Nested/Embedded Documents

Nested/Embedded Documents

- It contain a document inside another document.

- When a collection has a document, this document contains another document, another document contains another sub-document, and so on, then such types of documents are known as embedded or nested documents.

```
> db.Tools.insertMany([
... {module: "JavaScript",
... time_days: {beginners: 15, moderate: 10, advanced: 10},
... tags: "Frontend Development", _id: "DC-001"
... },
... {module:"React-Redux",
... time_days: {beginners: 12, moderate: 4, advanced: 4},
... tags: "Frontend Development", _id: "DC-002"
... },
... {module: "MySQL",
... time_days: {beginners: 8, moderate: 4, advanced: 3},
... tags: "Backend Development", _id: "DC-003"
... },
... {module: "MongoDB",
... time_days: {beginners: 7, moderate: 5, advanced: 3},
... tags: "Backend Development", _id: "DC-004"
... },
... {module: "Server-API-git",
... time_days: {beginners: 12, moderate: 4, advanced: 4},
... tags: "Backend Development", _id: "DC-005"
... }
... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                "DC-001",
                "DC-002",
                "DC-003",
                "DC-004",
                "DC-005"
        ]
}
```

Query on Nested Documents

- An embedded document can be accessed by using the .find() method and using dot notation in our query filter to access the embedded field.

- **Syntax:** "field.nestedField": value

- Example: db.inventory.insertMany([
  { item: "journal", instock: [{ warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 }] },
  { item: "notebook", instock: [{ warehouse: "C", qty: 5 }] }
]);

Query for a Document Nested in an Array:

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

Output:

```
[
  {
    _id: ObjectId('65b9047e73af0e04477a021e'),
    item: 'journal',
    instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
  }
]
```

- Equality matches on the whole embedded/nested document require an *exact* match of the specified document, including the field order. For example, the following query does not match any documents in the inventory collection:

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

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

Example: db.First_Collection.insertMany([

{ item: "journal", instock: [{ warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 }] },

{ item: "notebook", instock: [{ warehouse: "C", qty: 5 }] },

]);

```
db.First_Collection.find({"instock.qty":{$lte: 10}})
```

- This query 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 10.

```
{
  _id: ObjectId('65b9047e73af0e04477a021e'),
  item: 'journal',
  instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
},
{
  _id: ObjectId('65b9047e73af0e04477a021f'),
  item: 'notebook',
  instock: [ { warehouse: 'C', qty: 5 } ]
}
```

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

- Using dot notation, you can specify query conditions for field in a document at a particular index or position of the array. The array uses zero-based indexing.
- When querying using dot notation, the field and index must be inside quotatic

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

- This query 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.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 } ] },
```

```
db.First_Collection.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 },
    { warehouse: "C", qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },
]);
```

- Query 1: `db.First_Collection.find({"instock.0.qty":{$lte: 10}})`
- Query 2: `db.First_Collection.find({"instock.1.qty":{$gte: 10}})`

```
db.First_Collection.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C",
     qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },
]);
```

- Query 1:

```
db.First_Collection.find({"instock.0.qty":{$lte: 10}})
```

- Output:

```
{
  _id: ObjectId('65b9047e73af0e04477a021e'),
  item: 'journal',
  instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
},
{
  _id: ObjectId('65b9047e73af0e04477a021f'),
  item: 'notebook',
  instock: [ { warehouse: 'C', qty: 5 } ]
}
```

- Query 2:

```
db.First_Collection.find({"instock.1.qty":{$gte: 10}})
```

- Output:

```
{
  _id: ObjectId('65b9047e73af0e04477a021e'),
  item: 'journal',
  instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
}
```

Specify Multiple Conditions for an Array of Documents

- When specifying conditions on more than one field nested in an array of documents, you can specify the query such that either a single document meets these condition or any combination of documents (including a single document) in the array meets the conditions.

❑ **A Single Nested Document Meets Multiple Query Conditions on Nested Fields**

- Use $elemMatch operator to specify multiple criteria on an array of embedded documents such that at least one embedded document satisfies all the specified criteria.

```
db.inventory.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] }
]);
```

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

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

```
db.inventory.insertMany( [
   { item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C",
     qty: 15 } ] },
   { item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] }
]);
```

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

- This 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.

Output:

```
{
  _id: ObjectId('65b9047e73af0e04477a021e'),
  item: 'journal',
  instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
}
```

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

- This 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.

Output:

```
{
  _id: ObjectId('65b9047e73af0e04477a021e'),
  item: 'journal',
  instock: [ { warehouse: 'A', qty: 5 }, { warehouse: 'c', qty: 15 } ]
}
```

❑ Combination of Elements Satisfies the Criteria

- If the compound query conditions on an array field do not use the $elemMatch operator, the query selects those documents whose array contains any combination of elements that satisfies the conditions.
- Example:

```
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 } ] }
]);
```

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

- This query matches documents where any document nested in the *instock* array has the *qty* field greater than 10 and any document (but not necessarily the same embedded document) in the array has the qty field less than or equal to 20.

```
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 } ] }
]);
```

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

- This example queries for documents where the *instock* array has at least one embedded document that contains the field *qty* equal to 5 and at least one embedded document (but not necessarily the same embedded document) that contains the field warehouse equal to A.