

# Complementos de Bases de Dados

2018/2019



Licenciatura em Engenharia Informática

**Projeto – Época Especial**

O projeto final da disciplina de Complementos de Bases de Dados visa a familiarização com a administração de bases de dados relacionais. O trabalho incidirá ao nível da BD e SGBD e sobre a lógica subjacente à utilização dos dados para os requisitos colocados; **não se pretende o desenvolvimento das camadas da lógica aplicacional e apresentação.** Apesar do enunciado se encontrar dividido por 3 partes, com vista a uma melhor organização do trabalho; este trabalho é de entrega e discussão únicas nas data definidas na secção 3.

## 1 Descrição Geral

A empresa de material de ciclismo, AdventureWorks, está ser alvo de uma reestruturação e necessitará de desenvolver um portal de *e-commerce* uma vez que faz parte da sua estratégia a potenciação do aumento das vendas via Internet. A empresa já vendia via Internet mas os seus processos e sistemas estão desatualizados face às novas solicitações que agora se impõem.

Contudo a experiência e conhecimento que os já clientes têm da empresa não deverá ser negligenciada, e como tal, deverá ser possível no novo portal, o acesso aos seus dados de conta, encomendas anteriores, assim como à nomenclatura já existente de produtos e suas categorias.

O suporte eficiente ao portal torna necessário modelar e integrar numa nova Base de Dados os fragmentos de informação que foram exportados das aplicações existentes que suportavam até à data as vendas.

Constitui-se anexo a este enunciado um ficheiro contendo uma Base de Dados onde estão persistidos os dados existentes. No entanto estes apresentam-se fracamente relacionados, carecendo de uma otimização segundo as boas práticas de modelação e regras da normalização, com vista a providenciar um suporte eficiente das operações e ainda permitir intervenções no âmbito da sua escalabilidade.

## 2 Desenvolvimento

### 2.1 Primeira Parte

#### 2.1.1 Modelo Relacional

Considerando os requisitos abaixo enumerados deverá ser desenvolvido, e carregado, um Modelo Relacional que os assegure de forma eficiente:

1. Existência de dois tipos de utilizador do novo Portal: Cliente e Cliente VIP (que no nível aplicacional terão acesso a algumas funcionalidades diferenciadas)
2. A autenticação perante o portal será feita por recurso à conta de email e password
  - a. Para os clientes existentes a password do sistema anterior será alvo de reset

3. Os produtos estão organizados em Sub-Categorias que por sua vez se agrupam em Categorias mais gerais (deve ser facilitada a navegação e filtragem por Categoria e/ou Sub-Categoria)
4. As encomendas on-line dos utilizadores Cliente são realizadas através de um carrinho de compras que pode conter mais do que um produto (e em qualquer quantidade de cada produto, desde que disponível)
5. Contemplar a existência de promoções sobre os produtos considerando:
  - a. designação da promoção (e.g. BlackFriday), data/hora de início e fim da promoção
  - b. percentagem de desconto associada a cada produto no âmbito de cada promoção
  - c. número de unidades de cada produto disponível para promoção
6. Um produto pode participar em várias promoções e uma promoção abrange um ou mais produtos
7. Refletir na “associação” do produto às vendas a eventual promoção que abrangesse o respetivo produto
8. Gestão de utilizadores:
  - a. Editar, Adicionar e Remover Utilizadores
9. Gestão de Produtos, Categorias e Subcategorias:
  - a. Editar, Adicionar e Remover Produtos, Categorias e Sub-Categorias
  - b. Associar Produto a Sub-Categoria/Categoria
  - c. Alterar as datas (DueDate e ShipDate) de Encomendas
  - d. Alterar o Estado dos Produtos

No desenvolvimento do modelo relacional deverá considerar as opções na definição pertinente dos tipos de dados, default values, constraints (primary key, foreign key, unique e check) e triggers.

### 2.1.2 Regras (triggers e/ou SPs)

- Sempre que é realizada a venda (fecho da encomenda) de um produto em promoção, deve ser atualizado o campo “número de unidades disponíveis”
- Só é possível o cliente beneficiar dos preços das promoções que vigoram à data/hora de aquisição (fecho da encomenda), e ainda não tenha sido atingida o nº de unidades do produto em promoção
- Em caso de conflito, i.e. coexistência de promoções sobre o mesmo produto será aplicado o valor de desconto mais alto
- Quando é criada uma nova promoção deverá enviar-se aos Clientes VIPs (simular com tabela) uma mensagem com os produtos que constam na promoção

### 2.1.3 Programação

A implementação dos requisitos relativos aos pontos 8 e 9 deverá ser feita com recurso a *stored procedures*. Particularmente, relativamente à implementação dos requisitos 8.a e 9.a consultar a secção 2.1.6.

**Defina ainda** as *stored procedures* de “Criação de Encomenda/Carrinho”, “Adição de Produto a Encomenda”, “Alteração de Quantidade de Produto na Encomenda” e “Remoção de Produto de Encomenda” (*poderá ter de considerar o estado e valor da encomenda*)

## **Tratamento de Erros**

Todo o código desenvolvido no projeto deverá fazer as validações necessárias e incluir tratamento de erros. O tratamento de erros deve ser gerido de forma centralizada. Na ocorrência de um erro deverá ser enviada uma mensagem amigável ao utilizador e criar um *log* de erro que identifique o erro gerado, o utilizador (sql server) e o *timestamp* do evento.

### 2.1.4 Verificação da nova BD

Produza um conjunto de queries que dirigindo às duas bases de dados permita verificar a conformidade dos dados no novo modelo.

Requisitos mínimos:

- Total monetário de vendas por ano
- Total monetário de vendas por ano por “*Sales Territory Country*”
- Total monetário de vendas por ano por “*Product Subcategory*”
- Total monetário de vendas por ano por “*Product Category*”
- Número de Clientes por ano por “*Sales Territory Country*”

### 2.1.5 Layout da BD

Para a definição do *layout* da base de dados, deverá suportar-se num conjunto de dados, nomeadamente:

- Espaço ocupado por registo de cada tabela;
- Espaço ocupado por cada tabela com o número atual de registos;
- Propor uma taxa de crescimento por tabela (inferindo dos dados existentes);
- Dimensionar o nº e tipos de acessos.

A definição dos Filegroups com o respetivo tipo, dimensão inicial, taxa de crescimento e dimensão máxima prevista deverá ser informada pelo levantamento enumerado.

### 2.1.6 Catalogo/Metadados

#### **Stored procedures “geradores”**

As *stored procedures* a definir para suportar os requisitos 8.a e 9.a da secção 2.1.1 deverão ser geradas automaticamente a partir do catalogo da BD.

Assim com base nos metadados desenvolva um conjunto de *stored procedures*, denominados de “geradores”, que têm como argumento de entrada o nome da respetiva tabela, e como ação a criação de:

- um *stored procedure* por tabela referida que implementa a operação de *insert*;
- um *stored procedure* por tabela referida que implementa a operação de *update*;
- um *stored procedure* por tabela referida que implementa a operação de *delete*.

Os *stored procedures* gerados, devem ter como nome, *sp\_<nome\_tabela>\_<comando>* (ex: *sp\_categoria\_insert*), e no seu código devem ser verificadas todas as restrições (*not null*, *primary key*, ...) sobre os dados que estão definidas na respetiva tabela.

Quando os dados não satisfazem as restrições é parada a execução e é devolvido um identificador do erro gerado. **Cada erro deve ser identificado por um código único (ver secção anterior, “Tratamento de Erros”).**

### **Apoio à monitorização**

- ❖ Deverão ainda ser criados para efeitos de apoio à monitorização os seguintes objectos na BD:
  - Uma *stored procedure* que recorra ao catalogo para gerar entradas numa tabela(s) dedicada(s) onde deve constar a seguinte informação relativa à bases de dados: todos os campos de todas as tabelas, com o seus tipos de dados, tamanho respetivo e restrições associadas (no caso de chaves estrangeiras, deve ser indicada qual a tabela referenciada e o tipo de acção definido para a manutenção da integridade referencial nas operações de “*update*” e “*delete*”. Deverá manter histórico de alterações do esquema da BD nas sucessivas execuções da *usp*.
  - Uma *view* que disponibilize os dados relativos à execução mais recente, presentes na tabela do ponto anterior
  - Uma *stored procedure* que registe, também em tabela dedicada, por cada tabela da base de dados o seu número de registos e estimativa mais fiável do espaço ocupado. Deverá manter histórico dos resultados das sucessivas execuções da *usp*.
- ❖ **A execução destas procedures deve ser feita por recurso à criação de agendamento que possibilite a execução periodica automática.**

## 2.2 Segunda Parte

### 2.2.1 Índices e Views

Considerando que os utilizadores no papel de Gestor de Vendas necessitam de fazer alguma análises e relatórios sobre a atividade dos clientes na plataforma por forma a poder gerir *stocks*, aquisição ou abolição de produtos, etc, deverá haver um suporte eficiente aos seguintes requisitos:

1. Qual o volume de vendas por Subcategoria em trimestres homologos entre os ultimos 3 anos
2. Calcular por produto o volume de vendas total e o volume de vendas efetuado em promoção
3. Qual percentagem de vendas por produto efetuada com promoção
4. Qual o volume de vendas por produto, considerando o Top 10, em trimestres homologos entre os ultimos 3 anos
5. Prazo médio entre data de encomenda e envio por Região Geográfica, (consideração dos ultimos 2 anos)
6. Stored procedure que realiza o fecho de cada ano inscrevendo numa tabela de arquivo o valor total gasto por Cliente em cada ano e onde figura também o seu *YearlyIncome*
7. Criar uma view (e.g. *vPromoBlackFriday*) por cada promoção, com a lista de produtos que foi adquirida (vendas) durante a promoção:

Nome do produto (inglês), Categoria, quantidade vendida, valor total vendido (com a promoção incluída), território da venda

Com vista à otimização da execução das consultas e *procedure* propostas, se necessário reveja o MR e, defina, justificadamente, os índices e/ou *views* pertinentes.

### **SQL Profiler e Tuning Advisor**

Contraste os planos de execução das *queries* anteriores sobre a base de dados que otimizou e as *queries* equivalentes sobre a base de dados original (para a *querie* nº4 gere uma tabela adicional de teste).

Inclua no relatório o comentário ao que observa.

Compare ainda, a execução dos comandos necessários para a alteração da associação de uma categoria a um produto, em ambos os cenários: base de dados revista e original.  
Inclua no relatório o comentário ao que observa.

### 2.2.2 Controlo de Concorrência

- Defina os níveis de isolamento em que terão de ser executadas as *stored procedures*: “Adição de Produto a Encomenda”, “Alteração de Quantidade de Produto” e “Alterar Estado de Produto”, se executadas “simultaneamente” em sessões concorrentes num cenário de conflito.
- Implemente o controlo de transações necessário para garantir que quando a tabela de arquivo do fecho anual das contas de clientes está a ser escrita, não é possível aceder às vendas
- Simular a suspensão de venda de um determinado produto  
Através de um *stored procedure* cujo os argumentos de entrada são o id de um produto em promoção (a validar) e o nº de minutos e durante este período de tempo não permite o acesso (select,..) ao respetivo produto na tabela de produtos;

## 2.3 Terceira Parte

### 2.3.1 Controlo de Acessos

Considere a necessidade de implementar na bases de dados contas de acesso diferenciadas para os seguintes perfis de utilização: Clientes, Gestor de Vendas e Administrador.

Defina os objetos necessários e os privilégios de acesso aos objetos da base de dados para garantir os seguintes requisitos mínimos:

- Só os utilizadores Gestor de Vendas e Administrador, autenticados na respetiva conta do SGBD/BD poderão executar as *stored procedures* de Gestão de Produtos, Categorias e Sub-Categorias.
- Só os utilizadores Administrador, autenticados na respetiva conta do SGBD/BD poderão executar as *stored procedures* de Gestão de Utilizadores.
- Os utilizadores Clientes não podem ver as tabelas e respetiva estrutura (deverão realizar consultas necessárias via *views* e a inserção, atualização e remoção de encomendas só pode ser feita por via das respetivas *stored procedures*.

### 2.3.2 Replicação

Considerando que a sede da empresa se situa na América do Norte, defina justificadamente e implemente os mecanismos de distribuição e replicação de forma a que existam réplicas (para os dados pertinentes) distribuídas pelas regiões presentes em “SalesTerritoryGroup”, considerando:

- O catálogo de produtos e respectivas promoções é comum em todas as regiões e gerido pela sede
  - Cada região gere as suas vendas/encomendas
    - São, contudo, centralizadas na sede num servidor dedicado a *Reporting*
- ❖ No fim do período de cada promoção, todos os sites devem enviar para uma BD central (que deverá conter uma tabela agregadora) as vendas no âmbito de cada promoção.

### 2.3.3 Encriptação

Implemente o código necessário à encriptação/hashing dos campos email e password dos utilizadores do sistema.

### 2.3.4 Backups

Defina justificadamente e implemente uma estratégia de *backup* e recuperação, concretizando aspetos como:

1. Modelo de *Recuperação*
  2. Tipo de *Backup* (integral, diferencial e log) e periodicidade. Para este efeito, apresente os pressupostos quanto à carga do sistema, de forma a distribuir justificadamente o tipo de *backup* e definir a rotação dos dispositivos de armazenamento.
- ❖ Considerando os tipos, periodicidades e coberturas crie um plano de manutenção para os backups da(s) base(s) de dados.

## 3 Regras de Execução

O não seguimento de qualquer das regras elencadas nas secções seguintes pode ser motivo de anulação do projeto.

### 3.1 Grupos de Trabalho e Acompanhamento

O projeto deverá ser desenvolvido por grupos de 2 alunos, só em casos excecionais e devidamente justificados poderá ser executado individualmente, sob parecer positivo do responsável da UC (Prof. Cláudio Sapateiro). O grupo de trabalho será acompanhado (dúvidas) e avaliado pelo respetivo docente do horário/turma prevista segundo o SI.

### 3.2 Calendário e Entregas

O projeto rege-se pelo seguinte calendário:

Data	Descrição
15/09/2019	Entrega
17 e 18 /09/2019	Discussões Serão agendadas e publicadas no moodle durante o dia 16/02 pelos respetivos docentes

As entregas previstas devem ser realizadas através da plataforma *Moodle*, em *link* disponível para o efeito.

Deverá compor a entrega um ficheiro .ZIP com o seguinte conteúdo:

- Relatório
  - Documentando o projeto e justificações para as principais decisões tomadas em todas as secções de requisitos.
  - Deverá seguir a *template* disponibilizada no *moodle* para o efeito
- Scripts desenvolvidos
  - O código implementado deverá ser devidamente comentado

O nome do ficheiro ZIP segue a seguinte nomenclatura: *Docente\_numeroAluno1\_numeroAluno2.zip*

Entregas fora de prazo serão alvo de uma penalização de 1 valor por hora nas 10 horas subsequentes à entrega, findas as quais o projeto será anulado.

Todos os materiais entregues (nomeadamente relatório e scripts) serão executados no software anti-plágio da instituição.

### 3.3 Avaliação

#### **Componentes de Avaliação**

A nota mínima final é de 9,5 valores.

#### **Discussões**

Em cada discussão o grupo deverá ter tudo preparado para realizar a demonstração e responder às questões do docente, assim que for solicitado no horário que lhe será atribuído.

Durante a discussão as questões podem ser individualizadas. Consequentemente podem resultar notas diferentes para os elementos do grupo de trabalho.

### Anexos

Constituem-se anexos a este enunciado, disponíveis no moodle: 1) o *backup* da Base de Dados original e 2) o diagrama da mesma.