**03 - Buscando no banco**

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 trazermos 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.

|  | **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\_alunos'])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.

muitas\_matriculas = pd.read\_sql\_table('matriculas', engine, columns=['nome\_do\_curso', 'quantidade\_de\_alunos'])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\_matriculasCOPIAR CÓDIGO

|  | **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.