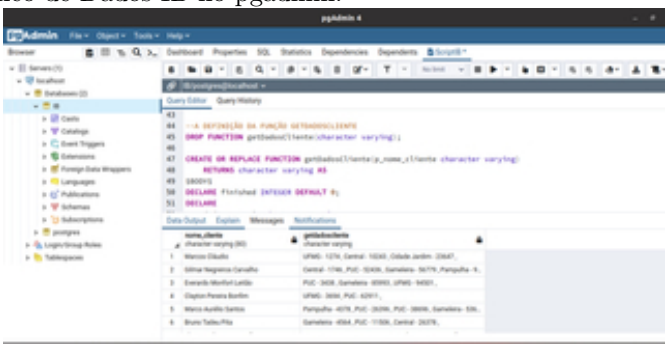
1 Trabalho de Banco de Dados 2

1.1 Exercício 10

Banco de Dados IB no pgadmin.



1.2 Exercício 11

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. Lembra-se de nossa velha consulta:

11.1 Selecione os nomes dos clientes e seus respectivos números de conta e nome de agência que fizeram depósitos e empréstimos ao mesmo tempo.

```
select nome_cliente, numero_conta, nome_agencia from emprestimo
intersect
select nome_cliente, numero_conta, nome_agencia from deposito
```

Agora vem a sua tarefa: 11.2 Construa a consulta equivalente a este exemplo utilizando a cláusula JOIN.

```
-- -- Tempo de Execucao 296 msecs
select nome_cliente, numero_conta, nome_agencia
from emprestimo natural join deposito
```

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

```
-- -- Tempo de Execucao 505 msecs
select distinct e.nome_cliente, e.numero_conta, e.nome_agencia
from emprestimo as e, deposito as d
where e.nome_cliente = d.nome_cliente
and e.numero_conta = d.numero_conta
and e.nome_agencia = d.nome_agencia
```

1.3 Exercício 12

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

	INTERSECT	JOIN	PRODUTO
	181	296	505
	186	239	209
	156	277	202
	294	190	244
	149	277	221
	184	231	276
	205	220	206
	157	234	255
	197	243	216
	638	702	982
	628	573	1084
	826	309	1065
	576	1320	1067
	964	1511	1057
	834	1553	1154
	1071	1094	1010
	1385	1046	1245
	1438	1154	1080
	726	982	1288
	1064	1362	1152
	587	1133	1544
	659	1357	701
	1240	974	659
	1400	863	1134
	670	1993	875
	776	914	881
	1065	1563	1252
	1298	1453	780
	1121	1685	1212
MEDIA	689,1666667	858,2666667	785,2

1.4 Exercício 13

Novamente, você precisa executar uma consulta em SQL para retornar dados de clientes. Os dados armazenados te permitem inferir o quão interessantes são estes clientes para receberem um novo tipo de cartão, com melhores taxas e mais crédito. Para este propósito foram criadas três faixas de clientes: A , B e

C cujas as somas dos valores depositados ultrapassem seis mil, quatro mil e um mil Reais, respectivamente. O foco principal são os clientes classes A, mas existe a possibilidade de que uma parte dos novos clientes deste cartão seja oriunda de clientes do tipo B. Desta forma solicita-se que no relatório final conste apenas o nome do cliente (e outros dados de contato, por exemplo) acompanhado da letra que denomina a faixa de classificação do cliente, as somas das quantias depositadas não devem ser exibidas no relatório. Novamente eu te pergunto: como você vai codificar em letras as faixas de depósitos dos clientes em uma única linha de consulta SQL?

1.4.1 Cenário 1

Novamente, você precisa executar uma consulta em SQL para retornar dados de clientes. Os dados armazenados te permitem inferir o quão interessantes são estes clientes para receberem um novo tipo de cartão, com melhores taxas e mais crédito. Para este propósito foram criadas três faixas de clientes: A , B e C cujas as somas dos valores depositados ultrapassem seis mil, quatro mil e um mil Reais, respectivamente. O foco principal são os clientes classes A, mas existe a possibilidade de que uma parte dos novos clientes deste cartão seja oriunda de clientes do tipo B. Desta forma solicita-se que no relatório final conste apenas o nome do cliente (e outros dados de contato, por exemplo) acompanhado da letra que denomina a faixa de classificação do cliente, as somas das quantias depositadas não devem ser exibidas no relatório. Novamente eu te pergunto: como você vai codificar em letras as faixas de depósitos dos clientes em uma única linha de consulta SQL?

```
1
2  --A DEFINICAO DA FUNCAO GETDADOSCLIENTE
3  DROP FUNCTION getDadosCliente(character varying);
4
5  CREATE OR REPLACE FUNCTION getDadosCliente(p_nome_cliente character
       varying)
6    RETURNS character varying AS
7  $BODY$
8  DECLARE finished INTEGER DEFAULT 0;
9  DECLARE
10   dados_agencia character varying;
11   dados_conta int;
12   dados_cliente character varying;
13   conta character varying;
14   cursor_relatorio cursor FOR SELECT nome_cliente, nome_agencia,
       numero_conta FROM conta
15   WHERE nome_cliente = p_nome_cliente;
16 BEGIN
17   OPEN cursor_relatorio;
18     conta = '';
19     LOOP
20       FETCH cursor_relatorio into dados_cliente, dados_agencia,
       dados_conta;
21       IF FOUND THEN
22         conta = conta || dados_agencia || ' - ' || dados_conta || '
       , ';
```

```
      END IF;
      IF NOT FOUND THEN
        CLOSE cursor_relatorio;
        RETURN conta;
      END IF;
    END LOOP;
END
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION getDadosCliente(character varying)
OWNER TO postgres;

select nome_cliente,getDadosCliente(nome_cliente) from conta
```

1.4.2 Cenário 2

) Novamente, você precisa executar uma consulta em SQL para retornar dados de clientes. Os dados armazenados te permitem inferir o quão interessantes são estes clientes para receberem um novo tipo de cartão, com melhores taxas e mais crédito. Para este propósito foram criadas três faixas de clientes: A , B e C cujas as somas dos valores depositados ultrapassem seis mil, quatro mil e um mil Reais, respectivamente. O foco principal são os clientes classes A, mas existe a possibilidade de que uma parte dos novos clientes deste cartão seja oriunda de clientes do tipo B. Desta forma solicita-se que no relatório final conste apenas o nome do cliente (e outros dados de contato, por exemplo) acompanhado da letra que denomina a faixa de classificação do cliente, as somas das quantias depositadas não devem ser exibidas no relatório. Novamente eu te pergunto: como você vai codificar em letras as faixas de depósitos dos clientes em uma única linha de consulta SQL?

```

CREATE OR REPLACE function getClassificacao(p_numero_conta integer,
     p_nome_agencia character varying, p_nome_cliente character
    varying)
returns character varying as
$BODY$
declare
    soma_deposito float;
    classificacao character varying;
    cursor_relatorio cursor for select sum(d.saldo_deposito) as
    total_dep
        from conta c natural left outer join deposito d
        where c.nome_cliente = p_nome_cliente
        and c.nome_agencia = p_nome_agencia
        and c.numero_conta = p_numero_conta
        group by c.nome_cliente, c.nome_agencia, c.numero_conta;
begin
    open cursor_relatorio;
        fetch cursor_relatorio into soma_deposito;
            if found then
                if soma_deposito is null then soma_deposito = 0;
    end if;
```

```
19                 if soma_deposito > 6000 then classificacao = 'A';
    end if;
20                 if soma_deposito between 6000 and 4000 then
    classificacao = 'B'; end if;
21                 if soma_deposito < 4000 then classificacao = 'C';
    end if;
22             end if;
23         close cursor_relatorio;
24     return classificacao;
25 end
26 $BODY$
27 language plpgsql volatile
28 cost 100;
29 alter function getClassificacao(integer, character varying,
    character varying)
30 owner to postgres;
31
32 select numero_conta, nome_agencia,nome_cliente, getClassificacao(
    numero_conta, nome_agencia,nome_cliente) from conta c
```