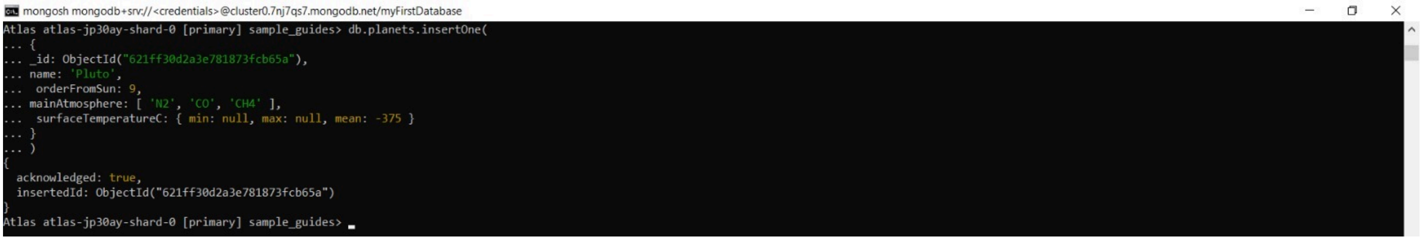
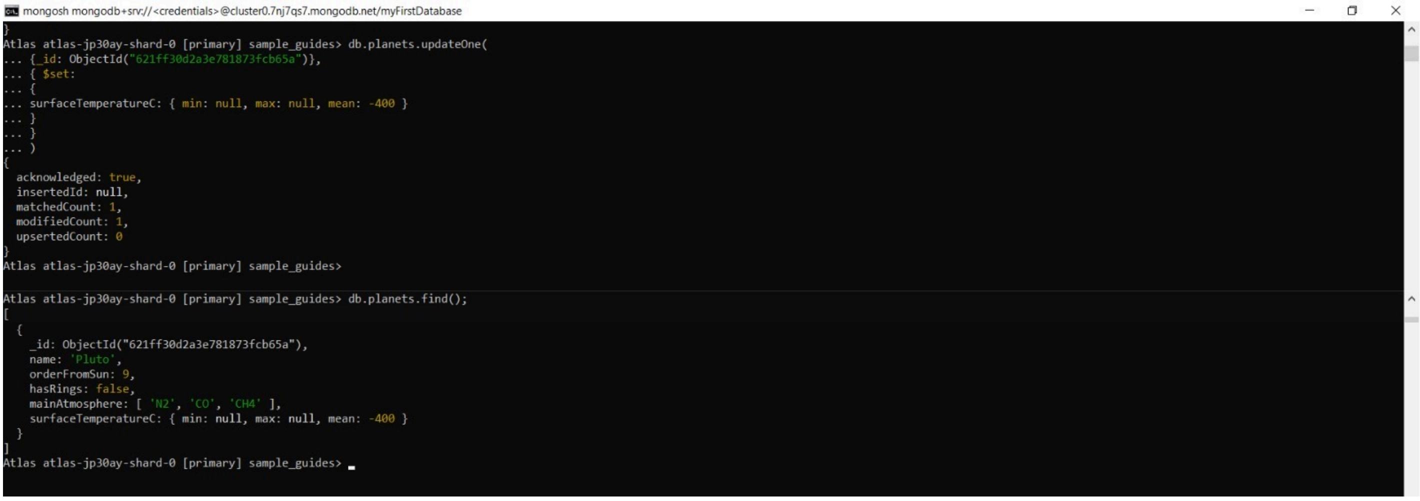
Lab 1

- Insert document in database.


mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

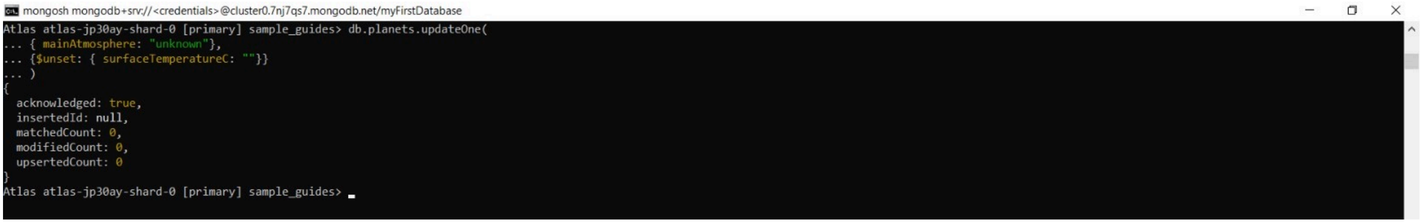
```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.insertOne(
... {
... _id: ObjectId("621ff30d2a3e781873fcb65a"),
... name: 'Pluto',
...  orderFromSun: 9,
... mainAtmosphere: [ 'N2', 'CO', 'CH4' ],
...  surfaceTemperatureC: { min: null, max: null, mean: -375 }
... }
... )
{
  acknowledged: true,
  insertedId: ObjectId("621ff30d2a3e781873fcb65a")
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> _
```

- Executing update operations using all update operators ($set, $unset, $push, etc).

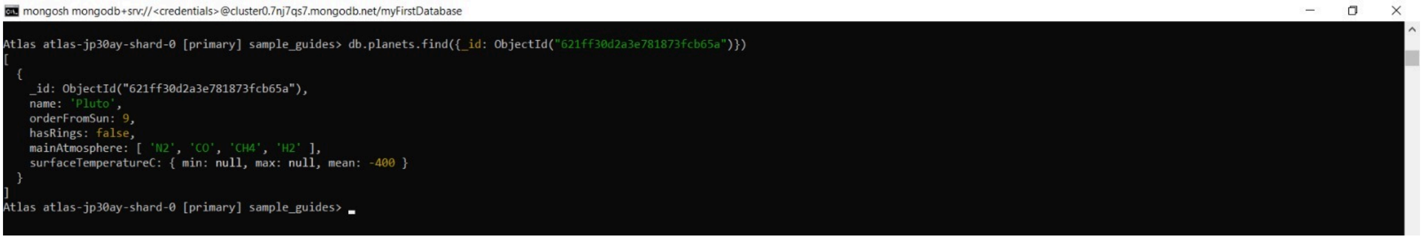

mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.updateOne(
... {_id: ObjectId("621ff30d2a3e781873fcb65a")},
... { $set:
... {
... surfaceTemperatureC: { min: null, max: null, mean: -400 }
... }
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides>
```

```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.find();
[
  {
    _id: ObjectId("621ff30d2a3e781873fcb65a"),
    name: 'Pluto',
    orderFromSun: 9,
    hasRings: false,
    mainAtmosphere: [ 'N2', 'CO', 'CH4' ],
    surfaceTemperatureC: { min: null, max: null, mean: -400 }
  }
]
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> _
```

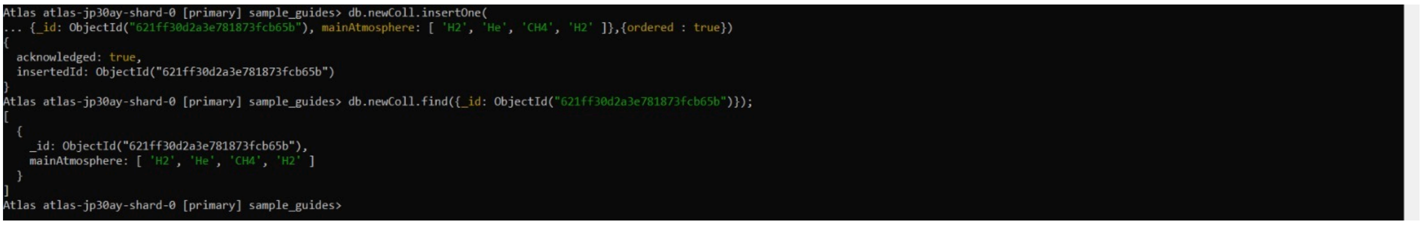

mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.updateOne(
... { mainAtmosphere: "unknown"},
... {$unset: { surfaceTemperatureC: ""}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> _
```

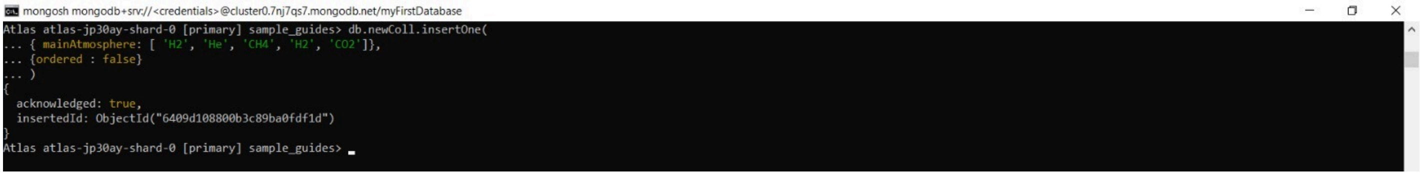

mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.find({_id: ObjectId("621ff30d2a3e781873fcb65a")})
[
  {
    _id: ObjectId("621ff30d2a3e781873fcb65a"),
    name: 'Pluto',
    orderFromSun: 9,
    hasRings: false,
    mainAtmosphere: [ 'N2', 'CO', 'CH4', 'H2' ],
    surfaceTemperatureC: { min: null, max: null, mean: -400 }
  }
]
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> _
```

- Executing insertion queries with both insertOne and insertMany with ordered flag.

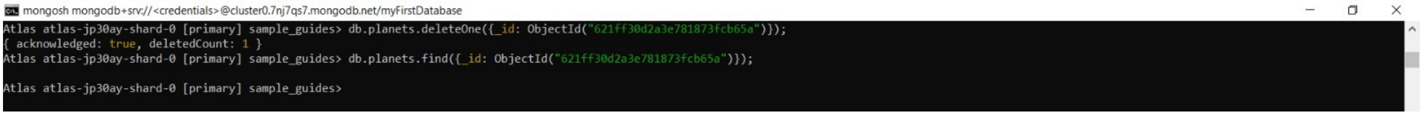


```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.newColl.insertOne(
... {_id: ObjectId("621ff30d2a3e781873fcb65b"), mainAtmosphere: [ 'H2', 'He', 'CH4', 'H2' ]},{ordered : true})
{
  acknowledged: true,
  insertedId: ObjectId("621ff30d2a3e781873fcb65b")
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.newColl.find({_id: ObjectId("621ff30d2a3e781873fcb65b")});
[
  {
    _id: ObjectId("621ff30d2a3e781873fcb65b"),
    mainAtmosphere: [ 'H2', 'He', 'CH4', 'H2' ]
  }
]
Atlas atlas-jp30ay-shard-0 [primary] sample_guides>
```


mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

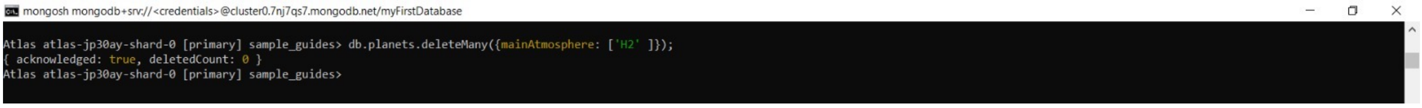
```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.newColl.insertOne(
... { mainAtmosphere: [ 'H2', 'He', 'CH4', 'H2', 'CO2']},
... {ordered : false}
... )
{
  acknowledged: true,
  insertedId: ObjectId("6409d108800b3c89ba0fdf1d")
}
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> 
```

- Executing deleteOne and deleteMany operations.


mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

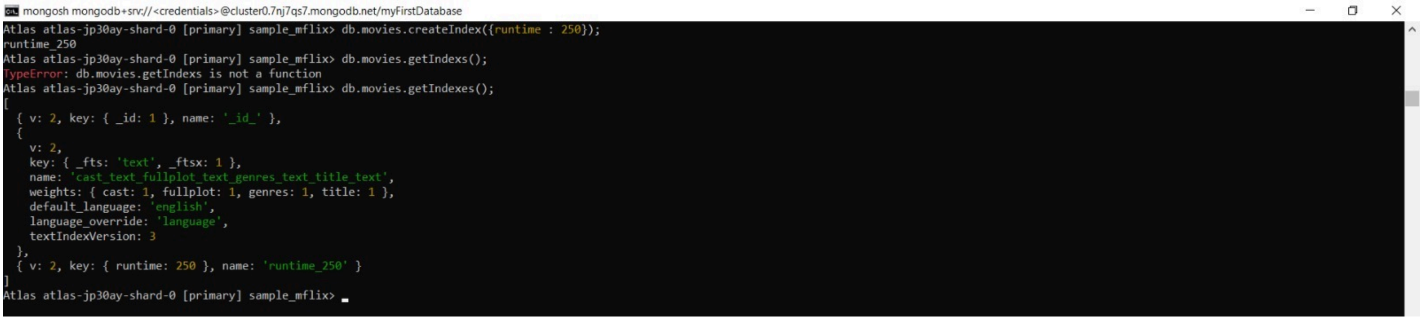
```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.deleteOne({_id: ObjectId("621ff30d2a3e781873fcb65a")});
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.find({_id: ObjectId("621ff30d2a3e781873fcb65a")});

Atlas atlas-jp30ay-shard-0 [primary] sample_guides>
```

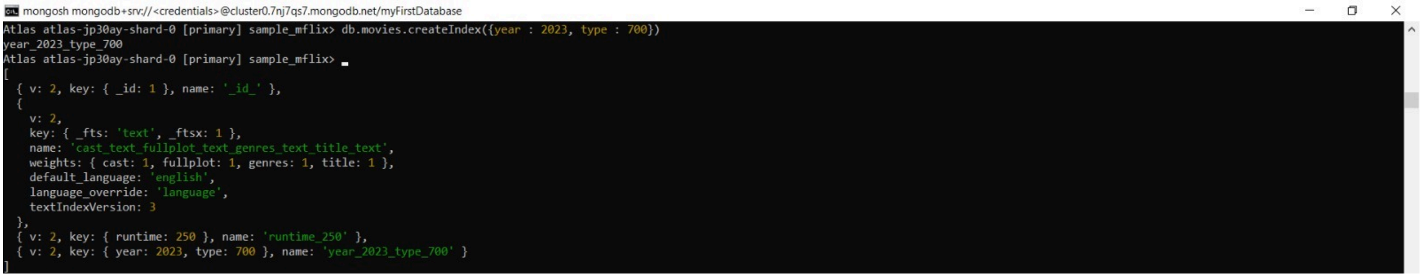

mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_guides> db.planets.deleteMany({mainAtmosphere: ['H2' ]});
{ acknowledged: true, deletedCount: 0 }
Atlas atlas-jp30ay-shard-0 [primary] sample_guides>
```

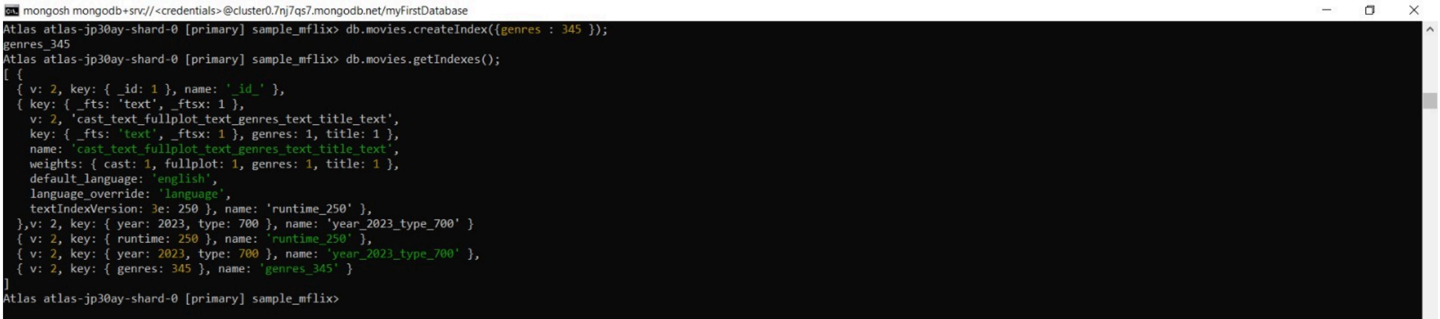
- Creating indexes (single field, compound, multi-key and text indexes).


mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.createIndex({runtime : 250});
runtime_250
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.getIndexs();
TypeError: db.movies.getIndexs is not a function
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  },
  { v: 2, key: { runtime: 250 }, name: 'runtime_250' }
]
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> 
```

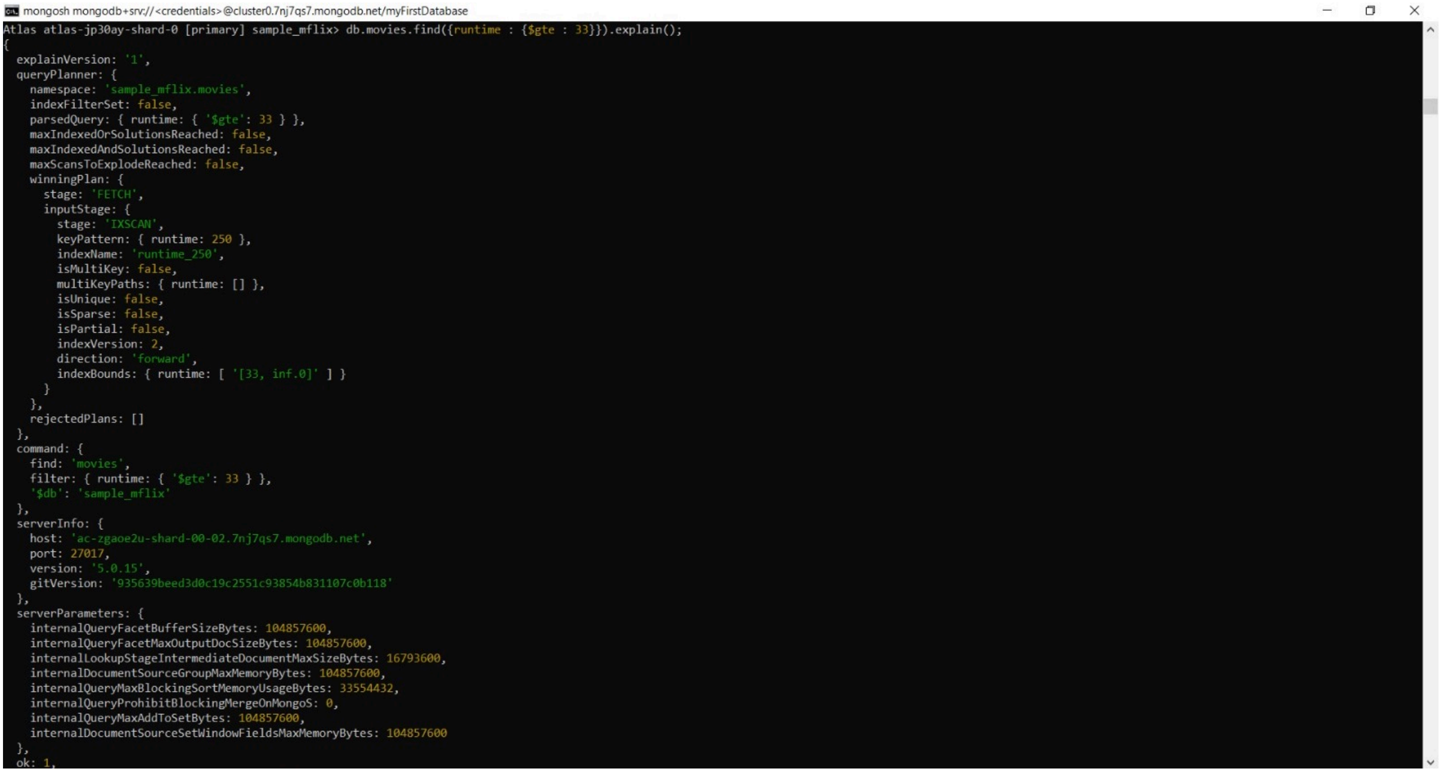

mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.createIndex({year : 2023, type : 700})
year_2023_type_700
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> 
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  },
  { v: 2, key: { runtime: 250 }, name: 'runtime_250' },
  { v: 2, key: { year: 2023, type: 700 }, name: 'year_2023_type_700' }
]
```


mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

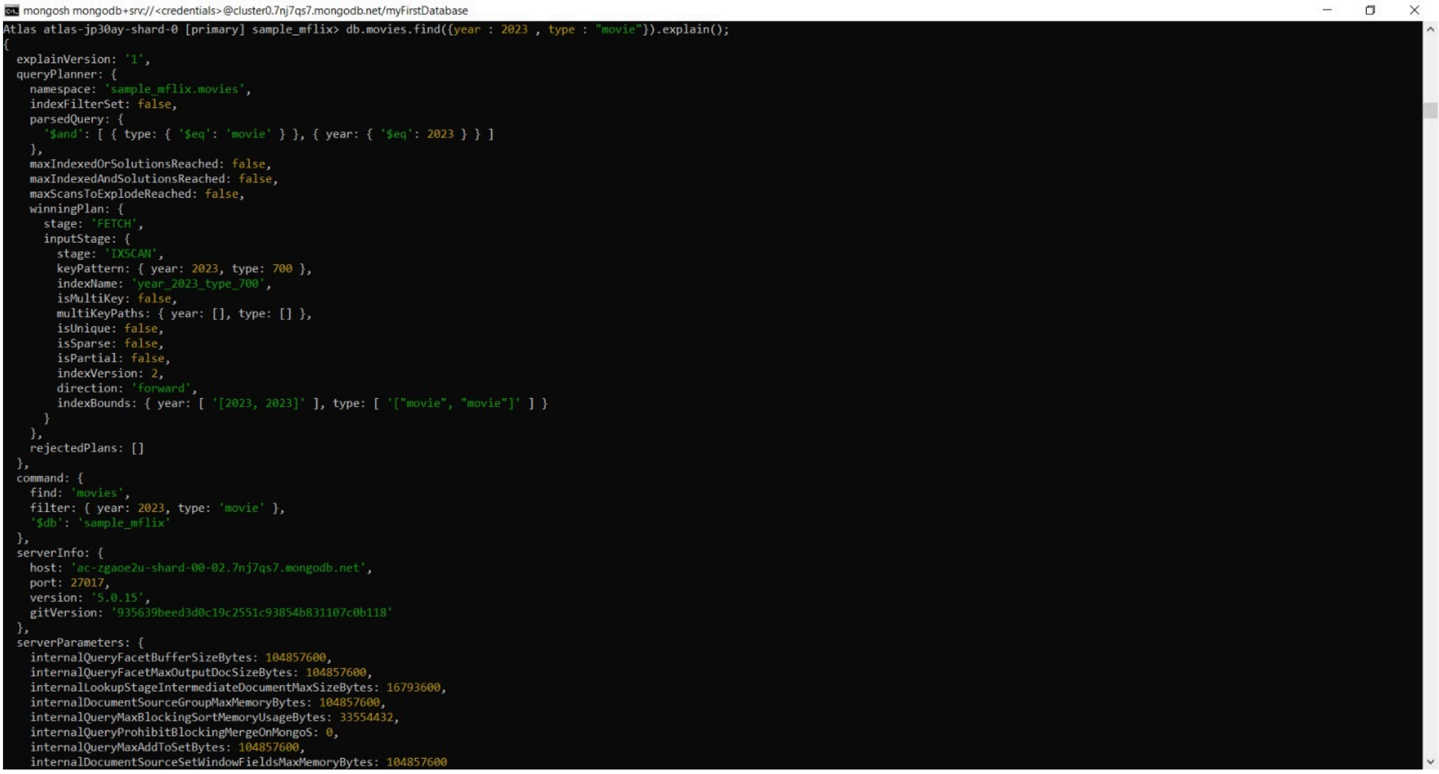
```
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.createIndex({genres : 345 });
genres_345
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.getIndexes();
[ {
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { key: { _fts: 'text', _ftsx: 1 },
    v: 2, 'cast_text_fullplot_text_genres_text_title_text',
    key: { _fts: 'text', _ftsx: 1 }, genres: 1, title: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3e: 250 }, name: 'runtime_250' },
  },v: 2, key: { year: 2023, type: 700 }, name: 'year_2023_type_700' }
  { v: 2, key: { runtime: 250 }, name: 'runtime_250' },
  { v: 2, key: { year: 2023, type: 700 }, name: 'year_2023_type_700' },
  { v: 2, key: { genres: 345 }, name: 'genres_345' }
]
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix>
```

- Using cursor.explain with different queries and reading and understanding the winningPlan.



mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.find({runtime : {$gte : 33}}).explain();
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'sample_mflix.movies',
    indexFilterSet: false,
    parsedQuery: { runtime: { '$gte': 33 } },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { runtime: 250 },
        indexName: 'runtime_250',
        isMultiKey: false,
        multiKeyPaths: { runtime: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { runtime: [ '[33, inf.0]' ] }
      }
    },
    rejectedPlans: []
  },
  command: {
    find: 'movies',
    filter: { runtime: { '$gte': 33 } },
    '$db': 'sample_mflix'
  },
  serverInfo: {
    host: 'ac-zgaoe2u-shard-00-02.7nj7qs7.mongodb.net',
    port: 27017,
    version: '5.0.15',
    gitVersion: '935639beed3d0c19c2551c93854b831107c0b118'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 16793600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 33554432,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600
  },
  ok: 1,
```



mongosh mongodb+srv://<credentials>@cluster0.7nj7qs7.mongodb.net/myFirstDatabase

```
Atlas atlas-jp30ay-shard-0 [primary] sample_mflix> db.movies.find({year : 2023 , type : "movie"}).explain();
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'sample_mflix.movies',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [ { type: { '$eq': 'movie' } }, { year: { '$eq': 2023 } } ]
    },
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { year: 2023, type: 700 },
        indexName: 'year_2023_type_700',
        isMultiKey: false,
        multiKeyPaths: { year: [], type: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { year: [ '[2023, 2023]' ], type: [ '["movie", "movie"]' ] }
      }
    },
    rejectedPlans: []
  },
  command: {
    find: 'movies',
    filter: { year: 2023, type: 'movie' },
    '$db': 'sample_mflix'
  },
  serverInfo: {
    host: 'ac-zgaoe2u-shard-00-02.7nj7qs7.mongodb.net',
    port: 27017,
    version: '5.0.15',
    gitVersion: '935639beed3d0c19c2551c93854b831107c0b118'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 16793600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 33554432,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600
```