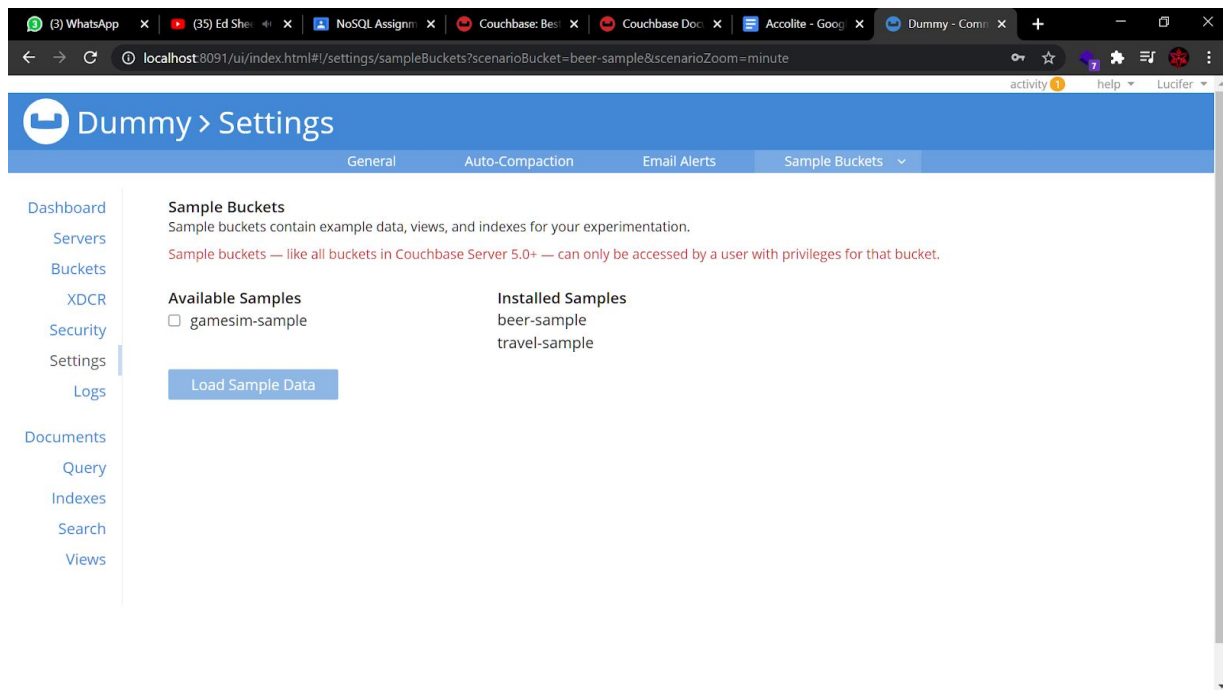


NoSQL

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"
LIMIT 100
```

The screenshot shows the Dummy Query Workbench interface with the following SQL query in the Query Editor:

```
1 SELECT *
2 FROM `travel-sample` r
3 JOIN `travel-sample` a ON r.airlineid = META(a).id
4 WHERE r.sourceairport = "SFO"
5 LIMIT 100
```

The Query Results table shows the following data in JSON format:

1	2
{	{
"a": {	"callsign": "JETBLUE",
"country": "United States",	"iata": "B6",
"icao": "JBU",	"id": "3029",
"name": "JetBlue Airways",	"type": "airline"
}	}

The Data Insights panel on the right shows the same information as the first screenshot.

LEFT JOIN:

```
SELECT *
FROM `travel-sample` r
LEFT JOIN `travel-sample` a
ON r.airlineid = META(a).id
WHERE r.sourceairport = "SFO"
LIMIT 100
```

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

```
1 SELECT *
2 FROM `travel-sample` r LEFT
3 JOIN `travel-sample` a ON r.airlineid = META(a).id
4 WHERE r.sourceairport = "SFO"
5 LIMIT 100
```

The Query Results section displays the results in JSON format. The first result is a document with the following structure:

```
{
  "r": {
    "airline": "A1",
    "airlineid": "airline_218",
    "destinationairport": "HKG",
    "distance": 11128.182035009515,
    "equipment": "77W",
    "id": 10624,
    "schedule": [
      {
        "start": "2013-01-01T00:00:00Z",
        "end": "2013-01-01T00:00:00Z"
      }
    ]
  }
}
```

The Data Insights section 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%
 - [Indexes](#)
- dummy1** sampled 2 of 2
 - ``name` = "new hotel", `type2` = "ol"`
 - [Indexes](#)
- 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%
 - [Indexes](#)

A Refresh button is located at the bottom of the Data Insights section.

RIGHT JOIN:

```
SELECT *
FROM `travel-sample` r
RIGHT JOIN `travel-sample` a
ON r.airlineid = META(a).id
WHERE r.sourceairport = "SFO"
LIMIT 100
```

The screenshot shows the Dummy Query Workbench interface. The top navigation bar includes 'Dashboard', 'Servers', 'Buckets', 'XDCR', 'Security', 'Settings', 'Logs', 'Documents', 'Query', 'Indexes', 'Search', and 'Views'. The main area is divided into three sections:

- Query Editor:** Contains a SQL query:


```
1 SELECT *
2 FROM `travel-sample` r RIGHT
3 JOIN `travel-sample` a ON r.airlineid = META(a.id)
4 WHERE r.sourceairport = "SFO"
5 LIMIT 100
```

 Below the query are buttons for 'Execute', 'Explain', and 'Format'. A status bar indicates 'External Query Advisor success just now | elapsed: 1.4s | execution: 1.4s | docs: 100 | size: 366095 bytes'.
- Query Results:** Displays the results in JSON format. The first result is:


```
{
  "a": {
    "callsign": "CITRUS",
    "country": "United States",
    "iata": "FL",
    "icao": "TRS",
    "id": 1316,
    "name": "AirTran Airways",
    "type": "airline"
  }
}
```
- Data Insights:** Shows 'Queryable Buckets' for 'beer-sample', 'dummy1', and 'travel-sample'. For 'travel-sample', it lists:
 - 'stops' = 0, 'type' = 'route': 79.9%
 - 'type' = 'landmark': 11.8%
 - 'type' = 'hotel': 2.7%
 - 'type' = 'airline': 0.5%
 - 'type' = 'airport': 5.1%

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);
}
```

The screenshot shows the Dummy Query Workbench interface with a MapReduce job configured. The top navigation bar is the same as the previous screenshot. The main area is divided into three sections:

- Documents:** Shows a snippet of a document:


```
10 "flight": "LS224",
11 "utc": "13:06:00"
12 }, {
13 "day": 1,
14 "flight": "LS119",
```
- View Index Code:** Contains the Map function:


```
1 function (doc, meta) {
2   if(doc.type == "route")
3     emit(meta.id, null);
4 }
```
- Reduce:** Contains the Reduce function:

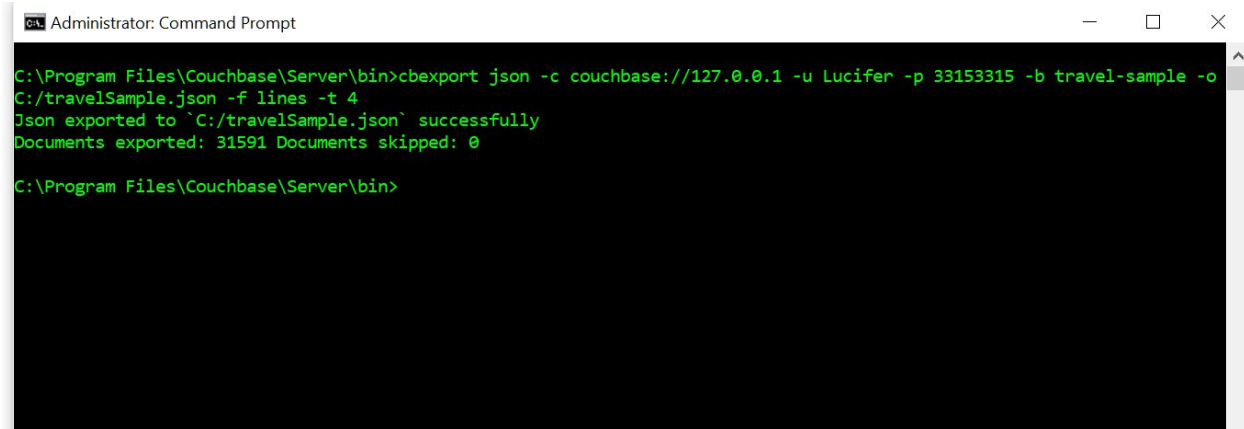

```
1 _count
```

At the bottom, there are buttons for 'Make Copy' and 'Save Changes'. Below these, there is a 'Results' section with a filter: 'filter: ?limit=6&state=false&connection_timeout=60000&inclusive_end=true&skip=0&full_set='. There are buttons for 'Development Time Subset', 'Full Cluster Data Set', and 'Show Results'.

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

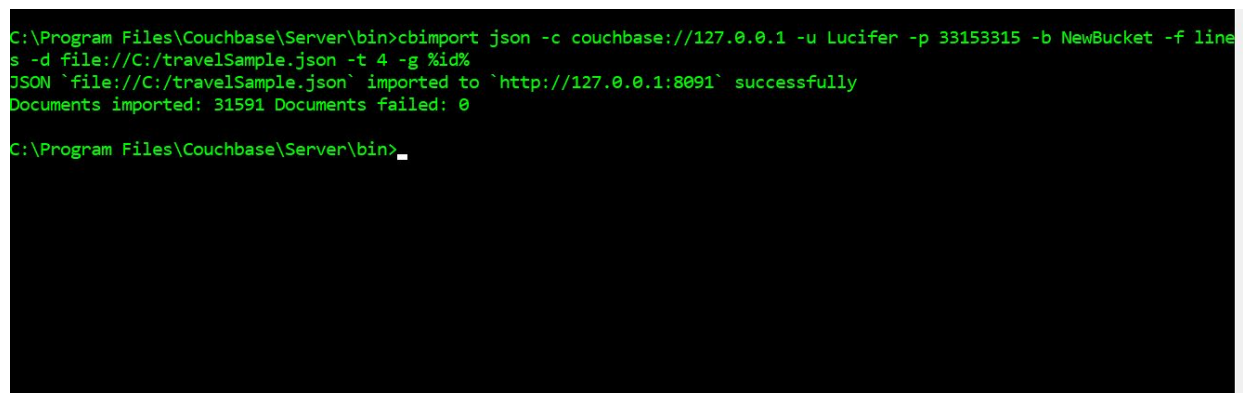
cbexport json -c couchbase://127.0.0.1 -u <USER NAME> -p <password> -b travel-sample -o C:/travelSample.json -f lines -t 4



```
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 dashboard. On the left is a navigation menu with links to 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 segments for other buckets (300 MB), this bucket (100 MB), and remaining (3.86 GB).
- Disk:** Shows total cluster storage (202 GB) with segments for other buckets (107 MB), this bucket (28.4 MB), and remaining (141 GB).

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

Imported data:

The screenshot shows the Couchbase Dummy Documents editor. The top navigation bar includes 'CLASSIC EDITOR' and 'ADD DOCUMENT'. Below this is a 'Document Editor' section with fields for Bucket (NewBucket), Limit (50), Offset (0), Document ID (optional...), and N1QL WHERE (no indexes available...). A 'Retrieve Docs' button is on the right.

Below the editor fields, it says '50 Results for NewBucket, limit: 50, offset: 0'. There is a toggle for 'enable field editing' and navigation links for '< prev batch' and 'next batch >'. The main area displays a list of documents with columns for id and JSON content:

id	JSON Content
10	{ "iata": "Q5", "icao": "MLA", "id": 10, "name": "40-Mile Air", "type": "airline", "callsign": "MILE-AIR", "country": "United States" }
10000	{ "type": "route", "airline": "AF", "airlineid": "airline_137", "destinationairport": "MRS", "distance": 2881, "id": 10000, "sourceairport": "TLV", "equipment": "320", "schedule": [{ "day": 0, "flight": "AF198...", "utc": "15:05:00" }, { "day": 0, "flight": "AF392", "utc": "11:04:00" }, { "day": 0, "flight": "AF114", "utc": "07:35:00" }, { "day": 0, "flight": "AF062", "utc": "02:03:00" }, { "day": 1, "flight": "AF198...", "utc": "15:05:00" }, { "day": 1, "flight": "AF392", "utc": "11:04:00" }, { "day": 1, "flight": "AF114", "utc": "07:35:00" }, { "day": 1, "flight": "AF062", "utc": "02:03:00" }] }
10001	{ "destinationairport": "NCE", "distance": 2735.2013399811754, "equipment": "320", "schedule": [{ "day": 0, "flight": "AF248", "utc": "21:24:00" }, { "day": 1, "flight": "AF517", "utc": "13:36:00" }, { "day": 1, "flight": "AF198...", "utc": "15:05:00" }, { "day": 1, "flight": "AF392", "utc": "11:04:00" }, { "day": 0, "flight": "AF114", "utc": "07:35:00" }, { "day": 0, "flight": "AF062", "utc": "02:03:00" }, { "day": 1, "flight": "AF198...", "utc": "15:05:00" }, { "day": 1, "flight": "AF392", "utc": "11:04:00" }, { "day": 1, "flight": "AF114", "utc": "07:35:00" }, { "day": 1, "flight": "AF062", "utc": "02:03:00" }] }
10002	{ "airline": "AF", "schedule": [{ "day": 0, "flight": "AF986", "utc": "22:26:00" }, { "day": 0, "flight": "AF962", "utc": "14:25:00" }, { "day": 1, "flight": "AF301", "utc": "12:11:00" }, { "day": 0, "flight": "AF923", "utc": "04:31:00" }, { "day": 1, "flight": "AF986", "utc": "22:26:00" }, { "day": 1, "flight": "AF962", "utc": "14:25:00" }, { "day": 0, "flight": "AF301", "utc": "12:11:00" }, { "day": 0, "flight": "AF923", "utc": "04:31:00" }] }
10003	{ "airline": "AF", "airlineid": "airline_137", "destinationairport": "AMS", "equipment": "777", "schedule": [{ "flight": "AF545", "utc": "13:22:00", "day": 0 }, { "day": 0, "flight": "AF350", "utc": "01:21:00" }, { "day": 0, "flight": "AF114", "utc": "07:35:00" }, { "day": 0, "flight": "AF062", "utc": "02:03:00" }, { "day": 1, "flight": "AF545", "utc": "13:22:00", "day": 1 }, { "day": 1, "flight": "AF350", "utc": "01:21:00", "day": 1 }, { "day": 1, "flight": "AF114", "utc": "07:35:00", "day": 1 }, { "day": 1, "flight": "AF062", "utc": "02:03:00", "day": 1 }] }
10004	{ "airlineid": "airline_137", "destinationairport": "MNL", "schedule": [{ "day": 0, "flight": "AF117", "utc": "11:04:00" }, { "day": 0, "flight": "AF096", "utc": "16:52:00" }, { "day": 1, "flight": "AF210", "utc": "14:25:00" }, { "day": 1, "flight": "AF117", "utc": "11:04:00" }, { "day": 1, "flight": "AF096", "utc": "16:52:00" }, { "day": 0, "flight": "AF210", "utc": "14:25:00" }, { "day": 0, "flight": "AF117", "utc": "11:04:00" }] }
10005	{ "airline": "AF", "distance": 365.28208402314306, "schedule": [{ "day": 0, "flight": "AF288", "utc": "05:02:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF288", "utc": "05:02:00" }, { "day": 1, "flight": "AF320", "utc": "10:40:00" }, { "day": 1, "flight": "AF288", "utc": "05:02:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF288", "utc": "05:02:00" }] }
10006	{ "airline": "AF", "distance": 365.28208402314306, "schedule": [{ "day": 0, "flight": "AF288", "utc": "05:02:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF288", "utc": "05:02:00" }, { "day": 1, "flight": "AF320", "utc": "10:40:00" }, { "day": 1, "flight": "AF288", "utc": "05:02:00" }, { "day": 0, "flight": "AF320", "utc": "10:40:00" }, { "day": 0, "flight": "AF288", "utc": "05:02:00" }] }