Postgresql

# Prints postgresql version

SELECT version();

# Prints PostGIS version (must be in a db with postgis)

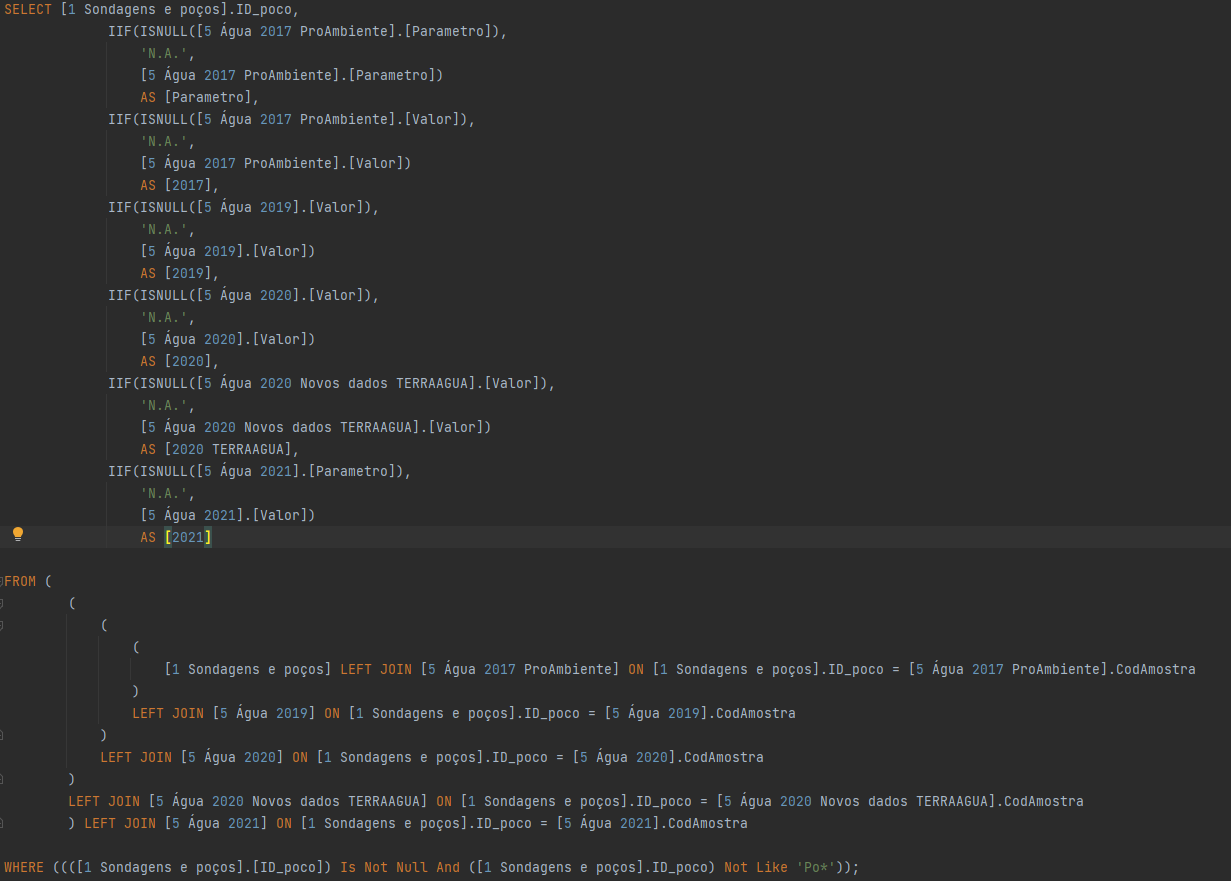
SELECT PostGIS\_full\_version();

# Show databases

\list

# Connects to <database\_name>

\c <database\_name>



Microsoft Access

O Microsoft Access (MS Access) é um aplicativo dentro da família Microsoft Office. Ele combina a Microsoft Jet Engine como sistema de gerenciamento de banco de dados com as ferramentas de um ambiente de desenvolvimento integrado, cuja interface gráfica é particularmente adequada para o grupo alvo de usuários finais para a produção de aplicações de banco de dados.

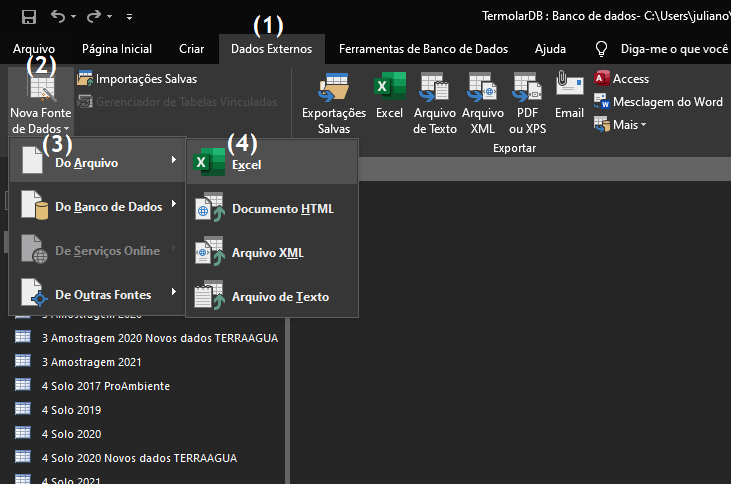
O MS Access suporta (com algumas limitações) a linguagem de banco de dados SQL a partir da versão 2007.

O MS Access pode servir para armazenamentos de dados de um projeto, facilitando consultas de dados de diferentes campanhas.

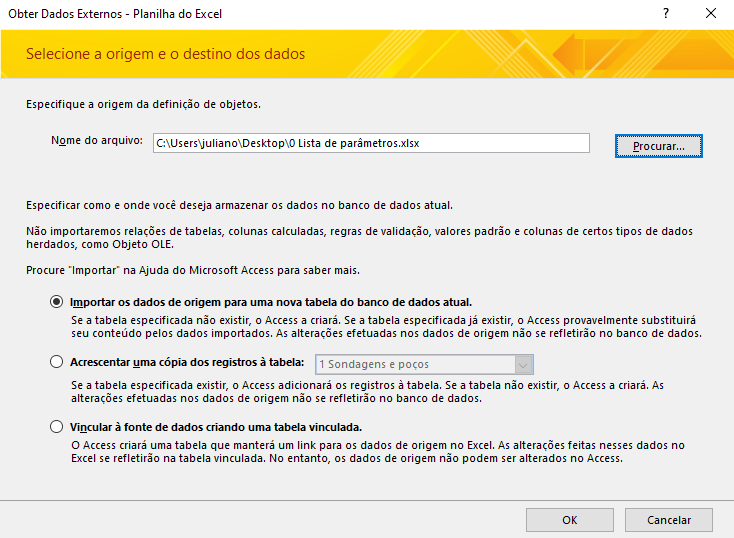
Inserção de dados

* Inserção de registro
* Inserção de tabela

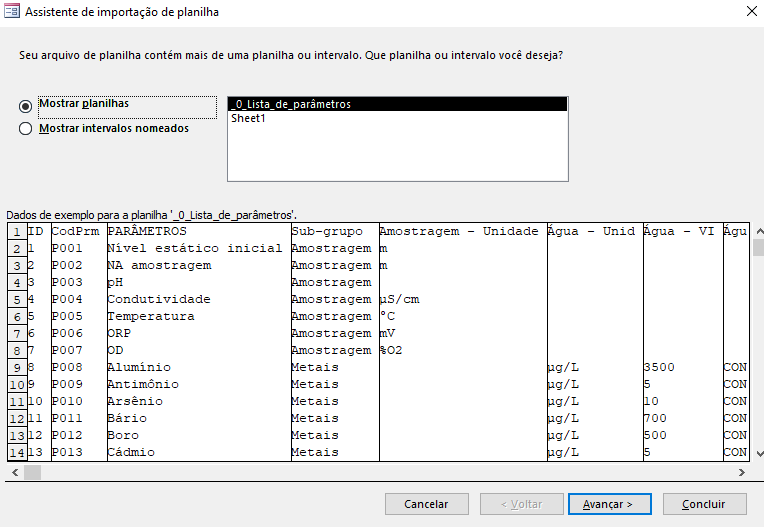
Para inserir tabelas, uma opção é partindo de um arquivo excel



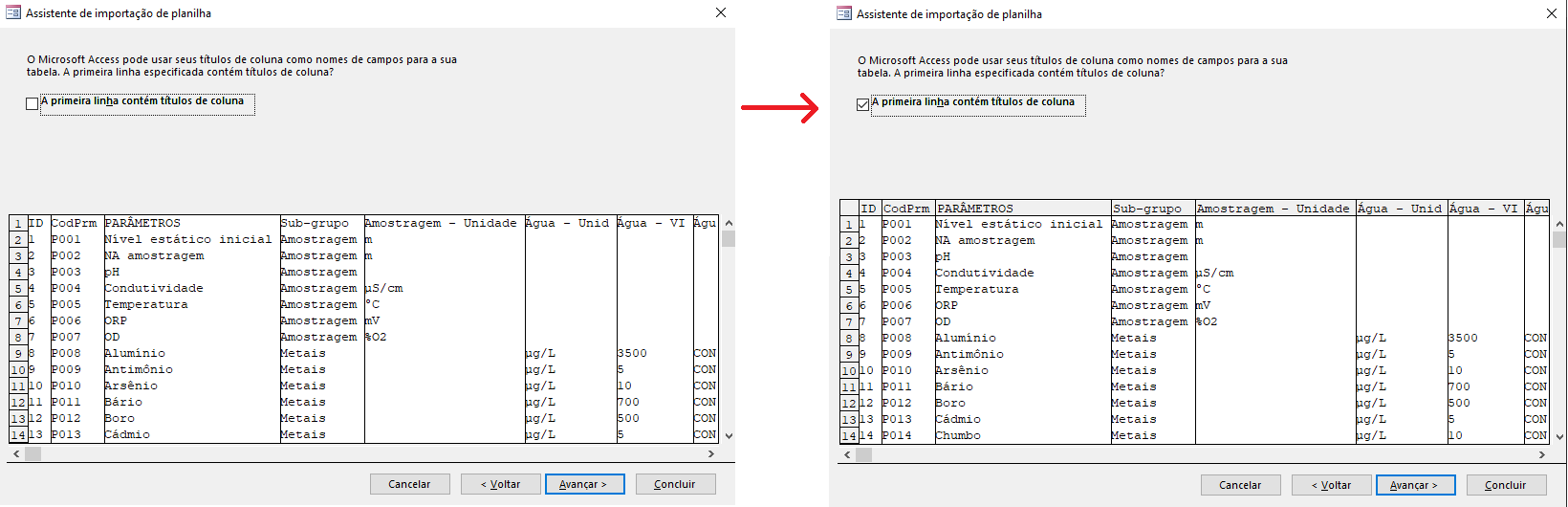
Uma nova janela de assistente de importação abrirá. Nela se seleciona o arquivo clicando em Procurar... e clica OK



Deve-se selecionar qual planilha do Excel deve ser importada. No meu caso, “\_0\_Lista\_de\_parâmetros”.



Na próxima tela, marcar caso a primeira linha contém títulos de coluna, que é o caso do exemplo. Depois, clicar “Avançar >”.



Agora, serão definidos os tipos de variáveis de cada coluna/campo. Automaticamente o Access define o tipo das variáveis baseado nos primeiros registros. Basta clicar em cada campo e alterar o tipo de dados (note que o nome do campo pode ser alterado nessa etapa pelo usuário se necessário).

Tipos de dados:

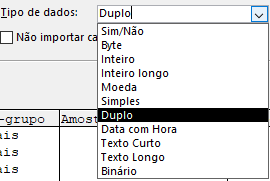
< <https://support.microsoft.com/pt-br/office/tipos-de-dados-para-bancos-de-dados-da-%C3%A1rea-de-trabalho-do-access-df2b83ba-cef6-436d-b679-3418f622e482>>

Sim/Não – booleano True/False

Duplo:

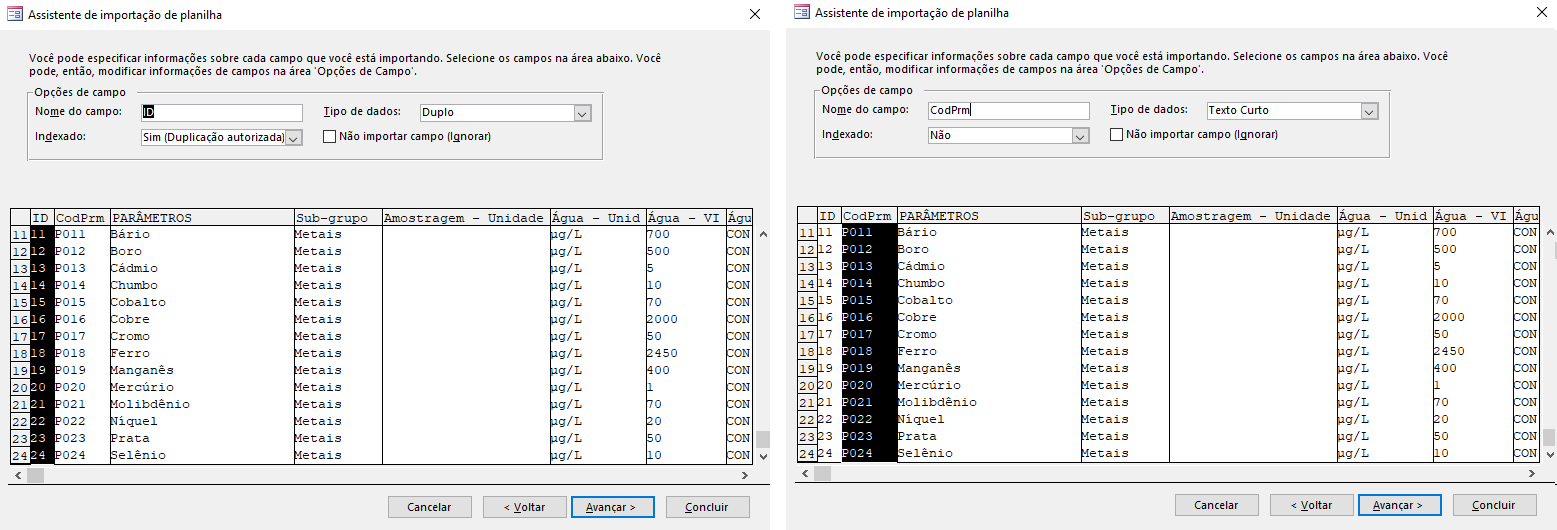
Texto Curto: até 255 caracteres

Texto Longo: até cerca de 1GB, mas o Access permite exibir apenas os primeiros 64.000 caracteres

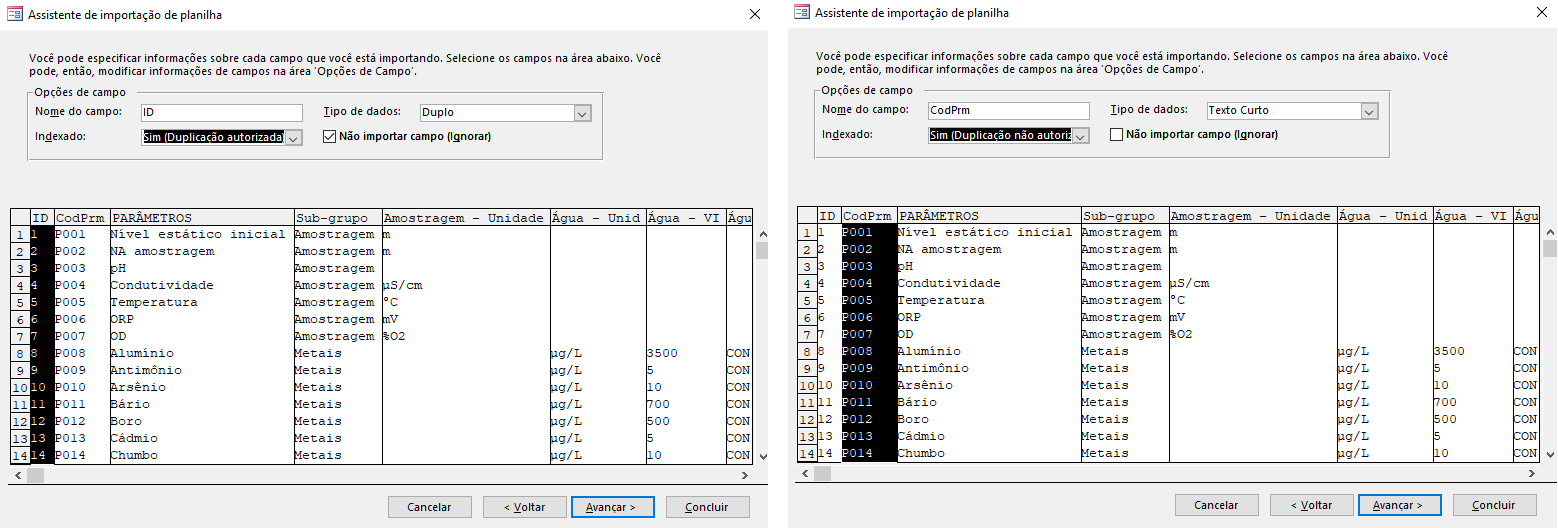


Se os primeiros registros levam a crer que o campo é numérico, mas algum registro depois é na verdade um texto, o ideal é alterar o tipo de dados para textual nessa coluna. Quando não o faz, o assistente depois retornará um erro, avisando onde não pôde converter a informação para o tipo de dados selecionado.

Na figura abaixo, o Access identificou que a coluna ID é do tipo Duplo e que CodPrm é do tipo Texto Curto:



Eu não vou importar a coluna de ID, pois ela era a coluna criada como chave primária pelo Access, que não mais usarei. Utilizarei CodPrm como chave primária, indexando com restrição de não duplicável.

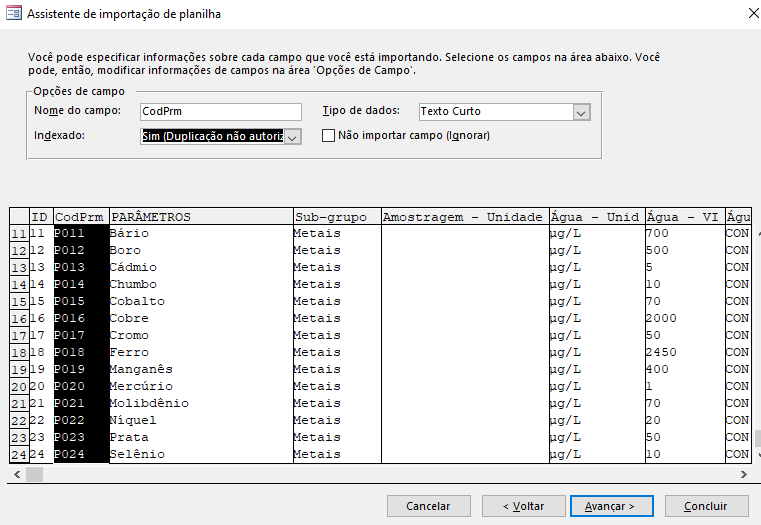


Quando indexar um campo?

*Você pode criar índices baseados em um único campo ou em vários campos. Você provavelmente desejará indexar campos pesquisados com mais frequência, os campos em que é realizada a classificação e os campos associados a campos em outras tabelas em várias consultas de tabela. Os índices podem agilizar as pesquisas e as consultas, mas podem também diminuir o desempenho quando você adiciona ou atualiza dados. Quando você insere dados em uma tabela que contém um ou mais campos indexados, o Access precisa atualizar os índices sempre que um registro é adicionado ou alterado. É provável também que adicionar registros usando uma consulta acréscimo ou acrescentando registros importados tornará o processo mais lento se a tabela de destino contiver índices.*

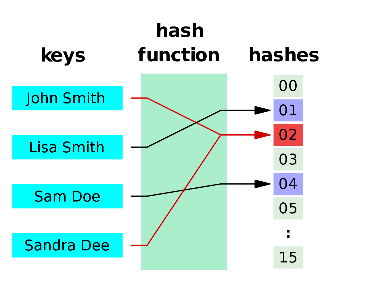
< <https://support.microsoft.com/pt-br/office/criar-e-usar-um-%C3%ADndice-para-melhorar-o-desempenho-0a8e2aa6-735c-4c3a-9dda-38c6c4f1a0ce>>

Criei o CodPrm como Indexado: Sim (Duplicação não autorizada) porque sei que vou utilizar ele bastante para fazer buscas (Queries) e sei que ele será sempre único na lógica que utilizamos na Sapotec. 1 parâmetro : 1 CodPrm

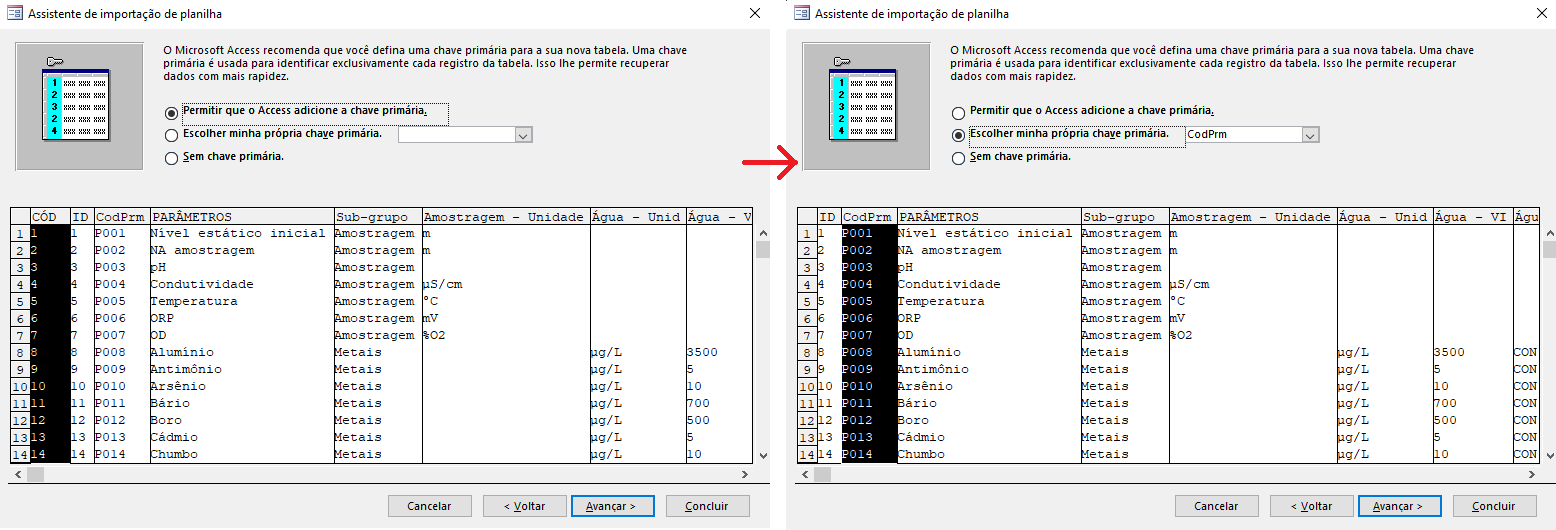


O próximo passo é sobre a chave primária. Pode-se (1) permitir que o Access adicione a chave primária, (2) escolher um campo como chave primária, ou (3) que não haja chave primária. Porém, o que é uma chave primária?

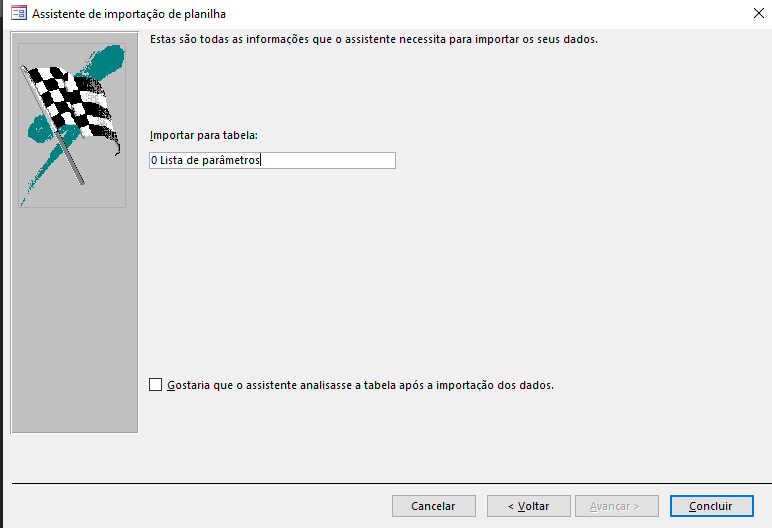
É um campo que identifica unicamente cada diferente registro de uma tabela. Ele é utilizado para relacionar diferentes tabelas. Por isso, ele é automaticamente indexado na opção de duplicação não autorizada. Nas tabelas de resultados haverá PXXX, elas serão chaves estrangeiras (FK – Foreign Keys) que permitirão chamar informações da lista de parâmetros, como a unidade e os valores de referência do parâmetro, a partir da chave primária (PK – Primary Key) que será única. Para que as relações sejam possíveis, os campos devem ter o mesmo **tipo de dado**. Registros indexados são normalmente passados por uma função hash, virando um código mais curto e de rápido acesso. Ao fazer uma busca de um registro de um campo indexado, a busca ocorrerá em O(1) porque quase não haverá perda de tempo buscando pelo campo por esse estar indexado/hashed.



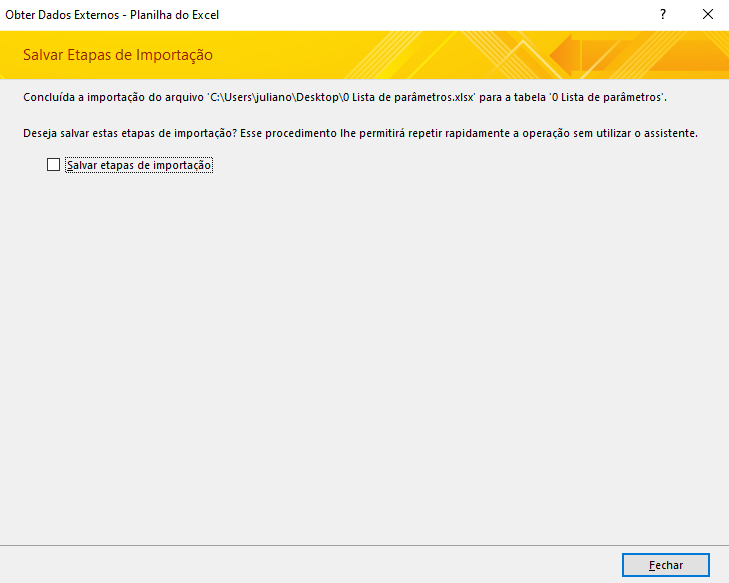
Já que essa tabela já estava no banco de dados anteriormente eu a exportei para realizar algumas modificações, o Access já havia criado a chave primária antes. Portanto, utilizarei ela novamente, clicando em “Escolher minha própria chave primária.” e selecionando o campo “ID”.



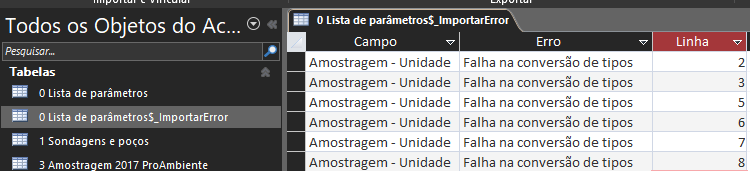
Por fim, é perguntado o nome da tabela. O nome deve ser digitado e depois clicar em concluir.



Essa será a última janela:



Se erros acontecerem durante a importação, aparecerá uma tabela de “NOME$\_importarError”. Algo assim, por exemplo:



* Inserção SQL

Conectar-se pelo Python a um Access

* Operações de atualização

2023-02-08 11:55:47 2023-02-08 14:55:47,539 ERROR [jeeves] - JeevesContextLoaderListener: Error creating bean with name 'entityManagerFactory' defined in URL [jar:file:/usr/local/tomcat/webapps/geonetwork/WEB-INF/lib/domain-3.10-SNAPSHOT.jar!/config-spring-geonetwork.xml]: Cannot resolve reference to bean 'jdbcDataSource' while setting bean property 'dataSource'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jdbcDataSource' defined in URL [file:/usr/local/tomcat/webapps/geonetwork/WEB-INF/config-db/defaultJdbcDataSource.xml]: Initialization of bean failed; nested exception is java.lang.RuntimeException: java.sql.SQLException: Cannot create PoolableConnectionFactory (Connection to postgres:5432 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.)

2023-02-08 11:55:47 org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in URL [jar:file:/usr/local/tomcat/webapps/geonetwork/WEB-INF/lib/domain-3.10-SNAPSHOT.jar!/config-spring-geonetwork.xml]: Cannot resolve reference to bean 'jdbcDataSource' while setting bean property 'dataSource'; nested exception is org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jdbcDataSource' defined in URL [file:/usr/local/tomcat/webapps/geonetwork/WEB-INF/config-db/defaultJdbcDataSource.xml]: Initialization of bean failed; nested exception is java.lang.RuntimeException: java.sql.SQLException: Cannot create PoolableConnectionFactory (Connection to postgres:5432 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.BeanDefinitionValueResolver.resolveReference(BeanDefinitionValueResolver.java:359)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.BeanDefinitionValueResolver.resolveValueIfNecessary(BeanDefinitionValueResolver.java:108)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.applyPropertyValues(AbstractAutowireCapableBeanFactory.java:1481)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCapableBeanFactory.java:1226)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireCapableBeanFactory.java:543)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapableBeanFactory.java:482)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(AbstractBeanFactory.java:306)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistry.java:230)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:302)

2023-02-08 11:55:47 at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:197)

2023-02-08 11:55:47 at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:1076)

2023-02-08 11:55:47 at org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicationContext.java:851)

2023-02-08 11:55:47 at org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:541)

2023-02-08 11:55:47 at jeeves.config.springutil.JeevesContextLoaderListener.contextInitialized(JeevesContextLoaderListener.java:87)

2023-02-08 11:55:47 at org.apache.catalina.core.StandardContext.listenerStart(StandardContext.java:4759)

2023-02-08 11:55:47 at org.apache.catalina.core.StandardContext.startInternal(StandardContext.java:5229)

2023-02-08 11:55:47 at org.apache.catalina.util.LifecycleBase.start(LifecycleBase.java:183)

2023-02-08 11:55:47 at org.apache.catalina.core.ContainerBase.addChildInternal(ContainerBase.java:753)

2023-02-08 11:55:47 at org.apache.catalina.core.ContainerBase.addChild(ContainerBase.java:727)

2023-02-08 11:55:47 at org.apache.catalina.core.StandardHost.addChild(StandardHost.java:695)

2023-02-08 11:55:47 at org.apache.catalina.startup.HostConfig.deployDirectory(HostConfig.java:1177)

2023-02-08 11:55:47 at org.apache.catalina.startup.HostConfig$DeployDirectory.run(HostConfig.java:1925)

2023-02-08 11:55:47 at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)

2023-02-08 11:55:47 at java.util.concurrent.FutureTask.run(FutureTask.java:266)

2023-02-08 11:55:47 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)

2023-02-08 11:55:47 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)

2023-02-08 11:55:47 at java.lang.Thread.run(Thread.java:750)

2023-02-08 11:55:47 Caused by: org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'jdbcDataSource' defined in URL [file:/usr/local/tomcat/webapps/geonetwork/WEB-INF/config-db/defaultJdbcDataSource.xml]: Initialization of bean failed; nested exception is java.lang.RuntimeException: java.sql.SQLException: Cannot create PoolableConnectionFactory (Connection to postgres:5432 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.)