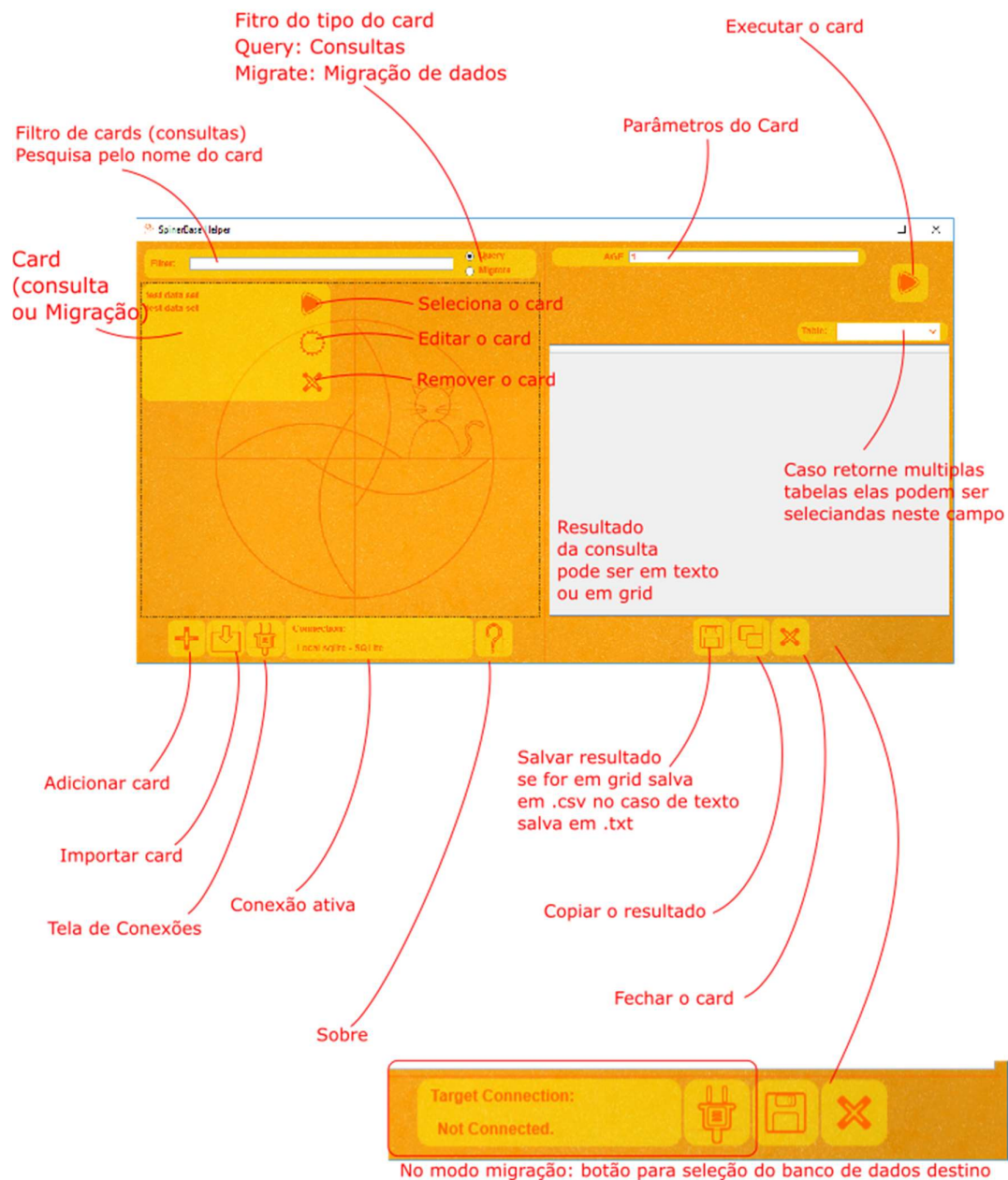


Tela Principal



No modo migração: botão para seleção do banco de dados destino

Edição de cards

Nome do card

Descrição do card

Tipo do banco de dados

Tipo do card
Query: Consulta
Migrate: Migração

Tipo de resultado do card

Parâmetros extraídos

Nome de exibição do parâmetro

Extrair parâmetros do comando

Comando em SQL

Comando em python para tratar o parâmetro antes do mesmo ser integrado a consulta.

Os parâmetros são compostos da seguinte maneira:
<%nome@TIPO%>
os tipos podem ser:
NUMBER = para numeros inteiros
DATE = Para Datas
TEXT = Para texto
Caso não seja informado ou seja outro valor será considerado texto

Comando em python para o comando SQL antes da substituição dos parâmetros

Comando em python para tratar o resultado da consulta ou migração (no caso de migração trata o resultado antes de executa-lo no banco de dados de destino)

Salvar o card

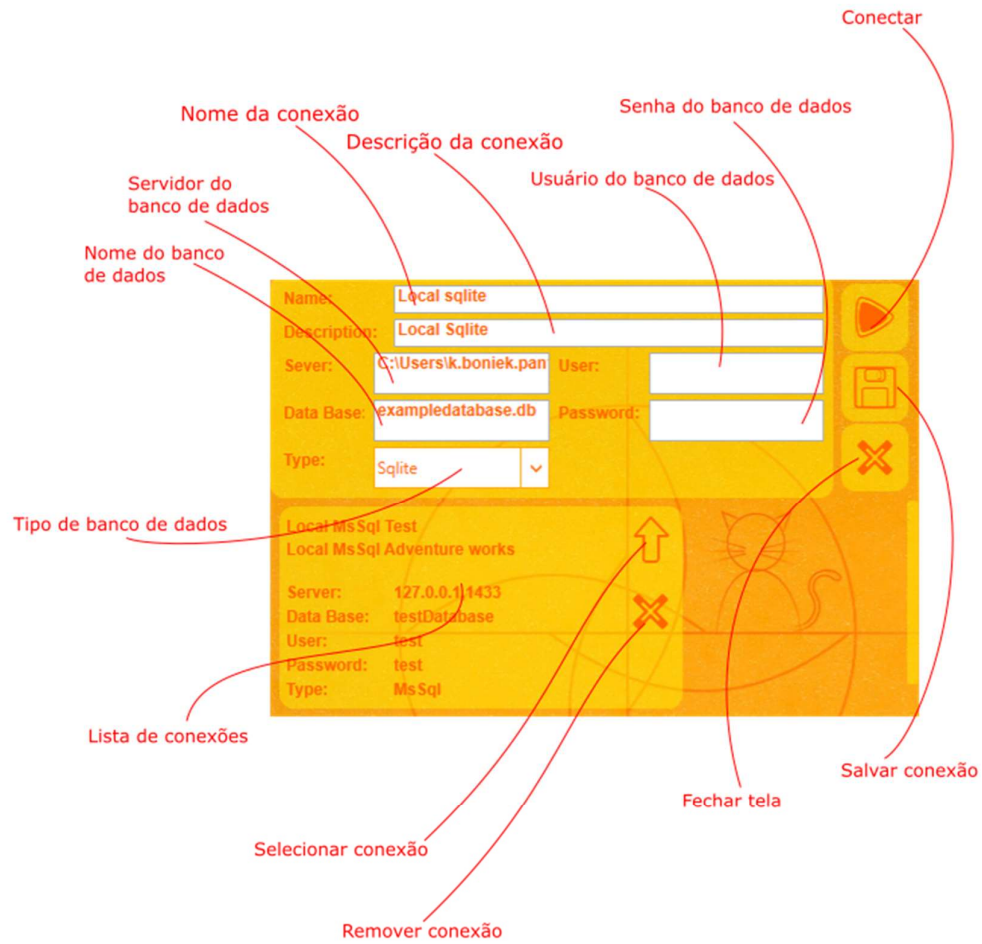
Exportar o card

O comando pode ser dividido em partes com a palavra chave:
<!STATEMENT!> cada parte retornará uma tabela de resultados, uma parte pode acessar o resultado das outras com a palavra chave <!campo@tabela!> o campo é o nome da coluna ou indice e a tabela pode ser o nome da mesma ou indice, para referenciar uma tabela pelo nome deve-se nomear a mesma, para isso utilizar a palavra chave <!TABLENAME@nomeDaTabela!> dentro do statement.

Form fields and controls:

- Name: test data set
- Description: test data set
- Tag: <%AGE@NUMBER%>
- Description: AGE
- Data Base Type: ☐ MySQL ☒ SQLite ☐ MySQL ☐ Oracle
- Card Type: ☒ Query ☐ Migrate
- Result Type: ☒ Grid ☐ Text
- Example: <%PARAMETER%>
- SQL Command: select * from testTable where age > <%AGE@NUMBER%>
<!TABLENAME@tabela_A!>
<!STATEMENT!>
select * from testTable where name = '<!name@tabela_A!>
<!TABLENAME@tabela_B!>
- Python Command: Os parâmetros são compostos da seguinte maneira: <%nome@TIPO%> os tipos podem ser: NUMBER = para numeros inteiros DATE = Para Datas TEXT = Para texto Caso não seja informado ou seja outro valor será considerado texto
- Python Command: Comando em python para o comando SQL antes da substituição dos parâmetros
- Python Command: Comando em python para tratar o resultado da consulta ou migração (no caso de migração trata o resultado antes de executa-lo no banco de dados de destino)
- Buttons: Pre, Pos, Save, Export

Edição de conexões



Editor Python

Adiciona método
main obrigatório

Adiciona import para o clr e System.Data
necessário para tratar retornos
em DataSet (modo resultado em grid)



```
import clr
clr.AddReference("System.Data")
from System import Data
from System.Data import DataTable
from System.Data import DataSet
from System.Data import DataColumn
from System.Data import DataRow

def main(input):
    objReturn = DataSet()
    objReturn.Tables.Add(DataTable())
    objReturn.Tables[0].Columns.Add(DataColumn("return"))
    strAuxBaseInsert = ""
    strAuxInsert = ""

    strAuxBaseInsert = "insert into tabletest("
    for column in input.Tables[0].Columns:
        strAuxBaseInsert = strAuxBaseInsert + str(column.Co
        if input.Tables[0].Columns.IndexOf(column) != (input.1
            strAuxBaseInsert = strAuxBaseInsert + ", "

    strAuxBaseInsert = strAuxBaseInsert + ") values ("

    for row in input.Tables[0].Rows:
        strAuxInsert = strAuxBaseInsert
```

Create an function called main with a single parameter
Example: def main(input): return "Hello"

Comando:

Salvar

O comando deve obrigatoriamente conter a definição
de uma função chamada main recebendo um parâmetro:
def main(input):

Esta função deve retornar uma string no caso de tratamento de
parâmetros e resultado em texto e retornar um DataSet no caso
de tratamento de resultado em grid;

Podem ser adicionadas quantas funções forem necessárias além
da função main.