



Enunciado - Parte 3

A terceira parte do trabalho prático (TP) de Bases de Dados tem como objetivo a implementação de algumas estruturas em PL/SQL. As estruturas a implementar consistem em funções, procedimentos e *triggers*, sobre a base de dados (BD) desenvolvida na segunda parte do trabalho.

Descrição das estruturas a implementar

Funções

1. Implementar uma função designada **func_stock_artigo_armazem**, para retornar a quantidade total de um artigo num determinado armazém. A função tem como parâmetros, o identificador do artigo e o identificador do armazém. No caso de o identificador do artigo ou do armazém não existir na BD, a função deve retornar o valor NULL.
2. Implementar uma função designada **func_razao_veiculo_transporte**, para obter a razão entre o número de veículos que efetuaram viagens com o maior número de encomendas e o número de veículos que efetuaram viagens com o menor número de encomendas. O cálculo desta razão deve ser feito para um dado armazém num dado intervalo de tempo. A função tem como parâmetros, o identificador do armazém e o período de tempo (data início e data fim). No caso de o identificador do armazém não existir na BD, a função deve retornar o valor NULL.
3. Implementar uma função designada **func_zona_maior_armazenamento**, para obter um cursor com os identificadores das zonas físicas de um dado armazém, que possuem a maior quantidade de artigos em *stock*. A função tem como parâmetro o identificador do armazém. No caso de o identificador do armazém não existir na BD ou a maior quantidade de artigos em *stock* for zero, a função deve retornar o valor NULL.



Procedimentos

4. Implementar um procedimento designado **proc_faturas_nao_liquidadas**, para realizar uma listagem com as faturas, de um dado período de tempo, que ainda não foram liquidadas. O procedimento tem como parâmetro o período (data de início e de fim) e no caso de não ser indicada uma das datas serão analisadas todas as faturas não liquidadas. Deve ser apresentada informação (identificador, data de faturação e valor total) das faturas em falta de cada cliente, ordenada pela data de faturação, bem como, o subtotal a liquidar por cliente. De igual forma deve ser apresentado o total de todas as faturas, de todos os que clientes, que faltam ser liquidadas à IsepBricolage.
5. Implementar um procedimento designado **proc_tratar_nota_encomenda**, para processar uma dada nota de encomenda. O procedimento tem como parâmetro o identificador da nota de encomenda. Caso não seja possível processar a nota de encomenda, deve ser armazenada toda a informação de todos os produtos em falta. No entanto, caso seja possível processar a nota de encomenda, deve ser atualizada a informação relativa ao armazém, e zona física, de onde o artigo vai sair retirado. Para além disso deve ser gerada a respetiva guia de saída. No caso de o identificador da nota de encomenda não existir na BD o procedimento deve lançar uma exceção adequada. Se se entender necessário podem ser realizadas alterações ao modelo relacional.
6. Elaborar um procedimento designado **proc_atualizar_tipo_cliente**, que atualize o valor do tipo de cliente, para todos os clientes, de acordo com o volume de negócio que efetuaram (ver Tabela 1). O período de volume de negócios a ser considerado é o dos 12 meses anteriores. No entanto, só os clientes que foram atualizados há mais de 12 meses é que podem ser atualizados. Se se entender necessário podem ser realizadas alterações ao modelo relacional.

Tabela 1 – Classificação de Tipo de Cliente

Tipo de Cliente	Volume de Negócio (€)
VIP	> 50 000
Grande Cliente	[30 000, 50 000]
Pequeno Cliente	< 30 000



Triggers

7. Implementar um *trigger* designado **trg_nota_encomenda_impedir_registo**, que impeça que seja registada uma nota de encomenda de um cliente se este tiver faturas por pagar há mais de três meses.
8. Implementar um *trigger* designado **trg_viagem_impedir_atribuicao_veiculo**, que impeça que seja atribuído um veículo a uma nova viagem, se este já efetuou o número máximo de quatro viagens permitidas por semana ou, se está indisponível.
9. Implementar um *trigger* designado **trg_guia_saida_atualizar_stock**, que garanta que o *stock* é corretamente atualizado de acordo com a informação das guias de saída, ou seja, quando se cria, altera ou apaga uma linha de uma determinada guia de saída.

Informações sobre a entrega

Data limite para entrega: **2 de dezembro até às 23:59.**

Observações:

O nome do ficheiro zip a submeter no Moodle em cada entrega intermédia deverá ter o seguinte formato:

TurmaXX_GrupoN_ParteY.zip onde,

XX - Representa a turma (e.g., TurmaDE)

N – Número do grupo (e.g., Grupo9)

Y – Indicação do número da parte do TP a entregar (e.g., Parte3)

Tenha em atenção que o ficheiro zip a ser entregue deve conter os seguintes ficheiros:

- relatório, incluindo o modelo relacional e justificações às alterações implementadas ao mesmo, da segunda para a terceira parte;
- *script* para a criação das tabelas da BD;
- *script* para preenchimento das tabelas, com registos necessários para testar as estruturas PL/SQL implementadas;
- *script* com as estruturas PL/SQL.

Bom Trabalho!