

# Complementos de Bases de Dados 2022/2023



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

A Wide World Importers (WWI), é uma empresa importadora e distribuidora de produtos que opera no mercado de vendas a retalho. O seu sistema de informação, que está baseado numa base de dados não normalizada e num conjunto de ficheiros em formato de texto e Excel, está desatualizado e coloca entraves ao crescimento da WWI.

A administração tomou a decisão de reformular o sistema existente, num novo ERP que lhe permita gerir, de forma integrada, todo o processo de vendas.

Com a implementação de um novo ERP torna-se necessário modelar e integrar, numa nova Base de Dados, que será denominada de WWIGlobal, os fragmentos de informação que foram exportados do sistema antigo.

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

Como foi referido a informação disponibilizada nos ficheiros extraídos do sistema antigo, encontra-se fracamente relacionada e com problemas de informação, tais como, informação repetida e campos compostos não normalizados.

Deste modo é necessário proceder a definição de um novo modelo relacional, e respetiva base de dados, que suporte toda a informação presente no sistema antigo, e cumpra as regras de normalização, neste caso a 3FN.

O novo modelo relacional a ser desenvolvido deve ter em consideração os seguintes requisitos adicionais:

1. Existência de utilizadores de acesso à aplicação, para os clientes (Customers)
  - a. A autenticação perante a aplicação será feita com recurso à conta de email e password
2. Gestão de utilizadores:
  - a. Editar, Adicionar e Remover Utilizadores
  - b. Recuperar Password – o sistema deve gerar automaticamente um token e enviar para o email do utilizador, para este alterar a password. Este token tem uma validade de 24 horas.
3. Gestão de promoções:
  - a. Definir o conceito de promoção sobre um ou mais produtos
  - b. Alterar as datas de Início e Fim de uma promoção (não deve ser possível atribuir uma promoção que não esteja ativa)

Nota: É aconselhável que estes novos requisitos sejam implementados em tabelas dedicadas.

No desenvolvimento do modelo relacional deverá considerar seguintes as opções:

- Recomendado utilizar as convenções de escrita apresentadas em:  
<https://www.tutorialsteacher.com/sqlserver/sql-server-naming-conventions>
- 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. No entanto devem ser criados obrigatoriamente os store procedures (SP), user defined functions (UDF) e triggers que implementem os seguintes processos de negócio:

- Implementação dos requisitos relativos aos pontos 2 e 3 da secção 2.1.1 (gestão de utilizadores e promoções);
- Criar uma venda;
- Adicionar um produto a uma venda;
- Alterar a quantidade de um produto numa venda;
- Remover um produto de uma venda. Recebe um parâmetro que indica se a venda é removida no caso de não ter mais produtos associados;
- Calcular o preço total de uma venda;
- Implementar a regra de negócio que verifique se a data de entrega está de acordo com o tempo previsto de entrega de um produto (“Lead Time Days”);
- Não permitir uma venda conter produtos com e sem “Chiller Stock”.

Nota: Podem ser desenvolvidos SPs, UDFs auxiliares, que facilitem a implementação dos processos de negócio enunciados.

### **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 “WWI\_OldData” 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:

- Nº de “Customers”
- Nº de “Customers” por “Category”
- Total de vendas por “Employee”
- Total monetário de vendas por “Stock Item”
- Total monetário de vendas por ano por “Stock Item”
- Total monetário de vendas por ano por “City”

Notas:

- Apenas deve ser contabilizada uma venda, mesmo que esta contenha vários produtos associados;
- Valor monetário obtido por *quantity\*Unit Price*;
- O ano deve ser retirado da coluna “Delivery Date Key”

#### 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 2.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, <nome\_tabela>\_<comando> (ex: utilizador\_insert). Opcionalmente no seu código devem ser verificadas todas as restrições (*not null*, *primary key*, ...) sobre os dados que estão definidas na respetiva tabela.

### **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 base de dados: todos os campos de todas as tabelas, com os seus tipos de dados, tamanho respetivo e restrições associadas. 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. João Portelinha). Os elementos do grupo de trabalho têm de ter o mesmo Prof. de laboratório e serão acompanhados e avaliados pelo respetivo docente.

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:

Datas	Descrição
Semana 7 (14/11 a 18/11)	Entrega da Fase 1, até às 23H55 de 18/11
Semana 8 e 9	Discussões da fase 1 entregue Publicação do enunciado da 2ª Fase (final) do projeto
Semana 15 (16/01 a 20/01)	Entrega da Fase 2, até às 23h55 de 20/01
Semana 16	Discussões da fase 2

As entregas previstas na fase 1 e 2, devem ser realizadas através da plataforma *Moodle*, em *link* disponível para o efeito (descriminado por turno de laboratório).

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

- Relatório (atualizado na fase 2 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 7 valores. A nota mínima final é de 9,5 valores.

### **Grelha de avaliação**

<b>Componente</b>	<b>Valor</b>
Modelo	15%
Programação e Tratamento de Erros	20%
Layout + Migração de Dados	30%
Metadados	20%
Relatório e Apresentação	15%
<b>Total</b>	<b>100%</b>

### **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.

Fim do enunciado