NosDB
NoSQL Document Database for .NET

NosDB SQL Query Language Cheat Sheet

This cheat sheet provides references to commonly used SQL queries to manage and access data in NosDB as simple JSON documents.

Example Group JSON Documents

```
{
   "id":"Group0001",
   "name":"Peterson",
   "members":[
      {
         "firstName":"Jacques"
      },
      {
         "firstName":"Markus"
      },
      {
         "firstName":"Samantha",
         "gender":"female",
         "toys":[
            {
               "givenName":"Barbie"
            }
         ]
      }
   ],
   "address":{
      "state":"PA",
      "county":"New Castle",
      "city":"Massachusets"
   },
   "creationDate":"2015-01-03T12:00Z",
   "isRegistered":true,
   "location":{
      "type":"Point",
      "coordinates":[
         31.9,
         -4.8
      ]
   }
},
{
   "id":"Group0002",
   "members":[
      {
         "firstName":"Smith"
      },
      {
         "firstName":"Jade"
      },
      {
         "firstName":"Jolie",
         "gender":"female",
         "toys":[
            {
               "givenName":"Barbie"
            }
         ]
      },
      {
         "firstName":"Timothy",
         "gender":"male",
         "toys":[
            {
               "givenName":"Goofy"
            },
            {
               "givenName":"Lisa"
            }
         ]
      }
   ],
   "address":{
      "state":"CL",
      "county":"Colorado",
      "city":"Colorado"
   },
   "creationDate":"2015-07-20T12:00Z",
   "isRegistered":false
}
```

SQL SELECT Query

```
-- Get all Group(s) which has id equals to
'Group0001'.

SELECT * FROM GROUPS WHERE id = 'Group0001'
```

```
[
  {
    "id": "Group0001",
    "name": "Peterson", ...
  }
]
```

SQL SELECT Query + JSON

```
-- Get items(s)by JSON.

SELECT *
FROM Groups
WHERE address =
{
  "state":"CL",
  "county":"Colorado",
  "city":"Colorado"
}
```

```
[
  {
    "id": "Group0002",
    "members":..., ...
  }
]
```

SQL Query + UDF in .NET

```
-- UDF definition:
function (input, pattern)
{    return Regex.IsMatch(input, pattern); }

SELECT
function(Families.address.city, ".*eattle")
FROM "Groups"
```

```
[
  {
    "function(input, pattern)": true
  },
  {
    "function(input, pattern)": false
  }
]
```

Operators

Arithmetic	+, -, *, /, %
Logical	AND, OR, NOT
Comparison	=, !=, >, >=, <, <=, <>
String	+ (concatenate)

SQL Insert Query

```
-- Inserting a simple document into collection
"Groups".

INSERT INTO Groups ("id","lastName")
VALUES ("Group0001","Peterson")
```

```
1 document inserted into the "Groups"
collection.
```

SQL Insert + JSON Child

```
-- Inserting a document which has a JSON
document as a child.

INSERT INTO Groups
("id","lastName","address")
VALUES
(
   "Group0001", "Peterson",
   {
     "state":"PA",
     "county":"New Castle",
     "city":"Massachusetts"
   }
)
```

```
1 document inserted into the "Groups"
collection.
```

SQL Insert + JSON Array

```
-- Inserting a JSON document which has JSON
document in a JSON array.

INSERT INTO "Groups"
( "id","lastName","members" )
VALUES
(
  "Group0001", "Peterson",
   [
    {
      "firstName":"Jacques"
    },
    {
      "firstName":"Samantha",
      "gender":"female",
      "toys":
       [
        {
          "givenName":"Barbie"
        }
       ]
     }
    ]
 )
```

```
1 document inserted into the "Groups"
collection.
```

Built-in Functions (Case insensitive)

Mathematics	avg, sum, round
String	lcase, ucase, len, mid, format
Date Time	Now
Type Indifferent	Max, Min, Count, First, Last
Custom Methods	Define UDF in .NET

SQL Update Query

```
-- Adding/updating "passports"=null in all
documents

UPDATE Groups SET ("Passports"= null)
```

```
2 affected documents ("passports":null added
in both).
```

www.alachisoft.com | Tweet: @nosdb | Download this cheat sheet from Alachisoft Website

Alachisoft®

NosDB
NoSQL Document Database for .NET

NosDB SQL Query Language Cheat Sheet

This cheat sheet provides references to commonly used SQL queries to manage and access data in NosDB as simple JSON documents.

Data Update Query + Filter

```
-- Deleting the key "Passports" where ID
Exists.

UPDATE Groups SET (DELETE "Passports")
WHERE id EXISTS
```

```
2 affected documents. "Passports" deleted
from both.
```

SQL UPDATE Addition in an Array

```
-- Adding a pet in childrens' pets if
the family is registered.

UPDATE Groups SET
(
 children.pets ADD
 (
  {
   "givenname": "Diesel"
  }
 )
)
WHERE isRegistered = false
```

```
1 affected document.
```

Update Options

EQUALS (=)	Replaces the value of a key if exists, else adds the key-value pair in the document.
ADD	Adds the value(s) given at the end of the array existing against an attribute.
INSERT	Adds the value(s) at the end of array if they do not exist in the array against an attribute.
REMOVE	Removes the given value(s) from the array against an attribute.
REPLACE	Replaces given values by their pairs given in the array existing against an attribute.
RENAME TO	Renames a given attribute in the document to the specified name.
DELETE	Deletes the specified attribute from the document if exists.

SQL Delete + Filter

```
Deleting the registered group.

DELETE FROM "Families" WHERE isRegistered =
false
```

```
1 affected document.
```

Sample SELECT Queries

Comparison (range) operators	`SELECT *`
	`FROM Groups`
	`WHERE children.grade >= 5`

Logical operators	`SELECT *`
	`FROM Groups`
	`WHERE children.grade >= 5 AND isRegistered = true`

ORDER BY keyword	`SELECT id, address.city`
	`FROM Groups`
	`ORDER BY address.city`

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

Constant Evaluation	`SELECT 1+2 AS NumberThree`
	`FROM Groups`

Parameterized SQL	`SELECT *`
	`FROM Groups`
	`WHERE lastName = @lastName AND address.state = @addressState`

String Built-in functions	`SELECT Families.id, address.city`
	`FROM Groups`
	`WHERE STRLEN(Families.id) == 5`

Exists Keyword	`SELECT *`
	`FROM Groups`
	`WHERE grade EXISTS`

Array Projection	`SELECT (location.coordinates) SLICE (0,1)`
	`FROM Groups`
	`WHERE grade EXISTS`

Delimited Identifiers	`SELECT *`
	`FROM $Groups$`
	`WHERE "grade" EXISTS`

Embedded Attribute with indexer	`SELECT *`
	`FROM Groups`
	`WHERE members.toys[0].givenname = 'Lisa'`

Sample DML Queries

Parameterized Insert

```
INSERT INTO Groups
("Name","Collection")
VALUES (@name, @array)
```

Inserting Custom DateTime

```
INSERT INTO Groups
("Name","DateTime")
VALUES ('Josh', DateTime('12/30/2011'))
```

Replacing items in an array

```
UPDATE Groups
SET (Collection REPLACE (1=3, 4=5, 7=10))
```

Adding items in an array

```
UPDATE Groups
SET (
 Collection ADD
  (
   {
     "id":1
   },
   2
  )
)
```

Removing items from an array

```
UPDATE Groups
SET (Collection REMOVE
 (
  {
    "id":1
  }
 )
)
```

Renaming an Attribute

```
UPDATE Groups
SET (RENAME Collection TO 'Items')
```

Alachisoft®