



Computational Journalism

Lecture 3: Advanced Python Programming

Ting Wang

Outlines

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





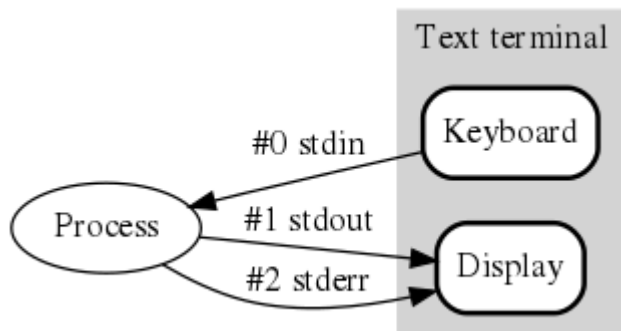
input text from web or documents and output it

File and Stream I/O

File and Stream I/O

Stream 流

a sequence of data elements made available over time.

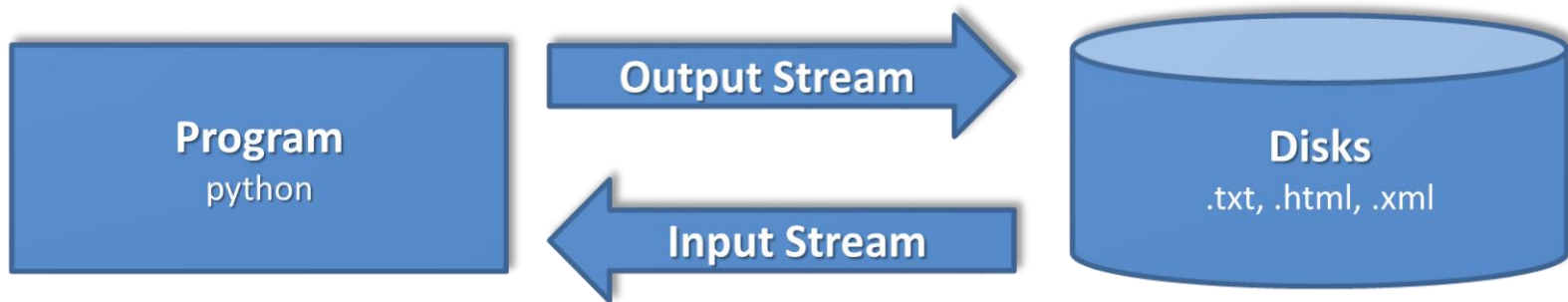


Eg: The standard streams for input, output, and error.



File and Stream I/O

Files and Stream



File and Stream I/O

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



File and Stream I/O

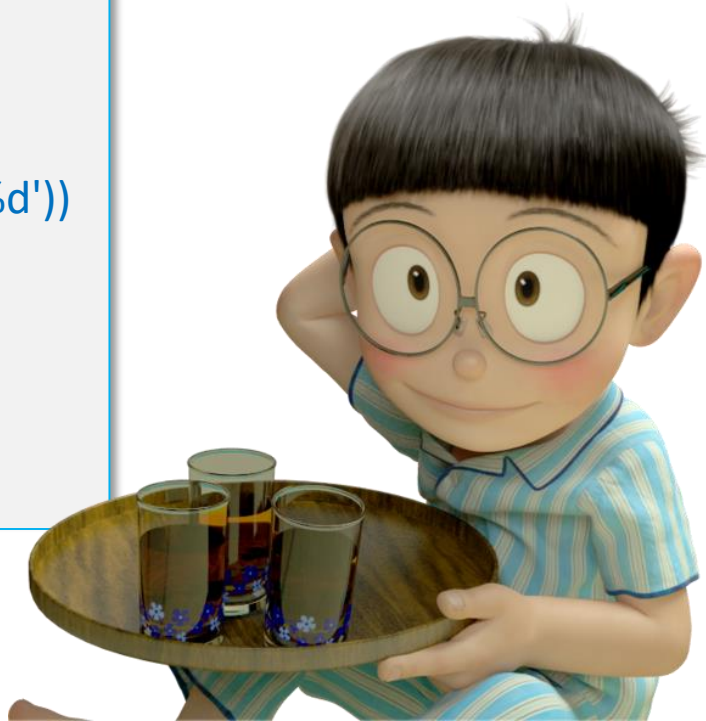
EXAMPLE 1:
Write and Read Files



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)
```



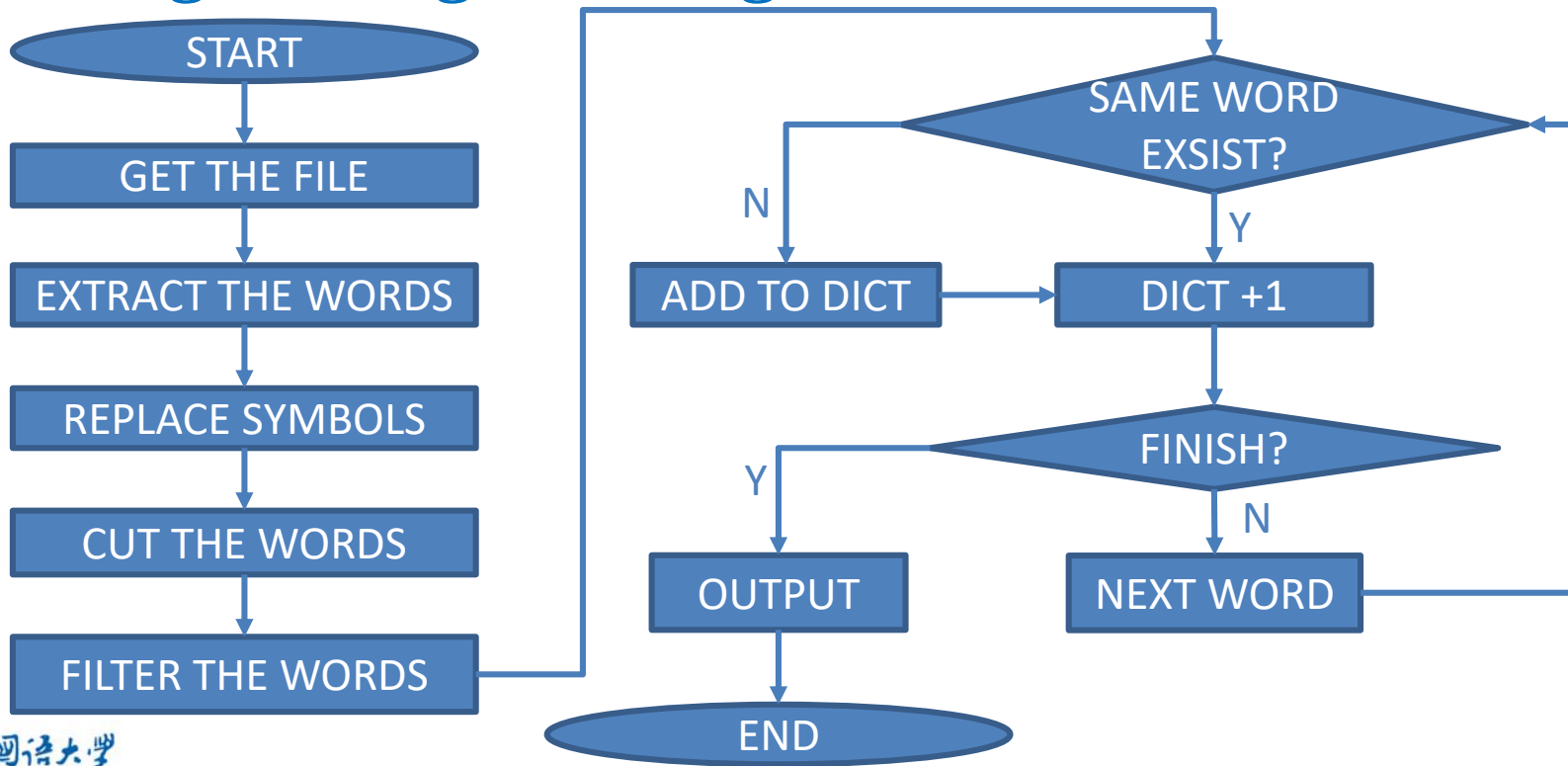
File and Stream I/O

**EXAMPLE 2:
Word Frequency**



File and Stream I/O

Thinking in Programming



File and Stream I/O

```
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_array[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 + "\\t" + str(number) + "\\n"  
        f.write(result)
```





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

my first website using web framework

Web Programming

Python Web Framework

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



Install Flask: pip install flask



```
命令提示符
Microsoft Windows [版本 10.0.16299.248]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\Ting-SISU>pip install flask
Collecting flask
  Downloading Flask-0.12.2-py2.py3-none-any.whl (83kB)
    100% |#####| 92kB 164kB/s
Collecting Werkzeug>=0.7 (from flask)
  Downloading Werkzeug-0.14.1-py2.py3-none-any.whl (322kB)
    100% |#####| 327kB 114kB/s
Collecting click>=2.0 (from flask)
  Downloading click-6.7-py2.py3-none-any.whl (71kB)
    100% |#####| 71kB 84kB/s
Collecting Jinja2>=2.4 (from flask)
  Downloading Jinja2-2.10-py2.py3-none-any.whl (126kB)
    100% |#####| 133kB 66kB/s
Collecting itsdangerous>=0.21 (from flask)
  Downloading itsdangerous-0.24.tar.gz (46kB)
    100% |#####| 51kB 43kB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.4->flask)
  Downloading MarkupSafe-1.0.tar.gz
Installing collected packages: Werkzeug, click, MarkupSafe, Jinja2, itsdangerous, flask
  Running setup.py install for MarkupSafe ... done
  Running setup.py install for itsdangerous ... done
Successfully installed Jinja2-2.10 MarkupSafe-1.0 Werkzeug-0.14.1 click-6.7 flask-0.12.2 itsdangerous-0.24

C:\Users\Ting-SISU>
```

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?



EXAMPLE 3: Flask Login



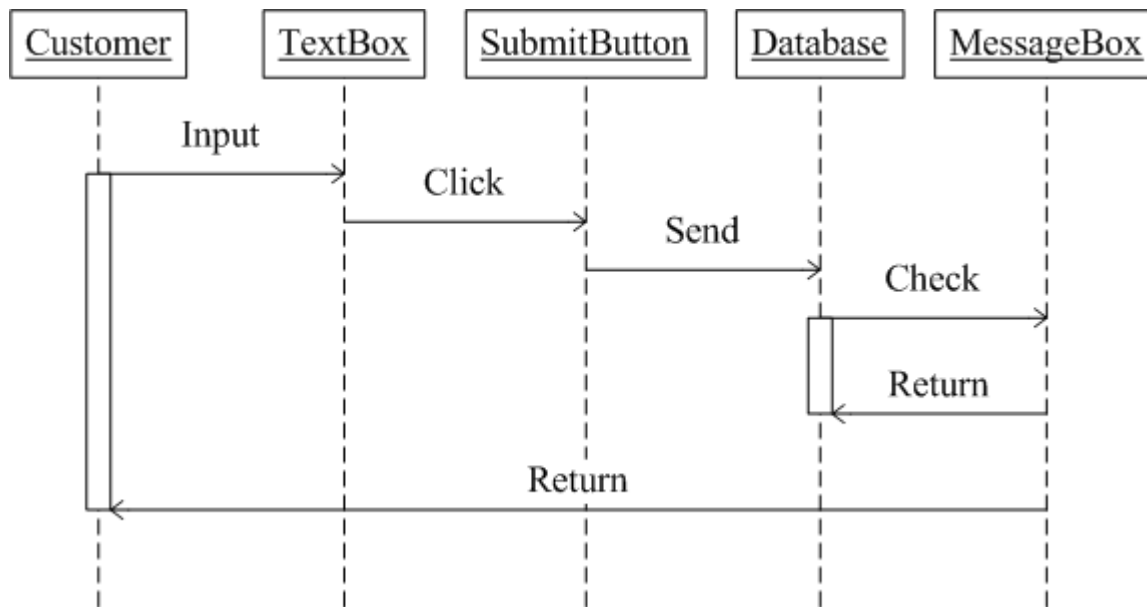
Adam .

Sign-in options

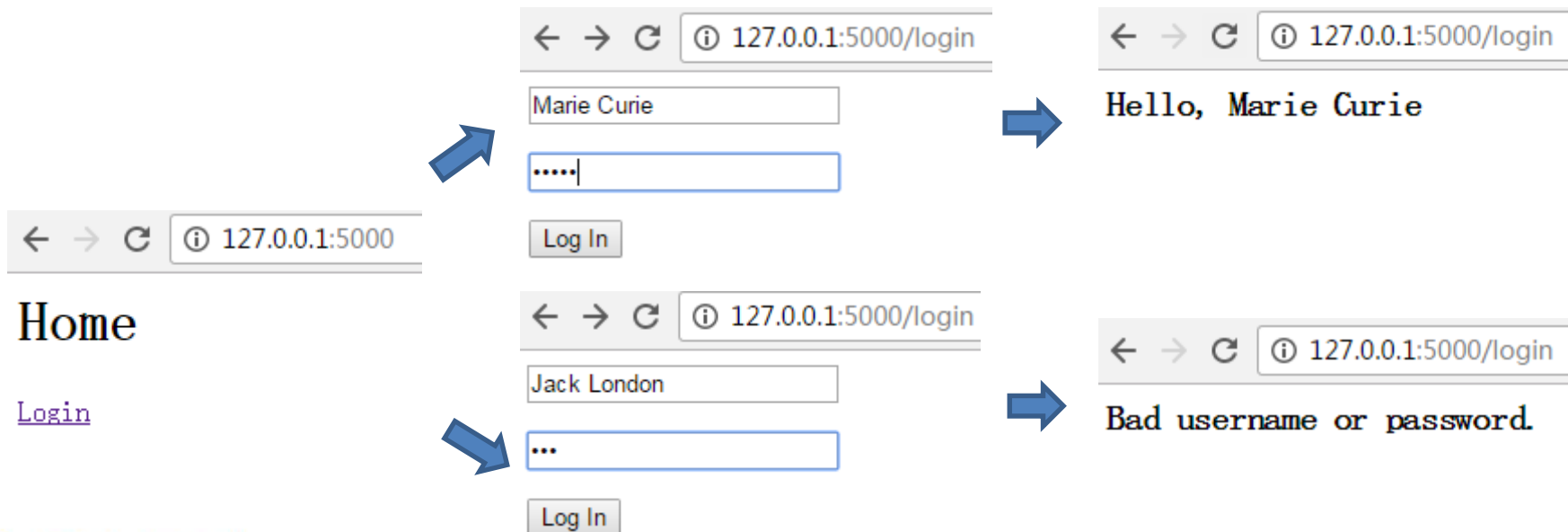
ENG
US



Login



Final Performance



Web Programming

- A small step:

```
from flask import Flask
from flask import request

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def home():
    return '''<h1>Home</h1>
    <p><a href="/login">Login!</a></p>'''

@app.route('/login', methods=['GET'])
def login_form():
    return '''<form action="/login" method="post">
    <p> username <input name="username"></p>
    <p> password <input name="password" type="password"></p>
    <p><button type="submit">Log In</button></p>
    </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 '<h3>hi!you are wrong again! fighting!!!! </h3>'

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





connect python with MySQL

Connect to Database

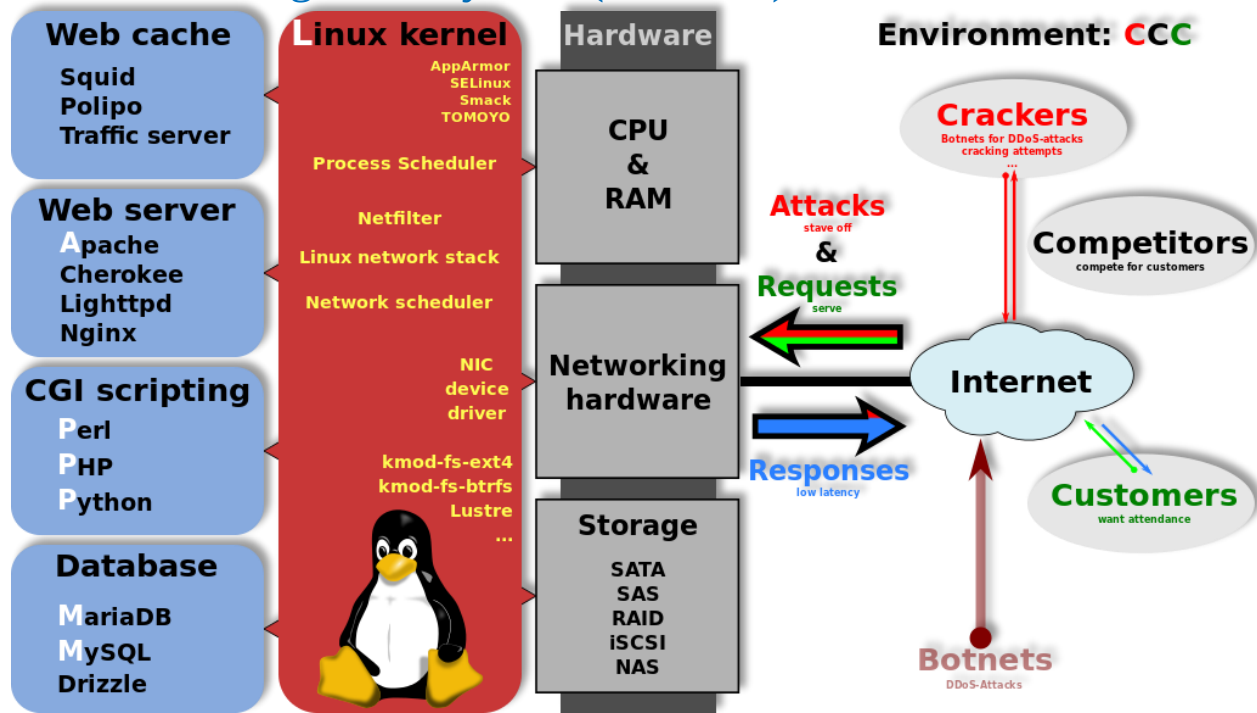
Connect to Database

MySQL

an open-source relational database management system (RDBMS)

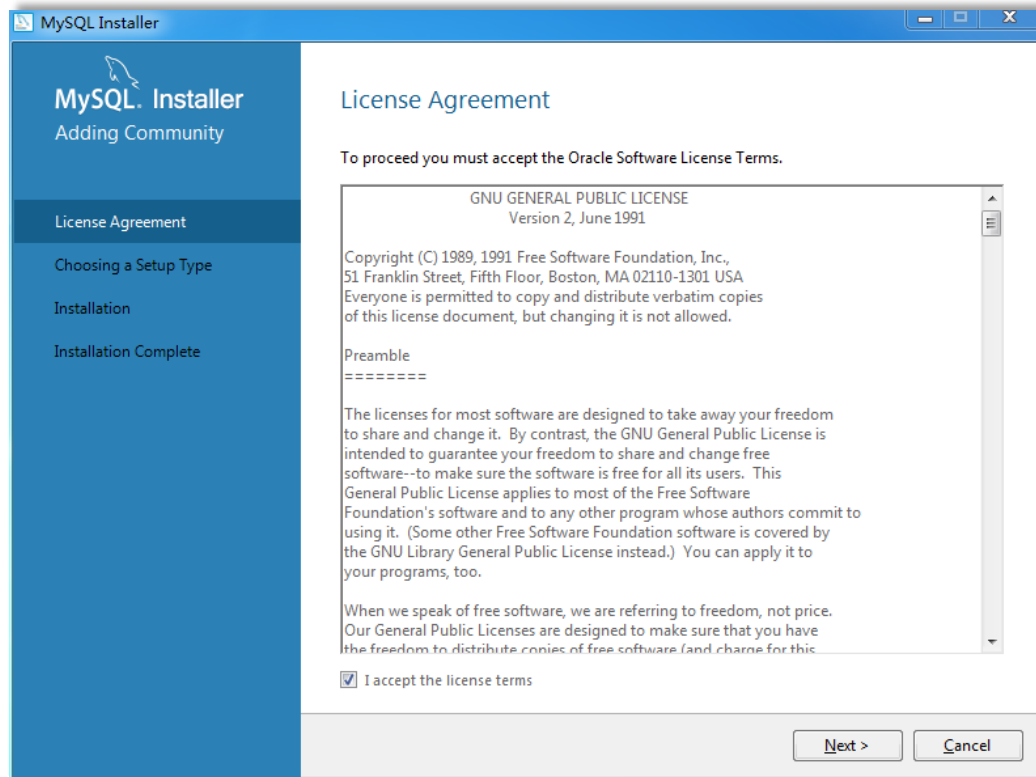
LAMP

- **L**inux
- **A**pache
- **P**erl+**P**HP+**P**ython
- **M**ySQL



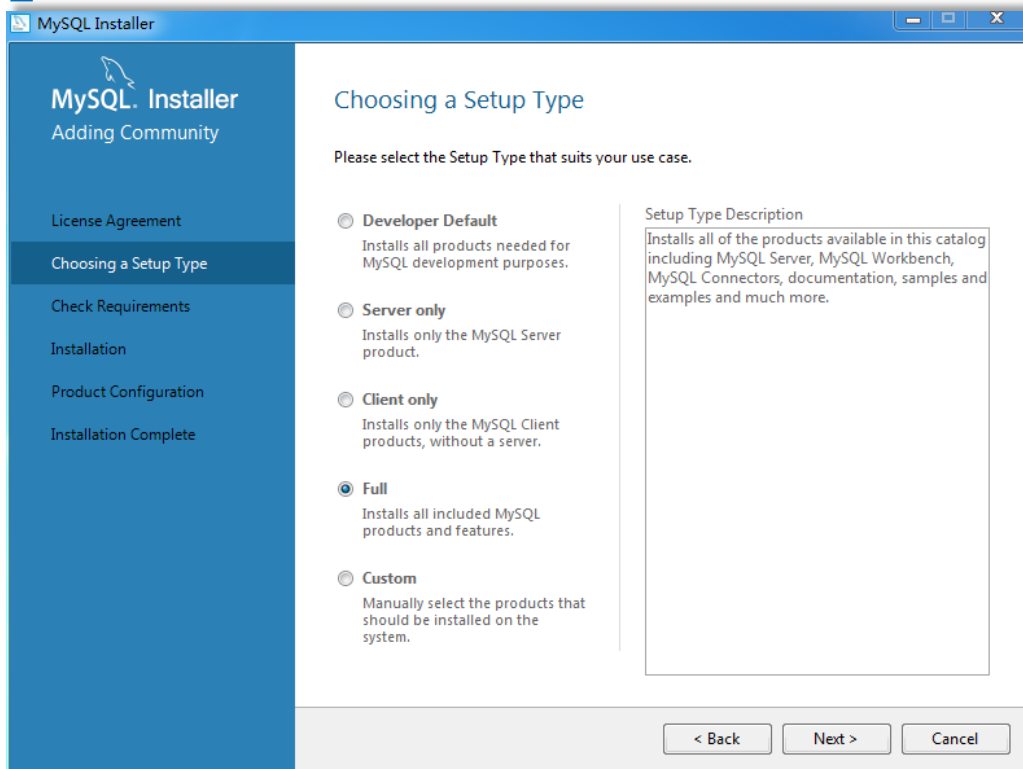
Connect to Database

MySQL Installation



Connect to Database

Choosing a Setup Type

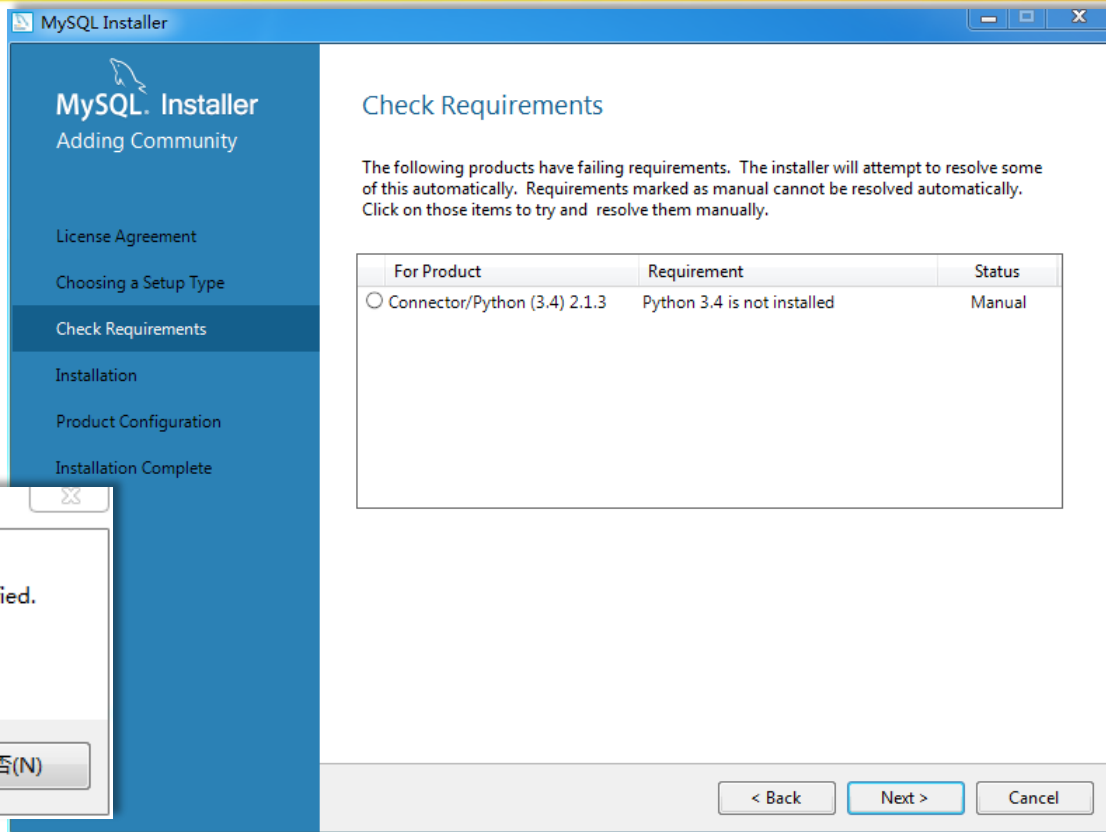


Connect to Database

A Requirement:

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

Just ignore it!



Connect to Database

Connector/Python 2.1.3

Select Platform:

Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer Python 2.7 (mysql-connector-python-2.1.3-py2.7-win32.msi)	2.1.3	1.4M	Download
Windows (x86, 32-bit), MSI Installer Python 3.3 (mysql-connector-python-2.1.3-py3.3-win32.msi)	2.1.3	1.4M	Download
Windows (x86, 32-bit), MSI Installer Python 3.4 (mysql-connector-python-2.1.3-py3.4-win32.msi)	2.1.3	1.4M	Download
Windows (x86, 64-bit), MSI Installer Python 2.7 (mysql-connector-python-2.1.3-py2.7-winx64.msi)	2.1.3	1.5M	Download
Windows (x86, 64-bit), MSI Installer Python 3.3 (mysql-connector-python-2.1.3-py3.3-winx64.msi)	2.1.3	1.5M	Download
Windows (x86, 64-bit), MSI Installer Python 3.4 (mysql-connector-python-2.1.3-py3.4-winx64.msi)	2.1.3	1.5M	Download

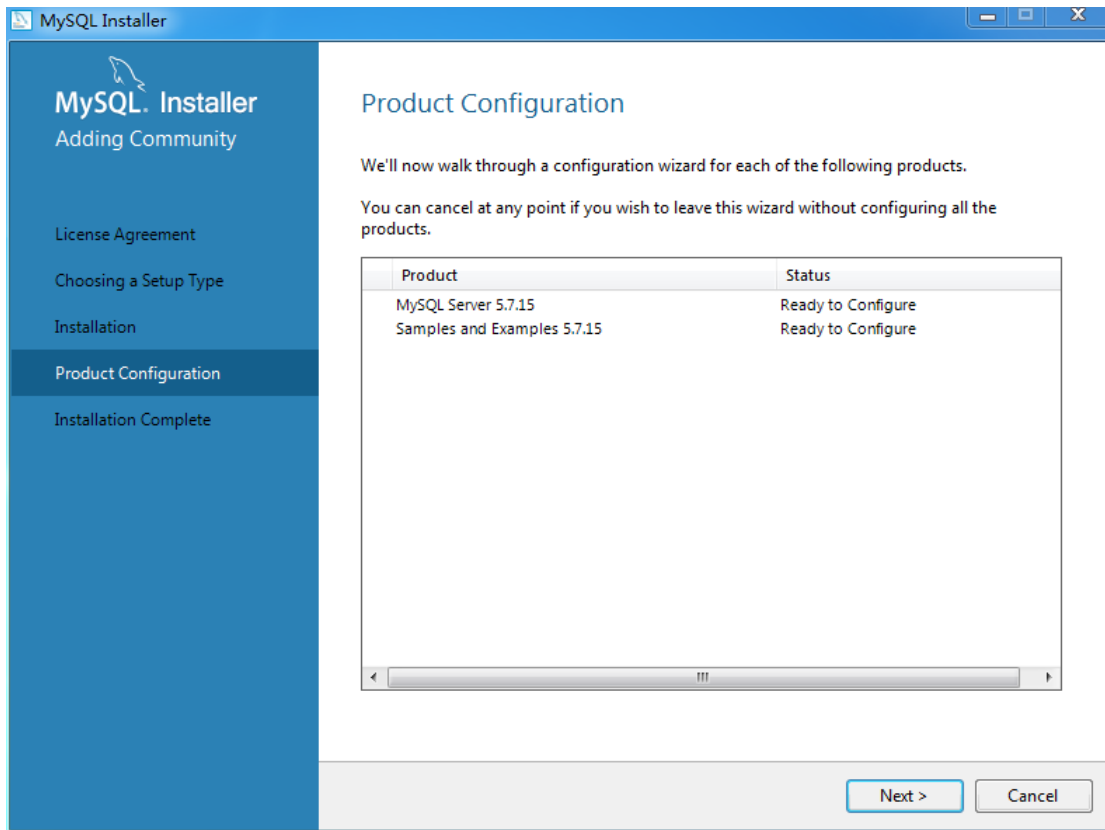
No Python 3.5!!

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



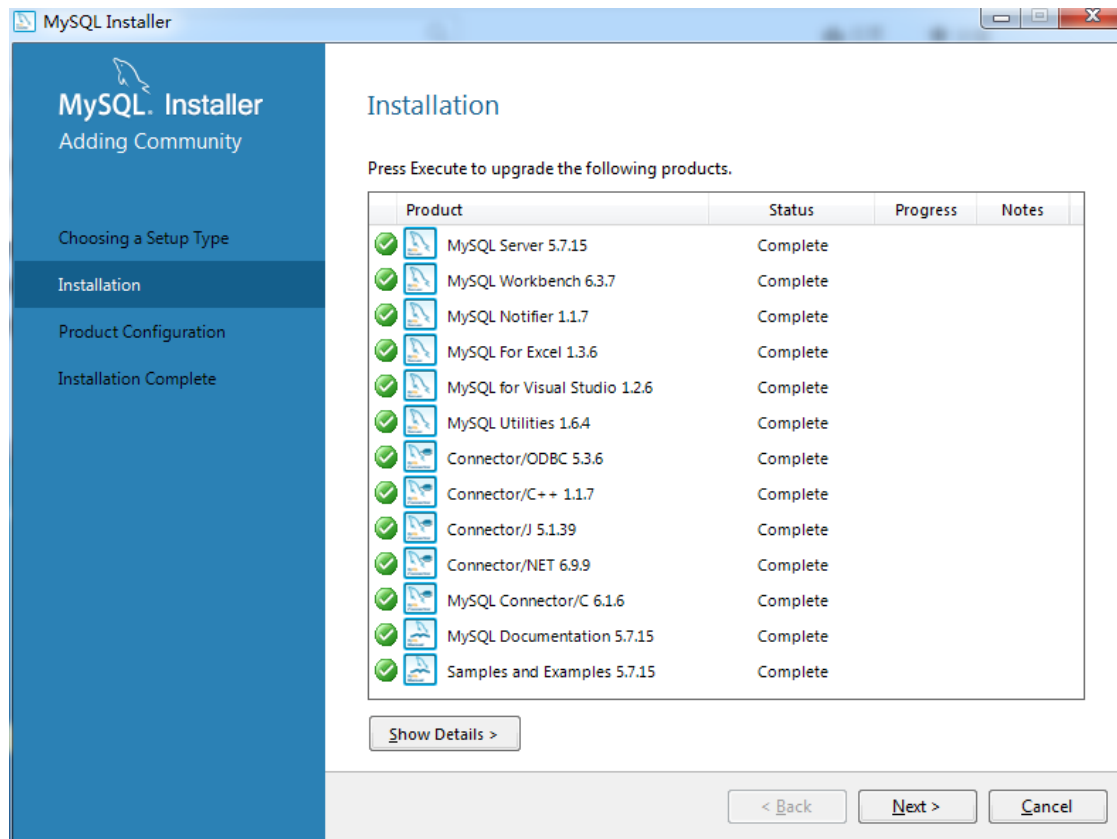
Connect to Database

Product Configuration

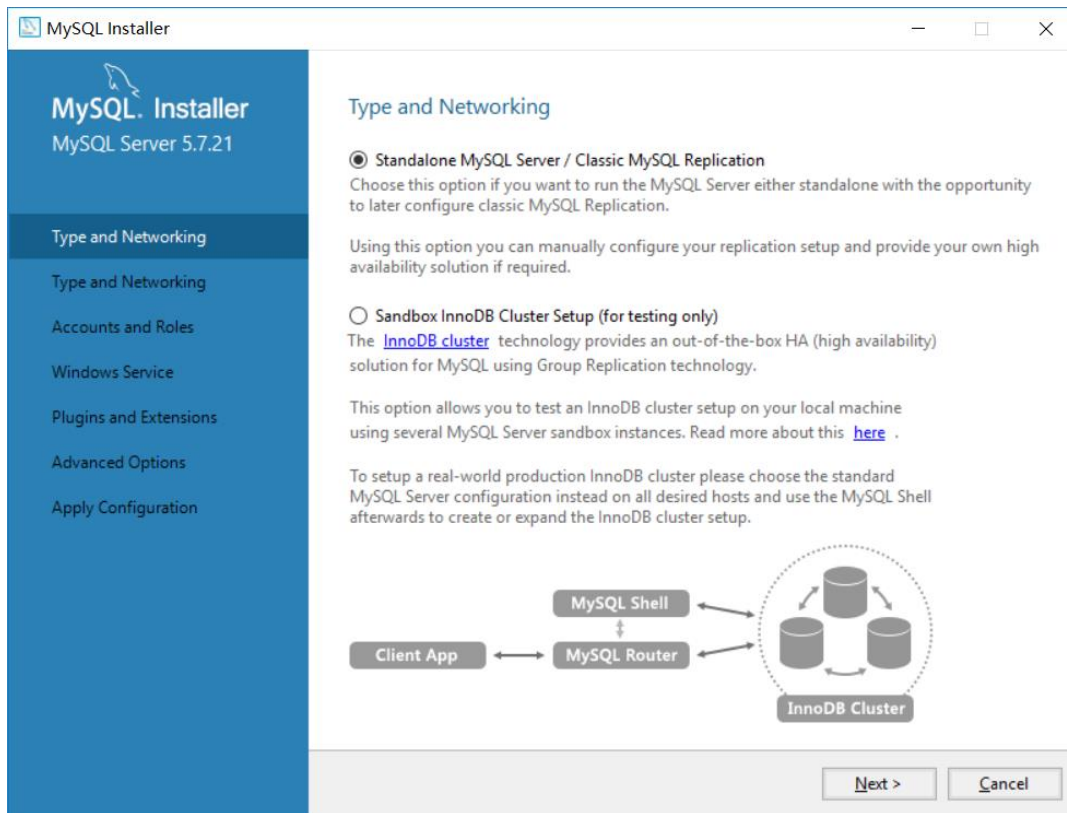


Connect to Database

Installation



Connect to Database



Connect to Database

Show Advanced Options

The screenshot shows the MySQL Installer window for MySQL Server 5.7.15. The left sidebar lists several tabs: 'Type and Networking' (selected), 'Accounts and Roles', 'Windows Service', 'Plugins and Extensions', 'Advanced Options', and 'Apply Server Configuration'. The main area is titled 'Type and Networking' and contains the following sections:

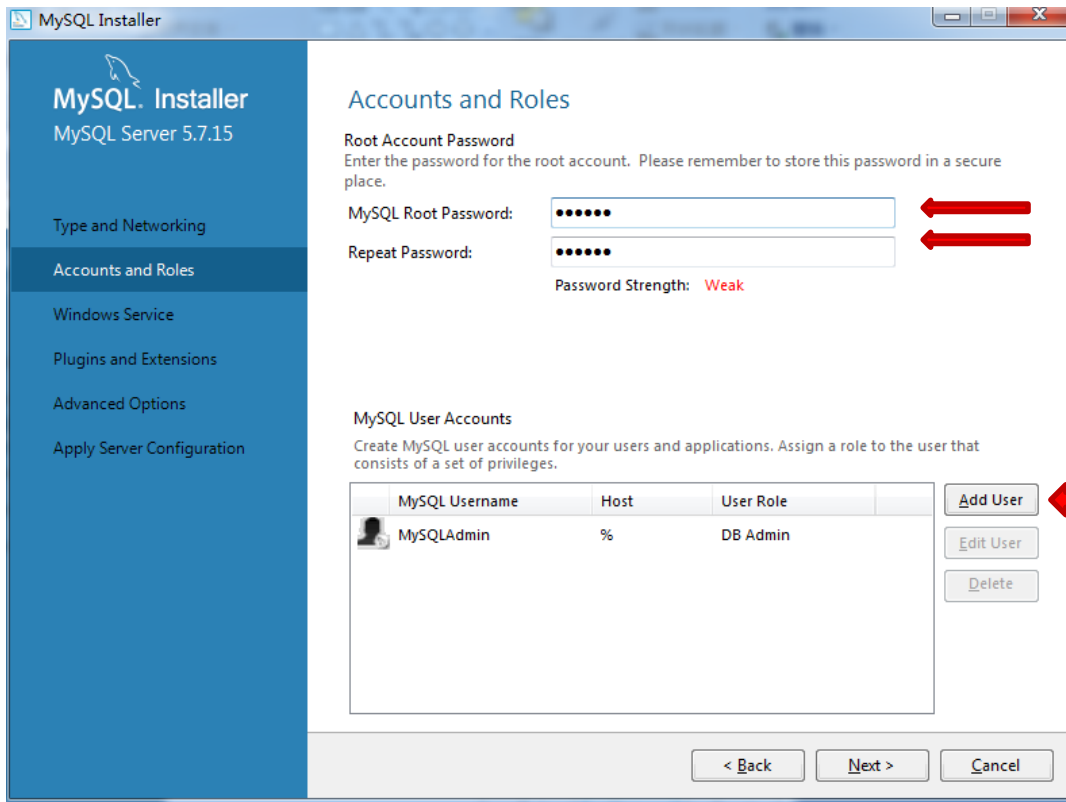
- Server Configuration Type**: A text box explaining that this setting defines how much system resources are assigned to the MySQL Server instance.
- Config Type**: A dropdown menu currently set to 'Development Machine'.
- Connectivity**: A section with the instruction 'Use the following controls to select how you would like to connect to this server.' It includes:
 - ☒ TCP/IP: Port Number: 3306
 - ☒ Open Firewall port for network access
 - ☐ Named Pipe: Pipe Name: MYSQL
 - ☐ Shared Memory: Memory Name: MYSQL
- Advanced Configuration**: A section with the instruction 'Select the checkbox below to get additional configuration page where you can set advanced options for this server instance.' It includes:
 - ☒ Show Advanced Options

A large red arrow points to the 'Show Advanced Options' checkbox. At the bottom right, there are 'Next >' and 'Cancel' buttons.



Connect to Database

Account and Roles



The screenshot shows the MySQL Installer window for MySQL Server 5.7.15. The left sidebar contains navigation options: Type and Networking, Accounts and Roles (selected), Windows Service, Plugins and Extensions, Advanced Options, and Apply Server Configuration. The main area is titled 'Accounts and Roles' and contains two sections: 'Root Account Password' and 'MySQL User Accounts'.

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password:

Repeat Password:

Password Strength: **Weak**

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

MySQL Username	Host	User Role	
 MySQLAdmin	%	DB Admin	Add User Edit User Delete

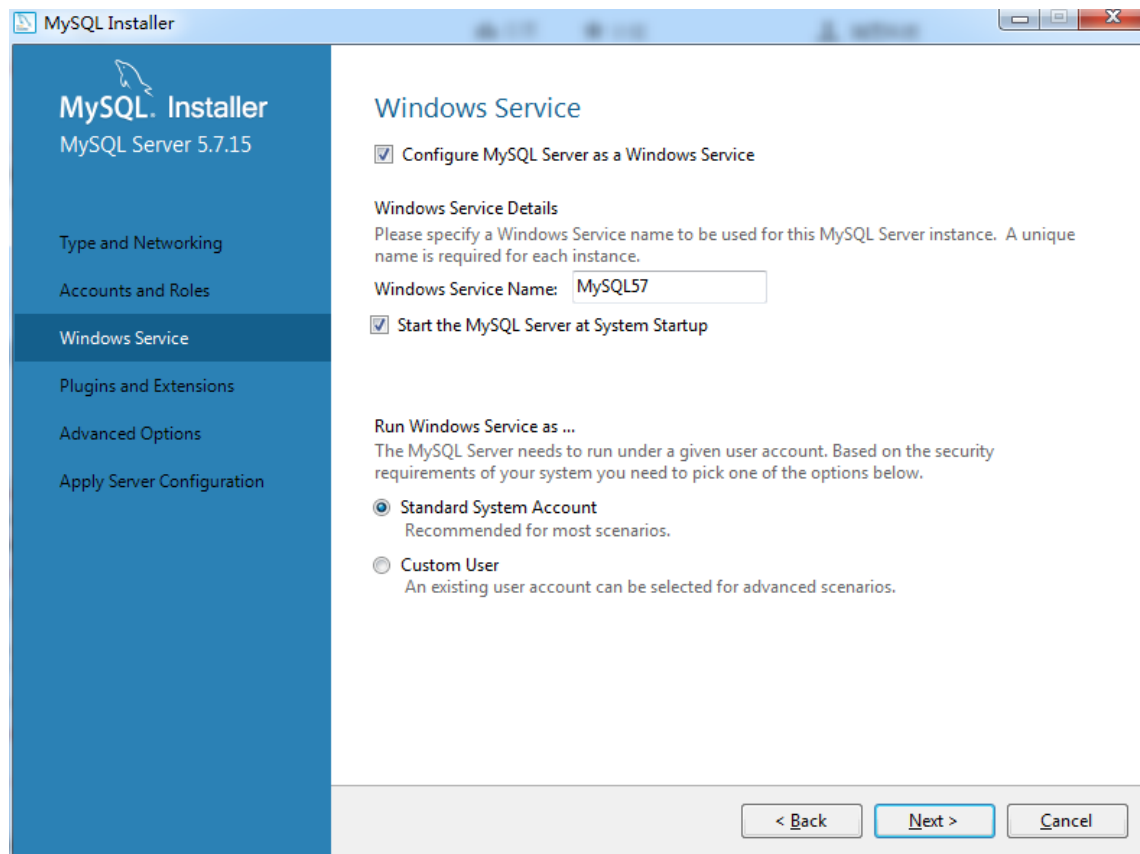
Navigation buttons at the bottom: < Back, Next >, Cancel.

Red arrows point to the password fields, the 'Add User' button, and the 'Weak' password strength indicator.



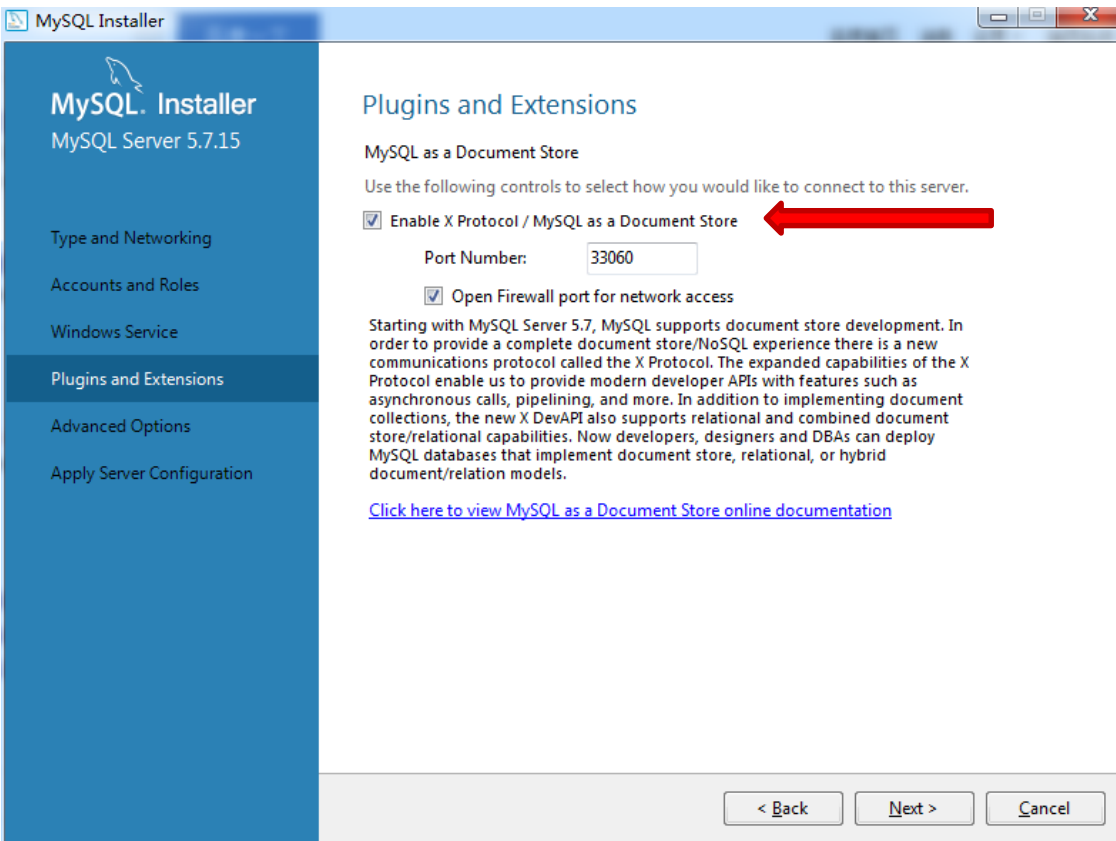
Connect to Database

Next



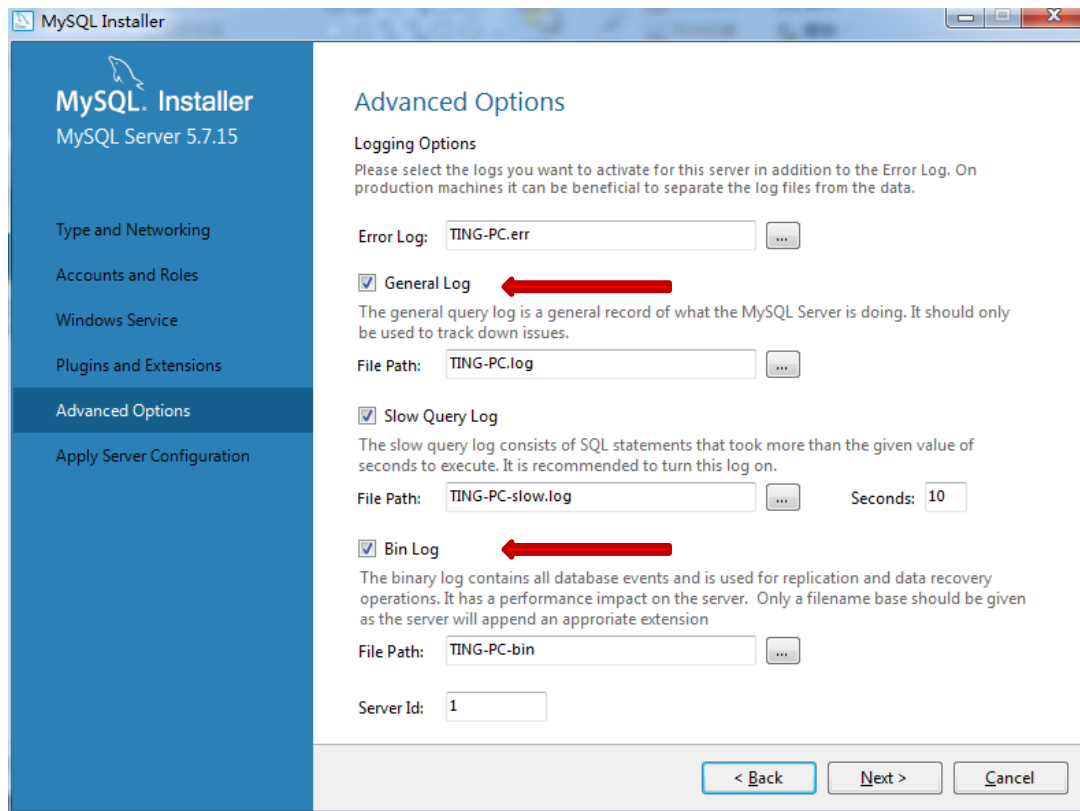
Connect to Database

New MySQL Function for NoSQL



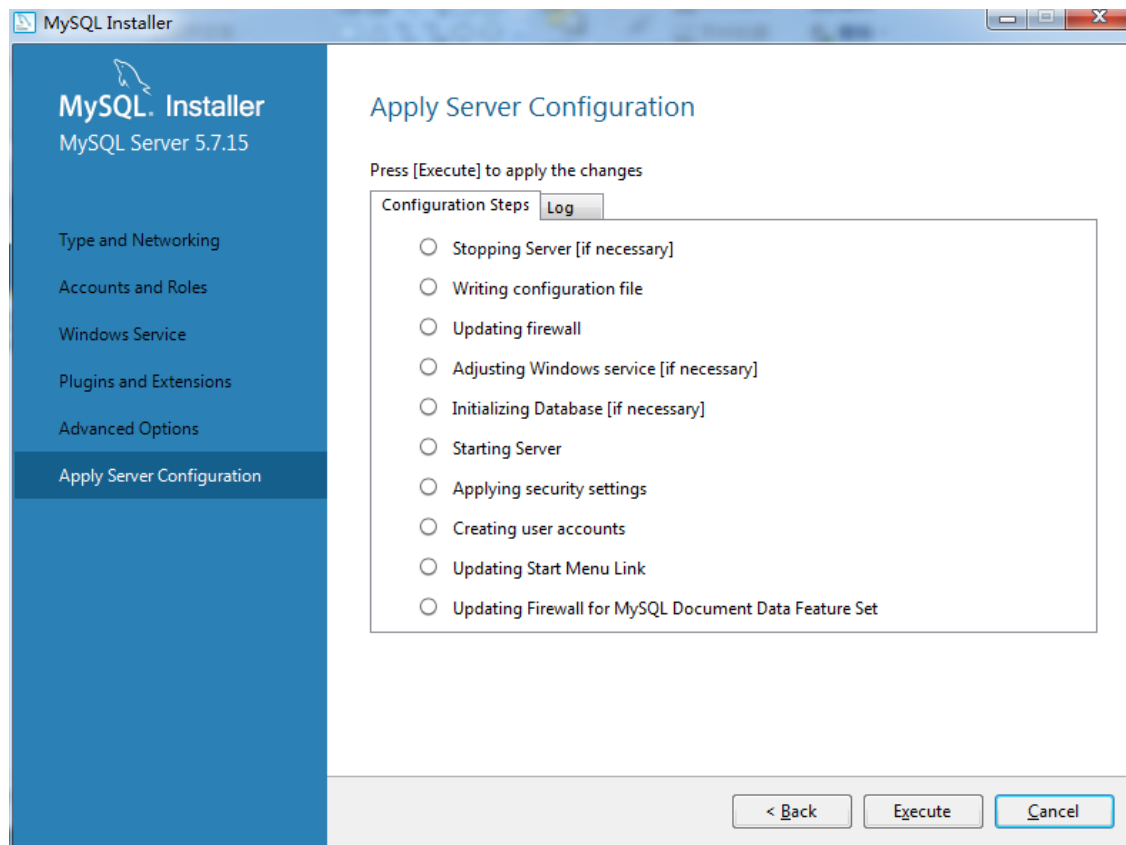
Connect to Database

Logs



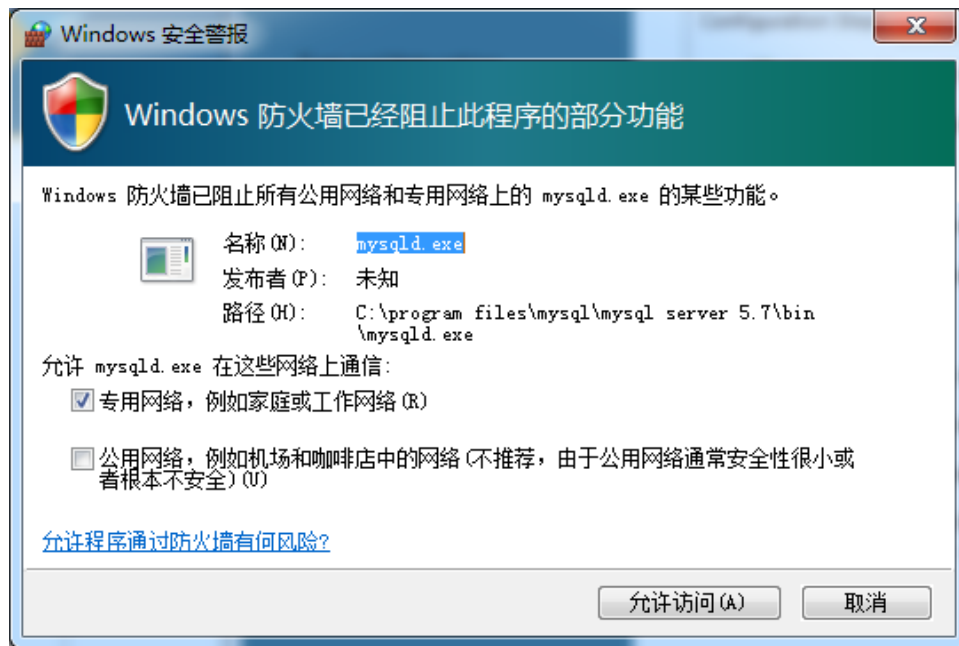
Connect to Database

MySQL Server Server Configuration



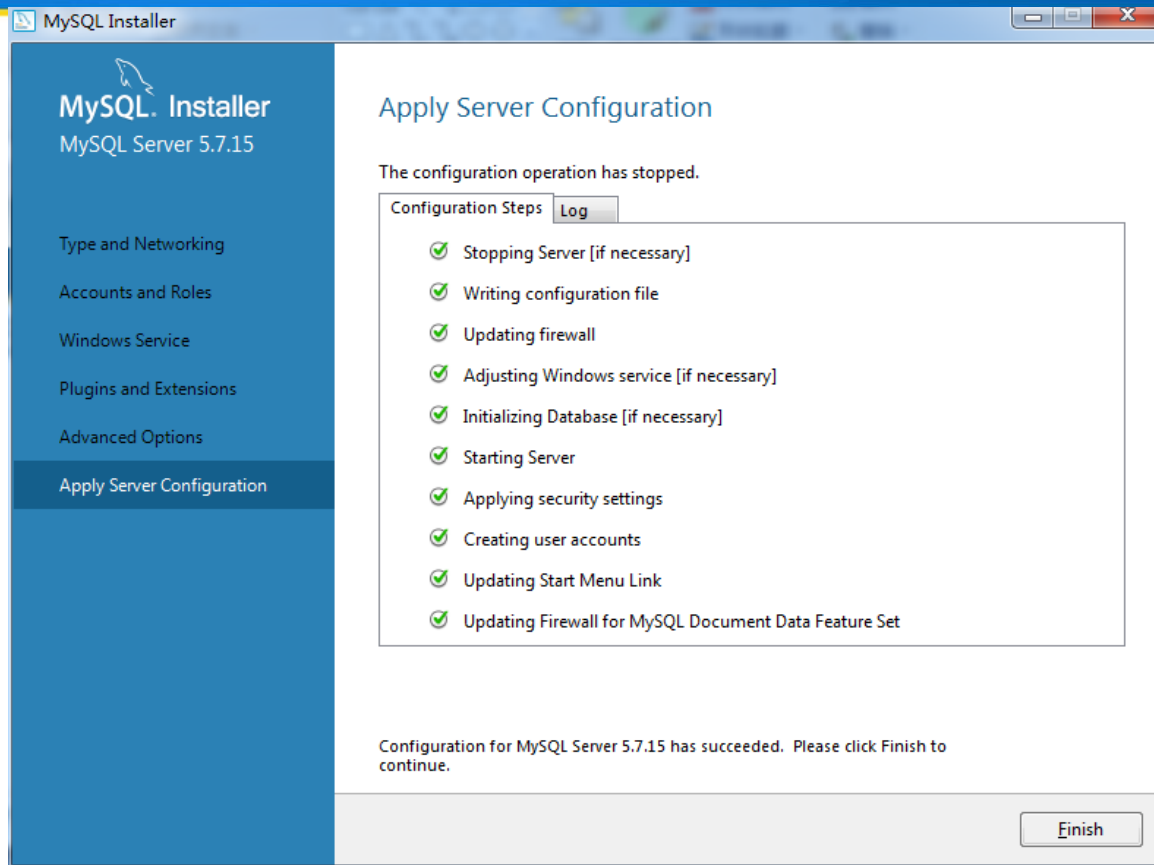
Connect to Database

Sometimes, maybe...if you are using Wifi



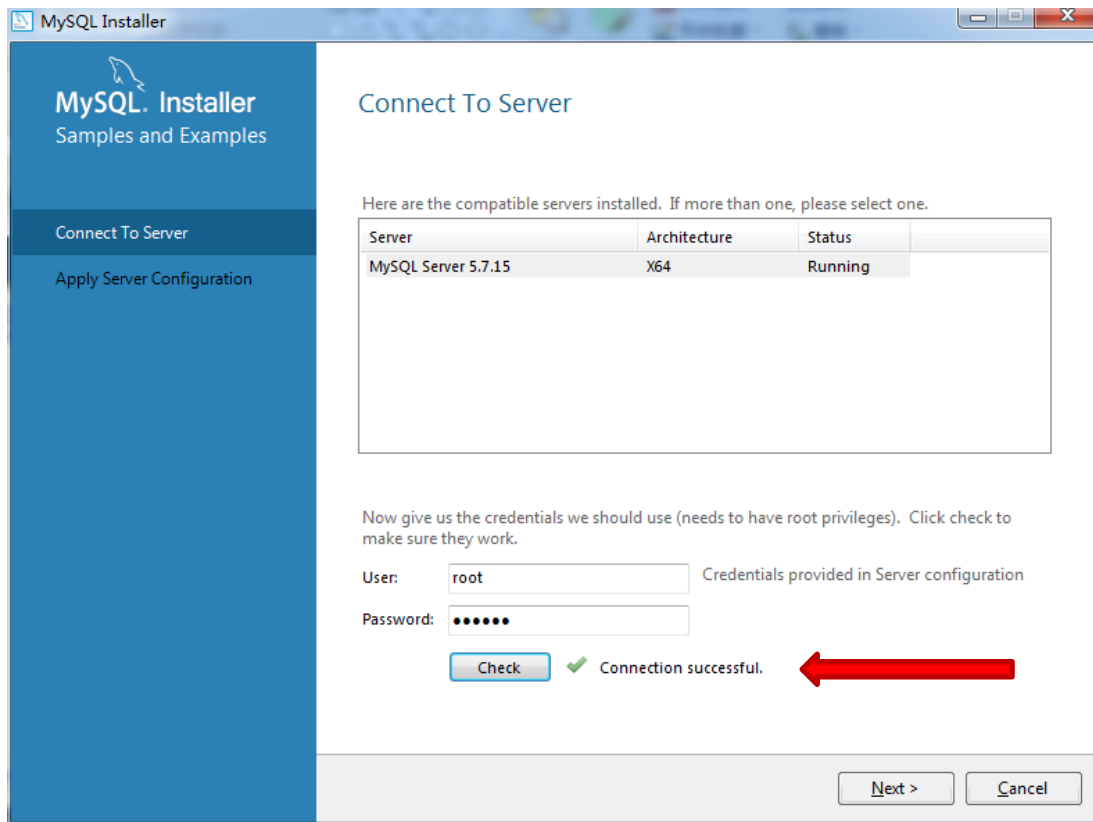
Connect to Database

MySQL Server Apply Server Configuration

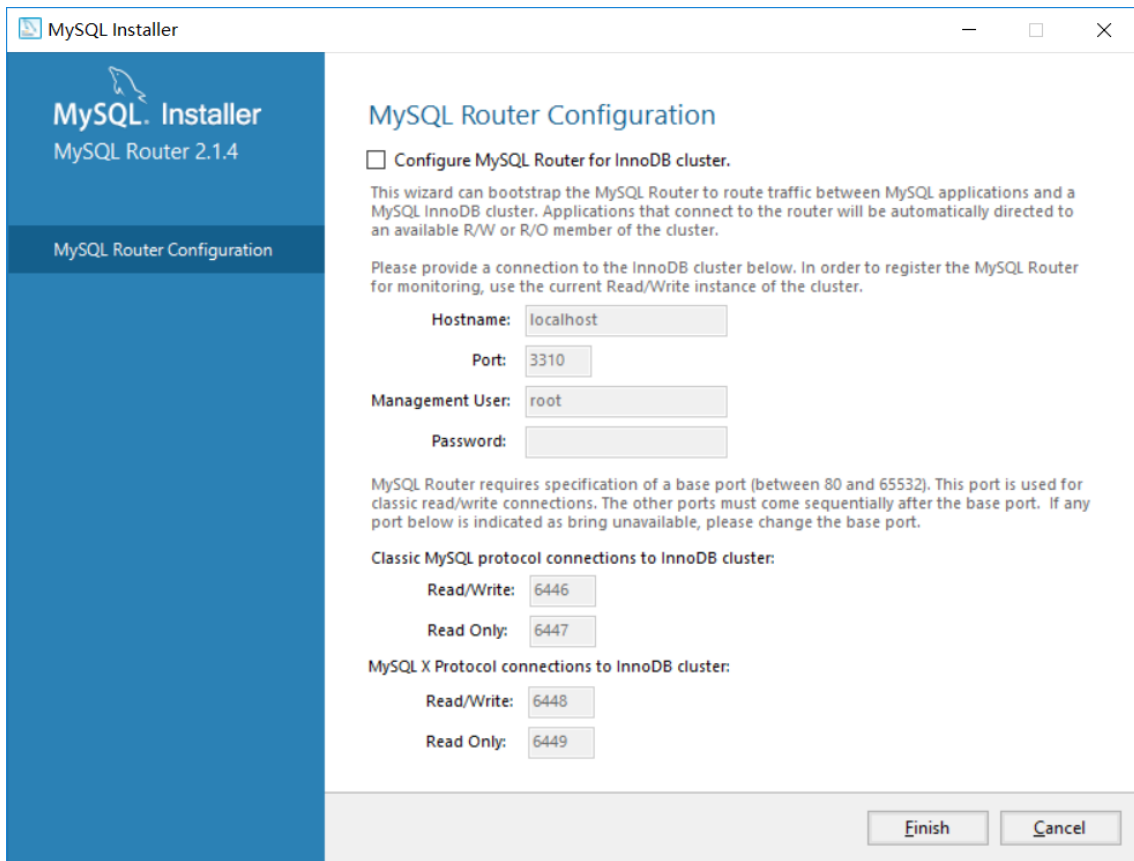


Connect to Database

*Prepare to install
Samples
– Check!*



Connect to Database



The image shows a screenshot of the MySQL Installer application window, specifically the "MySQL Router Configuration" step. The window has a title bar that says "MySQL Installer". On the left side, there is a blue sidebar with the text "MySQL. Installer" and "MySQL Router 2.1.4". Below this, the current step "MySQL Router Configuration" is highlighted. The main area of the window contains the following content:

MySQL Router Configuration

☐ Configure MySQL Router for InnoDB cluster.

This wizard can bootstrap the MySQL Router to route traffic between MySQL applications and a MySQL InnoDB cluster. Applications that connect to the router will be automatically directed to an available R/W or R/O member of the cluster.

Please provide a connection to the InnoDB cluster below. In order to register the MySQL Router for monitoring, use the current Read/Write instance of the cluster.

Hostname:

Port:

Management User:

Password:

MySQL Router requires specification of a base port (between 80 and 65532). This port is used for classic read/write connections. The other ports must come sequentially after the base port. If any port below is indicated as being unavailable, please change the base port.

Classic MySQL protocol connections to InnoDB cluster:

Read/Write:

Read Only:

MySQL X Protocol connections to InnoDB cluster:

Read/Write:

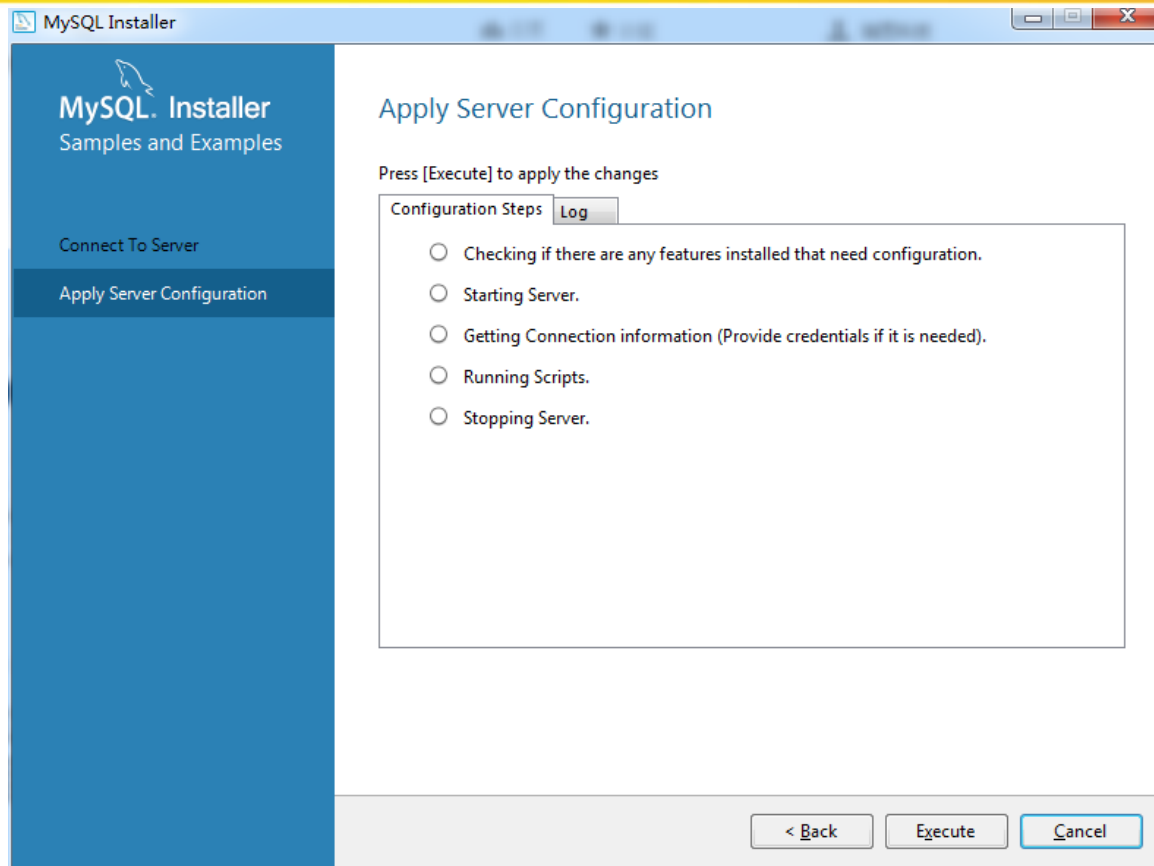
Read Only:

At the bottom right of the window, there are two buttons: "Finish" and "Cancel".



Connect to Database

Samples Apply Server Configuration



Connect to Database

MySQL Installer

MySQL. Installer
Samples and Examples

Connect To Server

Apply Configuration

Connect To Server

Here are the compatible MySQL Svr instances installed in this computer.
Please select the ones where the sample schemas and data will be created.

☒ Show MySQL Server instances maybe running in read-only mode

	Server	Port	Arch...	Type	Status
☒	MySQL Server 5.7.21	3306	X64	Stand-alone Server	Connection succeeded.

Now give us the credentials we should use (needs to have root privileges).
Click "Check" to make sure they work.

User: Credentials provided in Server configuration

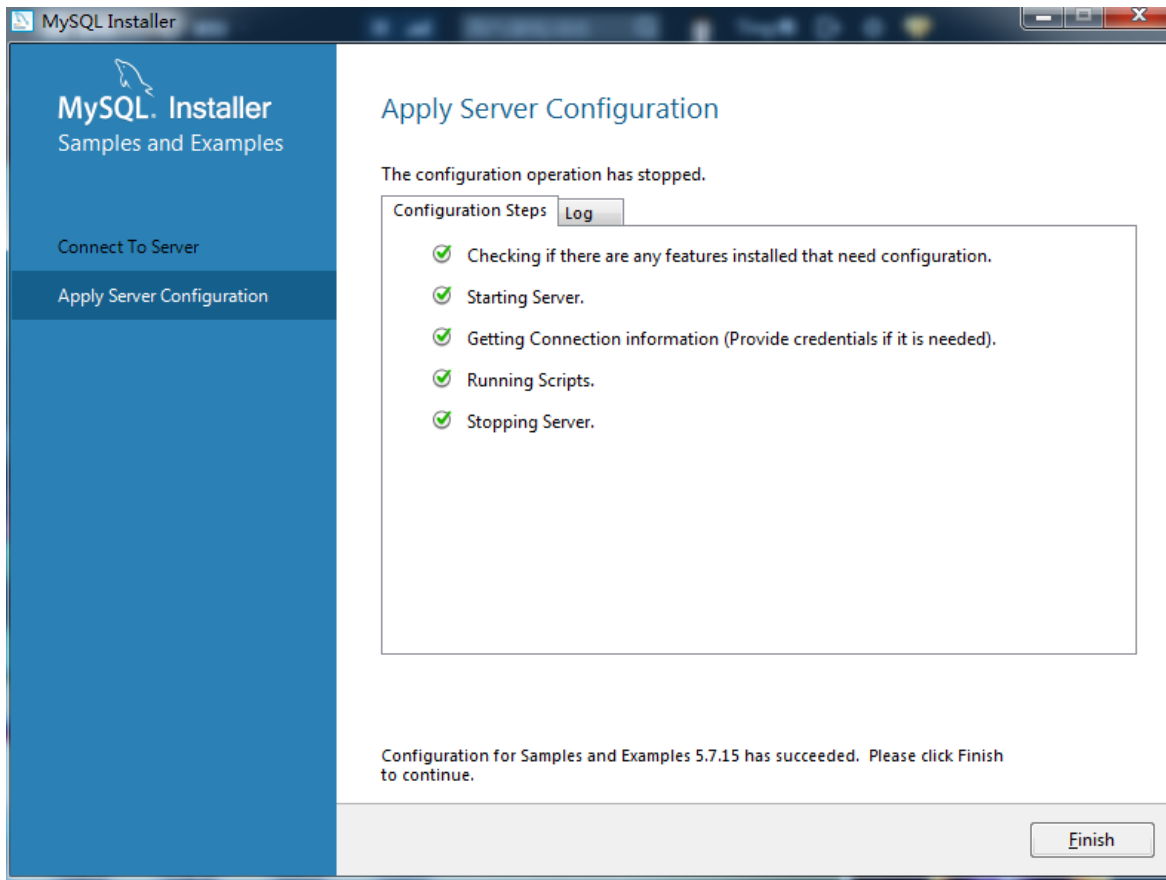
Password:

 All connections succeeded.



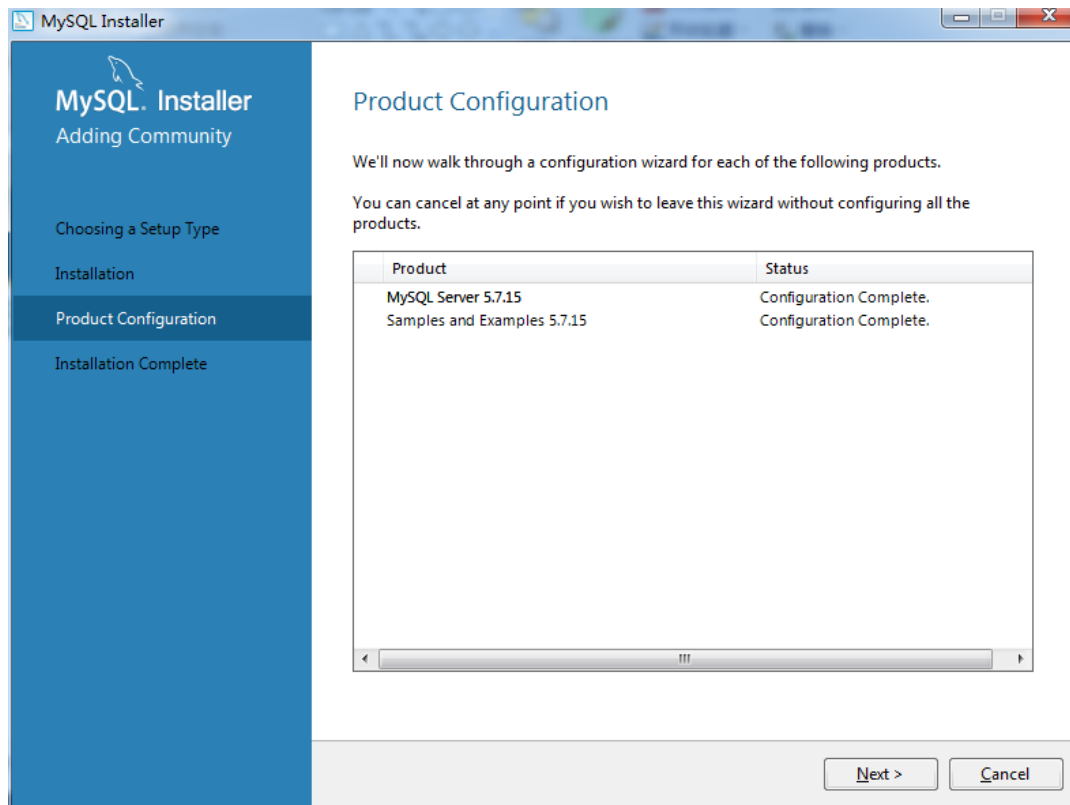
Connect to Database

Finish Server Configuration



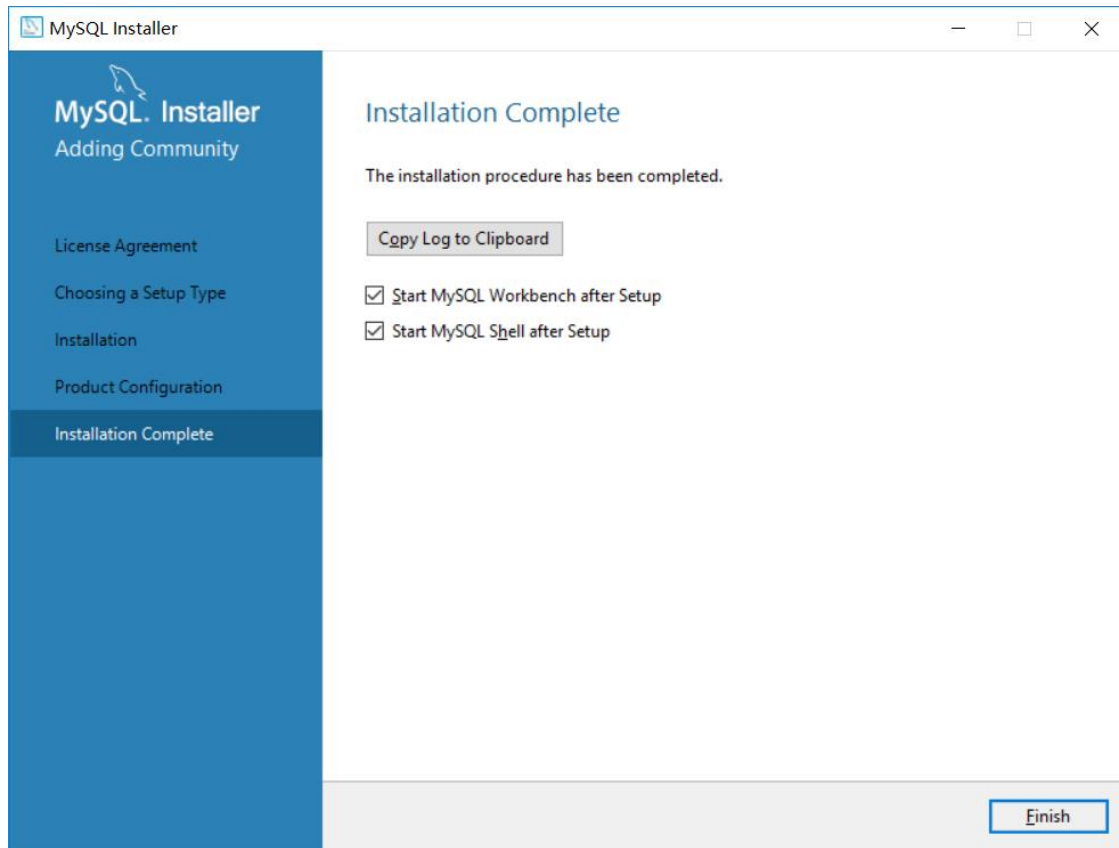
Connect to Database

Next



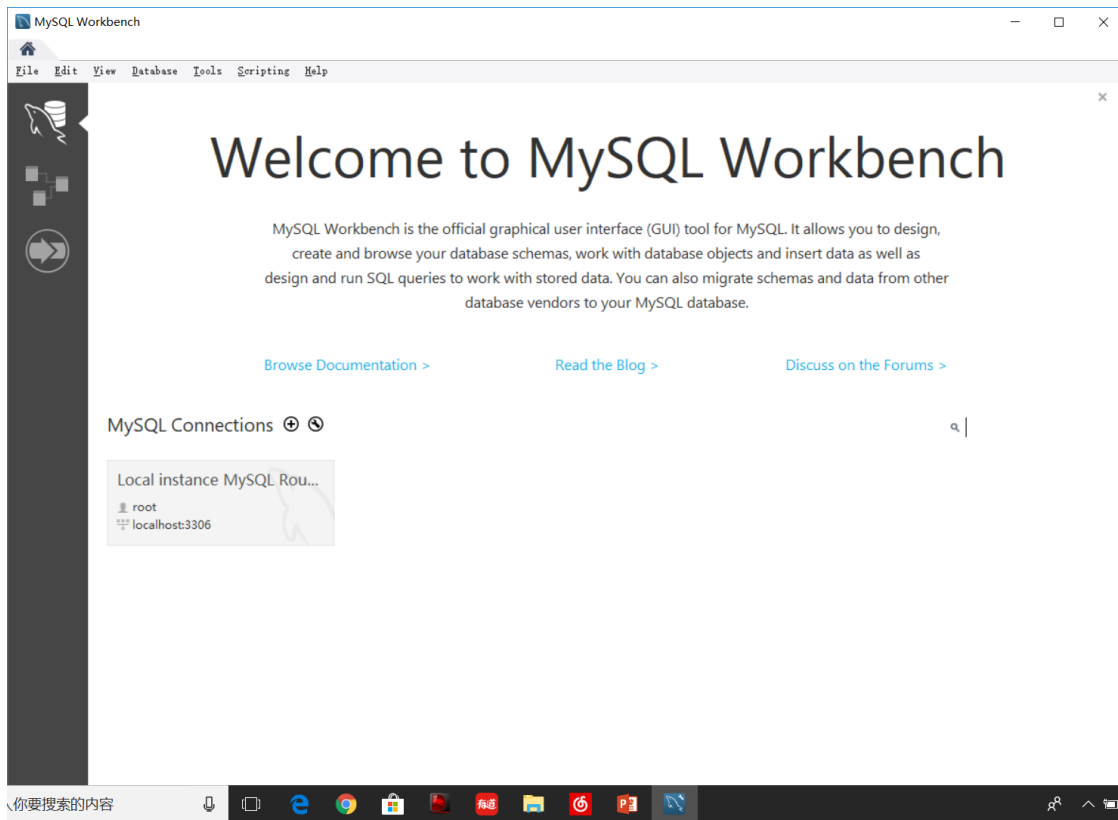
Connect to Database

Complete



Connect to Database

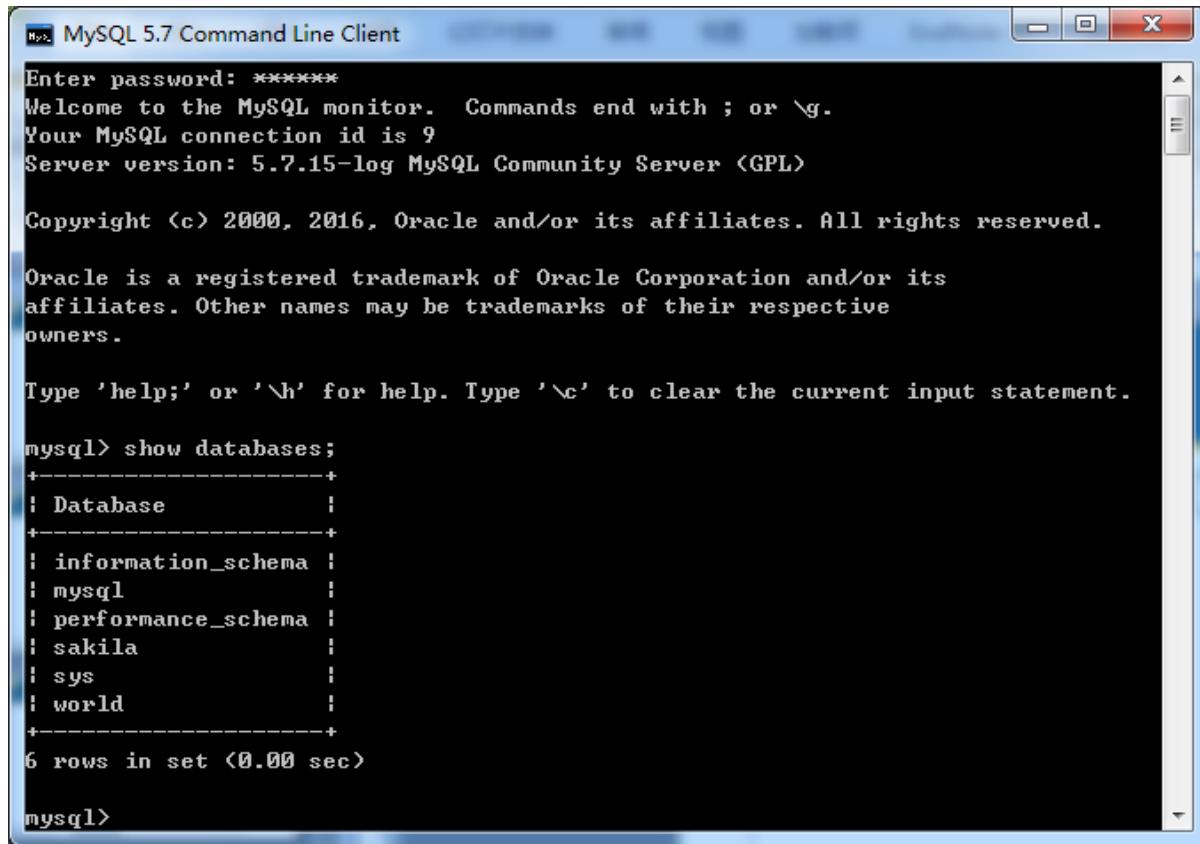
- Welcome



Connect to Database

Test

- *Successful?*
- *Test it!*



```
MySQL 5.7 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \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.

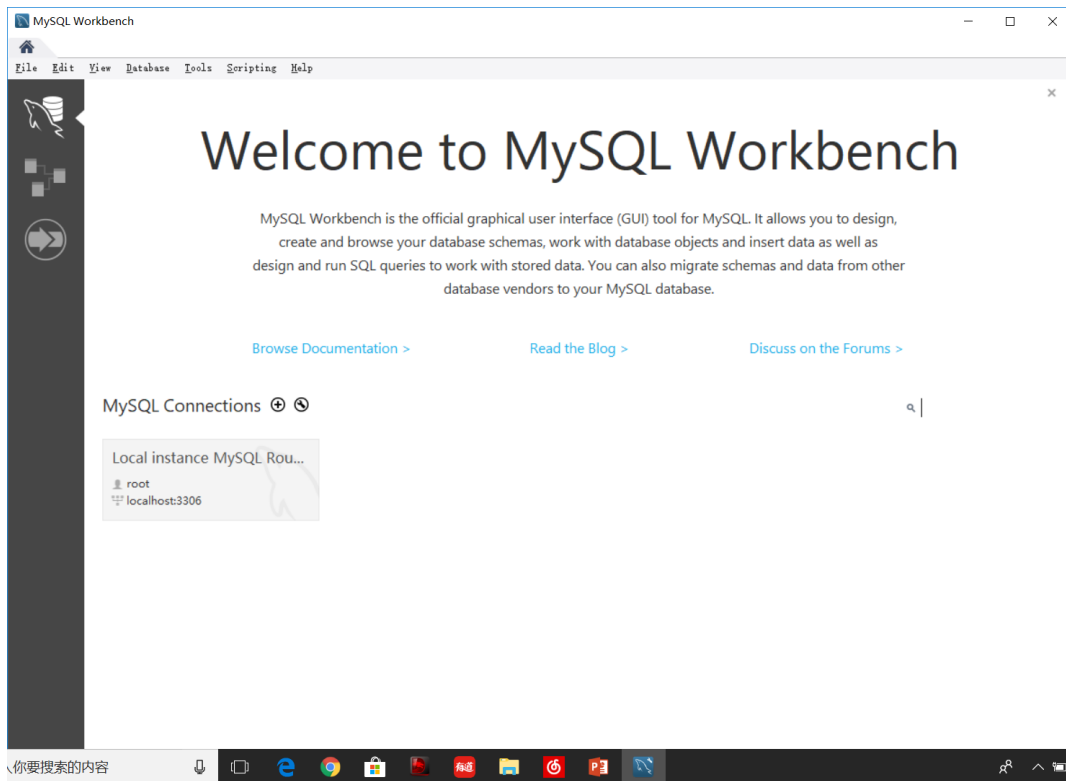
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.00 sec)

mysql>
```



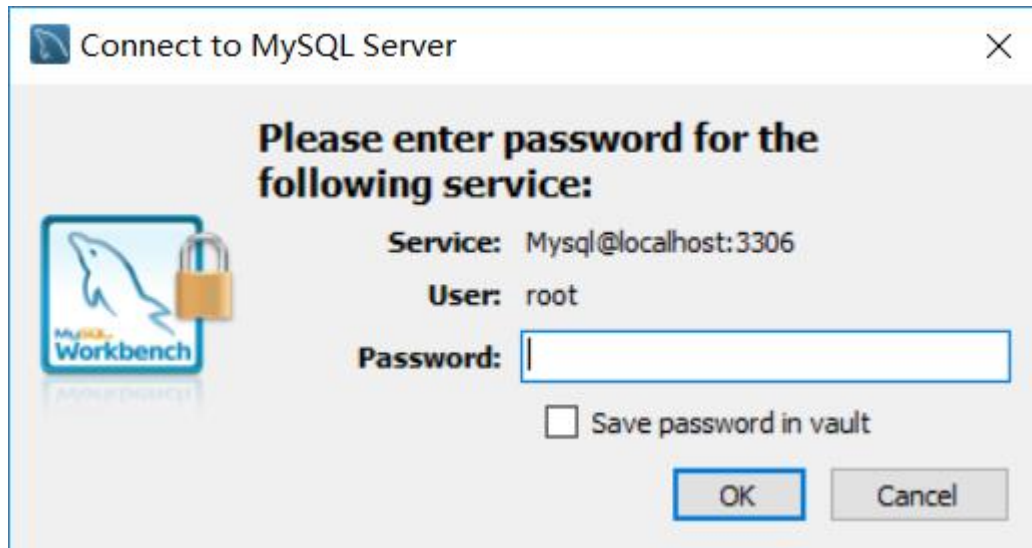
Connect to Database

- Click the “root”

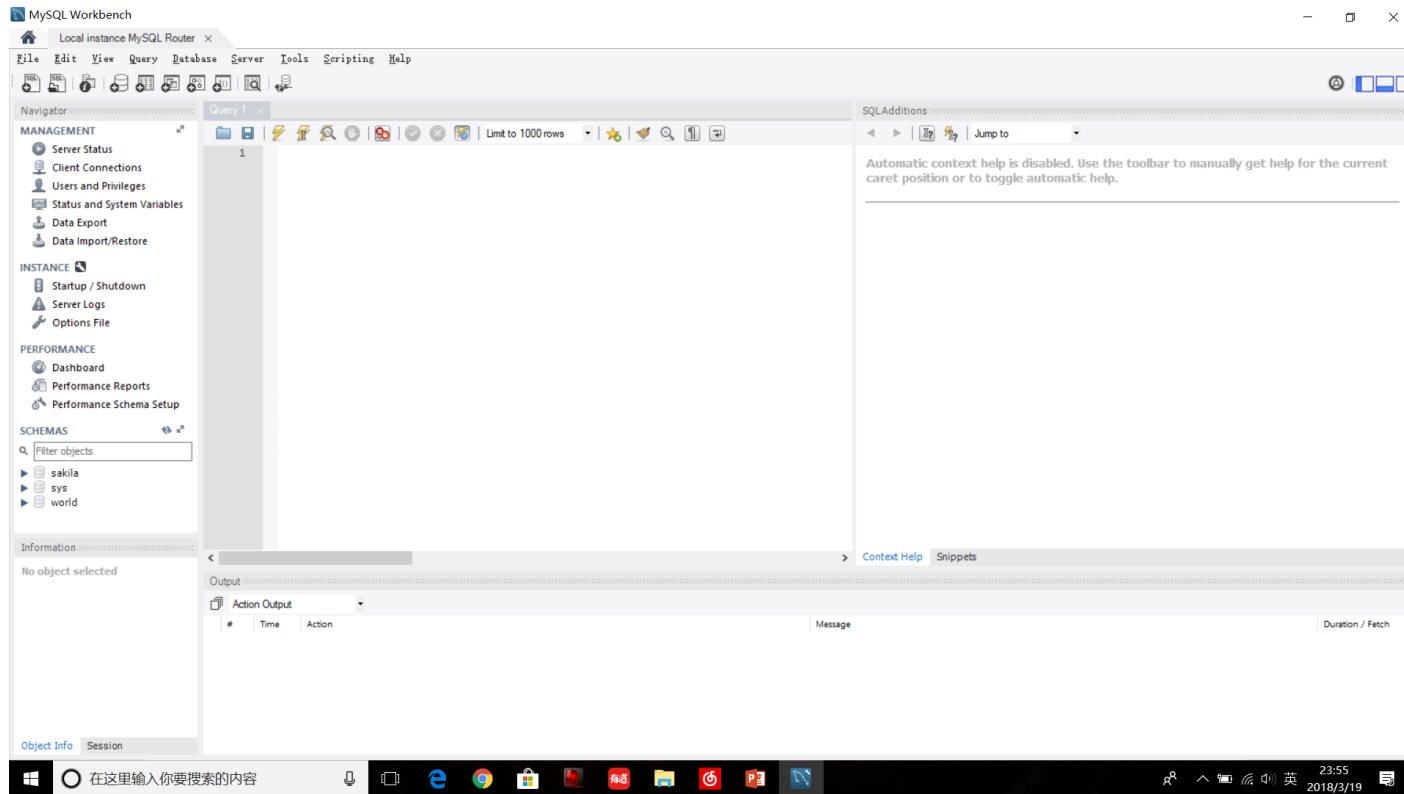


Connect to Database

Start MySQL Workbench



Connect to Database



Connect to Database

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*



Connect to Database

Tips

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





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

an approach to store data

Foundations of Data Base

Foundations of Data Base

Widely Used Data Base

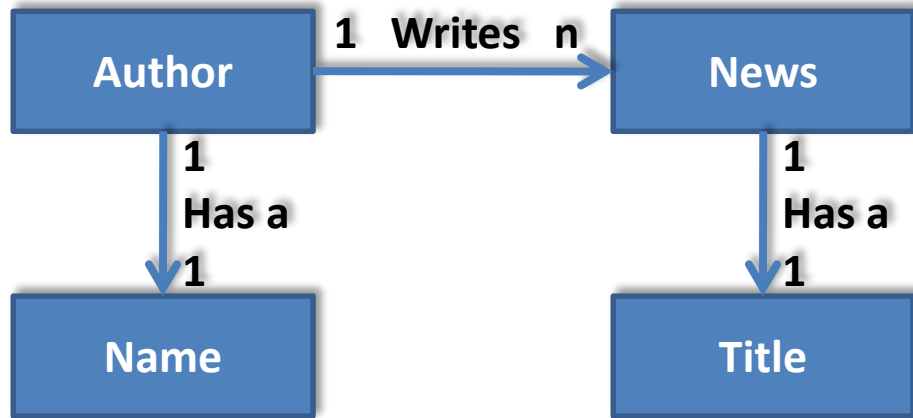
- MySQL
- SQL Server
- Oracle



Foundations of Data Base

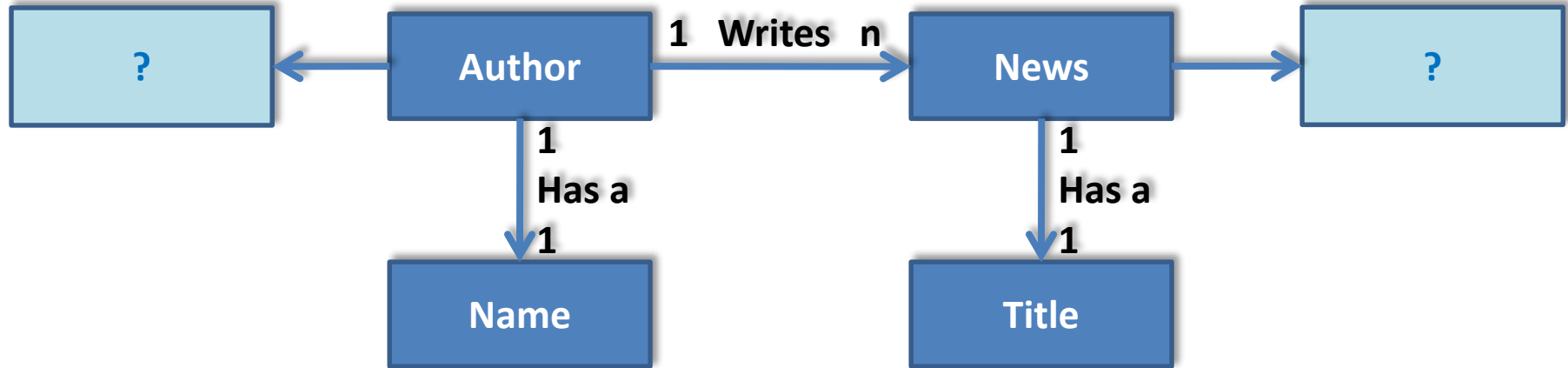
Relational Database

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



Foundations of Data Base

ER Diagram



Foundations of Data Base

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



Foundations of Data Base

Constraints

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



Foundations of Data Base

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).



Foundations of Data Base

Create

CREATE TABLE *TABLE_NAME*

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



Foundations of Data Base

Drop

DROP TAME TABLE_NAME

```
DROP TABLE CUSTOMER|
```



Foundations of Data Base

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')
```



Foundations of Data Base

Delete

DELETE FROM *TABLE_NAME* WHERE *CONDITION*

```
DELETE FROM CUSTOMER WHERE CUSTOMER_NAME='Marie Curie'
```



Foundations of Data Base

Update

UPDATE *TABLE_NAME* SET
COLUMN=COLUMN_VALUE WHERE
CONDITION

```
UPDATE CUSTOMER SET CUSTOMER_PWD='shisu' WHERE CUSTOMER_NAME='Albert Einstein'
```



Foundations of Data Base

Select

SELECT - FROM - WHERE

SELECT *COLUMN_NAME* FROM *TABLE_NAME* WHERE *CONDITION*
GROUP BY *CONDITION*
ORDER BY *CONDITION*

```
SELECT CUSTOMER_NAME FROM CUSTOMER WHERE CUSTOMER_PWD='sisu'
```



Foundations of Data Base

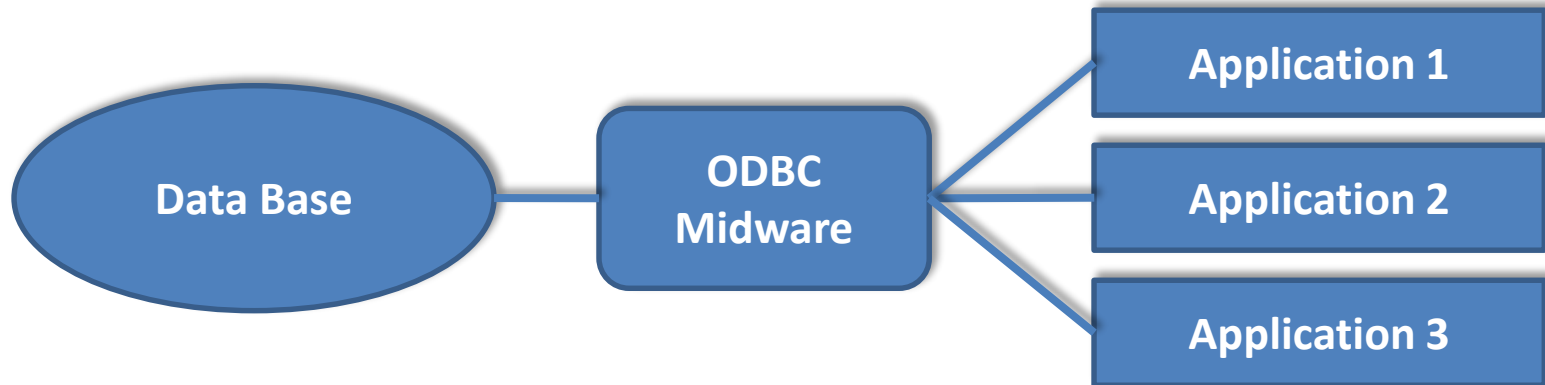
EXAMPLE 4: Data Base



Foundations of Data Base

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.



Foundations of Data Base

Distributed Data Base

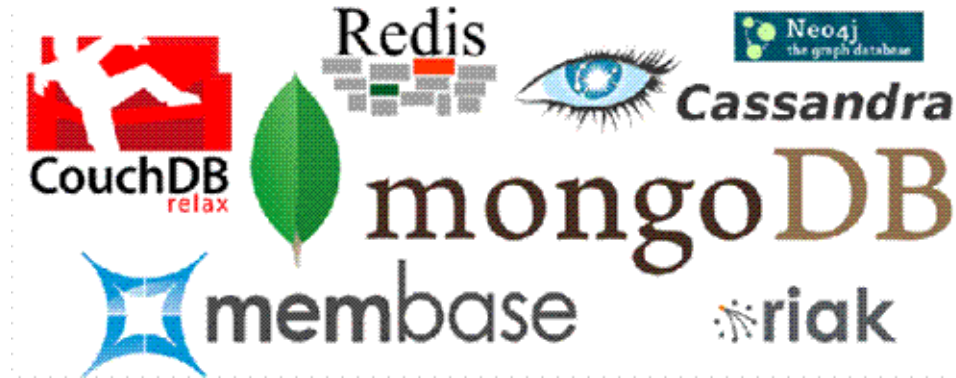
a database in which data are stored in multiple computers, located in the same physical location; or may be dispersed over a network of interconnected computers.



Foundations of Data Base

NoSQL (Not Only SQL)

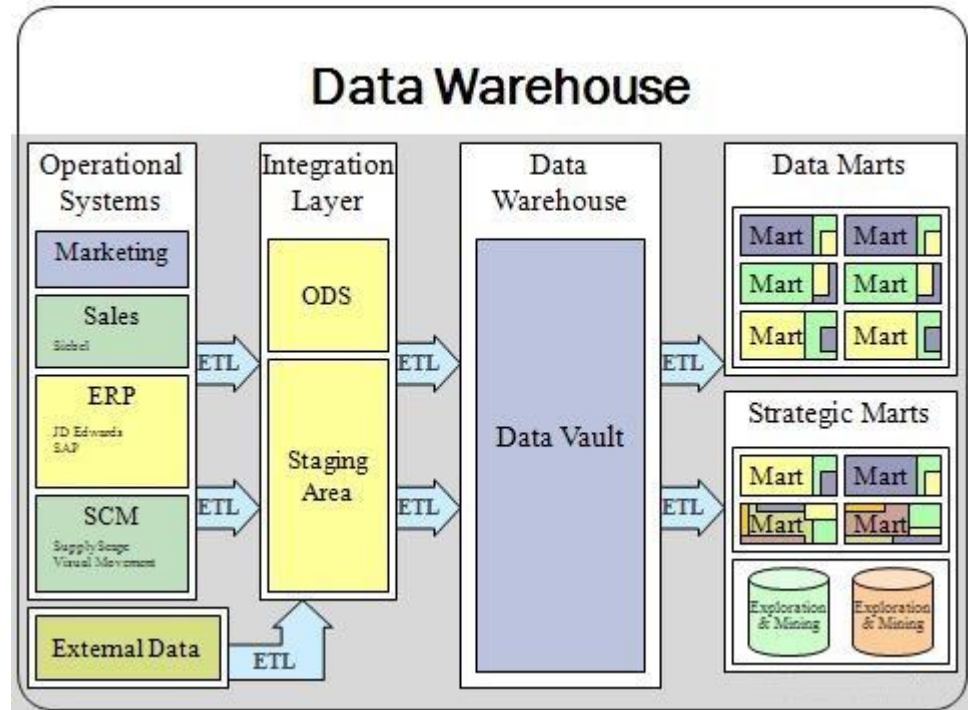
triggered by the needs of Web 2.0 companies such as Facebook, Google, and Amazon.com, NoSQL databases are increasingly used in big data and real-time web applications.



Foundations of Data Base

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



Foundations of Data Base

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

DB Application

DB Application

EXAMPLE 5:
python – MySQL



DB Application

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



DB Application

Create Tables

1. Create a new schema



2. Set as a default schema (right click)

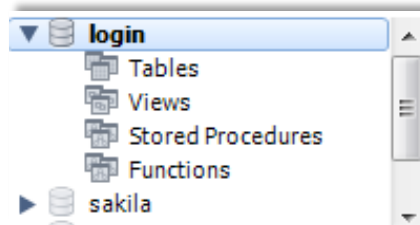
3. Run CreateMySQL.sql



```
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');
```



5. Result

1 x select * from customer

	CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_PWD
1	1	Thomas Edison	sisu
2	2	Albert Einstein	sisu
3	3	Marie Curie	shisu
*	NULL	NULL	NULL

DB Application

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

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

get data from request:

```
if numrows>0:  
    print('Hello, Thomas Edison! ' )  
else:  
    print('Hi!You are wrong! ' )
```

Result:



```
"C:\Program Files\Python35\python.exe"  
Hi!You are wrong!  
  
Process finished with exit code 0
```



EXAMPLE 6: Flask Login with DB



Adam .

Sign-in options

DB Application

```
from flask import Flask
from flask import request
import pymysql
```

```
app = Flask(__name__)
```

```
@app.route('/', methods=['GET', 'POST'])
def home():
    return '''<h1>Home</h1>
    <p><a href="/login">Login</a></p>'''
```

```
@app.route('/login', methods=['GET'])
def login_form():
    return '''<form action="/login" method="post">
    <p><input name="username"></p>
    <p><input name="password" type="password"></p>
    <p><button type="submit">Log In</button></p>
    </form>'''
```

```
@app.route('/login', methods=['POST'])
def login():
```

```
    # get data from database:
```

```
    # 连接到MySQL数据库
```

```
    # 1.Connection Open
```

```
    conn = pymysql.connect(user='root', password='123456', database='login')
```

```
    # 2.Cursor Creating:
```

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

```
    # 3. SQL Execution
```

```
    # 执行SQL语句, 循环插入记录:
```

```
    sqlstr='select * from customer where
customer_name=\''+request.form['username']+\'
\' and customer_pwd=\''+request.form['password']+\' \'
```

```
    # 4. Cursor Moving
```

```
    # 体验游标
```

```
    # 执行, 游标移至当前位置
```

```
    cursor.execute(sqlstr)
```

```
    numrows = int(cursor.rowcount)
```

```
    # 提交事务:
```

```
    conn.commit()
```

```
    # 5.Connection Close
```

```
    # 关闭Cursor:
```

```
    cursor.close()
```

```
    # 关闭Connection:
```

```
    conn.close()
```

```
    # get data from request:
```

```
    if numrows>0:
```

```
        return '<h3>Hello, '+request.form['username']+ '</h3>'
```

```
    return '<h3>Bad username or password.</h3>'
```

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





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

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

数据库系统概论（第5版）

王珊，萨师煊 著

高等教育出版社





上海外国语大学
SHANGHAI INTERNATIONAL STUDIES UNIVERSITY

Home Work

*Design a simple web site and connect your databases to this web site (**deadline: Nov.9**)*

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





The End of Lecture 3

Thank You

<http://www.wangting.ac.cn>

