--Calcular por produto o volume de vendas total e o volume de vendas efetuado em

promoção;

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2019-01-01', '2019-01-01';

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2018-01-01', '2018-01-01';

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2017-01-01', '2017-01-01';

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2016-01-01', '2016-01-01';

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2015-01-01', '2015-01-01';

EXEC sp\_PopularPromotions '10%','10% off on all purchases','2014-01-01', '2014-01-01';

GO

--Calcular por produto o volume de vendas total

CREATE OR ALTER VIEW VolumeVendasTotal AS

select sod.ProductID, sum(OrderQty) as VolumeVenda from Promotions p

inner join SalesOrder so on so.DueDate BETWEEN p.PromotionStartDate and p.PromotionEndDate

inner join SalesOrderDetail sod on so.SalesOrderNumber = sod.SalesOrderNumber

group by sod.ProductID

GO

--volume de vendas efetuado em promoção

CREATE OR ALTER VIEW VolumeVendasPromocao AS

select sod.ProductID, sum(OrderQty) as VolumeVenda from SalesOrderDetail sod GROUP BY sod.ProductID

GO

--Qual percentagem de vendas por produto efetuada com promoção;

CREATE OR ALTER VIEW PorcentagemVendasProduto AS

select sode.ProductID, ((

select sum(sod.OrderQty) from Promotions p

inner join SalesOrder so on so.DueDate BETWEEN p.PromotionStartDate and p.PromotionEndDate

inner join SalesOrderDetail sod on so.SalesOrderNumber = sod.SalesOrderNumber

where sod.ProductID = sode.ProductID

group by sod.ProductID

) / cast(sum(OrderQty) as float)) as '%' from SalesOrderDetail sode GROUP BY sode.ProductID

GO

--Calcular o valor total de vendas anual por Região Geográfica;

CREATE OR ALTER VIEW TotalVendasAnualRegiao AS

select stg.SalesTerritoryGroupName, sum(so.SalesAmount) ValorTotal from Customer cu

inner join Country co on co.CountryCode = cu.CountryCode

inner join SalesTerritoryGroup stg on stg.SalesTerritoryGroupID = co.SalesTerritoryGroupID

inner join SalesOrder so on so.CustomerID = cu.CustomerID

group by stg.SalesTerritoryGroupID, stg.SalesTerritoryGroupName

GO

--Obter para cada ano a Região Geográfica com o maior valor total de vendas;

CREATE OR ALTER FUNCTION fnMaxSalesTerritory (@Year VARCHAR(255))

RETURNS TABLE

AS

RETURN

( select top 1 stg.SalesTerritoryGroupName from Customer cu

inner join Country co on co.CountryCode = cu.CountryCode

inner join SalesTerritoryGroup stg on stg.SalesTerritoryGroupID = co.SalesTerritoryGroupID

inner join SalesOrder so on so.CustomerID = cu.CustomerID where format(so.DueDate, 'yyyy') = @Year

group by stg.SalesTerritoryGroupName order by sum(so.SalesAmount) DESC)

GO

CREATE OR ALTER VIEW MaiorTotalVendas AS

select format(so.DueDate, 'yyyy') as Year, (select \* from fnMaxSalesTerritory(format(so.DueDate, 'yyyy'))) as SalesTerritoryGroup from Customer cu

inner join Country co on co.CountryCode = cu.CountryCode

inner join SalesTerritoryGroup stg on stg.SalesTerritoryGroupID = co.SalesTerritoryGroupID

inner join SalesOrder so on so.CustomerID = cu.CustomerID

group by format(so.DueDate, 'yyyy')

GO

--Prazo médio entre data de encomenda e envio por Região Geográfica, (consideração dos ultimos 2 anos).

CREATE OR ALTER VIEW PrazoMedioEntrega AS

select stg.SalesTerritoryGroupName,

sum(datediff(day, so.OrderDate, so.DueDate)) / count(datediff(day, so.OrderDate, so.DueDate)) as AvgOrderDateDueDate from Customer cu

inner join Country co on co.CountryCode = cu.CountryCode

inner join SalesTerritoryGroup stg on stg.SalesTerritoryGroupID = co.SalesTerritoryGroupID

inner join SalesOrder so on so.CustomerID = cu.CustomerID

group by stg.SalesTerritoryGroupID, stg.SalesTerritoryGroupName

GO

--SELECT \* FROM VolumeVendasTotal

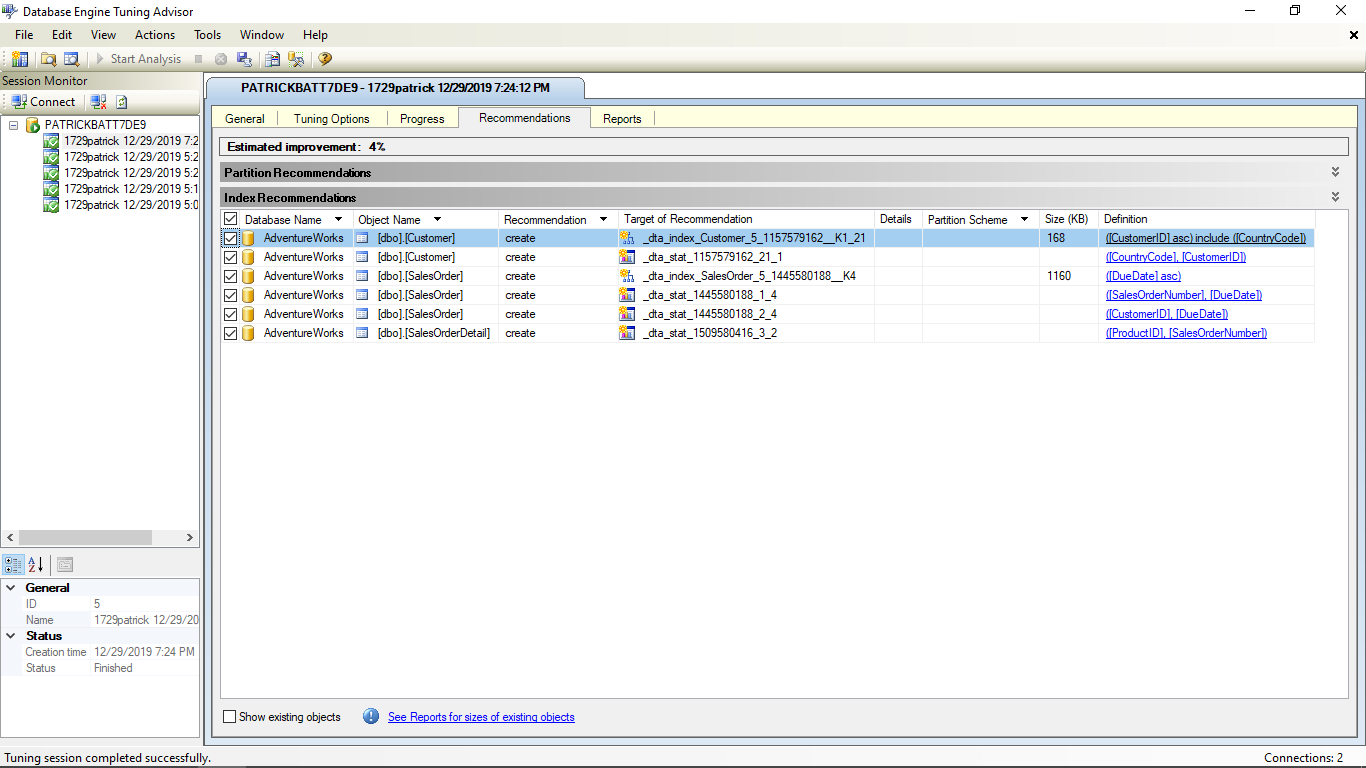
--SELECT \* FROM VolumeVendasPromocao

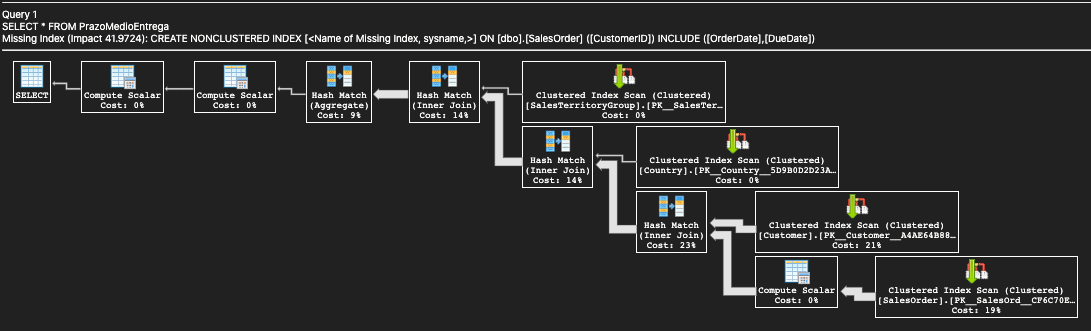
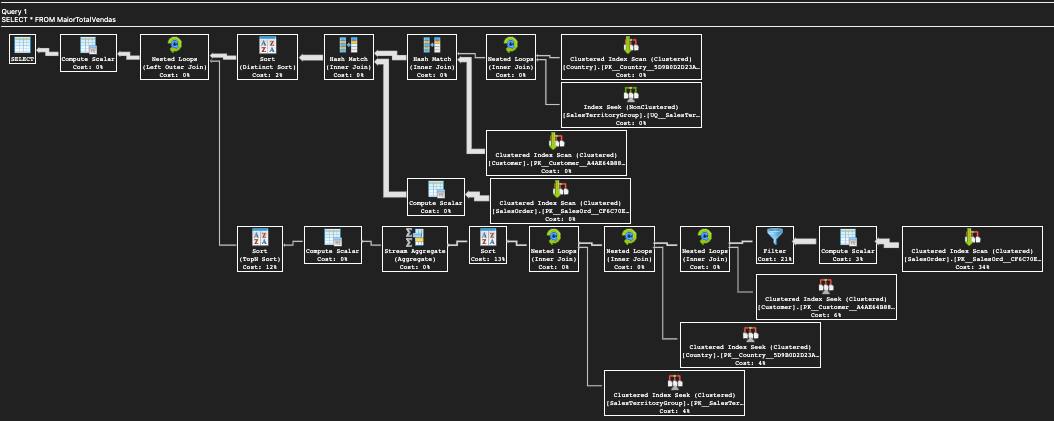
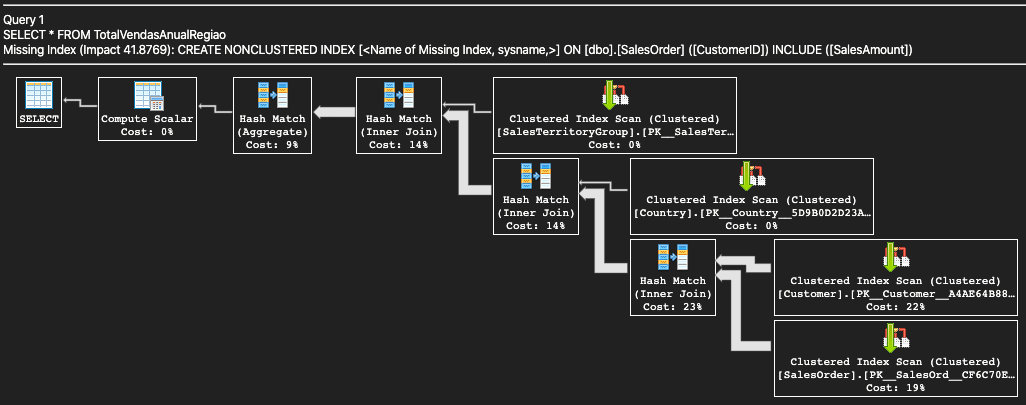
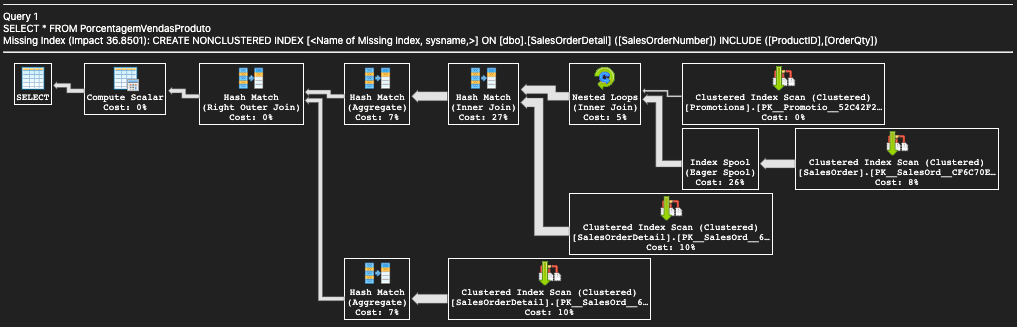
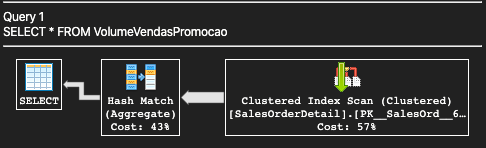
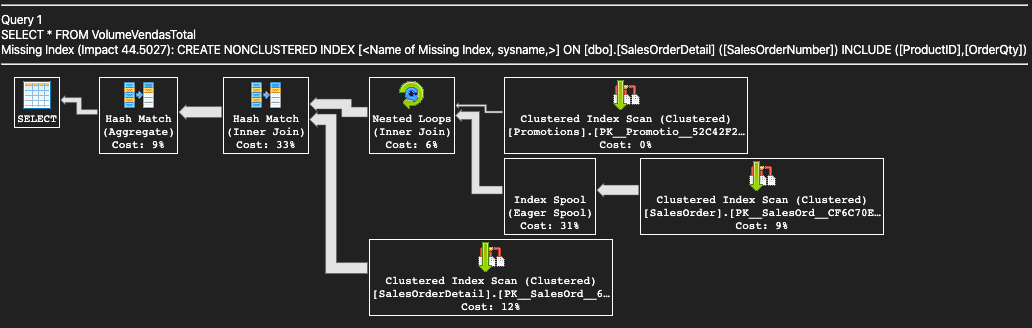
--SELECT \* FROM PorcentagemVendasProduto

--SELECT \* FROM TotalVendasAnualRegiao

--SELECT \* FROM MaiorTotalVendas

--SELECT \* FROM PrazoMedioEntrega



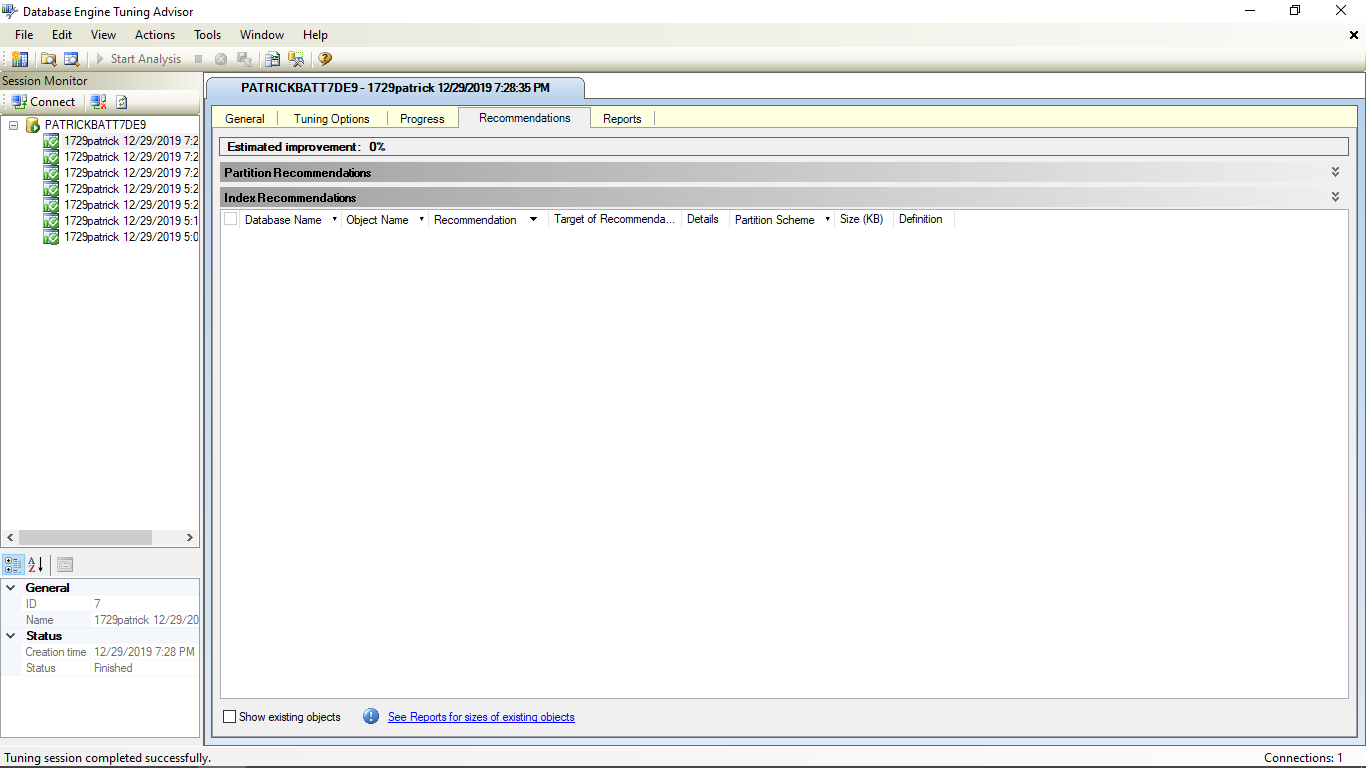


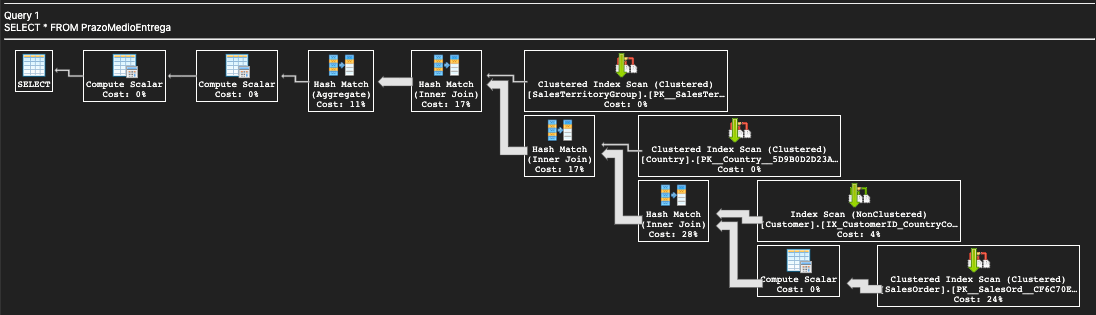
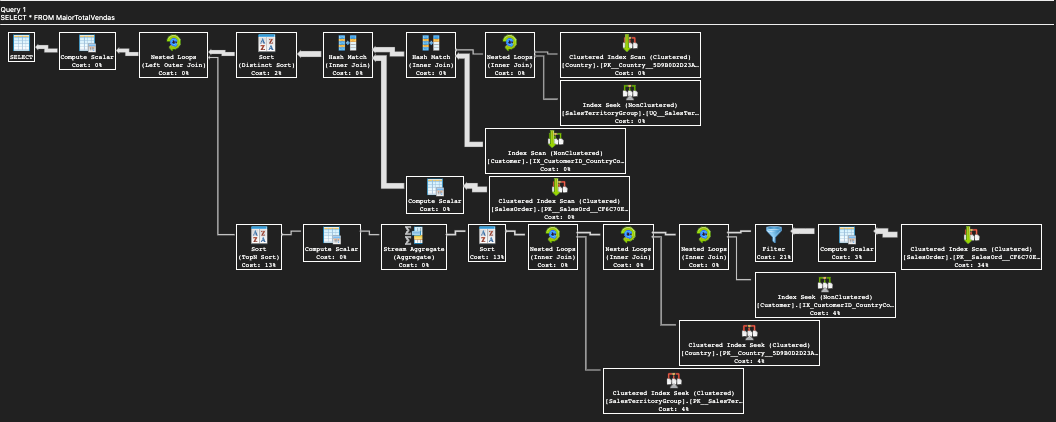
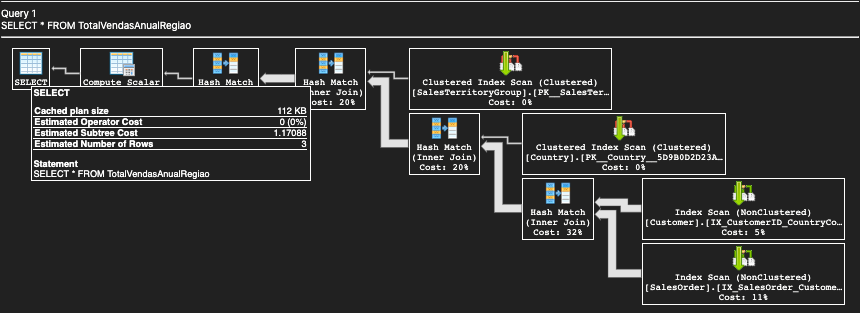
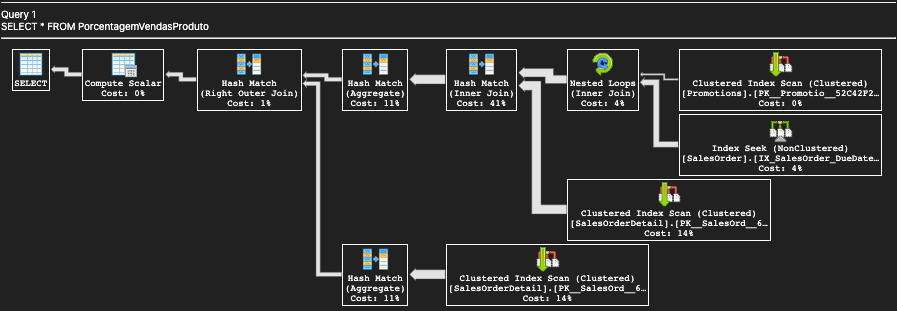
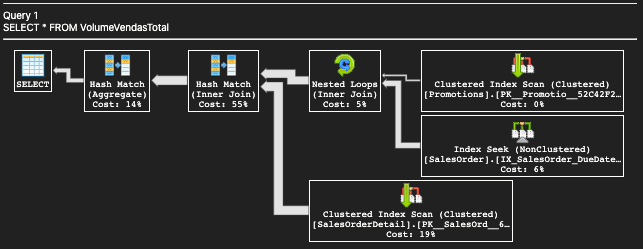
CREATE NONCLUSTERED INDEX IX\_SalesOrderDetail\_\_SalesOrderNumber\_ProductID ON SalesOrderDetail (SalesOrderNumber) INCLUDE (ProductID)

CREATE NONCLUSTERED INDEX IX\_SalesOrder\_CustomerID\_SalesAmount ON SalesOrder (CustomerID) INCLUDE (SalesAmount)

CREATE NONCLUSTERED INDEX IX\_CustomerID\_CountryCode ON Customer (CustomerID) INCLUDE (CountryCode)

CREATE NONCLUSTERED INDEX IX\_SalesOrder\_DueDate ON SalesOrder (DueDate)





Optamos por incluir a colunas em alguns índices pelo motivo que desta forma o otimizador de consulta pode localizar todos os valores da coluna no índice, sem necessidade de acessar o disco; dados de tabela ou índice de cluster não são acessados, resultando em menos operações de E / S de disco

Todas as consultas foram transformadas em views, desta forma, os registros dos relatórios serão acessados mais facilmentes pelo Gestor de Vendas. Além de melhorar a confiabilidade na análise de uma consulta para outra.