

## New Media Data Analytics and Application

Lecture 3: Advanced Python Programming

Ting Wang

#### Outlines

- File and Stream I/O
- Web Programming
- Connect to Database





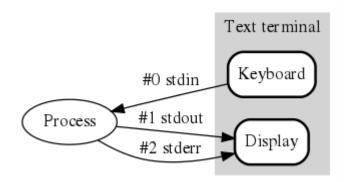


input text from web or documents and output it

## File and Stream I/O

## Stream 流

a sequence of data elements made available over time.

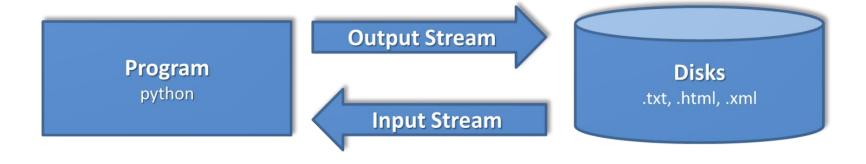


Eg: The standard streams for input, output, and error.





#### Files and Stream





## Read Files (Input)

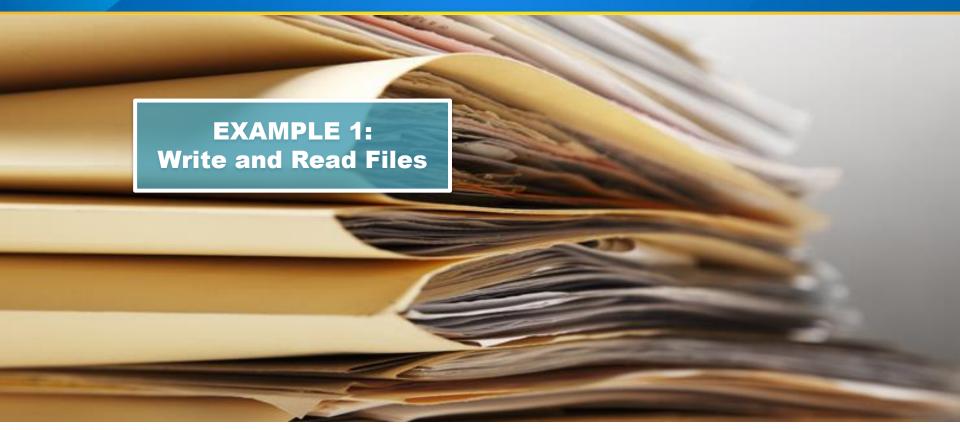
with open('FILE\_PATH', 'r/rb') as VARIABLE\_NAME
VARIABLE\_NAME.read()

## Write Files (Output)

with open('FILE\_PATH', 'w/wb') as VARIABLE\_NAME
VARIABLE\_NAME.write('TEXT')

## Absolute and Relative Paths 绝对和相对路径







#### Example: Document Write-Save-Read

```
import datetime
with open('test.txt', 'w') as f:
  f.write('今天是')
  f.write(datetime.datetime.now().strftime('%Y-%m-%d'))
with open('test.txt', 'r') as f:
  s = f.read()
  print('open for read...')
  print(s)
```

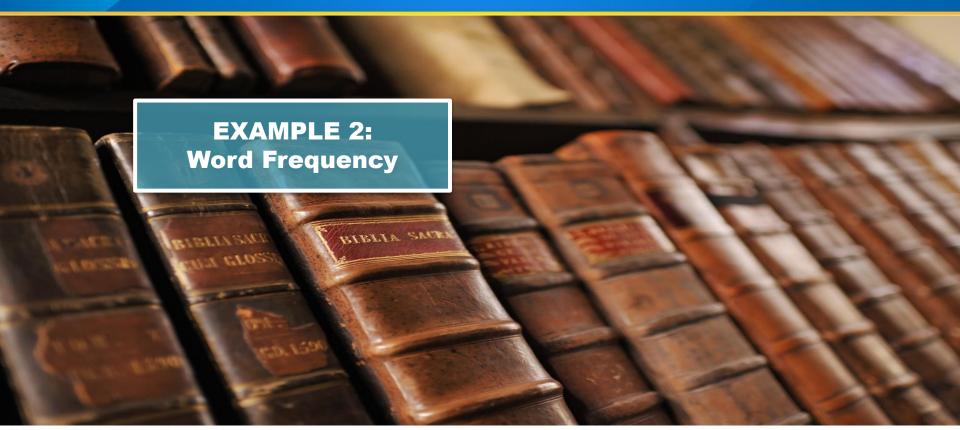


#### Example: Webpage Crawling

```
import urllib.request
response = urllib.request.urlopen('http://www.shisu.edu.cn/about/introducing-sisu')
HTMLText = response.read()
```

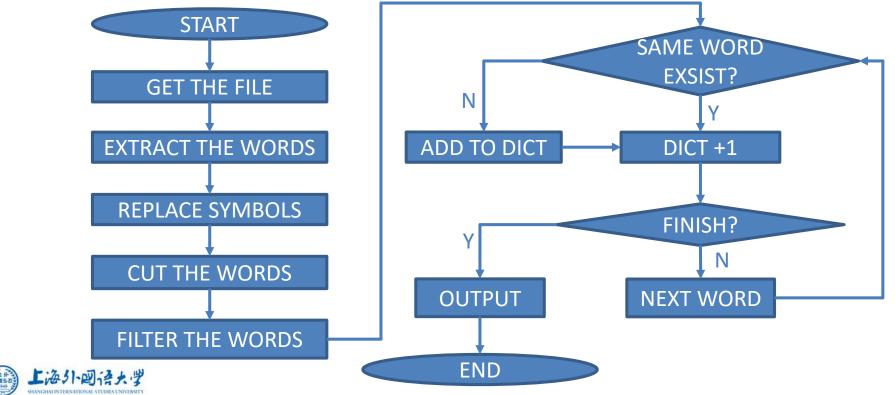
with open('Files/shisu.html', 'wb') as f: f.write(HTMLText)







Thinking in Programming



```
with open ('text flu. txt', 'rU') as f:
    s array = str(f.read())
s_array = s_array.replace(',',')
s_array = s_array.replace('.','')
s_array = s_array.replace('\'','')
s_array = s_array.replace('\"','
s_array = s_array.replace(':','')
s_array = s_array.replace(':',''
s_array = s_array.replace('?'.')
s_array = s_array.replace(''',''')
s_array = s_array.replace('~',',')
s_array = s_array.replace('!'.'')
s_array = s_array.replace('@','')
s_array = s_array.replace('#',' '
s_array = s_array.replace('%'.'')
s array = s array.replace('$',''
s_array = s_array.replace('^','')
s_array = s_array.replace('&','')
s array = s array.replace('*','')
s_array = s_array.replace('('.''))
s_array = s_array.replace(')'.''

s_array = s_array.replace('-'.')
s_array = s_array.replace('+','')
```

```
s_array = s_array.replace('_', '')
s_array = s_array.replace('='.'')
s_array = s_array.replace('{',''
s_array = s_array.replace(')',''
s_array = s_array.replace('[',''
s_array = s_array.replace(']'.''
s_array = s_array.replace('|','')
s_array = s_array.replace('\\','')
s_array = s_array.replace('<','')
s_array = s_array.replace('>','')
s array = s array.split()
word dict = {}
for i in range (len (s array)):
    if s array[i] not in word dict:
        word dict[s arrav[i]] = 1
    else:
        word dict[s array[i]] += 1
with open ('result flu test. txt', 'w') as f:
    for word, number in word dict. items():
        result = word + \sqrt[m]{t''} + str(number) + \sqrt[m]{n''}
        f. write (result)
```





my first website using web framework

## Web Programming

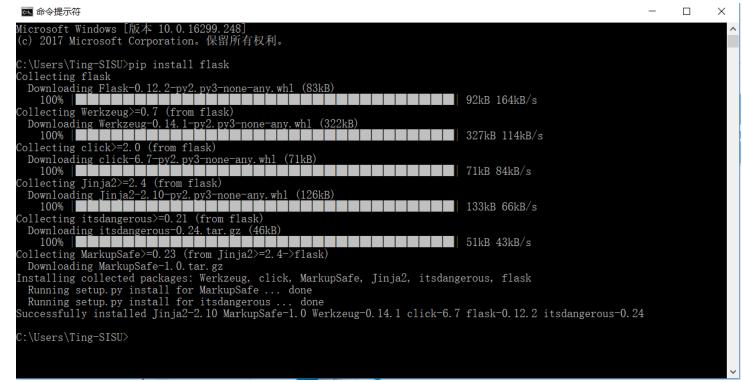
#### Python Web Framework

- Flask
- Django
- Web.py
- Tornado
- Bottle





## Install Flask: pip install flask







#### The simplest Flask code

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello World!'

if __name__ == '__main__':
    app.run()
```

#### HTML?

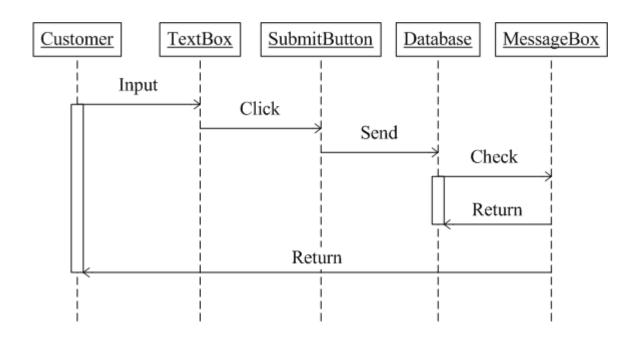






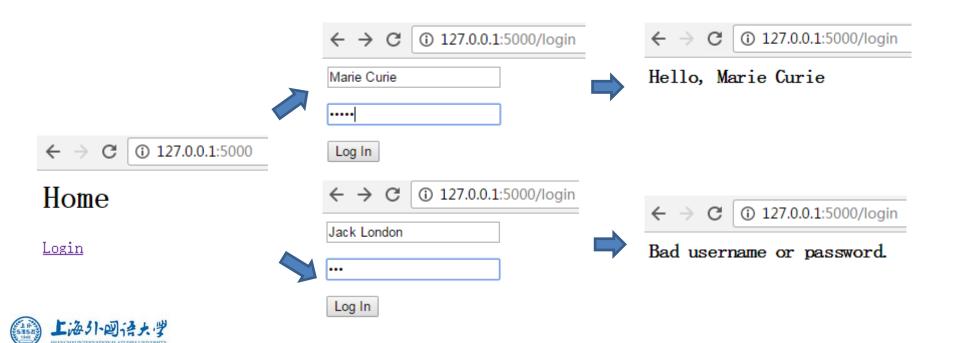


## Login





#### Final Performance



• A small step:

```
from flask import request
app = Flask( name )
@app. route('/', methods=['GET', 'POST'])
def home():
               return ''' <h1>Home</h1>
                                                  <a href="/login">Login!</a>'''
@app. route ('/login', methods=['GET'])
def login form():
               return ''' \( form action="/login" method="post" \)
                                                       username <input name="username">
                                                      \( \p \) password \( \lambda \) input name="password" \( \tau \) \( \p \) \( \lambda \) \( \tau \) \( \p \) \( \lambda \) \( \tau \) \( \lambda \) \( \lambda \) \( \tau \) \( \lambda \) \( \tau \) \( \lambda \) \( \lambd
                                                      <button type="submit">Log In</button>
                                                      </form>'''
@app. route('/login', methods=['POST'])
def login():
               if request.form['username'] == 'abc' and request.form['password'] == '123':
                              return '<h3>Hello, ' + request. form['username'] + '</h3>'
               else:
                              return '\(\hat{h3}\)hi!vou are wrong again! fighting!!!! \(\hat{h3}\)'
if name == ' main ':
               app. run()
```

from flask import Flask





connect python with MySQL

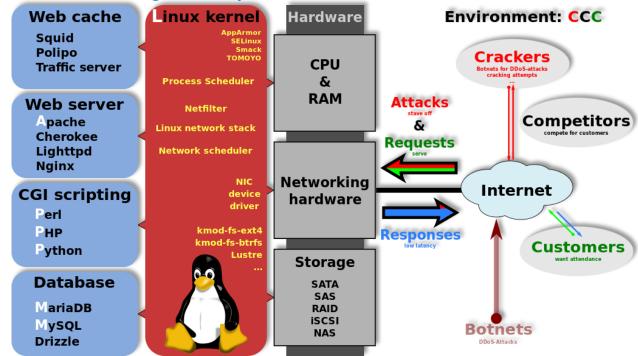
#### Connect to Database

## MySQL

an open-source relational database management system (RDBMS)

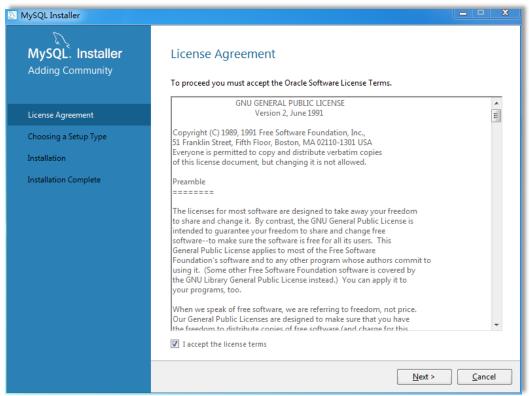
#### **LAMP**

- Linux
- Apache
- Perl+PHP+Python
- MySQL



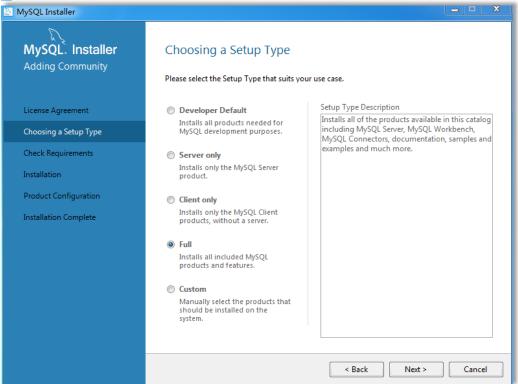


## MySQL Installation





Choosing a Setup Type





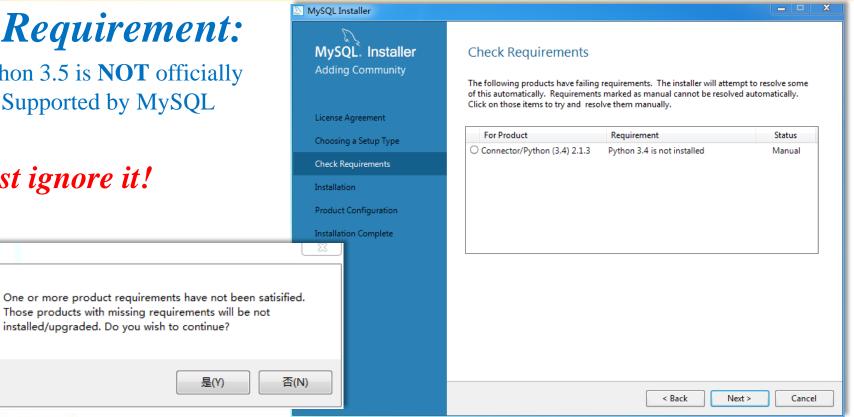
## A Requirement:

Python 3.5 is **NOT** officially Supported by MySQL

installed/upgraded. Do you wish to continue?

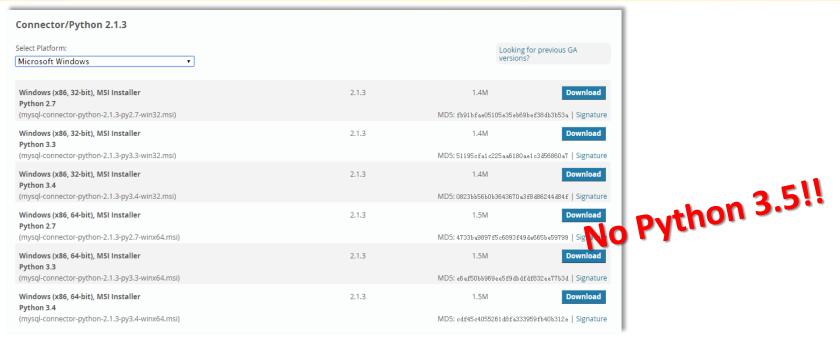
是(Y)

#### Just ignore it!





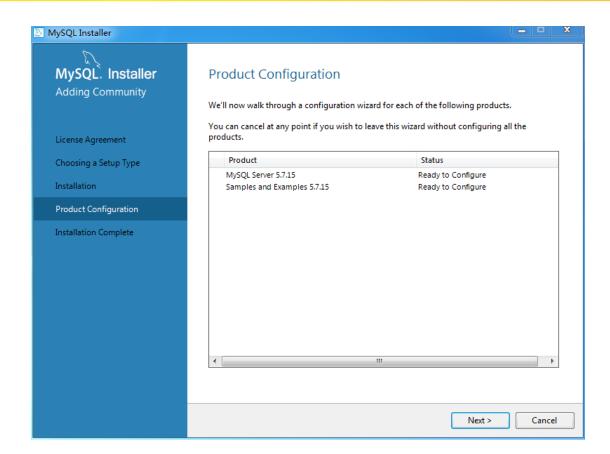
Confirm



http://dev.mysql.com/downloads/connector/python/



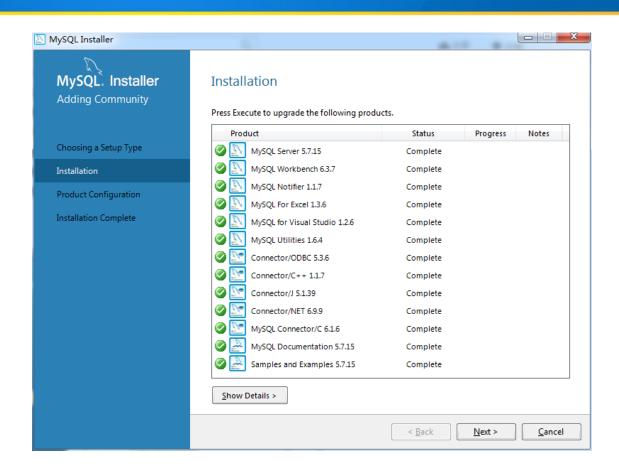
## Product Configuration



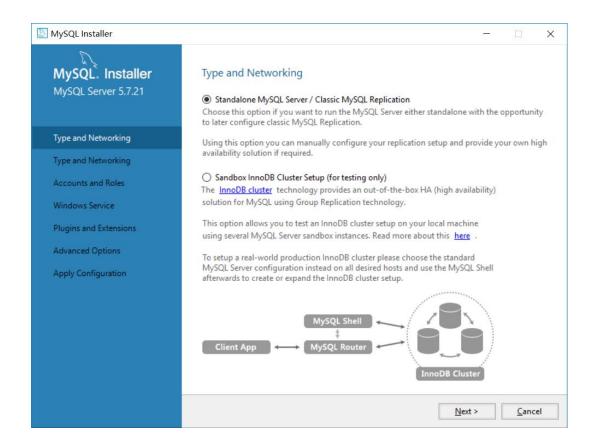


#### **Installation**



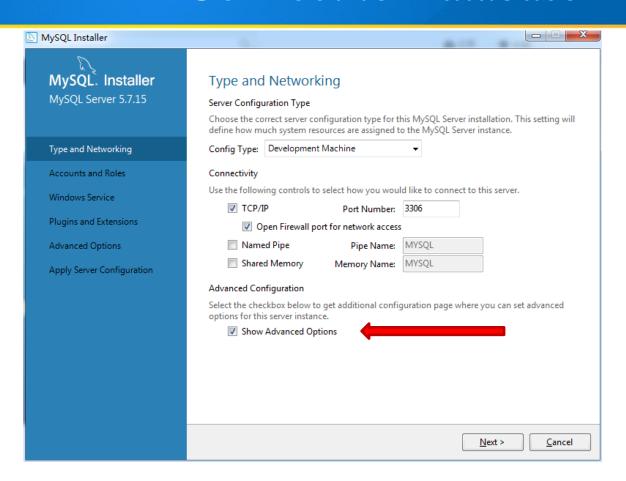






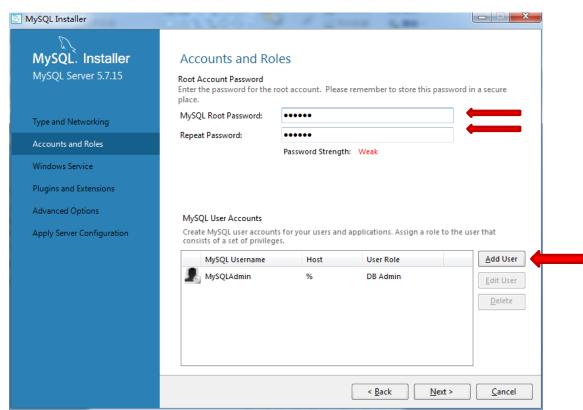


# Show Advanced Options



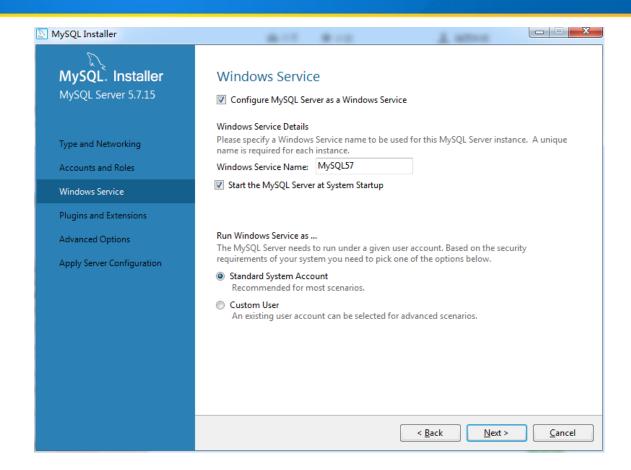


#### Account and Roles



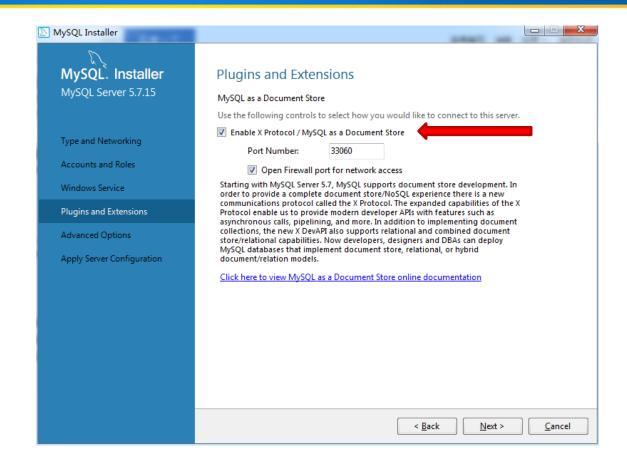


#### Next



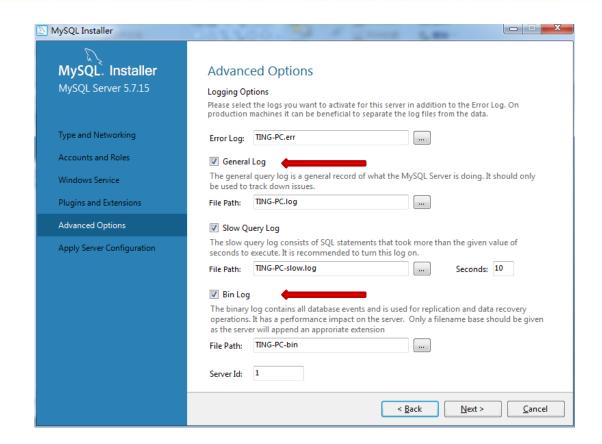


## New MySQL Function for NoSQL



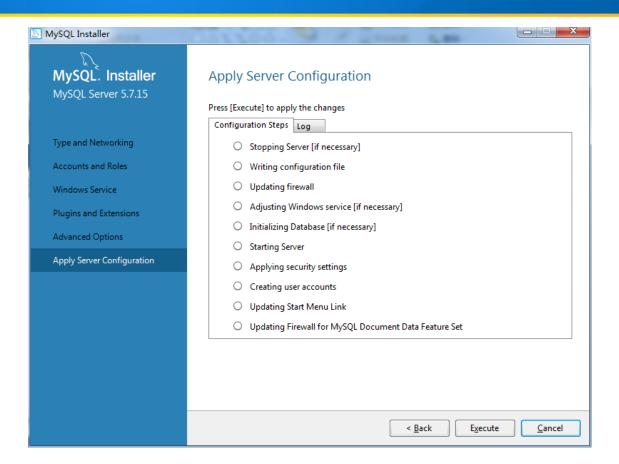


## Logs





MySQL Server Server Configuration





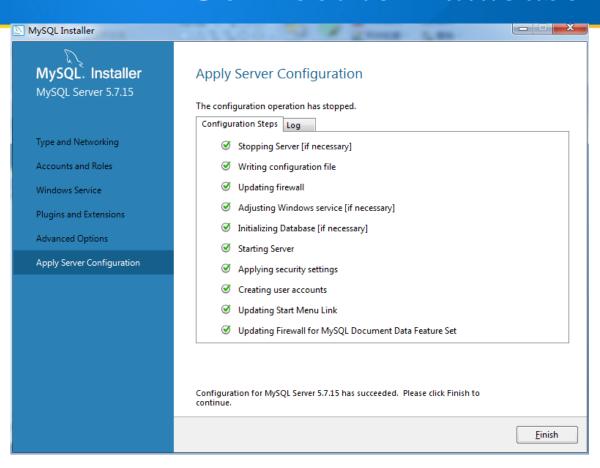
## Sometimes, maybe...if you are using Wifi





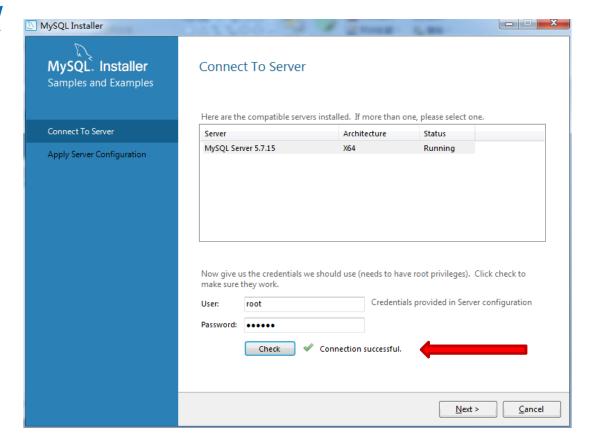


MySQL Server
Apply
Server
Configuration

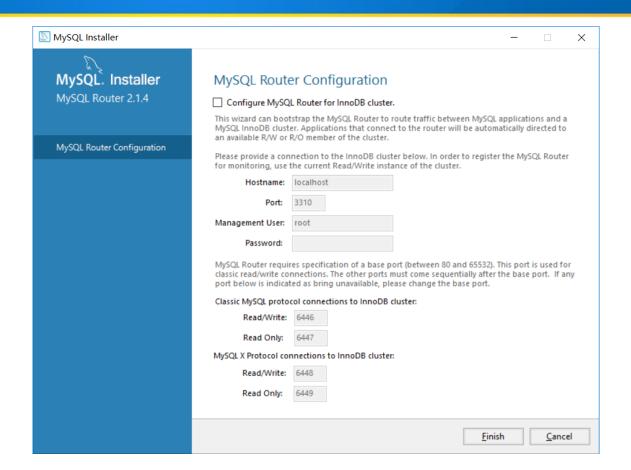




# Prepare to install Samples - Check!

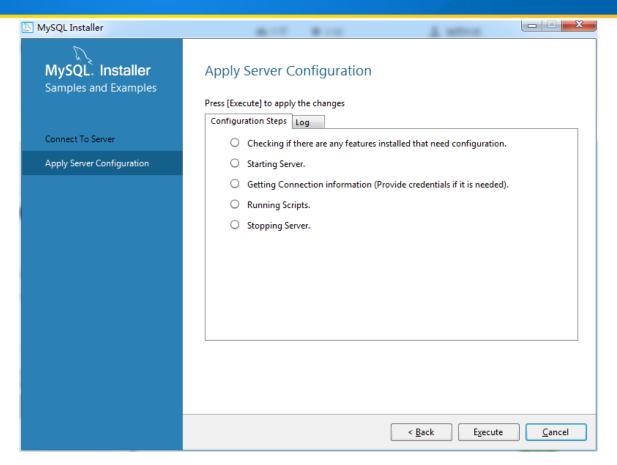




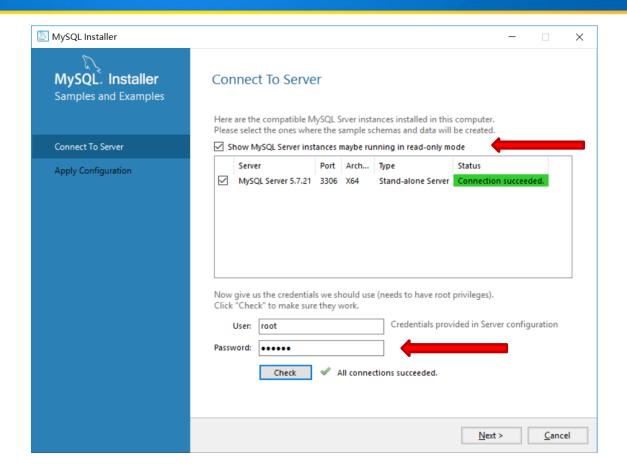




Samples
Apply
Server
Configuration

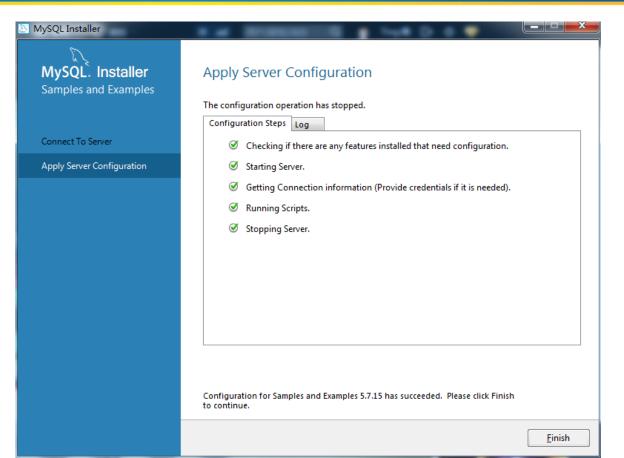






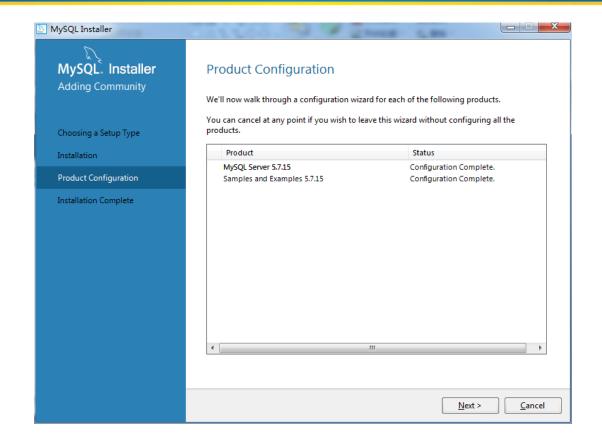


# Finish Server Configuration



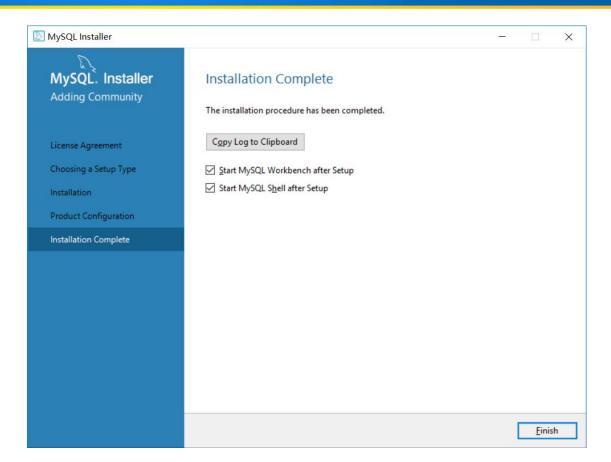


#### Next



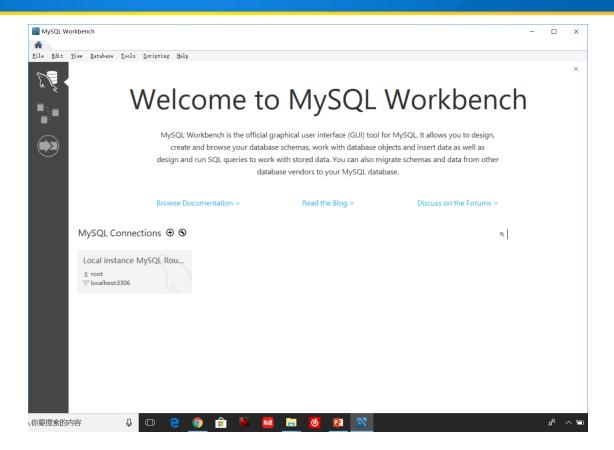


# **Complete**





Welcome





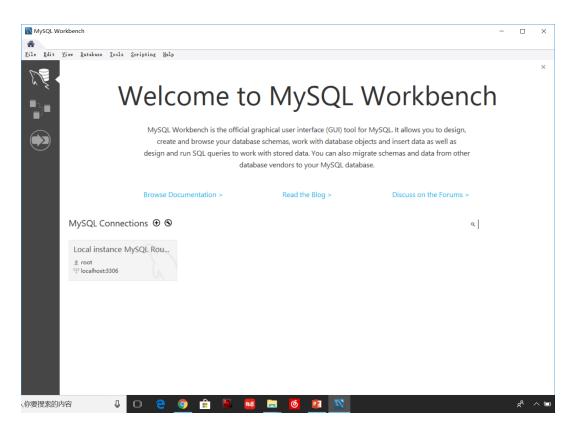
#### Test

- Successful?
- Test it!

```
MySQL 5.7 Command Line Client
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \setminus g .
Your MySQL connection id is 9
Server version: 5.7.15-log MySQL Community Server (GPL)
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases;
 Database
  information_schema
 mysq1
  performance_schema
 sakila
 s ys
 world
 rows in set (0.00 sec)
mysql>
```



• Click the "root"



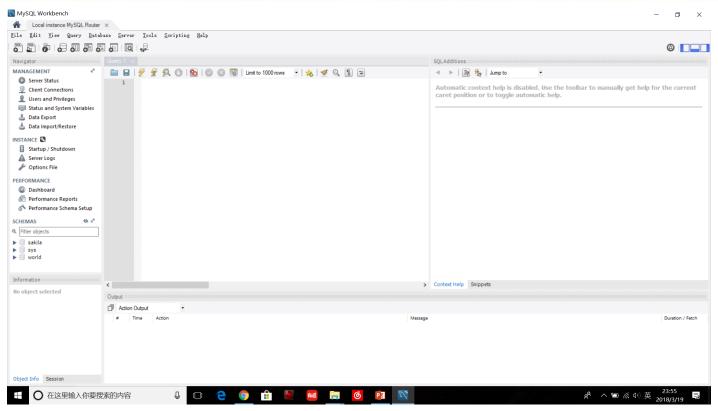


# Start MySQL Workbench











Install a third-party module for the connection between Python and MySQL: Preparation pip install pymysql

• pymysql, a third-party module for the connection between Python and MySQL



**Tips** 

Do NOT forget to restart your IDE after you install some new third-party modules.







an approach to store data

# Foundations of Data Base

# Widely Used Data Base

- MySQL
- SQL Server
- Oracle





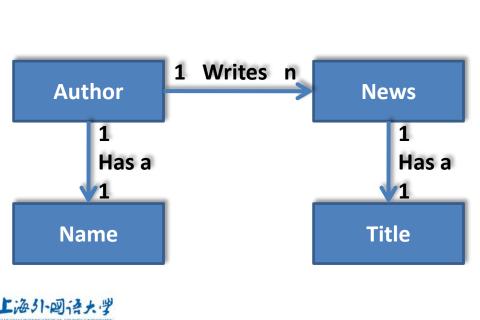






#### Relational Database

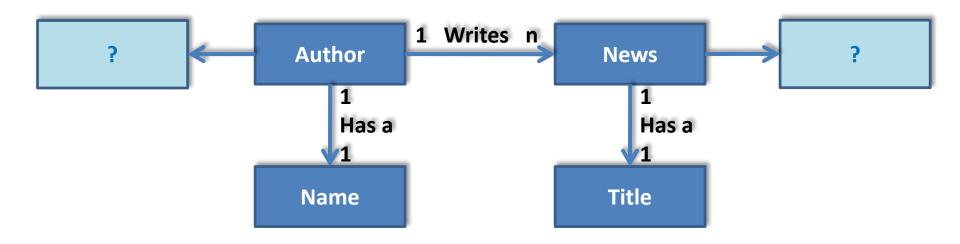
a digital database whose organization is based on the **relational model** of data







# ER Diagram





# Structure Design of Data Base

- 1:1, in the same table
- 1:n, mark "1" in "n" 's table
- n:m, create a new table to store





#### **Constraints**

- 1. Primary Key: Unique ID
- 2. Foreign Key: The same thing
- 3. Index: a quicker access to data



# Structured Query Language (SQL)

a special-purpose programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).





#### Create

# CREATE TABLE TABLE\_NAME

```
CUSTOMER_ID INT IDENTITY(1,1) PRIMARY KEY CLUSTERED,
CUSTOMER_NAME CHAR(100) NOT NULL,
CUSTOMER_PWD CHAR(100) NOT NULL,
)
```



**Drop**DROP TAME TABLE\_NAME

DROP TABLE CUSTOMER



#### Insert

```
INSERT INTO TABLE_NAME(COLUMN1,
COLUMN2,...)
VALUES('STRING',NUMBER)
```

```
INSERT INTO CUSTOMER(CUSTOMER NAME, CUSTOMER PWD) VALUES('Thomas Edison', 'sisu')
INSERT INTO CUSTOMER(CUSTOMER NAME, CUSTOMER PWD) VALUES('Albert Einstein', 'sisu')
INSERT INTO CUSTOMER(CUSTOMER NAME, CUSTOMER PWD) VALUES('Marie Curie', 'shisu')
```



#### Delete

DELETE FROM TABLE\_NAME WHERE CONDITION





Update
UPDATE TABLE\_NAME SET
 COLUMN=COLUMN\_VALUE WHERE
 CONDITION

UPDATE CUSTOMER SET CUSTOMER PWD='shisu' WHERE CUSTOMER NAME='Albert Einstein'



#### Select

SELECT - FROM - WHERE

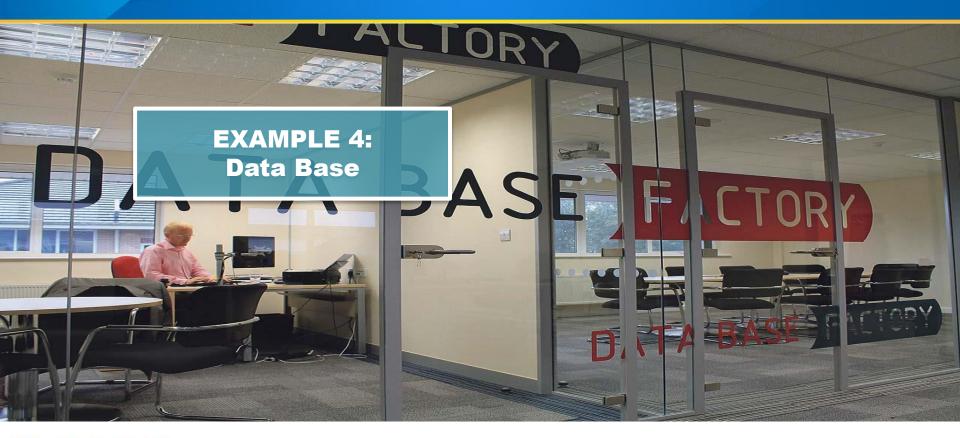
SELECT COLUMN\_NAME FROM TABLE\_NAME WHERE CONDITION

GROUP BY CONDITION

ORDER BY CONDITION



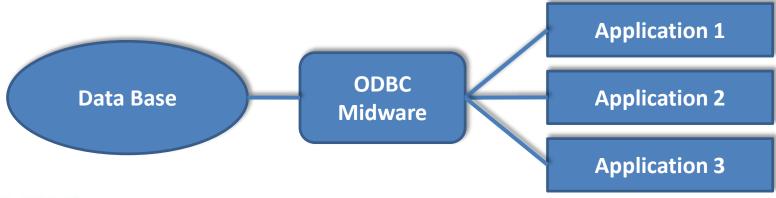






# Open Database Connectivity (ODBC)

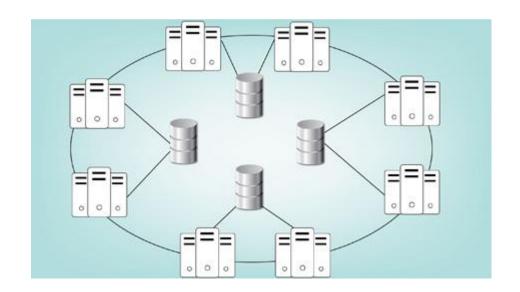
a standard application programming interface (API) for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of database systems and operating systems. An application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.





#### Distributed Data Base

a database in which data are stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers.





# NoSQL (Not Only SQL)

triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com, NoSQL databases are increasingly used in big data and real-time web applications.



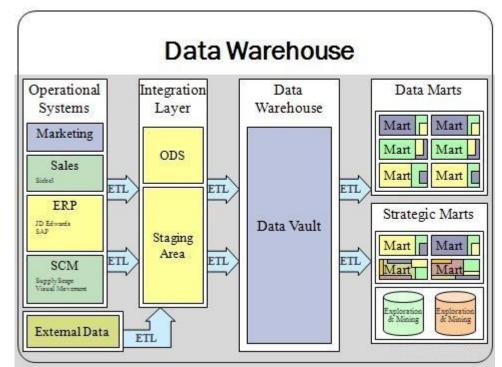


# Data Warehouse 数据仓库

a system used for reporting and data analysis, and is considered as a core component of

**business intelligence** environment







# Data Mining 数据挖掘

an interdisciplinary subfield of computer science, which is the computational process of **discovering patterns** in large data sets involving methods at the intersection of **artificial intelligence**, **machine learning**, **statistics**, and **database systems**.









use database





- How to use database in programming?
  - Steps:
    - 1. Connection Open
    - 2. Cursor Creating
    - 3. SQL Execution
    - 4. Cursor Moving
    - 5. Connection Close

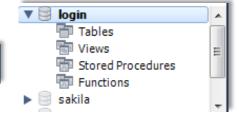




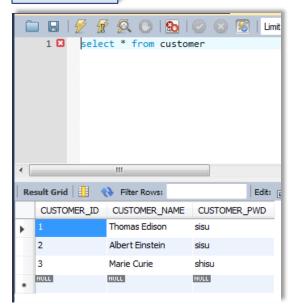


- 1. Create a new schema
- 2. Set as a default schema (right click)
- 3. Run CreateMySQL.sql





#### 5. Result



CREATE TABLE CUSTOMER (

CUSTOMER\_ID INT(10) PRIMARY KEY AUTO\_INCREMENT,

CUSTOMER\_NAME VARCHAR(100) NOT NULL,

CUSTOMER\_PWD VARCHAR(100) NOT NULL);

4. Run InsertMySQL.sql

INSERT INTO CUSTOMER(CUSTOMER\_NAME,CUSTOMER\_PWD) VALUES('Thomas Edison','sisu'); INSERT INTO CUSTOMER(CUSTOMER\_NAME,CUSTOMER\_PWD) VALUES('Albert Einstein','sisu'); INSERT INTO CUSTOMER(CUSTOMER\_NAME,CUSTOMER\_PWD) VALUES('Marie Curie','shisu');

#### Code

```
import pymysql
#1.Connection Open
conn = pymysql.connect(user='root',
password='123456', database='login')
#2.Cursor Creating:
cursor = conn.cursor()
#3.SQL Execution
sqlstr = 'select * from customer where
customer name="Thomas Edison"
and customer_pwd="shisu""
```

```
#4.Cursor Moving
cursor.execute(sqlstr)
numrows = len(cursor.fetchall())
conn.commit()
#5.Connection Close
                                   Result:
cursor.close()
conn.close()
                                             "C:\Program Files\Python35\python.exe"
                                            Hi!You are wrong!
# get data from request:
                                            Process finished with exit code 0
if numrows>0:
  print('Hello, Thomas Edison! ')
else:
  print('Hi!You are wrong! ')
```











```
# 3. SQL Execution
from flask import Flask
                                                                                # 执行SQL语句,循环插入记录:
from flask import request
                                                                                salstr='select * from customer where
import pymysql
                                                                            customer name=\''+request.form['username']+\
                                                                                      '\' and customer pwd=\''+request.form['password']+'\''
app = Flask( name )
                                                                                # 4. Cursor Moving
@app. route('/', methods=['GET', 'POST'])
                                                                                # 体验游标
def home():
   return ''' <h1>Home</h1>
                                                                                # 执行, 游标移至当前位置
            <a href="/login">Login</a>'''
                                                                                cursor. execute (sqlstr)
                                                                                numrows = int(cursor.rowcount)
@app. route('/login', methods=['GET'])
                                                                                # 提交事务:
def login form():
                                                                                conn.commit()
   return ''' \( form action="/login" method="post" \)
             <input name="username">
                                                                                # 5. Connection Close
             <input name="password" type="password">
                                                                                # 关闭Cursor:
             <button type="submit">Log In</button>
                                                                                cursor, close()
             </form>'''
                                                                                # 关闭Connection:
                                                                                conn. close()
@app. route('/login', methods=['POST'])
def login():
                                                                               # get data from request:
   # get data from database:
                                                                                if numrows>0:
                                                                                   return '<h3>Hello, '+request. form['username']+'</h3>'
    # 连接到MySQL数据库
                                                                                return '<h3>Bad username or password. </h3>'
    # 1. Connection Open
   conn = pymysql.connect(user='root', password='123456', database='login')
                                                                            if name == ' main ':
    # 2. Cursor Creating:
                                                                                app. run()
   cursor = conn. cursor()
```





# Reference

# Reference

# Django

http://www.ziqiangxuetang.com/django/django
 -tutorial.html

# Python操作MySQL数据库9个实用实例

http://www.jb51.net/article/76231.htm



## Reference

Reference 数据库系统概论(第5版) 王珊,萨师煊著 高等教育出版社







# Home Work

## Home Work

Design a simple web site and connect your databases to this web site (deadline: Apr. 1)

- It should contain:
  - Source code, including Registration and Login
  - Databases
  - A Report (very important!!)







# The End of Lecture 3

Thank You



