Create Table tblProducts  
(  
 [Id] int identity primary key,  
 [Name] nvarchar(50),  
 [Description] nvarchar(250)  
)  
  
Create Table tblProductSales  
(  
 Id int primary key identity,  
 ProductId int foreign key references tblProducts(Id),  
 UnitPrice int,  
 QuantitySold int  
)

Insert into tblProducts values ('TV', '52 inch black color LCD TV')  
Insert into tblProducts values ('Laptop', 'Very thin black color acer laptop')  
Insert into tblProducts values ('Desktop', 'HP high performance desktop')  
  
Insert into tblProductSales values(3, 450, 5)  
Insert into tblProductSales values(2, 250, 7)  
Insert into tblProductSales values(3, 450, 4)  
Insert into tblProductSales values(3, 450, 9)  
  
**Write a query to retrieve products that are not at all sold?**  
This can be very easily achieved using subquery as shown below. Select [Id], [Name], [Description]  
from tblProducts  
where Id not in (Select Distinct ProductId from tblProductSales)  
  
**Most of the times subqueries can be very easily replaced with joins.** The above query is rewritten using joins and produces the same results. Select tblProducts.[Id], [Name], [Description]  
from tblProducts  
left join tblProductSales  
on tblProducts.Id = tblProductSales.ProductId  
where tblProductSales.ProductId IS NULL  
  
**In this example, we have seen how to use a subquery in the where clause.**  
  
**Let us now discuss about using a sub query in the SELECT clause.**Write a query to retrieve the NAME and TOTALQUANTITY sold, using a subquery.Select [Name], (Select SUM(QuantitySold) from tblProductSales where ProductId = tblProducts.Id) as TotalQuantity  
from tblProducts  
order by Name  
  
**Query with an equivalent join that produces the same result.**  
Select [Name], SUM(QuantitySold) as TotalQuantity  
from tblProducts  
left join tblProductSales  
on tblProducts.Id = tblProductSales.ProductId  
group by [Name]  
order by Name  
  
**From these examples,** it should be very clear that, a subquery is simply a select statement, that returns a single value and can be nested inside a SELECT, UPDATE, INSERT, or DELETE statement.   
  
It is also possible to nest a subquery inside another subquery.  
  
According to MSDN, subqueries can be nested upto 32 levels.  
  
Subqueries are always encolsed in paranthesis and are also called as inner queries, and the query containing the subquery is called as outer query.  
  
The columns from a table that is present only inside a subquery, cannot be used in the SELECT list of the outer query.