

ASSESSMENT



Instituto Infnet

Escola Superior de Tecnologia da Informação (ESTI – EAD)

Análise e Desenvolvimento de Sistemas

Projeto de Bloco

Python

Aluno: Alceu Wanderlei Valim de Lima Junior

Matrícula (CPF): 00936806001

Professor: Adalberto Igor de Souza

24 de setembro de 2023

ETAPA 1

Criação da base SQLite

Comecei baixando e instalando o DB Browser for SQLite, uma ferramenta gráfica de código aberto para trabalhar com bases de dados SQLite. Após a instalação, abri o DB Browser for SQLite e selecionei a opção "Nova Base de Dados" para iniciar o processo de criação. Com a nova base de dados criada, utilizei a funcionalidade "Importar" e selecionei "Tabela a partir de arquivo CSV". Naveguei até o local do CSV fornecido e o carreguei na base de dados. Com os dados carregados, comecei a criar e testar as consultas solicitadas no PB, usando o editor SQL integrado ao DB Browser.

Exercícios de SQL

1. A partir da base carregada completa, crie uma nova tabela contendo apenas as duas variáveis escolhidas (uma numérica e uma categórica).

```
CREATE TABLE interest AS SELECT LotArea, Street FROM housing_data;
```

2. A partir da tabela criada no item 01, apresente o agrupamento da variável categórica, incluindo a contagem de cada resultado possível.

```
SELECT Street, COUNT(*) as count
FROM interest
GROUP BY Street;
```

3. A partir da tabela criada no item 01, apresente a listagem de elementos da variável numérica que estejam contidos entre o valor mínimo e a média (excluindo ambos os limites).

```
SELECT LotArea
FROM interest
WHERE LotArea > (SELECT MIN(LotArea) FROM interest)
AND LotArea < (SELECT AVG(LotArea) FROM interest);
```

4. A partir da tabela criada no item 01, elimine os dados que pertencem à categoria mais representativa da base, em quantidade de elementos.

```
DELETE FROM interest
WHERE Street = (
    SELECT Street
    FROM interest
    GROUP BY Street
    ORDER BY COUNT(*) DESC
    LIMIT 1
);
```

ETAPA 2

Exercício no Google Looker Studio

<https://lookerstudio.google.com/reporting/f606daa0-1582-4e52-811b-f44ad67d43fd>

ETAPA 3

OBS: O SCRIPT PODE SER CONFERIDO NO SEGUINTE LINK

https://github.com/Alceu-wv/python_pb_at

1. Faça o consumo dos dados a partir da base SQLite criada para a execução do PB e guarde os dados em uma estrutura apropriada da linguagem (dataframe).

```
import sqlite3
import pandas as pd

# Conectar com a base SQLite
conn = sqlite3.connect('tp9.sqlite')

# Carrega a tabela
df = pd.read_csv("AmesHousing.csv")
df.to_sql('housing', conn, if_exists='replace', index=False)

# Carregar os dados em um dataframe desde o banco de dados
df = pd.read_sql_query("SELECT `Lot Area`, Street FROM housing", conn)
```

2. A partir dos dados carregados na estrutura do item 09, apresente o conteúdo das duas variáveis indicadas.

```
print(df[['Lot Area', 'Street']])
```

3. A partir dos dados carregados na estrutura do item 09, apresente os valores máximo, mínimo e a média dos valores da variável numérica escolhida.

```
print("Máximo:", df['Lot Area'].max())  
print("Mínimo:", df['Lot Area'].min())  
print("Média:", df['Lot Area'].mean())
```

4. A partir dos dados carregados na estrutura do item 09, apresente uma listagem de itens únicos da variável categórica escolhida.

```
print(df['Street'].unique())
```

ETAPA 4

1. A partir dos dados carregados na estrutura do item 09, crie uma lista contendo apenas os dados da variável numérica.

```
lista_num = df['Lot Area'].tolist()
```

2. A partir dos dados carregados na estrutura do item 09, crie uma nova lista, agora contendo apenas os dados da variável categórica.

```
lista_cat = df['Street'].tolist()
```

3. A partir da lista dos dados da variável numérica do item 13, crie uma estrutura de repetição que apresente na tela a soma dos valores acima da média dos valores da própria variável.

```
media = sum(lista_num) / len(lista_num)
soma_valores_acima_media = sum([x for x in lista_num if x > media])
print("Soma dos valores acima da média:", soma_valores_acima_media)
```

4. A partir da lista dos dados da variável categórica do item 14, crie uma função que, ao ser aplicada na lista (a lista deve ser passada para a função), retorne a contagem de ocorrência dos itens individuais da variável categórica escolhida.

```
def contagem_itens(lista):
    return {item: lista.count(item) for item in set(lista)}

print(contagem_itens(lista_cat))
```