Course code	: **CSE3009**
Course title	: **No SQL Data Bases**
Module	: **4**
Topic	: **2**

Querying

Objectives

This session will give the knowledge about

- Querying

Querying

Insert a Single Document:

db.collection.insertOne() inserts a single document into a collection.

The following example inserts a new document into the inventory collection. If the document does not specify an _id field, MongoDB adds the _id field with an ObjectId value to the new document.

```
db.inventory.insertOne(
   { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }
)
```

Querying

Insert Multiple Documents

db.collection.insertMany() can insert multiple documents into a collection. Pass an array of documents to the method.

The following example inserts three new documents into the inventory collection. If the documents do not specify an _id field, MongoDB adds the _id field with an ObjectId value to each document.

```
db.inventory.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], size: { h: 14, w: 21, uom: "cm" } },
   { item: "mat", qty: 85, tags: ["gray"], size: { h: 27.9, w: 35.5, uom: "cm" } },
   { item: "mousepad", qty: 25, tags: ["gel", "blue"], size: { h: 19, w: 22.85, uom: "cm" } }
])
```

Querying

Additional Methods for Inserts

- db.collection.update()
- db.collection.updateOne()
- db.collection.updateMany()
- db.collection.findAndModify()
- db.collection.findOneAndUpdate()
- db.collection.findOneAndReplace()
- db.collection.save().
- db.collection.bulkWrite().

Querying

```
db.inventory.insertMany([
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

Querying

Select All Documents in a Collection

To select all documents in the collection, pass an empty document as the query filter parameter to the find method. The query filter parameter determines the select criteria:

```
db.inventory.find( {} )
```

Specify Equality Condition

The following example selects from the inventory collection all documents where the status equals "D":

```
db.inventory.find( { status: "D" } )
```

Querying

Specify Conditions Using Query Operators

The following example retrieves all documents from the inventory collection where status equals either "A" or "D":

```
db.inventory.find( { status: { $in: [ "A", "D" ] } } )
```

Specify AND Conditions

The following example retrieves all documents in the inventory collection where the status equals "A" and qty is less than ($lt) 30:

```
db.inventory.find( { status: "A", qty: { $lt: 30 } } )
```

Querying

Specify OR Conditions

The following example retrieves all documents in the collection where the status equals "A" or qty is less than ($lt) 30:

```
db.inventory.find( { $or: [ { status: "A" }, { qty: { $lt: 30 } } ] } )
```

Specify AND as well as OR Conditions

In the following example, the compound query document selects all documents in the collection where the status equals "A" and either qty is less than ($lt) 30 or item starts with the character p:

```
db.inventory.find( { status: "D" } )
```

Querying

```
db.inventory.insertMany( [
   { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
   { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
   { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
   { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

Querying

Match an Embedded/Nested Document

To specify an equality condition on a field that is an embedded/nested document, use the query filter document { <field>: <value> } where <value> is the document to match.

For example, the following query selects all documents where the field size equals the document { h: 14, w: 21, uom: "cm" }:

```
db.inventory.find( { size: { h: 14, w: 21, uom: "cm" } } )
```

Querying

Query on Nested Field

The following example selects all documents where the field uom nested in the size field equals "in":

```
db.inventory.find( { "size.uom": "in" } )
```

Specify Match using Query Operator

The following query uses the less than operator ($lt) on the field h embedded in the size field:

```
db.inventory.find( { "size.h": { $lt: 15 } } )
```

Querying

Specify AND Condition

The following query selects all documents where the nested field h is less than 15, the nested field uom equals "in", and the status field equals "D":

```
db.inventory.find( { "size.h": { $lt: 15 }, "size.uom": "in", status: "D" } )
```

Querying

```
db.inventory.insertMany([
   { item: "journal", qty: 25, tags: ["blank", "red"], dim_cm: [ 14, 21 ] },
   { item: "notebook", qty: 50, tags: ["red", "blank"], dim_cm: [ 14, 21 ] },
   { item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim_cm: [ 14, 21 ] },
   { item: "planner", qty: 75, tags: ["blank", "red"], dim_cm: [ 22.85, 30 ] },
   { item: "postcard", qty: 45, tags: ["blue"], dim_cm: [ 10, 15.25 ] }
]);
```

Querying

Match an Array

The following example queries for all documents where the field tags value is an array with exactly two elements, "red" and "blank", in the specified order:

```
db.inventory.find( { tags: ["red", "blank"] } )
```

If, instead, you wish to find an array that contains both the elements "red" and "blank", without regard to order or other elements in the array, use the $all operator:

```
db.inventory.find( { tags: { $all: ["red", "blank"] } } )
```

Querying

Query an Array for an Element

The following example queries for all documents where tags is an array that contains the string "red" as one of its elements:

```
db.inventory.find( { tags: "red" } )
```

The following operation queries for all documents where the array dim_cm contains at least one element whose value is greater than 25.

```
db.inventory.find( { dim_cm: { $gt: 25 } } )
```

Querying

Specify Multiple Conditions for Array Elements

The following example queries for documents where the dim_cm array contains elements that in some combination satisfy the query conditions; e.g., one element can satisfy the greater than 15 condition and another element can satisfy the less than 20 condition, or a single element can satisfy both:

```
db.inventory.find( { dim_cm: { $gt: 15, $lt: 20 } } )
```

The following example queries for documents where the dim_cm array contains at least one element that is both greater than ($gt) 22 and less than ($lt) 30:

```
db.inventory.find( { dim_cm: { $elemMatch: { $gt: 22, $lt: 30 } } } )
```

Querying

Query for an Element by the Array Index Position

The following example queries for all documents where the second element in the array dim_cm is greater than 25:

```
db.inventory.find( { "dim_cm.1": { $gt: 25 } } )
```

Query an Array by Array Length

The following selects documents where the array tags has 3 elements.

```
db.inventory.find( { "tags": { $size: 3 } } )
```

Querying

```
db.inventory.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },
   { item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },
   { item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },
   { item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
]);
```

Querying

Query for a Document Nested in an Array

The following example selects all documents where an element in the instock array matches the specified document:

```
db.inventory.find( { "instock": { warehouse: "A", qty: 5 } } )
```

The following query does not match any documents in the inventory collection:

```
db.inventory.find( { "instock": { qty: 5, warehouse: "A" } } )
```

Querying

Specify a Query Condition on a Field Embedded in an Array of Documents

If you do not know the index position of the document nested in the array, concatenate the name of the array field, with a dot (.) and the name of the field in the nested document.

The following example selects all documents where the instock array has at least one embedded document that contains the field qty whose value is less than or equal to 20:

```
db.inventory.find( { 'instock.qty': { $lte: 20 } } )
```

Querying

Use the Array Index to Query for a Field in the Embedded Document

The following example selects all documents where the instock array has as its first element a document that contains the field qty whose value is less than or equal to 20:

```
db.inventory.find( { 'instock.0.qty': { $lte: 20 } } )
```

NOTE

When querying using dot notation, the field and index must be inside quotation marks.

Querying

A Single Nested Document Meets Multiple Query Conditions on Nested Fields

The following example queries for documents where the instock array has at least one embedded document that contains both the field qty equal to 5 and the field warehouse equal to A:

```
db.inventory.find( { "instock": { $elemMatch: { qty: 5, warehouse: "A" } } } )
```

The following example queries for documents where the instock array has at least one embedded document that contains the field qty that is greater than 10 and less than or equal to 20:

```
db.inventory.find( { "instock": { $elemMatch: { qty: { $gt: 10, $lte: 20 } } } } )
```

Querying

Query for Null or Missing Fields

Equality Filter

The { item : null } query matches documents that either contain the item field whose value is null or that do not contain the item field.

```
db.inventory.find( { item: null } )
```

Type Check

The { item : { $type: 10 } } query matches only documents that contain the item field whose value is null; i.e. the value of the item field is of BSON Type Null (type number 10) :

```
db.inventory.find( { item : { $type: 10 } } )
```

Querying

Existence Check

The following example queries for documents that do not contain a field.

The { item : { $exists: false } } query matches documents that do not contain the item field:

```
db.inventory.find( { item : { $exists: false } } )
```

Summary

This session will give the knowledge about

- Querying
- References:
  - https://docs.mongodb.com/manual/tutorial/
  - “Seven NoSQL Databases in a Week” Author: Aaron Ploetz Aaron