

Complementos de Bases de Dados 2019/2020



Licenciatura em Engenharia Informática

Projeto – fase 1

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.**

1 Descrição Geral

O grupo *Adventure*, da qual faz parte a empresa de material de ciclismo, *AdventureWorks*, está ser alvo de uma reestruturação e a administração do grupo decidiu adquirir uma nova empresa de serviços à qual deu o nome de *AdventureServices*.

Uma das medidas adotadas pela administração passa por desenvolver um novo ERP que lhe permita gerir, de forma integrada, todo o processo de vendas das duas companhias. Até ao momento uma das companhias gere toda a informação num ERP desatualizado e com o auxílio de ficheiros Excel, a segunda companhia não tem qualquer sistema informático implementado.

Com a implementação de um novo ERP torna-se necessário modelar e integrar, numa nova Base de Dados, os fragmentos de informação que foram exportados do ERP e aplicações existentes que suportavam até à data a *AdventureWorks*. Por outro lado é requisito obrigatório que o mesmo ERP faça a gestão da informação de todas as companhias existentes.

Constitui-se anexo a este enunciado um conjunto de ficheiros, que contêm os dados extraídos dos outros sistemas. 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 Fase 1

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 utilizadores de acesso à aplicação.
2. A autenticação perante a aplicação será feita com recurso à conta de email e password
 - a. Para os novos utilizadores a password do sistema deverá ser alvo de reset na primeira entrada.
 - b. Sempre que é adicionado um novo utilizador ou este solicita a recuperação de password, o sistema deverá automaticamente gerar uma password e enviar e-mail ao mesmo com essa informação (no âmbito do projeto, poderá ser efetuada uma simulação de escrita numa tabela “sentEmails” em vez de configuração de servidor de e-mail).
3. Todas as informações de vendas são divididas por companhia. A informação relativa aos clientes e produtos são comuns a todas as companhias.
4. 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)
5. Gestão de utilizadores:
 - a. Editar, Adicionar e Remover Utilizadores
 - b. Recuperar Password – Um utilizador para poder recuperar a password deve ter definidas 3 questões às quais deve responder corretamente para efetuar a recuperação. Cada utilizador terá à escolha uma lista de questões às quais deve responder, sendo obrigatório selecionar **pelo menos 3**.
6. Gestão de Produtos, Categorias e Subcategorias:
 - a. Editar, Adicionar e Remover Produtos, Categorias e Sub-Categorias
 - b. Associar Produto a Sub-Categoria/Categoria
 - c. Definir uma promoção na encomenda
 - d. Alterar as datas de Início e Fim de uma promoção (não deve ser possível atribuir uma promoção que não esteja ativa)
 - e. 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 Programação

Todos os requisitos apresentados deverão ser implementados através da solução apropriada e justificada, devendo ser criados Stored Procedures, Functions, Triggers ou outros de acordo com a solução encontrada.

A implementação dos requisitos relativos aos pontos 5 e 6 deverá ser feita com recurso a *stored procedures*. Particularmente, relativamente à implementação dos requisitos 5a e 6.a consultar a secção 2.1.5.

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

2.1.3 Verificação da nova BD

Dados originais: Para os dados originais defina uma nova base de dados “AdventureOldData” para onde deve importar toda a informação do sistema antigo.

Migração de dados: Defina os scripts necessários para a migração dos dados antigos para a nova base de dados.

Produza um conjunto de queries que dirigindo às duas bases de dados permita verificar a conformidade dos dados no novo face aos dados originalmente fornecidos.

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”

2.1.4 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.

2.1.5 Catálogo/Metadados

Stored procedures “geradores”

As *stored procedures* a definir para suportar os requisitos 5.a e 6.a da secção 2.1.1 deverão ser geradas automaticamente a partir do catálogo 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, bem como a ação de 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 os 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 ação 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 *sp*.
 - 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 *sp*.
- **A execução destas procedures deve ser feita por recurso à criação de agendamento que possibilite a execução periódica automática.**

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). Os elementos do grupo de trabalho têm de pertencer ao mesmo horário de laboratório e serão acompanhados e avaliados pelo respetivo docente desse horário.

Para o acompanhamento estarão disponíveis as aulas conforme calendário da secção seguinte e os horários de dúvidas de acordo com o publicado no *moodle*.

3.2 Calendário e Entregas

O projeto rege-se pelo seguinte calendário:

Data	Descrição
Semana 4	Aulas de laboratório de apoio ao projeto
Semana 7: 23h55, 15/11/2019	Entrega da Fase 1

	Aulas de laboratório desta semana serão de apoio ao projeto
Semana 8	Discussões da fase 1 entregue Publicação do enunciado da 2ª Fase (final) do projeto
Semana 12	Aulas de laboratório de apoio ao projeto
Semana 14: 23h55, 17/01/2020	Entrega da Fase final Aulas de laboratório desta semana serão de apoio ao projeto
Semana 15	Discussões da fase final entregue

As entregas previstas na fase 1 e final, devem ser realizadas através da plataforma *Moodle*, em *link* disponível para o efeito (descriminado por docente de laboratório).

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

- Relatório (atualizado na fase final face ao entregue na fase 1)
 - Documentando o projeto e justificações para as principais decisões tomadas em todas as secções de requisitos relativos a cada uma das fases de desenvolvimento.
 - Em ambas as fases seguirá (nas secções adequadas a cada fase) 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: TurmaLab_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

Percentagens de avaliação: $0,5 * \text{Fase 1} + 0,5 * \text{Fase 2}$

A nota mínima por fase é de 9 valores. 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 e previamente publicado no moodle .

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) ficheiro zip com os dados originais.