**Week 1 – MONGO Database(No SQL)**

| **Problem Statement** | Use MongoDB and create a sample database and perform the following operations on it  1. Create documents with varying fields.  2. Use the insert and save commands.  3. Drop an existing database and a collection and use remove() to delete documents.  4. Update an existing document in a collection. (update() ) 5. Use find() and findOne() for querying the database. |
| --- | --- |

**MongoDB CRUD Operations**

CRUD operations create, read, update, and delete .

**1.Create Operations**

Create or insert operations add new documents to a collection. If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

**db.collection.insertOne()**

**db.collection.insertMany()**

ex: 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

**Insert one :**

**db.inventory.insertOne(**

**{ item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }**

**)**

**Insert many :**

**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" } }])**

**2. Read Operations**

Read operations retrieve documents from a collection; i.e. query a collection for documents. MongoDB provides the following methods to read documents from a collection:

**db.collection.find()**

**example :**

**The following query uses $gt to return documents where the value of qty is greater than 4.**

**db.collection.find( { qty: { $gt: 4 } } )**

**Try using findone()**

You can specify query filters or criteria that identify the documents to return.

**3.Update Operations**

Update operations modify existing documents in a collection. MongoDB provides the following methods to update documents of a collection:

**db.collection.updateOne() New in version 3.2**

**db.collection.updateMany() New in version 3.2**

**db.collection.replaceOne() New in version 3.2**

**The updateOne() method has the following syntax:**

**db.collection.updateOne(**

**<filter>,**

**<update>,**

**{**

**upsert: <boolean>,**

**writeConcern: <document>,**

**collation: <document>,**

**arrayFilters: [ <filterdocument1>, ... ],**

**}**

**)**

| **Parameter** | **Type** | **Description** |
| --- | --- | --- |
| [filter](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-update-one-filter) | document | The selection criteria for the update. The same [query selectors](https://docs.mongodb.com/manual/reference/operator/query/#std-label-query-selectors) as in the [find()](https://docs.mongodb.com/manual/reference/method/db.collection.find/#mongodb-method-db.collection.find) method are available.  Specify an empty document { } to update the first document returned in the collection. |
| [update](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-update-one-update) | document or pipeline | The modifications to apply. Can be one of the following:   |  |  | | --- | --- | | [Update document](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-update-one-method-update-document) | Contains only [update operator expressions](https://docs.mongodb.com/manual/reference/operator/update/#std-label-update-operators).  For more information, see [Update with an Update Operator Expressions Document](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-updateOne-behavior-update-expressions) | | [Aggregation pipeline](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-update-one-method-agg-pipeline) (*Starting in MongoDB 4.2*) | Contains only the following aggregation stages:   * [$addFields](https://docs.mongodb.com/manual/reference/operator/aggregation/addFields/#mongodb-pipeline-pipe.-addFields) and its alias [$set](https://docs.mongodb.com/manual/reference/operator/aggregation/set/#mongodb-pipeline-pipe.-set) * [$project](https://docs.mongodb.com/manual/reference/operator/aggregation/project/#mongodb-pipeline-pipe.-project) and its alias [$unset](https://docs.mongodb.com/manual/reference/operator/aggregation/unset/#mongodb-pipeline-pipe.-unset) * [$replaceRoot](https://docs.mongodb.com/manual/reference/operator/aggregation/replaceRoot/#mongodb-pipeline-pipe.-replaceRoot) and its alias [$replaceWith](https://docs.mongodb.com/manual/reference/operator/aggregation/replaceWith/#mongodb-pipeline-pipe.-replaceWith).   For more information, see [Update with an Aggregation Pipeline](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/#std-label-updateOne-behavior-aggregation-pipeline). |   To update with a replacement document, see [db.collection.replaceOne()](https://docs.mongodb.com/manual/reference/method/db.collection.replaceOne/" \l "mongodb-method-db.collection.replaceOne). |
| upsert | boolean | Optional. When true, [updateOne()](https://docs.mongodb.com/manual/reference/method/db.collection.updateOne/" \l "mongodb-method-db.collection.updateOne) either:   * Creates a new document if no documents match the filter. For more details see [upsert behavior](https://docs.mongodb.com/manual/reference/method/db.collection.update/" \l "std-label-upsert-behavior). * Updates a single document that matches the filter.   To avoid multiple [upserts](https://docs.mongodb.com/manual/reference/glossary/" \l "std-term-upsert), ensure that the filter field(s) are [uniquely indexed](https://docs.mongodb.com/manual/core/index-unique/#std-label-index-type-unique).  Defaults to false, which does *not* insert a new document when no match is found. |

Ex:

db.people.update(

{ name: "Andy" },

{

name: "Andy",

rating: 1,

score: 1

},

{ upsert: true }

)

In MongoDB, update operations target a single collection. All write operations in MongoDB are atomic on the level of a single document.

**4.Delete Operations**

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

**db.collection.deleteOne() New in version 3.2**

**db.collection.deleteMany() New in version 3.2**

**ex:**

**db.inventory.deleteMany({ status : "A" })**

**try deleteone()**