

# **Máster Universitario en Inteligencia Computacional e Internet de las Cosas de la Universidad de Córdoba**

**Course:** *Análisis, Diseño y Procesamiento de Datos Aplicados a las Ciencias y a las Tecnologías.*

**Year:** 2022-23

## **Assignment 2: Neo4j & Python driver**

### **DELIVERABLE/ENTREGABLE**

Make a small report with the answers to the exercises of this practice to later generate a PDF document that will be the "deliverable" of this practice.

Take care of the cover page, table of contents, content, style, format, bibliographical references, structure, etc. of the deliverables.

Not all the points of the practice involve including something in the deliverable. Only those points of the exercise in which it is explicitly indicated and all those in which you are asked for the code or instructions necessary to carry out an exercise will be included in the deliverable, in which case you will indicate the code and the explanation that you consider necessary.

In the deliverable always indicate the exercise number and its statement.

Free text answers to exercise questions, or additional explanations to exercises, should be no more than 4-5 paragraphs maximum each exercise, of about 30-40 words each.

Always write in the deliverable the commands you use to solve each exercise in the corresponding section.

### **EXERCISES**

Load the Movie Graph that will serve as the basis for these exercises and do the following in Cypher:

1. Query to display the schema of your database.
2. Query to retrieve all nodes from the database.
3. All Person nodes
4. All Movies nodes
5. Retrieve all Movie nodes that have a released property value of 2003.

6. Retrieve all Movies released in 2006, returning their titles.
7. Retrieve all Movie nodes from the database and return the title, released, and tagline values.
8. Modify the query you just ran so that the headings for the columns of the table returned are more descriptive.
9. Retrieve all people who wrote the movie Speed Racer.
10. Explica brevemente lo que hace la siguiente instrucción:  
`MATCH (m:Movie)-[rel]-(:Person {name: 'Tom Hanks'}) RETURN m.title, type(rel)`
11. Explica brevemente lo que hace la siguiente instrucción:  
`MATCH (m:Movie)-[rel:ACTED_IN]-(:Person {name: 'Tom Hanks'}) RETURN m.title, rel.roles`
12. Explica brevemente lo que hace la siguiente instrucción:  
`MATCH (gene:Person)-[:ACTED_IN]->(m:Movie) ←[:ACTED_IN]-(other:Person)  
 WHERE gene.name= 'Gene Hackman'  
 AND exists( (other)-[:DIRECTED]->(m) )  
 RETURN gene, other, m`
13. Devuelve las personas que hayan actuado y dirigido a la vez una película.

### **ADDITIONAL AND OPTIONAL CYPHER EXERCISES**

(additional optional exercises to raise marks)

14. Use MERGE to update (ON MATCH) the existing Production node for Forrest Gump to add the company property with a value of Paramount Pictures.
15. Añadir la relación [:DIRECTED] entre Robert Zemeckis y el film "Forrest Gump":
16. Borrar el nodo de la producción "Forrest Gump"
17. Finding who directed Cloud Atlas movie
18. Finding all people who have co-acted with Tom Hanks in any movie
19. Finding all people related to the movie Cloud Atlas in any way
20. Finding Movies and Actors that are 3 hops away from Kevin Bacon
21. Devuelve para cada actor, una lista de películas en las que ha actuado (usa collect()).
22. Todos los actores que han coactuado alguna vez con Tom Hanks.

### **PYTHON PY2NEO DRIVER EXERCISES**

1. Analiza los siguientes scripts python colgados en moodle de la asignatura. Edita dichos scripts para que se conecten a tu base de datos y se ejecuten correctamente. Analiza el código y comenta brevemente en el entregable qué hace cada uno:
  - a) `movies1.py` (change 'connecting to DBMS' with your password)
  - b) `movies_full_dataset.py` (change 'connecting to DBMS' with your password)
  - c) `query1.py` (change 'connecting to DBMS' with your password)
2. Realiza los siguientes scripts en Python:
  - a) Hacer el script (`personas.py`) que escriba en un fichero de salida (`personas.txt`) el nombre y apellidos y fecha de nacimiento de cada persona. Delante del nombre de cada persona aparecerá un número entero comenzando en 1 que se irá incrementando en 1 con cada nueva persona de la siguiente forma:  
1- Keanu Reeves, 1964  
2- Carrie-Anne Moss, 1967  
3- Laurence Fishburne, 1961  
4- Hugo Weaving, 1960  
.....
  - b) Hacer un script (`personas_1960.py`) que escriba en un fichero de salida (`personas_1960.txt`) el número, nombre y apellidos y fecha de nacimiento de cada persona cuya fecha de nacimiento sea posterior a 1960. El formato del fichero de salida será idéntico al del ejercicio anterior. Usa la `match(...).where(...)` en el `NodeMatcher`.
  - c) Hacer el script `actores.py`, que escriba en un fichero de salida (`actores.txt`) una lista numerada de los actores y actrices sin repetir ninguno.
  - d) Hacer un script `actores_1960.py` que escriba en un fichero de salida (`actores_1960.txt`) una lista numerada con los nombres de los actores y actrices del grafo nacidas después de 1960 con el siguiente formato en cada línea del fichero de salida:

Número – Nombre y Apellidos : fechas de nacimiento

Antes de hacer este ejercicio añadir al nodo de la actriz “Naomie Harris” su fecha de nacimiento como se ha indicado en la transparencia “Ejercicio con MERGE”.