### Example Family JSON Documents

```json
{  
  "id":"AndersenFamily",
  "lastName":"Andersen",
  "parents": [
    {  
      "firstName":"Henriette Thaulow",
      "gender":"female",
      "grade":5,
      "pets": [  
        {  
          "givenName": "Fluffy"
        }
      ]
    },
    {  
      "firstName":"Mary Kay",
      "gender":"female",
      "grade":4,
      "pets": [  
        {  
          "givenName": "Goofy"
        },
        {  
          "givenName": "Shadow"
        }
      ]
    }
  ],
  "children": [  
    {  
      "firstName": "Jesse",
      "givenName": "Jesse",
      "gender": "male",
      "grade": 8,
    },
    {  
      "firstName": "Ben",
      "givenName": "Ben",
      "gender": "male",
      "grade": 8,
    }
  ],
  "city": "NY",
  "county": "Manhattan",
  "address": {},
  "gender": "female",
  "givenName": "Lisa",
  "familyName": "Miller",
  "pets": [],
  "creationDate": "2015-07-20T12:00Z",
  "isRegistered": true
}
```

```json
{  
  "id":"WakefieldFamily",
  "parents": [  
    {  
      "firstName": "Wakefield",
      "familyName": "Wakefield",
      "gender": "male",
      "grade": 5,
      "isRegistered": true
    }
  ],
  "children": [  
    {  
      "givenName": "Jesse",
      "gender": "male",
      "grade": 5,
    },
    {  
      "givenName": "Ben",
      "gender": "male",
      "grade": 8,
    },
    {  
      "givenName": "Robin",
      "familyName": "Wakefield",
      "gender": "female",
      "grade": 5,
    }
  ],
  "city": "seattle",
  "county": "King",
  "state": "WA",
  "address": {},
  "gender": "female",
  "givenName": "Lisa",
  "familyName": "Miller",
  "pets": [],
  "creationDate": "2015-01-03T12:00Z",
  "isRegistered": true
}
```

### SQL Query

**-- Find Families by ID**

```sql
SELECT * FROM Families f WHERE f.id = "AndersenFamily"
```

**-- Find families where City equals State and return Name and City**

```sql
SELECT \{"FirstName":f.firstName, \"City":f.address.city\} AS Family FROM Families f WHERE f.address.city = f.address.state
```

**SQL + JSON**

```sql
-- Get the child names using an intra-document JOIN
SELECT c.givenName FROM Families f JOIN c IN f.children WHERE f.id = 'WakefieldFamily' ORDER BY f.address.city ASC
```

**SQL + JavaScript UDF**

```sql
-- Register UDF for REGEX_MATCH with this code
function (input, pattern) {
  return input.match(pattern) !== null;
}
```

```sql
-- Use JavaScript
SELECT udf.REGEX_MATCH(Families.address.city, ".*eattle")
```

### Sample Queries

**Comparison (range) operators**

```sql
SELECT * FROM Families.children[8] c WHERE c.grade >= 5 AND c.isRegistered = true
```

**Logical operators**

```sql
SELECT * FROM Families.children[8] c WHERE c.grade >= 5 AND c.isRegistered = true
```

**ORDER BY keyword**

```sql
SELECT f.id, f.address.city FROM Families f ORDER BY f.address.city
```

**IN keyword**

```sql
SELECT * FROM Families WHERE Families.address.state IN ("NY", "WA", "CA", "PA", "OH", "OR", "MI", "WI")
```

**Terminy (?) and Coalesce (?) operators**

```sql
SELECT c.givenName FROM Families f JOIN c IN f.children
```

**Escape/quoted accessor**

```sql
SELECT \"Last\"name\" FROM Families f WHERE f.id = "AndersenFamily"
```

**Object/Array Creation**

```sql
SELECT \{f.address.city, f.address.state\} AS CityState FROM Families f
```

**Value keyword**

```sql
SELECT VALUE \"Hello World\"
```

**Intra-document JOINS**

```sql
SELECT c.givenName FROM Families f JOIN c IN f.children WHERE f.id = 'WakefieldFamily' ORDER BY f.address.city ASC
```

**Parameterized SQL**

```sql
SELECT * FROM Families f WHERE f.lastName = @lastName AND f.address.state = @addressState
```

**String Built-in functions**

```sql
SELECT Families.id, Families.address.city FROM Families WHERE STARTSWITH(Families.id, "Wakefield")
```

**Array Built-in functions**

```sql
SELECT Families.id FROM Families WHERE ARRAY_CONTAINS(Families.parents, {  
  \"givenName\": "Robin",
  \"familyName\": "Wakefield"
})
```

**Math Built-in functions**

```sql
SELECT VALUE ABS(-4)
```

**Type Built-in functions**

```sql
SELECT IS_DEFINED(f.lastName), IS_NUMBER(f.givenName), IS_STRING(p.firstName), IS_BOOLEAN(p.givenName)
```

**BETWEEN keyword**

```sql
SELECT * FROM Families.children[8] c WHERE c.grade BETWEEN 1 AND 5
```

**TOP keyword**

```sql
SELECT TOP 10 * FROM Families f
```

**Geospatial functions**

```sql
SELECT * FROM Families WHERE ST_DISTANCE(f.location, ("type":"Point", \"coordinates\":\[31.9, -4.8\])) < 30000
```