

**Instituto Superior Técnico**

**BASE DE DADOS SQL**

**Formação em Contexto de Trabalho**

**Daniel Salvador Maurício Duarte**

Índice

[Introdução 5](#_Toc136329175)

[Ligação do SQL ao HTML via Python 7](#_Toc136329176)

[Ligação do SQL ao Python 9](#_Toc136329177)

[Selecionar Informação 9](#_Toc136329178)

[Modificar Tabelas 9](#_Toc136329179)

# Introdução

# Este relatório tem como objetivo explorar a integração entre a linguagem SQL (Structured Query Language) e Python, duas poderosas ferramentas utilizadas na programação e análise de dados. O SQL é uma linguagem padrão usada para gerenciar e manipular bases de dados relacionais, enquanto o Python é uma linguagem de programação versátil e de fácil aprendizagem.

# Neste relatório, abordaremos a interligação entre SQL e HTML usando Python. Por meio do uso das bibliotecas sqlite3 e Flask, descobrimos uma forma de conectar o SQL a páginas HTML. Utilizando o SQLite como SGBD, pudemos criar consultas SQL em Python e exibi-las em um site HTML. Isso permitiu-nos visualizar os dados de uma tabela em formato tabular.

# Além disso, exploramos a conexão entre o SQL e o Python, aprendendo a selecionar informações de tabelas usando consultas SQL e a modificar tabelas por meio de comandos SQL. Utilizamos o módulo sqlite3 para estabelecer a conexão com a base de dados, executar consultas e realizar ações de modificação.

# Com esse conhecimento, pudemos extrair dados de uma tabela e exibi-los no output, bem como realizar modificações nas tabelas, como inserção, atualização e exclusão de registros.

# Neste relatório, iremos fornecer uma visão geral dos conceitos e códigos essenciais para a interligação entre SQL e Python, demonstrando a sua utilidade na manipulação e visualização de dados em um contexto web.

# Ligação do SQL ao HTML via Python

Devido a já ter conhecimento de SQL aprofundado, começei logo a pesquisar com interligar o SQL ao HTML, e usando o ChatGPT, fonte de pesquisa de todo o meu trabalho, encontrei uma forma via Python que me permitira fazer essa interligação, utilizando então o pycharm baixei as bibliotecas sqlite3 e Flask. Primeiramente escolhi o SQLite pois permitia-me linhas de código muito mais simples em python e foi a opção que menos me deu problemas de instalação, então recorrendo às bibliotecas antes referidas pude então copiar um select da base de dados para um site em HTML usando este código:

import sqlite3  
from flask import Flask, render\_template  
  
app = Flask(\_\_name\_\_) – criação da instância ‘app’  
  
  
@app.route('/')- define uma rota para a raiz do aplicativo web  
def table():  
 conn = sqlite3.connect('vectrix.db') – conexao a BDcursor = conn.cursor() – criação do cursor para executar consultas   
 cursor.execute('SELECT \* FROM Cliente') – executa o comando SQL nele escrito neste caso o SELECTdata = cursor.fetchall() – recebe os dados selecionados e retorna-os como uma lista de tuplas  
 conn.close() – fecha a conexão com a BD  
 return render\_template('table.html', data=data)- retorna os dados ‘fetched’ como argumento  
  
  
if \_\_name\_\_ == '\_\_main\_\_': - Faz com que o servidor Flask seja somente executado quando o script for executado corretamente  
 app.run()- inicia o servidor de desenvolvimento do Flask

E o código em HTML sendo:

<!DOCTYPE html>  
<html>  
<head>  
 <title>SQLite to HTML</title>  
</head>  
<body>  
 <table> - cria uma tabela para exibir dados  
 <thead> - define cabeçalho da tabela  
 <tr> - define uma linha da tabela   
 <th>ID</th>   
 <th>NOME</th>  
 <th>IDADE</th>  
 <th>Data\_registo</th>  
 <th>Telefone</th>  
 <th>Morada</th>  
 <th>Cod\_Postal</th>  
 <th>Email</th>  
 </tr> - define fim da linha do cabeçalho  
 </thead> - marca fim do cabeçalho  
 <tbody> - define a secção do corpo da tabela  
 {% for row in data %} – define loop que percorrerá cada linha de dados  
 <tr>   
 <td>{{ row[0] }}</td>  
 <td>{{ row[1] }}</td>  
 <td>{{ row[2] }}</td>  
 <td>{{ row[3] }}</td>  
 <td>{{ row[4] }}</td>  
 <td>{{ row[5] }}</td>  
 <td>{{ row[6] }}</td>  
 <td>{{ row[7] }}</td>  
 </tr>  
 {% endfor %} – marca o fim do loop  
 </tbody>  
 </table>  
</body>  
</html>

Tendo então descoberto uma maneira de interligar o SQL ao HTML passei então a procurar maneiras de conectar o SQL ao Python tópico que irei abordar agora.

# Ligação do SQL ao Python

## Selecionar Informação

Para a ligação do SQL ao Python eu usei a mesma lógica da ligação do SQL ao Html pegando então na função *table* e tirando o que tinha mais haver com o Flask fiquei com esta syntax:

conn = sqlite3.connect('vectrix.db')  
conn.execute("PRAGMA foreign\_keys = ON")#Codigo usado para que seja respeitada as regras das chaves externascursor = conn.cursor()  
cursor.execute(("SELECT \* FROM Cliente"))

data = cursor.fetchall()  
conn.close

Tendo a necessidade de alterar o código de forma a que desse print do Select no output pesquisei um vídeo no Youtube sobre prints de selects chegando assim então a esta syntax:

conn = sqlite3.connect('vectrix.db')  
cursor = conn.cursor()  
cursor.execute("SELECT \* FROM CLIENTE WHERE ID = 11")  
Cliente = cursor.fetchall()  
print('Cliente:',Cliente[0])  
conn.close()

No caso esta alteração consiste em adicionar a variável ***Cliente*** que vai carregar os valores Selecionados da tabela e no print que vai escrever os dados do cliente que, neste caso, tem o ID = 11 sendo o ***Cliente [0]*** o mesmo que pedir ao ***Cliente*** para retornar apenas a primeira linha da lista.

## Modificar Tabelas

A este ponto em que já sabia selecionar dados da tabela, decidi então descobrir se era possível modificar as tabelas sendo assim ao pesquisar encontrei esta syntax:

conn = sqlite3.connect('vectrix.db')  
conn.execute("PRAGMA foreign\_keys = ON")cursor = conn.cursor()  
cursor.execute(("Comando em SQL"),(Valores a inserir))  
conn.commit() – commit serve para que as alterações desejadas sejam validadas na tabela  
conn.close

Como se pode ver tirando o commit não há grande diferença para o Select já que acaba por ser a mesma logica de importar o sqlite3, conectar a Database e efetuar ações com a mesma.

# Conclusão

A integração entre SQL e Python oferece uma gama de combinações de ferramentas para o gerenciamento e análise de dados. Através da biblioteca ***sqlite3*** do Python, é possível estabelecer uma conexão com bases de dados SQLite e executar consultas SQL de forma eficiente.

Ao utilizar o ***Flask***, um framework web em Python, é possível interligar SQL com HTML, permitindo exibir os resultados dos Selects da Database em páginas web.

Além disso, a integração entre SQL e Python também permite realizar modificações nas tabelas, como inserir, atualizar ou excluir registros. Através do uso do comando ***execute ()*** e do ***commit (),*** as alterações desejadas podem ser aplicadas e validadas na tabela.

A combinação dessas duas linguagens, SQL e Python, proporciona um ambiente flexível e eficiente para a manipulação de dados. Com o conhecimento adequado das sintaxes SQL e das bibliotecas Python, é possível realizar uma variedade de tarefas, desde consultas simples até operações complexas de modificação e análise de dados.

# Bibliografia

ChatGPT - <https://chat.openai.com/c/68ff354f-2e90-4175-b8d0-f8d82590acc>

# Python & SQLite: Print Data from Tables - <https://www.youtube.com/watch?v=l2IQ7uKE9LE>

# <../pythonProject/funcoes.py>

<../pythonProject/Vectrix.py>

# <../pythonProject/templates/table.html>

<../pythonProject/untitled.py>