

Introdução aos Bancos de dados

Renato Rocha Souza
rsouza.fgv@gmail.com

Programa:

Linguagem SQL – Parte I

- Introdução ao MySQL – CLI
- Consultas simples
- Consultas em mais de uma linha
- Cancelando comandos
- Usando bancos
- Manipulando tabelas
- Carregando dados
- Selecionando dados
- Ordenando os dados
- Trabalhando com NULL
- Usando Pattern Matching
- Contando Linhas
- Trabalhando em “batch”



Introdução ao MySQL

Slides inspirados no tutorial:

<http://dev.mysql.com/doc/refman/5.7/en/tutorial.html>

MySQL

- MySQL é um banco de dados popular, open source.
- Pode ser usado para bancos de dados de tamanho considerável.
- Provê um *shell* interativo para utilização em linha de comando, e também interfaces gráficas (ex: MySQL Workbench)

Sessão interativa (CLI)

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
rsouza@CNPQnote:~$ mysql -u root -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 1  
Server version: 5.7.4-m14 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2014, 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> █
```

- Para terminar a sessão, digite QUIT or EXIT:

```
mysql> QUIT
```

```
mysql> exit
```

Consultas (queries) simples

- Uma vez em que esteja logado, podem-se tentar consultas simples.
- Por exemplo:

```
mysql> SELECT VERSION() , CURRENT_DATE;
```

```
+-----+-----+  
| VERSION() | CURRENT_DATE |  
+-----+-----+  
| 3.23.49   | 2002-05-26   |  
+-----+-----+
```

```
1 row in set (0.00 sec)
```

- Os comandos são delimitados com ponto e vírgula (;)
- MySQL também informa o número de linhas retornadas e o tempo necessário para efetuar a consulta.

Consultas (queries) simples

- Os comandos não são *case-sensitive*.
- As consultas a seguir são equivalentes:

```
mysql> SELECT VERSION() , CURRENT_DATE;
```

```
mysql> select version() , current_date;
```

```
mysql> SeLeCt vErSiOn() , current_DATE;
```

Consultas (queries) simples

- Outro exemplo, demonstrando o uso de MySQL como uma calculadora simples:

```
mysql> SELECT SIN (PI () /4) , (4+1) *5 ;
```

+-----+-----+	
SIN (PI () /4)	(4+1) *5
+-----+-----+	
0.707107	25
+-----+-----+	

Consultas (queries) simples

- Podem-se encadear vários comandos, separando-os com ponto e vírgula (;)

```
mysql> SELECT VERSION() ; SELECT NOW() ;
```

```
+-----+
| VERSION() |
+-----+
| 3.22.20a-log |
+-----+
+-----+
| NOW() |
+-----+
| 2004 00:15:33 |
+-----+
```

Consultas de mais de uma linha

- Você pode usar várias linhas para uma consulta, terminando o comando com ponto e vírgula (;)

```
mysql> SELECT  
      -> USER()  
      -> ,  
      -> CURRENT_DATE;
```

```
+-----+-----+  
| USER()          | CURRENT_DATE |  
+-----+-----+  
| joesmith@localhost | 1999-03-18   |  
+-----+-----+
```

Cancelando um comando

- Você pode cancelar um comando multilinha usando \c

```
mysql> SELECT  
      -> USER()  
      -> \c  
mysql>
```

Usando um banco de dados

- Verifique os bancos existentes no servidor com o comando SHOW:

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| mysql    |  
| test     |  
+-----+  
2 rows in set (0.01 sec)
```

Usando um banco de dados

- Para criar um novo banco de dados, utilize o comando “create database” <nome>:
 - `mysql> create database veterinario;`
- Para selecionar um banco de dados, utilize o comando “use” <nome>:
 - `mysql> use veterinario;`

Criando uma tabela

- Uma vez em que tenha selecionado um banco de dados, você pode examinar suas tabelas:

```
mysql> show tables;
```

```
Empty set (0.02 sec)
```

- Um conjunto vazio indica que não há tabelas neste banco.

Criando uma tabela

- Vamos criar uma tabela para guardar informações sobre animais de estimação.
- Tabela: pets
 - name: VARCHAR(20)
 - owner: VARCHAR(20)
 - species: VARCHAR(20)
 - sex: CHAR(1)
 - birth: DATE
 - date: DATE

Tipos de dados do MySQL

Consultar:

<http://dev.mysql.com/doc/refman/5.6/en/data-types.html>

Criando uma tabela

- Para criar a tabela, use o comando CREATE TABLE:

```
mysql> CREATE TABLE pet (  
    -> name VARCHAR(20) ,  
    -> owner VARCHAR(20) ,  
    -> species VARCHAR(20) ,  
    -> sex CHAR(1) ,  
    -> birth DATE, death DATE) ;  
Query OK, 0 rows affected (0.04 sec)
```

Exibindo as tabelas

- Para verificar se a tabela foi criada:

```
mysql> show tables;
```

```
+-----+
```

```
| Tables_in_test |
```

```
+-----+
```

```
| pet            |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

Examinando uma tabela

- Para examinar a estrutura de uma tabela, use o comando DESCRIBE:

```
mysql> describe pet;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
owner	varchar(20)	YES		NULL	
species	varchar(20)	YES		NULL	
sex	char(1)	YES		NULL	
birth	date	YES		NULL	
death	date	YES		NULL	

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

Apagando uma tabela

- Para apagar uma tabela completamente, use o comando DROP TABLE:

```
mysql> drop table pet;
```

```
Query OK, 0 rows affected (0.02 sec)
```

Carregando dados

- Use o comando INSERT para carregar dados em uma tabela.
- Por exemplo:

```
INSERT INTO pet VALUES  
    ('Fluffy', 'Harold', 'cat', 'f',  
    '1999-02-04', NULL);
```

- Complete a tabela como mostrado no próximo slide:

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

Carregando dados

- Você poderia criar um arquivo de texto 'pet.txt' contendo um registro por linha.
- Os valores devem estar separados por <TAB> e dispostos na ordem em que as colunas foram criadas pelo comando CREATE TABLE.
- Carregue então os dados usando o comando LOAD DATA.

Carregando dados

•Fluffy	Harold	cat	f	1993-02-04	\N
•Claws	Gwen	cat	m	1994-03-17	\N
•Buffy	Harold	dog	f	1989-05-13	\N
•Fang	Benny	dog	m	1990-08-27	\N
•Bowser	Diane	dog	m	1979-08-31	1995-07-29
•Chirpy	Gwen	bird	f	1998-09-11	\N
•Whistler		Gwen	bird	\N	1997-12-09 \N
•Slim	Benny	snake	m	1996-04-29	\N

- Para carregar estes dados na tabela pet:

```
mysql> LOAD DATA LOCAL INFILE "pet.txt" INTO TABLE pet;
```


Verifique a tabela criada:

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

Selecionando dados

- O comando SELECT é usado para recuperar informações de uma tabela.
- O formato geral é:

```
SELECT <o_que_selecionar>  
FROM <qual_tabela>  
WHERE <condições_a_satisfazer>;
```

Selecionando todos os dados

- A forma mais simples do SELECT recupera tudo que há em uma tabela:

```
mysql> select * from pet;
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1999-02-04	NULL
Claws	Gwen	cat	f	1994-03-17	NULL
Buffy	Harold	dog	f	1989-05-13	NULL
Fang	Benny	dog	m	1999-08-27	NULL
Bowser	Diane	dog	m	1998-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	NULL
Whistler	Gwen	bird		1997-12-09	NULL
Slim	Benny	snake	m	1996-04-29	NULL

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

Selecionando dados específicos

- Você pode selecionar linhas específicas.
- Por exemplo, para examinar somente o registro do “Bowser”, use WHERE <condição>:

```
mysql> SELECT * FROM pet WHERE name = "Bowser";
```

name	owner	species	sex	birth	death
Bowser	Diane	dog	m	1998-08-31	1995-07-29

1 row in set (0.00 sec)

Selecionando dados específicos

- Todos os animais nascidos após 1998:

```
SELECT * FROM pet WHERE birth >= "1998-1-1";
```

- Todas as cadelas (uso do AND):

```
SELECT * FROM pet WHERE species = "dog" AND sex = "f";
```

- Cobras ou pássaros (uso do OR):

```
SELECT * FROM pet WHERE species = "snake"  
OR species = "bird";
```

Selecionando dados específicos

- Da mesma forma que se podem limitar os registros (linhas) recuperados, podem-se visualizar apenas algumas colunas;
- Basta nomear as colunas desejadas após o `SELECT`, separando com vírgulas;
- Por exemplo, se quiser saber quando os animais nasceram, selecione apenas nome e data de nascimento.

Selecionando dados específicos

```
mysql> select name, birth from pet;
```

```
+-----+-----+
| name   | birth   |
+-----+-----+
| Fluffy | 1999-02-04 |
| Claws  | 1994-03-17 |
| Buffy  | 1989-05-13 |
| Fang   | 1999-08-27 |
| Bowser | 1998-08-31 |
| Chirpy | 1998-09-11 |
| Whistler | 1997-12-09 |
| Slim   | 1996-04-29 |
+-----+-----+
8 rows in set (0.01 sec)
```

Modificando dados

```
mysql> mysql> UPDATE pet
-> SET owner = 'Diana',
-> birth = '1989-08-31',
-> WHERE name = 'Bowser';
```

Query OK, 1 row affected (0.04 sec)

Rows matched: 1 Changed: 1 Warnings: 0;

```
mysql> SELECT * FROM pet WHERE name = "Bowser";
```

name	owner	species	sex	birth	death
Bowser	Diana	dog	m	1989-08-31	1995-07-29

1 row in set (0.00 sec)

Ordenando os registros

- Para ordenar os resultados, use a condição ORDER BY;
- Por exemplo, para ver as datas de nascimento ordenadas por data:

```
mysql> SELECT name, birth FROM pet ORDER BY birth;
```

```
+-----+-----+
| name      | birth      |
+-----+-----+
| Buffy      | 1989-05-13 |
| Claws      | 1994-03-17 |
| Slim       | 1996-04-29 |
| Whistler   | 1997-12-09 |
| Bowser     | 1989-08-31 |
| Chirpy     | 1998-09-11 |
| Fluffy     | 1999-02-04 |
| Fang       | 1999-08-27 |
+-----+-----+
```

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

Ordenando os registros

- Para reverter a ordem, use DESC.

```
mysql> SELECT name, birth FROM pet ORDER BY birth DESC;
```

```
+-----+-----+
| name      | birth      |
+-----+-----+
| Fang       | 1999-08-27 |
| Fluffy     | 1999-02-04 |
| Chirpy     | 1998-09-11 |
| Bowser     | 1989-08-31 |
| Whistler   | 1997-12-09 |
| Slim       | 1996-04-29 |
| Claws      | 1994-03-17 |
| Buffy      | 1989-05-13 |
+-----+-----+
8 rows in set (0.02 sec)
```

Trabalhando com registros Nulos

- O valor NULL preenche campos de valores faltantes ou desconhecidos.
- Para testar se um campo é NULL, não é possível usar operadores aritméticos como =, < ou <>.
- Ao invés disso, é necessário usar IS NULL e IS NOT NULL.

Trabalhando com registros Nulos

- Por exemplo, para achar os animais que já morreram:

```
mysql> select name from pet where death IS  
      NOT NULL;
```

```
+-----+
```

```
| name |
```

```
+-----+
```

```
| Bowser |
```

```
+-----+
```

```
1 row in set (0.01 sec)
```

Pattern Matching

- MySQL provê:
 - Pattern matching padrão do SQL; e
 - Pattern matching por expressões regulares, similar àqueles disponíveis no Linux como “grep” e “sed”.
- SQL Pattern matching:
 - Para usar pattern matching, use os operadores de comparação LIKE ou NOT LIKE;
 - Por padrão, padrões não diferenciam maiúsculas de minúsculas.
 - Caracteres especiais:
Para substituir um caracter simples: _
Para substituir um número arbitrário de caracteres: %

Pattern Matching

- Nomes que começam com 'b':

```
mysql> SELECT * FROM pet WHERE name LIKE "b%";
```

name	owner	species	sex	birth	death	
Buffy	Harold	dog	f	1989-05-13	NULL	
Bowser	Diana	dog	m	1989-08-31	1999-07-29	

Pattern Matching

- Nomes que terminam com `fy`:

```
mysql> SELECT * FROM pet WHERE name LIKE "%fy";
```

name	owner	species	sex	birth	death	
Fluffy	Harold	cat	f	1993-02-04	NULL	
Buffy	Harold	dog	f	1989-05-13	NULL	

Pattern Matching

- Nomes que contenham um 'w':

```
mysql> SELECT * FROM pet WHERE name LIKE "%w%";
```

name	owner	species	sex	birth	death
Claws	Gwen	cat	m	1994-03-17	NULL
Bowser	Diana	dog	m	1989-08-31	1999-07-29
Whistler	Gwen	bird	NULL	1997-12-09	NULL

Pattern Matching

- Nomes que contenham exatamente 5 caracteres:

```
mysql> SELECT * FROM pet WHERE name LIKE "_____";
```

name	owner	species	sex	birth	death	
Claws	Gwen	cat	m	1994-03-17	NULL	
Buffy	Harold	dog	f	1989-05-13	NULL	

Matching por Expressões Regulares

- O outro tipo de pattern matching disponível no MySQL utiliza expressões regulares.
- Quando você testa por um casamento (match) para este tipo de padrão (pattern), utiliza os operadores REGEXP e NOT REGEXP (ou RLIKE e NOT RLIKE, que são sinônimos).
- Algumas características do Matching por expressões regulares:
 - Qualquer caractere singular: `.`
 - Uma classe de caracteres: `[...]`
 - Por exemplo, `[abc]` recupera a, b, ou c. Para atingir uma faixa de caracteres, use um traço. `[a-z]` recupera qualquer letra em minúscula, e `[0-9]` recupera qualquer dígito.
 - Zero ou mais ocorrências do que o preceder: `*`
 - Por exemplo, `x*` recupera qualquer quantidade de caracteres “x”, e `[0-9]*` recupera qualquer quantidade de dígitos. `.*` recupera qualquer quantidade de qualquer coisa 😊
 - Para “ancorar” um padrão, de forma que ele valha para o início ou o fim de um valor testado, use: `^` (início) ou: `$` (fim).

Matching por Expressões Regulares

- Exemplo: Nomes que começam com b:

```
mysql> SELECT * FROM pet WHERE name REGEXP "^b";
```

name	owner	species	sex	birth	death
Buffy	Harold	dog	f	1989-05-13	NULL
Bowser	Diana	dog	m	1989-08-31	1999-07-29

Matching por Expressões Regulares

- Exemplo: Nomes que terminam com `fy`:

```
mysql> SELECT * FROM pet WHERE name REGEXP "fy$";
```

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	NULL
Buffy	Harold	dog	f	1989-05-13	NULL

Contando registros

- Uma questão comum em consultas é do tipo: “Quantas vezes este valor ocorre em uma tabela?”
- Por exemplo, você pode querer saber quantos animais de estimação um determinado dono possui.
- A função COUNT() conta o número de registros não nulos.

Contando registros

- Contando o número total de animais na tabela:

```
mysql> SELECT COUNT(*) FROM pet;
```

```
+-----+  
| COUNT(*) |  
+-----+  
|          9 |  
+-----+
```

Usando um arquivo de comandos (batch mode)

- Nos exemplos anteriores, você utilizou mysql interativamente para realizar consultas e visualizar os resultados.
- Você também pode escrever suas consultas em um arquivo texto e rodar o mysql em modo “batch”:
- **`shell> mysql < batch-file`**