

# neo4j-social-graph

## Социальный граф на neo4j

### 1. Скачиваем Docker-образ neo4j:

```
test@test>docker pull neo4j:latest
latest: Pulling from library/neo4j
Digest: sha256:b817478d92e2bc20dba98742f40927c8988ec0a090be38cc096ba6222bd16041
Status: Image is up to date for neo4j:latest
docker.io/library/neo4j:latest
```

### 2. Запускаем Docker-контейнер neo4j:

```
test@test>docker run \
--name testneo4j \
-p7474:7474 -p7687:7687 \
-d \
-v $HOME/neo4j/data:/data \
-v $HOME/neo4j/logs:/logs \
-v $HOME/neo4j/import:/var/lib/neo4j/import \
-v $HOME/neo4j/plugins:/plugins \
--env NEO4J_AUTH=neo4j/password \
neo4j:latest\
bcd3b21db2711f025925a07608fbc64539c9560ca0727ae348906a638ba419f6
```

### 3. Заходим Docker-контейнер neo4j:

```
test@test>docker exec -ti testneo4j bash
```

### 4. Заходим в командную оболочку neo4j:

```
root@bcd3b21db271:/var/lib/neo4j# cypher-shell -u neo4j -p password
Connected to Neo4j using Bolt protocol version 5.0 at neo4j://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
```

### 5. Создаём ограничение на уникальность имён людей:

```
neo4j@neo4j> CREATE CONSTRAINT FOR (person:PERSON) REQUIRE person.name IS UNIQUE;
0 rows
ready to start consuming query after 88 ms, results consumed after another 0 ms
Added 1 constraints
```

### 6. Создаём записи и устанавливаем отношения между ними (1 запись <====> 1 узел графа):

```
neo4j@neo4j> CREATE
  (john:Person {name: "John", sex: "male"}),
  (paul:Person {name: "Paul", sex: "male"}),
  (peter:Person {name: "Peter", sex: "male"}),
  (kate:Person {name: "Kate", sex: "female"}),
  (lisa:Person {name: "Lisa", sex: "female"}),
  (sara:Person {name: "Sara", sex: "female"}),
  (tyler:Person {name: "Tyler", sex: "male"}),
  (luke:Person {name: "Luke", sex: "male"}),
  (peter)-[:Parent]->(john),
  (kate)-[:Parent]->(john),
  (paul)-[:Parent]->(peter),
  (lisa)-[:Parent]->(peter);
0 rows
ready to start consuming query after 251 ms, results consumed after another 0 ms
Added 8 nodes, Created 4 relationships, Set 16 properties, Added 8 labels
```

## База знаний создана)

### 7. Выполняем запрос на поиск всех дедушек Джона:

```
neo4j@neo4j> MATCH (grandfather:Person {sex: "male"})-[:Parent]-()-[:Parent]-(:Person {name: "John"})
RETURN grandfather;
+-----+
```

```
| grandfather |  
+-----+  
| (:Person {name: "Paul", sex: "male"}) |  
+-----+
```

```
1 row  
ready to start consuming query after 414 ms, results consumed after another 14 ms
```

## 8. Выходим из командной оболочки neo4j:

```
neo4j@neo4j> :exit
```

```
Bye!
```

## 9. Выходим из Docker-контейнера neo4j:

```
root@bcd3b21db271:/var/lib/neo4j# exit  
exit
```

## 10. Останавливаем Docker-контейнер neo4j:

```
test@test>docker stop testneo4j  
testneo4j
```

## 11. Удаляем Docker-контейнер neo4j:

```
test@test>docker rm testneo4j  
testneo4j
```