



Colégio ULBRA São Lucas

27ª Coordenadoria Regional de Educação - Deliberação nº 451/2016 - CEE/RN
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL



APOSTILA 8

BANCO DE DADOS



CRÉDITOS:

Autora: Camille Ferreira | Capa: Camille e Gabriel Sandes | Revisão técnica: Prof. Jeferson Faleiro Leon e Prof. Éder Oliveira de Rosso. [Curso de Banco de Dados] [VC Ensinos]



SUMÁRIO

INTRODUÇÃO AO SQL	3
1. TIPOS DE DADOS MYSQL	3
RESTRIÇÕES DO MYSQL	4
1. RESTRIÇÕES DE INTEGRIDADE ESSENCIAIS NO SQL	4
TIPOS DE RESTRIÇÕES E MANIPULAÇÃO DE DADOS	5
COMO CRIAR UM BANCO DE DADOS	6
REFERÊNCIAS BIBLIOGRÁFICAS	8



INTRODUÇÃO AO SQL(Structured Query Language)

O MySQL é o banco de dados de código aberto mais conhecido no mundo. De acordo com o DB-Engines, o MySQL é o segundo banco de dados mais popular, ficando atrás do Oracle Database. O MySQL alimenta muitas das aplicações mais acessadas, como Facebook, Twitter, Netflix, Uber, Airbnb, Shopify e Booking.com.

Como o MySQL é de código aberto, ele inclui vários recursos desenvolvidos em estreita cooperação com os usuários há mais de 25 anos. Portanto, é muito provável que sua aplicação ou linguagem de programação favorita seja compatível com o MySQL Database.

1. TIPO DE DADOS MYSQL

- **INTEGER:** armazena números inteiros exemplo CPF integer.
- **NUMBER:** armazena números reais. Nesse tipo de dado é possível que seja definido a precisão e a escala do número exemplo salario number (15,2) onde a precisão seria esses 15 dígitos antes da vírgula e dois usado como escala (número representativo depois da vírgula).
- **DATE:** armazena datas. Nesse tipo de dado são armazenadas as datas como por exemplo dia mês e ano. A padronização de data no mysql é: ano/mês/dia
- **VARCHAR:** armazena apenas a quantidade de caracteres definidos na criação. Por exemplo, se definir uma coluna da tabela como VARCHAR(10) e armazenar um caractere, ele vai armazenar somente o caractere.
- **VARCHAR2:** é utilizado para armazenar variáveis de texto com tamanho variável. O texto informado no campo ocupa apenas o espaço do seu tamanho.



RESTRIÇÕES DO MYSQL

Considere um cenário em que temos uma tabela que armazena informações do usuário para um aplicativo da Web. Alguns dados, como a idade do usuário, podem ser opcionais porque não impedem que os usuários acessem nosso aplicativo. No entanto, ter uma senha para cada usuário é essencial para fins de login. Para resolver isso, implementamos uma restrição de integridade na coluna de senha da tabela do nosso usuário para garantir que cada entrada nessa tabela inclua uma senha.

Em essência, as restrições de integridade são cruciais para:

- Evitar a falta de dados.
- Garantir que todos os dados estejam de acordo com os tipos e intervalos de valores esperados.
- Manter links adequados entre os dados de diferentes tabelas.

1. Restrições de integridade essenciais no SQL:

- **PRIMARY KEY:** Identifica de forma exclusiva cada registro em uma tabela.
- **NOT NULL:** Garante que uma coluna não pode conter valores NULL.
- **UNIQUE:** Garante que todos os valores em uma coluna ou grupo de colunas sejam exclusivos.
- **DEFAULT:** Fornece um valor padrão para uma coluna quando nenhum é especificado.
- **CHECK:** Garante que todos os valores em uma coluna satisfaçam uma condição específica.
- **FOREIGN KEY:** Estabelece relações entre tabelas ao fazer referência a uma chave primária em outra tabela.



TIPOS DE RESTRIÇÕES E MANIPULAÇÃO DE DADOS

A linguagem de definição e manipulação de dados no MySQL é a linguagem de consulta estruturada (SQL) e a linguagem de manipulação de dados (DML):

SQL: A linguagem de consulta estruturada (SQL) é usada para definir, consultar, controlar e manipular dados.

DML: A linguagem de manipulação de dados (DML) é um conjunto de comandos SQL usados para manipular dados em tabelas de banco de dados. A DML é responsável por adicionar, editar ou excluir dados de um banco de dados.

Para manipular dados no MySQL, é importante:

Criar o banco de dados e as suas tabelas. Aprender a visualizar o banco de forma gráfica através do MySQL Workbench Alguns comandos DML são:

- **INSERT:** Adiciona uma nova linha à tabela.
- **UPDATE:** Edita o registro da conta bancária.
- **DELETE:** Apaga a entrada no banco de dados.

O MySQL é um Sistema Gerenciador de Banco de Dados (SGBD) relacional que é multiusuário e multi tarefas. É conhecido por ser fácil de usar e compatível com a maioria dos sistemas operacionais.



COMO CRIAR UM BANCO DE DADOS

10 passos para criar um banco de dados MySQL

Criar um banco de dados é uma tarefa fundamental para quem trabalha com tecnologia da informação, e o MySQL é um dos mais populares do mercado. Se você está começando a trabalhar com esse RDBMS, ou deseja aprimorar suas habilidades, confira esses 10 passos fáceis.

Passo 1: Instale o MySQL

Antes de criar um banco de dados, você precisa ter a ferramenta instalada no seu computador. A maneira mais fácil de fazer isso é baixar a versão mais recente no site oficial, que oferece diversas opções de download, dependendo do seu sistema operacional.

Passo 2: Abra o prompt de comando do MySQL

Após instalar, abra o prompt de comando. Se você estiver usando Windows, vá até a barra de pesquisa e digite "CMD" e clique no prompt de comando. Se você estiver usando o Linux, abra o terminal.

Passo 3: Faça login no MySQL

Digite o comando "mysql -u root -p" no prompt de comando e pressione enter. Isso irá solicitar que você digite a senha do seu usuário "root". Digite a senha e pressione novamente. Agora você está conectado ao servidor.

Passo 4: Crie um banco de dados

Digite o comando "create database nome_do_banco_de_dados;" e pressione enter. Substitua "nome_do_banco_de_dados" pelo nome que você deseja dar ao seu banco de dados.

Passo 5: Selecione o banco de dados

Digite o comando "use nome_do_banco_de_dados;" e pressione enter. Isso irá selecionar o seu banco de dados recém-criado.

Passo 6: Crie uma tabela

Digite o comando "create table nome_da_tabela (coluna1 tipo_de_dado, coluna2 tipo_de_dado, coluna3 tipo_de_dado);" e pressione enter. Substitua "nome_da_tabela" pelo nome que você deseja dar à sua tabela. Defina o tipo de dado para cada coluna, por exemplo, "int" para números inteiros, "varchar" para strings, "date" para datas, entre outros.



Passo 7: Insira dados na tabela

Digite o comando "insert into nome_da_tabela (coluna1, coluna2, coluna3) values (valor1, valor2, valor3);" e pressione enter. Substitua "nome_da_tabela" pelo nome da tabela que você criou. Insira os valores que você deseja na tabela.

Passo 8: Visualize os dados da tabela

Digite o comando "select * from nome_da_tabela;" e pressione enter. Isso irá exibir todos os dados armazenados na tabela que você criou.

Passo 9: Atualize os dados da tabela

Digite o comando "update nome_da_tabela set coluna1 = novo_valor where condição;" e pressione enter. Substitua "nome_da_tabela" pelo nome da tabela que você criou. Atualize o valor de uma determinada coluna com um novo valor, com base em uma condição específica.

Passo 10: Exclua dados da tabela

Eventualmente, pode ser necessário excluir dados específicos de uma tabela. Para fazer isso, você pode usar a cláusula DELETE. É possível filtrar os dados com base em diferentes critérios, assim se torna mais fácil manter o ambiente "clean".



REFERÊNCIAS BIBLIOGRÁFICAS

BLOG DO SQL. **Entendendo Constraints em Bancos de Dados SQL**. Disponível em: <https://blogdosql.con.br>. Acesso em: 27 nov. 2024.

ESCOLA DNC. **Guia Rápido para Criar um Banco de Dados MySQL**. Disponível em: <https://www.escoladnc.com.br>. Acesso em: 29 nov. 2024.

MYSQL. **MySQL Documentation**. Disponível em: <https://dev.mysql.com.br/doc/>. Acesso em: 2 dez. 2024.

MYSQL. MySQL Workbench Manual. Disponível em: <https://dev.mysql.com/doc/workbench/en/>. Acesso em: 2 dez. 2024.

SILVA, João. **Guia Prático: Criando Banco de Dados no MySQL**. Disponível em: <https://exemplo.com/guia-mysql>. Acesso em: 4 dez. 2024.