Assignment: 01

(Group B)

Problem Statement:

**MongoDB Queries:**

Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc)

Implementation

**Execute following queries in MongoDB:**

1. Consider a collection “Inventory” with the following documents:

{ "\_id" : 1, "item" : "f1", type: "food", quantity: 500 }

{ "\_id" : 2, "item" : "f2", type: "food", quantity: 100 }

{ "\_id" : 3, "item" : "p1", type: "paper", quantity: 200 }

{ "\_id" : 4, "item" : "p2", type: "paper", quantity: 150 }

{ "\_id" : 5, "item" : "f3", type: "snacks", quantity: 300 }

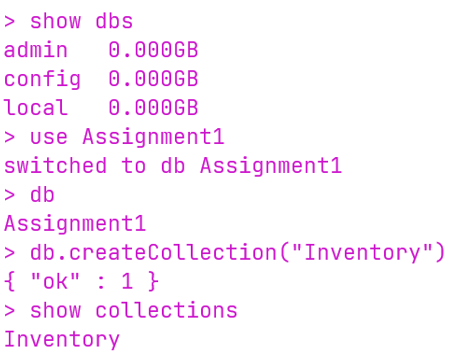
{ "\_id" : 6, "item" : "t1", type: "toys", quantity: 500 }

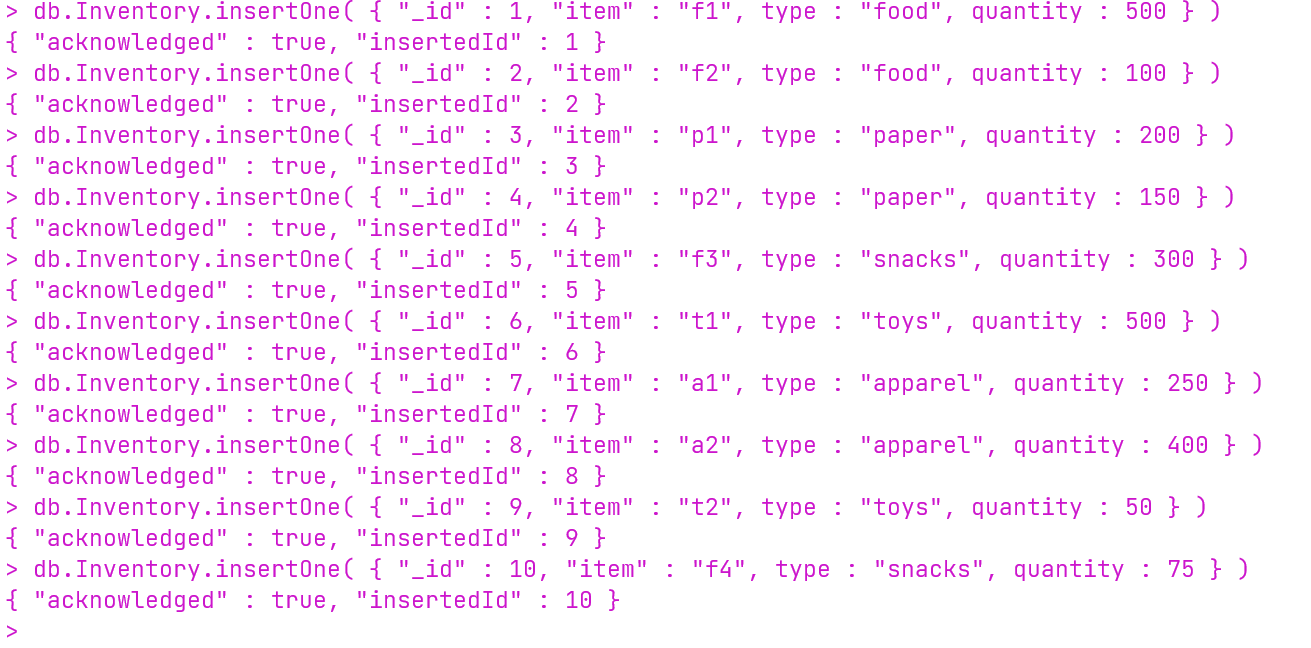
{ "\_id" : 7, "item" : "a1", type: "apparel", quantity: 250 }

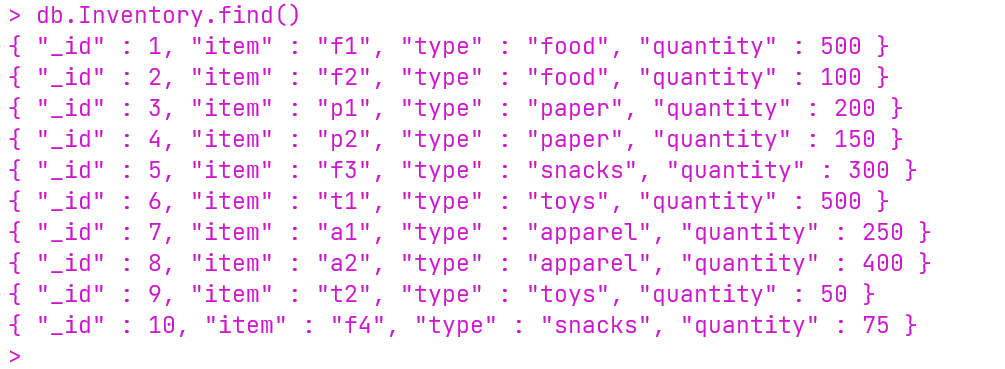
{ "\_id" : 8, "item" : "a2", type: "apparel", quantity: 400 }

{ "\_id" : 9, "item" : "t2", type: "toys", quantity: 50 }

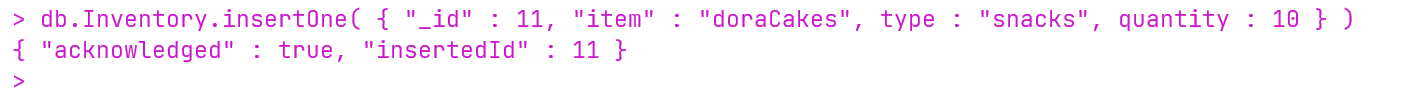
{ "\_id" : 10, "item" : "f4", type: "snacks", quantity: 75 }



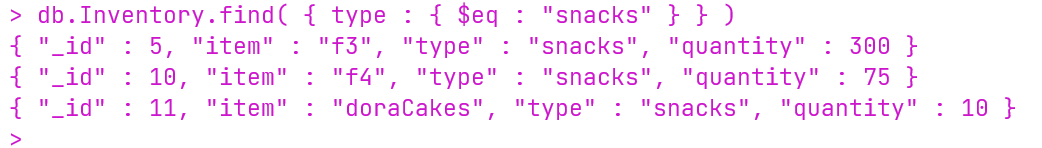




1. Insert a new document into a collection named “Inventory”



1. Find all documents where the type field has the value snacks



1. Insert an Array of Documents

var mydocuments =

[

{

item: "ABC2",

details: { model: "14Q3", manufacturer: "M1 Corporation" },

stock: [ { size: "M", qty: 50 } ],

category: "clothing"

},

{

item: "MNO2",

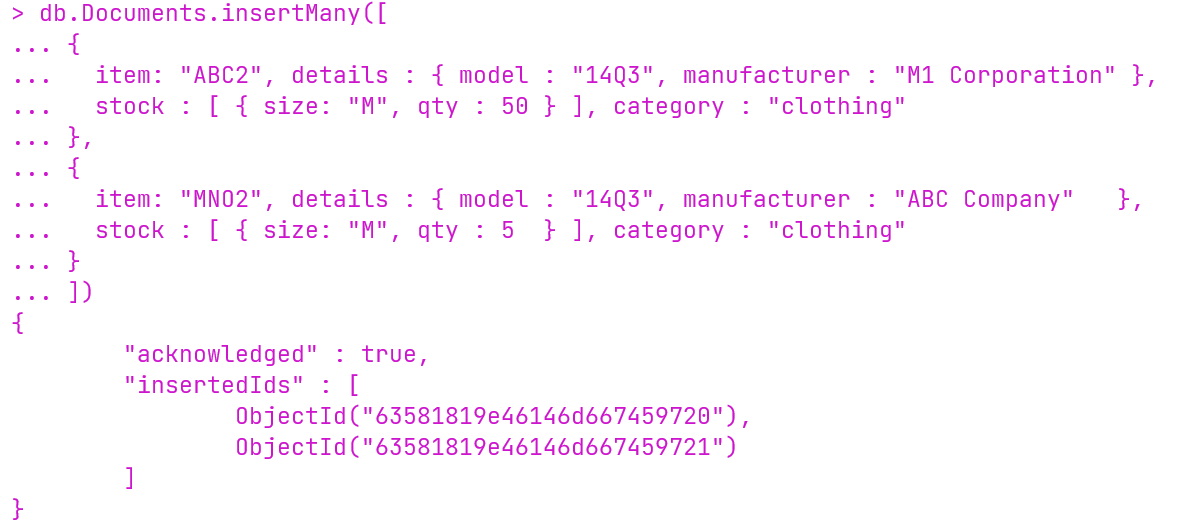
details: { model: "14Q3", manufacturer: "ABC Company" },

stock: [ { size: "S", qty: 5 }, { size: "M", qty: 5 }, { size: "L", qty: 1 } ],

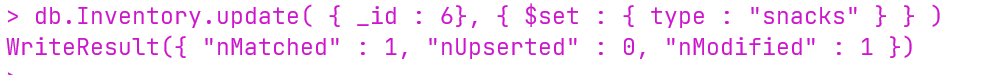
category: "clothing"

}

]



1. Update a document

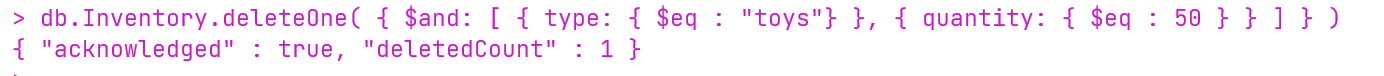


1. Update Specific Fields

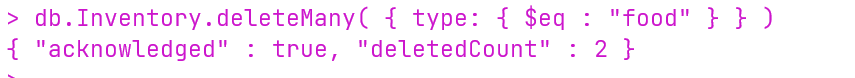
**For the document with item equal to "MNO2", update the category field and the details field**.

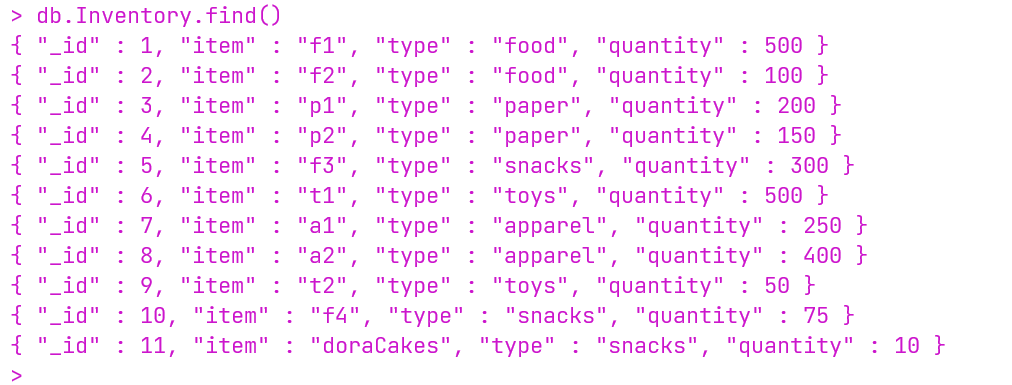


1. Remove a single document where type is toys and qty is 50

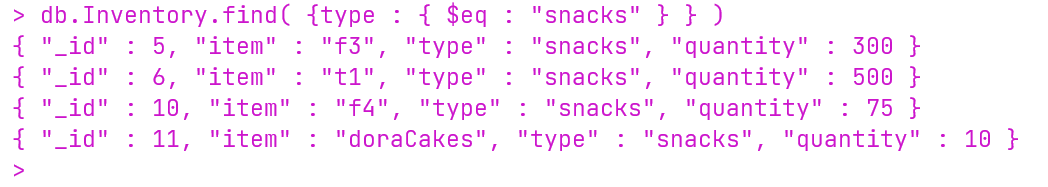


1. Remove all documents where type is “food”



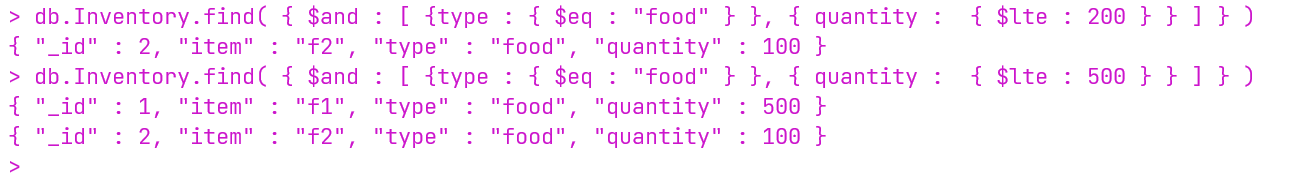
1. Select all documents in a Collection
2. 
3. Specify Equality Condition

**Find all documents where the type field has the value snacks**

****

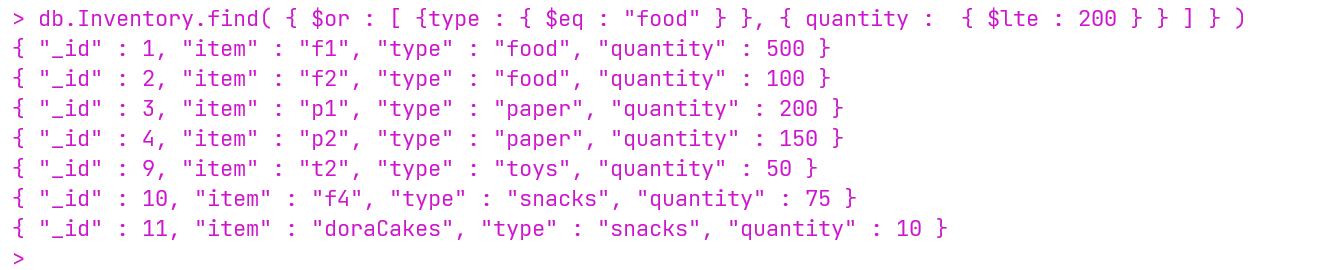
1. Specify AND Conditions

**All documents where the type field has the value 'food' and the quantity of the price field is less than 200**



1. Specify OR Conditions

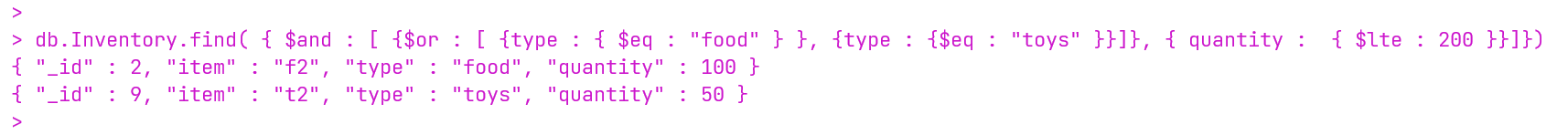
**All documents in the collection where the field qty has a value greater than 200 or the name is “food”**



1. Specify AND as well as OR Conditions

**Find all documents in the collection where the value of the type field is 'food' and either the qty has a less than 200 or the type is either “food” or**

“toys”.



1. Exact Match on the Embedded Document

**Find all documents where the value of the field producer is an embedded document that contains only the field company with the value 'ABC123' and the field address with the value '123 Street', in the exact order**

* **Data Insufficient, not provided**

1. Exact Match on an Array

**Find all documents where the field ratings is an array that holds exactly three elements, 5, 8, and 9**

* **Data Insufficient, not provided**

1. Match an Array Element

**Find all documents where ratings is an array that contains 5 as one of its elements**

* **Data Insufficient, not provided**

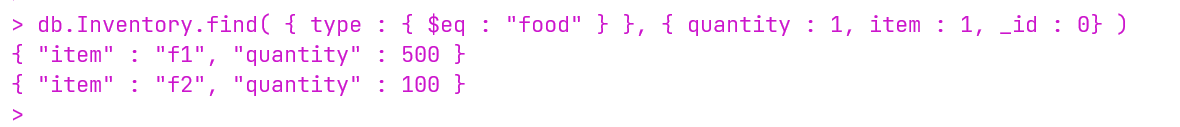
1. Match a Specific Element of an Array

**Find all documents where the ratings array contains 5 as the first element**

* **Data Insufficient, not provided**

1. Return Specified Fields Only

**Return food only with item and qty fields from the documents.**



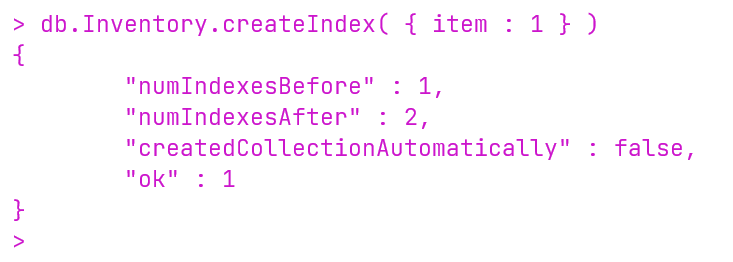
1. Sort documents using one field

**Sort documents on qty in descending order**



1. Create Index on some field

**Create Index on item field**

****