

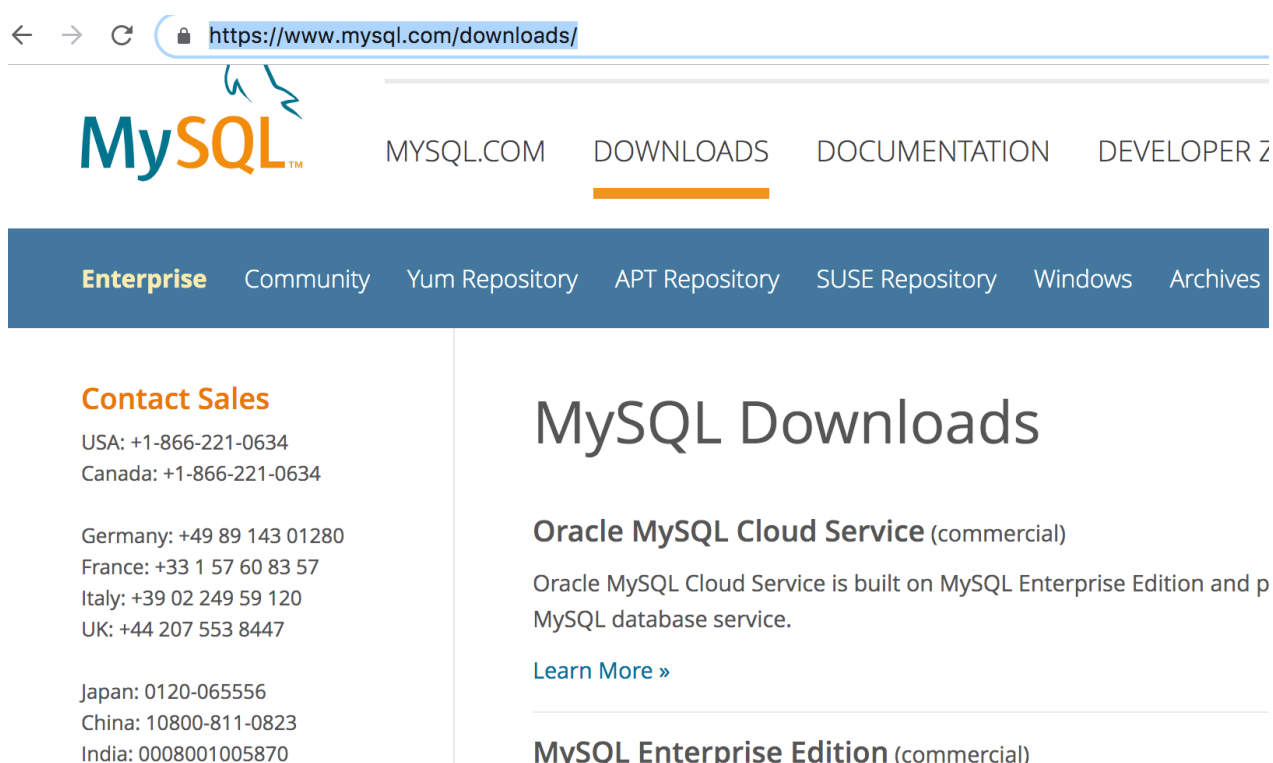
菜鸟学Python趣味项目-数据库

1.准备环境

这次的数据库，主要熟悉MySQL和Mongodb,目前这两个数据库是主流的数据库，也是用的非常多！在企业里面MySQL 会多一些，当然Mongodb也有很多中小企业在用。

1).安装环境

大家先在自己的电脑上安装MySQL和Mongodb这个两个数据库，具体可以百度一下。



备注：Mysql需要配置一下配置文件。

What is MongoDB?

MongoDB is a document database with the scalability and flexibility with the querying and indexing that you need

[Download the latest release](#)

4.0

[Explore the documentation](#)

如果大家不想安装的话，我会提供Linux云服务给大家，里面有我安装好的MySQL和Mongodb

2).数据库的可视化工具

一般我们操作数据库都是用可视化的工具，这些工具比较多，比如有大名鼎鼎的Navicat系列。



Robo 3T对应的是管理Mongodb

Navicat 是一个比较有名的数据库管理工具，但是需要收费，需要破解！

当然工具有很多也可以用其他的比如workbench

- › MySQL on Windows
- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell

Download MySQL Workbench

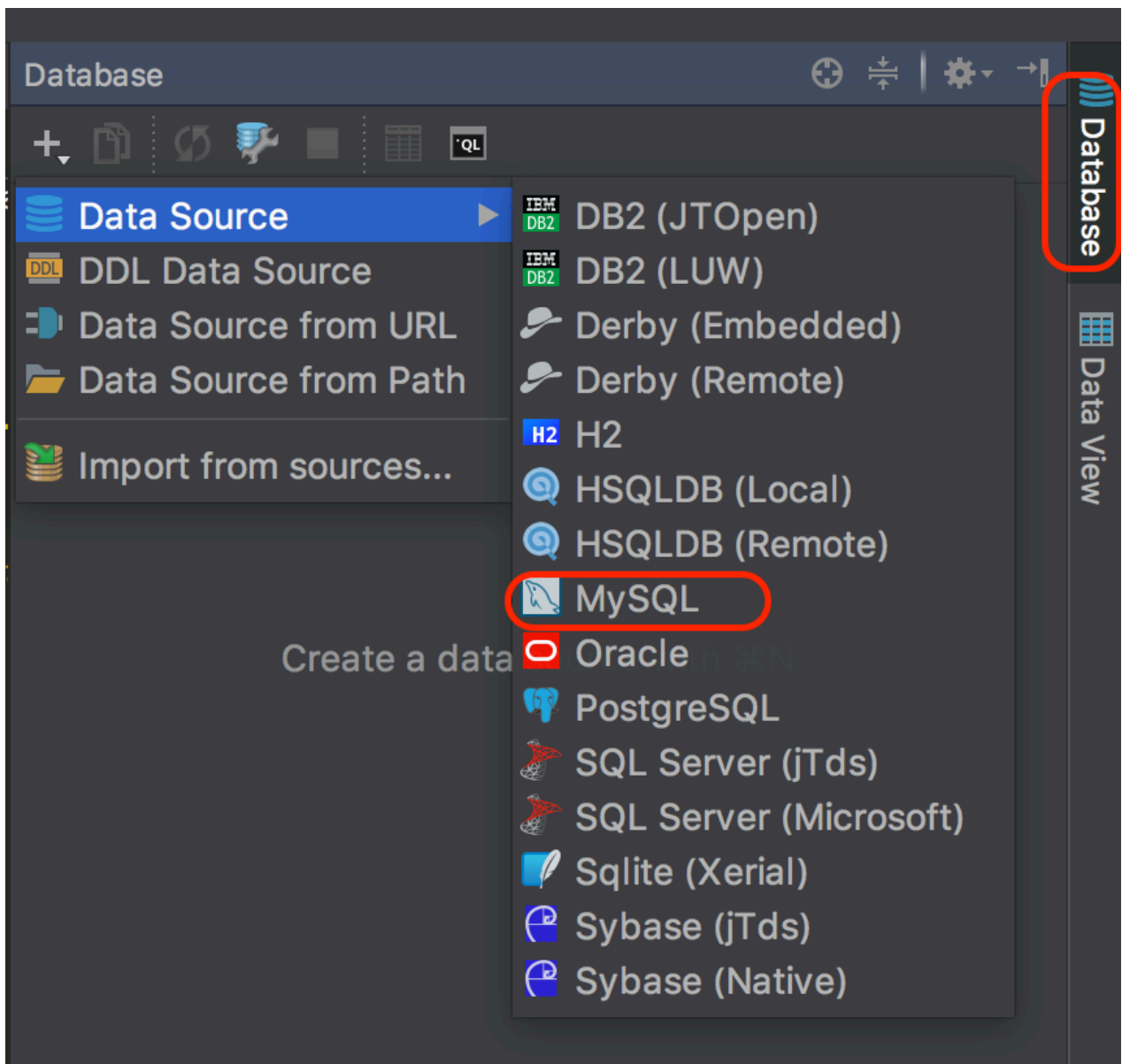
MySQL Workbench provides DBAs and developers an integrated tools environment for:

- Database Design & Modeling
- SQL Development
- Database Administration
- Database Migration

The Community (OSS) Edition is available from this page under the [GPL](#).

Download source packages of LGPL libraries: [\[+\]](#)

3).用Pycharm连数据库



选择MySQL然后配置一下即可

Name: @129.211.

General

SSH/SSL

Schemas

Options

Advanced

Host:

129.211.125.85

Database:

User:

Password:

●●●●

URL:

jdbc:mysql://129.211.125.85:3306

Overrides settings above

Test Connection

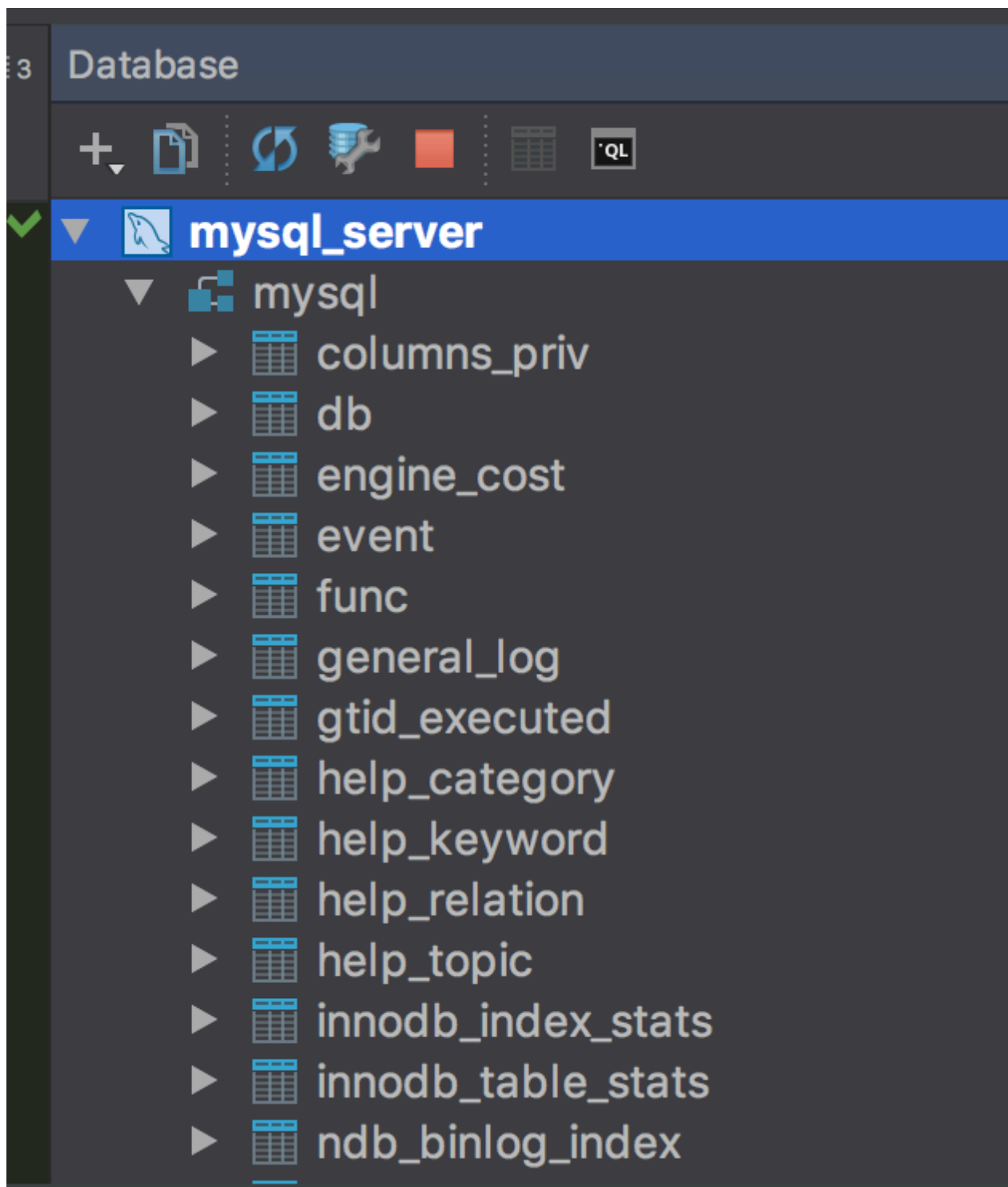
Successful

Details

Driver:

MySQL

然后连接一下，选择mysql 表即可



当然也可以用Navicat连接

ty_mysql_server

information_schema

mysql

表

columns_priv

db

engine_cost

event

func

general_log

gtid_executed

help_category

help_keyword

help_relation

help_topic

innodb_index_stats

innodb_table_stats

ndb_binlog_index

plugin

proc

procs_priv

proxies_priv

server_cost

servers

对象

名

行

数据长度

引擎

columns_priv

0

0 byte

MyISA

db

2

976 字节

MyISA

engine_cost

2

16.00 KB

InnoDB

event

0

0 byte

MyISA

func

0

0 byte

MyISA

general_log

2

0 byte

CSV

gtid_executed

0

16.00 KB

InnoDB

help_category

41

16.00 KB

InnoDB

help_keyword

513

96.00 KB

InnoDB

help_relation

1474

64.00 KB

InnoDB

help_topic

399

1.52 MB

InnoDB

innodb_index_stats

7

16.00 KB

InnoDB

innodb_table_stats

2

16.00 KB

InnoDB

ndb_binlog_index

0

0 byte

MyISA

plugin

0

16.00 KB

InnoDB

proc

48

293.48 KB

MyISA

procs_priv

0

0 byte

MyISA

proxies_priv

1

837 字节

MyISA

server_cost

6

16.00 KB

InnoDB

servers

0

16.00 KB

InnoDB

slave_master_info

0

16.00 KB

InnoDB

slave_relay_log_info

0

16.00 KB

InnoDB

slave_worker_info

0

16.00 KB

InnoDB

Mongodb也是类似的，不过Pycharm默认好像没有支持mongodb的数据库的gui,需要插件！不过用Robo 3T也一样！

2. 操练阶段

这个阶段主要熟悉两大数据的用法，了解常见的命令，可以通过上面的可视化工具进行建标！

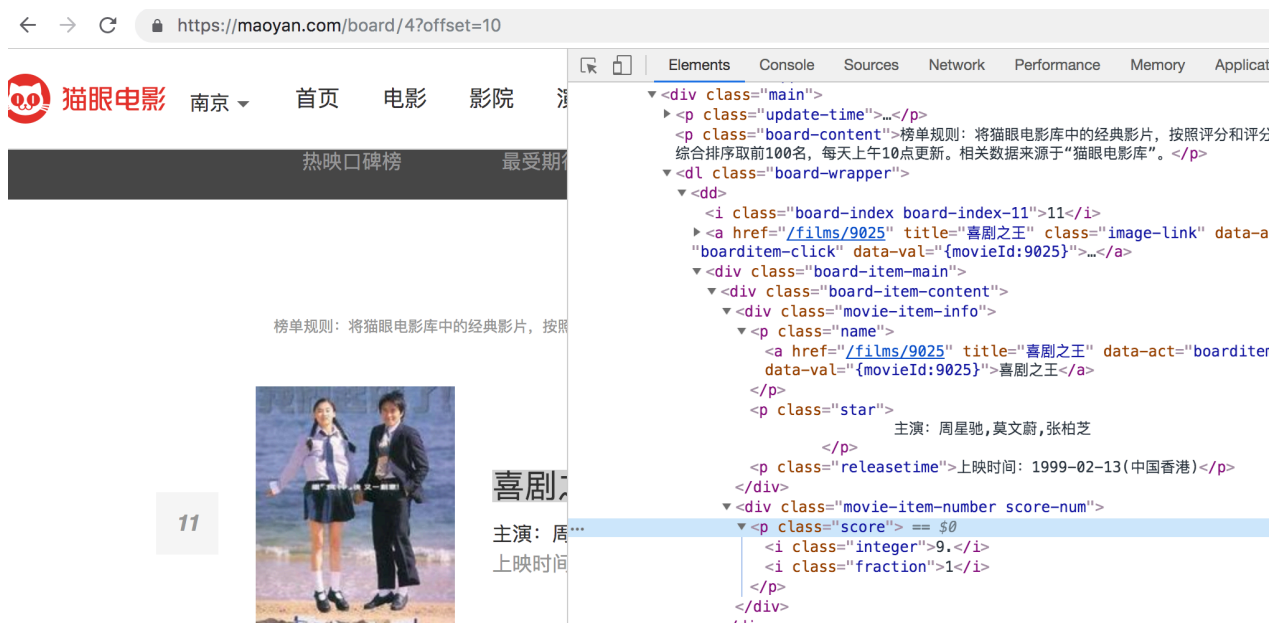
然后还是简单大家先在敲一些命令，加深印象，建议如下：

- 1).手动敲命令，来建表，查询，熟悉mysql和mongo的常见命令
- 2).用Gui工具来操作，同上

3. Python实战

数据库离不开数据，为了照顾很多新手同学，我们这次插入100条数据，数据要有趣，有意思！我想来想去，找了很多数据源的轮子，Github上的微信公众号，豆瓣，链家，知乎等等爬虫的轮子，发现大部分不好用。后来我找了tushare库里面，虽然可以很容易的到每日的票房的排名数据，但是需要周期行的获取！

所以反复对比之后，我们决定用爬虫爬取猫眼的Top100 电影，没有反爬，而且页面非常规整：



The screenshot shows the Maoyan.com website's Top 100 movies page. The URL is <https://maoyan.com/board/4?offset=10>. The page displays a list of movies, with the 11th movie, '喜劇之王' (King of Comedy), highlighted. The movie's details, including the actors (周星驰, 莫文蔚, 张柏芝) and the release time (1999-02-13), are visible. The right side of the image shows the browser's developer tools with the HTML structure of the movie item, including the score and release time.

大概只有10-20行就搞定，非常简单。我多加了些做了封装和完善，如果不懂爬虫的，我把源码贴出：

```
from pyquery import PyQuery
from pprint import pprint
import requests
import time

def decode_html(html_text):
    '''解析网页里面的电影数据'''
    doc=PyQuery(html_text)
    for item in doc.items('.board-wrapper dd'):
        yield {
            'name':item.find('.name').text(),
            'actors':item.find('.star').text(),
            'time':item.find('.releasetime').text(),
            'score':item.find('.score').text()
        }

def download_html(url):
    '''下载网页的html text'''
    try:
        r = requests.get(url)
        if r.status_code==200:
```

```
        return r.text
    else:
        print ('request failed ,status is {}'.format(r.status_code))
        return None
except Exception as e:
    print (e)
    return None

def parse_url(offset):
    '''解析网页'''
    url = 'https://maoyan.com/board/4?offset={}'.format(offset)
    html_text=download_html(url)
    for each_movie in decode_html(html_text):
        pprint(each_movie)

def main():
    '''多页面, 1-10'''
    for offset in range(10,100,10):
        parse_url(offset)
```