

Cypher - World Trades

Para el desarrollo de esta actividad, hemos usados como fuente los datos de exportaciones de 2016 de algunos productos. Los datos originales pueden ser consultados y descargados de <https://resourcetrade.earth/data>

En la base de datos que se ha compartido se han creado previamente todos los nodos de los países y los datos de sus exportaciones de algunos productos, los cuales son banano, crudo, diamantes, flores, carbón, café y langostas.

Para crear los nodos hemos usado sentencias similares a la siguiente, una por cada país:

CREATE

```
(:Country {code:"COL", name:"Colombia", population:"48653419", region:"South America"})
```

Las relaciones las creamos con sentencias similares a la siguiente:

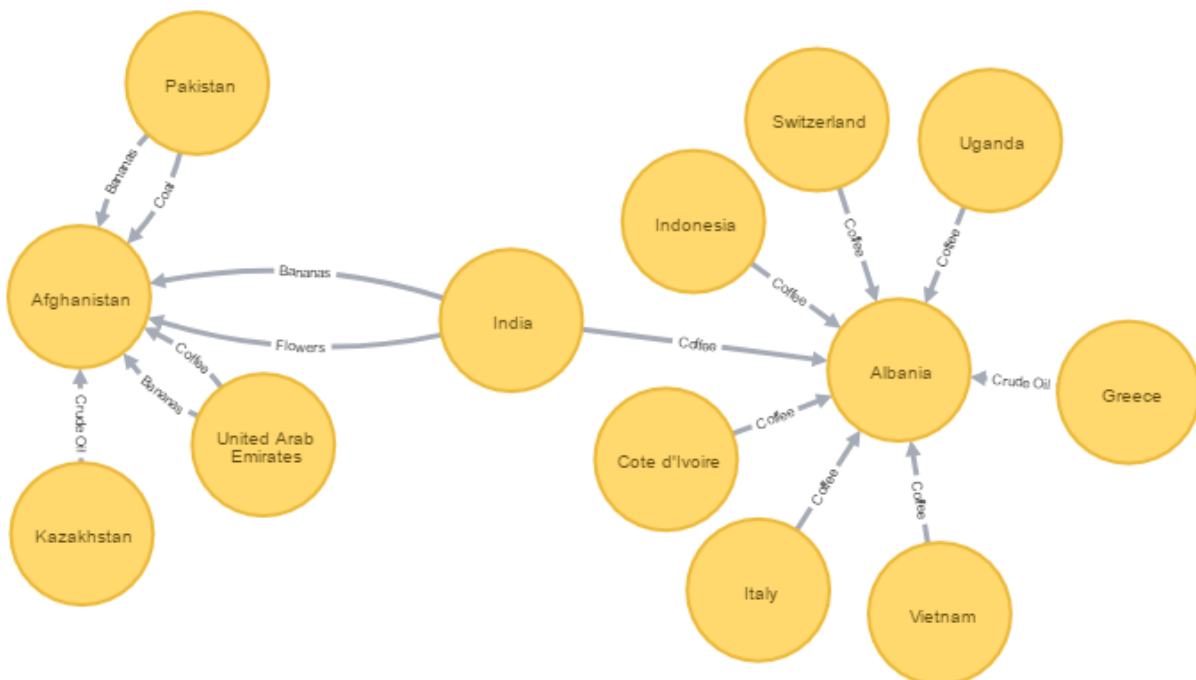
```
MATCH (exporter:Country), (importer:Country)
```

```
WHERE exporter.code = "BRA" AND importer.code = "COL"
```

```
CREATE (exporter)-[:EXPORTS
```

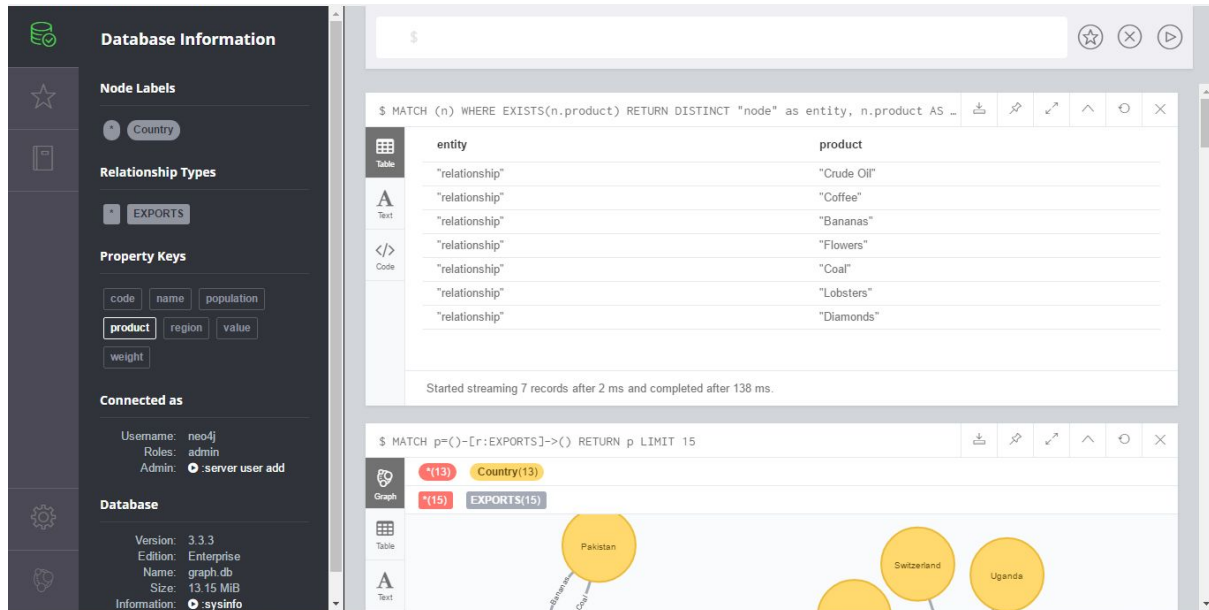
```
{product:"Coffee",value:"1056.2684",weight:"608.4633"}]->(importer);
```

De forma general el grafo tiene la siguiente estructura:



Actividad

1. Inicie sesión en el browser de Neo4j y explore los datos haciendo clic en las etiquetas de las propiedades de los datos ubicadas en la parte izquierda de la interfaz para familiarizarse un poco con los datos que están disponibles en la base de datos.



2. Para consultar algunos de los datos de los países ejecute la siguiente consulta:

```
MATCH (pais)
RETURN pais.name, pais.population
```

3. Consulte los datos de un país exportador, para este ejemplo hemos escogido Samoa, puede probar con otro país teniendo en cuenta los nombres y propiedades de los países registrados en la base de datos, tenga en cuenta los resultados de la consulta anterior

```
MATCH (exportador)-[relacion:EXPORTS]->(importador)
WHERE exportador.name = 'Samoa'
RETURN exportador, relacion, importador LIMIT 4
```

4. Neo4j permite especificar los datos de los filtros usando una sintaxis diferente, la misma consulta anterior puede definirse como:

```
MATCH (e {name:'Samoa'})-[r:EXPORTS]->(i)
RETURN e,r,i LIMIT 4
```

En este tipo de consulta los datos de los filtros se especifican directamente en la relación, el primer tipo de sintaxis puede ser más cómoda para las personas familiarizadas con SQL

5. Cuando se desean conocer valores específicos de las propiedades, se puede modificar la consulta de la siguiente forma:

```
MATCH (e)-[r]->(i)
WHERE e.name = 'Samoa'
RETURN e.name, r.product, r.value, i.name LIMIT 4
```

6. Si revisamos las consultas anteriores, podemos ver que todas ellas retornan varios objetos (e, r, i), que representan los datos de la siguiente forma:

e = país exportador
r = la relación entre los países (Exporta)
i = el país importador

Cuando se quiere retornar todos los datos de la relación se puede crear un objeto adicional que los almacena y retornar únicamente este valor, este tipo de sintaxis puede resultar muy útil cuando se manejan consultas más complejas que involucran varios objetos, la consulta quedaría así:

```
MATCH p = (e)-[r{product:'Oil'}]->(i)
WHERE e.name = 'Samoa'
RETURN p LIMIT 4
```

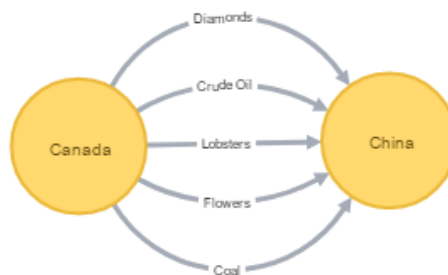
7. Si queremos conocer los datos de un país importador, la consulta sería la siguiente:

```
MATCH (e)-[r:EXPORTS]->(i)
WHERE i.name = 'China'
RETURN e, r, i LIMIT 12
```



8. Consultar los datos de exportación entre dos países:

```
MATCH (e)-[r:EXPORTS]->(i)
WHERE e.name = 'Canada'
AND i.name = 'China'
RETURN e, r, i
```



9. Consultar los países y el número de tipos de exportaciones de cada uno, para visualizar mejor los datos escoja la vista de texto:

```
MATCH (e)-[r:EXPORTS]->(i)
WITH e, COUNT(r) as exportaciones
ORDER BY exportaciones DESC
RETURN e, exportaciones
```

\$ MATCH (e)-[r:EXPORTS]->(i) WITH e, count(r) as exportaciones ORDER BY exportacion...

"e"	"exportaciones"
{ "name": "Netherlands", "code": "NLD", "region": "Europe", "population": 17018408 }	568
{ "name": "United States", "code": "USA", "region": "North America", "population": 323127513 }	396
{ "name": "United Kingdom", "code": "GBR", "region": "Europe", "population": 65637239 }	309
{ "name": "Colombia", "code": "COL", "region": "South America", "population": 48653419 }	301

10. Consultar y agrupar en una lista los países de Sur America

```
MATCH (e {region: "South America"})
RETURN collect(e.name)
```

11. Consultar los datos de exportaciones de Petróleo

```
MATCH(e)-[r:EXPORTS]->(i)
WHERE r.product = 'Oil'
RETURN e,r,i LIMIT 20
```

12. Consultar los países que exportan más de USD 500.000 en carbón a otro país

```
MATCH (e)-[r:EXPORTS{product:'Coal'}]->(i)
WHERE toFloat(r.value) > 500000
RETURN e,r,i
```

13. Consultar los países que exportan más de USD 6000000 en petroleo a otro país

```
MATCH (e)-[r:EXPORTS]->(i)
WHERE toFloat(r.value) > 6000000
AND r.product = 'Crude Oil'
RETURN e, r, i
```

14. Consultar los datos de exportación hacia China desde Sur América

```
MATCH (e)-[r:EXPORTS]->(i)
WHERE i.name = 'China'
AND e.region = 'South America'
RETURN e, r, i
```

15. Consultar los principales países exportadores de carbón

```
MATCH (e)-[r:EXPORTS{product:'Coal'}]->(i)
RETURN e, SUM(toFloat(r.value))
ORDER BY SUM(toFloat(r.value)) desc LIMIT 20
```

16. Con los nombres de los países de la consulta anterior muestre el grafo del top 3

```
MATCH (e)-[r:EXPORTS{product:'Coal'}]->(i)
WHERE e.name IN ["Indonesia","Australia","South Africa"]
RETURN e, r, i
```

17. Consultar los principales importadores de carbón

```
MATCH (e)-[r:EXPORTS{product:'Coal'}]->(i)
RETURN i, SUM(toFloat(r.value))
ORDER BY SUM(toFloat(r.value)) desc LIMIT 20
```

18. Con los nombres de los países de la consulta anterior muestre el grafo del top 4

```
MATCH (e)-[r:EXPORTS{product:'Coal'}]->(i)
WHERE i.name IN ["India","China","Malaysia","Germany"]
RETURN e,r,i
```