CSCU9YQ - NoSQL Databases
Lecture 8: Cursors and Projections in MongoDB

Gabriela Ochoa

Cursors

- The db.collection.find() method returns a cursor.
- A cursor is a ‘pointer’ to the results.
- To access the documents, you need to iterate the cursor.

Cursors

- If the returned cursor is not assigned to a variable using the var keyword, then the cursor is automatically iterated up to 20 times (print up to the first 20 documents in the result)
- When assigned to a a variable, to access the whole set of results, you need to
  - Iterate the cursor manually
  - Use the iterator index.

Manually Iterate the Cursor

```
var myCursor = db.users.find( { type: 2 } );

myCursor
```

- If you call the cursor variable in the shell, it will iterate up to 20 times and print the matching
- The shell will say: type ‘it’ for more

```
var myCursor = db.users.find( { type: 2 } );

while (myCursor.hasNext()) {
   print(tojson(myCursor.next()));
}
```

- Use the cursor method next() to access all the documents
- Print results with tojson()

```
var myCursor = db.users.find( { type: 2 } );

while (myCursor.hasNext()) {
   printjson(myCursor.next());
}
```

- As an alternative print operation, use printjson() helper method to replace print(tojson())

Iterator Index

```
var myCursor = db.inventory.find( { type: 2 } );
var documentArray = myCursor.toArray();
var myDocument = documentArray[3];
```

- You can use the toArray() method to iterate the cursor and return the documents in an array
- The toArray() method
    - Loads into RAM all documents returned by the cursor
    - Exhausts the cursor.

Projections

- By default, queries in MongoDB return all fields in matching documents.
- To limit the amount of data, you can include a projection document to specify or restrict fields to return.

https://docs.mongodb.com/manual/tutorial/project-fields-from-query-results/

Example: Inventory data

```
db.inventory.insertMany( [
  { item: "journal", status: "A", size: { h: 14, w: 21, uom: "cm" },
   instock: [ { warehouse: "A", qty: 5 } ] },
  { item: "notebook", status: "A",  size: { h: 8.5, w: 11, uom: "in" },
    instock: [ { warehouse: "C", qty: 5 } ] },
  { item: "paper", status: "D", size: { h: 8.5, w: 11, uom: "in" },
   instock: [ { warehouse: "A", qty: 60 } ] },
  { item: "planner", status: "D", size: { h: 22.85, w: 30, uom: "cm" },
    instock: [ { warehouse: "A", qty: 40 } ] },
  { item: "postcard", status: "A", size: { h: 10, w: 15.25, uom: "cm" },
    instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }
]);
```

- If you do not specify a projection document, the db.collection.find() method returns all fields in the matching documents.
- db.inventory.find({ status: "A" })
  - Returns all fields from all documents where the status equals "A":

Return the Specified Fields

- A projection can explicitly include several fields by setting the <field> to 1 in the projection document.
- The _id field is shown by default
- db.inventory.find({ status: "A" }, { item: 1, status: 1 })

```
{"_id": ObjectID("6shssgahs..."),  item: "journal", status: "A”}
{"_id": ObjectID("3sfhf7jnsh..."),  item: "notebook", status: "A”},
{"_id": ObjectID(”8sasgahyv..."),  item: "postcard", status: "A” }
```

- You van supress the _id field by setting it to 0 in the projection
- db.inventory.find({ status: "A" }, { item: 1, status: 1, _id: 0 })

```
{item: "journal", status: "A”}
{item: "notebook", status: "A”}
{item: "postcard", status: "A” }
```

Return all but the Excluded Fields

- You can use a projection to exclude specific fields
- Notice that, with the exception of the _id field, you cannot combine inclusion and exclusion statements in projection documents.
- db.inventory.find({ status: "A" }, { status: 0, instock: 0 })

```
{"_id": ObjectID("6shssgahs..."),  item: "journal", size: { h: 14, w: 21, uom: "cm" }},
{"_id": ObjectID("3sfhf7jnsh..."),  item: "notebook”,  size: { h: 8.5, w: 11, uom: "in" }},
{"_id": ObjectID(”8sasgahyv..."),  item: "postcard”, size: { h: 10, w: 15.25, uom: "cm" }}
```

Return Specific Fields in Embedded Documents

- You can return specific fields in an embedded document.
- Use the dot notation to refer to the embedded field and set to 1 in the projection document.

```
db.inventory.find( { status: "A" }, { item: 1, status: 1, "size.uom": 1, _id: 0 }
```

```
{item: "journal", status: : "A", size: { uom: "cm" }}
{item: "notebook", status: "A”, size: { uom: "cm" }}
{item: "postcard", status: "A”, size: { uom: "cm" }}
```

The uom field remains embedded in the size document.

Supress Specific Fields in Embedded Documents

- You can return specific fields in an embedded document.
- Use the dot notation to refer to the embedded field and set to 0 in the projection document.

```
db.inventory.find( { status: "A" }, {"size.uom": 0, _id: 0 })
```

```
{item: "journal", status: "A", size: { h: 14, w: 21 },
 instock: [ { warehouse: "A", qty: 5 } ] },
{item: "notebook", status: "A",  size: { h: 8.5, w: 11},
 instock: [ { warehouse: "C", qty: 5 } ] },
{ item: "postcard", status: "A", size: { h: 10, w: 15.25}
 instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 }
```

Projection on Embedded Documents in an Array

- Use dot notation to project specific fields inside documents embedded in an array (use " ")

```
db.inventory.find( { status: "A" }, {item: 1, status: 1, "instock.qty": 1, _id: 0 })
```

```
{item: "journal", status: "A", instock: [{ warehouse: "A", qty: 5 } ] }
{item: "notebook", status: "A", instock: [ { qty: 5 } ] }
{ item: "postcard", status: "A”, instock: [{qty: 15 }, {qty: 35 } ]}
```

Project Specific Array Elements in the Returned Array

- MongoDB provides the following projection operators for manipulating arrays:
  - $elemMatch: limits the contents of an <array> field from the query results to contain only the first element matching the $elemMatch condition.
  - $slice: controls the number of items of an array that a query returns.
  - $: limits the contents of an <array> from the query results to contain only the first element matching the query document.
- You cannot project specific array elements using the array index; e.g. { "instock.0": 1 }

Project Specific Array Elements in the Returned Array

- Example using the $slice projection operator to return the last element in the instock array

```
db.inventory.find( { status: "A" }, {_id:0, item: 1, status: 1, instock: {$slice: -1}})
```

```
{item: "journal", status: "A", instock: [{ qty: 5 } ] }
{item: "notebook", status: "A", instock: { warehouse: "C", qty: 5 } ] },
{item: "postcard", status: "A”, instock: [{ warehouse: "C", qty: 35 } ]}
```

Summary

- Cursors
  - The db.collection.find() method returns a cursor.
  - To access the whole set of results, you need to
    - Iterate the cursor manually
    - Use the iterator index.
- Projections
  - Specify or restrict fields to return from a query
  - Return or suppress specified fields
  - Return or supress specific fields in embedded documents
  - Project embedded documents in an array
  - Project specific array elements in the returned array