Using MySQL-connector in Python

Updated June 2019-20

by Somnath PaulChoudhury

Version 1.1 last updated June 23, 2019

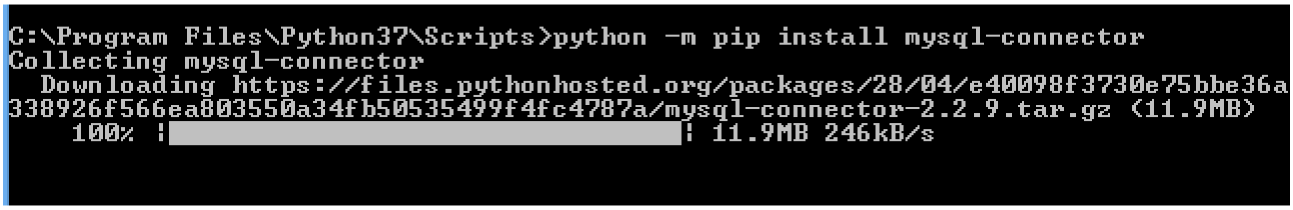
Python is used in database applications

MySQL must be installed successfully first then to access MySQL databases Python needs **MySQL connector**

pip is a package-management system used to install and manage software packages written in Python.

We need to go to the folder (using Windows OS!) in Python where PIP is located mostly under Python and sub-folder Scripts where we need to type

python -m pip install mysql-connector



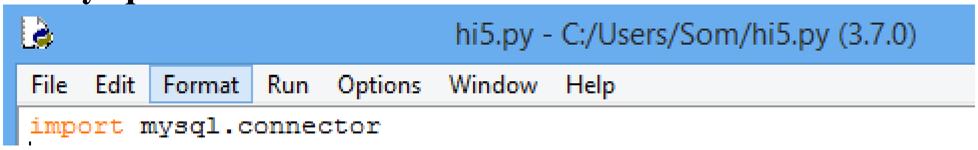
```
C:\Program Files\Python37\Scripts>python -m pip install mysql-connector
Collecting mysql-connector
  Downloading https://files.pythonhosted.org/packages/28/04/e40098f3730e75bbe36a
338926f566ea803550a34fb50535499f4fc4787a/mysql-connector-2.2.9.tar.gz (11.9MB)
    100% |████████████████████████████████| 11.9MB 246kB/s
```

Once MySQL-connector is installed it can be used by Python

If we type the following in a py file and run it will not show any error now

(successfully installed)

import mysql.connector

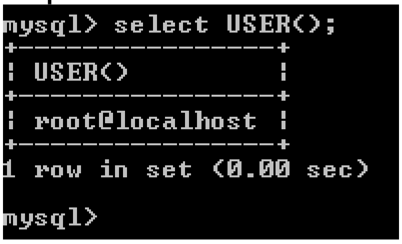


hi5.py - C:/Users/Som/hi5.py (3.7.0)
File Edit Format Run Options Window Help

```
import mysql.connector
```

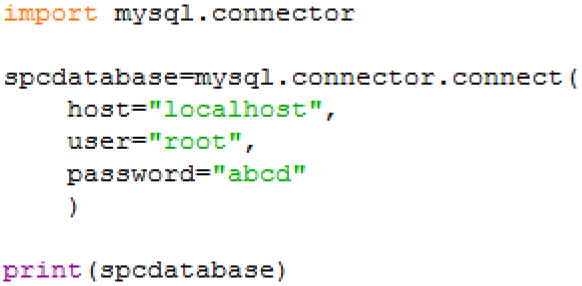
Create a connection to the database

You can check the host and username as follows from MySQL command prompt as



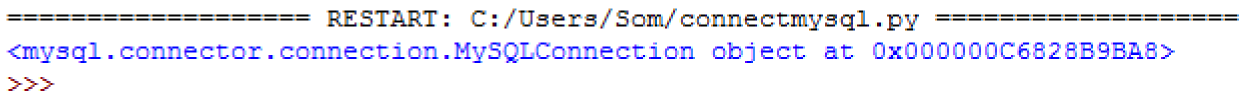
```
mysql> select USER();
+----------------+
| USER()         |
+----------------+
| root@localhost |
+----------------+
1 row in set (0.00 sec)

mysql>
```

user is **root** host is **localhost** and say password given as **abcd,** type the following in a py file and run



```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="abcd"
    )

print(spcdatabase)
```

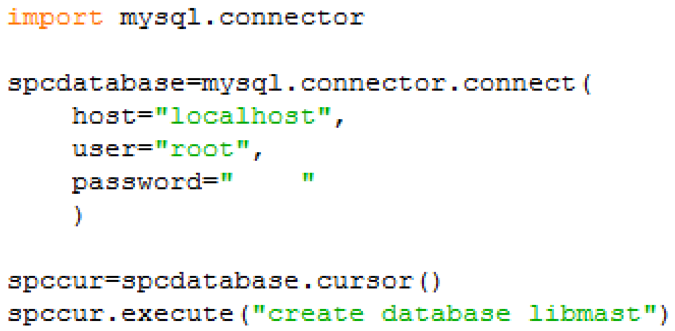
The output will be as below



```
=================== RESTART: C:/Users/Som/connectmysql.py ===================
<mysql.connector.connection.MySQLConnection object at 0x000000C6828B9BA8>
>>>
```

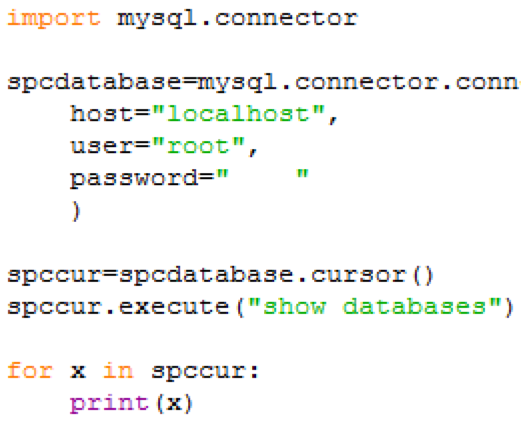
And Python is ready for SQL query!!

Creating the database **libmast**



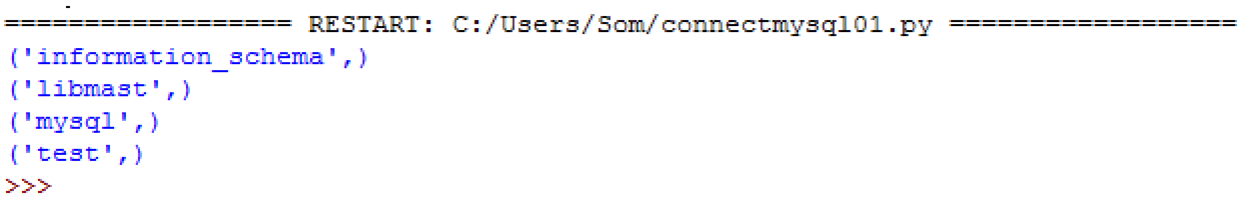
```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    "
    )

spccur=spcdatabase.cursor()
spccur.execute("create database libmast")
```

Now you can check if its created



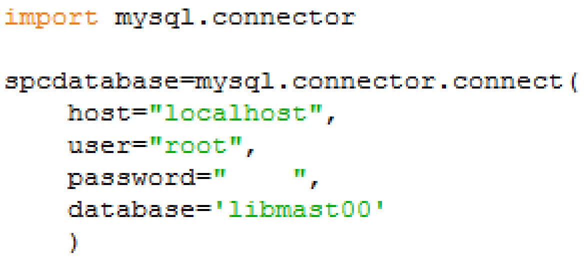
```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    "
    )

spccur=spcdatabase.cursor()
spccur.execute("show databases")

for x in spccur:
    print(x)
```

The output is



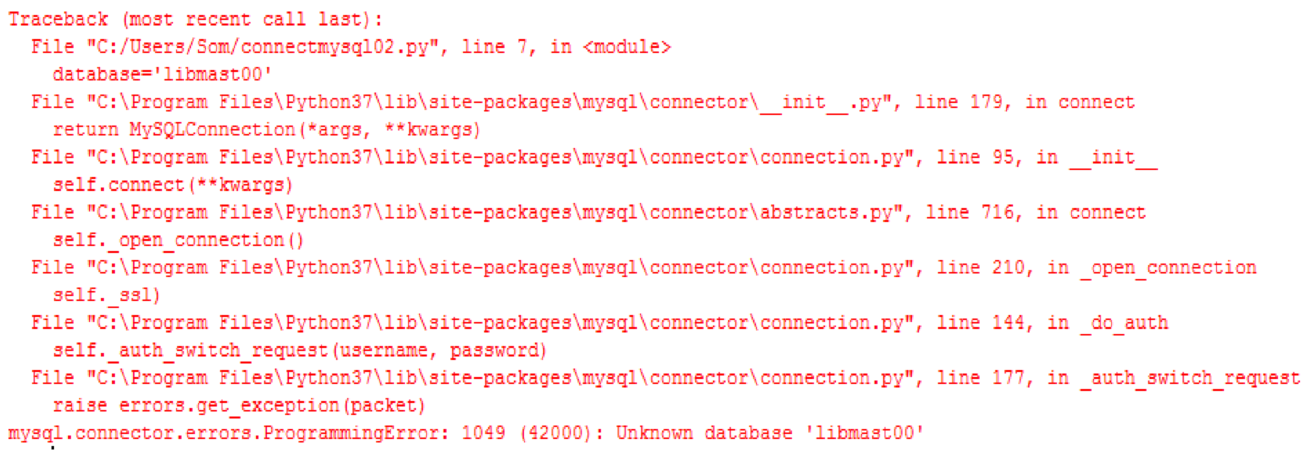
```
================== RESTART: C:/Users/Som/connectmysql01.py ==================
('information_schema',)
('libmast',)
('mysql',)
('test',)
>>>
```

Suppose we want to check if libmast00 database exists we can also do this



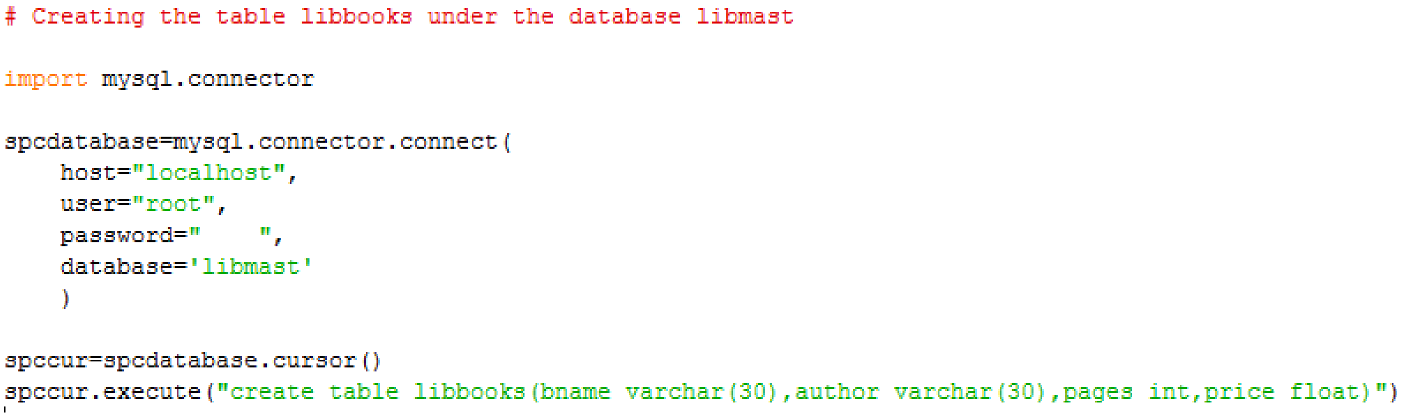
```
import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast00'
    )
```

Since the database libmast00 does not exist as of now we will get the following error, otherwise with database **libmast** there will be no error



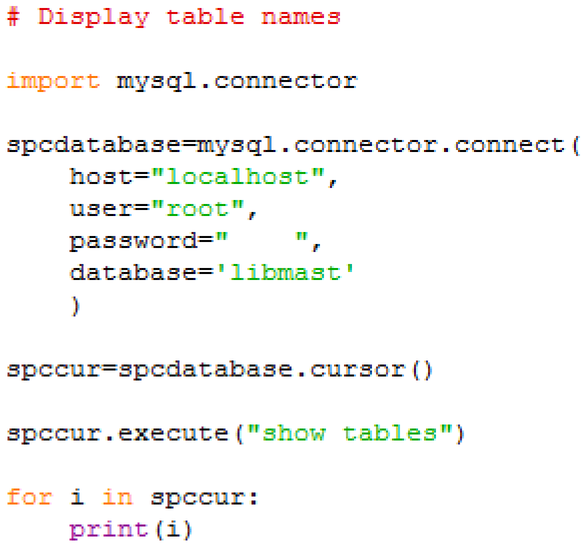
```
Traceback (most recent call last):
  File "C:/Users/Som/connectmysql02.py", line 7, in <module>
    database='libmast00'
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\__init__.py", line 179, in connect
    return MySQLConnection(*args, **kwargs)
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\connection.py", line 95, in __init__
    self.connect(**kwargs)
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\abstracts.py", line 716, in connect
    self._open_connection()
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\connection.py", line 210, in _open_connection
    self._ssl)
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\connection.py", line 144, in _do_auth
    self._auth_switch_request(username, password)
  File "C:\Program Files\Python37\lib\site-packages\mysql\connector\connection.py", line 177, in _auth_switch_request
    raise errors.get_exception(packet)
mysql.connector.errors.ProgrammingError: 1049 (42000): Unknown database 'libmast00'
```

Now let us create a table **libbooks** under libmast database,

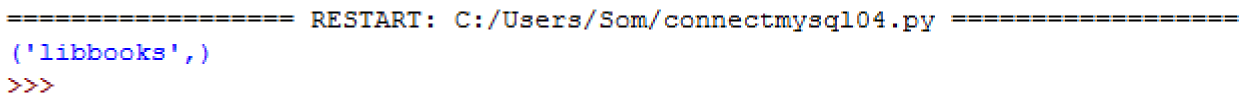


```
# Creating the table libbooks under the database libmast

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()
spccur.execute("create table libbooks(bname varchar(30),author varchar(30),pages int,price float)")
```

Check if the table has been created

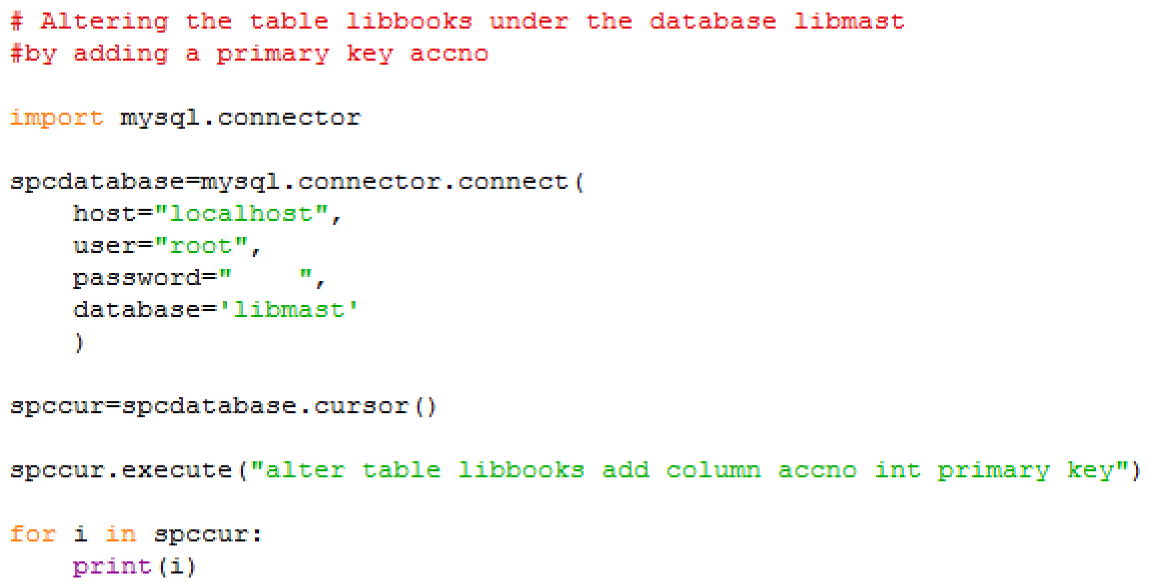


```
# Display table names

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()

spccur.execute("show tables")

for i in spccur:
    print(i)
```



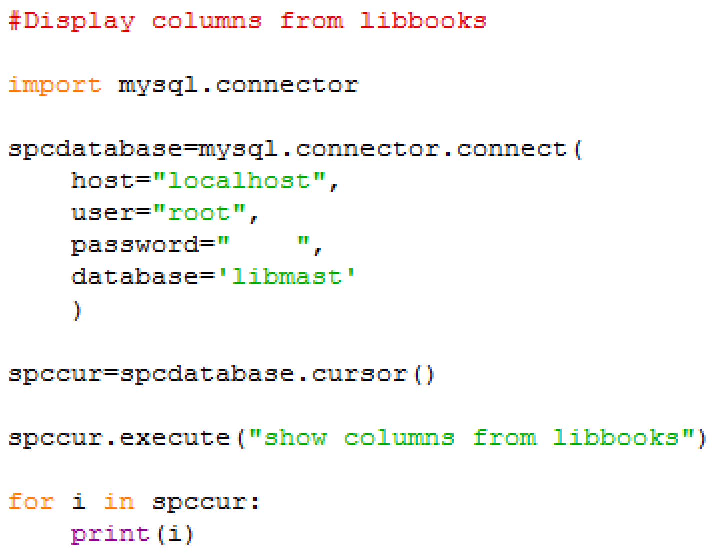
```
================== RESTART: C:/Users/Som/connectmysql04.py ==================
('libbooks',)
>>>
```

Adding a new column in the table by using **alter command**



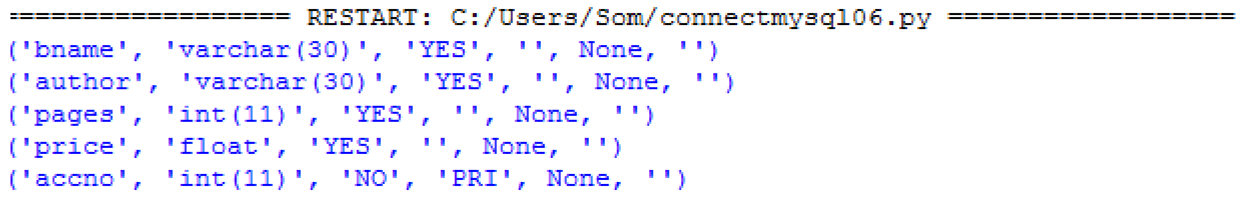
```
# Altering the table libbooks under the database libmast
#by adding a primary key accno

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()

spccur.execute("alter table libbooks add column accno int primary key")

for i in spccur:
    print(i)
```

Now to check the altered table structure



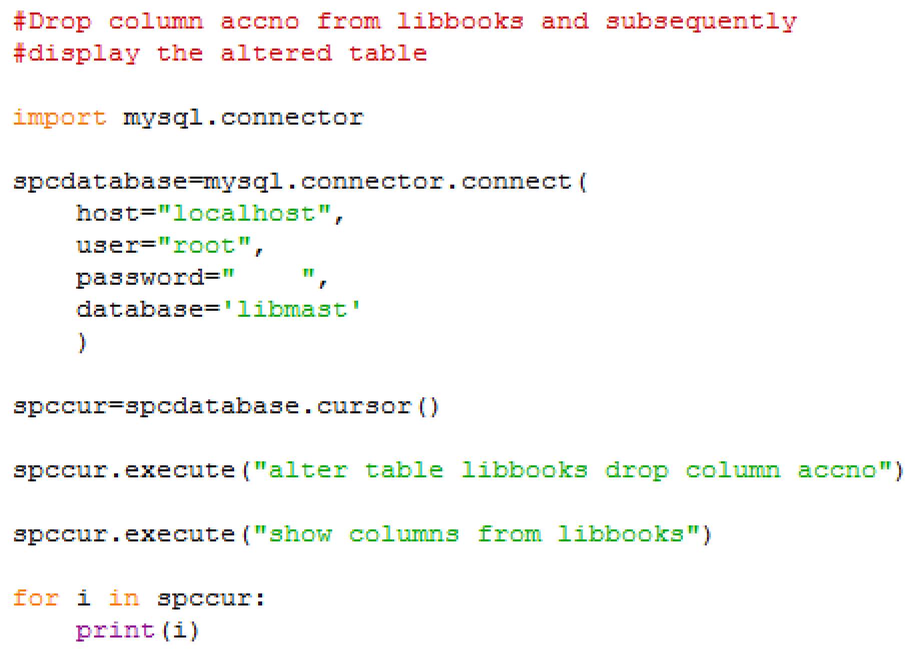
```
#Display columns from libbooks

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()

spccur.execute("show columns from libbooks")

for i in spccur:
    print(i)
```

The output is



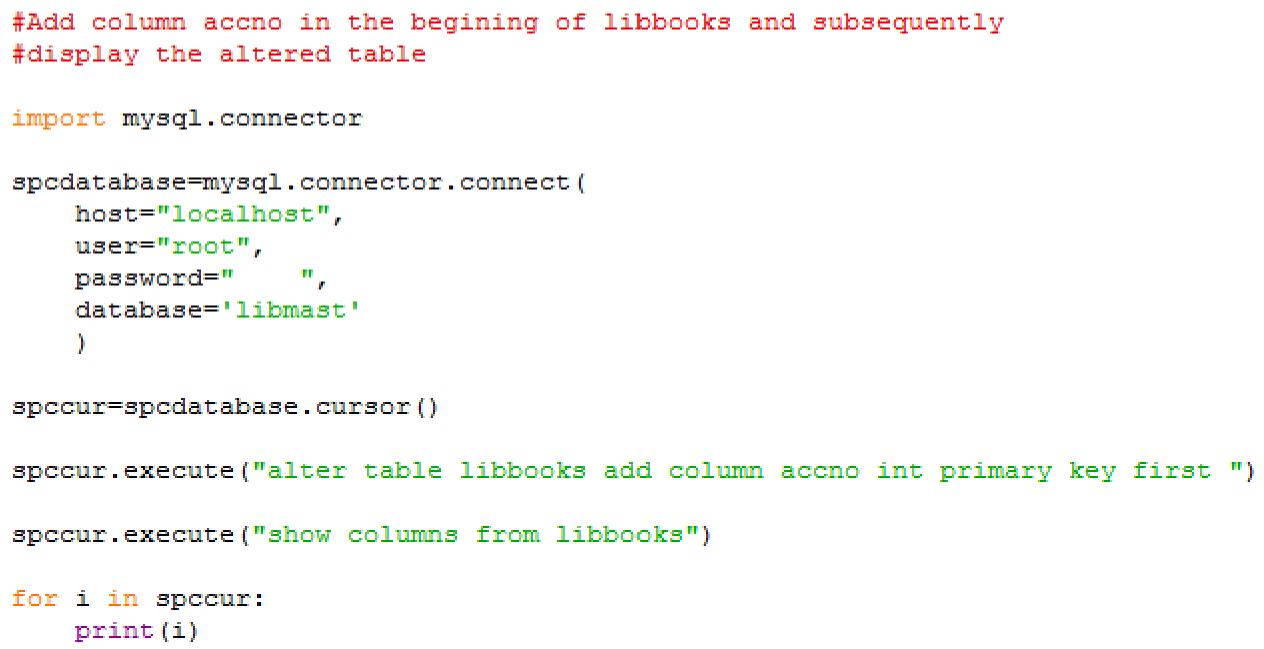
```
================== RESTART: C:/Users/Som/connectmysql06.py ==================
('bname', 'varchar(30)', 'YES', '', None, '')
('author', 'varchar(30)', 'YES', '', None, '')
('pages', 'int(11)', 'YES', '', None, '')
('price', 'float', 'YES', '', None, '')
('accno', 'int(11)', 'NO', 'PRI', None, '')
```

To drop the column accno we need to change in the script as



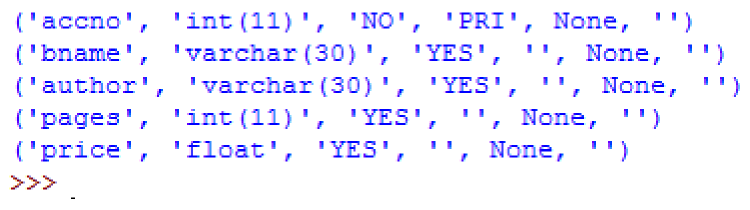
```
#Drop column accno from libbooks and subsequently
#display the altered table

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()

spccur.execute("alter table libbooks drop column accno")

spccur.execute("show columns from libbooks")

for i in spccur:
    print(i)
```

To add the field accno at the begining



```
#Add column accno in the begining of libbooks and subsequently
#display the altered table

import mysql.connector

spcdatabase=mysql.connector.connect(
    host="localhost",
    user="root",
    password="    ",
    database='libmast'
    )

spccur=spcdatabase.cursor()

spccur.execute("alter table libbooks add column accno int primary key first ")

spccur.execute("show columns from libbooks")

for i in spccur:
    print(i)
```

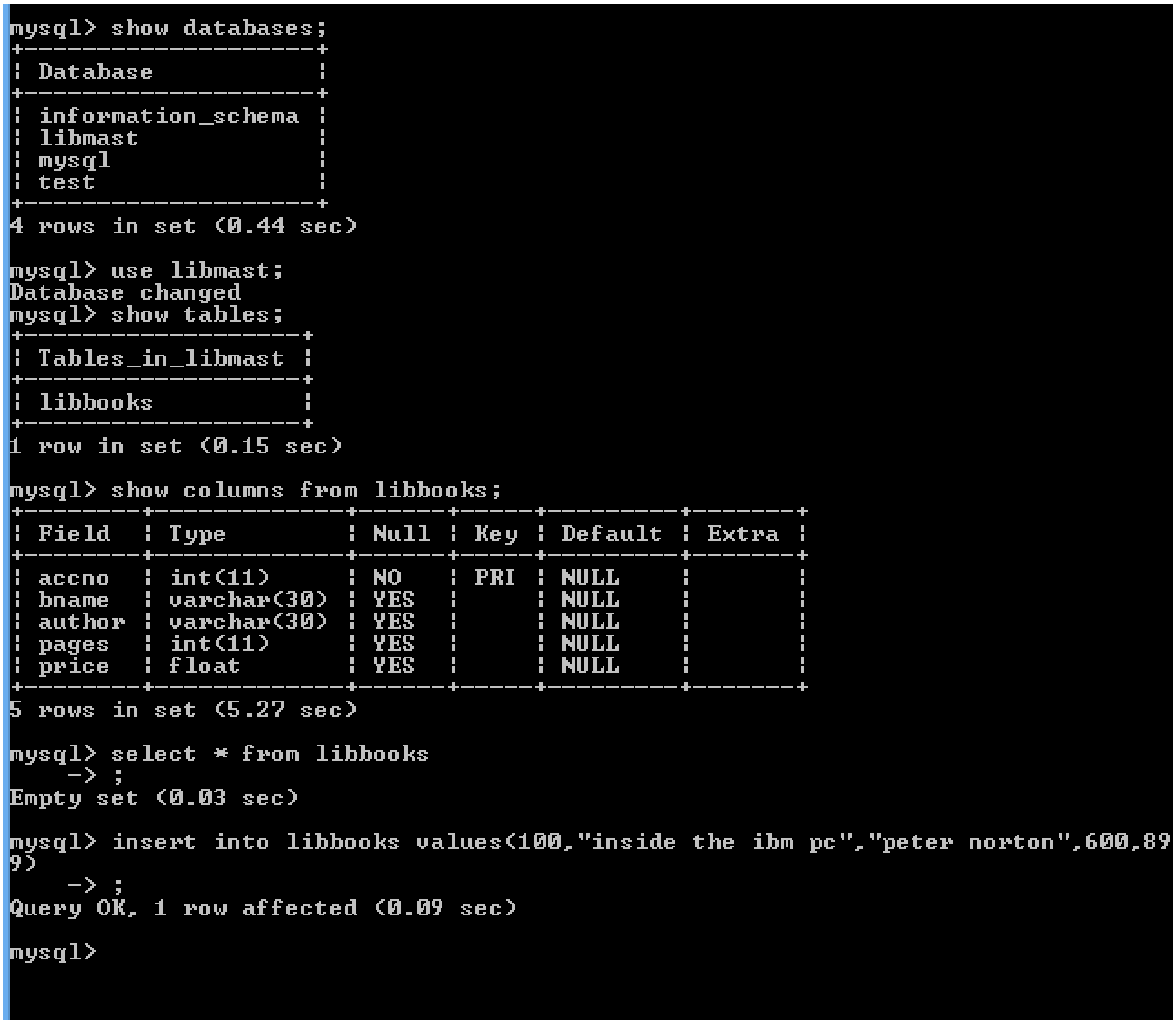
The output is



```
('accno', 'int(11)', 'NO', 'PRI', None, '')
('bname', 'varchar(30)', 'YES', '', None, '')
('author', 'varchar(30)', 'YES', '', None, '')
('pages', 'int(11)', 'YES', '', None, '')
('price', 'float', 'YES', '', None, '')
>>>
```

Inserting the first record in the table libbooks of database libmast using mysql prompt....

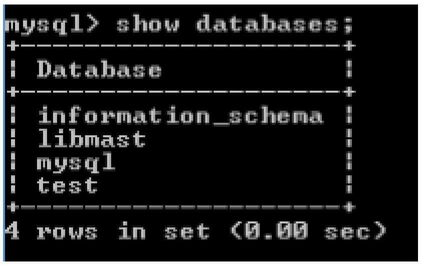
Also checking the same done above in Python...



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| libmast            |
| mysql              |
| test               |
+--------------------+
4 rows in set (0.44 sec)

mysql> use libmast;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_libmast |
+-------------------+
| libbooks          |
+-------------------+
1 row in set (0.15 sec)

mysql> show columns from libbooks;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| accno  | int(11)     | NO   | PRI | NULL    |       |
| bname  | varchar(30) | YES  |     | NULL    |       |
| author | varchar(30) | YES  |     | NULL    |       |
| pages  | int(11)     | YES  |     | NULL    |       |
| price  | float       | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (5.27 sec)

mysql> select * from libbooks
    -> ;
Empty set (0.03 sec)

mysql> insert into libbooks values(100,"inside the ibm pc","peter norton",600,89
9)
    -> ;
Query OK, 1 row affected (0.09 sec)

mysql>
```

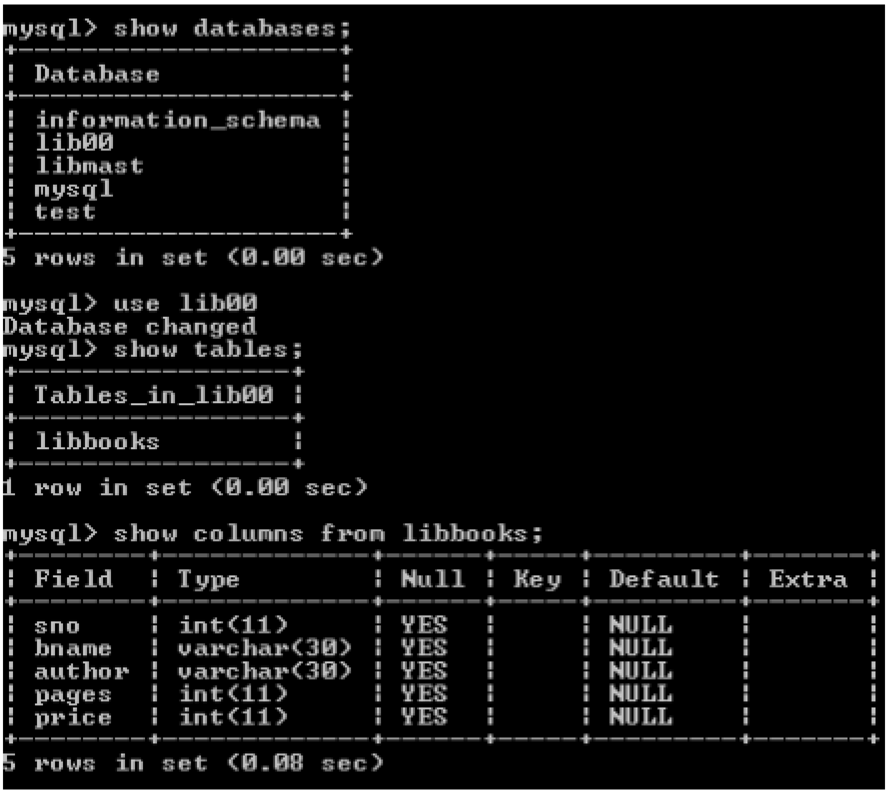
It turns out that we have suceeded in creating a real table in a real database in MySql using Python!

Lets code in Python to add database name , table in execution time the result of the code can be checked in MySql command prompt later as below...it shows that the database lib00 was created (not existing before)



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| libmast            |
| mysql              |
| test               |
+--------------------+
4 rows in set (0.00 sec)
```

After the execution of the Python code to create at execution time we check the same in MySql...



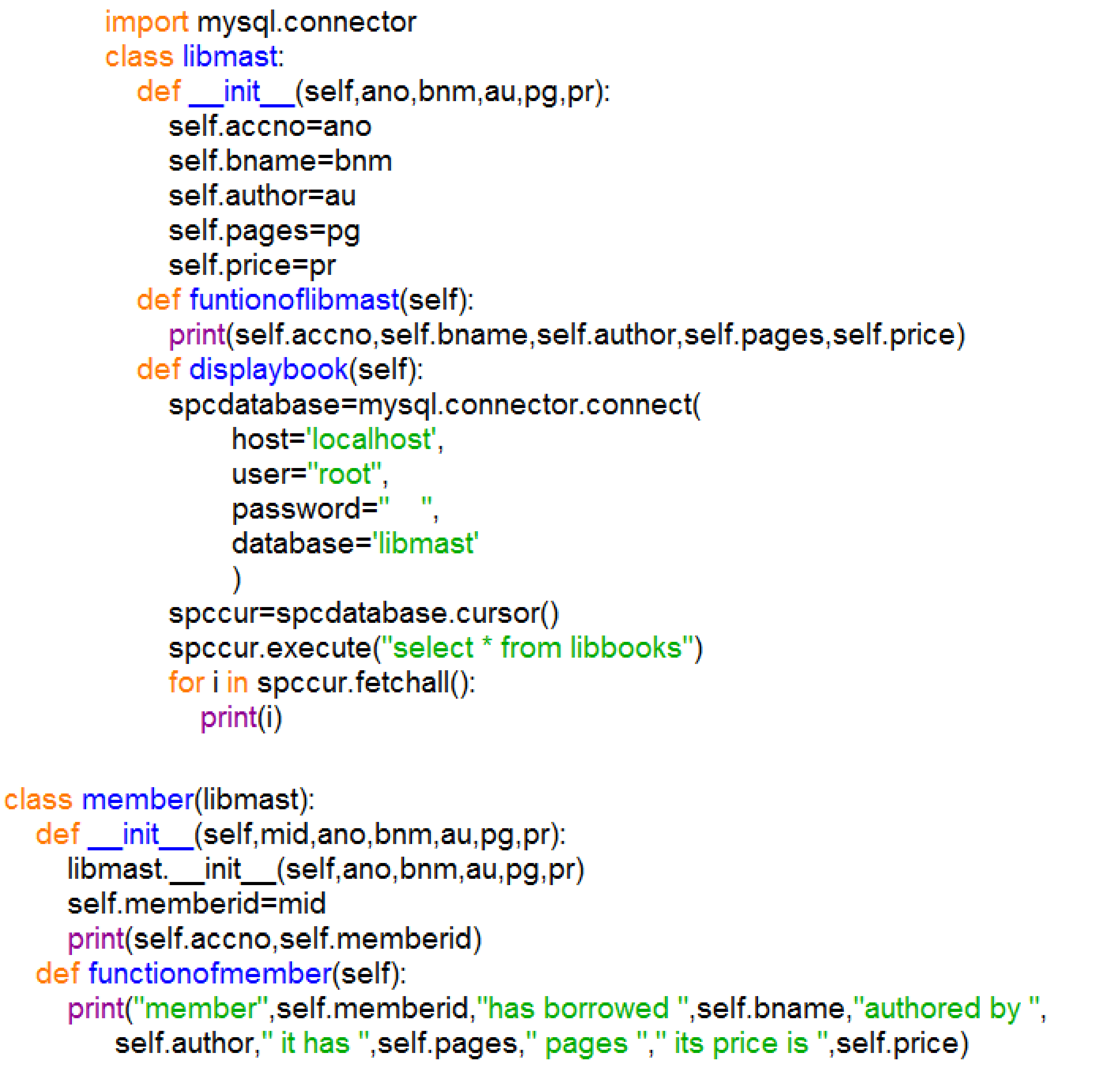
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| lib00              |
| libmast            |
| mysql              |
| test               |
+--------------------+
5 rows in set (0.00 sec)

mysql> use lib00
Database changed
mysql> show tables;
+-----------------+
| Tables_in_lib00 |
+-----------------+
| libbooks        |
+-----------------+
1 row in set (0.00 sec)

mysql> show columns from libbooks;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| sno    | int(11)     | YES  |     | NULL    |       |
| bname  | varchar(30) | YES  |     | NULL    |       |
| author | varchar(30) | YES  |     | NULL    |       |
| pages  | int(11)     | YES  |     | NULL    |       |
| price  | int(11)     | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
5 rows in set (0.08 sec)
```

Note that show databases command displays a database list having lib00

Here is the Python code to create database and table at execution time

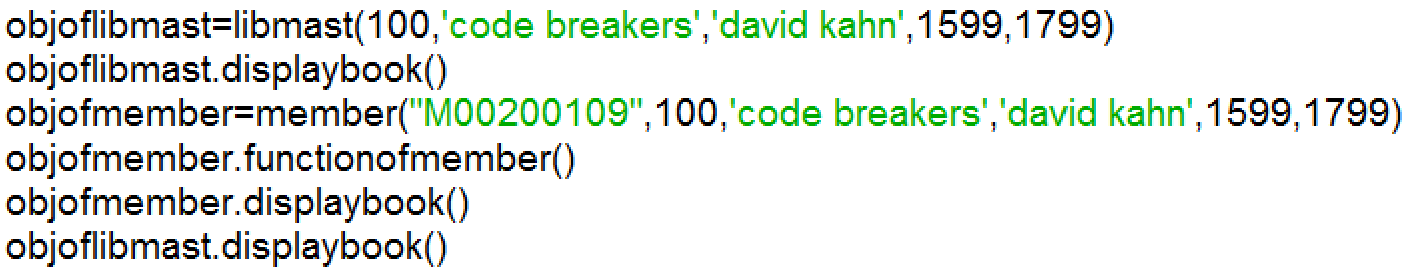
```python
import mysql.connector
spcdatabase=mysql.connector.connect(
   host="localhost",
   user="root",
   password="   "
   )
spccur=spcdatabase.cursor()
databasename=input("Enter a name for the database to be created ")
spccur.execute("create database {}".format(databasename))
spccur.execute("show databases")
for x in spccur:
   print(x)
spcdatabase=mysql.connector.connect(
   host="localhost",
   user="root",
   password="   ",
   database=databasename
   )
spccur=spcdatabase.cursor()
tablename=input("Enter a name for the SQL table ")
nooffields=int(input("Enter no of fields "))
field=input("Enter the field name ")
typeorsize=input("Enter type for int, float or varchar(size) for varchar ")
spccur.execute("create table {} ({} {})".format(tablename,field,typeorsize))
i=1
while(i<=nooffields-1):
   field=input("Enter the field name ")
   typeorsize=input("Enter type for int, float or size for varchar ")
   spccur.execute("alter table {} add column {} {}".format(tablename,field,typeorsize))
   i+=1
```

The following two classes exhibits multilevel inheritance. Observe the base class function displaybook() ...here's the class libmast followed by member



```
import mysql.connector
class libmast:
  def __init__(self,ano,bnm,au,pg,pr):
    self.accno=ano
    self.bname=bnm
    self.author=au
    self.pages=pg
    self.price=pr
  def funtionoflibmast(self):
    print(self.accno,self.bname,self.author,self.pages,self.price)
  def displaybook(self):
    spcdatabase=mysql.connector.connect(
        host='localhost',
        user="root",
        password="    ",
        database='libmast'
        )
    spccur=spcdatabase.cursor()
    spccur.execute("select * from libbooks")
    for i in spccur.fetchall():
      print(i)


class member(libmast):
  def __init__(self,mid,ano,bnm,au,pg,pr):
    libmast.__init__(self,ano,bnm,au,pg,pr)
    self.memberid=mid
    print(self.accno,self.memberid)
  def functionofmember(self):
    print("member",self.memberid,"has borrowed ",self.bname,"authored by ",
       self.author," it has ",self.pages," pages "," its price is ",self.price)
```

Lets create objects of the two classes and run the code



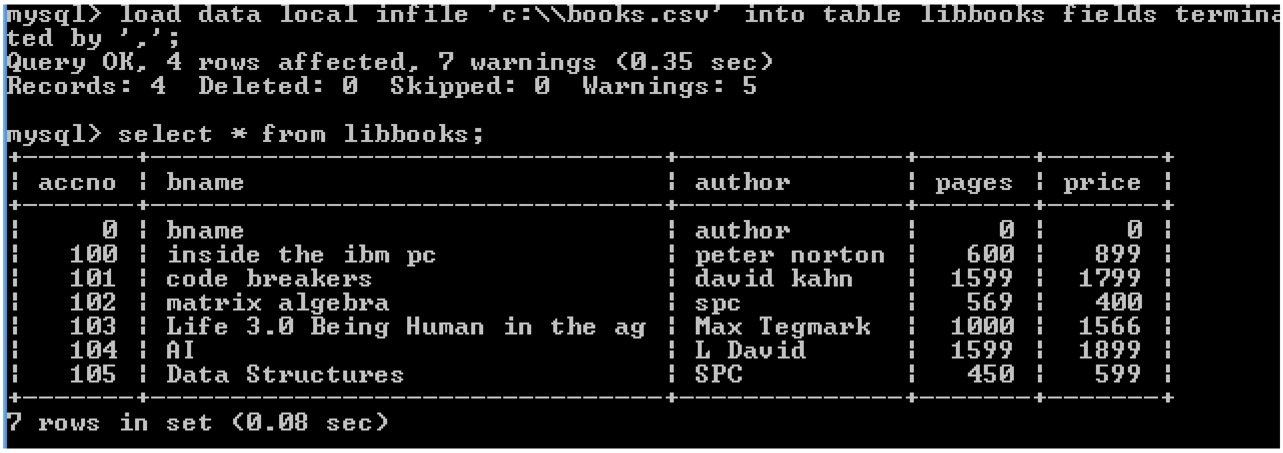
```
objoflibmast=libmast(100,'code breakers','david kahn',1599,1799)
objoflibmast.displaybook()
objofmember=member("M00200109",100,'code breakers','david kahn',1599,1799)
objofmember.functionofmember()
objofmember.displaybook()
objoflibmast.displaybook()
```

The output is as below

```
(100, 'code breakers', 'david kahn', 1599, 1799.0)
(101, 'inside the ibm pc', 'peter norton', 599, 699.0)
(102, 'python programming', 'spc', 399, 599.0)
(103, 'data science with python', 'kevin lee', 250, 500.0)
100 M00200109
member M00200109 has borrowed  code breakers authored by  david kahn  it has  1599  pages   its price is  1799
(100, 'code breakers', 'david kahn', 1599, 1799.0)
(101, 'inside the ibm pc', 'peter norton', 599, 699.0)
(102, 'python programming', 'spc', 399, 599.0)
(103, 'data science with python', 'kevin lee', 250, 500.0)
(100, 'code breakers', 'david kahn', 1599, 1799.0)
(101, 'inside the ibm pc', 'peter norton', 599, 699.0)
(102, 'python programming', 'spc', 399, 599.0)
(103, 'data science with python', 'kevin lee', 250, 500.0)
```

High speed data transfer from a csv file

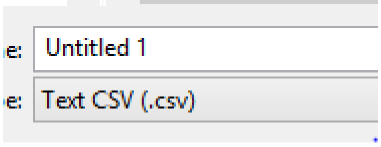
It's not possible to add records one by one also source data needs cleansing and formatting before they are uploaded. There may be missing values or impossible values and these can be dealt with some uniform policy..look at the ouput of a simple MySql command and think how it will be like adding 1000 or 20000 records one by one!! The table shows some missing data



```
mysql> load data local infile 'c:\\books.csv' into table libbooks fields termina
ted by ',';
Query OK, 4 rows affected, 7 warnings (0.35 sec)
Records: 4  Deleted: 0  Skipped: 0  Warnings: 5

mysql> select * from libbooks;
+-------+---------------------------------+--------------+-------+-------+
| accno | bname                           | author       | pages | price |
+-------+---------------------------------+--------------+-------+-------+
|     0 | bname                           | author       |     0 |     0 |
|   100 | inside the ibm pc               | peter norton |   600 |   899 |
|   101 | code breakers                   | david kahn   |  1599 |  1799 |
|   102 | matrix algebra                  | spc          |   569 |   400 |
|   103 | Life 3.0 Being Human in the ag  | Max Tegmark  |  1000 |  1566 |
|   104 | AI                              | L David      |  1599 |  1899 |
|   105 | Data Structures                 | SPC          |   450 |   599 |
+-------+---------------------------------+--------------+-------+-------+
7 rows in set (0.08 sec)
```

We can create comma separated values file in open office Calc just

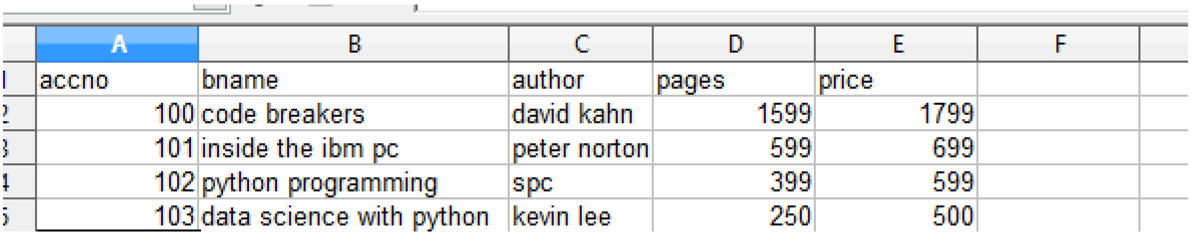
save as .csv type



e: Untitled 1

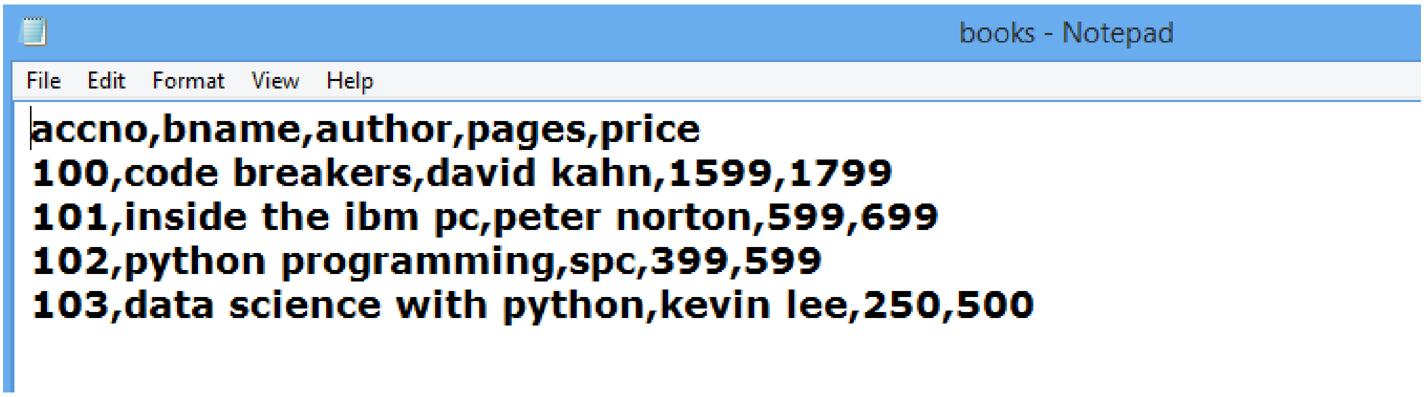
e: Text CSV (.csv)

Data shown below is cleaned and formatted and ready to save as .csv



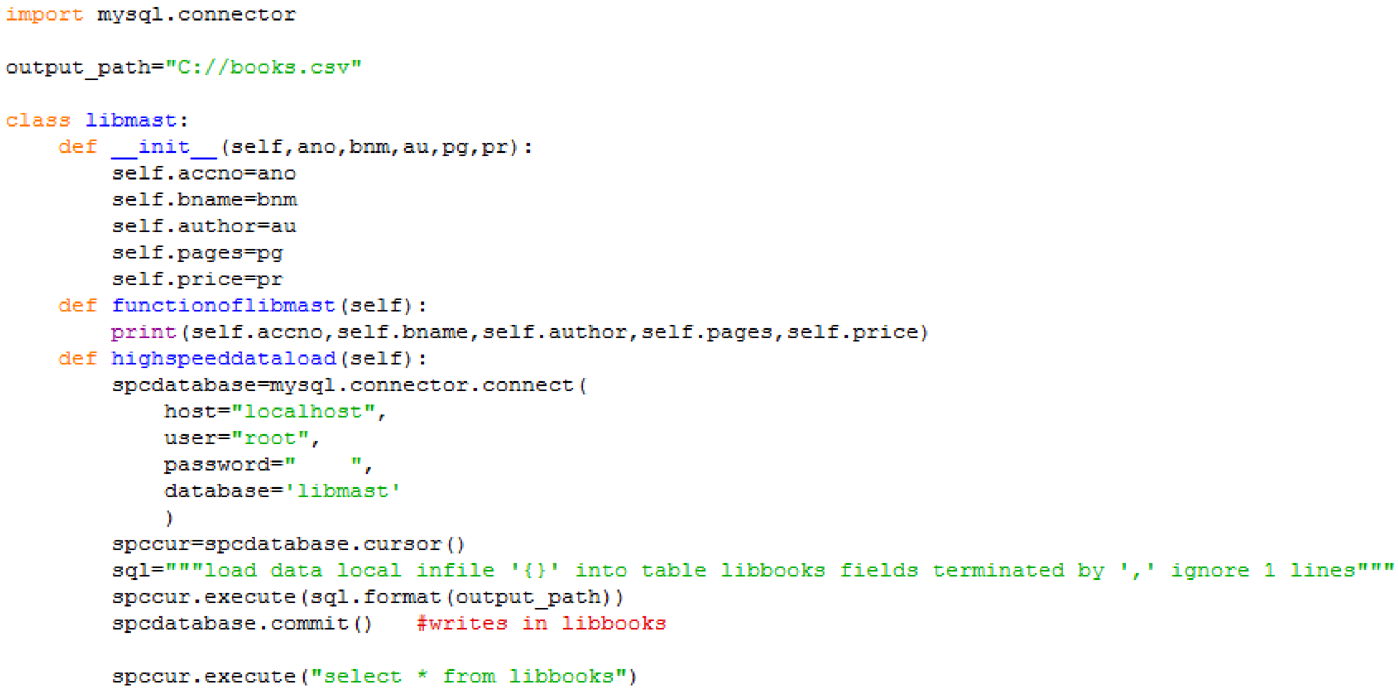
	A	B	C	D	E	F
1	accno	bname	author	pages	price	
2	100	code breakers	david kahn	1599	1799	
3	101	inside the ibm pc	peter norton	599	699	
4	102	python programming	spc	399	599	
5	103	data science with python	kevin lee	250	500	

Data saved as .csv



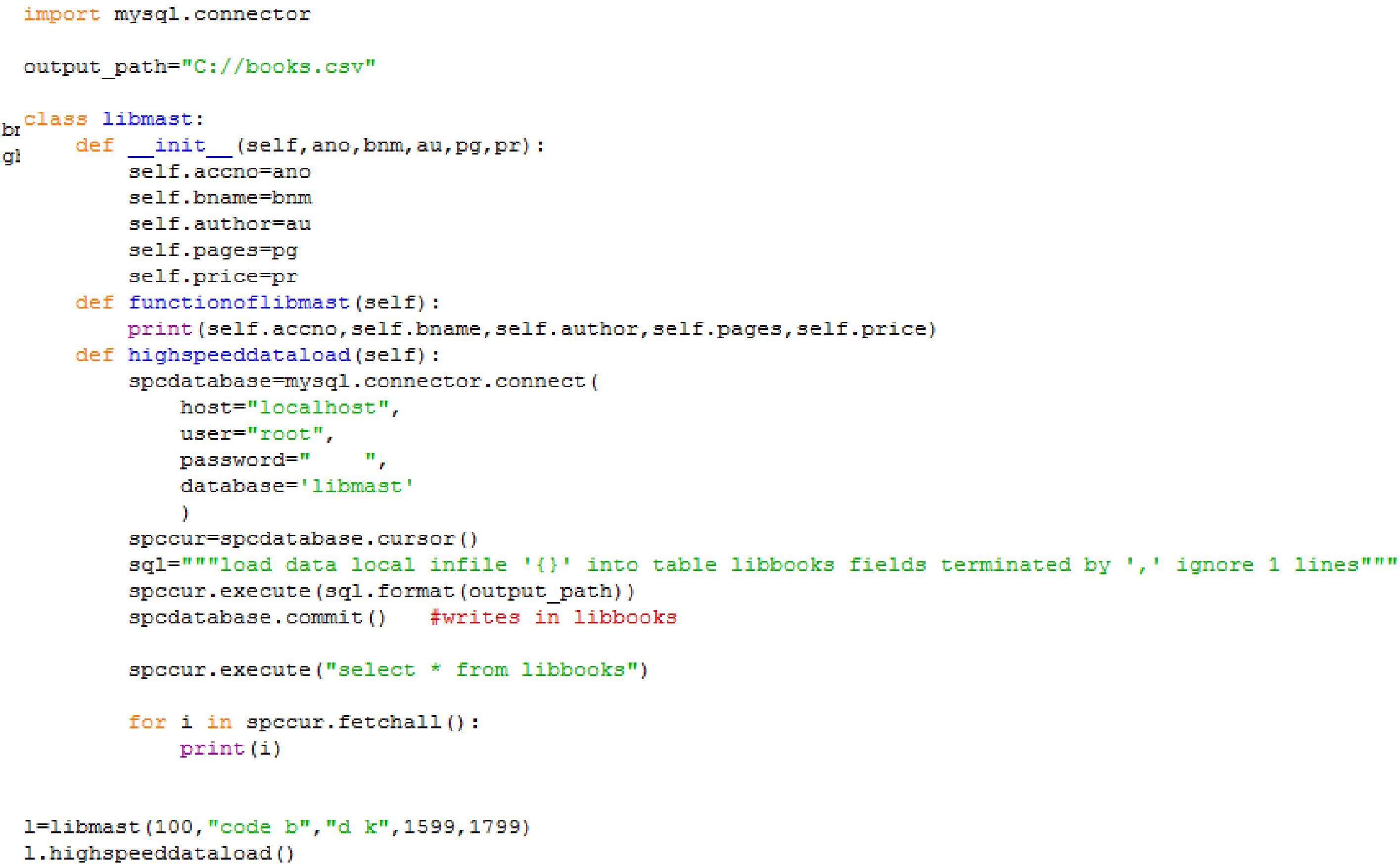
```
accno,bname,author,pages,price
100,code breakers,david kahn,1599,1799
101,inside the ibm pc,peter norton,599,699
102,python programming,spc,399,599
103,data science with python,kevin lee,250,500
```

What about writing a Python program for high speed data upload?



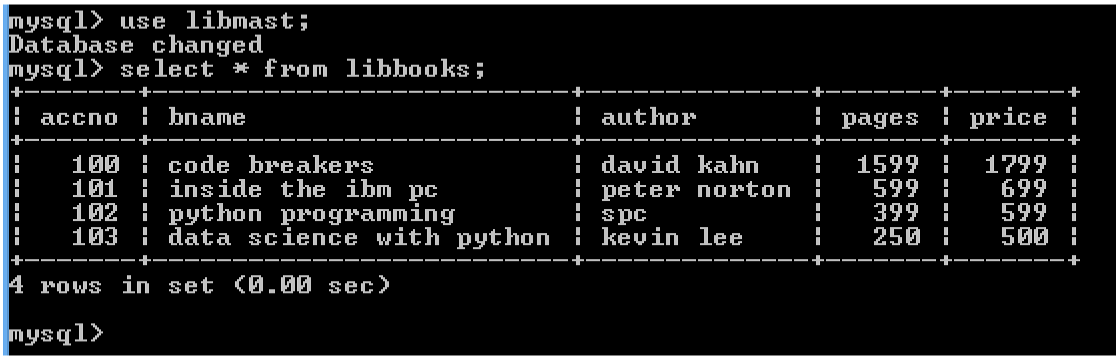
```
import mysql.connector

output_path="C://books.csv"

class libmast:
    def __init__(self,ano,bnm,au,pg,pr):
        self.accno=ano
        self.bname=bnm
        self.author=au
        self.pages=pg
        self.price=pr
    def functionoflibmast(self):
        print(self.accno,self.bname,self.author,self.pages,self.price)
    def highspeeddataload(self):
        spcdatabase=mysql.connector.connect(
            host="localhost",
            user="root",
            password="    ",
            database='libmast'
            )
        spccur=spcdatabase.cursor()
        sql="""load data local infile '{}' into table libbooks fields terminated by ',' ignore 1 lines"""
        spccur.execute(sql.format(output_path))
        spcdatabase.commit()   #writes in libbooks

        spccur.execute("select * from libbooks")
```

```
l=libr
l.higl
```



```
import mysql.connector

output_path="C://books.csv"

class libmast:
    def __init__(self,ano,bnm,au,pg,pr):
        self.accno=ano
        self.bname=bnm
        self.author=au
        self.pages=pg
        self.price=pr
    def functionoflibmast(self):
        print(self.accno,self.bname,self.author,self.pages,self.price)
    def highspeeddataload(self):
        spcdatabase=mysql.connector.connect(
            host="localhost",
            user="root",
            password="    ",
            database='libmast'
            )
        spccur=spcdatabase.cursor()
        sql="""load data local infile '{}' into table libbooks fields terminated by ',' ignore 1 lines"""
        spccur.execute(sql.format(output_path))
        spcdatabase.commit()   #writes in libbooks

        spccur.execute("select * from libbooks")

        for i in spccur.fetchall():
            print(i)


l=libmast(100,"code b","d k",1599,1799)
l.highspeeddataload()
```

We can see how it is



```
mysql> use libmast;
Database changed
mysql> select * from libbooks;
+--------+--------------------------+--------------+-------+-------+
| accno  | bname                    | author       | pages | price |
+--------+--------------------------+--------------+-------+-------+
|    100 | code breakers            | david kahn   |  1599 |  1799 |
|    101 | inside the ibm pc        | peter norton |   599 |   699 |
|    102 | python programming       | spc          |   399 |   599 |
|    103 | data science with python | kevin lee    |   250 |   500 |
+--------+--------------------------+--------------+-------+-------+
4 rows in set (0.00 sec)

mysql>
```

by displaying the same using MySQL command after high speed upload

Somnath PaulChoudhury

interested in

Data Science Machine Learning and AI