Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação.

Disciplina: Programacao II

AD2 - 1° semestre de 2011 - GABARITO

Execício 1:

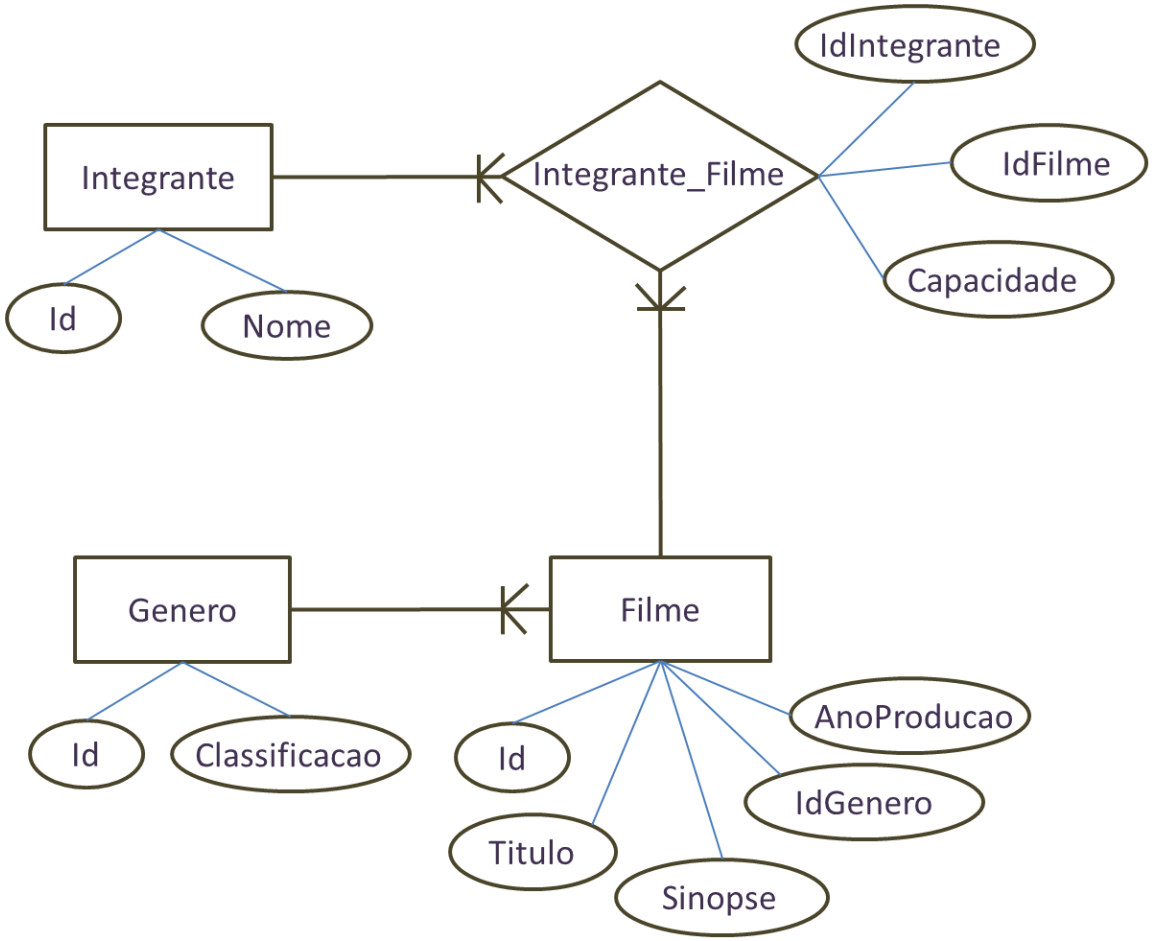
A implementação abaixo contém algumas simplificações, dando enfoque a parte conceitual. Para ver o exemplo completo, consulte o anexo no final do arquivo.

```
/* Constante que indica casa fechada. */
define('F', -2);
/* Constante que indica casa aberta com uma bomba = fim de jogo. */
define('B', -1);
/* Dimensão do tabuleiro. */
define('DIM', 8);
/* Quantidade total de minas. */
define('TOTAL_MINAS', 10);

/**
* Altera o tabuleiro de acordo com a casa selecionada.
* @param array $minas matriz contendo as posições de minas
* @param array $tab matriz contendo o estado atual do tabuleiro
* @param int $linha da casa selecionada nesta jogada
* @param int $coluna da casa selecionada nesta jogada
*/
function sweep($minas, &$tab, $linha, $coluna) {

        //Se houver mina na posição escolhida, você perdeu o jogo
        if($minas[$linha][$coluna]) {
            echo "Voce perdeu";
            $tab[$linha][$coluna] = B;

        } else {
            if($tab[$linha][$coluna] != F) return;

            $contador_minas = 0;
            //itere sobre as 8 casas ao redor da casa atual
            for ($i = $linha-1; $i <= $linha+1; $i++) {
                for ($j = $coluna-1; $j <= $coluna+1; $j++) {

                    //Se alguns dos índices do loop sair dos limites da matriz,
                    //ou for igual à casa atual, ignore a iteração
                    if($i < 0 || $j < 0 || $i >= DIM || $j >= DIM
                        || ($i == $linha && $j == $coluna)) continue;

                    //Do contrario, é um vizinho válido, e se neste houver mina,
                    //deve entrar no contador
                    if($minas[$i][$j]) $contador_minas++;
                }
            }
            //casa atual recebe o número de minas na vizinhança
            $tab[$linha][$coluna] = $contador_minas;

            //expande casas vizinhas caso o contador seja zero
            if($contador_minas == 0) {
                  for ($i = $linha-1; $i <= $linha+1; $i++) {
                        for ($j = $coluna-1; $j <= $coluna+1; $j++) {

                              if($i < 0 || $j < 0 || $i >= DIM || $j >= DIM
                                 || ($i == $linha && $j == $coluna)) continue;
```

```
                        sweep($minas, $tab, $i, $j);
                    }
                }
            }


            //testa condicão de vitória
            $casas_fechadas = 0;
            for ($i = 0; $i < DIM; $i++) {
                for ($j = 0; $j < DIM; $j++) {
                    if($tab[$i][$j] == F) $casas_fechadas++;
                }
            }
            //Se todas as casas fechadas corresponderem a minas

            if($casas_fechadas == TOTAL_MINAS) {
                echo "Voce venceu";
            }
        }

}
```

Exercício 2:

1) Uma modelagem sugerida:



A estrutura de dados ficará:

```
CREATE TABLE filme (
  Id int(11) NOT NULL AUTO_INCREMENT,
  Titulo varchar(255) NOT NULL,
  Sinopse text,
  AnoProducao date NOT NULL,
  IdGenero int(11) NOT NULL,
  PRIMARY KEY (Id),
  KEY FK_Genero (IdGenero)
);

CREATE TABLE genero (
  Id int(11) NOT NULL AUTO_INCREMENT,
  Classificacao varchar(255) NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE integrante (
  Id int(11) NOT NULL AUTO_INCREMENT,
  Nome varchar(255) NOT NULL,
  PRIMARY KEY (Id)
);

CREATE TABLE integrante_filme (
  IdIntegrante int(11) NOT NULL,
  IdFilme int(11) NOT NULL,
  Capacidade varchar(255) NOT NULL,
  PRIMARY KEY (`IdIntegrante`,`IdFilme`),
  KEY FK_Integrante (IdIntegrante),
  KEY FK_Filme (IdFilme)
);
```

OBS: As CONSTRAINTS de chaves estrangeiras foram omitidas.

2) Consultas SQL para este banco. As consultas abaixo assumem que o banco de dados em questão acaba de ser criado e está vazio. Do contrário, precisaríamos garantir que a operação de comparação relacionasse necessariamente campos com constraints tipo UNIQUE ou chaves estrangeiras (e para tanto precisaríamos de código PHP para guardar os ids inseridos e reutilizá-los):

i) Inserção de Filme Harry Porter e atores

```
INSERT INTO `genero` (Classificacao) VALUES ('Aventura');

INSERT INTO `filme` (Titulo, AnoProducao, IdGenero) SELECT 'Harry Potter e as relíquias da morte – Parte I', '2010-01-01', `genero`.Id FROM `genero` WHERE Classificacao = 'Aventura';

INSERT INTO `integrante` (Nome) VALUES ('Daniel Radcliffe'),('Emma Watson'),('Rupert Grint'),('David Yates'),('Steve Kloves'),('J. K. Rowling');

INSERT INTO integrante_filme (IdIntegrante, IdFilme, Capacidade) SELECT integrante.id, filme.id, 'Ator' FROM integrante, filme WHERE integrante.Nome IN ('Daniel Radcliffe', 'Emma Watson', 'Rupert Grint') AND filme.Titulo = 'Harry Potter e as relíquias da morte – Parte I';

INSERT INTO integrante_filme (IdIntegrante, IdFilme, Capacidade) SELECT integrante.id, filme.id, 'Diretor' FROM integrante, filme WHERE integrante.Nome = 'David Yates' AND filme.Titulo = 'Harry Potter e as relíquias da morte – Parte I';

INSERT INTO integrante_filme (IdIntegrante, IdFilme, Capacidade) SELECT integrante.id, filme.id, 'Escritor' FROM integrante, filme WHERE integrante.Nome IN ('Steve Kloves', 'J. K. Rowling') AND filme.Titulo = 'Harry Potter e as relíquias da morte – Parte I';
```

ii) Filmes em que Daniel Radcliffe foi ator

```
SELECT filme.Titulo FROM `integrante_filme`,`filme` WHERE integrante_filme.IdFilme = filme.Id AND Capacidade = "Ator" AND IdIntegrante = (SELECT Id FROM `integrante` WHERE Nome = 'Daniel Radcliffe');
```

iii) Filmes em que Daniel e Emma atuaram juntos ordenados por ano

```
SELECT filme.Titulo FROM `integrante_filme` i1,`integrante_filme` i2,`filme` WHERE i1.IdFilme = filme.Id AND i2.IdFilme = filme.Id AND i1.Capacidade = "Ator" AND i2.Capacidade = "Ator" AND i1.IdIntegrante = (SELECT Id FROM `integrante` WHERE Nome = 'Daniel Radcliffe') AND i2.IdIntegrante = (SELECT Id FROM `integrante` WHERE Nome = 'Emma Watson') ORDER BY filme.AnoProducao
```

3) A implementação das telas PHP+HTML pode ser feita da seguinte forma:

- Campos <form> englobando a região de exibição dos resultados, cujo atributo action redireciona para a própria página php, utilizando method = get ou post.
- Um botão de atributo type = submit, que é responsável pelo envio das informações do formulário de volta ao PHP. Ele deve estar dentro da região do form.
- Quando a pagina é recarregada, deve haver um if em php para saber se valores foram enviados para a variável global $_GET ou $_POST, variando de acordo com o method. Caso positivo, o PHP deve recuperar os valores que estão nestas variáveis, realizar a consulta ao banco de dados, iterar sobre esses resultados, e imprimir usando echo a estrutura tabular que se deseja exibir.

- As chamadas do PHP ao MySQL seguem o seguinte modelo:

```
mysql_connect("localhost", "user", "password") or die(mysql_error());
mysql_select_db("meu_banco");
$query = "SELECT id, nome FROM tabela";
$result = mysql_query($query);

while ($row = mysql_fetch_array($result)) {
    // Código que gera a estrutura dos resultados
}
```

- As consultas para cada caso serão:

a) Ficha Técnica do Filme

```
$query = 'SELECT integrante.Nome, integrante_filme.Capacidade FROM integrante_filme,
integrante WHERE integrante_filme.IdIntegrante = integrante.Id AND
integrante_filme.IdFilme = '.$id_filme;
```

b) Filmes dos quais um determinado integrante participou

```
$query = 'SELECT filme.Titulo FROM integrante_filme,filme,integrante WHERE
integrante_filme.IdFilme = filme.Id AND integrante_filme.IdIntegrante = integrante.Id
AND integrante.Nome LIKE "%'.$nome_integrante.'%"';
```

*Anexo: Implementação para testar funcionamento do campo minado do exercício 1

```php
<?php

/* Constante que indica casa fechada. */
define('F', -2);
/* Constante que indica casa aberta com uma bomba = fim de jogo. */
define('B', -1);
/* Dimensão do tabuleiro. */
define('DIM', 8);
/* Quantidade total de minas. */
define('TOTAL_MINAS', 10);
/* Constante que indica que o jogo não acabou. */
define('JOGO_CONTINUA', 2);
/* Constante que indica que o jogador venceu. */
define('VITORIA', 1);
/* Constante que indica que o jogador perdeu. */
define('DERROTA', 0);

/**
 * Classe que representa um tabuleiro do jogo campo minado.
 */
class JogoCampoMinado {

    /**
     * Inicializa o tabuleiro para um novo jogo.
     * @param &tab referência para tabuleiro.
     */
    function novo_jogo(&$tab, &$minas) {
        for ($i = 0; $i < DIM; $i++) {
            for ($j = 0; $j < DIM; $j++) {
                $tab[$i][$j] = F;
                $minas[$i][$j] = FALSE;
            }
        }
        $minas_colocadas = 0;
        while ($minas_colocadas < TOTAL_MINAS) {
            $sorteio = rand(0, DIM * DIM - 1);
            $i = $sorteio / DIM;
            $j = $sorteio % DIM;
            if (!$minas[$i][$j]) {
                $minas[$i][$j] = TRUE;
                $minas_colocadas++;
            }
        }

    }

    /**
     * Altera o tabuleiro de acordo com a casa selecionada.
     * @param array $minas matriz contendo as posições de minas
     * @param array $tab matriz contendo o estado atual do tabuleiro
     * @param int $linha da casa selecionada nesta jogada
     * @param int $coluna da casa selecionada nesta jogada
     * @return constante indicando estado do jogo
     */
    function sweep($minas, &$tab, $linha, $coluna) {
        //Se houver mina na posição escolhida, você perdeu o jogo
        if($minas[$linha][$coluna]) {
            echo "Voc&ecirc; perdeu";
            $tab[$linha][$coluna] = B;
            return DERROTA;
        } else {
            if($tab[$linha][$coluna] != F) return JOGO_CONTINUA;

            $contador_minas = 0;
            //itere sobre as 8 casas ao redor da casa atual
```

```
            for ($i = $linha-1; $i <= $linha+1; $i++) {
                for ($j = $coluna-1; $j <= $coluna+1; $j++) {
                    //Se alguns dos índices do loop sair dos limites da matriz, ou for
igual a casa atual, ignore a iteração
                    if($i < 0 || $j < 0 || $i >= DIM || $j >= DIM || ($i == $linha &&
$j == $coluna)) continue;
                    //Do contrario, é um vizinho válido, e se neste houver mina, deve
entrar no contador
                    if($minas[$i][$j]) $contador_minas++;
                }
            }
            //casa atual recebe o número de minas na vizinhança
            $tab[$linha][$coluna] = $contador_minas;

            //expande casas vizinhas caso o contador seja zero
            if($contador_minas == 0) {
                  for ($i = $linha-1; $i <= $linha+1; $i++) {
                        for ($j = $coluna-1; $j <= $coluna+1; $j++) {

                              if($i < 0 || $j < 0 || $i >= DIM || $j >= DIM
                                 || ($i == $linha && $j == $coluna)) continue;

                              $this->sweep($minas, $tab, $i, $j);
                        }
                  }
            }


            //testa condicao de vitória
            $casas_fechadas = 0;
            for ($i = 0; $i < DIM; $i++) {
                for ($j = 0; $j < DIM; $j++) {
                    if($tab[$i][$j] == F) $casas_fechadas++;
                }
            }
            //Se todas as casas fechadas corresponderem a minas
            if($casas_fechadas == TOTAL_MINAS) {
                echo "Voc&ecirc; venceu";
                return VITORIA;
            }
        }
        return JOGO_CONTINUA;
    }

    /**
     * Função utilitária para imprimir o campo minado.
     * @param array $tab grid do campo minado
     */
    function imprime_tabela($tab) {
        echo '<table border=1>';
        for ($i = 0; $i < DIM; $i++) {
            echo '<tr>';
            for ($j = 0; $j < DIM; $j++) {
                echo '<td> ';
                $valor = $tab[$i][$j];
                switch($valor){
                    case F: echo 'F';
                            break;
                    case B: echo '<b>B<b>';
                            break;
                    default:  echo $tab[$i][$j];
                }
                echo ' </td>';
            }
            echo '</tr>';
        }
        echo '</table>';
```

```
    }
}

$jogo = new JogoCampoMinado();
$jogo->novo_jogo($tab, $minas);
do {
    $sorteio = rand(0, DIM * DIM - 1);
    $linha = $sorteio / DIM;
    $coluna = $sorteio % DIM;
    $jogo->imprime_tabela($tab);
} while($jogo->sweep($minas, $tab, $linha, $coluna) == JOGO_CONTINUA);

$jogo->imprime_tabela($tab);
?>
```