

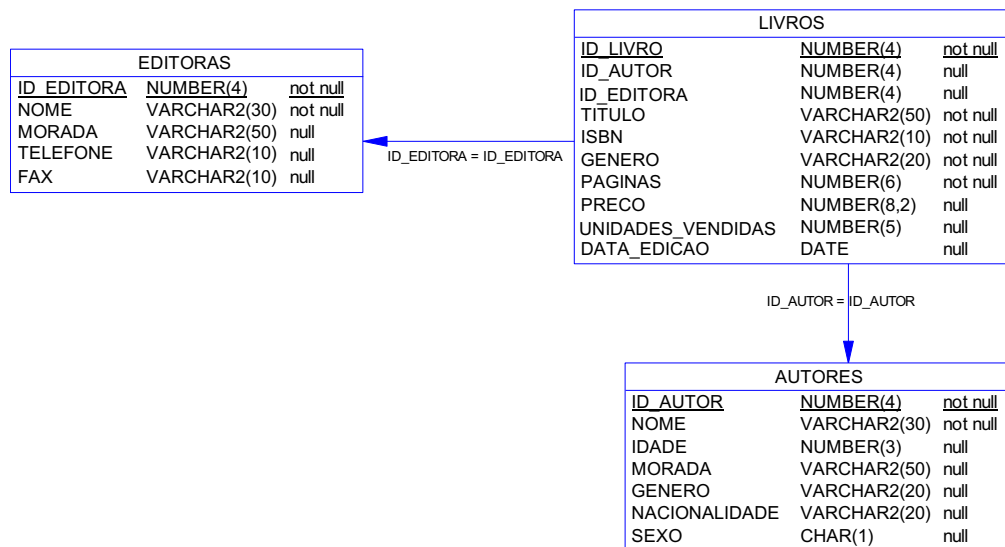
Ficha Prática 9

PL/pgSQL – Cursores

Bases de Dados

Departamento de Engenharia Informática

As tabelas seguintes pretendem modelar uma livraria com uma base de dados muito simples (o código necessário para criar as tabelas e inserir alguns dados está disponível no *inforestudante*). Existe apenas informação sobre editoras, livros e autores. Todos os livros são editados por uma editora. Essa relação é indicada através da coluna de chave forasteira, *id_editora*, na tabela *livros*, que indica um valor existente na coluna de chave primária *id_editora* da tabela *editoras*. Cada livro é escrito por um, e apenas um autor. Essa relação é indicada através da coluna de chave forasteira, *id_autor*, na tabela *livros*, que indica um valor existente na coluna de chave primária *id_autor* da tabela *autores*.



Baseando-se nas tabelas anteriores, resolva os seguintes problemas:

1. Usando a linguagem SQL, crie a seguinte tabela:

```
temp(col1 numeric(10),col2 numeric(20),message varchar(60))
```

2. Faça um programa PL/pgSQL que insira dez registos na tabela *temp* com o seguinte aspeto:

COL1	COL2	MESSAGE
1	100	Col1 é impar
2	200	Col1 é par
3	300	Col1 é impar
...
10	1000	Col1 é par

3. Apague todo o conteúdo da tabela *temp*. Faça um procedimento PL/pgSQL que receba o número de um livro (*id_livro*) por parâmetro e se este pertencer ao autor 17 o mude para o autor 80. Como este autor ainda não existe na tabela de autores, o programa deverá começar por criá-lo com a seguinte informação: *id_autor* = 80, *nome* = Luís Moreno Campos, *morada* = Lisboa, *sexo* = Masculino, *nacionalidade* = Portuguesa, *género* = Informática. Pode usar o seguinte código como referência para a criação do procedimento:

```
create or replace procedure ex3(p_livro livros.id_livro%type)
language plpgsql
as $$
declare
    ...
begin
    ...
end;
$$;
```

Pode executar o procedimento da seguinte forma:

```
call ex3(10);
```

4. Execute o programa anterior outra vez e verifique o erro obtido. A que se deve tal erro?
5. Faça um procedimento em PL/pgSQL que receba o número de um livro (*id_livro*) por parâmetro e aumente o preço desse livro de acordo com o seguinte critério:
- se custar menos de 25 é aumentado 10%;
 - se custar 25 ou mais é aumentado 6%;
 - se for um livro de aventuras não é aumentado o seu preço.
6. Analise e teste o seguinte programa em PL/pgSQL, que aumenta os preços dos livros dos géneros romance e aventuras da tabela livros, de acordo com o seguinte critério:
- os livros que custem menos de 25 são aumentados em 10%;
 - os livros que custem mais de 25 e menos de 50 são aumentados em apenas 6%;
 - os livros que custem mais de 50 não são aumentados.

```
do $$
declare
    v_id_livro livros.id_livro%type;
    v_preco livros.preco%type;
    c1 cursor for
        select id_livro,preco
        from livros
        where genero in ('Aventura','Romance') and preco<=50
        for update;
begin
    open c1;
    loop
        fetch c1 into v_id_livro,v_preco;
        exit when not found;
        if v_preco<=25 then
            update livros set preco=preco*1.1 where current of c1;
        else
            update livros set preco=preco*1.06 where current of c1;
        end if;
    end loop;
end;
$$;
```

7. Volte a escrever o programa anterior utilizando um ciclo “*for de cursor*”.
8. Faça um procedimento em PL/pgSQL que receba um gênero de livro (*genero*) por parâmetro e aumente os preços dos livros com esse gênero em 10%, de acordo com o seguinte critério:
 - só aumenta preços se o preço total de todos os livros com esse gênero for inferior a 150.
 - se obedecer à condição anterior, então começa por aumentar os livros que tenham o preço mais baixo;
 - para assim que o aumento de um livro levar à violação da primeira condição.
9. Faça um programa PL/pgSQL que copie os campos *id_livro*, *preco* e *titulo* dos 8 livros mais caros da tabela *livros* para a tabela *temp* (criada na pergunta 1).