



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

DATABASE AND SQL FOR DATA SCIENCE

Truy cập Database với Python

Phòng LT & Mạng

<http://csc.edu.vn/kiem-thu-phan-mem>

2019



**Data
Science**

- 1. Python và Database**
- 2. Sử dụng DB-API**
- 3. Sử dụng Python code thao tác dữ liệu**
- 4. Sử dụng Magic SQL (%)**

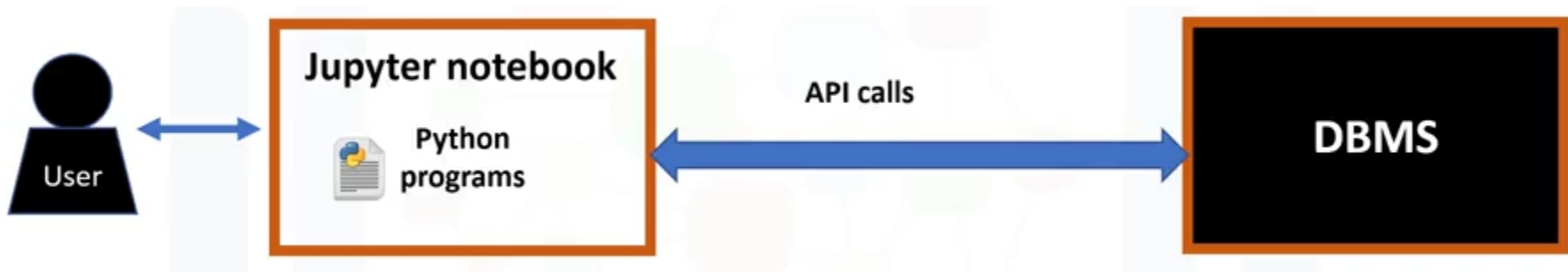
❑ Ưu điểm của Python khi làm việc với DB:

- Cực mạnh trong việc xử lý các loại dữ liệu chuỗi, tập hợp. Thích hợp với ứng dụng bóc tách, chuyển đổi, phân tích dữ liệu: big data - data mining
- Cung cấp nhiều thư viện hỗ trợ khi làm việc với các loại DB, đặc biệt là các thư viện rất mạnh trong việc xử lý dữ liệu: NumPy, Pandas, Matplotlib, SciPy,...

- ❑ Ưu điểm của Python khi làm việc với DB:
 - Dễ Code, dễ cài đặt
 - Có thể chạy trên nhiều nền tảng hệ điều hành khác nhau
 - Hỗ trợ tất cả các loại cơ sở dữ liệu quan hệ
 - Tài liệu hướng dẫn chi tiết, đội ngũ developer đông đảo

❑ Mô hình truy cập DB:

- Python truy cập DB thông qua việc gọi các API (Application Programming Interface) tương ứng với từng loại DB
- API: là 1 tập hợp các hàm được cung cấp để có thể gọi để có quyền truy cập vào hệ thống.

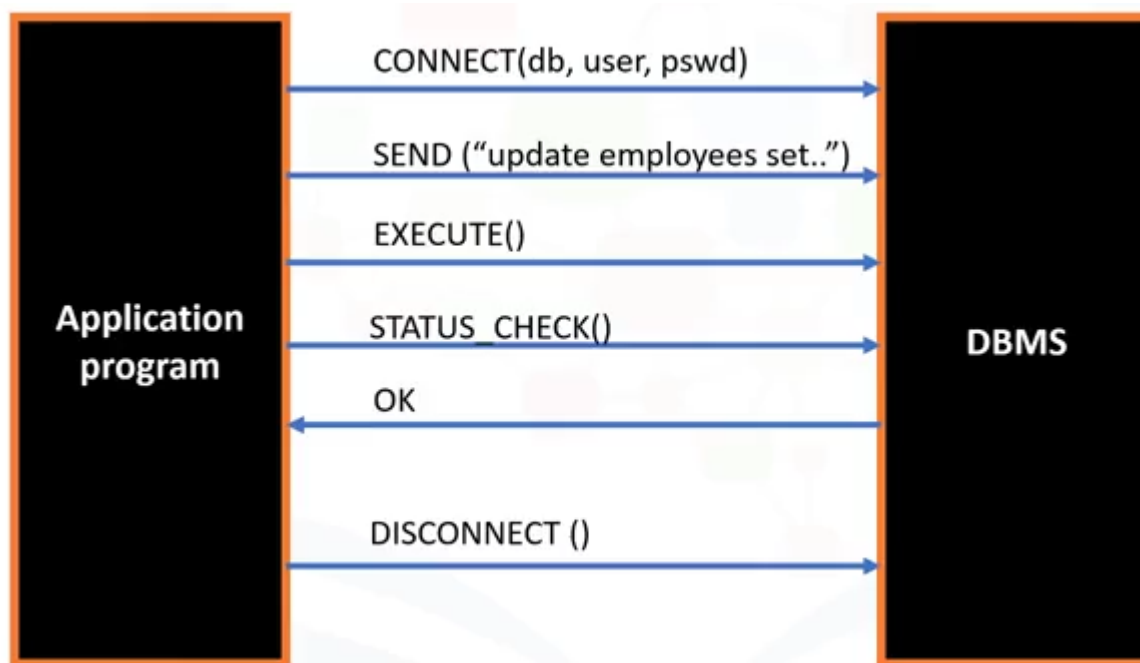


❑ SQL APIs:

- SQL APIs là tập hợp các API được cung cấp để truy cập vào DBMS
- Để truyền các câu lệnh SQL cho DBMS, chương trình ứng dụng gọi các hàm trong API và nó gọi các hàm khác để lấy kết quả truy vấn và thông tin trạng thái từ DBMS.

□ SQL APIs:

- Hoạt động cơ bản của SQL APIs:



❑ SQL APIs:

- APIs được sử dụng trên 1 số DBMS phổ biến:

Application or Database	SQL API
MySQL	MySQL C API
PortgreSQL	psycopg2
IBM DB2	ibm_db
SQL Server	dblib API
Database access for Microsoft Windows OS	ODBC
Oracle	OCI
Java	JDBC

1. Python và Database
- 2. Sử dụng DB-API**
3. Sử dụng Python code thao tác dữ liệu
4. Sử dụng Magic SQL (%)

Sử dụng DB-API

- ❑ DB-API là thư viện chuẩn của Python dùng để truy cập cơ sở dữ liệu quan hệ.
- ❑ DB-API cho phép Python kết nối đến bất kỳ loại cơ sở dữ liệu quan hệ nào

Sử dụng DB-API

□ Ưu điểm khi sử dụng DB-API

- Dễ hiểu, dễ cài đặt
- Được định nghĩa tương tự các module của Python dùng để truy cập cơ sở dữ liệu
- Đảm bảo tính nhất quán → module dễ hiểu, dễ sử dụng
- Không phụ thuộc vào bất kỳ DB nào

❑ Một số thư viện thường dùng để kết nối:

Database	DB API
DB2 Warehouse on Cloud	ibm_db
MySQL	MySQL Connector/ Python
PostgreSQL	psycopg2
MongoDB	PyMongo
SQL Server	pymssql

❑ Các đối tượng chính trong Python DB API

- Đối tượng Connection

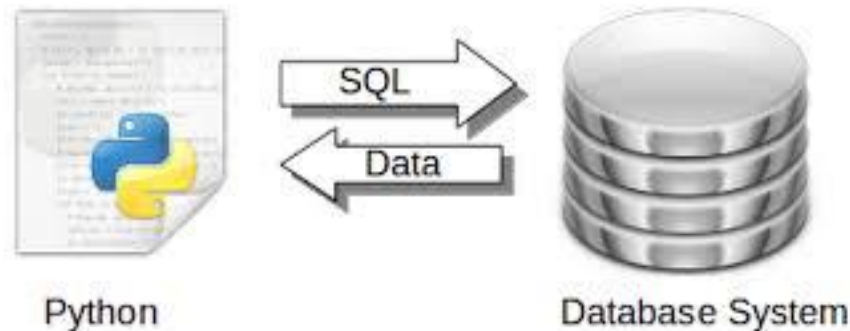
- Thực hiện kết nối đến Database
- Quản lý các phiên (session) làm việc

- Đối tượng Cursor

- Thực hiện các truy vấn dữ liệu thông qua Connection

□ Các phương thức của Connection

- cursor(): trả về 1 đối tượng cursor (sử dụng cursor để thực hiện lệnh)
- commit(): dùng để cập nhật các thay đổi về Database
- rollback(): hủy bỏ các thay đổi, không cập nhật về Database
- close(): n



- ❑ Cài đặt thư viện pymysql kết nối đến **MySQL** bằng lệnh:

```
pip install mysql-connector-python
```

- ❑ Tạo kết nối đến Database

```
from mysql.connector import connect
```

```
# Open database connection
```

```
db = connect(host, user, password, database[, port])
```

❑ Tạo kết nối đến Database MySQL

```
from mysql.connector import connect, Error

# Open database connection
connection = connect(host='localhost', database='testdb', user='root', password='')

if connection.is_connected():
    db_Info = connection.get_server_info()
    print("Connected to MySQL database... MySQL Server version on ",db_Info)

    # prepare a cursor object using cursor() method
    cursor = connection.cursor()
    cursor.execute("select database();")

    # Fetch a single row using fetchone() method.
    record = cursor.fetchone()
    print ("Your connected to - ", record)

if(connection.is_connected()):
    # disconnect from server
    cursor.close()
    connection.close()
    print("MySQL connection is closed")
```

```
Connected to MySQL database... MySQL Server version on 5.7.19
Your connected to - ('testdb',)
MySQL connection is closed
```


Sử dụng DB-API

- ❑ Cài đặt thư viện pymssql kết nối đến **SQL Server** bằng lệnh:

```
pip install pymssql
```

- ❑ Tạo kết nối đến Database

```
from pymssql import connect
```

```
# Open database connection
```

```
db = connect(server, user, password, database[, port])
```

❑ Tạo kết nối đến Database SQL Server

```
import pymssql as sql
```

```
sql.connect(server='127.0.0.1',user='sa',  
            password='123456',database='BANHANG')
```

```
<pymssql.Connection at 0x1b66c660708>
```

Sử dụng DB-API

□ Các phương thức của Cursor

- `callproc():`
- `execute():`
- `executemany():`
- `fetchone():`
- `fetchmany():`
- `fetchall():`
- `close():`

❑ Các phương thức của Cursor

- callproc(): thi hành các stored procedure trong Database
- Tạo stored procedure:

```
DELIMITER $$  
CREATE PROCEDURE find_customer_by_phone(IN p_phone VARCHAR(14),  
    OUT p_firstname VARCHAR(255), OUT p_lastname VARCHAR(255))  
BEGIN  
    SELECT firstname, lastname INTO p_firstname, p_lastname  
    FROM customers  
    WHERE phone = p_phone;  
END $$  
DELIMITER ;
```

□ Các phương thức của Cursor

- callproc(): thi hành các stored procedure trong Database
- Gọi stored procedure:

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

args = ['(171) 555-7788', None, None]
result_args = cursor.callproc('find_customer_by_phone', args)

print(result_args[1], result_args[2])

cursor.close()
conn.close()
```

Thomas Hardy

□ Các phương thức của Cursor

- `execute()`: thực thi câu lệnh, truy vấn dữ liệu

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

cursor.execute("select * from customers where City = 'London'")

# iterate over result
for row in cursor:
    print(row)

cursor.close()
conn.close()
```

```
(4, 'Thomas', 'Hardy', 'London', 'UK', '(171) 555-7788')
(11, 'Victoria', 'Ashworth', 'London', 'UK', '(171) 555-1212')
(16, 'Elizabeth', 'Brown', 'London', 'UK', '(171) 555-2282')
(19, 'Ann', 'Devon', 'London', 'UK', '(171) 555-0297')
(53, 'Simon', 'Crowther', 'London', 'UK', '(171) 555-7733')
(72, 'Hari', 'Kumar', 'London', 'UK', '(171) 555-1717')
```

□ Các phương thức của Cursor

- execute(): thực thi nhiều câu lệnh, truy vấn dữ liệu

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

sql1 = "select version()"
sql2 = "select * from customers where City=%s limit %s"
sql3 = "select 1+1"

queries = [sql1, sql2, sql3]
data = ('London', 4)

results = cursor.execute(";".join(queries), data, multi=True)
count = 1

for result in results:
    print("Query {0} - {1} :".format(count, result.statement))
    if result.with_rows:
        for row in result:
            print(row)
            count = count + 1
    else:
        print("No result found")
    print()

cursor.close()
conn.close()
```

□ Các phương thức của Cursor

- `execute()`: thực thi nhiều câu lệnh, truy vấn dữ liệu

```
Query 1 - select version() :  
( '5.7.19', )
```

```
Query 2 - select * from customers where City='London' limit 4 :  
(4, 'Thomas', 'Hardy', 'London', 'UK', '(171) 555-7788')  
(11, 'Victoria', 'Ashworth', 'London', 'UK', '(171) 555-1212')  
(16, 'Elizabeth', 'Brown', 'London', 'UK', '(171) 555-2282')  
(19, 'Ann', 'Devon', 'London', 'UK', '(171) 555-0297')
```

```
Query 3 - select 1+1 :  
(2, )
```


□ Các phương thức của Cursor

- `execute()`: thực thi câu lệnh với nhiều bộ dữ liệu:

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

sql = """update products set UnitPrice = %(unit_price)s
where ProductName = %(name)s and SupplierId = %(sid)s"""

data_list = [
    {'unit_price': 20.0, 'sid': 1, 'name': 'Chang'},
    {'unit_price': 31.5, 'sid': 4, 'name': 'Ikura'},
    {'unit_price': 23.5, 'sid': 6, 'name': 'Tofu'},
]

for data in data_list:
    cursor.execute(sql, data)

# db.commit() # call commit() method to save the changes
# db.rollback() # call rollback() method to unsave the changes

cursor.close()
conn.close()
```

❑ Các phương thức của Cursor

- `executemany()`: thực thi câu lệnh với nhiều bộ dữ

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

sql = """update products set UnitPrice = %(unit_price)s
where ProductName = %(name)s and SupplierId = %(sid)s"""

data_list = [
    {'unit_price': 20.0, 'sid': 1, 'name': 'Chang'},
    {'unit_price': 31.5, 'sid': 4, 'name': 'Ikura'},
    {'unit_price': 23.5, 'sid': 6, 'name': 'Tofu'},
]

cursor.executemany(sql, data_list)

print("Rowcount:", cursor.rowcount) # rows updated

cursor.close()
conn.close()
```

Rowcount: 3

Sử dụng DB-API

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

sql1 = """
create temporary table customer(
    id int AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    age SMALLINT NOT NULL
)"""

cursor.execute(sql1)

sql2 = "insert into customer(name, age) VALUES (%(name)s, %(age)s)"

data_list = [
    {'name': 'John', 'age': 45 }, {'name': 'Max', 'age': 25 }, {'name': 'Jane', 'age': 20 }, {'name': 'Bob', 'age': 34 },
]

cursor.executemany(sql2, data_list)

print("Query:", cursor.statement)
print("Rowcount:", cursor.rowcount) # rows inserted

conn.commit() # commit the changes

cursor.close()
conn.close()
```

Query: insert into customer(name, age) VALUES ('John', 45),('Max', 25),('Jane', 20),('Bob', 34)
Rowcount: 4

□ Các phương thức của Cursor

- `fetchone()`: trả về dòng tiếp theo trong tập kết quả. Nếu không có dòng dữ liệu sẽ trả về `None`

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

r = cursor.execute("select * from customers limit 5")

# first row
row = cursor.fetchone()

# loop over the remaining result set
while row:
    print(row)
    row = cursor.fetchone()

cursor.close()
conn.close()
```

```
(1, 'Maria', 'Anders', 'Berlin', 'Germany', '030-0074321')
(2, 'Ana', 'Trujillo', 'México D.F.', 'Mexico', '(5) 555-4729')
(3, 'Antonio', 'Moreno', 'México D.F.', 'Mexico', '(5) 555-3932')
(4, 'Thomas', 'Hardy', 'London', 'UK', '(171) 555-7788')
(5, 'Christina', 'Berglund', 'Luleå', 'Sweden', '0921-12 34 65')
```

□ Các phương thức của Cursor

- `fetchmany([numRows])`: Trả về số dòng được chỉ định. Nếu không còn dòng nào sẽ trả về [], mặc định `numRows = 1`

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

r = cursor.execute("select * from customers limit 5")

# first row
print(cursor.fetchmany())
print(cursor.fetchmany(2))
print(cursor.fetchmany(2))
print(cursor.fetchmany())

cursor.close()
conn.close()
```

```
[(1, 'Maria', 'Anders', 'Berlin', 'Germany', '030-0074321')]
[(2, 'Ana', 'Trujillo', 'México D.F.', 'Mexico', '(5) 555-4729'), (3, 'Antonio', 'Moreno', 'México D. F.', 'Mexico', '(5) 555-3932')]
[(4, 'Thomas', 'Hardy', 'London', 'UK', '(171) 555-7788'), (5, 'Christina', 'Berglund', 'Luleå', 'Sweden', '0921-12 34 65')]
[]
```

□ Các phương thức của Cursor

- `fetchall()`: trả về tất cả các dòng (hoặc các dòng còn lại) trong tập kết quả

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

r = cursor.execute("select * from customers limit 5")

# first row
print(cursor.fetchone())
print(cursor.fetchall())

cursor.close()
conn.close()
```

```
(1, 'Maria', 'Anders', 'Berlin', 'Germany', '030-0074321')
[(2, 'Ana', 'Trujillo', 'México D.F.', 'Mexico', '(5) 555-4729'), (3, 'Antonio', 'Moreno', 'México D. F.', 'Mexico', '(5) 555-3932'), (4, 'Thomas', 'Hardy', 'London', 'UK', '(171) 555-7788'), (5, 'Christina', 'Berglund', 'Luleå', 'Sweden', '0921-12 34 65')]
```

❑ Các thuộc tính của Cursor

- `column_names`: danh sách cột trả về trong truy vấn
- `with_rows`: trả về True nếu truy vấn có dữ liệu, ngược lại trả về False
- `description`: Trả về thông tin của các cột trong truy vấn bao gồm: `name`, `type_code`, `display_size`, `internal_size`, `precision`, `scale`, `null_ok`

❑ Các thuộc tính của Cursor

- lastrowid: Trả về Id của dòng được Update hoặc Insert cuối cùng hoặc trả về None khi không có dòng Insert hoặc Update. Nếu có nhiều dòng được Insert hoặc Update đồng thời thì trả về Id của dòng đầu tiên
- statement: trả về nội dung câu lệnh truy vấn cuối cùng được thực thi
- rowcount: trả về số dòng dữ liệu có trong truy vấn

□ Các thuộc tính của Cursor

```
from mysql.connector import connect, Error

conn = connect(host='localhost', database='testdb', user='root', password='')
cursor = conn.cursor()

cursor.execute("select * from customers where City='London' limit 5")

print("rowcount (initial):", cursor.rowcount, end='\n\n')

# reading rows but not displaying
for row in cursor: pass

print("column_names:", cursor.column_names, end='\n\n')
print("with_rows:", cursor.with_rows, end='\n\n')

print("description: ", end="")
print(cursor.description)

print("\nstatement:", cursor.statement, end='\n\n')

print("lastrowid:", cursor.lastrowid, end='\n\n')

print("rowcount (final):", cursor.rowcount, end='\n\n')

cursor.close()
conn.close()
```

□ Các thuộc tính của Cursor

```
rowcount (initial): 0
```

```
column_names: ('Id', 'FirstName', 'LastName', 'City', 'Country', 'Phone')
```

```
with_rows: True
```

```
description: [('Id', 3, None, None, None, None, 0, 53251), ('FirstName', 253, None, None, None, None, 0, 4097), ('LastName', 253, None, None, None, None, 0, 4097), ('City', 253, None, None, None, None, 1, 0), ('Country', 253, None, None, None, None, 1, 0), ('Phone', 253, None, None, None, None, 1, 0)]
```

```
statement: select * from customers where City='London' limit 5
```

```
lastrowid: 0
```

```
rowcount (final): 5
```

1. Python và Database
2. Sử dụng DB-API
- 3. Sử dụng Python code thao tác dữ liệu**
4. Sử dụng Magic SQL (%)

❑ Sử dụng Python tạo bảng mới trong Database

- Bước 1: Tham chiếu thư viện mysql connector

```
from mysql.connector import connect, Error
```

- Bước 2: Tạo kết nối đến Database

```
conn = connect(host='localhost', database='testdb',  
               user='root', password='')
```

```
cursor = conn.cursor()
```

❑ Sử dụng Python tạo bảng mới trong Database

- Bước 3: Khai báo câu lệnh tạo bảng

```
sql = """  
create table customers(  
    id int AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    age SMALLINT NOT NULL  
)"""
```

❑ Sử dụng Python tạo bảng mới trong Database

- Bước 4: Thực thi câu lệnh tạo bảng

try:

```
cursor.execute(sql)
```

except Exception as er:

```
print('Lỗi: ', er)
```

else:

```
conn.commit()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python tạo bảng mới trong Database

- Bước 5: Ngắt kết nối đến Database

`cursor.close()`

`conn.close()`

❑ Sử dụng Python xóa bảng trong Database

- Bước 1: Tham chiếu thư viện mysql connector

```
from mysql.connector import connect, Error
```

- Bước 2: Tạo kết nối đến Database

```
conn = connect(host='localhost', database='testdb',  
              user='root', password='')
```

```
cursor = conn.cursor()
```

- Bước 3: Khai báo câu lệnh xóa bảng

```
sql = "drop table customers"
```


❑ Sử dụng Python xóa bảng trong Database

- Bước 4: Thực thi câu lệnh tạo bảng

try:

```
cursor.execute(sql)
```

except Exception as er:

```
print('Lỗi: ', er)
```

else:

```
conn.commit()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python xóa bảng trong Database

- Bước 5: Ngắt kết nối đến Database

`cursor.close()`

`conn.close()`

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm 1 dòng dữ liệu vào bảng

- Bước 1: Tham chiếu thư viện mysql connector

```
from mysql.connector import connect, Error
```

- Bước 2: Tạo kết nối đến Database

```
conn = connect(host='localhost', database='testdb',  
user='root', password='')
```

```
cursor = conn.cursor()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm 1 dòng dữ liệu vào bảng

- Bước 3: Khai báo câu lệnh thêm dữ liệu vào bảng

```
sql = "insert into customers(name, age)
```

```
VALUES (%s, %s)"
```

- Bước 4: khai báo dữ liệu truyền vào tham số câu lệnh
insert

```
data = ('John', 45 )
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm 1 dòng dữ liệu vào bảng

- Bước 5: Thực thi câu lệnh tạo bảng

try:

```
cursor.execute(sql, data)
```

except Exception as er:

```
print('Lỗi: ', er)
```

else:

```
conn.commit()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm 1 dòng dữ liệu vào bảng

- Bước 6: Ngắt kết nối đến Database

`cursor.close()`

`conn.close()`

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm nhiều dòng dữ liệu vào bảng

- Bước 1: Tham chiếu thư viện mysql connector

```
from mysql.connector import connect, Error
```

- Bước 2: Tạo kết nối đến Database

```
conn = connect(host='localhost', database='testdb',  
user='root', password='')
```

```
cursor = conn.cursor()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm nhiều dòng dữ liệu vào bảng

- Bước 3: Khai báo câu lệnh thêm dữ liệu vào bảng

```
sql = "insert into customers(name, age) VALUES  
      %(name)s, %(age)s)"
```

- Bước 4: khai báo dữ liệu truyền vào tham số câu lệnh insert

```
data_list = [  
    {'name': 'John', 'age': 45 }, {'name': 'Max', 'age': 25 },  
    {'name': 'Jane', 'age': 20 }, {'name': 'Bob', 'age': 34 },  
]
```


Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm nhiều dòng dữ liệu vào bảng

- Bước 5: Thực thi câu lệnh tạo bảng

try:

```
cursor.executemany(sql, data_list)
```

except Exception as er:

```
print('Lỗi: ', er)
```

else:

```
conn.commit()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thêm nhiều dòng dữ liệu vào bảng

- Bước 6: Ngắt kết nối đến Database

`cursor.close()`

`conn.close()`

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thực thi câu lệnh truy vấn

- Bước 1: Tham chiếu thư viện mysql connector

```
from mysql.connector import connect, Error
```

- Bước 2: Tạo kết nối đến Database

```
conn = connect(host='localhost', database='testdb',  
user='root', password='')
```

```
cursor = conn.cursor()
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thực thi câu lệnh truy vấn

- Bước 3: Khai báo câu lệnh truy vấn

```
sql = "select * from customers where City=%s limit %s"
```

- Bước 4: khai báo dữ liệu truyền vào tham số câu lệnh
Select (nếu có)

```
data = ('London', 5)
```

Sử dụng Python code thao tác dữ liệu

❑ Sử dụng Python thực thi câu lệnh truy vấn

- Bước 5: Thực thi câu lệnh truy vấn

`cursor.execute(sql, data)`

- Bước 6: Ngắt kết nối đến Database

`cursor.close()`

`conn.close()`

1. Python và Database
2. Sử dụng DB-API
3. Sử dụng Python code thao tác dữ liệu
4. Sử dụng Magic SQL (%)

Sử dụng Magic SQL (%)

❑ Cài đặt thư viện

```
pip install ipython-sql
```

```
pip install pymysql
```

❑ Tham chiếu thư viện magic sql trước khi sử dụng

```
%load_ext sql
```

Sử dụng Magic SQL (%)

❑ Kết nối đến Database MySQL

```
connection_string =  
    "mysql+pymysql://{user}:{password}@{hostname}/  
    {database}?charset=utf8"  
  
%sql $connection_string
```

```
%load_ext sql
```

```
user = 'root'  
password = ''  
connection_string = "mysql+pymysql://{user}:{password}@localhost/testdb?charset=utf8".format(user=user,  
%sql $connection_string
```

```
'Connected: root@testdb'
```


Sử dụng Magic SQL (%)

❑ Kết nối đến Database SQL Server

```
connection_string =  
    "mssql+pymssql://{user}:{password}@{hostname}[:port]  
    /{database}?charset=utf8"  
  
%sql $connection_string
```

```
%load_ext sql
```

```
connection_string = '''mssql+pymssql://sa:123456@  
                        localhost/BANHANG?charset=utf8'''  
%sql $connection_string
```

```
'Connected: sa@BANHANG'
```

Sử dụng Magic SQL (%)

❑ Thực thi câu lệnh truy vấn

%sql <nội dung câu lệnh truy vấn>

● Ví dụ:

In [12]: **%sql** select * **from** customers limit 5

* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.

Out[12]:

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
4	Thomas	Hardy	London	UK	(171) 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

- ❑ Sử dụng Pandas đọc dữ liệu từ truy vấn
- ❑ Cách 1: sử dụng phương thức DataFrame

```
df = pd.DataFrame(SQL_Query_result, columns=[
    'field1', 'field2',...])
```

• Ví dụ 1:

```
In [32]: result = %sql select * from customers limit 5
df = pd.DataFrame(result)
```

```
* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
```

```
In [33]: print(df)
```

	0	1	2	3	4	5
0	1	Maria	Anders	Berlin	Germany	030-0074321
1	2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
2	3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
3	4	Thomas	Hardy	London	UK	(171) 555-7788
4	5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

❑ Sử dụng Pandas đọc dữ liệu từ truy vấn

• Ví dụ 2:

```
In [34]: result = %sql select * from customers limit 5
df = pd.DataFrame(result, index=range(1,6),
                  columns=['Id', 'FirstName', 'LastName', 'City', 'Country', 'Phone'])
```

```
* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
```

```
In [35]: print(df)
```

	Id	FirstName	LastName	City	Country	Phone
1	1	Maria	Anders	Berlin	Germany	030-0074321
2	2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
3	3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
4	4	Thomas	Hardy	London	UK	(171) 555-7788
5	5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

❑ Sử dụng Pandas đọc dữ liệu từ truy vấn

❑ Cách 2: sử dụng phương thức read_sql

`df = pd.read_sql(<câu lệnh truy vấn>, <đối tượng kết nối>)`

● Ví dụ:

```
In [36]: user = 'root'
password = ''
connect_string = "mysql+pymysql://{user}:{password}@localhost/testdb?charset=utf8"
            .format(user=user, password=password)
```

```
In [37]: from sqlalchemy.engine import create_engine
engine = create_engine(connect_string)
```

Sử dụng Magic SQL (%)

❑ Sử dụng Pandas đọc dữ liệu từ truy vấn

❑ Cách 2: sử dụng phương thức read_sql

`df = pd.read_sql(<câu lệnh truy vấn>, <đối tượng kết nối>)`

● Ví dụ:

```
In [38]: df2 = pd.read_sql('select * from customers limit 5', engine)
```

```
In [39]: print(df2)
```

	Id	FirstName	LastName	City	Country	Phone
0	1	Maria	Anders	Berlin	Germany	030-0074321
1	2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
2	3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
3	4	Thomas	Hardy	London	UK	(171) 555-7788
4	5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

- ❑ Sử dụng Pandas đọc dữ liệu từ truy vấn
- ❑ Cách 3: sử dụng phương thức DataFrame của tập kết quả

```
In [44]: result = %sql select * from customers limit 5
df3 = result.DataFrame()

* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
```

```
In [45]: print(df3)
```

	Id	FirstName	LastName	City	Country	Phone
0	1	Maria	Anders	Berlin	Germany	030-0074321
1	2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
2	3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
3	4	Thomas	Hardy	London	UK	(171) 555-7788
4	5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

- ❑ Sử dụng Python Variables trong SQL Statements
- ❑ Truyền tham số cho câu lệnh truy vấn
 - Sử dụng dấu **:** trước tên tham số trong câu truy vấn

```
In [49]: city='London'
%sql select * from customers where City = :city limit 5

* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
```

```
Out[49]:
```

Id	FirstName	LastName	City	Country	Phone
4	Thomas	Hardy	London	UK	(171) 555-7788
11	Victoria	Ashworth	London	UK	(171) 555-1212
16	Elizabeth	Brown	London	UK	(171) 555-2282
19	Ann	Devon	London	UK	(171) 555-0297
53	Simon	Crowther	London	UK	(171) 555-7733

Sử dụng Magic SQL (%)

- ❑ Sử dụng Python Variables trong SQL Statements
- ❑ Gán kết quả cho biến từ câu lệnh truy vấn
 - Với câu lệnh chỉ có 1 dòng

<biến> = %sql <nội dung truy vấn>

```
In [54]: result = %sql select * from customers limit 5
result

* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
```

```
Out[54]:
```

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
4	Thomas	Hardy	London	UK	(171) 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

Sử dụng Magic SQL (%)

- ❑ Sử dụng Python Variables trong SQL Statements

- ❑ Gán kết quả cho biến từ câu lệnh truy vấn
 - Với câu lệnh có nhiều dòng

```
%%sql <biến> <<  
<nội dung truy vấn>
```

Sử dụng Magic SQL (%)

- Ví dụ:

```
In [59]: %%sql result <<
select *
from customers
limit 5

* mysql+pymysql://root:***@localhost/testdb?charset=utf8
5 rows affected.
Returning data to local variable result
```

```
In [60]: result
```

```
Out[60]:
```

Id	FirstName	LastName	City	Country	Phone
1	Maria	Anders	Berlin	Germany	030-0074321
2	Ana	Trujillo	México D.F.	Mexico	(5) 555-4729
3	Antonio	Moreno	México D.F.	Mexico	(5) 555-3932
4	Thomas	Hardy	London	UK	(171) 555-7788
5	Christina	Berglund	Luleå	Sweden	0921-12 34 65

