**CRUD**

1)  
mongorestore - -gzip - -archive=dump/archive.gz –db PostDB

2)  
Find tow articles which author is bbc news + projection just title and published date  
  
db.articles.find({author\_id:"bbc\_news"}, {title: 1, publishedAt: 1}).limit(2)

3)

create one comment:

db.comments.insertOne({

author: "Maxim",

text: "This is fantastic!!",

date: "2016-11-14T13:32:17Z",

artilce\_id: "article\_2"

});  
  
and you can check result:  
db.comments.find({author: 'Maxim'})  
  
4)  
update   
db.comments.replaceOne({author: 'Maxim'}, {author: 'Maxim', text: 'This is awful', date: '2016-12-14T13:32:17Z', article\_id: 'article\_4'})  
and you can check result:  
db.comments.find({author: 'Maxim'})  
  
5)   
  
delete comments for article\_2  
  
db.comments.deleteMany({article\_id: 'article\_2'})  
  
result: 6 were deleted

6)

Find articles with comments\_ids great than 12350:

db.articles.find({comments\_ids: {$gt: 12350}}, {comments\_ids: 1})

7)

find the article with Donald Trump in title  
db.articles.find({title: {$regex: /Donald Trump/}});  
8)

For example we want to find articles with similar tags (football)  
  
db.articles.find({tags: {$regex: /football/}});

**Indexing**:

1)

Want to create Index for Search by /Donald Trump/ text

db.articles.createIndex({title: ‘text’})

after that we can find  
db.articles.find($text: {$search: ‘Donald Trump’});

2)

very often wants to find article by tags, for it we can create Index

db.articles.createText({‘tags’: 1})

after that we can find

db.articles.find({tags: ‘BBC’});

3)

**Aggregation**:

Firstly:   
mongoimport --db StudentsDB --collection students --upsertFields=\_id --file grapes.json

db.students.aggregate([{$project: {student\_id: '$student\_id', class\_id: '$class\_id', scores:{$filter: {input: '$scores', as: 'score', cond: {$ne: ['$$score.type', 'quiz']}}}}}, {$unwind: '$scores'}, {$group: {\_id: {class\_id: '$class\_id', student\_id: '$student\_id'}, avgStudentScore: {$avg: '$scores.score'}}}, {$group: {\_id: '$\_id.class\_id', avgClassScore: {$avg: '$avgStudentScore'} }}, {$sort: {avgClassScore: -1}}, {$group:{ \_id: false, maxAverage: { $first: '$avgClassScore'}}}])

**Replication**:

create directory   
/replication/db/rs0

/replication/db/rs1

/replication/db/rs2

create replications   
mongod --port 27017 --replSet 'rs0' --dbpath replication/rs0

mongo

rs.initiate()

rs.conf();

mongod --port 27018 --replSet 'rs0' --dbpath replication/rs1

mongod --port 27019 --replSet 'rs0' --dbpath replication/rs2

rs.add(‘<yourpc>:27018’)

{ "ok" : 1 }

rs.add(‘<yourpc>:27019’)

{ "ok" : 1 }