CREATE DOCUMENTS IN A GIVEN COLLECTION

db.<*myCollection*>.insert({<*field*> : <*value*>, <*field*> : <*value*>, …})

db.<*myCollection*>.insert([{<*field*> : <*value*>, <*field*> : <*value*>, …},
{<*field*> : <*value*>, <*field*> : <*value*>, …},
…])

db.<*myCollection*>.insertOne({<*field*> : <*value*>, <*field*> : <*value*>, …})

db.<*myCollection*>.insertMany([{<*field*> : <*value*>, <*field*> : <*value*>, …},
{<*field*> : <*value*>, <*field*> : <*value*>, …},
…])

SELECT DOCUMENTS FROM COLLECTION

db.<*myCollection*>.find( )
db.<*myCollection*>.find( ).count( )
db.<*myCollection*>.find( ).pretty( )

db.<*myCollection*>.find({<***conditions***>}, {<***projections***>})

{<***conditions***>} = {<*field*> : <*value*>}
{<*field*> : /<*pattern*>/<*options*>}
{<*field*> : { $<*operator*> : <*value*> } }
$<*operator*> = $gt, $gte, $lt, $lte, $eq, $ne
= $exists (<*value*> can only be 1/true or 0/false)
= $type (<*value*> must be a bson type)
{<*field*> : { $in : [<*value*>, <*value*>, …] } }
{<*field*> : { $nin : [<*value*>, <*value*>, …] } }
{<*field*> : <*value*>, <*field*> : <*value*>, …} (implicit AND)
{ $and : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] } (explicit AND)
{ $or : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] }
{ $expr : { $<*operator*> : ["$<*field*>", "$<*other.field*>"] } } (e.g. x > y)

{<***projections***>} = {<*field*> : 1, <*field*> : 1, …, _id : 0} (_id is independent of the rest (default 1))
{<*field*> : 0, <*field*> : 0, …} (consistently 1 or consistently 0, except for _id)

DELETE DOCUMENTS FROM COLLECTION

db.<*myCollection*>.remove({ })
db.<*myCollection*>.remove({<***conditions***>}, {<***options***>})

{<***conditions***>} = {<*field*> : <*value*>}
{<*field*> : /<*pattern*>/<*options*>}
{<*field*> : { $<*operator*> : <*value*> } }
$<*operator*> = $gt, $gte, $lt, $lte, $eq, $ne
= $exists (<*value*> can only be 1/true or 0/false)
= $type (<*value*> must be a bson type)
{<*field*> : { $in : [<*value*>, <*value*>, …] } }
{<*field*> : { $nin : [<*value*>, <*value*>, …] } }
{<*field*> : <*value*>, <*field*> : <*value*>, …} (implicit AND)
{ $and : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] } (explicit AND)
{ $or : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] }
{ $expr : { $<*operator*> : ["$<*field*>", "$<*other.field*>"] } } (e.g. x > y)

{<***options***>} = {justOne : true} (false by default)

db.<*myCollection*>.deleteOne({<***conditions***>}) (deletes first matching document)
db.<*myCollection*>.deleteMany({<***conditions***>}) (deletes all matching documents)

DELETE COLLECTION

db.<*myCollection*>.drop( )

UPDATE VALUES IN EXISTING DOCUMENTS

(By default, it updates only the first document matching <*conditions*> with the *entire* <*update*> replacing whatever existed before)

db.<*myCollection*>.update({<***conditions***>}, {<***update***>}, {<***options***>})
db.<*myCollection*>.updateOne({<***conditions***>}, {<***update***>}, {<***options***>})
db.<*myCollection*>.updateMany({<***conditions***>}, {<***update***>}, {<***options***>})

{<***conditions***>} = {<*field*> : <*value*>}
{<*field*> : /<*pattern*>/<*options*>}
{<*field*> : { $<*operator*> : <*value*> } }
$<*operator*> = $gt, $gte, $lt, $lte, $eq, $ne
= $exists (<*value*> can only be 1/true or 0/false)
= $type (<*value*> must be a bson type)
{<*field*> : { $in : [<*value*>, <*value*>, …] } }
{<*field*> : { $nin : [<*value*>, <*value*>, …] } }
{<*field*> : <*value*>, <*field*> : <*value*>, …} (implicit AND)
{ $and : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] } (explicit AND)
{ $or : [{<*field*> : <*value*>}, {<*field*> : <*value*>}, …] }
{ $expr : { $<*operator*> : ["$<*field*>", "$<*other.field*>"] } } (e.g. x > y)

{<***update***>} = {<*field*> : <*value*>, <*field*> : <*value*>, …}
{ $<*operator*> : {<*field*> : <*value*>, <*field*> : <*value*>, …} }
$<*operator*> = $set (PARTIAL update, leaves as is whatever is not updated)
$inc
$mul
$rename
$unset (removes a field, <*value*> must be "")

For arrays:
$push (add several values to array at specified location)
$push : {<*field*> : {$each : [<*value*>, <*value*>, …],
$position : <*value*> } }
$push (sort an array)
$push : {<*field*> : {$each : [<*value*>, <*value*>, …],
$sort : -1 } }
$pull (remove several values from array)
$pull : {<*field*> : {$in : [<*value*>, <*value*>, …] } }

{<***options***>} = {multi : true} (false by default, true only valid with $<*operator*> updates)
{upsert : true} (false by default, inserts a row if no document matches <*conditions*>)

AGGREGATIONS OF DOCUMENTS IN A GIVEN COLLECTION

```
db.<myCollection>.aggregate( [ {$group : {
                                    _id : "$continent",
                                    countries : {$sum : 1},
                                    people : {$sum : "$population"}
                                  }
                             } ] )

db.<myCollection>.aggregate( [ {$group : {
                                    _id : {
                                            Company : "$Company",
                                            Date : "$Date"
                                          },
                                    Documents : {$sum : 1},
                                    TotalVolume : {$sum : "$Stock.volume"},
                                    Highest : {$max : "$Stock.high"},
                                    Lowest : {$min : "$Stock.low"}
                                  }
                             } ] )
```