



Fundação CECIERJ – Vice Presidência de Educação Superior à Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação II
AD2 – 2º Semestre de 2016

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.
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.
3. Faça uma função em PHP que identifica entre as matrículas cadastradas no semestre informado (o `ID_MATRICULA` deve ser um parâmetro 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.
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.