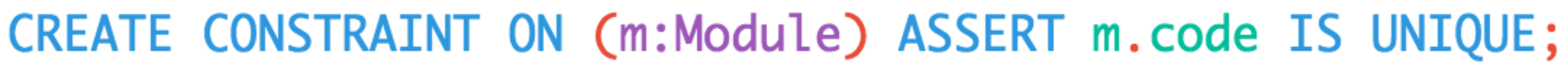
Cypher Query Language

Dr James Xue

James.xue@northampton.ac.uk

Create Constraints

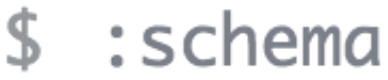
- Usually a constraint needs to be created BEFORE creation of actual nodes.



```
CREATE CONSTRAINT ON (m:Module) ASSERT m.code IS UNIQUE;
```

The constraint above ensures no module with duplicated *code* is allowed

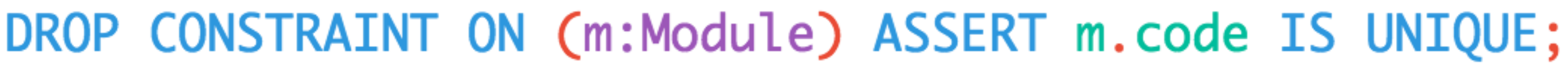
- Use the following command to verify the created constraints



```
$ :schema
```

DROP Constraints

- To DROP a constraint, type



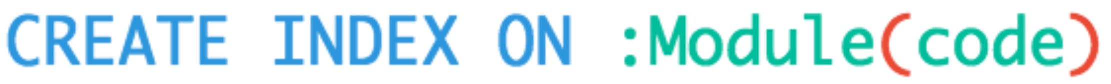
```
DROP CONSTRAINT ON (m:Module) ASSERT m.code IS UNIQUE;
```

Try It Out

- Start Neo4j
- Create a Module node with code: CSY3024
- Create another Module node with the same code: CSY3024
- What happens? Why?
- Recall the duplicated actor/movie nodes and think about a solution to prevent it.

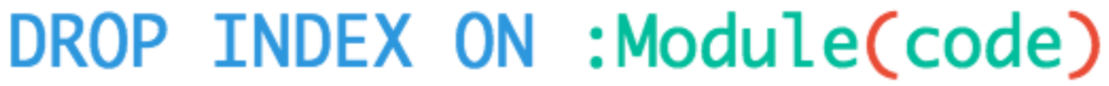
CREATE and DROP index

- An index on a single property for all nodes that have a particular label can be created with the command below:



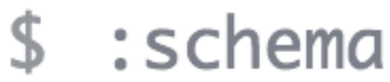
```
CREATE INDEX ON :Module(code)
```

- An index on all nodes that have a label and single property combination can be dropped with the command below:



```
DROP INDEX ON :Module(code)
```

- Using the following command to verify the created index



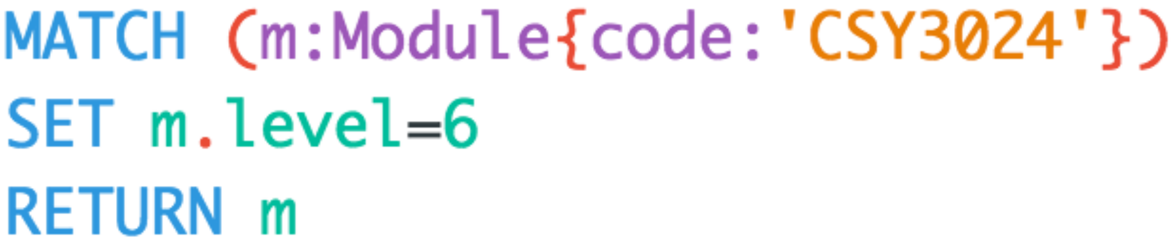
```
$ :schema
```

SET clause

- The SET clause is used to update labels on nodes and properties on nodes and relationships.
- Setting labels on a node is an idempotent operations – i.e., if you try to set a label on a node that already has that label on it, nothing happens.

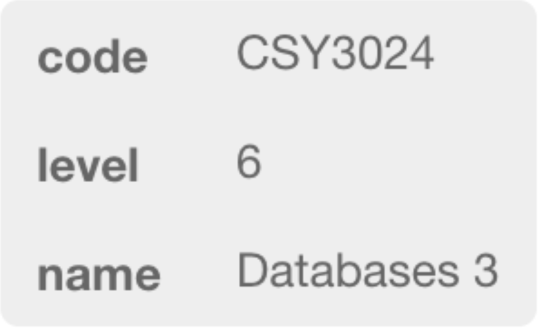
SET clause…

- New attributes can be added to using the SET clause:



```
MATCH (m:Module{code:'CSY3024'})
SET m.level=6
RETURN m
```

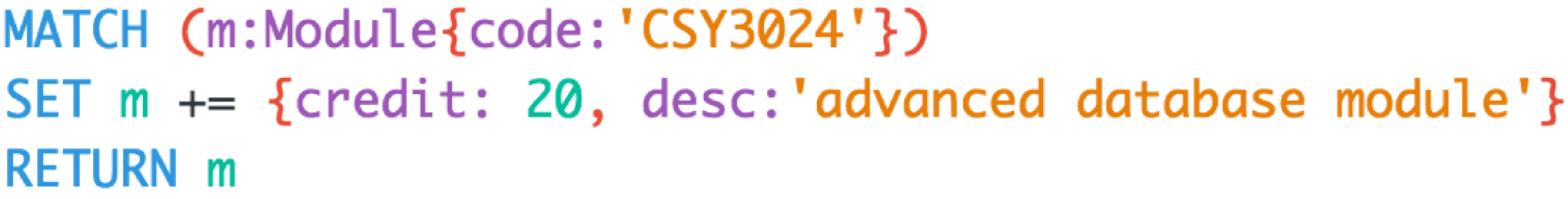
Output:



```
code    CSY3024
level   6
name    Databases 3
```

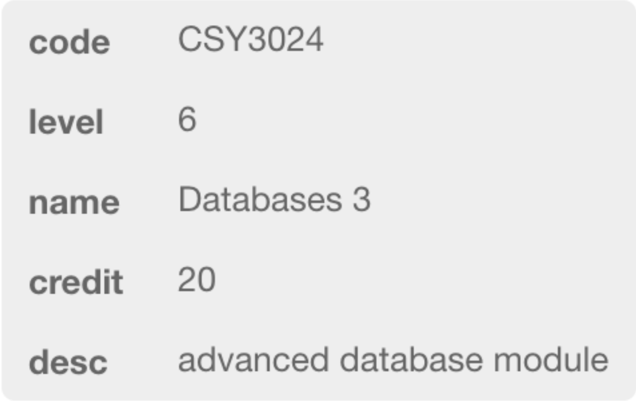
Adding Properties From Maps

- When setting properties from a map (literal, paremeter, or graph element), you can use the += form of SET to only add properties, and not remove any of the existing properties on the graph element.



```
MATCH (m:Module{code:'CSY3024'})
SET m += {credit: 20, desc:'advanced database module'}
RETURN m
```

Output:



```
code      CSY3024
level     6
name      Databases 3
credit    20
desc      advanced database module
```

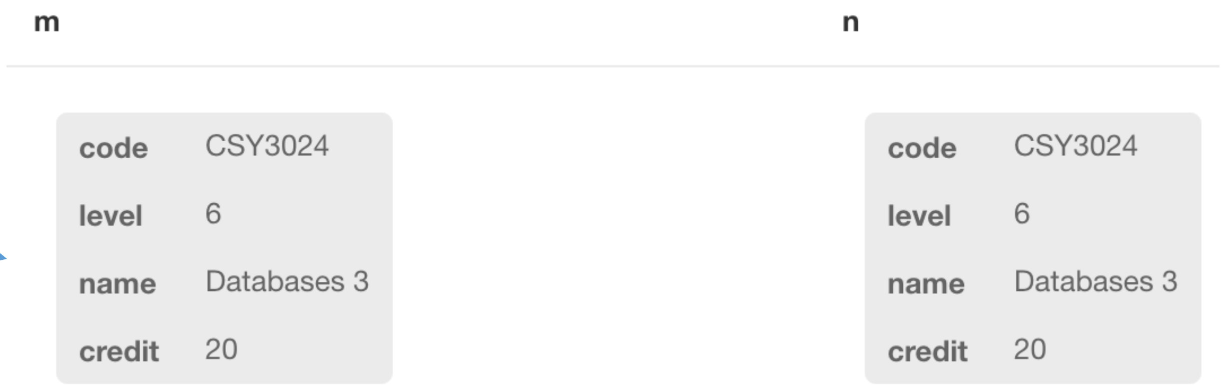
Copying Properties Between Nodes And between Relationships

- You can also use SET to copy all properties from one graph element to another. Remember that doing this will remove all other properties on the receiving graph element.

```
CREATE (n:Module{code:'CSY2038',name:'Databases 2'})

MATCH (m:Module{code:'CSY3024'}),(n:Module{code:'CSY2038'})
SET n=m RETURN m,n
```

The attributes and values of CSY2038 have been replaced by those in CSY3024



m	n
code CSY3024 level 6 name Databases 3 credit 20	code CSY3024 level 6 name Databases 3 credit 20

Exercise (optional)

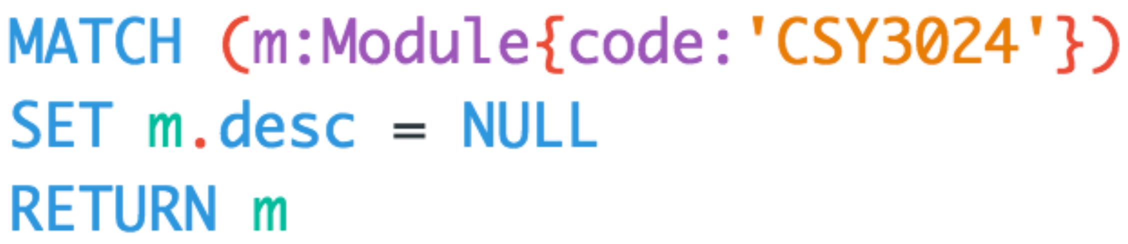
- Find out how to copy properties between relationships?

REMOVE clause

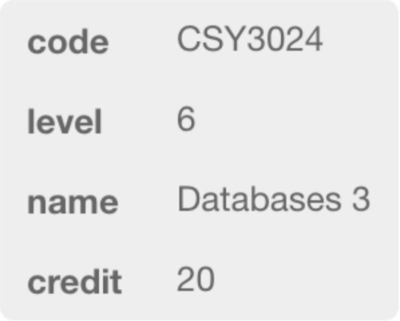
- The REMOVE clause is used to remove properties and labels from graph elements.

Remove a Property using NULL

- Neo4j doesn’t allow storing null in properties. Instead, if no value exists, the property is just not there.



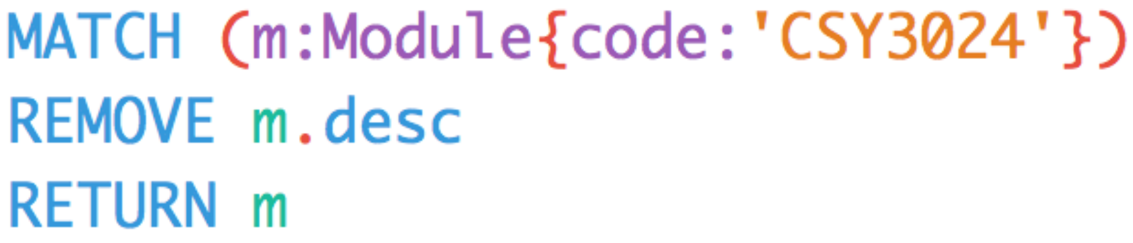
```
MATCH (m:Module{code:'CSY3024'})
SET m.desc = NULL
RETURN m
```



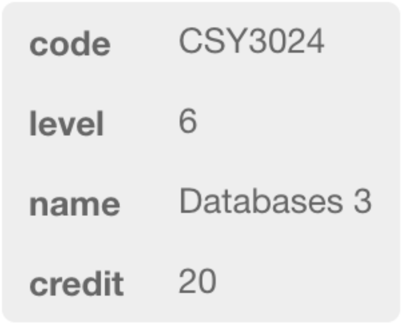
code	CSY3024
level	6
name	Databases 3
credit	20

Remove a property Using REMOVE

- To remove a property value on a node or a relationship, is also done with REMOVE.



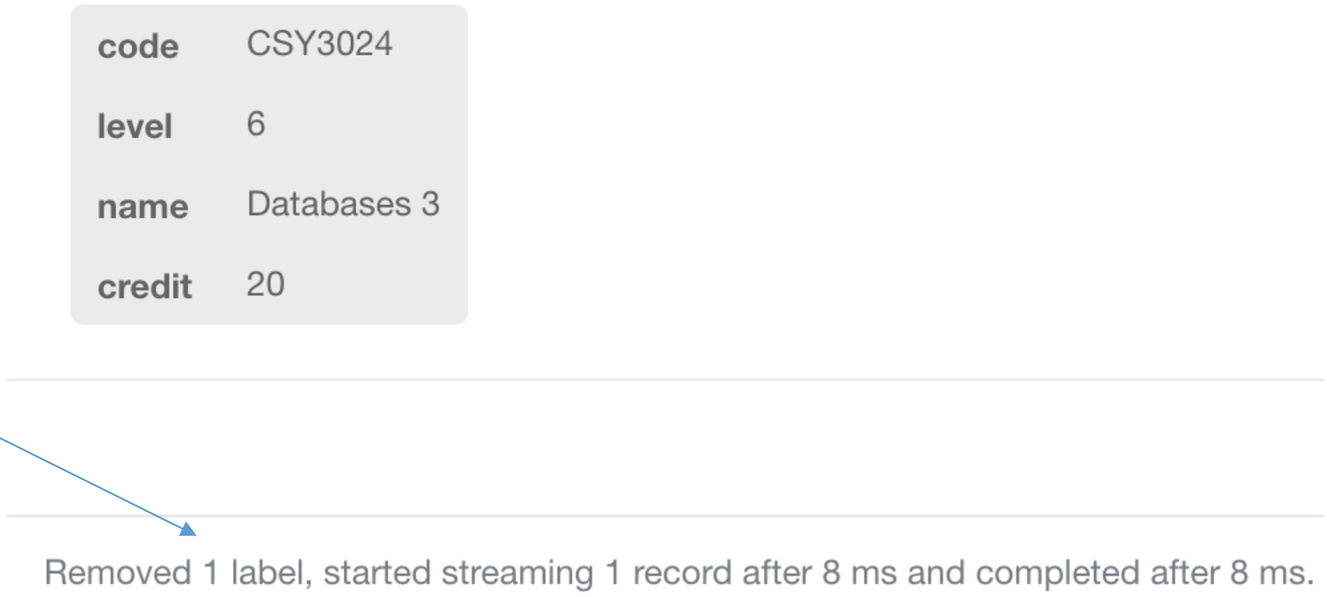
```
MATCH (m:Module{code:'CSY3024'})
REMOVE m.desc
RETURN m
```



code	CSY3024
level	6
name	Databases 3
credit	20

REMOVE labels

```
MATCH (m:Module{code:'CSY3024'})
REMOVE m:Module
RETURN m
```



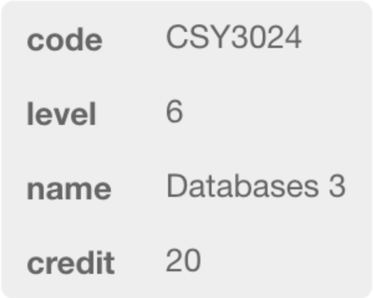
```
code     CSY3024
level    6
name     Databases 3
credit   20
```

Removed 1 label, started streaming 1 record after 8 ms and completed after 8 ms.

After the Label Module is removed, we still can run the following command:

```
MATCH (m{code:'CSY3024'})
RETURN m
```

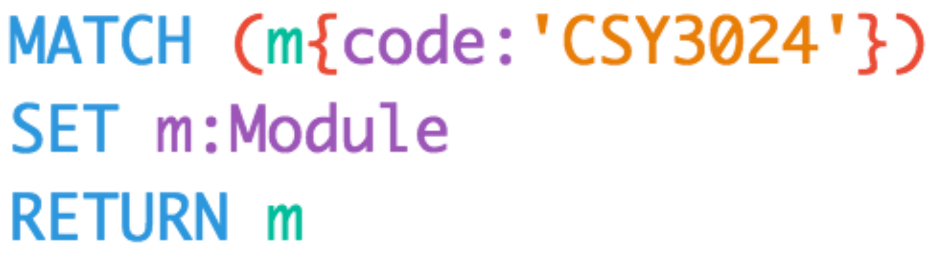
Output:



```
code     CSY3024
level    6
name     Databases 3
credit   20
```

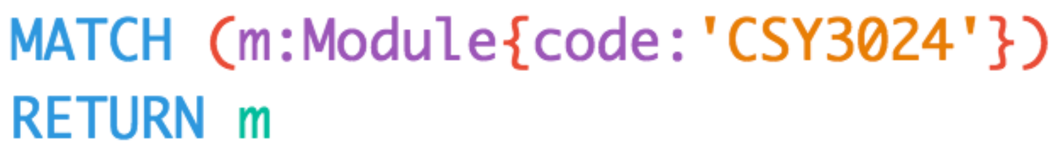
Add Labels

- If a label is needed after a node is created, the following command can be used:



```
MATCH (m{code:'CSY3024'})
SET m:Module
RETURN m
```

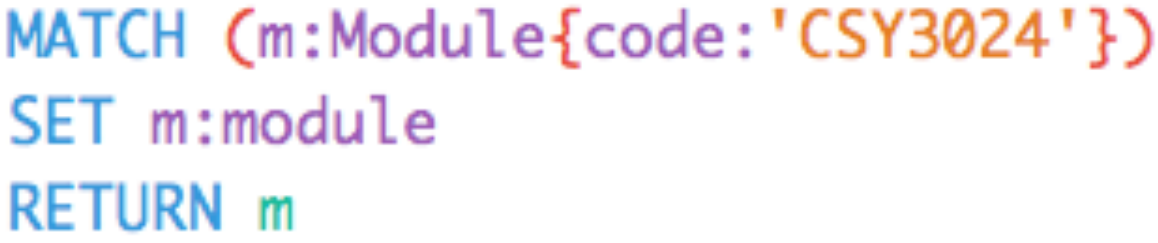
A MATCH command can be used to verify the added label:



```
MATCH (m:Module{code:'CSY3024'})
RETURN m
```

Update Label

- Does the following command update the label :Module to :module?



```
MATCH (m:Module{code:'CSY3024'})
SET m:module
RETURN m
```

- Why?
- How to update then?

Exercises

- Start the Neo4j database
- Using the reference codes in the early slides and practice how to add, remove properties in nodes/relationships and add and remove labels.
- Log your learning activities and reflection in the diary.