CEFET-MG

Banco de Dados

Prof.: Carlos Storck

SQL

SQL = Structured Query Language

- Linguagem de Consulta Estruturada;
- Criado pela IBM em San Jose - CA, em 1970 para o produto DB2;
- SQL foi adotado pela ANSI (pela American National Standards Institute) em 1986 e pela ISO (International Organization for Standardization) em 1987;
- SQL usa uma combinação de construtores em álgebra e cálculo relacional;
- Tornou-se padrão para os SGBD.

SQL

DDL (Data Definition Language): linguagem de definição de dados:

Permite a especificação da base de dados;

Definir as tabelas;

Comandos para esquemas de relação;

Criação de índices;

Ex.: create table, alter, drop, …

SQL

DML (Data Manipulation Language): linguagem de manipulação de dados

Permite a consulta e atualização de informações;

Abrange a álgebra e o cálculo relacional de tuplas;

Comandos para inserção, exclusão e modificação;

Ex.: select, insert, delete, …

SQL

Os comandos básicos da DML são:

SELECT - É geralmente o mais usado do DML, comanda e permite ao usuário especificar uma query como uma descrição do resultado desejado.

INSERT - É usado para somar um registro (formalmente uma TUPLA) a uma tabela existente. Ou seja, adicionar uma linha na tabela.

UPDATE – Usado para mudar os valores de dados em um registro de uma tabela existente.

DELETE - Permite remover registros existentes em uma tabela;

SQL

DCL (Data Control Language - Linguagem de Controle de Dados):

Controla os aspectos de autorização de dados;

Também as licenças concedidas aos usuários;

Controle de quem tem acesso e quem pode manipular dados dentro do banco de dados.

Ex.: grant, revoke, alter password, …

SQL

DTL (Data Transaction Language - Linguagem de Transação de Dados):

Inclui comandos para a especificação de iniciação e finalização de transações;

Algumas implementações permitem o bloqueio de dados para controle de concorrência;

Ex.: start transaction, commit, rollback, …

START TRANSACTION, pode ser usado para marcar o começo de uma transação de banco de dados que pode ser completada ou não.

COMMIT - Grava todos os dados modificados ou inseridos, permanentemente.

ROLLBACK - Faz com que as mudanças nos dados existentes desde que o último COMMIT ou ROLLBACK sejam descartadas.

CREATE TABLE

```
CREATE TABLE < nome_tabela >  (
        nome_atributo1 < tipo > [ NOT NULL ],
        nome_atributo2 < tipo > [ NOT NULL ],
            ......
            nome_atributoN < tipo > [ NOT NULL ]
) ;
```

Onde:

nome_table - Indica o nome da tabela a ser criada.
nome_atributo - Indica o nome do campo a ser criado na tabela.
tipo - Indica a definição do tipo de atributo (integer, char, real, date...);

Exemplo:

```
CREATE TABLE departamento  (
        cod_depto NUMERIC NOT NULL,
        nome_depto VARCHAR [50] NOT NULL,
        descricao_depto VARCHAR [200]
) ;
```

CREATE TABLE

Exemplo Definindo Chaves:

```
CREATE TABLE Departamento (
CodDepto INT NOT NULL,
Nome CHAR(15) NOT NULL,
CodGerente CHAR(9) NOT NULL,
DataInicioGerencia DATE,
PRIMARY KEY (CodDepto),
FOREIGN KEY (CodGerente) REFERENCES Empregado(CodEmp));
```

ALTER TABLE

ALTER TABLE < nome_tabela >
ADD < nome_coluna > < tipo_coluna >;
| DROP < nome_coluna >;
| ALTER [COLUMN] < nome_coluna > < cláusula >;

< cláusula > = TO < novo_nome_coluna >
| TYPE < novo_tipo_coluna >
| POSITION < nova_posição_coluna >

Onde:

nome_table - Indica o nome da tabela a ser excluída.
nome_coluna – Indica o nome da coluna que será inserida.
novo_nome_coluna – Indica o novo nome para a coluna.
novo_tipoj_coluna – Indica o novo tipo para a coluna.
nova_posição_coluna – Indica a posição que a coluna assumirá na tabela

Exemplo:

```
ALTER TABLE Empregado RENAME COLUMN Nome TO NomeEmp;

ALTER TABLE Empregado ADD CONSTRAINT PKEmp PRIMARY KEY (CodEmp);

ALTER TABLE Empregado DROP CONSTRAINT FKEmpDepto;

ALTER TABLE Empregado DROP COLUMN DataAdm;
```

DROP TABLE

Sintaxe:

DROP TABLE < nome_tabela >;

Onde:

nome_table - Indica o nome da tabela a ser excluída.

Ex.: DROP TABLE departamento;

INSERT

O comando para inserção de Registros (Linhas ou Tuplas) numa tabela é o comando INSERT:

Sintaxe - Inserção Unitária:

INSERT INTO <tabela>

(<lista-de-colunas>)

VALUES (<lista-de-valores>)

Inserir Cod_Peca, Nome_Peca, Preco

INSERT INTO Peca

(Cod_Peca, Nome_Peca,Preco)

VALUES (380,'Peca W',77.00)

INSERT

Inserir uma Peca com todos os atributos:

INSERT INTO Peca

VALUES (423,'Peca K',100.00,15)

Sintaxe - Inserção em Massa:

INSERT INTO <tabela1>

(<lista-de-colunas>)

SELECT ...

Exemplo:

INSERT INTO Empregado_BH (SELECT * FROM Empregado

WHERE Cidade = 'BH');

UPDATE

O comando para atualizar Registros (Linhas ou Tuplas) numa tabela é o comando UPDATE:

UPDATE <tabela>

SET <coluna1> = <expressão1>,

<coluna2> =<expressão2>, ...

WHERE <condição-de-alteração>

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

UPDATE Peca

SET Peso = 90.00

WHERE Cod_Peca = 200

UPDATE

Alteração de salário

```
UPDATE Empregado SET Salario = Salario * 1.10;
```

Alteração de salário por cidade

```
UPDATE Empregado SET Salario = Salario * 1.10
WHERE Cidade = ‘BH’;
```

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

```
UPDATE Peca
SET Peso = 90.00
WHERE Cod_Peca = 200
```

DELETE

DELETE (Exclusão)

DELETE FROM <tabela>

WHERE <condição-de-exclusão>

Excluir a peça 200:

DELETE FROM Peca WHERE Cod_Peca = 200

Excluir as peças que tem mais de 1000 na Qte:

DELETE FROM Peca WHERE Qte > 1000

Excluir empregados por cidade:

DELETE FROM Empregado WHERE Cidade = ‘SP’;

ATIVIDADE

Criar a tabela e inserir os dados abaixo, através da linguagem SQL:

Matricula	Nome	Endereço	Depto.
1001	João	Rua 5	108
1003	Pedro	Rua 8	123
1004	Manoel	Rua 6	120
1005	Pedro	Rua 7	123
1007	Maria	Rua6	132