

PRACTICAL - 6: Working with NoSQL Databases. (MongoDB)

1. Install MongoDB on your operating system. (Windows/Linux/Mac)

```
C:\Users\parth>mongo
MongoDB shell version v5.0.1
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0492765d-3ff3-42e6-a8d6-62fdb771196b") }
MongoDB server version: 5.0.1
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed
in an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-07-24T17:13:38.330+05:30: Access control is not enabled for the database. Read and write access to
on is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

2. Create a database of your name on MongoDB. verify whether the database shows in the list or not.

```
> use ParthKukadiya
switched to db ParthKukadiya
>
```

3. Create a collection named "stories" and verify whether the database shows in the list or not.

```
> db.createCollection("stories");
{ "ok" : 1 }
> show collections
stories
> db
ParthKukadiya
>
```

4. Create documents under stories, where various documents contains: time of putting the story, name of the person who viewed the story, comment on the story, number of shares of the story and so on.

```
> db.stories.insert({name : 'fairy tale' , time : '11.11 AM' , viewedBy : 'person 1' , comment : 'charming' , shares : '22'})
WriteResult({ "nInserted" : 1 })
> db.stories.insert({name : 'horror' , time : '11.11 PM' , viewedBy : 'person 2' , comment : 'thrilling' , shares : '121'})
WriteResult({ "nInserted" : 1 })
```

5. Enlist all the documents under the collection stories. – Just like the sql query of “select * from the stories”.

```
> db.stories.find()
{ "_id" : ObjectId("60fc23f48df50300ebc985bf"), "name" : "fairy tale", "time" : "11.11 AM", "viewedBy" : "person 1", "comment" : "charming", "shares" : "22" }
{ "_id" : ObjectId("60fc241b8df50300ebc985c0"), "name" : "horror", "time" : "11.11 PM", "viewedBy" : "person 2", "comment" : "thrilling", "shares" : "121" }
>
```