

Introdução a Banco de Dados – Trabalho 1

Daniel A. P. Castro¹, Eric Y. Fukuyama¹

¹Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba – Paraná – Brasil

danielcastro@alunos.utfpr.edu.br, ericfukuyama@alunos.utfpr.edu.br

Abstract. *This report refers to the project of the discipline Introduction to Database, offered in the period 2022/2 by Professor Leandro Batista de Almeida. The task consists in the development of a database management tool, based on CSV data loading and queries processed in the tables. The project was carried out in Python language and had a code that extensively used string manipulation. It was successful in executing the task, with complete and well-structured functions.*

Resumo. *Este relatório se refere ao trabalho da disciplina Introdução a Banco de Dados, ofertada no período de 2022/2 pelo Professor Leandro Batista de Almeida. A tarefa consiste no desenvolvimento de uma ferramenta de gerenciamento de bancos de dados, baseada em carregamento de dados de CSV e consultas processadas nas tabelas. O projeto foi realizado em linguagem Python e contou com um código que utilizou extensivamente a manipulação de strings. Houve êxito em executar a tarefa, com funções completas e bem estruturadas.*

1. Introdução

Foi orientado que o programa de carregamento e consultas deveria executar as seguintes funções: importação de dados – selecionar um diretório onde estarão os arquivos de dados em formato CSV, carregando um arquivo para cada tabela, com o nome do arquivo dando o nome à tabela; os dados deverão ser armazenados em arrays em memória; permitir a consulta aos dados, em formato SQL, com as seguintes cláusulas possíveis:

```
projeção (lista de campos ou *)  
filtros (where)  
ordenação (order by)
```

Instruiu-se ainda que os filtros e ordenação poderiam ser feitos por um ou dois campos, com modificadores AND e OR.

A linguagem de programação recorrida ao trabalho proposto foi o Python [Foundation 2022], devido à fluidez e facilidade que ela permite ao trabalhar com dados ou *strings*.

2. Carregamento de Dados

Para fins de teste, foram carregados dados do exemplo Employee, disponível em um repositório do GitHub [Maxia 2022]. Para isso, utilizou-se a interface gráfica do próprio

MySQL Workbench [Oracle 2022] a qual possui um mecanismo de conversão de tabelas em arquivos CSV.

Após esse processo, gerou-se um projeto na linguagem escolhida e foi criado um arquivo principal, intitulado "Trabalho1.py", ao qual inicialmente carrega os dados de CSV e os converte em listas de *strings*. Para isso, é solicitado ao usuário especificar o caminho até o arquivo, bem como as tabelas que ele desejaria consultar. Essas informações são passadas para a função *criarTabela()* e assim é retornada uma lista de tabelas.

3. Processamento das Queries

Após o carregamento de dados, é pedido que se digite um comando de consulta na tabela, por meio de outro arquivo que trata disso (Comandos.py). Inicialmente o comando é dividido em listas que separam cada cláusula com suas respectivas informações. Isso possibilita organização para realizar consultas efetivas.

Após esse processo, é chamada a função *fromTudo* que retorna a tabela solicitada pela cláusula *From*. Caso tenha sido escrita mais de uma tabela (separada por vírgula), ela é unida de modo a repetir todas as linhas de uma para cada linha de outra. Isso é realizado gerando-se uma nova tabela que chama seu método pré-definido *append()*.

Além disso, para conseguir filtrar as condições presentes na cláusula *where*, foram feitas algumas funções. Desse modo, a função "procuraWhere" possui como parâmetro uma string com a instrução do *WHERE* presente no comando de SQL, tal função serve para identificar qual é a operação de comparação. Dessa forma, ela retorna essa informação, junto com os dois elementos que vão ser comparados. Ademais, a função "comparacao" possui como intuito comparar os elementos de acordo com o comando, assim, dado certos atributos de entrada, a "comparacao" retorna o booleano *True* caso a comparação descrita pelo comando seja verdadeira. Por fim, a função "where" utiliza das duas funções auxiliares citadas nesse parágrafo para retornar uma tabela com todas as condições satisfeitas. Tal lógica utilizou diversas comparações para o tratamento da string.

Após o processo de especificar condições sobre os dados e atributos, existe a possibilidade de ordenar as linhas conforme a cláusula *Order By*. Para isso, bastou utilizar o algoritmo Merge Sort – baseado em um código pronto [Programiz 2022] –, cujo tempo de execução é $O(n \lg n)$, a fim de realizar a ordenação em ordem crescente. Contudo, para quando for dada a especificação *DESC* (decrescente), utilizou-se o método pré-definido *reverse()*, cujo tempo é $O(n)$ para inverter a ordem da Query.

Por último, são filtrados os dados com a cláusula *Select* conforme um algoritmo que seleciona o índice de coluna correspondente aos atributos especificados e em seguida são armazenados os dados referentes a ela dentro de uma tabela auxiliar criada (que é retornada no final). Caso o usuário escrever "*", é retornada a tabela inteira.

Em um repositório específico do GitHub [Fukuyama 2022] é possível visualizar o processo de implementação do projeto, bem como as etapas ao trabalhar com cada cláusula. Na Figura 1 encontra-se uma simulação da interface criada em terminal e um comando completo executado.

```
Qual a pasta que está as tabelas? C:\\Users\\danie\\Documents\\Faculdade\\2022.2\\Introducao a Banco de Dados\\BancoDeDados\\Trabalho1
Nome de quais tabelas serão usadas? Escrever entre espaços: departments
Qual o comando SQL? select dept_no, dept_name from departments where dept_no < d005 and dept_name > D order by dept_no
['dept_no', 'dept_name']
['d001', 'Marketing']
['d002', 'Finance']
['d003', 'Human Resources']
['d004', 'Production']
```

Figura 1. Simulação de Interface e Comando

4. Considerações Finais

O projeto cumpriu com seu papel de simular a consulta de dados do MySQL através da programação, em prol das ferramentas e artifícios proporcionados pela linguagem Python. Dessa maneira, evidencia-se a aplicação dos conceitos trabalhados até então na disciplina de Introdução a Banco de Dados.

Referências

- Foundation, P. S. (2022). Python. Technical report, <https://www.python.org/>. Acesso em: 23 October 2022.
- Fukuyama, E. Y. (2022). Bancodedados. Technical report, <https://github.com/EricFukuyama/BancoDeDado>. Acesso em: 27 October 2022.
- Maxia, G. (2022). test_db. Technical report, https://github.com/datacharmer/test_db. Acesso em: 23 October 2022.
- Oracle (2022). Mysql workbench. Technical report, <https://www.mysql.com/products/workbench/>. Acesso em: 23 October 2022.
- Programiz (2022). Merge sort algorithm. Technical report, <https://www.programiz.com/dsa/merge-sort>. Acesso em: 27 October 2022.