

RELATÓRIO EP MAC0350

Matheus Sanches Jurgensen - 12542199

Paulo Henrique Albuquerque dos Anjos de Souza - 12542251

11 de Julho de 2023

PROJETOS CONCEITUAL E LÓGICO

Destaca-se que, além deste relatório e alguns outros *scripts*, a entrega deste projeto contém também os modelos conceitual e lógico do problema solicitado. Ambos estes modelos estão dentro da pasta “*Conceitual_e_Lógico*”. Dentro dela, é possível encontrá-los em dois formatos: PDF ou XML. Utilizando os arquivos .xml, é possível visualizá-los pelo *draw.io*, que foi o software que utilizamos para desenhar esses modelos.

COMO EXECUTAR

Além deste relatório, a entrega consiste em outros 3 *scripts*. Deve-se dar uma atenção maior ao script *load_db.sh*, pois é ele que executa os principais comandos, executando os outros *scripts*. Nesse sentido, o **único arquivo a ser executado é o *load_db.sh***. Para facilitar a organização, os outros 2 arquivos (um em SQL e o outro em Bash) foram colocados na pasta /aux, visto que eles não precisam ser executados diretamente.

Em resumo, o que ele faz é, em primeiro lugar, copiar o SQL *script* (***ScriptEP.sql***) para o diretório /tmp, porque tivemos dificuldades com o usuário *postgres* não conseguir acessar arquivos na *home* de outros usuários. Nesse sentido, copiamos o SQL *script* ao diretório /tmp para que ele fique acessível ao usuário *postgres*.

Após isso, acessado o *postgres*, é feita a criação do banco de dados ***bd_mac0350***, no qual serão armazenadas as estruturas deste projeto. Em seguida, esse banco é acessado e, dentro dele, é executado o SQL *script*, o qual é responsável por criar as tabelas, preenchê-las e criar as funções que armazenam as consultas solicitadas.

Depois, é executado um *script* adicional (***output_to_file.sh***), que apenas direciona as chamadas de todas as funções armazenadas a um arquivo, caso isso facilite a correção das consultas pelo professor. O arquivo de saída é o */tmp/output_file.txt*.

Por fim, chama-se um comando para que *psql* seja mais uma vez executado e o banco *bd_mac0350* acessado, para que o professor possa acessar as estruturas armazenadas (tabelas, funções, etc.) pela linha de comando.

Vale ressaltar que, caso algum dos scripts não estejam com permissão de execução, basta executar, no terminal, o comando “*chmod +x <nome_do_arquivo>*”.

CONTEÚDO DAS TABELAS

Conforme poderá ser conferido, as tabelas foram todas preenchidas com valores fictícios, apenas com o intuito de que as consultas possam ser verificadas. Nesse sentido, como já preenchemos todas as tabelas com os dados necessários, além de termos enfrentado dificuldades para implementar a interface do banco de dados, optamos por acessá-lo pelo terminal e pelo **pgAdmin**. Pensamos que isso fez sentido, visto que, como explicitado no enunciado e em aula, a principal função da interface web é carregar as tabelas para que sejam utilizadas as consultas solicitadas.

Nesse sentido, após ser executado o *script* que cria o banco e o carrega com as estruturas e informações solicitadas, o banco de dados pode ser acessado e visualizado tanto por terminal, como pelo *pgAdmin*.

CONSULTAS

Foram implementadas todas as consultas solicitadas no enunciado. Apenas um caso específico deve ser destacado, com relação à terceira consulta: no mundo real, podem ocorrer disciplinas de mesmo código, em um mesmo intervalo de tempo, mas como “turmas” diferentes da disciplina. Além disso, uma única turma de uma disciplina pode ser composta por mais de um professor.

Com isso, tratando o oferecimento como um relacionamento triplo, conforme solicitado pelo professor, não é possível distinguir “turmas” considerando-se que elas podem ter mais de um professor. Em outras palavras, não é possível saber, dadas duas entradas na tabela “oferecimento”, com mesmas datas de início e mesma disciplina, mas professores diferentes, se são uma mesma turma da disciplina, com dois professores, ou se são turmas diferentes.

Nesse sentido, implementamos duas funções para a terceira consulta, cada uma interpretando de forma diferente essa problemática. A primeira delas, nomeada *disciplinas_mais_oferecidas1()*, considera que cada professor corresponde a uma turma diferente. Ou seja, assume-se que não há turmas com mais de um professor. Por outro lado, a segunda função, nomeada *disciplinas_mais_oferecidas2()*, considera que não pode haver turmas simultâneas (ou, em outras palavras, iniciadas em um mesmo dia) de uma mesma disciplina.

A tabela de oferecimentos foi cuidadosamente preenchida de forma a destacar a diferença entre essas duas funções implementadas. Analisando-se a disciplina de ID 8, percebe-se que cada uma das funções conta de forma diferente a quantidade de vezes que ela foi oferecida.

Por fim, destaca-se a forma como uma função pode ser chamada pelo terminal: “*SELECT * FROM nome_da_função();*”. Deve-se, apenas, alterar o nome da função pelo nome de alguma das 5 funções implementadas:

- Consulta 1: *serviços_por_perfil()*
- Consulta 2: *serviços_usados_perfil()*
- Consulta 3:
 - Versão 1: *disciplinas_mais_oferecidas1()*
 - Versão 2: *disciplinas_mais_oferecidas2()*
- Consulta 4: *docentes_desde_2020*

Vale destacar, para finalizar, que, implementando a consulta 4, consideramos que, quando o enunciado se refere a “*disciplinas oferecidas no período de maio de 2020 a maio de 2023*”, são disciplinas cuja data de início é a partir de maio de 2020 e data de fim é até maio de 2023.