

BÀI 8: KẾT NỐI MY SQL VỚI PYTHON

Cài đặt môi trường MySQL cho Python

Để xây dựng các ứng dụng, kết nối với cơ sở dữ liệu là điều cần thiết cho các ngôn ngữ lập trình. Python cho phép chúng ta kết nối ứng dụng với các cơ sở dữ liệu như MySQL, SQLite, MongoDB và nhiều ứng dụng khác.

Bài này chúng ta sẽ tìm hiểu về cài đặt môi trường MySQL cho Python.

Nội dung chính

- [Cài đặt mysql.connector](#)
- [Cách 1: sử dụng trình cài đặt pip](#)
- [Cách 2: cài đặt phần mềm mysql.connector](#)
- [Kiểm tra kết nối Python với MySQL](#)

Cài đặt mysql.connector

Để kết nối ứng dụng python với cơ sở dữ liệu MySQL, chúng ta phải import mô-đun mysql.connector trong chương trình.

Mysql.connector không phải là mô-đun tích hợp đi kèm với cài đặt python. Chúng ta cần cài đặt nó như sau:

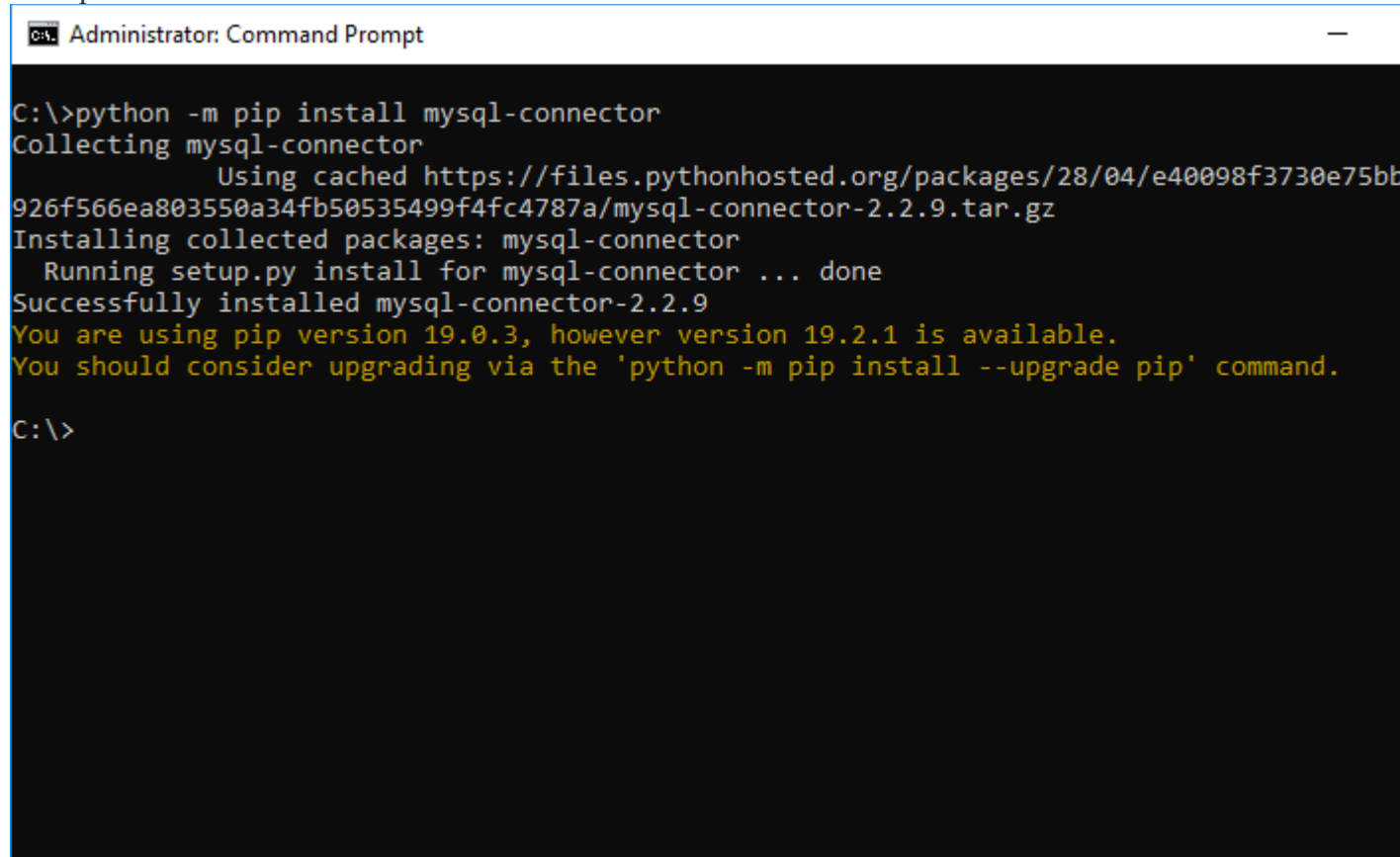
Cách 1: sử dụng trình cài đặt pip

Thực hiện lệnh sau để cài đặt nó bằng trình cài đặt pip:

?

```
1 > python -m pip install mysql-connector
```

Kết quả:



```
Administrator: Command Prompt

C:\>python -m pip install mysql-connector
Collecting mysql-connector
  Using cached https://files.pythonhosted.org/packages/28/04/e40098f3730e75bb926f566ea803550a34fb50535499f4fc4787a/mysql-connector-2.2.9.tar.gz
Installing collected packages: mysql-connector
  Running setup.py install for mysql-connector ... done
Successfully installed mysql-connector-2.2.9
You are using pip version 19.0.3, however version 19.2.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\>
```

Hoặc làm theo các bước sau:

Cách 2: cài đặt phần mềm mysql.connector

1. Open link sau và tải file driver tương ứng cho hệ điều hành của bạn.

<https://dev.mysql.com/downloads/connector/python/>

2. Cài đặt file vừa tải về.

Ví dụ, cấu hình máy tính của mình như sau:

- Hệ điều hành Window 10
- Python 3.7.3

Mình cần tải mysql.connector như trong ảnh sau:

The screenshot shows the MySQL Connector/Python 8.0.17 download page. The left sidebar contains links to Connector/C, MySQL Native Driver for PHP, and Other Downloads. The main content area has a 'Select Operating System' dropdown set to 'Microsoft Windows' and a 'Select OS Version' dropdown set to 'All'. Below these is a 'Recommended Download' section for the MySQL Installer for Windows. A red arrow points to the 'Other Downloads' section, which contains a table of download links.

Other Downloads:	
Windows (x86, 32-bit), MSI Installer Python 3.7 (mysql-connector-python-8.0.17-py3.7- windows-x86-32bit.msi)	8.0.17 MD5: 6ca9e...
Windows (x86, 64-bit), MSI	8.0.17

Kiểm tra kết nối Python với MySQL

Ví dụ 1 được sử dụng để kiểm tra kết nối Python với MySQL:

[?](#)

```

1 import mysql.connector
2     #Create the connection object
3 myconn = mysql.connector.connect(host = "localhost", user = "root",
4     passwd = "123456")
5
6     #printing the connection object
7     print(myconn)
8

```

Kết quả:

?

```

1 <mysql.connector.connection.MySQLConnection object at 0x002DF810>

```

II. Kết nối Python với MySQL

Nội dung chính

- Các bước để kết nối Python với MySQL Server.
 - 1. Import mô-đun mysql.connector
 - 2. Tạo đối tượng kết nối Python với MySQL
 - 3. Tạo một đối tượng con trỏ
 - 4. Thực hiện truy vấn

Các bước để kết nối Python với MySQL Server.

Đầu tiên các bạn cần phải **Cài đặt môi trường MySQL cho Python**

1. Import mô-đun mysql.connector
2. Tạo đối tượng kết nối.
3. Tạo đối tượng con trỏ.
4. Thực hiện truy vấn.

2. 1. Import mô-đun mysql.connector

Cú pháp:

?

```

1 import mysql.connector

```

2. Tạo đối tượng kết nối Python với MySQL

Phương thức connect() của mô-đun mysql.connector được sử dụng để tạo đối tượng connection kết nối giữa cơ sở dữ liệu MySQL và ứng dụng python.

Truyền các thông tin của database như HostName, username và password, cú pháp như sau:

?

```

1 Connection-Object= mysql.connector.connect(host = <host-name>,
2     user = <username>, passwd = <password>)

```

Ví dụ2 tạo đối tượng connection:

?

```

1 import mysql.connector
2
3     # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "123456")
6
7     # in đối tượng connection ra màn hình
8     print(myconn)

```

Kết quả:

?

```
1 <mysql.connector.connection.MySQLConnection object at 0x010DF810>
```

Ở đây, chúng ta phải lưu ý rằng chúng ta có thể chỉ định tên cơ sở dữ liệu trong phương thức connect() nếu chúng ta muốn kết nối với một cơ sở dữ liệu cụ thể.

Ví dụ 3:

```
?
```

```
1 import mysql.connector
2
3 # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "123456", database = "mydb")
6
7 # in đối tượng connection ra màn hình
8 print(myconn)
```

Kết quả:

```
?
```

```
1 <mysql.connector.connection.MySQLConnection object at 0x010FF870>
```

3. Tạo một đối tượng con trỏ

Đối tượng con trỏ có thể được định nghĩa là một sự trừu tượng hóa được chỉ định trong Python DB-API 2.0. Nó tạo điều kiện cho chúng ta có nhiều môi trường làm việc riêng biệt thông qua cùng một kết nối với cơ sở dữ liệu. Chúng ta có thể tạo đối tượng con trỏ bằng cách gọi hàm cursor() của đối tượng connection. Đối tượng con trỏ là một khía cạnh quan trọng của việc thực hiện các truy vấn đến cơ sở dữ liệu.

Cú pháp:

```
?
```

```
1 <my_cur> = conn.cursor()
```

Ví dụ 4: tạo đối tượng con trỏ:

```
?
```

```
1 import mysql.connector
2
3 # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "123456", database = "mydb")
6
7 # in đối tượng connection ra màn hình
8 print(myconn)
9
10 # tạo đối tượng cursor
11 cur = myconn.cursor()
12
13 # in đối tượng cursor ra màn hình
14 print(cur)
```

```
?
```

```
1 <mysql.connector.connection.MySQLConnection object at 0x010BF9B0>
```

```
2 MySQLCursor: (Nothing executed yet)
```

4. Thực hiện truy vấn

Tạo new database MySQL trong Python

Trong bài hướng dẫn này, chúng ta sẽ tạo cơ sở dữ liệu mới có tên PythonDB.

Nội dung chính

- Lấy danh sách các cơ sở dữ liệu hiện có
- Tạo new database MySQL trong Python

Lấy danh sách các cơ sở dữ liệu hiện có

Chúng tôi có thể lấy danh sách tất cả các cơ sở dữ liệu bằng cách sử dụng truy vấn MySQL sau đây.

?

```
1 show databases;
```

Ví dụ 5:

?

```
1 import mysql.connector
2
3 # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "123456")
6
7 # tạo đối tượng cursor
8 cur = myconn.cursor()
9
10 try:
11     dbs = cur.execute("show databases")
12 except:
13     myconn.rollback()
14 for x in cur:
15     print(x)
16 myconn.close()
```

Kết quả:

```
('information_schema',)
('mydb',)
('mysql',)
('performance_schema',)
('phpmyadmin',)
('test',)
```

Tạo new database MySQL trong Python

Cơ sở dữ liệu mới có thể được tạo bằng cách sử dụng truy vấn SQL sau.

?

```
1 create database <database-name>
```

Ví dụ 6 tạo new database MySQL trong Python:

?

```
1 import mysql.connector
2
3 # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "123456")
6
7 # tạo đối tượng cursor
8 cur = myconn.cursor()
9
```

```

10     try:
11         cur.execute("create database PythonDB")
12         dbs = cur.execute("show databases")
13     except:
14         myconn.rollback()
15     for x in cur:
16         print(x)
17     myconn.close()

```

Kết quả:

```

('information_schema',)
('mydb',)
('mysql',)
('performance_schema',)
('phpmyadmin',)
('pythondb',)
('test',)

```

Tạo bảng MySQL trong Python

Nội dung chính

- Create table
- Alter table
- Create table
- Trong bài này, chúng ta sẽ tạo ra một bảng MySQL mới có tên là Employee. Chúng ta có thể tạo bảng mới bằng cách sử dụng câu lệnh CREATE TABLE của SQL. Bảng Employee sẽ có bốn cột bao gồm name, id, salary và department_id.
- Câu lệnh SQL sau được sử dụng để tạo bảng Employee:

```

1 create table Employee (
2     name varchar(20) not null,
3     id int primary key,
4     salary float not null,
5     dept_Id int not null
6 )

```

- *Ví dụ 7:*
-

```

1 import mysql.connector
2
3 # tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "1234567890", database = "PythonDB")
6
7 # tạo đối tượng cursor
8 cur = myconn.cursor()
9
10 try:
11     # tạo bảng Employee gồm 4 cột name, id, salary, và department id
12     dbs = cur.execute("create table Employee(name varchar(20) not null, "
13         + "id int(20) not null primary key, salary float not null, "
14         + "dept_id int not null)")
15 except:

```

16 myconn.rollback()

17

18 myconn.close()

- Kết quả:



#	Tên	Kiểu	Bảng mã đối chiếu	Thuộc tính	Null	Mặc định	Ghi chú	Thêm	Hành động
1	name	varchar(20)	latin1_swedish_ci		Không	Không			Thay đổi Xóa Thêm
2	id	int(20)			Không	Không			Thay đổi Xóa Thêm
3	salary	float			Không	Không			Thay đổi Xóa Thêm
4	dept_id	int(11)			Không	Không			Thay đổi Xóa Thêm

-
-
-

Alter table

- Đôi khi, chúng ta có thể quên tạo một số cột hoặc chúng tôi có thể cần cập nhật table schema. Câu lệnh ALTER được sử dụng trong trường hợp này. Ở đây, chúng ta sẽ thêm cột tên_bảng vào bảng Employee như sau.

- ?

1 alter table Employee

2 add branch_name varchar(20) not null

- Ví dụ 8:

- ?

1 import mysql.connector

2

3 # tạo đối tượng connection

4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5 passwd = "1234567890", database = "PythonDB")

6

7 # tạo đối tượng cursor

8 cur = myconn.cursor()

9

10 try:

11 # thêm cột branch name vào bảng Employee

12 cur.execute("alter table Employee add branch_name varchar(20) not null")

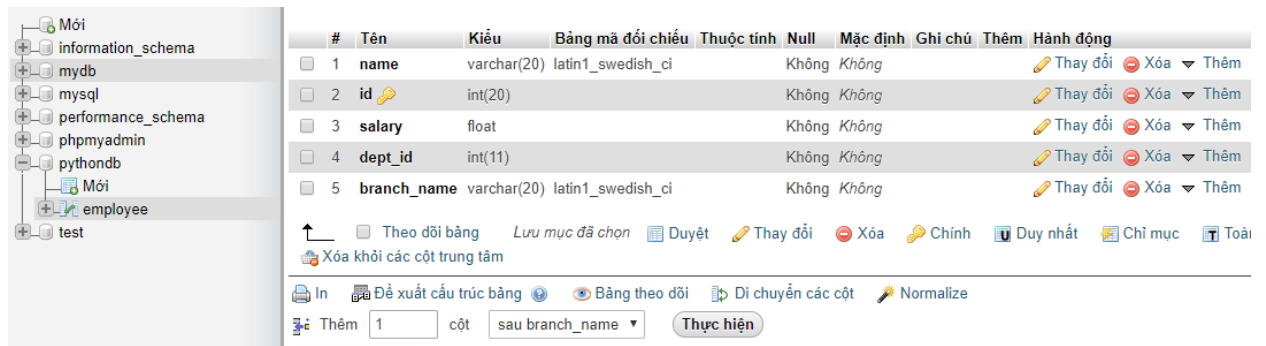
13 except:

14 myconn.rollback()

15

16 myconn.close()

- Kết quả:



Lệnh INSERT MySQL trong Python

âu lệnh INSERT INTO được sử dụng để thêm một bản ghi vào bảng. Trong python, chúng ta có thể sử dụng (%) thay cho các giá trị.

Nội dung chính

- INSERT 1 bản ghi
- INSERT nhiều bản ghi

INSERT 1 bản ghi

Phương thức cursor.execute() được sử dụng để insert 1 bản ghi vào table.

Chúng ta cung cấp các giá trị thực tế trong một tuple trong phương thức cursor.execute().

Ví dụ 9

```

1  import mysql.connector
2
3  #tạo đối tượng connection
4  myconn = mysql.connector.connect(host = "localhost", user = "root",
5      passwd = "1234567890", database = "PythonDB")
6
7  #tạo đối tượng cursor
8  cur = myconn.cursor()
9  sql = ("insert into Employee(name, id, salary, dept_id, branch_name) "
10      + "values (%s, %s, %s, %s, %s)")
11
12  #giá trị của một row được cung cấp dưới dạng tuple
13  val = ("The Mac", 10001, 25000.00, 101, "Hanoi")
14
15  try:
16      #inserting the values into the table
17      cur.execute(sql,val)
18
19      #commit the transaction
20      myconn.commit()
21
22  except:
23      myconn.rollback()
24
25  print(cur.rowcount,"record inserted!")
26  myconn.close()

```

Kết quả:

Message	Result1	Profile	Status		
	name	id	salary	dept_id	branch_name
► The Mac		10001	25000	101	Hanoi

INSERT nhiều bản ghi

Phương thức `cursor.executemany()` được sử dụng để insert 1 bản ghi vào table.

Ví dụ 10

```

1  import mysql.connector
2
3  #tạo đối tượng connection
4  myconn = mysql.connector.connect(host = "localhost", user = "root",
5      passwd = "1234567890", database = "PythonDB")
6
7  #tạo đối tượng cursor
8  cur = myconn.cursor()
9  sql = ("insert into Employee(name, id, salary, dept_id, branch_name) "
10      "values (%s, %s, %s, %s, %s)")
11
12  #giá trị của một row được cung cấp dưới dạng tuple
13  val = [("Vinh", 10002, 26000.00, 101, "Hanoi"),
14      ("Trung", 10003, 26000.00, 102, "Danang")]
15
16  try:
17      #inserting the values into the table
18      cur.executemany(sql, val)
19
20      #commit the transaction
21      myconn.commit()
22
23  except:
24      myconn.rollback()
25
26  print(cur.rowcount,"record inserted!")
27  myconn.close()

```

Kết quả:

Message	Result1	Profile	Status		
name	id	salary	dept_id	branch_name	
► The Mac	10001	25000	101	Hanoi	

Lệnh SELECT MySQL trong Python

Câu lệnh SELECT được sử dụng để đọc các giá trị từ cơ sở dữ liệu.

Nội dung chính

- Phương thức `fetchall()`
- Phương thức `fetchone()`

- Định dạng kết quả

Phương thức fetchall()

Python cung cấp phương thức fetchall() trả về dữ liệu được lưu trữ bên trong bảng dưới dạng các hàng. Chúng ta có thể lặp lại kết quả để có được các hàng riêng lẻ.

Ví dụ 11

[?](#)

```
1 import mysql.connector
2
3 #tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "1234567890", database = "PythonDB")
6
7 #tạo đối tượng cursor
8 cur = myconn.cursor()
9
10 try:
11     # select dữ liệu từ database
12     cur.execute("SELECT * FROM Employee")
13
14     # tìm nạp các hàng từ đối tượng con trỏ
15     result = cur.fetchall()
16
17     for x in result:
18         print(x);
19
20 except:
21     myconn.rollback()
22
23 myconn.close()
```

Kết quả:

```
('The Mac', 10001, 25000.0, 101, 'Hanoi')
('Vinh', 10002, 26000.0, 101, 'Hanoi')
('Trung', 10003, 26000.0, 102, 'Danang')
('Tân', 10004, 10000.0, 101, 'Hanoi')
('Quân', 10005, 30000.0, 103, 'HCM')
```

Ví dụ 12: đọc cột cụ thể

[?](#)

```
1 import mysql.connector
2
3 #tạo đối tượng connection
4 myconn = mysql.connector.connect(host = "localhost", user = "root",
5     passwd = "1234567890", database = "PythonDB")
6
7 #tạo đối tượng cursor
8 cur = myconn.cursor()
9
10 try:
11     # select dữ liệu từ database
```

```

12     cur.execute("SELECT name, id, salary FROM Employee")
13
14     # tìm nạp các hàng từ đối tượng con trỏ
15     result = cur.fetchall()
16
17     for x in result:
18         print(x);
19
20 except:
21     myconn.rollback()
22
23     myconn.close()

```

Kết quả:

```

('The Mac', 10001, 25000.0)
('Vinh', 10002, 26000.0)
('Trung', 10003, 26000.0)
('Tân', 10004, 10000.0)
('Quân', 10005, 30000.0)

```

Phương thức fetchone()

Phương thức fetchone() được sử dụng để chỉ tìm nạp một hàng từ bảng. Phương thức fetchone () trả về hàng tiếp theo của tập kết quả.

Ví dụ13:

```

1     import mysql.connector
2
3     #tạo đối tượng connection
4     myconn = mysql.connector.connect(host = "localhost", user = "root",
5         passwd = "1234567890", database = "PythonDB")
6
7     #tạo đối tượng cursor
8     cur = myconn.cursor()
9
10    try:
11        # select dữ liệu từ database
12        cur.execute("SELECT name, id, salary FROM Employee")
13
14        # tìm nạp hàng đầu tiên từ đối tượng con trỏ
15        result = cur.fetchone()
16        print(result);
17
18        # tìm nạp hàng tiếp theo từ đối tượng con trỏ
19        result = cur.fetchone()
20        print(result);
21
22    except:

```

```

23     myconn.rollback()
24
25     myconn.close()

```

Kết quả:

```

('The Mac', 10001, 25000.0)
('Vinh', 10002, 26000.0)

```

Định dạng kết quả

Ví dụ 14:

?

```

1     import mysql.connector
2
3     #tạo đối tượng connection
4     myconn = mysql.connector.connect(host = "localhost", user = "root",
5         passwd = "1234567890", database = "PythonDB")
6
7     #tạo đối tượng cursor
8     cur = myconn.cursor()
9
10    try:
11        # select dữ liệu từ database
12        cur.execute("SELECT name, id, salary FROM Employee")
13
14        # tìm nạp các hàng từ đối tượng con trỏ
15        result = cur.fetchall()
16
17        print("Name   ID   Salary")
18
19        for row in result:
20            print("%s   %d   %d"%(row[0],row[1],row[2]))
21
22    except:
23        myconn.rollback()

```

Kết quả:

```

Name  ID   Salary
The Mac  10001  25000
Vinh   10002  26000
Trung  10003  26000
Tân    10004  10000
Quân   10005  30000

```

Lệnh UPDATE MySQL trong Python

Lệnh UPDATE-SET được sử dụng để cập nhật giá trị của một hoặc nhiều cột trong table. Câu lệnh sau đây được sử dụng để cập nhập một cột.

?

```

1     update Employee set name = 'Đạt' where id = 10001

```

Ví dụ 15:

?

```

1     import mysql.connector
2

```

```

3      #tạo đối tượng connection
4      myconn = mysql.connector.connect(host = "localhost", user = "root",
5          passwd = "1234567890", database = "PythonDB")
6
7      #tạo đối tượng cursor
8      cur = myconn.cursor()
9
10     try:
11         # cập nhật name cho bảng Employee
12         cur.execute("update Employee set name = 'Đạt' where id = 10001")
13         myconn.commit()
14
15     except:
16         myconn.rollback()
17
18     myconn.close()

```

Lệnh DELETE MySQL trong Python

ệnh DELETE FROM được sử dụng để xóa một bản ghi được chỉ định từ table. Ở đây, chúng ta phải sử dụng câu điều kiện WHERE để chỉ định bản ghi mà bạn muốn xóa, nếu không thì tất cả sẽ bị xóa.

Câu lệnh sql sau được sử dụng để xóa employee có id là 110:

```

1      delete from Employee where id = 10001

```

Ví dụ 16:

```

1      import mysql.connector
2
3      #tạo đối tượng connection
4      myconn = mysql.connector.connect(host = "localhost", user = "root",
5          passwd = "1234567890", database = "PythonDB")
6
7      #tạo đối tượng cursor
8      cur = myconn.cursor()
9
10     try:
11         # cập nhật name cho bảng Employee
12         cur.execute("delete from Employee where id = 10001")
13         myconn.commit()
14
15     except:
16         myconn.rollback()
17
18     myconn.close()

```