



**Fundação CECIERJ – Vice Presidência de Educação Superior à Distância**  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Programação II**

**Gabarito da AD2 – 2º Semestre de 2016**

A seguir é apresentado o esquema relacional (script de criação de tabelas) relativo ao cadastramento de receitas e avaliações de receitas para o site da Palmirinha, famosa doceira da TV. Considere esse esquema para resolver as questões a seguir.

```
CREATE TABLE PESSOA
(
  ID_PESSOA integer NOT NULL,
  NOME character varying,
  LOGIN character varying,
  SENHA character varying,
  CONSTRAINT PK_PESSOA PRIMARY KEY
  (ID_PESSOA) );

CREATE TABLE RECEITA
(
  ID_RECEITA integer NOT NULL,
  DATA_ENVIO date,
  TITULO character varying,
  MODO_PREPARO character varying,
  ID_PESSOA integer,
  CONSTRAINT PK_RECEITA PRIMARY KEY (ID_RECEITA),
  CONSTRAINT FK_PESSOA FOREIGN KEY (ID_PESSOA) REFERENCES
  PESSOA (ID_PESSOA)
  ON UPDATE NO ACTION ON DELETE NO ACTION
);
```

```
CREATE TABLE INGREDIENTE
(
  ID_INGREDIENTE integer NOT
  NULL, DESCRICAO character
  varying, MEDIDA character varying,
  CONSTRAINT PK_INGREDIENTE PRIMARY KEY
  (ID_INGREDIENTE) );
```

```
CREATE TABLE RECEITA_INGREDIENTE
(
  ID_RECEITA integer NOT NULL,
  ID_INGREDIENTE integer NOT
  NULL, QUANTIDADE double
  precision,
  CONSTRAINT PK_INGREDIENTE_RECEITA PRIMARY KEY (ID_RECEITA,
  ID_INGREDIENTE),
  CONSTRAINT FK_INGREDIENTE FOREIGN KEY (ID_INGREDIENTE)
  REFERENCES INGREDIENTE (ID_INGREDIENTE) ON UPDATE NO ACTION ON
  DELETE NO ACTION,
  CONSTRAINT FK_RECEITA FOREIGN KEY (ID_RECEITA)
  REFERENCES RECEITA (ID_RECEITA) ON UPDATE NO ACTION ON DELETE NO
  ACTION
  );
```

```
CREATE TABLE VOTACAO
(
  ID_VOTO integer NOT NULL,
  NOTA double precision,
  ID_PESSOA integer,
  ID_RECEITA integer,
  CONSTRAINT PK_VOTO PRIMARY KEY (ID_VOTO),
  CONSTRAINT FK_PESSOA FOREIGN KEY (ID_PESSOA)
  REFERENCES PESSOA (ID_PESSOA) ON UPDATE NO ACTION ON DELETE NO
  ACTION,
  CONSTRAINT FK_RECEITA FOREIGN KEY (ID_RECEITA)
  REFERENCES RECEITA (ID_RECEITA) ON UPDATE NO ACTION ON DELETE NO
  ACTION
  );
```

1. Faça uma função em PHP chamada INSERE\_RECEITA que insira uma nova receita no banco de dados da Palmirinha. Sua função deve receber como parâmetros o ID da pessoa que está postando a receita, o título da receita, a data de postagem e o modo de preparo e insira uma nova receita no banco de dados (tabela RECEITA). Além disso, a sua função deve receber como parâmetros dois **arrays**: o primeiro contendo os nomes dos ingredientes utilizados na receita e o segundo as quantidades dos mesmos utilizadas na receita (assumindo a unidade padrão cadastrada na tabela INGREDIENTE campo MEDIDA).

Observação importante: deverá ser efetuado o teste para verificar se o ingrediente correspondente ao nome informado existe na tabela INGREDIENTE. Caso contrário, sua função deve exibir uma mensagem de erro. Atente também que na tabela RECEITA a referência ao ingrediente é realizada por meio do campo ID\_INGREDIENTE e não por seu nome.

R: Como podemos perceber, o script de criação das tabelas não foi dado inicialmente em MySQL, mas numa pseudo-linguagem. A fim de responder as questões a seguir, considere o script abaixo:

```
CREATE TABLE PESSOA
(
    ID_PESSOA INT NOT NULL,
    NOME VARCHAR(255),
    LOGIN VARCHAR(255),
    SENHA VARCHAR(255),
    CONSTRAINT PK_PESSOA PRIMARY KEY (ID_PESSOA)
);

CREATE TABLE RECEITA
(
    ID_RECEITA INT NOT NULL,
    DATA_ENVIO DATE,
    TITULO VARCHAR(255),
    MODO_PREPARO TEXT,
    ID_PESSOA INT,
    CONSTRAINT PK_RECEITA PRIMARY KEY (ID_RECEITA),
    CONSTRAINT FK_RECEITA_PESSOA FOREIGN KEY (ID_PESSOA) REFERENCES
    PESSOA (ID_PESSOA)
);
```

```

CREATE TABLE INGREDIENTE
(
    ID_INGREDIENTE INT NOT NULL,
    DESCRICAO VARCHAR(255),
    MEDIDA VARCHAR(3),
    CONSTRAINT PK_INGREDIENTE PRIMARY KEY (ID_INGREDIENTE)
    CONSTRAINT UK_DESCRICAO_INGREDIENTE UNIQUE (DESCRICAO)
);

CREATE TABLE RECEITA_INGREDIENTE
(
    ID_RECEITA INT NOT NULL,
    ID_INGREDIENTE INT NOT NULL,
    QUANTIDADE DOUBLE,
    CONSTRAINT PK_INGREDIENTE_RECEITA PRIMARY KEY (ID_RECEITA,
    ID_INGREDIENTE),
    CONSTRAINT FK_INGREDIENTE_RECEITA_INGREDIENTE FOREIGN KEY
    (ID_INGREDIENTE) REFERENCES INGREDIENTE (ID_INGREDIENTE),
    CONSTRAINT FK_INGREDIENTE_RECEITA_RECEITA FOREIGN KEY
    (ID_RECEITA) REFERENCES RECEITA (ID_RECEITA)
);

CREATE TABLE VOTACAO
(
    ID_VOTO INT NOT NULL,
    NOTA DOUBLE,
    ID_PESSOA INT,
    ID_RECEITA INT,
    CONSTRAINT PK_VOTO PRIMARY KEY (ID_VOTO),
    CONSTRAINT FK_VOTACAO_PESSOA FOREIGN KEY (ID_PESSOA) REFERENCES
    PESSOA (ID_PESSOA),
    CONSTRAINT FK_VOTACAO_RECEITA FOREIGN KEY (ID_RECEITA) REFERENCES
    RECEITA (ID_RECEITA)
);

```

A assinatura da função requer alguns parâmetros que devem ser referenciados pelo id enquanto os demais seriam passados como string para o método.

```

//Inicialização:
$id_pessoa = 1; //pessoa já existente
$titulo_receita = 'Strogonoff de carne';
$data_postagem = '2016-10-02';
$modo_preparo = "
Em uma panela, adicione o óleo, a carne, a cebola, os tomates, o caldo
de carne e deixe cozinhar por 20 minutos.
Acrescente o ketchup e o champignon e deixe cozinhar até obter um
molho consistente e cremoso.
Desligue o fogo e acrescente o creme de leite sem soro.
Mexe até incorporar o molho ao creme.
";
$nomes_ingredientes = array('carne', 'cebola', 'pitadas de sal',
'pitadas de pimenta-do-reino', 'lata de creme de leite', 'tomate
picado' 'colher de sopa de azeite', 'colher de sopa de catchup', 'cubo
de caldo de carne');
$quantidades_ingredientes = array(1.0 /*kg*/, 3, 'n/a', 'n/a', 2, 3,
3, 2, 1);

//Chamada da função:
INSERE_RECEITA($id_pessoa, $titulo_receita, $data_postagem,
$modo_preparo, $nomes_ingredientes, $quantidades_ingredientes);

//Resposta da questão 1:
//OBS: Não precisa ter usado BEGIN, ROLLBACK e COMMIT para ser correto
<?php

function INSERE_RECEITA($id_pessoa, $titulo_receita, $data_postagem,
$modo_preparo, $nomes_ingredientes, $quantidades_ingredientes) {

    $link = mysql_connect('localhost','root','yourPassword');
    mysql_select_db('ad2_s2_2016', $link);
    mysql_query("BEGIN");

    $ids_ingredientes = array();

```

```

foreach($nomes_ingredientes as $nome_ingrediente) {
    $sql = "SELECT id_ingrediente FROM INGREDIENTE WHERE descricao
= '$nome_ingrediente'";
    $result = mysql_query($sql);
    if(mysql_num_rows($result)==0) {
        die("Ingrediente nao cadastrado: " . $nome_ingrediente);
    }
    $ids_ingredientes []= mysql_result($result, 0);
}
$sql = "INSERT INTO RECEITA VALUES (default, '$data_postagem',
'$titulo_receita', '$modo_preparo', $id_pessoa)";
mysql_result(mysql_query($sql), 0) or die("Erro durante insercao"
. mysql_error());

$id_receita = mysql_last_insert_id();

for($i = 0; $i < count($ids_ingredientes); $i++) {
    $id_ingrediente = $ids_ingredientes[$i];
    $qt_ingrediente = $quantidades_ingredientes[$i];
    $sql = "INSERT INTO RECEITA_INGREDIENTE VALUES ($id_receita,
$id_ingrediente, $qt_ingrediente)";
    mysql_result(mysql_query($sql), 0) or die("Erro durante
insercao" . mysql_error());
}

mysql_query("COMMIT");
}

```

2. Estenda a função INSERE\_RECEITA implementada na Questão 1 de forma que no momento da inserção de um ingrediente para uma receita, a mesma verifique a compatibilidade da quantidade do ingrediente informada na receita e sua medida. Por exemplo, ingredientes cuja medida seja "Unidade" só podem ter quantidades inteiras. Todas as outras medidas permitem valores fracionados. Caso a quantidade não seja compatível com a medida, sua função deve exibir uma mensagem informando que o valor foi aproximado por arredondamento para o inteiro mais próximo. Esse valor arredondado deve ser o armazenado no banco de dados.

R: Na primeira questão temos o seguinte loop:

```
$ids_ingredientes = array();
foreach($nomes_ingredientes as $nome_ingrediente) {
    $sql = "SELECT id_ingrediente FROM INGREDIENTE WHERE descricao
= '$nome_ingrediente'";
    $result = mysql_query($sql);
    if(mysql_num_rows($result)==0) {
        die("Ingrediente nao cadastrado: " . $nome_ingrediente);
    }
    $ids_ingredientes []= mysql_result($result, 0);
}
```

Ele pode ser adaptado da seguinte forma (em azul) para atender à questão 2:

```
$ids_ingredientes = array();
$i = 0; //para nao ter de reescrever o foreach como for
foreach($nomes_ingredientes as $nome_ingrediente) {
    $sql = "SELECT id_ingrediente, medida FROM INGREDIENTE WHERE
descricao = '$nome_ingrediente'";
    $result = mysql_query($sql);
    if(mysql_num_rows($result)==0) {
        die("Ingrediente nao cadastrado: " . $nome_ingrediente);
    }
    $unidade_de_medida = mysql_result($result, 0, 1);
    $qtd_ingrediente = $quantidades_ingredientes[$i];

    //Na minha tabela, o campo unidade possui 3 caracteres
    //A sigla para Unidade ficou representada por UND
    if($unidade_de_medida == 'UND') {
        if(!is_int($qtd_ingrediente)) {
            echo("O ingrediente $nome_ingrediente precisou ter sua
quantidade arredondada.");
            $quantidades_ingredientes[$i] =
round($qtd_ingrediente);
        }
    }
    $i++;
}
```

```

        $ids_ingredientes []= mysql_result($result, 0, 0);
    }

```

R:

3. Faça uma função em PHP que identifica entre as receitas cadastradas no último mês aquela que possui a maior média (calculada a partir do campo NOTA) dentre aquelas que tiveram pelo menos 1.000 votos. Em caso de empate a receita postada mais recentemente deve ser a selecionada.

R: Basicamente uma questão de SQL com algum código PHP mínimo (para os feras)

```

<?php
    function melhor_receita_ultimo_mes() {
        $sql = '
SELECT V.ID_RECEITA, AVG(V.NOTA)
FROM VOTACAO V NATURAL JOIN RECEITA R
WHERE DATA_ENVIO BETWEEN
    DATE_FORMAT(NOW() ,'%Y-%m-01') - INTERVAL 1 MONTH
AND
    DATE_FORMAT(NOW() ,'%Y-%m-01') - INTERVAL 1 DAY
AND (SELECT COUNT(1) FROM VOTACAO WHERE ID_RECEITA = V.ID_RECEITA)
> 1000
GROUP BY V.ID_RECEITA
ORDER BY R.DATA_ENVIO DESC LIMIT 1;
        ';

        return mysql_result(mysql_query($sql), 0) or die("Erro ao
executar consulta: " . mysql_error());

    }

?>

```

4. Faça uma função em PHP que recebe um título de uma receita e verifica se a mesma pode ser fracionada por um fator (que é um parâmetro de entrada da função). Uma receita fracionada é aquela em que seus ingredientes podem ser divididos por aquele fator. Por exemplo, uma "meia-receita" se refere a metade da quantidade de cada ingrediente utilizado na receita completa. Para verificar se uma receita pode ser fracionada, devemos checar as medidas dos seus ingredientes. As medidas sempre possíveis de serem fracionadas são Kilograma, Grama e Litro. Ingredientes cuja medida seja "Unidade" só são fracionáveis por um fator caso sejam múltiplos desse fator. Por exemplo, se uma receita contiver 2 ovos e o fator informado for 0,5, esta receita poderá ser fracionada. Usando o fator 0,3 para a mesma quantidade de ovos, a resposta seria negativa. Sua função deve exibir como saída o título da receita e o seu modo de preparo, além do nome e da quantidade fracionada de cada ingrediente.



R:

Principal erro que deve ser evitado: Fator é um número entre 0 e 1, representando uma fração. O fator deve ser MULTIPLICADO pelo valor do ingrediente e não dividido, do contrário, supondo quantidade = 2 e fator = 0,5 ou  $\frac{1}{2}$ :  $2 / \frac{1}{2} = 2 * 2 = 4$ . No lugar de dividir a receita, vocês a estariam dobrando.

```
function fraciona_receita($titulo, $fator) {

    $link = mysql_connect('localhost','root','yourPassword');
    mysql_select_db('ad2_s2_2016', $link);

    $sql = "SELECT * FROM RECEITA WHERE titulo = '$titulo'";

    $result = mysql_query($sql) or die("Erro ao executar consulta: "
    . mysql_error());

    if(mysql_num_rows($result)==0) {
        die("Receita nao cadastrada: " . $nome_ingredientes);
    }

    $receita = mysql_fetch_object($result);

    $sql = "SELECT r.id_ingredientes, r.quantidade, i.descricao,
    i.medida FROM RECEITA_INGREDIENTE r NATURAL JOIN INGREDIENTE i WHERE
    id_receita = $id_receita;";

    $result = mysql_query($sql) or die("Erro ao executar consulta: "
    . mysql_error());

    if(mysql_num_rows($result)==0) {
        die("Receita nao possui ingredientes.");
    }
    $receita->ingredientes = array();
    while ($row = mysql_fetch_assoc($result)) {
        if($row["medida"] == 'UND') {
            //O jeito mais simples de verificar se nao há parte
            //fracionaria é com módulo de 1 igual a zero
            if(!(($row["quantidade"] * $fator) % 1 == 0)) {
                die("Receita não pode ser fracionada");
            }
        }
    }
}
```

```
        }
    }
    $receita->ingredientes []= array(
        'descricao' => $row["descricao"],
        'quantidade' => $quantidades[] = $row["quantidade"] *
    $fator
    );

}
print_r($receita);
}
```