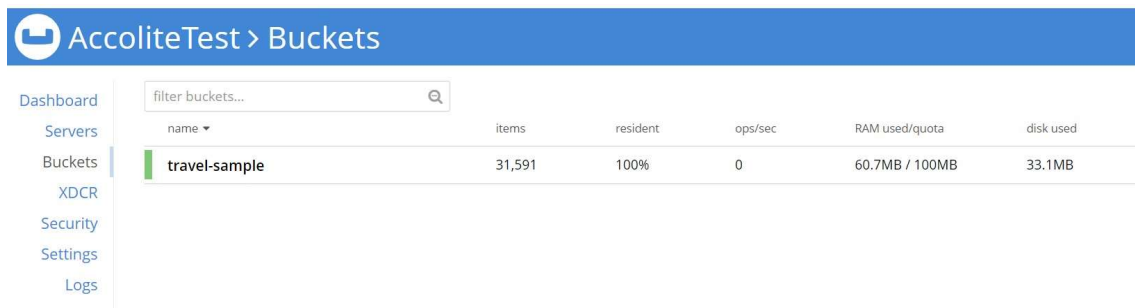


NoSQL Assignment

By Raj Vignesh Karunakaran

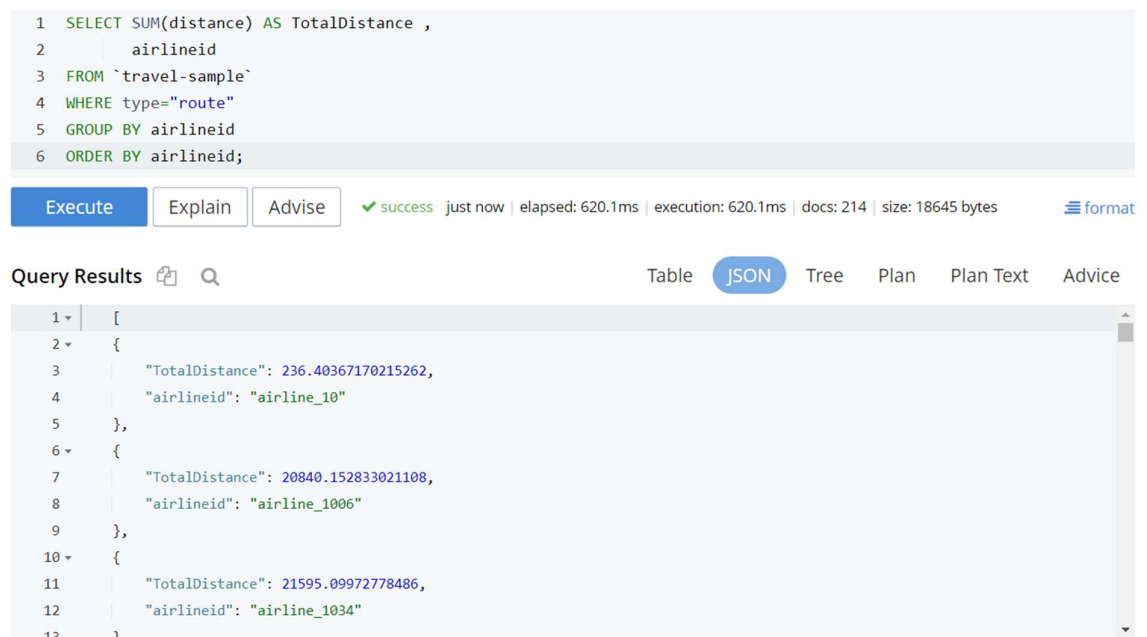
1. Import "Travel_sample" bucket;



The screenshot shows the AccoliteTest interface with a sidebar on the left containing links to Dashboard, Servers, Buckets, XDCR, Security, Settings, and Logs. The main area is titled 'AccoliteTest > Buckets' and features a search bar labeled 'filter buckets...'. Below the search bar is a table with columns: name, items, resident, ops/sec, RAM used/quota, and disk used. The table contains one entry: 'travel-sample' with 31,591 items, 100% resident, 0 ops/sec, 60.7MB / 100MB RAM used, and 33.1MB disk used.

name	items	resident	ops/sec	RAM used/quota	disk used
travel-sample	31,591	100%	0	60.7MB / 100MB	33.1MB

2. Sum aggregate by airline_id



The screenshot shows the AccoliteTest query execution interface. At the top, a SQL query is entered in a text area:

```
1 SELECT SUM(distance) AS TotalDistance ,
2     airlineid
3 FROM `travel-sample`
4 WHERE type="route"
5 GROUP BY airlineid
6 ORDER BY airlineid;
```

Below the query area are buttons for 'Execute', 'Explain', and 'Advise'. To the right of these buttons, a status bar indicates: 'success just now | elapsed: 620.1ms | execution: 620.1ms | docs: 214 | size: 18645 bytes'. A 'format' link is also present.

Below the status bar, the 'Query Results' section is displayed. It includes a search icon and a 'Table' button. The 'JSON' button is selected, and the results are shown in a JSON format. The results are a list of three objects, each containing 'TotalDistance' and 'airlineid'.

```
1 [
2   {
3     "TotalDistance": 236.40367170215262,
4     "airlineid": "airline_10"
5   },
6   {
7     "TotalDistance": 20840.152833021108,
8     "airlineid": "airline_1006"
9   },
10  {
11    "TotalDistance": 21595.09972778486,
12    "airlineid": "airline_1034"
13  }
14 ]
```

3. JOINS

INNER:

```
1 SELECT * FROM `travel-sample` route
2 INNER JOIN `travel-sample` airline
3 ON route.airlineid = META(airline).id
4 WHERE route.type = "route" and airline.type = "airline"
5 AND route.sourceairport = "SFO"
```

Execute Explain Advise ✓ success just now | elapsed: 53ms | execution: 53ms | docs: 183 | size: 669953 bytes [format](#)

Query Results [Q](#) Table **JSON** Tree Plan Plan Text Advice

```
1 + [
2 + {
3 +   "airline": {
4 +     "callsign": "UNITED",
5 +     "country": "United States",
6 +     "iata": "UA",
7 +     "icao": "UAL",
8 +     "id": 5209,
9 +     "name": "United Airlines",
10 +    "type": "airline"
11 +  },
12 +  "route": {
13 +    "airline": "UA",
14 +    "airlineid": "airline 5209".
```

LEFT:

```
1 SELECT * FROM `travel-sample` route
2 LEFT JOIN `travel-sample` airline
3 ON route.airlineid = META(airline).id
4 WHERE route.type = "route" and airline.type = "airline"
5 AND route.sourceairport = "SFO"
```

Execute Explain Advise ✓ success just now | elapsed: 56ms | execution: 56ms | docs: 183 | size: 669953 bytes [format](#)

Query Results [Q](#) Table **JSON** Tree Plan Plan Text Advice

```
1 + [
2 + {
3 +   "airline": {
4 +     "callsign": "UNITED",
5 +     "country": "United States",
6 +     "iata": "UA",
7 +     "icao": "UAL",
8 +     "id": 5209,
9 +     "name": "United Airlines",
10 +    "type": "airline"
11 +  },
12 +  "route": {
13 +    "airline": "UA",
14 +    "airlineid": "airline 5209".
```

RIGHT:

```
1 SELECT * FROM `travel-sample` route
2 RIGHT JOIN `travel-sample` airline
3 ON route.airlineid = META(airline).id
4 WHERE route.type = "route" and airline.type = "airline"
5 AND route.sourceairport = "SFO"
```

Execute Explain Advise success 32 min ago | elapsed: 2s | execution: 2s | docs: 183 | size: 669953 bytes format

Query Results Table JSON Tree Plan Plan Text Advice

```
1 {
2   {
3     "airline": {
4       "callsign": "CITRUS",
5       "country": "United States",
6       "iata": "FL",
7       "icao": "TRS",
8       "id": 1316,
9       "name": "AirTran Airways",
10      "type": "airline"
11    },
12    "route": {
13      "airline": "FL"
```

4. MAPREDUCE views for number of documents based on entity – type:

MAP:

```
function (doc, meta) {
  if(doc.type){
    emit(doc.type,null)
  }
}
```

REDUCE:

```
_count
```

Filter:

```
Group_level = 1
```

Output:

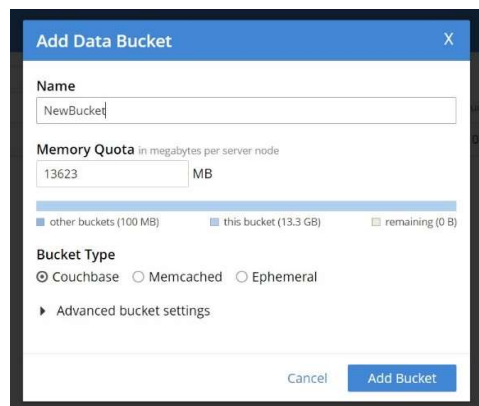
Key		Value
"airline"	undefined	187
"airport"	undefined	1968
"hotel"	undefined	917
"landmark"	undefined	4495
"route"	undefined	24824

5. CLI INTERFACE:

STEPS:

1. Open CMD in Admin mode and navigate to couchbase/bin
2. Use command "cbexport json -c couchbase://127.0.0.1 -u Admin -p Password -b travel-sample -o C:/travelSample.json -f lines -t 4" to export the bucket
3. Created a new bucket in UI named "NewBucket"
4. Used command "cbimport json -c couchbase://127.0.0.1 -u Admin -p Password -b NewBucket -f lines -d file://C:/travelSample.json -t 4 -g %id%" to import data t new bucket

Screenshots:



Bucket Name	Documents	Size	Index Size	Index Count	Index Size / Index Count	Index Size	Actions
NewBucket	28,579	100%	6198	65.7MB / 13.3GB	4.05MB		Documents Statistics
travel-sample	31,591	100%	0	62.5MB / 100MB	47.4MB		Documents Statistics

```
Administrator: Command Prompt
C:\Program Files\Couchbase\Server\bin>cbexport json -c couchbase://127.0.0.1 -u Admin -p [REDACTED] -b travel-sample -o
C:/travelSample.json -f lines -t 4
JSON exported to 'C:/travelSample.json' successfully
Documents exported: 31591 Documents skipped: 0

C:\Program Files\Couchbase\Server\bin>cbimport json -c couchbase://127.0.0.1 -u Admin -p [REDACTED] -b NewBucket -f line
s -d file://C:/travelSample.json -t 4 -g %id%
JSON 'file://C:/travelSample.json' imported to 'http://127.0.0.1:8091' successfully
Documents imported: 31591 Documents failed: 0

C:\Program Files\Couchbase\Server\bin>
```