

## Programação II

### Avaliação à Distância 1 - Gabarito

- 1) Considere uma aplicação web típica onde um usuário acessa uma página web para fazer uma compra numa loja virtual. Assuma que a aplicação web foi implementada no modelo de 3 camadas. Dê uma possível sequência de passos desde a consulta ao artigo ao desejado até a aquisição do produto. Descreva cada etapa claramente e indique em que camada cada etapa é realizada.

Resp.

1. Login na loja virtual
  1. Entra usuário e senha (1a camada - Navegador)
  2. Servidor recebe dados do usuário e consulta o banco de dados (2a camada - Servidor)
  3. Banco de dados executa a consulta e envia ao servidor (3a camada – Banco de Dados)
  4. Servidor recebe a resposta do banco de dados, valida os dados do usuário e envia a resposta para o cliente (2a camada - Servidor)
  5. É exibida página com login validado ou é re-exibido o formulário de login (1a camada - Navegador)
2. Consulta de produtos
  1. Usuário preenche detalhes da busca (1a camada)
  2. Servidor recebe os dados da busca e consulta no banco de dados (2a camada)
  3. Banco de dados executa a consulta e retorna os dados encontrados para o servidor (3a camada)
  4. Servidor retorna os dados formatados para o cliente (2a camada)
  5. Resultado da consulta é exibida no cliente (1a camada)
3. Incluir produto no carrinho
  1. Submete inclusão do item na compra (1º camada)
  2. Servidor recebe o pedido e o guarda numa variável temporária (2º camada) / ou o Cliente guarda numa variável temporária (1º camada)
4. Repete-se os passos 2 e 3
5. Totalizar compra
  1. Pedido para totalizar a compra (1a camada)
  2. Servidor recebe o pedido, e consulta o banco de dados para confirmar os valores dos produtos (2a camada)
  3. Banco de dados retorna os valores dos produtos solicitados (3a camada)
  4. Servidor faz o cálculo do total da compra e envia para o cliente (2a camada)
  5. Cliente exibe o resultado (1a camada)
6. Comprar

1. Confirmação de compra (1a camada)
2. Servidor recebe a confirmação e envia o formulário para pagamento (2a camada)
3. Usuário preenche os dados do pagamento (1a camada)
4. Servidor verifica os dados (2a camada)
5. Servidor finaliza a compra e gera nota de venda e o envia para o cliente (2a camada)
6. Cliente recebe a nota de venda (1a camada).

2) O que é exibido pelo seguinte fragmento de código PHP?

```
foreach (array("x"=>1, "y", 3=>"z") as $a=>$b) {
    echo "$a=$b <br/>";
}
```

Resp.

```
x=1
0=y
3=z
```

3) Considere a seguinte saída:

```

      *
     * *
    *  *
   *    *
  *      *
 *        *
*          *
 *        *
  *      *
   *    *
    *  *
     * *
      *
```

```

      *
     * *
    *  *
   *    *
  *      *
 *        *
*          *
 *        *
  *      *
   *    *
    *  *
     * *
      *
```

```

      *
     * *
    *  *
   *    *
  *      *
 *        *
*          *
 *        *
  *      *
   *    *
    *  *
     * *
      *
```

Este texto foi obtido com o auxílio da função `losango` como no trecho de código abaixo. Escreva o corpo dessa função de tal forma que a saída acima seja obtida.

```
function losango($n)
{
    ...
}
```

```

losango (5);
losango (4);
losango (3);

```

Resp.:

```

function losango($n)
{
    echo "<pre>\n";
    $s = '*'.str_repeat(' ', 2*$n-3).'*';
    for ($i = -($n-1); $i < $n; $i++) {
        echo str_repeat(' ', abs($i));
        echo substr_replace ($s, ' ', 1, 2*abs($i) );
        echo "\n";
    }
    echo "</pre>";
}

```

- 4) Frequentemente, o uso de aspas em textos web é feito com aspas simples, como por exemplo:  
**O candidato respondeu: "Esta questão é fácil".**

Como você faria para empregar expressões regulares para transformar textos contendo aspas simples em textos equivalentes, porém usando aspas tipográficas, isto é, caracteres abre- e fecha-aspas, cujos códigos html são, respectivamente, &ldquo; e &rdquo;. O texto acima, por exemplo seria exibido como:

**O candidato respondeu: “Esta questão é fácil”.**

Dica: Pode-se inferir que um abre-aspas ocorre sempre no início do texto ou então com um caractere branco à sua esquerda, enquanto que um fecha-aspas segue um caracter alfanumérico ou um sinal de pontuação.

Resp.

```

$text = ereg_replace("\"([A-Za-z0-9])\"", "&ldquo;\1", $text);
$text = ereg_replace(" \\"", " &ldquo;", $text);
$text = ereg_replace("([A-Za-z0-9.,;:!?])\"", "\\1&rdquo;", $text);

```

- 5) Explique brevemente a diferença entre as seguintes junções

- JOIN
- CROSS JOIN
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- NATURAL JOIN

Para validar sua justificativa crie duas tabelas simples, preencha-as, e escreva consultas utilizando

estas junções.

Resp.:

São cláusulas SQL que servem para fazer consultas relacionando duas tabelas. JOIN, INNER JOIN e CROSS JOIN realizam junções *internas*, isto é, cada linha do resultado corresponde a uma seleção de linhas do produto cartesiano das duas tabelas. NATURAL JOIN é semelhante, no entanto já subentende que a condição de seleção é a igualdade de atributos com mesmo nome em ambas as tabelas.

Por outro lado, LEFT JOIN e RIGHT JOIN são junções *externas*, isto é, as linhas dos resultados podem ser constituídas de valores nulos caso não seja possível encontrar um emparelhamento que atenda à condição de seleção. No caso do LEFT JOIN, atributos da tabela à esquerda podem ser emparelhados com nulos, enquanto que no RIGHT JOIN, atributos da tabela à direita podem ser emparelhados com valores nulos.

- 6) Considere as seguintes tabelas, **esportes** e **nomes** respectivamente. Este banco de dados simples é empregado para armazenar nomes de pessoas e os esportes de que elas gostam.

**esportes**

id	esporte
1	Basquete
2	Futebol
3	Vôlei
4	Corrida
5	Caminhada

**nomes**

id	id_esporte	nome
1	1	Jorge
2	2	Rafael
3	2	Julio
4	4	Elissa
5	2	Ricardo
6	3	Cesar
7	2	Flavio
8	2	Adriano
9	4	Viviane
10	4	Jayme
11	3	Carla
12	1	Robson
13	1	Marcelo

Por exemplo, se nós quiséssemos saber qual é o esporte favorito do **Julio**, teríamos que escrever

uma consulta similar a:

```
SELECT n.nome, e.esporte
FROM nomes AS n, esportes AS e
WHERE n.id_esporte = e.id AND n.nome = "Julio";
```

Pergunta-se:

- a. Escreva uma consulta que permita listar todas as pessoas registradas e os esportes de que elas gostam.

Resp.:

```
SELECT n.nome, e.esporte FROM nomes as n, esportes as e
WHERE n.id_esporte = e.id;
```

- b. Escreva uma consulta para obter o mesmo resultado utilizando construções JOIN.

Resp.

```
SELECT n.nome, e.esporte FROM nomes as n JOIN esportes as e
ON n.id_esporte = e.id;
```

- c. Encontre uma consulta que produza o seguinte resultado

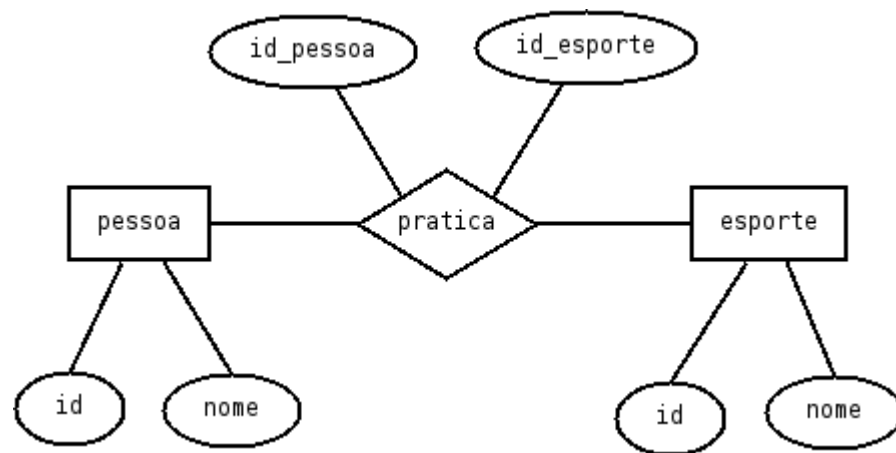
nome	esporte
Jorge	basquete
Marcelo	basquete
Robson	basquete
Elissa	corrida
Jayme	corrida
Viviana	corrida
Adriano	futebol
Flavio	futebol
Julio	futebol
Rafael	futebol
Ricardo	futebol
Carla	vôlei
Cesar	vôlei

Resp.

```
SELECT n.nome, e.esporte FROM nomes as n, esportes as e
WHERE n.id_esporte = e.id order by e.esporte, n.nome;
```

- d. Este banco de dados assume que cada pessoa só gosta de um esporte. Suponha agora que qualquer pessoa pode gostar de mais de um esporte, ou mesmo de nenhum esporte. Pedese:
  1. Crie um Modelo de Entidade Relacionamento apropriado.

Resp.



2. Escreva um modelo físico correspondente.

Resp.

```
create table pessoa(id int not null auto_increment,
nome varchar(20),
primary key (id)
);
create table esporte(
id int not null auto_increment,
nome varchar(20),
primary key (id)
);
create table pratica(
id_pessoa int not null,
id_esporte int not null,
primary key(id_pessoa, id_esporte)
);
```

3. Que consulta você usaria para obter os nomes das pessoas que não gostam de nenhum esporte?

Resp.:

```
SELECT p.nome FROM pessoa p left join pratica pr on
p.id = pr.id_pessoa group by p.nome having
count(pr.id_esporte) = 0
```