

Curso: Análise e Desenvolvimento de Sistemas – ADS

Ano: 2023/1

Orientação Técnica (OT) – 14

OT14 - Banco de Dados

Introdução

Antes de iniciarmos a programação do Registro de Produtos, é necessário que você adquira alguns conceitos sobre mais um assunto: Banco de Dados.

Até agora todos os programas que temos desenvolvido recebem determinadas informações do usuário que são perdidas após o término de sua execução. Esta abordagem é necessária quando se trata de um primeiro contato com programação, porém, é inviável na maioria dos programas comerciais: já imaginou perder todas as suas informações bancárias após o sistema do banco ser reiniciado? Para evitar este tipo de situação faz-se necessário armazenar determinadas informações coletadas pelo seu programa de forma duradoura e confiável, para que elas não sejam perdidas durante suas várias execuções. Uma das formas de se atingir este objetivo é armazená-las em um banco de dados.

Assim, realize os fichamentos indicados a seguir.

FICHAMENTOS

- Banco de dados
 - Definição
 - Principais modelos/tipos de banco de dados da atualidade + Vantagens/Desvantagens (ex.: noSQL...)
- Banco de Dados Relacional
 - Definição
 - Tabelas
 - Colunas/Campos
 - Registros/tuplas
 - Chave Primária
 - Chave estrangeira
- Linguagem SQL
 - Definição
 - Visão geral dos subconjuntos do SQL
 - DDL
 - DML
 - DQL
 - Sintaxe dos comandos para
 - Criação de Banco de Dados
 - Criação de tabelas
 - Inserção de registros em uma tabela
 - Alteração de registros de uma tabela
 - Exclusão de registros de uma tabela
 - Consulta de registros de uma tabela
- Sistemas Gerenciadores de Banco de Dados
 - Definição
 - Como funcionam
 - MySQL
 - Definição

Forma de entrega: No formato de trabalho academico seguindo a NBR 14724 ABNT em PDF anexado ao AVA e enviado ao e-mail do professor orientador.

Segunda parte apresentação relacionada ao tema para apresentação dos conteúdos elencados.

Prática para aquisição de conhecimentos

Para praticar o que viu nessa pesquisa, você vai criar um banco de dados via MySQL Workbench, chamado **“mercado”** e executar nele alguns comandos DML e DQL, seguindo as instruções da **OT1 de Banco de Dados**.

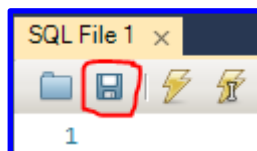
Criação do BD do projeto para a Coldigo



Agora que já desenvolveu a primeira prática em um exemplo genérico, vamos criar o banco de dados da Coldigo.

Primeiro, no Workbench, crie um novo arquivo SQL :



A intenção desse arquivo é **reaproveitarmos os códigos** que criaremos, levando em consideração principalmente o fato de que **a cada dia de trabalho você deverá recriar o banco de dados se estiver nos nossos computadores**. Assim, salve-o em algum local de fácil acesso dentro de sua workspace.



Dica sobre executar os comandos: até o fim dessa OT, você terá vários comandos SQL nesse mesmo arquivo. Caso não tenha notado, existem 2 botões no Workbench para executá-los: o , que executa todos os comandos SQL do arquivo ou a porção selecionada (seja essa seleção de um ou mais comandos), e o , que executa o comando em que você tiver posicionado o cursor. Assim, nada de ficar deletando o comando depois de executar, pelo contrário, mantenha-o no arquivo, afinal de contas essa é mesmo a ideia: criarmos UM arquivo com TODOS os comandos necessários para preparar o banco para uso...

UniSENAI

Agora que já temos um arquivo onde armazenaremos nossos códigos, devemos **criar o banco** em si:

```
CREATE DATABASE IF NOT EXISTS bdcoldigo DEFAULT CHARACTER SET utf8 ;
```

Repare que já o colocamos como UTF-8, para evitar qualquer problema de codificação de caracteres, já que nosso projeto foi criado assim.

Agora, precisamos **criar as tabelas neste banco**, mas para isso **algum comando deve ser feito antes...** Qual? Bem, isso é com você!

Depois de fazer o que faltava, vamos **criar a tabela marcas**. Ela é necessária para identificar a marca de cada produto.

```
CREATE TABLE IF NOT EXISTS marcas (  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
  nome VARCHAR(45) NOT NULL,  
  PRIMARY KEY (id)  
);
```

Veja que ela tem apenas 2 campos, sendo **id** a chave primária, o que a torna simples de entender.

Desafio: O que é aquele unsigned no id?

Agora, finalizaremos com a **criação da tabela produtos**, onde armazenaremos cada produto registrado no sistema. Aplique em seu código:

```
CREATE TABLE IF NOT EXISTS produtos (  
    id INT(5) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT,  
    categoria TINYINT(1) UNSIGNED NOT NULL,  
    modelo VARCHAR(45) NOT NULL,  
    capacidade INT(4) UNSIGNED NOT NULL,  
    valor DECIMAL(7,2) UNSIGNED NOT NULL,  
    marcas_id INT UNSIGNED NOT NULL,  
    PRIMARY KEY (id),  
    INDEX fk_produtos_marcas_idx (marcas_id ASC),  
    CONSTRAINT fk_produtos_marcas  
        FOREIGN KEY (marcas_id)  
        REFERENCES marcas (id)  
);
```

Veja que ela tem 6 campos, sendo **id** a chave primária. Repare também naquele “**CONSTRAINT**”, que indica que **marcas_id** é uma **chave estrangeira** que serve de **referência** para o campo **id** da tabela **marcas**.

Desafio: Pesquise sobre zerofill e os tipos de campo utilizados.

Agora, nesse mesmo arquivo, **crie usando o comando INSERT 3 marcas** nesse mesmo arquivo SQL para popularmos o BD. As usaremos quando formos registrar nossos produtos. **Não insira produtos!** Isso será feito diretamente pelas páginas que criaremos :)

Com isso, **finalizamos a criação do banco de dados da Coldigo** (pelo menos a parte que utilizaremos nas próximas atividades). Não esqueça de **salvar o arquivo**, incluindo o código que insere as 3 marcas, para não perder o que fez: ele será usado em **TODO** o restante do projeto!

Conclusão

Depois de adquirir esses conhecimentos todos, estamos aptos a começar a programação backend!

Na OT seguinte, iniciaremos o Registro de Produtos, pela parte de cadastrar um novo produto no banco de dados do nosso sistema. Até lá!

Após finalizar a OT, comunique um orientador para novas instruções.