**需求分析及初始设计**

在图谱构建的初期，我们需完成数据库的设计，并将清洗后的数据导入sqlite。在下一步的工作中，为适配知识图谱的构建，我们需要将原有数据库转化为三元组表，从而形成知识图谱。并在此基础上初步进行图谱的图形化展示。

1.数据库设计

Class：省份-分数

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 关系 | 备注 |
| 省份名称 | string | 主键 | 例：’Jiangsu’ |
| 年份 | int | 四位。例：2018 |
| 文理科 | string | ‘art’ or ‘science’ or ‘no’ |
| 一本线 | int |  | 例：420 |
| 二本线 | int |  | 例：400 |
| 专科线 | int |  | 例：300 |

Class：一分一档

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 关系 | 备注 |
| 省份名称 | string | 主键 |  |
| 年份 | int |  |
| 文理科 | string |  |
| 分数 | int |  |
| 排名 | int |  |  |

Class：省份-学校

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 关系 | 备注 |
| 省份名称 | string | 主键 |  |
| 文理科 | string |  |
| 省内可选学校 | Class 学校 |  |  |

需求分析；

省份为中心实体

1）用户可以通过省份来查询排名信息

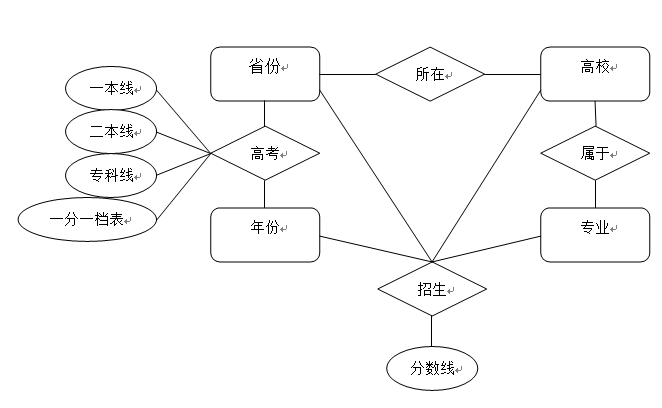
2）用户可以选择某一特定省份内的学校

3）用户可以判断自己的分数能够达到的本科或专科线

4）实体对齐问题，后期可考虑将不同学校的同类专业设置出统一标签

2.知识图谱表示与建模

在RDF中，知识总是以三元组的形式出现，每一份知识可以被分解为如下形式：(subject, predicate, object). 其中，RDF的主语是一个个体，个体是类的实例。RDF中的谓语是一个属性，属性可以连接两个个体，也可以连接一个个体和一个数据类型的实例。因此，RDF中的宾语既可以是一个个体，也可以是一个数据类型的实例。例如，“东南大学在江苏”可写为以下三元组：(SEU, in, Jiangsu). 'SEU' 和 'Jiangsu' 别是高校和省份的实例。故需提取出数据中的实体和关系。具体如下：



3.知识存储

构建完成后，需针对图谱设计底层存储方式，完成项目知识的存储。目前计划采用基于现有数据库（sqlite）的存储方案。需确定实体类别及三元组的谓词种类。

4.运用图形化工具，实现知识图谱可视化

构建知识图谱并存储后，运用图形化工具实现图谱可视化，需展示：

·实体与实体间的联系

·实体与属性间的联系

可视化模块由RDF三元组提取模块和html数据可视化模块组成。整个图谱通过html文件来构建，以结点代表实例，以边代表实例之间的关系。

RDF三元组提取模块由python语言完成，将表中数据以“{type: '', target: '{实体B}', source: '{实体A}', 'rela': '{关系}'},”的格式导出，导入html文件中，完成数据可视化。

Html图形化的设计采用GitHub的开源工具，图的布局为射线型结构。点击某个结点即可使边加重颜色。示例如下：

