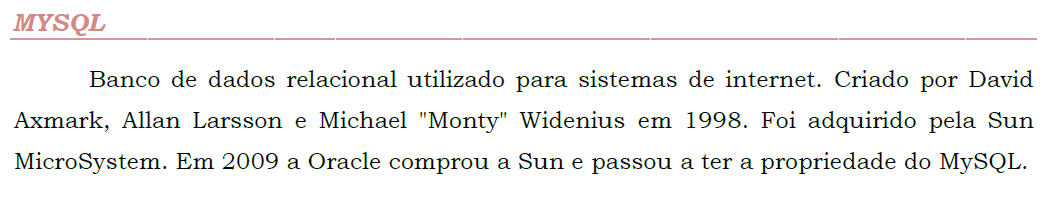
Aula 2

Iniciando no Mundo do MySQL





<https://www.youtube.com/watch?v=5JbAOWJbgIA&list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r&index=3&ab_channel=CursoemV%C3%ADdeo>

História do MySQL: (minute 1:19 até 5:37)

**SQL**

***SQL*** *=* ***S****tructure* ***Q****uery* ***L****anguage*

O SQL é uma linguagem de **Consulta Estruturada** que surgiu dentro dos laborátorios de pesquisa da IBM na década de 70.



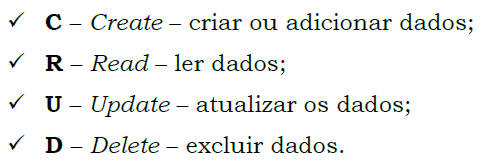
A linguagem SQL foi declarada um padrão pelo ANSI (American National Standarts Institute) e pela ISO (International Organization for Standardizion) com diversos **comandos padrões** lançados desde seu desenvolvimento.



O conceito de SQL basicamente está divido em duas partes:

1. Comandos que **definem** e **manipulam** a **estrutura** **de** **armazenamento** e **procedimentos**;
2. Comandos que **manipulam dados**.

As operações básicas de Banco de Dados podem ser memorizadas como: CRUD



<https://www.youtube.com/watch?v=5JbAOWJbgIA&list=PLHz_AreHm4dkBs-795Dsgvau_ekxg8g1r&index=3&ab_channel=CursoemV%C3%ADdeo>

Padrões de uma Base de Dados: (minute 6:02 até 9:08)

**Comandos SQL (Parte I):**

Lembram dos padrões que a gente falou na ‘’Aula 1’’ de Banco de Dados?

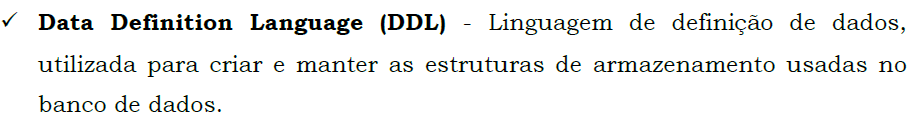
1. Base de Dados;
2. Sistema Gerenciador de Banco de Dados;
3. Linguagem de Exploração;
4. Programas Adicionais.

Então, o MySQL tem tudo isso e muito mais.

Por padrão, o MySQL possui dentro dele algums **comandos padrões**, como por exemplo:

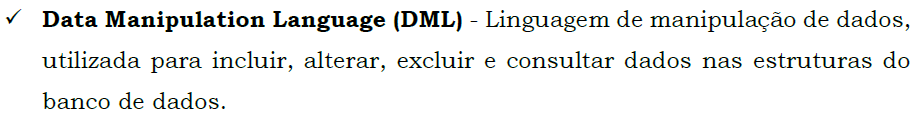


Os comandos SQL são divididos em:



**Resumo:**

Ou seja, você vai criar uma tabela, alterar esse Banco de Dados. Ou seja, qualquer comando de definição da estrutura da base de dados é mantida, dentro do MySQL, pela porção DDL.



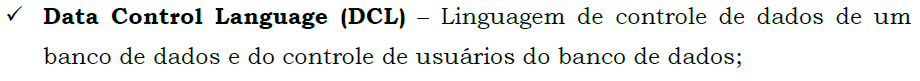
**Resumo:**

Ou seja, como vai poder incluir novos dados, excluir dados, manipular dados de qualquer forma, alterar a composição dos dados e etc.



**Resumo:**

Ou seja, uma linguagem de solicitações, ou você pode fazer o SELECT e de uma parte dos dados, filtrar aquilo que se busca.



**Resumo:**

Ou seja, como essa é uma linguagem de controle, aqui a gente pode definir quais usuários podem acessar o nosso Banco de Dados, qual tipo de acesso ele vai poder fazer, que tipo de comandos ele vai poder executar e muito mais.



**Resumo:**

Ou seja, qualquer solicitação que possa ser feita a um Banco de Dados, e ele vai te atender da melhor maneira possível seguindo os quatro princípios que chamamos de **DICA**.

**D**urabilidade

**I**solamento

**C**onsistência

**A**tomicidade

Um breve resumo sobre cada um deles...

1. Durabilidade: Todo dado que é colocado, alterado, ou manipulado tem que permanecer durável. Ou seja, ele tem que permancer dessa maneira, enquanto eu quiser que esse dado esteja lá.
2. Isolamento: Se eu tenho duas transações feitas ao mesmo tempo, elas devem ser executadas sem uma uma interferir na outra. Ou seja, elas são isoladas.
3. Consistência: Toda a transação tem que levar o Banco de Dados de um estado consistente à outro consistente. Ou seja, se tudo estava ‘’OK’’ antes, tudo deve continuar ‘’OK’’ depois.
4. Atomicidade: Toda transação tem que ser atômica Ou seja, ou TODA elas acontece, ou NADA acontece. Ou TUDO dá certo, ou ele dá um ‘’ctrl+z’’ para um estado anteriormente consistente.

**[Exercício 1]** Escreva em uma lista as seguintes características sobre 3 pessoas, preferencialmente que cada característica, entre cada uma das pessoas, seja diferente.

Dados:

1. Nome;
2. Idade;
3. Sexo;
4. Peso;
5. Altura;
6. Nacionalidade.



Nome = Shiryu de Dragão

Idade = 14

Sexo = Masculino

Peso = 53 Kg

Altura = 1,72 m

Nacionalidade = Japão



Nome = Aldebaran de Touro

Idade = 20

Sexo = Masculino

Peso = 130 Kg

Altura = 2,10 m

Nacionalidade = Brasil



Nome = Hyoga de Cisnei

Idade = 14

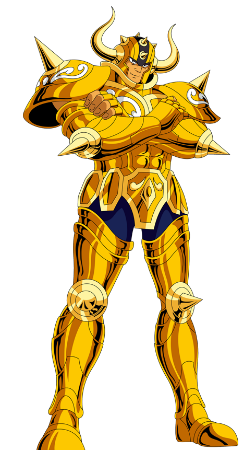
Sexo = Masculino

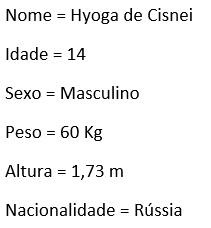
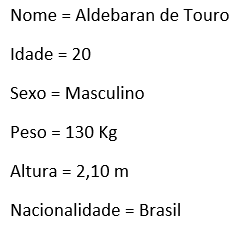
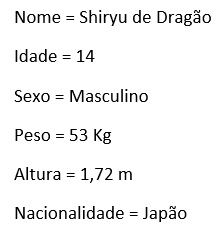
Peso = 60 Kg

Altura = 1,73 m

Nacionalidade = Rússia

Todo so três tem caracteristicas semelhantes!





Porém, cada um deles as suas próprias características.

E isso os faz instâncias, ou seja, pessoas diferentes uma das outras.

E esse é o objetivo do banco de dados:

“Registrar instâncias separadas de ‘coisas’ que tem características semelhantes!”

Ou seja, ultilizando essa padrão, eu posso cadastrar qualquer tipo pessoa, visto que todas elas vão ter as mesmas características.

Caracteristicas iguais.

Valores diferentes.

1. Pegar as pessoas e colocar num container;
2. Escrever no container do lado de fora “pessoas”:

Todas vez que eu tiver uma pessoas e quiser guardar essa pessoas, eu vou colocar elas no container escrito “pessoa”.

E todas as instâncias que estiver nesse container, vão ter essas características. Algumas podem ser opcionais, mas existe a possibilidade de cadastro dessa nacionalidade.

E eu posso “criar” containers diferentes para coisas diferentes, como, por exemplos, jogos, visto que jogos possuem caracteristicas diferentes e por isso precisam estar em containers diferentes.

1. Agora eu vou pegar esses container e colocar em um navio.

No mundo do Banco de Dados isso tudo tem um nome:

1. O navio = Banco de Dados:

Então o Banco de Dados são coleções de dados que são de características separadas, mas que estão organizados em locais específicos (container).

1. Locais específicos = Tabelas:

Tabelas guardam dados de coisas que têm características semelhantes. E se você quiser, você pode ter várias tabelas dentro de um Banco de Dados. E os dados dentro das tabelas também têm um nome específico.

1. Dados das tabelas = registros:

Aonde ficam cada “coisa” dentro daquela tabela. “coisa” pode ser pessoas, ou jogo, ou whatever you want.



Então, pra começar, a primeira coisa que eu tenho que criar é o meu banco de dados.

E como se cria um Bando de Dados usando o MySQL?

Muito simple!

Se fosse em português, como seria um comando?



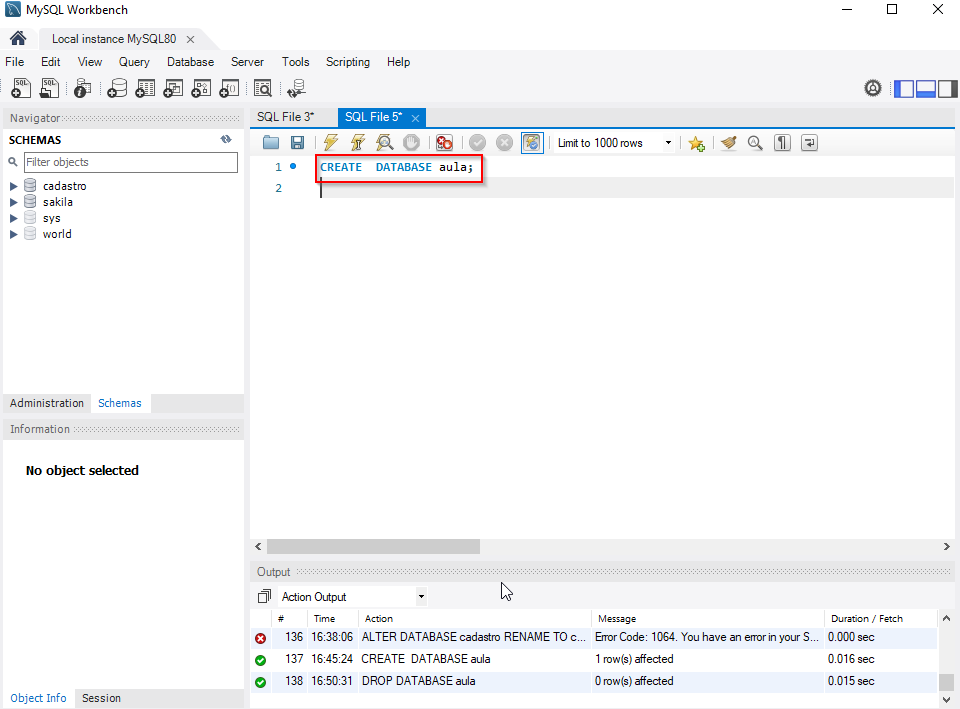
Então em inglês será...



Comandos:

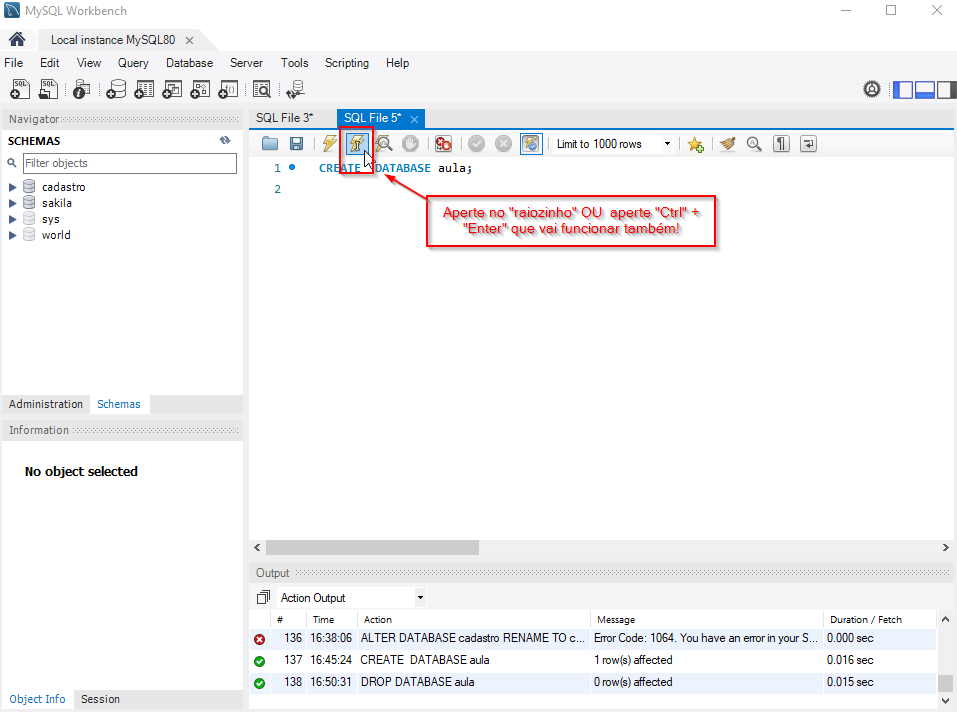
1. Primeiro passo:

CREATE DATABASE cadastro;



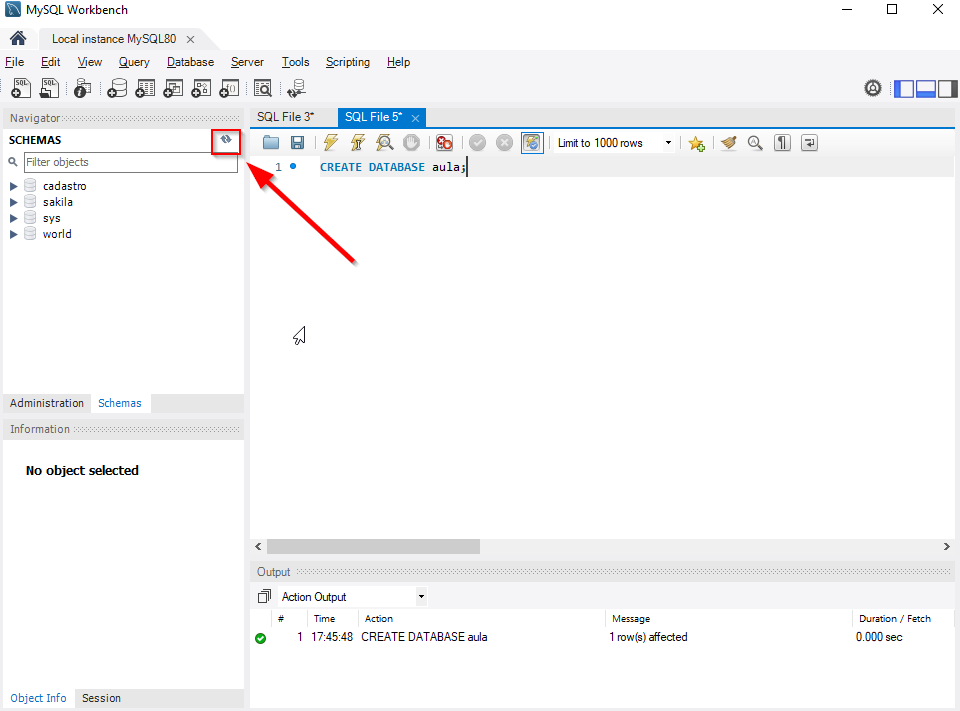
1. Segundo passo:

Precione o raiozinho;



1. Terceiro passo:

Aperte em “Refresh” para que o Banco de Dados seja visto;



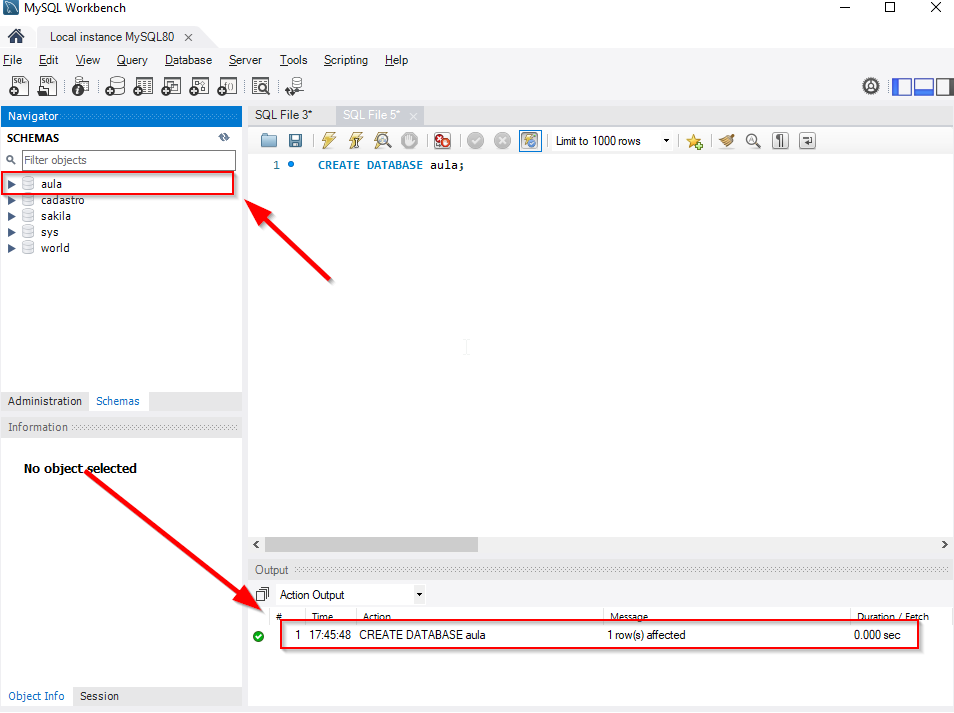
1. OBSERVE 1:

4.1) O Banco de dados “aula” foi criado;

4.2) “1 row(s) affected” = 1 LINHA afetada:

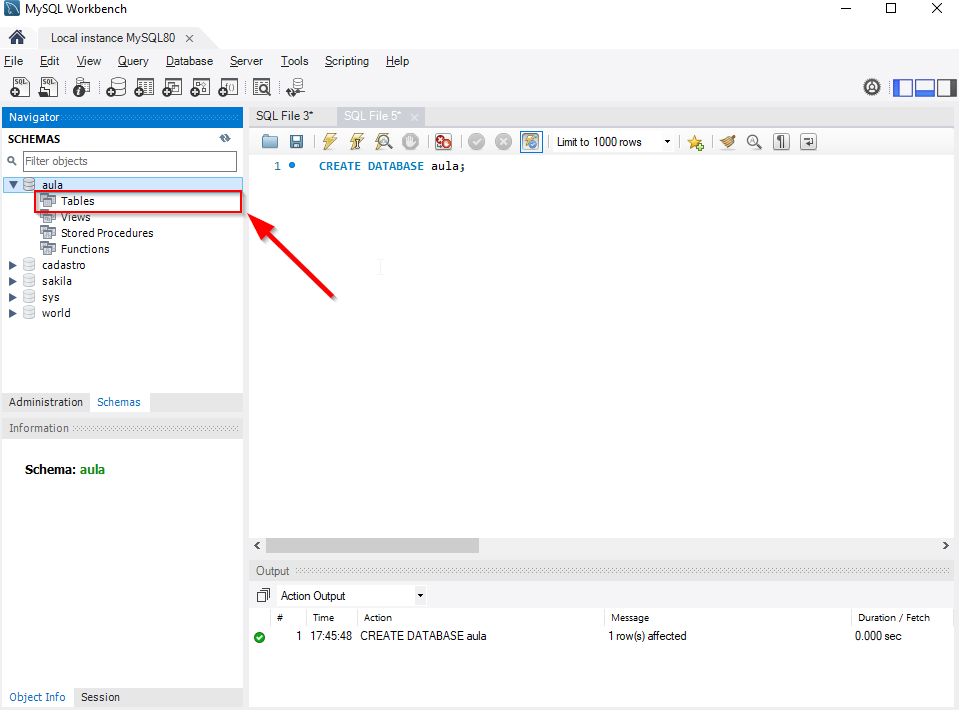
Ou seja, eu não tinha um Banco de Dados.

Quando eu creie o Banco de Dados, eu afetei uma linha.



1. OBSERVE 2:

O nosso Banco de Dados também tem “Tables”.



Beleza, agora a gente já criou o nosso Banco de Dados!

~~O nosso “Navio”.~~

Então o nosso próximo passo vai ser criar as tabelas!

RELEMBRANDO O QUE A GENTE SABE SORBE AS TABELAS:



Resumo:

(Campos) ⊂ (Registros) ⊂ (Tabelas) ⊂ (Banco de Dados)

Então agora, vamos criar a nossa tabela:

Se fosse em português, como seria um comando?

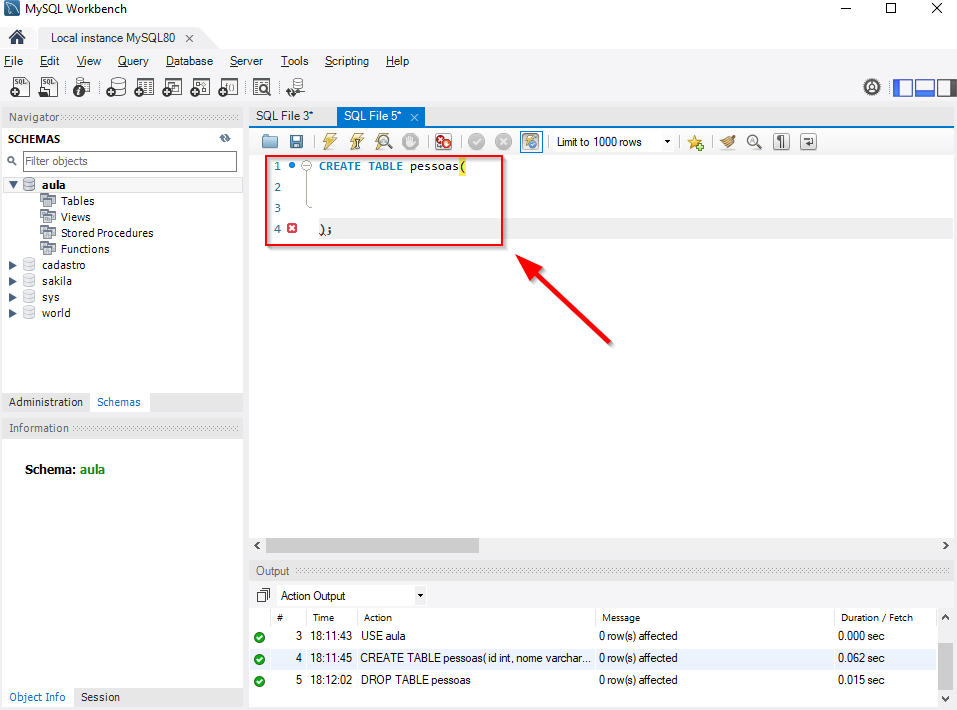


Então em inglês será...



Comandos:

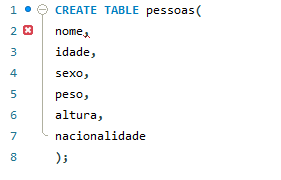
1. Primeiro passo:



1. Segundo passo:

Preencher os campos que a gente quer.

#Observe que o último campo, não precisa de vírgula!



Agora eu preciso informar ao programa qual é o tipo de cada um desses campos!

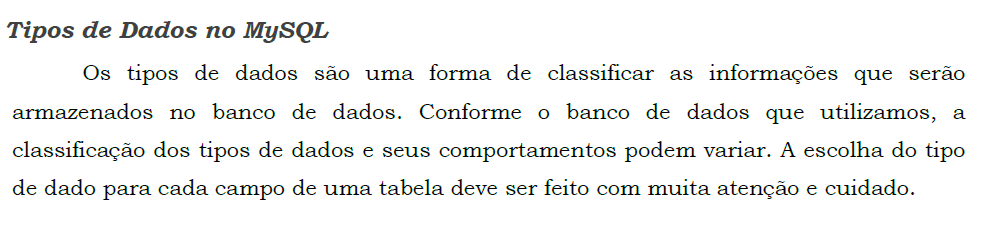
O tipo é o que nós chamamos de **tipos primitivos**.

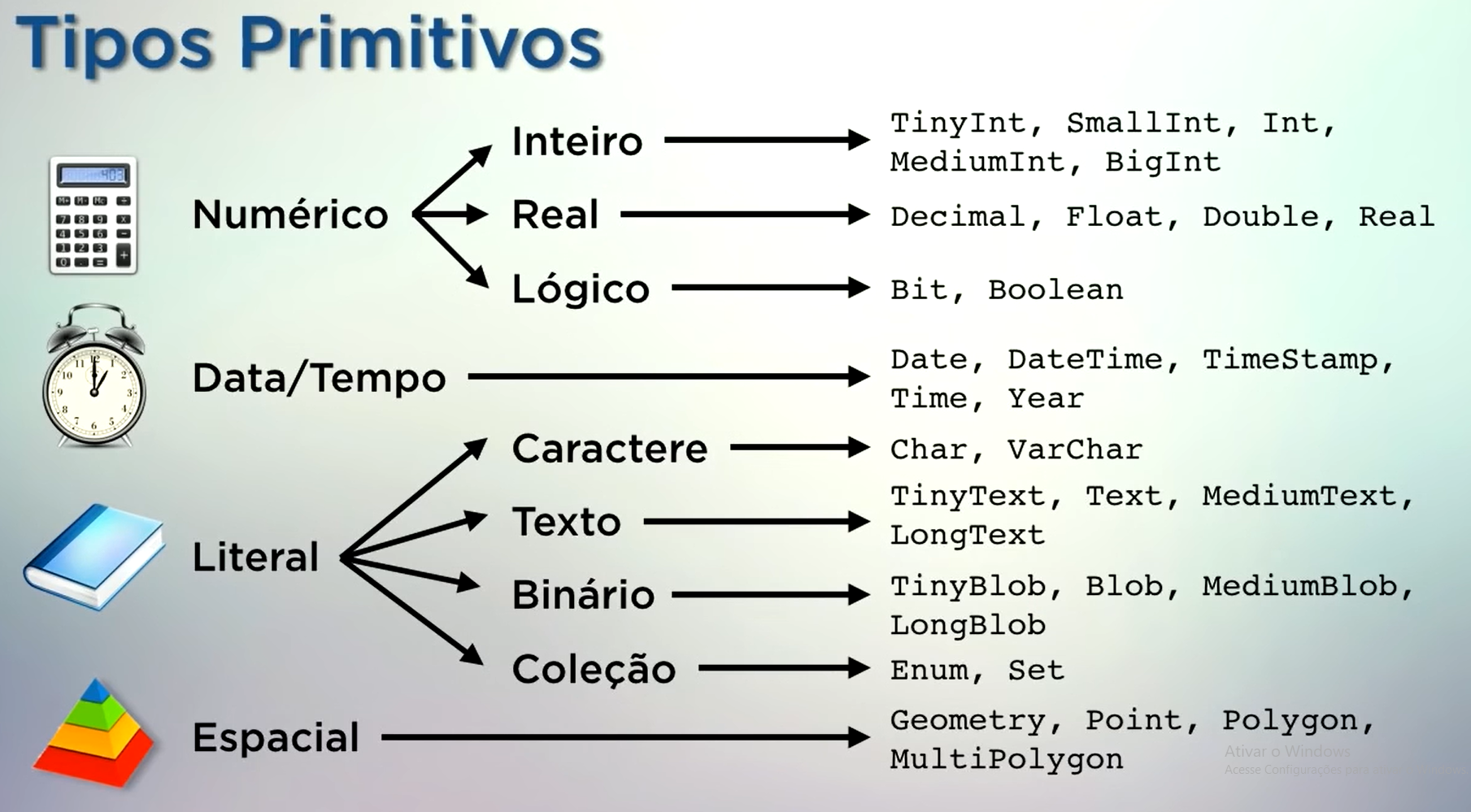
Cada linguagem de programação tem a usar. Para SQL não seria diferente.

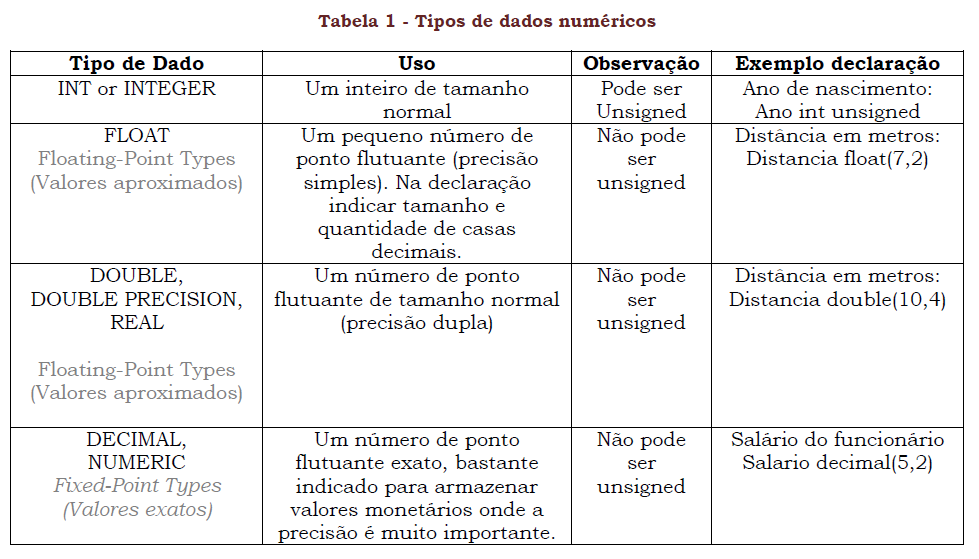
~~(Teoria de simulacros e simulações)~~

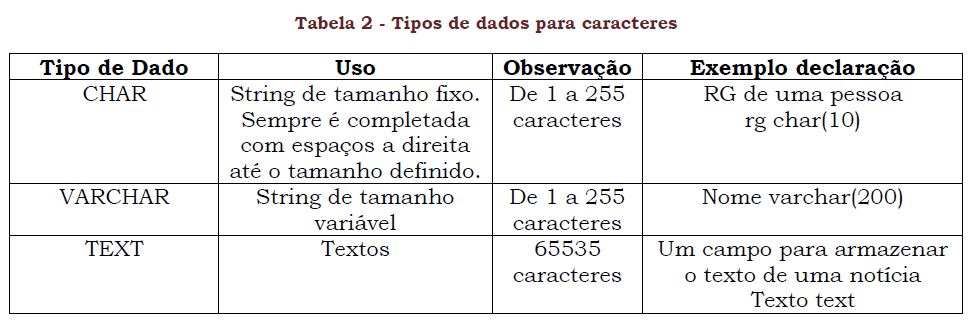
Em SQL são quatro famílias de tipos primitivos: **Numérico, Data/Tempo, Literal e Espaciais.**

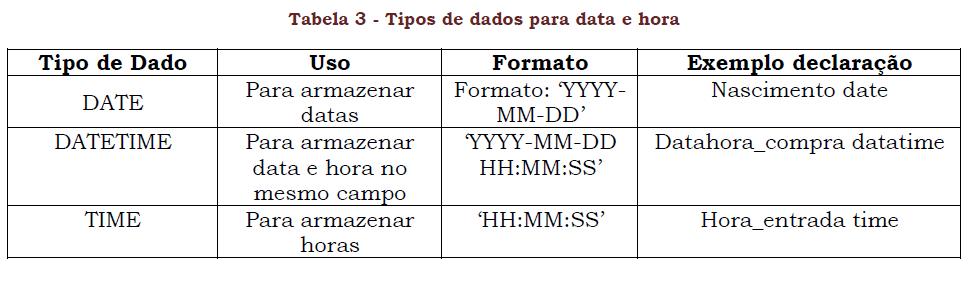
E cada família tem sua **subdivisão** e os seus **subtipos** ainda.







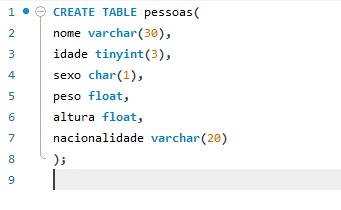




1. Terceiro passo:

Agora sim! Colocar o tipo primitivo de cada variável.

#Curiosidade: tinyint(3) não precisa escrever o 3 ali, visto que ele ocupa 3 bytes. O tupi int ocupa 11 bytes, por exemplo.



**[Pergunta]** Quantos comandos existem na tela?

Apenas um!

1. Quarto passo:

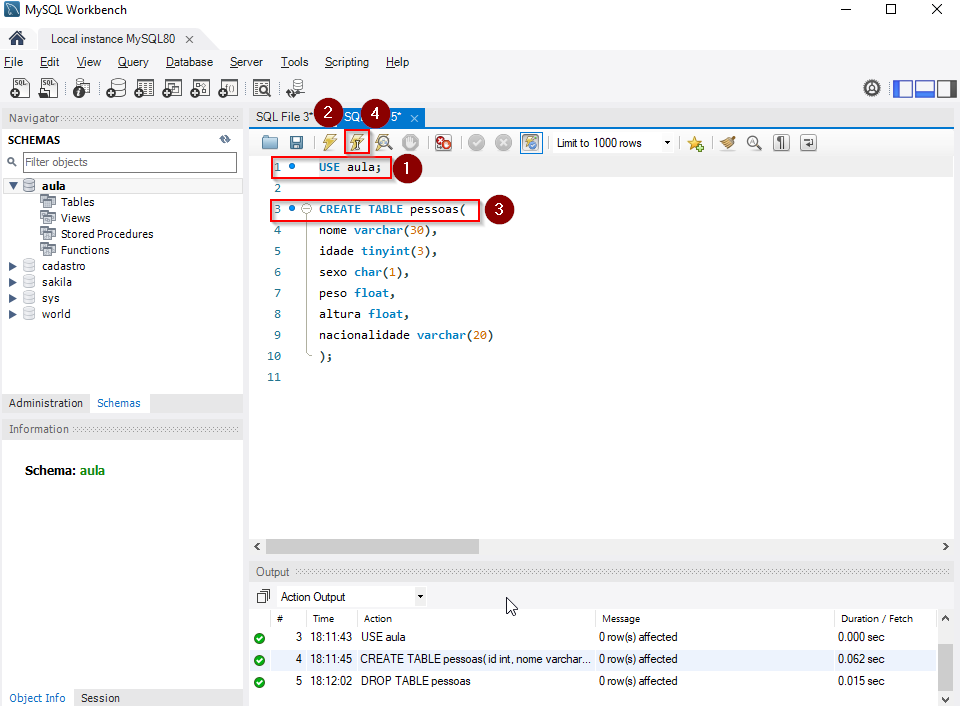
Execute o comando! (Ctrl + Enter)

Deu ERRO!? Calma...

Primeiro selecione o schema “aula” ou então digite:

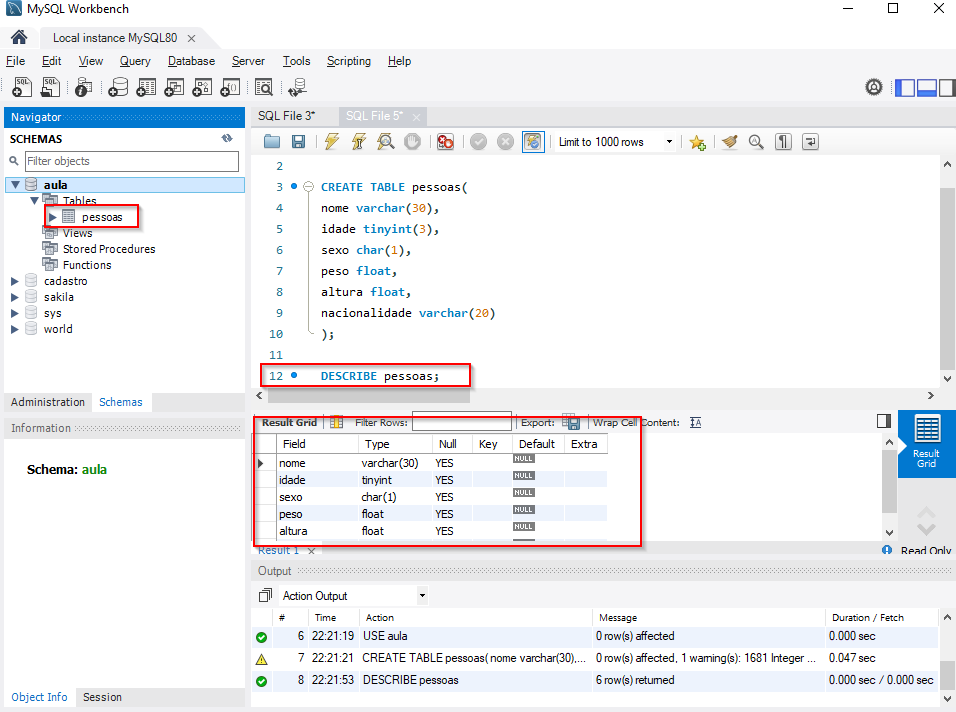


Depois execute o código novamente.



Você quer ver como funciona, mais ou menos, a estrutura interna de uma tabela?

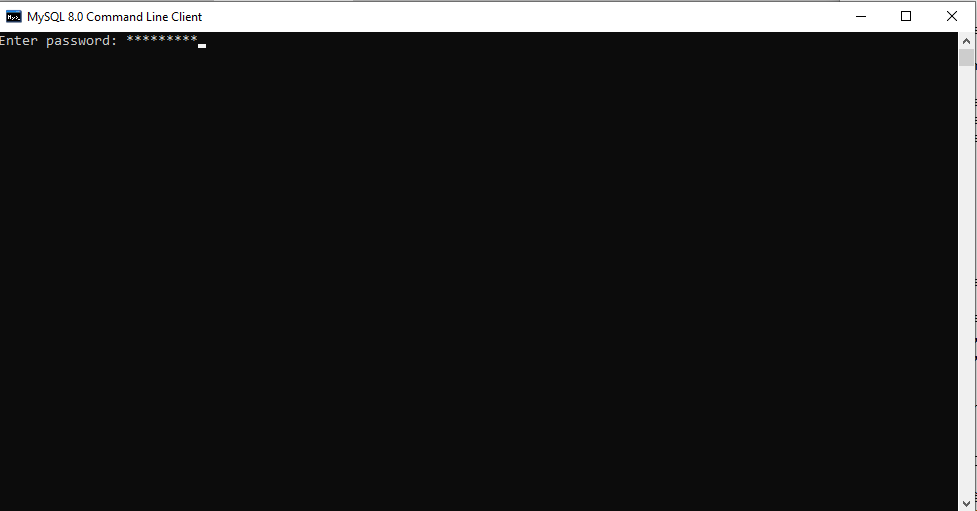




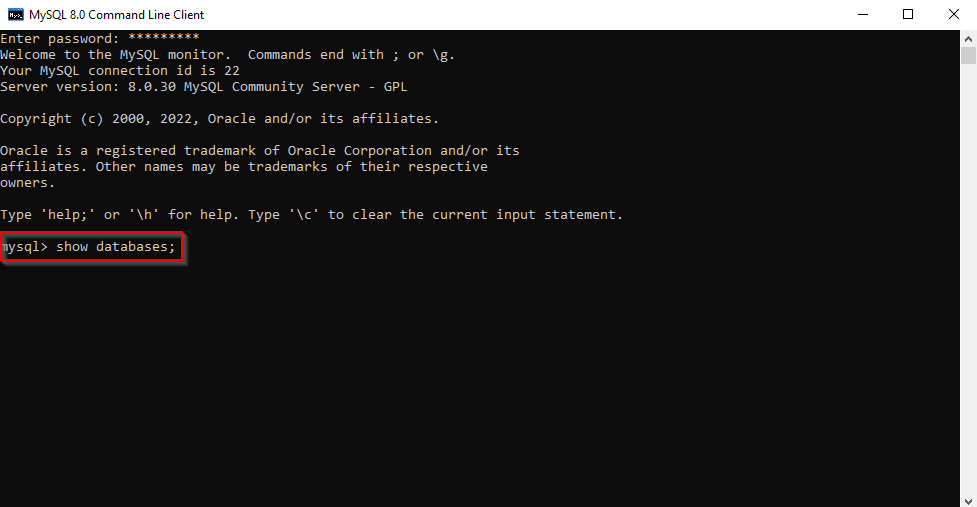
Agora façamos tudo isso atravéz do prompt de comandos:

1. Primeiro passo:

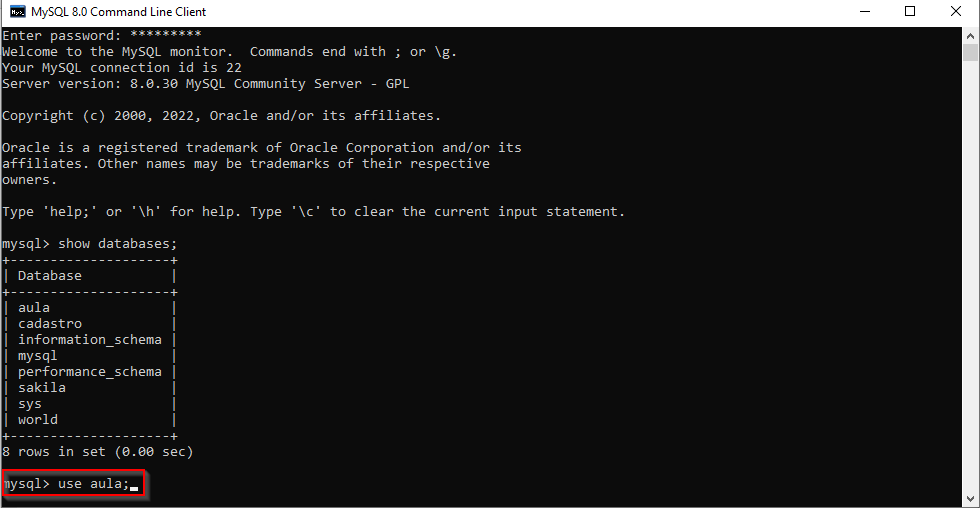
Abra o “Command Line Client” e digite a sua senha.



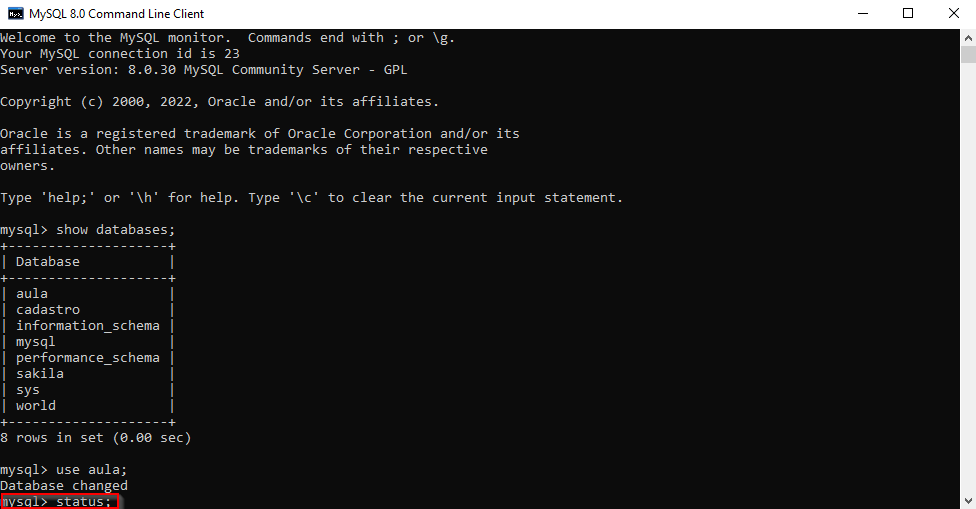
1. Segundo passo:



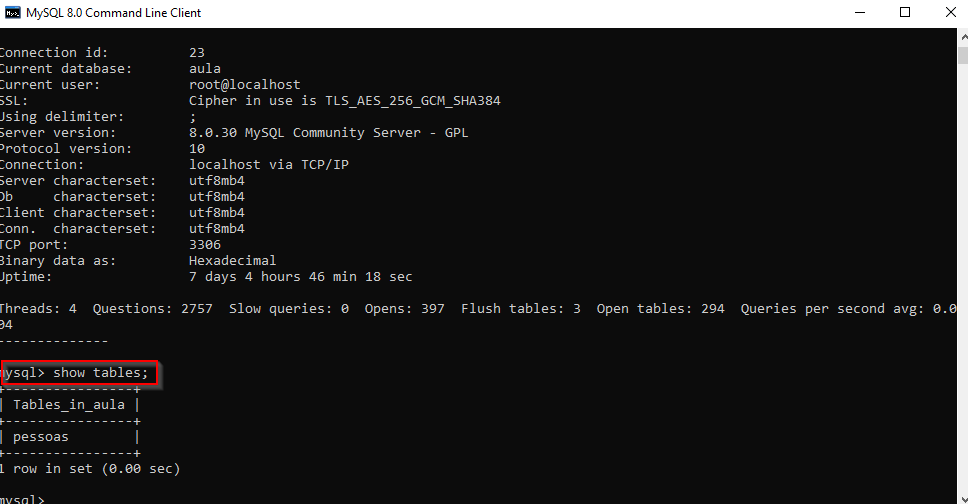
1. Terceiro passo:



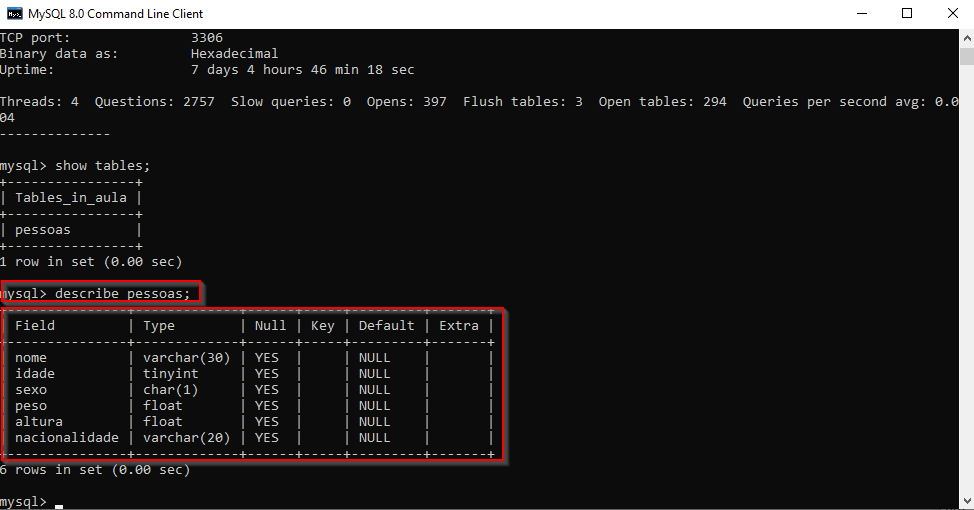
1. Quarto passo:



1. Quinto passo:



1. Sexto passo:



1. Setimo passo: “exit;”

Agora, para cadastrar novos registros, a gente vai ver um outro commando que é o INSERT INTO na próxima aula.

Porém até aqui nos temos um problema!

A gente pode cadastrar duas vezes a mesma pessoa. E esse problema a gente vai aprender como resolver na próxima aula!

Voltando um pouco pra teoria.

Estrutura de comandos:



