Neo4J
Graph Data Science Library

IE 2021/2022

Índice

- Introducción a Neo4J - GDS
- Esquema básico de uso
- Graph Catalog
- Algoritmos de grafos en Neo4J con GDS
- ¿Machine Learning con GDS?

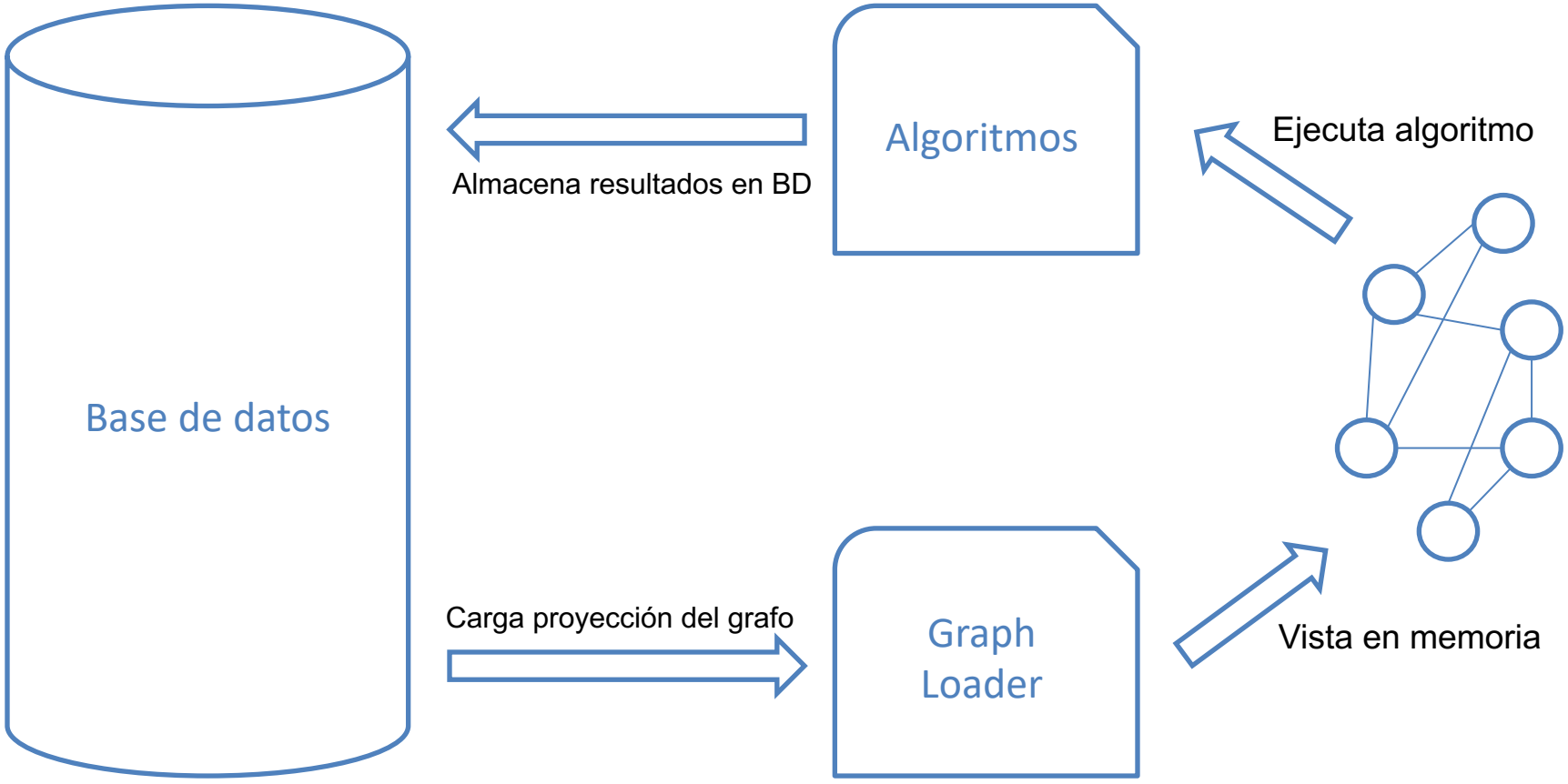
INTRODUCCIÓN A NEO4J - GDS

Graph Data Science Library

- Algoritmos sobre grafos
- Manejo de subgrafos como *vistas*
- Traits
  - Directed
  - Undirected
  - Homogeneous: se tratarán todos los nodos y aristas como si fueran de un mismo tipo.
  - Heterogeneous
  - Weighted
- Catálogo razonablemente extenso de algoritmos

ESQUEMA BÁSICO DE USO

Esquema básico de uso



GRAPH CATALOG

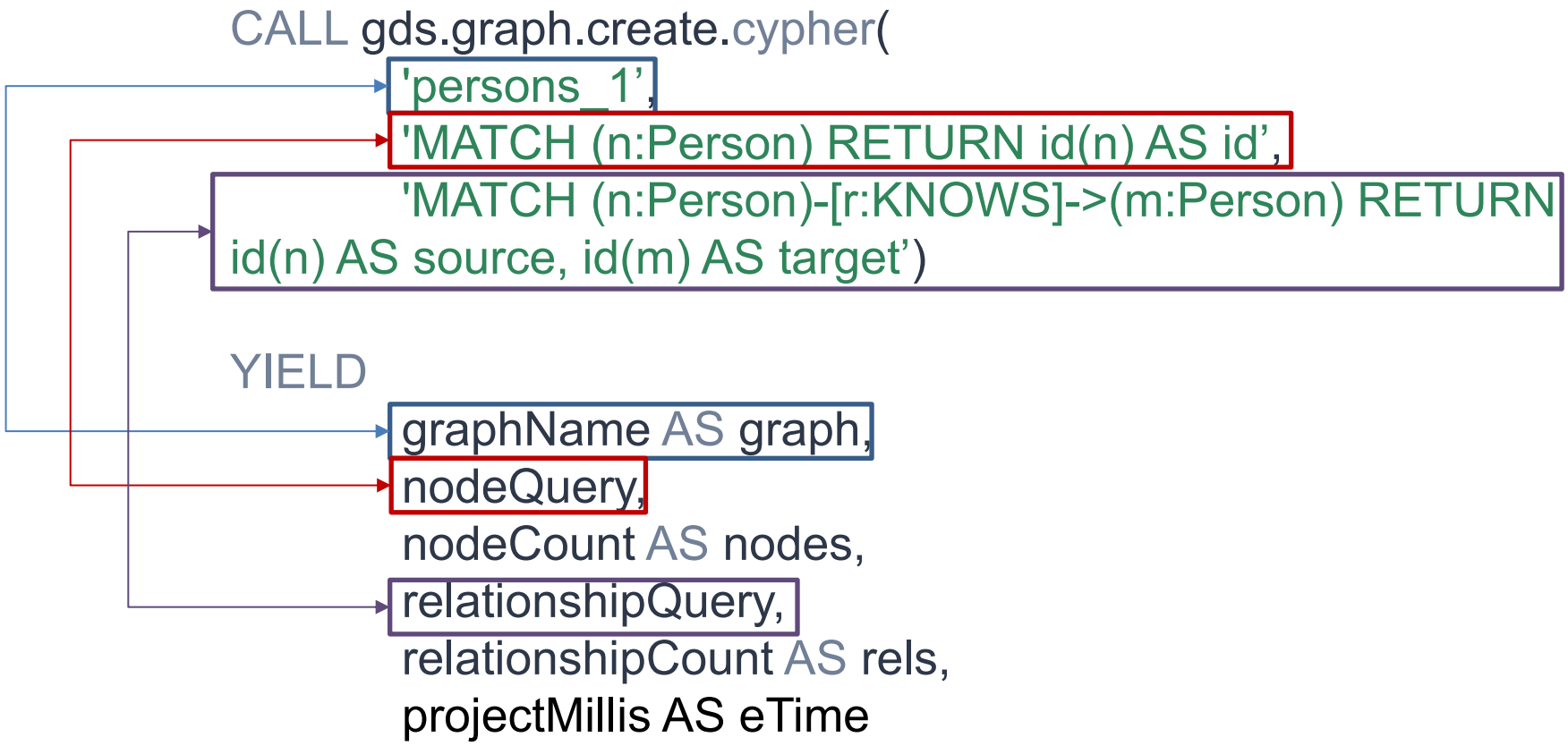
Graph Catalog

- Los algoritmos en GDS se ejecutan sobre vistas o **proyecciones** del grafo principal.
- Estas proyecciones se cargan en memoria cuando se invoca un algoritmo sobre ellas.
- Pueden ser proyecciones anónimas o con nombre.
- Se pueden crear distintas proyecciones, todas ellas formarán parte del **Graph Catalog** y podemos referirnos a ellas mediante su nombre.

Vista en memoria

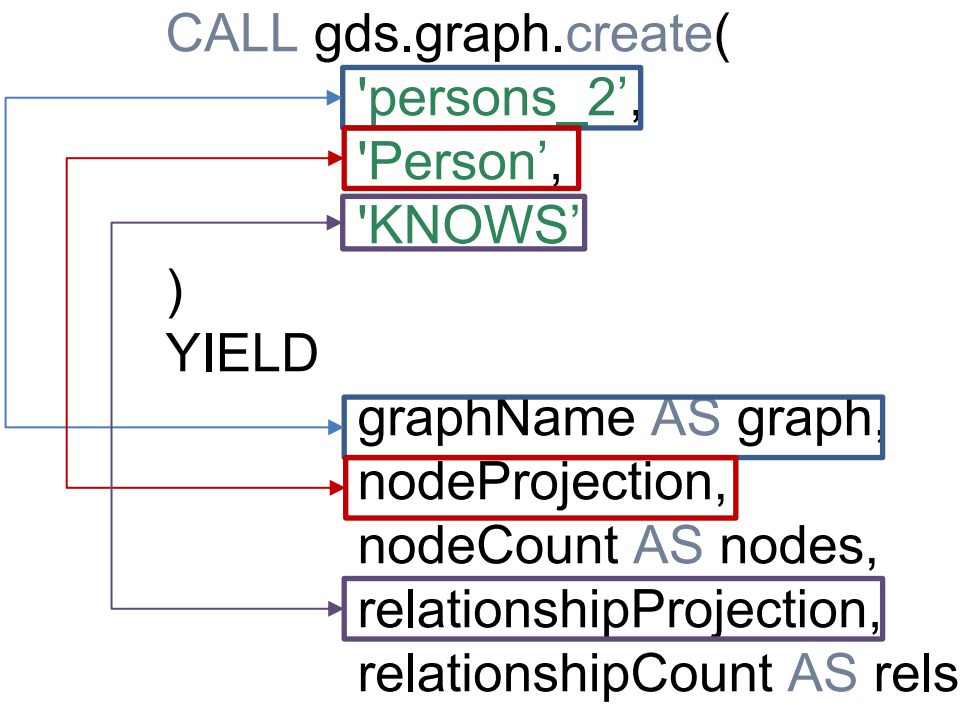
Graph Catalog

- Creación de una proyección con nombre usando consultas Cypher:



Graph Catalog

- Creación de una proyección con nombre de forma *nativa*:



ALGORITMOS DE GRAFOS EN NEO4J CON GDS

Algoritmos de grafos en Neo4J con GDS

- Agrupados en:
  - **Centrality**
  - **Community detection**
  - **Similarity**
  - **Path finding**
  - **Node embeddings**
  - **Topological link prediction**

Buscad ejemplos de algunos de estos algoritmos.

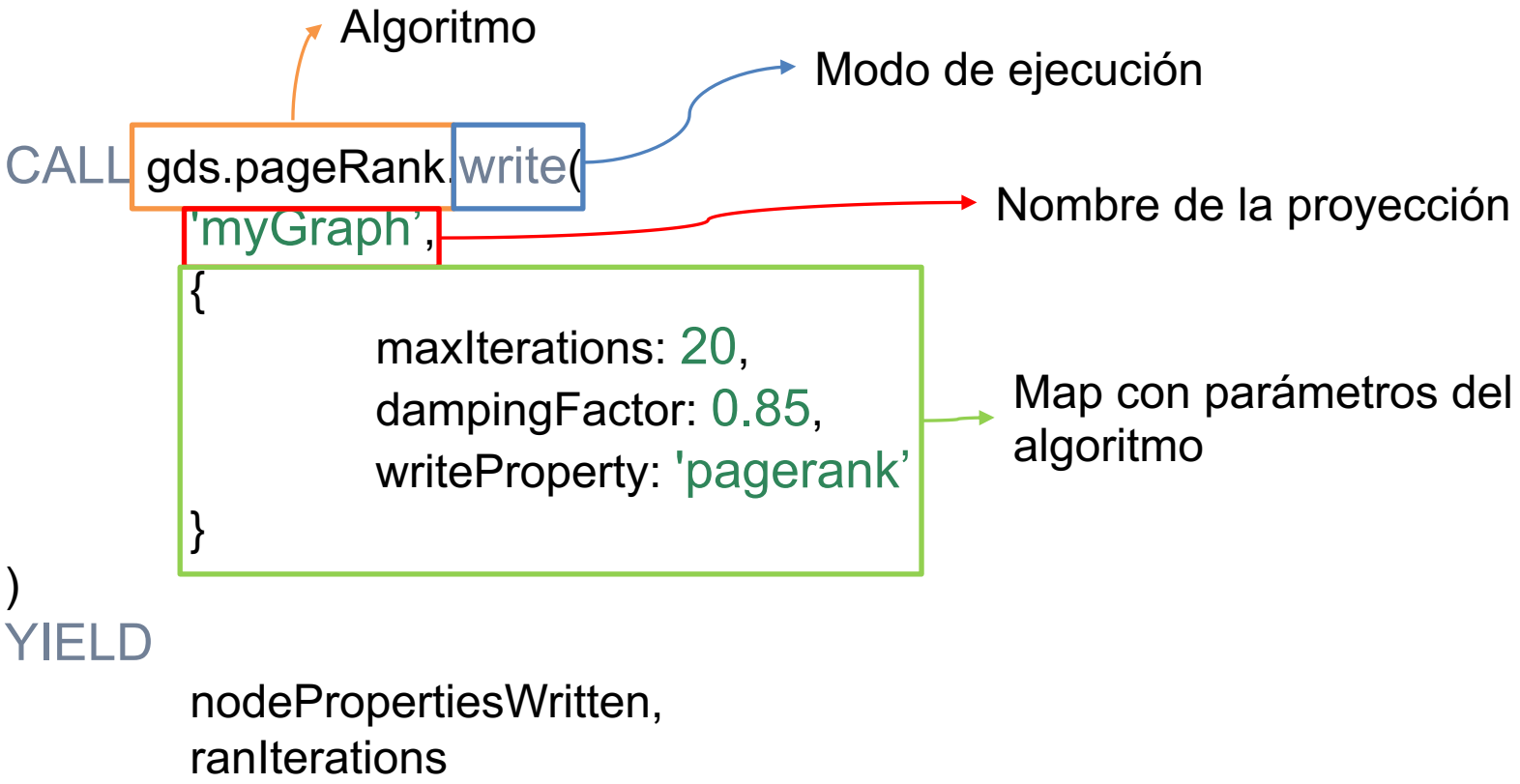
Algoritmos de grafos en Neo4J con GDS

- Modos de ejecución:
  - **stream**: devuelve los resultados en forma de stream.
  - **stats**: devuelve un registro con estadísticas resumidas.
  - **mutate**: escribe los resultados en la proyección sobre la que se ejecuta, y devuelve lo que ***stats***
  - **write:** escribe los resultados en la base de datos de Neo4J y devuelve lo que ***stats.***

Adicionalmente, podemos añadir la directiva **estimate**, que realiza una estimación de la memoria necesitada por el algoritmo.

Algoritmos de grafos en Neo4J con GDS

- Ejemplo de llamada: PageRank



¿MACHINE LEARNING CON GDS?

¿Machine learning con GDS?

- Sí pero no… veamos:
  - Buenas ideas e intenciones
    - Pipeline catalog
    - Model catalog
    - Uso de algoritmos para obtener features
  - Pero poco más
    - Pocos algoritmos
    - Poca documentación
    - Muy inmaduro
    - Mejores opciones en el mercado

Ejemplo

- Carga de datos de GOT:

```
LOAD CSV WITH HEADERS FROM
"https://www.macalester.edu/~abeverid/data/stormofswords.csv" AS row
MERGE (src:Character {name: row.Source})
MERGE (tgt:Character {name: row.Target})
MERGE (src)-[r:INTERACTS]->(tgt)
ON CREATE SET r.weight = toInteger(row.Weight)
```

Ejemplo

- Creación de proyección:

```
CALL gds.graph.create(
        'got1',
        'Character',
        'INTERACTS'
)
YIELD
        graphName AS graph,
        nodeProjection,
        nodeCount AS nodes,
        relationshipProjection,
        relationshipCount AS rels
```

Ejemplo

- Ejecución de PageRank en streaming:

```
CALL gds.pageRank.stream(
        'got1',
        {
        maxIterations: 20,
        dampingFactor: 0.85
        }
)
```

Ejemplo

- Ejecución de PageRank guardando resultado en el grafo:

```
CALL gds.pageRank.write(
        'got1',
        {
                maxIterations: 20,
                dampingFactor: 0.85,
                writeProperty: 'pagerank'
        }
)
YIELD
        nodePropertiesWritten,
        ranIterations
```

Ejemplo

- ¿Cómo sería con pesos?

```
CALL gds.graph.create(
        'got2',
        'Character',
        'INTERACTS',
        {
            relationshipProperties: 'weight'
        }
)
YIELD
        graphName AS graph,
        nodeProjection,
        nodeCount AS nodes,
        relationshipProjection,
        relationshipCount AS rels
```

Ejemplo

- Ejecución de PageRank con pesos:

```
CALL gds.pageRank.write(
        'got2',
        {
                maxIterations: 20,
                dampingFactor: 0.85,
                relationshipWeightProperty: 'weight',
                writeProperty: 'wPR'
        }
)
YIELD
        nodePropertiesWritten,
        ranIterations
```

Referencia

https://neo4j.com/docs/graph-data-science/current/