COMPUTAÇÃO UFCG

Banco de Dados 1

SQL - Consultas - Parte 1

Neste roteiro, continuaremos trabalhando com a linguagem **SQL**.

Pela primeira vez:

* trabalharemos com um esquema diferente do esquema padrão do PostgreSQL (public);
* utilizaremos comandos contidos em arquivos;

Além disso, iremos praticar consultas SQL em cenários variados.

ATENÇÃO: antes de começar, verifique se deseja fazer **backup** de algo que tenha criado no seu banco de dados (esquemas e/ou dados). Caso deseje, use o comando pg_dump (já visto em roteiros anteriores) para gerar seu backup e depois use o comando scp para transferir o arquivo gerado do servidor para sua máquina local.

Se quiser apagar os elementos produzidos nos roteiros anteriores, **execute** o comando:

```
DROP OWNED BY CURRENT_USER;
```

Criando o Esquema Company

Acesse o seu banco de dados via psql.

Crie um novo esquema chamado “company”:

```
CREATE SCHEMA company;
```

Para confirmar que o schema foi criado, **execute** o comando:

```
\dn
```

Existindo mais de um esquema, surge a necessidade de usar qualificadores para referenciar os elementos dos esquemas, tais como tabelas. Por exemplo, “nomeDoEsquema.nomeDaTabela”. Mas isso é chato. Para facilitar a vida dos usuários do BD (i.e., desenvolvedores, DBAs, etc), existe o search_path. Os elementos dos esquemas que constam no search_path não precisam de qualificadores para serem acessados. Porém, quando recursos de mesmo nome constam em esquemas diferentes, o qualificador se faz necessário para eliminar a ambiguidade.

Veja como está o seu search_path. Ele deve conter o esquema padrão (i.e., public) e não deve conter o esquema recentemente criado (company).

```
SHOW search_path;
```

Adicione o novo esquema ao search_path:

```
SET search_path TO company,public;
```

Visualize como está agora seu search_path. Agora devem constar os dois esquemas.

```
SHOW search_path;
```

Se você não alterou a ordem dos esquemas ao redefinir o search_path, o esquema “company” terá prioridade. Ou seja, ao criar uma tabela sem qualificador, ela será criada nesse esquema. **Faça um teste**:

- Crie uma tabela vazia:
  - CREATE TABLE test();
- Veja em qual esquema a tabela foi criada.
  - \dt public.*
  - \dt company.*
- Crie uma tabela de mesmo nome no esquema public.
- Remova as tabelas.

Atenção! Embora o search_path tenha sido alterado, esta modificação terá validade apenas para esta sessão. Ou seja, caso você saia do psql (com o comando \q) e depois se reconecte ao banco, o search_path terá voltado ao estado anterior. Neste caso, será necessário ajustar o search_path todas as vezes que iniciar uma nova sessão. **Faça um teste**.

Isso é útil pois podem haver muitos usuários (e aplicações) diferentes, acessando o mesmo banco de dados, porém interessados em esquemas diferentes. Além disso, um mesmo usuário pode estar interessado em esquemas diferentes em momentos diferentes (sessões diferentes).

Se desejar alterar o search_path de forma permanente, execute os dois comandos abaixo, nesta ordem. Para executar o primeiro comando, é preciso mencionar o nome do banco de dados, que é o nome utilizado para acessar o psql (como padrão, estamos nomeando como “seuusuario_db”).

```
ALTER DATABASE seuusuario_db SET search_path TO company,public;
SET search_path TO company,public;
```

Definindo os elementos do esquema COMPANY

Baixe o arquivo **company.sql** disponível no site da disciplina. Este arquivo é um script contendo comandos para definição do esquema COMPANY.

Abra uma nova janela do terminal para trabalhar na máquina local.

```
scp -P 45600 pathParaOArquivo seuUsuario@150.165.15.11:/home/seuUsuario
```

Volte para a janela de terminal que está conectada ao BD através do PSQL.

Execute o comando a seguir para executar o script SQL. Este script cria todas as tabelas do esquema COMPANY:

```
\i company.sql
```

Observe que os comandos DROP e CREATE que constam neste arquivo contêm a instrução IF EXISTS. Isto é útil para evitar erros quando o comando não pode ser executado. Por exemplo, ao tentar remover uma tabela que não existe ou ao tentar criar uma tabela que já existe.

Assim, quando queremos remover um elemento que não existe, uma mensagem é exibida no terminal apenas para informar que o comando foi ignorado, porém o demais comandos são executados normalmente. No arquivo que acabamos de executar, mensagens desse tipo foram exibidas para os comandos DROP, visto que as tabelas não existiam. Caso não fosse utilizado IF EXISTS.

Verifique se as 6 tabelas foram criadas.

Inserindo os dados do esquema COMPANY

Baixe o arquivo **company-data.sql** disponível no site da disciplina. Este arquivo é um script contendo comandos INSERT para papular o BD.

Execute o script de forma similar ao que foi feito para definição do esquema.

Executando Consultas

Para as consultas abaixo, **não é permitido usar:**

- LIMIT;
- WITH;
- CASE/WHEN;
- EXISTS;
- Consultas aninhadas - IN.
- SOME, ANY, ALL;
- JOIN (nenhuma variação). As junções **devem** ser feitas na cláusula WHERE (produto cartesiano + filtragem).

Você é responsável por conferir se os resultados da sua consulta estão corretos. Para isso, basta analisar os dados das tabelas e conferir manualmente.

Ao final, você deve enviar suas respostas em um formulário. Há um campo do formulário para cada questão (cada consulta). Copie e cole a consulta exatamente como foi executada, inclusive com o ponto-e-vírgula do final. Não acrescente nada mais nos campos além das consultas, sob o risco de não ter a consulta executada durante a avaliação.

O arquivo enviado não deve conter erros de sintaxe, ou seja, deve ser executado sem erros.

QUESTÕES:

1. Retornar todos os elementos da tabela department.
2. Retornar todos os elementos da tabela dependent;
3. Retornar todos os elementos da tabela dept_locations;
4. Retornar todos os elementos da tabela employee;
5. Retornar todos os elementos da tabela project;
6. Retornar todos os elementos da tabela works_on;
7. Retornar os nomes (primeiro e último) dos funcionários homens;
8. Retornar os nomes (primeiro) dos funcionários homens que não possuem supervisor;
9. Retornar os nomes dos funcionários (primeiro) e o nome (primeiro) do seu supervisor, apenas para os funcionários que possuem supervisor;

10. Retornar os nomes (primeiro) dos funcionários cujo supervisor se chama Franklin;
11. Retornar os nomes dos departamentos e suas localizações;
12. Retornar os nomes dos departamentos localizados em cidades que começam com a letra ‘S’;
13. Retornar os nomes (primeiro e último) dos funcionários e seus dependentes (apenas para os funcionários que possuem dependentes);
14. Retornar o nome completo dos funcionários que possuem salário maior do que 50 mil. A relação de retorno deve ter apenas duas colunas “full_name” e “salary”. O nome completo deve ser formado pela concatenação dos valores das 3 colunas com dados sobrenome. Use o operador || para concatenar strings;
15. Retornar os projetos (nome) e os departamentos responsáveis (nome);
16. Retornar os projetos (nome) e os gerentes dos departamentos responsáveis (primeiro nome). Retornar o resultados apenas para os projetos com código maior do que 30;
17. Retornar os projetos (nome) e os funcionários que trabalham neles (primeiro nome);
18. Retornar os nomes dos departamentos dos funcionários que trabalham no projeto 91. Retornar também o nome (primeiro) do funcionário e o relacionamento entre eles. Retornar os atributos na mesma ordem que aparecem abaixo.

Resultado da consulta 18:

```
fname  | dependen_name | relationship
---------+---------------------+-----------------
Alex     | Johnny              | Son
Alex     | Tommy               | Son
Bonnie | Chris               | Spouse
Alec     | Sam                 | Spouse
```