

## Neo4j

1. Encuentre todos los Riders del Team “TEAM KATUSHA”

```
MATCH (r:Rider) --> (t:Team)
WHERE (t.name = "TEAM KATUSHA")
RETURN *
```

o

```
MATCH (r:Rider) --> (t:Team {name: "TEAM KATUSHA"})
RETURN *
```



2. Teniendo en cuenta que el nodo Team tiene como atributo un país. ¿Cuántos equipos por países participan del TDF 2014?

```
MATCH (t:Team)
RETURN DISTINCT t.country, count(t.name) AS teamsPerCountry
```

t.country	teamsPerCountry
"ESP "	1
"GER "	1
"FRA "	5
"GBR "	1
"RUS "	2
"BEL "	2
"NED "	2
"ITA "	2
"USA "	3
"KAZ "	1
"AUS "	1
"SUI "	1

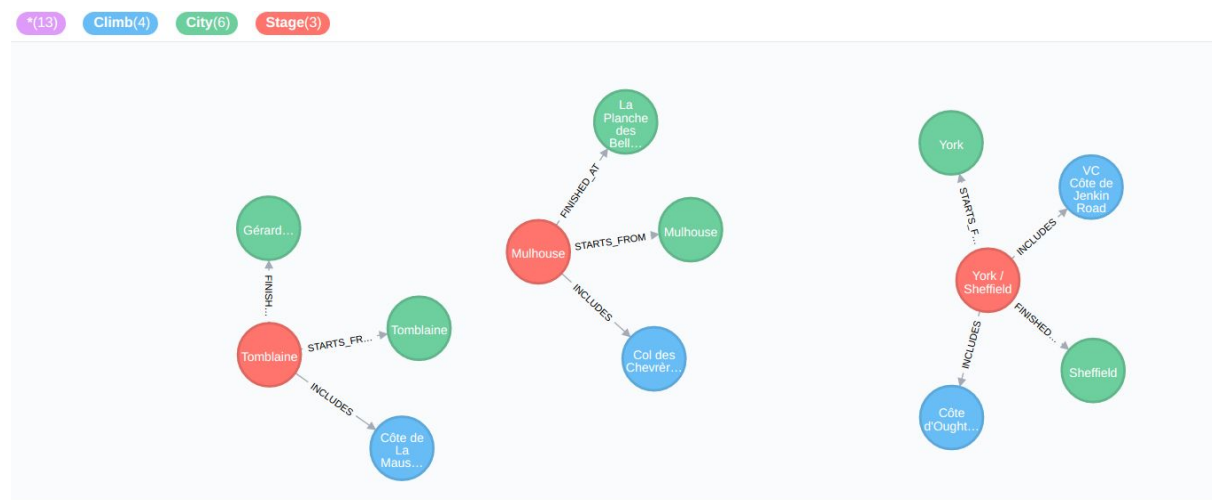
3. Teniendo en cuenta que el nodo Climb tiene como atributo los kilómetros de subida, ¿Cuántos kilómetros de subida recorrerá cada ciclista por estación?

MATCH (s:Stage)-[:INCLUDES]->(c:Climb)  
 RETURN s.name, sum(c.distance)

"Le Touquet-Paris-Plage / Lille Métropole"	2.3
"Arras / Reims"	2.4
"Maubourguet Pays du Val d'Adour / Bergerac"	1.3
"Tomblaine / Gérardmer La Mouselaine"	12.4
"Carcassonne / Bagnères-de-Luchon"	28.0
"Saint-Étienne / Chamrousse"	40.3
"Besançon / Oyonnax"	15.399999999999999
"Épernay / Nancy"	4.5
"Grenoble / Risoul"	65.6
"Saint-Gaudens / Saint-Lary Pla d'Adet"	39.099999999999994
"Pau / Hautacam"	35.300000000000004
"Leeds / Harrogate"	9.1
"Mulhouse / La Planche des Belles Filles"	44.0
"Évry / Paris Champs-Élysées"	0.0
"York / Sheffield"	18.3
"Gérardmer / Mulhouse"	37.1
"Bourg-en-Bresse / Saint-Étienne"	30.6

4. El grafo con las ciudades de inicio y fin de una Stage que tienen subidas con índice de inclinación mayor a 9

MATCH (c1:City)<-[:STARTS\_FROM]-(s:Stage)-[:FINISHED\_AT]-(c2:City)  
 MATCH (s)-->(c: Climb)  
 WHERE c.averageSlope > 9  
 RETURN \*



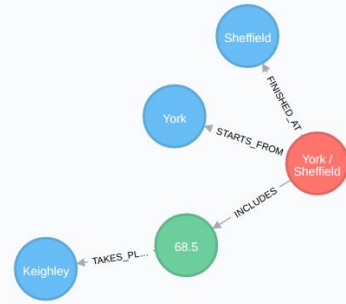
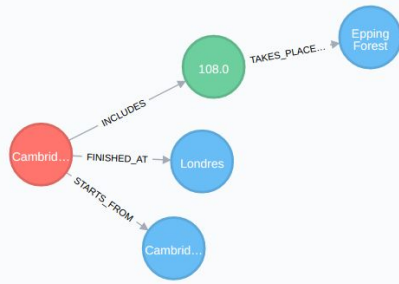
5. Teniendo en cuenta que las ciudades tienen un atributo size. Encuentre las estaciones que parten en una ciudad más grande que en la que terminan e incluye un "IntermediateSprint" en Inglaterra.

MATCH(c1:City)<-[:STARTS\_FROM]-(s1:Stage)-[:FINISHED\_AT]-(c2:City)  
 WHERE c1.size > c2.size  
 match(s1)-[:INCLUDES]->(i:IntermediateSprint)-[:TAKES\_PLACE\_IN]-(c3:City)  
 WHERE c3.country = "INGLATERRA"  
 return \*

City(6)

IntermediateSprint(2)

Stage(2)



## PANDAS

```
import pandas as pd
import numpy as np

info_alumnos = {
    'numero_alumno': ['1221', '232742', '32211', '46332', '566222', '67531', '71347'],
    'nombre': ['martin', 'juan', 'pedro', 'maria', 'samuel', 'flores', 'elisa'],
    'edad': [22,23,21,19,18,19,20]}

notas_alumnos = {'numero_alumno': ['1221', '232742', '32211', '46332', '566222', '67531', '71347'],
    'i1': [4.3,2.3,5.4,3.4,5.2,4.0,4],
    'i2': [2.1,4.2,2.2,5.4,5.2,4.5,5.5],
    'i3': [4.1,4.3,6.2,4.4,5.2,4.5,1]}

# Utilizando pandas imprima la tabla de alumnos con su nombre y notas

frame1 = pd.DataFrame(info_alumnos)
frame2 = pd.DataFrame(notas_alumnos)
#print(pd.merge(frame1, frame2, on="numero_alumno")[["nombre", "i1", "i2", "i3"]])

# Ahora nos encontramos con la situacion de que martin boto el ramo y no figura en las notas
# las notas de los alumnos son las siguientes

notas_alumnos = {'numero_alumno': ['232742', '32211', '46332', '566222', '67531', '71347'],
    'i1': [2.3,5.4,3.4,5.2,4.0,4],
    'i2': [4.2,2.2,5.4,5.2,4.5,5.5],
    'i3': [4.3,6.2,4.4,5.2,4.5,1]}

frame2 = pd.DataFrame(notas_alumnos)
# print(pd.merge(frame1, frame2, on="numero_alumno", how='left'))

# Que hubiese pasado si hacemos how = "right"??

#Cual es el promedio por prueba?
#print(frame2.mean())

# Teniendo en cuenta que martin boto el ramo
# Entregue las notas de los alumnos con edad mayor a 20, que pasa con martin
# print(pd.merge(frame1[frame1["edad"] > 20], frame2, on="numero_alumno", how='outer'))

# Como queda con left, right, como queda con outer?

# Recordar como funcionan los filtros!!
# print(frame1[frame1["edad"] > 20])
```

## SQL

1. Mostrar el nombre del empleado que tiene el salario más alto en el departamento de 'VENTAS'.

```
SELECT nombre FROM Empleado AS E, Departamento AS D
WHERE E.depto_no = D.depto_no AND D.nombre_depto='VENTAS' AND E.salario = (
    SELECT max(salario) FROM Empleado AS E, Departamento AS D
    WHERE E.depto_no = D.depto_no AND nombre_depto='VENTAS' AS foo)
```

2. Mostrar el nombre, salario y nombre del departamento de los empleados que tengan el mismo oficio que "Leonel Alfonso Esquivel"

```
SELECT nombre, salario, nombre_depto FROM Empleado AS E, Departamento AS D
WHERE E.dept_no=D.dept_no AND E.oficio =
    (SELECT oficio from EMPLEADO AS E
    WHERE E.nombre= 'Leonel Alfonso Esquivel' AS foo)
```

3. Encuentre los empleados que trabajan con todos los empleados de la tabla Empleado.

```
SELECT nombre FROM Empleados WHERE Empleados.eid NOT IN (
    SELECT eid FROM (
        SELECT e1.eid, e2.eid FROM Empleado e1, Empleado e2
        WHERE e1.eid <> e2.eid
        EXCEPT (
            SELECT eid1, eid2 FROM Trabaja
            UNION
            SELECT eid2, eid1 FROM Trabaja)
    ) AS no_trabajan_con_todos
)
```