Banco de Dados 1

SQL/DDL Básico - Roteiro

Abra a Documentação do Postresql 9.5*, pois pode ser útil ao longo do roteiro.

*Há uma versão traduzida da documentação, porém esta é referente à versão 8.0 do PostgreSQL. Caso prefira ler em português, acesse: http://pgdocptbr.sourceforge.net/pg80/index.html

No roteiro de hoje, vocês vão conhecer os comandos de criação de tabelas e restrições. Este documento contém um resumo sobre o uso destes comandos. Veja mais detalhes na documentação do PostgreSQL.

Para o roteiro de HOJE, vocês devem usar apenas os seguintes tipos:

- INTEGER - inteiro;
- SERIAL - inteiro de autoincremento;
- CHAR(length) - cadeia de caracteres (string) de tamanho fixo. Exemplo: CHAR(5) para comprimento de exatamente 5 caracteres.
- VARCHAR(length) - cadeia de caracteres (string) de tamanho variável. Exemplo: VARCHAR(20) para comprimento de até 20 caracteres.
- TEXT - cadeia de caracteres com comprimento ilimitado;
- BOOLEAN - true / false;
- NUMERIC - representação geral para números em ponto flutuante;
- DATE - data. Exemplo '2017-12-31';
- TIMESTAMP - data/hora. Exemplo: '2017-12-31 14:05:06';

ATENÇÃO:

Você deverá criar um banco de dados para uma **seguradora de automóveis**. Siga os passos a seguir.

Cada comando executado com sucesso deve ser adicionado em um arquivo de texto chamado:

roteiro1-matricula.sql

Adicione os comandos na **mesma ordem** que os executar.

Você pode usar o **gedit** com a função *highlight* para SQL (View → Highlight Mode…)

Para **colar os comandos no terminal**, use Ctrl+Shift+v

Use **comentários** no seu código para sinalizar onde **cada questão** do roteiro está começando e para anotar qualquer outra coisa que desejar. As linhas que denotam comentários iniciam com '--'. Exemplo de comentário:

Caso tenha interesse em utilizar colunas para representar **chaves artificiais** (códigos), basta defini-los como tipo serial. Não é preciso se preocupar ainda como esses valores seriam inseridos/gerados.

Para valores **monetários**, neste momento, basta definir o campo como NUMERIC. Veremos outras formas de representação monetária mais adiante.

```
-- isto é uma linha comentários
CREATE TABLE pessoa (
nome varchar(40)
)
```

1. Abaixo consta uma lista de tabelas a serem criadas no seu banco de dados. Defina os atributos que julga essenciais para cada uma das tabelas. Não é preciso pensar nas constraints ainda.

    - AUTOMOVEL
    - SEGURADO
    - PERITO
    - OFICINA
    - SEGURO
    - SINISTRO
    - PERICIA
    - REPARO

    Dicas:

    - Não sabe o que é um sinistro? Veja aqui.
    - Um "seguro" representa o serviço que um segurado compra. Um seguro não dura para sempre.
    - No mundo real, é possível fazer um seguro para um carro no nome de outra pessoa.
    - Quando um segurado se envolve em um acidente (sinistro), ele informa à seguradora e seu carro é encaminhado para uma oficina. Em seguida, um perito realiza análise no veículo (uma perícia) no veículo para avaliar a dimensão do dano.

A tabela perícia deve guardar as informações sobre essa análise feita pelo perícia. É com base nestas informações, por exemplo, que a seguradora decide se vai preferir consertar o veículo (realizar um "reparo") ou pagar o prêmio total ao segurado (declarar perda total).

2. Crie as tabelas com os atributos que definiu acima.

3. Defina as chaves primárias das tabelas e adicione essas constraints através de comandos ALTER TABLE. Se decidir remover ou adicionar colunas neste momento (devido as suas decisões sobre chaves primárias), faça essas alterações usando comando ALTER TABLE.

4. Defina as chaves estrangeiras e adicione essas constraints através de comandos ALTER TABLE. Se decidir remover ou adicionar colunas neste momento (devido as suas decisões sobre chaves primárias), faça essas alterações usando comando ALTER TABLE.

5. Pense em atributos adicionais que podem ser úteis para suas tabelas. Pense também nos atributos que poderiam/deveriam ser NOT NULL. Não adicione ainda esses atributos e constraints ao seu banco.

6. Remova todas as tabelas que você criou. Observe que as remoções não podem acontecer em qualquer ordem, visto que agora há constraints envolvendo mais de uma tabela (FOREIGN KEYS).

7. Reescreva seus comandos CREATE TABLE de forma a incorporar todas definições de constraints (PRIMARY KEY, FOREIGN KEY, NOT NULL).

8. Crie as tabelas com os comandos definidos no passo anterior. Execute cada comando CREATE TABLE separadamente de forma a melhor corrigir possíveis erros.

9. Remova novamente todas as tabelas que você criou.

10. Você definiria outras tabelas para este banco? Fique à vontade para recriar seu banco (com os comandos que já estão prontos) e adicionar mais tabelas, caso deseje. Discutiremos em sala de aula suas sugestões.

Envio do Roteiro:

Você deve enviar seu arquivo SQL, contendo comandos e comentários. Os comentários são importantes para saber quais regras de negócio ou simplificações do mundo real você considerou na elaboração do roteiro, visto que há várias respostas possíveis de acordo com cada caso.

Seu arquivo deve ser nomeado da seguinte maneira:

roteiro1-matricula.sql

Esta mesma informação (sobre o nome do arquivo) foi dita antes. Caso não tenha visto, isto significa que você não leu o documento com bastante atenção. Neste caso, talvez seja o caso de você reler o documento novamente antes de enviar sua solução para ter certeza que não deixou passar outras informações cruciais.