

**Metadados e layout**

Grupo Nº 6

Alexandre Coelho, Nº 190221093

Sérgio Veríssimo, Nº 190221128

**Licenciatura em Engenharia Informática**

**COMPLEMENTOS de BASES DE Dados**

**laboratório 3**

**Índice**

[Etapa 1 2](#_Toc23118551)

[Etapa 2 3](#_Toc23118552)

[Etapa 3 5](#_Toc23118553)

[Etapa 4 6](#_Toc23118554)

# Etapa 1

**Crie um conjunto de queries para uma determinada tabela, que:**

1. Visualize para todas as colunas, o nome e se contém a restrição NOT NULL;

SELECT COLUMN\_NAME, IS\_NULLABLE

FROM AdventureWorksLT2012.INFORMATION\_SCHEMA.COLUMNS

WHERE TABLE\_NAME = 'Product';

1. Visualize o atributo IDENTITY;

SELECT is\_identity AS 'IDENTITY'

FROM sys.columns sc

JOIN sys.tables st

ON sc.object\_id = st.object\_id

WHERE st.name = 'Product';

1. Visualize a(s) coluna(s) que constituem a chave primária;

SELECT pk.name AS pk\_name,

ic.index\_column\_id AS column\_id,

col.name AS column\_name

FROM sys.tables tab

INNER JOIN sys.indexes pk

ON tab.object\_id = pk.object\_id

AND pk.is\_primary\_key = 1

INNER JOIN sys.index\_columns ic

ON ic.object\_id = pk.object\_id

AND ic.index\_id = pk.index\_id

INNER JOIN sys.columns col

ON pk.object\_id = col.object\_id

AND col.column\_id = ic.column\_id

WHERE tab.name = 'Product';

1. Visualize para as chaves estrangeiras, o nome da coluna e a tabela/coluna que é referenciada;

SELECT

tab1.name AS NomeDaTabela,

col1.name AS NomeDaColuna,

tab2.name AS TabelaReferenciada,

col2.name AS ColunaReferenciada

FROM sys.foreign\_key\_columns fkc

JOIN sys.tables tab1

ON tab1.object\_id = fkc.parent\_object\_id

JOIN sys.columns col1

ON col1.column\_id = parent\_column\_id AND col1.object\_id = tab1.object\_id

JOIN sys.tables tab2

ON tab2.object\_id = fkc.referenced\_object\_id

JOIN sys.columns col2

ON col2.column\_id = referenced\_column\_id AND col2.object\_id = tab2.object\_id;

# Etapa 2

**Faça um stored procedure que recebe como argumento o nome de uma tabela (@tabela), e cria outro stored procedure (“sp\_Ins” + @tabela”) que permite inserir um registo na tabela (@tabela).**

**Altere o stored procedure para:**

1. Retirar do comando insert a coluna IDENTITY;
2. Validar os valores das colunas NOT NULL;
3. Validar os valores da chave primária;
4. Validar os valores das colunas que contêm chaves estrangeiras.

CREATE OR ALTER PROCEDURE [SalesLT].etapa2

@table NVARCHAR(128)

AS

BEGIN

DECLARE @table\_columns\_type NVARCHAR(MAX)

DECLARE @table\_columns NVARCHAR(MAX)

DECLARE @table\_columns\_values NVARCHAR(MAX)

DECLARE @null\_validation NVARCHAR(MAX)

DECLARE @column\_name NVARCHAR(128)

DECLARE @data\_type NVARCHAR(128)

DECLARE @data\_length int

DECLARE columns\_cursor CURSOR LOCAL FOR

SELECT isc.[COLUMN\_NAME],

isc.[DATA\_TYPE],

isc.[CHARACTER\_MAXIMUM\_LENGTH]

FROM [INFORMATION\_SCHEMA].[COLUMNS] isc

WHERE isc.[TABLE\_NAME] LIKE @table

OPEN columns\_cursor

FETCH NEXT FROM columns\_cursor INTO @column\_name, @data\_type, @data\_length

WHILE @@FETCH\_STATUS = 0

BEGIN

--ONLY ADD THE PARAMETER IF COLUMN ISN'T IDENTITY

IF (COLUMNPROPERTY(OBJECT\_ID('SalesLT.' + @table), @column\_name, 'IsIdentity') = 0)

BEGIN

--CHECK IF A NULL VALUE IS ALLOWED

IF (@data\_length IS NULL OR @data\_length = -1)

BEGIN

SET @table\_columns\_type = CONCAT(@table\_columns\_type, '@' + @column\_name + ' ' + @data\_type + ',');

END

ELSE

BEGIN

SET @table\_columns\_type = CONCAT(@table\_columns\_type, '@' + @column\_name + ' ' + @data\_type + '(' + CONVERT(nvarchar,@data\_length) + '),');

END

SET @table\_columns = CONCAT(@table\_columns, @column\_name + ',');

SET @table\_columns\_values = CONCAT(@table\_columns\_values, '@' + @column\_name + ',');

IF (COLUMNPROPERTY(OBJECT\_ID('SalesLT.' + @table), @column\_name, 'AllowsNull') = 0)

BEGIN

SET @null\_validation = CONCAT(@null\_validation, ' IF (@', @column\_name, ' IS NULL) BEGIN RAISERROR (''Column ', @column\_name , ' can not be null'', 16, 1); RETURN; END;');

END

END

FETCH NEXT FROM columns\_cursor INTO @column\_name, @data\_type, @data\_length

END

CLOSE columns\_cursor

DEALLOCATE columns\_cursor

SET @table\_columns\_type = LEFT(@table\_columns\_type, LEN(@table\_columns\_type) - 1);

SET @table\_columns = LEFT(@table\_columns, LEN(@table\_columns) - 1);

SET @table\_columns\_values = LEFT(@table\_columns\_values, LEN(@table\_columns\_values) - 1);

EXEC('CREATE OR ALTER PROCEDURE [SalesLT].sp\_Ins' + @table + '

' + @table\_columns\_type + '

AS

BEGIN

' + @null\_validation + '

INSERT INTO ' + @table + ' (' + @table\_columns + ') VALUES (' + @table\_columns\_values + ');

END');

END

GO

EXEC [SalesLT].etapa2 'ProductCategory';

# Etapa 3

**Com base nos metadados, faça um stored procedure que preencha a tabela de estatísticas que foi criada no laboratório 1.**

CREATE PROCEDURE [SalesLT].[spEstatisticas]

AS

BEGIN

INSERT INTO [SalesLT].[Estatisticas]

SELECT SCHEMA\_NAME([schema\_id]) AS [schema],

[t].[Name] AS [table],

SUM([p].[rows]) AS [total\_row\_count]

FROM [sys].[tables] AS [t]

INNER JOIN [sys].[partitions] AS [p]

ON [p].[object\_id] = [t].[object\_id]

AND [p].[index\_id] IN (0, 1)

GROUP BY SCHEMA\_NAME([schema\_id]), [t].[name];

END

GO

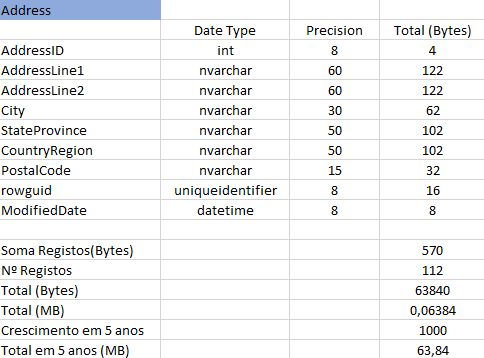
# Etapa 4

**Com base na tabela de estatísticas preenchida na etapa anterior, planeie o layout da nova base de dados AdventureWorksCBD. Admita que o número de novos registos é 1000 vezes maior.**

1. Defina os ficheiros e respetivos filegroups, e o tamanho inicial e final (ao fim de 5 anos);

FILEGROUP 1 - Customer\_Address

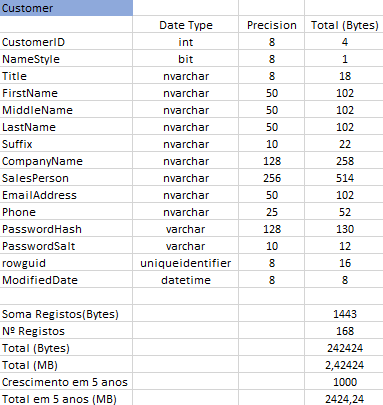
Ficheiro: address.ndf



Tamanho inicial: 0,06384 MB

Tamanho final: 63,84 MB

Ficheiro: customer.ndf

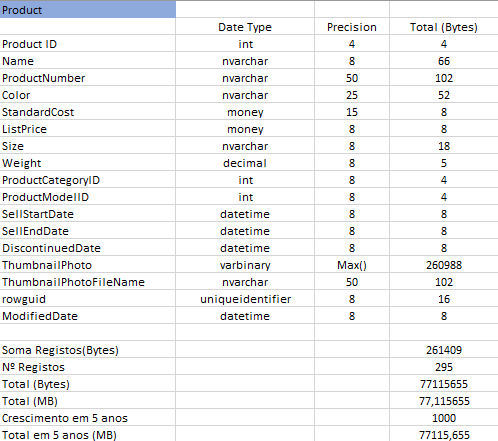


Tamanho inicial: 2,42 MB

Tamanho final: 2424,24 MB

FILEGROUP 2 - Products

Ficheiro: products.ndf



Tamanho inicial: 77,115655 MB

Tamanho final: 77115,655 MB

O valor declarado no excel para o campo varbinary presente (ThumbnailPhoto), que contem precisão máxima, foi feito a partir do tamanho realmente utilizado de cada campo (aplicando apenas uma média de utilização), numa amostra de 100 registos. Ou seja, o somatório dos valores em bytes de 100 registos (tamanho do registo + 2). Foi observada através da seguinte query:

select sum(SizeInBytes) from(

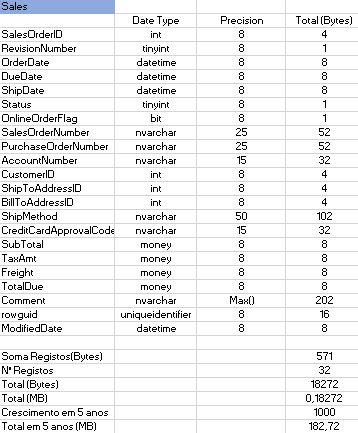
select Size + 2 as SizeInBytes, rowNumber from(

select DATALENGTH(ThumbNailPhoto) as Size, ROW\_NUMBER() over(order by ProductId) as rowNumber

from SalesLT.Product) as teste where rowNumber <= 100) as Counter;

FILEGROUP 3 - Sales

Ficheiro: sales.ndf



Tamanho inicial: 0,18272 MB

Tamanho final: 182,72 MB

Quanto ao campo “Comment”, não existe precisão exactamente definida (encontra-se como max()), em vez de assumirmos o valor exponencial de 2^30 (valor aplicado quando existe um nvarchar com precisão máxima), tentariamos fazer por taxa de ocupação (como feito anteriormente), mas todos os registos dessa tabela, encontram-se preenchidos a NULL. Logo, assumimos um valor de precisão de 100 (em bytes 100\*2+2= 202), que é um valor apropriado para um campo de comentários.

1. Implemente o script SQL para criar a base de dados de acordo com o layout planeado.

CREATE DATABASE [AdventureWorksCBD]

ON PRIMARY (

NAME = [AdventureWorksCBD],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AdventureWorksCBD.mdf',

SIZE = 3MB,

FILEGROWTH = 1MB

),

FILEGROUP [Products] (

NAME = [products],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\products.ndf,',

SIZE = 77,115655 MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 77115,655 MB

),

FILEGROUP [Sales] (

NAME = [sales],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\sales.ndf',

SIZE = 0,18272 MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 182,72 MB

),

FILEGROUP [Customer\_Address] (

NAME = [customers],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\customers.ndf',

SIZE = 2,42 MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 2424,24 MB

), (

NAME = [address],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\address.ndf',

SIZE = 0,06384 MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 63,84 MB

)

LOG ON (

NAME = [AdventureWorksCBD\_LOG],

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\AdventureWorksCBD.ldf',

SIZE = 2MB,

FILEGROWTH = 10%

)

**Admita que a taxa de alteração da informação relativa aos produtos é muito reduzida, mas o número de pesquisas sobre ela é muito elevado.**

1. Altere o planeamento anterior utilizando a solução RAID que melhor se adequa a cada um dos filegroups identificados.

FILEGROUP [Products]: RAID 5

FILEGROUP [Sales]: RAID 10

FILEGROUP [Customer\_Address]: RAID 10