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

airport*runways*by*iata*code.n1ql

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

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

airport*runways*by*icao*code.n1ql

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

Both queries will yield the same exact result.

Result


```
[
  {
    "high_bearing": {
      "displaced_threshold": null,
      "elevation": 1330,
      "ident": "19R",
      "latitude": 37.6616,
      "longitude": -97.4338,
      "magnetic_heading": 200
    },
    "lighted": true,
    "low_bearing": {
      "displaced_threshold": null,
      "elevation": 1313,
      "ident": "01L",
      "latitude": 37.635,
      "longitude": -97.446,
      "magnetic_heading": 20
    },
    "runway_id": 240444,
    "runway_length": 10301,
    "runway_width": 150,
    "surface": "CON"
  },
  {
    "high_bearing": {
      "displaced_threshold": null,
      "elevation": 1320,
      "ident": "19L",
      "latitude": 37.6616,
      "longitude": -97.4177,
      "magnetic_heading": 200
    },
    "lighted": true,
    "low_bearing": {
      "displaced_threshold": null,
      "elevation": 1321,
      "ident": "01R",
      "latitude": 37.6428,
      "longitude": -97.4263,
      "magnetic_heading": 20
    }
```

```
    "runway_id": 240443,
    "runway_length": 7301,
    "runway_width": 150,
    "surface": "CON"
  },
  {
    "high_bearing": {
      "displaced_threshold": null,
      "elevation": 1322,
      "ident": "32",
      "latitude": 37.6426,
      "longitude": -97.4292,
      "magnetic_heading": 330
    },
    "lighted": true,
    "low_bearing": {
      "displaced_threshold": null,
      "elevation": 1332,
      "ident": "14",
      "latitude": 37.6575,
      "longitude": -97.4401,
      "magnetic_heading": 150
    },
    "runway_id": 240445,
    "runway_length": 6301,
    "runway_width": 150,
    "surface": "CON"
  }
]
```


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

airport*runway*idents*by*iata_code.n1ql

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

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

airport*runway*idents*by*icao_code.n1ql

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

Both queries will yield the same exact result.

Result

```
[
  {
    "runway": "01L/19R"
  },
  {
    "runway": "01R/19L"
  },
  {
    "runway": "14/32"
  }
]
```

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

airport*with*runway*idents*by*iata*code.n1ql

```
SELECT airports.airport_id, airports.airport_name, airports.airport_type,
    airports.iso_region, airports.municipality,
    IFNULL( airports.airport_iata, airports.airport_icao, airports.airport_ident ) AS ai
    ARRAY
        runway.low_bearing.ident || IFNULL('/' || runway.high_bearing.ident, '')
        FOR runway IN IFMISSING(runways, [])
        WHEN runway.closed = false
    END AS runways
FROM `flight-data` AS codes
USE KEYS 'airport_code_ICT'
INNER JOIN `flight-data` AS airports ON KEYS 'airport_' || TOSTRING( codes.id )
LEFT NEST `flight-data` AS runways ON KEYS (
    ARRAY runway.runway_id FOR runway IN (
        SELECT 'runway_' || TOSTRING( runway_id ) AS runway_id
        FROM `flight-data` AS runway_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_runways'
        UNNEST runway_lookup.runways AS runway_id
    ) END
)
```

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

airport*with*runway*idents*by*icao*code.n1ql

```
SELECT airports.airport_id, airports.airport_name, airports.airport_type,
    airports.iso_region, airports.municipality,
    IFNULL( airports.airport_iata, airports.airport_icao, airports.airport_ident ) AS ai
    ARRAY
        runway.low_bearing.ident || IFNULL('/' || runway.high_bearing.ident, '')
        FOR runway IN IFMISSING(runways, [])
        WHEN runway.closed = false
    END AS runways
FROM `flight-data` AS codes
USE KEYS 'airport_code_KICT'
INNER JOIN `flight-data` AS airports ON KEYS 'airport_' || TOSTRING( codes.id )
LEFT NEST `flight-data` AS runways ON KEYS (
    ARRAY runway.runway_id FOR runway IN (
        SELECT 'runway_' || TOSTRING( runway_id ) AS runway_id
        FROM `flight-data` AS runway_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_runways'
        UNNEST runway_lookup.runways AS runway_id
    ) END
)
```

Both queries will yield the same exact result.

Result


```
[
  {
    "airport_code": "ICT",
    "airport_id": 3605,
    "airport_name": "Wichita Dwight D. Eisenhower National Airport",
    "airport_type": "large_airport",
    "iso_region": "US-KS",
    "municipality": "Wichita",
    "runways": [
      "01R/19L",
      "14/32",
      "01L/19R"
    ]
  }
]
```