**Querying**

**insert**

> db.marks.insert({roll:1,name:"raj",phone:9999999999});

WriteResult({ "nInserted" : 1 })

> db.marks.insert({roll:2,name:"sham",phone:9999999998});

WriteResult({ "nInserted" : 1 })

> db.marks.insert({roll:3,name:"ram",phone:9999999997});

WriteResult({ "nInserted" : 1 })

**find();**

> db.marks.find();

{ "\_id" : ObjectId("54095871c2dab0fe124e0d4d"), "roll" : 1, "name" : "raj", "pho

ne" : 9999999999 }

{ "\_id" : ObjectId("54095899c2dab0fe124e0d4e"), "roll" : 2, "name" : "sham", "ph

one" : 9999999998 }

**update**

> db.marks.update({roll:2},{$set:{name:"Lahu"}});

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

**Find the exact record in db:**

> db.marks.find({"roll" : 5 ,"name" : "mukesh"});

{ "\_id" : ObjectId("540958e0c2dab0fe124e0d51"), "roll" : 5, "name" : "mukesh", "

phone" : 9999999995 }

**Query Conditionals:**

This would find all documents where the “roll" field was greater than or equa 2 to 5

AND less than or equal to 5.

> db.marks.find({"roll" : {"$gte" : 2, "$lte" : 5}})

{ "\_id" : ObjectId("54095899c2dab0fe124e0d4e"), "roll" : 2, "name" : "Lahu", "ph

one" : 9999999998 }

**OR Querying**

"$not" is a metaconditional: it can be applied on top of any other criteria. As an example,

let’s consider the modulus operator, "$mod". "$mod" queries for keys whose values, when

divided by the first value given, have a remainder of the second value:

> db.marks.find({"roll" : {"$mod" : [5, 1]}});

{ "\_id" : ObjectId("54095871c2dab0fe124e0d4d"), "roll" : 1, "name" : "raj", "pho

ne" : 9999999999 }

**$not**

"$not" is a metaconditional: it can be applied on top of any other criteria. As an example,

let’s consider the modulus operator, "$mod". "$mod" queries for keys whose values, when

divided by the first value given, have a remainder of the second value:

> db.marks.find({"roll" : {"$not" : {"$mod" : [5, 1]}}});

{ "\_id" : ObjectId("54095899c2dab0fe124e0d4e"), "roll" : 2, "name" : "Lahu", "ph

one" : 9999999998 }

> db.marks.find({"roll" : {"$not" : {"$mod" : [5, 3]}}});

{ "\_id" : ObjectId("54095871c2dab0fe124e0d4d"), "roll" : 1, "name" : "raj", "pho

ne" : 9999999999 }

{ "\_id" : ObjectId("54095899c2dab0fe124e0d4e"), "roll" : 2, "name" : "Lahu", "ph

**Indexing**

// Creating Server Connections

C:\mongo\bin>

C:\mongo\bin> cd \

C:\>

C:\> cd C:\mongo\bin

C:\mongo\bin>mongod

// Creating a client

C:\Users\admin1>

C:\Users\admin1>cd \

C:\>

C:\> cd C:\mongo\bin

C:\mongo\bin>mongo

//Create Database in Client

db

test

use mydb

Create Collection

> db.createCollection("student12")

Insert Documents in Collection

> db.student12.insert({name:"amit",Roll\_no:1,sub:"TOC",Marks:80})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"amita",Roll\_no:2,sub:"TOC",Marks:100})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"ram",Roll\_no:3,sub:"TOC",Marks:90})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"ramesh",Roll\_no:4,sub:"TOC",Marks:95})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"amit",Roll\_no:1,sub:"DBMS",Marks:70})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"Swami",Roll\_no:5,sub:"DBMS",Marks:80})

WriteResult({ "nInserted" : 1 })

> db.student12.insert({name:"SwamiRaj",Roll\_no:6,sub:"DBMS",Marks:10})

WriteResult({ "nInserted" : 1 })

> db.student12.find();

> db.student12.ensureIndex({Roll\_no:1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 2,

"numIndexesAfter" : 3,

"ok" : 1

}

> db.student12.find({Rollno:7}).explain();

> db.abc.find().explain();