Airline Queries

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

Airline By ID

The following query will get an Airline by its Document ID.

Query

airline_by_document_id.n1ql

```
SELECT airlines.*
FROM `flight-data` AS airlines
USE KEYS 'airline_2009'
```

Result

```
[
  {
    "_id": "airline_2009",
    "active": true,
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines",
    "callsign": "DELTA",
    "doc_type": "airline",
    "iso_country": "US"
  }
]
```

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

Query

airlines_by_document_id.n1ql

```
SELECT airlines.*
FROM `flight-data` AS airlines
USE KEYS ['airline_2009', 'airline_24']
```

Result


```
[
  {
    "_id": "airline_2009",
    "active": true,
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines",
    "callsign": "DELTA",
    "doc_type": "airline",
    "iso_country": "US"
  },
  {
    "_id": "airline_24",
    "active": true,
    "airline_iata": "AA",
    "airline_icao": "AAL",
    "airline_id": 24,
    "airline_name": "American Airlines",
    "callsign": "AMERICAN",
    "doc_type": "airline",
    "iso_country": "US"
  }
]
```


Airlines in a Country

The following index and queries allows for finding airlines based in a given country by creating an index on the `iso_country` where the `doc_type` is `airline`

Index

idx_airlines_iso_country.n1ql

```
CREATE INDEX idx_airlines_iso_country ON `flight-data`( iso_country, doc_type )
WHERE iso_country IS NOT NULL
    AND doc_type = 'airline'
USING GSI
```

Query

airline_by_country.n1ql

```
SELECT airlines.*
FROM `flight-data` AS airlines
WHERE airlines.iso_country = 'FI'
    AND airlines.doc_type = 'airline'
LIMIT 1
```

Result


```
[
  {
    "_id": "airline_14073",
    "active": false,
    "airline_iata": null,
    "airline_icao": "FN1",
    "airline_id": 14073,
    "airline_name": "Finlandian",
    "callsign": null,
    "doc_type": "airline",
    "iso_country": "FI"
  }
]
```


In the returned results there is an `active` attribute. We will want to update our query account for this, so that only active airlines are returned.

Query

airline_by_country_active.n1ql

```
SELECT airlines.*
FROM `flight-data` AS airlines
WHERE airlines.iso_country = 'FI'
    AND airlines.doc_type = 'airline'
    AND airlines.active = true
LIMIT 1
```

Result


```
[
  {
    "_id": "airline_1427",
    "active": true,
    "airline_iata": "KF",
    "airline_icao": "BLF",
    "airline_id": 1427,
    "airline_name": "Blue1",
    "callsign": "BLUEFIN",
    "doc_type": "airline",
    "iso_country": "FI"
  }
]
```


Now we can retrieve all airlines for a given country. Returning the `airline_id`, `airline_name` and `airline_code`. Some airlines may have an `airline_iata` code, while others may have an `airline_icao` code, and some may have both. To normalize these values into a single `airline_code` we will use the `IFNULL` conditional statement

Query

airlines_by_country_active.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    IFNULL( airlines.airline_iata, airlines.airline_icao ) AS airline_code
FROM `flight-data` AS airlines
WHERE airlines.iso_country = 'AE'
    AND airlines.doc_type = 'airline'
    AND airlines.active = true
ORDER BY airlines.airline_name ASC
```

Result

```
[
  {
    "airline_code": "MO",
    "airline_id": 502,
    "airline_name": "Abu Dhabi Amiri Flight"
  },
  {
    "airline_code": "G9",
    "airline_id": 329,
    "airline_name": "Air Arabia"
  },
  {
    "airline_code": "8L",
    "airline_id": 2942,
    "airline_name": "Cargo Plus Aviation"
  },
  {
    "airline_code": "EK",
    "airline_id": 2183,
    "airline_name": "Emirates"
  },
  {
    "airline_code": "EY",
    "airline_id": 2222,
    "airline_name": "Etihad Airways"
  },
  {
    "airline_code": "FZ",
    "airline_id": 14485,
    "airline_name": "Fly Dubai"
  }
]
```


Airline Codes

Each Airline has 2 identifying codes a 2 character IATA / FAA Code and a 3 character ICAO code. Each of these attributes are stored as separate attributes on the airlines document as `airline_iata` and `airline_icao`.

Index

idx_airlines_codes.n1ql

```
CREATE INDEX idx_airlines_codes ON `flight-data`( airline_iata, airline_icao, doc_type )
WHERE (
        airline_iata IS NOT NULL
        OR
        airline_icao IS NOT NULL
    )
    AND doc_type = 'airline'
USING GSI
```

Query

airline_by_iata_or_icao.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS airlines
WHERE (
        airlines.airline_iata = 'DL'
        OR (
            airlines.airline_iata <> 'DL'
            AND
            airlines.airline_icao = 'DL'
        )
    )
    AND airlines.doc_type = 'airline'
LIMIT 1
```

Results

```
[
  {
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines"
  }
]
```

This works but is slow because of the `OR` statement that we have to use to attempt to match either code. We can improve this by creating 2 separate indexes on each code the query using a `UNION` statement.

Index

Drop the index we just created, since it will no longer be used.

idx_airlines_codes_drop.n1ql

```
DROP INDEX `flight-data`.idx_airlines_codes
```

Create index for Airline IATA codes

idx_airlines_iata_codes.n1ql

```
CREATE INDEX idx_airlines_iata_codes ON `flight-data`( airline_iata, doc_type )
WHERE airline_iata IS NOT NULL
    AND doc_type = 'airline'
USING GSI
```

Create index for Airline ICAO codes

idx_airlines_icao_codes.n1ql

```
CREATE INDEX idx_airlines_icao_codes ON `flight-data`( airline_icao, doc_type )
WHERE airline_icao IS NOT NULL
    AND doc_type = 'airline'
USING GSI
```

Query

airline_by_iata_union_icao.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS airlines
WHERE airlines.airline_iata = 'DL'
    AND airlines.doc_type = 'airline'
UNION
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS airlines
WHERE airlines.airline_icao = 'DL'
    AND airlines.doc_type = 'airline'
LIMIT 1
```

Results

```
[
  {
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines"
  }
]
```

This performs much better as we are now using 2 different indexes for the IATA and ICAO codes. However, we can improve this query even more. One of our data models Codes is a lookup document for Airline, Airport, and Navaid IATA, ICAO and Ident Codes. We can query the codes, then when a matching code is found join back to its parent document.

Index

Drop the previously created indexes as they will no longer be used.

idx_airlines_iata_codes_drop.n1ql

```
DROP INDEX `flight-data`.idx_airlines_iata_codes
```

idx_airlines_icao_codes_drop.n1ql

```
DROP INDEX `flight-data`.idx_airlines_icao_codes
```

Create a new index on the `code` and `designation` attributes where the `doc_type = 'code'`

idx_codes.n1ql

```
CREATE INDEX idx_codes ON `flight-data`( code, designation, doc_type )
WHERE doc_type = 'code'
USING GSI
```

Query

Query by the IATA code

airline_by_iata_designation.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS codes
INNER JOIN `flight-data` AS airlines
    ON KEYS 'airline_' || TOSTRING( codes.id )
WHERE codes.code = 'DL'
     AND codes.designation = 'airline'
     AND codes.doc_type = 'code'
LIMIT 1
```

Query by the ICAO code

airline_by_icao_designation.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS codes
INNER JOIN `flight-data` AS airlines
    ON KEYS 'airline_' || TOSTRING( codes.id )
WHERE codes.code = 'DAL'
     AND codes.designation = 'airline'
     AND codes.doc_type = 'code'
LIMIT 1
```

Results

Both queries will yield the same exact result.

```
[
  {
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines"
  }
]
```

Our Codes model is keyed by `{{designation}}_code_{{code}}` i.e. `airline_code_DL`. Because of how these documents are keyed, we do not even need an index. Using this predictive key pattern the code is used as part of the key name on the codes document.

Index

Drop the previously created index, as it will no longer be used.

idx_codes_drop.n1ql

```
DROP INDEX `flight-data`.idx_codes
```

Query

Query by the IATA code

airline*by*iata_code.n1ql

```
    SELECT airlines.airline_id, airlines.airline_name,
        airlines.airline_iata, airlines.airline_icao
    FROM `flight-data` AS codes
    USE KEYS 'airline_code_DL'
    INNER JOIN `flight-data` AS airlines
        ON KEYS 'airline_' || TOSTRING( codes.id )
    LIMIT 1
```

Query by the ICAO code

airline*by*icao_code.n1ql

```
SELECT airlines.airline_id, airlines.airline_name,
    airlines.airline_iata, airlines.airline_icao
FROM `flight-data` AS codes
USE KEYS 'airline_code_DAL'
INNER JOIN `flight-data` AS airlines
   ON KEYS 'airline_' || TOSTRING( codes.id )
LIMIT 1
```

Results

```
[
  {
    "airline_iata": "DL",
    "airline_icao": "DAL",
    "airline_id": 2009,
    "airline_name": "Delta Air Lines"
  }
]
```