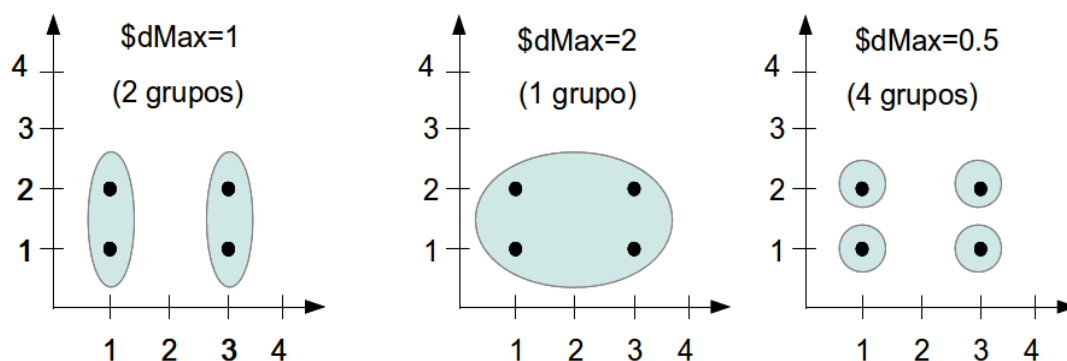


1. (5 pontos) Escreva a função `agrupaPontos($q, $dmax)` que agrupa os pontos do array `$q` com base na distância máxima `$dmax`. Um ponto `A` de `$q` pertence a um grupo `G` se existe algum outro ponto `A'` em `G` tal que a distância entre `A` e `A'` é menor ou igual a `$dmax`. Isto implica que se `A` está a uma distância maior que `$dmax` de qualquer outro ponto em `$q`, então `A` está sozinho em um grupo. De forma semelhante, dois grupos distintos `G / G'` não podem possuir pontos `A ∈ G / A' ∈ G'` tal que a distância entre `A` e `A'` seja menor ou igual a `$dmax`. Cada ponto é representado por um array de dois elementos correspondentes às coordenadas `x` e `y` do ponto. O resultado é um array de grupos, onde cada grupo é um array de pontos.

Eis um exemplo de utilização da função `agrupaPontos` sobre um mesmo conjunto `$q` e três valores de `$dmax`:



A figura acima corresponde ao trecho de código abaixo:

```
$q = array (array (1,1), array (1,2), array (3,1), array (3,2));  
echo "Para distância maxima 1:"; print_r(agrupaPontos ($q, 1));  
echo "Para distância maxima 2:"; print_r(agrupaPontos ($q, 2));  
echo "Para distância maxima 0.5:"; print_r(agrupaPontos ($q, 0.5));
```

Cujo resultado impresso é:

```
Para distância maxima 1:Array (  
    [0] => Array (  
        [0] => Array (  
            [0] => 1  
            [1] => 1
```

```

        )
        [1] => Array (
            [0] => 1
            [1] => 2
        )
    )
    [1] => Array (
        [0] => Array (
            [0] => 3
            [1] => 2
        )
        [1] => Array (
            [0] => 3
            [1] => 1
        )
    )
)
Para distância maxima 2:Array (
    [0] => Array (
        [0] => Array (
            [0] => 1
            [1] => 1
        )
        [1] => Array (
            [0] => 1
            [1] => 2
        )
        [2] => Array (
            [0] => 3
            [1] => 2
        )
        [3] => Array
        (
            [0] => 3
            [1] => 1
        )
    )
)
Para distância maxima 0.5:Array (
    [0] => Array (
        [0] => Array (
            [0] => 1
            [1] => 1
        )
    )
    [1] => Array (
        [0] => Array (
            [0] => 1
            [1] => 2
        )
    )
    [2] => Array (
        [0] => Array

```

```

        (
            [0] => 3
            [1] => 1
        )
    )
    [3] => Array (
        [0] => Array (
            [0] => 3
            [1] => 2
        )
    )
)

```

Resposta:

```

function agrupaPontos($q, $dmax) {
    if(!is_array($q) || count($q) <= 1)
        return array();

    $indice_grupo = array_fill(0, count($q), -1);
    $grupos = array();

    for($i=0; $i<(count($q)-1); $i++) {
        for($j=$i+1; $j<count($q); $j++) {
            if(euclidian_dist($q[$i][0], $q[$i][1],
                              $q[$j][0], $q[$j][1]) <= $dmax) {
                if($indice_grupo[$i] != -1) {
                    if($indice_grupo[$j] == -1) {
                        $indice_grupo[$j] = $indice_grupo[$i];
                        $grupos[$indice_grupo[$i]][] = $q[$j];
                    }
                } else {
                    $indice_grupo[$i] = count($grupos);
                    $indice_grupo[$j] = $indice_grupo[$i];
                    $grupos[$indice_grupo[$i]] = array($q[$i], $q[$j]);
                }
            }
        }
    }

    for($i=0; $i<count($q); $i++)
        if($indice_grupo[$i] == -1)
            $grupos[] = $q[$i];

    return $grupos;
}

function euclidian_dist($x1, $y1, $x2, $y2) {
    return sqrt(($x2-$x1)*($x2-$x1)+($y2-$y1)*($y2-$y1));
}

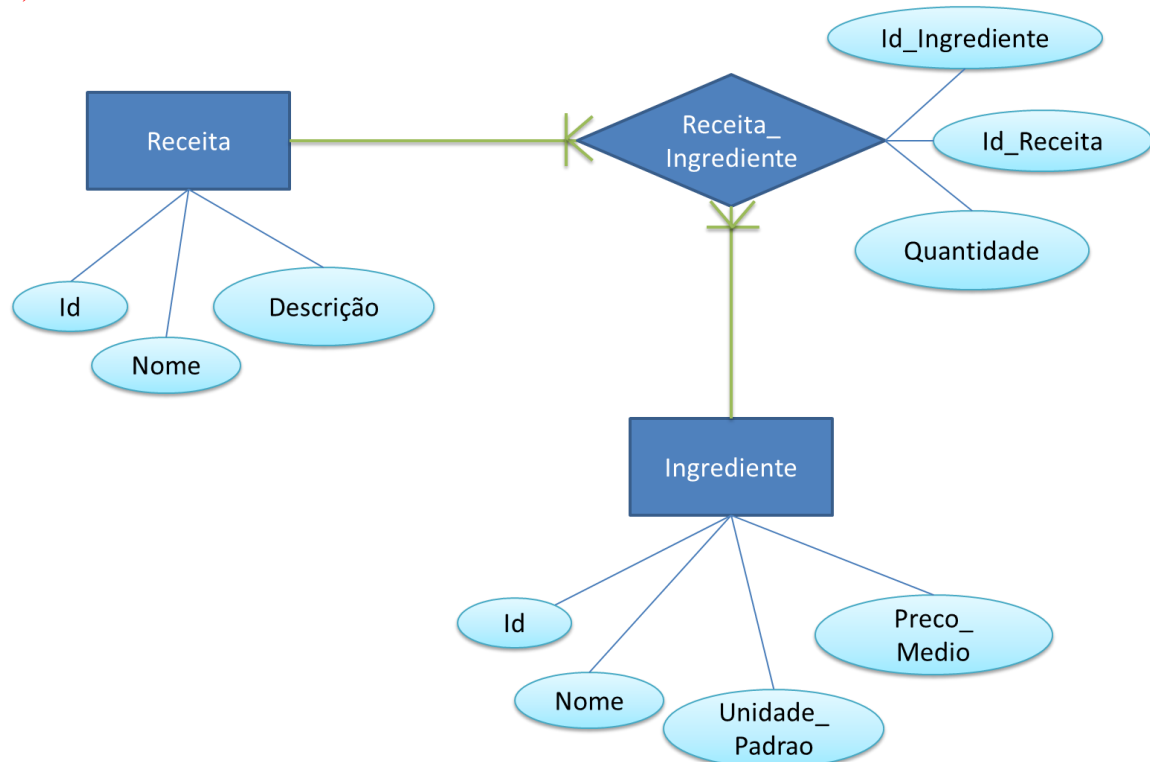
?>

```

2. (5 pontos) Deseja-se montar um banco de dados de receitas culinárias. Cada receita tem um nome (string), um texto de descrição (string) e uma lista de ingredientes. Cada tipo de ingrediente tem as propriedades nome (string), o nome da unidade padrão com que normalmente constam em receitas - 'gramas' para manteiga, 'unidades' para ovos, 'pitadas' para sal, etc, e um preço médio por unidade padrão (decimal). Pede-se

- (1 ponto) Desenhe um esquema de entidades e relacionamentos para seu banco de dados e escreva uma modelagem física em SQL
- (1 ponto) Escreva em SQL comandos para registrar no banco o ingrediente 'manteiga' com unidades-padrão 'gramas' e preço médio por grama '0.5'.
- (1 ponto) Escreva em SQL comandos para registrar uma receita de bolo com 200 gramas de manteiga, 4 xícaras de farinha, 4 ovos e 300 gramas de açúcar.
- (2 pontos) Escreva em PHP uma função `listaReceitas($ingr)` que gera um documento html com todas as receitas que levam o ingrediente com nome igual a `$ingr` ordenadas por quantidade daquele ingrediente.

a)



b)

```

CREATE TABLE receita
(
    id          INT NOT NULL auto_increment,
    nome        VARCHAR(255) UNIQUE NOT NULL,
    descricao   VARCHAR(1000),
    PRIMARY KEY (id)
);

CREATE TABLE ingrediente
(
    id          INT NOT NULL auto_increment,
    nome        VARCHAR(255) UNIQUE NOT NULL,
    unidade_padrao VARCHAR(127) NOT NULL,
    preco_medio DECIMAL(15, 2) NOT NULL,
    PRIMARY KEY (id)
);
  
```

```
CREATE TABLE receita_ingrediente
(
    id_receita      INT NOT NULL,
    id_ingrediente  INTEGER NOT NULL,
    quantidade      VARCHAR(127) NOT NULL,
    PRIMARY KEY (id_receita, id_ingrediente),
    FOREIGN KEY (id_receita) REFERENCES receita(id),
    FOREIGN KEY (id_ingrediente) REFERENCES ingrediente(id)
);
```

c) Supondo o banco pré-populado com os ingredientes especificados na pergunta, a cláusula UNIQUE na propriedade 'nome' em 'Ingrediente' e 'Receita' nos garante que o cadastro pode ser feito pelas duas queries a seguir:

```
INSERT INTO receita
    (nome,
     descricao)
VALUES      ('bolo', NULL);

INSERT INTO receita_ingrediente
    (id_receita,
     id_ingrediente,
     quantidade)
(SELECT r.id,
     i.id,
     200
  FROM   receita r,
        ingrediente i
 WHERE  r.nome = "bolo"
        AND i.nome = "manteiga")
UNION
(SELECT r.id,
     i.id,
     4
  FROM   receita r,
        ingrediente i
 WHERE  r.nome = "bolo"
        AND i.nome = "farinha")
UNION
(SELECT r.id,
     i.id,
     4
  FROM   receita r,
        ingrediente i
 WHERE  r.nome = "bolo"
        AND i.nome = "ovo")
UNION
(SELECT r.id,
     i.id,
     300
```

```

FROM    receita r,
        ingrediente i
WHERE   r.nome = "bolo"
        AND i.nome = "açúcar");

```

d)

```
<?php
```

```

function listaReceitas($ingr)
{
    $html = '<html><body>';
    $link = mysql_connect('localhost', 'mysql_user', 'mysql_password');

    if (!$link)
    {
        die('Nao foi possivel conectar: ' . mysql_error());
    }

    $result = mysql_query('SELECT    r.nome,
                                   ri.quantidade
                           FROM      receita_ingrediente ri
                           INNER JOIN receita r
                                   ON r.id = ri.id_receita
                           INNER JOIN ingrediente i
                                   ON i.id = ri.id_ingrediente
                           WHERE i.nome = ' . $ingr . '
                           ORDER BY ri.quantidade DESC ');

    if (!$result)
    {
        return $html . '<p>Nenhuma receita foi encontrada.</p></body></html>';
    }

    $html.= '<ul>';
    while ($row = mysql_fetch_assoc($result))
    {
        $html.= '<li>' . $row['nome'] . ' ' . $row['quantidade'] . '</li>';
    }
    $html.= '</ul>';

    mysql_close($link);

    return $html . '</body></html>';
}

```