

Índice

MySQL	1
Criação de uma database	1
Execução	2
Comandos	2
Tipos de dados	4
Schemas	5
Links e afins	5
Dúvidas	5

Assuntos: DDL com MySQL.

Abertura Alunos:

Reflexão diária: Uso responsável do ChatGPT
Yuri Henrick e Daniel Lugli.

MySQL

Clique no título para ser direcionado as anotações anteriores de SQL

MySQL é um SGBD, ou seja, ele auxilia na criação e gerenciamento de um banco de dados.

Crie uma nova conexão.

Use as configurações padrão.

ANOTE BEM A SENHA, NÃO PODE SER MUDADA!!!!

A Guia Query do MySQL Workbench possui duas grande áreas:

A primeira é o Editor, onde você irá escrever as instruções SQL;

A segunda é o Output, onde você visualiza o resultado da sua instrução, ou seja, se deu certo ou errado.

Organização da SQL

- **DQL** - Linguagem de Consulta de Dados
 - SELECT
- **DML** - Linguagem de Manipulação de Dados
 - INSERT, UPDATE e DELETE
- **DDL** - Linguagem de Definição de Dados
 - CREATE, ALTER e DROP
- **DCL** - Linguagem de Controle de Dados
 - GRANT e REVOKE
- **DTL** - Linguagem de Transação de Dados
 - BEGIN, COMMIT e ROLLBACK

Criação de uma database

Para criar o database use o comando **"CREATE DATABASE"** com o nome a seguir.

Por boa prática deixar o comando em letra maiúscula e o nome em minúscula.

Por boa prática os nomes costumam ser **"db_nome"**.

Fica **CREATE DATABASE db_nome;**

MYSQL não é case sensitive para comando.

É boa prática identificar tabelas com **"tb_nome"**;

Execução



1. Executa a linha de comando selecionada.
2. Executa todos os comandos no cursor.
3. Executa o comando explain no cursor.

PS: Mantenha a Atenção aos Detalhes ao utilizar o comando Executar. O SQL não funciona da mesma forma que as Linguagens de Programação tradicionais. Cada instrução é única e o seu final é determinado pelo ponto e virgula (;). Na prática funciona como se fossem comandos do Terminal: Cada instrução faz uma ação e uma instrução nem sempre dependerá das instruções anteriores.

Comandos

CREATE DATABASE db_nome;

Comando para se criar uma database, semelhante a quando a gente nomeava a variável usamos agora Coluna e Campo.

USE db_nome;

USE é usado para indicar o database a ser usado, precisa indicar para o SQL para ele entender qual database usar na parte editar.

Shift + Home

Atalho para selecionar a linha inteira.

CREATE TABLE tb_nome ();

Comando para criar uma tabela. Dentro dos parênteses é que identificamos os atributos da tabela.

nome TIPOVARIÁVEL,

Sintaxe para definição de variável. Atenção para vírgula para separação de variáveis.

Auto_increment

Cláusula para criar um novo incremento sem repetir o número, usado logo após a primeira declaração de argumento.

NOT NULL

Cláusula para não permitir que um campo seja nulo, ou seja, não pode ficar vazio.

SELECT

Comando usado para extrair informação da database.

UPDATE

Comando para a atualização de informações na database.

DELETE

Comando para deletar a informação da database.

INSERT INTO tb_nome ()

Comando para inserir informações novas na database.

Ex: **INSERT INTO** tb_produtos(nome, quantidade, datavalidade, preco)
VALUES ("tomate",100, "2023-12-15", 8.00);

VALUES ()

Comando para se definir os valores dentro de uma tabela, geralmente se usa depois do **INSERT INTO**. É como implementar um método.

ALTER DATABASE

Comando para modificar a database.

ALTER TABLE

Comando para modificar a tabela.

DROP TABLE

Comando para deletar a tabela

CREATE INDEX

Cria o index (chave de pesquisa)

DROP INDEX

Deleta a index

Edit > Format > UPPERCASE Keywords

Atalho para formatar as palavras chave para maiúsculo de uma vez

PRIMARY KEY (nome);

Comando usado para indicar a chave primária, geralmente usamos ID pois não se repete e usamos o comando auto_Increment

EDIT > PREFERENCES > SQL EDITOR > QUERY EDITOR

Configuração para as palavras chave ficarem maiúsculas de forma automática.

SELECT * FROM tb__nomedatabela;

Comando que verifica as informações que constam na tabela, * significa que estamos selecionando tudo.

SELECT campo1 , campo2 FROM tb__nomedatabela;

Comando que verifica as informações campos específicos que constam na tabela.

SELECT campo1 , campo2 FROM tb__nomedatabela WHERE campo condição (condições)

Comando para verificar informações de campos e linhas específicas em uma tabela

DELETE FROM tb__ nome WHERE campo condição

Comando para deletar informações em um campo específico.

AND

É um conector de condições do comando

Ex (SELECT campo1 , campo2 FROM tb_nomedatabela WHERE campo
condição AND condição)

ALTER TABLE tb__tabela MODIFY campo VARIÁVEL(alteração);

Comando para alterar a estrutura de um campo

Ex: **ALTER TABLE** tb_tabela **MODIFY** preco **DECIMAL**(6,2);

ADD

Cláusula para se adicionar campos em uma tabela.

Ex: **ALTER TABLE** tb_tabela **ADD** descricao **VARCHAR**(255) **NOT NULL**;

SET SQL_SAFE_UPDATES = 0;

Comando exclusivo para o SQL, que desabilita a medida de segurança que previne atualizações dentro do database sem o comando WHERE que seria uma alteração na tabela inteira.

SET

Comando que indica que o programa deve inserir um valor.

UPDATE

Comando para se atualizar um valor em uma tabela

Ex: **UPDATE** tb_tabela **SET** preco = 5.99 **WHERE** id = 5;

UPDATE define que se vai fazer um update, o SET define o que será feito, nesse caso colocar um valor, e o WHERE indica em qual index, o id seria nossa chave primária

ALTER TABLE tb__tabela DROP COLUMN coluna

Comando para remover uma coluna de uma tabela.

FOREIGN KEY (nomechave) REFERENCES tb__tabela (chaveprimaria)

Comando para associar os valores de uma chave estrangeira a uma chave primária.

nomecoluna LIKE '%palavra%'

Comando para se pesquisar em uma cadeia de caracteres. É possível pesquisar sem colocar a palavra completa, é como um ctrl+F.

Tipos de dados

Dado SQL	Dado Java	Função
VARCHAR(n)	String	Cadeia de caracteres
CHAR(n)	char	Caractere
BLOB(n)	<i>null</i>	Armazena campos do tipo BLOB
DATE	<i>null</i>	Data
DATETIME	<i>null</i>	Data e hora
TIMESTAMP	<i>null</i>	Armazena data e hora no formato Timestamp
INT	int	Número inteiros
DECIMAL	<i>null</i>	Decimais
FLOAT	float	Números reais
DOUBLE	double	Números reais com mais casas decimais
DECIMAL	<i>null</i>	Decimal alterável padrão (10,0)
BOOLEAN	boolean	Booleano, definido por valores de verdadeiro ou falso
Bit	<i>null</i>	Valores pequenos, podendo ser 0 e 1.

BLOB: Armazena objetos binários grandes como imagens, por exemplo.
Armazena até 65.535 bytes de dados

Timestamp: Os valores TIMESTAMP são armazenados como o número de segundos desde a época Unix ('1970-01-01 00:00:00' UTC). Formato: AAAA-MM-DD hh:mm:ss.

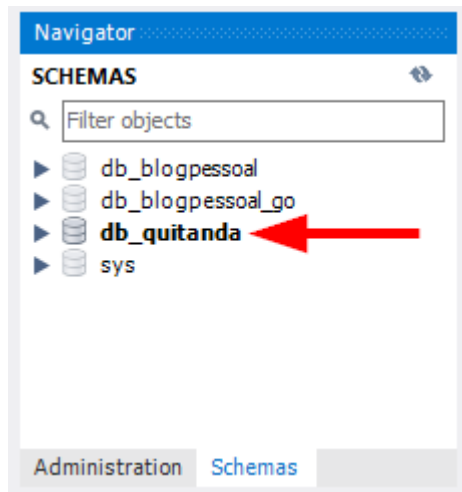
O intervalo suportado é de '1970-01-01 00:00:01' UTC a '2038-01-09 03:14:07' UTC.

A inicialização e atualização automáticas para a data e hora atuais podem ser especificadas usando DEFAULT CURRENT_TIMESTAMP e ON UPDATE CURRENT_TIMESTAMP na definição da coluna.

Schemas

Os Schemas são uma coleção de objetos dentro de um determinado database (banco de dados), servem para agrupar objetos no nível de aplicação como também para simplesmente fazer divisões departamentais, ou seja, onde fica a sua biblioteca de de databases.

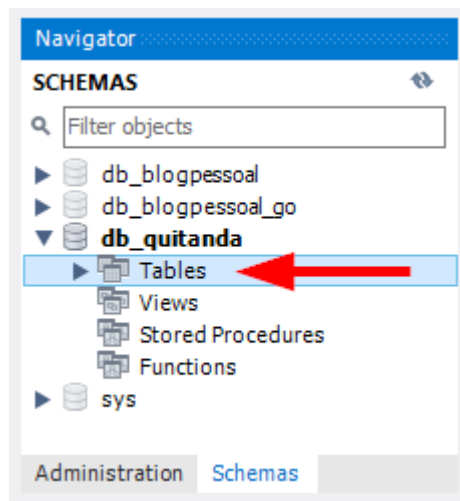
Geralmente localizado do lado esquerdo:



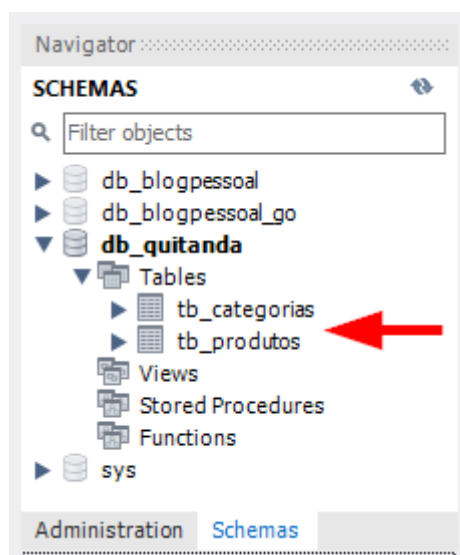
É uma boa prática sempre atualizar após criar sua nova database



É onde também você encontra tudo que você tem dentro da sua database:



Usando os dropdowns você acessa os detalhes como as tabelas:



Links e afins

CookBook mySQL, MD02 :

https://github.com/conteudoGeneration/cookbook_java_fullstack/blob/main/03_mysql/02.md

Tutorial SQL indicado no CookBook:

<https://www.w3schools.com/sql/default.asp>

Dúvidas