SQL - Criação de Tabelas

André Restivo

Faculdade de Engenharia da Universidade do Porto

February 24, 2012

Sumário

1. Introdução
2. Tabelas
3. Colunas
4. Restrições de Integridade
5. Modificação de Tabelas
6. Domínios

Structured Query Language

- SQL - Linguagem para definir, manipular e questionar uma Base de Dados Relacional.
- SQL = DDL + DML + DQL + ...
  - DDL = Data Definition Language.
  - DML = Data Manipulation Language.
  - DQL = Data Query Language.

SQL Versões

- 1986 (SQL-86 e SQL-87) Publicado pela ANSI e ratificado pela ISO.
- 1989 (SQL-89)
- 1992 **(SQL-92)** Também conhecido como SQL2.
- 1999 (SQL:1999) Também conhecido como SQL 3. Inclui expressões regultares, queries recursivas, gatilhos, tipos não escalares e algumas funcionalidades orientadas a objectos.
- 2003 (SQL:2003) Inclui suporte a XML e colunas com numeração automática.

Criação de Tabelas

Na sua forma mais básica é preciso apenas indicar o nome da tabela, os nomes das várias colunas e o tipo de cada uma delas.

Exemplo

```
create table <nometabela> (
  <nomecoluna> <tipocoluna>,
  <nomecoluna> <tipocoluna>
);
```

Remoção de Tabelas

Para remover uma tabela basta indicar o seu nome.

Exemplo

```
drop table <nometabela>;
```

Tipos de Dados

Os tipos de dados mais utilizados são:

- **char**(n) ou character(n) - Cadeia de caractéres de tamanho fixo n
- **varchar**(n) ou character varying(n) - Cadeia de caractéres com tamanho máximo n
- **text** - Cadeia de caractéres sem tamanho definido
- **int** ou integer - Números inteiros (4 bytes)
- **numeric**(precisão, escala) - Números reais sem limite de tamanho
- **date** e time - Data e hora
- **timestamp** - Data + hora no mesmo campo
- **boolean** - Valores booleanos

Exemplo

Exemplo

```
create table empregado (
   bi integer,
   nome varchar(256),
   salario numeric(9,2),
   datanascimento date
);
```

Valores por Omissão

Podem ainda ser definidos valores por omissão para cada coluna usando a palavra-chave **default**.

Exemplo

```
create table empregado (
  bi integer,
  nome varchar(256),
  salario numeric(9,2) default 0,
  datanascimento date
);
```

Restrições de Integridade

Em SQL podem ser definidas restrições de integridade de vários tipos.

- Check
- Not Null
- Unique (Chaves candidatas não primárias)
- Primary Key (Chaves candidatas)
- Foreign Key (Chaves estrangeiras)

As restrições podem ser de dois tipos: de coluna (referem-se a apenas uma coluna e são descritas em frente à coluna em causa) ou de tabela (referem-se a mais do que a uma coluna e ficam separadas da definição das colunas).

Restrições Check

As restrições do tipo CHECK permitem garantir que uma ou mais colunas seguem uma determinada regra que pode ser expressa como uma expressão matemática.

Exemplo

```
create table empregado (
  bi integer,
  nome varchar(256),
  salario numeric(9,2)
    default 0
    check (salario >= 0),
  datanascimento date
);
```

Restrições Check

Podemos e devemos sempre dar nomes às restrições para que seja mais fácil identificar a razão pela qual a inserção de dados falha.

Exemplo

```
create table empregado (
  bi integer,
  nome varchar(256),
  salario numeric(9,2)
    default 0
    constraint sal_positivo check (salario >= 0),
  datanascimento date
);
```

Restrições Check

No caso da restrição abranger mais de uma coluna temos de usar uma restrição de tabela.

Exemplo

```
create table empregado (
  bi integer,
  nome varchar(256),
  salario numeric(9,2),
  descontos numeric(9,2),
  constraint desconto_menor_salario
    check (desconto < salario)
  ...
);
```

Restrições Not Null

Para garantir que uma coluna não vai ter valores nulos podemos usar uma restrição do tipo **not null**.

Exemplo

```
create table empregado (
  bi integer,
  nome varchar(256) not null,
  salario numeric(9,2),
  datanascimento date
);
```

Restrições Chave Primária

- Podemos definir uma, e só uma, chave primária para a tabela.
- Uma chave primária não pode conter valores nulos nem pode ter valores repetidos.

Exemplo

```
create table empregado (
  bi integer primary key,
  nome varchar(256) not null,
  salario numeric(9,2),
  datanascimento date
);
```

Restrições Chave Primária

- Uma chave primária pode ser composta por mais de do que um atributo.
- Nesse caso temos de usar uma restrição de tabela.

Exemplo

```
create table empregado (
   pnome varchar(256),
   unome varchar(256),
   salario numeric(9,2),
   datanascimento date,
   primary key (pnome, unome)
);
```

Restrições Chaves Candidatas

- Chaves candidatas alternativas podem ser definidas usando restrições do tipo **unique**.
- Estas restrições são equivalentes às restrições de chave primária mas não obrigam os valores a ser não nulos.

Exemplo

```
create table empregado (
  bi integer primary key,
  nif integer unique,
  nome varchar(256) not null,
  salario numeric(9,2),
  datanascimento date
);
```

Restrições Chaves Candidatas

Tal como as outras restrições devem ser nomeadas e no caso de incluirem mais de uma coluna devem ser declaradas como restrições de tabela.

Exemplo

```
create table empregado (
  bi integer primary key,
  nif integer constraint nif_unico unique,
  pnome varchar(256),
  unome varchar(256),
  ...
  constraint nome_unico
    unique (pnome, unome)
);
```

Restrições Chaves Estrangeiras

- Uma restrição do tipo **foreign key** permite declarar chaves estrangeiras.
- Uma chave estrangeira deve sempre referenciar uma chave primária ou única.

Exemplo

```
create table empregado (
   bi integer primary key,
   depid integer references departamento(id)
   ...
);
```

Restrições Chaves Estrangeiras

No caso da coluna (ou colunas) referenciada ser a chave primária de outra tabela, podemos omitir o nome da coluna referenciada.

Exemplo

```
create table empregado (
   bi integer primary key,
   depid integer references departamento
   ...
);
```

Restrições Chaves Estrangeiras

No caso da chave estrangeira ser composta por mais de uma coluna usa-se uma restrição de tabela:

Exemplo

```
create table empregado (
  bi integer primary key,
  rua varchar(256),
  cidade varchar(256),
  foreign key (rua, cidade) references rua,
  ...
);
```

Restrições Chaves Estrangeiras

Devemos sempre definir o que acontece quando uma chave estrangeira é violada durante uma operação de remoção ou alteração.

Exemplo

```
create table empregado (
  bi integer primary key,
  depid integer references departamento
    on delete set null on update cascade,
  ...
);
```

Neste exemplo estamos a definir que no caso de um departamento ser apagado os empregados que pertençam a esse departamento devem ficar com o *depid* a null. No caso da chave primária do departamento ser modificada, essa modificação deve propagar-se e modificar também o *depid* da tabela empregado.

Restrições Chaves Estrangeiras

- As alternativas para os valores de **on delete** e **on update** são:
  - – **restrict** – Não deixa efectuar a operação
  - – **cascade** – Apaga os registos associados (delete) ou altera a chave estrangeira (update).
  - – **set null** – A chave estrangeira passa a **null**.
  - – **set default** – A chave estrangeira passa a ter o valor por omissão.
- No caso de não ser possível corrigir o erro, as chaves estrangeiras funcionam como se a restrição fosse do tipo **restrict**.

Modificação de tabelas

Depois de criada, uma tabela pode ser modificada.

Exemplo

```
ALTER TABLE empregado ADD COLUMN nome character varying(64);
ALTER TABLE empregado DROP COLUMN nome;
ALTER TABLE empregado ADD CHECK (salario > 0);
ALTER TABLE empregado ADD CONSTRAINT nif_unico UNIQUE (nif);
ALTER TABLE empregado ADD FOREIGN KEY (depid) REFERENCES departamento;
ALTER TABLE empregado ALTER COLUMN salario SET NOT NULL;
ALTER TABLE empregado DROP CONSTRAINT nif_unico;
ALTER TABLE empregado ALTER COLUMN salario DROP NOT NULL;
ALTER TABLE empregado ALTER COLUMN salario SET DEFAULT 0;
ALTER TABLE empregado ALTER COLUMN salario DROP DEFAULT;
ALTER TABLE empregado RENAME COLUMN depid TO departamento_id;
ALTER TABLE empregado RENAME TO funcionario;
```

Domains

- Os domains são tipos de dados definidos pelo utilizador de forma a evitar repetições.
- Um domain pode conter um valor default, restrições do tipo not null e check.

Exemplo

```
CREATE DOMAIN d_salario integer
  NOT NULL CHECK (salario > 0);
CREATE TABLE empregado (
   bi integer,
   salario d_salario
);
```