Final Project Output

This file consists of ten commands passed through the Python Program, and output queries verified by passing them to Neo4j.

“Find” commands were able to run directly through API, provided by Neo4j.

Commands : Create and Delete were tested by manually copy and pasting the program output into the Neo4j browser to check if node is created or deleted (screenshots from the Browsers have been attached).

Following Ten Natural Language Sentences were tested:

1. Make a movie called Twilight.
2. Delete the actor Robin Williams.
3. Create an actor Kate Sharma.
4. Find Tom Hanks.
5. See if Tom Cruise exists in the graph.
6. Find Johnny Depp in the graph.
7. See if the movie Indiana Jones is in the graph.
8. Find the movie The Matrix.
9. Make a movie Indiana Jones.
10. Delete the movie Indiana Jones.

1. Query: Make a movie called Twilight.

Output of Program:

```
Please enter command to be converted into Cypher Query Make a movie Twilight.
Name:  Twilight Label: WORK_OF_ART
"Query Type is Create
CREATE (a: Movie {title:'Twilight'}) RETURN a
```

Output from Neo4j Graph:

2. Delete the actor, Robin Williams.

Program Output:

```
Name:  Robin Williams Label: PERSON
"Query Type is Delete
MATCH (a: Person {name:'Robin Williams'}) DETACH DELETE a
```

Neo4j Output:

3. Create an actor Kate Sharma.

Program Output:

```
Please enter command to be converted into Cypher Query Create an actor Kate Sharma.
Name:  Kate Sharma Label: PERSON
"Query Type is Create
CREATE (a: Person {name:'Kate Sharma'}) RETURN a
```

Neo4j Output:

4. Find Tom Hanks.

```
Please enter command to be converted into Cypher Query Find Tom Hanks.
Name:  Tom Hanks Label: PERSON
Query Type is Match
<neo4j._sync.work.result.Result object at 0x7f2702ebe430>
['Tom Hanks']
Found person: Tom Hanks
```

5. See if Tom Cruise exists in the graph.

```
Please enter command to be converted into Cypher Query See if Tom Cruise exists in the graph.
Name:  Tom Cruise Label: PERSON
Query Type is Match
<neo4j._sync.work.result.Result object at 0x7f27064dba60>
['Tom Cruise']
Found person: Tom Cruise
```

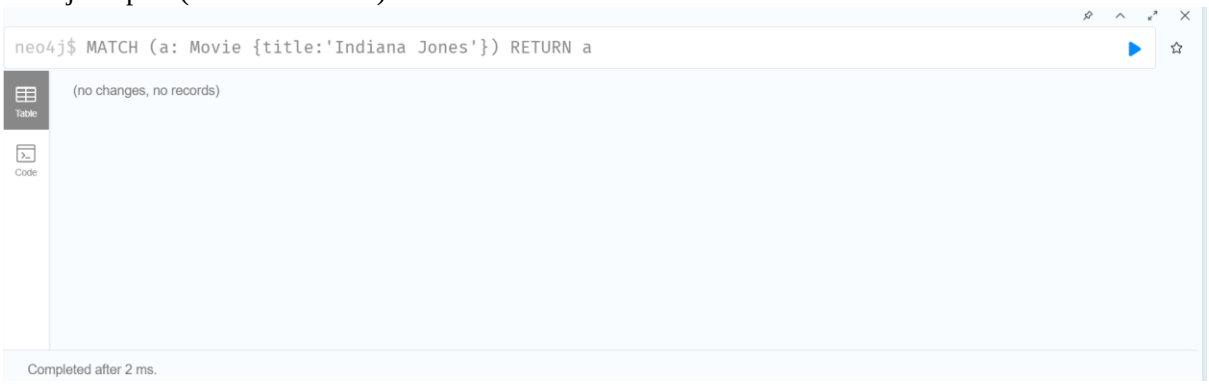
6. Find Johnny Depp in the graph.

```
Please enter command to be converted into Cypher Query Find Johnny Depp in the graph.
Name:  Johnny Depp Label: PERSON
Query Type is Match
<neo4j._sync.work.result.Result object at 0x7fd222eafe50>
[]
MATCH (a: Person {name:'Johnny Depp'}) RETURN a
```

7. See if the movie Indiana Jones is in the graph.

```
Please enter command to be converted into Cypher Query See if the movie Indiana Jones is in the graph
Name:  Indiana Jones Label: ORG
Query Type is Match
<neo4j._sync.work.result.Result object at 0x7f27064dbdc0>
[]
MATCH (a: Movie {title:'Indiana Jones'}) RETURN a
```

Neo4j Output: (no record found)



8. Find the movie The Matrix.
   Program Output:

```
Please enter command to be converted into Cypher Query Find the movie The Matrix.
Name:  Matrix Label: ORG
Query Type is Match
<neo4j._sync.work.result.Result object at 0x7f27064e69a0>
[]
MATCH (a: Movie:'Matrix'}) RETURN a
```

9. Make a movie Indiana Jones.

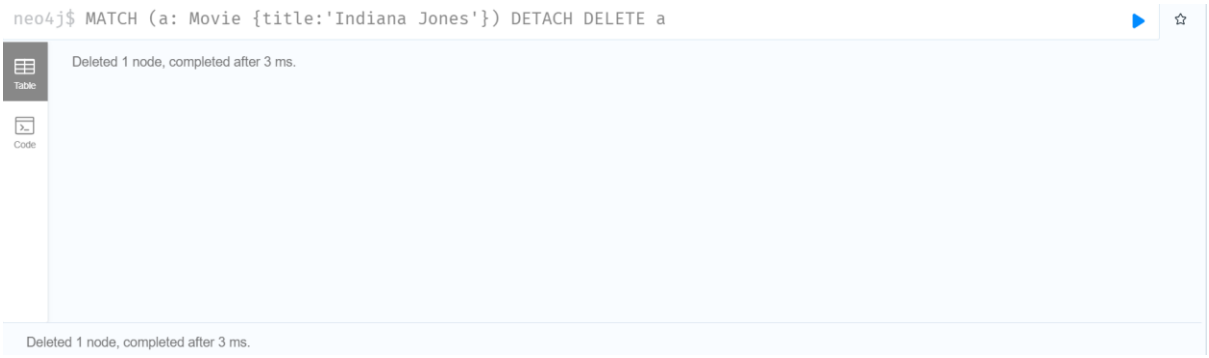
```
Please enter command to be converted into Cypher Query !
Name:  Indiana Jones Label: ORG
"Query Type is Create
CREATE (a: Movie {title:'Indiana Jones'}) RETURN a
```

10. Delete the movie Indiana Jones.

Program Output:

```
Please enter command to be converted into Cypher Query Delete the movie Indiana Jones.
Name:  Indiana Jones Label: ORG
"Query Type is Delete
MATCH (a: Movie {title:'Indiana Jones'}) DETACH DELETE a
```

Neo4j Output:


neo4j$ MATCH (a: Movie {title:'Indiana Jones'}) DETACH DELETE a
Table
Code
Deleted 1 node, completed after 3 ms.
Deleted 1 node, completed after 3 ms.