



Laboratório 1: Introdução ao MySQL

15/09/2023



O objetivo deste laboratório: Apresentar o sistema de gerenciamento de banco de dados MySQL, bem como o cliente `mysql-client`. Será demonstrado como executar instruções SQL simples, como: criar, alterar e apagar tabelas; inserir, alterar e excluir registros nas tabelas.

1 Executando servidor MySQL

É possível instalar o servidor MySQL no Linux, macOS ou Windows. Por padrão no Linux, assim como qualquer outro servidor, o MySQL será iniciado sempre que a máquina for iniciada. Caso não deseje isso, então poderá usar o comando `systemctl disable nome-do-serviço` para impedir a subida no serviço no *boot*. Outra opção seria executar o servidor MySQL dentro de contêiner Docker.

1.1 Instalando o servidor no Ubuntu Linux

```
sudo apt-get install -y mysql-server mysql-client
```

1.2 Executando servidor MySQL dentro de contêiner Docker

Você precisará ter o Docker instalado e deixar o *daemon* do serviço em execução. Para isto veja a documentação oficial em <https://docs.docker.com/get-docker/>. Para executar o servidor MySQL 8 dentro de um contêiner Docker, execute a seguinte linha de comando:

```
docker run --rm -d -p 3306:3306 -e MYSQL_ROOT_PASSWORD=senhaRoot -e MYSQL_ROOT_HOST='%' \
-e MYSQL_DATABASE=bcd -e MYSQL_USER=aluno -e MYSQL_PASSWORD=alunoAluno \
-v $(pwd)/mysql-data:/var/lib/mysql --name meumysql mysql/mysql-server:latest
```

Você terá um servidor em execução, com um mapeamento da porta 3306 da máquina hospedeira (seu computador) para a porta 3306 dentro do contêiner. Será feito um mapeamento do diretório do contêiner onde o MySQL guarda as bases de dados (`/var/lib/mysql`) para o diretório no local `mysql-data`. Assim, os dados continuarão no sistema de arquivo do computador hospedeiro mesmo que o contêiner seja destruído.

Será criado um usuário chamado `aluno`, com a senha `aluno` e um esquema (banco) chamado `bcd`. Também foi definida a senha `senhaRoot` para o usuário `root` e foi permitindo que o `root` possa acessar o banco a partir de qualquer endereço IP (o padrão seria somente na `localhost`).

- `docker exec -it meumysql bash` – Obtendo uma shell dentro do contêiner
 - Dentro do contêiner você pode executar o cliente do `mysql`.
 - Exemplo: `mysql -u root -psenhaRoot`
- `docker stop meumysql` – parando o contêiner
- `docker start meumysql` – iniciando o contêiner

1.3 Exemplo usando Docker Compose

*Docker Compose*¹ é uma ferramenta para facilitar o trabalho com aplicações que precisam ser executadas com múltiplos contêineres Docker ou que precisam passar diversos parâmetros para execução do contêiner. Crie um arquivo com o nome `docker-compose.yml`, contendo as linhas presentes na listagem abaixo.

```
version: "3.9"
services:
  db:
    image: mysql:latest
    container_name: meumysql
    environment:
      - MYSQL_DATABASE=bcd
      - MYSQL_ROOT_PASSWORD=senhaRoot
      - MYSQL_ROOT_HOST='%'
      - MYSQL_USER=aluno
      - MYSQL_PASSWORD=senhaAluno
    volumes:
      # Dados ficam no host e nao dentro do container
      - ./mysql-data:/var/lib/mysql
      # Todos arquivos *.sql armazenados em scritps-para-iniciar-o-banco serao executados (em
      # ordem alfabetica) na primeira vez que o container for iniciado
      - ./scritps-para-iniciar-o-banco:/docker-entrypoint-initdb.d
    cap_add:
      - SYS_NICE
    ports:
      - 3306:3306
```

Para executar o contêiner, execute: `docker compose start`



Todos os arquivos com a extensão `.sql`, e armazenados no diretório `/docker-entrypoint-initdb.d`, serão lidos na primeira vez que o contêiner for iniciado. Use isso para povoar o banco de dados. Veja mais informações na documentação oficial em https://hub.docker.com/_/mysql.

2 Usando o cliente MySQL

O cliente MySQL aceita como parâmetros o endereço IP ou nome do servidor MySQL (se omitir ele tentará no localhost), o nome de um usuário do MySQL (não necessariamente o root) e a opção para indicar que fornecerá a senha posteriormente.

```
# Sintaxe: mysql -h IP-ou-nome-maquina -u usuario -p[senha]
mysql -u aluno -paluno

# Para executar o mysql client dentro do contêiner Docker que está em execução com o servidor
docker exec -it meu-mysqlserver -u aluno -p
```

```
-- alterando a codificação de caracteres do cliente para UTF-8
mysql> charset utf8;

-- para ativar o auto completar com a tecla TAB
mysql> \#

-- para limpar a tela CTR-L
-- para sair do cliente MySQL pode-se usar: exit, quit ou CTRL+d
mysql> exit
```

¹<https://docs.docker.com/compose>

2.1 Registrando os comandos de uma sessão

O comando `tee` permite registrar toda a atividade em uma sessão do cliente MySQL em um arquivo. O comando `notee` encerra o registro das atividades.

```
mysql> tee labmysql-registro.txt
-- diversos comandos
mysql> notee
mysql> quit
```

2.2 Criando um banco de dados e dando todas as permissões para um novo usuário

```
mysql> CREATE DATABASE lab;

-- Criando usuário aluno, com a senha '1234' e delegando todos direitos sobre o banco 'lab'
mysql> CREATE USER IF NOT EXISTS 'aluno'@'%' IDENTIFIED WITH mysql_native_password by '1234';
mysql> GRANT ALL ON lab.* TO 'aluno'@'%';
```

2.3 Criando tabelas e outras instruções

```
mysql -h 127.0.0.1 -u aluno -p
```

```
-- listando todos os bancos de dados
mysql> SHOW DATABASES;
-- Escolhendo o banco: lab
mysql> USE lab;

-- Criando a tabela Aluno(matricula :int, nome :varchar, email :varchar)
mysql> CREATE TABLE Aluno (matricula INT NOT NULL AUTO_INCREMENT, nome VARCHAR(80), email VARCHAR
(80), PRIMARY KEY (matricula));

-- Obtendo informacoes sobre a estrutura da tabela. Os seguintes comandos são equivalentes no
MySQL: desc, describe, explain, show columns from
mysql> DESC Aluno;
-- Exibindo a instrução SQL que criou a tabela Aluno
mysql> SHOW CREATE TABLE Aluno;
```

2.3.1 Criando tabelas com relacionamento

```
-- Criar tabela Departamento
CREATE TABLE Departamento(
idDepartamento INT NOT NULL,
dNome VARCHAR(255) NOT NULL,
Orcamento DOUBLE NOT NULL,
PRIMARY KEY (idDepartamento))

-- Criar tabela Funcionarios
CREATE TABLE Funcionario(
idFuncionario INT NOT NULL,
Nome VARCHAR(45) NOT NULL,
Sobrenome VARCHAR(45) NOT NULL,
idDepartamento INT NOT NULL,
PRIMARY KEY (idFuncionario),
CONSTRAINT fk_Funcionario_Departamento FOREIGN KEY (idDepartamento)
REFERENCES Departamento (idDepartamento))
```

2.4 Trabalhando com scripts

É possível executar scripts SQL que estão salvos no sistema de arquivos de duas maneiras: (1) de forma interativa dentro do console do cliente MySQL; (3) processando em lote, estando fora do console do cliente MySQL.

```
-- Conteúdo do arquivo: lab.sql
CREATE TABLE Curso (idCurso INT NOT NULL AUTO_INCREMENT, nome VARCHAR(80),
    PRIMARY KEY (idCurso));

INSERT INTO Curso (nome) VALUES ('Telecomunicações'),('Computação');

SELECT * FROM Curso;
```

```
-- Executando o script de forma interativa
mysql> source lab.sql
```

```
# Executando o script na shell. O banco de dados é 'lab' e o usuário é 'aluno'.
mysql lab < lab.sql -u aluno -p
```

2.5 Fazendo backup de um banco de dados em arquivo

O aplicativo mysqldump permite salvar o DML e DDL de um banco em um arquivo texto.

```
# salvando no arquivo backup-lab.sql o DML e DDL do banco 'lab01'
mysqldump lab -u aluno -p > backup-lab.sql
```

2.6 Banco de dados de exemplo

No site do MySQL² é possível baixar o “world database”, um banco de dados de exemplo. O banco de dados possui 4 tabelas: city, country, countryinfo e countrylanguage. Baixe o arquivo com as instruções DDL e DML em https://emersonmello.me/ensino/bcd/labs/world_x.sql.

²<https://dev.mysql.com/doc/world-setup/en/>