

Exercícios

Rodar o script abaixo e criar as funções pedidas em SQL:

```
create table curso(  
  id                numeric(5),  
  nome              varchar(100) not null,  
  constraint curso_pkey primary key (id)  
);  
  
create table aluno(  
  rgm                numeric(6),  
  nome               varchar(60) not null,  
  curso              numeric(5) not null,  
  constraint pk_aluno_rgm primary key (rgm),  
  constraint aluno_curso_fkey foreign key (curso) references curso (id)  
    on update no action on delete no action  
);  
  
create table disciplina(  
  codigo             numeric(6),  
  nome               varchar(60) not null,  
  constraint pk_disciplina_codigo primary key (codigo)  
);  
  
create table tiponota(  
  codigo             numeric(6),  
  nome               varchar(60) not null check (nome in('p1', 'p2', 'exame')),  
  constraint tiponota_pkey primary key (codigo)  
);  
  
create table notas(  
  id                 numeric(5),  
  rgm_aluno           numeric(6),  
  codigo_disciplina   numeric(6),  
  tipo_nota           numeric(6),  
  nota                numeric(4,2),  
  constraint notas_pkey primary key (id),  
  constraint notas_codigo_disc_fkey foreign key (codigo_disciplina)  
    references disciplina (codigo) on update no action on delete no action,  
  constraint notas_rgm_aluno_fkey foreign key (rgm_aluno)  
    references aluno (rgm) on update no action on delete no action,  
  constraint notas_tipo_nota_fkey foreign key (tipo_nota)  
    references tiponota (codigo) on update no action on delete no action  
);  
  
create table situacao(  
  aluno              numeric(5),  
  situacao            varchar(100) check (situacao in('aprovado', 'reprovado', 'cursando', 'dp')),  
  disciplina          numeric(6) not null,  
  id                  serial not null,  
  constraint id primary key (id),  
  constraint disc foreign key (disciplina) references disciplina (codigo)  
    on update restrict on delete restrict,  
  constraint pessoa foreign key (aluno) references aluno (rgm)  
    on update restrict on delete restrict  
);
```

```
insert into curso (id, nome) values (1, 'tds');  
insert into curso (id, nome) values (2, 'sisinfo');
```

```
insert into aluno (rgm, nome, curso) values (1, 'bruno', 1);  
insert into aluno (rgm, nome, curso) values (2, 'ze', 2);  
insert into aluno (rgm, nome, curso) values (3, 'joao', 1);
```

```
insert into disciplina (codigo, nome) values (1, 'bd');  
insert into disciplina (codigo, nome) values (2, 'ioo');
```

```
insert into tiponota (codigo, nome) values (1, 'p1');  
insert into tiponota (codigo, nome) values (2, 'p2');
```

```
insert into notas (id, rgm_aluno, codigo_disciplina, tipo_nota, nota)  
values (1, 1, 1, 1, 9.00);  
insert into notas (id, rgm_aluno, codigo_disciplina, tipo_nota, nota)  
values (2, 1, 1, 2, 7.00);  
insert into notas (id, rgm_aluno, codigo_disciplina, tipo_nota, nota)  
values (3, 2, 1, 1, 8.00);  
insert into notas (id, rgm_aluno, codigo_disciplina, tipo_nota, nota)  
values (4, 2, 1, 2, 6.00);
```

```
insert into situacao (aluno, situacao, disciplina, id)  
values (1, 'aprovado', 1, 1);  
insert into situacao (aluno, situacao, disciplina, id)  
values (3, 'reprovado', 1, 2);
```

1. Criar uma função para saber quais notas são de determinado tipo, passando o tipo da nota por parâmetro.
2. Criar uma função para verificar se o tipo da nota existe.
3. Criar uma função para inserir uma nota na tabela de notas usando o plpgsql. Mas antes de inserir a nota verificar se o tipo da nota existe na tabela tiponota.
4. Criar uma função que retorne a nota p1 do aluno passando como parâmetro rgm e a disciplina.
5. Criar uma função que retorne a nota p2 do aluno passando como parâmetro o rgm e a disciplina.
6. Criar uma função que retorne a media do aluno $(p1+p2)/2$, passando o rgm do aluno e a disciplina como parâmetros.