Banco de Dados II
Prática 2
Profª. Vanessa Souza

Assunto: Conexão com Banco de Dados

Objetivo de Aprendizagem: Praticar os conceitos vistos em sala sobre segurança (*injection*), driver de conexão e mapeamento objeto-relacional.

1. Introdução

O padrão de design de software MVC (*Model-View-Controller*) é um dos mais utilizados no desenvolvimento de software. Cada camada tem uma responsabilidade clara e separada, o que facilita a manutenção, escalabilidade e teste durante o desenvolvimento da aplicação. Essa separação clara de responsabilidades facilita o desenvolvimento por diferentes equipes.

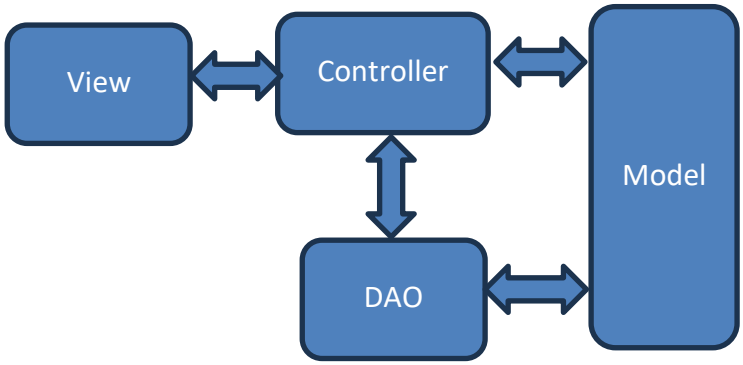
A *View* é responsável pela interação com o usuário. O *Model* é responsável por representar os dados e pela lógica do negócio. O *Controller* faz a ligação adequada entre a View e o Model.

O padrão MVC gera uma aplicação com 3 camadas. No entanto, existe uma forma ainda melhor de desacoplamento entre aplicação e banco de dados, chamada **DAO - *Data Access Object***. DAO é um padrão de design utilizado em programação para abstrair e encapsular todas as interações com o banco de dados. Ele separa a lógica de negócios da lógica de acesso a dados, promovendo uma arquitetura mais modular e de fácil manutenção. O uso do DAO faz a chamada arquitetura de 3.5 camadas. Sendo assim, todas as funções que acessam o banco diretamente ficam encapsuladas dentro do DAO. Ou seja, o DAO isola o código de acesso aos dados do restante da aplicação, tornando-o independente de detalhes de implementação de armazenamento de dados.

No diagrama da Figura 1 há um exemplo de como fica a arquitetura da aplicação utilizando MVC e DAO. No caso, o Controller sempre acessa o DAO para operações relativas à persistência do dado e, nesse caso, o DAO interage com o Model para executar as operações necessárias no banco de dados.

Quando o Controller precisa realizar operações de lógica da aplicação, mas que não são operações de persistência do dado, ele acessa o Model diretamente. Por exemplo, se o objetivo é incluir um funcionário no banco de dados, o Controller recebe os dados desse novo funcionário na View, instancia um objeto funcionário e chama a função de inserir funcionário no módulo DAO. O DAO faz a persistência efetiva do dado no banco.

No entanto, caso fosse necessário validar alguma informação do funcionário como, por exemplo, se seu CPF é válido, o Controller pode chamar um método direto no Model para fazer isso e, posteriormente, acessar o DAO para inserção do novo funcionário.



2. Descrição da Atividade

Nessa atividade vocês deverão implementar uma aplicação escrita em python e que utilize os padrões MVC e DAO. O objetivo é inserir um novo pedido no banco de dados northwind. No entanto, vocês deverão implementar essa aplicação de duas maneiras. Na primeira, utilizando apenas o driver de conexão psycopg. Posteriormente, adaptar a mesma aplicação para utilizar o ORM SqlAlchemy.

Na camada *View*, o usuário deverá ter acesso a uma maneira de informar o nome do cliente, nome do vendedor, dados do pedido e itens do pedido. Para tanto, façam um formulário e, na **implementação com drive**, vocês devem simular uma *SQL Injection*. Ou seja, o formulário teria dois *backends*. Um que permite a *injection* e outro que não permite.

Dados da tabela que podem ser nulos, podem ser ignorados para criar um novo pedido.

Para gerar o modelo de classes (Model), vocês deverão utilizar o aplicativo sqlacodegen[1].

ATENÇÃO: Na implementação apenas com o driver de conexão, manter o model. Porém, sem as informações de relacionamento no objeto. Ou seja, podem gerar o modelo de classes com o sqlalcodegen, mas devem adaptá-lo para usar com o driver de conexão.

[1] https://github.com/agronholm/sqlacodegen

Além da inserção de um novo pedido, a aplicação deverá implementar dois relatórios:

a) Informações completas sobre um pedido:
    - Número do pedido
    - Data do pedido
    - Nome do cliente
    - Nome do vendedor
    - Os itens do pedido
        i. Produto, quantidade e preço

b) Ranking dos funcionários por intervalo de tempo
    - Nome do funcionário
    - Total de pedidos realizados
    - Soma dos valores vendidos

Os relatórios devem ser feitos na aplicação e não podem acessar views.

Antes de iniciar o desenvolvimento da aplicação, certifique-se de que o banco de dados possui as chaves estrangeiras listadas abaixo. Caso não tenha, vocês devem cria-las.

- Tabela Orders: chave estrangeira para customers e employess
- Tabela Order_Details: chave estrangeira para orders e products

3. Entrega

A atividade pode ser feita em grupo de 3 pessoas.
A entrega deverá ser um documento que contenha:

- Nome dos integrantes do grupo
- Link para o github com o código do projeto
- Link para um vídeo onde o grupo explica e demonstra a implementação da atividade.

ATENÇÃO: Apenas um dos integrantes do grupo deve fazer a entrega no SIGAA.

4. Nota

Cada parte da atividade terá uma nota diferente e será lançada como se fossem atividades independentes:

1. Aplicação com driver
2. Implementação da injection
3. Aplicação com ORM
4. Relatórios