

Banco de dados

Linguagem SQL – Comandos DDL

Prof. Eldane Vieira

Introdução

- SQL – Structured Query Language.
- Utilizada para consultar, criar tabelas, inserir, excluir e alterar dados do banco de dados.

DDL – Data Definition Language

- Possibilita a criação, exclusão e modificação de definições de relações ou tabelas.

Comando CREATE DATABASE

- Quando iniciamos um projeto o primeiro passo é criar o banco de dados com que trabalharemos.
- A sintaxe do comando é:
 - CREATE DATABASE <nome do banco de dados> ou
 - CREATE SCHEMA <nome do banco de dados>

Comando DROP DATABASE

- Este comando nos dá a opção de excluir o banco de dados criado.
- A sintaxe do comando é:
 - DROP DATABASE <nome do banco de dados>
- Nem todo SGBD disponibiliza esse comando, fornecendo outra forma de exclusão de banco de dados.

Tipos de dados em SQL

- Antes de criar as tabelas é preciso conhecer alguns dos tipos de dados para definirmos os campos de maneira correta.

Tipos de dados	Descrição
CHARACTER(n) / CHAR(n)	String (n>0)
VARCHAR(n)	String (n>0), tamanho variável
BIT(n)	String com n Bits (n>0)
NUMERIC(n,d) / DECIMAL (n,d)	Valor numérico, n dígitos (n>0) e d decimais (d>=0)
INTEGER / INT	Inteiro
FLOAT	Ponto Flutuante
DATE	Data
TIME	Hora
TIMESTAMP	Intervalo de tempo

Comando CREATE TABLE

- Comando usado para criar as tabelas, onde serão especificados os campos e as chaves.
- Sintaxe do comando é:
 - CREATE TABLE <nome da tabela>
(<nome do atributo> <tipo de dado> [NOT NULL],...,
<nome alternativo> PRIMARY KEY (<nome do atributo>,<nome do atributo2>,
...)
<nome alternativo> FOREIGN KEY (<nome do atributo>) REFERENCES
<nome da tabela>(<nome do atributo na tabela de origem>));
- PRIMARY KEY indica os atributos que formam a chave primária, podendo ser mais de um, formando uma chave composta.
- FOREIGN KEY indica os atributos que formam a chave estrangeira, chaves oriundas de outras tabelas, nem toda tabela precisa ter chave estrangeira.

Comando CREATE TABLE

- O recurso NOT NULL indica que aquele campo não pode ter valor NULL, ou seja, um valor sempre deve ser definido para aquele campo.
 - A chave primária deverá ser NOT NULL.
- O recurso AUTO_INCREMENT pode ser usado junto a um campo para que o seu valor seja gerado automaticamente de forma incremental. Então, sempre que um dado for inserido o campo com AUTO_INCREMENT receberá o último valor gerado +1 como sendo seu valor.

Especificando a base de dados

- Antes da criação da tabela pode ser preciso especificar em qual esquema a tabela deve ser criada. Para isso, pode-se utilizar o seguinte comando:
 - *use <nome da base de dados>;*

Exemplo 1 CREATE TABLE

- Esquema da tabela:
 - Aluno (matrícula, nome, endereço)
- Comando de criação da tabela:
 - CREATE TABLE Aluno(
matricula decimal(11) NOT NULL,
nome char(20) NOT NULL,
endereco char(30),
PRIMARY KEY (matricula));

Exemplo 2 CREATE TABLE

- Esquema da tabela:
 - Cursos(código, nome, matrículaAluno)
- Comando de criação da tabela:
 - 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));

Tabela com chave estrangeira deve ser criada depois que a tabela com a chave de referência já estiver criada.

Comando ALTER TABLE

- Usado para modificar as tabelas. Podendo adicionar uma coluna, modificar ou excluir uma coluna, renomear uma tabela.
- A sintaxe do comando é:
 - ALTER TABLE <nome da tabela>
[ADD <nome do atributo> <tipo de dado>;]
CHANGE <nome do atributo> <novo nome ou continua com o mesmo><tipo de dado>;]
RENAME <nome novo da tabela>;]
DROP <nome do atributo>;]

Exemplo 1 ALTER TABLE

- No exemplo a seguir alteramos o nome do atributo “nome” para “nomeAluno” e seu tamanho para 40 caracteres.

- ALTER TABLE Aluno

CHANGE nome nomeAluno VARCHAR(40) not null;

- Por ter utilizado o VARCHAR, mesmo se inserir um nome de tamanho 10, só será ocupado o espaço de 10 caracteres. Caso tivesse mantido CHAR, a ocupação é fixa, ou seja, é ocupado o espaço total da declaração do atributo.

Exemplo 2 ALTER TABLE

- A seguir é apresentado um exemplo em que é adicionado o atributo cidade à tabela Alunos.
- ALTER TABLE Aluno
ADD cidade VARCHAR(20);

Exemplo 3 ALTER TABLE

- No exemplo a seguir será excluído o atributo cidade.
- ALTER TABLE Aluno
DROP cidade;

Exemplo 4 ALTER TABLE

- O exemplo a seguir mostra como alterar o nome de uma tabela.
- ALTER TABLE Cursos
RENAME CursosEAD;

Exemplo 5 ALTER TABLE

- O exemplo a seguir mostra como adicionar uma chave primária em uma tabela já criada.
- ALTER TABLE Aluno
ADD PRIMARY KEY (matricula);

Exemplo 6 ALTER TABLE

- O exemplo a seguir mostra como adicionar uma chave estrangeira em uma tabela já criada.
- ALTER TABLE Cursos
ADD FOREIGN KEY (matriculaAluno) REFERENCES
Aluno(matricula);

Comando DROP TABLE

- Este comando serve para excluir tabelas, mas verifique os relacionamentos existentes.
- Devemos verificar se algum atributo da tabela a ser deletada é chave estrangeira em outra tabela. Como ficaria uma chave estrangeira sem o atributo de referência?
- A sintaxe do comando é a seguinte:
 - DROP TABLE <nome da tabela>

Restrições de integridade

- Os SGBDs verificam as restrições de chave estrangeira para evitar inconsistência na base.
- *Constraint* é uma solução para especificar regras na restrição de chaves estrangeiras.

Restrições de integridade

- Violações podem ocorrer nas operações de UPDATE ou DELETE.
- O uso de *constraint* deve ocorrer na declaração das chaves da tabela.
- Antes das declarações de PRIMARY KEY ou FOREIGN KEY podemos colocar a palavra CONSTRAINT.
 - No caso da chave primária serve para reforçar a integridade, o que pode ser redundante.
 - No caso da chave estrangeira serve para indicar que o conteúdo do campo deve referenciar a um outro campo.
 - CONSTRAINT <nome opcional> FOREIGN KEY (matriculaAluno) REFERENCES Aluno(matricula)

Restrições de integridade

- Os comandos a seguir servem para programarmos eventos quando ocorrerem deleções ou atualizações.
 - CASCADE: Propaga-se a alteração feita em uma tabela com alguma chave para as outras tabelas que a referenciam.
 - ON DELETE CASCADE
 - ON UPDATE CASCADE
 - SET NULL: Pode-se atribuir NULL a uma coluna já que a chave referenciada não existe mais, essa opção não se aplica a chave primária, pois ela não deve assumir valor NULL.
 - ON DELETE SET NULL
 - ON UPDATE SET NULL
 - NO ACTION: Não realiza ações.
 - ON DELETE NO ACTION
 - ON UPDATE NO ACTION
 - Exemplo:
 - FOREIGN KEY (matriculaAluno) REFERENCES Aluno(matricula)
ON UPDATE CASCADE ON DELETE CASCADE;

Exercício 1

- Dados os esquemas a seguir, com a linguagem SQL codifique o que é pedido nos itens:

Cliente (ID_Cliente, nome, endereço, cidade, telefone, tipo)

Empresa (ID_Empresa, CNPJ, endereço, telefone)

Cliente_Empresa (ID_Empresa, ID_Cliente)

- a) Crie uma base de dados onde as tabelas serão criadas.
- b) Crie todas as tabelas. Inclua a restrição de efeito cascata às chaves ID_Empresa e ID_Cliente na tabela Cliente_Empresa caso forem deletados ou atualizadas.
- c) Inclua o campo contato na tabela Empresa.
- d) Exclua o campo tipo da tabela Cliente.
- e) Altere o campo endereço para VARCHAR(80) na tabela Empresa.
- f) Altere o nome da tabela Cliente para Pessoas.