

MYSQL - PRIMEIROS COMANDOS

CRIAÇÃO DO BD E DAS TABELAS, INSERÇÃO E CONSULTA DE REGISTROS

1. Manuseio do banco de dados

As seguintes instruções manuseiam o banco de dados:

cria o banco	create database [if not exists] <nome>
mostra os bancos existentes	show databases
coloca um BD em uso	use <nome>
exclui um BD	drop database <nome>

Para testar estas instruções, digite as instruções abaixo, no prompt do MySQL. No Windows, podem ser usadas maiúsculas e minúsculas sem problema; no Linux, não. Alguns autores adotam a convenção de escrever todos os comandos SQL em maiúsculas.

```
show databases;  
create database teste;  
show databases;
```

Alguns BDs listados são privativos do MySQL: são usados para guardar as informações sobre os BDs criados pelos usuários. Abra o Meu Computador e faça o caminho onde está instalado o MySQL e verifique na pasta Data a criação do BD *teste*.

Para verificar o erro e a criação de um BD já existente, digite:

```
create database teste;  
show databases;  
create database if not exists teste;  
show databases;
```

Coloque o BD teste em uso, digite: *use teste*; Apague o BD, digite: *drop database teste*; Depois, verifique novamente os BDs existentes, com *show databases*; Abra o Meu Computador e faça o caminho onde está instalado o MySQL e verifique na pasta Data a exclusão do BD *teste*.

2. Criação de tabelas

A tabela para o cadastro de funcionários deve se chamar *cadfun*, utiliza alguns dos principais comandos da linguagem SQL e tem a seguinte estrutura:

<i>Campo</i>	<i>Tipo*</i>	<i>Descrição</i>
CodFun**	integer	código do funcionário (não nulo)
Nome	varchar(40)	nome do funcionário (não nulo)
Depto	char (2)	departamento onde está locado o funcionário
Funcao	char (20)	função do funcionário
Salario	decimal (10, 2)	salário do funcionário

* os tipos de dados possíveis no MySQL estão na tabela anexa TIPOS DE DADOS

** chave primária.

Antes de criar a tabela, vamos criar um novo BD *virtual*, que conterà a maior parte dos dados que usaremos nas lições seguintes. Para isto, digite o seguinte:

```
create database virtual;
use virtual;
```

Para criar a tabela, digite o comando abaixo. Compare a sintaxe do comando digitado com a descrição da estrutura da tabela. Para efeito didático, a instrução foi digitada em várias linhas para destacar a endentação; observe a colocação das vírgulas e parênteses e que o ; (ponto e vírgula) encerra a instrução.

```
create table CadFun (
    CODFUN integer not null primary key,
    NOME varchar(40) not null,
    DEPTO char( 2),
    FUNCAO char(20),
    SALARIO decimal(10, 2)
);
```

Para mostrar as tabelas do BD em uso, digite: *show tables;*

Para mostrar a estrutura da tabela, ou de apenas um campo, digite:

```
describe CadFun;
describe CadFun CODFUN;
```

3. Inserção de registros

A inserção de registros pode ser feita pelos comandos INSERT e pelo comando LOAD DATA. O primeiro insere diretamente os registros na tabela, após a execução de cada *Insert*; já *Load data* permite inserir de uma vez os comandos gravados em um arquivo. *Insert* tem o seguinte formato (atenção com os parênteses):

```
INSERT INTO <tabela>
(campo1, campo2, campo3..., campoN)
VALUES (valor1, valor2, valor3..., valorN);
```

Insira o primeiro registro. Observe os parênteses, as vírgulas e as aspas para os campos tipo char, bem como o ponto decimal e o ; (ponto e vírgula).

```
INSERT INTO cadfun
(CODFUN, NOME, DEPTO, FUNCAO, SALARIO)
VALUES (12, 'CARLOS ALBERTO', '3', 'VENDEDOR', 1530.00);
```

Só por castigo, ou melhor, para treinar um pouco mais, acrescente mais dois registros:

<i>codfun</i>	<i>nome</i>	<i>depto</i>	<i>funcao</i>	<i>salario</i>
15	MARCOS HENRIQUE	2	GERENTE	1985.75
7	APARECIDA SILVA	3	SECRETARIA	1200.50

Quando um campo deva ser inserido com NULL, basta suprimir o nome deste campo da lista de campos. Na nossa tabela, os campos CodFun e Nome não podem ser nulos. Por exemplo, o comando a seguir insere um registro sem informar o campo FUNCAO.

```
INSERT INTO cadfun
(CODFUN, NOME, DEPTO, SALARIO)
VALUES (44, 'JACIBA DA SILVA', '3', 1500.00);
```

Quando estiver inserindo todos os campos do registro e na sua ordem correta, você pode suprimir a primeira parte da instrução. Insira mais dois registros digitando o seguinte:

```
INSERT INTO cadfun VALUES (
    2, 'WILSON DE MACEDO', '3', 'PROGRAMADOR', 1050.00);
INSERT INTO cadfun VALUES (
    5, 'AUGUSTO SOUZA', '3', 'PROGRAMADOR', 1050.00);
```

4. Consulta de registros

A consulta pode ser feita com a seguinte instrução:

```
SELECT [tipo] <campos> FROM <tabela> [condição];
```

- Tipo: pode ser ALL (todos os registros) ou DISTINCT (registros distintos).
- Campos: é uma lista dos campos, separada por vírgula; caso se use * (asterisco), deseja-se ver todos os campos.
- Condição: é dada por outras cláusulas WHERE, ORDER BY..., etc, que serão detalhadas mais à frente.

Para ver todos os funcionários cadastrados, digite: *SELECT * FROM cadfun;*

Para ver somente o nome e a função, digite: *SELECT NOME, FUNCAO FROM cadfun;*

Para ver quem trabalha em um departamento, digite:

```
SELECT NOME, CODFUN FROM cadfun  
WHERE DEPTO = '3';
```

Para mudar a ordem da consulta, ascendente e descendente, digite estes exemplos:

```
SELECT NOME, SALARIO FROM cadfun  
ORDER BY NOME;
```

```
SELECT NOME, SALARIO FROM cadfun  
ORDER BY NOME DESC;
```

```
SELECT NOME, SALARIO FROM cadfun  
ORDER BY NOME DESC;
```

```
SELECT DEPTO, NOME FROM cadfun  
ORDER BY DEPTO, NOME DESC;
```

```
SELECT NOME FROM cadfun  
WHERE DEPTO = '3'  
ORDER BY NOME;
```

5. Alteração de registros

A alteração de registros pode ser feita com a seguinte instrução:

```
UPDATE <tabela> SET <campo1> = <expressão1>, <campo2> = <expressão2>, ...,  
<campoN> = <expressãoN>  
[condição];
```

- *Campo*: nome da coluna da tabela que se deseja atualizar. *Expressão*: valor que se deseja atribuir ao campo. A mesma instrução UPDATE pode alterar vários campos do registro.
- *Condição*: é dada pela cláusula WHERE.

Vamos alterar o salário do funcionário 7 e transferir a Ana Bastos do departamento 5 para o 3. Os SELECT mostram o estado antes e depois dos UPDATE.

```
SELECT * FROM cadfun WHERE CODFUN = 7;
```

```
UPDATE cadfun SET SALARIO = 2300.56 WHERE CODFUN = 7;
```

```
SELECT * FROM cadfun WHERE CODFUN = 7;
```

```
SELECT * FROM cadfun WHERE NOME = 'ANA BASTOS';
```

```
UPDATE cadfun SET DEPTO = '3' WHERE NOME = 'ANA BASTOS';
```

```
SELECT * FROM cadfun WHERE NOME = 'ANA BASTOS';
```

Vamos também conceder um reajuste de 10% para todos os empregados. Digite:

```
SELECT * FROM cadfun;
```

```
UPDATE cadfun SET SALARIO = SALARIO * 1.10;
```

```
SELECT * FROM cadfun;
```

6. Exclusão de registros

A exclusão de registros pode ser feita com a seguinte instrução:

```
DELETE FROM <tabela> [condição];
```

- *Condição*: é dada pela cláusula WHERE. Atenção!, pois se esta cláusula for omitida, é feita a exclusão de todos os registros da tabela.

Vamos excluir o departamento 2 inicialmente e depois o empregado Marcelo Souza. Os SELECT mostram o estado antes e depois dos DELETE.

```
SELECT * FROM cadfun;
```

```
SELECT DISTINCT DEPTO FROM cadfun;
```

```
DELETE FROM cadfun WHERE DEPTO = '2';
```

```
SELECT * FROM cadfun;
```

```
SELECT DISTINCT DEPTO FROM cadfun;
```

```
DELETE FROM cadfun WHERE NOME = 'MARCELO SOUZA';
```

```
SELECT * FROM cadfun;
```

Embora possamos excluir todos os registros de uma tabela com DELETE, a instrução abaixo faz isto com maior eficiência:

```
TRUNCATE TABLE <tabela>;
```

7. Alteração da estrutura da tabela

A alteração da estrutura da tabela pode ser feita com a seguinte instrução:

```
ALTER TABLE <tabela> <operação>;
```

- *Operação*: permite adicionar e excluir colunas de uma tabela:
 - ADD <campo> <tipo>: insere uma nova coluna <campo>, informando em seguida seu <tipo> de dados,
 - DROP <campo>: elimina um campo da tabela.

Vamos colocar na tabela o campo ADMISSAO (data de admissão). Em seguida, vamos preencher os dados destes campos. No mySql o formato da data é: 'aaaa-mm-dd'; em outros SGBD este formato pode ser diferente. Os SELECT mostram o estado da tabela antes e depois.

```
SELECT * FROM cadfun;
```

```
ALTER TABLE cadfun ADD ADMISSAO DATE;
```

```
SELECT * FROM cadfun;
```

```
UPDATE cadfun SET ADMISSAO = '2006-01-15' WHERE CODFUN = 2;
```

```
UPDATE cadfun SET ADMISSAO = '1999-10-21' WHERE CODFUN = 3;
```

```
UPDATE cadfun SET ADMISSAO = '2004-10-21' WHERE CODFUN = 4;
```

```
UPDATE cadfun SET ADMISSAO = '2006-04-26' WHERE CODFUN = 5;
```

```
UPDATE cadfun SET ADMISSAO = '1980-05-10' WHERE CODFUN = 7;
```

```
UPDATE cadfun SET ADMISSAO = '1999-12-15' WHERE CODFUN = 9;
```

```
UPDATE cadfun SET ADMISSAO = '2000-12-21' WHERE CODFUN = 12;
```

```
UPDATE cadfun SET ADMISSAO = '2000-10-21' WHERE CODFUN = 25;
```

```
SELECT * FROM cadfun;
```

Vamos aproveitar e testar o campo de data. O primeiro select usa uma função para extrair o mês; o segundo usa >=. Digite:

```
SELECT NOME, ADMISSAO FROM cadfun WHERE MONTH(ADMISSAO) = 12;
```

```
SELECT NOME, ADMISSAO FROM cadfun WHERE ADMISSAO >= '2000-01-01';
```

Para preparar a tabela para os próximos exercícios, vamos recriar o departamento 2, readmitir um empregado e admitir outros três. Digite:

```
INSERT INTO cadfun VALUES (  
    15, 'MARCOS HENRIQUE', '2', 'GERENTE', 2184.33, '2006-05-25');
```

```
INSERT INTO cadfun VALUES (  
    20, 'AUDREY TOLEDO', '2', 'SUPERVISORA', 1700.00, '2006-07-05');
```

```
INSERT INTO cadfun VALUES (  
    22, 'SANDRA MANZANO', '2', 'ANALISTA', 2000.00, '2006-07-01');
```

```
INSERT INTO cadfun VALUES (  
    24, 'MARCIO CANUTO', '2', 'PROGRAMADOR', 1200.00, '2006-07-10');
```

```
SELECT * FROM cadfun;
```

A microempresa deve ter ficado com doze empregados.

CONSULTAS COM OPERADORES

1. Operadores aritméticos

Os operadores aritméticos (soma, subtração, multiplicação, divisão e resto) estão descritos na tabela anexa “tabela de operadores e funções matemáticas”. Os operadores podem ser usados dentro de um comando Select: eles criam um campo calculado dentro da consulta, mas não afetam os valores das tabelas. Verifique, após os exemplos abaixo, que a tabela permanece inalterada. Digite:

```
SELECT NOME, SALARIO FROM cadfun;
```

```
SELECT NOME, SALARIO + 100 FROM cadfun;
```

```
SELECT NOME, SALARIO FROM cadfun;
```

```
SELECT NOME, SALARIO * 1.20 FROM cadfun;
```

```
SELECT NOME, SALARIO FROM cadfun;
```

2. Operadores de comparação ou relacionais

Os operadores relacionais que podem ser usados no mySql constam na tabela a seguir.

<i>Operador</i>	<i>Descrição</i>
>	Maior do que
<	Menor do que
=	Igual a
<> ou !=	Diferente de
>=	Maior ou igual a
<=	Menor ou igual a
<=>	Igual para Null

Por exemplo, digite:

```
SELECT * FROM cadfun WHERE DEPTO = '5';
```

```
SELECT * FROM cadfun WHERE FUNCAO = 'VENDEDOR';
```

```
SELECT * FROM cadfun WHERE SALARIO <= 1700;
```

```
SELECT * FROM cadfun WHERE SALARIO > 1700 + 50;  
SELECT * FROM cadfun WHERE SALARIO > (1700 + 50);
```

3. Operadores lógicos

Os operadores lógicos conhecidos na programação também podem ser usados no mySql e sua notação consta na tabela a seguir.

<i>Operador</i>	<i>Operador</i>	<i>Descrição</i>
AND	&&	E – conjunção
OR		Ou – disjunção
NOT	!	Não – negação
XOR		Ou exclusivo

Por exemplo, digite:

```
SELECT * FROM cadfun  
  WHERE (DEPTO = '3') AND (FUNCAO = 'PROGRAMADOR');  
SELECT * FROM cadfun  
  WHERE (DEPTO = '3') OR (DEPTO = '5');  
SELECT * FROM cadfun  
  WHERE NOT (FUNCAO = 'VENDEDOR');  
SELECT * FROM cadfun  
  WHERE (DEPTO = '5') XOR (FUNCAO = 'PROGRAMADOR');
```

4. Operadores adicionais

Alguns operadores adicionais podem ser usados no mySql e sua notação e seu significado consta na tabela a seguir.

<i>Operador</i>	<i>Descrição</i>
IS NULL	Verifica se um campo está vazio.
BETWEEN	Verifica se um valor está dentro de um intervalo.
IN	Verifica se um valor está em um conjunto de valores.
LIKE	Busca valores semelhantes.

Para testar estes operadores, crie um novo campo (a quantidade de filhos) nas tabelas CadFun e Morto e depois verifique seus valores. Para isto, digite:

```
ALTER TABLE cadfun ADD FILHOS SMALLINT;  
ALTER TABLE morto ADD FILHOS SMALLINT;  
SELECT NOME, FILHOS FROM cadfun;  
SELECT CODFUN, NOME, FILHOS FROM cadfun  
  WHERE FILHOS IS NULL;  
SELECT CODFUN, NOME, FILHOS FROM cadfun  
  WHERE NOT FILHOS IS NULL;  
SELECT CODFUN, NOME, FILHOS FROM cadfun  
  WHERE FILHOS IS NOT NULL;  
SELECT CODFUN, NOME, FILHOS FROM cadfun  
  WHERE NOT (FILHOS IS NULL);
```

Vamos alterar os valores deste campo para melhorar os exemplos. Para isto, digite:

```
UPDATE cadfun SET FILHOS = 1 WHERE CODFUN = 2;
UPDATE cadfun SET FILHOS = 3 WHERE CODFUN = 3;
UPDATE cadfun SET FILHOS = 2 WHERE CODFUN = 5;
UPDATE cadfun SET FILHOS = 1 WHERE CODFUN = 9;
UPDATE cadfun SET FILHOS = 4 WHERE CODFUN = 20;
UPDATE cadfun SET FILHOS = 3 WHERE CODFUN = 25;
SELECT CODFUN, NOME, FILHOS FROM cadfun;
SELECT CODFUN, NOME, FILHOS FROM cadfun
    WHERE FILHOS IS NULL;
SELECT CODFUN, NOME, FILHOS FROM cadfun
    WHERE NOT (FILHOS IS NULL);
SELECT * FROM cadfun
    WHERE SALARIO BETWEEN 1700 AND 2000;
SELECT * FROM cadfun
    WHERE SALARIO NOT BETWEEN 1700 AND 2000;
SELECT * FROM cadfun
    WHERE DEPTO IN ('2', '3');
SELECT * FROM cadfun
    WHERE DEPTO NOT IN ('2', '3');
```

O operador LIKE é ainda mais poderoso, pois pode trabalhar com dois caracteres curinga:

% (percentagem) significa qualquer quantidade (0, 1, 2...) de caracteres combinando, (alguns BDs usam * asterisco),

_ (sublinhado) significa apenas um caractere igual.

Estes caracteres podem ser usados no início, meio e fim do campo, permitindo inúmeras situações e combinações. Alguns exemplos:

<i>Exemplo: Where...</i>	<i>Operação</i>
Salario like '11%'	Encontra valores que começam com 11.
Salario like '%8%'	Encontra valores que tenham 8 em qualquer posição.
Salario like '_0%'	Encontra valores que tenham 0 na segunda posição.
Salario like '1_%_ %'	Encontra valores que começam com 1 e tenham 3 caracteres de comprimento (?? no mínimo??)
Salario like '%6'	Encontra valor que termine com 6.
Salario like '_1%6'	Encontra valores que tenham 1 na segunda posição e termine com 6.

Digite os comandos a seguir para testar o operador Like:

```
SELECT * FROM cadfun WHERE NOME LIKE 'A%';
SELECT * FROM cadfun WHERE NOME LIKE '_A%';
SELECT * FROM cadfun WHERE NOME LIKE '%AN%';
SELECT * FROM cadfun WHERE SALARIO LIKE '%6';
SELECT * FROM cadfun WHERE SALARIO LIKE '_5%2';
SELECT * FROM cadfun WHERE NOME NOT LIKE '%AN%';
```

Agora digite os comandos a seguir completando de acordo com a tabela anterior, para verificar os exemplos apresentados:

```
SELECT NOME, SALARIO FROM cadfun WHERE SALARIO LIKE ...(complete de acordo com a tabela de exemplos).
```