



O **MySQL** é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo

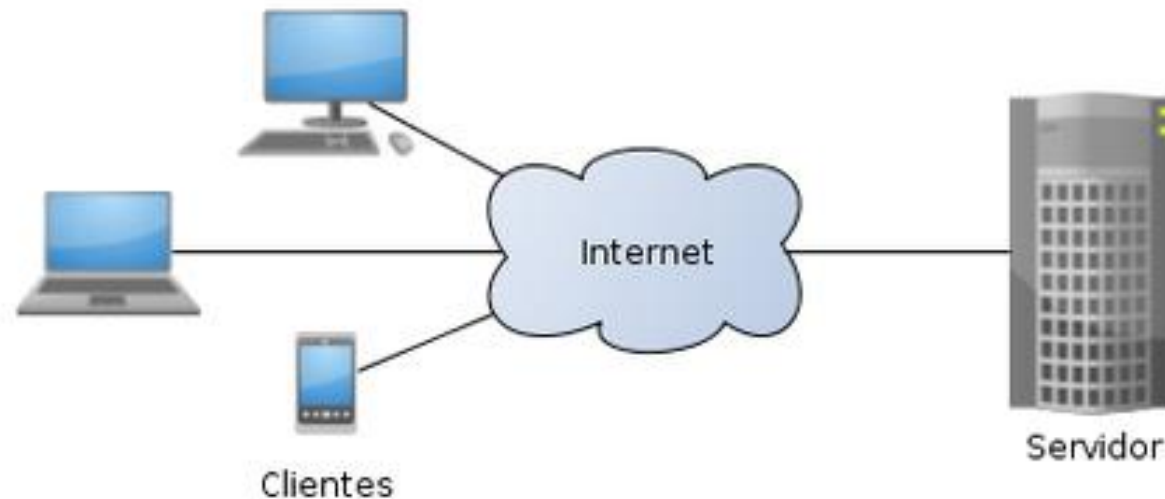
O que é SGBD ?



Um **Sistema de Gerenciamento de Banco de Dados** (SGBD) — do inglês *Data Base Management System* (DBMS) — é o conjunto de [softwares](#) responsáveis pelo gerenciamento de um [banco de dados](#). Seu principal objetivo é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a persistência, a manipulação e a organização dos dados. O SGBD disponibiliza uma [interface](#) para que seus clientes possam incluir, alterar ou consultar dados previamente armazenados. Em [bancos de dados relacionais](#) a interface é constituída pelas [APIs](#) (*Application Programming Interface*) ou [drivers](#) do SGBD, que executam comandos na linguagem [SQL](#) (*Structured Query Language*).



O **modelo cliente-servidor** (em inglês *client/server model*), em computação, é uma estrutura de aplicação distribuída que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como servidores, e os requerentes dos serviços, designados como clientes

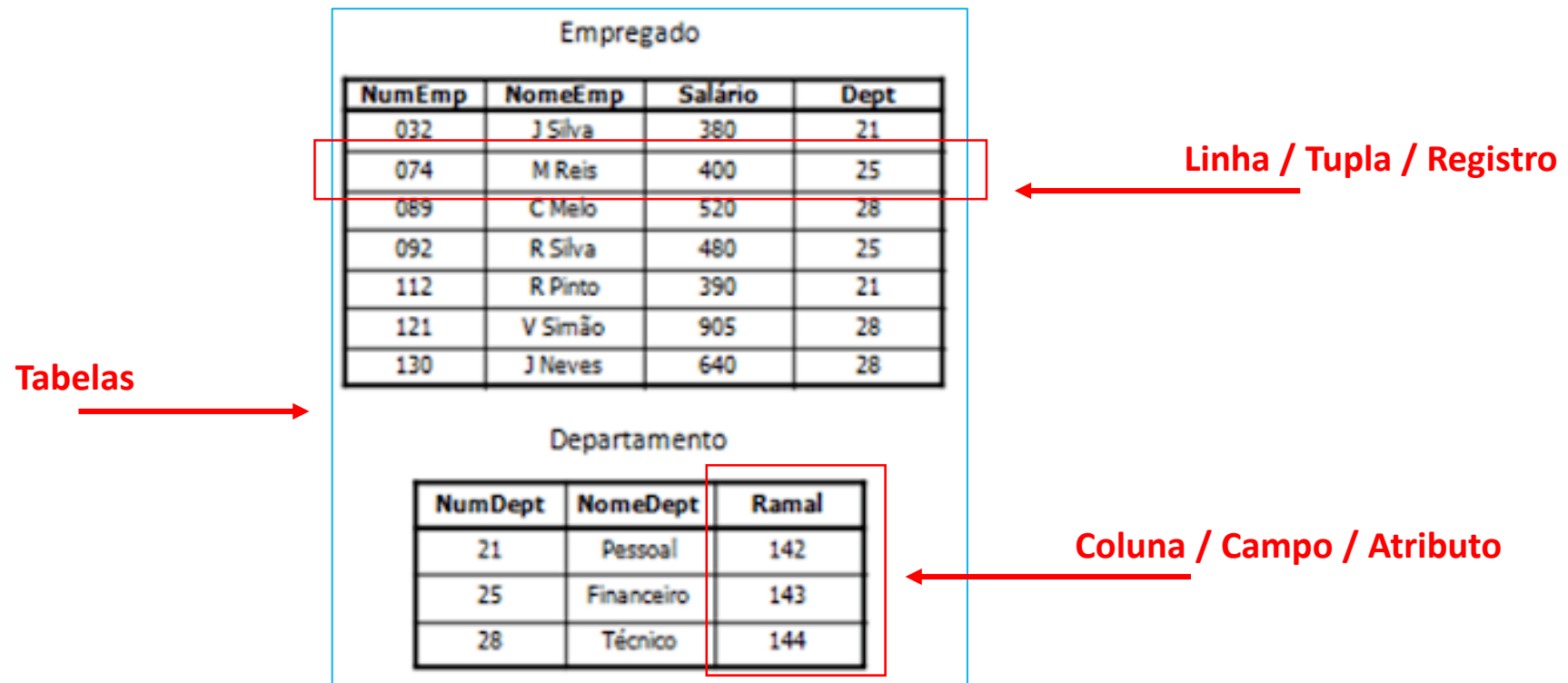


Perguntas ?????



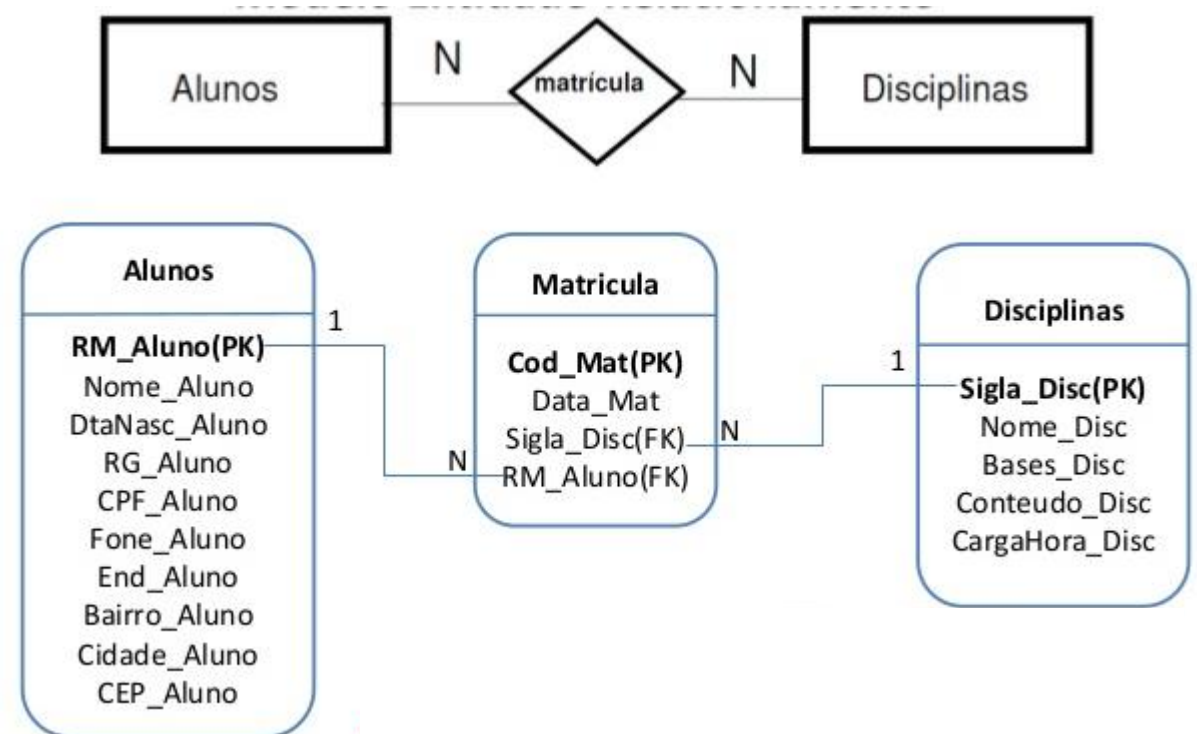
- Posso instalar um banco de dados no meu computador?
- O banco de dados também pode ficar na nuvem?
- Qual o nome do profissional que administra o banco de dados ?
- O que é Big Data?

A arquitetura de um banco de dados relacional pode ser descrita de maneira informal ou formal. Na descrição informal estamos preocupados com aspectos práticos da utilização e usamos os termos tabela, linha e coluna. Na descrição formal estamos preocupados com a semântica formal do modelo e usamos termos como relação (tabela), tupla(linhas) e atributo(coluna).



Todos os dados de um banco de dados relacional (BDR) são armazenados em tabelas. Uma tabela é uma simples estrutura de linhas e colunas. Em uma tabela, cada linha contém um mesmo conjunto de colunas. Em um banco de dados podem existir uma ou centenas de tabelas, sendo que o limite pode ser imposto tanto pela ferramenta de software utilizada, quanto pelos recursos de hardware disponíveis no equipamento.

As tabelas associam-se entre si por meio de regras de relacionamentos, que consistem em associar um ou vários atributos de uma tabela com um ou vários atributos de outra tabela



Registros (ou tuplas)

Cada linha formada por uma lista ordenada de colunas representa um **registro**, ou tupla. Os registros não precisam conter informações em todas as colunas, podendo assumir valores nulos quando assim se fizer necessário.

Colunas (atributos)

As colunas de uma tabela são também chamadas de atributos. Ex.: O campo Nome, ou endereço de uma tabela de um BD relacional.

Chave

As tabelas relacionam-se umas as outras através de chaves. Uma chave é um conjunto de um ou mais atributos que determinam a unicidade de cada registro.

- Chave primária: (*PK - Primary Key*) é um identificador exclusivo de todas as informações de cada registro dando-lhe unicidade. A chave primária nunca se repetirá.

- Chave Estrangeira: (*FK - Foreign Key*) é a chave formada através de um relacionamento com a chave primária de outra tabela. Define um relacionamento entre as tabelas e pode ocorrer repetidas vezes. Caso a chave primária seja composta na origem, a chave estrangeira também o será.

O SQL é uma linguagem e um grande padrão de banco de dados relacional. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado daqueles que se iniciam na linguagem.

Subconjuntos da linguagem SQL

DDL - Linguagem de Definição de Dados

DML - Linguagem de Manipulação de Dados

DQL - Linguagem de Consulta de Dados

DCL - Linguagem de Controle de Dados

DTL - Linguagem de Transação de Dados

SQL - Structured Query Language

(Linguagem de Consulta Estruturada)



DDL (Data Definition Language)

- Linguagem de Definição de Dados
- CREATE: Cria uma estrutura
- ALTER: Altera uma estrutura
- DROP: Exclui uma estrutura



SQL - Structured Query Language

(Linguagem de Consulta Estruturada)



DML (Data Manipulation Language)

- Linguagem de Manipulação de Dados
- INSERT: Insere dados
- UPDATE: Altera dados
- DELETE: Exclui dados



SQL - Structured Query Language

(Linguagem de Consulta Estruturada)



DQL (Data Query Language)

- Linguagem de Consulta de Dados
- SELECT: Retorna dados
- Ordenação de dados
- Agrupamento de dados
- Funções aritméticas
- Filtros de seleção



SQL - Structured Query Language

(Linguagem de Consulta Estruturada)



DCL (Data Control Language)

- Linguagem de Controle de Dados
- GRANT: Habilita acesso a dados e operações
- REVOKE: Revoga acesso a dados e operações



SQL - Structured Query Language

(Linguagem de Consulta Estruturada)



DTL (Data Transaction Language)

- Linguagem de Transação de Dados
- **START TRANSACTION**: Inicia a transação
- **COMMIT**: Concretiza a transação
- **ROLLBACK**: Anula a transação





Conhecer os principais tipos de dados existentes do MySQL é muito importante, uma vez que será algo que vai impactar no funcionamento de seu sistema. Com experiência, ficará automático decidir qual tipo de dado utilizar em cada coluna. O MySQL, como a maioria dos outros SGBD, possui 3 categorias de tipos de dados: **texto, numero e data/tempo**.

Tipo texto

- **CHAR(tamanho)** : guarda um numero fixo de caracteres. Pode conter letras, números e caracteres especiais. O tamanho deve ser declarado entre parênteses. Guarda ate 255 caracteres.
- **VARCHAR(tamanho)**: ele possui as características do tipo CHAR, com a diferença de que, se você criar com mais de 255 caracteres, ele transforma para o tipo TEXT. Ou seja, se for criar algum campo com mais de 255, já crie como TEXT.
- **TEXT**: guarda uma string: com o tamanho máximo de 65.535 caracteres.
- **BLOB**: é o tipo de dado medido pela quantidade de bytes, em vez de pela quantidade de caracteres, conforme a maioria. Pode salvar imagens, por exemplo, com o máximo de 65.535 bytes de arquivo.

Tipo numérico

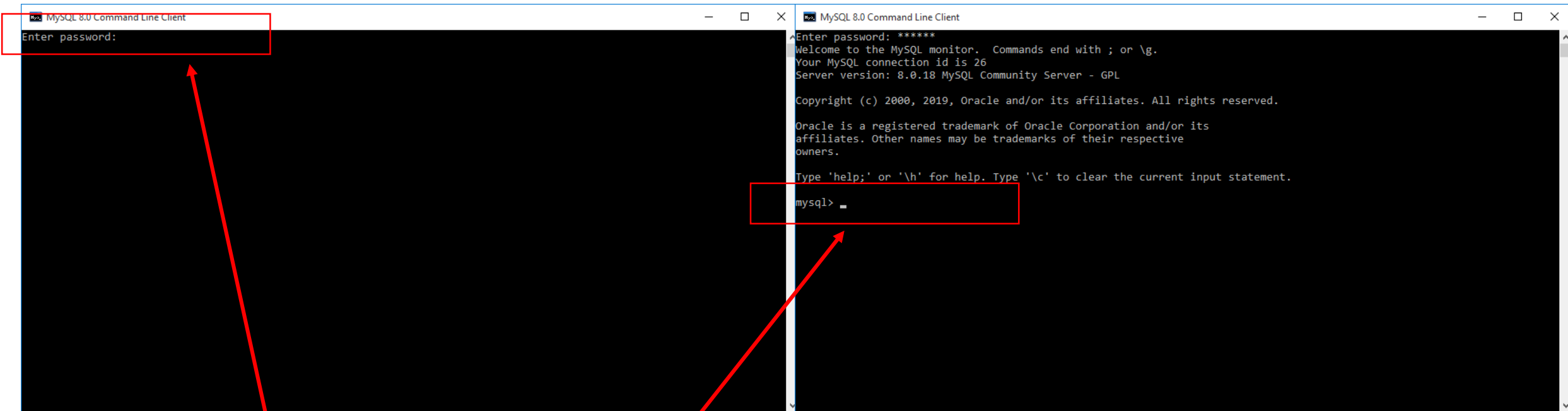
- **TINYINT**: guarda números do tipo inteiro. Suporta de -128 ate 127 caracteres.
- **SMALLINT**: guarda números do tipo inteiro. Suporta de -32768 ate 32767 caracteres.
- **MEDIUMINT**: guarda números do tipo inteiro. Suporta de -8388608 ate 8388607 caracteres.
- **INT(tamanho)**: guarda números inteiros. Suporta de -2147483648 ate 2147483647 caracteres. O numero máximo de caracteres pode ser especificado entre parênteses.
- **BIGINT**: guarda números do tipo inteiro. Suporta de - 9223372036854775808 ate 9223372036854775807 caracteres.
- **FLOAT(tamanho,decimal)**: guarda números REAIS. O numero máximo de caracteres pode ser especificado entre parênteses. Deve-se especificar o tamanho inteiro e o tamanho numérico da coluna.
- **DOUBLE(tamanho,decimal)**: guarda números REAIS. O numero máximo de caracteres pode ser especificado entre parênteses. Deve se especificar o tamanho inteiro e o tamanho numérico da coluna. Esse tipo armazena um quantidade maior de numero do que os campos do tipo FLOAT.



Tipo date/time

- **DATE()**: tipo de campo que vai armazenar datas no: YYYY-MM-DD, onde Y refere-se ao ano, M ao mes e D ao dia;
- **DATETIME()**: a combinação de data e tempo, no formato YYYY-MMDD HH:MI:SS;
- **TIME()**: armazena horas, minutos e segundos no formato HH:MI:SS.

O banco de dados MySql pode ser administrado de algumas formas diferentes dependendo das ferramentas e instalação feita no servidor, a ferramenta mais básica para manipular o banco de dados MySql é o **Command Line Client** que vem junto com a instalação do servidor.



```
MySQL 8.0 Command Line Client
Enter password:

MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Após inserir a senha já podemos enviar comandos para nosso servidor usando o terminal

Quando o Command Line Client não é instalado podemos usar o **cmd** do próprio windows para acessar o servidor Mysql conforme é mostrado abaixo, mas antes devemos configurar o **path nas variáveis de ambiente** para executar os comandos no terminal

```
Prompt de Comando - mysql -h localhost -u root -p

C:\Users\Flavio Mota>mysql -h localhost -u root -p
Enter password:
```

```
Prompt de Comando - mysql -h localhost -u root -p

C:\Users\Flavio Mota>mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

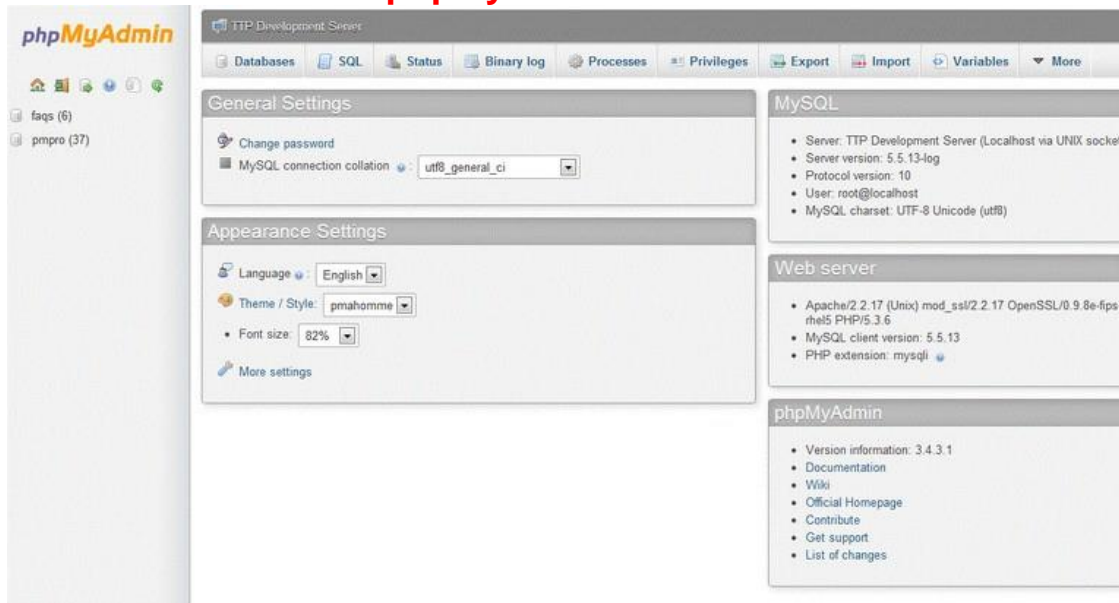
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> _
```

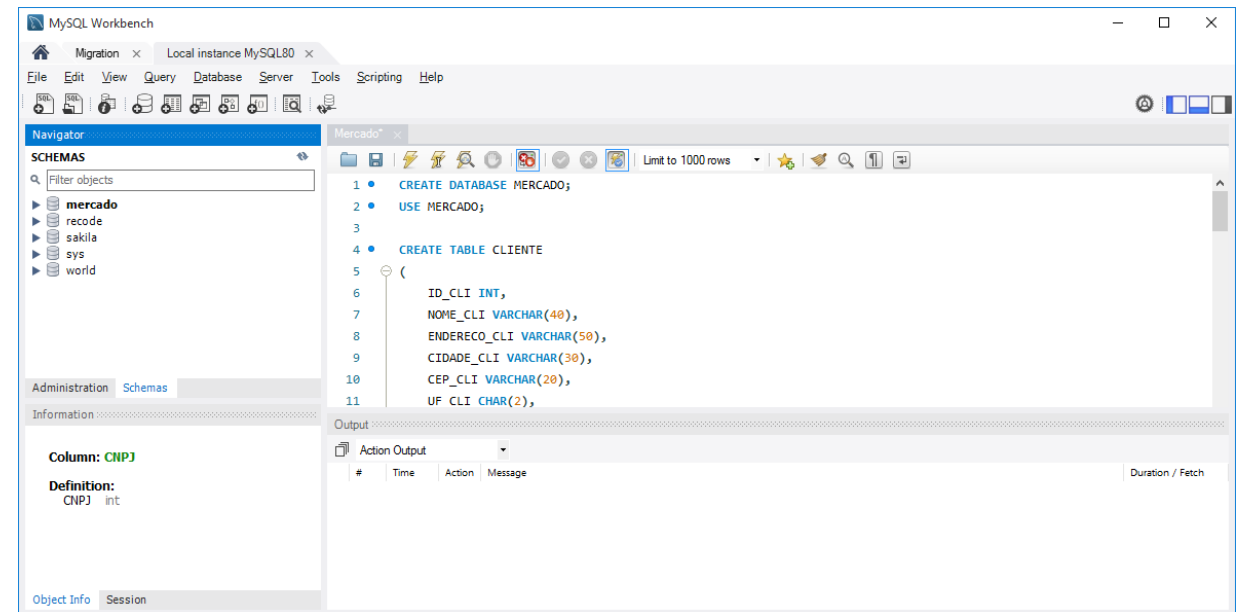
Após inserir os dados do servidor, usuário e senha já podemos enviar comandos para nosso servidor usando o terminal

Geralmente os desenvolvedores não usam o terminal para manipular o servidor MySql, para agilizar o processo de criação e manipulação das bases de dados são usadas ferramentas gráficas como o phpMyadmin e o Workbench no caso servidor Mysql

phpMyadmin

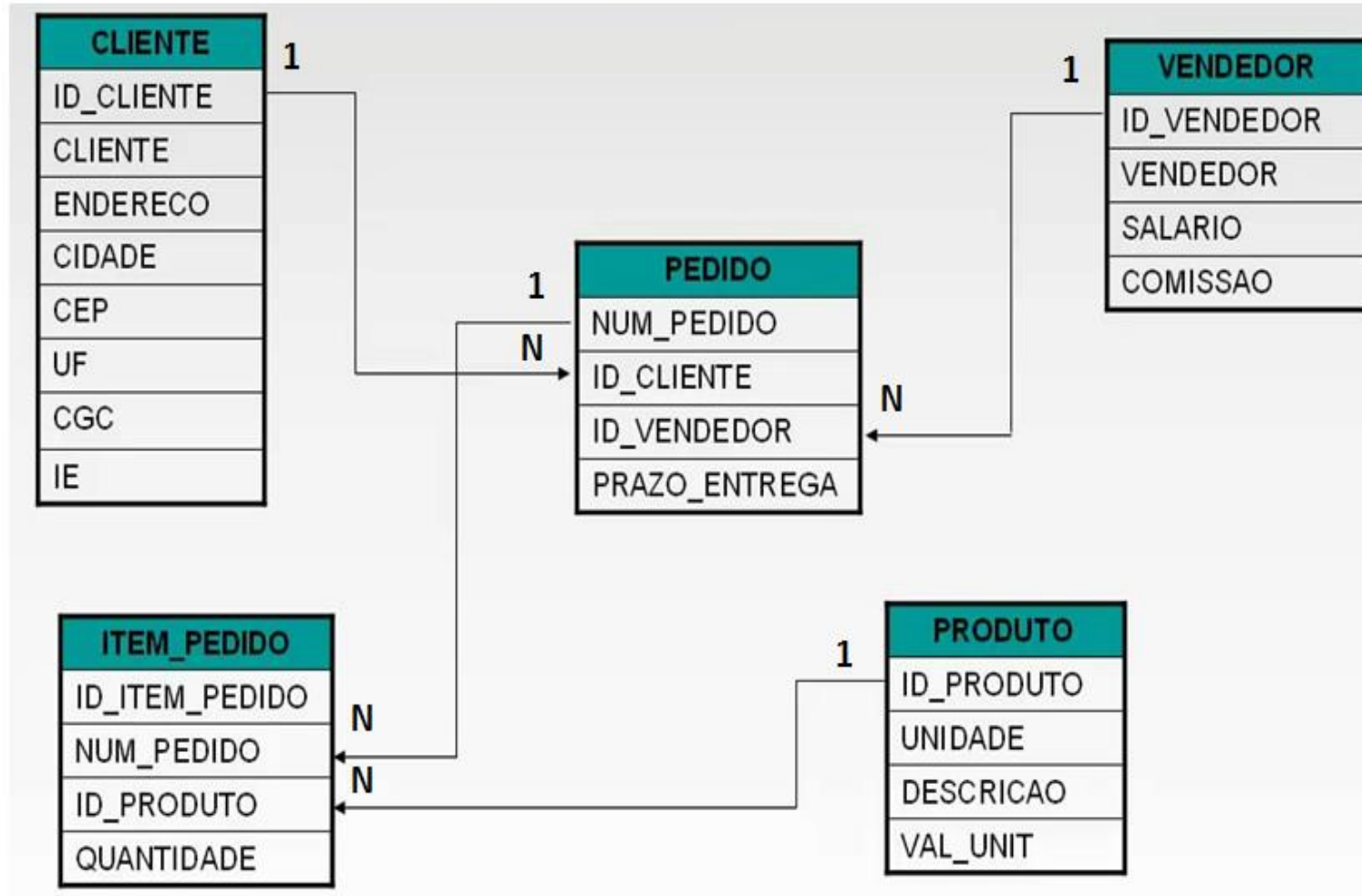


Workbench



Independente da ferramenta utilizada vamos sempre usar o SQL como linguagem.

Flávio Mota



```

CREATE DATABASE MERCADO
USE MERCADO
    
```

CRIANDO A TABELA CLIENTE

```
CREATE TABLE CLIENTE  
(  
    ID_CLI INT,  
    NOME_CLI VARCHAR(40),  
    ENDERECO_CLI VARCHAR(50),  
    CIDADE_CLI VARCHAR(30),  
    CEP_CLI VARCHAR(20),  
    UF_CLI CHAR(2),  
    CGC_CLI VARCHAR(20),  
    IE_CLI VARCHAR(20),  
    PRIMARY KEY(ID_CLI)  
)
```

CLIENTE
ID_CLIENTE
CLIENTE
ENDERECO
CIDADE
CEP
UF
CGC
IE

CRIANDO A TABELA VENDEDOR

```
CREATE TABLE VENDEDOR  
(  
    ID_VENDEDOR INT,  
    NOME_VEND VARCHAR(40),  
    SALARIO_VEND FLOAT,  
    COMISSAO FLOAT,  
  
    PRIMARY KEY (ID_VENDEDOR)  
)
```

VENDEDOR
ID_VENDEDOR
VENDEDOR
SALARIO
COMISSAO

CRIANDO A TABELA PRODUTO

```
CREATE TABLE PRODUTO  
(  
    ID_PRODUTO INT,  
    UNIDADE CHAR(2),  
    DESCRICAO VARCHAR(20),  
    VAL_UNIT FLOAT,  
    PRIMARY KEY(ID_PRODUTO),  
);
```

PRODUTO
ID_PRODUTO
UNIDADE
DESCRICAO
VAL_UNIT

CRIANDO A TABELA PEDIDO

```
CREATE TABLE PEDIDO
(
    NUM_PEDIDO INT,
    ID_CLI INT,
    ID_VENDEDOR INT,
    PRAZO_ENTREGA DATE,

    PRIMARY KEY (NUM_PEDIDO),
    FOREIGN KEY (ID_CLI) REFERENCES CLIENTE (ID_CLI),
    FOREIGN KEY (ID_VENDEDOR) REFERENCES VENDEDOR(ID_VENDEDOR)
)
```

PEDIDO
NUM_PEDIDO
ID_CLIENTE
ID_VENDEDOR
PRAZO_ENTREGA

CRIANDO A TABELA ITENS PEDIDO

```
CREATE TABLE ITENS_PEDIDO
(
    ID_ITEM_PEDIDO INT,
    NUM_PEDIDO INT,
    ID_PRODUTO INT,
    QUANTIDADE INT,
    PRIMARY KEY(ID_ITEM_PEDIDO),
    FOREIGN KEY(NUM_PEDIDO) REFERENCES PEDIDO(NUM_PEDIDO),
    FOREIGN KEY(ID_PRODUTO) REFERENCES PRODUTO(ID_PRODUTO)
)
```

ITEM_PEDIDO
ID_ITEM_PEDIDO
NUM_PEDIDO
ID_PRODUTO
QUANTIDADE



DROP TABLE ITENS_PEDIDO; // Apaga a tabela especificada

DROP DATABASE MERCADO; // Apaga o banco especificado

Alterando a estrutura de uma tabela usando o ALTER TABLE

Apagando uma coluna

```
ALTER TABLE CLIENTE DROP COLUMN IE_CLI;
```

Adicionando uma coluna

```
ALTER TABLE CLIENTE ADD COLUMN IE_CLI VARCHAR(20);
```

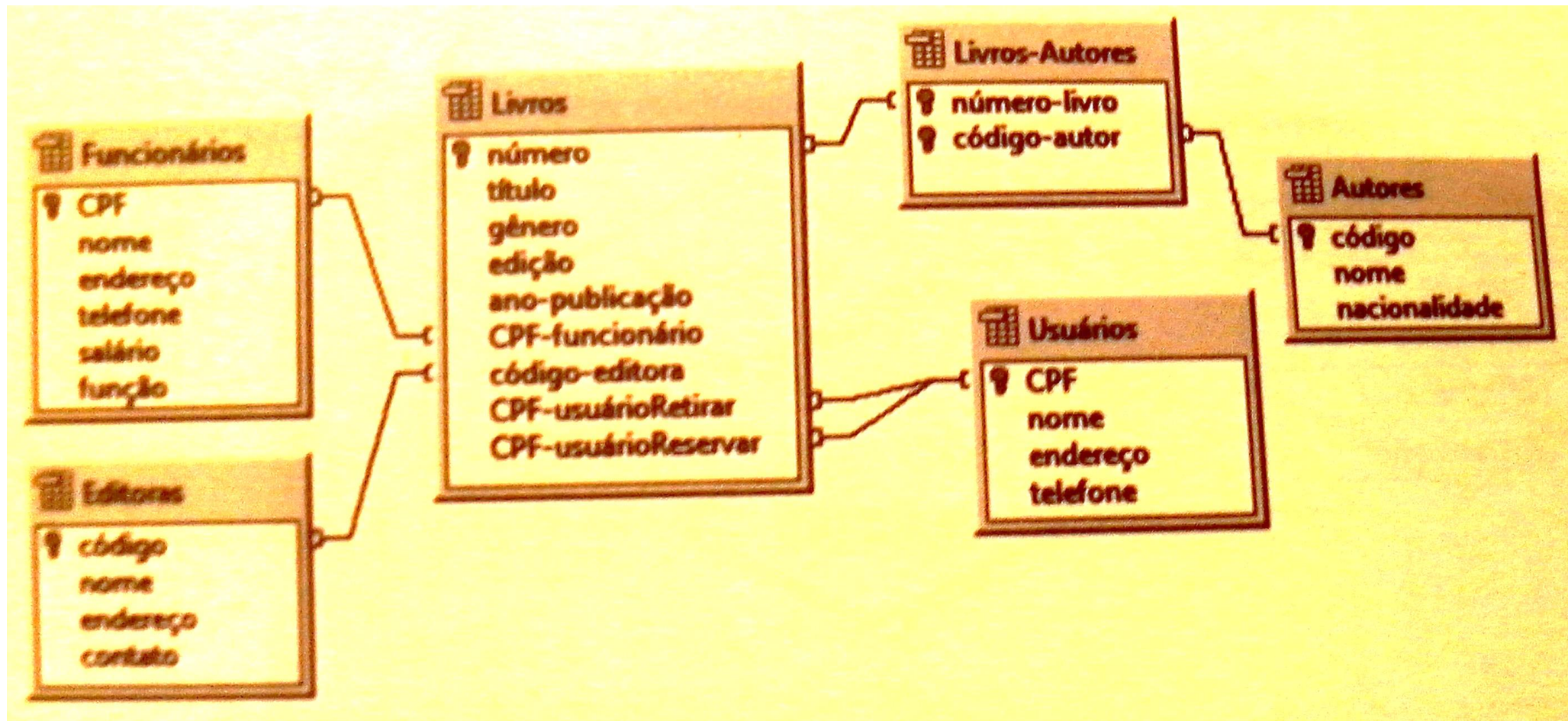
Mudando o nome de uma coluna

```
ALTER TABLE CLIENTE CHANGE CGC_CLI CNPJ VARCHAR(20);
```

Mudando o tipo de uma coluna

```
ALTER TABLE CLIENTE MODIFY CNPJ INT;
```

Crie no Mysql o seguinte banco de dados respeitando seus relacionamentos conforme está especificado na modelagem.





Após criar o banco de dados anterior faça as seguintes alterações:

- Altere o campo telefone do funcionário para tel_Res
- Adicione um campo celular para o funcionário do tipo varchar(20)
- Adicione um campo endereço para a tabela editora
- Apague o campo gênero da tabela livro
- Apague a tabela Livros-Autores

Observação: Crie um novo banco de dados igual ao anterior com outro nome para fazer as alterações e preservar o banco original