Cloud Data Management Project

MongoDB

Team: Tyler Bounds, Steve Brown, Chris Hight

Source Code Examples

QUERY #1:

```
// Finds all speeds greater than 100 and sums the number returned
var results = db.freeway_loopdata.find( { speed : { $gt : 100 } } ).count();

// Print the results
print("Number of speeds > 100 MPH: " + results);
```

QUERY #6:

```
// The station we're going to
var toStation = db.freeway_stations.find( { locationtext : "Columbia to I-205 NB" } ).toArray();

// The station we're going from
var fromStation = db.freeway_stations.find( { locationtext: "Johnson Cr NB" } ).toArray();

// Initialize variable to hold route
var route = "";

// Loop through and find all downstream stationids until you find the station you're going to
for(var i = fromStation[0].stationid; i != toStation[0].stationid;){

        var cursor = db.freeway_stations.find({ stationid : i});

        if(cursor.hasNext()){
                route += cursor[0].locationtext + "\n";
                i = cursor[0].downstream;
        }
}
route += toStation[0].locationtext;

// Prints out the final route by locationtext
print(route);
```

	Returned Result	Execution Times (ms)	Number of Rows Matched
Query #1	6972	6853	6972
Query #2	{ " id" : null, "summedVolume" : 56130 }	7065	12960
Query #3	(1 hour out of 24 shown) { "_id" : { "hour" : 23, "5_minute_intervals" : 55 }, "result_in_seconds" : 99.57741067998464 } { "_id" : { "hour" : 23, "5_minute_intervals" : 50 }, "result_in_seconds" : 99.22887612797375 } { "_id" : { "hour" : 23, "5_minute_intervals" : 45 }, "result_in_seconds" : 100.99893730074389 } { "_id" : { "hour" : 23, "5_minute_intervals" : 40 }, "result_in_seconds" : 103.16417910447761 } { "_id" : { "hour" : 23, "5_minute_intervals" : 35 }, "result_in_seconds" : 105.7627118644068 } { "_id" : { "hour" : 23, "5_minute_intervals" : 30 }, "result_in_seconds" : 98.38509316770187 } { "_id" : { "hour" : 23, "5_minute_intervals" : 25 }, "result_in_seconds" : 99.6923076923077 } { "_id" : { "hour" : 23, "5_minute_intervals" : 20 }, "result_in_seconds" : 101.1063829787234 } { "_id" : { "hour" : 23, "5_minute_intervals" : 15 }, "result_in_seconds" : 97.75456919060053 } { "_id" : { "hour" : 23, "5_minute_intervals" : 10 }, "result_in_seconds" : 99.48627103631533 } { "_id" : { "hour" : 23, "5_minute_intervals" : 5 }, "result_in_seconds" : 97.30642890661662 }	7110	12960
Query #4	{ "_id" : null, "result_in_seconds" : 98.22110012002815 }	11421	2166
Query #5	{ "_id" : null, "result_in_minutes" : 10.118927408327334 }	11577	15884
Query #6	Johnson Cr NB Foster NB Powell to I-205 NB Division NB Glisan to I-205 NB Columbia to I-205 NB	~0	6

Data Model Critique

- Documents are independent units and makes it easier to distribute data across multiple servers while preserving its locality.
- Application logic is easier to write. You don't have to translate between objects in your application and SQL queries, you can just turn the object model directly into a document.
- Unstructured data can be stored easily, since a document contains whatever keys and values the application logic requires. In addition, costly migrations are avoided since the database does not need to know its information schema in advance.

MongoDB Advice

- No joins. If you need data from more than one collection, you need to do more than one query.
- Extensive and well-written documentation.
- Setup tables with proper types before imports. Conversion after import is time intensive.
- Write failure errors are not explicitly given to the user.

MongoDB - What We Learned

- Using MongoDB feels familiar to those accustomed to OOP as MongoDB uses JavaScript.
- Queries are easily optimized after their initial development.
- Query results can be saved into variables for future use.
- Handles importing multiple data collections conveniently.

Post Project Analysis

- While building the queries they were rebuilt multiple times to make them more efficient. They are still at a point to where they could be rebuilt again.
- We initially were working with Elasticsearch and it required us to try to nest multiple tables which made implementation difficult.
- Time spent on Elasticsearch reduced time using MondoDB.