



13.3k xp



66%

Transcrição

Agora que temos a tabela de matrículas no banco, queremos realizar uma busca nesta tabela a partir de uma *query* SQL, algo que faremos em uma nova seção "Buscando do banco sql". Criaremos uma variável `query` na qual definiremos a nossa busca, começando com `select * from`. A ideia é trazer todos os cursos com menos de 20 pessoas matriculadas, portanto, `select * from matriculas where quantidade_de_alunos < 20`.

```
query = 'select * from matriculas where quantidade_de_alunos < 20'
```

COPIAR CÓDIGO

Em seguida, chamaremos a função `pd.read_sql()` passando como parâmetros a `query` criada e nossa `engine`.

```
pd.read_sql(query, engine)
```

COPIAR CÓDIGO

Como resultado, teremos uma tabela contendo `id_curso`, `quantidade_de_alunos` e `nome_do_curso`, listando somente os cursos com menos de 20 inscritos.



13.3k xp

a



66%

	id_curso	quantidade_de_alunos	nome_do_curso
0	6	17	PHP com MySql
1	10	12	Código limpo com C#
2	16	16	Estatística Básica
3	20	18	Orientação objetos com Java

A partir do comando `read_sql_table()` , podemos buscar uma tabela inteira, bastando passarmos o nome da tabela (`matriculas`), a `engine` e quais colunas desejamos filtrar. Dessa vez, pediremos primeiramente o `nome_do_curso` seguido da `quantidade_de_alunos` .

```
pd.read_sql_table('matriculas', engine, columns=['nome_do_curso', 'quantidade_de
```

COPIAR CÓDIGO



	nome_do_curso	quantidade_de_alunos
0	Lógica de programação	62
1	Java para Web	52
2	C# para Web	25
3	Ruby on Rails	33
4	Cursos de Python	59

Atribuiremos essa consulta a uma variável `muitas_matriculas` .



13.3k xp

a



66%

```
muitas_matriculas = pd.read_sql_table('matriculas', engine, columns=['nome_do_cu
```

COPIAR CÓDIGO

O próprio Pandas nos disponibiliza uma função capaz de criar *queries*, chamada `query()`. Com ela é possível, por exemplo, filtrar uma `quantidade_de_alunos` maior do que `60`.

```
muitas_matriculas.query('quantidade_de_alunos > 60')
```

COPIAR CÓDIGO

	nome_do_curso	quantidade_de_alunos
0	Lógica de programação	62
1	TDD com Java	5270
2	Preparatório para certificação Java	81
3	Análise de dados	82
4	Boas práticas em Java	67

Podemos repetir o processo apenas para cursos com mais de `80` inscritos e atribuir o resultado a uma variável.

```
muitas_matriculas = muitas_matriculas.query('quantidade_de_alunos > 80')
muitas_matriculas
```

COPIAR CÓDIGO



13.3k xp



66%

	nome_do_curso	quantidade_de_alunos
2	Preparatório para certificação Java	81
3	Análise de dados	82

É possível escrevermos tal resultado como um campo da nossa base de dados local. No Colab, criaremos uma seção "Escrevendo no banco". Em seguida, a partir de `muitas_matriculas`, chamaremos a função `to_sql()` nomeando o campo `muitas_matriculas` e passando um parâmetro `con=engine`.

```
muitas_matriculas.to_sql('muitas_matriculas', con=engine)
```

COPIAR CÓDIGO

Ao imprimirmos os nomes das tabelas em nosso banco com `engine.table_names()`, teremos:

```
print(engine.table_names())
```

COPIAR CÓDIGO

```
['matriculas', 'muitas_matriculas']
```

Agora sabemos criar e trabalhar com um banco de dados local, inclusive realizando *queries* tanto em SQL quanto utilizando o próprio Pandas.

ATIVIDADES
3 de 6

DISCORD
ALURA

FÓRUM DO
CURSO

VOLTAR
PARA
DASHBOARD

MODO
NOTURNO



13.3k xp

a