

# Complementos de Bases de Dados 2019/2020

Licenciatura em Eng<sup>a</sup>. Informática

1ª Fase Relatório Técnico

Turma:02

Horário de Laboratório: 4:30

Docente:João Lazaro

Grupo nº 9

nº110221031 Nome Eduardo Ferreira

nº160221075 Nome André Nascimento

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

### 1. Sumário Executivo

Este projeto foi proposto no âmbito da unidade curricular Complementos de Base de Dados(CBD), com o objetivo de normalizar uma base de dados e migrar os dados antiga para a nova.

Na realização do projeto, foram aplicados vários conceitos que foram lecionados ao longo UC, sendo estes:

- Metadados – Que iram garantir a automatização da criação de procedimentos
- Filegroups – Estes separam as tabelas em vários ficheiros, facilitando a sua gestão e performance no seu acesso.
- Backups – Permitirá recuperar a base de dados caso algum erro tenha ocorrido.
- Índices – Permitirá melhorar a performance das queries mais essenciais.
- Transações - Permitirá corrigir/prevenir alguns dos erros phantom read/ repeatable read e dirty read.
- Permissões – Irá permitir simular uma melhor estrutura para a base de dados, criando 3 tipos de users com as suas próprias permissões para com o que podem fazer na base de dados.

### 2. Especificação de Requisitos

#### 2.1 Requisitos funcionais

ID	Descrição	Implementado (S/N)
RF01	O sistema deverá permitir a criação de novos utilizadores com passwords aleatórias geradas automaticamente	S
RF02	O sistema deverá permitir a criação de um Produto	S
RF03	O sistema deverá permitir a criação de uma encomenda	S
RF04	O sistema deverá permitir ao utilizador apagar produtos	S
RF05	O sistema deverá permitir ao utilizador apagar um utilizador	S
RF06	O sistema deverá permitir ao utilizador adicionar um produto à encomenda	S
RF07	O sistema deverá permitir ao utilizador alterar a quantidade de produto na encomenda	S

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

RF08	O sistema deverá permitir ao utilizador remover uma encomenda	S
RF09	O sistema deverá permitir ao utilizador fazer login	S
RF10	O sistema deverá permitir ao utilizador fazer logout	S
RF11	O sistema deverá permitir ao utilizador fazer reset à sua password	S
RF12	O sistema deverá permitir ao utilizador receber perguntas de segurança para poder fazer reset à sua password	S
RF13	O sistema deverá permitir ao utilizador alterar o seu email	S
RF14	O sistema deverá permitir ao utilizador associar um produto a uma categoria e subcategoria	S
RF15	O sistema deverá permitir ao utilizador editar uma subcategoria	S
RF16	O sistema deverá permitir ao utilizador editar uma categoria	S
RF17	O sistema deverá permitir ao utilizador alterar o estado do produto	S
RF18	O sistema deverá permitir ao utilizador editar, criar ou remover uma promoção	S
RF19	O sistema deverá permitir ao utilizador visualizar o total de vendas por ano	S
RF20	O sistema deverá permitir ao utilizador visualizar o total de vendas por ano e país	S
RF21	O sistema deverá permitir ao utilizador visualizar o total de vendas por ano e subcategoria	S
RF22	O sistema deverá permitir ao utilizador visualizar o total de vendas por ano e categoria	S
	O sistema deverá permitir ao utilizador visualizar o número de clientes por	S

*1ª Fase Relatório Técnico – Complementos de Bases de Dados*

RF23	ano e país de território de vendas	
RF24	O sistema deverá poder restaurar a base de dados caso ele seja perdida/danificada.	S
RF25	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular o volume de vendas total por produto.	S
RF26	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular o volume de vendas total por produto com promoção.	S
RF27	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular a percentagem de vendas por produto efetuada com promoção.	S
RF28	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular o total de vendas anual por Região Geográfica	S
RF29	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular para cada ano a Região Geográfica com o maior valor total de vendas	S
RF30	O sistema deverá possuir um índice para melhorar o performance da query com o objetivo de calcular o prazo médio entre data de encomenda e envio por Região Geográfica dos últimos dois anos	S
RF31	O Sistema deverá possuir o tipo e número de transações adequadas para a realização do stored procedure dedicado a alterar o status do produto.	S
RF32	O Sistema deverá possuir o tipo e número de transações adequadas para a realização do stored procedure dedicado a alterar a quantidade de produto numa encomenda.	S
RF33	O Sistema deverá possuir o tipo e número de transações adequadas para a realização do stored procedure dedicado a adicionar um produto a uma encomenda.	S
RF34	O Sistema deverá possuir um utilizador do tipo administrador com todas as	S

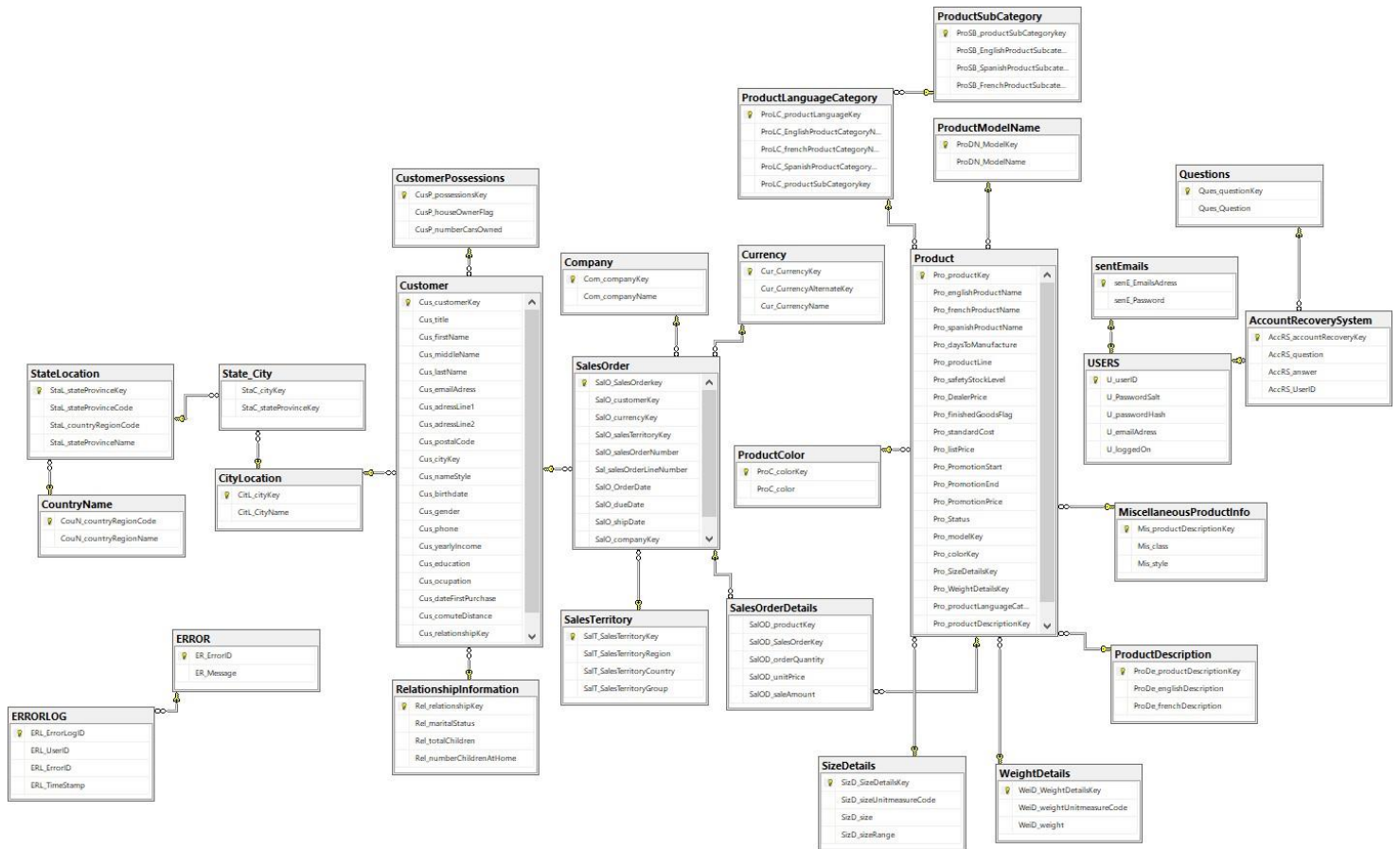
## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

	suas permissões necessárias.	
RF35	O Sistema deverá possuir um utilizador do tipo registado com todas as suas permissões necessárias.	S
RF36	O Sistema deverá possuir um utilizador do tipo Não-registado com todas as suas permissões necessárias.	S
RF37	O sistema deverá ter backups do tipo full assim como o seu Schedule (a que horas/dias o tipo de backup é realizado)	S
RF38	O sistema deverá ter backups do tipo diferencial assim como o seu Schedule (a que horas/dias o tipo de backup é realizado)	S
RF39	O sistema deverá ter backups do tipo log assim como o seu Schedule (a que horas/dias o tipo de backup é realizado)	S
Rf40	O sistema deverá conseguir recuperar os dados caso a base de dados seja apagada ou danificada.	S

### 2.2 Atores

Nome	Descrição
USERS	São funcionários da companhia AdventureWorks
Costumers	São clientes da loja AdventureWorks
Permission Users	São os users de teste para as permissões de uma forma generalizada .

### 3. Modelo Relacional (Modelo de dados)



## 4. Dimensionamento do Layout

Em relação ao dimensionamento do base de dados, criaram-se ps seguintes filegroups:

Primary:

- Este Filegroup tem um tamanho inicial de 22mb(o tamanho foi obtido através do comando `exex sp_space_used`).
- Tem um max size de 48mb, esse número foi obtido através de um crescimento(filegrowth) de 30 % ano a ano durante 3 anos.
- O ficheiro Projeto\_Log tem exatamente o mesmo tamanho que o Projeto\_dat

FG\_Customers

- Este filegroup tem um tamanho inicial de 9264Kb(este valor foi obtido através da soma do tamanho das tabelas CountryName, StateLocation, State\_City, CityLocation, CustomerPossessions, RelationshipInformation e por fim Customer)
- Tem um max size de 16009Kb, esse número foi obtido através de um crescimento(filegrowth) de 20 % ano a ano durante 3 anos.

FG\_Products:

- Este filegroup tem um tamanho inicial de 1336Kb(este valor foi obtido através da soma do tamanho das tabelas ProductColor, ProductSubCategory, ProductLanguageCategory, ProductDescription, Product, ProductModelName, WeightDetails e a tabela SizeDetails)
- Tem um max size de 2309Kb, esse número foi obtido através de um crescimento(filegrowth) de 20 % ano a ano durante 3 anos.

FG\_Sales:

- Este filegroup tem um tamanho inicial de 7256Kb(este valor foi obtido através da soma do tamanho das tabelas Currency, SalesTerritory, Company, SalesOrder e a tabelas SalesOrderDetails)
- Tem um max size de 12539Kb, esse número foi obtido através de um crescimento(filegrowth) de 20 % ano a ano durante 3 anos

FG\_AccountManagement:

- Este filegroup tem um tamanho inicial de 856Kb(este valor foi obtido através da soma do tamanho das tabelas Questions, AccountRecoverySystem, SentEmails, USERS, ERROR e a tabela ERRORLOG)
- Tem um max size de 1480Kb, esse número foi obtido através de um crescimento(filegrowth) de 20 % ano a ano durante 3 anos

## 5. Schemas

Nome	Descrição
dbo	Aceder a todas as tarefas permitidas pela base de dados

## 6. Functions

Nome	Tipo	Atributos	Descrição
fnCodifica	HASHBYTES	@wordToHash varchar(50)	Codifica a password e o email do utilizador.

## 7. Stored procedures

Nome	Atributos	Descrição
sp_SalesPerYear2	Não tem	Permite vizualizar o total de vendas por ano
Sp_SalesPerYearAndCountry2	Não tem	Permite vizualizar o total de vendas por ano e pais
Sp_SalesPerYearAndSubcategory	Não tem	Permite vizualizar o total de vendas por ano e subcategoria
Sp_SalesPerYearAndCategory	Não tem	Permite vizualizar o total de vendas por ano e categoria
Sp_CustomerYearAndCountry	Não tem	Permite vizualizar o número de clientes por ano e pais de territorio de vendas
Sp_CreatingProduct	@englishProductName me varchar(50),	Permite criar um producto



## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

	@Color nvarchar(20), @EnglishProductC at egoryName nvarchar(50), @englishProductS ub categoryName nvarchar(50), @userID int	
Sp_DeleteProduct	@ProductKey int, @userID int	Permite apagar um producto
Sp_associateProduct 2	@ProductKey int, @EnglishProductC at egoryID int, @EnglishProductS ub categoryID int, @userID int	Permite associar um producto a uma categoria e subcategoria
Sp_EditSubCat	@EnglishProductS ub categoryID int, @englishSubName nvarchar(50), @SpanishSubNam e nvarchar(50), @FrenchSubName nvarchar(50), @userID int	Permite editar uma subcategoria
Sp_EditCat	@EnglishProductc at egoryID nvarchar(50), @englishName nvarchar(50), @SpanishName nvarchar(50), @FrenchName nvarchar(50), @subCategoryKey int, @userID int	Permite editar uma categoria
Sp_AlterStat	@ProductKey int, @newStatus nvarchar(20), @userID	Permite alterar o estado do produto

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

Sp_EditPromo	@ProductKey int, @promotionStart date, @promotionEnd date, @Pro_promotionPrice money, @userID int	Permite editar, criar ou remover uma promoção
Sp_CreateDelivery	@CustomerKey int, @CurrencyKey int, @salesTerritoryKey int, @CompanyKey int, @salesOrderNumber nvarchar(10) , @OrderDate datetime2, @dueDate datetime2, @shipDate datetime2, @userID int	Permite criar uma encomenda
Sp_AddProduct	@productKey int, @orderQuantity int, @SalesOrderKey int, @userID int	Permite adicionar um produto à encomenda
Sp_AlterQuantity	@SalesOrderKey int, @orderQuantity int, @userID int	Permite alterar a quantidade de produto na encomenda
Sp_RemoveDelivery	@SalesOrderKey int, @userID int	Permite remover uma encomenda
Sp_NewUser	@emailAddress nvarchar(50)	Permite criar um novo utilizador com uma password gerada aleatoriamente
Sp_LoginUser	@email nvarchar(50),	Permite que o utilizador faça o login

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

	@passwordWithSalt nvarchar(50)	
Sp_LogoutUser	@userID int	Permite que o utilizador faça logout
Sp_getSecurityQuestion	@questionNum int, @answer nvarchar(20), @userID int	Permite ao utilizador receber questões de segurança para que possa fazer reset à sua password
Sp_resetPass	@userID int	Permite ao utilizador fazer reset à password
Sp_DeleteUser	@userID int	Permite apagar um utilizador
Sp_editUser	@userID int, @OLDemail nvarchar(50), @NEWemail nvarchar(50)	Permite alterar o email do utilizador
Sp_InsertProcedure	@SchemaName nvarchar(40), @TableName nvarchar(40)	Permite criar procedures
Sp_createDeleteProcedure	@SchemaName nvarchar(40), @TableName nvarchar(40)	Permite apagar procedures
Sp_createUpdateProcedure	@SchemaName nvarchar(40), @TableName nvarchar(40)	Permite atualizar procedures
sp_changeFirstDayToPromo	Não tem	Permite tornar todas as vendas do dia 1 de janeiro como promoção (como pedia no guia)

## 8. Consultas

### 8.1 Verificação da conformidade dos dados

Total de vendas por ano:

Results Messages		
	PROFIT	ANO
1	3266373,65...	2014
2	6530343,52...	2015
3	9791060,29...	2016
4	9770899,74	2017

	(No column name)	ANO
1	3266373,6566	2014
2	6530343,5264	2015
3	9791060,2977	2016
4	9770899,74	2017

Total de vendas por ano e país:

Results Messages			
	PROFIT	Country	ANO
1	1309047,19...	Australia	2014
2	2154284,88...	Australia	2015
3	3033784,21...	Australia	2016
4	2563884,29	Australia	2017
5	146829,8074	Canada	2014
6	621602,3823	Canada	2015
7	535784,4624	Canada	2016
8	673628,21	Canada	2017
9	180571,692	France	2014
10	514043,0131	France	2015

	(No column name)	Sales TerritoryCountry	(No column name)
1	1309047,1978	Australia	2014
2	2154284,8835	Australia	2015
3	3033784,2131	Australia	2016
4	2563884,29	Australia	2017
5	146829,8074	Canada	2014
6	621602,3823	Canada	2015
7	535784,4624	Canada	2016
8	673628,21	Canada	2017
9	180571,692	France	2014
10	514043,0131	France	2015

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

Total de vendas por ano e subcategoria:

Results Messages			
	PROFIT	ProSB_EnglishProductSubcategoryName	ANO
1	16440,00	Bike Racks	2016
2	22920,00	Bike Racks	2017
3	20670,00	Bike Stands	2017
4	18921,00	Bike Stands	2016
5	23280,27	Bottles and Cages	2016
6	33517,92	Bottles and Cages	2017
7	7956,15	Caps	2016
8	11731,95	Caps	2017
9	3044,85	Cleaners	2016
10	4173,75	Cleaners	2017

	(No column name)	EnglishProductSubcategoryName	(No column name)
1	16440,00	Bike Racks	2016
2	22920,00	Bike Racks	2017
3	18921,00	Bike Stands	2016
4	20670,00	Bike Stands	2017
5	23280,27	Bottles and Cages	2016
6	33517,92	Bottles and Cages	2017
7	7956,15	Caps	2016
8	11731,95	Caps	2017
9	3044,85	Cleaners	2016
10	4173,75	Cleaners	2017

Total de vendas por ano e categoria:

	PROFIT	ProLC_EnglishProductCategoryName	ANO
1	293709,71	Accessories	2016
2	407050,25	Accessories	2017
3	3266373,65...	Bikes	2014
4	6530343,52...	Bikes	2015
5	9359102,61...	Bikes	2016
6	9162324,85	Bikes	2017
7	138247,97	Clothing	2016
8	201524,64	Clothing	2017

	(No column name)	EnglishProductCategoryName	(No column name)
1	293709,71	Accessories	2016
2	407050,25	Accessories	2017
3	3266373,6566	Bikes	2014
4	6530343,5264	Bikes	2015
5	9359102,6177	Bikes	2016
6	9162324,85	Bikes	2017
7	138247,97	Clothing	2016
8	201524,64	Clothing	2017

## 1ª Fase Relatório Técnico – Complementos de Bases de Dados

Número de clientes por ano e país de território de vendas:

	Customers	Country	ANO
1	394	Australia	2014
2	859	Australia	2015
3	5335	Australia	2016
4	6757	Australia	2017
5	47	Canada	2014
6	226	Canada	2015
7	3086	Canada	2016
8	4261	Canada	2017
9	59	France	2014
10	222	France	2015

	Customers	Sales TerritoryCountry	(No column name)
1	394	Australia	2014
2	859	Australia	2015
3	5335	Australia	2016
4	6757	Australia	2017
5	47	Canada	2014
6	226	Canada	2015
7	3086	Canada	2016
8	4261	Canada	2017
9	59	France	2014
10	222	France	2015

### 8.2 Outras consultas

- Traces do SQL Profiler, que contêm a execução dos códigos com e sem índice
- Demonstração vídeo da base de dados a ser recuperada perante alguns erros.
- Vídeo de execução de transações em funcionamento através do wait time como forma de teste.
- Video de login como teste para com as permissões.
- Video de demonstração de backups em ação após a database ter sido apagada.
- Imagens de demonstração de como a replicação é criada.

## 9. Índices

Designação	Tabela	Justificação/Consultas
IX_ProductKeyWithEnglishName	Product	O índice de múltiplos valores, encapsulando a Pro_productKey assim como a Pro_englishProductName, ter estes dois organizados num índice foi considerado pelo grupo como essencial para a query dedicada a encontrar a organização de produtos por nome.
IX_SalesOrderDetails_Promo_productKey	SalesOrderDetails	O índice de múltiplos valores, encapsulando a SalOD_productKey assim como a SalOD_Promo, ter estes dois organizados num índice foi considerado pelo grupo como essencial para a query dedicada a encontrar produtos em promoção.
IX_SalesOrderDetails_Promo	SalesOrderDetails	O índice de uma só coluna, foi considerado pelo grupo como uma pequena necessidade para a query visto que já possuíamos um índice semelhante dedicado a encontrar produtos em promoção.
IX_SalesOrder_SalesOrderkey_salesTerritoryKey	SalesOrder	O índice de múltiplos valores, encapsulando a SalO_SalesOrderkey assim como a SalO_salesTerritoryKey, ter estes dois organizados num índice foi considerado pelo grupo como essencial para a query dedicada a encontrar as vendas por região geográfica e data.
IX_SalesOrder_salesTerritoryKey_OrderDate	SalesOrder	O índice de múltiplos valores, encapsulando a SalO_salesTerritoryKey assim como a SalO_OrderDate, ter estes dois organizados num índice foi considerado pelo grupo como essencial para a query dedicada a encontrar as vendas por região geográfica e data.

## 10. Otimização e Execução de Consultas

O grupo começou por utilizar o sql profiler para fazer trace das queries criadas de forma a poder verificar o numero de “Reads” efectuados pela query assim como a sua duração. Após a sua verificação o grupo utilizou o tuning advisor para cada uma das queries de forma a possuir-mos uma ideia de quais os índices a serem feitos assim como qual seria o aumento de performance. Embora isto alguns dos índices aconselhados pelo tuning advisor não foram aplicados pois o grupo considerou-os como demais ou não necessários. Esses índices seriam chamados de IX\_SalesOrderDetails\_ProductKey onde iria existir um índice no atributo ProductKey na tabela SalesOrderDetails mas como o índice IX\_ProductKeyWithEngName já existe e utiliza tanto o ProductKey como o englishProductName de forma a ter uma query mais organizada foi considerado então desnecessário. O índice

IX\_SalesOrderDetails\_SalesOrderKey foi considerado desnecessário pelo mesmo motivo que o anterior, já existia um índice (IX\_SalesOrder\_SalesOrderkey\_salesTerritoryKey) que cumpria essa função mas de uma melhor/mais ordenada forma e ter ambos ao mesmo tempo resultou numa perda de performance. Após os índices terem sido criados o grupo utilizou o sql profiler mais uma vez para verificar o aumento de performance.

No volume de vendas em promoção por produto houve um aumento de performance significativo, onde o numero de reads foi reduzido a quase metade (632 para 345) na primeira leitura e a duração foi extremamente reduzida também (39906 para 8171)

Na percentagem de vendas por produto efetuada com promoção o numero de reads foi reduzido (623 para 484) embora a redução de reads a sua duração aumentou ligeiramente (14859 para 21747 nós pensamos que isto tenha ocorrido devido ao facto das durações serem demasiado pequenas/pouco significativas).

Nas total de vendas anual por Região Geográfica o numero de reads foi reduzido significativamente (2267 para 568) assim como a sua duração (138136 para 74561).

Em para cada ano a Região Geográfica com o maior valor total de vendas Geográfica o numero de reads foi reduzido significativamente (954 para 568), embora a redução de reads a sua duração foi praticamente a mesma.

No Prazo médio entre data de encomenda e envio por Região Geográfica não existiu nenhum aumento de performance tanto nos reads como na duração.



## 11. Controlo de Concorrência

No Procedure **sp\_alterStat** o grupo optou por utilizar o nível de isolamento repeatable read de forma a evitar os problemas de dirty read e nonrepeatable read, Desta forma podemos evitar que o status do produto seja lido com informação errada (dirty read) ou o problema da segunda transação alterar o status do produto sem que a primeira tenha sido lida o que pode causar problemas.

No Procedure **sp\_AddProduct** o grupo optou por utilizar o nível read committed para a verificação do produto que vai ser adicionado na encomenda de forma a evitar o dirty read, de seguida optamos também por utilizar o serializable no final do procedure onde o produto vai ser adicionado á encomenda, embora o serialize seja o mais “pesado” dos tipos de isolamento e deva ser evitado, nós achamos necessário devido ao facto de que se duas **sp\_AddProduct** fossem ser executadas concorrentemente, o resultado seria catastrófico. Caso alguém tentá-se inserir uma nova row/encomenda enquanto outra estará a ser inserida poderia acontecer ambas terem o mesmo id de encomenda/uma falhava completamente.

No Procedure **sp\_alterQuantity** o grupo optou por utilizar o nível de isolamento read committed durante a verificação da salesOrderKey de forma a poder obter de uma forma mais “correta” a leitura do produto da encomenda a ser alterado, para além disso utilizamos também o nível de isolamento repeatable read no final do procedure (onde a alteração decorre) de forma a que a leitura do numero de produtos seja “melhor” (isto pode ser observado no vídeo de demonstração localizado na pasta “Video Demonstration of Transactions at work” de uma certa forma...).

## 12. Segurança e Controlo de Acessos

Para os controlos de acesso foram criados quatro roles (AdminRole, MarketingManager, AnonUser, RegisteredUser) que estavam especificados no guia do projeto, o grupo criou também quatro logins (admingLog, marketingLog, anonLog, registeredLog) cada um com a sua própria password, de seguida criamos também quatro utilizadores (Administrator, Manager, Kaguya, Mokou) que foram associados aos quatro logins previamente criados de forma a poder similar o login de um utilizador no sql server. Após termos os utilizadores operacionais demos a cada um o seu próprio role assim como as suas permissões adequadas (grants) para que se possa testar as funcionalidades de cada user **(estamos cientes de que o Manager só deveria fazer Select á tabela de promoção mas como a nossa base de dados tem as promoções como atributos do Products onde verifica se as datas são validas e se o preço é inferior ao normal este é a única maneira...)**. Foi também criado o código de simulação de login de um utilizador assim como a utilização de selects juntamente com a função fn\_my\_permissions para que seja possível verificar as permissões de cada utilizador.

Em relação á segurança o grupo optou por utilizar o código previamente criado na fase 1 e melhorar a partir de aí, ou seja escolhemos usar o hashing para ambos os campos, não só pelo facto que certa parte já se encontrava feita/criada mas também por questões de performance e utilidade. O grupo decidiu que seria desnecessário a criação de uma Master Key -> Criação de um Certificate -> criação

de uma chave simétrica ou assimétrica e open/close sempre que fôssemos encriptar ou desencriptar pois estes dados não são processados de uma mas simplesmente utilizados quando um utilizador efetua login, logo embora o hashing faz com que não seja possível esses dados serem desencriptados como os valores da encriptação serão sempre únicos com hashing e todos os utilizadores recebem via email a pass, os utilizadores de sistema poderão sempre efectuar o seu login.

### 13. Esquema de Replicação

Ao efectuar a replicação, foi decidido entre os membros do grupo, usar o método de replicação transaccional. Pois quando acontecem alterações à database principal, as mesmas alterações vão acontecer aos seus subscritores com uma maior brevidade que o método Snapshot.

Para efectuar uma replicação, é preciso seleccionar a pasta “Replication”, de seguida seleccionar, com o botão direito do rato, a pasta “local Publications” e depois seleccionar a opção “New Publication”.

Em seguida, clicar no botão “next”, de seguida é preciso escolher a base de dados que se deseja replicar. Depois de escolher a base de dados que irá ser replicada, é necessário escolher o tipo de replicação, após seleccionar o tipo de replicação, clique “next”, de seguida é preciso escolher as tabelas que vão ser replicadas, clique no “+”, em seguida escolha-se as tabelas e clique “next”. Em seguida pode-se adicionar filtros às linhas das tabelas (neste caso o grupo optou por não adicionar filtros) de seguida clique “next”. Em seguida, clique na primeira opção e clique next. Em seguida é escolhido o “Agent”, clique na botão “Security Settings” e em seguida clique na opção “Run under the SQL Server Agent service account”, abaixo deixe ficar como está e em seguida clique “OK”, após ter feito isso, clique no botão “next”.

De seguida deixe ficar seleccionada a opção “Create the publication” e clique no botão “next” (neste caso o grupo optou também por gerar um script para poder criar a publicação passo a passo), em seguida clique no botão “next”. De seguida deixe ficar como está e clique “next”. Neste passo vai ter de dar um nome à publicação, após ter dado um nome à publicação, clique no botão “finish”. Espere alguns segundos e a publicação será efectuada.

Para efectuar uma publicação, clique com o botão direito do rato em “Local Subscription”, de seguida seleccione a opção “New Subscriptions” e em seguida clique no botão “next”, a seguir é preciso escolher a publicação que se vai subscrever, seleccione a publicação que foi feita anteriormente e clique no botão “next”. De seguida deixe ficar como está e clique “next”. De seguida é necessário o subscritor (Subscriber) e a base de dados que vai subscrever, após ter sido seleccionada a o subscritor e a base de dados subscritora, clique no botão “next”.

Em seguida é preciso escolher a conexão com o distribuidor e com o subscriber, clicando no botão “...”, onde é seleccionada a opção “Run under the SQL Agent service account”, o resto pode ficar como está, de seguida clique “OK”. Após ter escolhido as conexões, clique no botão “next”. De seguida, deixa-se estar escolhida a opção “Run continuously” como está e clique na opção “next”, em seguida, clique em “next” deixando escolhida a opção “Immediately”. De seguida, clique nas duas opções que são apresentadas e clique no botão “next”. De seguida, pode-se deixar como está e clique “next”. De seguida clique no botão “Finish” e espere alguns segundos para que a publicação seja feita.

## 14. Política de Backups

Na política de backups nós optamos pela opção de utilizar “Backup Devices”, um plano lógico que pode “armazenar” todos os .bak/logs necessários para o restore de uma base de dados danificada/ apagada, criamos também um Sistema/Schedule de backups de se encontra descrito no documento Word “10 - Backup plan”, explicando tudo numa forma mais sucinta.

Estamos cientes que os backups vão ser criados no disco C: , é preferível criar os backups noutra unidade onde a DB não existe, isto é simplesmente para ser melhor verificado.

Como extra criamos também o código para efectuar a tail log backup e o restore que possuía os 4 juntos.

## 15. Descrição da Demonstração

### (Fase 1)

Neste Projeto o grupo decidiu separar ao máximo todas as funcionalidades em vários ficheiros SQL assim como atribuir a cada ficheiro um número no seu nome, criando assim uma ordem em que cada deve ser corrido de forma a simplificar a criação da Database (from scratch), para além disso criamos também um ficheiro .bak para que a database possa ser importada sem quaisquer problemas.

Primeiramente no ficheiro “1 – Filegroups.sql” onde a layout do projeto se encontra, o grupo optou por escolher o directório “C:\Projeto\_de\_CBD\MSSQL\DATA\”, esta localização foi escolhida devido a ser mais fácil de localizar e de trabalhar. Todos os ficheiros irão ser armazenados nesse directório pelos diversos Filegroups. Caso opte por não utilizar o ficheiro.bak este ficheiro pode ser todo selecionado e corrido de uma só vez.

O ficheiro “2 - Creates.sql” contém simplesmente todos os creates tables que são utilizados na data base, é um ficheiro que pode todo ser selecionado e corrido de uma só vez.

O ficheiro “3 – DataMigration.sql” contém a migração dos dados da AdventureOldData.bak para a nova database denominada de PROJETO que é uma database mais normalizada que a anterior. Este ficheiro pode também ser corrido todo de uma só vez caso opte por não utilizar o ficheiro.bak.

O ficheiro “4 - Error Table and SecuritySystemQuestions.sql” contém simplesmente uns pequenos inserts para as tables Questions e ERROR, estes inserts basicamente injetam nessas tabelas as questões possíveis de segurança que a database vai ter assim como os possíveis erros que irão/podem ser encontrados caso algum procedure sejam mal executado.

O ficheiro “5 - Foreign Keys.sql” simplesmente contém todas as foreign key constraints para que se possa criar o modelo relacional do projeto assim como identificar as foreign keys que cada table possui.

O ficheiro “6 - Minimum Requirements.sql” Contém todos os requisitos mínimos que eram

pedidos no guia assim como as queries para a AdventureOldData para que possam ser comparadas e comprovar que os resultados são idênticos antes e pós migração.

O ficheiro “7 - USERS and Password Encryption Related Queries.sql” Contêm todas as queries relacionadas com o user assim como a password. A function “fnCodificaPassword” serve para codificar a password para que não seja tão fácil de detectar, é também uma função que é utilizada pelos procedures relacionados com passwords. O procedure sp\_NewUser permite que um novo utilizador seja criado no Sistema, no final desta procedure existe 4 comandos exec os primeiros 2 irão funcionar sem erros enquanto que os últimos irão demonstrar alguns dos erros possíveis de detectar pelo nosso ERROR/ERRORLOG tables.

Após ter criado um utilizador ser-lhe á enviado a password decriptada via “email” (table sentEmails), como as password são geradas aleatoriamente deverá utilizar a query de sentEmails que se encontra em comentário e fazer copy-paste da password para o utilizador com email “**MokouisLove@gmail.com**”. A próxima procedure é o sp\_LoginUser que necessita de um email e uma password (como a password é feita aleatoriamente terá de copiar a pass criada anteriormente e substituir no comando exec que se encontra no final desta SP). Se o login foi feito com sucesso então a coluna LoggedOn irá ter o valor bit de 1

A Procedure sp\_LogoutUser simplesmente precisa do um userID, a procedure verifica se esse user está online e caso esteja fará logout.

A procedure sp\_getSecurityQuestion basicamente irá associar as questões de segurança a um user e utiliza como parâmetros a @questionNum (id da questão), @answer (resposta), @userId(User a ser associado), existem 3 exec no final da SP que podem ser corridos.

A procedure sp\_resetPass irá criar uma nova password para um utilizador já existente, recebe simplesmente como parâmetro o userID e verifica se esse user possui pelo menos 3 questões respondidas. Após o reset ter sido feito pode tentar efectuar o login novamente “seguindo exatamente os passos que foram seguidos anteriormente.

O sp\_DeleteUser Procedure elimina um user do sistema (aconselho a correr este procedure no final).

O procedure sp\_EditUser permite que o user do Sistema possa modificar o seu email (existe um exec que pode ser corrido para testar).

O ficheiro “8 - Product related Queries” contém todas as procedures relacionadas com os produtos e encomendas. Todos estes procedures recebem um UserID e verifica se esse user se encontra online, caso não esteja então o ERRORLOG devolverá uma mensagem.

Sp\_CreatingProduct irá criar um produto com os parâmetros que recebe e caso algum esteja errado a verificação de erros deverá detetar. (existe dois execs no final deste SP para testar.)

Sp\_DeleteProduct irá remover um produto da nossa base de dados, o procedure verifica se as sua category e subCategory são únicas e caso sejam então irá elimina-las também, caso exista algum outro produto a utilizar a mesma category/subcategory associação então só o produto será apagado.

Sp\_AssociateProduct2 irá associar um produto com uma category e subcategory (essa ligação category/subcategory tem de existir caso contrário terá ainda de ser criada primeiro), existem dois execs no final para testar.

O sp\_EditSubCat e sp\_EditCat permitem a edição de uma subcategory e category respectivamente (cada um terá dois execs no final para testar, o segundo de cada deverá dar erro devido ao facto de não existir um ProductcategoryID = 90 / EnglishProductSubcategoryID = 90).

O sp\_alterStat irá alterar um estado do produto para o valor de null or current, caso seja current então esse produto poderá ser posta na encomenda.

O sp\_EditPromo permite criar ou editar uma promoção, a promoção foi feita como atributos na tabela Product onde existe as colunas (Pro\_PromotionStart, Pro\_PromotionEnd, Pro\_PromotionPrice), que indicam o início de uma promoção, o seu fim assim como o seu preço, existem verificações de erros caso o Pro\_PromotionStart seja maior que Pro\_PromotionEnd e caso o Pro\_PromotionPrice seja maior que o preço base (listPrice) de um produto, isto pode ser verificado com os 4 execs no final desta Procedure.

O sp\_CreateDelivery permite criar encomendas, este procedure irá receber um grande número de parâmetros e fazer todas as suas verificações para garantir que a encomenda possa ser feita sem quaisquer problemas (existe um exec no final como exemplo assim como um select query para verificar a sua criação). No final, esta procedure irá devolver uma salesOrderkey de deverá ser utilizada para a adição de um produto á encomenda no próximo procedure.

O sp\_AddProduct procedure irá adicionar um produto a uma encomenda, (deverá ser utilizado um salesOrderKey existente de forma a poder adicionar o produto á encomenda) assim como os outros parâmetros necessários (já existe um exec no final como exemplo).

O sp\_AlterQuantity procedure irá permitir que a quantidade de um produto já existente seja alterados caso seja possível (se o produto já estiver sido enviado então não será possível fazer alteração), existe um exec no final como exemplo.

O sp\_RemoveDelivery irá remover um produto de uma encomenda caso essa encomenda exista.

O ficheiro “9 – Metadados” contém simplesmente os 3 geradores de procedures que eram pedidos no guia (sem a verificação de erros) para os correr basicamente correr os seus execs que se encontram no final de cada SP.

## **(Fase 2)**

Nesta fase o grupo decidiu separar ao máximo todas as funcionalidades em vários ficheiros SQL assim como atribuir a cada ficheiro um número no seu nome, assim como foi feito na fase. Para poder correr o .bak o utilizador deverá clicar com o lado direito nas pastas databases do sql server, ir a restore database e depois escolher a opção device e de seguida ir para o directorio de backups e seleccionar o backupDevice que se encontra na pasta de backups.

Após a restauro ter sido feito, o utilizador pode abrir o ficheiro sql nº 9 - Metadados(new) para verificar as correções/ a melhoria nos geradores. Em principio os stored procedures já devem ter sido criados assim que o restauro aconteceu por isso deverá bastar executar o “**USE PROJETO**” da linha um e de seguida utilizar os execs que se encontram respectivamente nas linhas 142 / 214 e 416, desta vez os geradores deverão mostrar as verificações de erros.

Após esta melhoria ter sido revisitada o utilizador poderá o ficheiro numero 10 - Backups seleccione Use Projeto que se encontra na primeira linha e de seguida crie as pastas dentro do disco C de forma a que esta directoria exista **C:\Projeto\_de\_CBD\MSSQL\BACKUPS\** após ter sido criado esta localização corra ambos os **EXEC sp\_addumpdevices** de forma a criar dois devices lógicos, agora execute o código da linha 15 e 16 de forma a criar um backup full de toda a informação da database. A partir deste momento caso algum erro danifica a base de dados o utilizador pode simplesmente correr o código das linhas 39 e 40 para executar um restore só com o full( ver vídeo na pasta “Video Demonstration of backup at work” para melhor perceber). Após a criação de um backup full, corra agora a linha nº18 para poder criar um user na tabela de users com o email de KaguyaSucks@gmail.com , assim que este utilizador for criado com sucesso, corra agora o código das linhas 21 e 22 para criar um backup diferencial que irá guardar este user que acabou de ser criado e que o full não possui, agora que já temos um diferencial poderemos correr o código das linhas 43 a 45 de forma a fazer um restore que utilize o full + o diferencial. Após a criação de um backup full e diferencial, corra agora a linha nº24 para poder criar um user na tabela de users com o email de GetCaved@gmail.com, assim que este utilizador for criado com sucesso, corra agora o código das linhas 26 e 27 para criar um backup transacional que irá guardar este user que acabou de ser criado e que o full e o diferencial não possuem, agora caso a database esteja danificada ou seja apagada o utilizador pode simplesmente correr o código da linha 48 a 51.

No ficheiro 11 - Indices and Queries irá ser encontrado um conjunto de queries assim como índices e um store procedure denominado sp\_changeFirstDayToPromo que serve para mudar todos os produtos vendidos nos dias 1 de janeiro como se tivessem tido promoção de forma a poder correr a querie que se encontra imediatamente abaixo (linha 58). Neste file o utilizador pode verificar o resultado dessas queries com e sem índices, (caso o utilizador deseje executar um índice basta simplesmente executar

todos os códigos que comecem com “CREATE NONCLUSTERED”) caso deseje remover os índices basta simplesmente correr os códigos de drop index. Existe mais informação detalhada na secção de otimização e execução de consultas Assim como nos ficheiros que se encontram na pasta “Traces do SQL Profiler (sem e com Indexes)” que contem os traces com e sem índices para o utilizador verificar a diferença de performance. Caso o Utilizador deseje verificar por si mesmo, nesse caso deverá ir a tools na parte de cima do SSMS escolher sql server profiler e fazer a autenticação via Windows que irá levar o utilizador a uma janela com as trace properties, aqui o utilizador deverá ir a Events selection e remover todas os eventos menos os dois últimos e por fim clicar em run. Agora sempre que uma das queries for executada o numero de reads e a sua duração deverá aparecer to trace.

No ficheiro 12 - Transactions o utilizador deverá correr o código todo desde a linha 1 a 245 (basicamente não correr o exec), embora estes stored procedures já existiam na fase 1 como foram alterados para o propósito de testar a concorrência decidimos não os ter já preparados no ficheiro .bak.

Após todos os stored procedures terem sido criados , abra também o ficheiro 12.1 - Transactions 2nd query. Assim que tiver ambos abertos tente correr o código da linha 1 e 3 do ficheiro 12.1. Irá dar um erro “Important Column Value doesnt Exist.” Que afirma que esta encomenda embora tenha sido criada não existe nenhum produto para alterar a quantidade, agora corra o exec do ficheiro 12 (linha 248) e rapidamente mude para o ficheiro 12.1 e tente executar o sp\_AlterQuantity novamente. Deverá agora existir um período de espera onde a o sp\_AlterQuantity irá esperar que os novos dados sejam efetivados e só depois é que verificara que existe um produto na encomenda e irá alterar a sua quantidade.Pode também ver vídeo da pasta Video Demonstration of Transactions at work para perceber melhor como a transacção é efectuada.

No ficheiro 13 – Permission, o utilizador pode correr o código todo da linha 1 a 85, isto irá criar os roles para garantir uma flexibilidade nas permissões (Grant / deny / revoke), assim como os logins, os users,a adição de certos users a determinados roles, e por fim os grants para lhes dar permissão. Após tudo isto ter sido feito o utilizador pode verificar no código abaixo (linhas 90+) o que o cada user pode fazer, para alem disto o user pode também ir ao connect do object explorer do lado esquerdo e fazer login como um dos users que foi criado, (ver vídeo da pasta Video Demonstration of Permissions Login test) para perceber melhor como o login é efectuado.

Em relação á replicação por favor ver o que foi escrito na secção 13 que contem uma forma extremamente bem detalhada de como realizar a replicação(ver também as imagens da pasta “Criação de uma replicação”) (assim como os ficheiros 14 e 15), infelizmente o grupo teve com problemas em conseguir finalizar a replicação a 100%.

