Tutorial 9a - Solution
Neo4j Basic

File Template for Answer

A file template (`Week10_Neo4jExercise.cypher`) is provided for you on Moodle (under the Week 10 resources) to answer the tutorial work. You can use any word editor application (e.g. Atom, Notepad++) to edit your answer in the provided file template.

You are required to implement a graph database of the following scenario.

MonUni has been storing their student enrolment records in a relational database system. The database system contains mainly four tables, which are STUDENT, SUBJECT, and ENROLMENT (shown in Table 1-3). The STUDENT table contains the personal details of students. The SUBJECT table keeps all the records of the available subjects. After a student selects a subject she/he is interested, the details are stored in the ENROLMENT table.

Table 1. Student

SID	Slname	Sfname	Sgender	SEmail	Age
10001	Tan	Mirriam	F	mtan@student.monuni.com	20
10002	Murray	Juan	M	jmur@student.monuni.com	18
10003	Lay	Andy	M	alay@student.monuni.com	28

Table 2. Subject

Ucode	Utitle	Ucredit
IT001	Database	5
IT002	Java	5
IT003	SAP	10
IT004	Network	5

Table 3. Enrolment

SID	UCode
10001	IT001
10001	IT002
10002	IT001
10002	IT004
10003	IT001

Due to the growing popularity of graph database system, the management board of MonUni is interested to move their data from the traditional relational database system to a graph database system. As part of the database management team of MonUni, you have been allocated to design the new graph database and in charge of transferring the current data from a relational database to your newly designed graph database.

Tasks:

1. Create a new project in Neo4j and name it as `Tutorial9a`.

2. Add a new graph. Name the graph to `MonUniGraph` and set the password to `graph`.

3. After your graph has been created, start the graph then open Neo4j Browser.

4. Create the nodes for the students.

```
CREATE (mirriam:Student {name:'Mirriam Tan', gender:'F',
email:'mtan@student.monuni.com', age: 20})

CREATE (juan:Student {name:'Juan Murray', gender:'M',
email:'jmur@student.monuni.com', age: 18})

CREATE (andy:Student {name:'Andy Lay', gender:'M',
email:'alay@student.monuni.com', age: 28})
```

5. Retrieve the nodes you recently created.

```
MATCH (n) RETURN n
```

6. Create the nodes for the units.

```
CREATE (it001:Unit {code:'IT001', title:'Database', credit:5})
CREATE(it002:Unit {code:'IT002', title:'Java', credit:5})
CREATE(it003:Unit {code:'IT003', title:'SAP', credit:10})
CREATE(it004:Unit {code:'IT004', title:'Network', credit:5})
```

7. Create the relationship between students and units (hint: you need to use match to find the nodes before creating the relationship).

```
MATCH (mirriam:Student {name:'Mirriam Tan'})
MATCH (it001:Unit {code:'IT001'})
CREATE (mirriam)-[:ENROL]->(it001)

MATCH (mirriam:Student {name:'Mirriam Tan'})
MATCH (it002:Unit {code:'IT002'})
CREATE (mirriam)-[:ENROL]->(it002)

MATCH (juan:Student {name:'Juan Murray'})
MATCH (it001:Unit {code:'IT001'})
CREATE (juan)-[:ENROL]->(it001)

MATCH (juan:Student {name:'Juan Murray'})
MATCH (it004:Unit {code:'IT004'})
CREATE (juan)-[:ENROL]->(it004)

MATCH (andy:Student {name:'Andy Lay'})
MATCH (it001:Unit {code:'IT001'})
CREATE (andy)-[:ENROL]->(it001)
```

As you have created your database, answer the following queries:

1. Display all the students's email.

```
MATCH (n:Student)
RETURN n.email
```

2. Find Database unit.

```
MATCH (n:Unit {title:'Database'})
RETURN n
```

OR

```
MATCH (n:Unit)
WHERE n.title='Database'
RETURN n
```

3. Show Andy's age.

```
MATCH (n:Student)
WHERE n.name="Andy Lay"
RETURN n.age
```

4. Find all students whose age range is in between 20-30.

```
MATCH (n:Student)
WHERE n.age>=20 and n.age<=30
RETURN n
```

5. Update Java’s credit to 7.

```
MATCH (n:Unit {title:"Java"})
SET n.credit=7
RETURN n
```

The End