-- We (Juhee Lee, Seonhye Hyeon) declare that the attached assignment is our own work in

--accordance with the Seneca Academic Policy.

--No part of this assignment has been copied manually or electronically from any other source

--(including web sites) or distributed to other students.

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

-- Name: Juhee Lee

-- ID: 128261203

-- Name: Seonhye Hyeon

-- ID: 125635193

-- Date: November 29, 2021

-- Purpose: Lab 8 DBS311

-- \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Instructions

1. Switch to database called Lab8. Type use Lab8

Even if it is non-existing, MongoDB will create a database called Lab8. Type again db

The operation should return Lab8

1. Use insert() to insert the following item in the products collection.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| code | desc | indate | qoh | min | price | Colors | Size | Package |
| 11QER/31 | Power painter, 15 psi., 3-nozzle | 2011-11-03 | 100 | 5 | 109.99 | Black, red | l: 24  w:19.99  h:14.99  uom: "cm" | weight: 1.79 uom: "kg" |

db.products.insert({"code":"11QER/31", "desc":"Power painter, 15 psi., 3-nozzle", "indate":new Date("2011-11-03"), "qoh": 100, "min":5, "price": 109.99, "color":["black", "red"], "size":{"l":24, "w":19.99, "h":14.99, "uom":"cm"},"package":{"weight":1.79, "uom":"kg"}})

1. Use findOne() and show the document in the products collection. Use pretty() for formatting.

db.products.findOne().pretty()

1. Use insertMany() to bulk insert the following items in the products collection. When the data is missing for a specific attribute, the document should not contain that attribute as key in its structure.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| code | desc | indate | qoh | min | price | Colors | Size | Package |
| 13-Q2/P2 | 7.25-in. pwr. saw blade | 2011-12-13 | 32 | 15 | 14.99 | - | w: 7.25  h: 0.25  uom: in | - |
| 14-Q1/L3 | 9.00-in. pwr. saw blade | 2011-11-13 | 18 | 12 | 17.49 | Metallic | l: 9  uom: in | weight: 0.6  uom: lbs |
| 1546-QQ2 | Hrd. cloth, 1/4-in., 2x50 | 2012-01-15 | 15 | 8 | 39.95 | silver | l:50  h:2  uom: ft | weight: 19.5  uom: lbs |
| 1558-QW1 | Hrd. cloth, 1/2-in., 4x50 | 2012-01-15 00:00:00.000 | 23 | 5 | 43.99 | - | l:50  h:4  uom: ft | weight: 40  uom: lbs |

db.products.insertMany([

{"code":"13-Q2/P2", "desc":"7.25-in. pwr. saw blade", "indate":new Date("2011-12-13"), "qoh":32, "min":15, "price":14.99, "size":{"l":7.25, "w":0.25, "h":14.99, "uom":"in"}},

{"code":"14-Q1/L3", "desc":"9.00-in. pwr. saw blade", "indate":new Date("2011-11-13"), "qoh":18, "min":12, "price":17.49,"color":"metallic", "size":{"l":9, "uom":"in"},"package":{"weight":0.6, "uom":"lbs"}},

{"code":"1546-QQ2", "desc":"Hrd. cloth, 1/4-in., 2x50", "indate":new Date("2012-01-15"), "qoh":15, "min":8, "price":39.95,"color":"silver", "size":{"l":50, "h":2, "uom":"ft"},"package":{"weight":19.5, "uom":"lbs"}},

{"code":"1558-QW1", "desc":"Hrd. cloth, 1/2-in., 4x50", "indate":new Date("2012-01-15T00:00:00.000"), "qoh":23, "min":5, "price":43.99, "size":{"l":50, "h":4, "uom": "ft"},"package":{"weight":40, "uom":"lbs"}}])

## Tasks

1. Write a query to return desc and *price* for products of color silver.

|  |
| --- |
| db.products.find({"color": "silver"},{"desc":1, "price":1}).pretty() |

1. Write a query to return desc and *price* for products with price between $12 and $20 (Values *12* and *20* are included).

|  |
| --- |
| db.products.find({"price":{$gt:12, $lt:20}},{"desc":1,"price":1}).pretty() |

1. Write a query to return code, desc, *price*, and color for products that are not of color metallic.

|  |
| --- |
| db.products.find({"color":{"$ne":"metallic"}}, {"\_id":0, "code":1, "desc":1, "price":1, "color":1}) |

1. Write a query to return code, desc, *price*, and color for products with color metallic or silver.

|  |
| --- |
| db.products.find({"$or" : [{"color":" metallic"}, {"color":"silver"}]}, {"\_id":0, "code":1, "desc":1, "price":1, "color":1}) |

1. Write a query to return code, desc, *price*, and color for products that do have the color key.

|  |
| --- |
| db.products.find({ color: {$exists: true }}, { "desc": 1, "price": 1, "color": 1 }).pretty() |

1. Write a query to return code, desc, *price*, and color for products that their color is both black and red.

|  |
| --- |
| db.products.find({ $and: [{ color: "black" }, { color: "red"}]}, { desc: 1, price: 1, code: 1 }).pretty() |

**Updating Multiple Documents**

1. Use updateMany() function and the “$set” operator to roll out a new feature “tag” to all products as an array of tags. Consider the value “tools” as a tag element.

|  |
| --- |
| db.products.updateMany({}, {"$set" : {"tag" : ["tools"]}}) |

**Update many using the $push operator.**

1. Add the tag “durable” to the tag array for all products in the products collection.

|  |
| --- |
| db.products.updateMany({},{"$push": {"tag": "durable"}}) |

**Remove elements from an array based on specific criteria.**

1. Pull the black color from the colors array for product 1558-QW1.

|  |
| --- |
| db.products.update({"code":"1558-QW1"},{"$pull":{"Colors" : "black"}}) |

**Add multiple elements and avoid duplicates.**

1. use "$addToSet" in conjunction with "$each" to add the colors black and white to the colors array for product 1558-QW1. This product already has the black color in the list. The operation should not add duplicate values in the colors array.

|  |
| --- |
| db.products.update({"code": "1558-QW1"}, {"$addToSet": {"color": {$each:["black","white"]}}}) |