Banco de Dados

"Ou não comece ou, tendo começado, não desista."
(Provérbio Chinês)

NOTAS DE AULA

(Laboratório)

Arquivo 1

Rafael Dias Ribeiro, M.Sc.
rafaeldiasribeiro@gmail.com.br

Sócio Efetivo da:

Sociedade Brasileira de Computação

Interagindo com a Interface do MySQL

Ajuda do MySQL: help

Conectando-se ao MySQL : mysql –h servidor –u usuário –p (banco)

Desconectando-se de um banco: QUIT

Exemplos de comandos:

```
SELECT VERSION( );

SELECT CURRENT_DATE;

SELECT SIN(PI( )/4);

SELECT VERSION( ), CURRENT_DATE;
```

Interagindo com a Interface do MySQL

Múltiplas Insrtuções em uma linha:

```
SELECT VERSION( ); SELECT NOW( );
```

As instruções em MySQL são sempre terminadas por " ; " exceto o QUIT e o USE, isto é, a instrução será aceita pelo MySQL quando for detectado o ; <ENTER> e não apenas <ENTER>

EX: SELECT
USE( )
;

é igual a SELECT USE( );

Significado do Prompt

mysql> Pronto para novo comando.

-> Esperando pela próxima linha de comando com múltiplas linhas.

'> Esperando pela próxima linha, coletando uma string que comece com uma aspas simples (‘’’).

"> Esperando pela próxima linha, coletando uma string que comece com aspas duplas (‘”’).

`> Esperando pela próxima linha, coletando uma string que comece com crase (‘`’).

\C Cancela o Comando

Ex:
mysql> SELECT * FROM minha_tabela WHERE nome = "Smith AND idade < 30;
"> "\c
mysql>

Consultando as Bases de Dados Existentes:

Utilize a instrução SHOW para saber quais bancos de dados existem atualmente no servidor:

Ex:

```
mysql> SHOW DATABASES;
+----------+
| Database |
+----------+
| mysql |
| test    |
| tmp    |
+----------+
```

Utilizando uma Base de dados Existente:

mysql> USE test
Database changed

OBS: Perceba que o USE, como o QUIT, não necessitam de um ponto e vírgula. (Você pode terminar tais declarações com uma ponto e vírgula se gostar; Isto não importa) A instruçãoo USE é especial em outra maneira, também: Ela deve ser usada em uma única linha.

Visualizando as tabelas de uma Base de Dados:

mysql> SHOW TABLES;
Empty set (0.00 sec) -> *SIGNIFICA QUE A BASE DE DADOS ESTÁ VAZIA*

Criando uma Base de Dados

Sintaxe:

mysql> CREATE DATABASE [IF NOT EXISTS] aula;

IF NOT EXISTS: Permite que o comando seja executado sem que o usuário tenha certeza de que a base de dados não exista. Caso exista, o comando é ignorado

O SGBD ao executar o comando de criação da Base de Dados cria um diretório com o nome dado. É neste diretório onde as tabelas criadas no database serão armazenadas.
Caminho padrão: c:\mysql\data\nome-database\>

OBS: No Unix, nomes de bancos de dados são caso sensitivo (ao contrário das palavras chave SQL), portanto você deve sempre fazer referência ao seu banco de dados como aula e nãoAula, AULA ou outra variação. Isto também é verdade para nomes de tabelas. (No Windows, esta restrição não se aplica, entiretanto você deve referenciar os bancos de dados e tabelas usando o mesmo caso em toda a parte da consulta.)

Exercício:
Crie o database aula.

mysql> CREATE DATABASE aula;

Selecionando uma Base para Uso

Criar um bancos de dados não o seleciona para o uso; Você deve fazer isso de forma explícita.

Para fazer uso do banco de dados atual, use o comando:

```
mysql> USE aula
Database changed
```

Excluindo a Base de Dados

DROP DATABASE apaga um banco de dados e todas suas tabelas, assim tenha **CUIDADO COM ESTE COMANDO** !

Sintaxe:
DROP DATABASE [IF NOT EXISTS] <nome-database>;

Exemplo:
mysql> DROP DATABASE aula;

Criando uma Tabela

CREATE TABLE para especificar o layout de sua tabela:

SINTAXE:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] <nome da tabela>
( <nome-atributo> <tipo-dado> [NULL | NOT NULL] [DEFAULT <default-value>]
[AUTO_INCREMENT] [PRIMARY KEY] )
{ENGINE | TYPE}=engine_name
AUTO_INCREMENT = value
COMMENT='string'
MAX_ROWS = value
MIN_ROWS =  value
PACK_KEYS = {0 | 1 | default}
ROW_FORMAT = {default | dynamic | fixed | compressed | redundant | compact }
```

Existem diversos atributos de configuração que não são abordados neste material. Para maiores detalhes utilize o guia de referência oficial do produto disponível no site do fabricante (http://www.mysql.com)

Criando uma Tabela

SINTAXE:

Uma outra forma de especificar uma nova tabela é utilizando uma tabela já existente como modelo

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] <nome da tabela>
(LIKE <nome da tabela existente>)

Criando uma Tabela

Ex:

```
mysql>    CREATE TABLE IF NOT EXISTS funcionario (
          cod_funcionario INT(4) NOT NULL AUTO_INCREMENT,
          nome_funcionario VARCHAR(5) NOT NULL DEFAULT '...',
          INDEX idxFuncionarioNome(nome_funcionario),
          PRIMARY KEY (cod_funcionario))
          TYPE = MYISAM
          MIN_ROWS=0
          MAX_ROWS=9000
          AUTO_INCREMENT=1
          PACK_KEYS=DEFAULT
          ROW_FORMAT=DEFAULT
          COMMENT='Cadastro de funcionarios';

mysql>    commit;
```

Criando uma Tabela

CREATE TABLE IF NOT EXISTS funcionario (

IF NOT EXISTS: Mesma definição apresentada no tópico de CRETE DATABASE

cod_funcionario INT(4) NOT NULL AUTO_INCREMENT,

NOT NULL / NULL : Colunas *not null* não podem ser deixadas vazias, quando uma nova linha é inserida.

AUTO_INCREMENT : Usando uma coluna tem a propriedade *auto_increment*, ela funciona como um índice. A cada inserção na tabela, o número dessa coluna é aumentado. As colunas *auto_increment* precisam ser *not null* e precisam ser número inteiros (int, longint, mediumint ou tinyint). Por serem automáticos, os dados dessa coluna não podem ser alterados.

Criando uma Tabela

nome_funcionario VARCHAR(5) NOT NULL DEFAULT '...',

DEFAULT :Define um valor padrão que será atribuido ao campo caso não seja alterado no processo de entrada de dados

PRIMARY KEY (cod_funcionario)

PRIMARY KEY : Uma *primary key* (chave primária) é um campo de identificação única do registro. Um campo definido como *primary key* não pode ter um falor nulo ou duplicado.

OBS: Um atributo pode ser definido como chave primária colocando a instrução primary key em sua linha de declaração

Ex:

```
CREATE TABLE IF NOT EXISTS funcionario (
cod_funcionario INT(4) NOT NULL AUTO_INCREMENT PRIMARY KEY,
nome_funcionario VARCHAR(5) NOT NULL DEFAULT '...',
```

Criando uma Tabela

TYPE = MYISAM
Define o tipo de tabela com MyIsam.

MIN_ROWS=0
Define que poderá não ter registros

MAX_ROWS=9000
Define que terá no máximo 9000 registros

AUTO_INCREMENT=1
Define o valor inicial para o auto increment como 1

PACK_KEYS=DEFAULT
Define a compactação da chave de acordo com o padrão

ROW_FORMAT=DEFAULT
Define o formato do registro de acordo com o padrão

COMMENT='Cadastro de funcionarios';
Informa o propósito da tabela (comentários sobre a tabela)

Os itens de configuração da tabela não são obrigatórios.
Caso não sejam especificados, o MySQL utilizará os valores padrão que estão no arquivo de configuração do produto (my.ini)

Criando uma Tabela

Considerações:

O MySQL representa cada tabela com a extensão **.frm,** dentro do diretório de banco de dados onde ela foi criada. De acordo com o tipo de estrutura de tabela, outras extensões podem ser criadas para tabela.Tabelas do tipo MyIsam criam extensões . FRM, .MYI e .MYD

Criando uma Tabela

Exercício:

Crie a tabela carros dentro do database aula

```
mysql> CREATE TABLE carros (
            placa VARCHAR(7),
            marca VARCHAR(10),
            modelo VARCHAR(15),
            compra DATE,
            venda DATE);
```

```
mysql> SHOW TABLES;
+------------------+
| Tables in aula   |
+------------------+
| carros           |
+------------------+
```

Visualização da Estrutura da Tabela

Para se visualizar a estrutura de uma tabela já criada, utilisa-se o comando DESCIBE OU DESC.
Sintaxe:

DESCRIBE <nome-tabela>;
DESC <nome-tabela>;

Exercício:

Exiba a estrutura da tabela carros.

```
mysql> DESCRIBE carros;
+---------+------------+------+-----+---------+-------+
| Field   | Type       | Null | Key | Default | Extra |
+---------+------------+------+-----+---------+-------+
| placa | varchar(7) | YES |       | NULL |        |
| marca |varchar(10) | YES |       | NULL |        |
| modelo | varchar(15) | YES |    | NULL |        |
|compra | date        | YES |      | NULL |        |
| venda| date         | YES |      | NULL |        |
+---------+------------+------+-----+---------+-------+
```

Uma alternativa ao comando describe
é a utilização do comando SHOW

Ex:
SHOW COLUMNS FROM <nome-tabela>

Alterando a Estrutura da Tabela

o comando ALTER TABLE permite alterar a estrutura da tabela existente. Por exemplo, você pode adicionar ou excluir colunas, criar ou remover índices, alterar o tipo de coluna existentes, ou renomear coluna ou tabelas.

Sintaxe:

ALTER TABLE <nome-tabela> <especificação da alteração> [<especificação da alteração>,...]

Alterando a Estrutura da Tabela

Adicionando Atributos:

ALTER TABLE <nome-tabela> ADD <atributo>[<atributo>,...] [AFTER | BEFORE] <atributo>;

Ex: Adicionar um atributo 'at3' do tipo integer 4 posições depois do atributo 'at2' na tabela exemplo

```
mysql> ALTER TABLE exemplo ADD at3 INTEGER(4) AFTER at2;
```

Modificando Atributos:

ALTER TABLE <nome-tabela> MODIFY <atributo>[<atributo>,...] [AFTER | BEFORE]<atributo>;

Ex1:Modificar um atributo 'at3' do tipo integer 4 para TINYINT NOT NULL (mantendo o mesmo atributo) e alterar o atributo 'at2' do tipo CHAR de 10 para 20 posições e modificando seu nome para 'at21'

```
mysql> ALTER TABLE exemplo MODIFY at3 TINYINT NOT NULL, CHANGE at2 at21(20);
```

Alterando a Estrutura da Tabela

Modificando Atributos:

Ex2: O comando básico para alteração de atributos e nome de colunas é CHANGE, seguida da denominação da coluna a ser alterada e dos novos atributos. Para mudar os atributos da coluna fone, utilizaremos a seguinte query:

```
ALTER TABLE carros CHANGE marca marca char(30);
/* Note que alteramos o tamanho do CHAR */
```

Vocês devem ter percebido que a palavra ‘marca' foi utilizada duas vezes. Isso ocorre porque se indica primeiro a coluna e depois seus novos atributos, e o nome da coluna é um de seus atributos.

Supomos que queiramos agora somente mudar o nome da coluna, e manter seus demais atributos:

```
ALTER TABLE carros CHANGE marca fabricante char(30);
```

Veja as alterações com DESCRIBE clientes;

Alterando a Estrutura da Tabela

Excluindo Atributos:

ALTER TABLE <nome-tabela> DROP <atributo>[<atributo>,...]

Ex: Excluir um atributo de nome 'at3'

mysql> ALTER TABLE exemplo DROP COLUMN ;

Exercício:

Adicione o atributo cor após o atributo marca, na tabela carro

mysql> ALTER TABLE carro ADD cor varchar(15) after marca;

Utilize o comando de visualização de estrututa para verificar a alteração

mysql> DESC carro;

Alterando a Estrutura da Tabela

Renomeando Tabelas

Sintaxe:

ALTER TABLE <nome-tabela> RENAME <novo-nome>;

Utiliza tabelas temporárias enquanto o comando RENAME TABLE não

Ex:
mysql> ALTER TABLE tabela RENAME tabelanova;

Ou utiliza-se o comando RENAME TABLE

Sintaxe:

RENAME TABLE <nome-tabela> TO <novo-nome> [,<nome-tabela> TO <novo-nome>,...]

Ex: RENAME TABLE tabela TO tabelanova;

Deletando Tabela(s)

DROP TABLE remove uma ou mais tabelas. Todos os dados e definições de tabela são *removidos*, assim tenha **CUIDADO COM ESTE COMANDO !**

Sintaxe:

DROP TABLE [IF EXISTS] <nome-tabela> [,<nome-tabela>,...] [RESTRICT | CASCATE]

[RESTRICT | CASCATE] -> Até a versão 5.0 não tinham funcionalidade implementada

mysql> DROP TABLE nomedatabela;