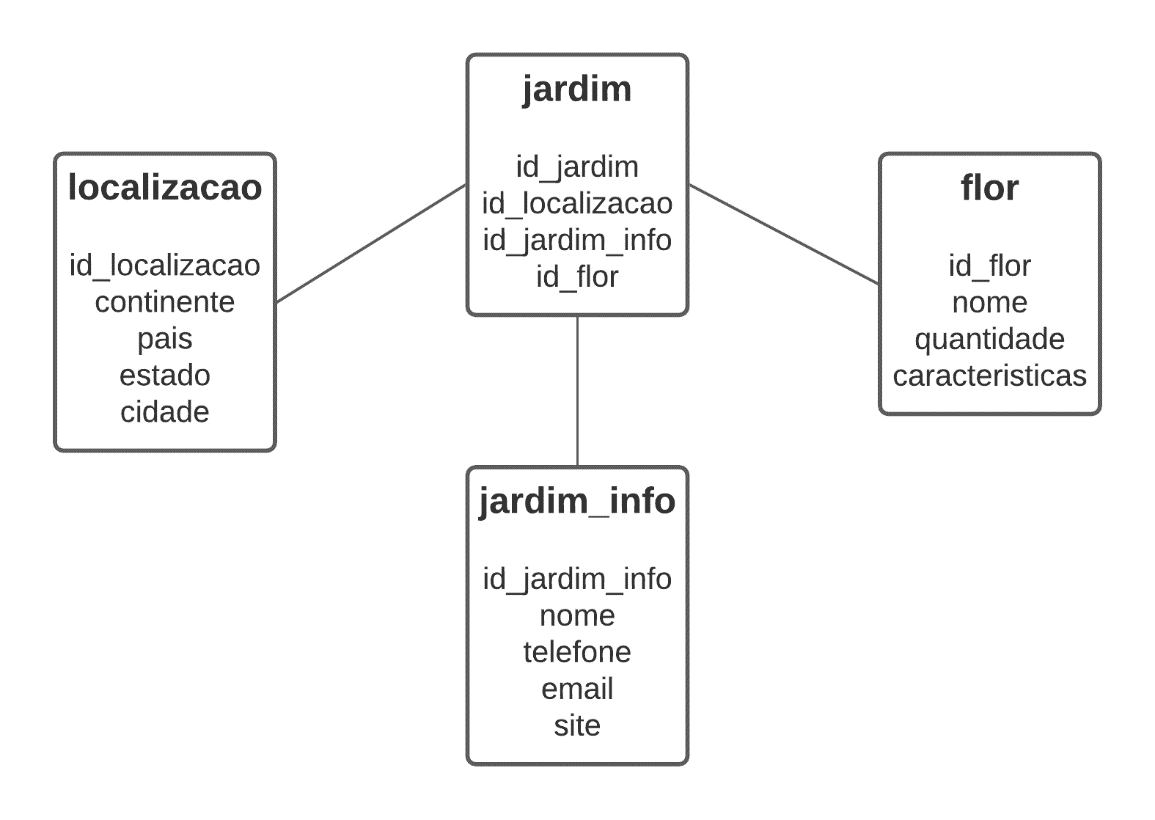
**Universidade Tecnológica Federal do Paraná - Campus Dois Vizinhos   
Especialização em Ciência de Dados**   
  
  
  
  
  
  
  
  
  
  
**Projeto Integrador**  
  
  
  
**Alexandre Hannisch   
Bruno Faustino Amorim   
Leonardo Garcia   
Nicolas Soffi**

**Dois Vizinhos - PR   
2021**

**Processamento Analítico de Dados**

**Projeto logico (desenho estrutural):**



**Projeto físico (Criação das tabelas)**

**create** **schema** jardim;

**create** **table** jardim.jardim\_info

(

id\_jardim\_info serial **primary** **key**,

nome **VARCHAR**,

telefone **VARCHAR**,

email **VARCHAR**,

site **VARCHAR**

);

**create** **table** jardim.localizacao

(

id\_localizacao serial **primary** **key**,

continente **varchar**,

pais **VARCHAR**,

estado **varchar**,

cidade **VARCHAR**

);

**create** **table** jardim.flor

(

id\_flor serial **primary** **key**,

nome **varchar**,

quantidade **INTEGER**,

caracteristicas **FLOAT**[]

);

**create** **table** jardim.jardim

(

id\_jardim serial **primary** **key**,

id\_jardim\_info **INTEGER**,

id\_localizacao **INTEGER**,

id\_flor **INTEGER**,

**CONSTRAINT** fk\_jardim\_info

**FOREIGN** **KEY**(id\_jardim)

**REFERENCES** jardim.jardim\_info(id\_jardim\_info),

**CONSTRAINT** fk\_localizacao

**FOREIGN** **KEY**(id\_localizacao)

**REFERENCES** jardim.localizacao(id\_localizacao),

**CONSTRAINT** fk\_flor

**FOREIGN** **KEY**(id\_flor)

**REFERENCES** jardim.flor(id\_flor)

);

**Descrever no documento quais são as hierarquias identificada**

Localização: (all) ⪯ (continente) ⪯ (pais) ⪯ (estado) ⪯ (cidade)

**Descrever no documento quais são as medidas identificadas**

Flor: (quantidade)

**Criar 4 grupos de consultas analíticas no data warehouse e suas características**

Consulta 1:

**select** j.id\_jardim, ji.nome, l.pais, f.nome

**from**

jardim.jardim **as** j, jardim.jardim\_info **as** ji,

jardim.localizacao **as** l, jardim.flor **as** f

**where**

j.id\_jardim\_info = ji.id\_jardim\_info

**and** j.id\_localizacao = l.id\_localizacao

**and** j.id\_flor = f.id\_flor

**and** l.pais = 'Brasil'

Consulta o id, nome, país e nome das flores que existem nos jardins do Brasil

Consulta 2:

**select** l.continente, f.nome, **sum**(f.quantidade)

**from**

jardim.jardim **as** j, jardim.jardim\_info **as** ji,

jardim.localizacao **as** l, jardim.flor **as** f

**where**

j.id\_jardim\_info = ji.id\_jardim\_info

**and** j.id\_localizacao = l.id\_localizacao

**and** j.id\_flor = f.id\_flor

**group** **by**

l.continente, f.nome

**order** **by**

l.continente

Consulta a soma da quantidade de flores de cada espécie por continente

Consulta 3:

**select** f.nome, **avg**(f.quantidade)

**from**

jardim.jardim **as** j, jardim.jardim\_info **as** ji,

jardim.localizacao **as** l, jardim.flor **as** f

**where**

j.id\_jardim\_info = ji.id\_jardim\_info

**and** j.id\_localizacao = l.id\_localizacao

**and** j.id\_flor = f.id\_flor

**group** **by**

f.nome

Consulta a média da quantidade de flores por espécie nos jardins existentes na base de dados

Consulta 4:

**select** f.nome, l.pais, **max**(f.quantidade) **as** maior\_quantidade

**from**

jardim.jardim **as** j, jardim.jardim\_info **as** ji,

jardim.localizacao **as** l, jardim.flor **as** f

**where**

j.id\_jardim\_info = ji.id\_jardim\_info

**and** j.id\_localizacao = l.id\_localizacao

**and** j.id\_flor = f.id\_flor

**and** f.nome **in** ('bluebell','buttercup','iris','daffodil','crocus')

**group** **by**

f.nome, l.pais

**order** **by**

f.nome, maior\_quantidade **desc**

Consulta a maior quantidade de flores por espécie e país, dada as espécies bluebell, buttercup, íris, daffodil e crocus.

**Introdução ao Big Data**

**Qual é o NoSQL escolhido e por que da sua escolha?**

Optamos pelo MongoDB, pois contém um grupo de documentos que é a coleção, existe dentro de um banco de dados, pode possuir um conjunto de diferentes campos, os documentos possuem uma estrutura similar, não precisa de um esquema rígido, escala bem horizontalmente, é aprimorado para dados no volume de petabytes e segue o modelo BASE.

**Qual é o modelo de dados do NoSQL escolhido?**

Escolhemos o modelo orientado a documentos que é uma extensão do modelo chave-valor, onde o conceito básico do dado é o documento e cada documento pertence a uma coleção que por sua vez pertence a um database. Outras características é que é possível gerenciar informações de dados semi-estruturados.

**Por que seria interessante migrar do PostgreSQL ao NoSQL escolhido? detalhar a motivação.**

Seria interessante migrar para o MongoDB pois o foco é em flexibilidade e desempenho, é projetado para dados não-estruturados, é possível escalar horizontalmente por meio de sharding que particiona os dados por intervalos e atribui dados em varias instâncias e oferece grande disponibilidade pela reaplicação de dados que é uma solução que permite servidos de banco de dados tenham os mesmos dados, assim deixando mais fácil o balanceamento de carga e a redundância.

**Como implantar as tabelas de dimensão e de fato no NoSQL escolhido?**

É possível ter as tabelas de fato e dimensão da maneira abaixo através das várias coleções e gerar referencias entre os documentos delas, como abaixo:

Texto

Descrição gerada automaticamente

**Qual seria a sintaxe das consultas analíticas no NoSQL escolhido? apresente exemplos.**

Abaixo algumas sintaxes de consultas no MongoDB:

db.localizacao.find( {‘pais’:’Brasil’} )

db.flor.fin().sort( {‘nome’,’quantidade’} )

db.jardim.find().limit(5)

db.localizacao.distinct(‘cidade’)

db.flor.find({quantidade: { $gte: 10000}})

**Recuperação de Informação Baseada em Conteúdo**

Nosso DW contém a modelagem para análise de dados de plantas. Uma das tabelas possui os vetores de características já calculados a armazenados. Assim, podemos utilizar consultas por similaridade para comparar as imagens, por exemplo, usando cálculos de funções de distância.

Em uma das nossas dimensões temos nossos vetores de características armazenados:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Acima, realizamos a criação das funções de similaridade para os cálculos de distância. Escolhemos como ponto de comparação a planta bluebell de id = 1

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Realizando as consultas vemos que a consulta por KNN identificou as flores buttercup e crocus como sendo as mais similares quando comparadas a nossa flor consultada (bluebell). Abaixo, temos imagens de exemplo: bluebell e crocus, respectivamente.