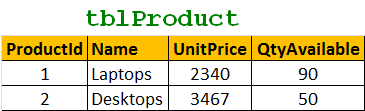
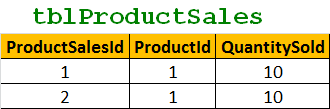
**Suggested SQL Server Videos**  
[Part 57 - Transactions in SQL Server](http://csharp-video-tutorials.blogspot.com/2012/10/transactions-in-sql-server-part-57.html)   
  
A transaction is a group of database commands that are treated as a single unit. A successful transaction must pass the "ACID" test, that is, it must be   
A - Atomic   
C - Consistent   
I - Isolated   
D - Durable   
  
**Atomic** - All statements in the transaction either completed successfully or they were all rolled back. The task that the set of operations represents is either accomplished or not, but in any case not left half-done. For example, in the **spUpdateInventory\_and\_Sell**stored procedure, both the UPDATE statements, should succeed. If one UPDATE statement succeeds and the other UPDATE statement fails, the database should undo the change made by the first UPDATE statement, by rolling it back. In short, the transaction should be ATOMIC.  
  
   
  
   
  
Create Procedure spUpdateInventory\_and\_Sell  
as  
Begin  
  Begin Try  
    Begin Transaction  
      Update tblProduct set QtyAvailable = (QtyAvailable - 10)  
      where ProductId = 1  
  
      Insert into tblProductSales values(3, 1, 10)  
    Commit Transaction  
  End Try  
  Begin Catch   
    Rollback Transaction  
  End Catch   
End   
  
   
  
**Consistent** - All data touched by the transaction is left in a **logically consistent state**. For example, if stock available numbers are decremented from **tblProductTable**, then, there has to be a related entry in **tblProductSales** table. The inventory can't just disappear.   
  
**Isolated** - The transaction must affect data without interfering with other concurrent transactions, or being interfered with by them. This prevents transactions from making changes to data based on uncommitted information, for example changes to a record that are subsequently rolled back. **Most databases use locking to maintain transaction isolation**.  
  
**Durable** - Once a change is made, it is permanent. If a system error or power failure occurs before a set of commands is complete, those commands are undone and the data is restored to its original state once the system begins running again.