Hello World at Foo Café

Hello World!

Let's graph our class!

Open your browser to: https://bit.ly/2J3nNNU
Password: neo
Don't hesitate! This is just a play environment

CREATE a node for yourself

CREATE yourself:

```
CREATE (p:Person {name: 'Your Name'})
RETURN p
```


If I were to do this, it would be:

```
CREATE (p:Person {name: 'Maria'})
RETURN p
```

MATCH and RETURN yourself

Find yourself:

```
MATCH (p:Person {name: 'Your Name'})
RETURN p;
```

If I were to do this, it would be:

```
MATCH (p:Person {name: 'Maria'})
RETURN p
```

SET and update properties

Find yourself and SET your city as a property.

```
MATCH (p:Person {name: 'Your Name'})
SET p.city = 'Your City'
RETURN p
```


If I were to do this, it would be:

```
MATCH (p:Person {name: 'Maria'})
SET p.city = 'Lund'
RETURN p
```

Find your neighbour

Ask the person to your left their name.
Find the pair of you:

```
MATCH (p1:Person {name: 'Your Name'})
MATCH (p2:Person {name: 'Your Neighbour'})
RETURN p1, p2
```

Or in one statement:

```
MATCH (p1:Person {name: 'Your Name'}),
      (p2:Person {name: 'Your Neighbour'})
RETURN p1, p2
```

name: Your Name
city: Your City
Person

name: Your Neighbour
city: Their City
Person

If I were to do this, it would be:

```
MATCH (p1:Person {name: 'Maria'}),
      (p2:Person {name: 'Ola'})
RETURN p1, p2
```
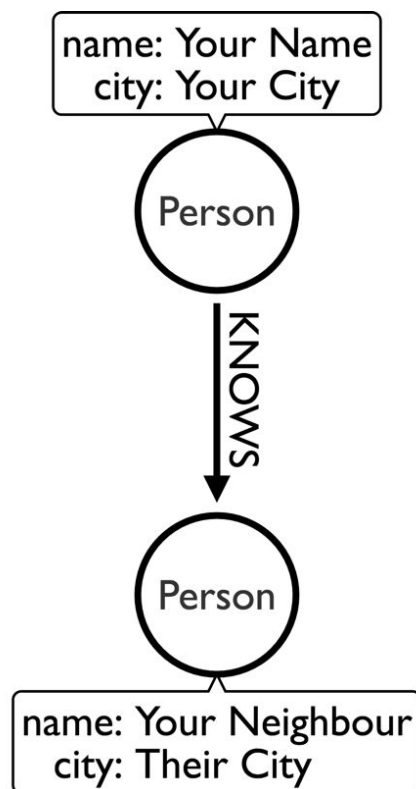
Creating relationships

Create a relationship between you and the person sitting next to you.

```
MATCH (p1:Person {name: 'Your Name'})
MATCH (p2:Person {name: 'Your Neighbour'})
CREATE (p1)-[:KNOWS]->(p2)
```


If I were to do this, it would be:

```
MATCH (p1:Person {name: 'Maria'}),
      (p2:Person {name: 'Ola'})
CREATE (p1)-[:KNOWS]->(p2)
```

Finding skills

Your query should look something like this:

```
MATCH (s:Skill)
RETURN s
```

or

```
MATCH (s:Skill)
RETURN s.name
```

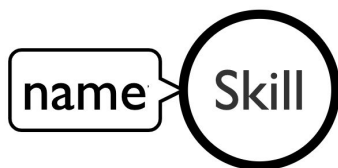| s.name |
| --- |
| "Java" |
| "Python" |
| "C#" |
| "Go" |
| "JavaScript" |
| "Scala" |
| "Bash" |
| "LINUX" |
| "Perl" |
| "HTML" |
| "CSS" |
| "Oracle" |

Missing skills

The database contains a bunch of languages and databases.
Chances are you are missing your skills in there!
Can you create some?

NOTE: Each skill should have:

- a label called **Skill**
- a property called **name**

```
CREATE (:Skill { name: 'Your skill' })
```

My create statement could look something like this:

```
CREATE (:Skill { name: 'Technical writing' })
```

Your skills

Now let's associate you to some of those skills.
Can you write a query to add a HAS_SKILL relationship from you to whichever skills you have?
Hint: Don't forget to use the MERGE clause!
Your query should look something like this:

```
MATCH (me:Person { name: 'Your Name' })
MATCH (skill:Skill { name: 'Name of Skill' })
MERGE (me)-[:HAS_SKILL]->(skill)
RETURN me, skill
```

if you have many skills, you could do something like this:

```
UNWIND ["skill 1", "skill 2", "skill 3"] as nameOfSkill
MATCH (skill:Skill { name: nameOfSkill })
MATCH (me:Person { name: "My Name" })
MERGE (me)-[:HAS_SKILL]->(skill)
```


My query could look like this:

```
MATCH (me:Person { name: 'Maria' })
MATCH (skill1:Skill { name: 'Technical writing' })
MATCH (skill2:Skill { name: 'SQL' })
MERGE (me)-[:HAS_SKILL]->(skill1)
MERGE (me)-[:HAS_SKILL]->(skill2)
RETURN me, skill1, skill2
```
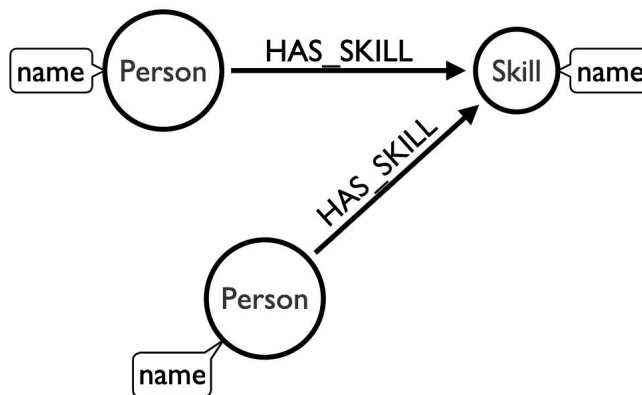
Do your friends share any skills?

Now let's write a query to check if our neighbours have any of the same skills as us. Your query should look something like this:

```
MATCH (me:Person { name: 'Your Name' })
MATCH
(friend)-[:HAS_SKILL]->(skill)<-[:HAS_SKILL]-(me)
RETURN friend, skill
```


My query would look like this:

```
MATCH (me:Person { name: 'Maria' })
MATCH
(friend)-[:HAS_SKILL]->(skill)<-[:HAS_SKILL]-(me)
RETURN friend, skill
```

What skills do you want to learn?

Now let's associate you to some skills that you want to learn
Can you write a query to add a WANTS_TO_LEARN relationship from you to whichever skills you want to learn?

Your query should look something like this:

```
MATCH (me:Person { name: 'Your Name' })
MATCH (skill:Skill { name: 'Name of Skill' })
MERGE (me)-[:WANTS_TO_LEARN]->(skill)
RETURN me, skill
```
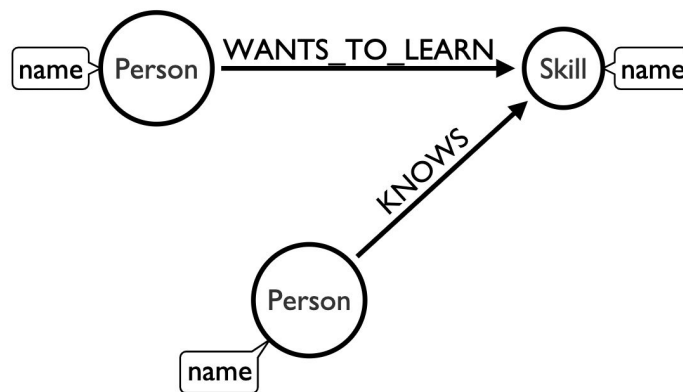

My query would look like this:

```
MATCH (me:Person { name: 'Maria' })
MATCH (skill:Skill { name: 'Python' })
MERGE (me)-[:WANTS_TO_LEARN]->(skill)
RETURN me, skill
```

Can your friends teach you?

Can you write a query to find out if any of your friends already have the skills that you want to learn?

Your query should look something like this:

```
MATCH (me:Person { name: 'Your Name' })
MATCH (friend:Person)-[:HAS_SKILL]-(skill)-[:WANTS_TO_LEARN]-(me)
RETURN friend, skill
```


My query would look like this:

```
MATCH (me:Person { name: 'Maria' })
MATCH (friend:Person)-[:HAS_SKILL]-(skill)-[:WANTS_TO_LEARN]-(me)
RETURN friend, skill
```

Already done?

If you want to continue exploring, you can type the following into your browser and follow the
self-guided tutorial:

```
:play https://guides.neo4j.com/fundamentals/basics.html
:play https://guides.neo4j.com/fundamentals/advanced.html
:play https://guides.neo4j.com/fundamentals/import.html
```

Alternatively:

```
:play movie-graph
```