

Manipulando a base de dados

- Uma base de dados pode ser manipulada com quatro operações básicas:
 - Incluir,
 - Apagar,
 - Alterar e
 - Pesquisar.
- Vale lembrar que como toda linguagem para computadores, o MySQL tem suas regras. Um erro de parênteses que seja pode resultar no inverso do que você espera. Portanto, fique atento a sintaxe de seus comandos.

Manipulando a base de dados

- **Inserindo registros**
 - Para se adicionar dados a uma tabela, usamos o comando INSERT, que diz por si só sua função, como o exemplo que segue:
 - `mysql>INSERT INTO teste VALUES (NULL, 'Elaine', 'elaine.brito@cotil.unicamp.br', '34444444');`

Observações:

- Todos os campos que contêm texto, ou seja, CHAR, VARCHAR, BLOB, TEXT, etc. têm de ficar entre apóstrofos
- Para campos do tipo número, não se usam apóstrofos.
- A entrada NULL em um campo do tipo auto-incremento, permite que o MySQL providencie o conteúdo deste campo de forma automática. No caso do primeiro campo, o valor será 1, no segundo 2, no terceiro 3 e assim consecutivamente.
- Se possuíssemos um campo DATE, a entrada NULL faria com que o valor gravado no registro se torne a data atual.

Observações:

- É importante lembrar-se sempre de passar para o comando INSERT um número de parâmetros igual ao número de campos na tabela que está recebendo os dados. Caso contrario, você obterá uma mensagem de erro.

Pesquisando registros

- As pesquisas no MySQL são feitas através do comando SELECT.

- Exemplo:

```
mysql>SELECT * FROM teste;
```

- Resultado:
 - Lista todos os campos(*) de todos os registros da tabela teste.

Pesquisando registros

- Se queremos ver apenas alguns campos da tabela, especificamos os nomes das colunas desejadas, separadas por virgulas.

- Exemplo:

```
mysql>SELECT codigo,nome FROM teste;
```

- Resultado:
 - Lista os campos codigo e nome de todos registros da tabela teste.

Pesquisando registros

- Ação:

```
mysql>SELECT * FROM teste WHERE  
      (nome = 'Elaine');
```

- Resultado:
 - Lista todos os registros da tabela teste que possui 'Elaine' no campo nome.

Alterando registros

- Ação:

```
mysql>UPDATE teste SET nome = 'Elaine Brito' WHERE nome = 'Elaine';
```

- Resultado: Procura na tabela um registro que contenha no campo nome o conteúdo 'Elaine', definido pelo comando WHERE. Encontrado o registro, ele é substituído pelo nome definido no comando SET, que é 'Elaine Brito'.

Apagando registros

- Ação:
 - mysql>DELETE FROM
teste WHERE (telefone =
‘34444444’);
- Resultado: Apaga da tabela teste todos os registros que têm o conteúdo ‘34444444’ no campo telefone.

Operadores Aritméticos:

- São responsáveis pela execução de operações matemáticas simples:

+	Adição
-	Subtração
*	Multiplicação
/	Divisão

Operadores Relacionais:

- São utilizados quando precisamos fazer comparações entre dois valores:

>	Maior que
<	Menor que
=	Igual a
<>	Diferente de
>=	Maior ou igual a
<=	Menor ou igual a

Operadores lógicos:

- AND (&&)
 - O operador lógico AND, ou E, deve ser usado em uma pesquisa que se deseja entrar dois valores.
 - O AND, verifica ambas as cláusulas da comparação, e só retorna algum valor se as duas tiverem uma resposta verdadeira.
 - Observe o exemplo:

```
mysql>SELECT * FROM teste WHERE (nome =  
        'ELAINE') AND (telefone = '34444444');
```

Esta pesquisa mostrara todos os registros que contém no campo nome o conteúdo 'Elaine', E (AND) no campo telefone, o conteúdo '34444444'.

Operadores lógicos:

- OR (||)
 - O operador lógico OR, ou OU, deve ser usado em uma pesquisa que se deseja entrar dois valores.
 - O OR, verifica ambas as cláusulas da comparação, e retorna valores se qualquer um dos membros obtiver resultado.

```
mysql>SELECT * FROM teste WHERE (nome  
= 'Elaine') OR (telefone = '34444444');
```

Esta pesquisa fará com que todos os resultados que contenham o conteúdo 'Elaine' no campo nome, OU telefone '34444444' sejam exibidos na tela.

Operadores lógicos:

- NOT (!)
 - O operador lógico NOT, ou NÃO, realiza uma pesquisa, excluindo valores determinados do resultado.
- `mysql>SELECT * FROM teste WHERE`
- `(nome != 'Elaine');`
- Esta pesquisa listará todos os registros da base de dados teste, NÃO (NOT) mostrando aqueles que possuem 'Elaine' como conteúdo do campo nome.

Operadores lógicos:

- ORDER BY
 - O operador lógico ORDER BY, ou ORDENAR POR, simplesmente lista os registros, colocando-os em ordem de acordo com o campo solicitado.
- `mysql>SELECT * FROM teste WHERE`
- `(nome = 'Elaine') ORDER BY telefone;`
- O resultado desta busca resultara em todos os registros contendo 'Elaine' no campo nome, e a listagem será organizada de acordo com a ordem do telefone.

ORDER BY

- ASC e DESC especificam o tipo de classificação e são, respectivamente, abreviações das palavras em ingles ascending e descending, ou seja, classificação crescente ou decrescente.
- Quando não especificamos nenhum , o padrão é ascendente

Exercícios

- Crie uma tabela Empregados como a seguir:

Campo	Tipo	Descrição
codigo	Integer	Código do funcionário(não nulo)
nome	Char(40)	Nome do funcionário (não nulo)
setor	Char(2)	Setor onde o funcionário trabalha
cargo	Char(20)	cargo do funcionário
salario	Decimal(10,2)	salário do funcionário
Chave Primária		Será o campo codigo

Inserção de registros

codigo	nome	setor	cargo	salario
1	Cleide Campos	1	Secretária	1000
3	Andreia Batista	6	Programadora	1500
4	Cristiano Souza	6	Programador	1500
6	Mario Souza	4	Analista	2200
7	Ana Silva	4	Secretária	1000
9	Silvia Soares	5	Supervisora	1650
10	José da Silva	1	Programador	1500
15	Manoel Batista	1	Projetista	2500
25	João Silva	4	Supervisor	1650

Exemplo:

Insert into empregados values (1,'Cleide Campos','1','secretaria',1000);

Listagem de registros

- Apresentar a listagem completa dos registros da tabela Empregados;
- Apresentar uma listagem dos nomes e dos cargos de todos os registros da tabela Empregados;
- Apresentar uma listagem dos nomes dos empregados do setor 1
- Listagem dos nomes e dos salários por ordem de nome (a-z)
- Listagem dos nomes e dos salários por ordem de nome em formato descendente (z-a)
- Listagem dos setores e nomes colocados por ordem do campo setor em formato ascendente e do campo nome em formato descendente.
- Listagem de nomes ordenados pelo campo nome em formato ascendente, dos empregados do setor 4.

Alteração de Registros

- O empregado de código 7 teve um aumento de salário para 2500.50.
- Andreia Batista foi transferida do departamento 5 para o departamento 3.
- Todos os empregados da empresa tiveram um aumento de salário de 20%.
- Todos os empregados do setor 1 foram demitidos , exclua-os.
- Mario Souza pediu demissão, exclua-o.

Modificação da estrutura de uma tabela

- Inserir na tabela Empregados o campo admissao que conterà a data de admissão dos empregados.
- Em seguida será necessário atualizar a tabela com as datas de admissão dos empregados ativos.

codigo	nome	seto	cargo	salario	admissao
3	Andreia Batista	3	Programadora	1800	2000-10-20
4	Cristiano Souza	6	Programador	1800	1999-09-10
7	Ana Silva	4	secret ria	3000.6	2005-05-15
9	Silvia Soares	5	Supervisora	1980	2008-04-25
25	João Silva	4	Supervisor	1980	2000-11-15

Praticando

- Apresente a listagem dos empregados que foram admitidos em 20/10/2000
- Apresente a listagem dos funcionários que foram admitidos após 01/01/2000
- O departamento 2 foi reaberto e admitiu-se os seguintes empregados:

codigo	nome	setor	cargo	salário	admissão
20	Aline Brito	2	Supervisora	1700.00	2009-09-05
22	Silvia Mendes	2	Gerente	2000.00	2009-09-01
24	Moacir Campos	2	Programador	2000.00	2009-09-10
26	Marco Silva	2	Programador	2000.00	2009-09-15

Exercícios com Operadores

- Apresentar nome e salário dos empregados que ganham acima de 1700.00(valor do salário) mais uma comissão de 50.00 (totalizando 1750.00).
- Listar os empregados do setor 5.
- Listar os empregados cujo cargo é programador.
- Listar empregados com salário até 2000.00

Exercícios - Operadores Lógicos

- Listar programadores do setor 2.
- Listar empregados que sejam supervisor ou supervisora.
- Listar empregados que não sejam gerentes.

Operadores auxiliares

- **Between**

- Definição de intervalos de valores para a cláusula where.

<expressão> [Not] BETWEEN <mínimo> and <máximo>

- **IN**

- Algumas vezes não é possível definir um intervalo sequencial de valores.

<expressão> [Not] IN <valor1,valor2,...,valorN>

Exercícios

- Listar empregados com salário entre 1700.00 e 2000.00
- Listar programadores e programadoras

Verificação de caracteres

- Para verificar seqüência de caracteres dentro de um campo do tipo string (char ou varchar), pode-se utilizar junto com a clausula where uma condição baseada no uso do operador LIKE.

<expressão> [NOT] LIKE <valor>

- Exemplos:
- ‘A%’ – começa com letra A
- ‘_A%’ – segunda letra do nome A
- ‘%AN%’ - possui AN em qualquer posição

Exercícios

- Listar empregados cujo nome comece com a letra A
- Listar empregados cujo nome tem a segunda letra A
- Listar empregados que tem a sequência AN em qualquer posição do nome.

Se vazio

- Uma ocorrência bastante útil é verificar a existência de campos que possuam valores em branco ou não. Para isso usa-se junto ao where o operador IS NULL.

<expressão> IS [NOT] NULL

Exemplo:

Select * from Empregados where nome is null

Funções Agregadas

- AVG() – média aritmética
- MAX() – Maior valor
- MIN() – Menor valor
- SUM() – Soma dos valores
- COUNT() – Número de valores
 - ALL- contagem dos valores não vazios
 - Distinct – contagem dos valores não vazios e únicos

Função([all]<expressão>/[distinct]<expressão>)

Exercícios

- Média aritmética dos salários de todos os empregados
- Média aritmética dos salários de todos os empregados do setor 3
- Soma dos salários de todos os empregados
- Soma dos salários de todos os empregados do setor 5
- Maior salário existente entre todos os empregados
- Menor salário existente entre todos os empregados
- Numero de empregados do setor 3
- Número de empregados que ganham mais que 2000.00
- Número de setores existentes no cadastro de empregados.