



Exercício

Considerando o seguinte esquema de uma base de dados relacional:

Sócios (nºsócio, nome, morada, telefone, BI, Data_Nasc, Data_Insc)

Filmes (cod_filme, título, duração)

Modalidades (modalid, preço, multa_diária)

Cópias (cod_filme, nºcópia, formato, data_aquisição, preço)

Alugueres (nºaluguer, nºsócio, cod_filme, nºcópia, modalid, data_aluguer, data_entrega, preço, multa)

Sendo que:

Se (multa = 0) então (o aluguer terminou sem multa)

Se (multa > 0) então (o aluguer terminou com multa)

Se (multa é null) então (o aluguer ainda não terminou)

Responder às seguintes questões utilizando a linguagem SQL:

- 1 – Quais as cassetes (cod_filme e nºcópia) que, neste momento, não se encontram alugadas?
- 2 – Quais os sócios (nºsócio e nome) que nunca alugaram DVDs?
- 3 – Quais os filmes (cod_filme e título) que existem no clube em formato cassete e DVD?
- 4 – Relativamente a cada aluguer que terminou em multa, identificar qual o sócio envolvido (nºsócio e nome), qual o valor da multa paga, e de que filme (título) se tratava.
- 5 – Quais os sócios (nºsócio e nome) que já efectuaram alugueres em todas as modalidades existentes?
- 6 – Listar, para cada filme (cod_filme, título) o número de cassetes existentes no clube.
- 7 – Qual o maior valor de multa alguma vez ocorrido? Qual o sócio (nºsócio e nome) e filme (cod_filme e título) envolvidos?

Utilização de *views*

Recorrendo a mecanismos do tipo *view*, resolva as seguintes questões:

- 1) Qual o sócio (nº sócio e nome) com maior número de alugueres terminados em multa até ao momento?
- 2) Qual o filme (título) mais rentável do vídeo-club?

Armazenamento de código na Base de Dados

- 1) Desenvolver uma função “*exemplares_dentro*” que, dada a identificação de um filme (*cod_filme*), retorne o número de exemplares (de todos os tipos) que neste momento não se encontram alugados.
- 2) Desenvolver uma função “*multa_a_pagar*” que dada a modalidade em que um aluguer foi realizado, juntamente com o número de dias em atraso, retorna o correspondente valor da multa a pagar.
- 3) Desenvolver um procedimento “*abrir_aluguer*” com um número mínimo de parâmetros de entrada. Este procedimento deverá ser invocado sempre que se quiser iniciar um novo aluguer.
- 4) Desenvolver um procedimento “*fechar_aluguer*” com um número mínimo de parâmetros de entrada. Este procedimento deverá ser invocado sempre que se quiser terminar um novo aluguer.
- 5) Desenvolver um *trigger* que implemente a seguinte regra de negócio: “todos os clientes com mais de 5 alugueres terminados em multa, só poderão poder fazer novos alugueres na modalidade mais cara”.