



Fundação CECIERJ – Vice Presidência de Educação Superior à Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação II
AD2 – 1º Semestre de 2017

Apresentamos a seguir um **esquema** relacional simplificado (esse **script** deve ser adaptado para ser rodado no MySQL ou em outro Sistema de Banco de Dados) relativo ao cadastramento de alunos na Universidade Federal Fluminense. Considere esse esquema para resolver as questões a seguir.

```
CREATE TABLE ALUNO
(
  ID_ALUNO integer NOT NULL,
  NOME character varying,
  ENDERECO character varying,
  MATRICULA character
  varying, CR integer,
  CONSTRAINT PK_ALUNO PRIMARY KEY (ID_ALUNO)
);
```

```
CREATE TABLE PROFESSOR
(
  ID_PROFESSOR integer NOT
  NULL, NOME character varying,
  ENDERECO character varying,
  SIAPE character varying,
  MATRICULA_UFF character varying,
  CONSTRAINT PK_PROFESSOR PRIMARY KEY
  (ID_PROFESSOR) );
```

```
CREATE TABLE DISCIPLINA
(
  ID_DISCIPLINA integer NOT
  NULL, TITULO character varying,
```

```
TIPO character varying,  
ID_PROFESSOR integer,  
CONSTRAINT PK_DISCIPLINA PRIMARY KEY (ID_DISCIPLINA),  
CONSTRAINT FK_PROFESSOR" FOREIGN KEY (ID_PROFESSOR) REFERENCES  
PROFESSOR (ID_PROFESSOR)  
);
```

```
CREATE TABLE TURMA  
(  
ID_TURMA integer NOT NULL,  
ID_DISCIPLINA integer NOT NULL,  
ANO character varying,  
NUMERO_DE_VAGAS integer,  
PERIODO character varying,  
CONSTRAINT PK_TURMA PRIMARY KEY (ID_TURMA),  
CONSTRAINT FK_DISCIPLINA" FOREIGN KEY (ID_DISCIPLINA) REFERENCES  
DISCIPLINA (ID_DISCIPLINA)  
);
```

```
CREATE TABLE MATRICULA  
(  
ID_MATRICULA integer NOT NULL,  
ID_TURMA integer NOT NULL,  
ID_ALUNO integer NOT NULL,  
NOTA1 double,  
NOTA2 double,  
NOTA_FINAL double,  
CONSTRAINT PK_MATRICULA PRIMARY KEY (ID_MATRICULA),  
CONSTRAINT FK_ALUNO FOREIGN KEY (ID_ALUNO)  
REFERENCES ALUNO (ID_ALUNO),  
CONSTRAINT FK_TURMA FOREIGN KEY (ID_TURMA)  
REFERENCES TURMA (ID_TURMA)  
);
```

```
CREATE TABLE LISTA_ESPERA  
(  
ID_ESPERA integer NOT NULL,  
ID_DISCIPLINA integer NOT NULL,  
ID_ALUNO integer NOT NULL,  
CONSTRAINT PK_ESPERA PRIMARY KEY (ID_ESPERA),  
CONSTRAINT FK_ALUNO FOREIGN KEY (ID_ALUNO)  
REFERENCES ALUNO (ID_ALUNO) ,
```

```
CONSTRAINT FK_DISCIPLINA FOREIGN KEY (ID_DISCIPLINA)  
REFERENCES DISCIPLINA (ID_DISCIPLINA)  
);
```

1. Faça uma função em PHP chamada MATRICULA_ALUNO que matricula um determinado aluno em uma determinada turma da disciplina informada. Sua função deve receber como parâmetros o ID do aluno que será matriculado e o nome da disciplina escolhida e insere uma nova matrícula no banco de dados (tabela MATRICULA). Se a disciplina escolhida possuir mais de uma turma cadastrada, o aluno pode ser cadastrado em qualquer turma que possua vagas disponíveis (verificar o NUMERO_DE_VAGAS em comparação com os alunos já matriculados). Se não houver vaga para o aluno, a sua função deve inserir o mesmo em uma lista de espera (tabela ESPERA) informando qual a disciplina que não possui vagas para o aluno.

R: Script equivalente em MySQL:

```
CREATE TABLE ALUNO
(
    ID_ALUNO integer NOT NULL AUTO_INCREMENT,
    NOME VARCHAR(255),
    ENDERECO VARCHAR(255),
    MATRICULA VARCHAR(255),
    CR integer,

    CONSTRAINT PK_ALUNO PRIMARY KEY (ID_ALUNO)
);

CREATE TABLE PROFESSOR
(
    ID_PROFESSOR integer NOT NULL AUTO_INCREMENT,
    NOME VARCHAR(255),
    ENDERECO VARCHAR(255),
    SIAPE VARCHAR(255),
    MATRICULA_UFF VARCHAR(255),

    CONSTRAINT PK_PROFESSOR PRIMARY KEY (ID_PROFESSOR)
);

CREATE TABLE DISCIPLINA
(
    ID_DISCIPLINA integer NOT NULL AUTO_INCREMENT,
    TITULO VARCHAR(255),
    TIPO VARCHAR(255),
    ID_PROFESSOR integer,

    CONSTRAINT PK_DISCIPLINA PRIMARY KEY (ID_DISCIPLINA),
    CONSTRAINT FK_DISCIPLINA_PROFESSOR FOREIGN KEY (ID_PROFESSOR)
REFERENCES PROFESSOR (ID_PROFESSOR)
);
```

```

CREATE TABLE TURMA
(
    ID_TURMA integer NOT NULL AUTO_INCREMENT,
    ID_DISCIPLINA integer NOT NULL,
    ANO VARCHAR(255),
    NUMERO_DE_VAGAS integer,
    PERIODO VARCHAR(255),

    CONSTRAINT PK_TURMA PRIMARY KEY (ID_TURMA),
    CONSTRAINT FK_TURMA_DISCIPLINA FOREIGN KEY (ID_DISCIPLINA)
REFERENCES DISCIPLINA (ID_DISCIPLINA)
);

CREATE TABLE MATRICULA
(
    ID_MATRICULA integer NOT NULL AUTO_INCREMENT,
    ID_TURMA integer NOT NULL,
    ID_ALUNO integer NOT NULL,
    NOTA1 double,
    NOTA2 double,
    NOTA_FINAL double,

    CONSTRAINT PK_MATRICULA PRIMARY KEY (ID_MATRICULA),
    CONSTRAINT FK_MATRICULA_ALUNO FOREIGN KEY (ID_ALUNO) REFERENCES
ALUNO (ID_ALUNO),
    CONSTRAINT FK_MATRICULA_TURMA FOREIGN KEY (ID_TURMA) REFERENCES
TURMA (ID_TURMA)
);

CREATE TABLE LISTA_ESPERA
(
    ID_ESPERA integer NOT NULL AUTO_INCREMENT,
    ID_DISCIPLINA integer NOT NULL,
    ID_ALUNO integer NOT NULL,
    CONSTRAINT PK_ESPERA PRIMARY KEY (ID_ESPERA),
    CONSTRAINT FK_ESPERA_ALUNO FOREIGN KEY (ID_ALUNO) REFERENCES ALUNO
(ID_ALUNO),
    CONSTRAINT FK_ESPERA_DISCIPLINA FOREIGN KEY (ID_DISCIPLINA)
REFERENCES DISCIPLINA (ID_DISCIPLINA)
);

```

E o código:

```

function MATRICULA_ALUNO($ID_ALUNO, $ID_DISCIPLINA) {

    $link = mysql_connect('localhost', 'usuario', 'senha') or die('Não
foi possível conectar: ' . mysql_error());
    $db_selected = mysql_select_db('ad2s12017', $link);

    $queryTurma = "SELECT T.ID_TURMA
                    FROM    TURMA T

```

```

WHERE T.ID_DISCIPLINA = $ID_DISCIPLINA
AND T.PERIODO = (1 + (SELECT EXTRACT(MONTH FROM
CURDATE()) > 6))
AND T.ANO = (SELECT EXTRACT(YEAR FROM
CURDATE()))
AND (SELECT COUNT(1) FROM MATRICULA M, TURMA T2
WHERE M.ID_TURMA = T2.ID_TURMA AND T2.ID_TURMA = T.ID_TURMA) <
T.NUMERO_DE_VAGAS
LIMIT 1";

$result = mysql_query($queryTurma);

if($result && mysql_num_rows($result) > 0) {
    $row = mysql_fetch_row($result);
    $ID_TURMA = $row[0];
    $result = mysql_query("INSERT INTO MATRICULA(ID_ALUNO,
ID_TURMA) VALUES ($ID_ALUNO, $ID_TURMA)");
} else {
    $result = mysql_query("INSERT INTO LISTA_ESPERA(ID_DISCIPLINA,
ID_ALUNO) VALUES ($ID_DISCIPLINA, $ID_ALUNO)");
}
mysql_close($link);
}

```

2. Estenda a função MATRICULA_ALUNO implementada na Questão 1 de forma que no momento da inserção de uma matrícula para um aluno, caso não haja vaga para o aluno e exista um aluno com CR menor matriculado, o aluno com menor CR matriculado deve ir para a lista de espera e o novo aluno deve ser matriculado. Por exemplo, considere que um aluno com CR 8,0 quer se matricular e não há mais vagas. Porém, existem alunos matriculados com CR menor que o dele (7,0 e 4,0, por exemplo). Nesse caso, o aluno com CR 4,0 deve ir para a lista de espera e o aluno com CR 8,0 deve ser matriculado.

R: Para esta finalidade, só será preciso reescrever o else com letras em roxo na primeira questão:

```

(...)
} else {
    $queryAlunoMenorCR = "SELECT
        A.ID_ALUNO, T.ID_TURMA
    FROM TURMA T
    INNER JOIN MATRICULA M1 ON M1.ID_TURMA = T.ID_TURMA
    INNER JOIN ALUNO A ON M1.ID_ALUNO = A.ID_ALUNO
    WHERE A.CR < (SELECT CR FROM ALUNO WHERE ID_ALUNO =
$ID_ALUNO)
    AND T.ID_DISCIPLINA = $ID_DISCIPLINA
    AND T.PERIODO = (1 + (SELECT EXTRACT(MONTH FROM
CURDATE()) > 6))

```

```

        AND      T.ANO = (SELECT EXTRACT(YEAR FROM CURDATE()))
        AND      (SELECT COUNT(1) FROM MATRICULA M, TURMA T2 WHERE
M.ID_TURMA = T2.ID_TURMA AND T2.ID_TURMA = T.ID_TURMA) < T.NUMERO_DE_VAGAS
        ORDER BY A.CR
        LIMIT 1
    ";

```

```

$result = mysql_query($queryAlunoMenorCR);
if($result && mysql_num_rows($result) > 0) {
    $row = mysql_fetch_row($result);
    $ID_ALUNO_EXPULSO = $row[0];
    $ID_TURMA = $row[1];

    mysql_query("DELETE FROM MATRICULA WHERE
ID_ALUNO=$ID_ALUNO_EXPULSO AND ID_TURMA=$ID_TURMA");
    mysql_query("INSERT INTO MATRICULA(ID_ALUNO, ID_TURMA) VALUES
($ID_ALUNO, $ID_TURMA)");
    mysql_query("INSERT INTO LISTA_ESPERA(ID_DISCIPLINA, ID_ALUNO)
VALUES ($ID_DISCIPLINA, $ID_ALUNO_EXPULSO)");

} else {
    $result = mysql_query("INSERT INTO LISTA_ESPERA(ID_DISCIPLINA,
ID_ALUNO) VALUES ($ID_DISCIPLINA, $ID_ALUNO)");
}
}

```

3. Faça uma função em PHP que identifica entre as matrículas cadastradas no semestre informado (o PERIODO e ANO devem ser parâmetros da função) aquela que possui a maior média (campo NOTA_FINAL de MATRICULA) dentre os alunos que possuem CR maior que 8,0. Em caso de empate a matrícula inserida mais recentemente deve ser a selecionada.

R:

```

function MELHOR_MEDIA_ALUNOS_ALTO_RENDIMENTO($PERIODO, $ANO) {
    $link = mysql_connect('localhost', 'usuario', 'senha') or die('Não
foi possível conectar: ' . mysql_error());
    $db_selected = mysql_select_db('ad2s22016', $link);

    $queryMelhorMediaAlunosAltoCR =
        "SELECT ID_MATRICULA
        FROM    MATRICULA M
        INNER JOIN TURMA T ON M.ID_TURMA = T.ID_TURMA
        INNER JOIN ALUNO A ON M.ID_ALUNO = A.ID_ALUNO
        WHERE   T.PERIODO = $PERIODO
        AND     T.ANO = $ANO
        AND     A.CR > 8
        ORDER BY NOTA_FINAL DESC, ID_MATRICULA DESC";

    $result = mysql_query($queryMelhorMediaAlunosAltoCR);
}

```

```

        if($result && mysql_num_rows($result) > 0) {
            $row = mysql_fetch_row($result);
            return $row[0];
        }
        mysql_close($link);
    }
}

```

4. Faça uma função em PHP que recebe um **array** contendo IDs de alunos e nomes de disciplinas para matrículas em bloco. A sua função deve matricular os alunos em bloco seguindo as regras implementadas no item 2, i.e. os alunos com maiores CR tem prioridade de alocação nas turmas.

R:

```

function INSERE_MATRICULA_BATELADA($ARRAY) {
    $link = mysql_connect('localhost', 'root', '') or die('Não foi possível
conectar: ' . mysql_error());
    $db_selected = mysql_select_db('ad2s22016', $link);
    mysql_query("SET NAMES utf8", $link);

    foreach($ARRAY as $id_aluno => $nome_disciplina) {
        $queryIdDisciplina =
"SELECT    D.ID_DISCIPLINA
FROM      DISCIPLINA  D  INNER JOIN  TURMA  T  ON  T.ID_DISCIPLINA  =
D.ID_DISCIPLINA
WHERE     D.TITULO = '$nome_disciplina'
AND       T.PERIODO = (1 + (SELECT EXTRACT(MONTH FROM CURDATE()) > 6))
AND       T.ANO = (SELECT EXTRACT(YEAR FROM CURDATE()))
GROUP BY  D.ID_DISCIPLINA";

        $result = mysql_query($queryIdDisciplina);
        if($result && mysql_num_rows($result) > 0) {
            $row = mysql_fetch_row($result);
            $id_disciplina = $row[0];
            MATRICULA_ALUNO($id_aluno, $id_disciplina);
        } else {
            die("Nome inserido na listagem de matriculas é inválido.");
        }
    }
    mysql_close($link);
}

```