**What is a Transaction?**  
**A transaction is a group of commands** that change the data stored in a database. A transaction, is treated as a single unit. A transaction ensures that, either all of the commands succeed, or none of them. If one of the commands in the transaction fails, all of the commands fail, and any data that was modified in the database is rolled back. In this way, transactions maintain the integrity of data in a database.  
  
**Transaction processing follows these steps:**  
1. Begin a transaction.  
2. Process database commands.  
3. Check for errors.   
   If errors occurred,   
       rollback the transaction,   
   else,   
       commit the transaction  
  
**Let's understand transaction processing with an example.** For this purpose, let's Create and populate, tblMailingAddress and tblPhysicalAddress tables  
Create Table tblMailingAddress  
(  
   AddressId int NOT NULL primary key,  
   EmployeeNumber int,  
   HouseNumber nvarchar(50),  
   StreetAddress nvarchar(50),  
   City nvarchar(10),  
   PostalCode nvarchar(50)  
)  
  
Insert into tblMailingAddress values (1, 101, '#10', 'King Street', 'Londoon', 'CR27DW')   
  
   
  
Create Table tblPhysicalAddress  
(  
 AddressId int NOT NULL primary key,  
 EmployeeNumber int,  
 HouseNumber nvarchar(50),  
 StreetAddress nvarchar(50),  
 City nvarchar(10),  
 PostalCode nvarchar(50)  
)  
  
Insert into tblPhysicalAddress values (1, 101, '#10', 'King Street', 'Londoon', 'CR27DW')  
  
**An employee with EmployeeNumber 101**, has the same address as his physical and mailing address. His city name is mis-spelled as **Londoon** instead of **London**. The following stored procedure **'spUpdateAddress'**, updates the physical and mailing addresses. Both the UPDATE statements are wrapped between **BEGIN TRANSACTION** and **COMMIT TRANSACTION** block, which in turn is wrapped between **BEGIN TRY** and **END TRY** block.

So, if both the UPDATE statements succeed, without any errors, then the transaction is committed. If there are errors, then the control is immediately transferred to the catch block. The **ROLLBACK TRANSACTION** statement, in the CATCH block, rolls back the transaction, and any data that was written to the database by the commands is backed out.  
  
Create Procedure spUpdateAddress  
as  
Begin  
 Begin Try  
 Begin Transaction  
 Update tblMailingAddress set City = 'LONDON'   
 where AddressId = 1 and EmployeeNumber = 101  
   
 Update tblPhysicalAddress set City = 'LONDON'   
 where AddressId = 1 and EmployeeNumber = 101  
 Commit Transaction  
 End Try  
 Begin Catch  
 Rollback Transaction  
 End Catch  
End   
  
   
  
**Let's now make the second UPDATE statement, fail**. CITY column length in tblPhysicalAddress table is 10. The second UPDATE statement fails, because the value for CITY column is more than 10 characters.  
Alter Procedure spUpdateAddress  
as  
Begin  
 Begin Try  
 Begin Transaction  
 Update tblMailingAddress set City = 'LONDON12'   
 where AddressId = 1 and EmployeeNumber = 101  
   
 Update tblPhysicalAddress set City = 'LONDON LONDON'   
 where AddressId = 1 and EmployeeNumber = 101  
 Commit Transaction  
 End Try  
 Begin Catch  
 Rollback Transaction  
 End Catch  
End  
  
**Now, if we execute spUpdateAddress**, the first **UPDATE** statements succeeds, but the second **UPDATE** statement fails. As, soon as the second UPDATE statement fails, the control is immediately transferred to the CATCH block. The CATCH block rolls the transaction back. So, the change made by the first UPDATE statement is undone.