

Airport Runway Queries

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

Airport Runways by Code

This query uses the previously created `idx_airports_codes` index.

Query

This query will find the available runways and information by the 3 character IATA / FAA code of the airport

[airportrunwaysbyiatacode.n1ql](#)

```
1  SELECT runways.runway_id, runways.low_bearing, runways.high_bearing, runways.lighted,
2      runways.runway_length, runways.runway_width, runways.surface
3  FROM `flight-data` AS airport_codes
4  USE KEYS 'airport_code_ICT'
5  INNER JOIN `flight-data` AS airport_runways
6      ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_runways'
7  UNNEST airport_runways.runways AS runways_lookup
8  INNER JOIN `flight-data` AS runways
9      ON KEYS 'runway_' || TOSTRING( runways_lookup )
10 WHERE runways.closed = false
```

This query will find the available runways and information by the 4 character ICAO code of the airport

[airportrunwaysbyicaocode.n1ql](#)

```
1  SELECT runways.runway_id, runways.low_bearing, runways.high_bearing, runways.lighted,
2      runways.runway_length, runways.runway_width, runways.surface
3  FROM `flight-data` AS airport_codes
4  USE KEYS 'airport_code_KICT'
5  INNER JOIN `flight-data` AS airport_runways
6      ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_runways'
7  UNNEST airport_runways.runways AS runways_lookup
8  INNER JOIN `flight-data` AS runways
9      ON KEYS 'runway_' || TOSTRING( runways_lookup )
10 WHERE runways.closed = false
```

Both queries will yield the same exact result.

Result

```
1  [
2    {
3      "high_bearing": {
4        "displaced_threshold": null,
5        "elevation": 1330,
6        "ident": "19R",
7        "latitude": 37.6616,
8        "longitude": -97.4338,
9        "magnetic_heading": 200
10     },
11     "lighted": true,
12     "low_bearing": {
13       "displaced_threshold": null,
14       "elevation": 1313,
15       "ident": "01L",
16       "latitude": 37.635,
17       "longitude": -97.446,
18       "magnetic_heading": 20
19     },
20     "runway_id": 240444,
21     "runway_length": 10301,
22     "runway_width": 150,
23     "surface": "CON"
24   },
25   {
26     "high_bearing": {
27       "displaced_threshold": null,
28       "elevation": 1320,
29       "ident": "19L",
30       "latitude": 37.6616,
31       "longitude": -97.4177,
32       "magnetic_heading": 200
33     },
34     "lighted": true,
35     "low_bearing": {
36       "displaced_threshold": null,
37       "elevation": 1321,
38       "ident": "01R",
39       "latitude": 37.6428,
40       "longitude": -97.4263,
41       "magnetic_heading": 20
42     }
43   }
44 ]
```

```

42     },
43     "runway_id": 240443,
44     "runway_length": 7301,
45     "runway_width": 150,
46     "surface": "CON"
47 },
48 {
49     "high_bearing": {
50         "displaced_threshold": null,
51         "elevation": 1322,
52         "ident": "32",
53         "latitude": 37.6426,
54         "longitude": -97.4292,
55         "magnetic_heading": 330
56     },
57     "lighted": true,
58     "low_bearing": {
59         "displaced_threshold": null,
60         "elevation": 1332,
61         "ident": "14",
62         "latitude": 37.6575,
63         "longitude": -97.4401,
64         "magnetic_heading": 150
65     },
66     "runway_id": 240445,
67     "runway_length": 6301,
68     "runway_width": 150,
69     "surface": "CON"
70 }
71 ]

```

Return just the low and high bearing runway identifiers.

Query

This query will find the available runways by the 3 character IATA / FAA code of the airport

[airportrunwayidentsbyiata_code.n1ql](#)

```

1 SELECT runways.low_bearing.ident || '/' || runways.high_bearing.ident AS runway
2 FROM `flight-data` AS airport_codes
3 USE KEYS 'airport_code_ICT'
4 INNER JOIN `flight-data` AS airport_runways
5     ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_runways'
6 UNNEST airport_runways.runways AS runways_lookup
7 INNER JOIN `flight-data` AS runways
8     ON KEYS 'runway_' || TOSTRING( runways_lookup )
9 WHERE runways.closed = false

```

This query will find the available runways and information by the 4 character ICAO code of the airport

[airportrunwayidentsbyicao_code.n1ql](#)

```

1 SELECT runways.low_bearing.ident || '/' || runways.high_bearing.ident AS runway
2 FROM `flight-data` AS airport_codes
3 USE KEYS 'airport_code_ICT'
4 INNER JOIN `flight-data` AS airport_runways
5     ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_runways'
6 UNNEST airport_runways.runways AS runways_lookup
7 INNER JOIN `flight-data` AS runways
8     ON KEYS 'runway_' || TOSTRING( runways_lookup )
9 WHERE runways.closed = false

```

Both queries will yield the same exact result.

Result

```

1 [
2   {
3     "runway": "01L/19R"
4   },
5   {
6     "runway": "01R/19L"
7   },
8   {
9     "runway": "14/32"
10  }
11 ]

```

Airport Information with Runways

For this query we want to retrieve a single record with the airport information with a single attribute that is an array of each of the airports runway identifiers for active runways only.

Query

This query will find the available runways by the 3 character IATA / FAA code of the airport

[airportwithrunwayidentsbyiatacode.n1ql](#)

```
1  SELECT airports.airport_id, airports.airport_name, airports.airport_type,
2      airports.iso_region, airports.municipality,
3      IFNULL( airports.airport_iata, airports.airport_icao, airports.airport_ident ) AS ai
4      ARRAY
5          runway.low_bearing.ident || IFNULL('/', || runway.high_bearing.ident, '')
6      FOR runway IN IFMISSING(runways, [])
7      WHEN runway.closed = false
8      END AS runways
9  FROM `flight-data` AS codes
10  USE KEYS 'airport_code_ICT'
11  INNER JOIN `flight-data` AS airports ON KEYS 'airport_' || TOSTRING( codes.id )
12  LEFT NEST `flight-data` AS runways ON KEYS (
13      ARRAY runway.runway_id FOR runway IN (
14          SELECT 'runway_' || TOSTRING( runway_id ) AS runway_id
15          FROM `flight-data` AS runway_lookup
16          USE KEYS
17              'airport_' || TOSTRING(codes.id) || '_runways'
18          UNNEST runway_lookup.runways AS runway_id
19      ) END
20  )
```

This query will find the available runways and information by the 4 character ICAO code of the airport

[airportwithrunwayidentsbyicaocode.n1ql](#)

```

1  SELECT airports.airport_id, airports.airport_name, airports.airport_type,
2     airports.iso_region, airports.municipality,
3     IFNULL( airports.airport_iata, airports.airport_icao, airports.airport_ident ) AS ai
4     ARRAY
5         runway.low_bearing.ident || IFNULL('/', || runway.high_bearing.ident, '')
6     FOR runway IN IFMISSING(runways, [])
7     WHEN runway.closed = false
8     END AS runways
9  FROM `flight-data` AS codes
10 USE KEYS 'airport_code_KICT'
11 INNER JOIN `flight-data` AS airports ON KEYS 'airport_' || TOSTRING( codes.id )
12 LEFT NEST `flight-data` AS runways ON KEYS (
13     ARRAY runway.runway_id FOR runway IN (
14         SELECT 'runway_' || TOSTRING( runway_id ) AS runway_id
15         FROM `flight-data` AS runway_lookup
16         USE KEYS
17             'airport_' || TOSTRING(codes.id) || '_runways'
18         UNNEST runway_lookup.runways AS runway_id
19     ) END
20 )

```

Both queries will yield the same exact result.

Result

```

1  [
2    {
3      "airport_code": "ICT",
4      "airport_id": 3605,
5      "airport_name": "Wichita Dwight D. Eisenhower National Airport",
6      "airport_type": "large_airport",
7      "iso_region": "US-KS",
8      "municipality": "Wichita",
9      "runways": [
10       "01R/19L",
11       "14/32",
12       "01L/19R"
13     ]
14   }
15 ]

```