

nsd1912-py02-day04

复杂列表的排序

```
>>> result = {'172.40.58.150': 10, '172.40.58.124': 6, '172.40.58.101': 10, '127.0.0.1': 121,
'192.168.4.254': 103, '192.168.2.254': 110, '201.1.1.254': 173, '201.1.2.254': 119, '172.40.0.54':
391, '172.40.50.116': 244}
# 将字典转换成列表
>>> l1 = list(result.items())
>>> l1
[('172.40.58.150', 10), ('172.40.58.124', 6), ('172.40.58.101', 10), ('127.0.0.1', 121),
('192.168.4.254', 103), ('192.168.2.254', 110), ('201.1.1.254', 173), ('201.1.2.254', 119),
('172.40.0.54', 391), ('172.40.50.116', 244)]
# 列表的sort方法，支持一个名为key的参数。这个key接收一个函数，该函数作用于列表的每一项，将每
一项处理的结果作为排序依据
>>> def func1(seq):
...     return seq[-1]
...
>>> l1.sort(key=func1)
>>> l1
[('172.40.58.124', 6), ('172.40.58.101', 10), ('172.40.58.150', 10), ('192.168.4.254', 103),
('192.168.2.254', 110), ('201.1.2.254', 119), ('127.0.0.1', 121), ('201.1.1.254', 173),
('172.40.50.116', 244), ('172.40.0.54', 391)]
>>> l1.sort(key=lambda seq: seq[-1], reverse=True)
>>> l1
[('172.40.0.54', 391), ('172.40.50.116', 244), ('201.1.1.254', 173), ('127.0.0.1', 121),
('201.1.2.254', 119), ('192.168.2.254', 110), ('192.168.4.254', 103), ('172.40.58.101', 10),
('172.40.58.150', 10), ('172.40.58.124', 6)]
```

安装python软件包

- 官方网站：<https://pypi.org/>
- 安装python包使用pip，它类似于yum，yum是安装rpm包的，pip是安装python包的
- 默认在线安装使用的是国外站点，慢。可以采用国内镜像站点加速

```
[root@localhost ~]# mkdir ~/.pip
[root@localhost ~]# vim ~/.pip/pip.conf
[global]
index-url = http://mirrors.aliyun.com/pypi/simple/

[install]
trusted-host = mirrors.aliyun.com
```

- 在线安装pymysql

```
[root@localhost ~]# pip3 install pymysql
```

- 下载软件包后，本地安装

```
[root@localhost ~]# pip3 install zzg_pypkgs/pymysql_pkgs/*
```

安装pip

```
[root@localhost ~]# wget
https://files.pythonhosted.org/packages/08/25/f204a6138dade2f6757b4ae99bc3994aac28a56
02c97ddb2a35e0e22fbc4/pip-20.1.1.tar.gz
[root@localhost ~]# tar xf pip-20.1.1.tar.gz
[root@localhost ~]# cd pip-20.1.1/
[root@localhost pip-20.1.1]# python3 setup.py install
```

准备数据库

- 安装mariadb-server

```
[root@localhost ~]# yum install -y mariadb-server
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# systemctl enable mariadb
[root@localhost ~]# mysqladmin password tedu.cn
[root@localhost ~]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE nsd1912 DEFAULT CHARSET utf8;
```

- 假设你正在为一个小公司编写程序，需要用到的数据库要记录员工的个人信息以及发工资的情况。
 - 需要记录的字段有：员工ID、姓名、性别、出生日期、联系方式、部门、工资日、基本工资、奖金、实发工资
 - 关系型数据库应该避免数据冗余。数据冗余导致存储空间的额外占用，以及数据不一致。
 - 关系型数据库，需要按一定的规则把数据分别存放不同的表中
- 数据库范式。共有6全范式，至少要满足第三范式的要求
 - 所谓第一范式（1NF）是指在关系模型中，对于添加的一个规范要求，所有的域都应该是原子性的，即数据库表的每一列都是不可分割的原子数据项。根据1NF，联系方式需要继续拆分，如电话号码、email等
 - 第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。第二范式（2NF）要求数据库表中的每个实例或记录必须可以被唯一地区分。简单来说，就是每张表都要有一个主键
 - 第三范式（3NF）是第二范式（2NF）的一个子集，即满足第三范式（3NF）必须满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个关系中不包含已在其它关系已包含的非主关键字信息。简单来说，就是非主字段不能依赖于其他非主字段。实发工资是通过基本工资和奖金计算出来的，不应该出现在表中，需要的时候临时计算。
- 最终确认需要的三张表是：
 - 员工表：员工ID、姓名、性别、出生日期、email、电话、部门ID
 - 部门表：部门ID、部门名
 - 工资表：id、员工ID、工资日、基本工资、奖金

pymysql应用

sqlalchemy应用

https://yiyibooks.cn/wizard/sqlalchemy_11/index.html

- 可以操作oracle / mysql / sql server / db2等各种关系型数据库
- 无须编写sql语句，只要采用python语法即可
- 访问不同数据库时，只需要修改连接引擎，其他代码完全一样
- sqlalchemy采用ORM
- ORM
 - Object：对象，对应python中的class和实例
 - Relationship：关系，关系型数据库
 - Mapper：映射
 - 将数据库中的表与sqlalchemy中的class映射
 - 表中的字段与class的类变量映射
 - 字段的数据类型与sqlalchemy提前定义好的class映射
 - 表的一行记录与class的实例映射

```
[root@localhost ~]# pip3 install sqlalchemy
```

```
MariaDB [nsd1912]> CREATE DATABASE tedu1912 DEFAULT CHARSET utf8;
```