



Какво е графова СУБД?  
Примери за основни видове заявки и  
възможности на графовите СУБД

Webtex | отбор 3

# Какво е графова СУБД?

- База данни, специално проектирана и оптимизирана за съхранение и записване на графични данни.
- Графовите бази данни са вид NoSQL бази данни, замислени да адресират някои от ограниченията на релационните бази данни.

# Какво е специфичното на графовата СУБД?

- Графова база от данни пази възли и връзки, вместо таблици или документи. Информацията е запазена точно както бихме я скицирали на дъска.
- Данните се пазят без ограничение от предефиниран модел, позволявайки гъвкав начин на разглеждането и използването ѝ.





## Обходжане

- Заявките представляват обходжане на графа

# За какво са подходящи грфовите СУБД?

Социални  
мрежи

Анализиране  
на данни

Препоръчващи  
системи

Търсене на  
пътища

Други видове  
данни с  
много връзки

# Примери

- Социална мрежа
- Сайт за запознанства
- Онлайн магазин
- Сайт и мобилно приложение, агрегиращо онлайн концерти
- Анализ на поведението на потребители в сайт



# Каво е Neo4j

- Най-популярната графова база от данни
  - Отворен код (GPL)
- Предоставен за комерсиално използване

# Каво е Cypher

- Език за заявки към графови данни
  - Създаден специално за Neo4j
- OpenCypher - млад проект за разпространение на Cypher и към други СУБД
  - Декларативен език

# Пример

```
CREATE (doseOfHappiness:Movie {title:'Dose Of Happiness',  
released:2020, genre:'Romantic Drama'})
```

```
CREATE (Valentina:Person {name:'Валентина Каролева',  
born:1964})
```

```
CREATE (Dimitar:Person {name:'Димитър Николов', born:1967}),
```

```
(Valentina)-[:ACTED_IN {roles:['Весела']}]>(doseOfHappiness),
```

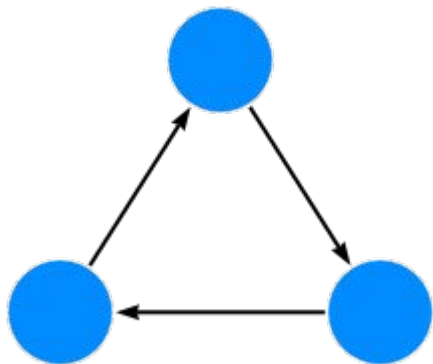
```
(Dimitar)-[:ACTED_IN {roles:['Жоро']}]>(doseOfHappiness)
```

```
MATCH (n) RETURN n
```



# Neo4j

## Основни концепции



### Възли

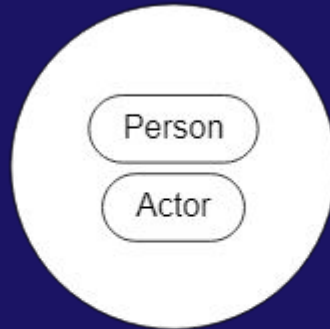
- Описват същности
- Имат 0 или повече етикети
- Имат свойства: двойки ключ-стойност

### Ребра

- Описват връзката между възел-източник и целеви възел
- Винаги имат посока
- Трябва да имат тип

# Възли

Самостоятелен  
възел:



name: 'Tom Hanks'  
born: 1956

```
CREATE (:Person:Actor {name: 'Tom Hanks', born: 1956})
```

## Етикети

Person Actor
name = 'Tom Hanks' born = 1956

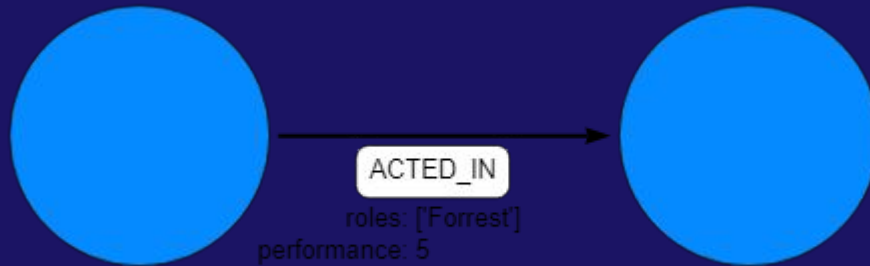
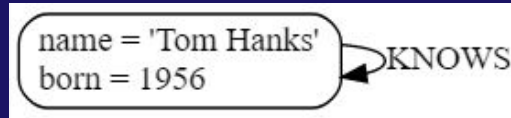
Movie
title = 'Forrest Gump' released = 1994

Person Director
name = 'Robert Zemeckis' born = 1951

→ групиране на възли в множества



# Ребра



- Тип
- Свойства

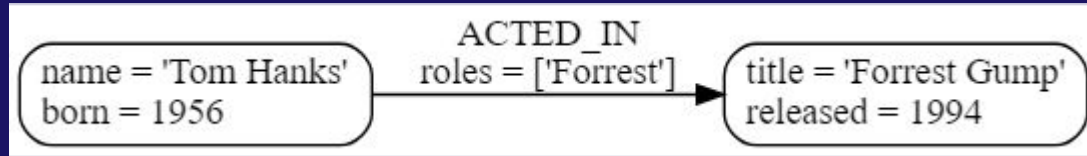
```
CREATE ()-[:ACTED_IN {roles: ['Forrest'], performance: 5}]->()
```

# Свойства

## Примери



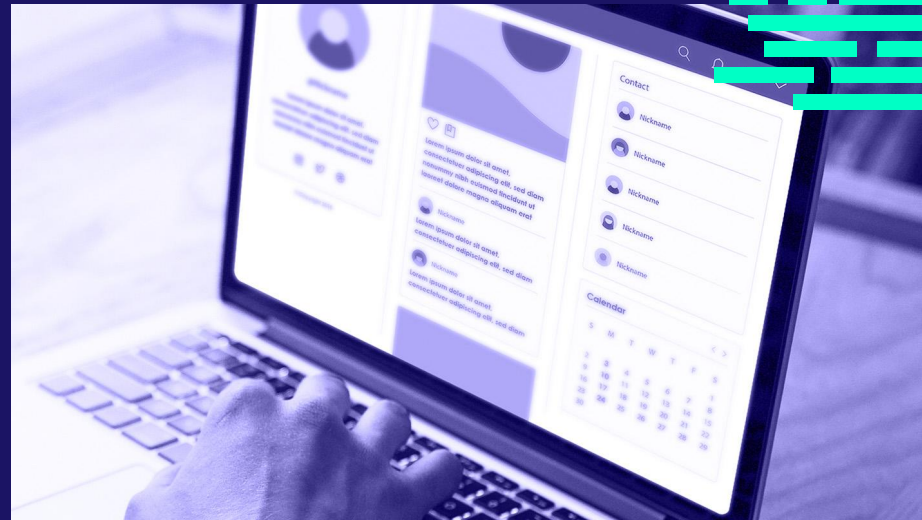
Тип на връзка  
между ребра



Свойства

```
CREATE (:Example {c: 'This is an example  
string', d: true, e: false})
```

```
CREATE (:Example {f: [1, 2, 3], g: [2.71,  
3.14], h: ['abc', 'example'], i: [true, true,  
false]})
```



# Демонстрация



```
CREATE (:Person:Actor {name: 'Tom Hanks', born: 1956})  
-[:ACTED_IN {roles: ['Forrest']}] -> (:Movie {title: 'Forrest  
Gump'}) <-  
[:DIRECTED]-(:Person {name: 'Robert Zemeckis', born: 1951})
```

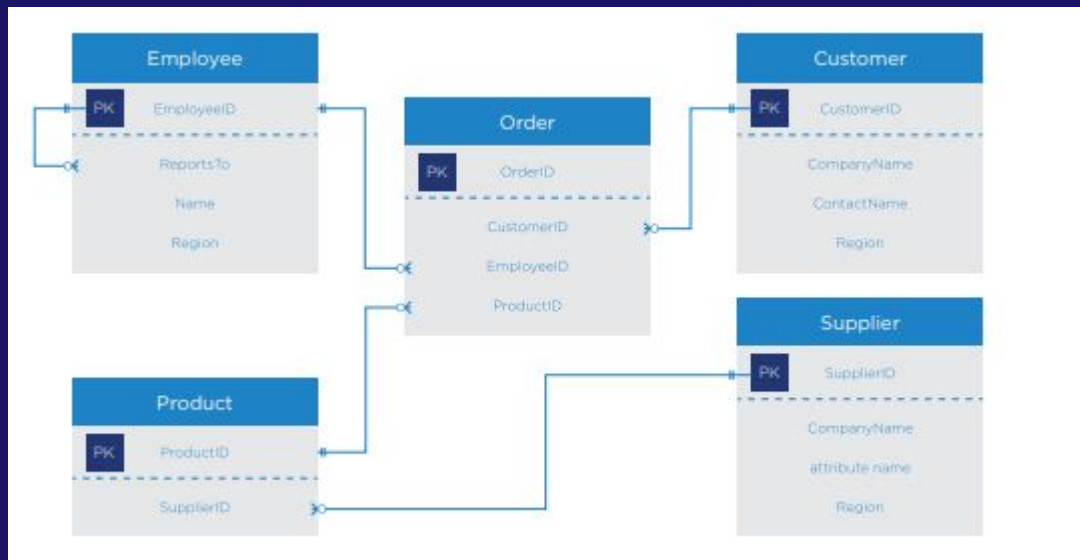
```
MATCH (n) RETURN n
```



# Релационна към графова база от данни

Преминаването от една база данни към друга се свежда до промяна на връзките в базата от един тип към друг.

Единица данна в релационната база от данни наричаме ред, докато в графовата - възел. По-конкретно можем да използваме външните ключове на релационната база от данни, за да направим ребрата, по който начин да трансформираме слабо свързаните записи на данни към логично създадени групи от възли.

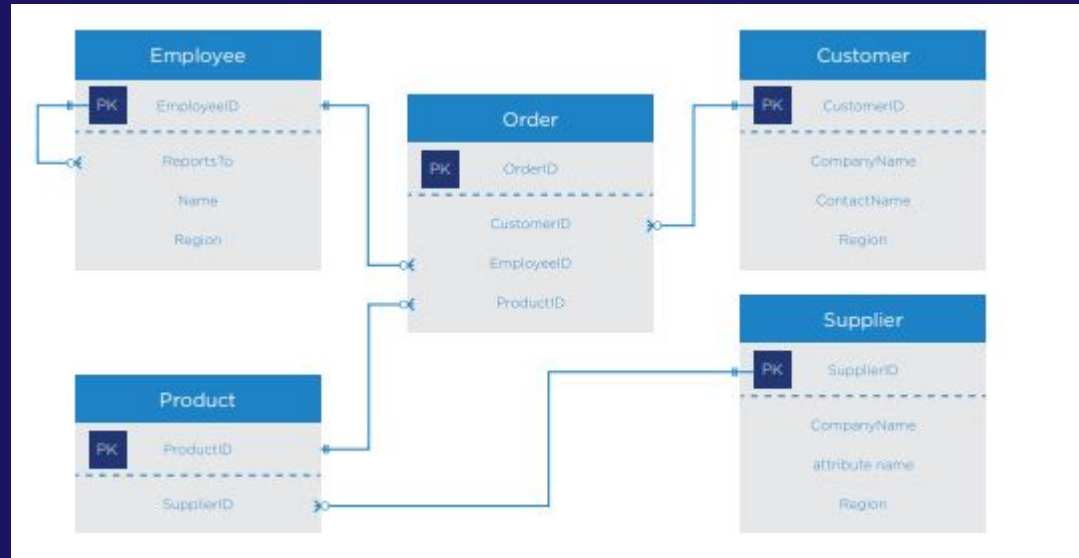


# Стъпка 1:

Първата стъпка е да идентифицираме типовете обекти.  
В този пример имаме 5 типа обекти:

- Employee
- Order
- Customer
- Product
- Supplier

Всяка таблица представлява различен тип информация, която ще манипулираме

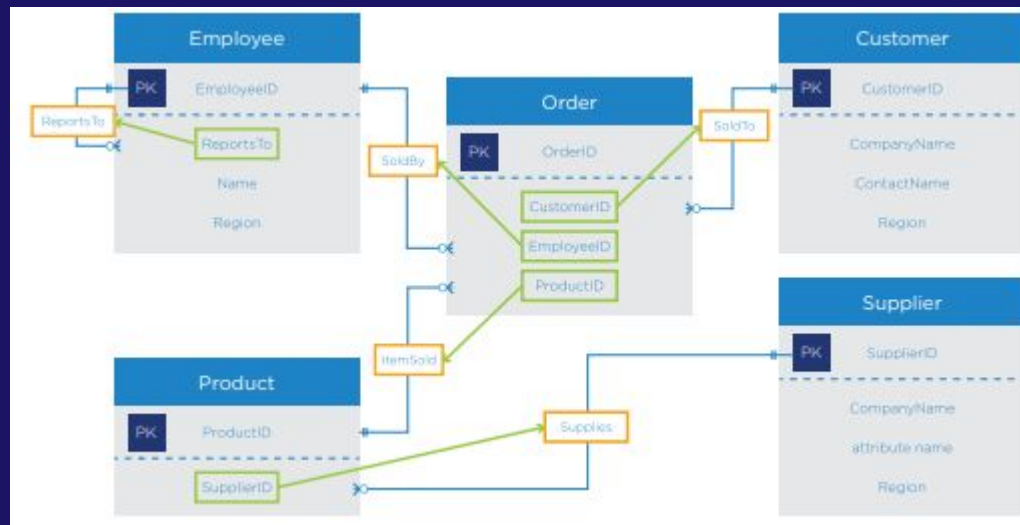




## Стъпка 2:

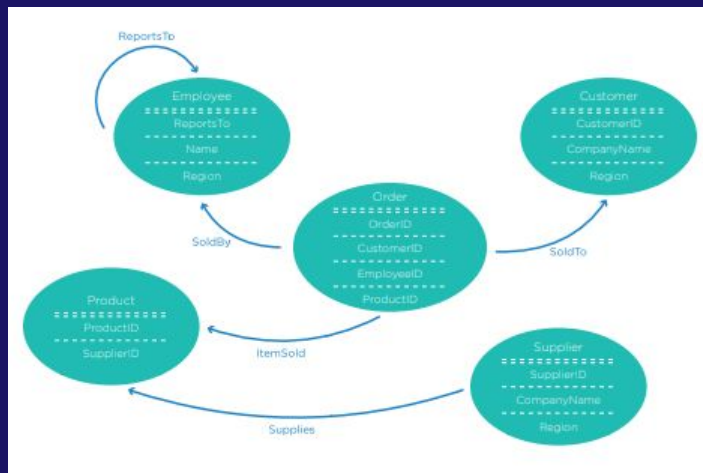
Във втора стъпка трябва да намерим семантични връзки между тези обекти

RELATION	LOGICAL LINK (ATTRIBUTE)
Order - SoldBy -> Employee	EmployID
Order - SoldTo -> Customer	CustomerID
Order -> SoldItem -> Product	ProductID
Supplier - Supplies -> Product	SupplierID
Employee - ReportsTo -> Employee	EmployeeID



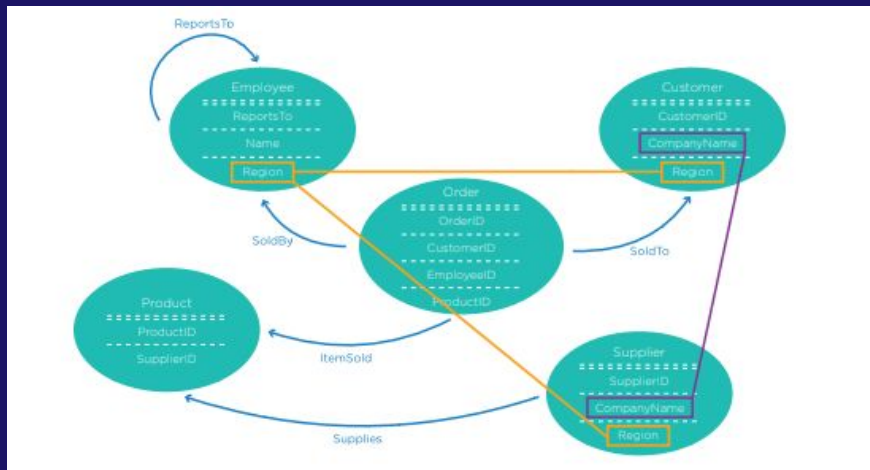
## Стъпка 3:

Когато обектите и връзките са правилно определени, можем да създадем възел за всеки обект и ребро за всяка връзка, която сме намерили. Така ще създадем графова база данни от релационния ни модел, базирана на външните ключове.



# Стъпка 4:

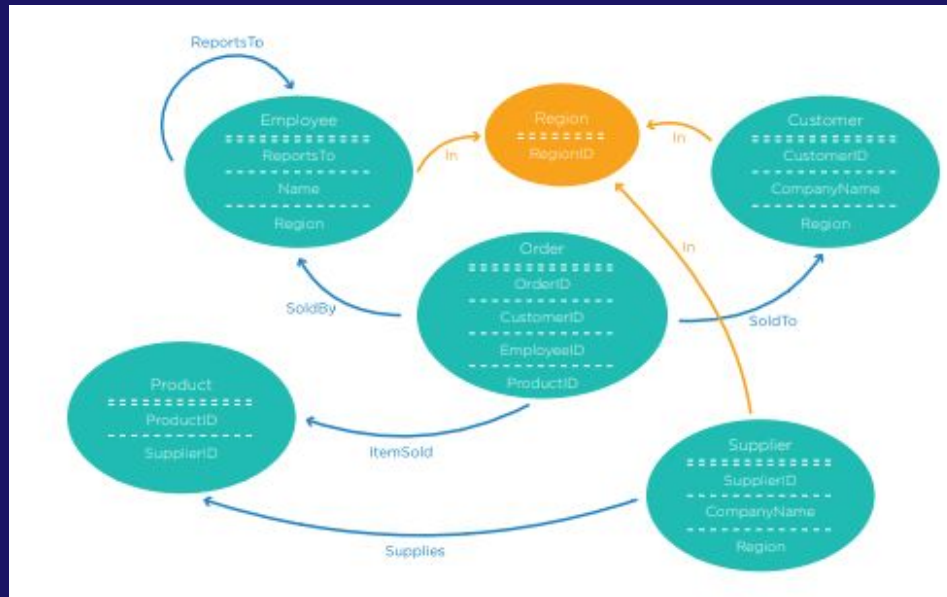
След разглеждане на диаграмата можем да забележим, че имаме два общи атрибута, използвани в повече от 1 възел.



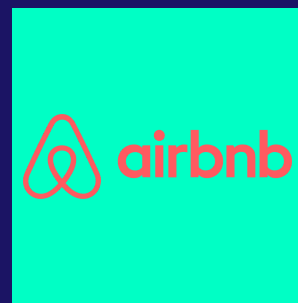
Превръщайки тези общи атрибути в отделни възли с правилни връзки към останалите би добавило стойност към нашия граф.

В този пример след промяната на Region към възел можем да зададем въпроса:

- *Кои са клиентите, които са в същия регион като Никола Василев?*



# Познати фирми използващи Neo4j



И много други..



**Благодарим за вниманието!**