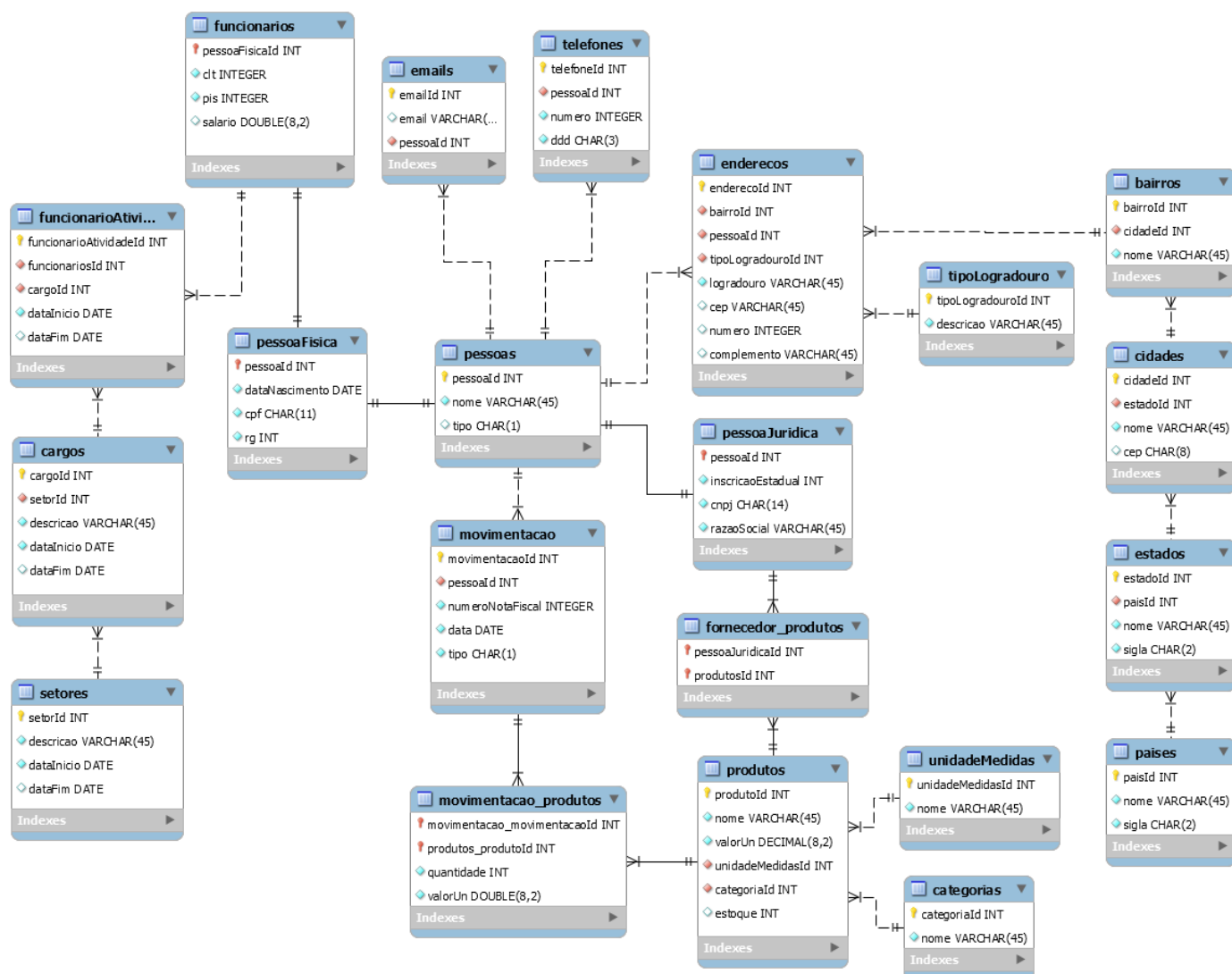


UNIVATES
BANCO DE DADOS 2016B
RESPOSTAS DA PROVA 2-1 – Sistema de Atacado
PROF. JULIANO DERTZBACHER

ALUNO: _____

Identifique-se na folha da prova. Esta prova é individual e não permite a consulta a qualquer material que seja de domínio público, somente os arquivos pessoais autorizados. A prova consiste em 10 pontos distribuídos nas diversas questões, o peso de cada questão está descrito ao lado do respectivo enunciado. Somente serão aceitos comentários sobre o conteúdo da prova nos primeiros 30 minutos, as questões devem ser expostas em voz alta para que todos possam ouvir. A interpretação das questões faz parte da avaliação.

Para resolver as questões 1, 2, 3, 4 e 5, utilize como base o modelo relacional abaixo. A implementação deste modelo no PostgreSQL pode ser recuperada a partir dos arquivos “Prova2-1.backup” ou “Prova2-1.sql”, disponíveis no Ambiente Virtual (Aula15). As respostas para estas questões devem ser submetidas na tarefa “Prova2-1”.



Questão 1 (1 ponto) – Elabore uma consulta que retorna a lista de todos os setores que possuem mais de um funcionário ativo, exibindo o nome do setor, os cargos vinculados ao mesmo e a quantidade de funcionários.

| Descricao | descricao | Count |
|-----------|-------------|-------|
| Vendas | Call Center | 2 |

```
SELECT setor.descricao, cargo.descricao, COUNT(funcionarioatividade.*)
FROM setor INNER JOIN cargo USING (setorid)
      INNER JOIN funcionarioatividade USING (cargoid)
WHERE funcionarioatividade.datafim IS NULL
GROUP BY setor.descricao, cargo.descricao
HAVING COUNT(funcionarioatividade.*) > 1;
```

Questão 2 (2 pontos) – Crie uma VIEW capaz de reunir o nome do produto, o nome do seu fornecedor e o código do seu fornecedor.

```
CREATE OR REPLACE VIEW view_produto_fornecedor (produto, fornecedor,
cod_fornecedor) AS
      SELECT produto.nome, pessoa.nome, pessoa.pessoaid
      FROM fornecedorproduto INNER JOIN produto USING (produtoid)
      INNER JOIN pessoa USING (pessoaid);
```

Utilizando a VIEW criada, desenvolva uma consulta que retorna a lista de todos os produtos cadastrados, agrupando por fornecedor, exibindo a razão social do fornecedor, além do nome do mesmo.

| produto | fornecedor | razaosocial |
|--------------|-----------------|-------------------------|
| Maminha | Barack Obama | Carne Boa Ltda |
| Torta | Jão Silva | Ki Pão Ltda |
| Farinha | Dilma Rousseff | Distribuidora Ltda |
| Abóbora | Augusto Pretto | Clean Ltda |
| Alface | Alcemar Cariado | Comércio de Frutas Ltda |
| Detergente | Barack Obama | Carne Boa Ltda |
| Cenoura | Alcemar Cariado | Comércio de Frutas Ltda |
| Feijão | Dilma Rousseff | Distribuidora Ltda |
| Cebola | Alcemar Cariado | Comércio de Frutas Ltda |
| Cerveja | Maria Souza | João e CIA Ltda |
| Picanha | Barack Obama | Carne Boa Ltda |
| Salsichão | Barack Obama | Carne Boa Ltda |
| Arroz | Dilma Rousseff | Distribuidora Ltda |
| Bolo | Jão Silva | Ki Pão Ltda |
| Pão | Jão Silva | Ki Pão Ltda |
| Refrigerante | Maria Souza | João e CIA Ltda |
| Tomate | Alcemar Cariado | Comércio de Frutas Ltda |
| Energético | Maria Souza | João e CIA Ltda |
| Desinfetante | Augusto Pretto | Clean Ltda |
| Alcatra | Barack Obama | Carne Boa Ltda |
| Bolacha | Dilma Rousseff | Distribuidora Ltda |

```
SELECT view_produto_fornecedor.produto, view_produto_fornecedor.fornecedor,
       pessoaJuridica.razaosocial
FROM view_produto_fornecedor INNER JOIN pessoaJuridica ON
       view_produto_fornecedor.cod_fornecedor = pessoaJuridica.pessoaid
GROUP BY 2,3,1;
```

Questão 3 (2,5 pontos) – Implemente uma STORED PROCEDURE, utilizando um RECORD, que recebe um intervalo de datas (data inicial e data final) e retorna todas as informações das movimentações que ocorreram no período.

| movimentacaoid | pessoaid | numeronotafiscal | data | tipo | valortotal |
|----------------|----------|------------------|------------|------|------------|
| 7 | 11 | 123456789 | 13/10/2015 | 1 | 1.75 |
| 10 | 14 | 369258147 | 12/10/2015 | 1 | 534.40 |

```
CREATE OR REPLACE FUNCTION obterMovimentacoes(p_dtIni DATE, p_dtFim DATE)
RETURNS SETOF movimentacao
AS $$
DECLARE
    sel_movimentacao RECORD;
BEGIN
    FOR sel_movimentacao IN (SELECT *
                             FROM movimentacao
                             WHERE movimentacao.data BETWEEN p_dtIni AND p_dtFim) LOOP
        RETURN NEXT sel_movimentacao;
    END LOOP;
END;
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM obterMovimentacoes ('2015-10-07', '2015-11-07');
```

Questão 4 (2 pontos) – Crie uma STORED PROCEDURE, utilizando a LANGUAGE sql, que recebe como parâmetro o código de um cargo e retorna todas as informações dos funcionários (atributos da tabela pessoa) que ocupam este cargo.

| pessoaid | nome | tipo |
|----------|-----------------|------|
| 3 | Alcemar Cariado | A |
| 3 | Alcemar Cariado | A |
| 3 | Alcemar Cariado | A |
| 3 | Alcemar Cariado | F |
| 10 | Joana Oliveira | F |
| 10 | Joana Oliveira | F |
| 10 | Joana Oliveira | F |

```
CREATE OR REPLACE FUNCTION obterFuncionariosDoCargo(p_cargoId INTEGER)
RETURNS SETOF pessoa
AS $$
    SELECT pessoa.pessoaid, pessoa.nome, pessoa.tipo
    FROM funcionarioatividade INNER JOIN pessoa ON
           pessoa.pessoaid = funcionarioatividade.pessoaid
    WHERE funcionarioatividade.cargoId = p_cargoId;
$$ LANGUAGE sql;
```

```
SELECT * FROM obterFuncionariosDoCargo(4);
```

Questão 5 (2,5 pontos) – Elabore uma TRIGGER com a função de verificar se o salário atribuído a um funcionário está dentro da faixa permitida pela gerência: mínimo de R\$200,00 e máximo de R\$20.000,00. Caso o salário estiver dentro deste intervalo a inserção ou atualização deverá ocorrer normalmente, caso contrário, a mensagem "Salário informado fora da faixa permitida!" deverá ser exibida e a operação cancelada.

```
CREATE OR REPLACE FUNCTION verificaSalarioFuncionario()  
  RETURNS TRIGGER  
  AS $$  
  DECLARE  
  BEGIN  
    IF NEW.salario < 200 OR NEW.salario > 20000 THEN  
      RAISE NOTICE 'Salário informado fora da faixa permitida!';  
      RETURN NULL;  
    ELSE  
      RETURN NEW;  
    END IF;  
  END;  
  $$ LANGUAGE plpgsql;  
  
CREATE TRIGGER tg_verificaSalarioFuncionario  
  BEFORE INSERT OR UPDATE  
  ON funcionario  
  FOR EACH ROW  
  EXECUTE PROCEDURE verificaSalarioFuncionario();
```

BOA PROVA!