Armazenando dados

Armazenando o resultado do scraping no MySQL

Armazenando dados

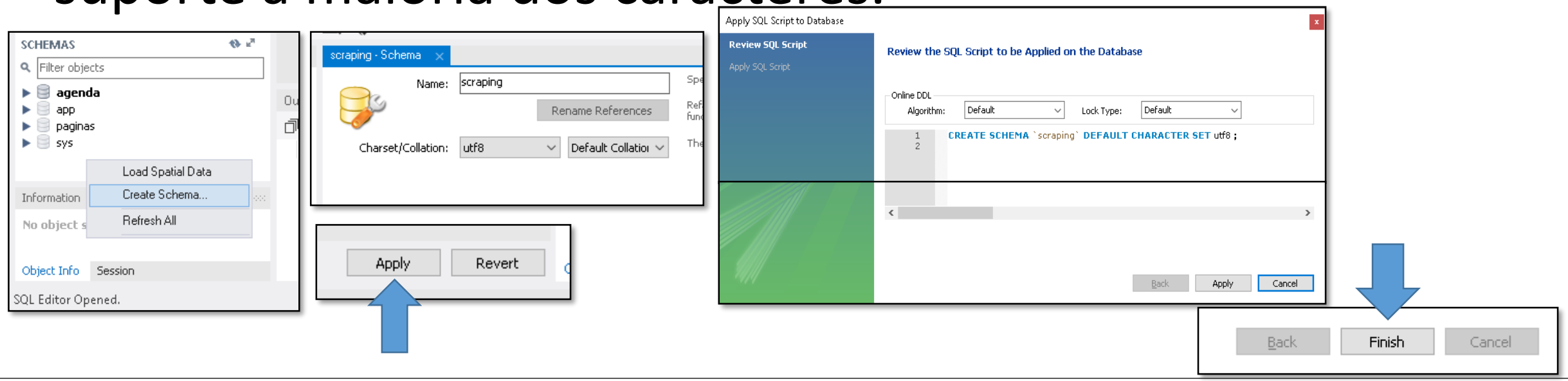
Objetivo da aula

Nesta aula vamos criar uma aplicação para navegar em artigos do Wikipedia, buscando os links do artigo e armazenando as urls, títulos e conteúdo das páginas percorridas.

Armazenando dados

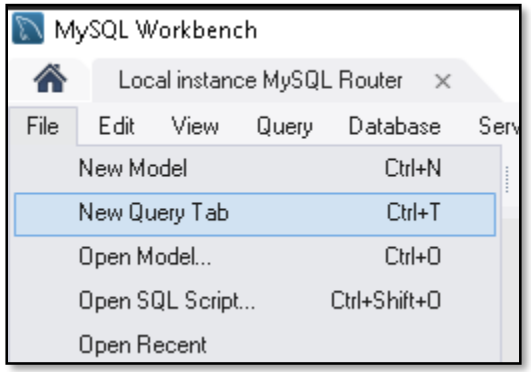
Criando o schema

Abra o MySQL Workbench, conecte ao seu banco de dados e, na área schema crie um schema chamado scraping. Escolha o charset utf8 para que o banco suporte a maioria dos caracteres.



Armazenando dados

Criando a tabela



Abra uma janela de consulta e execute o script para criação da tabela (código disponibilizado junto à aula).

```
CREATE TABLE `scraping`.`paginas` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `titulo` VARCHAR(200) NULL,
  `url` VARCHAR(200) NULL,
  `conteudo` VARCHAR(10000) CHARACTER SET 'utf8' NULL,
  `data` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;
```

Armazenando dados

Implementando o algoritmo

Primeiro vamos importar as bibliotecas necessárias.

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
import random
import mysql.connector
import re
```

Depois vamos implementar a conexão com o banco.

```
dados_conexao = {"user":"root", "password":"1234", "host":"127.0.0.1",
"database":"scraping", "charset":"utf8"}
conexao = mysql.connector.connect(**dados_conexao)
cursor = conexao.cursor()
```

Armazenando dados

Implementando o algoritmo

Implementando o método para gravar os dados.

```
def gravar(titulo, url, conteudo):
    cursor.execute('INSERT INTO paginas (titulo, url, conteudo)'
                   'VALUES (%s, %s, %s)', (titulo, url, conteudo))
    conexao.commit()
```

Armazenando dados

Implementando o algoritmo

Implementando o método para retornar os links da página.

```
def getLinks(urlArtigo):
    url = 'http://pt.wikipedia.org'+urlArtigo
    html = urlopen(url)
    bs = BeautifulSoup(html, 'html.parser')
    titulo = bs.find('h1').get_text()
    conteudo = bs.find('div', {'id':'mw-content-text'}).find('p').get_text()
    gravar(titulo, url, conteudo)
    return bs.find('div', {'id':'bodyContent'}).\
        findAll('a', href=re.compile('^(/wiki/)((?!:).)*$'))
```

Armazenando dados

Implementando o algoritmo

Finalizando o programa.

```
links = getLinks('/wiki/Copa_do_Mundo_FIFA_de_2026')

try:
    contador = 1
    while len(links) > 0 and contador <= 10:
        novoArtigo = links[random.randint(0, len(links)-1)].attrs['href']
        print(str(contador) + " -> " + novoArtigo)
        links = getLinks(novoArtigo)
        contador += 1
finally:
    cursor.close()
    conexao.close()
```

FIM