**PROJETO PL/SQL**

Informações gerais:

* Este projeto corresponde a 80% da nota G1;
* Pode ser feito individual ou em duplas;
* **Entregas**
  + **Apresentação | Demonstração: 13/10/2016 a partir das 19h30min**
  + **Códigos-fonte: 13/10/2016, até às 23h59min**
* **Enviar para** [**fahad.kalil@imed.edu.br**](mailto:fahad.kalil@imed.edu.br) **com o assunto: "[BDA-2016-2] Projeto PL/SQL"**
* **Verificando plágio, os trabalhos envolvidos terão nota final ZERO.**

Objetivo:

* Uso dos recursos da linguagem PL/SQL (Oracle) para manipulação de dados e garantia da consistência destes em um banco de dados.

Tema:

* Este projeto tem como tema a geração de um e-commerce simplificado.

**INSTRUÇÕES E ATIVIDADES**

1. Cada grupo deverá criar uma estrutura que possuam, no mínimo, as seguintes tabelas:
   1. **Cliente**

cod\_cliente [sequencial]

endereco

cidade

uf

telefone\_principal

* 1. **Produto**

cod\_produto [sequencial]

nome [ex.: Caneca Personalizada Porcelana Game Of Thrones]

status (1 = Ativo; 0 = Descontinuado)

preco\_unitario

cor

* 1. **Estoque**

*cod\_produto (primary foreign key)*

quantidade

estoque\_minimo

* 1. **Pedido**

cod\_pedido [sequencial]

data\_pedido [not null]

prazo\_entrega

data\_envio

data\_entrega

valor\_frete

desconto

* 1. **Pedido\_Entrega**

*cod\_pedido (primary foreign key)*

nome\_destinatario

endereco\_destinatario

cidade

uf

cep

* 1. **Produto\_Pedido** (muitos para muitos)

*cod\_produto (primary foreign key)*

*cod\_pedido (primary foreign key)*

quantidade

desconto

* 1. **Categoria**

cod\_categoria [sequencial]

nome

descricao

* 1. **Frete**

cod\_frete [sequencial]

cidade + uf [unique]

valor

prazo

* 1. **Fornecedores**

cod\_fornecedor [sequencial]

nome

endereço

cidade

uf

cep

pais

telefone

Restrições:

- Um pedido tem apenas um cliente relacionado;

- Um cliente pode realizar vários pedidos com um ou mais produtos incluídos;

- Cada produto deverá pertencer a uma categoria e ter apenas um fornecedor;

1. Para controle das chaves primárias sequenciais das tabelas deverão ser utilizadas *sequences* em conjunto com *triggers*, conforme demonstrado em sala de aula.
2. Criar **funções** com os seguintes propósitos:
   1. Retornar o estoque atual de determinado produto (código do produto) e disparar uma exceção caso este produto esteja descontinuado;

CREATE OR REPLACE FUNCTION estoque\_produto

(in\_cod\_produto IN Number)

return integer

AS

-- Declaracao de variavel

v\_status Number;

v\_estoque integer;

BEGIN

-- Codigo PL/SQL

-- Seleciona parametro para dentro da variavel

Select status Into v\_status from produto

where cod\_produto = in\_cod\_produto;

--

if v\_status = 0 THEN

raise\_application\_error(-20000,'Produto Descontinuado');

end if;

select e.quantidade into v\_estoque

from estoque e, produto p

where e.cod\_produto = p.cod\_produto;

return v\_estoque;

END;

select estoque\_produto('4') from dual;

* 1. Retornar a soma de todos os estoques de produtos de determinada categoria (código da categoria);
  2. Retornar o total de produtos enviados, mas ainda não entregues para determinado período de envio. Caso o período de envio seja maior que a data atual (SYSDATE), disparar uma exceção.

1. Criar **procedimentos** para executar as seguintes ações/regras:
   1. modificar\_estoque(cod\_produto IN NUMBER, quantidade IN NUMBER)
      1. Disparar exceção caso não exista o cod\_produto informado;
      2. Não permitir estoque negativo;
   2. criar\_pedido(data\_pedido IN DATE)
      1. Deve incluir dados nas tabelas de pedido;
      2. Deve retornar como OUT o código do pedido gerado.
   3. incluir\_produto\_pedido(cod\_pedido IN NUMBER, cod\_produto IN NUMBER, quantidade IN NUMBER, desconto IN NUMBER)
      1. Deve disparar uma exceção caso o cod\_pedido não existir ou produto estiver sem estoque, não executando as demais ações;
      2. Deve dar baixa na quantidade em estoque do produto indicado;
      3. Por padrão, deverá setar desconto como zero caso usuário passe valor null.
   4. definir\_entrega(cod\_pedido IN NUMBER, nome\_destinatario IN VARCHAR2, endereco\_destinatario IN VARCHAR2, cidade IN VARCHAR2, uf IN CHAR(2), cep CHAR(9))
      1. Incluir dados na tabela pedido\_entrega e setar o valor do frete na tabela pedido, conforme o valor presente na tabela de frete, bem como o prazo\_entrega.
   5. devolucao\_pedido(cod\_pedido IN NUMBER)
      1. Deve atualizar os estoques dos produtos pertencentes ao pedido;
         1. Usar CURSOR!
      2. Deve remover a entrada da tabela pedido\_entrega;
      3. Deve remover a entrada da tabela produto\_pedido;
      4. Setar a data de envio para vazio na tabela pedido;
      5. Por fim, deverá remover a entrada da tabela pedido.
2. Criar **triggers**:
   1. Ao excluir pedido, não efetivar caso este já possua data de envio preenchida.
   2. Ao excluir cliente, não efetivar caso este ainda possua pedidos não enviados ou entregues.
   3. Criar uma tabela chamada alerta que armazenará: código sequencial, data, código do produto, nome do produto, estoque atual e estoque mínimo.
      1. Inserir registro nessa tabela quando houver UPDATE na tabela estoque e o valor da coluna quantidade tenha ficado menor que o estoque\_minimo.
      2. Remover registro caso o estoque de determinado produto volte a ficar superior ou igual ao mínimo.
3. **CRIAR MAIS UM EXEMPLO DE FUNÇÃO, PROCEDIMENTO E TRIGGER A CRITÉRIO DO GRUPO.**
4. **GERAR UM ROTEIRO DE DEMONSTRAÇÃO DO FUNCIONAMENTO DOS CÓDIGOS PL/SQL GERADOS.**