## Creating an Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex({ "name": 1 });
name_1
```

## Creating an Compound Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex(
                                          { "major": 1, "age": -1 });
major_1_age_-1
```

## Creating an Index with Partial Expression

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex({ "age": 1 },
                         { partialFilterExpression: { "major": "Computer Science" } });
age_1
```

## Demonstration

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.find({name:"Eva Brown"})
[
  {
    _id: ObjectId("65336bb20d506ee59b537770"),
    name: 'Eva Brown',
    age: 19,
    major: 'Biology',
    courses: [ 'Genetics', 'Ecology' ]
  }
]
```

## Demonstration with explain

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.find({name:"Eva Brown"}).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'test.students',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'Eva Brown' } },
    queryHash: '64908032',
    planCacheKey: '39E6C6F3',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { name: 1 },
        indexName: 'name_1',
        isMultiKey: false,
        multiKeyPaths: { name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { name: [ '["Eva Brown", "Eva Brown"]' ] }
      }
    },
    rejectedPlans: []
  },
  command: { find: 'students', filter: { name: 'Eva Brown' }, '$db': 'test' },
  serverInfo: {
    host: 'ac-bgmpmhb-shard-00-01.0r9xgkx.mongodb.net',
    port: 27017,
    version: '6.0.11',
    gitVersion: 'f797f841eaf1759c770271ae00c88b92b2766eed'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 16793600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 33554432,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600
  },
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1697869177, i: 13 }),
    signature: {
      hash: Binary(Buffer.from("52256a8fd29aca656241d70a21cdaa0eda5d8a54", "hex"), 0),
      keyId: Long("7247763952019439618")
    }
  },
  operationTime: Timestamp({ t: 1697869177, i: 13 })
}
```

## Creating a TTL Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex({ "expireAt": 1 },
                                       { expireAfterSeconds: 1800 });
expireAt_1
```

## Creating a Multi-key Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex({ "courses": 1 });
courses_1
```

## Creating a Text Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.createIndex({ "name": "text",
"major": "text" });
name_text_major_text
```

## Dropping An Index

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.dropIndex("expireAt_1");
{
  nIndexesWas: 4,
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1697869822, i: 11 }),
    signature: {
      hash: Binary(Buffer.from("023e6e10bbb28c9c87e8150bc2e0e888438fad96", "hex"), 0),
      keyId: Long("7247763952019439618")
    }
  },
  operationTime: Timestamp({ t: 1697869822, i: 9 })
}
```

## Displaying all the indexes present in the collection

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.students.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { courses: 1 }, name: 'courses_1' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'name_text_major_text',
    weights: { major: 1, name: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
```

## Sum of a Field

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.sales.aggregate([
...   {
...     $group: {
...       _id: null,
...       totalRevenue: { $sum: { $multiply: ["$quantity", "$price"] } }
...     }
...   }
... ]);
[ { _id: null, totalRevenue: 1503.82 } ]
```

## Average of a Field

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.sales.aggregate([
...   {
...     $group: {
...       _id: null,
...       averagePrice: { $avg: "$price" }
...     }
...   }
... ]);
[ { _id: null, averagePrice: 17.544 } ]
```

## Group and Count

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.sales.aggregate([
...   {
...     $group: {
...       _id: "$product",
...       count: { $sum: 1 }
...     }
...   }
... ]);
[
  { _id: 'Widget A', count: 3 },
  { _id: 'Widget C', count: 3 },
  { _id: 'Widget B', count: 3 },
  { _id: 'Widget D', count: 1 }
]
```

## Filter and Count

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.sales.aggregate([
...   {
...     $match: {
...       price: { $gt: 10 }
...     }
...   },
...   {
...     $group: {
...       _id: null,
...       count: { $sum: 1 }
...     }
...   }
... ]);
[ { _id: null, count: 10 } ]
```

## Sort and Limit

```
Atlas atlas-a0ugoa-shard-0 [primary] test> db.sales.aggregate([
...   {
...     $group: {
...       _id: "$product",
...       totalRevenue: { $sum: { $multiply: ["$quantity", "$price"] } }
...     }
...   },
...   {
...     $sort: { totalRevenue: -1 }
...   },
...   {
...     $limit: 5
...   }
... ]);
[
  { _id: 'Widget C', totalRevenue: 566.53 },
  { _id: 'Widget B', totalRevenue: 426.5 },
  { _id: 'Widget A', totalRevenue: 330.79 },
  { _id: 'Widget D', totalRevenue: 180 }
]
```