SQL – COMANDOS DML

Prof: Ricado Satoshi

A Linguagem SQL - DML

- Os comandos DML são responsáveis pela manipulação dos dados armazenados nas estruturas físicas criadas pelos comandos DDL.
  - INSERT
  - UPDATE
  - DELETE
  - SELECT

Comando Insert - Sintaxes

- INSERT INTO <nome_tabela> (<lista de colunas>) VALUES (<lista de valores>)
  - <lista de colunas> - é a relação das colunas cujos valores serão mencionados em VALUES
    - se a lista de colunas for omitida, os valores de **todas** as colunas deverão ser informados em VALUES na **mesma ordem** de criação da tabela

Outra forma…

- INSERT INTO <nome_tabela> SET <nome_coluna1> = <valor1>, <nome_coluna2>= <valor2>, ….

Nesse último caso, os valores são atribuidos um a um, em qualquer ordem

Supondo - Tabela Aluno

```
CREATE TABLE TABALUNO(
    matricula int,
    nome varchar(20) not null,
    sobrenome varchar(30) not null,
    sexo char(1) default ‘m’,
    datanasc date,
    nota decimal(5,2) default 0
);
```

Comando Insert - Exemplos

INSERT INTO TABALUNO (matricula, nome, sobrenome) VALUES (14,’Maria’, ’Rios’);

INSERT INTO TABALUNO (matricula, nome, nota) VALUES (16, ‘Francisco ‘, 7.5);

INSERT INTO TABALUNO VALUES (18, ‘Renato ‘ , ’Cardoso’, ‘m’, ‘1998-10-16’, 7.5);

INSERT INTO TABALUNO SET matricula=20, nome=‘Marta’, sobrenome=‘Mendonça’, sexo=‘f’, datanasc= ‘1990-12-25’;

Comando Update – Sinataxe e Exemplos

UPDATE <nome_tabela> SET
<coluna>=<novo_valor>,[<coluna2>=<valor2>]…
[WHERE <condição>]

Exemplos:

UPDATE TABALUNO SET nota=7.0;

UPDATE TABaluno SET sexo='f', datanasc ='31-12-1997'
WHERE matricula=14;

UPDATE TABEMPREGADO SET nota = nota+0.5
WHERE nota < 6;

Comando DELETE – Sinataxe e Exemplos

- DELETE FROM <nome_tabela> WHERE <condição>

Exemplos:

DELETE FROM TABALUNO WHERE matricula=20;

DELETE FROM INATIVOS WHERE Datademissao < ‘1970-01-01’

Comando SELECT

- Facilita o estabelecimento de critérios de seleção e busca que satisfaçam as necessidades da aplicação.

- Exemplo:

```
SELECT  nome FROM tabpessoas WHERE idade > 18
                                   AND salario < 1000
                                   AND sexo ='F'
```

Esta instrução pode ser traduzida por:

Selecione todos os NOMEs das mulheres presentes na tabela TABPESSOAS que tenham IDADE superior à 18 anos e SALÁRIO menor que R$ 1.000,00

Comando SELECT

- Se quisermos ver além do nome outras características como idade e salário, que modificações devem ser feitas no comando abaixo?

```
SELECT  nome FROM tabpessoas WHERE idade > 18
                                  AND salario < 1000
                                  AND sexo ='F'
```

```
SELECT  nome, idade, salario  FROM tabpessoas
        WHERE idade > 18 AND salario < 1000
        AND sexo ='F'
```

Comando SELECT do SQL - sintaxe

- É a construção mais largamente utilizada em programas (VB, Java, Etc.)
- **Função**: Selecionar um conjunto de registros em uma ou mais tabelas que atendam a uma determinada condição

Sintaxe:

SELECT

{ * } ouu

{ [**DISTINCT/DISTINCTROW]** <nome-coluna>, ... **}**

FROM <nome-tabela> [, <nome-tabela>]

{ [**WHERE** <condição>]

[**ORDER BY** <nome-coluna> [ASC /_DESC]]

[**GROUP BY** <nome-coluna>] [**HAVING** <condição>] **}**

* mais opções....

Comando SELECT do SQL - sintaxe

Detalhes da Sintaxe:

{ * } → Todos as colunas da tabela

{ [**DISTINCT/DISTINCTROW]** <nome-coluna>,. **}** → apenas os valores distintos (diferentes) da coluna escolhida

FROM <nome-tabela> [, <nome-tabela>] → Tabela(s) de origem

{ [**WHERE** <condição>] → Critério de seleção

[**ORDER BY** <nome-coluna> [ASC / DESC]] → Ordenação

[**GROUP BY** <nome-coluna>] [**HAVING** <condição>] **}** → Grupamento

Algumas Construções Possíveis

SELECT * FROM <nome_tabela>;
seleciona todas as colunas da tabela

SELECT <coluna1>, <coluna2>, ... FROM <nome_tabela>;
seleciona coluna1, coluna2 da tabela

SELECT DISTINCT <coluna> FROM <nome_tabela>;
seleciona apenas os valores distintos daquela coluna

SELECT <coluna1>,<coluna2>,... FROM <nome_tabela>
WHERE <condição>;
seleciona coluna1, coluna2 da tabela se a condição for satisfeita

Algumas Construções Possíveis

SELECT * FROM <nome_tabela> ORDER BY <coluna>;

seleciona todas as colunas da tabela mostrando todas as colunas em ordem crescente da coluna indicada

SELECT * FROM <nome_tabela> WHERE <condição>
ORDER BY <coluna>;

seleciona todas as colunas da tabela desde que a condição seja satisfeita, mostrando em ordem crescente da coluna indicada

OPERADORES CONDICIONAIS - SQL*(cont.)*

Condição Where

- Diversos critérios podem ser combinados através dos operadores lógicos AND/OR

Operadores condicionais	Significado
• between ... and ...	entre dois valores (inclusive)
• in (....)	lista de valores
• like	com um padrão de caracteres usa-se combinado com '%'
• is null	se é um valor nulo

Exemplos

```
SELECT  NomeEmp, SalarioEmp   FROM  TABEMPREGADO
                    WHERE  SalarioEmp  BETWEEN  500  AND  1000;
```

SELECT NomeEmp, SalarioEmp   FROM TABEMPREGADO
WHERE SalarioEmp >=500 and SalarioEmp<=1000;

```
SELECT  NomeEmp,  DepNume   FROM  TABEMPREGADO
          WHERE  DepNume  IN  (10,30);
```

SELECT NomeEmp, DepNume   FROM TABEMPREGADO
WHERE DepNume=10 or DepNume=30;

Exemplos

```
SELECT  NomeEmp, FuncaoEmp    FROM  TABEMPREGADO
        WHERE   NomeEmp  LIKE  'F%';
```

Seleciona NomeEmp, FuncaoEmp da tabela TABEMPREGADO desde que o NomeEmp comece pela letra 'F'

```
SELECT  NomeEmp, FuncaoEmp    FROM  TABEMPREGADO
        WHERE   NomeEmp  LIKE  '%ana%';
```

Seleciona NomeEmp, FuncaoEmp da tabela TABEMPREGADO desde que o NomeEmp tenha 'ana' em qualquer parte do nome

```
SELECT NomeEmp, FuncaoEmp    FROM TABEMPREGADO
        WHERE  ComissaoEmp  IS  NULL;
```

```
SELECT NomeEmp, FuncaoEmp    FROM TABEMPREGADO
        WHERE  ComissaoEmp  IS  NOT NULL;
```

Exemplos

```
SELECT NomeEmp, SalarioEmp, FuncaoEmp  FROM
     TABEMPREGADO   WHERE  SalarioEmp BETWEEN  700 AND
    2000  AND ( FuncaoEmp =  'BALCONISTA'  OR
                                  FuncaoEmp =  'VENDEDOR' );
```

mostra o nome, salarioe função da tabela empregado desde que o salario esteja entre 700 e 2000 e que a funçaõ eja 'balconista' ou 'vendedor'

```
SELECT   *   FROM  TABEMPREGADO
          WHERE  AdmEmp =  '1980-01-01';
```

datas sempre no formato ano-mes-dia entre apóstrofes

```
SELECT DISTINCT FuncaoEmp  from TABEMPREGADO;
```

mostra apenas os valores distintos para a coluna FuncaoEmp

```
SELECT NomeEmp,  SalarioEmp FROM TABEMPREGADO
                    ORDER BY SalarioEmp DESC;
```

mostra o nome e salario da tabela empregado em ordem decrescente de salarios

Comando SELECT – (cont.)

- ALIAS
  - Uma coluna (campo) pode assumir um apelido durante uma consulta SELECT

Exemplos:

SELECT nome, numdep AS numero_departamento

FROM TABFUNC

SELECT mat AS Matricula, media AS 'Media Final' FROM ALUNOS

** Se o apelido tiver espaços em branco ou caracteres especiais deve ser escrito entre apóstrofes*

Comando SELECT – (cont.)

- ALIAS
  - O apelido também pode ser usado para nomear uma coluna (campo) calculado

Exemplos:

SELECT nome, qtdd, preco, qtdd*preco AS Total FROM TABPRODUTO

** Total é um campo calculado! Não existe na tabela TABPRODUTO*

Comando SELECT do SQL – (cont.)

- **FUNÇÕES AGREGADAS:**
  - Aplicam-se às tabelas como um todo, ou a grupos de colunas (GROUP BY)
  - Retornam **valores resumo**
    - COUNT (*/ [[DISTINCT] <nome_coluna>])
      - conta ocorrências de uma coluna qualquer (*) ou de uma coluna específica(<nome_coluna>), ou ainda de valores distintos de uma coluna(DISTINCT <nome_coluna>)
    - SUM (<nome_coluna>) - soma
    - AVG (<nome_coluna>) - calcula a média
    - MAX (<nome_coluna>) – valor máximo
    - MIN (<nome_coluna>) – valor mínimo

FUNÇÕES AGREGADAS - Exemplos

SELECT COUNT(*) as total FROM TABEMPREGADO;

mostra o numero de registros contidos na tabela

SELECT COUNT(NomeEmp) as total FROM TABEMPREGADO;

mostra o numero de ocorrências não nulas de NomeEmp da tabela

SELECT COUNT(distinct CidadeEmp) as total FROM TABEMPREGADO;

mostra o numero de ocorrências distintas da coluna CidadeEmp da tabela

SELECT MIN(SalarioEmp) as minimo FROM TABEMPREGADO;

mostra o valor do menor salario da tabela

SELECT MAX(SalarioEmp) as maximo FROM TABEMPREGADO;

mostra o valor do maior salario da tabela

FUNÇÕES AGREGADAS - Exemplos

SELECT AVG(SalarioEmp) as media FROM TABEMPREGADO;

mostra a media dos salários da tabela

SELECT SUM(SalarioEmp) as soma FROM TABEMPREGADO;

mostra a soma de todos os salários da tabela

SELECT DepNume as DEPTO, AVG(SalarioEmp) as media FROM TABEMPREGADO GROUP BY DepNume;

mostra o Número do depto, e a media dos salários, agrupado por Número do depto

SELECT DepNume, AVG(SalarioEmp) as media FROM TABEMPREGADO GROUP BY DepNume HAVING COUNT(*) > 3;

mostra o Número do depto, e a media dos salários agrupado por Número do depto, desde que cada depto tenha mais de 3 empregados