|  |  |
| --- | --- |
| **15** | **Mongo Aggregation**  Create a student collection consisting of fields like Roll No, name, class, marks, sports etc  use students  switched to db students  students> db.students.insert({"rollno":1,"name":"atharvaa","class":"TE","mar  ks":90,"sports":"badminton"});  DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451c0e4aa41441fbf9f766") }  }  students> **db.students.insert({"rollno":2,"name":"abhishek","class":"TE","marks":92,"sports":"swimming"});**  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451c3e4aa41441fbf9f767") }  }  students> db.students.update({"rollno":1},{$set:{"class":"TE A"}});  DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.  {  acknowledged: true,  insertedId: null,  matchedCount: 1,  modifiedCount: 1,  upsertedCount: 0  }  students> db.students.update({"rollno":2},{$set:{"class":"TE A"}});  {  acknowledged: true,  insertedId: null,  matchedCount: 1,  modifiedCount: 1,  upsertedCount: 0  }  students> db.students.insert({"rollno":3,"name":"soham","class":"TE A","mark  s":93,"sports":"gaming"});  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451d8d4aa41441fbf9f768") }  }  students> db.students.insert({"rollno":4,"name":"kartik","class":"TE b","marks":80,"sports":"football"});  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451dcb4aa41441fbf9f769") }  }  students> db.students.insert({"rollno":5,"name":"tiger","class":"TE b","marks":88,"sports":"football"});  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451de74aa41441fbf9f76a") }  }  students> db.students.insert({"rollno":6,"name":"varun","class":"TE b","marks":89,"sports":"cricket"});  {  acknowledged: true,  insertedIds: { '0': ObjectId("65451e0a4aa41441fbf9f76b") }  }  students> db.find().pretty();  TypeError: db.find is not a function  **students> db.students.find().pretty();**  [  {  \_id: ObjectId("65451c0e4aa41441fbf9f766"),  rollno: 1,  name: 'atharvaa',  class: 'TE A',  marks: 90,  sports: 'badminton'  },  {  \_id: ObjectId("65451c3e4aa41441fbf9f767"),  rollno: 2,  name: 'abhishek',  class: 'TE A',  marks: 92,  sports: 'swimming'  },  {  \_id: ObjectId("65451d8d4aa41441fbf9f768"),  rollno: 3,  name: 'soham',  class: 'TE A',  marks: 93,  sports: 'gaming'  },  {  \_id: ObjectId("65451dcb4aa41441fbf9f769"),  rollno: 4,  name: 'kartik',  class: 'TE b',  marks: 80,  sports: 'football'  },  {  \_id: ObjectId("65451de74aa41441fbf9f76a"),  rollno: 5,  name: 'tiger',  class: 'TE b',  marks: 88,  sports: 'football'  },  {  \_id: ObjectId("65451e0a4aa41441fbf9f76b"),  rollno: 6,  name: 'varun',  class: 'TE b',  marks: 89,  sports: 'cricket'  }  ]  1.Create an index on name field of employee collection. Also use the explain() function  **db.student.createIndex({name:1});**  name\_1  students> **db.student.find({name:"atharvaa"}).explain("executionStats")**  {  explainVersion: '2',  queryPlanner: {  namespace: 'students.student',  indexFilterSet: false,  parsedQuery: { name: { '$eq': 'atharvaa' } },  queryHash: '1AD872C6',  planCacheKey: '9B90F841',  maxIndexedOrSolutionsReached: false,  maxIndexedAndSolutionsReached: false,  maxScansToExplodeReached: false,  winningPlan: {  queryPlan: {  stage: 'FETCH',  planNodeId: 2,  inputStage: {  stage: 'IXSCAN',  planNodeId: 1,  keyPattern: { name: 1 },  indexName: 'name\_1',  isMultiKey: false,  multiKeyPaths: { name: [] },  isUnique: false,  isSparse: false,  isPartial: false,  indexVersion: 2,  direction: 'forward',  indexBounds: { name: [ '["atharvaa", "atharvaa"]' ] }  }  },  slotBasedPlan: {  slots: '$$RESULT=s11 env: { s3 = 1699028722633 (NOW), s6 = KS(3C617468617276616100FE04), s1 = TimeZoneDatabase(Pacific/Majuro...America/Argentina/San\_Luis) (timeZoneDB), s10 = {"name" : 1}, s2 = Nothing (SEARCH\_META), s5 = KS(3C6174686172766161000104) }',  stages: '[2] nlj inner [] [s4, s7, s8, s9, s10] \n' +  ' left \n' +  ' [1] cfilter {(exists(s5) && exists(s6))} \n' +  ' [1] ixseek s5 s6 s9 s4 s7 s8 [] @"ad19e831-d6c2-4574-a427-7ad82dd99f48" @"name\_1" true \n' +  ' right \n' +  ' [2] limit 1 \n' +  ' [2] seek s4 s11 s12 s7 s8 s9 s10 [] @"ad19e831-d6c2-4574-a427-7ad82dd99f48" true false \n'  }  },  rejectedPlans: []  },  executionStats: {  executionSuccess: true,  nReturned: 0,  executionTimeMillis: 22,  totalKeysExamined: 0,  totalDocsExamined: 0,  executionStages: {  stage: 'nlj',  planNodeId: 2,  nReturned: 0,  executionTimeMillisEstimate: 0,  opens: 1,  closes: 1,  saveState: 0,  restoreState: 0,  isEOF: 1,  totalDocsExamined: 0,  totalKeysExamined: 0,  collectionScans: 0,  collectionSeeks: 0,  indexScans: 0,  indexSeeks: 1,  indexesUsed: [ 'name\_1' ],  innerOpens: 0,  innerCloses: 0,  outerProjects: [],  outerCorrelated: [ Long("4"), Long("7"), Long("8"), Long("9"), Long("10") ],  outerStage: {  stage: 'cfilter',  planNodeId: 1,  nReturned: 0,  executionTimeMillisEstimate: 0,  opens: 1,  closes: 1,  saveState: 0,  restoreState: 0,  isEOF: 1,  numTested: 1,  filter: '(exists(s5) && exists(s6)) ',  inputStage: {  stage: 'ixseek',  planNodeId: 1,  nReturned: 0,  executionTimeMillisEstimate: 0,  opens: 1,  closes: 1,  saveState: 0,  restoreState: 0,  isEOF: 1,  indexName: 'name\_1',  keysExamined: 0,  seeks: 1,  numReads: 1,  indexKeySlot: 9,  recordIdSlot: 4,  snapshotIdSlot: 7,  indexIdentSlot: 8,  outputSlots: [],  indexKeysToInclude: '00000000000000000000000000000000',  seekKeyLow: 's5 ',  seekKeyHigh: 's6 '  }  },  innerStage: {  stage: 'limit',  planNodeId: 2,  nReturned: 0,  executionTimeMillisEstimate: 0,  opens: 0,  closes: 0,  saveState: 0,  restoreState: 0,  isEOF: 0,  limit: 1,  inputStage: {  stage: 'seek',  planNodeId: 2,  nReturned: 0,  executionTimeMillisEstimate: 0,  opens: 0,  closes: 0,  saveState: 0,  restoreState: 0,  isEOF: 0,  numReads: 0,  recordSlot: 11,  recordIdSlot: 12,  seekKeySlot: 4,  snapshotIdSlot: 7,  indexIdentSlot: 8,  indexKeySlot: 9,  indexKeyPatternSlot: 10,  fields: [],  outputSlots: []  }  }  }  },  command: { find: 'student', filter: { name: 'atharvaa' }, '$db': 'students' },  serverInfo: {  host: 'LAPTOP-7UVAN5JC',  port: 27017,  version: '7.0.2',  gitVersion: '02b3c655e1302209ef046da6ba3ef6749dd0b62a'  },  serverParameters: {  internalQueryFacetBufferSizeBytes: 104857600,  internalQueryFacetMaxOutputDocSizeBytes: 104857600,  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,  internalDocumentSourceGroupMaxMemoryBytes: 104857600,  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,  internalQueryProhibitBlockingMergeOnMongoS: 0,  internalQueryMaxAddToSetBytes: 104857600,  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,  internalQueryFrameworkControl: 'trySbeEngine'  },  ok: 1  }   1. Display the first 5 toppers of TE   students> **db.students.find({"class":"TE A"}).sort({"marks":-1}).limit(2);**  [  {  \_id: ObjectId("65451d8d4aa41441fbf9f768"),  rollno: 3,  name: 'soham',  class: 'TE A',  marks: 93,  sports: 'gaming'  },  {  \_id: ObjectId("65451c3e4aa41441fbf9f767"),  rollno: 2,  name: 'abhishek',  class: 'TE A',  marks: 92,  sports: 'swimming'  }  ]   1. Display marks of topper of each division   students> **db.students.aggregate([{$group:{\_id:"$class",topMarks:{$max:"$marks"}}}]);**  [ { \_id: 'TE b', topMarks: 89 }, { \_id: 'TE A', topMarks: 93 }   1. Display the average marks of each division   **students> db.students.aggregate([{$group:{\_id:"$class",avgMarks:{$avg:"$mark**  **s"}}}]);**  [  { \_id: 'TE A', avgMarks: 91.66666666666667 },  { \_id: 'TE b', avgMarks: 85.66666666666667 }  ]   1. Display the rollcall of TEComp A   **students> db.students.find({class:"TE A"},{rollno:1,name:1});**  [  {  \_id: ObjectId("65451c0e4aa41441fbf9f766"),  rollno: 1,  name: 'atharvaa'  },  {  \_id: ObjectId("65451c3e4aa41441fbf9f767"),  rollno: 2,  name: 'abhishek'  },  {  \_id: ObjectId("65451d8d4aa41441fbf9f768"),  rollno: 3,  name: 'soham'  }  ]   1. Display list of fail students from TE Comp A   **students> db.students.find({class:"TE A",marks:{$lt:91}},{name:1,marks:1});**  [  {  \_id: ObjectId("65451c0e4aa41441fbf9f766"),  name: 'atharvaa',  marks: 90  }  ]   1. Display Name, Class and Marks of all students   students> **db.students.find({},{"name":1,"class":1,"marks":1});**  [  {  \_id: ObjectId("65451c0e4aa41441fbf9f766"),  name: 'atharvaa',  class: 'TE A',  marks: 90  },  {  \_id: ObjectId("65451c3e4aa41441fbf9f767"),  name: 'abhishek',  class: 'TE A',  marks: 92  },  {  \_id: ObjectId("65451d8d4aa41441fbf9f768"),  name: 'soham',  class: 'TE A',  marks: 93  },  {  \_id: ObjectId("65451dcb4aa41441fbf9f769"),  name: 'kartik',  class: 'TE b',  marks: 80  },  {  \_id: ObjectId("65451de74aa41441fbf9f76a"),  name: 'tiger',  class: 'TE b',  marks: 88  },  {  \_id: ObjectId("65451e0a4aa41441fbf9f76b"),  name: 'varun',  class: 'TE b',  marks: 89  }  ]   1. Display list of students who play football   students> **db.students.find({sports:"football"},{name:1,rollno:1,sports:1});**  [  {  \_id: ObjectId("65451dcb4aa41441fbf9f769"),  rollno: 4,  name: 'kartik',  sports: 'football'  },  {  \_id: ObjectId("65451de74aa41441fbf9f76a"),  rollno: 5,  name: 'tiger',  sports: 'football'  }  ] |