【Experiment name】 Developer community data collection

【Purpose】

Master the construction of Python data acquisition environment in Windows environment

Understand the main functions of the developer community Segmentfault , Nuggets, etc.

Master the basics of HTML

Master Python database operation method

Master the Scrapy crawler framework, crawling strategy design, data storage table design

【Experimental content】

Topic : Use Scrapy to implement a developer community crawler, crawl developer community users, resources and other information, and store them in the MySQL database.

Requirements : 1. Set up a development environment, including installing Python interpreter and Scrapy (Anaconda is recommended [1]), Python development IDE (PyCharm is recommended Community version [2]), browser (Chrome is recommended [3]), MySQL database [4], visual database management tool (HediSQL is recommended [5]);

2. Analyze the main functions of Segmentfault [6]or the Nuggets [7]community, and design and store data tables for the information to be crawled;

3. Design the crawling strategy flow chart, give the initial page to be crawled, the URL extraction method to be crawled and other information;

4. Develop Scrapy crawler, crawl relevant data and store it in MySQL database.

Improvement (optional) : Consider the design strategy of parallel crawling by multiple machines, such as using centralized database management, distributed deployment of crawlers, unified acquisition of URLs to be crawled from the database, and unified storage of data.

[1] https://www.anaconda.com/distribution/

[2] https://www.jetbrains.com/pycharm/download/#section=windows

[3] https://www.google.com/intl/zh-CN/chrome/

[4] https://dev.mysql.com/downloads/installer/

[5] https://www.heidisql.com/download.php

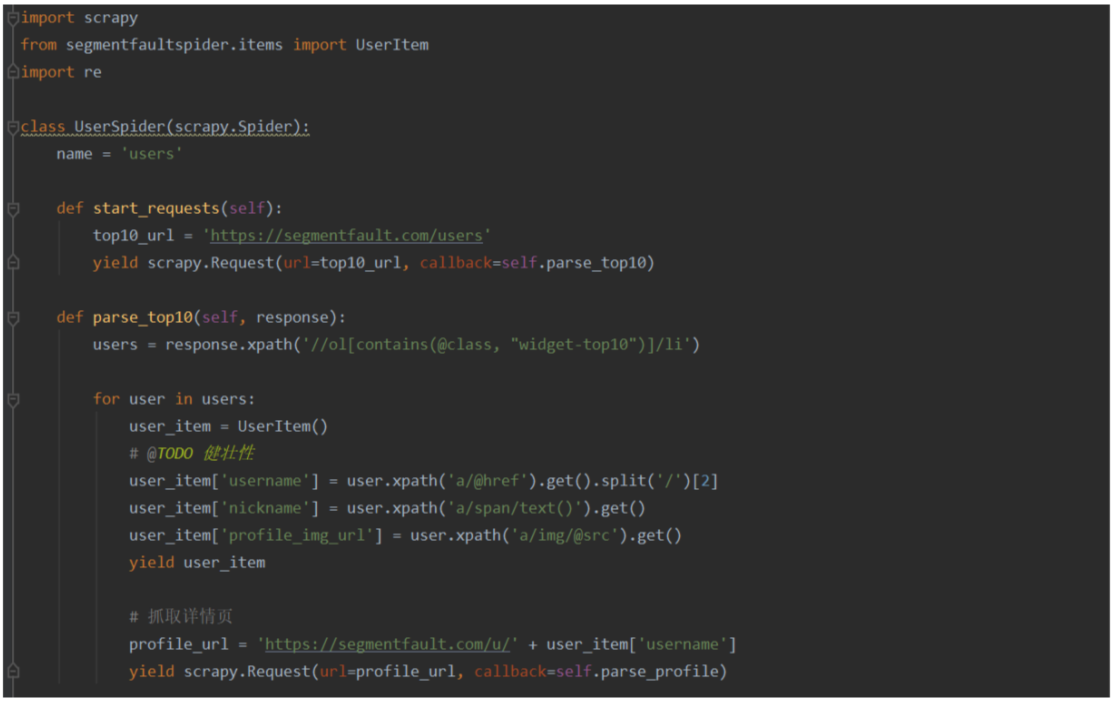
[6] https://segmentfault.com/

[7] https://juejin.im/

Experimental results (experimental steps and related core codes) :

user_spider.py

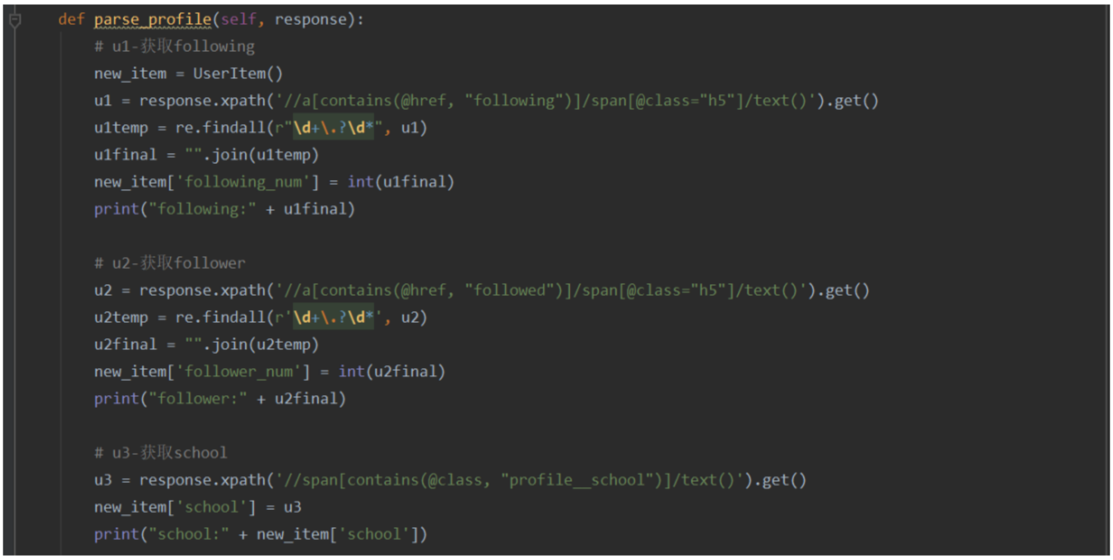
first part:



```python
import scrapy
from segmentfaultspider.items import UserItem
import re

class UserSpider(scrapy.Spider):
    name = 'users'

    def start_requests(self):
        top10_url = 'https://segmentfault.com/users'
        yield scrapy.Request(url=top10_url, callback=self.parse_top10)

    def parse_top10(self, response):
        users = response.xpath('//ol[contains(@class, "widget-top10")]/li')

        for user in users:
            user_item = UserItem()
            # @TODO 健壮性
            user_item['username'] = user.xpath('a/@href').get().split('/')[2]
            user_item['nickname'] = user.xpath('a/span/text()').get()
            user_item['profile_img_url'] = user.xpath('a/img/@src').get()
            yield user_item

            # 抓取详情页
            profile_url = 'https://segmentfault.com/u/' + user_item['username']
            yield scrapy.Request(url=profile_url, callback=self.parse_profile)
```

①Import the required library, scrapy and other libraries are used in this experiment op10 page information to be crawled

the second part:

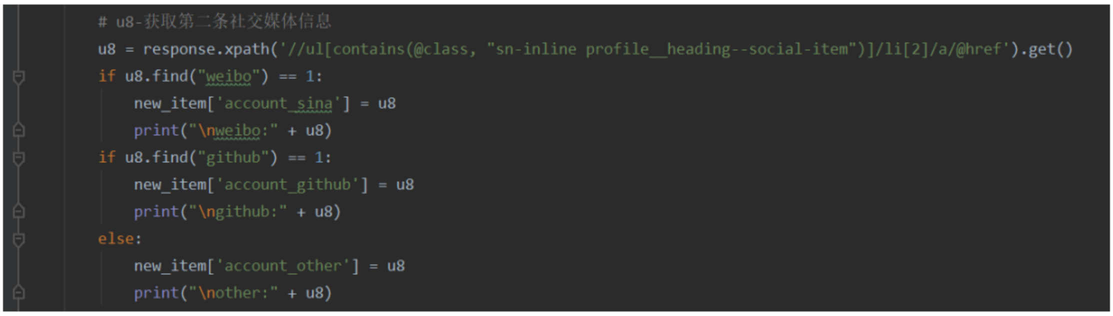


```python
    def parse_profile(self, response):
        # u1-获取following
        new_item = UserItem()
        u1 = response.xpath('//a[contains(@href, "following")]/span[@class="h5"]/text()').get()
        u1temp = re.findall(r"\d+\.?\d*", u1)
        u1final = "".join(u1temp)
        new_item['following_num'] = int(u1final)
        print("following:" + u1final)

        # u2-获取follower
        u2 = response.xpath('//a[contains(@href, "followed")]/span[@class="h5"]/text()').get()
        u2temp = re.findall(r'\d+\.?\d*', u2)
        u2final = "".join(u2temp)
        new_item['follower_num'] = int(u2final)
        print("follower:" + u2final)

        # u3-获取school
        u3 = response.xpath('//span[contains(@class, "profile__school")]/text()').get()
        new_item['school'] = u3
        print("school:" + new_item['school'])
```

①Based on the user information in top10 , we will further explore the information of various users associated with these users. I declare that u 1~u10 and other coming times refer to these information attributes.

②u 1, u2, u3 are respectively used to locate, process and obtain user information such as following, follower, school , etc.

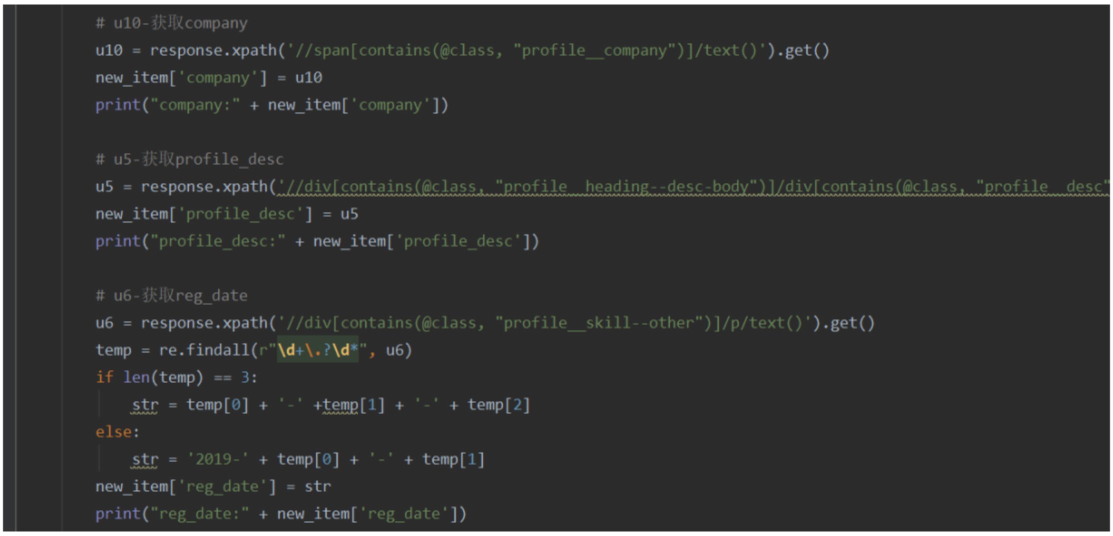
the third part:



```
# u8-获取第二条社交媒体信息
u8 = response.xpath('//ul[contains(@class, "sn-inline profile__heading--social-item")]/li[2]/a/@href').get()
if u8.find("weibo") == 1:
    new_item['account_sina'] = u8
    print("\nweibo:" + u8)
if u8.find("github") == 1:
    new_item['account_github'] = u8
    print("\ngithub:" + u8)
else:
    new_item['account_other'] = u8
    print("\nother:" + u8)
```

Because most users do not have github and weibo accounts at the same time , and some users will also upload twitter or zhihu accounts , but the number of accounts generally does not exceed 3. Based on this, we designed u 4 , u8 , and u9 to judge and store corresponding accounts, among which weibo and github store Weibo and github accounts as usual , and set up a separate "other column" to store other accounts.

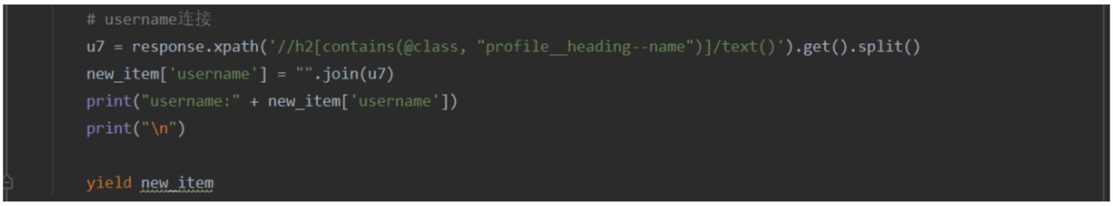
fourth part:



```
# u10-获取company
u10 = response.xpath('//span[contains(@class, "profile__company")]/text()').get()
new_item['company'] = u10
print("company:" + new_item['company'])

# u5-获取profile_desc
u5 = response.xpath('//div[contains(@class, "profile__heading--desc-body")]/div[contains(@class, "profile__desc"
new_item['profile_desc'] = u5
print("profile_desc:" + new_item['profile_desc'])

# u6-获取reg_date
u6 = response.xpath('//div[contains(@class, "profile__skill--other")]/p/text()').get()
temp = re.findall(r"\d+\.?\d*", u6)
if len(temp) == 3:
    str = temp[0] + '-' +temp[1] + '-' + temp[2]
else:
    str = '2019-' + temp[0] + '-' + temp[1]
new_item['reg_date'] = str
print("reg_date:" + new_item['reg_date'])
```

here u 10 , u 5, u6 store basic user information such as c ompany, profile_desc, reg_date as usual. Similar to the second part.

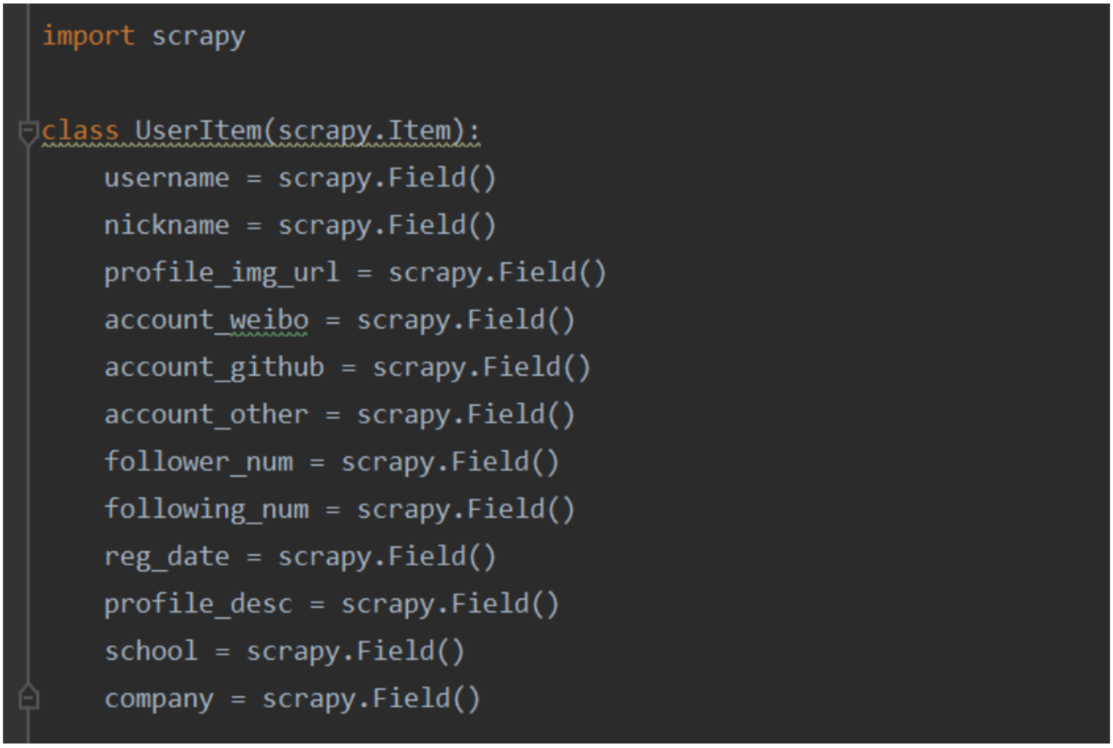
the fifth part:



```
# username连接
u7 = response.xpath('//h2[contains(@class, "profile__heading--name")]/text()').get().split()
new_item['username'] = "".join(u7)
print("username:" + new_item['username'])
print("\n")

yield new_item
```

Finally, the acquired column information is corresponding to the user name , and the final processing is performed on the table content.

items.py

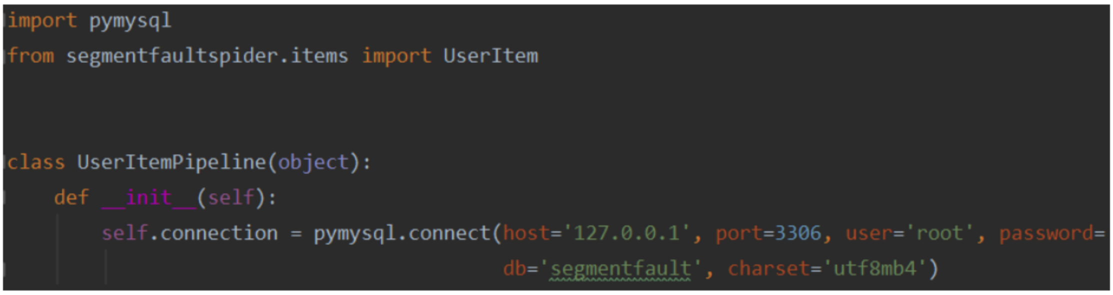


```python
import scrapy

class UserItem(scrapy.Item):
    username = scrapy.Field()
    nickname = scrapy.Field()
    profile_img_url = scrapy.Field()
    account_weibo = scrapy.Field()
    account_github = scrapy.Field()
    account_other = scrapy.Field()
    follower_num = scrapy.Field()
    following_num = scrapy.Field()
    reg_date = scrapy.Field()
    profile_desc = scrapy.Field()
    school = scrapy.Field()
    company = scrapy.Field()
```

① Select the name of the data column that the crawler program will eventually obtain and fill in this experiment. In this experiment, I designed and applied 12 columns .

pipelines.py

first part:

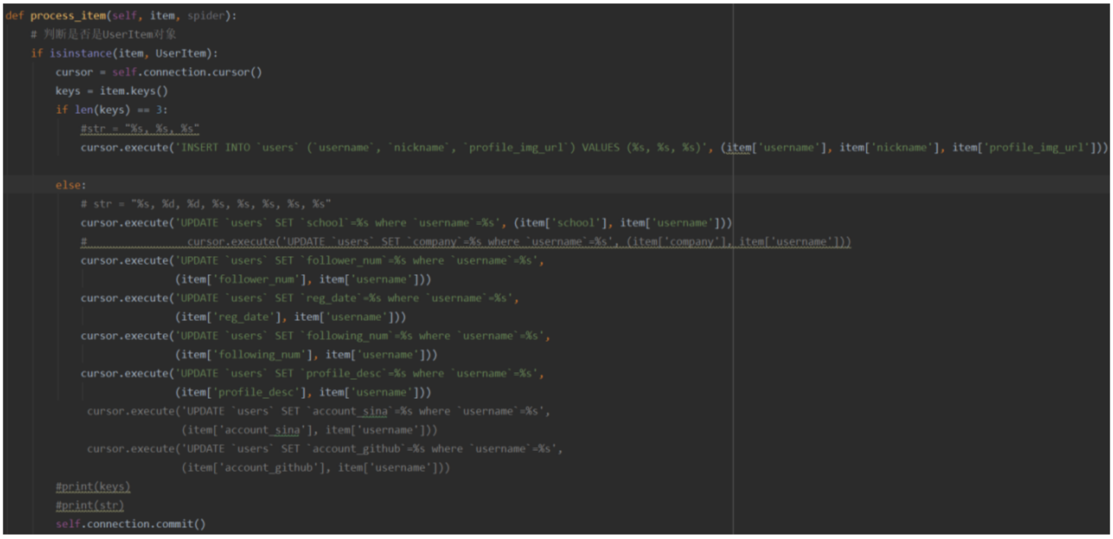


```python
import pymysql
from segmentfaultspider.items import UserItem


class UserItemPipeline(object):
    def __init__(self):
        self.connection = pymysql.connect(host='127.0.0.1', port=3306, user='root', password=
                                          db='segmentfault', charset='utf8mb4')
```

① Here, first configure the relevant library (s park has been automatically created at the beginning of creation)

② Configure our mysql information , such as host, port, password, etc.

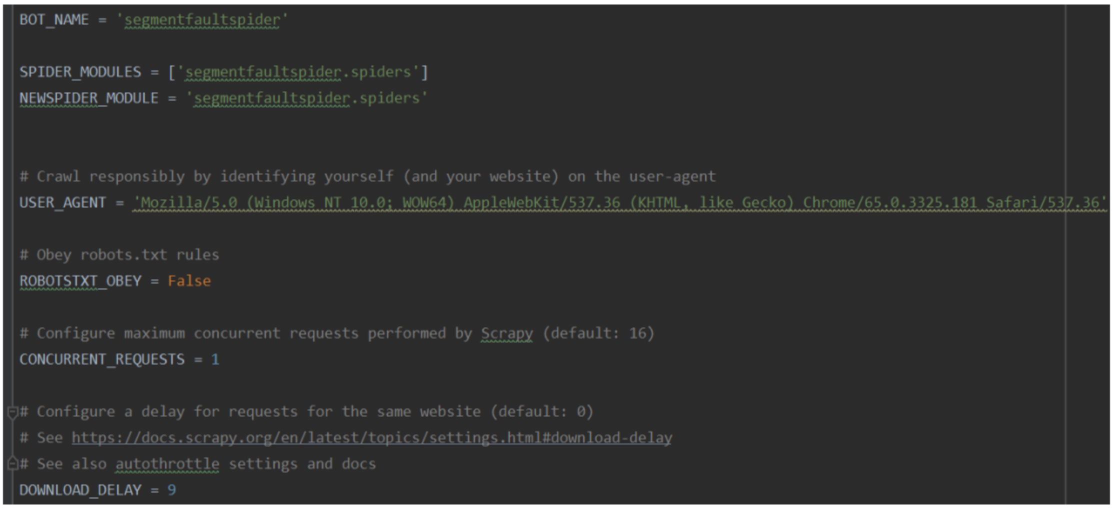
the second part:



```
def process_item(self, item, spider):
    # 判断是否是UserItem对象
    if isinstance(item, UserItem):
        cursor = self.connection.cursor()
        keys = item.keys()
        if len(keys) == 3:
            #str = "%s, %s, %s"
            cursor.execute('INSERT INTO `users` (`username`, `nickname`, `profile_img_url`) VALUES (%s, %s, %s)', (item['username'], item['nickname'], item['profile_img_url']))

        else:
            # str = "%s, %d, %d, %s, %s, %s, %s, %s"
            cursor.execute('UPDATE `users` SET `school`=%s where `username`=%s', (item['school'], item['username']))
            #                cursor.execute('UPDATE `users` SET `company`=%s where `username`=%s', (item['company'], item['username']))
            cursor.execute('UPDATE `users` SET `follower_num`=%s where `username`=%s',
                           (item['follower_num'], item['username']))
            cursor.execute('UPDATE `users` SET `reg_date`=%s where `username`=%s',
                           (item['reg_date'], item['username']))
            cursor.execute('UPDATE `users` SET `following_num`=%s where `username`=%s',
                           (item['following_num'], item['username']))
            cursor.execute('UPDATE `users` SET `profile_desc`=%s where `username`=%s',
                           (item['profile_desc'], item['username']))
             cursor.execute('UPDATE `users` SET `account_sina`=%s where `username`=%s',
                            (item['account_sina'], item['username']))
             cursor.execute('UPDATE `users` SET `account_github`=%s where `username`=%s',
                            (item['account_github'], item['username']))
        #print(keys)
        #print(str)
        self.connection.commit()
```

Here it is mainly to import the user information crawled from user_spider into the corresponding position of mysql . For this experiment, I will import it into my user table . By setting the corresponding information for each column, the Field ( ) data corresponding to each column in item.py can be accurately stored in the corresponding columns of the user table of mysql .

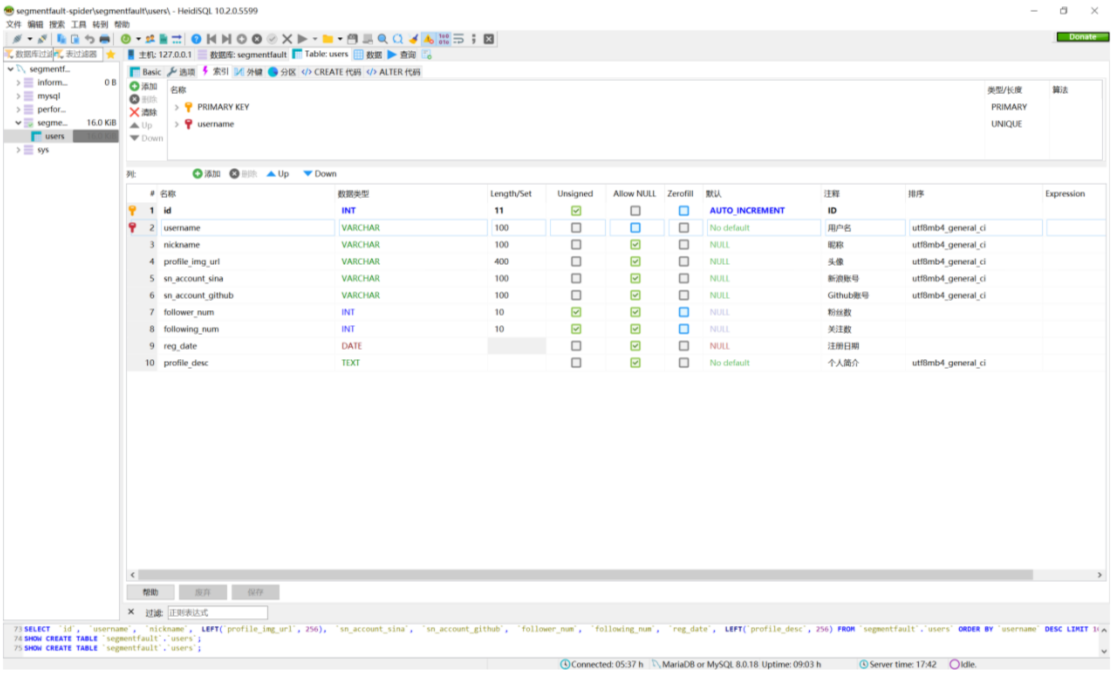
settings.py



```
BOT_NAME = 'segmentfaultspider'

SPIDER_MODULES = ['segmentfaultspider.spiders']
NEWSPIDER_MODULE = 'segmentfaultspider.spiders'


# Crawl responsibly by identifying yourself (and your website) on the user-agent
USER_AGENT = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36'

# Obey robots.txt rules
ROBOTSTXT_OBEY = False

# Configure maximum concurrent requests performed by Scrapy (default: 16)
CONCURRENT_REQUESTS = 1

# Configure a delay for requests for the same website (default: 0)
# See https://docs.scrapy.org/en/latest/topics/settings.html#download-delay
# See also autothrottle settings and docs
DOWNLOAD_DELAY = 9
```

Set some parameters and environment here, such as the part reminded by the teacher in the courseware, but also pay attention to the "# Crawl..." part in the figure to be set for the crawler carrier of the browser, so that the crawler program can run smoothly .

Users table design:



名称	类型/长度	算法
PRIMARY KEY	PRIMARY	
username	UNIQUE	

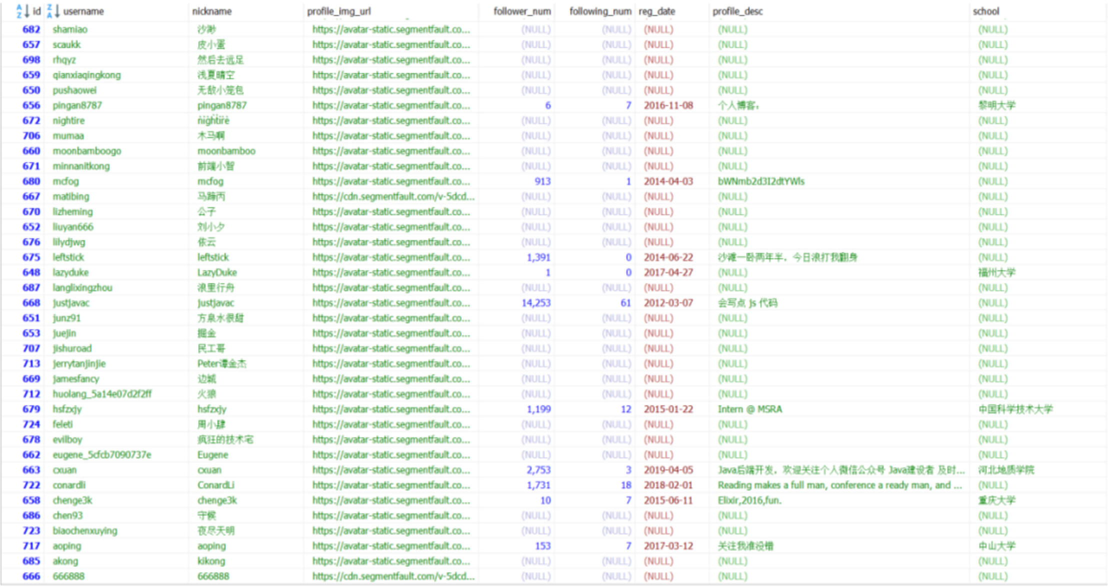
列:

#	名称	数据类型	Length/Set	Unsigned	Allow NULL	Zerofill	默认	注释	排序	Expression
1	id	INT	11	☑	☐	☐	AUTO_INCREMENT	ID		
2	username	VARCHAR	100	☐	☐	☐	No default	用户名	utf8mb4_general_ci	
3	nickname	VARCHAR	100	☐	☑	☐	NULL	昵称	utf8mb4_general_ci	
4	profile_img_url	VARCHAR	400	☐	☑	☐	NULL	头像	utf8mb4_general_ci	
5	sn_account_sina	VARCHAR	100	☐	☑	☐	NULL	新浪账号	utf8mb4_general_ci	
6	sn_account_github	VARCHAR	100	☐	☑	☐	NULL	Github账号	utf8mb4_general_ci	
7	follower_num	INT	10	☑	☑	☐	NULL	粉丝数		
8	following_num	INT	10	☑	☑	☐	NULL	关注数		
9	reg_date	DATE		☐	☑	☐	NULL	注册日期		
10	profile_desc	TEXT		☐	☑	☐	No default	个人简介	utf8mb4_general_ci	

```
73 SELECT `id`, `username`, `nickname`, LEFT(`profile_img_url`, 256), `sn_account_sina`, `sn_account_github`, `follower_num`, `following_num`, `reg_date`, LEFT(`profile_desc`, 256) FROM `segmentfault`.`users` ORDER BY `username` DESC LIMIT 1
74 SHOW CREATE TABLE `segmentfault`.`users`;
75 SHOW CREATE TABLE `segmentfault`.`users`;
```

Connected: 05:37 h   MariaDB or MySQL 8.0.18 Uptime: 09:03 h   Server time: 17:42   Idle.

User table data:



id	username	nickname	profile_img_url	follower_num	following_num	reg_date	profile_desc	school
682	shamiao	沙渺	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
657	scaukk	皮小蛋	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
698	rhqyz	然后去远足	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
659	qianxiaqingkong	浅夏晴空	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
650	pushaowei	无敌小笼包	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
656	pingan8787	pingan8787	https://avatar-static.segmentfault.co...	6	7	2016-11-08	个人博客。	黎明大学
672	nightire	nightire	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
706	mumaa	木马啊	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
660	moonbamboogo	moonbamboo	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
671	minnanitkong	前端小智	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
680	mcfog	mcfog	https://avatar-static.segmentfault.co...	913	1	2014-04-03	bWNmb2d3I2dtYWls	(NULL)
667	matibing	马蹄丙	https://cdn.segmentfault.com/v-5dcd...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
670	lizheming	公子	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
652	liuyan666	刘小夕	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
676	lilydjwg	依云	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
675	leftstick	leftstick	https://avatar-static.segmentfault.co...	1,391	0	2014-06-22	沙滩一卧两年半，今日浪打我翻身	(NULL)
648	lazyduke	LazyDuke	https://avatar-static.segmentfault.co...	1	0	2017-04-27	(NULL)	福州大学
687	langlixingzhou	浪里行舟	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
668	justjavac	justjavac	https://avatar-static.segmentfault.co...	14,253	61	2012-03-07	会写点 js 代码	(NULL)
651	junz91	方泉水很甜	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
653	juejin	掘金	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
707	jishuroad	民工哥	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
713	jerrytanjinjie	Peter谭金杰	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
669	jamesfancy	边城	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
712	huolang_5a14e07d2f2ff	火狼	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
679	hsfzxjy	hsfzxjy	https://avatar-static.segmentfault.co...	1,199	12	2015-01-22	Intern @ MSRA	中国科学技术大学
724	feleti	周小肆	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
678	evilboy	疯狂的技术宅	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
662	eugene_5cfcb7090737e	Eugene	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
663	cxuan	cxuan	https://avatar-static.segmentfault.co...	2,753	3	2019-04-05	Java后端开发，欢迎关注个人微信公众号 Java建设者 及时...	河北地质学院
722	conardli	ConardLi	https://avatar-static.segmentfault.co...	1,731	18	2018-02-01	Reading makes a full man, conference a ready man, and ...	(NULL)
658	chenge3k	chenge3k	https://avatar-static.segmentfault.co...	10	7	2015-06-11	Elixir,2016,fun.	重庆大学
686	chen93	守候	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
723	biaochenxuying	夜尽天明	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
717	aoping	aoping	https://avatar-static.segmentfault.co...	153	7	2017-03-12	关注我准没错	中山大学
685	akong	kikong	https://avatar-static.segmentfault.co...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)
666	666888	666888	https://cdn.segmentfault.com/v-5dcd...	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

【Experiment Summary】

In this lesson, I learned how to use Scrapy to implement a developer community crawler, crawl developer community users, resources and other information, and store this information in a MySQL database. This is the first crawler program I have completed. I am deeply impressed by the concise crawler commands and good crawler effects under the scrapy framework. Through this experimental class, I have deepened my understanding of crawler methods and the Scrapy framework .