

# Complementos de Bases de Dados 2022/2023

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

Relatório Técnico

Turma: 2<sup>o</sup>L EI-SW-01

Horário de Laboratório: 5<sup>a</sup>f 17h00

Docente: João Portelinha

Grupo

N<sup>o</sup>202100225, André Meseiro

N<sup>o</sup>202100230, Pedro Anjos

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

### 1. Introdução

O projeto final da disciplina de Complementos de Bases de Dados tem como objetivo a familiarização com a parte administrativa de bases de dados relacionais. O trabalho irá incidir tanto ao nível da base de dados como do sistema gestor de bases de dados e também sobre a lógica subjacente à utilização dos dados disponibilizados, ou seja, a partir de uma base de dados não normalizada e um conjunto de ficheiros em formato de texto e Excel (desatualizados, portanto) de uma empresa com o nome “World Wide Importers” (WWI), reformular este sistema para uma nova base de dados, de forma integrada, com todo o processo de vendas desta empresa.

### 2. Especificação de Requisitos

ID	Descrição	Implementado (S/N)
R01	Criação da nova base de dados normalizada	S
R02	Migração dos dados da base de dados antiga para a nova	S
R03	Consultas de verificação/validação da migração (requisitos mínimos)	S
R04	Gestão de vendas utilizando SPs, UDFs, e Triggers para os processos de negócio propostos	S
R05	SPs geradores e monitorização (metadados)	S
R06	Testes	S
R07	Tratamento de erros	S
R08	Definição do layout da nova base de dados	S
R09	Implementação de Índices para as queries relacionadas com as vendas	S
R10	Definição da estratégia de Backup e descrição da Recuperação para o Sistema de Informação	S
R11	Definição dos níveis de acesso à informação (objetos e privilégios de acesso aos mesmos)	S
R12	Implementação do código relativo à Encriptação das passwords dos utilizadores do sistema	S
R13	Definição dos níveis de isolamento para os processos referidos no enunciado	S
R14	Criação de um sistema autónomo que tem por base o SGBD MongoDB para as tarefas (e respetivos comandos) referidas no enunciado	S

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

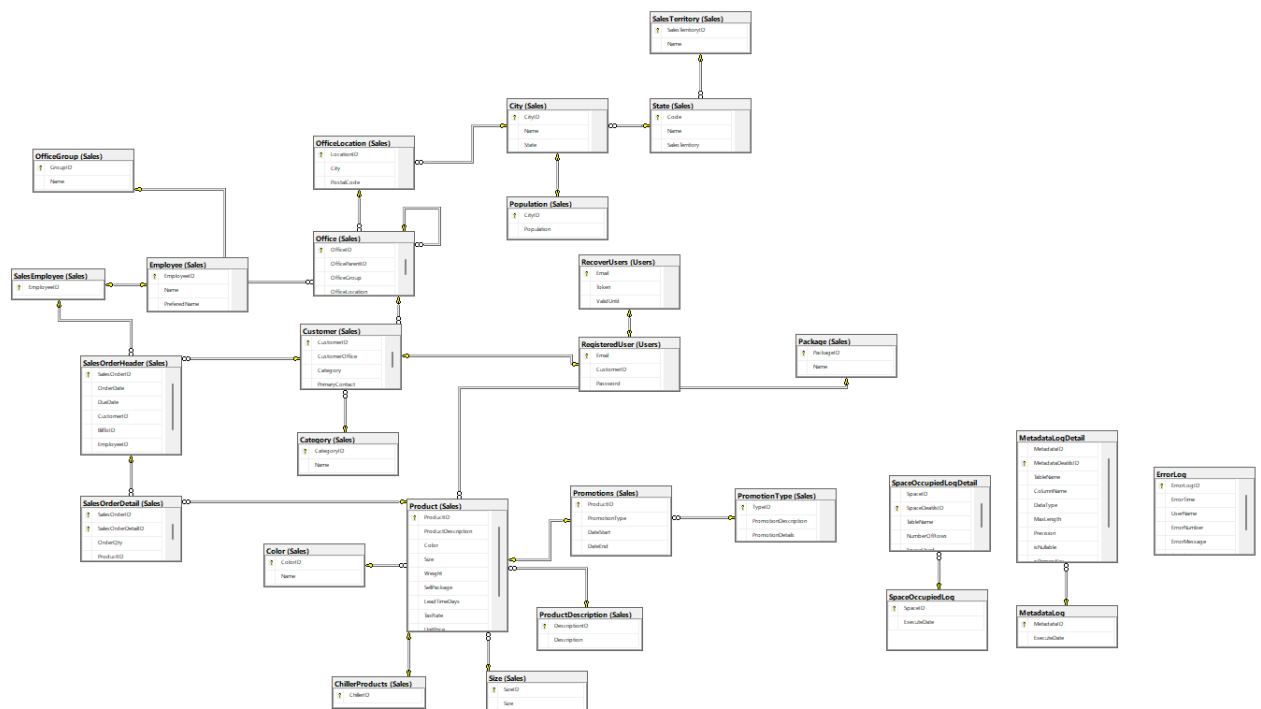
RM01	Existência de utilizadores de acesso à aplicação, para os clientes	S
RM02	Gestão de utilizadores	S
RM03	Gestão de promoções, para os produtos	S
RM04	Identificação de outros cenários possíveis que podem ocorrer no acesso concorrente ao site	S

### 3. Alterações/Melhorias ao Relatório da 1ª Fase

Filegroup SaleDat: alteração do tamanho máximo de 15MB para 20MB, pois não permitia a criação dos índices devido a não ter espaço livre disponível.

### 4. Relacional (Modelo de dados)

#### 4.1 Diagrama do Modelo Relacional



### 5. Definição do Layout

O layout proposto para a nova base de dados, considerando todas as tabelas criadas e o tipo de ficheiros existentes é o seguinte (assumir taxa de crescimento para 1 ano, respetiva para cada FG definida abaixo na tabela):

- Primary FileGroup: WWIGlobalDat com um primary file wwiglobaldat.mdf;

- LogFG WWIGlobalLog com log file wwigloballog.ldf;

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

- UFG1: ProductDat – ChillerProducts + Product + ProductDescription + Promotions + PromotionType (tudo o que é relacionado com produtos) wwiglobalproduct.ndf;
- UFG2: SaleDat – SalesEmployee + SalesOrderDetail + SalesOrderHeader + SalesTerritory (tudo o que é relacionado com produtos) wwiglobalsale.ndf
- UFG3: LocDat - City + State + Population (tudo o que é relacionado com localização) wwigloballoc.ndf
- UFG4: OfficeDat - Office + OfficeGroup + OfficeLocation (tudo o que é relacionado com escritórios) wwiglobaloffice.ndf
- UFG5: PersonDat - Customer + Employee (tudo o que é relacionado com pessoas) wwiglobalperson.ndf
- UFG6: MiscDat - Category + Color + Package + Size (tudo o que é informação variada/diversa) wwiglobalmisc.ndf
- UFG7: UserDat - RecoverUsers + RegisteredUser (tudo o que é relacionado com utilizadores) wwiglobaluser.ndf
- UFG8: MetaDat – MetadataLog + MetadataLogDetail + SpaceOccupiedLog + SpaceOccupiedLogDetail (tudo o que é relacionado com metadados) wwiglobalmeta.ndf

### 5.1 Identificação do espaço ocupado por tabela

Nome Tabela	Dimensão do Registo	Nº de Registos (inicial/final)
ErrorLog	548KB	Inicial: 0, final: 25000
ChillerProducts	76KB	Inicial: 8, final: 10
Product	128KB	Inicial: 229, final: 270
ProductDescription	176KB	Inicial: 92, final: 115
Promotions	86KB	Inicial: 0, final: 10
PromotionType	206KB	Inicial: 0, final: 3
SalesEmployee	76KB	Inicial: 10, final: 20
SalesOrderDetail	5208KB	Inicial: 115383, final: 120383
SalesOrderHeader	2534KB	Inicial: 63719, final: 68719
SalesTerritory	126KB	Inicial: 9, final: 12
City	1196KB	Inicial: 37940 , final: 38120

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

State	128KB	Inicial: 57 , final: 57
Population	528KB	Inicial: 26878, final: 27005
Office	88KB	Inicial: 402 , final: 410
OfficeGroup	106KB	Inicial: 2 , final: 2
OfficeLocation	90KB	Inicial: 402 , final: 410
Customer	124KB	Inicial: 402 , final: 410
Employee	136KB	Inicial: 19 , final: 39
Category	96KB	Inicial: 5 , final: 5
Color	106KB	Inicial: 9 , final: 9
Package	106KB	Inicial: 5 , final: 5
Size	116KB	Inicial: 44 , final: 44
RecoverUsers	230KB	Inicial: 0 , final: 17
RegisteredUsers	298KB	Inicial: 0 , final: 103
MetadataLog	84KB	Inicial: 0 , final: 1
MetadataLogDetail	211KB	Inicial: 0 , final: 72
SpaceOccupiedLog	84KB	Inicial: 0 , final: 1
SpaceOccupiedLogDetail	138KB	Inicial: 0 , final: 23

### 5.2 Especificação dos Filegroups

Nome Filegroup	Tabelas associadas	Parâmetros
----------------	--------------------	------------

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

WWIGlobalLog	-----	Dimensão inicial: 1MB, dimensão final: 15MB, taxa de crescimento: 14MB
ProductDat	ChillerProducts, Product, ProductDescription, Promotions, PromotionType	Dimensão inicial: 5MB, dimensão final: 6MB, taxa de crescimento: 1MB
SaleDat	SalesEmployee, SalesOrderDetail, SalesOrderHeader, SalesTerritory	Dimensão inicial: 10MB, dimensão final: 20MB, taxa de crescimento: 10MB
LocDat	City, State, Population	Dimensão inicial: 5MB, dimensão final: 6MB, taxa de crescimento: 1MB
OfficeDat	Office, OfficeGroup, OfficeLocation	Dimensão inicial: 2MB, dimensão final: 3MB, taxa de crescimento: 1MB
PersonDat	Customer, Employee	Dimensão inicial: 1MB, dimensão final: 4MB, taxa de crescimento: 3MB
MiscDat	Category, Color, Package, Size	Dimensão inicial: 1MB, dimensão final: 1MB, taxa de crescimento: 0MB
UserDat	RecoverUsers, RegisteredUsers	Dimensão inicial: 1MB, dimensão final: 5MB, taxa de crescimento: 4MB
MetaDat	MetadataLog, MetadataLogDetail, SpaceOccupiedLog, SpaceOccupiedLogDetail	Dimensão inicial: 1MB, dimensão final: 1MB, taxa de crescimento: 0MB

### 5.3 Schemas

Nome	Descrição
Sales	Este schema tem como objetivo representar tudo o que está relacionado com as vendas e a sua gestão
Users	Este schema tem como objetivo representar tudo o que está relacionado com utilizadores e a sua gestão

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

### 6. Verificação da migração de dados

#### 6.1 Consultas sobre as duas bases de dados

```
-- Verify Data Migration
-- Verifies the number of customers in both Databases
EXEC Sales.spNumberOfCustomersBothBDs;

-- Verifies the number of customers per category in both Databases
EXEC Sales.spNumberOfCustomersBothBDsPerCategory;

-- Verifies the number of sales per employee in both Databases
EXEC Sales.spNumberOfSalesBothBDsPerEmployee;

-- Verifies the total of sales per stock item in both Databases
EXEC Sales.spTotalValuePerProductBothBDs;

-- Verifies the total of sales per stock item per year in both Databases
EXEC Sales.spGetTotalSalesPerYearPerStockItem;

-- Verifies the number of sales per city per year in both Databases
EXEC spTotalSalesPerYearPerCity;
```

### 7. Programação

#### 7.1 Views

Nome	Descrição
vCityStates	Esta view permite obter a lista de cidades e estados
vStateSalesTerritory	Esta view permite obter a lista de estados e territórios
vCustomer	Esta view permite obter a lista de clientes com a respetiva informação relevante
vSalesEmployee	Esta view permite obter a lista de vendas associadas a cada empregado
vProducts	Esta view permite obter a lista de produtos com a respetiva informação relevante
vSalesHeader	Esta view permite obter a lista com informação relevante das “faturas” das vendas

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

<i>vSalesDetails</i>	<i>Esta view permite obter a lista com os detalhes relevantes das vendas</i>
----------------------	--

### 7.2 Functions

Nome	Atributos	Requisito	Descrição
<i>Sales.fnCalcTotalPriceSale</i>	<i>@SalesOrderDetailID INT</i>	R04	<i>Permite obter o preço total de uma venda</i>
<i>Sales.fnCheckIfSaleAllowsChillerProducts</i>	<i>@SalesOrderID INT</i>	R04	<i>Auxiliar - permite saber se uma venda pode ou não receber um produto do tipo “Chiller”</i>
<i>Sales.fnGetNumberCustomersPerCategoryOldDataBase</i>	<i>@Category VARCHAR(40)</i>	R03	<i>Permite obter o número de clientes, agrupado por categoria, da base de dados antiga</i>
<i>Sales.fnGetTotalSalesPerStockItemOldDataBase</i>	<i>@Description VARCHAR(100)</i>	R03	<i>Permite obter o total de vendas por stock item da base de dados antiga</i>
<i>Sales.fnCalculateTotalSalesYear</i>	<i>@year INT, @category VARCHAR(50)</i>	R09	<i>Permite ajudar no cálculo da taxa de crescimento de vendas por ano por categoria</i>

### 7.3 Stored procedures

Nome	Atributos	Requisito	Descrição
<i>Sales.spCreateSale</i>	<i>@DueDate DATE, @CustomerID INT, @BillToID INT, @EmployeeID INT, @TaxAmount MONEY</i>	R04	<i>Permite criar uma nova venda</i>
<i>Sales.spAddProductToSale</i>	<i>@SalesOrderID INT, @OrderQty INT, @ProductID INT, @UnitPrice MONEY</i>	R04	<i>Permite adicionar um produto a uma venda</i>
<i>Sales.spUpdateProductQty</i>	<i>@SalesOrderDetailID INT,</i>	R04	<i>Permite alterar a quantidade de um produto numa venda</i>



## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

	@NewOrderQty INT		
Sales.spRemoveProduct	@SalesOrderDetailID INT, @DeleteSale BIT	R04	Permite remover um produto de uma venda, com a opção de também remover a venda caso esta não tenha nenhum produto associado
Users.spAddUser	@email VARCHAR(50), @customerID INT, @password VARCHAR(100)	RM01	Permite adicionar um utilizador à base de dados
spLogError	-----	R07	Permite adicionar um erro à tabela de log de erros
Sales.spNumberOfCustomers BothBDs	-----	R03	Permite saber o número de clientes tanto da base de dados antiga como da nova
Sales.spTotalValuePerProductBothBDs	-----	R03	Permite obter o valor total por produto tanto da base de dados antiga como da nova
Sales.spNumberOfCustomers BothBDsPerCategory	-----	R03	Permite obter o número de clientes tanto da base de dados antiga como da nova
Sales.spNumberOfSalesBoth DBsPerEmployee	-----	R03	Permite obter o número de vendas agrupado por empregado tanto da base de dados antiga como da nova
color_migrate	-----	R02	Permite migrar os registos das cores da base de dados antiga para a nova
state_migrate	-----	R02	Permite migrar os registos dos estados da base de dados antiga para a nova
package_migrate	-----	R02	Permite migrar os registos dos pacotes da base de dados antiga para a nova
city_migrate	-----	R02	Permite migrar os registos das cidades da base de dados antiga para a nova
salesTerritory_migrate	-----	R02	Permite migrar os registos dos territórios das vendas da base de dados antiga para a nova

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

<i>category_migrate</i>		R02	Permite migrar os registos das categorias da base de dados antiga para a nova
<i>customer_migrate</i>		R02	Permite migrar os registos dos clientes da base de dados antiga para a nova
<i>employee_migrate</i>		R02	Permite migrar os registos dos empregados da base de dados antiga para a nova
<i>product_migrate</i>		R02	Permite migrar os registos dos products da base de dados antiga para a nova
<i>sale_migrate</i>		R02	Permite migrar os registos das vendas da base de dados antiga para a nova
<i>Sales.spGetTotalSalesBothBDsPerYearPerStockItem</i>		R03	Permite obter o número total de vendas das duas bases de por ano e por stock item
<i>Sales.spCreatePromotion</i>	@productID INT, @promotionDetail VARCHAR(30), @PromotionDescription VARCHAR(100), @DateStart DATE, @DateEnd DATE	RM03	Permite adicionar uma promoção a um produto
<i>Users.recoverPassword</i>	@email VARCHAR(100)	RM02	Permite enviar um token ao utilizador para este recuperar a sua password
<i>Users.spCheckTokenValid</i>		RM02	Permite verificar os tokens para dar reset à password e remove as que já expiraram
<i>Users.spEditUser</i>	@email VARCHAR(50), @propertyToChange VARCHAR(20), @value VARCHAR(100)	RM02	Permite editar uma propriedade de um utilizador

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

<i>Users.spRemoveUser</i>	@email VARCHAR(50)	RM02	Permite remover um utilizador
<i>Sales.spEditPromotionDate</i>	@productID INT, @startDate DATE, @endDate DATE	RM03	Permite alterar a data de uma promoção
<i>Sales.spUpdateProductPrice</i>	@ProductID INT, @NewUnitPrice INT	R13	Permite atualizar o preço de um produto

## 8. Catálogo/Metadados

### 8.1 Geradores

Nome	Atributos	Descrição
<i>spGenInserts</i>	@tableName VARCHAR(30)	Implementa o procedimento para inserir registos numa tabela
<i>spGenUpdates</i>	@tableName VARCHAR(30)	Implementa o procedimento para inserir updates numa tabela
<i>spGenDelete</i>	@tableName VARCHAR(30)	Implementa o procedimento para remover registos numa tabela

### 8.2 Monitorização

Nome	Atributos	Descrição
<i>vTableMetadata</i>	-----	Mostra a última execução da procedure <i>spAutoGenerateMetadata</i>
<i>vSizeOccupied</i>	-----	View auxiliar para listar o número de registos e espaço ocupado por tabela
<i>vTableSizeOccupied</i>	-----	Mostra a última execução da procedure <i>spAutoGenerateSpaceUsed</i>
<i>dbo.spGenerateMetadata PerTable</i>	@MetadataID INT, @tableName VARCHAR(50),	Implementa o procedimento para gerar os metadados para todas as tabelas da base de dados

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

	@schemaName VARCHAR(30)	
dbo.spGenMetadata	@MetadataID INT	Implementa o procedimento para gerar metadados para uma tabela da base de dados
dbo.spAutoGenerateMetadata		Implementa o procedimento para gerar de forma automática os metadados para todas as tabelas da base de dados
spGenerateSpacePerTable	@SpaceID INT, @tableName VARCHAR(50)	Implementa o procedimento para gerar o espaço ocupado por tabela da base de dados
dbo.spGenSpaceOccupied	@SpaceID INT	Implementa o procedimento para gerar o espaço ocupado por todas as tabelas da base de dados
dbo.spAutoGenerateSpaceUsed		Implementa o procedimento para gerar de forma automática o espaço utilizado e o número de registos de todas as tabelas da base de dados

## 9. Índices

### 9.1 Índices

Designação	Tabela	Justificação/Consultas
NONCI_SalesByCity	Sales.SalesOrderHeader	Otimiza o custo da consulta em 47.755% (de acordo com o MSSMS), indexa a coluna CustomerID e inclui a coluna EmployeeID.
NONCI_GrowthRate	Sales.SalesOrderHeader	Otimiza o custo da consulta em 58.2838% (de acordo com o MSSMS), indexa a coluna CustomerID e inclui a coluna OrderDate.

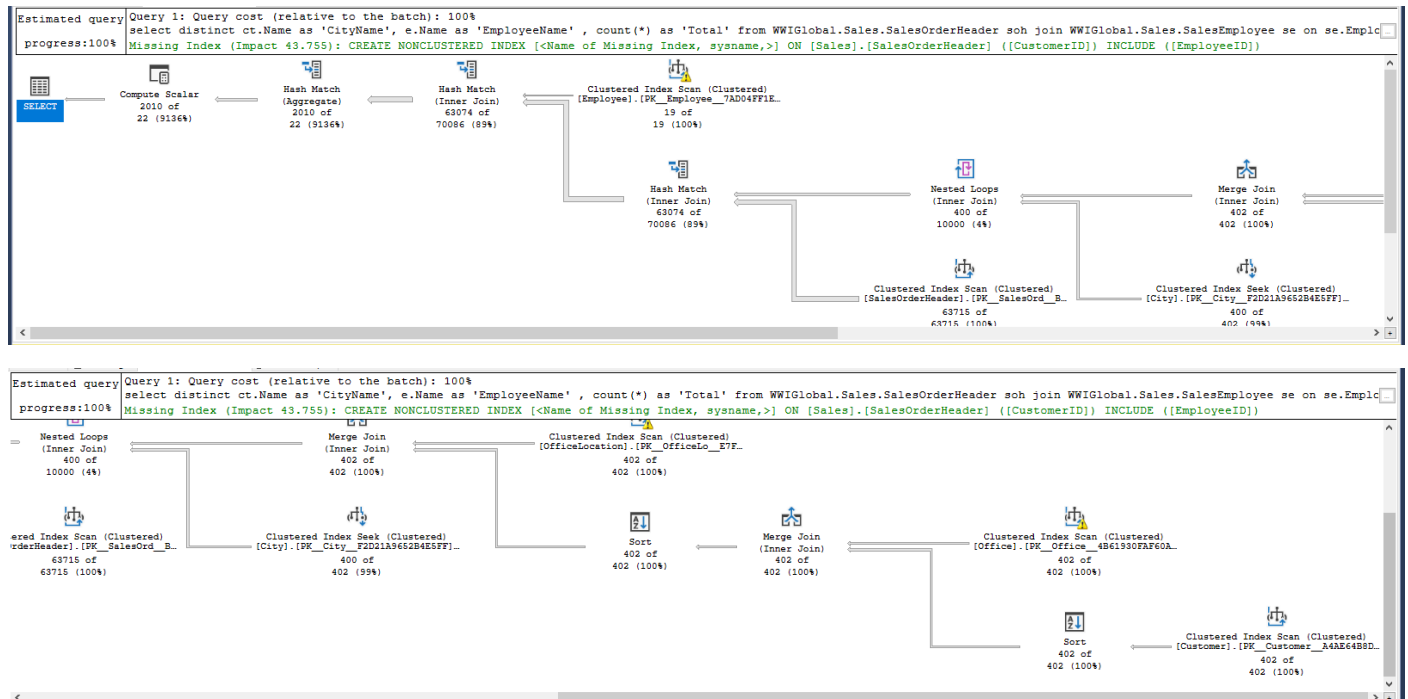
### 9.2 Otimização e Execução de Consultas

- Sobre a base de dados otimizada (normalizada) sem índices:

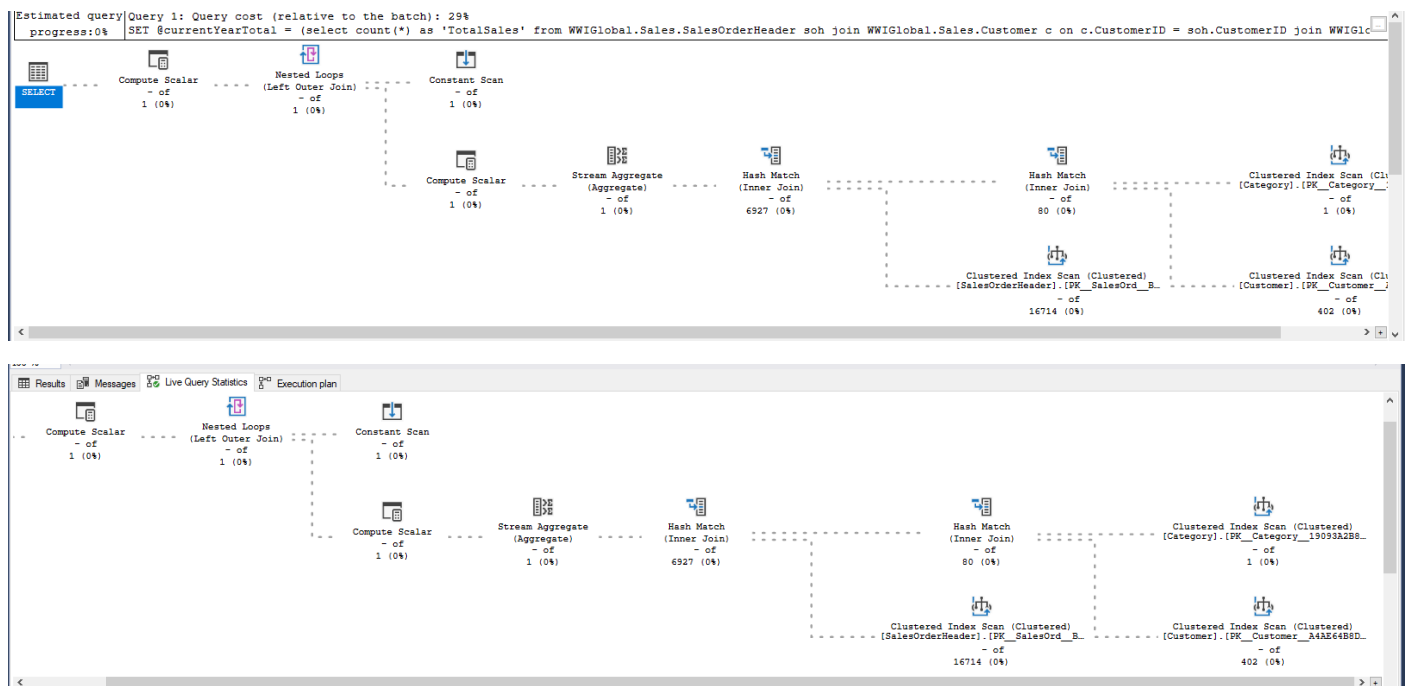
Estatísticas:

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

Query 1:

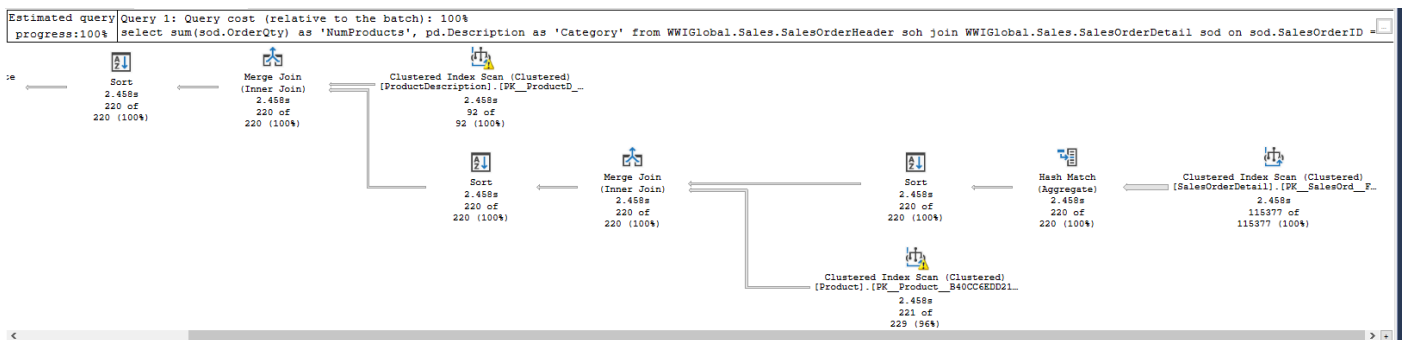
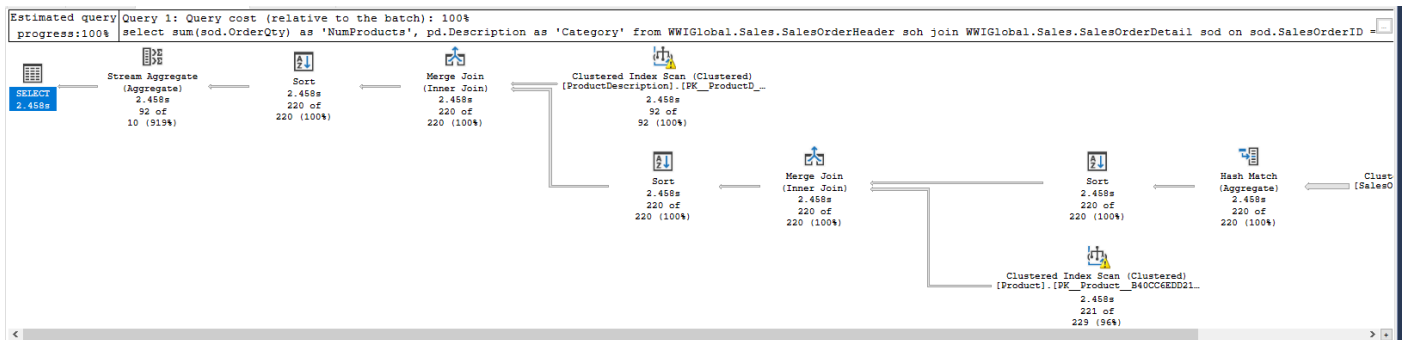


Query 2:



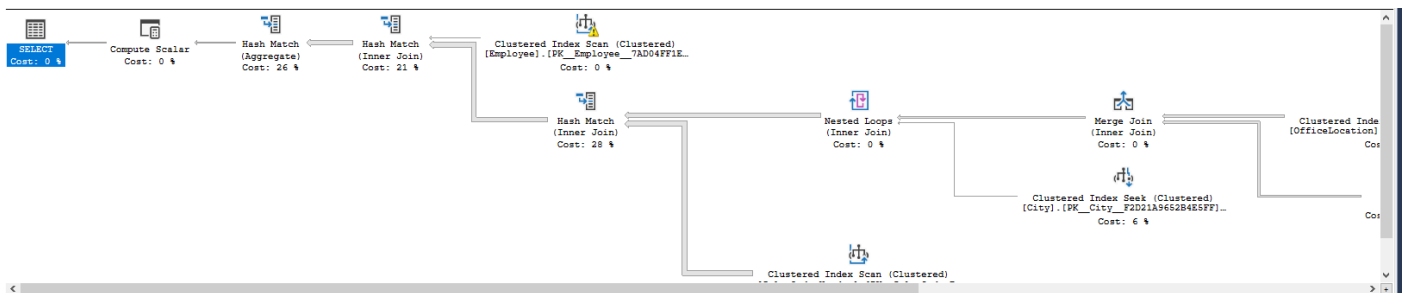
## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

### Query 3:

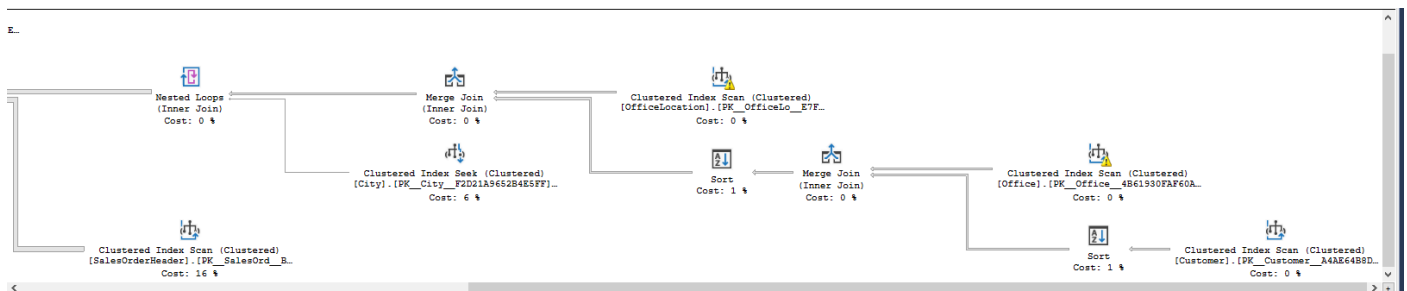


### Planos de execução:

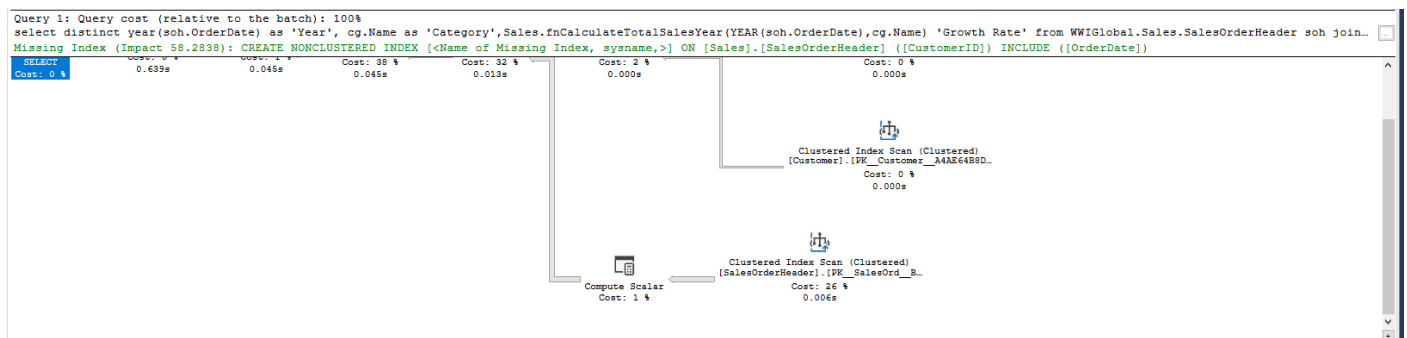
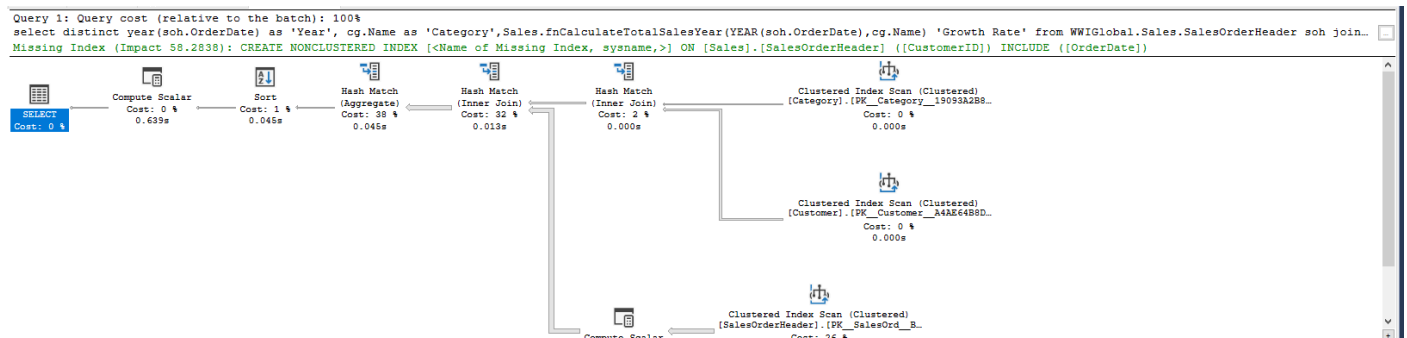
#### Query 1:



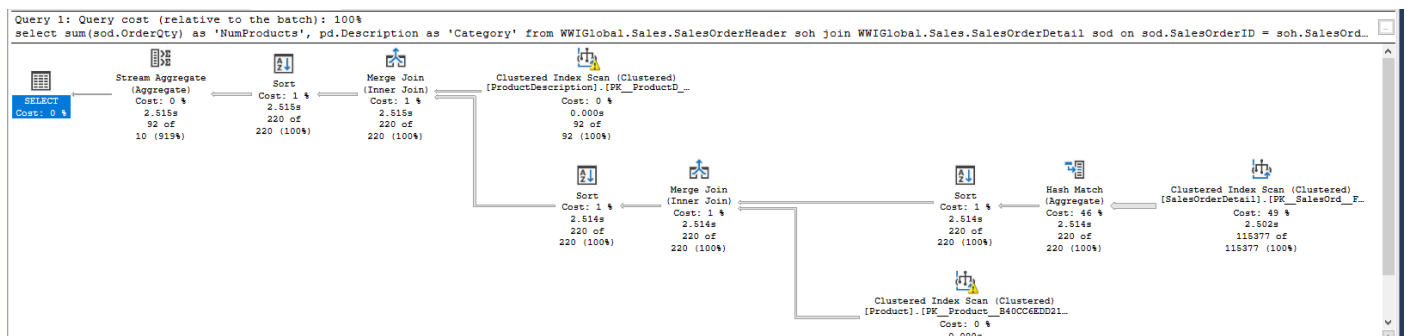
*2ª Fase Relatório Técnico – Complementos de Bases de Dados*



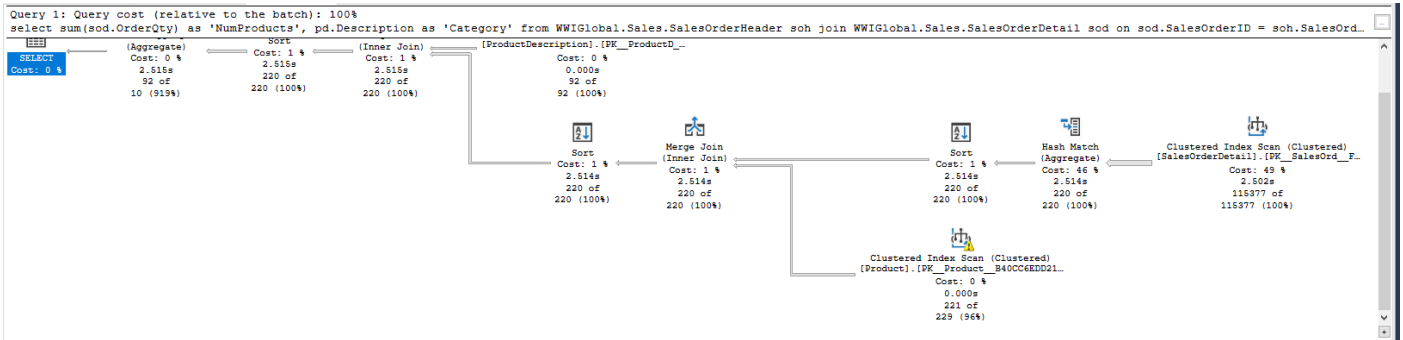
Query 2:



Query 3:



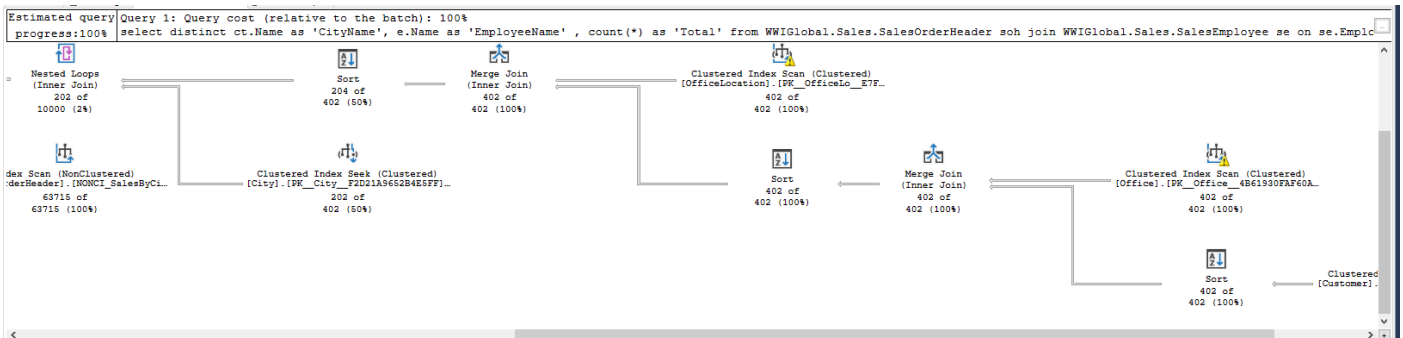
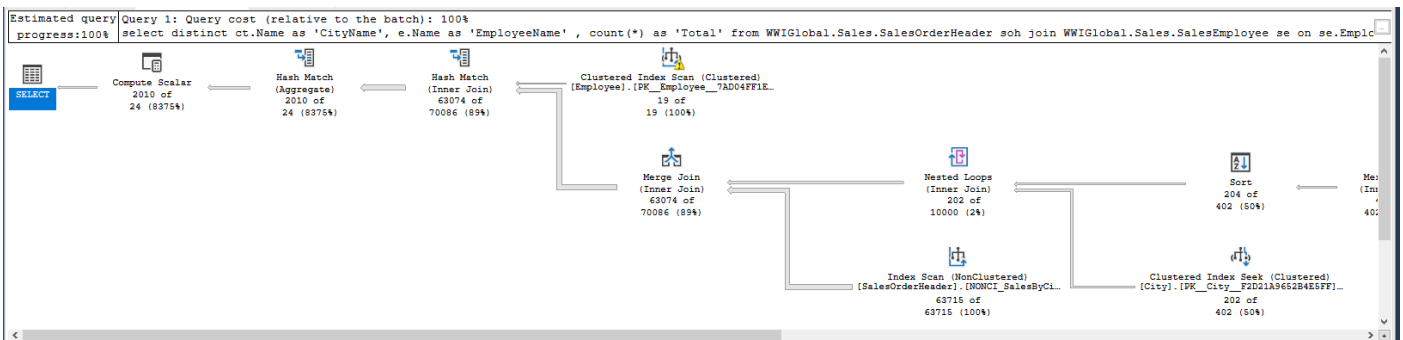
## 2ª Fase Relatório Técnico – Complementos de Bases de Dados



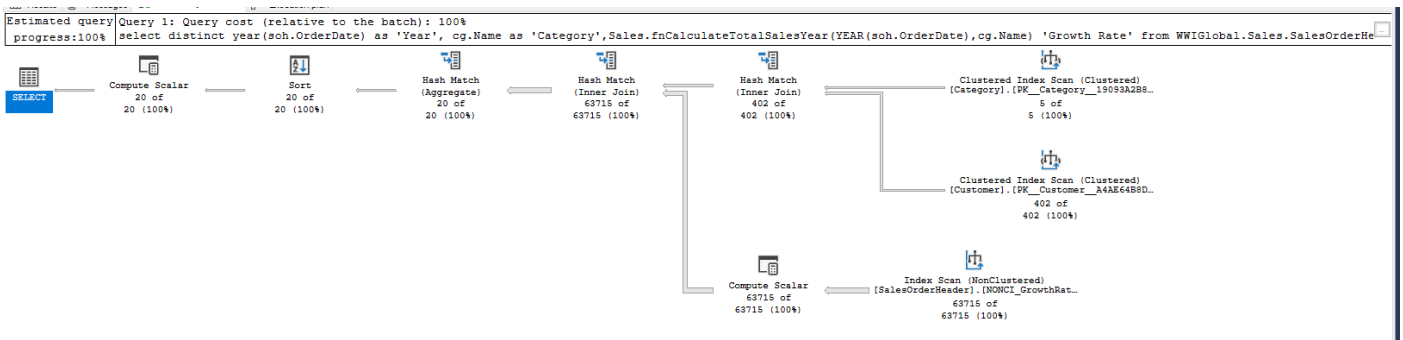
- Sobre a base de dados otimizada (normalizada) com índices:

Estatísticas:

Query 1:



Query 2:

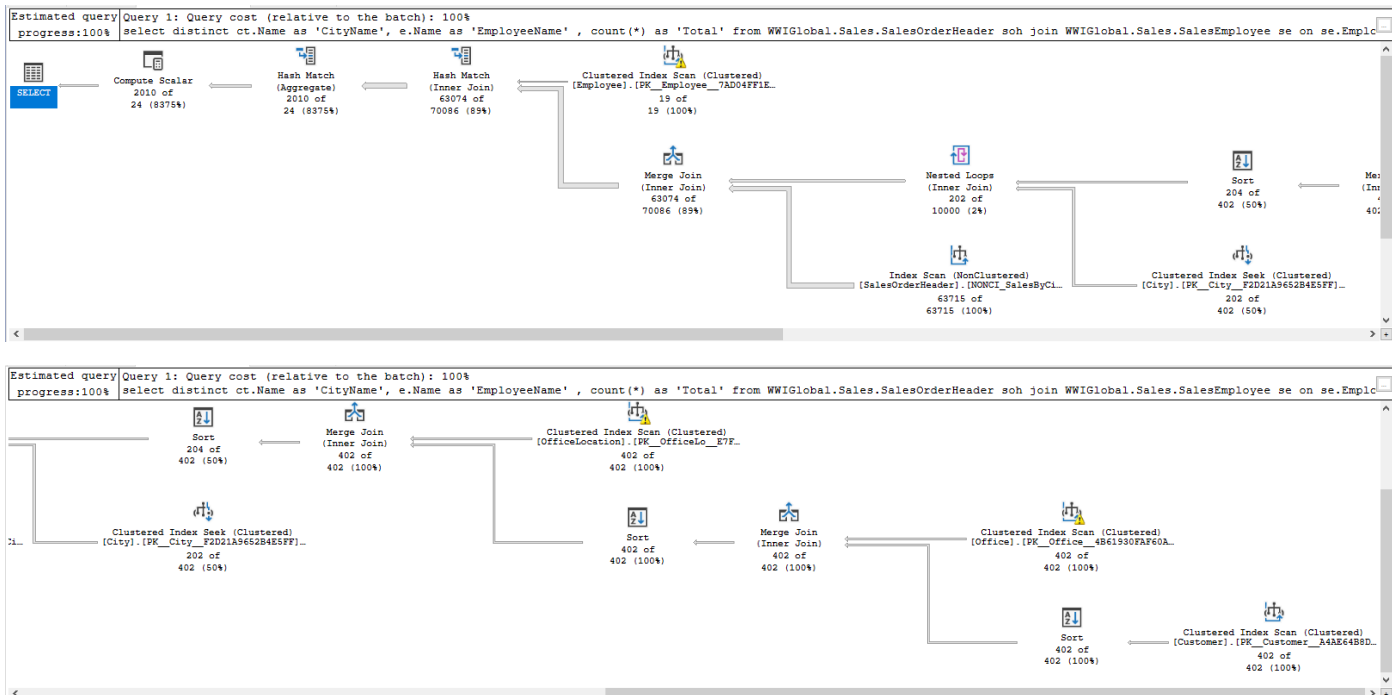


Planos de execução:

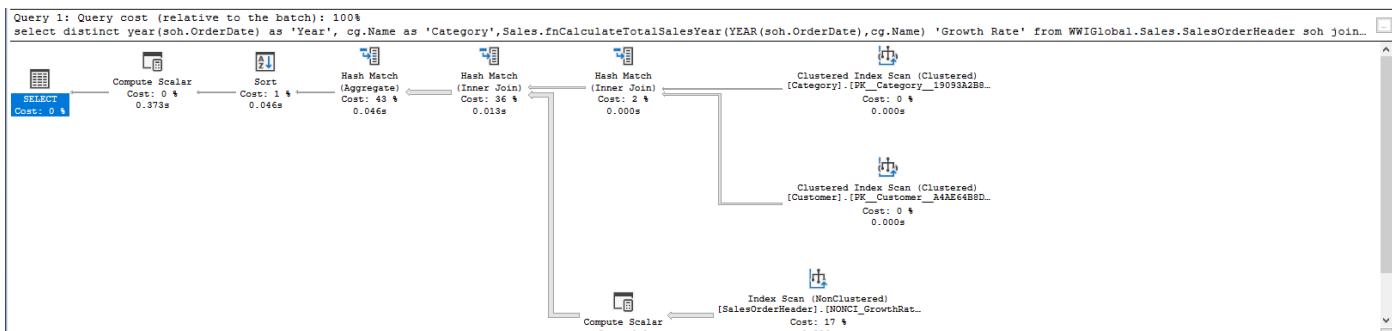


## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

Query 1:



Query 2:



Query 3: Não foi criado índice para esta consulta, uma vez que tinha uma boa performance, então não considerámos necessário fazê-lo.

## 10. Backup e Recuperação

- Modelo de Recuperação:

Uma vez que se trata de uma base de dados de uma empresa de tamanho considerável e que para além disso se trata de uma empresa que vende para o mundo inteiro, é de grande importância ter uma proteção elevada contra a perda de informação, daí o modelo de recuperação escolhido ser o Full.

- Sequência dos backups e justificação:

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

*Full Backup -> Log Backup\_01 -> Backup Diferencial\_1 -> Log Backup1\_1 -> Log Backup1\_2 -> Log Backup1\_3 -> Backup Diferencial\_2 -> Log Backup2\_1 -> Log Backup2\_2 -> Log Backup2\_3*

- Full Backup todos os dias à noite, quando o sistema tem menor carga;
- Primeiro Log Backup do dia 1 hora após o Full Backup;
- Primeiro Backup Diferencial passadas 3 horas do primeiro Log Backup;
- 3 Log Backups após o primeiro Backup Diferencial, com 1 hora de intervalo entre cada um deles;
- A partir deste ponto o seguinte ciclo mantém-se, até chegar a hora onde acontece o Full Backup: Backup Diferencial passado 1 hora da sequência de 3 Log Backups com 1 hora de intervalo entre eles;
- É usado este intervalo de 1 hora para prevenir que a perda de informação não seja superior a 1 hora.
- Cenário de ocorrência de “crash” da base de dados e sequência de recuperação - recuperação até ao momento da falha (ficheiro de dados corrompido):

*Componentes necessários para executar a recuperação: Backup das últimas (tail) atualizações no ficheiro de log de transações, backup completo e sequência de backups de log de transações, desde o último backup completo.*

*Passos a seguir:*

1. Fazer o backup do tail do transaction log;
  2. Recuperar a partir do backup completo;
  3. Recuperar os logs das transações (por exemplo dois logs);
  4. Recuperar o tail dos logs das transações.
- Otimizações do plano de backups:
- Por fim, de forma a otimizar o plano de backups, as tabelas de base deveriam ser colocadas em filegroups read-only. Desta forma poderiam ser feitos backups parciais, que não iriam guardar a informação contida das mesmas, visto que não são sujeitas a alterações frequentes, e poderia também posteriormente ser feito um backup diferencial e parcial quando necessário, com o objetivo de guardar esta informação.*

## 11. Segurança e Controlo de Acessos

*Definição de Utilizadores, Roles, Schemas e Encriptação.*

### 11.1 Níveis de acesso à informação

*Foram criados os roles:*

- Administrador, com acesso total à base de dados;
- EmployeeSalesPerson, com acesso a insert, update, delete nas tabelas de SalesOrderHeader e Sales OrderDetails, e acesso a select a todas as tabelas do shema sales;
- SalesTerritory, com acesso apenas às views correspondentes às sales apenas no território Rocky Mountain

*Foram também criados os users:*

- AdminWWIGlobal com o role Administrador;

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

- *EmployeeSales* com o role *SalesTerritory*;
- *SalesTerritoryUser* com o role *SalesTerritory*

### 11.2 Encriptação

A coluna das passwords da tabela de *RegisteredUsers* foi encriptada através de hashing, visto que vai ser necessário comparar as passwords para login, e as colunas dos preços da tabela *Product* foi encriptada através de uma chave simétrica.

## 12. Controlo de Concorrência

Os níveis de isolamento adotados para os processos referidos no enunciado foram os seguintes:

- *Repeatable Read* para a adição de produto a uma venda, uma vez que não deve ser possível adicionar um produto à mesma venda em simultâneo;
- *Read Committed* para a atualização do preço de um produto, de forma a que o mesmo não se altere numa venda que ainda não foi finalizada;
- *Snapshot* para o cálculo do preço total de uma venda, de forma a que não possam ser nem adicionados nem removidos novos produtos à mesma;

Outros cenários:

- Atualizar quantidade de um produto numa venda: *Repeatable Read*, uma vez que não deve ser possível atualizar a quantidade de um produto da mesma venda em simultâneo;
- Remover um utilizador: *Serializable*, uma vez que não deve ser possível remover um utilizador que já não exista, por exemplo, daí ser necessário o nível de isolamento mais completo.

## 13. MongoDB

O código implementado em MongoDB foi a criação de 3 collections e importados os dados da Base de Dados *WWIGlobal*:

- *sales*, corresponde à informação sobre as vendas;
- *salesValuePerYear*, corresponde à informação sobre o valor de vendas por ano;
- *salesValuePerMonthPerYear*, corresponde à informação sobre o valor de vendas por cada mês de cada ano.

Foram também implementadas funções para fazer a consulta à nova base de dados:

- *clientSalesHistory*, recebe o ID de cliente e o nome do produto a consultar, e retorna o histórico de vendas desse produto;
- *getTotalProductBought*, recebe o nome do produto, o mês e o ano, e retorna o número de produtos vendidos nesse mês desse ano;
- *getProductAverageMonthlySales*, recebe o nome do produto, e retorna o número médio de vendas;

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

- `getProductBoughtByBrand`, recebe o nome da marca, e retorna os produtos, e a sua quantidade, comprados pela marca.

### 14. Descrição da Demonstração

#### 14.1 Script de demonstração sobre a base de dados relacional

```
-- Test procedure to create a sale
EXEC Sales.spCreateSale '2022-11-30', 1, 3, 4, 4.99;
EXEC Sales.spCreateSale '2022-12-20', 1, 3, 4, 5.99;
EXEC Sales.spCreateSale '2022-12-06', 1, 3, 4, 6.99;
EXEC Sales.spCreateSale '2022-12-14', 1, 3, 4, 7.99;

SELECT *
FROM WWIGlobal.Sales.SalesOrderHeader
WHERE TaxAmount = 4.99;

-- to return to original state: delete this OrderHeader

-- Test procedure to add a product to a sale
EXEC Sales.spAddProductToSale 63716, 1, 1;
EXEC Sales.spAddProductToSale 63716, 1, 215;

EXEC Sales.spAddProductToSale 63717, 1, 220;
EXEC Sales.spAddProductToSale 63717, 1, 2;

EXEC Sales.spAddProductToSale 63718, 1, 215;
EXEC Sales.spAddProductToSale 63718, 1, 220;

EXEC Sales.spAddProductToSale 63719, 1, 1;
EXEC Sales.spAddProductToSale 63719, 1, 2;

SELECT *
FROM WWIGlobal.Sales.SalesOrderDetail
WHERE SalesOrderID = 63716

-- Check for errors
SELECT *
FROM dbo.ErrorLog;

-- to return to original state: delete this OrderDetail

-- Test procedure to update product quantity on a sale
select *
FROM WWIGlobal.Sales.SalesOrderDetail
WHERE SalesOrderDetailID = 115378;

EXEC Sales.spUpdateProductQty 115378, 2;

select *
FROM WWIGlobal.Sales.SalesOrderDetail
WHERE SalesOrderDetailID = 115378;

-- to return to original state: execute the same procedure using 1 instead of 2
```

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
-- Test procedure to remove product
SELECT *
FROM WWIGlobal.Sales.SalesOrderHeader
WHERE SalesOrderID = 63718;

EXEC Sales.spRemoveProduct 115378, 1;

SELECT *
FROM WWIGlobal.Sales.SalesOrderHeader
WHERE SalesOrderID = 63718;

-- Test function to calculate the total price of a sale
SELECT Sales.fnCalcTotalPriceSale(115375) 'Total';

-- Test add Promotion To Product Sale
EXEC Sales.spUpdateProductQty 115380, 7;
EXEC Sales.spUpdateProductQty 115381, 3;
SELECT sd.SalesOrderID, sd.SalesOrderDetailID, p.ProductID, sd.OrderQty, p.UnitPrice, Sales.fnCalcTotalPriceSale(sd.SalesOrderDetailID) SubTotal
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE SalesOrderID = 63718;

-- Create promotions
SELECT sd.SalesOrderID, sd.SalesOrderDetailID, p.ProductID, sd.OrderQty, p.UnitPrice, Sales.fnCalcTotalPriceSale(sd.SalesOrderDetailID) SubTotal
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE SalesOrderID = 63718;

EXEC Sales.spCreatePromotion 215, '50', '50% de Desconto', '2022-11-20', '2022-11-30'
EXEC Sales.spCreatePromotion 220, '100', '100% de Desconto', '2022-11-29', '2022-11-30'

SELECT p.ProductID, pt.PromotionDescription, p.DateStart, p.DateEnd
FROM WWIGlobal.Sales.Promotions p
JOIN WWIGlobal.Sales.PromotionType pt on pt.TypeID=p.PromotionType

SELECT sd.SalesOrderID, sd.SalesOrderDetailID, p.ProductID, sd.OrderQty, p.UnitPrice, Sales.fnCalcTotalPriceSale(sd.SalesOrderDetailID) SubTotal
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE SalesOrderID = 63718; table WWIGlobal.Sales.Product

EXEC Sales.spEditPromotionDate 215, '2022-11-29', '2022-11-30'

SELECT p.ProductID, pt.PromotionDescription, p.DateStart, p.DateEnd
FROM WWIGlobal.Sales.Promotions p
JOIN WWIGlobal.Sales.PromotionType pt on pt.TypeID=p.PromotionType

SELECT sd.SalesOrderID, sd.SalesOrderDetailID, p.ProductID, sd.OrderQty, p.UnitPrice, Sales.fnCalcTotalPriceSale(sd.SalesOrderDetailID) SubTotal
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE SalesOrderID = 63717;
```

```
-- Test user insert
DECLARE @counter INT
SET @counter = 1
while @counter <=10
BEGIN
    DECLARE @email VARCHAR(50)
    SET @email = CONCAT('user',@counter,'@email.com')
    DECLARE @password VARCHAR(50)
    SET @password = NEWID()
    exec Users.spAddUser @email,@counter,@password;
    Set @counter=@counter+1
END

SELECT *
FROM Users.RegisteredUser

EXEC Users.spEditUser 'user1@email.com','customerid','21'
--Recover a password from a user
EXEC Users.recoverPassword 'user1@email.com'

SELECT *
FROM WWIGlobal.Users.RecoverUsers
EXEC Users.spRemoveUser 'user1@email.com'

select *
from Sales.SalesOrderDetail

select Sales.fnCheckIfSaleAllowsChillerProducts(1) 'check'

-- Test accordance of delivery date to lead time days of product
SELECT sd.ProductID, p.LeadTimeDays, sh.OrderDate, sh.DueDate,Sales.fnVerifyDeliveryDate(sd.SalesOrderID) 'Accordance'
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.SalesOrderHeader sh on sh.SalesOrderID=sd.SalesOrderID
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE sd.SalesOrderID=63717

SELECT sd.ProductID, p.LeadTimeDays, sh.OrderDate, sh.DueDate,Sales.fnVerifyDeliveryDate(sd.SalesOrderID) 'Accordance'
FROM WWIGlobal.Sales.SalesOrderDetail sd
JOIN WWIGlobal.Sales.SalesOrderHeader sh on sh.SalesOrderID=sd.SalesOrderID
JOIN WWIGlobal.Sales.Product p on p.ProductID=sd.ProductID
WHERE sd.SalesOrderID=510

-- Executes the metadata sps
EXEC spAutoGenerateMetadata

SELECT *
FROM vTableMetadata v
ORDER BY v.[Table Name],v.[Column Name]

EXEC spAutoGenerateSpaceUsed

SELECT *
FROM vTableSizeOccupied v
ORDER BY v.[Number Of Rows] DESC, v.[Table Name]
```

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
-- Test Generate DML for RegisteredUser
EXEC spGenInserts 'RegisteredUser'
EXEC spGenUpdates 'RegisteredUser'
EXEC spGenDelete 'RegisteredUser'

EXEC registereduser_insert 'newEmail@email.com',250,'newPassword'
SELECT *
FROM Users.RegisteredUser
EXEC registereduser_update 'newEmail@email.com',137,'newNewPassword'
SELECT *
FROM Users.RegisteredUser
EXEC registereduser_delete 'newEmail@email.com'
SELECT *
FROM Users.RegisteredUser
```

### 14.2 Script de demonstração sobre a base de dados NoSQL

The screenshot displays two scripts and their results in a NoSQL database interface.

**Script 1: getTotalProductBought**

```
70 // Listar por Produto o valor total por mês/ano.
71 function getTotalProductBought(product, pMonth, pYear) {
72     return db.sales.aggregate([
73         { $match: { "product.description": product } },
74         {
75             $match: {
76                 $expr: {
77                     $gte: [{ $convert: { input: "$orderDate", to: "date" } }, new Date((String)(pYear) + "-" + (String)(pMonth) + "-01")]
78                 }
79             }
80         },
81         {
82             $match: {
83                 $expr: {
84                     $lt: [{ $convert: { input: "$orderDate", to: "date" } }, new Date((String)(pYear) + "-" + (String)(pMonth + 1) + "-01")]
85                 }
86             }
87         },
88         { $group: { _id: "$product.description", total: { $sum: "$product.price" } } },
89         { $project: { _id: 0, Product: "$_id", Total: "$total" } }
90     ])
91 }
92
93 getTotalProductBought("10mm Anti Static Bubble Wrap", 5, 2016)
94
95
```

**Result 1:**

Product	Total
10mm Anti Static Bubble Wrap	2.500 (2.5K)

**Script 2: clientSalesHistory**

```
57 //Listar por Produto o "histórico de vendas" adquiridos pelo cliente.
58 function clientSalesHistory(customerID, product) {
59     return db.sales.aggregate([
60         { $match: { "customer.id": customerID } },
61         { $match: { "product.description": product } },
62         { $project: { _id: 0, Product: "$product.description", "Aquired Date": "$orderDate" } },
63         { $sort: { "Aquired Date": 1 } }
64     ])
65 }
66
67 clientSalesHistory(200, "10mm Anti Static Bubble Wrap")
68
69
```

**Result 2:**

Product	Aquired Date
10mm Anti Static Bubble Wrap	2013-01-17
10mm Anti Static Bubble Wrap	2013-02-14
10mm Anti Static Bubble Wrap	2013-06-14
10mm Anti Static Bubble Wrap	2013-07-08
10mm Anti Static Bubble Wrap	2013-07-30
10mm Anti Static Bubble Wrap	2014-03-03
10mm Anti Static Bubble Wrap	2014-06-19
10mm Anti Static Bubble Wrap	2015-06-15
10mm Anti Static Bubble Wrap	2015-08-14
10mm Anti Static Bubble Wrap	2015-09-18
10mm Anti Static Bubble Wrap	2016-03-03

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
96 // Listar por Produto a média mensal.
97 function getProductAverageMonthlySales(product) {
98   let numberOfMonths = db.salesValuePerMonthPerYear.find().count()
99   return db.sales.aggregate([
100     { $match: { "product.description": product } },
101     { $group: { _id: "product.description", total: { $sum: 1 } } },
102     { $project: { _id: 0, Product: "$_id", "Avg Sales per Month": { $divide: ["$total", numberOfMonths] } } }
103   ])
104 }
105
106 getProductAverageMonthlySales("10mm Anti Static Bubble Wrap")
107
```

Product	Avg Sales per Month
1 10mm Anti Static Bubble Wrap	40.8293

```
109 // Listar por Marca, os produtos e quantidades adquiridos.
110 function getProductBoughtByBrand(brand) {
111   return db.sales.aggregate([
112     { $match: { "customer.brand": brand } },
113     { $group: { _id: "$product.description", total: { $sum: 1 } } },
114     { $project: { _id: 0, Product: "$_id", Quantity: "$total" } },
115     { $sort: { "Product": 1 } }
116   ])
117 }
118
119 getProductBoughtByBrand("Wingtip Toys")
120
```

Product	Quantity
1 "The Gu" T-Shirt	95
2 10mm Anti Static Bubble Wrap	12
3 10mm Double Sided Bubble Wrap	14
4 20mm Anti Static Bubble Wrap	11
5 20mm Double Sided Bubble Wrap	15
6 32mm Anti Static Bubble Wrap	18
7 32mm Double Sided Bubble Wrap	9
8 Air Cushion Film	10
9 Air Cushion Machine	5
10 Alien Officer Hoodie	13
11 Animal with Big Feet Slippers	18
12 Bubblewrap dispenser	8
13 Chocolate echidnas	1
14 Chocolate frogs	1
15 Clear packing tape	12

## 15. Conclusões

Após ter sido implementada a segunda fase do projeto, concluímos que foram cumpridos na sua maioria os requisitos pedidos nas duas fases, e que relativamente a esta mesma fase as partes onde houve maior dificuldade foram no controlo de concorrência e também na definição das políticas de backup, devido a não existirem certezas absolutas de como implementar ambas.