

Banco de dados

Linguagem SQL – Comandos DML

Prof. Eldane Vieira

Introdução

- Os comandos DML permitem a manipulação dos dados.
 - Inserção, atualização e exclusão de dados em uma tabela.

Tabelas dos exemplos

- Os exemplos ocorreram sobre as tabelas com os seguintes comandos de criação:
 - CREATE TABLE Aluno(
matricula decimal(11) NOT NULL,
nome char(20) NOT NULL,
endereco char(30),
PRIMARY KEY (matricula));
 - CREATE TABLE Cursos(
codigo decimal(7) NOT NULL,
nome char(20) NOT NULL,
matriculaAluno decimal(11),
PRIMARY KEY (codigo),
FOREIGN KEY (matriculaAluno) REFERENCES Aluno(matricula));

O comando INSERT

- O comando INSERT permite adicionar linhas de registros em uma tabela, e possui a seguinte sintaxe:
 - INSERT INTO <nome da tabela> [(<nome dos atributos>)]
VALUES (<valores dos atributos>),(<valores dos atributos>), ... ;
 - Exemplo:
 - Esquema da tabela: Aluno (matrícula, nome, endereço)
 - INSERT INTO Aluno (matricula, nome, endereco)
VALUES (1010, “Maria da Silva”, “Av. Afonso Pena, 1000”);
 - Se a ordem dos dados a serem inseridos respeita a ordem na tabela, o comando de inserção pode ficar assim:
 - INSERT INTO Aluno
VALUES (1010, “Maria da Silva”, “Av. Afonso Pena, 1000”);

O comando INSERT

- Não precisamos seguir a ordem dos campos, mas se isso ocorrer temos que indicar no comando INSERT qual ordem vamos seguir.
 - INSERT INTO Aluno (nome, matricula, endereco)
VALUES ("João da Silva", 1011, "Av. Afonso Pena, 1000");
- Alguns campos podem ser deixados em branco, neste caso podemos indicar o nome do campo e no valor do campo colocamos NULL.
 - INSERT INTO Aluno (nome, matricula, endereco)
VALUES ("André Ribeiro", 1012, NULL);
- Ou não indicar o nome do campo e consequentemente não atribuir valor para ele:
 - INSERT INTO Aluno (nome, matricula)
VALUES ("Bruno Gomes", 1013);
- *Mas em nenhum dos dois casos anteriores pode ter sido usado na criação da tabela o NOT NULL junto ao campo que ficará em branco.*

O comando INSERT

- Mais de um registro de dados pode ser inserido por vez utilizando o comando INSERT. Para isso basta separar os dados dos diferentes registros com vírgula, por exemplo:
 - INSERT INTO Aluno (nome, matricula, endereco)
VALUES ("João da Silva", 1011, "Av. Afonso Pena, 1000"), ("Maria da Silva", 1012, "Av. Floriano Peixoto, 2000");

Operadores lógicos

- Nos comandos de manipulação de dados, há cláusulas condicionais como o WHERE.
- Para que possamos utilizar mais de uma condição na mesma cláusula, utilizamos os operadores lógicos.
 - Quando utilizamos o operador AND, teremos como retorno verdadeiro se as duas condições forem verdadeiras.
 - Quando utilizamos o OR, teremos como retorno verdadeiro se uma das condições for verdadeira.
 - Quando utilizarmos o NOT, teremos como retorno verdadeiro se a condição for falsa.

O comando UPDATE

- Comando UPDATE é usado para atualização ou modificação de dados. Possui a seguinte sintaxe:
 - UPDATE <nome da tabela>
SET <alteração1, alteração2, ...>
[WHERE <condição>];

O comando UPDATE - Exemplo

- UPDATE Aluno
SET endereco = "Av. Floriano Peixoto, 1111"
WHERE matricula = 1012;

O comando UPDATE

- Para um novo exemplo será inserido o campo cidade à tabela Aluno.
 - ALTER TABLE Aluno
ADD cidade varchar(20);
- Em seguida atualize o valor de cidade, mas antes pode ser necessário desabilitar as opções de segurança que impedem atualização e deleção de dados sem especificar uma chave pelo comando WHERE.
 - Desabilita restrições de segurança:
 - SET SQL_SAFE_UPDATES = 0;
 - Atualiza o campo cidade:
 - UPDATE ALUNO
SET cidade = “Uberlândia”;

O comando UPDATE - Exemplo

- UPDATE Alunos
SET endereco = "Av. Getúlio Vargas, 222"
WHERE nome="Bruno Gomes"
AND cidade = "Uberlândia";

O comando DELETE

- O comando DELETE é utilizado para remoção de tuplas ou registros.
- A sintaxe desse comando é a seguinte:
 - DELETE FROM <nome da tabela>
[WHERE <condição>];

O comando DELETE

- Para excluir todos os DADOS da tabela Aluno basta executar o seguinte comando:
 - `DELETE FROM Aluno;`
- Podemos utilizar o comando WHERE para definir uma condição para a exclusão.
 - `DELETE FROM Aluno`
`WHERE matricula = 1011;`
- O comando DELETE exclui toda a linha na tabela que atende a condição e não somente alguns campos.

O comando SELECT

- Comando SELECT será apresentado aqui, de maneira simplificada, para poder verificar as alterações na tabela, mas este comando será estudado mais detalhadamente em outra aula.
 - `SELECT * FROM <nome da tabela>;`
 - Exemplo:
 - `SELECT * FROM Aluno;`
 - O asterisco (*) indica que todos os campo da tabela devem ser apresentados na consulta.

Exercício 1

- Considerando os seguintes esquemas de tabelas, crie as seguintes tabelas para realizar os exercícios a seguir:
 - Cliente (CPE, nome, numConta, DDD, telefone, cidade)
 - Carro (Chassi, modelo, cor, ano, valor)
 - Aluguel (CPF_Cliente, CHASSI, dataEntrada, dataSaida, total)
- a) Inclua 3 registros na tabela Carro.
- b) Inclua 3 registros utilizando um só comando na tabela Cliente.
- c) Inclua 3 registros na tabela Aluguel baseado nos dados existentes nas tabelas Cliente e Carro.
- d) Altere na tabela Cliente o campo DDD de todos os registros para 019.
- e) Altere os valores de cidade na tabela Cliente em todos os registros que possuem conta com número maior que 2000, o novo valor de cidade deve ser Brasília.
- f) Na tabela Carros modifique todos os valores dos carros para 20000.
- g) Exclua da tabela Cliente os registros com campo cidade igual a Campinas.
- h) Exclua da tabela Carro os registros de veículos dos anos 2003 e 2004.