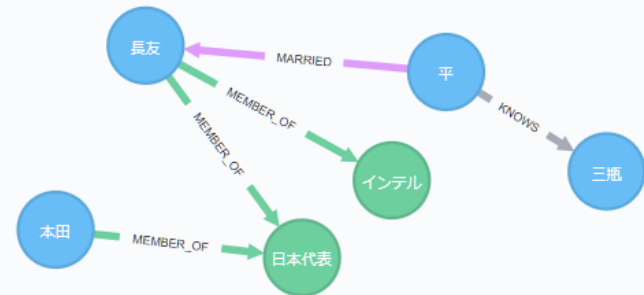


データ要素 - まとめ

- **Node**
 - Label \Rightarrow データレコード
 - プロパティ \Rightarrow Nodeを分類する
 - プロパティ \Rightarrow データレコード内の個別データ
- **Relationship**
 - Type \Rightarrow Node間の関係を示す
 - Type \Rightarrow Relationshipを分類する
 - プロパティ \Rightarrow Relationshipへも付与できる



データ要素の表現 - まとめ

- **Node...()**で括る

```
(:Person {name:"長友", age:31})
```

ラベル名

プロパティ

- **Relationship...[]**で括る

```
[:MEMBER_OF {since:2011}]
```

タイプ名

プロパティ

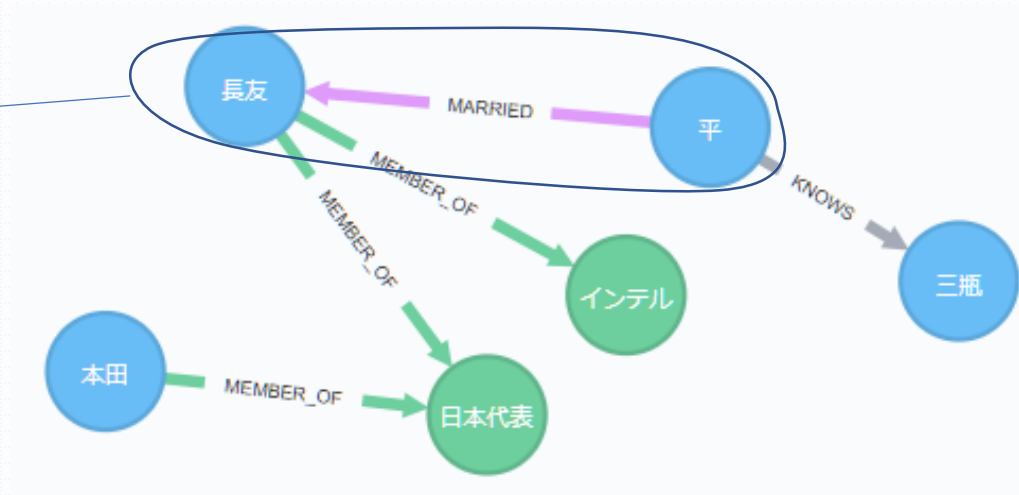
パスをCypherで表現する (1)

【例】

`() - [] -> ()`

`() <- [] - ()`

`() - [] -> () <- [] - ()`



`(:Person {name:"長友"}) <-[:MARRIED] - (:Person {name:"平"})`

・・・「長友と結婚した平愛梨」

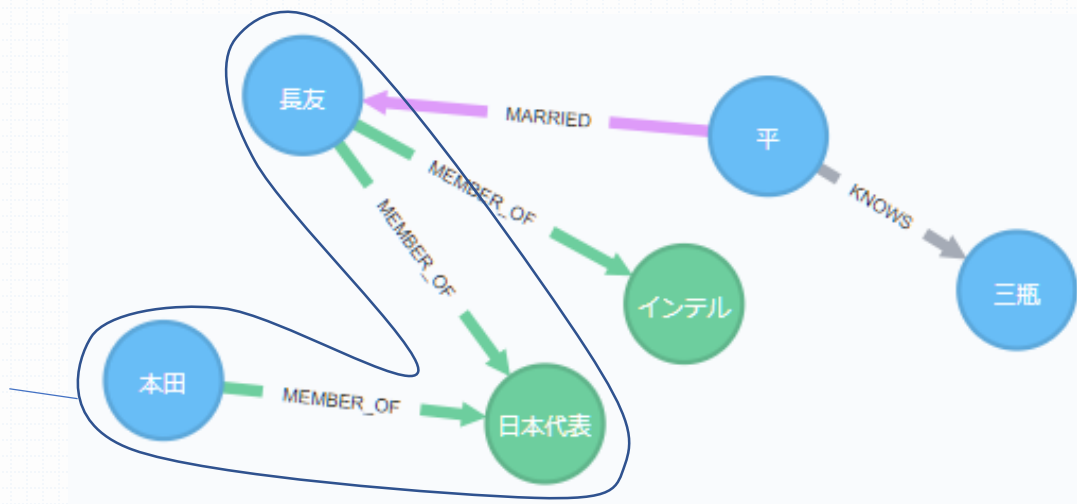
パスをCypherで表現する（1）

【例】

`()-[]->()`

`()<-[]-()`

`()-[]->()<-[]-()`



```
(:Person {name:"長友"})-[:MEMBER_OF]->(:Team)
<-[:MEMBER_OF]-(:Person {name="本田"})
```

- ・・・「長友が所属するチームに所属する本田」
＝「長友のチームメイトの本田」

パスをCypherで表現する (2)

- パスの長さを指定

```
()-[:KNOWS*2]->()
```

```
()-[:KNOWS*1..3]->()
```

```
()-[*]->()
```

- 長さ0のパス=Nodeのみマッチ

```
()-[:KNOWS*0..3]->()
```

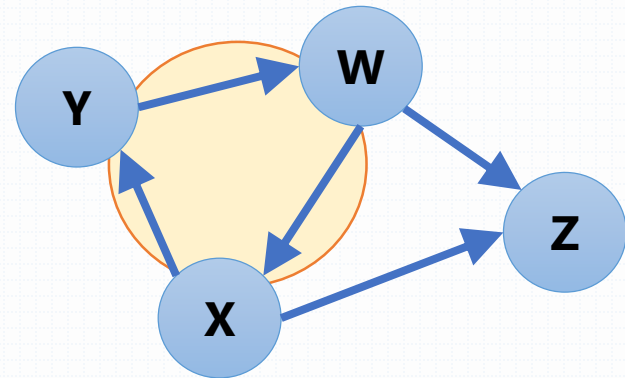
- 向きの指定は省略可能

```
()-[:KNOWS]-()
```

パスをCypherで表現する (3)

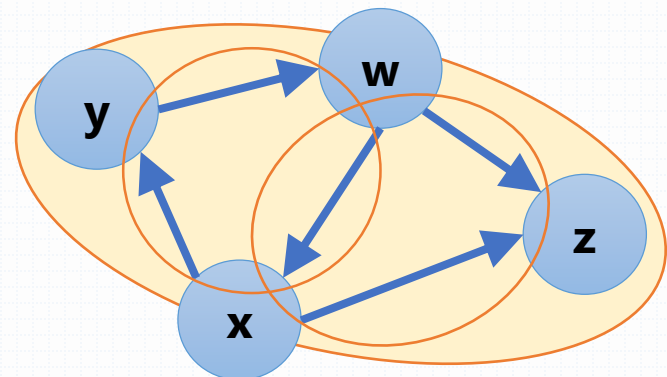
- ループの表現 (有向)

`(s)-[:GOTO*1..]->(s)`



- ループの表現 (無向)

`(s)-[:KNOWS*1..]-(s)`



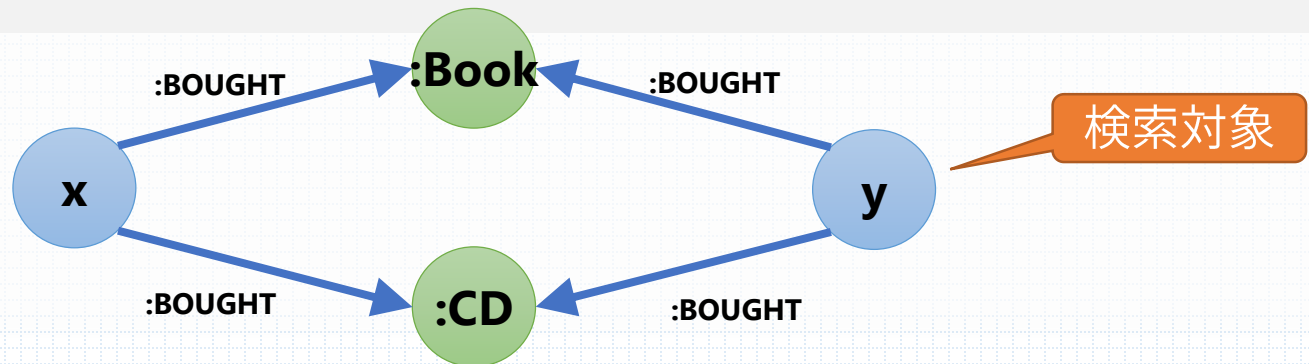
パスをCypherで表現する（4）

- xさんと同じ本とCDを買ったyさんを探す

```
MATCH (x)-[:BOUGHT]->(:Book)<-[:BOUGHT]-(y)
      -[:BOUGHT]->(:CD)<-[:BOUGHT]-(x)
RETURN y
```

or

```
MATCH (x)-[:BOUGHT]->(:Book)<-[:BOUGHT]-(y),
      (x)-[:BOUGHT]->(:CD)<-[:BOUGHT]-(y)
RETURN y
```



パスをCypherで表現する (5)

- xさんと同じ本を買ったyさんにCDをおすすめする

```
MATCH (x)-[:BOUGHT]->(:Book)<-[:BOUGHT]-(y)
      -[:BOUGHT]->(n:CD)<-[:BOUGHT]-(x)
RETURN n
```

or

```
MATCH (x)-[:BOUGHT]->(:Book)<-[:BOUGHT]-(y),
      (x)-[:BOUGHT]->(n:CD)<-[:BOUGHT]-(y)
RETURN n
```

