

1..Create a Database named 'MYWall'.

```
> use MYWall
switched to db MYWall
> db
MYWall
> █
```

2.Create two collections named 'user and "post' .

```
> db.createCollection("user")
{ "ok" : 1 }
> db.createCollection("post")
{ "ok" : 1 }
> show collections
post
user
```

3.Insert some post related information like post_id, title, description , posted_by, comments, created_at etc...

4.Comments should be an array of objects which has comment_id, message, timestamp.

Without comment_id:

```
> db.post.insert({title:'MongoDB Exercise',description:'Like the post',posted_by:'Devansh',createdAt:'ttn',likes:'90',comments:[{message:"Greenry post",time:new Date(),likes:90},{message:"Beach post",time:new Date(),tags:["Goa","Maldives"]},{message:"Technical post",time:new Date(),likes:60}]})
WriteResult({ "nInserted" : 1 })
> db.post.find()
{ "_id" : ObjectId("5eb038f5c15be3a198f625b4"), "title" : "MongoDB Exercise", "description" : "Like the post", "posted_by" : "Devansh", "createdAt" : "ttn", "likes" : "90", "comments" : [ { "message" : "Greenry post", "time" : ISODate("2020-05-04T15:47:01.237Z"), "likes" : 90 }, { "message" : "Beach post", "time" : ISODate("2020-05-04T15:47:01.237Z"), "tags" : [ "Goa", "Maldives" ] }, { "message" : "Technical post", "time" : ISODate("2020-05-04T15:47:01.237Z"), "likes" : 60 } ] }
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb038f5c15be3a198f625b4"),
  "title" : "MongoDB Exercise",
  "description" : "Like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : "90",
  "comments" : [
    {
      "message" : "Greenry post",
      "time" : ISODate("2020-05-04T15:47:01.237Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T15:47:01.237Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T15:47:01.237Z"),
      "likes" : 60
    }
  ]
}
```

Comment_id:

```
> db.post.find()
{ "_id" : ObjectId("5eb04427c15be3a198f625b7"), "title" : "MongoDB Exercise", "description" : "Description Updated to like the post", "posted_by" : "Devansh", "createdAt" : "ttn", "likes" : 100, "comments" : [ { "message" : "Updation in comment message", "time" : ISODate("2020-05-04T16:34:47.482Z"), "likes" : 90 }, { "message" : "Beach post", "time" : ISODate("2020-05-04T16:34:47.482Z"), "tags" : [ "Goa", "Maldives" ] }, { "message" : "Technical Post", "time" : ISODate("2020-05-04T16:34:47.482Z"), "likes" : 60 }, null ] }
{ "_id" : ObjectId("5eb04e90c15be3a198f625b8"), "title" : "MongoDB Exercise", "description" : "Like the post", "posted_by" : "Devansh", "createdAt" : "ttn", "likes" : "90", "comments" : [ { "message" : "Greenry post", "time" : ISODate("2020-05-04T17:19:12.612Z"), "likes" : 90 }, { "message" : "Beach post", "time" : ISODate("2020-05-04T17:19:12.612Z"), "tags" : [ "Goa", "Maldives" ] }, { "message" : "Technical post", "time" : ISODate("2020-05-04T17:19:12.612Z"), "likes" : 60 } ] }
{ "_id" : ObjectId("5eb04f01c15be3a198f625b9") }
{ "_id" : ObjectId("5eb04fab15be3a198f625ba"), "title" : "MongoDB Exercise", "description" : "Like the post", "posted_by" : "Devansh", "createdAt" : "ttn", "likes" : "90", "comments" : [ { "comment_id" : 100, "message" : "Greenry post", "time" : ISODate("2020-05-04T17:23:55.535Z"), "likes" : 90 }, { "comment_id" : 200, "message" : "Beach post", "time" : ISODate("2020-05-04T17:23:55.535Z"), "tags" : [ "Goa", "Maldives" ] }, { "comment_id" : 300, "message" : "Technical post", "time" : ISODate("2020-05-04T17:23:55.535Z"), "likes" : 60 } ] }
>
```

5. Write query to add a new comment for a post

```
> db.post.update({"posted_by": 'Devansh'}, {$push: {"comments": {"message": "New Value added", updateTime: new Date()}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb03d48c15be3a198f625b6"),
  "title" : "MongoDB Exercise",
  "description" : "Like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : "90",
  "comments" : [
    {
      "message" : "Greenry post",
      "time" : ISODate("2020-05-04T16:05:28.571Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T16:05:28.571Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T16:05:28.571Z"),
      "likes" : 60
    },
    {
      "message" : "New Value added",
      "updatedAt" : ISODate("2020-05-04T16:07:15.396Z")
    }
  ]
}
```

6. Write a query to update description of a post.

```
> db.post.update({'description': "Like the post"}, {$set: {'description': "Description Updated to like the post"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb04427c15be3a198f625b7"),
  "title" : "MongoDB Exercise",
  "description" : "Description Updated to like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : "90",
  "comments" : [
    {
      "message" : "Greenry post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 60
    }
  ]
}
```

7. Write query to update comment message.

```
> db.post.update({'createdAt':'ttn'},{$set: {"comments.0.message":"Updation in comment message"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb04427c15be3a198f625b7"),
  "title" : "MongoDB Exercise",
  "description" : "Description Updated to like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : 100,
  "comments" : [
    {
      "message" : "Updation in comment message",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 60
    }
  ]
}
```

8. Write query to remove a comment.

Before removing last comment:-

```
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb04427c15be3a198f625b7"),
  "title" : "MongoDB Exercise",
  "description" : "Description Updated to like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : 100,
  "comments" : [
    {
      "message" : "Greenry post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 60
    },
    {
      "message" : "New Value added",
      "updatedAt" : ISODate("2020-05-04T16:36:07.070Z")
    }
  ]
}
```

After removing last comment:-

```
> db.post.update({'createdAt':'ttn'},{$pop: {'comments':1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.post.find().pretty()
{
  "_id" : ObjectId("5eb04427c15be3a198f625b7"),
  "title" : "MongoDB Exercise",
  "description" : "Description Updated to like the post",
  "posted_by" : "Devansh",
  "createdAt" : "ttn",
  "likes" : 100,
  "comments" : [
    {
      "message" : "Greenry post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 90
    },
    {
      "message" : "Beach post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "tags" : [
        "Goa",
        "Maldives"
      ]
    },
    {
      "message" : "Technical post",
      "time" : ISODate("2020-05-04T16:34:47.482Z"),
      "likes" : 60
    }
  ]
}
```