

Python III

HIGHER DIPLOMA IN DATA ANALYTICS



neo4j-python-driver

- ▶ `from neo4j import GraphDatabase`
- ▶ `uri = "neo4j://localhost:7687"`
- ▶ `neo4jDriver = GraphDatabase.driver(uri, auth=("username", "password"))`



Transaction Types

- ▶ Auto-commit Transactions
- ▶ Explicit Transactions
- ▶ Managed Transactions
 - ▶ `read_transaction()`
 - ▶ `write_transaction()`

Transactions are executed in a [Session](#).



read_transaction()

- ▶ Execute a unit of work in a managed read transaction.
- ▶ This transaction will automatically be committed unless an exception is thrown during query execution or by the user code.



read_transaction()

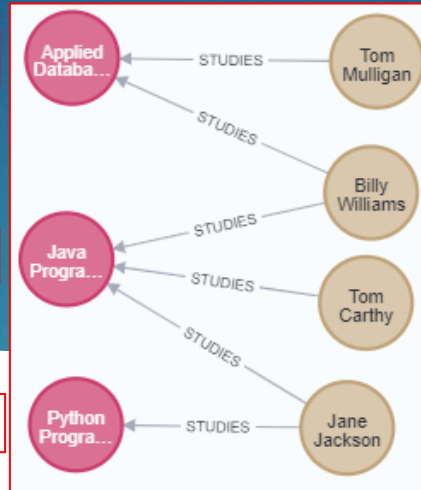
```
with neo4jDriver.session() as session:  
    result = session.read_transaction(function, parameters)
```



Module credits:5 mid:ADB **name**:Applied Databases

Module credits:5 mid:JAV **name**:Java Programming

Module credits:10 mid:PYT **name**:Python Programming



Student name:Tom Mulligan **sid**:G004

Student name:Billy Williams **sid**:G003

Student name:Tom Carthy **sid**:G001

Student name:Jane Jackson **sid**:G002

- ▶ Find the names of all students who study “Java Programming”.
- ▶ `MATCH(m:Module{name:"Java Programming"})<-[:STUDIES]-(s:Student) RETURN s.name`
 - ▶ s.name
 - ▶ Billy Williams
 - ▶ Tom Carthy
 - ▶ Jane Jackson



```
with neo4jDriver.session() as session:
```

```
    names = session.read_transaction(get_names, "Java Programming")
```

```
    for name in names:  
        print(name)
```

Billy Williams
Tom Carthy
Jane Jackson

```
def get_names(tx, module):
```

```
    query = "MATCH(m:Module{name:$name})<-[:STUDIES]-(s:Student) RETURN s.name"
```

```
    results = tx.run(query, name=module)
```

```
    names = []
```

```
    for result in results:
```

```
        names.append(result['s.name'])
```

```
    return names
```



write_transaction()

- ▶ Execute a unit of work in a managed write transaction.
- ▶ This transaction will automatically be committed unless an exception is thrown during query execution or by the user code.



write_transaction()

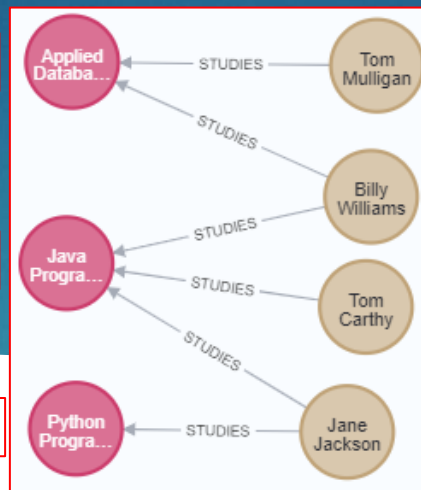
```
with neo4jDriver.session() as session:  
    result = session.write_transaction(function, parameters)
```



Module credits:5 mid:ADB **name**:Applied Databases

Module credits:5 mid:JAV **name**:Java Programming

Module credits:10 mid:PYT **name**:Python Programming



Student name:Tom Mulligan **sid**:G004

Student name:Billy Williams **sid**:G003

Student name:Tom Carthy **sid**:G001

Student name:Jane Jackson **sid**:G002

► Add the following student to the database: **name**: John Williams, **sid**: G001

► `CREATE(s:Student{name:"John Williams", sid:"G001"}) RETURN s`



```
with neo4jDriver.session() as session:
```

```
    try:
```

```
        session.write_transaction(add_student, {"name": "John Williams", "sid": "G001"})
```

```
        print("Student Created")
```

```
    except exceptions.ConstraintError as e:
```

```
        print("ERROR: ", e.message)
```

```
from neo4j import exceptions
```

```
def add_student(tx, student):
```

```
    query = "CREATE(s:Student{name:$name, sid:$sid}) RETURN s"
```

```
    result = tx.run(query, name=student["name"], sid=student["sid"])
```



Review

- ▶ MySQL
- ▶ Neo4j
- ▶ Python

