Airport Airline Queries

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

Airport Frequencies by Code

Query

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

airport*frequencies*by*iata*code.n1ql

```
SELECT frequencies.frequency_id, frequencies.description, frequencies.frequency_mhz, fre
FROM `flight-data` AS airport_codes
USE KEYS 'airport_code_SLN'
INNER JOIN `flight-data` AS airport_frequencies
    ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_frequencies'
UNNEST airport_frequencies.frequencies AS frequencies_lookup
INNER JOIN `flight-data` AS frequencies
    ON KEYS 'frequency_' || TOSTRING( frequencies_lookup )
ORDER BY frequencies.type ASC
```

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

airport*frequencies*by*icao*code.n1ql

```
SELECT frequencies.frequency_id, frequencies.description, frequencies.frequency_mhz, fre
FROM `flight-data` AS airport_codes
USE KEYS 'airport_code_KSLN'
INNER JOIN `flight-data` AS airport_frequencies
    ON KEYS 'airport_' || TOSTRING( airport_codes.id ) || '_frequencies'
UNNEST airport_frequencies.frequencies AS frequencies_lookup
INNER JOIN `flight-data` AS frequencies
    ON KEYS 'frequency_' || TOSTRING( frequencies_lookup )
ORDER BY frequencies.type ASC
```

Both queries will yield the same exact result.

Result

```
[
```

```
[
  {
    "description": "ATIS",
    "frequency_id": 66133,
    "frequency_mhz": 120.15,
    "type": "ATIS"
  },
  {
    "description": "KANSAS CITY CNTR",
    "frequency_id": 66134,
    "frequency_mhz": 134.9,
    "type": "CNTR"
  },
  {
    "description": "CTAF",
    "frequency_id": 66135,
    "frequency_mhz": 119.3,
    "type": "CTAF"
  },
  {
    "description": "GND",
    "frequency_id": 66136,
    "frequency_mhz": 121.9,
    "type": "GND"
  },
  {
    "description": "ARNG OPS",
    "frequency_id": 66137,
    "frequency_mhz": 49.95,
    "type": "OPS"
  },
  {
    "description": "WICHITA RDO",
    "frequency_id": 66138,
    "frequency_mhz": 122.4,
    "type": "RDO"
  },
  {
    "description": "TWR",
    "frequency_id": 66139,
    "frequency_mhz": 119.3,
    "type": "TWR"
  },
  {
    "description": "UNICOM",
    "frequency_id": 66140,
```

```
    "frequency_mhz": 122.95,
    "type": "UNIC"
  }
]
```

Airport Information with Frequencies

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 frequencies.

Query

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

airport*with*frequencies*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
        {
          "frequencies_mhz": frequency.frequency_mhz,
          "type": frequency.`type`
        }
        FOR frequency IN IFMISSING(frequencies, [])
    END AS frequencies
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 frequencies ON KEYS (
    ARRAY frequency.frequency_id FOR frequency IN (
        SELECT 'frequency_' || TOSTRING( frequency_id ) AS frequency_id
        FROM `flight-data` AS frequencies_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_frequencies'
        UNNEST frequencies_lookup.frequencies AS frequency_id
    ) END
)
```

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

airport*with*frequencies*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
        {
          "frequencies_mhz": frequency.frequency_mhz,
          "type": frequency.`type`
        }
        FOR frequency IN IFMISSING(frequencies, [])
    END AS frequencies
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 frequencies ON KEYS (
    ARRAY frequency.frequency_id FOR frequency IN (
        SELECT 'frequency_' || TOSTRING( frequency_id ) AS frequency_id
        FROM `flight-data` AS frequencies_lookup
        USE KEYS
            'airport_' || TOSTRING(codes.id) || '_frequencies'
        UNNEST frequencies_lookup.frequencies AS frequency_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"
  }
]
```