

## Assignment:

### 1. Import `travel-sample` bucket

The screenshot shows the Accolite Enterprise Edition 6.6.1 web interface. The browser address bar displays the URL: `localhost:8091/ui/index.html#!/settings/sampleBuckets?scenarioBucket=gamesim-sample&scenarioZoom=minute`. The page title is "Accolite > Settings". The navigation menu on the left includes: Dashboard, Servers, Buckets, XDCR, Security, Settings (selected), Logs, Documents, Query, Indexes, Search, Analytics, Eventing, and Views. The main content area is titled "Sample Buckets" and contains the following information:

- Sample Buckets**: Sample buckets contain example data, views, and indexes for your experimentation.
- Sample buckets** — like all buckets in Couchbase Server 5.0+ — can only be accessed by a user with privileges for that bucket.
- Available Samples**: ☐ beer-sample
- Installed Samples**: gamesim-sample, travel-sample
- Load Sample Data**: A blue button to load sample data.

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

The screenshot shows the Accolite Enterprise Edition 6.6.1 web interface, specifically the Query Editor. The page title is "Accolite > Query". The navigation menu on the left is the same as in the first screenshot. The main content area is titled "Query Editor" and contains the following information:

- Query Editor**: A text area with the following SQL query:

```
1 select sum(distance) as TOTAL_DISTANCE from `travel-sample`
2 where type like "route"
```
- Execute**: A blue button to execute the query.
- Explain**: A button to explain the query.
- Advise**: A button to advise the query.
- Success**: A green checkmark icon and the text "success just now | elapsed: 827.6ms | execution: 827.6ms | docs: 1 | size: 51 bytes".
- format**: A button to format the query results.
- Query Results**: A section showing the results of the query. The results are displayed in a table format with the following columns: Line Number, Key, and Value.

Line Number	Key	Value
1	*	[
2	*	{
3		"TOTAL_DISTANCE": 53538158.56144267
4		}
5		]

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

Inner join:

The screenshot shows the Accolite Query Editor interface. The query editor contains the following SQL code:

```
1 select * from `travel-sample` a
2 join `travel-sample` b
3 on a.airlineid = meta(b).id
4 where a.sourceairport="SFO"
```

Below the query editor, the status bar indicates: **Execute** **Explain** **Advise** ✓ success just now | elapsed: 83.8ms | execution: 83.8ms | docs: 183 | size: 668123 bytes [format](#)

The Query Results section shows the results in JSON format:

```
1 [
2 {
3   "a": {
4     "airline": "B6",
5     "airlineid": "airline_3029",
6     "destinationairport": "AUS",
7     "distance": 2416.0035377223094,
8     "equipment": "320",
9     "id": 14239,
10    "schedule": [
11      {
12        "day": 0,
13        "flight": "B6555",
14        "utc": "02:34:00"
15      },
16    ]
17  }
```

Left join:

The screenshot shows the Accolite Query Editor interface. The query editor contains the following SQL code:

```
1 select * from `travel-sample` a
2 left join `travel-sample` b
3 on a.airlineid = meta(b).id
4 where a.sourceairport="SFO"
```

Below the query editor, the status bar indicates: **Execute** **Explain** **Advise** ✓ success just now | elapsed: 111.7ms | execution: 111.7ms | docs: 249 | size: 887372 bytes [format](#)

The Query Results section shows the results in JSON format:

```
1 [
2 {
3   "a": {
4     "airline": "AI",
5     "airlineid": "airline_218",
6     "destinationairport": "HKG",
7     "distance": 11128.182035009515,
8     "equipment": "77W",
9     "id": 10624,
10    "schedule": [
11      {
12        "day": 0,
13        "flight": "AI472",
14        "utc": "09:43:00"
15      },
16    ]
17  }
```

Right join:

The screenshot shows the Accolite Query Editor interface. The top navigation bar includes 'Accolite > Query', 'Query Workbench', and 'Query Monitor'. The left sidebar contains a navigation menu with options: Dashboard, Servers, Buckets, XDCR, Security, Settings, Logs, Documents, Query, Indexes, Search, Analytics, Eventing, and Views. The main area is titled 'Query Editor' and contains a SQL query:

```
1 select * from `travel-sample` a
2 right join `travel-sample` b
3 on a.airlineid = meta(b).id
4 where a.sourceairport="SFO"
```

Below the query, there are buttons for 'Execute', 'Explain', and 'Advise'. A status bar indicates 'success just now | elapsed: 18.5s | execution: 18.5s | docs: 183 | size: 668123 bytes'. To the right is a 'format' link. Below the query editor is the 'Query Results' section, which has tabs for 'Table', 'JSON', 'Tree', 'Plan', 'Plan Text', and 'Advice'. The 'JSON' tab is selected, showing a JSON array of results. The first result is expanded, showing a document with fields like 'airline', 'airlineid', 'destinationairport', 'distance', 'equipment', 'id', and 'schedule'.

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

The screenshot shows the Accolite MapReduce interface. The top navigation bar includes 'Accolite - Enterprise Edition 6.6.1'. The left sidebar contains a navigation menu with options: Query, Indexes, Search, Analytics, Eventing, and Views. The main area is titled 'View Index Code' and contains a Map function:

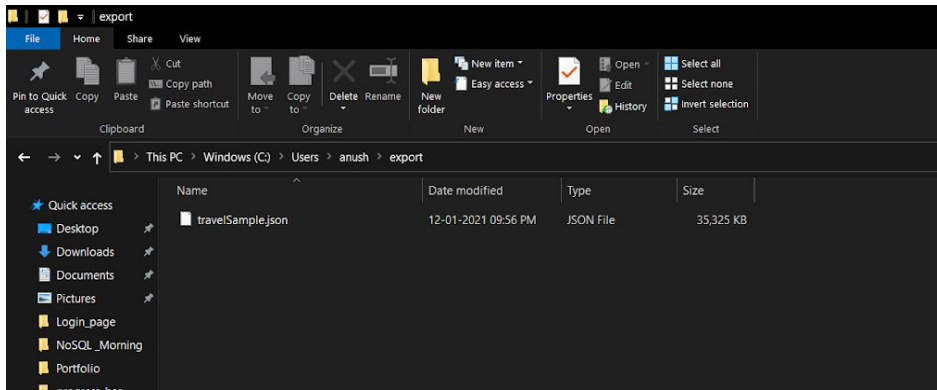
```
1 function (doc, meta) {
2   if(doc.type=="route")
3     emit(meta.id, null);
4 }
```

Below the map function, there is a 'Reduce (built in: \_count, \_sum, \_stats)' section with a single line of code: '1 \_count'. To the right are buttons for 'Make Copy' and 'Save Changes'. Below the code editor is the 'Results' section, which has 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'. The results table shows a single row with 'Key' as 'null undefined' and 'Value' as '34'.

5. Refer CLI interface & try to export the travel sample data. Below are the steps:

## EXPORT:

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



## IMPORT:

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

name	Items	resident	ops/sec	RAM used/quota	disk used	
gamesim-sample	586	100%	0	28.2MB / 100MB	14.4MB	Documents Statistics
NewBucket	28,579	100%	0	53.1MB / 8.38GB	27.5MB	Documents Statistics
travel-sample	31,591	100%	0	51.4MB / 100MB	51.7MB	Documents Statistics