

|  |
| --- |
| Complementos de Bases de Dados 2019/2020  Licenciatura em Engª. Informática |
| 1ª Fase Relatório Técnico |

Turma: SW-02

Horário de Laboratório: 6ªFeira 16:30/18:30

Docente: João Lazaro

Grupo nº3  
nº190221109 João Rosete

# Sumário Executivo

O presente relatório dispõe da documentação do desenvolvimento do projeto, no âmbito da Unidade Curricular Complementos de Base de Dados.

Este projeto ofereceu-nos o objetivo principal de corrigir e a atualizar um sistema de base de dados já existente que pertencia à empresa AdventureWorks que iria sofrer uma restruturação que lhe permita gerir, de forma integrada, todo o processo de vendas das duas companhias.

Para a recriação desta Base de Dados é necessário a execução dos ficheiros em anexo por esta ordem:

Restore AdventureServicesOLD.Bak > CREATE\_DATABASE.SQL > CREATE\_TABLES.SQL > MIGRATE\_DATA.SQL > TRIGGER.SQL > VIEWS.SQL > STORE\_PROCEDURES.SQL > FUNCTIONS.SQL

# Especificação de Requisitos

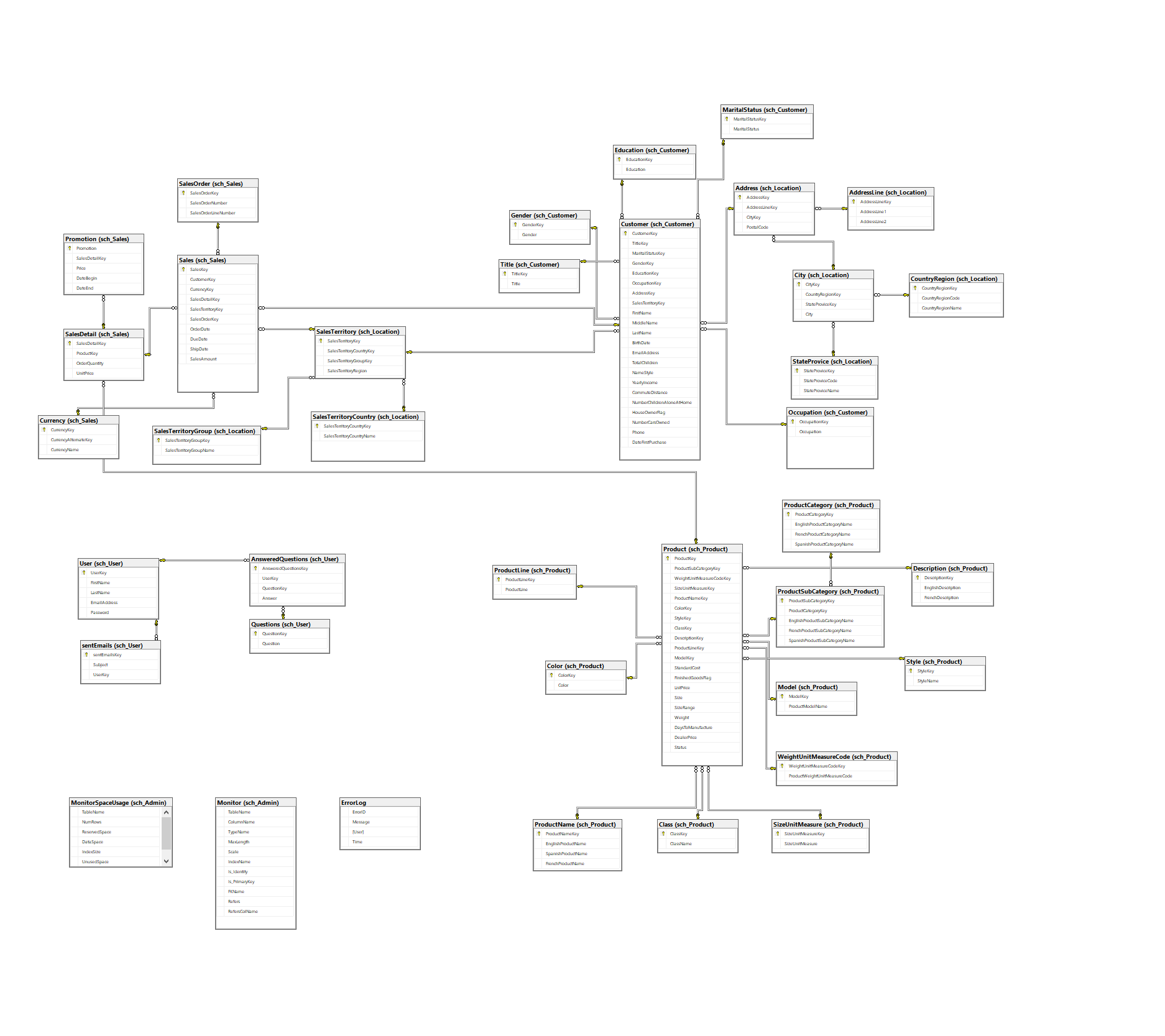
* 1. Requisitos funcionais

|  |  |  |
| --- | --- | --- |
| ID | Descrição | Implementado  (S/N) |
| RF01 | O sistema deverá permitir a existência de utilizadores de acesso à aplicação | S |
| RF02 | O sistema deverá permitir a autenticação de Utilizadores usando o seu email e password | S |
| RF03 | O sistema deverá permitir o reset da password para novos utilizadores | S |
| RF04 | O sistema deverá permitir enviar emails ‘sentemails’ para o Utilizador quando faz o reset da password | S |
| RF05 | O sistema deverá permitir a organização de produtos em SubCategorias/Categorias | S |
| RF06 | O sistema deverá permitir a gestão de Utilizadores, Categorias, SubCategorias e Produtos | S |

* 1. Atores

|  |  |
| --- | --- |
| Nome | Descrição |
| Customer | Este customer tem acesso a visualizar Views referentes ao Schema dos customers e tem a possibilidade de acesso aos Storage Procedures de gestão do carrinho de compras |
| Utilizadores | Têm a possibilidade de fazer a autenticação na aplicação. |
| Administradores | O Administrador tem como funções gerir produtos e clientes e para isto disponibiliza de acesso aos Storage procedures do schema dos clientes e aos storage procedures dos produtos |

# Modelo Relacional (*Modelo de dados*)

**

# Dimensionamento do Layout

O projeto tinha como necessidade principal manter os registos relativos às vendas, clientes e produtos da empresa e para isso uma vez que estes se encontravam fracamente normalizados houve a necessidade da criação de tabelas adicionais que permitissem a gestão mais detalhada de cada registo.

Em relação a criação da Encomenda foi feita a alteração para ser possível adicionar e remover produtos numa encomenda.

Também houve a necessidade da criação de tabelas que guardassem registos referentes a erros gerados na execução de procedures.

Quanto à organização dos filegroups obtou-se por dividir os dados por 5 filegroups:

* FG\_Admin – Neste filegroup ficam guardados os dados referentes a todo o schema sch\_Admin
  + Este filegroup irá conter a monitorização a informação dos catálogos de cada tabela.
* PRIMARY – Neste filegroup ira ser criado uma tabela de erros
  + Este filegroup ira ser guardar todos os tipos de erros das Store Procedures.
* FG\_User – Neste filegroup ficam guardados os dados referentes a todo o schema sch\_User
  + Este é um filegroup irá conter as informações dos Utilizadores das suas Questões e respostas como também dos seus emails.
* FG\_Location – Neste filegroup ficam guardados os dados referentes a todo o schema sch\_Location
  + Este é um filegroup que irá conter todas as localizações, address que vão ser utilizadas pelos os customers, products e sales.
* FG\_Customer– Neste filegroup ficam guardados os dados referentes a todo o schema sch\_Customer
  + Este é um filegroup que irá conter todas as informações associadas ao Customer
* FG\_Sales– Neste filegroup ficam guardados os dados referentes a todo o schema sch\_Sales
  + Este é um filegroup que irá conter todas as informações associadas não só as sales mas como também ao customer e products.
* FG\_Products – Neste filegroup ficam guardados os dados referentes a todo o schema sch\_Products
  + Este é um filegroup que irá conter todas as informações associadas aos products.

# Schemas

|  |  |
| --- | --- |
| Nome | Descrição |
| sch\_Customer | Este schema destina-se a agrupar registos, funções e procedimentos relativos às informações dos customers |
| sch\_Location | Este schema destina-se a agrupar registos, funções e procedimentos relativos às informações das sales, customers e products |
| sch\_Sales | Este schema destina-se a agrupar registos, funções e procedimentos relativos às informações de cada compra realizada pelo os customers |
| sch\_Users | Este schema destina-se a agrupar registos, funções e procedimentos relativos às informações pessoais dos users |
| sch\_Products | Este schema destina-se a agrupar registos, funções e procedimentos relativos às informações dos products |

# Views

|  |  |
| --- | --- |
| Nome | Descrição |
| sch\_Sales.v\_SalesPerYear | Esta view permite obter o total das monetário das sales por ano |
| v\_SalesPerYearOld | Esta view permite obter o total das monetário das sales por ano da antiga base de dados |
| sch\_Sales.v\_SalesPerYear\_SalesTerritoryCountry | Esta view permite obter o total das monetário das sales de SalesTerritoryCountry por ano |
| v\_SalesPerYear\_SalesTerritoryCountryOld | Esta view permite obter o total das monetário das sales de SalesTerritoryCountry por ano da antiga base de dados |
| sch\_Sales.v\_SalesPerYear\_ProductSubCategory | Esta view permite obter o total das monetário das sales de ProductSubCategory por ano |
| sch\_Sales.v\_SalesPerYear\_ProductCategor | Esta view permite obter o total das monetário das sales de ProductCategory por ano |
| v\_SalesPerYear\_ProductCategoryOld | Esta view permite obter o total das monetário das sales de ProductCategory por ano da antiga base de dados |
| v\_SalesPerYear\_ProductSubCategoryOld | Esta view permite obter o total das monetário das sales de ProductSubCategory por ano da antiga base de dados |
| sch\_Sales.v\_Customer\_SalesTerritoryCountry | Esta view permite obter o total de Customers de SalesTerritory por ano |
| sch\_Sales.v\_Customer\_SalesTerritoryCountryOld | Esta view permite obter o total de Customers de SalesTerritory por ano da antiga base de dados |
| sch\_Customer.v\_ProductPerCustomer | Esta view permite obter o total de Produtos adquiridos por cada Cliente |
| sch\_User.v\_QuestionPerUser | Esta view permite obter, para dcada utilizador, as suas perguntas de segurança, e quais as respetivas respostas. |
| sch\_Customer.v\_NumberOfCustomersPerCity | Esta view permite obter a quantidade de clientes que existem por pais |

# Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Atributos | Descrição |
| sch\_User.fn\_authUser | sch\_User.User | @EmailAddress NVARCHAR(25)  @Password NVARCHAR(25) | Faz a autenticação de um user usando o seu emailAddress e password |

# Stored procedures

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Atributos | Descrição | |
| [sch\_User].sp\_recoverPassword | @UserKey INT  @Answer1 NVARCHAR(25) @Answer2 NVARCHAR(25) @Answer3 NVARCHAR(25) | | Esta stored procedure permite recupera a palavra passe de um utilizador após a inserção de 3 respostas previamente selecionadas pelo mesmo, duma maneira correta. |
| [sch\_Product].sp\_alterProductStatus | @ProductKey INT @Status NVARCHAR(55) | | Esta stored procedure permite alterar o estado de um produto através do seu identificador. |
| [sch\_Sales].sp\_createOrder | @CustomerKey INT @CurrencyKey INT @SalesDetailKey INT @TerritoryKey INT @SalesOrderKey INT @OrderDate DATETIME @DueDate DATETIME @ShipDate DATETIME | | Esta stored procedure permite criar uma nova encomenda com os valores inseridos nos parâmetros. |
| [sch\_Sales].alterQuantity | @SalesDetailKey INT,  @OrderQuantity INT | | Esta stored procedure permite alterar a quantidade de produtos da encomenda. |
| [sch\_Sales].PromotionOrder | @SalesDetailKey INT,  @Price FLOAT,  @DateBegin DATETIME, @DateEnd DATETIME | | Esta stored procedure permite adicionar uma promoção a uma encomenda especifica, indicando-lhe o preço que deseja |
| [sch\_Product].sp\_changeProductSubCategory | @productID int, @newSubCategory nvarchar(25) | | Esta stored procedure permite alterar as datas de uma promoção, extendendo ou diminuindo a sua duração. |
| [sch\_Admin].sp\_sizeUsage |  | | Esta stored procedure insere na tabela [sch\_Admin].MonitorSpaceUsage os números totais de registo por tabela na base de dados e valores de espaço ocupado, reservado e não utilizado |
| [sch\_Admin].sp\_updateMonitorData |  | | Esta stored procedure atualiza os dados da tabela MonitorSpaceUsage, de forma a corresponderem com o tempo real |
| [sch\_Admin].sp\_showLatestData |  | | Esta stored procedure mostra ao utilizador os dados da view v\_DisplayMonitorData |
| dbo.sp\_GenerateGenerators | @table varchar(255) | | Esta stored procedure executa outras 3 SPs, as de gestão de tabela, ou seja, cria uma SP para inserção de dados, uma para remoção e outra para atualização. |
| dbo.sp\_GeneratorsInsert | @table varchar(255) | | Esta stored procedure cria uma stored procedure de **inserção** de dados para a tabela que foi inserida como argumento, |
| dbo.sp\_GeneratorsUpdate | @table varchar(255) | | Esta stored procedure cria uma stored procedure de **atualização** de dados para a tabela que foi inserida como argumento, |
| dbo.sp\_GeneratorsDelete | @table varchar(255) | | Esta stored procedure cria uma stored procedure de **remoção** de dados para a tabela que foi inserida como argumento, |

# Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Tabela | Descrição |
| Dbo.ResetPassword | AFTER INSERT | Sch\_User.User | Faz reset da password após um insert de um user |

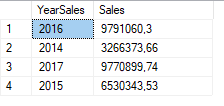
# Consultas

## Verificação da conformidade dos dados

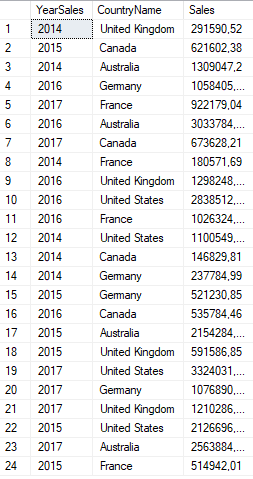
A verificação da conformidade dos dados foi realizada através de querys sobre as duas bases de dados. Querys estas que estão disponíveis no ficheiro “VIEWS.SQL” sendo que existe um par de querys para cada verificação dos dados sendo que a primeira é sobre a nova base de dados e a segunda sobre a antiga.

A baixo estarão apresentadas imagens sobre os resultados que se pretendiam para cada verificação.

**Total monetário de vendas por ano:**



**Total Monetário de vendas de Sales Territory Country por Ano:**

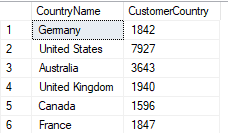


**Total Monetário de vendas por Product SubCategory por Ano:**



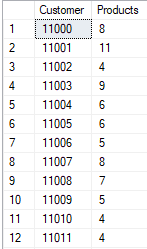
**Total Monetário de vendas por Product Category por Ano:**

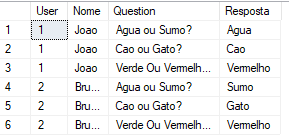
**Números de Clientes por Sales Territory Country:**



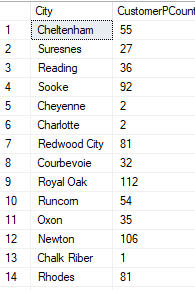
## Outras consultas

**Número de Compras Por Cliente:**



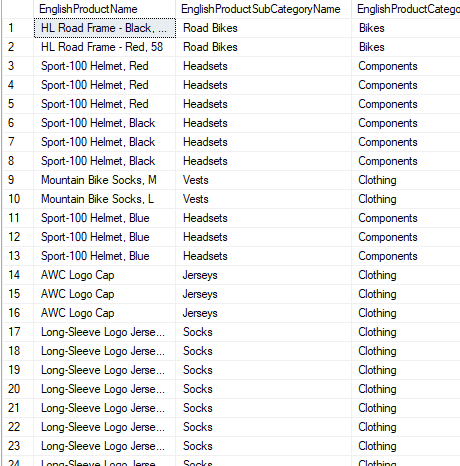
 **Questões e Respostas de Cada User:**

**Número de Clientes Por Cidade:**



Existem mais registos mas a imagem iria ficar demasiado grande portanto cortamos e mostramos apenas alguns registos

**Categoria e SubCategoria de um Producto:**



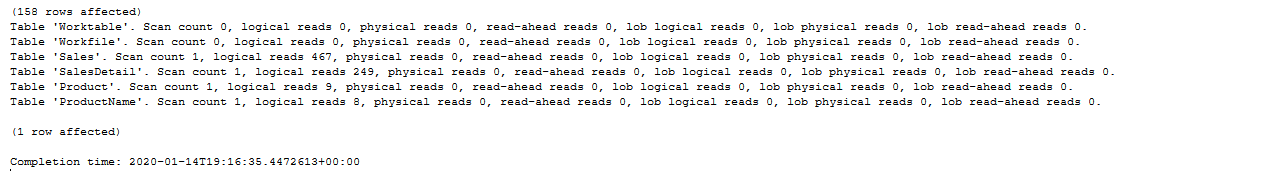
# Índices

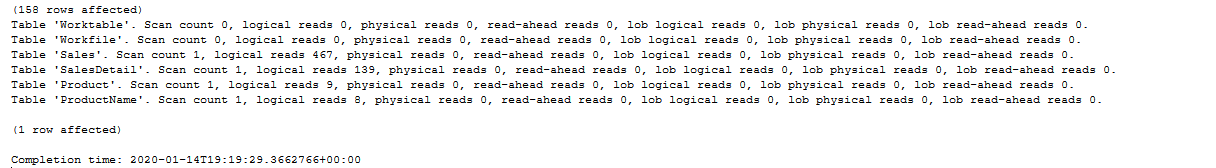
|  |  |  |
| --- | --- | --- |
| Designação | Tabela | Justificação/Consultas |
| v\_AllPackages\_In\_TheLast2Years\_INDEX | Sales | Índice sugerido pelo o Tunnng Advisor para a otimização da view v\_AllPackages\_In\_TheLast2Year |
| v\_AllPackages\_In\_TheLast2Years\_INDEX\_SalesTerritoryKey\_INDEX | SalesTerritoryCountry | Índice sugerido pelo o Tunnng Advisor para a otimização da view v\_AllPackages\_In\_TheLast2Year e da view v\_TotalValue\_Anual\_Per\_Country |
| v\_Volume\_Sales\_Per\_Product\_INDEX | SalesDetail | Índice adaptado seguindo a sugestão do Tuning Advisor para a otimização das queries da view v\_Volume\_Sales\_Per\_Product |
| v\_TotalValue\_Anual\_Per\_Country\_INDEX | Sales | Índice adaptado seguindo a sugestão do Tuning Advisor para a otimização das queries da view v\_TotalValue\_Anual\_Per\_Country e da view  MaxTotalValue\_Per\_Country |

# Otimização e Execução de Consultas

Serão apresentadas imagens relativas ao plano de execução que vão demonstrar a otimização das views pedidas no enunciado, será demonstrado com o plano sem índices e o plano com índices para se poder fazer a comparação.

**View - [sch\_Product].v\_Volume\_Sales\_Per\_Product**

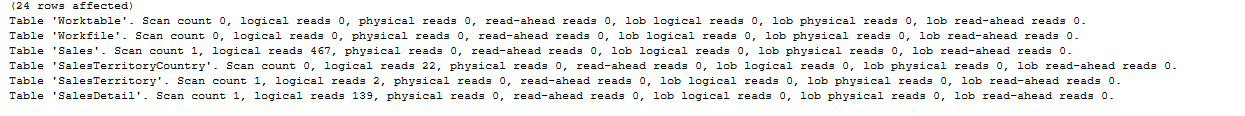
Sem Índices

Com Índices

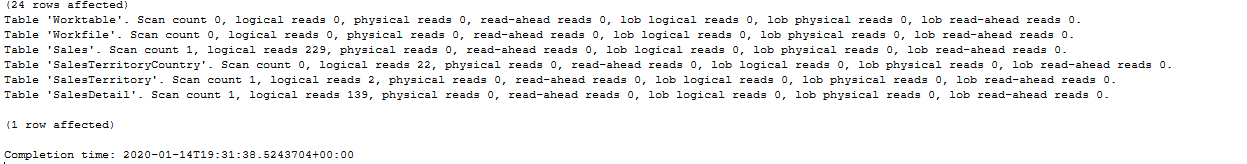
Como podemos verificar nestas imagens com a adição dos **índices non-clustered** obtivemos uma descida no número de leituras **“Logic Reads”** na tabela **“SalesDetail”** o que demonstra uma otimização significante na performance da query.

**View - [sch\_Sales].v\_TotalValue\_Anual\_Per\_Country**

Sem Índices

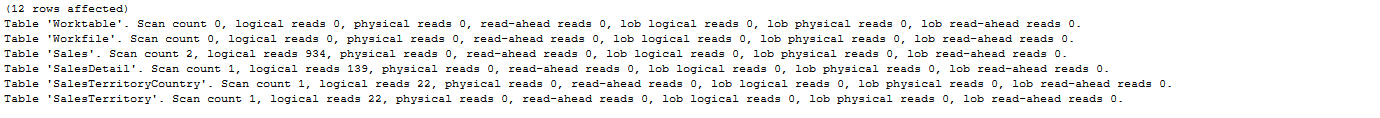
****

Com Índices

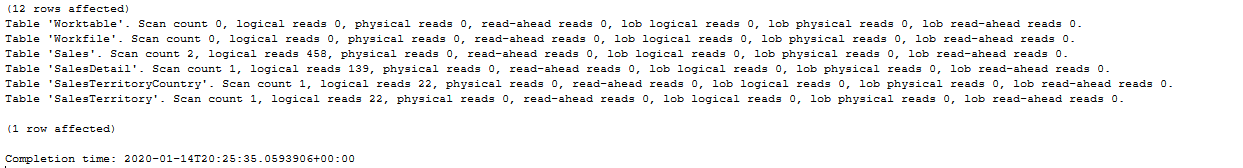


Quanto ao plano de execução desta view seguimos o Tunning Advisor melhorando os logical reads sendo 467 reads sem índice, e com índice 229 reads, da tabela Sales tendo assim uma otimização desta view.

**View - [sch\_Sales].v\_MaxTotalValue\_Per\_Country**

****Sem Índices

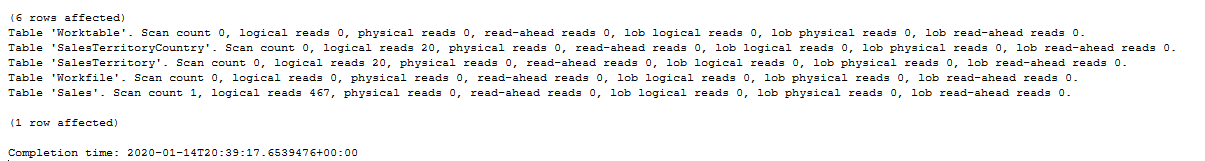
Com Índices



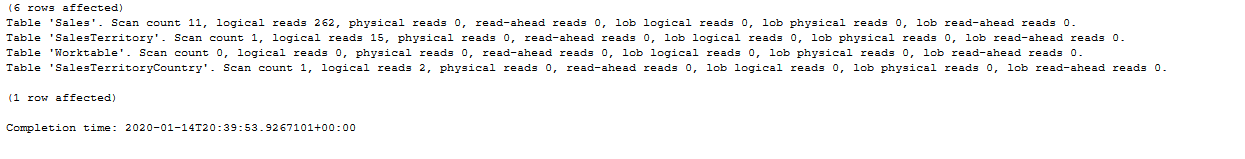
Ao utilizarmos o índice **“v\_TotalValue\_Anual\_Per\_Country\_INDEX”** que foi utilizado na view **v\_TotalValue\_Anual\_Per\_Country** podemos assim melhorar esta view porque utiliza as mesmas tabelas a execução da query passado de 934 logical reads sem índices, para 458 logical reads com índices, melhorando bastante a performance da query.

**View - [sch\_Sales].v\_AllPackages\_In\_TheLast2Years**

Sem Índices



Com Índices

**

Para a presente view tivemos que usar 2 índices para termos um decremento de logical reads na tabela Sales para a performance da view.

# Controlo de Concorrência

*Definir níveis de isolamento adotados e controlo transacional*

# Segurança e Controlo de Acessos

Neste capítulo iremos demonstrar e explicar como foi feita a administração de controlo de acessos a nossa base de dados:

* Logins
  + **login\_Administrador** com a password = “passAdmin”;
  + **login\_Gestor\_de\_Marketing** com a password = “passGestor”;
  + **login\_Utilizador\_Anonimo** com a password = “passUtilizadorA”;
  + **login\_Utilizador\_Registado** com a password = “passUtilizadorR”;
* Users
  + **user\_Administrador** com o login **login\_Administrador**;
  + **user\_Gestor\_de\_Marketing** com o login, **login\_Gestor\_de\_Marketing;**
  + **user\_Utilizador\_Anonimo** com o login, **login\_Utilizador\_Anonimo;**
  + **user\_Utilizador\_Registado** com o login, **login\_Utilizador\_Registado;**
* Permissões de acesso(Roles)
  + **role\_managment\_all**
    - Permissões para as operações SELECT,INSERT e UPDATE nos schemas: Admin, Customer, Location, Product, Sales e User;
  + **role\_view\_promotion\_campaigns**
    - Permissões para a operação SELECT na tabela Promotion do schema Sales;
  + **role\_view\_products**
    - Permissões para a operação SELECT no schema dos Products;
  + **role\_view\_customerInfo\_and\_salesInfo**
    - Permissões para a operação SELECT no schema dos Products, Customer e Sales;
* Atribuição dos Roles ao seus respectivos Users
  + **role\_managment\_all**
    - user\_Administrador
  + **role\_view\_promotion\_campaigns**
    - user\_Gestor\_de\_Marketing
  + **role\_view\_products**
    - user\_Utilizador\_Anonimo
  + **role\_view\_customerInfo\_and\_salesInfo**
    - user\_Utilizador\_Registado

Conforme foi pedido no enunciado foi criado 4 utilizadores diferentes para estes foram atribuídos 4 roles diferentes que permitem fazer uma série de ações a base de dados. Ao administrador ser possível controlar a base de dados toda. Ao Gestor de Marketing ser possível ver as promotions de cada encomenda. O utilizador anónimo só consegue ver os produtos da nossa base de dados por último o nosso utilizador autenticado consegue ver os seus dados pessoais, as suas compras e os produtos.

**Segurança:**

Para a realização da encriptação dos dados do Utilizador, nomeadamente Password e Email, foi optado fazer a encriptação do Email utilizando uma chave Simétrica (EncryptInformation) e para as passwords foi feita a encriptação usando a função Hash SHA1 para obtermos um hashing da password.

Foi feita uma encriptação diferente para ambos os campos pois no caso email este é um tipo de dados que usualmente é utilizado pelas aplicações para certas funcionalidades (Exemplo: Login) ou seja precisamos de poder aceder á informação real deste forma, encriptando, podemos através da chave simétrica desencriptar novamente para obtermos o seu valor original. No caso da password optamos por um hashing da mesma para que não pudesse obter o seu valor original mas sendo possível fazer a comparação num login de uma aplicação

# Esquema de Replicação

*Descrição do esquema de replicação implementado*

# Política de *Backups*

Foi decidido que a base de dados deste projeto irá utilizar o modelo de recuperação FULL pois esta base de dados é que contêm toda a informação necessário para o melhor funcionamento de uma aplicação.

No tipo de backups a serem utilizados na base de dados optado pela a combinação de 3 tipos diferentes sendo eles o Full, Diferencial e Transaction Log.

Tivemos em conta a carga de sistema de cada tipo de backup e optamos por um período de backup menor, para um backup com menos carga de sistema e um período de tempo maior para o backup com maior carga. Foi assim que concluímos que uma bola solução seria por realizar um backup Full semanalamente, um backup Diferencial diariamente e por fim um backup Transaction Log a cada 2 horas.

# Descrição da Demonstração

*Encadeamento de procedimentos que permita o teste visualizar o correto funcionamento da base de dados face aos requisitos*