Introdução ao Banco de Dados

André Luyde da Silva Souza

Aula 01: Introdução

DML - Definição

- A Linguagem DML (Data Manipulation Language) é composta por 4 operações de manipulação de dados que estão representadas abaixo:
  - Inserir dados – INSERT
  - Excluir dados – DELETE
  - Atualizar dados – UPDATE
  - Recuperar dados (Consultar) – SELECT

INSERT INTO (inserção)

- O comando INSERT insere dados em uma relação (tabela).

Sintaxe:

- INSERT INTO <nome_tabela> (<atributo1,atributo2, ..., atributoN>) VALUES (<valor1, valor2, ..., valorN>);

INSERT INTO (inserção)

- EXEMPLO: Inserir uma nova tupla (linha) na tabela Peca

Sintaxe:

- INSERT INTO Peca (Cod_Peca, Nome_Peca,Preco, Qte) VALUES (380,'Peca W',77.00, 23)

Cod_Peca	Nome_Peca	Preco	Qte
56	Peca X	23.90	10
99	Peca Y	56.99	5
200	Peca Z	80.00	0
380	**Peca W**	**77.00**	**23**

DELETE FROM (exclusão)

- O comando delete exclui dados em uma relação (tabela).
- ATENÇÃO: DROP x DELETE!!!
  - DROP é diferente de DELETE

Sintaxe:

- DELETE FROM <nome_tabela> WHERE <condição-de-exclusão>

DELETE FROM (exclusão)

- EXEMPLO: Excluir a peça 200 (toda a linha).

Sintaxe:

- DELETE FROM Peca WHERE Cod_Peca = 200

Cod_Peca	Nome_Peca	Preco	Qte
56	Peca X	23.90	10
99	Peca Y	56.99	5

UPDATE/ SET (alteração)

- O comando update atualiza dados em uma relação (tabela).
- Quando há mudança de endereço, nome, valor, etc...

Sintaxe:

- UPDATE <tabela> SET ¡coluna1> = <expressão1>, ..., <colunaN> =<expressãoN> WHERE <condição-de-alteração>

UPDATE/ SET (alteração)

- EXEMPLO: Alterar o Preco da peça 200 de 80,00 para 90,00

Sintaxe:

- UPDATE Peca SET Preço = 90.00 WHERE Cod_Peca = 200

Cod_Peca	Nome_Peca	Preco	Qte
56	Peca X	23.90	10
99	Peca Y	56.99	5
200	Peca Z	90.00	0

SELECT (selecionar)

- O resultado de uma consulta SQL é uma relação em que há duplicidade nas relações.

Sintaxe:

- SELECT <lista de atributos> FROM <nome das tabelas> WHERE <condição de pesquisa>

SELECT (selecionar)

- CLÁUSULA WHERE
  - A cláusula WHERE especifica as condições que devem ser satisfeitas.
  - Usa conectores lógicos:
    - AND (e)
    - OR (ou)
    - NOT (não)

SELECT (selecionar)

- CLÁUSULA WHERE
- Usa operadores de comparação:
  - $>$ (maior)
  - $<$ (menor)
  - $=$ (igual)
  - $<=$ (menor ou igual)
  - $>=$ (maior ou igual)
  - BETWEEN (entre)
    - facilita a especificação de condições numéricas que envolvam um intervalo, ao invés de usar os operadores $=<$ e $>=$.

SELECT (selecionar)

- EXEMPLO 1: Selecionar o código e o nome das peças com código menor do que 100:

Sintaxe:

```
SELECT Cod_Peca, Nome_Peca FROM Peca WHERE Cod_Peca < 100;
```

- Resultado:

Cod_Peca	Nome_Peca
56	Peca X
99	Peca Y

SELECT (selecionar)

- EXEMPLO 2: Selecionar o código e o nome das peças com preço maior que 50,00 e menor do que 70,00:

Sintaxe:

- SELECT Nome_Peca, Preco FROM Peca WHERE (Preco BETWEEN 50.00 AND 70.00)

- Resultado:

Nome_Peca	Preco
Peca Y	56.99

SELECT (selecionar)

- O (*) – ASTERISCO – seleciona todos os atributos da tabela.

- EXEMPLO 3: Selecionar todas informações das peças cuja quantidade seja maior ou igual a 10.

Sintaxe:

- SELECT * FROM Peca WHERE Qte >= 10

- Resultado:

Cod_Peca	Nome_Peca	Preco	Qte
56	Peca X	23.90	10

SELECT (selecionar)

- EXEMPLO 4: Selecionar o código, o nome, o preço e a quantidade de peças de código 56 que há no estoque:

Sintaxe:

- SELECT * FROM Peca WHERE Nome_Peca = "Peca Z"

- Resultado:

Cod_Peca	Nome_Peca	Preco	Qte
200	Peca Z	80.00	0

SELECT (selecionar)

- CLÁUSULA ORDER BY (ORDENAÇÃO E APRESENTAÇÃO DE TUPLAS)
  - Aplicado apenas à operação **SELECT**, depois da cláusula **WHERE**
  - A cláusula **ORDER BY** faz com que as tuplas do resultado de uma consulta apareçam em uma determinada ordem.
  - Para especificar a forma de ordenação, devemos indicar:
    - **Desc**: ordem descendente (decrescente)
    - **Asc**: ordem ascendente (crescente)

Sintaxe:

- ORDER BY <atributo> ASC ORDER BY <atributo> DESC

SELECT (selecionar)

- CLÁUSULA ORDER BY
  - EXEMPLO: Selecionar o nome e a quantidade de todas as peças que há no estoque:

Sintaxe:

- SELECT Nome_Peca, Qte FROM Peca ORDER BY Nome_Peca DESC

- Resultado:

Nome_Peca	Qte
Peca Z	0
Peca Y	5
Peca X	10

FUNÇÕES DE AGREGAÇÃO

- As funções de agregação servem para recuperar dados agregados, ou seja, dados que foram trabalhados sobre os dados armazenados.
- As principais são:
  - SUM (soma dos valores da coluna – deve ser númerico)
  - MAX (valor máximo)
  - MIN (valor mínimo)
  - AVG (average - média – deve ser númerico)
  - COUNT (contador de tuplas da relação).

FUNÇÕES DE AGREGAÇÃO

- EXEMPLO 1: Encontrar a soma dos preços de todas as peças, o maior preço, o menor preço e a média dos preços.

Sintaxe:

- SELECT SUM(Preco), MAX(Preco), MIN(Preco), AVG(Preco) FROM Peca; //sem espaço antes do "("

- Resultado:

SUM(Preco)	MAX(Preco)	MIN(Preco)	AVG(Preco)
160.89	80.00	23.90	53.62999999995

FUNÇÕES DE AGREGAÇÃO

- EXEMPLO 2: COUNT
- Recuperar o número de peças que são vendidas.

Sintaxe:

- SELECT COUNT(*) FROM Peça; //sem espaço entre COUNT e "("

- Resultado:

COUNT(*)
3

Dúvidas

Utilizem o fórum da disciplina que responderei a dúvida de todos.