INSTITUTO DE COMPUTAÇÃO - UNICAMP

MC536 - Bancos de Dados: Teoria e Prática

Prof. André Santanchè

Trabalho Final

Dupla LPT:

Gabriel Henrique Rosa Oswaldo - 172185
Lucas Silva Lopes Do Carmo - 202110

Dupla JSS:

Gabriel De Alcantara Bomfim Silveira - 197244
Vitor Coppo Ferreira - 206956

IC/UNICAMP
2019

GitHub

Todo o material apresentado aqui está presente no seguinte repositório no GitHub, seguindo a estrutura de organização exigida:

- https://github.com/gabrieloswaldo/mc536-trabalho

1. Análise relacional com SQL

Nesta seção são apresentadas as análises em SQL feitas por cada dupla, assim como os modelos relacional e lógico utilizados por cada uma.

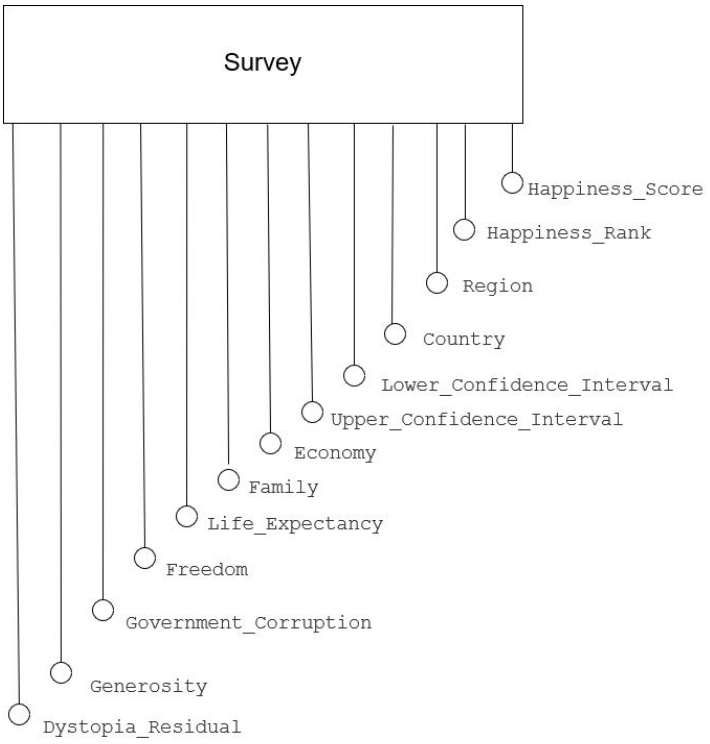
a) Dupla LPT

Nesta etapa do trabalho utilizamos como fonte de dados o World Happiness Report 2016, o qual contém pontuações e classificações de felicidade nos países, com um total de 157 instâncias e 13 variáveis. As pontuações são baseadas nas principais perguntas sobre avaliações sobre a vida das pessoas, feitas na pesquisa.

As colunas deste conjunto de dados incluem país, região, rank de felicidade, pontuação de felicidade, intervalo de confiança inferior e superior, Economia (PIB), Família, Saúde (Expectativa de Vida), Liberdade, Confiança no Governo (Corrupção), Generosidade, Distopia residual. O índice de felicidade é determinado por seis fatores - de Economia a Generosidade - que tornam as avaliações de vida mais altas em cada país.

Com este conjunto de dados, conseguimos obter o seguinte modelo lógico e relacional, respectivamente, apresentado abaixo:

```
Survey(Country, Region, Happiness_Rank,
      Happiness_Score, Lower_Confidence_Interval,
      Upper_Confidence_Interval, Economy, Family,
      Life_Expectancy, Freedom,
Government_Corruption,
      Generosity, Dystopia_Residual)
```



Partindo para as análises em SQL, focamos mais em análises exploratórias, tentando extrair algumas informações da base de dados e correlacionar os atributos da mesma. Primeiro analisamos qual região possuía, em média, um índice felicidade mais elevado, e quantos países faziam parte desta região. Com isso obtivemos o resultado apresentado abaixo, que nos mostra que as regiões da Oceania, América do Norte e Oeste Europeu possuem em média um índice mais elevado, enquanto a África Subsariana apresentou a pior média.

index	REGION	AVG_HAP_SCORE	COUNTRIES
0	Australia and New Zealand	7.323499999999999	2
1	North America	7.254	2
2	Western Europe	6.685666666666665	21
3	Latin America and Caribbean	6.10175	24
4	Eastern Asia	5.624166666666667	6
5	Middle East and Northern Africa	5.386052631578948	19
6	Central and Eastern Europe	5.3706896551724155	29
7	Southeastern Asia	5.338888888888889	9
8	Southern Asia	4.563285714285714	7
9	Sub-Saharan Africa	4.136421052631578	38

Figura 1 - Média do índice de felicidade por região

Como segunda análise, buscamos identificar o quanto os seis fatores influenciam no índice de felicidade. Para isso, estabelecemos correlações entre cada um dos seis fatores com o índice de felicidade, através do coeficiente de Pearson. Este coeficiente nos possibilitou avaliar o grau das relações entre a pontuação de felicidade e os seis fatores, determinando quais destes tem maior influência para ter uma pontuação mais elevada. Com isso, obtivemos o resultado apresentado abaixo, que nos

mostra que Economia (PIB), Família, Saúde (Expectativa de Vida) são mais importantes para obter um índice de felicidade mais elevado.

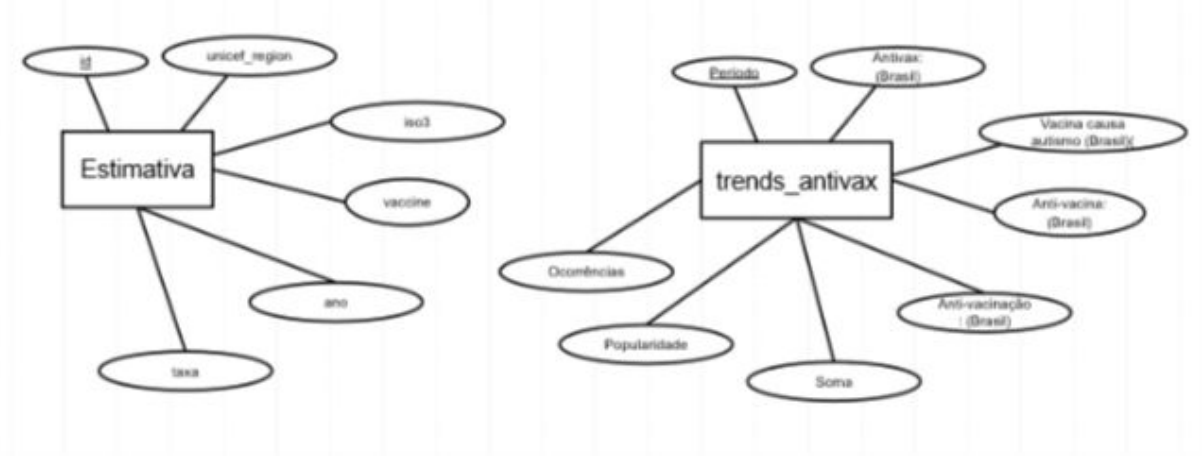
index	RELATION	COEFICIENT
0	economy	0.7903220167261241
1	life_expectancy	0.7653843344336755
2	family	0.7392515774070099
3	freedom	0.5668266730968905
4	gov_corruption	0.4020322451472926
5	generosity	0.15684779640360982

Figura 2 - Coeficientes de Pearson para os seis fatores

b) Dupla JSS

Para a primeira etapa do trabalho, decidimos trabalhar com o problema de imunização. Para tal, utilizamos a base de dados de imunização da Organização Mundial de Saúde (https://data.unicef.org/topic/child-health/immunization/), e restringimos nosso universo de análise ao Brasil. Esta base possui um total de 127 estimativas distribuídas ao longo de 13 vacinas diferentes nos últimos 10 anos.

```
    Immunization_Estimate(id, unicef_region, iso3,
vaccine, Ano, Taxa);
    Trends_antivax(Período, Antivax:(Brasil), Vacina
Causa autismo(Brasil), Anti-vacina:(Brasil),
Anti-vacinação:(Brasil), Soma, Popularidade,
Ocorrências);
```



Como uma análise inicial, utilizamos querys em SQL, para buscar entender um pouco melhor o comportamento temporal da imunização no Brasil. Assim, buscando a taxa média ao longo dos anos e, depois vendo quais dos anos analisados possuíam taxa de vacinação acima da média.

index	VACCINE	MEDIA_TAXA
0	DTP1	97
1	HIB3	94
2	DTP3	93
3	HEPB3	94
4	IPV1	91
5	HEPBB	87
6	BCG	96
7	MCV2	70
8	POL3	91
9	ROTAC	85
10	MCV1	95
11	RCV1	95
12	YFV	43
13	PCV3	80

Figura 3 - Média de Taxa de vacinação por vacina

Acima

index	ANO	COUNT(*)
1	2018	2
0	2017	5
9	2016	6
8	2015	14
7	2014	13
6	2013	12
5	2012	12
4	2011	12
3	2010	9
2	2009	10

Abaixo

index	ANO	COUNT(*)
1	2018	12
0	2017	9
7	2016	8
6	2013	1
5	2012	1
4	2011	1
3	2010	4
2	2009	2

Figuras 4 e 5 - Quantidade de vacinas acima e abaixo da média anual, respectivamente

Para melhor aproveitar estes dados, construímos uma base menor, a partir dos dados de pesquisa do Google sobre o movimento anti-vacina, obtidos através do Google Trends, com o intuito de verificar se poderia existir uma relação entre mudanças nas taxas de vacinação e o movimento anti-vacina.

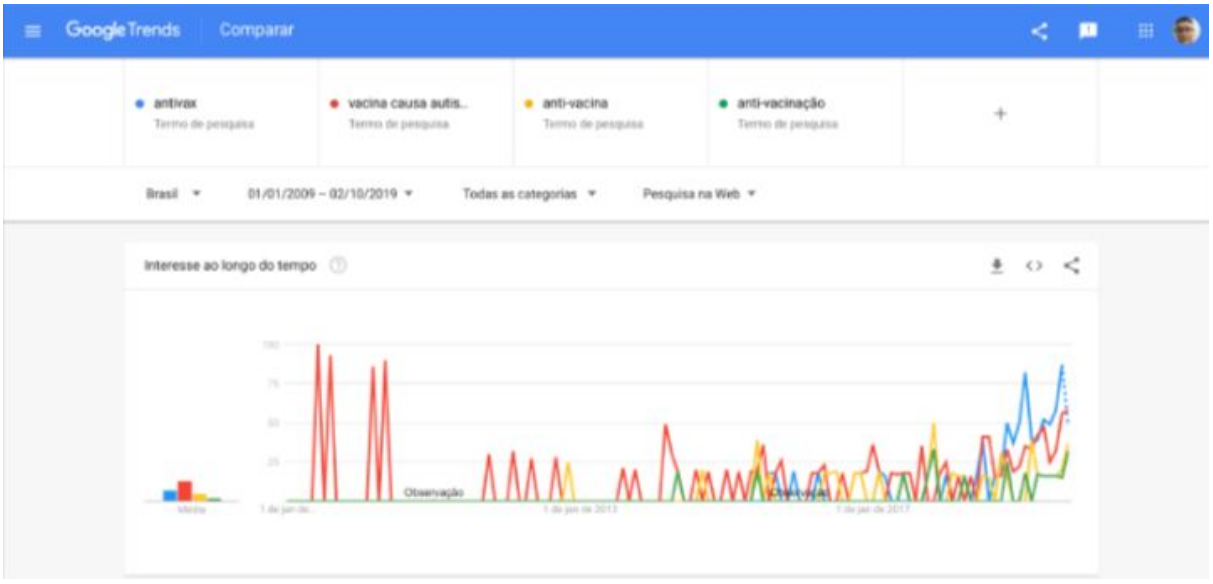


Figura 6 - Gráfico de tendências sobre anti-vacinação

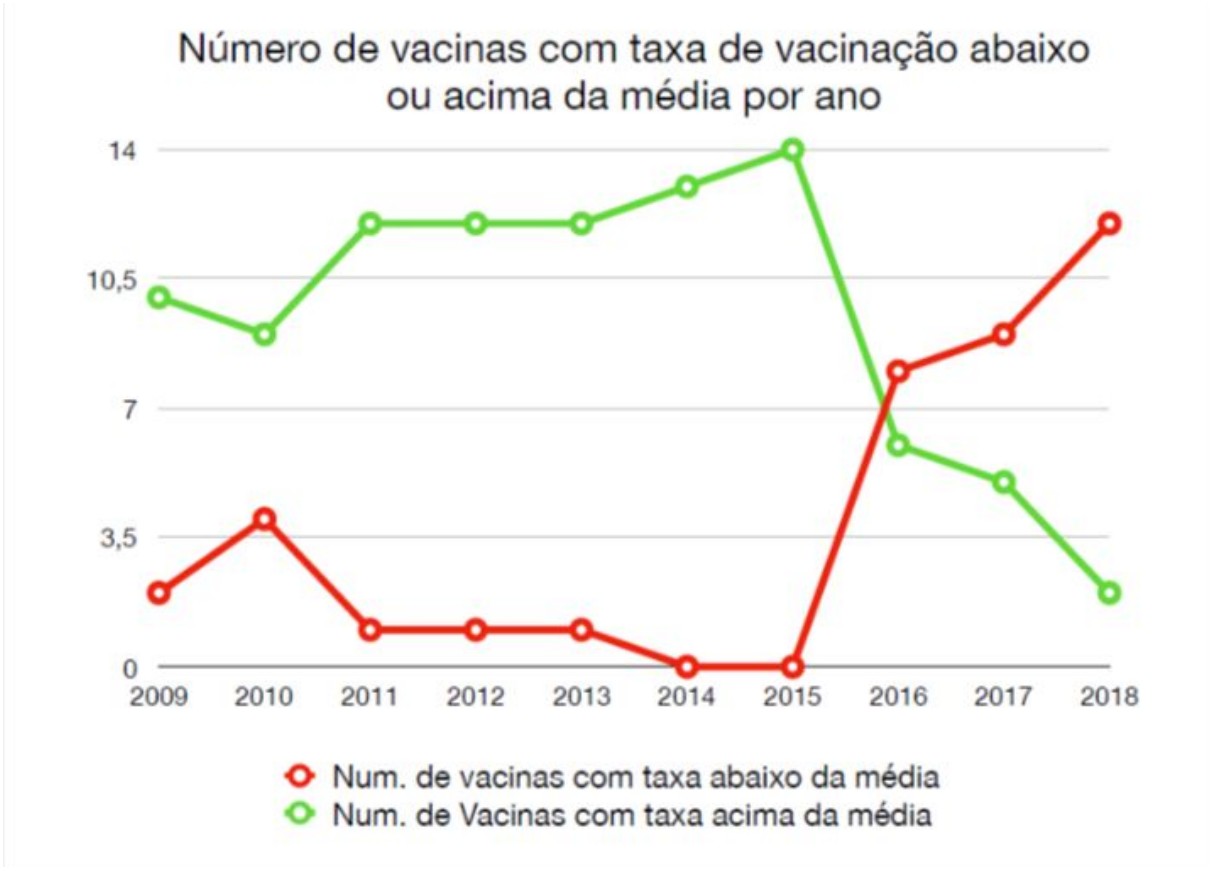


Figura 7 - Gráfico relacionando o número de vacinas acima e abaixo da média

Através de uma análise visual dos dois gráficos, podemos ver que é sim possível a existência de uma relação entre o declínio do número de vacinas abaixo da taxa média e o aumento de pesquisas sobre anti-vacinação, no entanto, para se poder confirmar tal relação, seria necessário um estudo de cunho diferente, não coberto no escopo da disciplina.

2. Análise hierárquica com XQuery

Para a realização do trabalho utilizamos os data sets gratuitos disponíveis em: Orphadata, especificamente a fonte relacionada a doenças raras relacionadas com seus fenótipos (http://www.orphadata.org/data/xml/en_product4_HPO.xml).

Para a elaboração das queries em XQuery nas bases XML utilizamos o programa gratuito BaseX, visto que o programa mostrado pelo professor (http://try.zorba.io/) não suportava o tamanho das fontes de dados.

Abaixo temos todas as queries realizadas, com suas respectivas descrições:

- **XQuery 01:** retorna a quantidade de doenças raras cadastradas na base.

```
for $c in //DisorderList
return count($c/Disorder)
```

- **XQuery 02:** retorna a quantidade de distúrbios associados a doença rara com o id especificado, no caso utilizamos a "Alexander Disease", que possui o id=2.

```
for $c in //Disorder[@id=2]//HPODisorderAssociationList
return count($c/HPODisorderAssociation)
```

- **XQuery 03:** retorna a lista formatada de distúrbios com a frequência acima de "Frequente", associadas a doença rara com o id especificado. No caso utilizamos a "Alexander Disease" com frequência "Frequent (79-30%)", "Very frequent (99-80%)" e "Obligate (100%)".

```
for $c in //Disorder[@id=2]//HPODisorderAssociation
where $c/HPOFrequency[Name="Frequent (79-30%)"] or
      $c/HPOFrequency[Name="Very frequent (99-80%)"] or
      $c/HPOFrequency[Name="Obligate (100%)"]
return <sintoma>
      <nome>{data($c//HPOTerm)}</nome>
      <frequencia>{data($c/HPOFrequency/Name)}</frequencia>
      </sintoma>
```

- **XQuery 04:** retorna uma lista de doenças que possuem o sintoma especificado (HPOTerm="Macrocephaly") com frequência acima de "Frequent (79-30%)".

```
for $c in //Disorder
where $c//HPODisorderAssociation/HPO[HPOTerm="Macrocephaly"]
and
      ($c//HPODisorderAssociation//HPOFrequency[Name="Very
      frequent (99-80%)"] or
      $c//HPODisorderAssociation//HPOFrequency[Name="Frequent
(79-30%)"] or
```

```
      $c//HPODisorderAssociation//HPOFrequency[Name="Obligate
      (100%)"])
return  data($c/Name)
```

- **XQuery 05:** retorna uma lista de doenças que possuem ambos os sintomas especificados com uma frequência acima de ocasional.

```
for $c in (//Disorder),
     $d in (//Disorder)
where  $c//HPODisorderAssociation/HPO[HPOTerm="Macrocephaly"]
and
      $d//HPODisorderAssociation/HPO[HPOTerm="Nystagmus"] and
      ($c[@id] = $d[@id])
      and
      (($c/HPODisorderAssociation/HPOFrequency[Name="Frequent
(79-30%)"] or
      $c//HPOFrequency[Name="Very frequent (99-80%)"] or
      $c//HPOFrequency[Name="Obligate (100%)"]) and
      ($d//HPOFrequency[Name="Frequent (79-30%)"] or
      $d//HPOFrequency[Name="Very frequent (99-80%)"] or
      $d//HPOFrequency[Name="Obligate (100%)"]))
return data($c/Name)
```

Vantagens de se utilizar o modelo Hierárquico

Neste caso, o modelo hierárquico com análises em XQuery se mostrou muito mais apropriado que outros modelos, como o relacional, visto que a estrutura hierárquica possibilita um modelo de busca em árvore, o que resulta numa maior consistência dos dados quando requisitados. O modelo hierárquico, tendo como base o xml, também possibilita guardar ou vincular dados em qualquer formato, graças à liberdade dada ao usuário de definir e estruturar suas marcações.

O modelo hierárquico também possui uma ótima escalabilidade, pois a inserção de dados se torna mais fácil. No entanto, a atualização dos dados, pode ser considerada uma desvantagem devido à quantidade de dados redundantes que esse modelo gera, assim se uma atualização não for bem arquitetada, podem gerar dados inconsistentes.

3. Análise de rede com Neo4J/Cypher

Para a análise de rede utilizamos os dados abertos sobre casos de suicídio disponíveis em https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016. Preparamos os dados em diferentes arquivos CSV e importamos para o sandbox online do Neo4J.

Abaixo seguem algumas das Queries realizadas para análise, importação e configuração:

Descrição: importa os países na base
Query1:

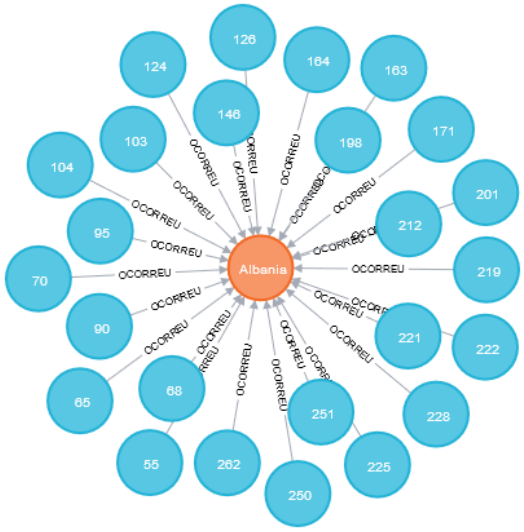
```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/gabrieloswaldo/mc536-trabal
ho/master/jupyter/data/suicidio-paises.csv' AS line
CREATE (:Pais {id: line.Id_pais, name: line.country})
```

Descrição: importa os casos de suicídio
Query2:

```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/gabrieloswaldo/mc536-trabal
ho/master/jupyter/data/suicidios-casos.csv' AS line CREATE
(:Suicidio {id: line.Id, sex: line.sex, age: line.age,
generation: line.generation})
```

Descrição: cria a relação entre os casos de suicídio e os países em que aconteceram
Query3:

```
CREATE INDEX ON :Suicidio(id)
CREATE INDEX ON :Pais(name)
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/gabrieloswaldo/mc536-trabal
ho/master/jupyter/data/suicidios-relations.csv' AS csvLine
MATCH (p:Pais {name: csvLine.country})
MATCH (c:Suicidio {id: csvLine.Id})
CREATE (c)-[:OCORREU {ano: csvLine.year}]->(p)
```



Descrição: cria um nó para cada geração

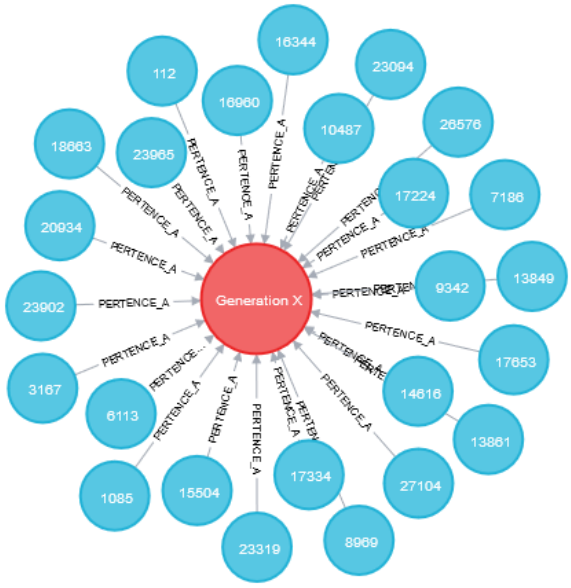
Query4:

```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/gabrieloswaldo/mc536-trabal
ho/master/jupyter/data/suicidios-casos.csv' AS line
MERGE (g:Generation {generation: line.generation})
```

Descrição: cria a relação entre os casos de suicídio e a qual geração a pessoa pertenceu

Query5:

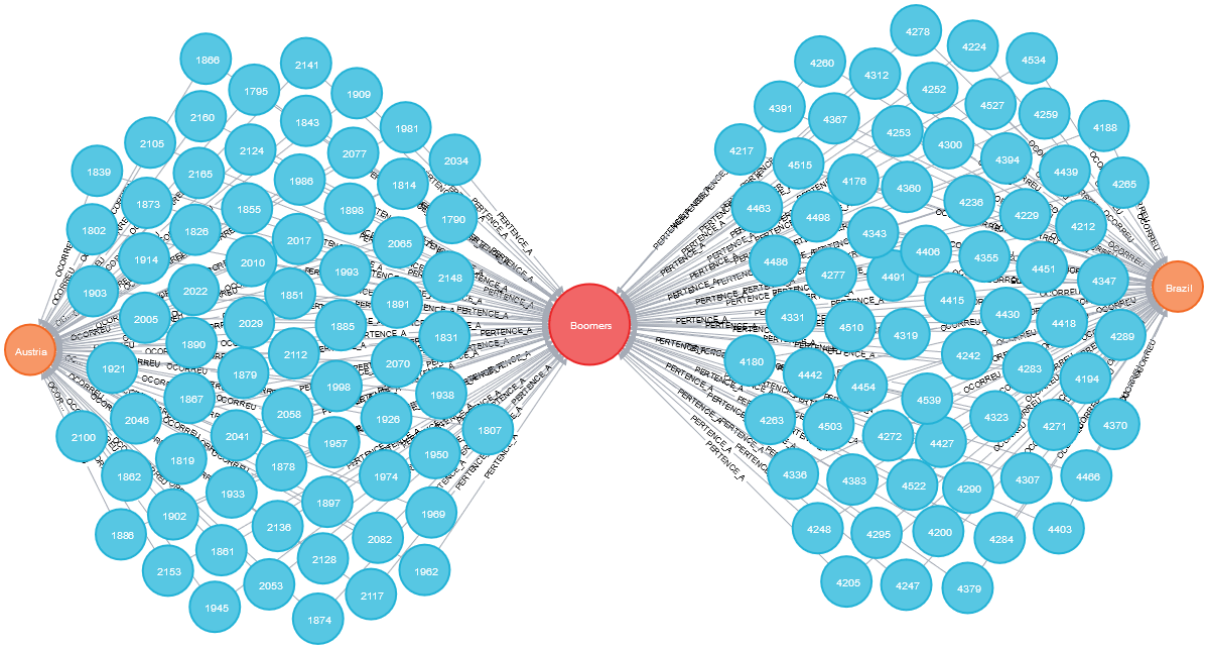
```
MATCH (g:Generation)
MATCH (s:Suicidio)
WHERE s.generation = g.generation
CREATE (s)-[:PERTENCE_A]->(g)
```



Descrição: retorna a rede de relações entre os suicídios de pessoas e qual geração ela pertenceu, especificada para dois países.

Query6:

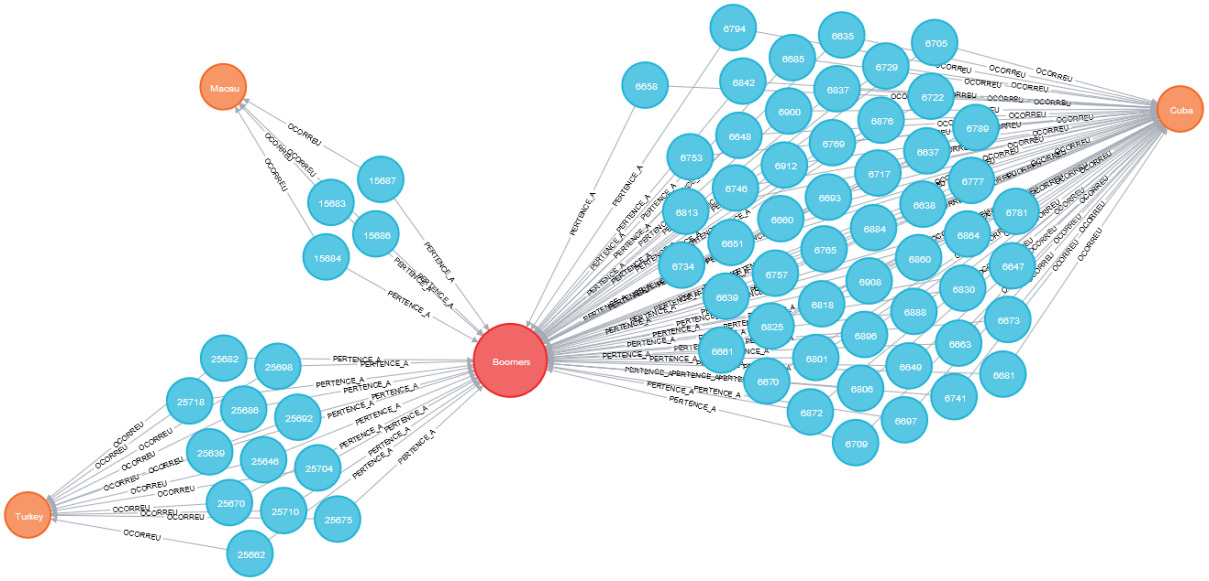
```
MATCH (g:Generation {generation:"Boomers"})
MATCH (s:Suicidio)-[:OCORREU]->(p)
WHERE p.name = "Brazil" OR p.name = "Austria"
RETURN (s)-[:PERTENCE_A]->(g), p
```



Descrição: retorna a rede de relações entre os suicídios de pessoas e qual geração ela pertenceu, especificada para três países.

Query7:

```
MATCH (g:Generation {generation:"Boomers"})
MATCH (s:Suicidio)-[:OCORREU]->(p)
WHERE p.name = "Cuba" OR p.name = "Macau" OR p.name = "Turkey"
RETURN (s)-[:PERTENCE_A]->(g), p
```


Macau
Cuba
Turkey
Boomers
15687
15683
15686
15684
6794
6635
6685
6705
6729
6658
6842
6837
6722
6900
6876
6789
6648
6769
6753
6912
6837
6717
6777
6746
6693
6813
6638
6781
6660
6884
6651
6884
6765
6860
6734
6757
6647
6639
6908
6830
6818
6888
6825
6673
6696
6663
6661
6801
6649
6681
6670
6741
6806
6872
6697
6709
25682
25698
25718
25686
25692
25639
25646
25704
25670
25710
25675
25662

Descrição: cria os nós representantes do sexo masculino (Male) e feminino(Female)

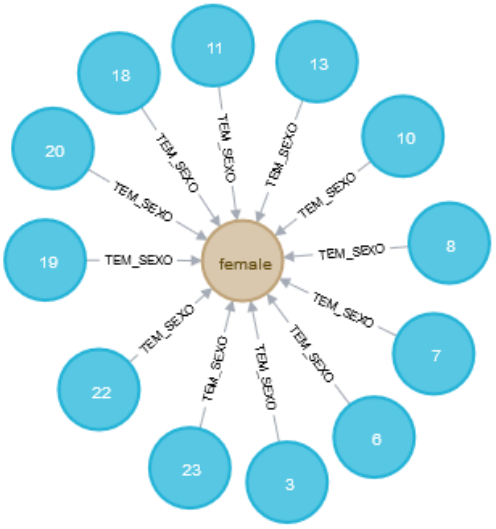
Query8:

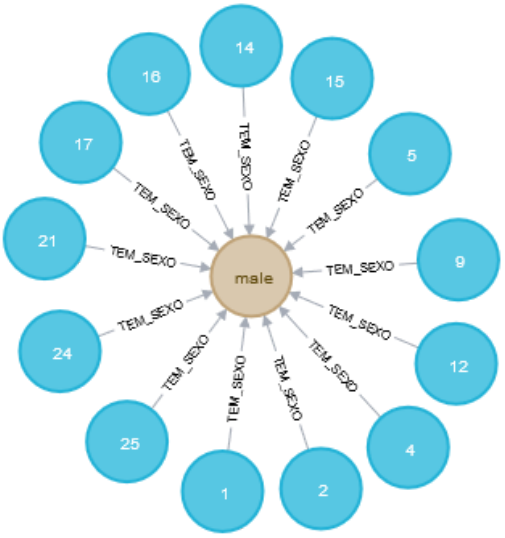
```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/gabrieloswaldo/mc536-trabal
ho/master/jupyter/data/suicidios-casos.csv' AS line
MERGE (s:Sex {sex: line.sex})
```

Descrição: Cria relações entre suicidas Homens com o nó Male e suicidas mulheres com o nó Female.

Query9:

```
MATCH (sex:Sex)
MATCH (sui:Suicidio)
WHERE sui.sex = sex.sex
CREATE (sui)-[:TEM_SEXO]->(sex)
```

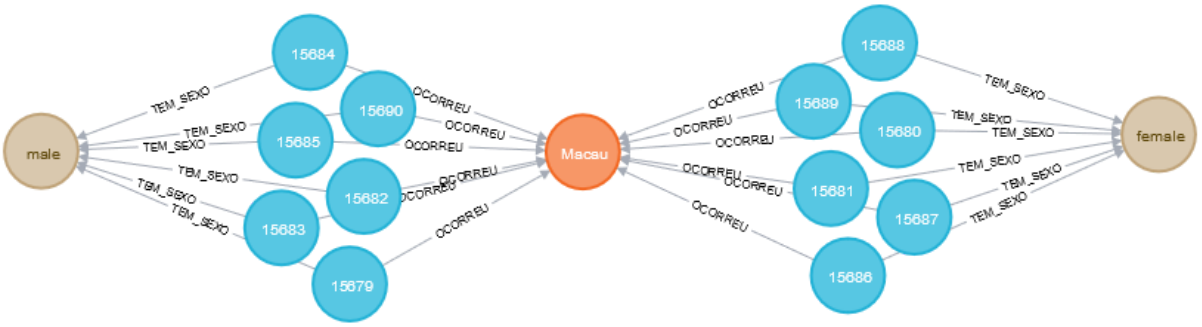




Descrição: retorna a relação entre suicídios de homens e mulheres em um país.

Query10:

```
MATCH (s:Suicidio)-[:OCORREU]->(p:Pais)
MATCH (s:Suicidio)-[:TEM_SEXO]->(sex:Sex)
WHERE p.name = "Macau"
RETURN s, sex, p
```



Descrição: roda o pagerank no grafo e retorna o score para os países com mais suicídios

Query11:

```
CALL algo.pageRank.stream('Page', 'LINKS', {iterations:20,
dampingFactor:0.85})
YIELD nodeId, score
RETURN algo.asNode(nodeId).name AS name,score
ORDER BY score DESC
```

name	score
"Austria"	48.85517120361328
"Iceland"	48.85517120361328
"Mauritius"	48.85517120361328
"Netherlands"	48.85517120361328
"Argentina"	47.58015823364258
"Belgium"	47.58015823364258
"Brazil"	47.58015823364258
"Chile"	47.58015823364258
"Colombia"	47.58015823364258
"Ecuador"	47.58015823364258
"Greece"	47.58015823364258
"Israel"	47.58015823364258
"Italy"	47.58015823364258
"Japan"	47.58015823364258
"Luxembourg"	47.58015823364258

Vantagens de se utilizar o modelo de Redes

A principal vantagem apresentada pelo modelo de redes é a fácil visualização dos resultados das queries a partir dos grafos gerados, os quais facilitam o entendimento e interpretação por parte de seus usuários. O fato dos dados estarem distribuídos em rede permite a utilização de algoritmos conhecidos sobre eles, como por exemplo, o PageRank, desenvolvido pelo Google com o intuito de rankear páginas de acordo com sua popularidade, algoritmo este que pode ser utilizado para visualizar quais vértices do grafo possuem mais conexões significativas.

A opção de se poder montar redes a partir de formatos tipicamente utilizados para armazenar dados, como o csv e tsv, o torna de fácil acesso.