Airport Navaid Queries

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

Airport Navaids by Code

Query

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

airport*navaids*by*iata*code.n1ql

```
SELECT navaids.navaid_id, navaids.navaid_ident, navaids.navaid_name, navaids.type,
    navaids.frequency_khz, navaids.geo, navaids.elevation, navaids.usage_type
FROM `flight-data` AS airport_codes
USE KEYS 'airport_code_SLN'
INNER JOIN `flight-data` AS airport_navaids
    ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_navaids'
UNNEST airport_navaids.navaids AS navaids_lookup
INNER JOIN `flight-data` AS navaids
    ON KEYS 'navaid_' || TOSTRING( navaids_lookup )
ORDER BY navaids.navaid_name ASC
```

This query will find the available frequencies by the 4 character ICAO code of the airport

airport*navaids*by*icao*code.n1ql

```
SELECT navaids.navaid_id, navaids.navaid_ident, navaids.navaid_name, navaids.type,
    navaids.frequency_khz, navaids.geo, navaids.elevation, navaids.usage_type
FROM `flight-data` AS airport_codes
USE KEYS 'airport_code_KSLN'
INNER JOIN `flight-data` AS airport_navaids
    ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_navaids'
UNNEST airport_navaids.navaids AS navaids_lookup
INNER JOIN `flight-data` AS navaids
    ON KEYS 'navaid_' || TOSTRING( navaids_lookup )
ORDER BY navaids.navaid_name ASC
```

Both queries will yield the same exact result.

Result


```
[
  {
    "elevation": 1315,
    "frequency_khz": 344,
    "geo": {
      "latitude": 38.68149948,
      "longitude": -97.64510345
    },
    "navaid_id": 93716,
    "navaid_ident": "SL",
    "navaid_name": "Flory",
    "type": "NDB",
    "usage_type": "TERMINAL"
  },
  {
    "elevation": 1310,
    "frequency_khz": 117100,
    "geo": {
      "latitude": 38.92509842,
      "longitude": -97.62139893
    },
    "navaid_id": 93733,
    "navaid_ident": "SLN",
    "navaid_name": "Salina",
    "type": "VORTAC",
    "usage_type": "BOTH"
  }
]
```


Airport Information with Navaids

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 navaids.

Query

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

airport*with*navaids*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
        {
          "elevation": navaid.elevation,
          "frequency_khz": navaid.frequency_khz,
          "geo": navaid.geo,
          "navaid_ident": navaid.navaid_ident,
          "type": navaid.`type`,
          "usage_type": navaid.usage_type
        }
        FOR navaid IN IFMISSING(navaids, [])
    END AS navaids
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 navaids ON KEYS (
    ARRAY navaid.navaid_id FOR navaid IN (
        SELECT 'navaid_' || TOSTRING( navaid_id ) AS navaid_id
        FROM `flight-data` AS navaids_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_navaids'
        UNNEST navaids_lookup.navaids AS navaid_id
    ) END
)
```

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

airport*with*navaids*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
        {
          "elevation": navaid.elevation,
          "frequency_khz": navaid.frequency_khz,
          "geo": navaid.geo,
          "navaid_ident": navaid.navaid_ident,
          "type": navaid.`type`,
          "usage_type": navaid.usage_type
        }
        FOR navaid IN IFMISSING(navaids, [])
    END AS navaids
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 navaids ON KEYS (
    ARRAY navaid.navaid_id FOR navaid IN (
        SELECT 'navaid_' || TOSTRING( navaid_id ) AS navaid_id
        FROM `flight-data` AS navaids_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_navaids'
        UNNEST navaids_lookup.navaids AS navaid_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",
    "frequencies": [
      {
        "frequencies_mhz": 125.7,
        "type": "CLD"
      },
      {
        "frequencies_mhz": 122.2,
        "type": "RDO"
      },
```

```
      {
        "frequencies_mhz": 32.71,
        "type": "APP"
      },
      {
        "frequencies_mhz": 122.95,
        "type": "UNIC"
      },
      {
        "frequencies_mhz": 118.2,
        "type": "TWR"
      },
      {
        "frequencies_mhz": 125.15,
        "type": "ATIS"
      },
      {
        "frequencies_mhz": 125.5,
        "type": "APP"
      },
      {
        "frequencies_mhz": 121.9,
        "type": "GND"
      },
      {
        "frequencies_mhz": 126.7,
        "type": "DEP"
      }
    ],
    "iso_region": "US-KS",
    "municipality": "Wichita"
  }
]
```