

Commands

- ☐ create : To create nodes.
- ☐ match : To find nodes.
- ☐ return : To display nodes.
- ☐ return distinct: To display unique nodes.
- ☐ where : To apply conditions.
- ☐ delete : To delete nodes.
- ☐ limit : To limit the number of nodes returned.
- ☐ id(n) : To get id of a node n.
- ☐ in : To check if a value in a list of values. ([1, 3, 4])
- ☐ or : Well, its OR.
- ☐ set : To update labels and properties in nodes.
- ☐ remove : To delete particular label or property from a node.
- ☐ labels(n) : To get labels-set of nodes n.
- ☐ toInt(n.value) : To typecast a property to integer.
- ☐ += : To update existing properties (used in set clause).
- ☐ exists : To see if a property exists.

Examples

Create Nodes

- ☒ create ()
- ☒ create (), ()
- ☒ create (:Student)
- ☒ create (:Student:Person)
- ☒ create ({name:"Manas"})
- ☒ create ({`first name`:"Manas"})
- ☒ create ({name:"Lakshya",hobby:"Dance"})
- ☒ create (:Student{name:"Manas"})
- ☒ create (divya:Student{name:"Divya"})

Labels and properties are case sensitive.

Match

Match must end with a return, set or delete clause. just append a return statement at the end of each statement.

- ☒ match (n)
- ☒ match (n:Student)
- ☒ match (n{name:"Manas"})
- ☒ match (n:Student:Person)
- ☒ match (n:Student{name:"Manas"})
- ☒ match (ma{name:"Manas"}, (di{name:"Divya"}) return ma, di

Create Relationships

First create variables a, b by using match commands. Exactly one label can be applied to a relationship (called 'type'). We cannot create bi-directional relationships using 'create'

- ☒ create (a)-[:LIKES]->(b)
- ☒ create (a)-[:HELPS]-(b)
- ☒ create (a)-[:FRIEND{since:"college", nickname:"callback"}]->(b)

Matching Relationships

- ☒ match (a)-[r]->(b)
- ☒ match (a)-[r:FRIEND]-(b)

Where

Prepend 'match (n)' and append 'return n'.

- ☒ where n:Person
- ☒ where n.name="Manas"
- ☒ where n:Person:Student
- ☒ where n:Person or n:Student
- ☒ where n.price < 1000
- ☒ where n.name in ["Manas", "Lakshya"]
- ☒ where n:Student and n.name="Manas"

Set

Prepend a match-clause (for n) and append return-clause.

- ☒ set n:Human
- ☒ set n.age=25
- ☒ set n:Animal:Atoms
- ☒ set n.age=25,n.name="Someone"
- ☒ set n += {age:26}
- ☒ set n1=n2 count

Remove

Prepend match-clause

- ☒ remove n:Student
- ☒ remove n:Student:Person

Misc

- ☒ match (n) return distinct labels(n)
- ☒ match (n) return count(distinct labels(n))
- ☒ match (n) return distinct labels(n),count(labels(n))

Property DataTypes

1. boolean
2. int
3. double
4. long
5. char
6. string
7. Array of primitives

NULL is not a valid value of any property.

Type	Description	Range of Values
Boolean		true/false
byte or byte[]	A single byte or array of bytes. It is an 8-byte Integer.	-128 to 127, inclusive of both ends.
short or short[]	A single short or array of shorts. It is a 16-byte Integer.	-32768 to 32767, inclusive of both ends.
int or int[]	A single integer or array of integers. It is a 32-byte Integer.	-2147483648 to 2147483647, inclusive of both ends.
long or long[]	A single long or array of longs. It is a 64-byte Integer.	-9223372036854775808 to 9223372036854775807, inclusive of both ends.
float or float[]	32-bit IEEE 754 floating-point number.	
double or double[]	64-bit IEEE 754 floating-point number.	
char or char[]	16-bit unsigned integers representing Unicode characters.	0 to 65535
String or String[]	It is a sequence of Unicode characters.	

Examples

- ✓ create ({title:"A string", names:["An", "array", "of", "strings"], price:10.5, pages:321, available:false})
- ✓ create ({an_array:["A string", 22]})

Array must be homogeneous.

Some things that may be needed in future

Importing CSV Data

Method 1

1. To enable neo4j to access filesystem to import csv files, in /etc/neo4j/neo4j.conf
 - dbms.security.allow_csv_import_from_file_urls=true
 - #dbms.directories.import=import
 - Then restart neo4j service.

```
id,company,last_name,first_name,email_address,job_title,business_phone,home_phone,mobile_phone,fax_number,address,city,state_province,zip_postal
1,"Northwind Traders","Freehafer","Nancy","nancy@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 1st Avenue","Seattle","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#",\N,""
2,"Northwind Traders","Cencini","Andrew","andrew@northwindtraders.com","Vice President, Sales","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 2nd Avenue","Bellevue","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Joined the
company as a sales representative, was promoted to sales manager and was then named vice president of sales."
3,"Northwind Traders","Kotas","Jan","jan@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",\N,"(123)555-0103","123
3rd Avenue","Redmond","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Was hired as a sales associate and was
promoted to sales representative."
4,"Northwind Traders","Sergienko","Mariya","mariya@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 4th Avenue","Kirkland","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#",\N,""
5,"Northwind Traders","Thorpe","Steven","steven@northwindtraders.com","Sales Manager","(123)555-0100","(123)555-0102",\N,"(123)555-0103","123
5th Avenue","Seattle","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Joined the company as a sales
representative and was promoted to sales manager. Fluent in French."
6,"Northwind Traders","Neipper","Michael","michael@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 6th Avenue","Redmond","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Fluent in
Japanese and can read and write French, Portuguese, and Spanish."
7,"Northwind Traders","Zare","Robert","robert@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 7th Avenue","Seattle","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#",\N,""
8,"Northwind Traders","Glussani","Laura","laura@northwindtraders.com","Sales Coordinator","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 8th Avenue","Redmond","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Reads and writes
French."
9,"Northwind Traders","Hellung-Larsen","Anne","anne@northwindtraders.com","Sales Representative","(123)555-0100","(123)555-0102",
\N,"(123)555-0103","123 9th Avenue","Seattle","WA","99999","USA","http://northwindtraders.com#http://northwindtraders.com/#","Fluent in French
and German."
```

1. load csv with headers from "file:///home/path/to/csv/file.csv" as row create
(:Employee{employeeID:toInt(row.id), firstName:row.first_name, lastName:row.last_name,
title:row.job_title});

Method 2 (more secure)

1. To enable neo4j to access filesystem to import csv files, in /etc/neo4j/neo4j.conf
 - #dbms.security.allow_csv_import_from_file_urls=true
 - dbms.directories.import=/var/lib/neo4j/import
 - Then restart the neo4j service.

You can modify the path to your preferred directory also.

2. Place your csv file into /var/lib/neo4j/import directory.
3. Execute same command as above method (just write the filename.csv of the file inside import directory).

Neo4j Specific

- ☐ To switch databases
 - ☐ :use [database-name]