

nsd_1908_py02_day04

字典排序

- 字典没有顺序
- 需要将字典转成列表，再排序

```
>>> adict = {'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}
>>> alist = list(adict.items())
>>> alist
[('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方法接受一个名为key的参数。key可以是一个函数，它将列表中的每一项作为参数处理，处理后的结果作为排序依据
>>> def func1(seq):
...     return seq[-1]
...
>>> alist.sort(key=func1)
>>> alist
[('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)]

# 使用匿名函数，并实现降序排列
>>> alist.sort(key=lambda seq: seq[-1], reverse=True)
>>> alist
[('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), ('172.40.58.150', 10), ('172.40.58.101', 10), ('172.40.58.124', 6)]
```

安装python额外的模块

- 官方网站：<https://pypi.org>

```
# 为了更好的下载速度，使用国内镜像站点
(nsd1908) [root@room8pc16 day04]# mkdir ~/.pip/
(nsd1908) [root@room8pc16 day04]# vim ~/.pip/pip.conf
[global]
index-url = http://pypi.douban.com/simple/
[install]
trusted-host=pypi.douban.com

# 在线安装软件包
```

```
(nsd1908) [root@room8pc16 day04]# pip install wget

# 本地安装python软件包
# ls /linux-soft/05/zzg_pypkgs.tar.gz
# 解压缩该软件包
[root@room8pc16 pub]# ls zzg_pypkgs
ansible-cmdb_pkgs  matplotlib_pkgs  requests_pkgs
ansible_pkg        paramiko_pkgs    sqlalchemy_pkgs
dj_pkgs            pymysql_pkgs     wordcloud_pkgs
jenkins            python3_pkg

# 在虚拟环境中，本地安装pymysql
(nsd1908) [root@room8pc16 day04]# pip install /var/ftp/pub/zzg_pypkgs/pymysql_pkgs/*
```

数据库应用

比如，你正在为一个小公司编写数据库，需要记录员工的基本信息和发工资的情况。那么，需要的字段有：姓名、性别、工号、出生日期、联系方式、部门、工资日、基本工资、奖金、实发工资

数据库范式

数据库范式有第一范式、第二范式、第三范式、巴斯-科德范式、第四范式、第五范式六种。

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，即数据库表的每一列都是不可分割的原子数据项。简单来说，每个字段都必须是原子性的，即不可再分。根据1NF的要求，联系方式需要进一步拆分成家庭住址、手机号、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，每张表都需要一个主键字段。

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，3NF要求非主属性，不能信赖其他非主属性。实发工资依赖于基本工资和奖金，基本工资和奖金不是主键，所以实发工资不应该出现在数据库中；当需要该字段时，由程序自动算出。

经过3NF的处理，需要创建的表有：

员工表：工号（主键）、姓名、出生日期、email、部门ID

部门表：部门ID（主键）、部门名称

工资表：id（主键）工资日、工号、基本工资、奖金

```
# 创建数据库
[root@room8pc16 pub]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE nsd1908 DEFAULT CHARSET utf8;
```

sqlalchemy

- 不限定于某一个数据库，支持大多数主流的关系型数据库

- 不必书写SQL语句，只要使用python语法即可
- sqlalchemy使用ORM模型
 - Object : 对象
 - Relationship : 关系
 - Mapper : 映射
 - 将每张表映射为一个class
 - 表中的字段映射为class的类变量，每个字段都使用Column类定义
 - 数据库中的每种数据类型，在sqlalchemy中都有对应的class
 - 表中的每个记录，对应sqlalchemy类的实例

```
(nsd1908) [root@room8pc16 day04]# pip install /var/ftp/pub/zzg_pypkgs/sqlalchemy_pkgs/*
```

```
MariaDB [nsd1908]> CREATE DATABASE tedu1908 DEFAULT CHARSET utf8;
```