***Inserting***

db.collection.insertOne() - Inserts a single document.

db.collection.insertMany() - Inserts multiple documents.

***Query Documents***

*db.collection.find({})*

SELECT \* FROM inventory

*db.collection.find({ status: 'D' })*

SELECT \* FROM inventory WHERE status = "D"

*db.collection.find({status:{$in:['A','D']}})*

SELECT \* FROM inventory WHERE status in ("A", "D")

*db.collection.find({status:'A',qty: {$lt:30}})*

SELECT \* FROM inventory WHERE status = "A" AND qty < 30

db.collection.find({ $or:[{status:'A'},{qty:{$lt:30}}]})

SELECT \* FROM inventory WHERE status = "A" OR qty < 30

db.collection.find({status:'A',$or:[{qty:{$lt:30}},{item:{ $regex:'^p'}}]})

SELECT \* FROM inventory WHERE status = "A" AND ( qty < 30 OR

item LIKE "p%")

***Match an Embedded/Nested Document***

For example, the following query selects all documents where the field size equals the document { h: 14, w: 21, uom: "cm" }:

db.collection.find({ size: { h: 14, w: 21, uom: 'cm' } });

Equality matches on the whole embedded document require an *exact* match of the specified <value> document, including the field order. For example, the following query does not match any documents in the inventory collection:

db.collection.find({ size: { w: 21, h: 14, uom: 'cm' } })

***Query on Nested Field:***

db.collection.find({ 'size.uom': 'in' })

db.collection.find({ 'size.h': { $lt: 15 } })

db.collection.find({ 'size.h': { $lt: 15 }, 'size.uom': 'in', status: 'D' })

***Query an Array:***

The following example queries for all documents where the field tags value is an array with exactly two elements, "red" and "blank", in the specified order:

db.collection.find({ tags: ['red', 'blank'] })

If, instead, you wish to find an array that contains both the elements "red" and "blank", without regard to order or other elements in the array, use the $all operator:

db.collection.find({tags: { $all: ['red', 'blank'] }})

The following example queries for all documents where tags is an array that contains the string "red" as one of its elements:

db.collection.find({ tags: 'red' })

The following operation queries for all documents where the array dim\_cm contains at least one element whose value is greater than 25.

db.collection.find({dim\_cm: { $gt: 25 }})

The following example queries for documents where the dim\_cm array contains elements that in some combination satisfy the query conditions; e.g., one element can satisfy the greater than 15 condition and another element can satisfy the less than 20 condition, or a single element can satisfy both:

db.collection.find({dim\_cm: { $gt: 15, $lt: 20 }})

Use $elemMatch operator to specify multiple criteria on the elements of an array such that at least one array element satisfies all the specified criteria.

The following example queries for documents where the dim\_cm array contains at least one element that is both greater than ($gt) 22 and less than ($lt) 30:

db.collection.find({dim\_cm:{$elemMatch:{ $gt: 22,$lt: 30 }}})