,

insertedId: ObjectId("635b36150ac0ff7a169fad0b")

}

mydatabase> db.Marks.insertOne({"Fname":"Muzamil ","Lname":"Hamid","Contact":03659886449,"Email":"Kamran@gmail.com"})

{

acknowledged: true,

insertedId: ObjectId("635b364d0ac0ff7a169fad0c")

}

mydatabase> db.Marks.insertOne({"Fname":"Aitzaz ","Lname":"khan","Contact":03759886748,"Email":"Aitzaza67@gmail.com"})

{

acknowledged: true,

insertedId: ObjectId("635b36850ac0ff7a169fad0d")

}

mydatabase> db.Marks.find().pretty()

[

{

\_id: ObjectId("635abd911bc43a2866caf5da"),

MarksObtained: 945,

'Total marks': 1100,

CGPA: 3.8

},

{

\_id: ObjectId("635b30ca0ac0ff7a169facfe"),

MarksObtained: 987,

'Total marks': 1100,

CGPA: 3.71

},

{

\_id: ObjectId("635b325b0ac0ff7a169facff"),

MarksObtained: 887,

'Total marks': 1100,

CGPA: 2.71

},

{

\_id: ObjectId("635b32790ac0ff7a169fad00"),

MarksObtained: 899,

'Total marks': 1100,

CGPA: 2.91

},

{

\_id: ObjectId("635b32a80ac0ff7a169fad01"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.82

},

{

\_id: ObjectId("635b32bf0ac0ff7a169fad02"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 2.81

},

{

\_id: ObjectId("635b32d70ac0ff7a169fad03"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.64

},

{

\_id: ObjectId("635b33740ac0ff7a169fad04"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 3.64

},

{

\_id: ObjectId("635b339e0ac0ff7a169fad05"),

MarksObtained: 1050,

'Total marks': 1100,

CGPA: 3.83

},

{

\_id: ObjectId("635b33b70ac0ff7a169fad06"),

MarksObtained: 1078,

'Total marks': 1100,

CGPA: 3.87

},

{

\_id: ObjectId("635b33cd0ac0ff7a169fad07"),

MarksObtained: 1096,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33db0ac0ff7a169fad08"),

MarksObtained: 1097,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33e90ac0ff7a169fad09"),

MarksObtained: 1093,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b35de0ac0ff7a169fad0a"),

Fname: 'Rahan ',

Lname: 'Sohail',

Contact: 3357856445,

Email: 'Rahan564@gmail.com'

},

{

\_id: ObjectId("635b36150ac0ff7a169fad0b"),

Fname: 'Zeeshan ',

Lname: 'khan',

Contact: 3659856445,

Email: 'Ranara964@gmail.com'

},

{

\_id: ObjectId("635b364d0ac0ff7a169fad0c"),

Fname: 'Muzamil ',

Lname: 'Hamid',

Contact: 3659886449,

Email: 'Kamran@gmail.com'

},

{

\_id: ObjectId("635b36850ac0ff7a169fad0d"),

Fname: 'Aitzaz ',

Lname: 'khan',

Contact: 3759886748,

Email: 'Aitzaza67@gmail.com'

}

]

mydatabase> db.Marks.insertOne({\_id:"1","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"ajman41288@gmail.com"})

{ acknowledged: true, insertedId: '1' }

mydatabase> db.Marks.insertOne({\_id:"2","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"ajman41288@outlook.com"})

{ acknowledged: true, insertedId: '2' }

mydatabase> db.Marks.insertOne({\_id:"3","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"ajman.khan40@ce.ceme.edu.pk.com"})

{ acknowledged: true, insertedId: '3' }

mydatabase> db.Marks.insertOne({\_id:"4","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"aullahkhan.ce40ceme@student.nust.edu.pk.com"})

{ acknowledged: true, insertedId: '4' }

mydatabase> db.Marks.insertOne({\_id:"5","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"ajmanullah745@yahoo.com"})

{ acknowledged: true, insertedId: '5' }

mydatabase> db.Marks.insertOne({\_id:"6","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"})

{ acknowledged: true, insertedId: '6' }

mydatabase> db.Marks.find().pretty()

[

{

\_id: ObjectId("635abd911bc43a2866caf5da"),

MarksObtained: 945,

'Total marks': 1100,

CGPA: 3.8

},

{

\_id: ObjectId("635b30ca0ac0ff7a169facfe"),

MarksObtained: 987,

'Total marks': 1100,

CGPA: 3.71

},

{

\_id: ObjectId("635b325b0ac0ff7a169facff"),

MarksObtained: 887,

'Total marks': 1100,

CGPA: 2.71

},

{

\_id: ObjectId("635b32790ac0ff7a169fad00"),

MarksObtained: 899,

'Total marks': 1100,

CGPA: 2.91

},

{

\_id: ObjectId("635b32a80ac0ff7a169fad01"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.82

},

{

\_id: ObjectId("635b32bf0ac0ff7a169fad02"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 2.81

},

{

\_id: ObjectId("635b32d70ac0ff7a169fad03"),

MarksObtained: 859,

'Total marks': 1100,

CGPA: 2.64

},

{

\_id: ObjectId("635b33740ac0ff7a169fad04"),

MarksObtained: 959,

'Total marks': 1100,

CGPA: 3.64

},

{

\_id: ObjectId("635b339e0ac0ff7a169fad05"),

MarksObtained: 1050,

'Total marks': 1100,

CGPA: 3.83

},

{

\_id: ObjectId("635b33b70ac0ff7a169fad06"),

MarksObtained: 1078,

'Total marks': 1100,

CGPA: 3.87

},

{

\_id: ObjectId("635b33cd0ac0ff7a169fad07"),

MarksObtained: 1096,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33db0ac0ff7a169fad08"),

MarksObtained: 1097,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b33e90ac0ff7a169fad09"),

MarksObtained: 1093,

'Total marks': 1100,

CGPA: 4

},

{

\_id: ObjectId("635b35de0ac0ff7a169fad0a"),

Fname: 'Rahan ',

Lname: 'Sohail',

Contact: 3357856445,

Email: 'Rahan564@gmail.com'

},

{

\_id: ObjectId("635b36150ac0ff7a169fad0b"),

Fname: 'Zeeshan ',

Lname: 'khan',

Contact: 3659856445,

Email: 'Ranara964@gmail.com'

},

{

\_id: ObjectId("635b364d0ac0ff7a169fad0c"),

Fname: 'Muzamil ',

Lname: 'Hamid',

Contact: 3659886449,

Email: 'Kamran@gmail.com'

},

{

\_id: ObjectId("635b36850ac0ff7a169fad0d"),

Fname: 'Aitzaz ',

Lname: 'khan',

Contact: 3759886748,

Email: 'Aitzaza67@gmail.com'

},

{

\_id: '1',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'ajman41288@gmail.com'

},

{

\_id: '2',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'ajman41288@outlook.com'

},

{

\_id: '3',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'ajman.khan40@ce.ceme.edu.pk.com'

}

]

Type "it" for more

mydatabase> it

[

{

\_id: '4',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'aullahkhan.ce40ceme@student.nust.edu.pk.com'

},

{

\_id: '5',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'ajmanullah745@yahoo.com'

},

{

\_id: '6',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'osamabrandstore8@gmail.com'

}

]

mydatabase> db.Marks.insert([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}])

Uncaught:

SyntaxError: Unexpected token (1:260)

> 1 | db.Marks.insert([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}])

| ^

2 |

mydatabase> db.Marks.insert([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"},])

Uncaught:

SyntaxError: Unexpected token (1:260)

> 1 | db.Marks.insert([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"},])

| ^

2 |

mydatabase> db.Marks.insert({{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}})

Uncaught:

SyntaxError: Unexpected token (1:17)

> 1 | db.Marks.insert({{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}})

| ^

2 |

mydatabase> db.Marks.insert([{"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"}])

Uncaught:

SyntaxError: Unexpected token (1:282)

> 1 | db.Marks.insert([{"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"}])

| ^

2 |

mydatabase> db.Marks.insert({"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"})

Uncaught:

SyntaxError: Unexpected token (1:281)

> 1 | db.Marks.insert({"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"})

| ^

2 |

mydatabase> db.Marks.insert({\_id:"6","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"},{\_id:"7","Fname":"ajman","Lname":"Khan","Contact":03324118288,"Email":"ajman543@gamil.com"})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

Uncaught:

MongoBulkWriteError: E11000 duplicate key error collection: mydatabase.Marks index: \_id\_ dup key: { \_id: "6" }

Result: BulkWriteResult {

result: {

ok: 1,

writeErrors: [

WriteError {

err: {

index: 0,

code: 11000,

errmsg: 'E11000 duplicate key error collection: mydatabase.Marks index: \_id\_ dup key: { \_id: "6" }',

errInfo: undefined,

op: {

\_id: '6',

Fname: 'Ajmanullah ',

Lname: 'khan',

Contact: 3324118288,

Email: 'osamabrandstore8@gmail.com'

}

}

}

],

writeConcernErrors: [],

insertedIds: [ { index: 0, \_id: '6' } ],

nInserted: 0,

nUpserted: 0,

nMatched: 0,

nModified: 0,

nRemoved: 0,

upserted: []

}

}

mydatabase> db.Marks.insertMany([{"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"}])

Uncaught:

SyntaxError: Unexpected token (1:286)

> 1 | db.Marks.insertMany([{"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"}])

| ^

2 |

mydatabase> db.Marks.insertMany([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}])

Uncaught:

SyntaxError: Unexpected token (1:264)

> 1 | db.Marks.insertMany([{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}])

| ^

2 |

mydatabase> db.Marks.insertMany({Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"})

Uncaught:

SyntaxError: Unexpected token (1:263)

> 1 | db.Marks.insertMany({Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"})

| ^

2 |

mydatabase> db.Marks.insertMany({{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}})

Uncaught:

SyntaxError: Unexpected token (1:21)

> 1 | db.Marks.insertMany({{Fname:"Farhan",Lname:"Khan",Email:"Farhan564@gmail.com"},{Fname:"Jabran",Lname:"Khan",Email:"Jabrankhan453@gmail.com"},{Fname:"Waqas",Lname:"khan",Email:"waqas564@gmail.com"},{Fname:"Osamabinkhalid",Lname:"Hassan","Osamabinkhalid45@gamil.com"},{Fname:"Jalili",Lname:"Khan",Email:"Jalilikhan453@gmail.com"},{Fname:"Waqar",Lname:"khan",Email:"Waqar675@gmail.com"}})

| ^

2 |

mydatabase> db.Marks.insert({\_id:"7","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"})

{ acknowledged: true, insertedIds: { '0': '7' } }

mydatabase> db.Marks.insert({\_id:"8","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"})

{ acknowledged: true, insertedIds: { '0': '8' } }

mydatabase> db.Marks.insert({\_id:"9","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"})

{ acknowledged: true, insertedIds: { '0': '9' } }

mydatabase> db.Marks.insertMany({\_id:"9","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"},{\_id:"10","Fname":"ajman","Lname":"khan","contact":0345637735,"Email":"ajman5234@gmail.com"})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

mydatabase> db.Marks.insertMany([{\_id:"11","Fname":"Ajmanullah ","Lname":"khan","Contact":03324118288,"Email":"osamabrandstore8@gmail.com"},{\_id:"10","Fname":"ajman","Lname":"khan","contact":0345637735,"Email":"ajman5234@gmail.com"}])

{ acknowledged: true, insertedIds: { '0': '11', '1': '10' } }

mydatabase> db.Marks.insertMany([{"Fname":"Farhan","Lname":"Khan","Email":"Farhan564@gmail.com"},{"Fname":"Jabran","Lname":"Khan","Email":"Jabrankhan453@gmail.com"},{"Fname":"Waqas","Lname":"khan","Email":"waqas564@gmail.com"},{"Fname":"Osamabinkhalid","Lname":"Hassan","Email":"Osamabinkhalid45@gamil.com"},{"Fname":"Jalili","Lname":"Khan","Email":"Jalilikhan453@gmail.com"},{"Fname":"Waqar","Lname":"khan","Email":"Waqar675@gmail.com"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("635b46940ac0ff7a169fad0e"),

'1': ObjectId("635b46940ac0ff7a169fad0f"),

'2': ObjectId("635b46940ac0ff7a169fad10"),

'3': ObjectId("635b46940ac0ff7a169fad11"),

'4': ObjectId("635b46940ac0ff7a169fad12"),

'5': ObjectId("635b46940ac0ff7a169fad13")

}

}

mydatabase> db.Marks.insertMany([{"Fname":"Osaman","Lname":"akram","Email":"usamakaram8764@gmail.com"},{"Fname":"zohri","Lname":"hassan","Email":"Zohrihassan783@gmail.com"},{"Fname":"Noman","Lname":"khan","Email":"Noman64@gmail.com"},{"Fname":"khalid","Lname":"Hassan","Email":"khalid4675665@gamil.com"},{"Fname":"Zubair","Lname":"Khan","Email":"zubairkhan453@gmail.com"},{"Fname":"sana","Lname":"khan","Email":"sanakhan565@gmail.com"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId("635b47720ac0ff7a169fad14"),

'1': ObjectId("635b47720ac0ff7a169fad15"),

'2': ObjectId("635b47720ac0ff7a169fad16"),

'3': ObjectId("635b47720ac0ff7a169fad17"),

'4': ObjectId("635b47720ac0ff7a169fad18"),

'5': ObjectId("635b47720ac0ff7a169fad19")

}

}

mydatabase>

