**MANUAL DE PROGRAMAÇÃO**

**Senac Vila Prudente**

**MySQL**

**O que é MySQL?**

MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) amplamente utilizado. Ele permite armazenar e recuperar dados de forma eficiente, facilitando a organização e manipulação de informações.

**Instalação do MySQL**

Antes de começar, é necessário instalar o MySQL no seu sistema. Você pode fazer isso baixando e instalando a versão adequada para o seu sistema operacional no site oficial do MySQL.

**Conceitos Básicos**

**Conectando ao MySQL**

A conexão ao MySQL pode ser feita por meio da linha de comando ou ferramentas gráficas. Utilizando o comando **mysql -u seu\_usuario -p**, você acessa o MySQL inserindo seu nome de usuário e senha quando solicitado.

**Criando um Banco de Dados**

A criação de um banco de dados é essencial. O comando **CREATE DATABASE nome\_do\_banco;** estabelece um novo banco de dados para armazenar informações.

****

**Selecionando um Banco de Dados**

A instrução **USE nome\_do\_banco;** seleciona o banco de dados no qual você deseja executar comandos.

****

**Manipulação de Tabelas**

**Criando Tabelas**

**Texto

Descrição gerada automaticamente com confiança baixa**As tabelas são estruturas fundamentais no MySQL. O comando **CREATE TABLE** define uma tabela especificando os nomes e tipos de dados das colunas.

**Inserindo Dados**

O MySQL permite a inserção de dados em uma tabela usando o comando **INSERT INTO**. Isso é fundamental para preencher a tabela com informações relevantes.

****

**Consultando Dados**

A instrução **SELECT** recupera dados de uma tabela. **SELECT \* FROM nome\_da\_tabela;** retorna todos os registros da tabela especificada.

****

**Atualizando Dados**

****Com o comando **UPDATE**, é possível modificar registros existentes em uma tabela. A cláusula **WHERE** define a condição para a atualização.

**Excluindo Dados**

O comando **DELETE** remove registros de uma tabela. Assim como o **UPDATE**, a cláusula **WHERE** é usada para identificar quais registros excluir.

****

**Índices**

Índices melhoram a eficiência das consultas. O comando C**REATE INDEX** cria um índice em uma coluna específica, acelerando a busca de dados.

****

**Consultas Avançadas**

**Cláusula WHERE**

A cláusula **WHERE** filtra os resultados da consulta com base em condições específicas. Exemplo: **SELECT \* FROM nome\_da\_tabela WHERE idade > 30;** retorna registros onde a idade é superior a 30.

****

**Cláusula ORDER BY**

A cláusula ORDER BY ordena os resultados da consulta. **SELECT \* FROM nome\_da\_tabela ORDER BY idade DESC;** exibe os resultados ordenados por idade de forma decrescente.

**Texto

Descrição gerada automaticamente com confiança média**

**Cláusula JOIN**

A cláusula JOIN combina dados de duas ou mais tabelas. Exemplo: **SELECT usuarios.nome, pedidos.produto FROM usuarios INNER JOIN pedidos ON usuarios.id = pedidos.usuario\_id;** recupera informações relacionadas de duas tabelas.

**Texto

Descrição gerada automaticamente**

**Funções de Agregação**

**COUNT**

****

**SUM**

****

**AVG**

****

Essas funções de agregação realizam cálculos em conjuntos de dados. **COUNT** conta o número de registros, **SUM** calcula a soma e **AVG** encontra a média.

**Stored Procedures**

**Criando Stored Procedures**

**Stored** procedures são conjuntos de instruções SQL que podem ser chamados pelo nome. O exemplo demonstra a criação de uma **stored procedure** para obter todos os usuários.

**Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente**

**Chamando Stored Procedures**

Stored procedures são invocadas com o comando CALL. No exemplo, CALL **obter\_usuarios();** executa a stored procedure criada.



**Segurança e Práticas Recomendadas**

**Usuários e Permissões**

Para garantir a segurança, criar usuários com senhas seguras e conceder permissões específicas é essencial. O exemplo ilustra a criação de um usuário e a concessão de privilégios.

**Tela preta com letras brancas

Descrição gerada automaticamente**

**Backup e Restauração**

**Texto

Descrição gerada automaticamente**Realizar backups regulares é uma prática recomendada. O exemplo mostra como criar um backup e restaurar um banco de dados a partir dele.

**Transações**

**Iniciando uma Transação**

Transações garantem a consistência dos dados. **START TRANSACTION;** inicia uma transação.

****

**Commit e Rollback**

**COMMIT;** efetiva as alterações em uma transação, enquanto **ROLLBACK;** desfaz as mudanças.

**Forma

Descrição gerada automaticamente com confiança média**

**MySQL e PHP**

**Conexão com PHP**

O PHP pode se conectar ao MySQL utilizando a classe mysqli. O exemplo ilustra a configuração de uma conexão básica.

**Texto

Descrição gerada automaticamente**

**Executando Consultas em PHP**

Com a conexão estabelecida, consultas podem ser executadas utilizando o método query. O resultado é então processado, e os dados podem ser manipulados em PHP.

**Texto

Descrição gerada automaticamente**

**Manipulação de Data e Hora**

**Trabalhando com Datas**

O MySQL oferece funções para manipulação de datas. Por exemplo, NOW() retorna a data e hora atuais, enquanto DATE\_FORMAT permite formatar datas de maneira específica.

Texto

Descrição gerada automaticamente

**Criando Triggers**

Triggers são conjuntos de instruções automáticas ativadas por eventos no banco de dados, como inserção, atualização ou exclusão de registros.

Texto

Descrição gerada automaticamente

**Views**

**Criando Views**

Views são consultas armazenadas como tabelas virtuais. Elas simplificam consultas complexas e podem ser usadas para restringir o acesso a determinadas informações.

Texto, Linha do tempo

Descrição gerada automaticamente

**Índices Avançados**

**Índices Compostos**

Índices compostos envolvem mais de uma coluna e são úteis para otimizar consultas que envolvem várias condições.



**Índices Full-Text**

Índices full-text aprimoram pesquisas de texto em grandes conjuntos de dados.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

**Replicação**

**Configuração de Replicação**

A replicação MySQL permite a criação de cópias idênticas de um banco de dados em servidores diferentes, garantindo redundância e disponibilidade.

Texto

Descrição gerada automaticamente

**Particionamento de Tabelas**

**Particionamento Horizontal**

O particionamento divide grandes tabelas em partes menores, facilitando a administração e otimizando consultas.

Texto

Descrição gerada automaticamente

**Funções JSON**

**Manipulação de Dados JSON**

O MySQL suporta funções para trabalhar com dados no formato JSON, facilitando a manipulação e extração de informações de documentos JSON.

Texto

Descrição gerada automaticamente

**Integridade Referencial**

**Chaves Estrangeiras**

As chaves estrangeiras mantêm a integridade referencial entre tabelas, garantindo que registros relacionados existam.

Texto

Descrição gerada automaticamente com confiança média

**MySQL Shell**

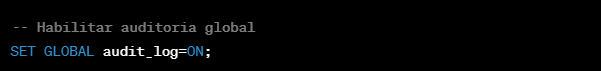
MySQL Shell é uma interface avançada para interagir com o MySQL. Além de SQL, oferece suporte a JavaScript e Python para automação e administração.



**Auditoria**

**Auditing**

O MySQL oferece recursos de auditoria para registrar atividades, como consultas executadas, proporcionando transparência e segurança.



**MySQL Workbench**

MySQL Workbench é uma ferramenta gráfica que simplifica o gerenciamento de bancos de dados MySQL, fornecendo uma interface intuitiva para desenvolvimento e administração.

**Sharding**

Sharding envolve a distribuição de dados em vários servidores para melhorar o desempenho e a escalabilidade.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente