TRANSACTION INTEGRITY

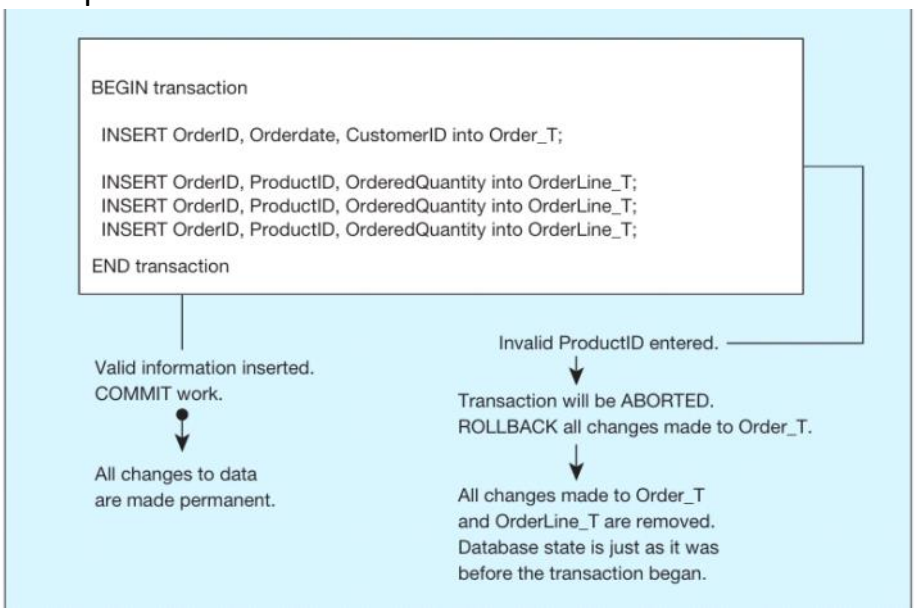
A business transaction is a sequence of steps that constitute some well-defined busi ness activity.

- Examples of business transactions are Admit Patient in a hospital and Enter Customer Order in a manufacturing company.
- Normally, a business transaction requires several actions against the database.
    - For example, consider the transaction Enter Customer Order. When a new customer order is entered, an application program might perform the following steps:
        1. Input the order data (keyed by the user).
        2. Read the CUSTOMER record (or ask user to key it in if a new customer).
        3. Accept or reject the order. If Balance Due plus Order Amount does not exceed Credit Limit, accept the order; otherwise, reject it.
        4. If the order is accepted, increase Balance Due by Order Amount. Store the updated (or new) CUSTOMER record. Insert the accepted ORDER record in the database.
- From a database perspective, a transaction is a complete set of closely-related update commands that must all be done (or none of them done) for the database to remain valid.
- When processing transactions, a DBMS must ensure that the transactions have four well-accepted characteristics, called the ACID properties:
    1. ***Atomic*** The transaction cannot be subdivided, and hence it must be processed in its entirety or not at all. Once the whole transaction is processed, we say that the changes are *committed*. If the transaction fails at any midpoint, we say that it has aborted. For example, suppose that the program accepts a new customer order, increases Balance Due, and stores the updated CUSTOMER record. However, suppose that the new ORDER record is not inserted successfully (perhaps due to a duplicate Order Number key or insufficient physical file space). In this case, we want none of the parts of the transaction to affect the database.
    2. ***Consistent*** Any database constraints that must be true before the transaction must also be true after the transaction. For example, if the inventory on-hand balance must be the difference between total receipts minus total issues, this will be true both before and after an order transaction, which depletes the on-hand balance to satisfy the order.
    3. ***Isolated*** Changes to the database are not revealed to users until the transaction is committed. For example, this property means that other users do not know what the on-hand inventory is until an inventory transaction is complete; this property then usually means that other users are prohibited from simultaneously updating and possibly even reading data that are in the process of being updated. We discuss this topic in more detail later in the discussion of concurrency controls and locking. A consequence of transactions being isolated from one another is that concurrent transactions (i.e., several transactions in some partial state of completion) all affect the database as if they were presented to the DBMS in serial fashion.
    4. ***Durable*** Changes are permanent. Thus, once a transaction is committed, no subsequent failure of the database can reverse the effect of the transaction.
- To maintain transaction integrity, DBMS provide facilities for the user or application program to define transaction boundaries (i.e., the logical beginning and end of a transaction), to commit the work of a transaction as a permanent change to the database, and to abort a transaction on purpose

and correctly if necessary.

- In SQL, the BEGIN TRANSACTION statement is placed in front of the first SQL command within the trans action, and the END TRANSACTION or COMMIT command is placed at the end of the transaction.
  - BEGIN TRANSACTION creates a log file and starts recording all changes (insertions, deletions, and updates) to the database in this file.
  - END TRANSACTION or COMMIT takes the contents of the log file and applies them to the database, thus making the changes permanent, and then empties the log file.
  - Any number of SQL commands may come in between these two commands; these are the database processing steps that perform some well-defined business activity, as explained earlier.
  - If a command such as ROLLBACK is processed after a BEGIN TRANSACTION is executed and before a COMMIT is executed, the DBMS aborts the transaction and undoes the effects of the SQL statements processed so far within the transaction boundaries.

- Example



  - When an order is entered into the Pine Valley database, all of the items ordered should be entered at the same time. Thus, either all OrderLine_T rows from this form are to be entered, along with all the information in Order_T, or none of them should be entered.
  - Here, the business transaction consists of the complete order, not the individual items that are ordered.
  - Alternatively, an application is likely be programmed to execute a ROLLBACK when the DBMS generates an error message per forming an UPDATE or INSERT command in the middle of the transaction.
  - The DBMS thus commits (makes durable) changes for successful transactions (those that reach the COMMIT statement) and effectively

rejects changes from transactions that are aborted (those that encounter a ROLLBACK).

- Some relational DBMSs also have an AUTOCOMMIT (ON/OFF) command that specifies whether changes are made permanent after each data modification command (ON) or only when work is explicitly made permanent (OFF) by the COMMIT com mand. Note that SET AUTOCOMMIT is an interactive command; therefore, a given user session can be dynamically controlled for appropriate integrity measures.

- Suggestions when organizing transactions
    - If possible, collect all user input before executing a database transaction. Also, to minimize the length of a transaction, check for possible errors, such as duplicate keys or insufficient account balance, as early in the transaction as possible so that portions of the database can be released as soon as possible for other users if the transaction is going to be aborted.
        - This is because when you are in the middle of a transaction that is executing other users will have to wait to see if that transaction fails or they move on while that transaction is happening.