Performing Transactions

Transactions are a mechanism that ensures data consistency. Transactions have the following four properties:

1. **Atomicity:** Either a transaction completes or nothing happens at all.
2. **Consistency:** A transaction must start in a consistent state and leave the system in a consistent state.
3. **Isolation:** Intermediate results of a transaction are not visible outside the current transaction.
4. **Durability:** Once a transaction was committed, the effects are persistent, even after a system failure.

The Python DB API 2.0 provides two methods to either *commit* or *rollback* a transaction.

Example

You already know how to implement transactions. Here is again similar example:

```
# Prepare SQL query to DELETE required records

sql = "DELETE FROM EMPLOYEE WHERE AGE > '%d'" % (20)

try:

   # Execute the SQL command

   cursor.execute(sql)

   # Commit your changes in the database

   db.commit()

except:

   # Rollback in case there is any error

   db.rollback()
```

COMMIT Operation

Commit is the operation, which gives a green signal to database to finalize the changes, and after this operation, no change can be reverted back.

Here is a simple example to call **commit** method.

```
db.commit()
```