

Resolução

Análise de dados

Com base na solução implantada responda aos seguintes questionamentos:

1. Escreva uma query que retorne a quantidade de linhas na tabela Sales.SalesOrderDetail pelo campo SalesOrderID, desde que tenham pelo menos três linhas de detalhes.

```
1 SELECT SalesOrderID, COUNT(*) AS QTD_LINHAS FROM dbo.tb_salesOrderDetail
2 GROUP BY SalesOrderID
```

Resultados Mensagens

Pesquisar para filtrar itens...

SalesOrderID	QTD_LINHAS
43659	12
43660	2
43661	15
43662	22
43663	1

✓ Êxito na consulta | 4s

2. Escreva uma query que ligue as tabelas Sales.SalesOrderDetail, Sales.SpecialOfferProduct e Production.Product e retorne os 3 produtos (Name) mais vendidos (pela soma de OrderQty), agrupados pelo número de dias para manufatura (DaysToManufacture).

```
1 SELECT TOP(3) pp.Name, pp.DaysToManufacture, sum(cast(sod.OrderQty as int)) as total_vendidos
2 FROM dbo.tb_salesOrderDetail sod
3 INNER JOIN dbo.tb_specialOfferProduct sop
4 ON sod.SpecialOfferID = sop.SpecialOfferID
5 INNER JOIN dbo.tb_productionProduct pp
6 ON sop.ProductID = pp.ProductID
7 group by pp.DaysToManufacture, pp.Name
8 order by total_vendidos desc
```

Resultados Mensagens

Pesquisar para filtrar itens...

Name	DaysToManufacture	total_vendidos
Full-Finger Gloves, L	0	270244
Women's Mountain Shorts, L	0	270244
Bike Wash - Dissolver	0	270159

3. Escreva uma query ligando as tabelas Person.Person, Sales.Customer e Sales.SalesOrderHeader de forma a obter uma lista de nomes de clientes e uma contagem de pedidos efetuados.

```
1 SELECT CONCAT(p.FirstName, ' ', p.MiddleName, ' ', p.LastName) AS FullName,
2 COUNT(soh.SalesOrderID) AS PedidosEfetuados
3 FROM [dbo].[tb_person] p
4 INNER JOIN [dbo].[tb_salesCustomer] sc
5 ON p.BusinessEntityID = sc.PersonID
6 INNER JOIN [dbo].[tb_salesOrderHeader] soh
7 ON soh.CustomerID = sc.CustomerID
8 GROUP BY CONCAT(p.FirstName, ' ', p.MiddleName, ' ', p.LastName)
```

Resultados Mensagens

Pesquisar para filtrar itens...

FullName	PedidosEfetuados
Ana NULL Henderson	1
Natalie C Young	1
Martin NULL Raman	2
Philip F Romero	2
Isabella C Bennett	2

Exito na consulta | 7s

4. Escreva uma query usando as tabelas Sales.SalesOrderHeader, Sales.SalesOrderDetail e Production.Product, de forma a obter a soma total de produtos (OrderQty) por ProductID e OrderDate.

```
1 SELECT pp.ProductID, soh.OrderDate, sum(cast(sod.OrderQty as int)) as total vendidos
2 FROM [dbo].[tb_salesOrderHeader] soh
3 INNER JOIN [dbo].[tb_salesOrderDetail] sod
4 ON sod.SalesOrderID = soh.SalesOrderID
5 INNER JOIN dbo.tb_specialOfferProduct sop
6 ON sod.SpecialOfferID = sop.SpecialOfferID
7 INNER JOIN [dbo].[tb_productionProduct] pp
8 ON sop.ProductID = pp.ProductID
9 GROUP BY pp.ProductID, soh.OrderDate
```

Resultados Mensagens


Failed to execute query. Error: A severe error occurred on the current command. The results, if any, should be discarded. Operation cancelled by user.

A consulta falhou

5. Escreva uma query mostrando os campos SalesOrderID, OrderDate e TotalDue da tabela Sales.SalesOrderHeader. Obtenha apenas as linhas onde a ordem tenha sido feita durante o mês de setembro/2011 e o total devido esteja acima de 1.000. Ordene pelo total devido decrescente.

```
1 SELECT SalesOrderID, OrderDate, TotalDue
2 FROM dbo.tb_salesOrderHeader
3 WHERE CAST(OrderDate AS DATE) BETWEEN '2011-09-01' AND '2011-09-30'
4 AND CAST(REPLACE(TotalDue, ',', '.') AS FLOAT) > 1000
5 ORDER BY CAST(REPLACE(TotalDue, ',', '.') AS FLOAT) DESC
```

Resultados Mensagens

 Pesquisar para filtrar itens...

SalesOrderID	OrderDate	TotalDue
44326	2011-09-01 00:00:00.000	3953,9884
44327	2011-09-02 00:00:00.000	3953,9884
44328	2011-09-02 00:00:00.000	3953,9884
44329	2011-09-02 00:00:00.000	3953,9884
44330	2011-09-02 00:00:00.000	3953,9884

Exito na consulta | 0s