

15_1

```
test> show dbs
admin      8.00 KiB
config    12.00 KiB
local      8.00 KiB

# create db
test> use store
switched to db store

# create collection
store> db.createCollection("products")
{ ok: 1 }

# check creation
store> db.getCollectionNames()
[ 'products' ]

# --terminal-- import data
mongoimport --db store --collection products --file products-data.json --jsonArray

# show data
store> db.products.find()
[
  {
    _id: ObjectId("6406f828958541fcfddad4f6"),
    type: 'Film',
    shipping: { weight: 25, dimensions: { width: 12, height: 21, depth: 2 } },
    pricing: { list: 5200, retail: 10100, savings: 500, pct_savings: 15 },
    details: {
      title: 'InterStellar',
      director: [ 'Christopher Nolan' ],
      writer: [ 'Christopher Nolan', 'Jonathan Nolan' ],
      aspect_ratio: '1.33:1'
    }
  },
  ...
]
```

```
# delete all
store> db.products.deleteMany({})

# add first
store> db.products.insertOne({ ... })
```

```

# add many
store> db.products.insertMany([ { ... } ])

# show
store> db.products.find()

# human readable show
store> db.products.find().pretty()

# show without a specific field
store> db.products.find({ _id: { $exists: false } })

# show a product with specific value for a field
store> db.products.find({ type: "Audio Album" })

# show pricing > retail is less than 5000
store> db.products.find({ "pricing.retail": { $lt: 5000 } })

# show type is not equal film
store> db.products.find ( { 'type': { $ne : "Film" } })

# show shipping > weight is greater than 15
store> db.products.find({ "shipping.weight": { $gt: 15 } })

# find by a field and update
store> db.products.updateOne({ "details.title": "The Matrix" }, { $set: { "pricing.list": 2500 } })

# find by 2 fields
store> db.products.find({ "type": "Film", "shipping.dimensions.depth": 1 })

# count
store> db.products.countDocuments({ "type": "Film" })

# include
store> db.products.find({ "details.writer": { $regex: ".*Jonathan Nolan.*" } })

# ?? sort --> Not work ??
store> db.products.find({ "savings.pricing": { $exists: true } }).sort({ "savings.pricing": -1 }).limit(1)

# include
store> db.products.find({ "details.title": { $regex: ".*x.*" } })

# delete
store> db.collection.deleteOne({ "details.aspect_ratio": "1.66:1" })

```

```
{ acknowledged: true, deletedCount: 0 }
```