**CREATE AND READ : insertOne(), insertMany(), find(), findOne(), limit(), skip()**

**Insert single data:** pass a single object into ‘insertOne.()’ method

db.products.insertOne({……….});

**insert Multiple data**: pass an array of object into ‘insertMany.()’ method.

db.products.insertMany([{}, {}, ….]);

**find single data** or single query based on property or id:

db.products.find({\_id: ObjectId(“24324234”)});

another method is pass a query object. It will return the first matching value.

Db.products.find({name:’masud’, age:’21’});

**Find using query operator:**

db.movies.find( { rated: { $in: [ **"PG"**, **"PG-13"** ] } } );

**find data using logical operator:** we can use also mongodb set for query.

db.movies.find( { countries: "Mexico", "imdb.rating": { $gte: 7 } } )

**Limit the search result property:** we need to pass an object into find() function as second parameter. Which property don’t want to see in our response. Just set that property value as 0.

db.products.find({name:”masud”, age:’21’}, {age:0}); it return only name. because age set to zero.

**Limit the search or find result:** we can use limit() method by passing the limit value on search result query.

Db.products.find({name:’masud’}).limit(2);

**Skip data and limit result:** using skip() method by passing number. We can skip those number of data.

It skip the first 1 data from the query result.

Db.products.find({…}).limit(3).skip(1);

**UPDATE AND DELETE: updateOne(), updateMany(), deleteOne(), deleteMany()**

We can use updateOne() method for updating single document. We can use updateMany() method for multiple document. When we will use mongo operator. Must use dollar( $ ) before this keywords. Because when we use $ sign before any text. Mongo understand this is mongo keywords. And others text is my text. We use $set operator for updating the document.

**Update single document:** we need to pass two object as parameter. Query and updated object.

Db.products.updateOne({name:’masud’, age:21}, {$set: { name:”masud rana sheikh”}});

db.products.updateOne({id:4}, {$set: { name:"masud rana"}});

db.products.updateOne({\_id: new ObjectId("6542a9688db913a45b98f451")}, {$set: { name:"masud rana sheikh"}});

**update multiple document:** pass a query and then pass a document to update

ecommerce> db.products.updateMany({}, {$set: { name:"masud rana"}});

{

acknowledged: true,

insertedId: null,

matchedCount: 5,

modifiedCount: 0,

upsertedCount: 0

}

**Delete single data:** it delete the single data based on search query

Db.products.deleteMany({}); there is not condition. We can pass condition. Those document will be deleted which will be meet up the query object. We can pass a query object to delete condition based delete the document or records.