



**FREE eBook**

# LEARNING

---

## neo4j

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#neo4j**

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with neo4j.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Installation & Starting a Neo4j server.....	2
<b>Start Neo4j from console (headless, without web server).....</b>	<b>2</b>
<b>Start Neo4j web server.....</b>	<b>3</b>
<b>Start Neo4j web server.....</b>	<b>3</b>
<b>Delete one of the databases.....</b>	<b>3</b>
Cypher Query Language.....	3
RDBMS Vs Graph Database.....	4
<b>Chapter 2: Cypher.....</b>	<b>6</b>
Introduction.....	6
Examples.....	6
Creation.....	6
<b>Create a node.....</b>	<b>6</b>
<b>Create a relationship.....</b>	<b>6</b>
<b>Query Templates.....</b>	<b>6</b>
Create an Edge.....	6
Deletion.....	6
<b>Delete all nodes.....</b>	<b>6</b>
<b>Delete all nodes of a specific label.....</b>	<b>7</b>
Match (capture group) and link matched nodes.....	7
Update a Node.....	7
Delete All Orphan Nodes.....	7
<b>Chapter 3: Python.....</b>	<b>8</b>
Examples.....	8
Install neo4jrestclient.....	8

Connect to neo4j.....	8
Create some nodes with labels.....	8
<b>You can associate a label with many nodes in one go.....</b>	<b>8</b>
Create relationships.....	8
<b>Bi-directional relationships.....</b>	<b>8</b>
Match using neo4jrestclient.....	8
<b>"db" as defined above.....</b>	<b>9</b>
<b>Print results.....</b>	<b>9</b>
<b>Output:.....</b>	<b>9</b>
<b>Credits.....</b>	<b>10</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [neo4j](#)

It is an unofficial and free neo4j ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official neo4j.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with neo4j

## Remarks

This section provides an overview of what neo4j is, and why a developer might want to use it.

It should also mention any large subjects within neo4j, and link out to the related topics. Since the Documentation for neo4j is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Go to [Install Neo4j](#) which should detect the OS platform via your web browser, download and follow the usual installation instructions for your OS.

Neo4j was created with Java, therefore will run on any platform with Java installed, however the Neo4j team has simplified installation by providing easy installation packages for popular platform (e.g. a .dmg for Mac, a .deb for Debian and Ubuntu, an .exe for Windows 64 and 32 bit platforms...).

To review other versions and platforms available, see [Other Neo4j Releases Page](#)

### Setup Neo4j as a Docker container :

```
## Required : Docker machine, docker cli

# Pull neo4j image from the docker hub
docker pull neo4j

# create the docker container
docker run \
  --publish=7474:7474 --publish=7687:7687 \
  --volume=$HOME/neo4j/data:/data \
  neo4j

# If you are running docker directly on the host (e.g ubuntu, RHEL, CentOS etc)
#   Access the neo4j console at http://localhost:7474
# If you are on OSX/ Windows
#   Access the neo4j console at http://<docker-machine-ip>:7474
```

## Installation & Starting a Neo4j server

Prerequisite steps:

- Install Java at your machine
- Visit [neo4j website](#) and click the link "Download Community Edition" or visit directly the [download link](#).
- Unzip the .tar downloaded file in your home directory

---

## Start Neo4j from console (headless, without web server)

- Visit the sub-directory `/bin` of the extracted folder and execute in terminal `./neo4j console`
- You can now execute neo4j queries in the terminal

---

## Start Neo4j web server

- Visit the sub-directory `/bin` of the extracted folder and execute in terminal `./neo4j start`
- Visit <http://localhost:7474/>
- Only the first time, you will have to sign in with the default account and change the default password. As of community version 3.0.3, the default username and password are neo4j and neo4j.
- You can now insert Neo4j queries in the console provided in your web browser and visually investigate the results of each query.

---

## Start Neo4j web server

Each Neo4j server currently (in the community edition) can host a single Neo4j database, so in order to setup a new database:

- Visit sub-directory `/bin` and execute `./neo4j stop` to stop the server
- Visit the sub-directory `/conf` and edit the file `neo4j.conf`, changing the value of the parameter `dbms.active_database` to the name of the new database that you want to create.
- Visit again the sub-directory `/bin` and execute `./neo4j start`
- The web server has started again with the new empty database. You can visit again <http://localhost:7474/> to work with the new database.
- The created database is located in the sub-directory `/data/databases`, under a folder with the name specified in the parameter `dbms.active_database`.

---

## Delete one of the databases

- Make sure the Neo4j server is not running; go to sub-directory `/bin` and execute `./neo4j status`. If the output message shows that the server is running, also execute `./neo4j stop`.
- Then go to sub-directory `/data/databases` and delete the folder of the database you want to remove.

### Cypher Query Language

This is the Cypher, Neo4j's query language. In many ways, Cypher is similar to SQL if you are

familiar with it, except SQL refers to items stored in a table while Cypher refers to items stored in a graph.

First, we should start out by learning how to create a graph and add relationships, since that is essentially what Neo4j is all about.

```
CREATE (ab:Object { age: 30, destination: "England", weight: 99 })
```

- You use **CREATE** to create data
- To indicate a node, you use parenthesis: ()
- The ab:Object part can be broken down as follows: a variable 'ab' and label 'Object' for the new node. Note that the variable can be anything, but you have to be consistent in a line of Cypher Query
- To add properties to the node, use brackets: {} brackets

Next, we will learn about finding MATCHes

```
MATCH (abc:Object) WHERE abc.destination = "England" RETURN abc;
```

MATCH specifies that you want to search for a certain node/relationship pattern (abc:Object) refers to one node Pattern (with label Object) which store the matches in the variable abc. You can think of this entire line as the following

```
abc = find the matches that is an Object WHERE the destination is England.
```

In this case, WHERE adds a constraint which is that the destination must be England. You must include a return at the end for all MATCH queries (neo4j will not accept just a Match...your query must always return some value [this also depends on what type of query you are writing...we will talk more about this later as we introduce the other types of queries you can make].

The next line will be explained in the future, after we go over some more elements of the Cypher Query Language. This is to give you a taste of what we can do with this language! Below, you will find an example which gets the cast of movies whose title starts with 'T'

```
MATCH (actor:Person)-[:ACTED_IN]->(movie:Movie)
WHERE movie.title STARTS WITH "T"
RETURN movie.title AS title, collect(actor.name) AS cast
ORDER BY title ASC LIMIT 10;
```

A complete list of commands and their syntax can be found at the official [Neo4j Cypher Reference Card](#) here.

## RDBMS Vs Graph Database

RDBMS	Graph Database
Tables	Graphs

RDBMS	Graph Database
Rows	Nodes
Columns and Data	Properties and its values
Constraints	Relationships
Joins	Traversal

Read **Getting started with neo4j** online: <https://riptutorial.com/neo4j/topic/945/getting-started-with-neo4j>



---

# Chapter 2: Cypher

## Introduction

Cypher is the query language used by Neo4j. You use Cypher to perform tasks and matches against a Neo4j Graph.

Cypher is "inspired by SQL" and is designed to be intuitive in the way you describe the relationships, i.e. typically the drawing of the pattern will look similar to the Cypher representation of the pattern.

## Examples

### Creation

---

## Create a node

```
CREATE (neo:Company) //create node with label 'Company'

CREATE (neo:Company {name: 'Neo4j', hq: 'San Mateo'}) //create node with properties
```

---

## Create a relationship

```
CREATE (beginning_node)-[:edge_name{Attribute:1, Attribute:'two'}]->(ending_node)
```

---

## Query Templates

Running neo4j locally, in the browser GUI (default: <http://localhost:7474/browser/>), you can run the following command to get a palette of queries.

```
:play query template
```

This helps you get started creating and merging nodes and relationships by typing queries.

### Create an Edge

```
CREATE (beginning_node)-[:edge_name{Attribute:1, Attribute:'two'}]->(ending_node)
```

### Deletion

---

# Delete all nodes

```
MATCH (n)
DETACH DELETE n
```

DETACH doesn't work in older versions (less than 2.3), for previous versions use

```
MATCH (n)
OPTIONAL MATCH (n)-[r]-()
DELETE n, r
```

---

## Delete all nodes of a specific label

```
MATCH (n:Book)
DELETE n
```

### Match (capture group) and link matched nodes

```
Match (node_name:node_type {}), (node_name_two:node_type_two {})
CREATE (node_name)-[:edge_name{}]->(node_name_two)
```

### Update a Node

```
MATCH (n)
WHERE n.some_attribute = "some identifier"
SET n.other_attribute = "a new value"
```

### Delete All Orphan Nodes

Orphan nodes/vertices are those lacking all relationships/edges.

```
MATCH (n)
WHERE NOT (n)--()
DELETE n
```

Read Cypher online: <https://riptutorial.com/neo4j/topic/3669/cypher>

---

# Chapter 3: Python

## Examples

### Install neo4jrestclient

```
pip install neo4jrestclient
```

### Connect to neo4j

```
from neo4jrestclient.client import GraphDatabase
db = GraphDatabase("http://localhost:7474", username="neo4j", password="mypass")
```

### Create some nodes with labels

```
user = db.labels.create("User")
u1 = db.nodes.create(name="user1")
user.add(u1)
u2 = db.nodes.create(name="user2")
user.add(u2)
```

---

## You can associate a label with many nodes in one go

```
Language = db.labels.create("Language")
b1 = db.nodes.create(name="C++")
b2 = db.nodes.create(name="Python")
beer.add(b1, b2)
```

### Create relationships

```
u1.relationships.create("likes", b1)
u1.relationships.create("likes", b2)
u2.relationships.create("likes", b1)
```

---

## Bi-directional relationships

```
u1.relationships.create("friends", u2)
```

### Match using neo4jrestclient

```
from neo4jrestclient import client

q = 'MATCH (u:User)-[r:likes]->(m:language) WHERE u.name="Marco" RETURN u, type(r), m'
```

---

## "db" as defined above

```
results = db.query(q, returns=(client.Node, str, client.Node))
```

---

## Print results

```
for r in results:
    print("(%s)-[%s]->(%s)" % (r[0]["name"], r[1], r[2]["name"]))
```

---

## Output:

```
(Marco)-[likes]->(C++)
(Marco)-[likes]->(Python)
```

Read Python online: <https://riptutorial.com/neo4j/topic/5990/python>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with neo4j	<a href="#">Andrew Lank</a> , <a href="#">Community</a> , <a href="#">cs_user2017</a> , <a href="#">Dimos</a> , <a href="#">Frank Pavageau</a> , <a href="#">Prosen Ghosh</a> , <a href="#">TinkerBotFoo</a> , <a href="#">wintersolider</a>
2	Cypher	<a href="#">frant.hartm</a> , <a href="#">Govind Singh</a> , <a href="#">JOG</a> , <a href="#">Liam</a> , <a href="#">Nicole White</a> , <a href="#">Renato Dinhani</a> , <a href="#">SerialDev</a> , <a href="#">TJ Walker</a>
3	Python	<a href="#">SerialDev</a>