MongoDB
Cursors, Limit() and Sort()

Prepared For: CS527

Prepared By: Josh Levine, Pranav Shivkumar, Pratik Mistry, Shounak Rangwala, Swapnil Kamate, Vikhyat Dhamija

Cursors

- It is a MongoDB Collection of the document which is returned upon the find() method execution

- Accessing records from cursor:
  - Can be automatically executed in loop using iterator
  - Explicitly specify the index you would like cursor to point to

- It is like a pointer which is pointing upon a specific index value

- Hence, when we call the find method, all the documents which are returned are saved in a virtual cursor

Cursors: Example (1)

- Sample Collection - fruits:
  {{ "_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 },
  { "_id": "oranges", "qty": { "in stock": 8, "ordered": 12 } },
  { "_id": "avocados", "qty": "fourteen" },
  { "_id": "mangoes", "qty": 9 }}

- var filtered_fruits = ***db.fruits.find({ qty: { $gt: 4 } })***

- Cursor (pointer) to the filtered documents is returned.

- filtered _fruits “points” to below result set:
  {"_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 }
  { "_id": "mangoes", "qty": 9 }

Cursors: Example (2)

- The document with _id equal to "avocados" is not returned because it’s qty value is of type string while the $gt operand is of type integer

- Accessing Records using loop:
  while(filtered_fruits.hasNext())
  { print(tojson(filtered_fruits.next())); }

  Result in JSON readable format:
  {"_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 },
  { "_id": "mangoes", "qty": 9 }

- Accessing Records by indexing:
  print(toJSON(filtered_fruits[0]))

  Result in JSON readable format:
  {"_id": "apples", "qty": 5 }

- Advanced flexibility when executing queries with cursors can be achieved by using count(), limit(), pretty(), etc. methods appended after the find() function

Limit

- Mongo DB provides query modifiers such as the “limit” clause to provide more flexibility when executing queries
- It is a function in MongoDB that is used to specify the maximum number of results to be returned
- limit() can be used with find() which will return a cursor(pointer) to the maximum specified records
- Example:
  db.collection.find() returns all the documents from collection
- To limit the number of records:
  db.collection.find().limit(*number*) return only <*number*> of documents from collection

Limit: Example

- Sample Collection - fruits:
  {{ "_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 },
  { "_id": "oranges", "qty": { "in stock": 8, "ordered": 12 } },
  { "_id": "avocados", "qty": "fourteen" },
  { "_id": "mangoes", "qty": 9 }}

- var filtered_fruits = ***db.fruits.find({ qty: { $gt: 4 } }).limit(2)***

- filtered _fruits “points” to below result set:
  {"_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 }

Sort

- Mongo DB provides query modifiers such as the “sort” clause to provide more flexibility when executing queries

- It is a function in MongoDB that is used to specify the order of documents to be returned based on ascending or descending order of any key in the collection

- sort() can be used with find() which will return cursor(pointer) to the ordered (ascending/descending) records

- Example:
  db.collection.find() returns all the documents from collection

- To sort the number of records:
  db.collection.find().sort({*key: 1 or -1*}) returns only documents from collection in ascending (1) or descending (-1) order against the value of the key specified

Sort: Example

- Sample Collection - fruits:
  {{ "_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 },
  { "_id": "oranges", "qty": { "in stock": 8, "ordered": 12 } },
  { "_id": "avocados", "qty": "fourteen" },
  { "_id": "mangoes", "qty": 9 }}

- Sorted Documents - Ascending
  var filtered_fruits = db.fruits.find({ qty: { $gt: 4 } }).sort({qty : 1})

  filtered _fruits “points” to below result set:
  {"_id": "apples", "qty": 5 },
  { "_id": "bananas", "qty": 7 },
  { "_id": "mangoes", "qty": 9 }

- Sorted Documents - Descending
  var filtered_fruits = db.fruits.find({ qty: { $gt: 4 } }).sort({qty : -1})

  filtered _fruits “points” to below result set:
  { "_id": "mangoes", "qty": 9 },
  { "_id": "bananas", "qty": 7 },
  {"_id": "apples", "qty": 5 }

Thank You