

# Explorando dados abertos com grafos no Autonomous Database

**Daniel Panizzo**

Trilha Inovação com dados em nuvem





# Explorando dados abertos com grafos no Autonomous Database

**Daniel Panizzo**

**Trilha Inovação com dados em nuvem**



Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-Compartilhalgual 4.0 Internacional. Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-sa/4.0/>.





# Daniel Panizzo

  dspanizzo

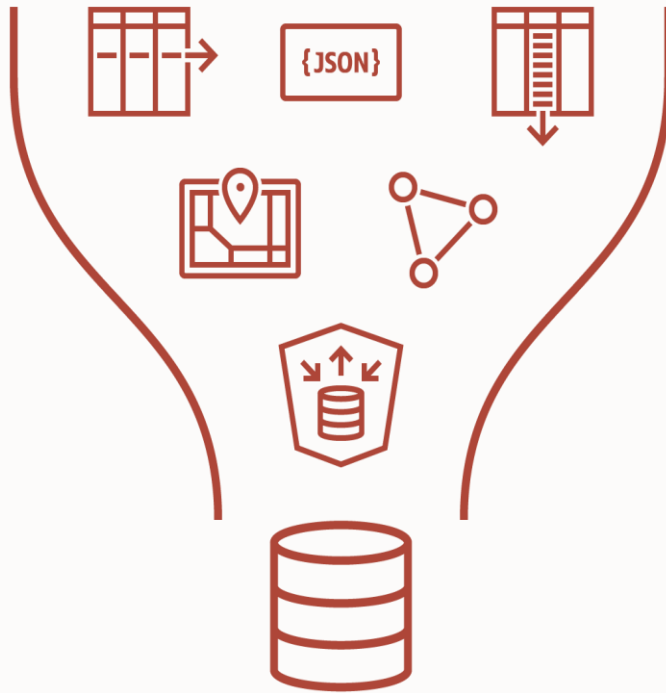


# Do que eu preciso para analisar grafos?

---

# Abordagem *multi-purpose*

Um banco de dados convergente



## Multi-model

Suporte nativo para diversos modelos de dados e métodos de acesso.

## Multitenant

Arquitetura de banco de dados em contêiner provê consolidação, isolamento e agilidade.

## Multi-workload

Uma série de otimizações que trabalham isoladamente ou em conjunto para entregar a melhor performance em qualquer tipo de carga de trabalho.



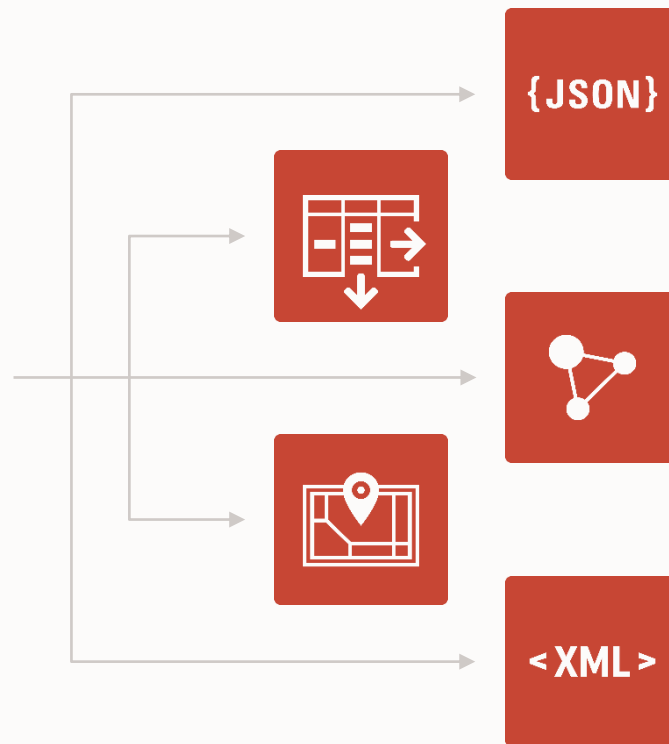
# Multi-model

Um motor, multiplas personalidades

## Cross-Model Data Access

### SQL & REST

Relacional, JSON,  
Espacial, Grafo, XML



## Model-Specific Data Access

### SODA

REST APIs p/ JSON

### PGQL

Property Graph

### SPARQL

RDF Graph

### XQUERY

W3C Standard

# Multi-model

Escopo do *workshop*





Open Knowledge  
Foundation

“Dados abertos são dados que podem ser livremente usados, modificados e compartilhados por qualquer um para qualquer propósito.”

**Open Knowledge Foundation**

Fonte: [The Open Definition](#)



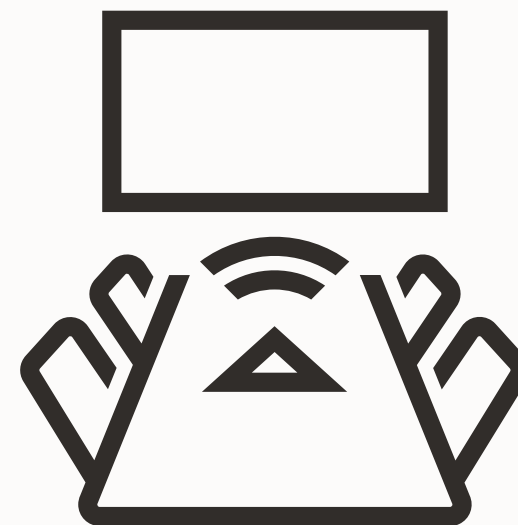
# Por quê dados abertos?



Transparência



Valor social e comercial



Governança Participativa

## Origem dos Dados

Tribunal Superior Eleitoral

Prestação de contas eleitorais por candidato - 2020

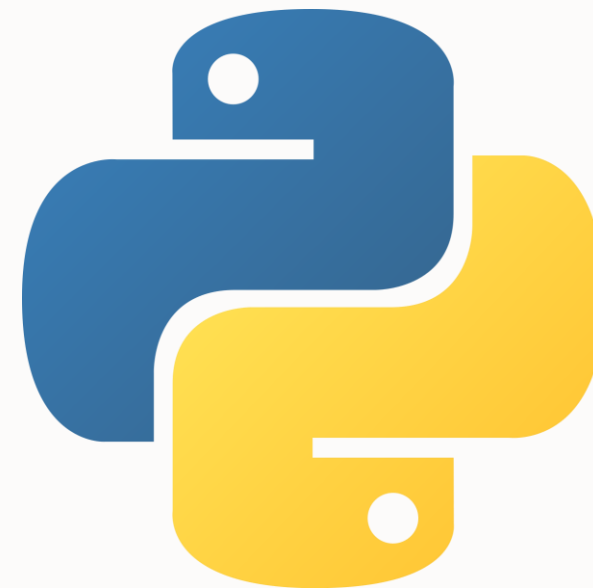


# Ferramentas

Coleta e preparação dos dados



**Jupyter**  
Notebooks



**Python**  
Anaconda

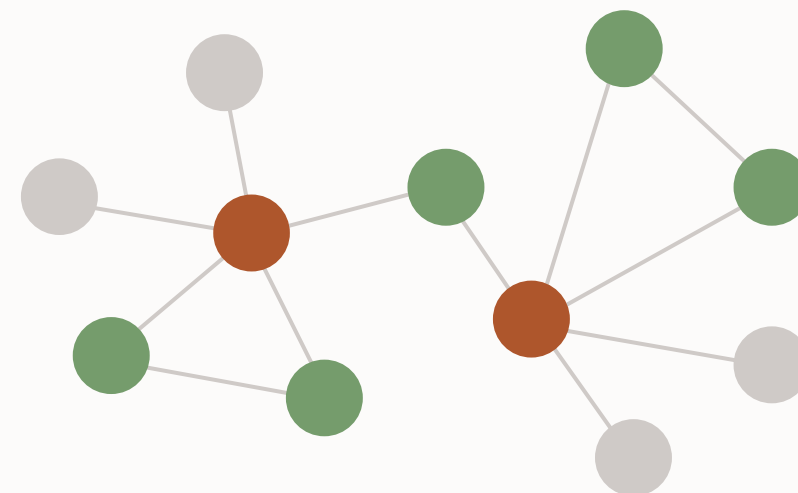


# Ferramentas

Processamento e análise dos dados



**Autonomous**  
Database



**Graph Studio**  
Database Actions

# Um olhar no código

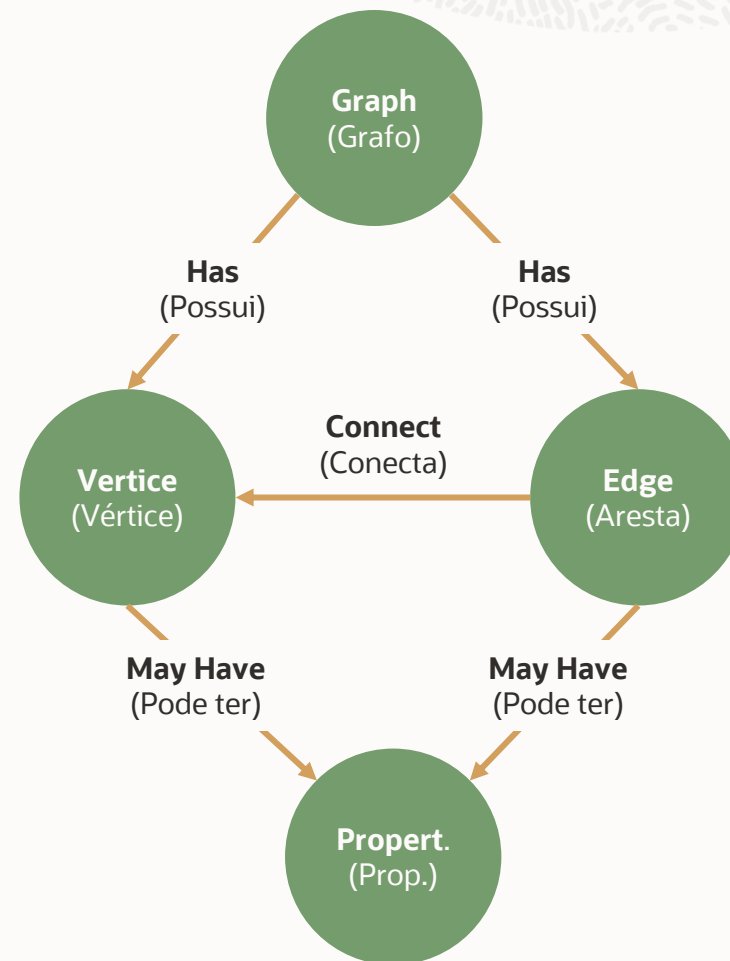
---

# Grafo

## Análise de relacionamentos

Grafo é a representação de um conjunto de objetos em que seus **pares de objetos** estão, de alguma maneira, **relacionados**.

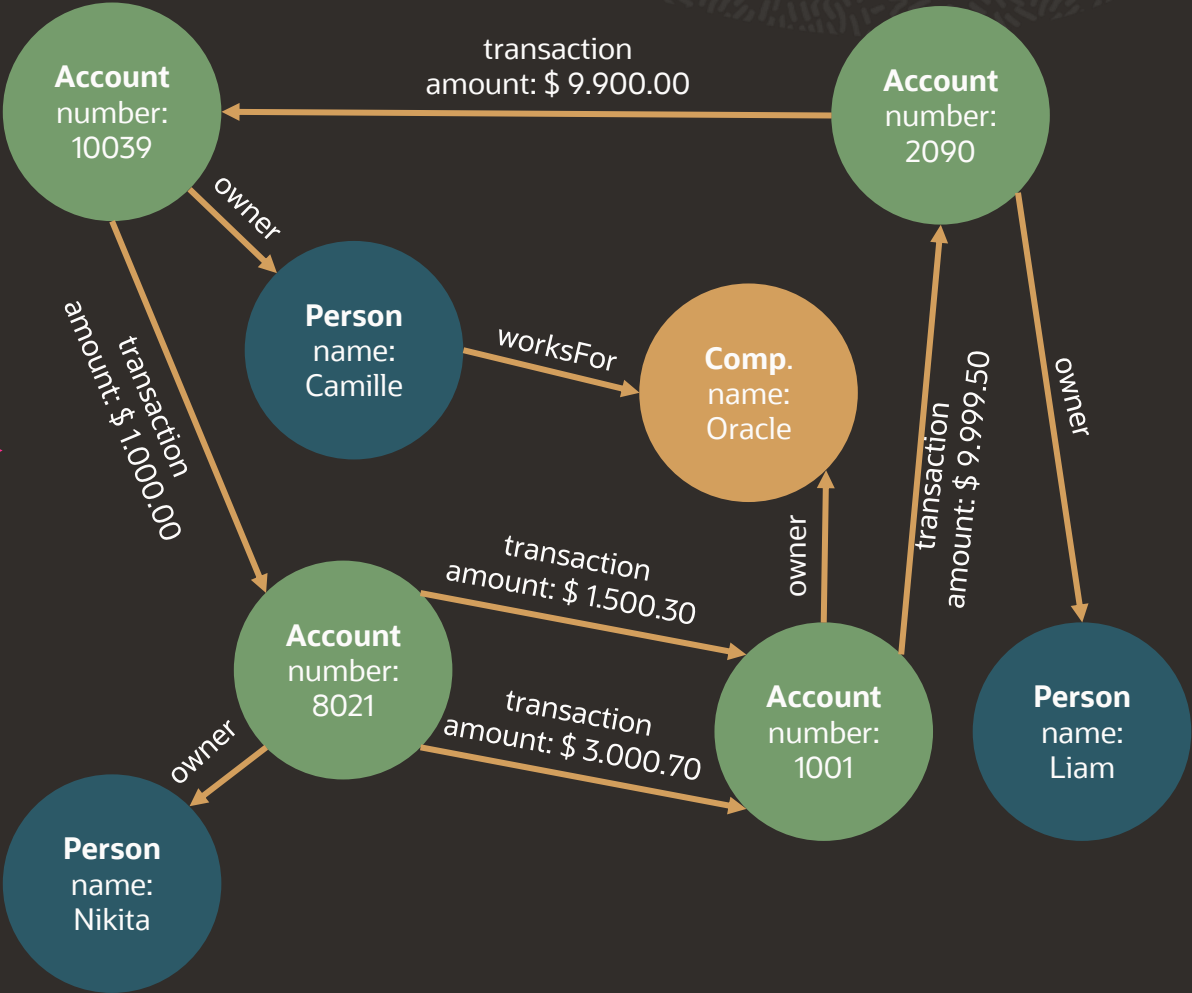
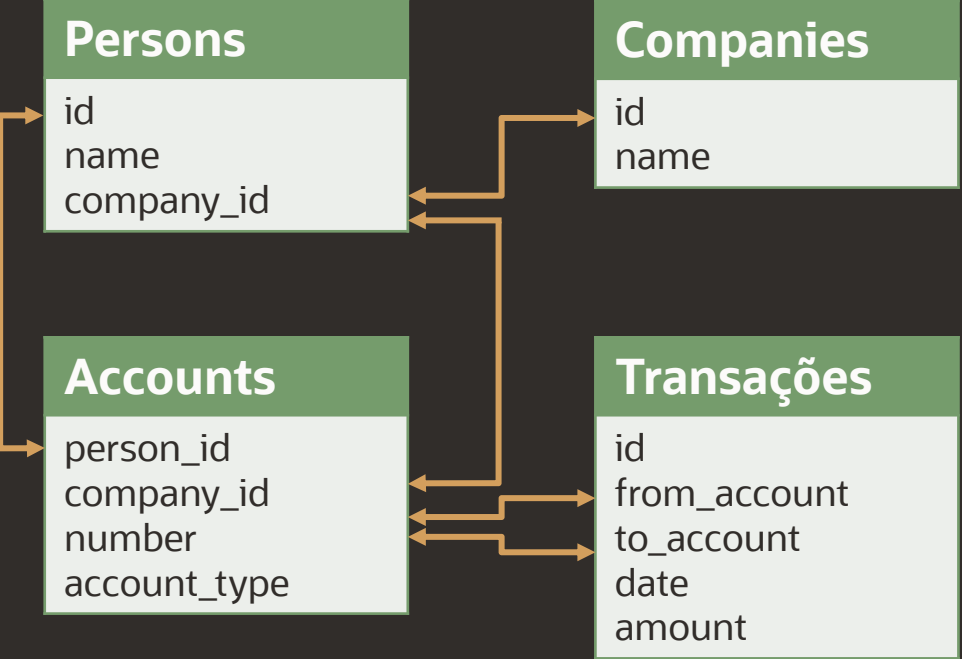
Apesar de possuir uma estrutura simples, pode representar conceitos complexos e seu maior potencial é alcançado utilizando algoritmos analíticos.





# Grafos

## Mapeamento de vértices e arestas



# Grafos

## Mapeamento de vértices e arestas

```
/* DEFINE VERTEX */  
CREATE PROPERTY GRAPH financial_transactions  
  VERTEX TABLES (  
    Persons LABEL Person PROPERTIES ( name ),  
    Companies LABEL Company PROPERTIES ( name ),  
    Accounts LABEL Account PROPERTIES ( number )  
  )  
/* CLICK TO CONTINUE... */
```

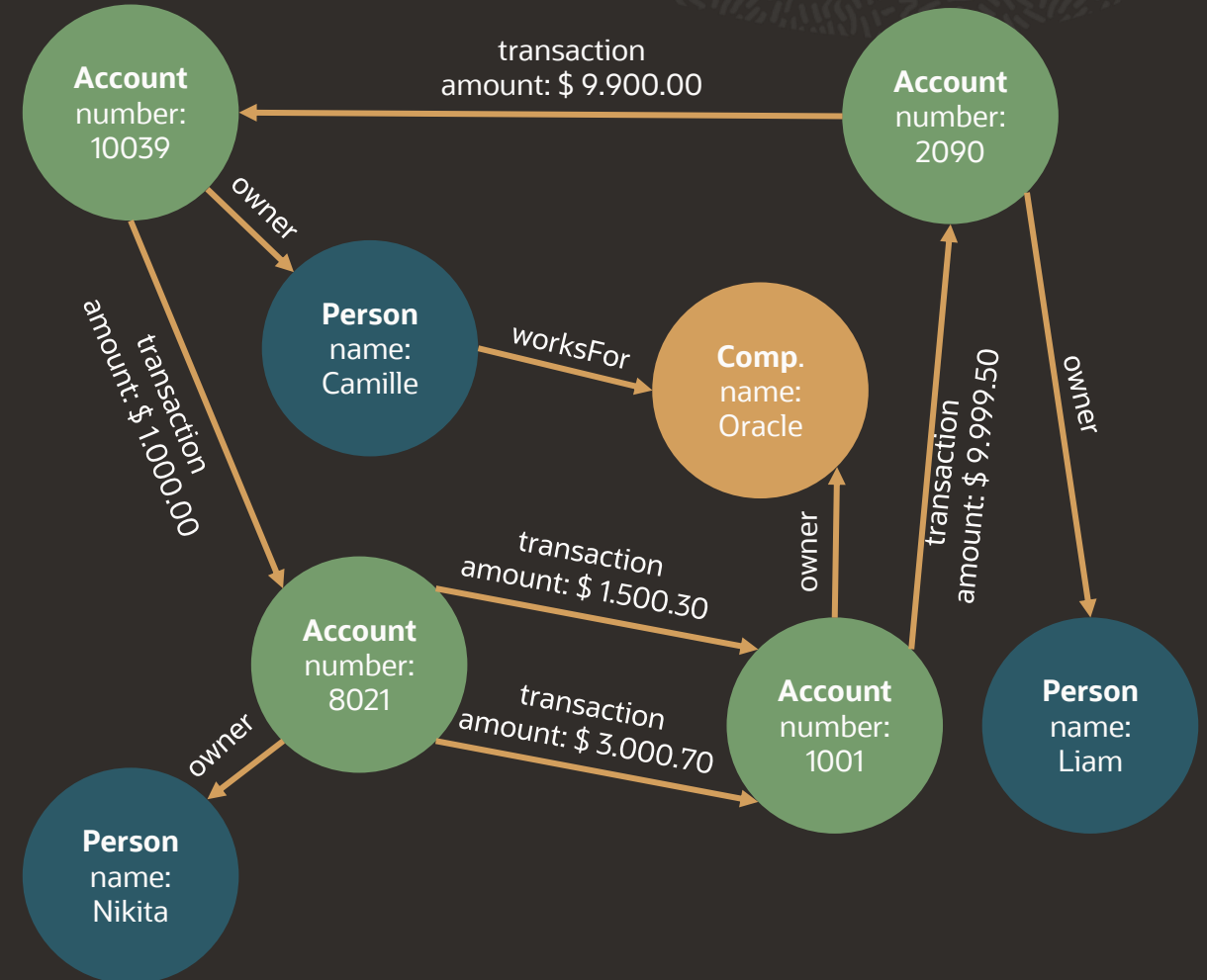


# Grafos

## Mapeamento de vértices e arestas

```
/*      ...      */  
/* DEFINE EDGE */
```

```
EDGE TABLES (  
  Transactions  
    SOURCE KEY (from_account) REFERENCES Accounts  
    DESTINATION KEY (to_account) REFERENCES Accounts  
    LABEL transaction PROPERTIES (amount),  
  Accounts AS PersonOwner  
    SOURCE KEY ( id ) REFERENCES Accounts  
    DESTINATION Persons  
    LABEL owner NO PROPERTIES,  
  Accounts AS CompanyOwner  
    SOURCE KEY ( id ) REFERENCES Accounts  
    DESTINATION Companies  
    LABEL owner NO PROPERTIES,  
  Persons AS worksFor  
    SOURCE KEY ( id ) REFERENCES Persons  
    DESTINATION Companies  
    NO PROPERTIES  
)
```





# Grafos

## Mapeamento de vértices e arestas

```
/* PGQL QUERY */
```

```
SELECT
```

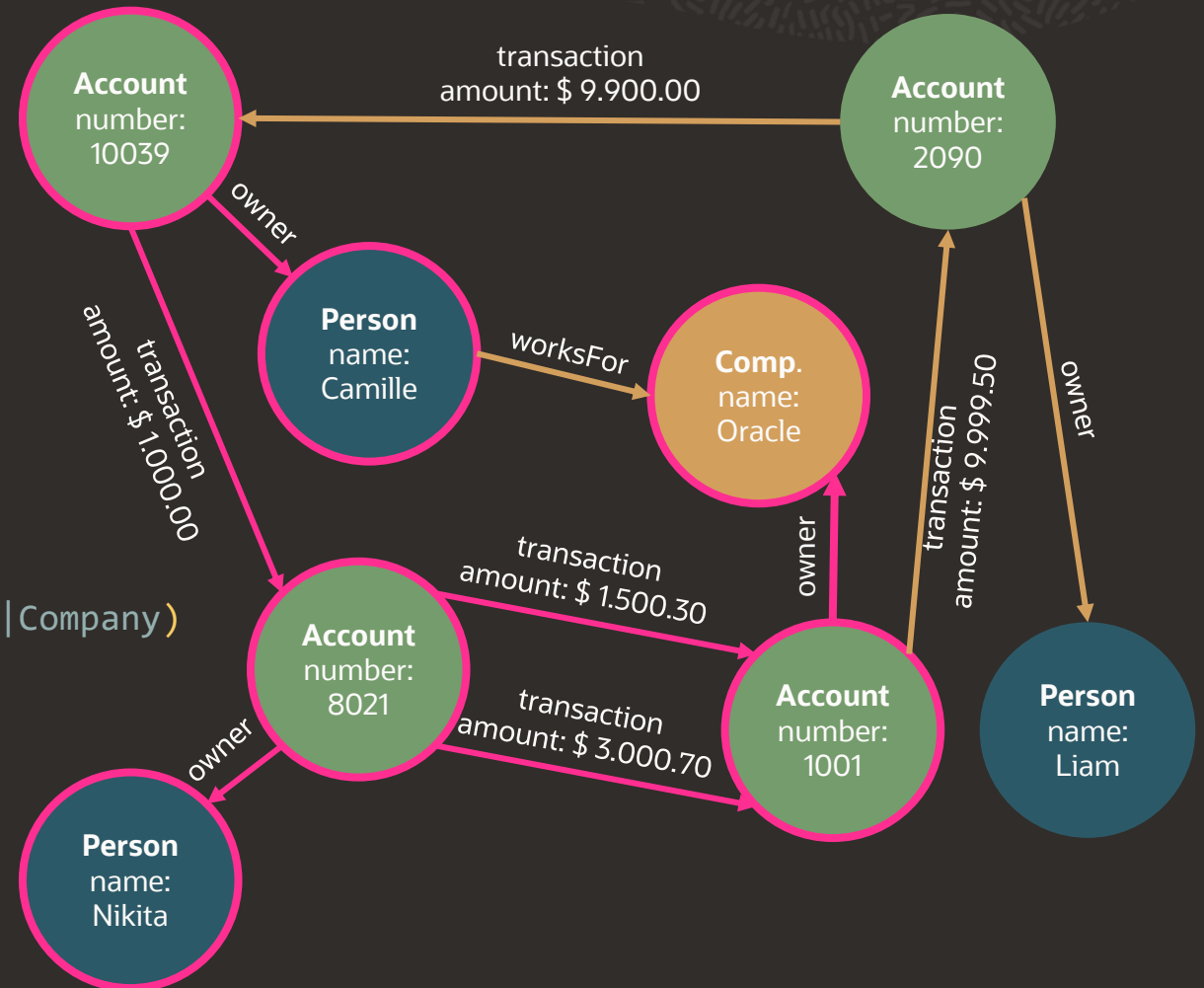
```
  owner.name AS account_holder,  
  SUM(t.amount) AS total_transacted
```

```
FROM
```

```
  MATCH (p:Person) <-[:owner]- (account1:Account),  
  MATCH (account1) -[t:transaction]- (account2),  
  MATCH (account2:Account) -[:owner]-> (owner:Person|Company)
```

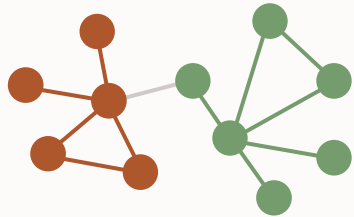
```
WHERE p.name = 'Nikita'
```

```
GROUP BY owner
```



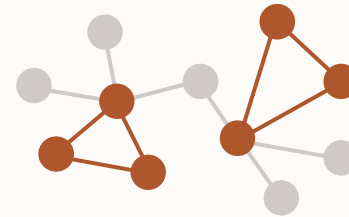
# Grafo

## Análise de relacionamentos



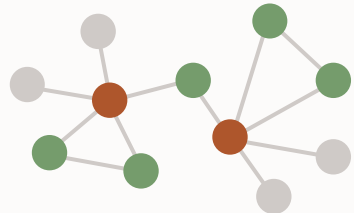
### Detecção de componentes e comunidades

Strongly Connected Components, Weakly Connected Components, Label Propagation, Conductance Minimization, Infomap



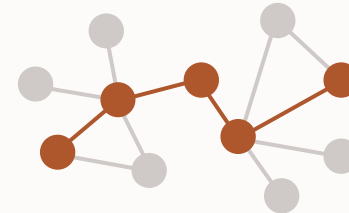
### Avaliação de estruturas

Adamic-Adar Index, Conductance, Cycle Detection, Degree Distribution, Eccentricity, K-Core, LCC, Modularity, Reachability Topological Ordering, Triangle Counting



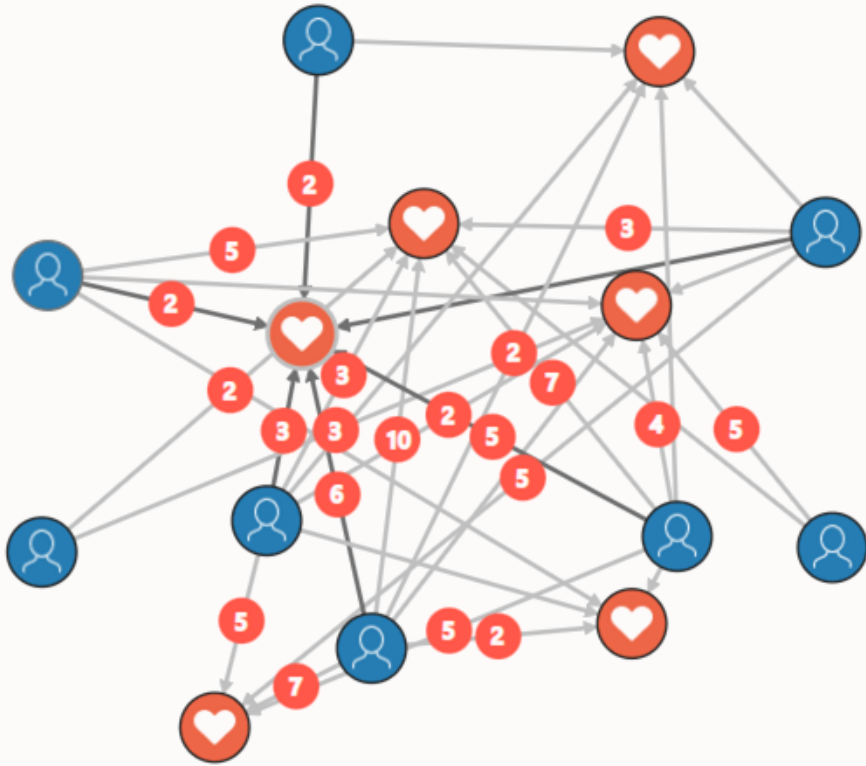
### Ranking and Walking

PageRank, Personalized PageRank, Degree Centrality, Closeness Centrality, Vertex Betweenness Centrality, Eigenvector Centrality, HITS, SALSA, Random Walk with Restart



### Path-finding

Shortest Path (Bellman-Ford, Dijkstra, Bidirectional Dijkstra), Fattest Path, Compute Distance Index, Enumerate Simple Paths, Fast Path Finding, Hop Distance



# Intermediação

*Betweenness Centrality*

“Na teoria dos grafos, a intermediação é a medida de centralidade em um grafo baseado nos menores caminhos. ”

- Wikipedia





Vamos trilhar este  
caminho juntos?



Parabéns!





[github.com/dspanizzo](https://github.com/dspanizzo)



[linkedin.com/in/dspanizzo](https://linkedin.com/in/dspanizzo)

---

Obrigado!

# Vamos juntos nesta trilha!



[linkedin.com/groups/8984009](https://linkedin.com/groups/8984009)



[youtube.com/c/Inovaçãocomdadosenuvem](https://youtube.com/c/Inovaçãocomdadosenuvem)



[anchor.fm/inova-dados-nuvm](https://anchor.fm/inova-dados-nuvm)



[github.com/taborda-cbip/inovacao-com-dados-em-nuvm](https://github.com/taborda-cbip/inovacao-com-dados-em-nuvm)



ORACLE