

1. 爬取安居客的信息，进行 redis 存储，存了之后对数据至少做 8 个 redis 操作

2. 代码：

```
from redis import StrictRedis
import pymysql
import requests
import re
from bs4 import BeautifulSoup

#爬取数据
url='https://chengdu.anjuke.com/sale/jinjiang/'

headers={
    'Host':'chengdu.anjuke.com',
    'Refer':'https://chengdu.anjuke.com/sale/jinjiang/',
    'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:66.0) Gecko/20100101 Firefox/66.0',
    'X-Requested-With':'XMLHttpRequest',
}
# 爬取锦江区二手房的名称、价格、大小、建造年份、联系人、地址、标签。

html=requests.get(url,headers=headers).content.decode('utf-8')

soup = BeautifulSoup(html,'html.parser')

items = soup.find_all(class_='list-item')

redis = StrictRedis(host='localhost',port=6379,db=1,password='')

for i in range(len(items)):
    detail = items[i].find(class_='house-details')
    pri = items[i].find(class_='pro-price')
    price = pri.find('strong').text

    Title = detail.select('a')[0].string.strip()

    Price = pri.find('strong').text + '万'

    Area = detail.select('span')[1].text

    Year = detail.select('span')[3].text

    people = detail.select('span')[4].text
```

```

Name = people[1:]

adds = detail.select('span')[5].text.strip()
Addre = re.sub('\n| ', "", adds)

tags = detail.find_all(class_='item-tags')
Tag = ""
for j in range(len(tags)):
    Tag += tags[j].string + ' '

#存入 redis 数据库中
redis.rpush('item'+str(i+1), 'Title:'+Title, 'Price:'+Price, 'Area:'+Area, 'Year:'+Year, 'Name:'+Name, 'Addre:'+Addre, 'Tag:'+Tag)

print("*****存入成功，下面进行各种操作*****")
print("1.获取随机的一个键")
print(redis.randomkey())
print("*****")
print("2.返回键为 item1 的列表的长度")
print(redis.llen('item1'))
print("*****")
print("3.返回键为 item4 的列表中 1 至 3 之间的元素")
print(redis.lrange('item4',1,3))
print("*****")
print("4.返回键为 item3 的列表中 2 位置的元素")
print(redis.lindex('item3',2))
print("*****")
print("5.返回并删除键为 item2 的列表中的尾元素")
print(redis.rpop('item2'))
print("*****")
print("6.返回并删除键为 item5 的列表中的首元素")
print(redis.lpop('item5'))
print("*****")
print("7.判断一个键是否存在")
print(redis.exists('item6'))
print("*****")
print("8.判断键类型")
print(redis.type('item7'))
print("*****")
print("9.获取当前数据库中键的数目")
print(redis.dbsize())

```

3. 截图:

```

***** 存入成功, 下面进行各种操作 *****
1. 获取随机的一个键
b"item4"
*****

2. 返回键为item1的列表的长度
7
*****

3. 返回键为item1的列表中1至3不同的元素
[b"Pr1c1e44.3", b"x1e2xb8", b"x87", b"Ar2a:37m\xc2\xb2", b"Year:2017", b"x5e1xb9", b"x41xe5", b"xb1xb8", b"xe9", b"x80", b"xae"]
*****

4. 返回键为item3的列表中2位置的元素
b"Area:49m\xc2\xb2"
*****

5. 返回并删除键为item2的列表中的尾元素
b"Tag:1xe5", b"xae", b"x2", b"xe5", b"x8e", b"x85", b"x6f", b"x9c", b"x9d", b"x5", b"x8d", b"x97", b"x41", b"xb1", b"x8d", b"x67", b"xb1", b"xae", b"xe5", b"xae", b"x89", b"x99", b"x9d", b"x99", b"x6", b"xb8", b"xe1", b"x5", b"x8c", b"x85", b"x5", b"x85", b"x5", b"x41", b"x8d", b"x8f"

6. 返回并删除键为item5的列表中的首元素
b"Title:1xe1", b"xb1", b"x84", b"x9e", b"x7e", b"x5b1", b"xb3", b"x5", b"x5b1", b"x82", b"x9e", b"xb9", b"x98", b"xb1", b"xb1", b"x8c", b"IF5", b"x5", b"x81", b"x82", b"x41", b"x8", b"xad", b"x5", b"xbf", b"x83", b"x6f", b"x98", b"x5", b"x67", b"x6f", b"x99", b"x68", b"xb7", b"xaf", b"xb1", b"x89", b"1", b"xe1", b"xb1", b"x8c", b"x6f", b"x9e", b"x6f", b"x41", b"xb1", b"x98", b"x30"
*****

7. 判断一个键是否存在
1
*****

8. 判断键类型
b"11a1c"
*****

9. 获取当前数据库中键的数目
60
*****

**Repl Closed**

```

The screenshot shows the DBeaver interface. On the left, the 'Database Explorer' pane displays a tree structure with a folder 'db1 (60/60)' expanded, showing a list of tables from 'item1' to 'item27'. The table 'item1' is selected and highlighted in blue. On the right, the 'Table Viewer' pane displays the data for the selected table 'item1'. It shows a table with two columns: 'row' and 'value'. The data rows are as follows:

row	value
1	Title: 套二带3阳台, ...
2	Price: 150万
3	Area: 86m²
4	Year: 2009年建造
5	Name: 钟婷
6	Address: 瑞升莲花逸都...
7	Tag: 近地铁 近学校

Below the table, the 'Value' field shows 'size in bytes: 0'.