**SET**

Um campo interessante, que permite que o usuário faça uma escolha dado determinado número de opções.

Cada campo pode conter até‚ 64 opções.

Exemplo:

vicio SET(‘café’, ‘cigarro’) NOT NULL;

Neste exemplo este campo pode conter apenas os seguintes itens:

""

"cafe"

"cigarro"

"cafe,cigarro"

**ENUM**

Um campo com funcionalidade semelhante ao SET, com a diferença que apenas um valor pode ser escolhido.

Exemplo:

virtude ENUM("programar", "amar") NOT NULL;

Neste exemplo este campo pode conter os seguintes valores:

""

"programar"

"amar"

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

**Pesquisando registros**

Imagine que possuímos um carro de última geração, com todos os recursos que a tecnologia atual pode permitir. Contudo, nosso carro não tem roda, ou seja, não serve para nada, a não ser ostentar.

As pesquisas no MySQL são feitas através do comando SELECT. Este comando pode fazer desde uma simples e trivial pesquisa até uma pesquisa extremamente complexa.

Pesquisa SQL(Structure query Language) – Linguagem estrutural de pesquisa.

Exemplos:

\*: Todos ou tudo

?: 1 caracter

Ação:

mysql>SELECT \* FROM teste;

Resultado:

Lista todos registros da tabela teste.

Ação:

mysql>SELECT \* FROM teste WHERE (nome = "Ernesto");

mysql>Select \* from teste where(salário>=2000,00); selecionar todos da tabela teste que o salário for maior ou igual a 2000,00.

mysql>Select \* from teste where(salário!=2000,00);

quando o salário for diferente de 2000,00

&& - And – e

**||** Or – ou

Mysql>Select \* from teste where(salário!=2000,00) and (cidade=Blumenau);

2000,00 ou 2000.00

Lista todos da tabela teste com o salário diferente de 2000,00 e da cidade de Blumenau.

Mysql>Select \* from teste where(salário>=2000.00) or (cidade=Blumenau);

Resultado:

**Apagando registros**

Quando um registro não nos é mais útil, podemos apagá−lo, para que a tabela não contenha dados obsoletos.

Isto pode ser facilmente feito usando o comando DELETE, que tem o funcionamento contrário ao comando INSERT.

Ação:

mysql>DELETE FROM teste WHERE (telefone = 2744747);

Resultado:

Apaga da tabela teste todos os registros que têm o conteúdo "2744747" no campo telefone.

Mysql>Delete from teste where(telefone=2744747) or (nomeAluno=Alexandre Augusto de Moraes);

**Alterando registros**

Suponhamos agora que sua namorada trocou o telefone. A primeira providência, é alterar sua tabela para que você não possua dados desatualizados. Para tal, existe o comando UPDATE. Observe os exemplos:

Ação:

mysql>UPDATE teste SET nome = 'Ernesto Augusto’ WHERE nome = 'Ernesto';

Resultado:

Procura na tabela um registro que contenha no campo nome o conteúdo 'Ernesto Augusto', definido pelo comando WHERE. Encontrado o registro, ele é substituído pelo nome definido no comando SET, que é 'Ernesto Augusto'.

Operadores lógicos:

O MySQL suporta todas operações lógicas, estando abaixo listadas as mais conhecidas:

**AND (&&)**

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

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

mysql>SELECT \* FROM teste WHERE

mysql>(nome = "Aline") AND (telefone = 2728918);

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Esta pesquisa mostrara todos os registros que contém no campo nome e conteúdo "Aline", E (AND) no campo telefone, o conteúdo 2728988.

**OR (||)**

O operador lógico OR, ou OU, deve ser usado em uma pesquisa que se deseja entrar dois valores. O OR,

verifica ambas as clausulas da comparação, e retorna valores se qualquer um dos membros obtiver resultado.

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

mysql>SELECT \* FROM teste WHERE

mysql>(nome = "Aline") OR (telefone = 2728918);

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Este pesquisa fará com que todos os resultados que contenham o conteúdo "Aline" no campo nome, OU 2728918 sejam exibidos na tela.

**NOT (!)**

O operador lógico NOT, ou NÃO, realiza uma pesquisa, excluindo valores determinados do resultado.

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

mysql>SELECT \* FROM teste WHERE (nome != "Aline");

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

Esta pesquisa listara todos os registros da base de dados teste, NÃO (NOT) mostrando aqueles que possuem "Aline" como conteúdo do campo nome.

**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 (salario>2000.00) ORDER BY nomFunc;

−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

O resultado desta busca resultara em todos os registros da tabela teste listagem será organizada de acordo com a ordem de nomes dos funcionários. 