

```

#*****
#*****
# This file is part of the program named "MyLib" and called by
# MyLib1.py (main file).
#
#*****
#*****

# *****
# ***Create database named MyLib and tables.***
# *****

def CreateDatabase(Cursor):
    Cursor.execute("drop DATABASE IF EXISTS MyLib")
    Cursor.execute("create DATABASE MyLib")
    Cursor.execute("use MyLib")

    # Create and use TABLE named MyBooks
    Cursor.execute("drop TABLE IF EXISTS MyBooks")
    Cursor.execute("""CREATE TABLE MyBooks(
    AccessNo integer(8) PRIMARY KEY NOT NULL,
    Name CHAR(50),
    Author CHAR(20),
    Type char(20),
    Language char(10),
    PubYear integer(5),
    IssuedTo integer(10) DEFAULT 0,
    IssueDate date DEFAULT NULL)""")

    # Notice that there is a mix of types (strings, ints) though we still
    # only use %s.
    sql = "INSERT INTO MyBooks (AccessNo, Name,Author, Type, Language,
    PubYear) VALUES (%s,%s,%s,%s,%s,%s)"
    vals = [
        (101,'Harry potter and the Philosopher\'s Stone','Jk Rowling',
        'Fiction', 'English',1997),
        (301,'Harry potter and the Philosopher\'s Stone','Jk Rowling',
        'Fiction', 'English',1997),
        (102,'Harry Potter and the Chamber of Secrets','Jk Rowling',
        'Fiction', 'English',1998),
        (302,'Harry Potter and the Chamber of Secrets','Jk Rowling',
        'Fiction', 'English',1998),
        (103,'Harry Potter and the Prisoner of Azkaban','Jk Rowling',
        'Fiction', 'English',1999),
        (303,'Harry Potter and the Prisoner of Azkaban','Jk Rowling',
        'Fiction', 'English',1999),
        (104,'Harry Potter and the Goblet of Fire','Jk Rowling',
        'Fiction', 'English',2000),
        (304,'Harry Potter and the Goblet of Fire','Jk Rowling',
        'Fiction', 'English',2000),
        (105,'Harry Potter and the Order of the Phoenix','Jk Rowling',
        'Fiction', 'English',2003),
        (305,'Harry Potter and the Order of the Phoenix','Jk Rowling',
        'Fiction', 'English',2003),
        (106,'Harry Potter and the Half-Blood Prince','Jk Rowling',
        'Fiction', 'English',2005),
        (306,'Harry Potter and the Half-Blood Prince','Jk Rowling',
        'Fiction', 'English',2005),
        (107,'Harry Potter and the Deathly Hallows','Jk Rowling',

```

'Fiction', 'English',2007),
 (307,'Harry Potter and the Deathly Hallows','Jk Rowling',
 'Fiction', 'English',2007),
 (111,'The Lightning Thief','Rick Riordan', 'Fiction/Myth',
 'English', 2005),
 (112,'The Sea of Monsters','Rick Riordan', 'Fiction/Myth',
 'English', 2006),
 (114,'The Titan\'s Curse','Rick Riordan', 'Fiction/Myth',
 'English', 2007),
 (115,'The Battle of the Labyrinth','Rick Riordan', 'Fiction/Myth',
 'English',2008),
 (116,'The Last Olympian','Rick Riordan', 'Fiction/Myth',
 'English', 2010),
 (117,'Camp Half-Blood Confidential','Rick Riordan',
 'Fiction/Myth', 'English', 2017),
 (121,'Percy Jacson\'s Greek Gods','Rick Riordan', 'Fiction/Myth',
 'English', 2014),
 (122,'Percy Jacson\'s Greek Heroes','Rick
 Riordan', 'Fiction/Myth', 'English',2014),
 (131,'The Lost Hero','Rick Riordan', 'Fiction/Myth', 'English',
 2010),
 (132,'The Son Of Neptune','Rick Riordan', 'Fiction/Myth',
 'English', 2011),
 (133,'The Mark Of Athena','Rick Riordan', 'Fiction/Myth',
 'English', 2012),
 (134,'The House Of Hades','Rick Riordan', 'Fiction/Myth',
 'English', 2013),
 (135,'The Blood Of Olympus','Rick Riordan', 'Fiction/Myth',
 'English', 2014),
 (141,'The Red Pyramid','Rick Riordan',' Fiction/Myth', 'English',
 2010),
 (142,'The Throne Of Fire','Rick Riordan', 'Fiction/Myth',
 'English', 2011),
 (143,'The Serpent\'s Shadow','Rick Riordan', 'Fiction/Myth',
 'English', 2012),
 (151,'The Sword Of Summer','Rick Riordan', 'Fiction/Myth',
 'English', 2015),
 (152,'The Hammer of Thor','Rick Riordan', 'Fiction/Myth',
 'English', 2016),
 (153,'The Ship Of The Dead','Rick Riordan', 'Fiction/Myth',
 'English', 2017),
 (154,'9 From The Nine Worlds','Rick Riordan', 'Fiction/Myth',
 'English', 2018),
 (161,'The Hidden Oracle','Rick Riordan', 'Fiction/Myth',
 'English', 2016),
 (162,'The Dark Prophecy','Rick Riordan', 'Fiction/Myth',
 'English', 2017),
 (163,'The Burning Maze','Rick Riordan', 'Fiction/Myth', 'English',
 2018),
 (164,'The Tyrant\'s Tomb','Rick Riordan', 'Fiction/Myth',
 'English', 2019),
 (165,'The Tower Of Nero','Rick Riordan', 'Fiction/Myth',
 'English', 2020),
 (171,'Other Worlds','Jon Scieszka', 'Fiction/Myth', 'English',2013),
 (181,'Aru Shah And The End Of Time','Roshani Chokshi',
 'Fiction/Myth', 'English', 2018),
 (182,'Aru Shah And The Song Of Death','Roshani Chokshi', 'Fiction/
 Myth', 'English', 2019),

```

        (183, 'Aru Shah And The Tree Of Wishes', 'Roshani Chokshi', 'Fiction/
Myth', 'English', 2020),
        (191, 'The Storm Runner ', 'Jennifer Cervantes', 'Fiction',
'English', 2018),
        (192, 'The Fire Keeper ', 'Jennifer Cervantes', 'Fiction',
'English', 2019),
        (201, 'Dragon Pearl ', 'Yoon Ha Lee', 'Fiction', 'English', 2019),
        (211, 'Sal And Gabi Break The Universe ', 'Carlos Hernandez',
'Fiction', 'English', 2019),
        (212, 'Sal And Gabi Fix The Universe', 'Carlos Hernandez', 'Fiction',
'English', 2019),
        (221, 'Race To The Sun', 'Rebecca Roanhorse', 'Fiction/Myth',
'English', 2020),
        (231, 'Tristan Strong Punches A Hole In The Sky', 'Kwame Mbalia',
'Fiction/Myth', 'English', 2019),
        (241, 'Paola Santiago And The River of Tears', 'Tehlor Kay Mejia',
'Fiction/Myth', 'English', 2020),
        (251, 'City Of The Plague God', 'Sarwat Chadda', 'Fiction/Myth',
'English', 2020),
        (261, 'The Last Fallen Star', 'Graci Kim', 'Fiction/Myth',
'English', 2016),
        (271, 'Ryan Higa\'s How To Write Good', 'Ryan Higa',
'Autobiography', 'English', 2017),
        (281, 'Start With Why', 'Simon Sinek', 'General', 'English', 2009),
        (282, 'Programming With Python', 'John Smith', 'Course', 'English',
2016)
    ]
    Cursor.executemany(sql, vals)

```

```

# Create and use TABLE named Students
Cursor.execute("drop TABLE IF EXISTS Students")
Cursor.execute("""CREATE TABLE Students(
RollNo integer(10) PRIMARY KEY NOT NULL,
Name CHAR(40),
Class integer(6),
Section char(8),
DOB date)""")

```

Notice that there is a mix of types (strings, ints) though we still only use %s.

```

sql = "INSERT INTO Students (RollNo,Name,Class,Section,DOB) VALUES
(%s, %s, %s, %s, %s)"

```

```

vals = [
    (21603 , 'wendy' , 11 , 'C1' , '2002-06-08' ),
    (20056 , 'Sneha Kumari' , 11 , 'B1' , '2003-11-20' ),
    (20006 , 'Shreya Jha' , 12 , 'D2' , '2000-01-14' ),
    (20567 , 'Riya Khan' , 12 , 'A4' , '2001-05-13' ),
    (25326 , 'Diya Gupta' , 12 , 'D1' , '2002-08-04' ),
    (26117 , 'Isha Banerjee' , 11 , 'C2' , '2002-12-29' ),
    (20001 , 'Tisha Roy' , 11 , 'A1' , '2003-11-23' ),
    (29706 , 'Freddie' , 12 , 'A2' , '2001-08-21' ),
    (20127 , 'Sayan Ray' , 12 , 'B1' , '2002-03-30' ),
    (20156 , 'Arnab Patra' , 11 , 'B4' , '2003-09-15' ),
    (20181 , 'Aaditri Khan' , 12 , 'B2' , '2001-08-05' ),
    (21111 , 'Asmita Maity' , 12 , 'B3' , '2002-03-21' ),
    (22121 , 'Avighna Ghosh' , 11 , 'B2' , '2003-04-17' ),
    (21129 , 'Divyanshu Ojha' , 12 , 'B1' , '2001-03-01' ),
    (22138 , 'Rishav Mukherjee' , 11 , 'A4' , '2004-05-23' ),

```

```
(21139, 'Aarush Gupta', 12, 'A1', '2001-03-16' ),
(23144, 'Arit Samanta', 12, 'A2', '2002-08-15' ),
(21170, 'Darshan Bose', 11, 'A2', '2003-11-16' ),
(20175, 'Gourav Chanda', 11, 'A3', '2004-08-24' ),
(24234, 'Divyanshu Singh Suman', 11, 'D1', '2002-12-10' ),
(25220, 'Vedanshi Kumari', 11, 'A2', '2002-07-25' ),
(21287, 'Kavya Jha', 12, 'C2', '2001-06-01' ),
(26228, 'Aayoshi Pathak', 11, 'C1', '2003-08-20' ),
(26276, 'Aritra Chatterjee', 12, 'C2', '2002-11-13' ),
(24284, 'Aranna Banerjee', 11, 'C1', '2003-12-28' ),
(31313, 'Bhumika Dawn', 12, 'D1', '2002-09-24' ),
(32317, 'Sreetama Das', 11, 'D2', '2003-06-11' ),
(30227, 'Falguni Ray', 11, 'D1', '2002-01-21' ),
(30256, 'Jayanta Kumar Patrat', 12, 'C3', '2001-02-09' ),
(30281, 'Swapnotonoo Khan', 12, 'C2', '2002-09-03' ),
(31211, 'Sourav Maity', 11, 'C1', '2003-11-10' ),
(32221, 'Bamdev Ghosh', 11, 'A2', '2003-04-26' ),
(31229, 'Rabindra Kumar', 11, 'A3', '2003-10-12' ),
(32238, 'Ritam Mukherjee', 11, 'A2', '2004-11-19' ),
(31239, 'Amar Nath Gupta', 11, 'A1', '2004-08-21' ),
(33244, 'Chayan Samanta', 12, 'B1', '2002-12-24' ),
(31270, 'Debashis Bose', 11, 'B2', '2003-10-04' ),
(30275, 'Manoj Chanda', 12, 'B4', '2002-04-30' ),
(34214, 'Sumanjay Kumar Suman', 11, 'B3', '2004-12-07' ),
(35226, 'Pinku Prasad', 12, 'A4', '2002-10-21' ),
(31227, 'Sanjeet Kumar', 11, 'A1', '2003-05-29' ),
(36228, 'Anirban Pathak', 12, 'B1', '2001-01-27' ),
(36276, 'Avinandan Chatterjee', 11, 'B1', '2003-11-18' ),
(34284, 'Debi Prosad Banerjee', 12, 'B2', '2002-06-29' ),
(31213, 'Pujan Kumar Dawn', 12, 'A4', '2001-08-10' ),
(32217, 'Goutam Das', 11, 'A3', '2004-05-01' )
```

```
]
Cursor.executemany(sql,vals)
```

```
# *****
# ***Use database named MyLib***
# *****
```

```
def UseExistingDatabase(Cursor):
    Cursor.execute("use MyLib")
```

```
#
# *****End of file*****
```