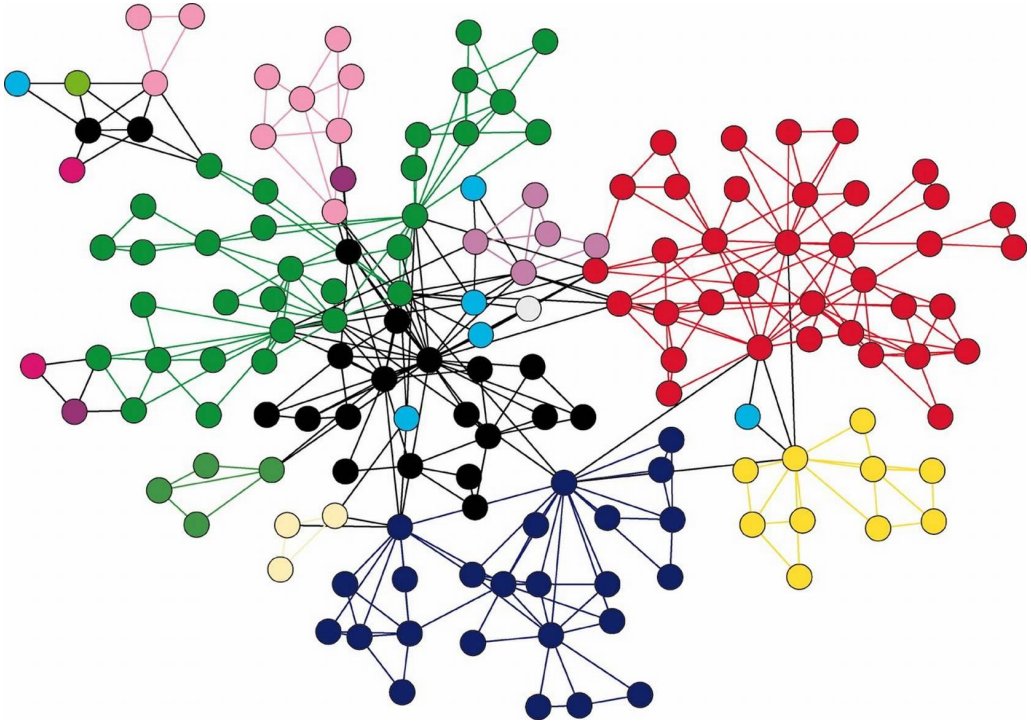
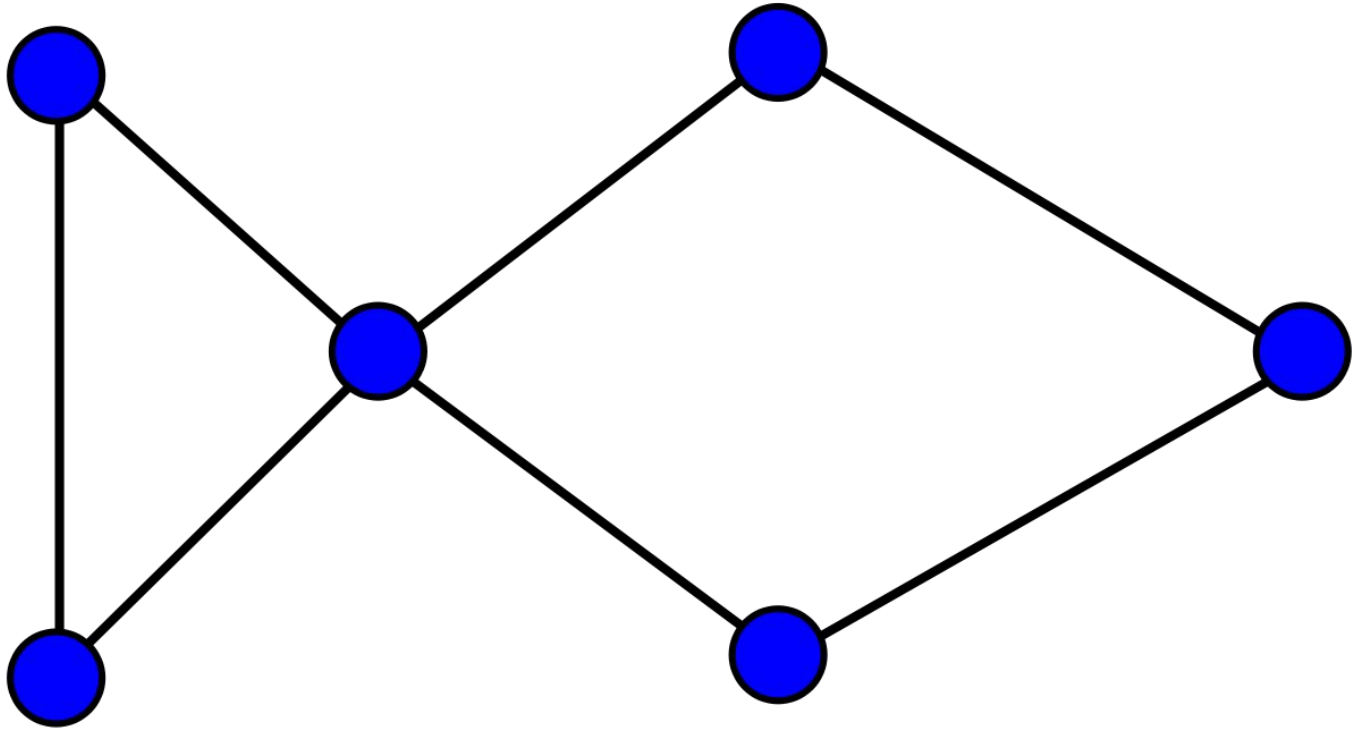


Neo4j

¿QUE ES?

Es una base de datos NOSQL orientadas a grafos

¿Qué es un Grafo?



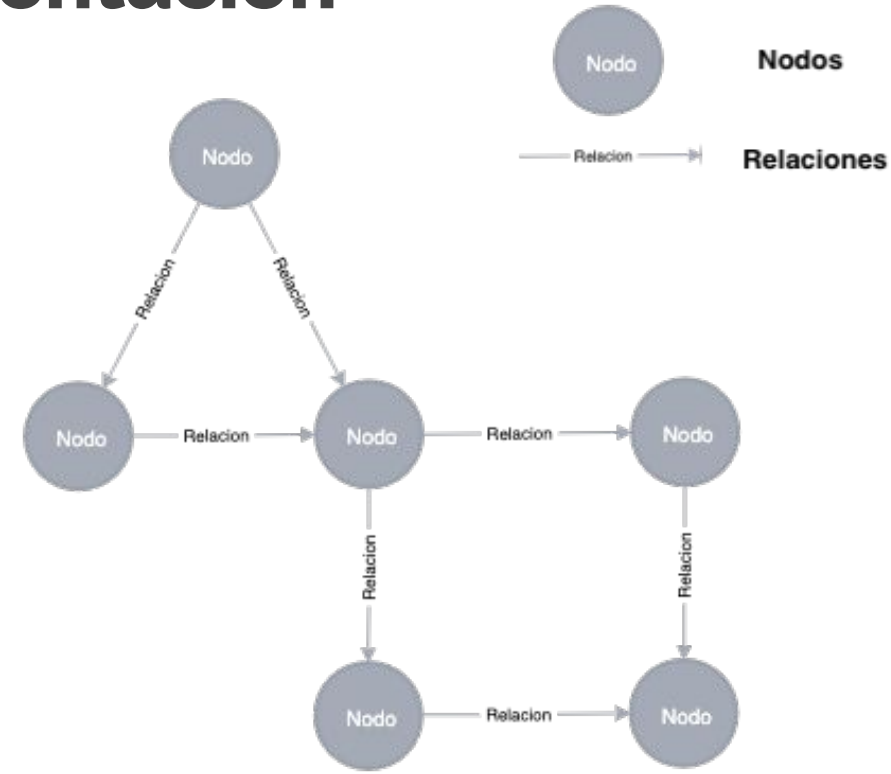
Tipos

- **Grafos no dirigidos:** los nodos y las relaciones son intercambiables, su relación se puede interpretar en cualquier sentido. Las relaciones de amistad en la red social Facebook, por ejemplo, son de este tipo.
- **Grafos dirigidos:** los nodos y la relaciones no son bidireccionales por defecto. Las relaciones en Twitter son de este tipo. Un usuario puede seguir a determinados perfiles en esta red social sin que ellos le sigan a él.
- **Grafos con peso:** en este tipo de gráficas las relaciones entre nodos tienen algún tipo de valoración numérica. Eso permite luego hacer operaciones.
- **Grafos con etiquetas:** estos grafos llevan incorporadas etiquetas que pueden definir los distintos vértices y también las relaciones entre ellos. En Facebook podríamos tener nodos definidos por términos como ‘amigo’ o ‘compañero de trabajo’ y la relaciones como ‘amigo de’ o ‘socio de’.
- **Grafos de propiedad:** es un grafo con peso, con etiquetas y donde podemos asignar propiedades tanto a nodos como relaciones (por ejemplo, cuestiones como nombre, edad, país de residencia, nacimiento). Es el más complejo.

Características:

- Los Nodos y Relaciones pueden tener propiedades
- Las Relaciones conectan Nonos
- Los Nodos pueden tener un tipo

Representación



Diferencias con las Relacionales

Bases de datos Relacionales	Bases de datos de Grafos
Tablas	Grafos
Filas	Nodos
Columnas y datos	Propiedades
Claves foráneas	Relaciones

HABEMUS ACID

A diferencia de otras bases de datos NOSQL, Neo4j cumple las propiedades ACID

- ATOMICIDAD
- CONSISTENCIA
- AISLAMIENTO
- DURABILIDAD

De forma sencilla:

Node

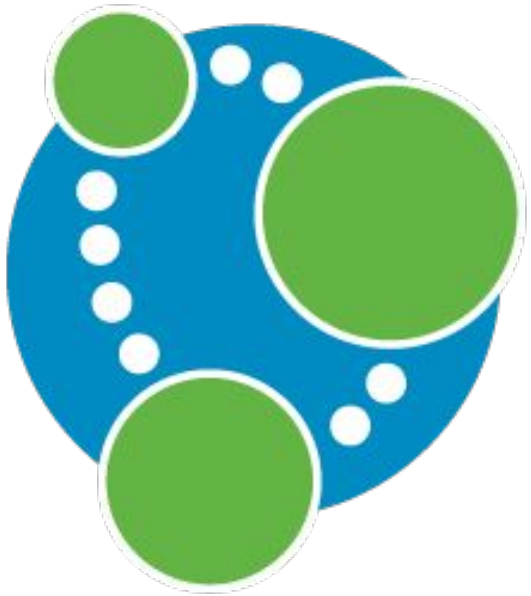
Representa un dato sencillo.

Relationship

Establece un vínculo entre dos nodos. Tiene dirección.

Labels para asignar roles o tipos, **properties** para guardar valores sencillos.

El lenguaje de consulta Cypher



Comandos básicos

CREATE

Hace falta decir para qué sirve?

MATCH

Establece un punto específico en el grafo, donde se llevará a cabo otra operación.

RETURN

Retornar valores.

CREATE

```
CREATE(person)
CREATE(person:Person)
CREATE(person:Person {name: "Roberto"})
```

=> Crea un nodo sin tipo y sin propiedades

=> Crea un nodo con el tipo **Person**

=>… **Person** y con la propiedad **name**

MATCH

```
MATCH(n) RETURN n
MATCH(p:Person) RETURN p
MATCH(p:Person {name: "Roberto"}) RETURN p
```

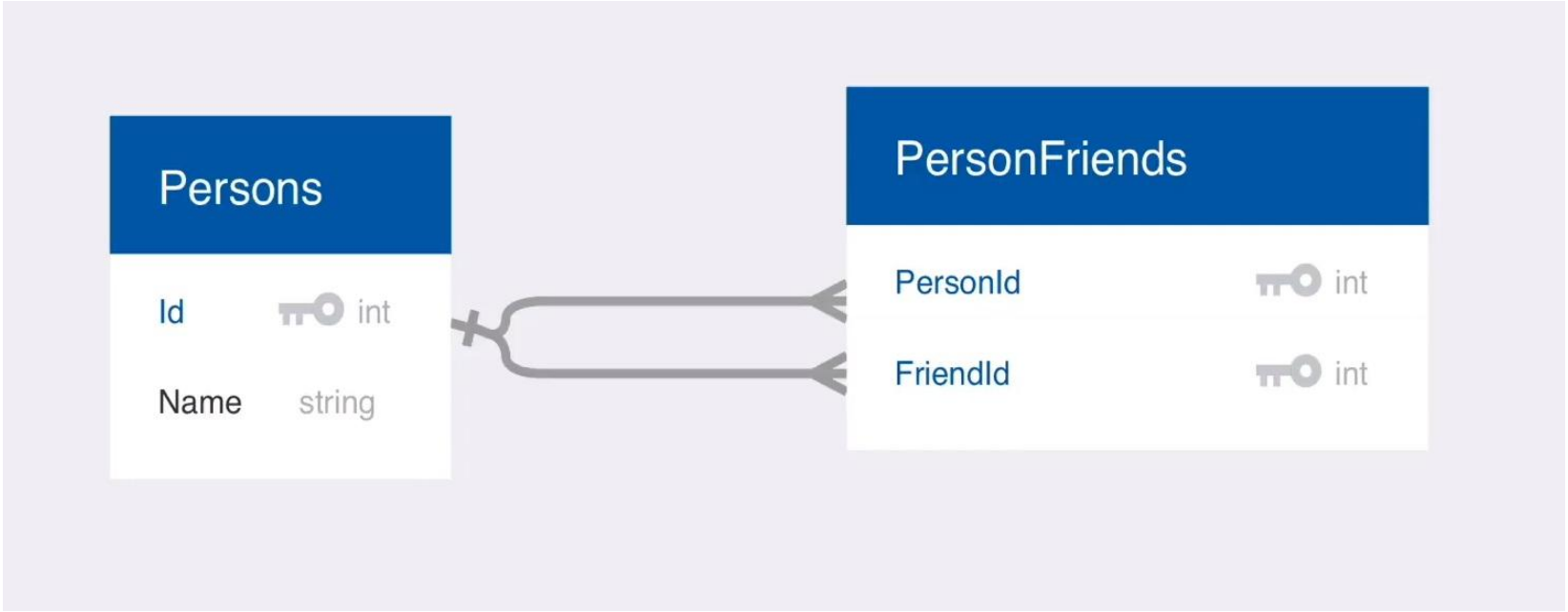
=> Retorna todos los nodos

=> Retorna todos los nodos del tipo **Person**

=> Retorna los nodos de tipo **Person** y con la propiedad **name** = "Roberto"

Comparado con SQL:

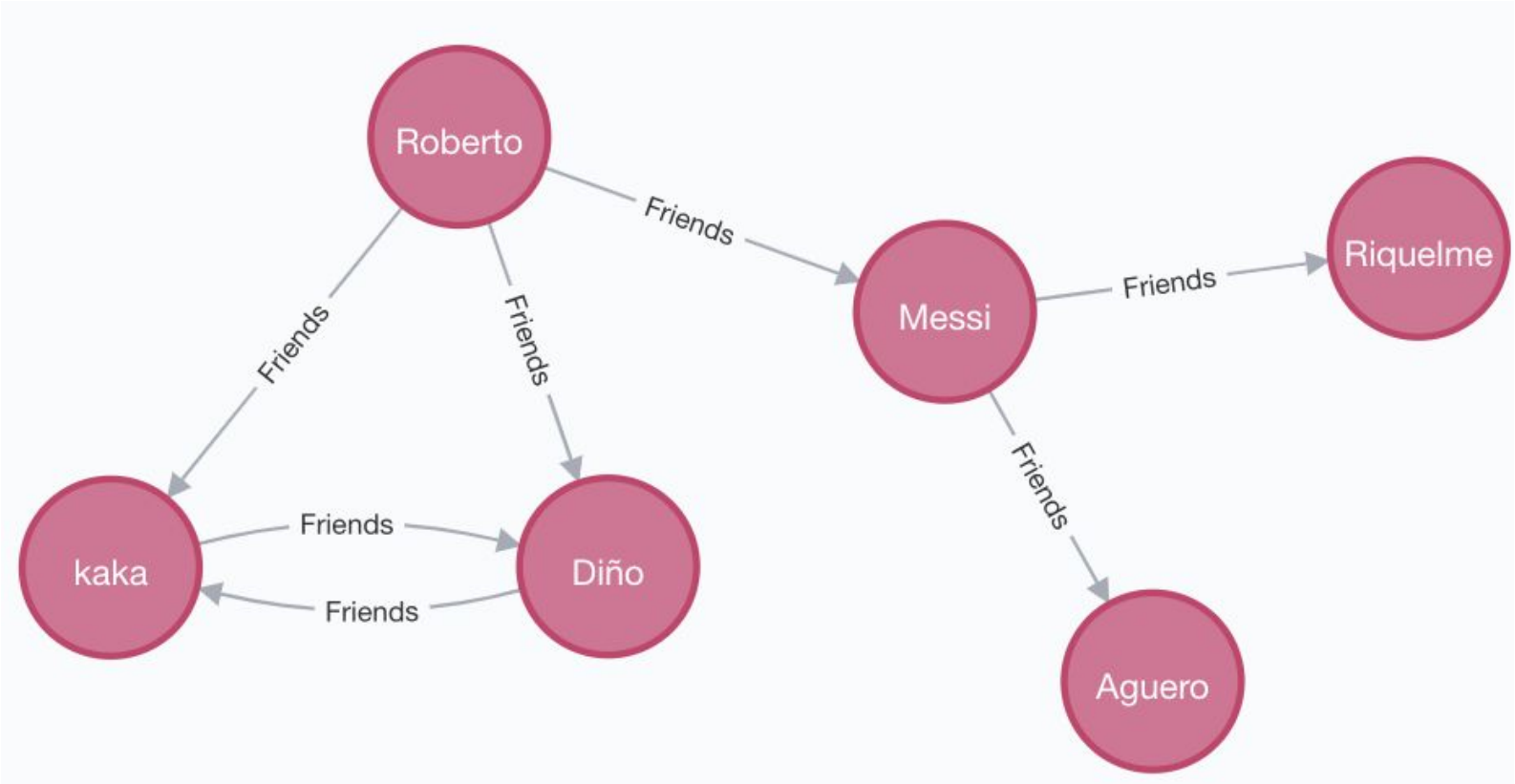
SELECT * FROM Person WHERE name = "Roberto"	MATCH (person:Person {name: "Roberto"}) RETURN person


Persons
Id
int
Name
string
PersonFriends
PersonId
int
FriendId
int

Cómo podemos resolver mis amigos y los amigos de mis amigos?

```sql
SELECT p1.Name AS Name, p2.Name FriendName
FROM PersonFriends pf1
     JOIN Persons p1 ON
          pf1.PersonId = p1.Id
     JOIN PersonFriends pf2 ON
          pf2.PersonId = pf1.FriendId
     JOIN Persons p2 ON
          pf2.FriendId = p2.Id
WHERE p1.Name = 'Roberto' AND
     pf2.FriendId <> p1.Id
```

```
MATCH (roberto:Person { name: "Roberto"})
MATCH (roberto)-[:Friends*2]-(friend)
RETURN roberto.name, friend.name
```


Roberto
Friends
Friends
Friends
Messi
Friends
Riquelme
Friends
kaka
Friends
Diño
Friends
Aguero

Comandos básicos

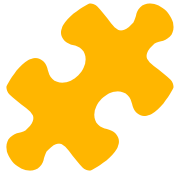
MERGE

Una mezcla entre el CREATE y el MATCH.

DELETE

Eliminar.

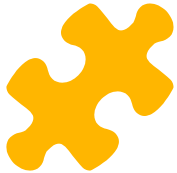
MERGE



```
MERGE(p:Person {name: "Roberto"})
```

Busca un nodo del tipo **Person** y **name** "Roberto" y si no lo encuentra, lo crea

DELETE



```
MATCH (n) DELETE n              => Elimina todos los nodos
MATCH (n) DETACH DELETE n       => Elimina todos los nodos y sus relaciones
```

Entonces:

Comandos:

- CREATE
- MATCH
- RETURN
- MERGE
- DELETE
- SET

Casos de uso

- Redes Sociales
- Recomendaciones de productos en tiempo real
- Modelos de red
- Gestionar accesos (ACL)

¿Neo4j es una base de datos relacional ?

¿Donde estaría Neo4j?