Name: Sanket Mahajan

TE-A   Roll.no. -71

MongoDB Queries:

DesignandDevelopMongoDBQueriesusingCRUDoperations.(UseCRUDoperations, SAVE method, logical operators etc.).

vishal@patil-HP-ProOne-600-G1-AiO:~$ mongosh

Current Mongosh Log ID: 6366542b434912fd1e71c21b

Connecting to: mongodb://127.0.0.1:27017/?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/

------

   The server generated these startup warnings when booting

   2022-11-05T15:57:49.120+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem

   2022-11-05T15:57:54.238+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

   2022-11-05T15:57:54.278+05:30: vm.max\_map\_count is too low

------

------

   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> use MongoDB01

switched to db MongoDB01

MongoDB01> db.createCollection("library");

{ ok: 1 }

MongoDB01> db.library.insert({bid:1,name:"C++"});

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

{

  acknowledged: true,

  insertedIds: { '0': ObjectId("636655c75e44f326a3addc5b") }

}

MongoDB01> db.library.insertOne({bid:2,name:"java"});

{

  acknowledged: true,

  insertedId: ObjectId("636655eb5e44f326a3addc5c")

}

MongoDB01> db.library.insertOne({bid:3,name:"python"});

{

  acknowledged: true,

  insertedId: ObjectId("636655fb5e44f326a3addc5d")

}

MongoDB01> db.library.find();

[

  { \_id: ObjectId("636655c75e44f326a3addc5b"), bid: 1, name: 'C++' },

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  { \_id: ObjectId("636655fb5e44f326a3addc5d"), bid: 3, name: 'python' }]

MongoDB01>

MongoDB01> db.library.remove({bid:1});

DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.

{ acknowledged: true, deletedCount: 1 }

MongoDB01> db.library.find();

[

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  {

    \_id: ObjectId("636655fb5e44f326a3addc5d"),

    bid: 3,

    name: 'python3.7'

  }

]

MongoDB01>

MongoDB01>

MongoDB01> db.library.findOne({name:"java"});

{ \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' }

MongoDB01>

MongoDB01>

MongoDB01> db.library.insertOne({bid:4,name:"java",desc:"fake book"})

{

  acknowledged: true,

  insertedId: ObjectId("6366587c5e44f326a3addc5e")

}

MongoDB01> db.library.find();

[

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  {

    \_id: ObjectId("636655fb5e44f326a3addc5d"),

    bid: 3,

    name: 'python3.7'

  },

  {

    \_id: ObjectId("6366587c5e44f326a3addc5e"),

    bid: 4,

    name: 'java',

    desc: 'fake book'

  }

]

MongoDB01> db.library.findOne({name:"java"});

{ \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' }

MongoDB01>

MongoDB01>

MongoDB01> db.library.find({$and:[{name:"java"},{desc:"fake book"}]})

[

  {

    \_id: ObjectId("6366587c5e44f326a3addc5e"),

    bid: 4,

    name: 'java',

    desc: 'fake book'

  }

]

MongoDB01>

MongoDB01> db.library.insertOne({bid:5,name:"The Book 1", cost:500})

{

  acknowledged: true,

  insertedId: ObjectId("63665a0f5e44f326a3addc5f")

}

MongoDB01> db.library.insertOne({bid:6,name:"The Book 2", cost:800})

{

  acknowledged: true,

  insertedId: ObjectId("63665a205e44f326a3addc60")

}

MongoDB01> db.library.insertOne({bid:7,name:"The Book 3", cost:60})

{

  acknowledged: true,

  insertedId: ObjectId("63665a345e44f326a3addc61")

}

MongoDB01> db.library.find();

[

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  {

    \_id: ObjectId("636655fb5e44f326a3addc5d"),

    bid: 3,

    name: 'python3.7'

  },

  {

    \_id: ObjectId("6366587c5e44f326a3addc5e"),

    bid: 4,

    name: 'java',

    desc: 'fake book'

  },

  {

    \_id: ObjectId("63665a0f5e44f326a3addc5f"),

    bid: 5,

    name: 'The Book 1',

    cost: 500

  },

  {

    \_id: ObjectId("63665a205e44f326a3addc60"),

    bid: 6,

    name: 'The Book 2',

    cost: 800

  },

MongoDB01> db.library.find({cost:{$gte:100}});

[

  {

    \_id: ObjectId("63665a0f5e44f326a3addc5f"),

    bid: 5,

    name: 'The Book 1',

    cost: 500

  },

  {

    \_id: ObjectId("63665a205e44f326a3addc60"),

    bid: 6,

    name: 'The Book 2',

    cost: 800

  }

]

MongoDB01> db.library.find({cost:{$lte:100}});

[

  {

    \_id: ObjectId("63665a345e44f326a3addc61"),

    bid: 7,

    name: 'The Book 3',

    cost: 60

  }

]

MongoDB01> db.library.find({cost:{$equ:500}});

MongoServerError: unknown operator: $equ

MongoDB01> db.library.find({cost:{$eq:500}});

[

  {

    \_id: ObjectId("63665a0f5e44f326a3addc5f"),

    bid: 5,

    name: 'The Book 1',

    cost: 500

  }

]

MongoDB01> db.library.find({cost:{$ne:500}});

[

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  {

    \_id: ObjectId("636655fb5e44f326a3addc5d"),

    bid: 3,

    name: 'python3.7'

  },

  {

    \_id: ObjectId("6366587c5e44f326a3addc5e"),

    bid: 4,

    name: 'java',

    desc: 'fake book'

  },

  {

    \_id: ObjectId("63665a205e44f326a3addc60"),

    bid: 6,

    name: 'The Book 2',

    cost: 800

  },

  {

    \_id: ObjectId("63665a345e44f326a3addc61"),

    bid: 7,

    name: 'The Book 3',

    cost: 60

  }

]

MongoDB01> db.library.find({cost:{$in:[100,200,300,400,500]}});

[

  {

    \_id: ObjectId("63665a0f5e44f326a3addc5f"),

    bid: 5,

    name: 'The Book 1',

    cost: 500

  }

]

MongoDB01>

MongoDB01> db.library.save({\_id:ObjectId("123456789abcdef123456789"),bid:8,name:"My save Query"})

TypeError: db.library.save is not a function

MongoDB01> db.library.insertOne({\_id:ObjectId("123456789abcdef123456789"),bid:7,name:"My save Query"})

{

  acknowledged: true,

  insertedId: ObjectId("123456789abcdef123456789")

}

MongoDB01> db.library.find();

[

  { \_id: ObjectId("636655eb5e44f326a3addc5c"), bid: 2, name: 'java' },

  {

    \_id: ObjectId("636655fb5e44f326a3addc5d"),

    bid: 3,

    name: 'python3.7'

  },

  {

    \_id: ObjectId("6366587c5e44f326a3addc5e"),

    bid: 4,

    name: 'java',

    desc: 'fake book'

  },

  {

    \_id: ObjectId("63665a0f5e44f326a3addc5f"),

    bid: 5,

    name: 'The Book 1',

    cost: 500

  },

  {

    \_id: ObjectId("63665a205e44f326a3addc60"),

    bid: 6,

    name: 'The Book 2',

    cost: 800

  },

  {

    \_id: ObjectId("63665a345e44f326a3addc61"),

    bid: 7,

    name: 'The Book 3',

    cost: 60

  },

  {

    \_id: ObjectId("123456789abcdef123456789"),

    bid: 7,

    name: 'My save Query'

  }

]

MongoDB01> exit