**T02 – Instalação Postgres, to Join or not to Join e funções**

Igor Augusto Reis Gomes – 12011BSI290 – [igor.augusto@ufu.br](mailto:igor.augusto@ufu.br)

Heitor Guimarães Da Fonseca Filho – 12011BSI203 – [heitor.filho@ufu.br](mailto:heitor.filho@ufu.br)

\*Este relatório contêm prints dos resultados das operações realizadas, o código em está localizado em um arquivo separado.

A seguir, um print do banco de dados conectado e populado no DataGrip (uma alternativa mais moderna ao pgadmin), sendo da mesma empresa responsável pela conhecida IDE de Java, o IntelliJ:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

1. Acrescente, por exemplo, 100.000 depósitos de R$ 1,00 (Um Real) na conta do cliente 'Germano Luiz de Paula', na agência 'Pampulha', na conta 93134. Você deve executar o código abaixo em uma janela de comandos do PostgreSQL, quando o banco de dados selecionado para consultas for o nosso banco IB.

**Tela de computador com fundo preto

Descrição gerada automaticamente**

1. Agora que temos o banco IB com uma tabela de tamanho significativamente maior, podemos testar a hipótese levantada no texto introdutório: o uso da cláusula JOIN é cômodo, mas custoso.
   1. Selecione os nomes dos clientes e seus respectivos números de conta e nome de agência que fizeram depósitos e empréstimos ao mesmo tempo.

Tela de computador

Descrição gerada automaticamente

Agora vem sua tarefa:

* 1. Construa a consulta equivalente a este exemplo utilizando a cláusula JOIN

**Tela de computador

Descrição gerada automaticamente**

* 1. Construa a consulta equivalente a este exemplo utilizando a SELECT DISTINCT e sem o JOIN.

**Tela de computador

Descrição gerada automaticamente**

1. Construa uma tabela em uma planilha (Programa Calc do Libre Office) com três colunas: intersect, distinct e join. Execute as três consultas pelo menos 30 vezes e registre na planilha o tempo de execução em milissegundos para a conclusão de cada uma das três versões da consulta. O tempo de execução de cada consulta é exibido no canto inferior direito da tela que executou uma consulta. Ao final, tire a média de cada coluna e conclua qual versão da consulta foi mais rápida.

****

1. Uma das principais vantagens obtidas pela utilização da SQL está no uso de uma linguagem não procedural, ou seja, é necessário dizer apenas “o que” é desejado que seja retornado em uma consulta sem se preocupar com os detalhes de “como” a consulta será executada. Entretanto, em determinadas situações nos encontramos enfrentando problemas nos quais uma simples consulta feita na SQL não nos atende. Vejamos alguns exemplos:
   1. Implemente primeiro (por ser mais simples) uma função em PL/pgSQL para retornar o relatório descrito no cenário 2 a partir desta consulta SQL;

select faixa\_cliente(nome\_cliente), nome\_cliente from cliente;

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

* 1. Implemente uma função em PL/pgSQL para retornar o relatório descrito no cenário 1

select nome\_cliente, contas\_cliente(nome\_cliente), cidade\_cliente from cliente;

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**