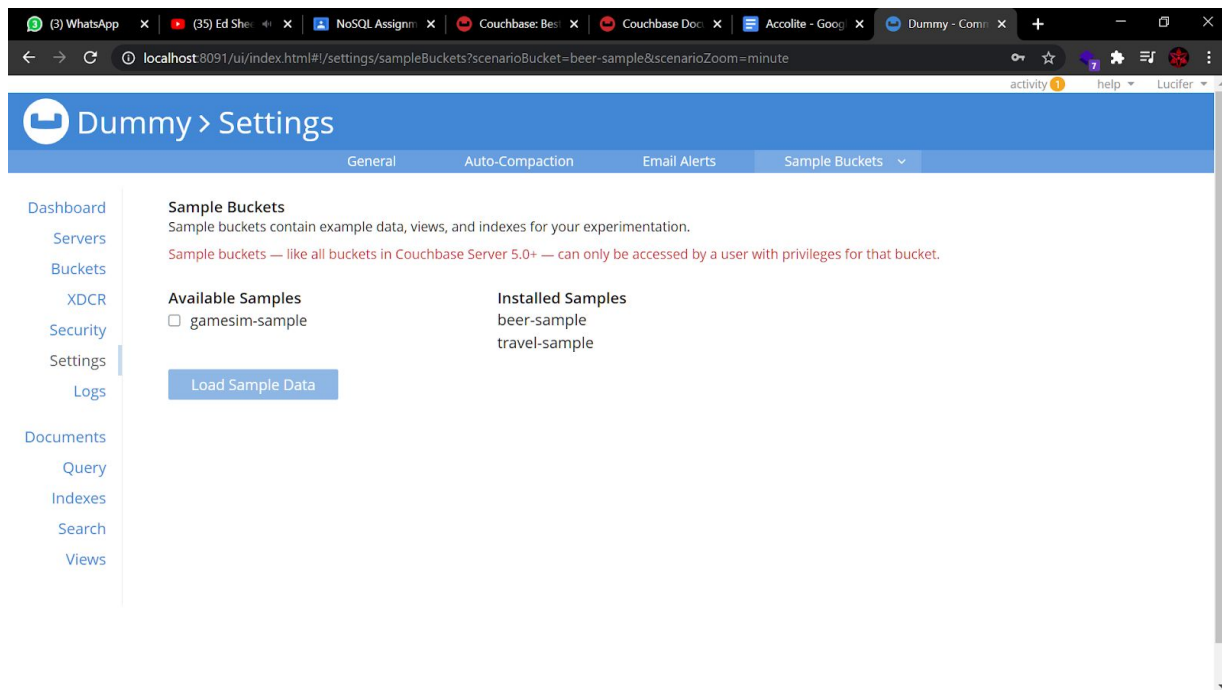


NoSQL

Prabhakar

1. Import `travel-sample` bucket:

travel-sample bucket imported.



2. Write a query to get the sum of all distances where type="route" for each airline id:

```
SELECT airlineid,  
       SUM(distance) AS distanceSum  
FROM `travel-sample`  
WHERE type="route"  
GROUP BY airlineid;
```

The screenshot shows the Dummy Query Workbench interface. The Query Editor contains the following SQL query:

```
1 SELECT airlineid,
2 SUM(distance) AS distanceSum
3 FROM `travel-sample`
```

The Query Results table shows the following data:

airlineid	distanceSum
airline_2622	3276.5051303826995
airline_397	11675.122046440949
airline_5416	244740.41777237272
airline_4533	158507.95890137457
airline_1946	4513.488243880396
airline_2750	3387.4542048073135
airline_3052	34090.50103307319
airline_2757	838.0613385351664
airline_324	347965.52063718165
airline_3179	43033.61840473161
airline_225	75433.5827954445

The Data Insights panel on the right shows the following information:

- Queryable Buckets:**
 - beer-sample** (sampled 1000 of 7303):
 - 'type' = "beer", 'upc' = 0: 82.5%
 - 'type' = "brewery": 17.5%
 - dummy1** (sampled 2 of 2):
 - 'name' = "new hotel", 'type2' = "ol": 100%
 - travel-sample** (sampled 1000 of 31591):
 - 'stops' = 0, 'type' = "route": 79.9%
 - 'type' = "landmark": 11.8%
 - 'type' = "hotel": 2.7%
 - 'type' = "airline": 0.5%
 - 'type' = "airport": 5.1%

3. Write queries to join(LEFT,RIGHT,INNER) type="route"&"airline" and fetch the data whose sourceairport="SFO":

INNER JOIN:

```
SELECT *
FROM `travel-sample` r
JOIN `travel-sample` a ON r.airlineid = META(a).id
WHERE r.sourceairport = "SFO"
```

The screenshot shows the Dummy Query Workbench interface with a JOIN query. The Query Editor contains the following SQL query:

```
1 SELECT *
2 FROM `travel-sample` r
3 JOIN `travel-sample` a ON r.airlineid = META(a).id
```

The Query Results are displayed in JSON format:

```
{
  "a": {
    "callsign": "JETBLUE",
    "country": "United States",
    "iata": "B6",
    "icao": "JBH",
    "id": "3029",
    "name": "JetBlue Airways",
    "type": "airline"
  },
  "r": {
    "callsign": "JETBLUE",
    "country": "United States",
    "iata": "B6",
    "icao": "JBH",
    "id": "3029",
    "name": "JetBlue Airways",
    "type": "airline",
    "sourceairport": "SFO"
  }
}
```

The Data Insights panel on the right shows the following information:

- Queryable Buckets:**
 - beer-sample** (sampled 1000 of 7303):
 - 'srm' = 0, 'type' = "beer": 81.0%
 - 'type' = "brewery": 19.0%
 - dummy1** (sampled 2 of 2):
 - 'name' = "new hotel", 'type2' = "ol": 100%
 - travel-sample** (sampled 1000 of 31591):
 - 'stops' = 0, 'type' = "route": 81.3%
 - 'type' = "landmark": 10.7%
 - 'type' = "hotel": 3.1%
 - 'callsign' = "LONGHORN", 'country' = "United States": 0.5%
 - 'type' = "airport": 4.8%

LEFT JOIN:

```
SELECT *
FROM `travel-sample` r
LEFT JOIN `travel-sample` a
ON r.airlineid = META(a).id
WHERE r.sourceairport = "SFO"
```

The screenshot displays the Dummy Query Workbench interface. The top navigation bar includes 'activity', 'help', and 'Lucifer'. The main header shows 'Dummy > Query' with 'IMPORT' and 'EXPORT' buttons. The left sidebar contains a navigation menu with 'Dashboard', 'Servers', 'Buckets', 'XDCR', 'Security', 'Settings', 'Logs', 'Documents', 'Query', 'Indexes', 'Search', and 'Views'. The 'Query Editor' section shows a SQL query for a LEFT JOIN between 'travel-sample' tables 'r' and 'a' on 'r.airlineid = META(a).id', filtered by 'r.sourceairport = "SFO"'. Below the editor are 'Execute' and 'Explain' buttons, and an 'External Query Advisor' status bar indicating success. The 'Query Results' section shows a JSON document for a flight record from SFO. The right sidebar features 'Data Insights' with 'Queryable Buckets' for 'beer-sample', 'dummy1', and 'travel-sample', each with a 'Refresh' button and a count.

Query Editor

```
1 SELECT *
2 FROM `travel-sample` r LEFT
3 JOIN `travel-sample` a ON r.airlineid = META(a).id
4 WHERE r.sourceairport = "SFO"
```

Execute Explain External Query Advisor success just now | elapsed: 73ms | execution: 73ms | docs: 249 | size: 887372 bytes | format

Query Results

1	2	3	4	5	6	7	8	9	10	11
[{	"r": {	"airline": "AI",	"airlineid": "airline_218",	"destinationairport": "HKG",	"distance": 11128.182035009515,	"equipment": "77W",	"id": 10624,	"schedule": [{

Data Insights

Queryable Buckets

- beer-sample sampled 1000 of 7303
 - 'srm' = 0, 'type' = "beer" 81.0%
 - 'type' = "brewery" 19.0%
 - Indexes
- dummy1 sampled 2 of 2
 - 'name' = "new hotel", 'type2' = "o"
 - Indexes
- travel-sample sampled 1000 of 31591
 - 'stops' = 0, 'type' = "route" 81.3%
 - 'type' = "landmark" 10.7%
 - 'type' = "hotel" 3.1%
 - 'callsign' = "LONGHORN", 'country' = "USA" 1.0%
 - 'type' = "airport" 4.8%
 - Indexes

Queryable By Doc ID Only

NewBucket Refresh 79

RIGHT JOIN:

```
SELECT *
FROM `travel-sample` r
RIGHT JOIN `travel-sample` a
ON r.airlineid = META(a).id
WHERE r.sourceairport = "SFO"
```

Query Editor

```

1 SELECT *
2 FROM 'travel-sample' r RIGHT
3 JOIN 'travel-sample' a ON r.airlineid = META(a).id
4 WHERE r.sourceairport = "SFO"

```

Query Results

```

1 [
2   {
3     "a": {
4       "callsign": "CITRUS",
5       "country": "United States",
6       "iata": "FI",
7       "icao": "FRS",
8       "id": 1316,
9       "name": "AirIran Airways",
10      "type": "airline"
11    },

```

Data Insights

Queryable Buckets

- beer-sample** sampled 1000 of 7303
 - 'srm' = 0, 'type' = "beer" 81.0%
 - 'type' = "brewery" 19.0%
 - Indexes
- dummy1** sampled 2 of 2
 - 'name' = "new hotel", 'type2' = "o"
 - Indexes
- travel-sample** sampled 1000 of 31591
 - 'stops' = 0, 'type' = "route" 81.3%
 - 'type' = "landmark" 10.7%
 - 'type' = "hotel" 3.1%
 - 'callsign' = "LONGHORN", 'country' = "USA" 2.9%
 - 'type' = "airport" 4.8%
 - Indexes

Queryable By Doc ID Only
NewBucket Refresh 79

4. Write a mapreduce to get the number of all documents based on entities(type):

```

function (doc, meta) {
  if(doc.type == "router")
    emit(meta.id, null);
}

```

Documents

```

10 "flight": "LS224",
11 "utc": "13:06:00"
12 }, {
13 "day": 1,
14 "flight": "LS119",

```

View Index Code

Map

```

1 function (doc, meta) {
2   if(doc.type == "route")
3     emit(meta.id, null);
4 }

```

Reduce (built in: _count, _sum, _stats)

```

1 _count

```

Results filter: ?limit=6&state=false&connection_timeout=60000&inclusive_end=true&skip=0&full_set=Development Time Subset

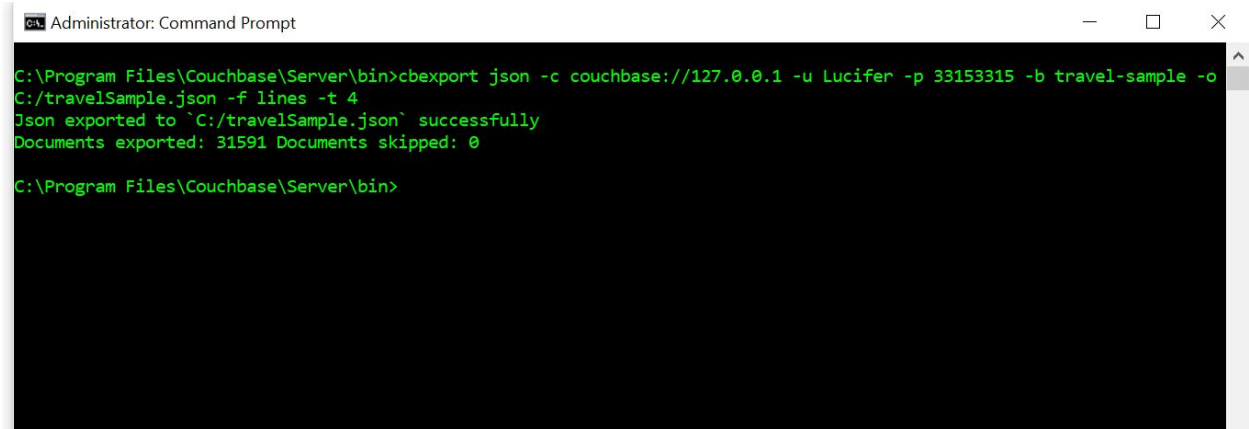
Development Time Subset Full Cluster Data Set

Key Value

5. Refer CLI interface & try to export the travel sample data.

- Go to Couchbase Bin folder - C:\Program Files\Couchbase\Server\bin
- Open Command Prompt from the bin folder
- Run the below command to export the data from travel-sample bucket

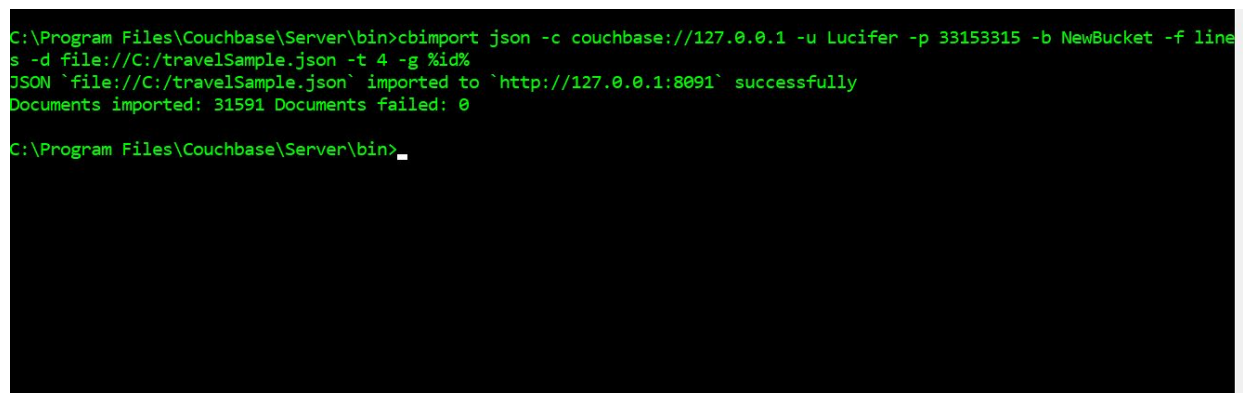
cbexport json -c couchbase://127.0.0.1 -u <USER NAME> -p <password> -b travel-sample -o C:/travelSample.json -f lines -t 4



```
Administrator: Command Prompt
C:\Program Files\Couchbase\Server\bin>cbexport json -c couchbase://127.0.0.1 -u Lucifer -p 33153315 -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>
```

- Create a new Bucket “NewBucket” & import the data to it:

cbimport json -c couchbase://127.0.0.1 -u <User Name> -p <password> -b NewBucket -f lines -d file://C:/travelSample.json -t 4 -g %id%



```
C:\Program Files\Couchbase\Server\bin>cbimport json -c couchbase://127.0.0.1 -u Lucifer -p 33153315 -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>
```

Imported Bucket:

The screenshot shows the Couchbase Dummy Buckets interface. On the left is a sidebar with navigation links: Dashboard, Servers, Buckets, XDCR, Security, Settings, Logs, Documents, Query, Indexes, Search, and Views. The main area displays a table of buckets:

name	items	resident	ops/sec	RAM used/quota	disk used	
beer-sample	7,303	100%	0	36.4MB / 100MB	24.8MB	Documents Statistics
dummy1	2	100%	0	33.2MB / 100MB	15.6MB	Documents Statistics
NewBucket	28,579	100%	0	67.1MB / 100MB	28.4MB	Documents Statistics
travel-sample	31,591	100%	0	75.2MB / 100MB	66.7MB	Documents Statistics

Below the table, the details for 'NewBucket' are shown:

- Type: Couchbase
- Bucket RAM Quota: 100MB
- Cluster RAM Quota: 4.25GB
- Replicas: 1
- Server Nodes: 1
- Ejection Method: Value-Only
- Conflict Resolution: Sequence Number
- Compaction: Not active

Two bar charts are displayed:

- Memory:** Shows cluster quota (4.25 GB) with a breakdown: other buckets (300 MB), this bucket (100 MB), and remaining (3.86 GB).
- Disk:** Shows total cluster storage (202 GB) with a breakdown: other buckets (107 MB), this bucket (28.4 MB), and remaining (141 GB).

At the bottom right, there are buttons for Delete, Compact, and Edit.

Imported data:

The screenshot shows the Couchbase Dummy Documents interface. On the left is a sidebar with navigation links: Dashboard, Servers, Buckets, XDCR, Security, Settings, Logs, Documents, Query, Indexes, Search, and Views. The main area displays a table of documents for the 'NewBucket' bucket:

id	Document ID	N1QL WHERE
10	optional...	no indexes available...
10000		
10001		
10002		
10003		
10004		
10005		
10006		

Below the table, the details for 'NewBucket' are shown:

- Type: Couchbase
- Bucket RAM Quota: 100MB
- Cluster RAM Quota: 4.25GB
- Replicas: 1
- Server Nodes: 1
- Ejection Method: Value-Only
- Conflict Resolution: Sequence Number
- Compaction: Not active

At the bottom right, there are buttons for Delete, Compact, and Edit.