疫情数据爬取和存储文档

1、环境准备

Python3.7、MySQL5.7、Access2010、Postman7.20、DBeaver6.3。

2、选择数据来源

目前疫情数据来源大致分为两种，一种是来自官方渠道，例如国家卫健委和各省区卫健委。这种渠道的优点：数据权威性高、更新速度快。缺点：由于卫健委是以新闻的形式发布每日疫情信息，需要从每日新闻的文字信息中提取疫情数据，学习门槛较高。

另一种则是从各大权威媒体渠道间接获得，例如腾讯新闻、丁香园。这种渠道的优点：由于媒体已经对疫情信息进行了正则化处理，访问其接口可以获取直接的疫情数据，学习门槛较低。缺点：数据权威性相较官方渠道稍差，且更新速度受媒体更新策略影响。

3、数据爬取

由于从新闻中提取疫情数据并非本文的主要目标，故在这里本文选择从各大权威媒体渠道获取数据。本文选择腾讯新闻作为数据获取来源，腾讯新闻的疫情数据访问地址为：https://view.inews.qq.com/g2/getOnsInfo?name=disease\_h5。使用Postman访问该链接可得如图1所示的Json字符串：

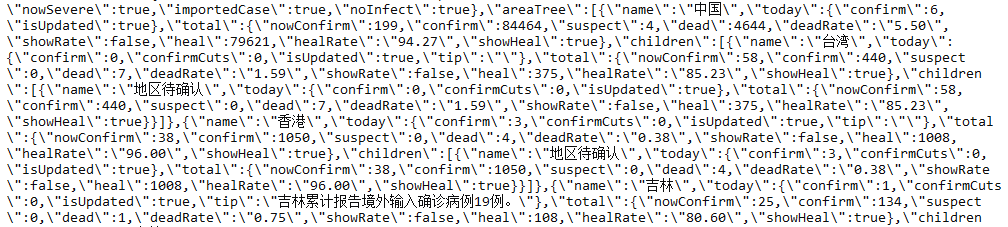


图1 疫情数据Json字符串

4、数据解析

从上述Json字符串图可以很明确地得出每日疫情数据，可以根据字段名来确定各数值所对应的疫情信息。具体来说，从“areaTree”对象中可以找到“name”字段为“中国”的对象，代表着中国的疫情数据。该对象中还包括“children”对象，顾名思义，“children”对象中包括了中国各地区的疫情数据。该对象里还包括“today”对象和“total”对象。其中“today”对象中的数据为今日疫情数据，“confirm”字段表示今日确诊数。而“total”对象则表示累计疫情数据，具体对应关系如表1所示：

表1 Json对象字段含义表

|  |  |
| --- | --- |
| 字段名 | 含义 |
| nowConfirm | 现存确诊数 |
| confirm | 累计确诊数 |
| suspect | 现存疑似数 |
| dead | 累计死亡数 |
| heal | 累计治愈数 |
| deadRate | 死亡率 |
| healRate | 治愈率 |

5、数据存储

爬取到数据后，需要将数据存储下来，存储方案主要可分为以下两种：

1）直接存储到数据库

这种方式会将数据实时存储到数据库中，可以使用编程语言（如Python）中的数据处理接口进行相关操作。这种方式简单直接，在数据量较小的情况下是非常优秀的存储方案，图2和图3分别展示了疫情数据被存储到主流数据库Access和MySQL的情形。（截取了前20条记录）

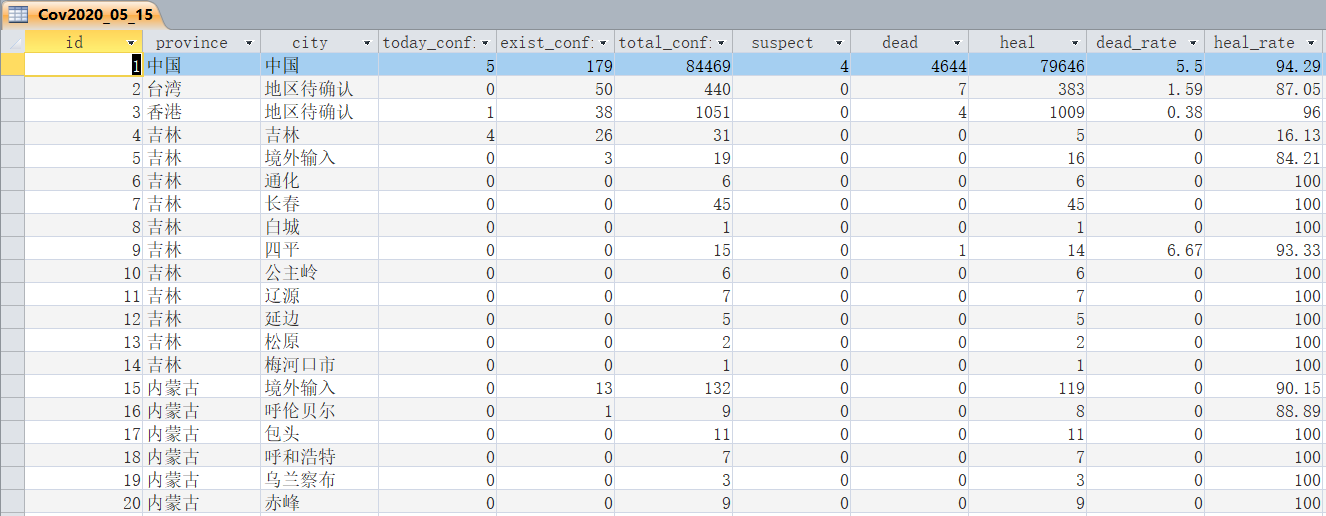


图2 疫情数据图-Access

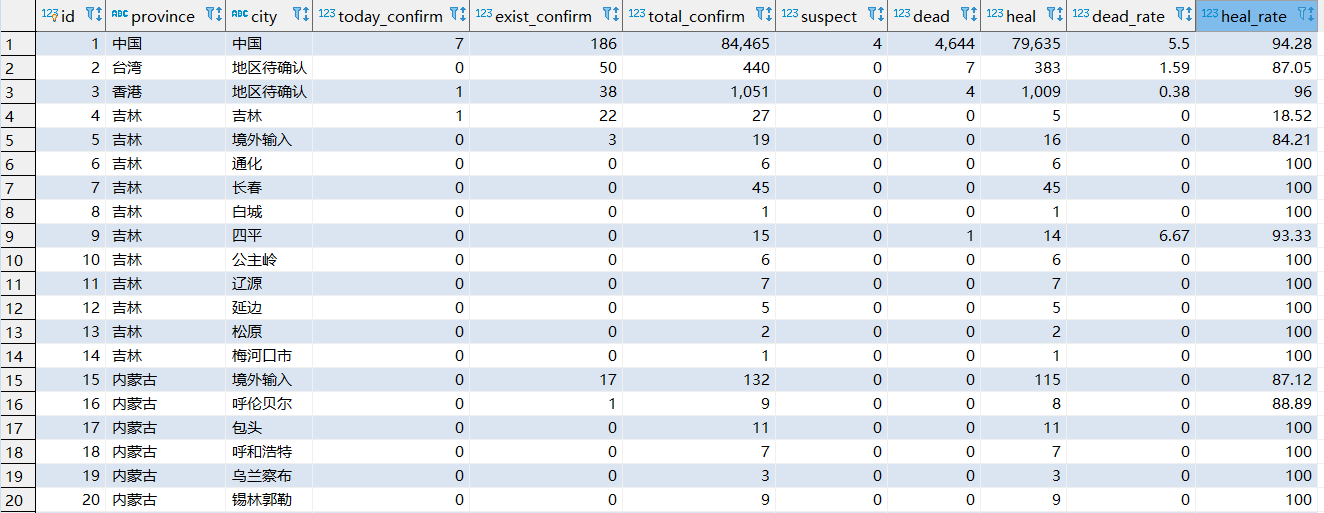


图3 疫情数据图-MySQL

每日疫情数据都会存储到对应的表中，表命名方式为“Cov2020\_MM\_DD”，用以存储2020年MM月DD日的疫情数据（如2020年04月13日的疫情数据存储到表“Cov2020\_04\_13”中）。

从图2和图3可见，表中首条数据记录了“province”列为“中国”的记录，用以表示当天全中国累计数据。数据库中各字段所对应的含义如表2所示。

表2 数据库字段含义表

|  |  |
| --- | --- |
| 字段名 | 含义 |
| province | 省份 |
| city | 城市 |
| today\_confirm | 今日确诊数 |
| exist\_confirm | 现存确诊数 |
| total\_confirm | 累计确诊数 |
| suspect | 现存疑似数 |
| dead | 累计死亡数 |
| heal | 累计治愈数 |
| dead\_rate | 死亡率 |
| heal\_rate | 治愈率 |

2）转存到中间文件

这种方式会先将数据存储到可视化中间文件中（如\*.csv文件），然后再编写转换程序储存到数据库中，如图3。（截取了前20条记录）

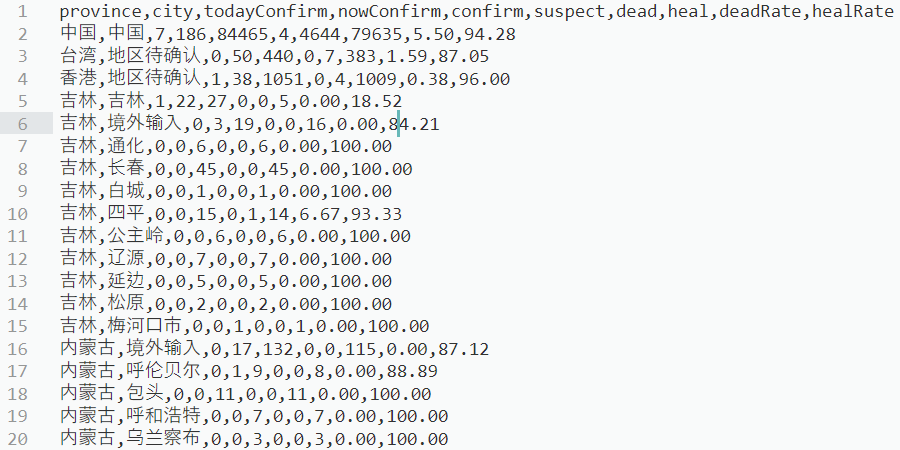


图4 疫情数据图-CSV文件

相比第一种方案，这种方式不需要连接数据库，可以方便地转存到其他数据库中，而且文件格式简单，查看文件更为直观，具有效率高灵活性好的优点。

6、代码展示

代码存储在GitHub，地址为：https://github.com/laputaq/Cov2020。

1）爬取数据



图5 数据爬取操作图

2）数据库连接

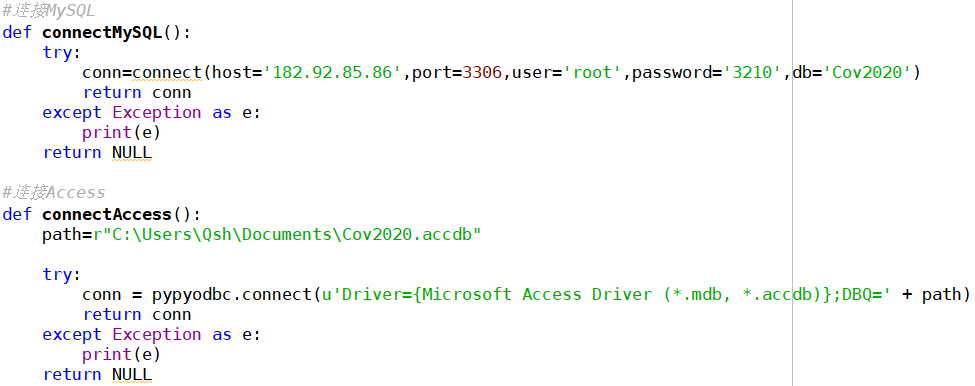


图6 数据库连接图

3）数据库存储

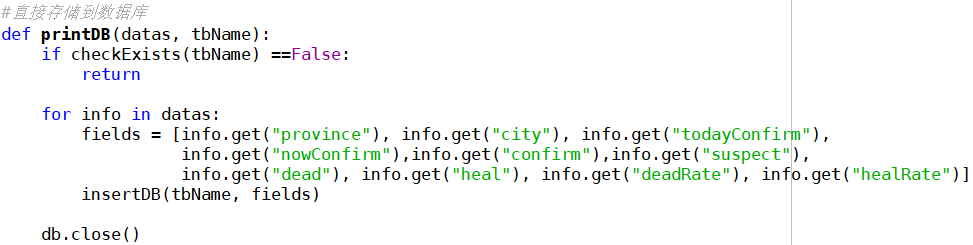


图7 数据库存储操作图

4）表创建操作

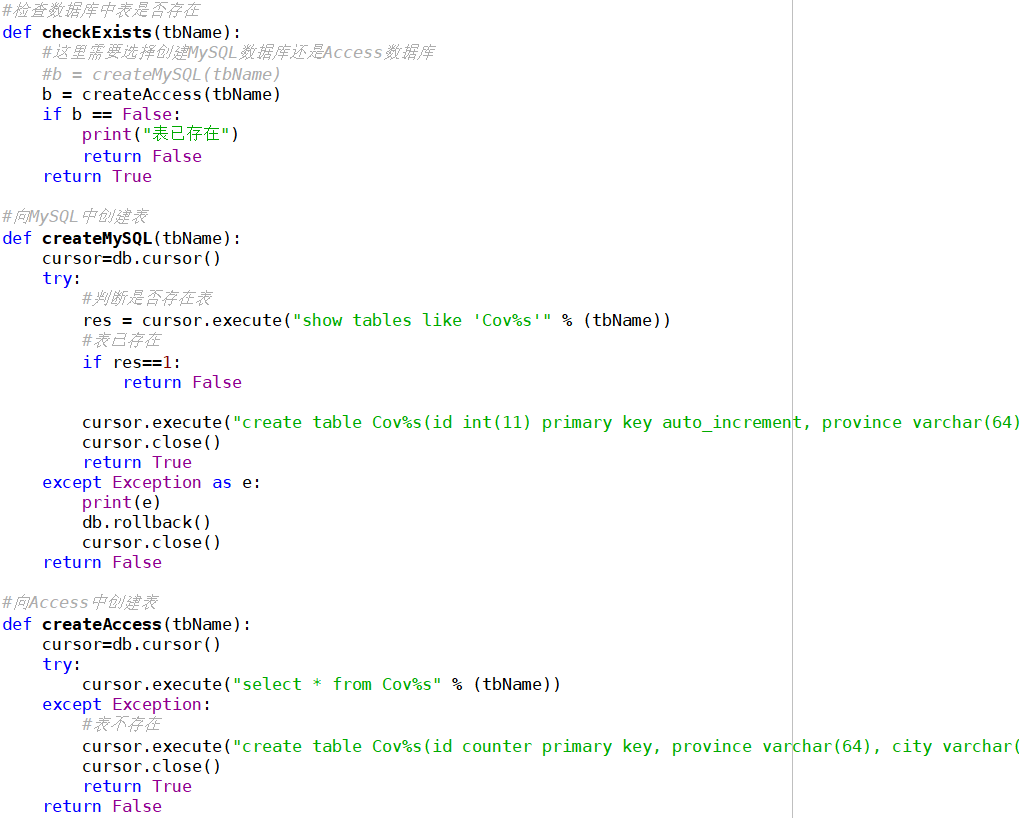


图8 表创建操作图

5）数据插入操作

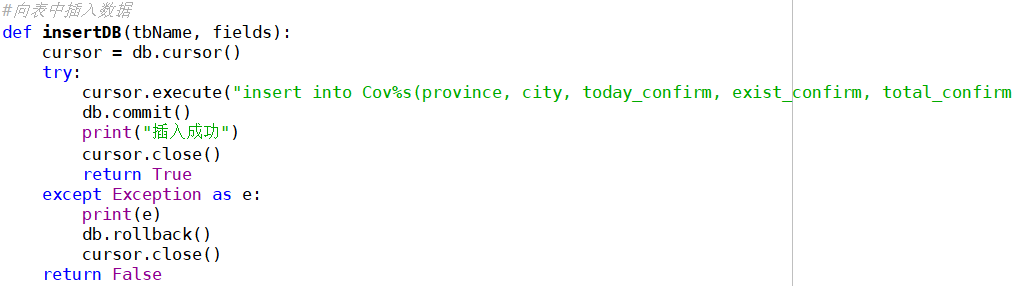


图9 数据插入操作图

6）中间文件存储



图10 中间文件存储操作图