ULBRA

SISTEMAS DE INFORMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO WEB

Prof. Taciano Balardin
taciano@ulbra.edu.br

E-MAIL:
taciano@ulbra.edu.br

SITE DA DISCIPLINA:
http://www.taciano.pro.br/

SENHA:
@lpweb


SISTEMAS DE INFORMAÇÃO
SI
XIV SEMANA ACADÊMICA
07 A 10 DE OUTUBRO DE 2014
19:00 - AUDITÓRIO ULBRA
GetWay
07.10
STARTUPS DE ALTO IMPACTO
PALESTRANTE: GUILHERME MASSERONI
GETWAY - CANOAS
SKYDRONES
VANTs, Imagens e Dados Aéreos
08.10
VANTs: LEGISLAÇÃO E USO
PALESTRANTE: ULF BOGDAWA
SKYDRONES - PORTO ALEGRE
ULBRA
08.10
SISTEMA DE NAVEGAÇÃO AÉREA TELEGUIADO UTILIZANDO ANDROID
PALESTRANTE: MAIQUEL LUIZ
ULBRA - CACHOEIRA DO SUL
creativar
09.10
GUIA DE BOLSO DO EMPREENDEDOR INICIANTE
PALESTRANTE: CAIO SANTANA CEZAR
CREATIVAR - PORTO ALEGRE
SI
10.10
BATE-PAPO COM EGRESSOS
CURSO DE SISTEMAS DE INFORMAÇÃO
ULBRA - CACHOEIRA DO SUL
ORGANIZAÇÃO:
casi
CENTRO ACADÊMICO DE SISTEMAS DE INFORMAÇÃO
ULBRA CACHOEIRA DO SUL
UNIVERSIDADE LUTERANA DO BRASIL

Exercício $_GET

Crie uma página web com a seguinte estrutura:

- **index.php** → arquivo que contem o topo da página, o menu lateral (Início – Empresa – Produtos – Contato), uma área central para conteúdo e o rodapé do site.
- **inc_inicial.php** → arquivo com o conteúdo padrão da área central;
- **inc_empresa.php** → arquivo com o conteúdo do menu empresa que abre na área central;
- **inc_produtos.php** → arquivo com o conteúdo do menu produtos que abre na área central e apresenta uma listagem de produtos;
- **inc_contato.php** → arquivo com o conteúdo do menu contato que abre na área central;

Revisão de Comandos SQL

LINGUAGEM DE PROGRAMAÇÃO WEB
AULA 12

Prof. Taciano Balardin

Classificação dos Comandos SQL

- DDL - Linguagem de **Definição** de Dados;
- DCL - Linguagem de **Controle** de Dados;
- DTL - Linguagem de **Transação** de Dados;
- DML - Linguagem de **Manipulação** de Dados.

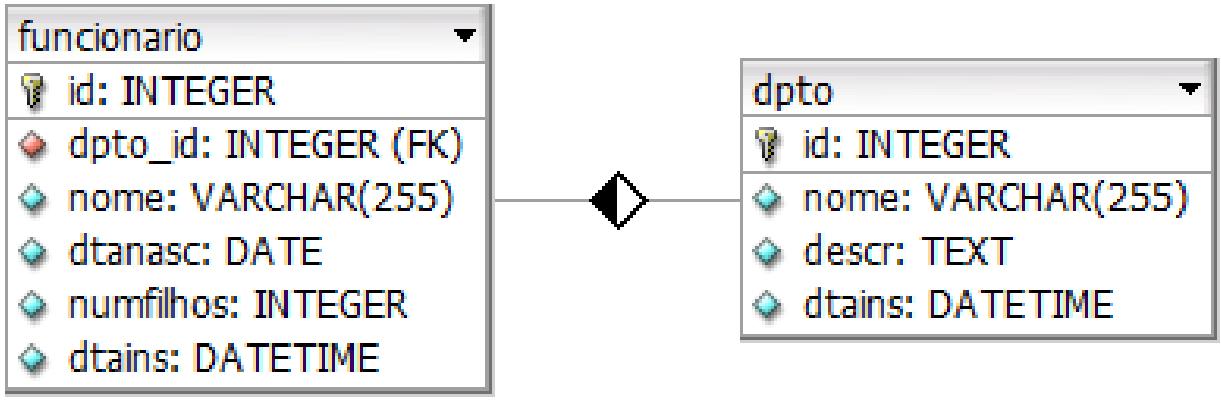
DDL - Data Definition Language

- Criação de Bases e Tabelas (CREATE)
- Alteração de Bases e Tabelas (ALTER)
- Remoção de Bases e Tabelas (DROP)

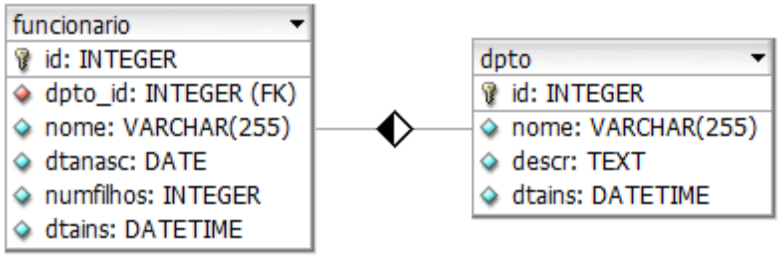
DML - Data Manipulation Language

- Inserção de registros (INSERT)
- Atualização de registros (UPDATE)
- Seleção de registros (SELECT)
- Remoçãode registros (DELETE)

Diagrama ER


funcionario
id: INTEGER
dpto_id: INTEGER (FK)
nome: VARCHAR(255)
dtanasc: DATE
numfilhos: INTEGER
dtains: DATETIME
dpto
id: INTEGER
nome: VARCHAR(255)
descr: TEXT
dtains: DATETIME

INSERT



INSERT INTO NomeTabela **(**campo1, campo2,...**)**

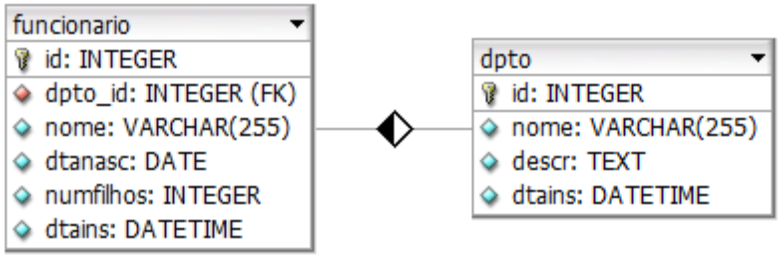
VALUES (valor1,valor2,...**)**

Os valores devem ser inseridos na ordem de declaração dos campos. Exemplo: 1º Valor para o 1º Campo

Se o campo for tipo que envolve Caracteres, então devem ser utilizadas aspas simples na especificação do valor.

INSERT INTO funcionario (nome, numfilhos) VALUES ('Fulano',3);

INSERT



INSERT INTO NomeTabela ~~**(campo1, campo2,...)**~~

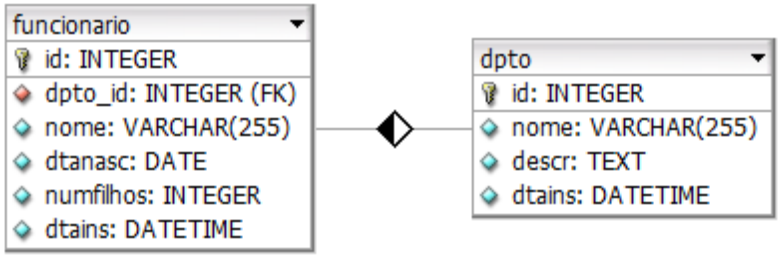
VALUES (valor1,valor2,...**)**

Não é obrigatório utilizar a sintaxe completa.

A declaração dos campos pode ser retirada, mas a quantidade de valores deve ser igual a quantidade de campos que existirem na tabela.

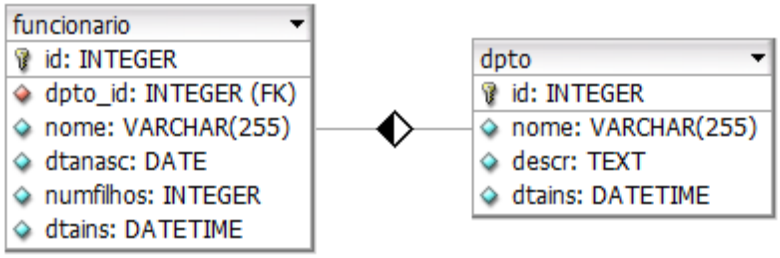
```
INSERT INTO funcionario
    VALUES (1,3,'Fulano',1990-01-01,2,now());
```

INSERT




```
INSERT INTO funcionario (nome, numfilhos)
VALUES   ('Fulano',2),
         ('Humatalde',1),
         ('Humatalde',0),
         ('Beltrano',3);
```

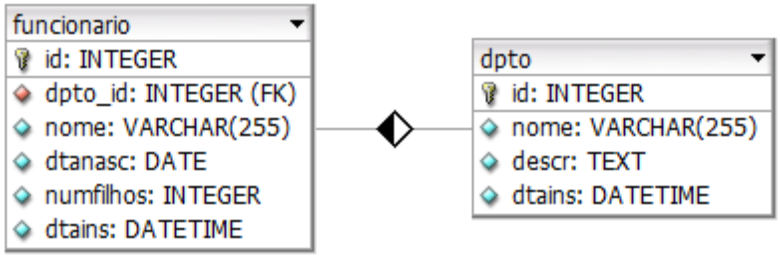
Cláusula WHERE



- Os demais comandos DML são normalmente utilizados com cláusulas que tem o papel de filtragem de registros.

- Por exemplo: listar somente os funcionários com mais de 2 filhos.

Cláusula WHERE



- Juntamente com o WHERE são utilizadas outras cláusulas de comparação e operação lógica.

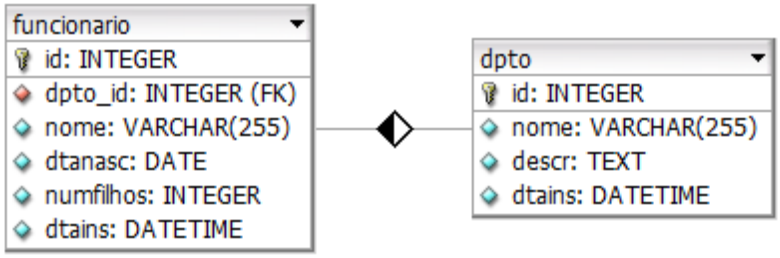
- Cláusulas de comparação:

  = > <

  => =< <> (ou !=)

  is null between (significa ENTRE, ou Intervalo)

Cláusula WHERE



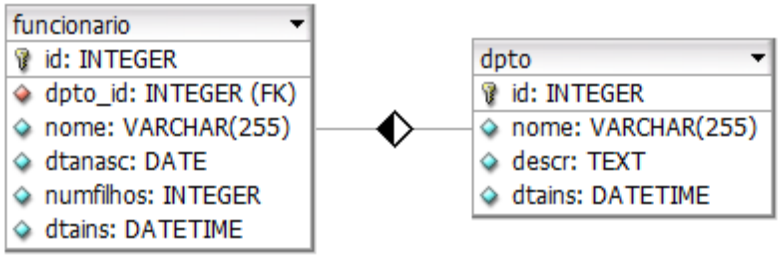
- Juntamente com o WHERE são utilizadas outras cláusulas de comparação e operação lógica.

- Operadores lógicos:

  AND

  OR

DELETE

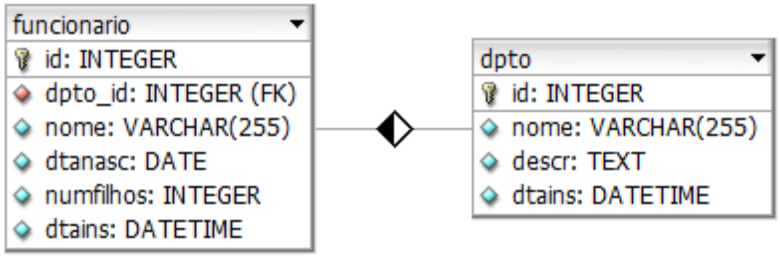


- Sintaxe completa:

DELETE FROM NomeTabela

Desta maneira <u>todos</u> os registros serão removidos. Para evitar que isto aconteça, utiliza-se a cláusula WHERE.

DELETE



EXEMPLOS:

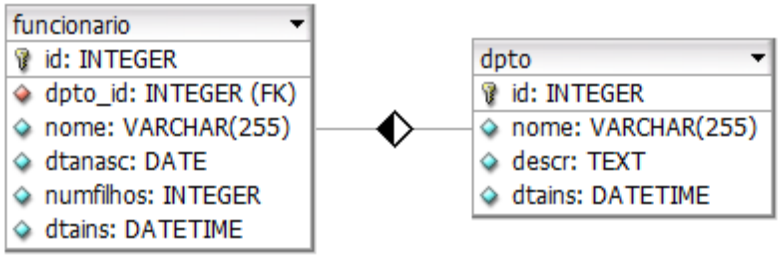
```
DELETE FROM funcionario WHERE dtanasc <= '1980-01-01';

DELETE FROM funcionario WHERE numfilhos BETWEEN 1 AND 3;

DELETE FROM funcionario WHERE numfilhos IS NULL;

DELETE FROM funcionario WHERE dtains = NOW( );
```

UPDATE

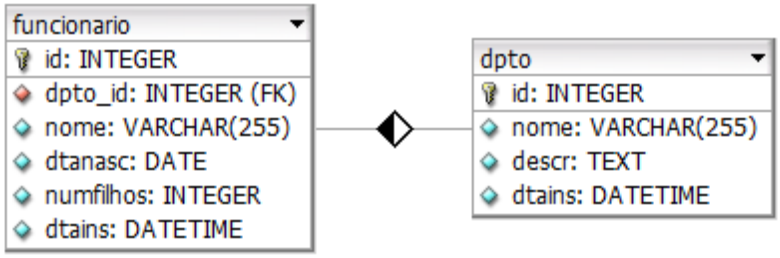


- Sintaxe padrão:

  UPDATE NomeTabela **SET** nomeCampo **=** valor

- Estes comandos atualizarão todos os registros da tabela.
- A cláusula WHERE pode ser utilizada na definição de quais campos devem ser afetados.

UPDATE



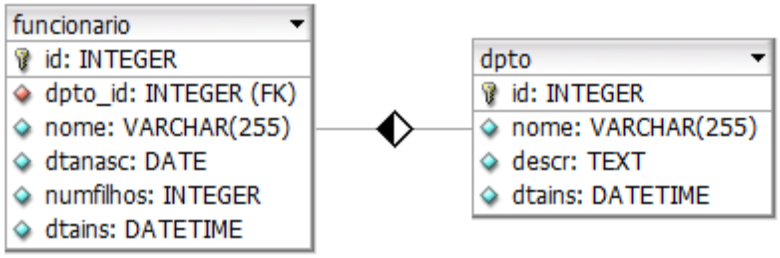
EXEMPLOS:

UPDATE funcionario **SET** numfilhos **=** 2 **WHERE** id **=** 16;

UPDATE dpto **SET** nome **=** 'TI' **WHERE** descricao **=** 'Tecnologia da Informação';

UPDATE funcionario **SET** dpto_id **=** 3 **WHERE** dtanasc **>=** '1990-01-01' and numfilhos **=** 0;

SELECT

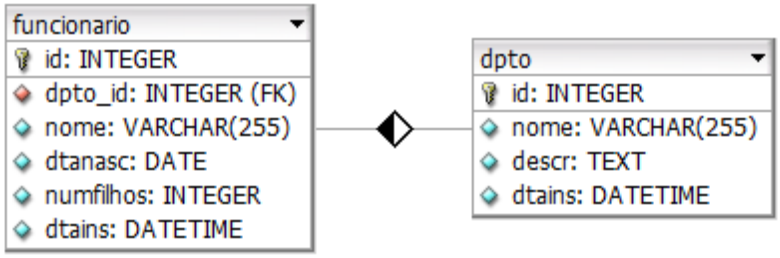


- Sintaxe padrão:

  SELECT colunas **FROM** NomeTabela

- O resultado do uso desta sintaxe é a seleção de todos os registros da Tabela.
- A cláusula WHERE também pode ser utilizada para seleção filtrada desses registros.

SELECT



EXEMPLOS:

Seleciona o nome dos dptos inseridos entre os dias 30/09/14 e 10/06/14:

SELECT nome **FROM** dpto **WHERE** dtains **between** '2014-09-30' **and** '2014-10-06';

Seleciona todos os campos da tabela Funcionário onde a FK é nula. Ou seja, todos os funcionários que não estão alocados em Departamento:

SELECT * **FROM** funcionario **WHERE** dpto_id **IS NULL**;

Seleciona somente o id dos Funcionários que tem filhos e estão alocados em algum Departamento:

SELECT id **FROM** funcionario **WHERE** numfilhos **>** 0 **AND** dpto_id **IS NOT NULL**;

Cliente MySQL