

Prova em Grupo

Thiago Outeiro Pereira Damasceno - DRE: 116038363

Giovani Tricarico Barros - DRE: 118051317

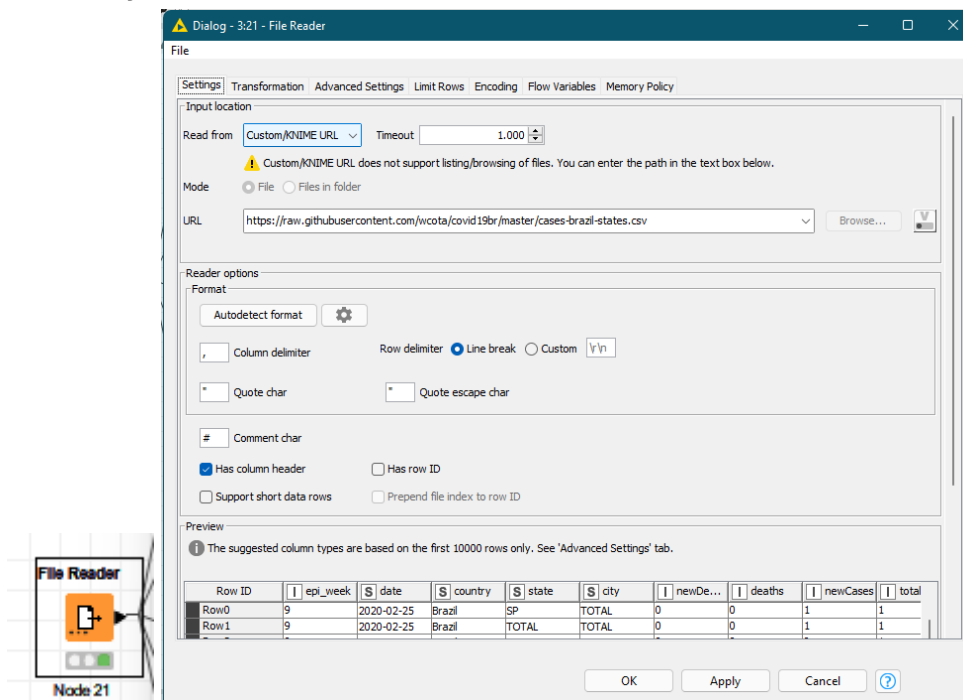
João Wendling Assayag - DRE: 118113834

1. Coleta de dados

Inicialmente nosso grupo faria o trabalho em grupo baseado no site “Painel COVID RJ” (<https://experience.arcgis.com/experience/38efc69787a346959c931568bd9e2cc4>) porém, os dados do site se mostraram fracos para o trabalho com isso decidimos buscar novos dados, chegando aos dados utilizados pela Johns Hopkins University, estes são os mesmos mostrados pelo google, ao realizar uma busca por “COVID 19 Brasil”, e para tal, eles utilizam como fonte o seguinte repositório do Github: <https://github.com/wcota/covid19br> que são dados disponibilizados pela UFV (Universidade Federal de Viçosa).

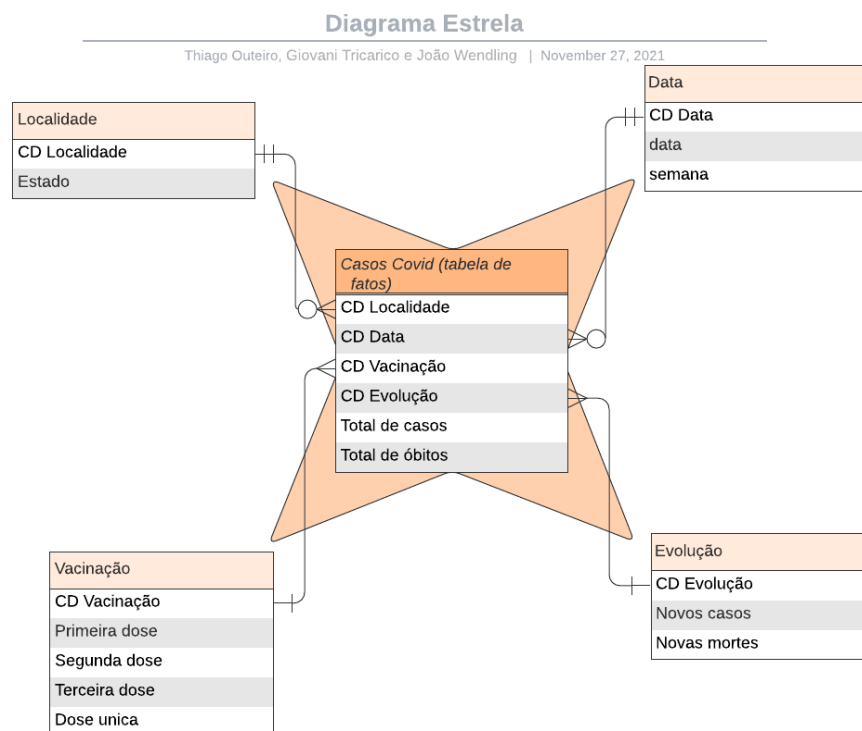
Neste repositório existem diversos tipos de dados, desde dados gerais sobre o brasil, a dados por cidades, porém para acompanhar a evolução da covid, os dados mais interessantes foram os dados por estado (<https://raw.githubusercontent.com/wcota/covid19br/master/cases-brazil-states.csv>).

Para a leitura dos dados foi utilizado o node “FileReader” com a configuração de “read from” como “Custom/KNIME URL” e passando a URL dos dados, o resto foi deixado como auto detecção.



2. Diagrama Estrela

Para a criação do modelo estrela foi observado os dados presentes no arquivo, e foi montado o seguinte diagrama:



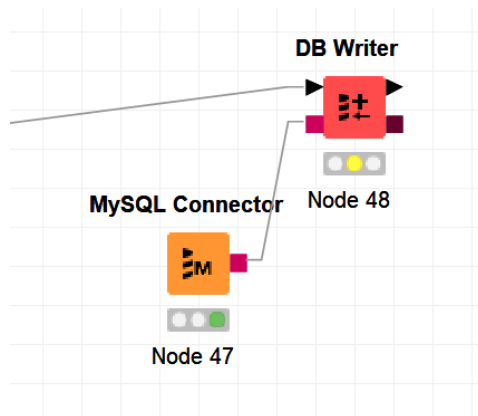
3. Criação do banco de dados

Para a criação do banco foi utilizado o aplicativo mySQL Workbench, e criado uma conexão na própria IDE como mostrado abaixo:

```
1 • create schema provaGrupoDW;  
2 • use provaGrupoDW;
```

Para a criação de tabelas foi utilizado o KNIME, no momento da carga dos dados, utilizando os nós MySQL Connector e DBWrite:

1. **MySQL Connector:** Utilizado para se conectar com o banco criado no MySQL WorkBench, sua saída é o banco que irá se conectar na entrada no DB Writer.
2. **DB Writer:** Utilizado para escrever no banco de dados recebido do MySQL Connector, irá criar a tabela passada nas configurações caso ela ainda não exista.



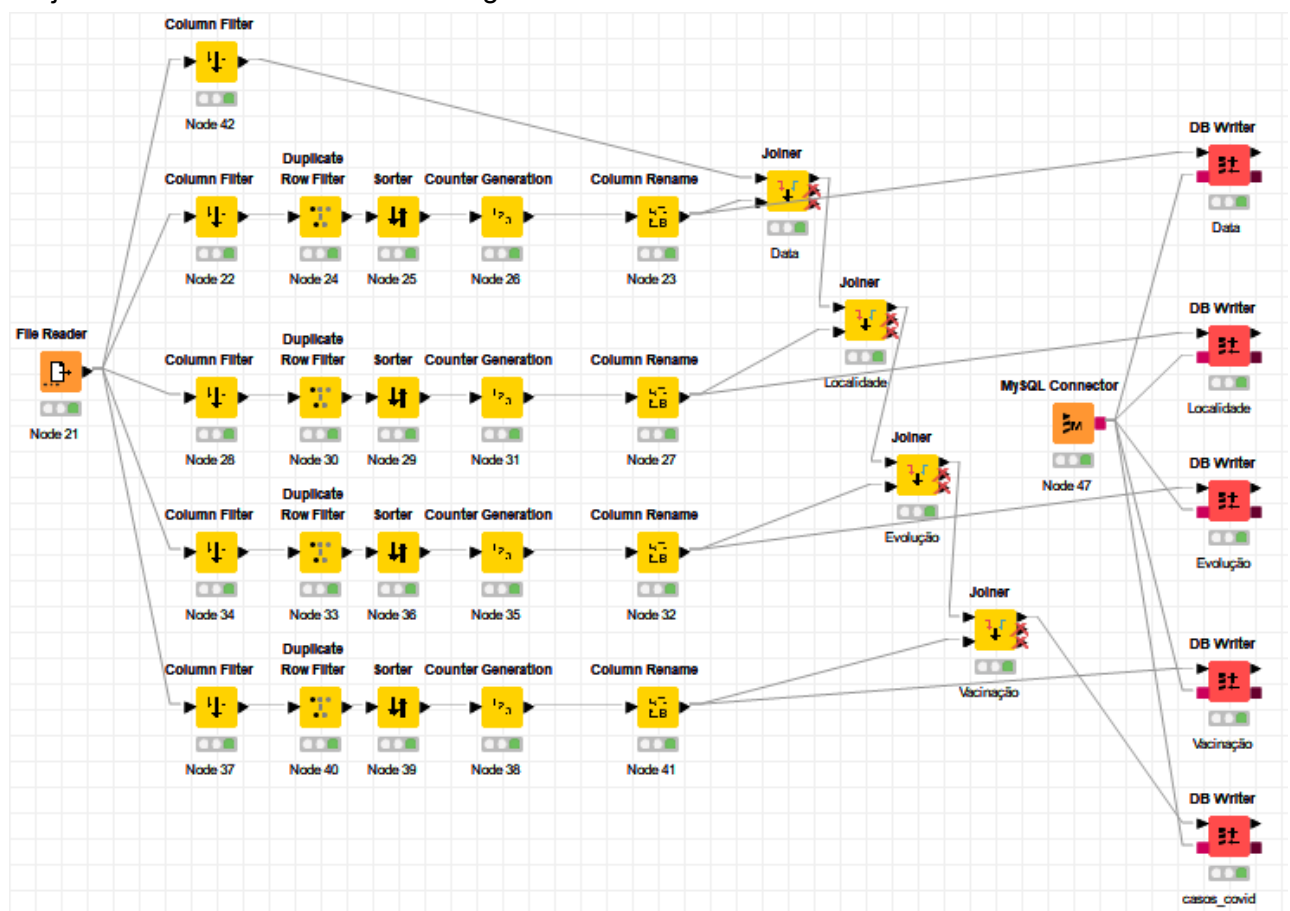
Para configurar esses nós é necessário que o banco seja criado no mySql Workbench e o schema seja criado com o seguinte script:

- 1 • `create schema provaGrupoDW;`
- 2 • `use provaGrupoDW;`

Após isso no node de Connector será necessário selecionar o banco (no caso foi utilizado o próprio main) e o no writer será selecionado o schema e a tabela a ser criada.

4. Carregamento dos dados

Para o carregamento dos dados foi utilizado o banco criado na questão 3, em conjunto com o KNIME obtendo o seguinte fluxo:

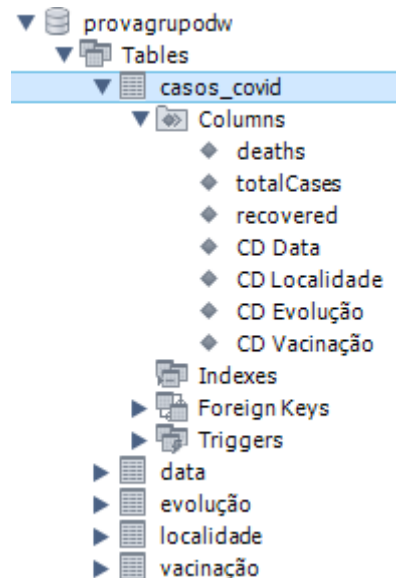


Para a carga de dados foram utilizados os nodes:

1. **Column Filter:** Utilizado para selecionar apenas as colunas necessárias para as tabelas.
2. **Duplicate Row Filter:** Utilizado nas dependências para remover as linhas duplicadas, ficando com entradas únicas.
3. **Sorter:** Não é necessário, porém foi utilizado para a visualização ficar melhor, organizando as linhas das tabelas.
4. **Counter Generation:** Utilizado para gerar os IDs (chaves) das tabelas de dependências
5. **Column Rename:** Utilizado para renomear as tabelas com o mesmo nome especificado no modelo.

Além destes 5 nodes, também foi utilizado o Node de Joiner, para realizar o pareamento da tabela de fatos com a chave para as tabelas de dependências. Todos os Joiners foram Inner Join, menos o último onde foi realizado o left outer join, pois como se tratam de dados de vacinas, no início dos casos não haviam vacinas. Como os dados eram concisos, o número de linhas entre o início e o final foi igual (17486).

Com isso na interface do banco temos:



Realizando um select na tabela de casos_covid:

- 3 • use provaGrupoDW;
- 4 • select * from casos_covid;

Result Grid							
		Filter Rows:		Export:		Wrap Cell Content:	
	deaths	totalCases	recovered	CD Data	CD Localidade	CD Evolução	CD Vacinação
▶	9403	369475	354985	555	11	4183	4402
	582243	20837967	19138604	555	27	13167	7717
	9411	369733	357475	556	11	3101	4408
	582983	20861571	19243454	556	27	13145	7718
	9418	369916	357475	557	11	2703	4412
	583621	20881896	19243454	557	27	13072	7719
	9429	370014	357475	558	11	4122	4414
	583892	20894557	19246432	558	27	12649	7720
	12313	567037	542182	559	7	4276	5333
	9434	370224	357475	559	11	1931	4415
	146567	4291993	4022314	559	25	6195	7517
	584112	20904729	19246432	559	27	12468	7721
	12318	567686	542182	560	7	2133	5335
	9439	370341	357475	560	11	1866	4418
	146595	4295149	4022314	560	25	8033	7518
	584439	20918333	19246432	560	27	12788	7722
	12333	568392	542182	561	7	5401	5345
	9447	370399	357475	561	11	3021	4421
	146610	4297229	4055982	561	25	5517	7519
	584688	20932515	19295628	561	27	12581	7723
	12341	569268	542182	562	7	3327	5350
	9455	370399	357475	562	11	3008	4422
	146828	4298180	4055982	562	25	12458	7521
	585512	20965338	19296570	562	27	13189	7724
	12350	570074	542182	563	7	3685	5356
	9459	370454	357475	563	11	1424	4428
	147020	4298851	4055982	563	25	12341	7524
	586148	20981700	19297240	563	27	13071	7725

casos_covid2 x

5. Análise de dados