



EBook Gratuito

APPENDIMENTO

neo4j

Free unaffiliated eBook created from
Stack Overflow contributors.

#neo4j

Sommario

Di.....	1
Capitolo 1: Iniziare con neo4j	2
Osservazioni.....	2
Examples.....	2
Installazione o configurazione.....	2
Installazione e avvio di un server Neo4j.....	2
Avvia Neo4j dalla console (senza testa, senza server web)	3
Avvia il server Web Neo4j	3
Avvia il server Web Neo4j	3
Elimina uno dei database	3
Cypher Query Language.....	4
RDBMS Vs Graph Database.....	5
Capitolo 2: cifra	6
introduzione.....	6
Examples.....	6
Creazione.....	6
Crea un nodo	6
Crea una relazione	6
Modelli di query	6
Crea un bordo.....	6
cancellazione.....	6
Elimina tutti i nodi	6
Elimina tutti i nodi di un'etichetta specifica	7
Abbina (cattura gruppo) e collega i nodi corrispondenti.....	7
Aggiorna un nodo.....	7
Elimina tutti i nodi orfani.....	7
Capitolo 3: Pitone	8
Examples.....	8
Installa neo4jrestclient.....	8

Connetti a neo4j.....	8
Creare alcuni nodi con etichette.....	8
È possibile associare un'etichetta con molti nodi in un colpo solo.....	8
Creare relazioni.....	8
Relazioni bidirezionali.....	8
Abbina usando neo4jrestclient.....	8
"db" come definito sopra.....	9
Stampa risultati.....	9
Produzione:.....	9
Titoli di coda.....	10

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

Capitolo 1: Iniziare con neo4j

Osservazioni

Questa sezione fornisce una panoramica di cosa sia neo4j e del perché uno sviluppatore potrebbe volerlo utilizzare.

Dovrebbe anche menzionare soggetti di grandi dimensioni all'interno di neo4j e collegarsi agli argomenti correlati. Poiché la documentazione di neo4j è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Examples

Installazione o configurazione

Vai su [Installa Neo4j](#) che dovrebbe rilevare la piattaforma del sistema operativo tramite il tuo browser web, scaricare e seguire le consuete istruzioni di installazione per il tuo sistema operativo.

Neo4j è stato creato con Java, quindi funzionerà su qualsiasi piattaforma con Java installato, tuttavia il team Neo4j ha semplificato l'installazione fornendo pacchetti di installazione facili per piattaforme popolari (ad esempio un .dmg per Mac, un .deb per Debian e Ubuntu, un file .exe per piattaforme Windows 64 e 32 bit ...).

Per rivedere altre versioni e piattaforme disponibili, vedere [Altre versioni di Neo4j](#)

Imposta Neo4j come contenitore Docker:

```
## 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
```

Installazione e avvio di un server Neo4j

Passi prerequisites:

- Installa Java sul tuo computer

- Visita il [sito Web neo4j](#) e fai clic sul link "Scarica Community Edition" o visita direttamente il [link per il download](#) .
- Decomprimi il file scaricato .tar nella tua home directory

Avvia Neo4j dalla console (senza testa, senza server web)

- Visita la sottodirectory `/bin` della cartella estratta ed esegui nel terminale `./neo4j console`
- Ora è possibile eseguire query neo4j nel terminale

Avvia il server Web Neo4j

- Visita la sottodirectory `/bin` della cartella estratta ed esegui nel terminale `./neo4j start`
- Visita <http://localhost:7474/>
- Solo la prima volta, dovrai accedere con l'account predefinito e cambiare la password predefinita. A partire dalla versione 3.0.3 della comunità, il nome utente e la password predefiniti sono neo4j e neo4j.
- È ora possibile inserire query Neo4j nella console fornita nel browser Web e analizzare visivamente i risultati di ciascuna query.

Avvia il server Web Neo4j

Ogni server Neo4j attualmente (nella versione community) può ospitare un singolo database Neo4j, quindi per impostare un nuovo database:

- Visita la sottodirectory `/bin` ed esegui `./neo4j stop` per fermare il server
- Visita la sottodirectory `/conf` e modifica il file `neo4j.conf` , cambiando il valore del parametro `dbms.active_database` sul nome del nuovo database che vuoi creare.
- Visitare nuovamente la sottodirectory `/bin` ed eseguire `./neo4j start`
- Il server Web è stato riavviato con il nuovo database vuoto. Puoi visitare nuovamente <http://localhost:7474/> per lavorare con il nuovo database.
- Il database creato si trova nella sottodirectory `/data/databases` , sotto una cartella con il nome specificato nel parametro `dbms.active_database` .

Elimina uno dei database

- Assicurarsi che il server Neo4j non sia in esecuzione; vai alla sottodirectory `/bin` ed esegui `./neo4j status` . Se il messaggio di output indica che il server è in esecuzione, eseguire anche `./neo4j stop` .
- Quindi vai nella sottodirectory `/data/database` ed elimina la cartella del database che vuoi

rimuovere.

Cypher Query Language

Questo è il linguaggio di query di Cypher, Neo4j. In molti modi, Cypher è simile a SQL se si ha familiarità con esso, ad eccezione di SQL si riferisce agli elementi memorizzati in una tabella mentre Cypher si riferisce agli elementi memorizzati in un grafico.

Per prima cosa, dovremmo iniziare imparando come creare un grafico e aggiungere relazioni, poiché questo è essenzialmente ciò di cui Neo4j si occupa.

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

- Usa **CREATE** per creare dati
- Per indicare un nodo, si utilizza la parentesi: ()
- La parte ab: Object può essere suddivisa come segue: una variabile 'ab' e label 'Object' per il nuovo nodo. Nota che la variabile può essere qualsiasi cosa, ma devi essere coerente in una riga di Cypher Query
- Per aggiungere proprietà al nodo, utilizzare le parentesi: {} parentesi

Successivamente, impareremo a trovare MATCH

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

MATCH specifica che si desidera cercare un determinato nodo / modello di relazione (abc: Object) fa riferimento a un nodo Pattern (con etichetta Object) che memorizza le corrispondenze nella variabile abc. Puoi pensare a questa intera linea come la seguente

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

In questo caso, WHERE aggiunge un vincolo che è che la destinazione deve essere l'Inghilterra. Devi includere un ritorno alla fine per tutte le query MATCH (neo4j non accetterà solo una corrispondenza ... la tua query deve sempre restituire un valore [questo dipende anche dal tipo di query che stai scrivendo ... parleremo più questo più avanti, mentre introduciamo gli altri tipi di query che puoi fare].

La prossima riga verrà spiegata in futuro, dopo aver esaminato alcuni elementi del linguaggio Cypher Query. Questo per darti un assaggio di cosa possiamo fare con questa lingua! Di seguito, troverai un esempio che ottiene il cast dei film il cui titolo inizia con '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;
```

Un elenco completo di comandi e la loro sintassi possono essere trovati nella [scheda di riferimento](#) ufficiale [Neo4j Cypher](#) qui .

RDBMS Vs Graph Database

RDBMS	Database grafico
tabelle	grafici
Righe	nodi
Colonne e dati	Proprietà e suoi valori
vincoli	Le relazioni
Si unisce	Traversal

Leggi Iniziare con neo4j online: <https://riptutorial.com/it/neo4j/topic/945/iniziare-con-neo4j>

Capitolo 2: cifra

introduzione

Cypher è il linguaggio di query usato da Neo4j. Si utilizza Cypher per eseguire attività e corrispondenze con un grafico Neo4j.

Cypher è "ispirato da SQL" ed è progettato in modo intuitivo nel modo in cui descrivi le relazioni, cioè in genere il disegno del modello sarà simile alla rappresentazione Cypher del modello.

Examples

Creazione

Crea un nodo

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

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

Crea una relazione

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

Modelli di query

Eseguendo neo4j localmente, nella GUI del browser (predefinita: [http://localhost:7474 / browser /](http://localhost:7474/browser/)), è possibile eseguire il seguente comando per ottenere una palette di query.

```
:play query template
```

Questo ti aiuta a iniziare a creare e unire nodi e relazioni digitando query.

Creazione

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

cancellazione

Elimina tutti i nodi

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

DETACH non funziona nelle versioni precedenti (meno di 2.3), per le versioni precedenti

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

Elimina tutti i nodi di un'etichetta specifica

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

Abbina (cattura gruppo) e collega i nodi corrispondenti

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

Aggiorna un nodo

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

Elimina tutti i nodi orfani

I nodi / vertici orfani sono quelli privi di tutte le relazioni / spigoli.

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

Leggi cifra online: <https://riptutorial.com/it/neo4j/topic/3669/cifra>

Capitolo 3: Pitone

Examples

Installa neo4jrestclient

```
pip install neo4jrestclient
```

Connetti a neo4j

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

Creare alcuni nodi con etichette

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

È possibile associare un'etichetta con molti nodi in un colpo solo

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

Creare relazioni

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

Relazioni bidirezionali

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

Abbina usando neo4jrestclient

```
from neo4jrestclient import client

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

"db" come definito sopra

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

Stampa risultati

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

Produzione:

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

Leggi Pitone online: <https://riptutorial.com/it/neo4j/topic/5990/pitone>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con neo4j	Andrew Lank , Community , cs_user2017 , Dimos , Frank Pavageau , Prosen Ghosh , TinkerBotFoo , wintersolider
2	cifra	frant.hartm , Govind Singh , JOG , Liam , Nicole White , Renato Dinhani , SerialDev , TJ Walker
3	Pitone	SerialDev