nsd1909-py02-day04

复杂列表的排序

- 列表的sort方法，接受一个名为key的参数
- key需要是一个函数
- key函数处理列表的每一项，将处理结果作为排序依据。
- key函数不改变列表

```
>>> 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}
>>> list(result.items())
[('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)]
>>> l = list(result.items())
>>> def get_last(seq):   # 接受一个序列对象
...   return seq[-1]     # 返回序列中的最后一项
...
>>> l.sort(key=get_last)
>>> l
[('172.40.58.124', 6), ('172.40.58.150', 10), ('172.40.58.101', 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)]

# 因为get_last函数非常简单，所以可以不定义它，直接使用匿名函数
>>> l.sort(key=lambda seq: seq[-1], reverse=True)  # 使作匿名函数，并降序排列
```

模块安装

本地安装

- 下载模块：https://pypi.org/
- 安装方法一：

```
[root@localhost 下载]# tar xf xxxx.tar.gz   # 解压
[root@localhost 下载]# cd xxxx              # 切换至解压目录
# 注意，python setup.py install将会将模块安装到python2中
[root@localhost xxxx]# python3 setup.py install
```

- 安装方法二：

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

在线安装

- pip工具类似于yum，yum安装rpm包，pip安装python包
- 在线直接安装

```
[root@localhost zzg_pypkgs]# pip3 install wget
```

- 如果直接安装的话，有可能速度比较慢，因为使用的是国外站点。
- 设置安装时，使用国内镜像站点

```
[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

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

准备mariadb数据库

```
[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 nsd1909 DEFAULT CHARSET utf8;
```

准备数据库

你正在为一个小公司编写程序，需要记录员工的基本信息和发工资的情况。

需要记录的字段：姓名、性别、出生日期、联系方式、部门、工资日、基本工资、奖金、工资总额

在建表的时候，要注意尽量避免数据冗余。为了减少数据冗余，可以将字段放到不同的表中。

员工表：姓名、性别、出生日期、联系方式、部门ID

部门表：部门ID、部门名称

工资表：姓名、工资日、基本工资、奖金、工资总额

数据库的字段是有要求的，否则不是标准的关系型数据库。指导原则参见数据库范式。

- 所谓第一范式（1NF）是指在关系模型中，对域添加的一个规范要求，所有的域都应该是原子性的。原子性，就是不可分割。根据1NF，联系方式不满足原子性的要求，因为它还可以再分成家庭住址、电话号码、email
- 第二范式（2NF）必须建立在1NF的基础之上，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，每张表都需要有一个主键。根据2NF，需要在每张表中有一个主键。员工表中应该添加员工编号字段作为主键；部门表中，部门ID作为主键；工资表中的姓名为了解决冗余等问题，应该是员工ID，工资表中无论用现有的哪个字段作为主键都不合适，所以强行加入一个名为id的字段作为主键。
- 3NF必须建立在2NF的基础之上，要求非主属性不能信赖于其他非主属性。工资总额是通过基本工资和奖金计算出来的，它不应该出现在数据库中。

根据数据库范式，最终得到的3张表是：

员工表：员工ID、姓名、email、部门ID

部门表：部门ID、部门名称

工资表：id、员工ID、工资日、基本工资、奖金

SQLAlChemy

- 可以连接各种各样的关系型数据库
- 不需要写SQL语句

ORM

- Object：对象
- Relationship：关系
- Mapper：映射
- 数据库中的每张表都映射成一个class
- 表中的字段映射成为class的类变量
- 表的记录映射为class的实例
- 字段的数据类型，映射为sqlalchemy的类

sqlalchemy应用

```
# 本地安装
[root@localhost zzg_pypkgs]# pip3 install sqlalchemy_pkgs/*
# 在线安装
[root@localhost zzg_pypkgs]# pip3 install sqlalchemy

# 创建新的数据库
MariaDB [nsd1909]> CREATE DATABASE tedu1909 DEFAULT CHARSET utf8;
```