

Region Queries

These are example N1QL queries that may can performed to retrieve region related data.

Country By ID

The following query will get a Region by its Document ID.

Query

[region_by_document_id.n1ql](#)

```
1 SELECT regions.*
2 FROM `flight-data` AS regions
3 USE KEYS 'region_US-KS'
```

Result

```
1 [
2   {
3     "_id": "region_US-KS",
4     "continent_code": "NA",
5     "doc_type": "region",
6     "iso_country": "US",
7     "local_code": "KS",
8     "region_code": "US-KS",
9     "region_id": 306092,
10    "region_name": "Kansas"
11  }
12 ]
```

The following query will retrieve multiple Regions by their Document ID.

Query

[regions_by_document_id.n1ql](#)

```
1 SELECT regions.*
2 FROM `flight-data` AS regions
3 USE KEYS ['region_US-KS', 'region_US-NC', 'region_US-MN']
```

Result

```
1  [
2    {
3      "_id": "region_US-KS",
4      "continent_code": "NA",
5      "doc_type": "region",
6      "iso_country": "US",
7      "local_code": "KS",
8      "region_code": "US-KS",
9      "region_id": 306092,
10     "region_name": "Kansas"
11   },
12   {
13     "_id": "region_US-NC",
14     "continent_code": "NA",
15     "doc_type": "region",
16     "iso_country": "US",
17     "local_code": "NC",
18     "region_code": "US-NC",
19     "region_id": 306103,
20     "region_name": "North Carolina"
21   },
22   {
23     "_id": "region_US-MN",
24     "continent_code": "NA",
25     "doc_type": "region",
26     "iso_country": "US",
27     "local_code": "MN",
28     "region_code": "US-MN",
29     "region_id": 306099,
30     "region_name": "Minnesota"
31   }
32 ]
```

Region By Code

As our region documents are keyed by the `region_code` , generally we will retrieve these documents directly through their document id. However, we can retrieve these same documents by querying on the `region_code` value within the document.

Index

[idx_regions_region_code.n1qI](#)

```

1 CREATE INDEX idx_regions_region_code ON `flight-data` ( iso_country, region_code )
2 WHERE doc_type = 'region'
3     AND iso_country IS NOT MISSING
4     AND region_code IS NOT MISSING
5 USING GSI

```

The following query will get a Region by its Document ID.

Query

[region_by_code.n1ql](#)

```

1 SELECT regions.region_code, regions.region_name
2 FROM `flight-data` AS regions
3 WHERE regions.iso_country = 'US'
4     AND regions.region_code = 'US-KS'
5     AND regions.doc_type = 'region'

```

Result

```

1 [
2   {
3     "region_code": "US-KS",
4     "region_name": "Kansas"
5   }
6 ]

```

The following query will retrieve multiple Regions by their `region_code`

Query

[regions_by_code_or.n1ql](#)

```

1 SELECT regions.region_code, regions.region_name
2 FROM `flight-data` AS regions
3 WHERE regions.iso_country = 'US'
4     AND (
5         regions.region_code = 'US-KS'
6         OR regions.region_code = 'US-NC'
7         OR regions.region_code = 'US-MN'
8     )
9     AND regions.doc_type = 'region'

```

Result

```

1  [
2    {
3      "region_code": "US-KS",
4      "region_name": "Kansas"
5    },
6    {
7      "region_code": "US-MN",
8      "region_name": "Minnesota"
9    },
10   {
11     "region_code": "US-NC",
12     "region_name": "North Carolina"
13   }
14 ]

```

While the previous query will return the correct results, it is not efficient or concise. We can update the query to use the `IN` statement defining our codes as an array.

Query

[regions_by_code_in.n1ql](#)

```

1  SELECT regions.region_code, regions.region_name
2  FROM `flight-data` AS regions
3  WHERE regions.iso_country = 'US'
4         AND regions.region_code IN ['US-KS', 'US-NC', 'US-MN']
5         AND regions.doc_type = 'region'

```

Result

```

1  [
2    {
3      "region_code": "US-KS",
4      "region_name": "Kansas"
5    },
6    {
7      "region_code": "US-MN",
8      "region_name": "Minnesota"
9    },
10   {
11     "region_code": "US-NC",
12     "region_name": "North Carolina"
13   }
14 ]

```

All Regions for a Country

Next we want to retrieve all region codes and names for a given country.

Query

[all_regions_by_country_code.n1ql](#)

```
1 SELECT regions.region_code, regions.region_name
2 FROM `flight-data` AS regions
3 WHERE regions.iso_country = 'US'
4       AND regions.region_code IS NOT MISSING
5       AND regions.doc_type = 'region'
6 ORDER BY regions.region_name ASC
```

Results

```
1 [
2   {
3     "region_code": "US-AL",
4     "region_name": "Alabama"
5   },
6   {
7     "region_code": "US-AK",
8     "region_name": "Alaska"
9   },
10  {
11    "region_code": "US-AZ",
12    "region_name": "Arizona"
13  },
14  {
15    "region_code": "US-AR",
16    "region_name": "Arkansas"
17  },
18  {
19    "region_code": "US-CA",
20    "region_name": "California"
21  },
22  ...
23 ]
```

Regions Country with Continent

Now lets say we also want to return both the continent and country codes and names for each region.

Query

[all_regions_by_country_code_continent.n1ql](#)

```
1  SELECT regions.region_code, regions.region_name,
2         countries.country_code, countries.country_name,
3         continents.continent_code, continents.continent_name
4  FROM `flight-data` AS regions
5  INNER JOIN `flight-data` AS countries
6      ON KEYS 'country_' || regions.iso_country
7  INNER JOIN `flight-data` AS continents
8      ON KEYS 'continent_' || regions.continent_code
9  WHERE regions.iso_country = 'US'
10     AND regions.region_code IS NOT MISSING
11     AND regions.doc_type = 'region'
12  ORDER BY regions.region_name ASC
```

Results

```
1  [
2    {
3      "continent_code": "NA",
4      "continent_name": "North America",
5      "country_code": "US",
6      "country_name": "United States",
7      "region_code": "US-AL",
8      "region_name": "Alabama"
9    },
10   {
11     "continent_code": "NA",
12     "continent_name": "North America",
13     "country_code": "US",
14     "country_name": "United States",
15     "region_code": "US-AK",
16     "region_name": "Alaska"
17   },
18   {
19     "continent_code": "NA",
20     "continent_name": "North America",
21     "country_code": "US",
22     "country_name": "United States",
23     "region_code": "US-AZ",
24     "region_name": "Arizona"
25   },
26   {
27     "continent_code": "NA",
28     "continent_name": "North America",
29     "country_code": "US",
30     "country_name": "United States",
31     "region_code": "US-AR",
32     "region_name": "Arkansas"
33   },
34   {
35     "continent_code": "NA",
36     "continent_name": "North America",
37     "country_code": "US",
38     "country_name": "United States",
39     "region_code": "US-CA",
40     "region_name": "California"
41   },
42   ...
43 ]
```

