**Comparison Operator:**

**#syntax:{field: {operator:value}}**

* $eq
* $ne
* $gt
* $gte
* $lt
* $lte
* $in
* $nin

Example: {"salary":{$lt:5000}}

db.trips.find({"tripduration":{$lte:500}})

**Logical Operator:**

**Syntax: {Operator: [{condition1,condition2},…..]}**

* **$and**
* **$or**
* **$nor :** return all documents that fail all of condition
* **$not :** {$not: condition}, return documents that do not match the condition

**Example: db.trips.find({$and : [{"tripduration":{$gt:400}},{"birth year":{$gt:1988}}]})**

**: db.trips.find({$or : [{"tripduration":{$gt:400}},{"birth year":{$gt:1988}}]})**

**Equal to: db.trips.find({"tripduration":{$gt:400},"birth year":{$gt:1988}})** // implicit $and operator

db.trips.find({"tripduration":{$lt:400,$gt : 300}})

db.inspections.find({"result":{$in:["unable to locate","Violation Issued"]}})

db.inspections.find({$or:[{"result":"Violation Issued"},{"result":"Unable to Locate"}]})

**$expr:**

db.routes.find({$expr:{$eq:["$src\_airport","$dst\_airport"]}})

# benefirs, when comparing two field value in same documents

**Element Operator:**

Return documents that contain specified field

* **$exists :** Return document that contain specified field

**Syntax:** **Db.collection.find({field:{$exists:<booelan>}})**

* $type: Return fileds that contain value of specified data type
* **Syntax:** **Db.collection.find({field:{$type:<BSon type>}})**

db.companies.find({"ipo":{$exists:true}})

**Curson Method:**

* **Count:** Return the number of documents in the result
* db.collection.find({query}).count()
* **Sort:** sort record based on specified filed
* db.collection.find({query}).sort({parameters})
* **limit:** db.collection.find({query}).limit(number)
* **limit:** db.collection.find({query}).skip(number)
* **size:** **limit:** db.collection.find({query}).skip(number)

**db.trips.find().count()**

**Projection:**

**db.companies.find({},{"name":1,"founded\_year":1,"\_id":0})**

**Embeded Query:**

**db.inspections.find({"address.zip":11427})**

**Query Arrays:**

db.posts.find({"tags":{$all:["roof","wealth"]}},{"tags":1})

$size

db.grades.find({"scores":{$elemMatch:{"type":"exam","score":{$gt:75}}}})

**Insert:**

**Db.collection\_name.insert({})**

**db.testcol.insert({"name":"khan","sal":150})**

**. db.testcol.insert([{"name":"khan","sal":150},{"name":"Banan","sal":190}])**

**Deleting Documents:**

**db.testcol.deleteOne({"name":"khan"})**

**db.testcol.deleteMany({"name":"khan"})**

**Update Documents:**

**updateOne({filter},{update},{options})**

**updateMany(({filter},{update},{options})**

**db.people.updateMany({"profession":"chef"},{$set:{"profession":"cooked101"}})**

**Update Operator:**

* $unset
* $inc
* $rename
* $push

**db.people.updateMany({},{$unset:{"profession":""},$rename:{"hobbies\_list":"hobbies"}})**

**db.people.updateMany({},{$push:{"hobbies":["apple","banana"]}})**

**Aggregation Framework:**

**$match**

**db.grades.aggregate([{$match:{"scores":{$elemMatch:{"type":"exam","score":{$gt:85}}}}}])**

**db.companies.aggregate([{$match:{"founded\_year":{$gte:2005,$lte:2010}}}])**

**$project:**

**db.companies.aggregate([{$match:{"founded\_year":{$gte:2005,$lte:2010}}},{$project:{"founded\_year":1,"category\_code":1,"\_id":0}}])**

**db.companies.aggregate([{$project:{"number\_of\_employees":1,"\_id":0,"new\_num\_emp":"$number\_of\_employees"}}])**

**Arithmetic Operator:**

* **$add**
* **$subtract**
* **$divide**
* **$multiply**
* **$round**

**String Expression Operator:**

* **$concate**
* **$toUpper**
* **$toLower**
* **$ltrim**
* **#split**

**db.companies.aggregate({$project:{"name":{$toUpper:"$name"}}})**

**Date expression operator:**

**Array Expression Operator:**

* **$isArray**
* **$arrayElemAt**
* **$first**
* **$last**
* **$size**
* **$map**

**db.posts.aggregate({$project:{"\_id":0,"tags":1,"is\_array":{$isArray:"$tags"}}})**

**db.posts.aggregate({$project:{"\_id":0,"tags":1,"is\_array":{$isArray:"$tags"},"first\_ele":{$first:"$tags"},"last\_ele":{$last:"$tags"},"array\_size":{$size:"$tags"}}})**

**Conditional Expression:**

* **$cond**
* **$ifNull**
* **$switch**

**db.companies.aggregate([{$project:{"number\_of\_employees":{$ifNull:["$number\_of\_employees",0]},"\_id":0}}])**

**db.companies.aggregate([{$project:{"\_id":0,"number\_of\_employees":1,"size\_class":{$cond:{if:{$gt:["$number\_of\_employees",1000]},then:"large",else:"not large"}}}}])**

**$addField:**

**db.trips.aggregate([{$addFields:{"new\_feed":"$end station name"}}]**

**Cursor Stage:**

* **$sort**
* **$limit**
* **$count**
* **#skipp**

**db.trips.aggregate([{$sort:{"tripduration":-1}}])**

**db.trips.aggregate([{$sort:{"tripduration":-1}},{$limit:5}])**