

# Exemplo de Uso MySQL via Linha de Comando

## Disciplina Banco de Dados

Profas. Marta Mattoso e Vanessa Braganholo

Tutor a Distância Ricardo Silva



GOVERNO DO  
**Rio de Janeiro**

SECRETARIA DE  
CIÊNCIA E TECNOLOGIA



Fundação Centro de Ciências e Educação Superior a Distância do Estado do Rio de Janeiro

# Diretivas gerais

- Este exemplo foi executado em plataforma Windows, no entanto, não há diferenças na forma de operação do MySQL nos ambientes Windows e LINUX

# Log in no MySQL

- Para efetuar login no MySQL entre com o seguinte comando:
  - `mysql -u <<nome do usuário>> -p`
- A diretiva -u indica o usuário e -p que será utilizada uma senha
- Após a execução deste comando a senha do usuário será solicitada

# Log in no MySQL

- Por exemplo:

```
C:\Documents and Settings\Ricktadeus>mysql -u root -p  
Enter password:
```

# Log in no MySQL

- Após entrar com a senha, um texto de boas vindas é exibido e o prompt passa ao formato `mysql>` (onde serão digitados os comandos SQL)

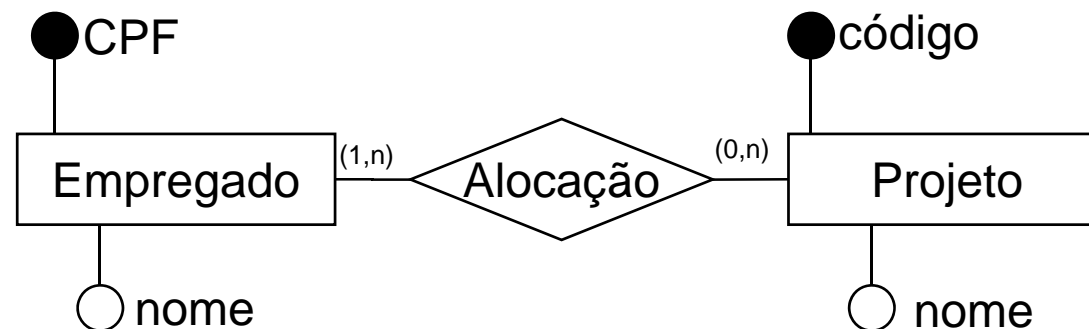
```
C:\Documents and Settings\Ricktadeus>mysql -u root -p
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 5.0.51b-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql >
```

# Modelo ER

- Vamos implementar o BD que representa os empregados de uma empresa e sua alocação a projetos
- O modelo ER correspondente é o que segue:



# Esquema Relacional

- Empregado (cpf, nome)
- Projeto (codigo, nome)
- Alocao (cpf, codigo)
  - Cpf referencia Empregado
  - Codigo referencia Projeto

# Criação do Esquema

- Inicialmente vamos criar um esquema para o nosso banco de dados.
- No MySQL isso é feito através do seguinte comando:
  - Create schema <<nome\_schema>>



# Criação do Esquema

- Vamos criar um esquema denominado empresa:

```
mysql > create schema empresa;
```

- Após a execução deste comando, o MySQL indica o resultado da execução

```
mysql > create schema empresa;  
Query OK, 1 row affected (0.00 sec)  
  
mysql >
```

# Selecionar o Esquema

- Após a criação do esquema, o mesmo deve ser selecionado para uso.
- No MySQL isso é feito com o comando:
  - Use `<<nome_schema>>;`

# Selecionar o Esquema

- Vamos selecionar o esquema empresa:

```
mysql > use empresa;
```

- Como resultado, o MySQL indica a troca do esquema utilizado

```
mysql > use empresa;  
Database changed  
mysql >
```

# Criação das Tabelas

- Agora vamos criar a tabela empregado através do seguinte script SQL:

```
create table Empregado(  
    CPF char(11) not null,  
    nome varchar(200) not null,  
    primary key(CPF),  
    unique(CPF)  
);
```

# Criação das Tabelas

- No console...

```
mysql> create table Empregado(  
->   CPF char(11) not null,  
->   nome varchar(200) not null,  
->   primary key(CPF),  
->   unique(CPF)  
-> );
```

- Resultando em...

```
mysql> create table Empregado(  
->   CPF char(11) not null,  
->   nome varchar(200) not null,  
->   primary key(CPF),  
->   unique(CPF)  
-> );  
Query OK, 0 rows affected (0.01 sec)  
  
mysql>
```

# Criação das Tabelas

- Agora vamos criar a tabela projeto que é definida com o seguinte script SQL:

```
create table Projeto (  
    codigo integer not null auto_increment,  
    nome varchar(200) not null,  
    primary key(codigo)  
);
```

# Criação das Tabelas

- No console...

```
mysql> create table Projeto (  
->   codigo integer not null auto_increment,  
->   nome varchar(200) not null,  
->   primary key(codigo)  
-> );
```

- Resultando em...

```
mysql> create table Projeto (  
->   codigo integer not null auto_increment,  
->   nome varchar(200) not null,  
->   primary key(codigo)  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql>
```

# Criação das Tabelas

- Agora vamos criar a tabela alocação. Essa tabela representa o relacionamento entre os empregados e sua alocação em um projeto

```
create table Alocacao(  
  CPFEmpregado char(11) not null,  
  codigoProjeto integer not null,  
  primary key(CPFEmpregado,codigoProjeto),  
  unique (CPFEmpregado,codigoProjeto),  
  foreign key(CPFEmpregado) references Empregado(CPF)  
    on delete restrict      on update cascade,  
  foreign key(codigoProjeto) references Projeto(codigo)  
    on delete restrict      on update cascade  
);
```



# Criação das Tabelas

- No console...

```
mysql> create table Alocacao(  
-> CPFEmpregado char(11) not null,  
-> codigoProjeto integer not null,  
-> primary key(CPFEmpregado, codigoProjeto),  
-> unique (CPFEmpregado, codigoProjeto),  
-> foreign key(CPFEmpregado) references Empregado(CPF)  
->     on delete restrict          on update cascade,  
-> foreign key(codigoProjeto) references Projeto(codigo)  
->     on delete restrict          on update cascade  
-> );
```

# Criação das Tabelas

- Resultando em...

```
mysql> create table Alocacao(  
-> CPFEmpregado char(11) not null,  
-> codi goProjeto integer not null,  
-> primary key(CPFEmpregado, codi goProjeto),  
-> unique (CPFEmpregado, codi goProjeto),  
-> foreign key(CPFEmpregado) references Empregado(CPF)  
-> on delete restrict on update cascade,  
-> foreign key(codi goProjeto) references Projeto(codi go)  
-> on delete restrict on update cascade  
-> );  
Query OK, 0 rows affected (0.00 sec)  
  
mysql>
```

# Verificar as tabelas criadas

- Para verificar as tabelas criadas, utilizamos o comando MySQL show tables

```
mysql > show tables;
+-----+
| Tables_in_empresa |
+-----+
| alocao              |
| empregado           |
| projeto             |
+-----+
3 rows in set (0.00 sec)

mysql >
```

# Verificar as tabelas criadas

- Já para verificar as colunas das tabelas, utilizamos o comando `show columns`:

```
mysql > show columns from alocao;
```

Field	Type	Null	Key	Default	Extra
CPFEmpregado	char(11)	NO	PRI	NULL	
codigoProjeto	int(11)	NO	PRI	NULL	

```
2 rows in set (0.02 sec)
```

```
mysql >
```

# Carregando a Base

- Agora vamos carregar as tabelas com dados
- Isso é feito via o comando insert

# Carregando a Base

- Os empregados a serem inseridos são:
  - Teobaldo Fernandes (CPF: 17974142XX)
  - Antônio de Souza (CPF: 263956372XX)
  - Luiz Alberto Cabral (CPF: 415675901XX)
  - Aline Ribeiro (CPF: 762108617XX)
  - Joana Helena (CPF: 840426843XX)
  - Elizabeth dos Reis Senna (CPF: 903617687XX)

# Carregando a Base

- Logo, os comandos SQL a serem executados são os que seguem:
  - INSERT INTO EMPREGADO VALUES ('179741420XX', 'Teobaldo Fernandes');
  - INSERT INTO EMPREGADO VALUES ('263956372XX', 'Antônio de Souza');
  - INSERT INTO EMPREGADO VALUES ('415675901XX', 'Luiz Alberto Cabral');
  - INSERT INTO EMPREGADO VALUES ('762108617XX', 'Aline Ribeiro');
  - INSERT INTO EMPREGADO VALUES ('840426843XX', 'Joana Helena');
  - INSERT INTO EMPREGADO VALUES ('903617687XX', 'Elizabeth dos Reis Senna');

# Carregando a Base

- Tais comandos devem ser executados um a um no console:

```
mysql> INSERT INTO EMPREGADO VALUES ('179741420XX', 'Teobaldo Fernandes');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO EMPREGADO VALUES ('263956372XX', 'Antônio de Souza');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO EMPREGADO VALUES ('415675901XX', 'Luiz Alberto Cabral');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO EMPREGADO VALUES ('762108617XX', 'Aline Ribeiro');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO EMPREGADO VALUES ('840426843XX', 'Joana Helena');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO EMPREGADO VALUES ('903617687XX', 'Elizabeth dos Reis Senna');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
```



# Carregando a Base

- Vamos verificar se os dados foram realmente inseridos
- Para isso, devemos efetuar um select sobre a tabela empregado

# Carregando a Base

- Como resultado da execução do comando, temos:

```
mysql > select * from empregado;
+-----+-----+
| CPF          | nome                |
+-----+-----+
| 179741420XX  | Teobal do Fernandes |
| 263956372XX  | Antônio o de Souza  |
| 415675901XX  | Lui z Al berto Cabral |
| 762108617XX  | Al i ne Ri bei ro   |
| 840426843XX  | Joana Hel ena       |
| 903617687XX  | El i zabeth dos Rei s Senna |
+-----+-----+
6 rows in set (0.02 sec)

mysql >
```

# Carregando a Base

- Da mesma forma, devemos entrar com os dados dos projetos. Em nosso caso, cada projeto desenvolve um produto:
  - Produto X, Produto Y e Produto Z

# Carregando a Base

- Em SQL:
  - INSERT INTO PROJETO VALUES (1, 'Produto X');
  - INSERT INTO PROJETO VALUES (2, 'Produto Y');
  - INSERT INTO PROJETO VALUES (3, 'Produto Z');

# Carregando a Base

- Tais comandos devem ser executados um a um no console:

```
mysql > INSERT INTO PROJETO VALUES (1, 'Produto X');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql > INSERT INTO PROJETO VALUES (2, 'Produto Y');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql > INSERT INTO PROJETO VALUES (3, 'Produto Z');  
Query OK, 1 row affected (0.00 sec)
```

```
mysql >
```

# Carregando a Base

- Verificando se os dados foram corretamente inseridos:

```
mysql > select * from projeto;
```

codi go	nome
1	Produto X
2	Produto Y
3	Produto Z

```
3 rows in set (0.00 sec)
```

```
mysql >
```

# Carregando a Base

- Agora vamos alocar as pessoas aos projetos da seguinte maneira:
  - Projeto X: Teobaldo e Antônio
  - Projeto Y: Luiz Alberto
  - Projeto Z: Aline, Joana e Elizabeth

# Carregando a Base

- Em SQL:
  - INSERT INTO ALOCACAO VALUES ('179741420XX','1');
  - INSERT INTO ALOCACAO VALUES ('263956372XX','1');
  - INSERT INTO ALOCACAO VALUES ('415675901XX','2');
  - INSERT INTO ALOCACAO VALUES ('762108617XX','3');
  - INSERT INTO ALOCACAO VALUES ('840426843XX','3');
  - INSERT INTO ALOCACAO VALUES ('903617687XX','3');



# Carregando a Base

- Tais comandos devem ser executados um a um no console:

```
mysql> INSERT INTO ALOCACAO VALUES (' 179741420XX' , ' 1' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO ALOCACAO VALUES (' 263956372XX' , ' 1' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO ALOCACAO VALUES (' 415675901XX' , ' 2' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO ALOCACAO VALUES (' 762108617XX' , ' 3' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO ALOCACAO VALUES (' 840426843XX' , ' 3' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO ALOCACAO VALUES (' 903617687XX' , ' 3' );  
Query OK, 1 row affected (0.00 sec)
```

```
mysql>
```

# Carregando a Base

- Agora vamos verificar as alocações:

```
mysql> select e.nome, p.nome from empregado e,  
projeto p, alocao a where a.CPFEmpregado=e.CPF  
and a.codigoProjeto=p.codigo;
```

nome	nome
Teobaldo Fernandes	Produto X
Antônio de Souza	Produto X
Luiz Alberto Cabral	Produto Y
Aline Ribeiro	Produto Z
Joana Helena	Produto Z
Elizabeth dos Reis Senna	Produto Z

6 rows in set (0.02 sec)

```
mysql>
```

# Atualização de Registros

- Para atualizar dados no banco, devemos escrever o comando SQL correspondente e executá-lo, no caso, o comando UPDATE
- Por exemplo, vamos mudar o nome do empregado Luiz Alberto Cabral (alocado ao projeto Y) para Alberto Luiz Cabral

# Atualização de Registros

- Em SQL...

```
mysql > update empregado  
      ->      set nome='Al berto Lui z Cabral '  
      ->  where cpf='415675901XX' ;  
Query OK, 1 row affected (0.03 sec)  
Rows matched: 1  Changed: 1  Warnings: 0  
  
mysql >
```

- Note que após a execução do comando, o MySQL retorna o resultado da execução.

# Atualização de Registros

- Para verificar se a alteração surtiu efeito, efetuamos um SELECT sobre a tabela:

```
mysql > select * from empregado where cpf=' 41567590160' ;
+-----+-----+
| CPF          | nome                |
+-----+-----+
| 415675901XX | Alberto Luiz Cabral |
+-----+-----+
1 row in set (0.00 sec)

mysql >
```

# Exclusão de Registros

- Da mesma forma, para excluirmos registros das tabelas do banco, devemos escrever o comando SQL correspondente e executá-lo
- Por exemplo, a empregada Aline Ribeiro não participará mais do Projeto Z

# Exclusão de Registros

- Em SQL...

```
mysql > delete from alocao where CPFEmpregado=' 762108617XX' ;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql >
```

- Note que, mais uma vez, o MySQL retorna o resultado da execução.

# Exclusão de Registros

- Para verificar o resultado da exclusão, efetuamos um SELECT sobre a tabela:

```
mysql > select * from alocao where CPFEmpregado=' 762108617XX' ;  
Empty set (0.00 sec)
```

```
mysql >
```



# Saindo do MySQL

- Para sair do MySQL, utilizamos o comando EXIT

```
mysql > exit  
Bye
```

```
C: \Documents and Settings\Ri cktadeus>
```

- Note que o MySQL se despede com um “Bye”