Python Database Communication (PDBC)

Every application required to store/save data permanently.
This data can be stored permanently using two approaches

1. Files
2. Database

Limitations of file management system

1. Files cannot hold large amount of data
2. Files are not secured because these are managed by OS
3. Files does not provides Query language

Advantage of database

1. It can hold large amount data
2. Data stored inside database is secured, this security is provided by database management system
3. Database provides a Query Language (SQL) for manipulating data

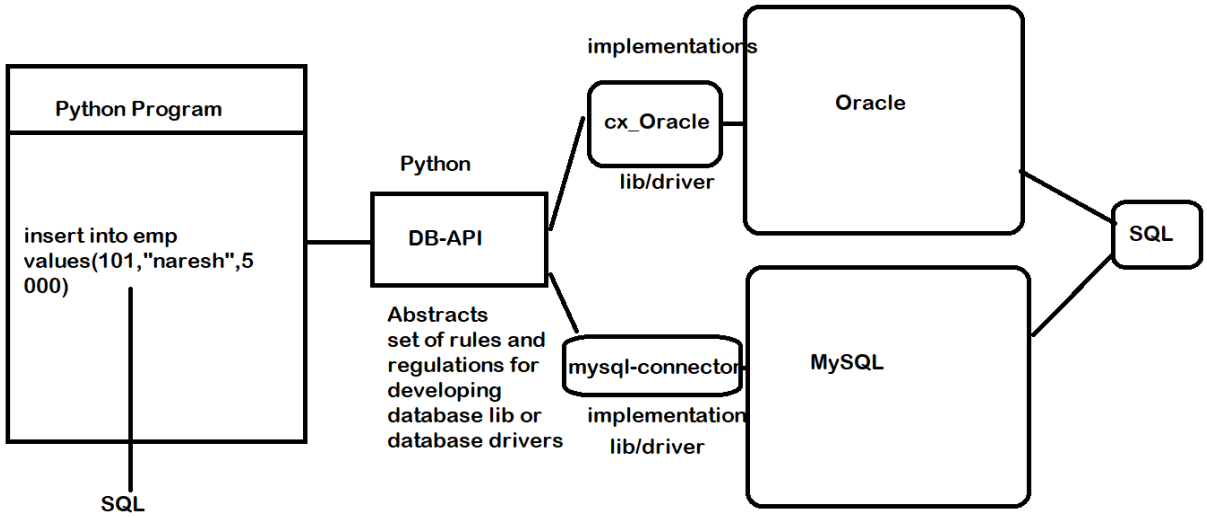
Database Applications or Database Software's

1. Oracle
2. MySQL
3. SQLServer
4. PostgreSQL
5. MongoDB

How Python program communicate with database software or database application?

What is a database API in Python?

DB-API. The Python Database API (DB-API) defines a standard interface for Python database access modules. It's documented in PEP 249. Nearly all Python database modules such as sqlite3 , psycopg , and mysql-python conform to this interface.




```
RBI

class Debitcard(abc.ABC):
  @abc.abstractmethod
   def withdraw(self):
     pass


class HDFCDebitcard(Debitcard):
     def withdraw(self):
         print("withdraw
     of HDFC")


class SBIDebitcard(Debitcard):
     def withdraw(self):
        print("withdraw of SBI")
```

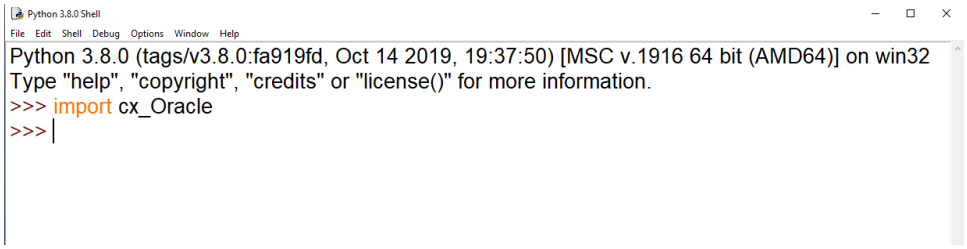
```
Python    (DB-API)

class    Connection(abc.ABC):
  @abc.abstractmethod
   def connect(self):
      pass

  (Oracle)
class OracleConnection( Connection):
      def connect(self):
        logic for
      communicate with
      oracle database
 (Mysql)
 class MySQLconnection(Connection):
      def connect(self):
         logic for communicate with mysql
          database
```

Database vendors provide libraries for communicating with database.

Oracle cx_Oracle
Mysql mysql-connector-python
PostgreSQL psycopg
MongoDB pymongo

Communicating with oracle database

1. **Install oracle database software (oracle 11g, 12c,..)**
2. **Install cx_oracle library for communicating with oracle database**
    a. **Pip install cx_oracle**



Basic steps to communicate with database

1. Establish connection to database
2. Create cursor object
3. Send SQL statements
4. Read Results
5. Close Connection

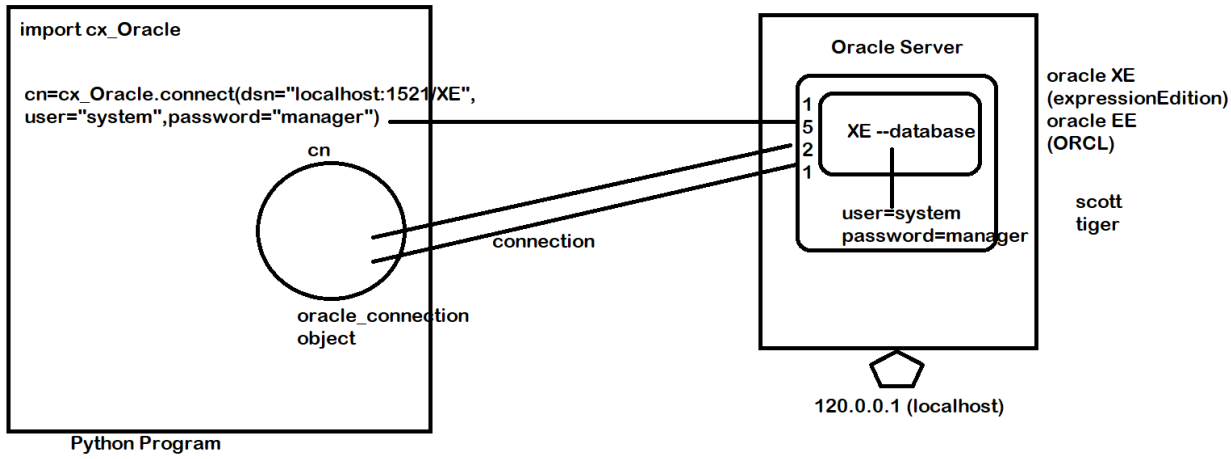
Establishing connection to oracle database

cx_oracle.connect() : This function establish connection to oracle database and returns connection object. Establishing connection is nothing creating session.

Syntax:

cx_oracle.connect(database,user,password,host)

database database name (dsn)
user database username
password database password
host ip-address or hostname where database is running



Example:

```
>>> import cx_Oracle
>>>
cn=cx_Oracle.connect(dsn="localhost:1521/XE",user="system",passw
ord="manager")
>>> print(cn)
<cx_Oracle.Connection to system@localhost:1521/XE>
>>>
```

Example:

```
>>> dsn = cx_Oracle.makedsn("localhost", 1521, service_name="XE")
```

```
>>> connection = cx_Oracle.connect(user="system",
password=”manager”, dsn=dsn)
```

Cursor object

Cursor object is used for sending SQL statements to database.

How to create cursor object?
connectionobject.cursor()

Example:

```
>>>
cn=cx_Oracle.connect(dsn="localhost:1521/XE",user="system",passw
ord="manager")
>>> print(cn)
<cx_Oracle.Connection to system@localhost:1521/XE>
>>> c=cn.cursor()
>>> print(c)
<cx_Oracle.Cursor on <cx_Oracle.Connection to
system@localhost:1521/XE>>
>>>
```

Cursor object provides the following methods for sending SQL statements to database.

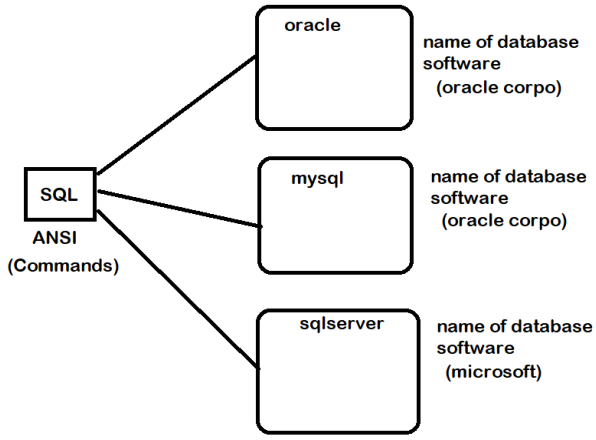
1. execute()
2. executemany()
3. executescript()

SQL

What is SQL?

SQL stands for Structure Query Language. SQL define set of rules and regulations to communicate with various Database software's or applications.

SQL provides set of commands which are used to communicate with various databases.



SQL commands are classified into different categories and these are called SQL sub languages.

1. DDL Data Definition Language
    a. CREATE
    b. ALTER
    c. DROP
2. DML Data Manipulation Language
    a. INSERT
    b. UPDATE
    c. DELETE
3. DRL/DQL Data Query Language
    a. SELECT

4. DCL Data Control Language
    a. GRANT
    b. REVOKE
5. TCL Transaction Control Language
    a. COMMIT
    b. ROLLBACK

In database data is stored in the form tables. A table is collection of rows and columns (fields).

Creating Table

Syntax:

Create table <table-name>(<column-name> <datatype>, <column-name> <datatype>,<column-name> <datatype>,…);