

Lab 3

Create a collection by the name blogPosts and it has 3 fields id, title and comments. In the collection the comments field is an array which consists of user details. Each collection

consists of two user details inside the comments array- user name and text

```
db.createCollection("blogPosts");
```

Demonstrate the following:

1. Adding an element into array

```
db.blogPosts.insert({_id:1, title: "Intro", comments:[{username:"Revanth", text: "Article"}]});
```

```
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.find()
[
  {
    _id: 1,
    title: 'Intro',
    comments: [ { username: 'Revanth', text: 'Article' } ]
  }
]
```

2. Display second element

```
db.blogPosts.update({_id: 1},{ $push:{comments: {username:" D", text:" uploading"}}})
```

```
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.update({_id: 1},{ $push:{comments: {username:" D", text:" uploading"}}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.find()
[
  {
    _id: 1,
    title: 'Intro',
    comments: [
      { username: 'Revanth', text: 'Article' },
      { username: ' D', text: ' uploading' }
    ]
  }
]
Atlas atlas-pn06xx-shard-0 [primary] test> |
```

3. Display size of the array

```
db.blogPosts.aggregate([{$project:{_id:1, title:1, comments:1, commentSize:{$size:
"$comments"}}}])
```

```
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.aggregate([{$project:{_id:1, title:1, comments:1, commentSize:{$
size: "$comments"}}}])
[
  {
    _id: 1,
    title: 'Intro',
    comments: [
      { username: 'Revanth', text: 'Article' },
      { username: 'D', text: 'uploading' }
    ],
    commentSize: 2
  },
  {
    _id: 2,
    title: 'Blog-one',
    comments: [
      { username: 'Revanth', text: 'Article' },
      { username: 'D', text: 'uploading' }
    ],
    commentSize: 2
  },
  {
    _id: 3,
    title: 'ExiteBlog',
    comments: [ { username: 'Xyz', text: 'Exciting Blog' } ],
    commentSize: 1
  }
]
Atlas atlas-pn06xx-shard-0 [primary] test> |
```

4. Display first two elements of the array

```
db.blogPosts.aggregate([ { $project: { _id: 1, title: 1, comments: { $slice: ["$comments",
2] } } } ])
```

```
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.aggregate([ { $project: { _id: 1, title: 1, comments: { $slice:
["$comments", 2] } } } ])
[
  {
    _id: 1,
    title: 'Intro',
    comments: [
      { username: 'Revanth', text: 'Article' },
      { username: 'D', text: 'uploading' }
    ]
  },
  {
    _id: 2,
    title: 'Blog-one',
    comments: [
      { username: 'Revanth', text: 'Article' },
      { username: 'D', text: 'uploading' }
    ]
  },
  {
    _id: 3,
    title: 'ExiteBlog',
    comments: [ { username: 'Xyz', text: 'Exciting Blog' } ]
  }
]
Atlas atlas-pn06xx-shard-0 [primary] test> |
```

5. Update the document with id 4 and replace the element present in 1st index position of the array with another array

```
db.blogPosts.update( { _id: 4 }, { $set: { "comments.1": [ { username: 'Jkl', text: 'Content prepared' }, { username: 'JKL', text: 'Content uploaded' } ] } } )
```

```
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.update( { _id: 4 }, { $set: { "comments.1": [ { username: 'Jkl', text: 'Content prepared' }, { username: 'JKL', text: 'Content uploaded' } ] } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-pn06xx-shard-0 [primary] test> db.blogPosts.find({_id:4})
[
  {
    _id: 4,
    title: 'IEEE',
    comments: [
      { username: 'Jkl', text: 'Blog on IEEE Paper' },
      [
        { username: 'Jkl', text: 'Content prepared' },
        { username: 'JKL', text: 'Content uploaded' }
      ]
    ]
  }
]
```