目录

[实习总结 2](#_Toc149557504)

[概述： 2](#_Toc149557505)

[任务一主要内容： 2](#_Toc149557506)

[概述： 2](#_Toc149557507)

[任务二主要内容： 4](#_Toc149557508)

[概述： 4](#_Toc149557509)

[任务二代码说明（详情见附件“任务二代码说明”包含代码）： 5](#_Toc149557510)

[任务三主要内容： 16](#_Toc149557511)

[概述： 16](#_Toc149557512)

[相关数据汇总（详情见附件“任务三知识图谱、航空要素等相关数据汇总”）： 16](#_Toc149557513)

[任务总结： 31](#_Toc149557514)

# 实习总结

## 概述：

**项目名称：** 全球导航数据制作平台

**项目时间：** 2023年8月2日-2023年11月1日

**项目职务：** 实习

**所用知识：** python程序设计、SQLite数据库操作、软件工程、QGIS空间数据处理

**所用工具: Navicat for SQLite, PyCharm, QGIS, GeoServer, jupyter notebook, spatialite\_gui, SQLiteStudio, python3, Xmind, Ding Talk, git**

**项目描述：**在这个项目中，我的职责是进行全球区域和地点数据的获取、整理处理和参与功能的全球导航的地图定位开发，协助对全球导航的地图定位进行了实验测试。用于测量发射站同步造成的误差。对于真实位置 国家、省州份、市和区县以及山川河流、海峡等特殊地方的数据处理，这里我采用的基于python操作SQLite(Spatialite)数据库来进行处理。定位的准确性和全面性得到了提高。第二个任务是进行格式的转化mbtiles，能使其再geoserver上呈现，拆分出来了四个功能点，1、旨在实现从文件夹读取图片->实现创建数据库、tiles和metadata表->直接创建（地址）输出.db数据库功能。然后要做整理资料的任务，即将以下数据：知识图谱、航空要素、地理信息、图表样式的数据类型、数据要素和相关图片整理，并形成文档。

**个人体会：**这个项目让我展示了自己与团队成员和顾问保持良好工作关系的能力。我还发现了团队内部的独立动机和动力对项目成功的重要性。期间我们持续每周一次或多次的在会议室等进行项目进度的总结，和项目难度的攻克，摊开重点和难点使得项目计划得到制定和解决。我也是持续三个月参与基于python操作SQLite、Spatilite数据库开发，和数据处理最后解决上诉的种种问题。QGIS和spatialite\_gui使用、geom空间数据的处理。形成全球的国家、省州份、市和区县以及山川河流等特殊地方的geom的点线面（POINT类型、LINESTRING和MULTIPOLYGON等类型）的区域、线和点数据，并能在QGIS中成像。

## 任务一主要内容：

### 概述：

在这个项目中，我的职责是进行全球区域和地点数据的获取、整理处理。对于真实位置 国家、省州份、市和区县以及山川河流、海峡等特殊地方的数据处理，这里我采用的基于python操作SQLite(spatialite)数据库来进行处理。定位的准确性和全面性得到了提高。

其中会遇到许多问题，如

一、海峡的归属，因为海峡点都在海洋区域没有被任何国家的陆地包含，Spatialite的函数ST\_Contains()函数(ST\_Contains()函数是当ST\_Contains(A,B) - 返回true时，当且仅在B中的所有点都在 A中，并且至少有一个B中的点位于A中，不存在上述问题。（速度慢）)失效，且海峡属于多个时常属于国家(如白令海峡属于俄罗斯和美国，马六甲海峡由新加坡、马来西亚和印度尼西亚三国共同管辖)，这里需要以第一个首要国家。这里的解决方式是百度百科结合文心一言，结合人工。补充五十个海峡的的parentUUID和parentName的数据。以及获取到的数据,查看10m\_culturalQGZ, 有国家（ch版）、省级、和美国的区级数据，缺少文档的（Administrative Areas (“adminareas”)）数据,需要对这些数据进行统一格式，和繁简转化（这里使用python的OpenCC库里的函数进行），数据较好，函数contains多数可以直接使用速度较快。个别大区域数据不全,仍需转化为处理点置信度，速度慢。然后外文转中文简体（translate翻译的有些地名错误）。同时需要查看OpenStreetMap 网站，寻找缺失的全球市级区域数据,存在大量区域只有OSM.pbf数据，用网站转为.shp太大被限制。尝试直接使用spatialite数据库的函数。然后使用ST\_Contains()函数和自己编写的置信度函数。慎用或者不用MBR\_Contains()函数，其误差太大。最后，有一些国外省没有点（50+），一些省同名，同国家，但地方不一样（如叶卡布皮尔斯 省等）(原姓名一样，则采用name+type\_en的形式。中文重名了，但是原姓名不一样的，采用英译原姓名的方式)，以及一些省的划分需要纠正，省的区域划分线，与chn视角的版图的国境线不一致（如西撒哈拉的那几个省等59个）(目前需要看中国)，有多余的空闲再检查一些边境点需要检查，特别是中国的乡镇岛屿点，然后数据库区域表worldChinaGeo和地点表worldPoiGeo合并为worldChinaGeo报错,worldGeo.db，合并过程中如有错误，去除geometry\_columns成像后即可合并。

其中保持八个原则，即1. 区域中心点的名字必须与区域数据的名字一致，中心点的数据与区域数据的名字要一致(国外的省级区域name与省级地点name名一致，可用name\_en那一列来对应统一,或者省级空间点在省级空间区域说明是该省则统一name，最后若有重复则去重）。2. 每个父节点下只能存在无法归到更小区域的兴趣点数据， province和直辖市的点归于国家层面，市级别点归在省份，其它归于区域下面。注意特殊情况：一些自治县是市级，且直接治理乡镇级别中间没有区县，且很多都是县级市，直接下辖乡镇街道没有官方的区域划分主要是海南省相关的：地级市：海口市、三亚市、三沙市、儋州市。县级市：五指山市、文昌市、琼海市、万宁市、东方市。县：定安县、屯昌县、澄迈县、临高县。自治县：白沙黎族自治县、昌江黎族自治县、乐东黎族自治县、陵水黎族自治县、保亭黎族苗族自治县、琼中黎族苗族自治县。海南没有区县数据，最低到市、如儋州市不设区县，直辖17个乡级行政区，其它还有广东省东莞市、广东省中山市，以及甘肃省嘉峪关市。3. 删除重复数据，做完后检查一遍数据库的表，去重。4、name列不是简体中文的要处理。如name列的特别字符（墨玉县 قاراقاش）阿合奇县 اقچئي(中国的部分外文和位置语言名和全球山川河流不是中文简体的，都放在nickname字段后，进行百科，或音译)，所有的name列都必须为中文简体。5、数据库区域表和地点表合并为worldChinaGeo,worldGeo.db。6、有地点没有数据的如（雄安新区，两江新区等），保留地点。7、存在一个字的地名，目前保留,name列表存在'?','/'，'(',')','\*'等特殊字符的删除。name中只有'村'的删除这一行。8、添加中心点数据更新至市级，直辖市为区县级。补充worldPoiGeo缺失的广西壮族自治区，甘肃省等中心点。有117个区县区域没有中心点，这里的处理是删除所有区县点，区县不使用中心点。9、将worldGeo中区县的type从district统一为和worldPoiGeo一样的名字county。 最后形成全球的国家、省州份、市和区县以及山川河流等特殊地方的geom的点线面（POINT类型、LINESTRING和MULTIPOLYGON等类型）的区域、线和点数据。

在之前的基础上要实现地图背景的展示，了解了瓦片级数的介绍，即级数是图块集为其提供数据的缩放级别，在获取瓦片级数图片后，需要形成可用的.db或mbtiles文件，其实现逻辑是分为两种情况：一、对于得到的是瓦片级数图片文件夹：先人工：创建.db数据库，及metadata表、tiles 视图（zoom\_level、tile\_column、tile\_row列、uptime）再调用程序: 传入包含级数的文件夹绝对地址（如D:\中导航\Convertmbtiles\GoogleMapNight1-7）按照其默认的方式(这里是以左上角为原点)存入.db数据库的tiles视图表中，然后如果是左上角则转化为2.2的情况进行处理。二、对于已存在的.db数据库：先人工：统一其图片表名为tiles、其字段名和字段顺序，判断是否需要添加metadata表。然后再SQLiteStudio或navicat判断原点是否满足要求，如果不满足，则进行程序：传入该.db数据库的绝对地址（如：D:\中导航\Convertmbtiles\GoogleMapNight1-7.db），再把握以左上角和左下角为原点的区别，即其x坐标一致，以左下角为原点的y坐标等于 (pow(2,level)-1 - 以左上角为原点的y坐标)。上述两种都能形成符合格式要求的.db文件。最后进行.db转化为.mbtiles，这里先手动添加metadata表，然后修改.db文件为后缀mbtiles。该任务持续参与基于python操作SQLite、Spatilite数据库开发，和数据处理最后解决上诉的种种问题。QGIS和spatialite\_gui使用、geom空间数据的处理。形成全球的国家、省州份、市和区县以及山川河流等特殊地方的geom的点线面（POINT类型、LINESTRING和MULTIPOLYGON等类型）的区域、线和点数据，并能在QGIS中成像。

## 任务二主要内容：

### 概述：

第二个任务是进行格式的转化，能使其再geoserver上呈现，拆分出来了四个功能点，1、旨在实现从文件夹读取图片->实现创建数据库、tiles和metadata表->直接创建（地址）输出.db数据库功能。2、旨在实现从左上角.db数据库变为左下角.db数据库功能。3、旨在实现传入一个数据库（其包含了以左下角为原点的表b）,为其创建metadata并将b表变为标准的tiles表的功能。4、前三者都会得到标准的数据库，最后一部修改文件后缀名为.mbtiles。最后在geoserver上发布mbtiles文件进行验证，解决发布时候图片混乱的问题。期间我的实现步骤是1、整理代码思路2、进行准确率测试3、进行性能测试4、各疑惑总结如遇到了geoserver无法配载插件mbtiles的问题，以及当实际情况与base.json不一致时候如何取舍的问题，最后再与团队成员进行约定时间地点的统一讨论交流的处理解决方案是geoserver需要匹配的jdk1.8.0\_301环境，并配置，和对第二个疑惑点metadata要以实际情况为主。结果是形成了包括项目目标、流程逻辑介绍、函数方法介绍、技术介绍、注意事项和mbtile文件介绍的Python代码说明文档模板，并将图片详情附于Xmind。

### 任务二代码说明（详情见附件“任务二代码说明”包含代码）：

【Python代码说明文档】(以下图片详情附于Xmind)

**[文档概述]**  
本文档为Python代码说明文档，记录代码的设计思路、实现细节及注意事项。将涵盖以下主要内容：项目目标、流程逻辑介绍、函数方法介绍、技术介绍。

**[项目目标]**  
1、旨在实现从文件夹读取图片->实现创建数据库、tiles和metadata表->直接创建（地址）输出.db数据库功能。

2、旨在实现从左上角.db数据库变为左下角.db数据库功能。

3、旨在实现传入一个数据库（其包含了以左下角为原点的表b）,为其创建metadata并将b表变为标准的tiles表的功能。

4、前三者都会得到标准的数据库，最后一部修改文件后缀名为.mbtiles。

最后在geoserver上发布mbtiles文件进行验证，解决发布时候图片混乱的问题。

**[环境]**

Miniconda内置模块:

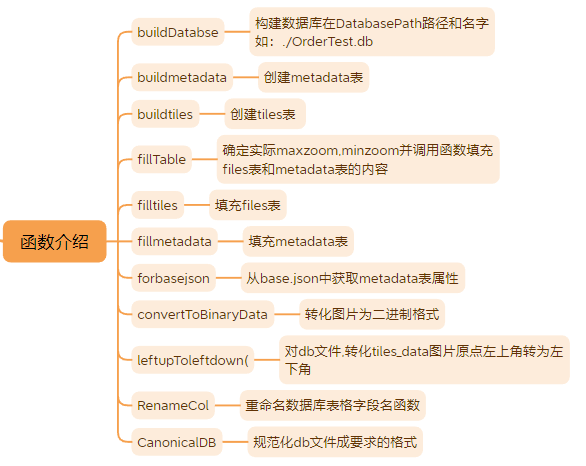
python :3.10.10

sqlite : 3.41.1

jupyter : 1.0.0

PyCharm 2023.1.2

**[函数方法介绍]**



该部分详细介绍函数方法。具体实现技术框架、依赖进行详细说明。同时，要给核心代码示例。

**1、**从文件夹读取图片->实现创建数据库、tiles和metadata表->直接创建（地址）输出.db数据库功能

1.1、依赖关系：该项目依赖于sqlite3、os、json、datatime库，这些是miniconda自带的库,无需额外安装。

依赖于函数方法:

|  |  |  |
| --- | --- | --- |
| name | meaning | 函数的**Parameter**和实现过程 |
| **Buildmetadata** | **创建metadata表** | **Parameter** :(sqliteConnection,cursor)  **实现过程:**调用执行创建语句：sql = '''CREATE TABLE metadata (name text, value text)''' |
| **Buildtiles** | **创建tiles表** | **Parameter** : (sqliteConnection,cursor)  **实现过程:**调用执行创建语句：sql = '''CREATE TABLE tiles (zoom\_level integer, tile\_column integer, tile\_row integer, tile\_data blob,uptime text)''' |
| **convertToBinaryData** | **转化图片为二进制格式** | **Parameter** : (filename)  **实现过程:**找到图片文件路径，用blobData = file.read()来获取该图片，则blobData为二进制格式 |
| **forbasejson** | **从base.json中获取metadata表属性** | **Parameter** : (jsonPath)  **实现过程:**先导入json文件：a = json.loads(content)为字典,然后提取里面的字段：  formatt = a["tileInfo"]['format'];description = a['description']  version = a["currentVersion"];  name = a["mapName"];tile\_width = a["tileInfo"]['rows'];  tile\_height = a["tileInfo"]['cols']; |
| **fillmetadata** | **填充metadata表** | **Parameter** : (sqliteConnection,cursor,maxzoom,minzoom)  **实现过程:** 先调用forbasejson获取到属性值，然后插入属性值。  sql = '''INSERT INTO metadata(name,value)  VALUES ('bounds',?),('format',?),('description',?), ('version',?),  ('type',?),('name',?),('tile\_width',?),('tile\_height',?), ('axis\_origin',?),('axis\_positive\_direction',?),('maxzoom',?),('minzoom',?)''' |
| **filltiles** | **填充tiles表** | **Parameter** : (sqliteConnection,cursor,maxzoom,minzoom)  **实现过程:** 共三次循环，  1、先以i在[minzoom,maxzoom]为索引，以filepath+i遍历文件夹下面的级数文件夹。  2、再分别进入各个级数文件夹，到里面后按照先大数字再小数字方式遍历行数文件夹。  3、最后按照观察原点由左上角变为左下角时，各元素的规律公式即其x坐标一致，即以左下角为原点的y坐标等于 (pow(2,level)-1 - 以左上角为原点的y坐标，原代码是：data\_tuple = (i,k,(orders-1)-j,empPhoto,datetime.datetime.now()) |
| **fillTable** | **确定实际maxzoom,minzoom并调用函数填充files表和metadata表的内容** | **Parameter** : (sqliteConnection,cursor)  **实现过程:** 先将文件夹下面各级数文件夹的名称填入整形列表temp(不能填入说明不是级数文件夹则try越过)，确定了实际的maxzoom=max(temp)和minzoom=min(temp),再调用fillmetadata, filltiles分别填充metadata表, tiles表 |
| **buildDatabse** | **构建数据库在DatabasePath路径和名字如：./OrderTest.db** | **Parameter** : (DatabasePath)  **实现过程:** 先使用函数sqlite3.connect(DatabasePath)创建数据库，再调用buildmetadata, buildtiles创建初始的metadata,files表，形成初始的数据库,最后调用函数fillTable(sqliteConnection,cursor)进行填表内容。 |
| **fileTodb** | **从文件夹->实现创建->直接创建（地址）输出.db数据库(NEED),** **默认以左下角为原点,将瓦片图片文件形成./OrderTest.db文件** | **Parameter** : (filepath,DatabasePath='./OrderTest.db',  bounds='-180.0,-90.0,180.0,90.0',  typee='baselayer',  axis\_origin='-180.0,-90.0',  axis\_positive\_direction='RightUp')  **实现过程:** 调用函数**buildDatabse(DatabasePath)来从文件夹创建数据库。** |

**2、**从左上角.db数据库变为左下角.db数据库功能

2.1、依赖关系：该项目依赖于sqlite3库，这是miniconda自带的库,无需额外安装。

依赖于函数方法:

|  |  |  |
| --- | --- | --- |
| name | meaning | parameter |
| **Buildmetadata** | **创建metadata表** | **Parameter** : (sqliteConnection,cursor)  **实现过程:** 调用执行创建语句：sql = '''CREATE TABLE metadata (name text, value text)''' |
| **Buildtiles** | **创建tiles表** | **Parameter** : (sqliteConnection,cursor)  **实现过程:** 调用执行创建语句：sql = '''CREATE TABLE tiles (zoom\_level integer, tile\_column integer, tile\_row integer, tile\_data blob,uptime text)''' |
| **leftupToleftdown** | **对db文件,转化tiles\_data图片原点左上角转为左下角** | **Parameter** : (dbpath,orderTableName,  zoom\_level='zoom\_level',tile\_column='tile\_column'  ,tile\_row='tile\_row',tile\_data='tile\_data',  bounds='-180.0,-90.0,180.0,90.0',  description='description',version='1.3',  typee='baselayer',  name='Layers',tile\_width=256,tile\_height=256,  axis\_origin='-180.0,-90.0',  axis\_positive\_direction='RightUp')  **实现过程:**  1、先调用buildmetadata，buildtiles构建metadata和tiles表。  2、然后将orderTableName表的内容复制到tiles中便于后面修正。  3、再开始填充metadata表，先从tiles表中得到maxzoom=max(zoom\_level),minzoom=min(zoom\_level),format=tile\_data的数据切片，再结合参数的默认值得以填充。  4、最后orderTable表和tiles表联系，以公式进行tiles表tile\_data值的填入,公式即即以左下角原点的y坐标等于 (pow(2,level)-1 - 以左上角为原点的y坐标。代码:  sql = 'update tiles set tile\_data = (select '+tile\_data+' from '+orderTableName+' where '+\  zoom\_level+'=tiles.zoom\_level and '+tile\_column+'=tiles.tile\_column and '+\  tile\_row+'=(1<<tiles.zoom\_level)-1-tiles.tile\_row)' |

**3、**传入一个数据库（其包含了以左下角为原点的表b）,为其创建metadata并将b表变为标准的tiles表的功能

3.1、依赖关系：该项目依赖于sqlite3、os、numpy、re、datatime库，这些是miniconda自带的库,无需额外安装。

依赖于函数方法:

|  |  |  |
| --- | --- | --- |
| name | meaning | parameter |
| **Buildmetadata** | **创建metadata表** | **Parameter** : (sqliteConnection,cursor)  **实现过程:** 调用执行创建语句：sql = '''CREATE TABLE metadata (name text, value text)''' |
| **RenameCol** | **重命名数据库表格字段名函数** | **Parameter** :(sqliteConnection,cursor,tablename,colname,newcolname)  **实现过程:** 使用sql语句: sql = 'alter table '+tablename+' rename column '+colname+' to '+newcolname修改表的字段名 |
| **CanonicalDB** | **传入数据库地址,变为mbtiles文件规范化db文件成要求的格式** | **Parameter** : (dbpath,orderTableName='tiles',  zoom\_level='zoom\_level',tile\_column='tile\_column'  ,tile\_row='tile\_row',tile\_data='tile\_data',  bounds='-180.0,-90.0,180.0,90.0',  description='description',version='1.3',  typee='baselayer',  name='Layers',tile\_width=256,tile\_height=256,  axis\_origin='-180.0,-90.0',  axis\_positive\_direction='RightUp')  **实现过程:** 1、先修改要规范(**Canonical**)的数据库的目标表名orderTableName为’tiles’。2、规范该表的字段名.3、调用buildmetadata函数创建metadata表。4、从tiles中获取到maxzoom=max(zoom\_level),minzoom=min(zoom\_level)。从tiles表中tile\_data中切片切出format。5、再结合一些默认如bounds等参数，插入metadata的各个值。 |

**4、**前三者都会得到标准的数据库，最后一部修改文件后缀名为.mbtiles。

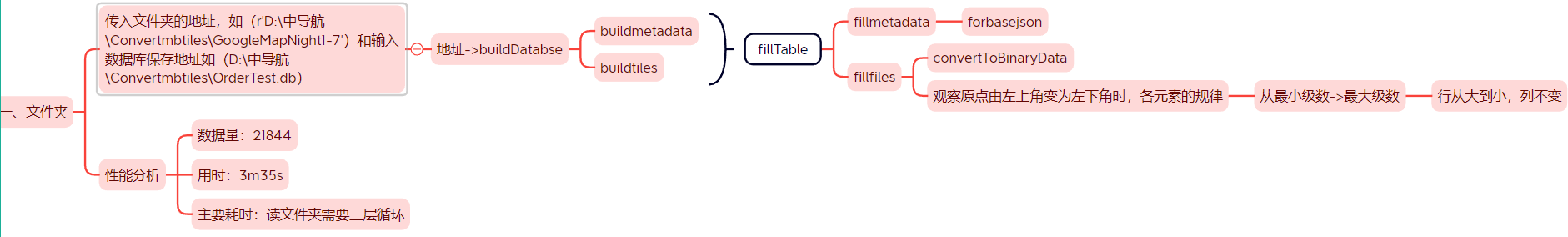
4.1、依赖关系：该项目依赖于os库，这是miniconda自带的库,无需额外安装。

依赖于函数方法:

|  |  |  |
| --- | --- | --- |
| name | meaning | parameter |
| **batch\_rename** | **修改文件的后缀名old\_ext到新的后缀名new\_ext** | **Parameter** : (file\_dir,fname ,old\_ext, new\_ext)  **实现过程:** 先从文件夹地址读入各个文件名，并对比是否为目标文件，如果是则改名，并结束循环 |

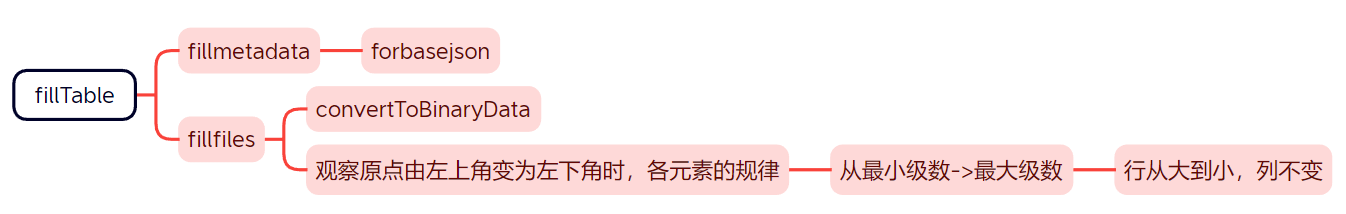
**[代码逻辑介绍]**  
这部分主要介绍项目的代码逻辑、核心代码。同时，针对可能出现的异常情况进行了try越过和except显示错误。

**功能一的逻辑，**先传入级数文件夹地址,用来作数据库的映射和输入数据库保存地址,若没有该数据库，则会自动创建在该目录下面。从文件夹读取图片->实现创建数据库、tiles和metadata表->直接创建（地址）输出.db数据库**：**



**核心代码（tiles表的映射）：**

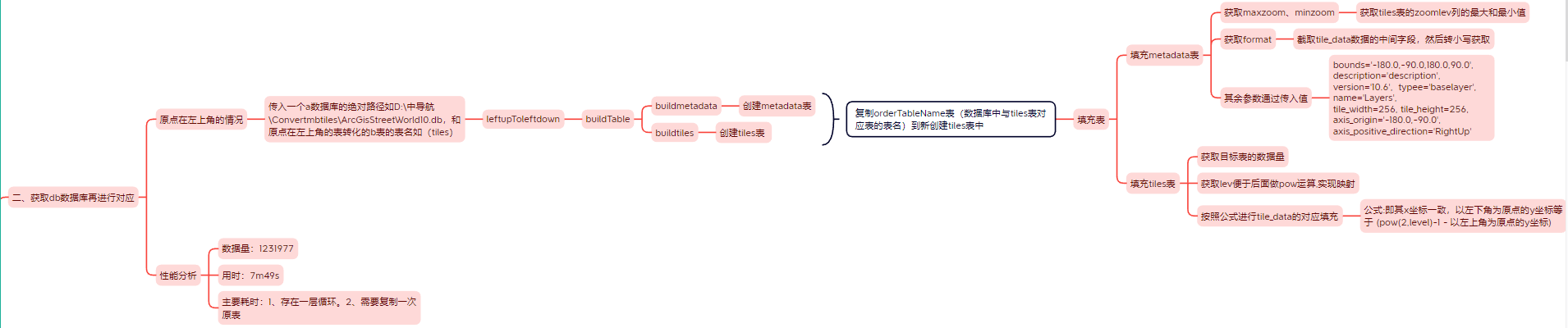
**思路图：**



**代码：**

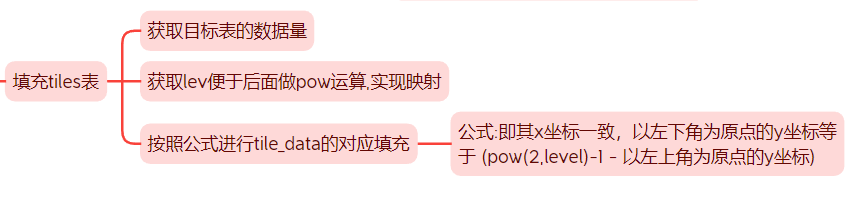
|  |
| --- |
| **#开始tiles表的映射**  **for i in range(minzoom,maxzoom+1):**  **path1 = filepath+'/'+str(i) #文件夹目录**  **files1= os.listdir(path1) #得到文件夹下的所有文件名称**  **#print(files1)**  **orders=pow(2,i)**  **try:**  **for j in range(orders-1,-1,-1):**  **path2 = path1+'/'+str(j)#文件夹目录**  **if formatt==' ':**  **files2= os.listdir(path2) #得到文件夹下的所有文件名称**  **pic=files2[0]**  **formatt=pic[pic.index('.')+1:len(pic)]**  **#print(formatt)**  **try:**  **for k in range(0,orders):**  **#tiles表开始填入数据**  **sqlite\_insert\_blob\_query = """ INSERT INTO tiles**  **(zoom\_level, tile\_column, tile\_row, tile\_data,uptime) VALUES (?, ?, ?, ?,?)"""**  **empPhoto = convertToBinaryData(path2+'/'+str(k)+'.'+formatt)**  **# Convert data into tuple format**  **data\_tuple = (i,k,(orders-1)-j,empPhoto,datetime.datetime.now())**  **cursor.execute(sqlite\_insert\_blob\_query, data\_tuple)**  **sqliteConnection.commit()**  **except sqlite3.Error as error:**  **print('error,映射tiles表出错,找不到文件,',k,'列下标图片是否连续？',error)**  **except sqlite3.Error as error:**  **print('error,映射tiles表出错,找不到文件,',j,'行文件数是否连续？',error)**  **print('tiles表映射结束。')** |

**功能二的逻辑，**传入一个a数据库的绝对路径如D:\中导航\Convertmbtiles\ArcGisStreetWorld10.db，和原点在左上角的表转化的b表的表名如（tiles）。将左上角.db数据库变为左下角.db数据库**：**



**核心代码：**

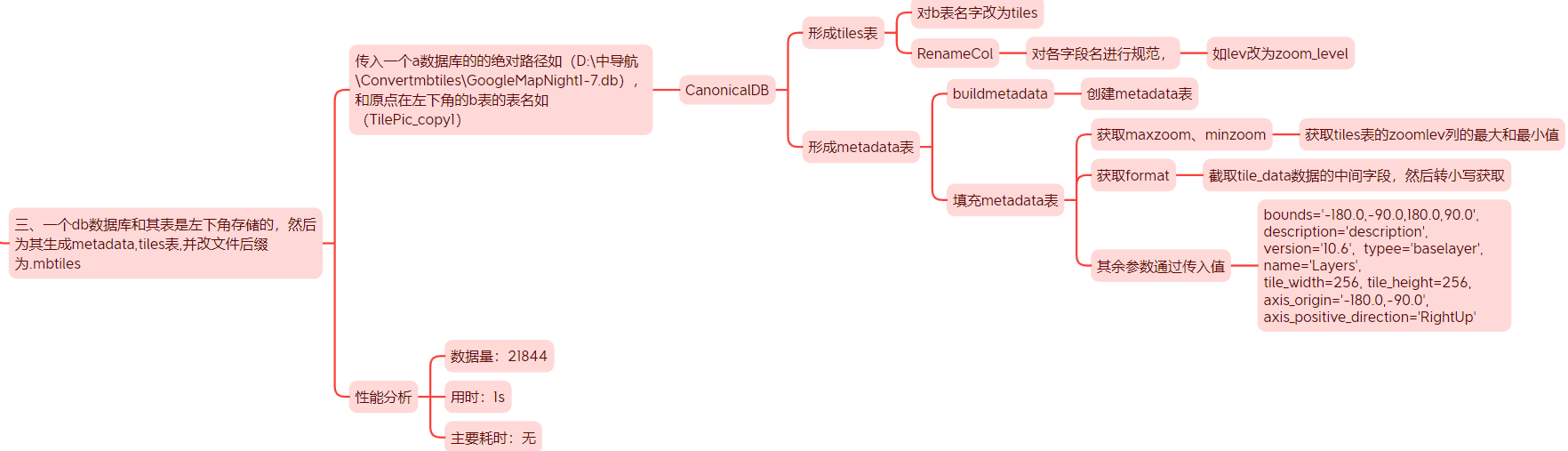
**思路图：**



**代码：**

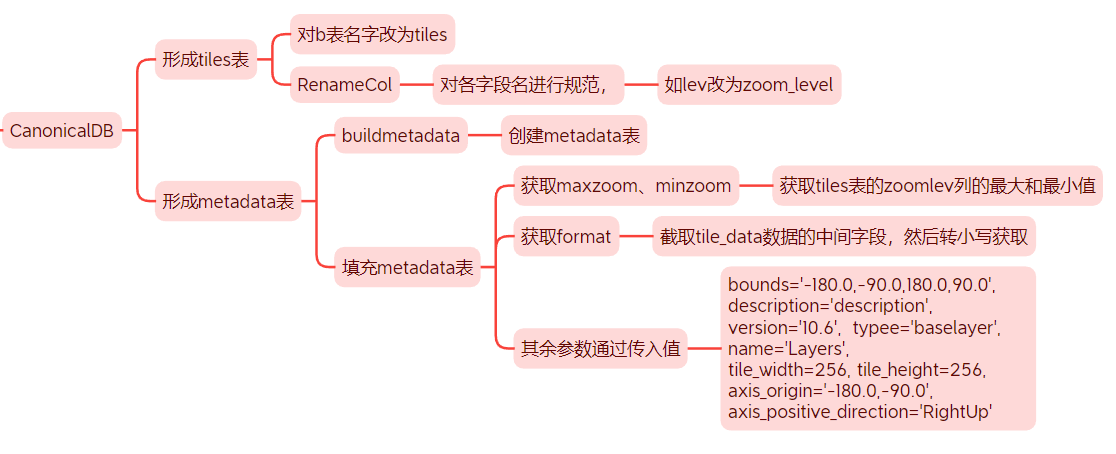
|  |
| --- |
| **print('开始tiles表的映射......')**  **sql = 'update tiles set tile\_data = (select '+tile\_data+' from '+orderTableName+' where '+\**  **zoom\_level+'=tiles.zoom\_level and '+tile\_column+'=tiles.tile\_column and '+\**  **tile\_row+'=(1<<tiles.zoom\_level)-1-tiles.tile\_row)'**  **#print(sql)**  **cursor.execute(sql)**  **sqliteConnection.commit()**  **print('tiles表映射结束。')** |

**功能三的逻辑，**传入一个a数据库的的绝对路径如（D:\中导航\Convertmbtiles\GoogleMapNight1-7.db），和原点在左下角的b表的表名如（TilePic\_copy1）,为其创建metadata并将b表变为标准的tiles表**：**



**核心代码（规范目标表为tiles表,获取metadata的maxzoom、minzoom、format属性值）**

**思路图：**



**代码：**

|  |
| --- |
| **#重命名数据库表格字段名函数,parameter:sqliteConnection,cursor,tablename,colname,newcolname**  **RenameCol(sqliteConnection,cursor,orderTableName,zoom\_level,'zoom\_level'); RenameCol(sqliteConnection,cursor,orderTableName,tile\_column,'tile\_column');**  **RenameCol(sqliteConnection,cursor,orderTableName,tile\_row,'tile\_row'); RenameCol(sqliteConnection,cursor,orderTableName,tile\_data,'tile\_data');**  **#1、获取maxzoom、minzoom**  **sql = '''SELECT max(zoom\_level),min(zoom\_level) FROM tiles'''#从规范的tiles中的zoom\_level列，获取到maxzoom、minzoom**  **cursor.execute(sql)**  **ll = cursor.fetchall()**  **maxzoom = ll[0][0];minzoom = ll[0][1]**  **#print(maxzoom,minzoom)**  **#2、获取format**  **formatt=''**  **sql = '''SELECT tile\_data FROM tiles where Rowid=1'''#得到tiles表中第一行tile\_data的数据，从中得到数据类型**  **cursor.execute(sql)**  **tile\_data1 = cursor.fetchall()**  **t = tile\_data1[0][0]#得到tiles表中第一行tile\_data的数据t**  **formatt=''.join(list(str(t[1:5]))[2:5]).lower()#从数据t中拆分出展示其数据类型的字段,如：png#print(formatt)** |

**功能四的逻辑，**从dbpath中切出文件夹地址，和文件名用于修改后缀，修改文件.db的后缀名，变.mbtiles文件**：**

**[技术介绍]**  
该项目主要采用以下技术实现：

Python语言。

Miniconda内置模块。

SQLite数据库。

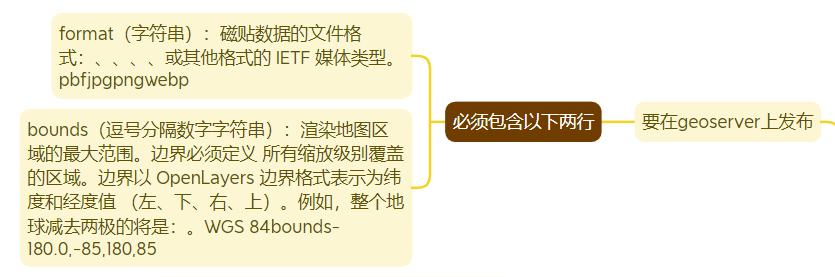
异常处理：对异常进行捕获和处理，确保系统的稳定性和可用性。

文档测试：定期进行文档测试，保证文档的质量和可读性。

**[mbtiles文件介绍]**

进行mbtiles测试流程:

用geoserver进行metadata表的属性删除实验，得出要在geoserver上发布至少包括bounds和format。



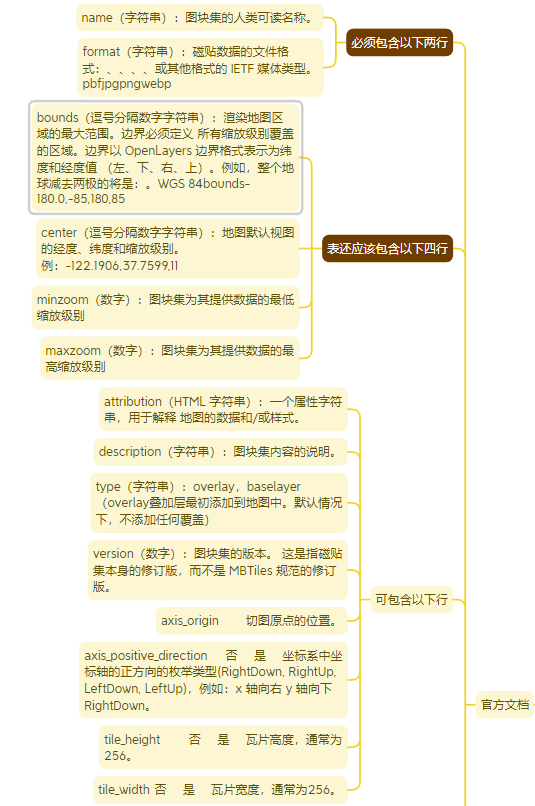
**.mbtiles文件特点:**

****

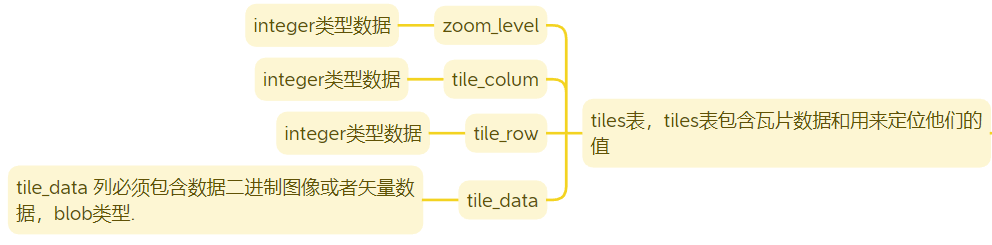
**.mbtiles文件构成:**

**数据库必须包含metadata元数据字段，tiles表，可以包含map 数据表，images 瓦片数据表。**

**metadata元数据字段:**



**tiles表:**

**map 数据表和images 瓦片数据表:**

****

**[注意事项]**



**Problem::**

**其中power在python的sqlite中会找不到如以下语句:**

**UPDATE tiles**

**SET tile\_data = (**

**SELECT**

**pic**

**FROM**

**TilePic**

**WHERE**

**lev = tiles.zoom\_level**

**AND mapCol = tiles.tile\_column**

**AND mapRow = power ( 2, tiles.zoom\_level )-1 - tiles.tile\_row**

**)**

**在python中运行会报错:no such function: power**

**这里用位运算符代替如power(2,0) = 1 = 1<<0; power(2,1)=2=1<<1...最多select (1 << 63)，即如果(1<<64)则返回0，超限。**

## 任务三主要内容：

### 概述：

然后要做整理资料的任务，即将以下数据：知识图谱、航空要素、地理信息、图表样式的数据类型、数据要素和相关图片整理，并形成文档，要求相关的都要包含，有图表和参考文献，形成了万字的整理文档，并和合作伙伴讨论，完成了这个任务。

### 相关数据汇总（详情见附件“任务三知识图谱、航空要素等相关数据汇总”）：

**知识图谱**

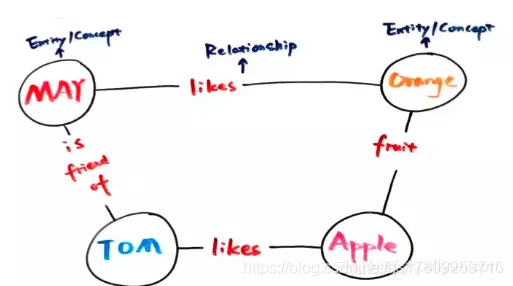
**定义：**



知识图谱：知识图谱是一种以图形化方式表达知识的工具，其中包含了实体、属性和关系。这种数据通常用于语义网络和人工智能领域，用于更好地理解和解析信息。

1.本质上是一种叫做语义网络（semantic network）的知识库，即具有有向图结构的一个知识库，其中图的结点代表实体（entity）或者概念（concept），而图的边代表实体/概念之间的各种语义关系，比如说两个实体之间的相似关系。

换个角度，从实际应用的角度出发其实可以简单地把知识图谱理解成多关系图（Multi-relational Graph）。



如图所示，你可以看到，如果两个节点之间存在关系，他们就会被一条无向边连接在一起，那么这个节点，我们就称为实体（Entity），它们之间的这条边，我们就称为关系（Relationship）。实体指的是现实世界中的事物比如人、地名、概念、药物、公司等， 关系则用来表达不同实体之间的某种联系，比如人-“居住在”-北京、张三和李四是“朋友”、逻辑回归是深度学习的“先导知识”等等**[1]**。

2.知识图谱的基本单位，便是“实体（Entity）-关系（Relationship）-实体（Entity）”构成的三元组（主语，谓词，宾语），这也是知识图谱的核心。

3.在Web视角下，知识图谱如同简单文本之间的超链接一样，通过建立数据之间的语义链接，支持语义搜索。

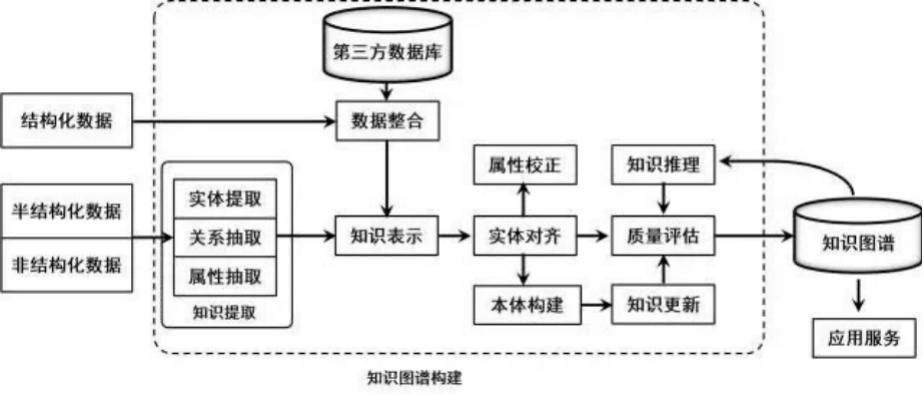
在自然语言处理视角下，知识图谱就是从文本中抽取语义和结构化的数据。

在知识表示视角下，知识图谱是采用计算机符号表示和处理知识的方法。

在人工智能视角下，知识图谱是利用知识库来辅助理解人类语言的工具。

在数据库视角下，知识图谱是利用图的方式去存储知识的方法**[2]**。

4.知识图谱的整体架构**[4]** 如下图所示，其中虚线框内的部分为知识图谱的构建过程，同时也是知识建立和更新的主要流程:



**数据类型**

知识图谱的数据类型主要有三种：

1. 结构化数据（Structed Data），例如关系数据库中的数据，具有特定的结构，以表格的形式呈现[1]。

也称作行数据，一般特点是：数据以行为单位，一行数据表示一个实体的信息，每一行数据的属性是相同的, 如下图[5]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | name | Gender | phone | address |
| 1 | 张一 | Female | 3337899 | 湖北省武汉市 |
| 2 | 王二 | Male | 3337499 | 广东省深圳市福田区 |
| 3 | 李三 | Female | 3339003 | 广东省深圳市南山区 |

1. 非结构化数据，如图片、音频、视频等，没有固定的结构，这类数据往往难以处理和利用[1]。

包括所有格式的办公文档、文本、图片、HTML、各类报表、图像和音频/视频信息等等。非结构化数据更难让计算机理解[5]。

1. 半结构化数据，如XML、JSON、百科等，具有一定的结构，但并不如结构化数据那样严格和固定[1]。

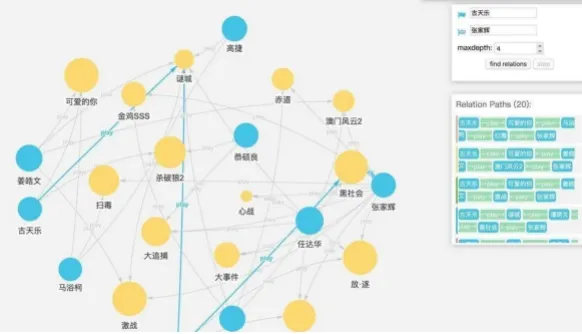
其是结构化数据的一种形式，虽不符合关系型数据库或其他数据表的形式关联起来的数据模型结构，但包含相关标记，用来分隔语义元素以及对记录和字段进行分层。因此，也被称为自描述的结构。

常见的半结构数据有XML和JSON，比如：如下表[5]。

|  |
| --- |
| <person>  <name>A</name>  <age>13</age>  <gender>female</gender>  </person> |

目前，主流的的知识存储解决方案包含单一式和混合式存储两种。其存储方式一般有两种选择:

一个是通过RDF（资源描述框架）这样的规范存储格式来进行存储，比较常用 的有Jena等。还有一种方法，就是使用图数据库来进行存储[4]，常用的如 Neo4j。

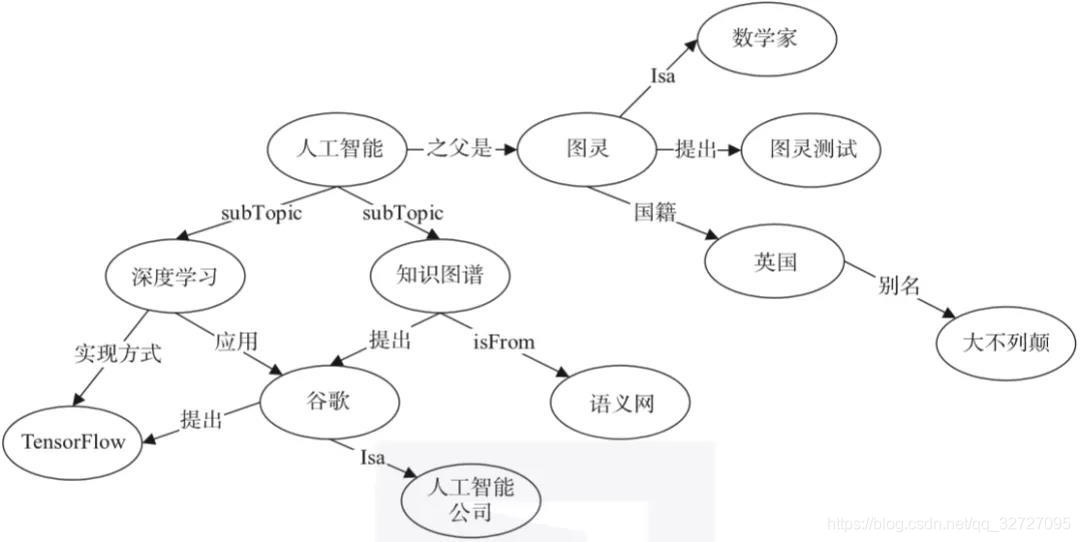


构建民航业知识图谱知识图谱属性类型[6]

|  |  |  |
| --- | --- | --- |
| 属性类型 | 含义 | 举例 |
| info | 年份或目录的整体概况 | 全年航空安全形势稳定，旅客运输和… |
| name | 各实体名称 | 2014，国内，货邮周转量 |
| value | 指标或地区的值 | 451.2，0，22 |
| unit | 指标或地区值的单位 | 亿吨公里，%，万元 |
| repr | 地区值所代表的含义 | 航线（不含港澳台），吞吐量占比 |
| child\_id | 同一（父）指标下不同（子）指标拥有相同child\_id的可视为构成父指标的一个角度 | 0，1，2 … |
| life | （结构关系）生命周期 | 0x179, 0x008 |

**数据要素**

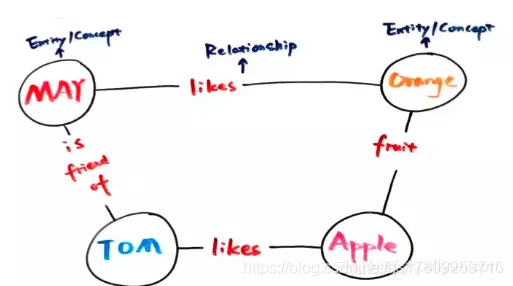
知识图谱的数据要素主要包括以下几类：

1. 实体：知识图谱中的基本元素，表示具有可区别性、独立存在的事物, 又叫作本体（Ontology），指客观存在并可相互区别的事物，可以是具体的人、事、物，也可以是抽象的概念或联系。实体是知识图谱中最基本的元素。。
2. 属性：实体的特征，指向特征值，类似于关系的“类别”, 知识图谱中的实体和关系都可以有各自的属性[3]。
3. 关系：实体间的关联, 在知识图谱中，表示知识图谱中的关系，用来表示不同实体间的某种联系。如下图所示，图灵和人工智能之间的关系，知识图谱和谷歌之间的关系，谷歌和深度学习之间的关系[3]。

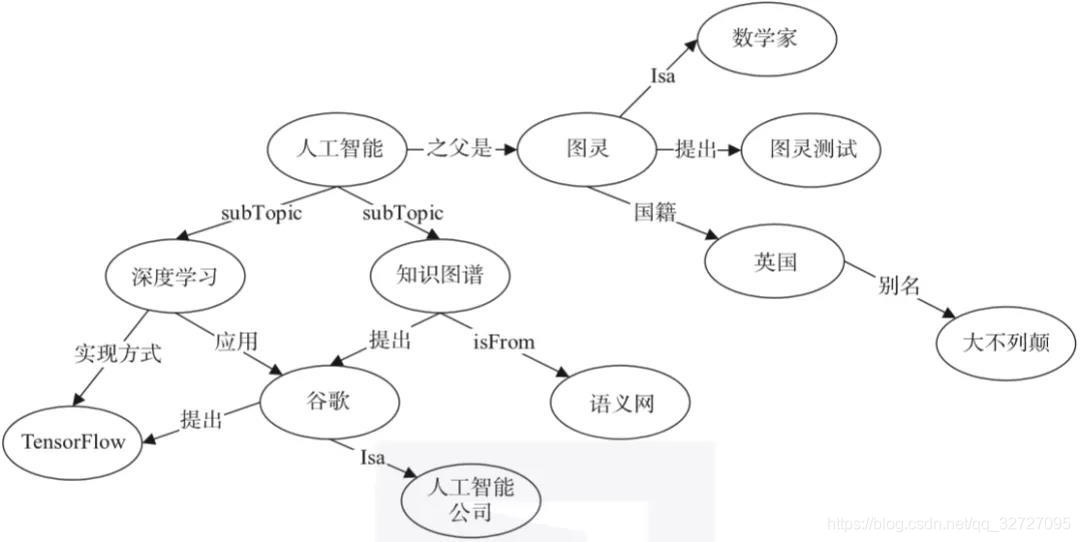
**相关图片**



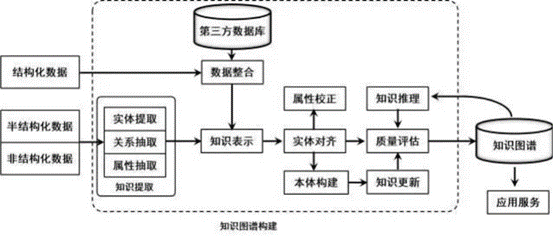
1,把知识图谱理解成多关系图（Multi-relational Graph）[1]



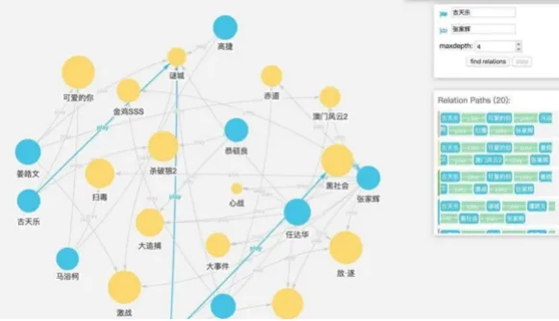
2,图灵和人工智能之间的关系，知识图谱和谷歌之间的关系，谷歌和深度学习之间的关系[3]



3,其中虚线框内的部分为知识图谱的构建过程，同时也是知识建立和更新的主要流程[4]



4, 使用图数据库来进行存储[4]，常用的如Neo4j



**图表样式**

知识图谱的图表样式通常包括以下几种：

实体关系图：实体关系图是知识图谱的基本组成形式，它以实体为节点，以关系为边，表示实体之间的相互关系。实体关系图通常采用图形化的方式展示实体和关系的结构和属性，可以清晰地展示实体之间的复杂关系。

树状图：树状图是用来展示层次结构的一种图表样式，常用于展示分类信息。在知识图谱中，树状图可以用来展示实体之间的层次关系和分类关系，例如实体之间的父子关系、公司组织结构等。

网状图：网状图是一种以节点和边表示复杂网络关系的图表样式，通常用于展示多个实体之间的复杂关系。在知识图谱中，网状图可以用来展示实体之间的多对多关系、交错关系等，例如社交网络中的人际关系、学术论文之间的引用关系等。

时序图：时序图是一种以时间轴为轴线的图表样式，用于展示数据或事件随时间的变化情况。在知识图谱中，时序图可以用来展示实体或事件随时间的变化情况，例如一个人的成长历程、公司股票价格的波动情况等。

地图：地图是一种用于展示地理信息的图表样式，通常包括地理区域、地形、交通路线等元素。在知识图谱中，地图可以用来展示实体的地理位置和分布情况，例如城市的人口分布、旅游景点的分布情况等。

这些图表样式可以根据实际需要和数据特点进行选择和调整，例如调整颜色、形状、标签等视觉元素，添加图例、标注、网格线等辅助元素，优化图表的交互效果等。同时，不同的图表样式也适用于不同的数据类型和展示需求，需要根据具体情况进行选择和调整。

**参考链接：**

[1][知识图谱综述（初步了解后的总结整理）\_知识图谱中图片、音频、视频、文章等数据-CSDN博客](https://blog.csdn.net/qs17809259715/article/details/103793220#:~:text=%E5%9B%9B%E3%80%81%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E7%9A%84%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B%E5%92%8C%E5%AD%98%E5%82%A8%E6%96%B9%E5%BC%8F%201%20%E7%BB%93%E6%9E%84%E5%8C%96%E6%95%B0%E6%8D%AE%EF%BC%88Structed%20Data%EF%BC%89%EF%BC%8C%E5%A6%82%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93,2%20%E9%9D%9E%E7%BB%93%E6%9E%84%E5%8C%96%E6%95%B0%E6%8D%AE%EF%BC%8C%E5%A6%82%E5%9B%BE%E7%89%87%E3%80%81%E9%9F%B3%E9%A2%91%E3%80%81%E8%A7%86%E9%A2%91%203%20%E5%8D%8A%E7%BB%93%E6%9E%84%E5%8C%96%E6%95%B0%E6%8D%AE%20%E5%A6%82XML%E3%80%81JSON%E3%80%81%E7%99%BE%E7%A7%91)

一、导读 二、知识图谱的概念演化

**三、 什么是知识图谱 四、知识图谱的数据类型和存储方式**

五、知识图谱的架构 六、知识图谱的构建技术

七、 知识图谱的应用 八、总结

[2] [终于有人把知识图谱讲明白了-CSDN博客](https://blog.csdn.net/zw0pi8g5c1x/article/details/113930459#:~:text=1%20%E5%AE%9E%E4%BD%93%EF%BC%9A%20%E5%8F%88%E5%8F%AB%E4%BD%9C%E6%9C%AC%E4%BD%93%EF%BC%88Ontology%EF%BC%89%EF%BC%8C%E6%8C%87%E5%AE%A2%E8%A7%82%E5%AD%98%E5%9C%A8%E5%B9%B6%E5%8F%AF%E7%9B%B8%E4%BA%92%E5%8C%BA%E5%88%AB%E7%9A%84%E4%BA%8B%E7%89%A9%EF%BC%8C%E5%8F%AF%E4%BB%A5%E6%98%AF%E5%85%B7%E4%BD%93%E7%9A%84%E4%BA%BA%E3%80%81%E4%BA%8B%E3%80%81%E7%89%A9%EF%BC%8C%E4%B9%9F%E5%8F%AF%E4%BB%A5%E6%98%AF%E6%8A%BD%E8%B1%A1%E7%9A%84%E6%A6%82%E5%BF%B5%E6%88%96%E8%81%94%E7%B3%BB%E3%80%82%20%E5%AE%9E%E4%BD%93%E6%98%AF%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E6%9C%80%E5%9F%BA%E6%9C%AC%E7%9A%84%E5%85%83%E7%B4%A0%E3%80%82%202,%E5%85%B3%E7%B3%BB%EF%BC%9A%20%E5%9C%A8%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%EF%BC%8C%E8%BE%B9%E8%A1%A8%E7%A4%BA%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E7%94%A8%E6%9D%A5%E8%A1%A8%E7%A4%BA%E4%B8%8D%E5%90%8C%E5%AE%9E%E4%BD%93%E9%97%B4%E7%9A%84%E6%9F%90%E7%A7%8D%E8%81%94%E7%B3%BB%E3%80%82%20%E5%A6%82%E5%9B%BE3-1%E6%89%80%E7%A4%BA%EF%BC%8C%E5%9B%BE%E7%81%B5%E5%92%8C%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E5%92%8C%E8%B0%B7%E6%AD%8C%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E8%B0%B7%E6%AD%8C%E5%92%8C%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%E3%80%82%203%20%E5%B1%9E%E6%80%A7%EF%BC%9A%20%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E7%9A%84%E5%AE%9E%E4%BD%93%E5%92%8C%E5%85%B3%E7%B3%BB%E9%83%BD%E5%8F%AF%E4%BB%A5%E6%9C%89%E5%90%84%E8%87%AA%E7%9A%84%E5%B1%9E%E6%80%A7%EF%BC%8C%E5%A6%82%E5%9B%BE3-2%E6%89%80%E7%A4%BA%E3%80%82)

**01 什么是知识图谱 02 知识图谱的价值 03 知识图谱的架构**

[3] [什么是知识图谱？通俗易懂\_一个写湿的程序猿的博客-CSDN博客](https://blog.csdn.net/qq_32727095/article/details/114290102#:~:text=1%20%E5%AE%9E%E4%BD%93%20%EF%BC%9A%E5%8F%88%E5%8F%AB%E4%BD%9C%E6%9C%AC%E4%BD%93%EF%BC%88Ontology%EF%BC%89%EF%BC%8C%E6%8C%87%E5%AE%A2%E8%A7%82%E5%AD%98%E5%9C%A8%E5%B9%B6%E5%8F%AF%E7%9B%B8%E4%BA%92%E5%8C%BA%E5%88%AB%E7%9A%84%E4%BA%8B%E7%89%A9%EF%BC%8C%E5%8F%AF%E4%BB%A5%E6%98%AF%E5%85%B7%E4%BD%93%E7%9A%84%E4%BA%BA%E3%80%81%E4%BA%8B%E3%80%81%E7%89%A9%EF%BC%8C%E4%B9%9F%E5%8F%AF%E4%BB%A5%E6%98%AF%E6%8A%BD%E8%B1%A1%E7%9A%84%E6%A6%82%E5%BF%B5%E6%88%96%E8%81%94%E7%B3%BB%E3%80%82%20%E5%AE%9E%E4%BD%93%E6%98%AF%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E6%9C%80%E5%9F%BA%E6%9C%AC%E7%9A%84%E5%85%83%E7%B4%A0%E3%80%82%202%20%E5%85%B3%E7%B3%BB,%EF%BC%9A%E5%9C%A8%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%EF%BC%8C%E8%BE%B9%E8%A1%A8%E7%A4%BA%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E7%94%A8%E6%9D%A5%E8%A1%A8%E7%A4%BA%E4%B8%8D%E5%90%8C%E5%AE%9E%E4%BD%93%E9%97%B4%E7%9A%84%E6%9F%90%E7%A7%8D%E8%81%94%E7%B3%BB%E3%80%82%20%E5%A6%82%E5%9B%BE3-1%E6%89%80%E7%A4%BA%EF%BC%8C%E5%9B%BE%E7%81%B5%E5%92%8C%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E5%92%8C%E8%B0%B7%E6%AD%8C%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%EF%BC%8C%E8%B0%B7%E6%AD%8C%E5%92%8C%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E4%B9%8B%E9%97%B4%E7%9A%84%E5%85%B3%E7%B3%BB%E3%80%82%203%20%E5%B1%9E%E6%80%A7%20%EF%BC%9A%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E7%9A%84%E5%AE%9E%E4%BD%93%E5%92%8C%E5%85%B3%E7%B3%BB%E9%83%BD%E5%8F%AF%E4%BB%A5%E6%9C%89%E5%90%84%E8%87%AA%E7%9A%84%E5%B1%9E%E6%80%A7%EF%BC%8C%E5%A6%82%E5%9B%BE3-2%E6%89%80%E7%A4%BA%E3%80%82%20%E2%96%B2%E5%9B%BE3-2%20%E7%9F%A5%E8%AF%86%E5%9B%BE%E8%B0%B1%E4%B8%AD%E7%9A%84%E5%B1%9E%E6%80%A7)

01 什么是知识图谱 02 知识图谱的价值 03 知识图谱的架构

[4] [知识图谱构建流程详解 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/145447330)

引言 知识图谱的基本定义 一、知识建模 二、知识存储 三、 信息抽取 四、 知识融 合 五、知识计算 六、知识图谱的应用

[5] [结构化数据和非结构化数据、半结构化数据的区别\_结构化数据 非结构化数据 半结构化数据-CSDN博客](https://blog.csdn.net/alice_tl/article/details/88142721)

[6] [构建民航业知识图谱并实现语义查询-CSDN博客](https://blog.csdn.net/qq_25000387/article/details/112436982)

知识图谱实体类型 知识图谱实体关系类型 知识图谱属性类型

**航空要素**

**定义：**

1.航空要素：指的是航空领域的各种数据元素，如飞机、导航台、机场、跑道、航线、天气情况等。这些数据通常以表格或数据库的形式存在，也可能包括地理信息。

2.航空要素是指直接服务于飞行的专门要素。航空图上用紫色突出表示[5]。

**数据类型**

数据类型：

1. 结构化数据：包括飞行数据、航班信息、航空器性能数据、飞行员资质信息、航材信息等。这些数据通常以表格或数据库的形式存储，具有固定的结构和格式。
2. 非结构化数据：包括航空图像、语音记录、文字记录、航空器维护记录等。这些数据没有固定的结构，需要借助自然语言处理等技术进行处理和分析。
3. 半结构化数据：包括航空领域的知识文章、研究报告、论文等。这些数据具有一定的结构，但并不如结构化数据那样严格和固定。

根据民航数据的内容和形式，可以将民航数据分为以下几个大类[1]：

1. 航空运输数据

航空运输数据包括航班信息、机场信息、航空公司信息等。航班信息包括航班号、起降时间、航线等；机场信息包括机场代码、机场位置、机场设施等；航空公司信息包括航空公司代码、航空公司性质、航空公司航线等。

2. 客运数据

客运数据主要包括旅客运输数据和货物运输数据。旅客运输数据包括旅客数量、旅客类型、旅客舱位等；货物运输数据包括货物数量、货物种类、货物舱位等。

3. 安全数据

安全数据主要包括航空事故数据和安全指标数据。航空事故数据包括事故类型、事故原因、事故地点等；安全指标数据包括飞行安全指标、地面安全指标等。

4. 运行数据

运行数据主要包括航班运行数据和机场运行数据。航班运行数据包括航班延误情况、航班准点率等；机场运行数据包括航班起降数量、航班滑行时间等。

5. 经济数据

经济数据包括航空公司经济数据和航空产业经济数据。航空公司经济数据包括航空公司营业收入、航空公司利润等；航空产业经济数据包括航空产值、航空就业情况等。

导航数据库内的数据分类[4]:

基础数据（Basic Data）也称标准数据（Standard Data）:

它来自于各国对外公布的航行资料（包括AIP等），数据供应商（比如JEPPESEN）自动获取并制成电子数据，并提供各厂商加入自身的导航数据库。数据类型有航路、导航台、位置点、机场和跑道位置信息、终端区程序。

公司数据（Company Data）也称客户化数据（Tailored Data）:

这部分数据未公布于航行资料，或者所公布的航行资料无法被JEPPESEN获取。因此，需要由航空公司根据实际需要，以规定的格式提交给导航数据库厂商加入数据库。类型包括导航台、位置点、机场和跑道位置信息、终端区程序、公司航路。

航图的分类[5]：

（1）航空设施（2）地磁资料（3）空中特区与垂直障碍物

**数据要素**

数据要素：

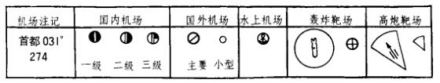
1. 航空器信息：包括航空器的型号、序列号、制造商、使用时间、维修记录等信息。
2. 航班信息：包括航班号、起飞时间、到达时间、起飞地点、目的地、航路信息、机组成员信息等。
3. 飞行数据：包括飞行高度、速度、航向、加速度、温度等飞行参数。
4. 安全信息：包括安全事故、安全隐患、安全警告等信息，用于分析安全态势和风险评估。
5. 运营信息：包括航空公司的运营计划、航班调度、机务维护等信息。
6. 乘客信息：包括乘客姓名、性别、年龄、购票信息等。
7. 位置信息：包括机场位置、航路位置、航空器位置等信息。
8. 气象信息：包括风速、风向、气温、气压等气象参数。
9. 油耗信息：包括航空器消耗的燃油量、燃油消耗率等参数。
10. 其他相关领域的信息，如政治、经济、文化等方面的信息，用于综合分析和决策。

这些数据要素可以用于构建航空领域的知识图谱，提供更加智能化的决策支持和服务。

**航图中航空要素[5]：**

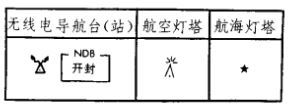
**（1）航空设施[5]**

指保证飞机安全起降和引导飞行的设施。

[](https://baike.baidu.com/pic/%E8%88%AA%E7%A9%BA%E5%9B%BE/4951315/0/77094b36acaf2eddc04a77c8841001e939019353?fr=lemma&fromModule=lemma_content-image)航空设施符号

**机场：**通常以特定符号表示，如右图。符号中心为主要跑道的中点，直径线方向表示跑道方向，符号旁注有机场名称、跑道标高和方位角。

**靶场：**为供飞机训练、试验和高炮射击的场区。图上用依比例尺和不依比例尺两种符号表示，如下图。

[](https://baike.baidu.com/pic/%E8%88%AA%E7%A9%BA%E5%9B%BE/4951315/0/29381f30e924b899ae11401967061d950a7bf6e5?fr=lemma&fromModule=lemma_content-image)灯塔与导航台符号

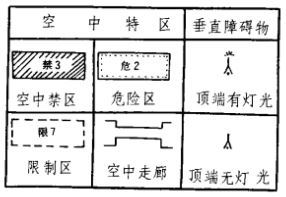
**导航台（站）：**图上符号如上图，右侧方框内的文字为台(站)名称，NDB表示该台(站)为无方向性无线电信标。若为VOR，则为甚高频全向信标。

**导航灯光：**通常只表示航空和航海灯塔及灯船。

**（2）地磁资料**

为便于利用罗盘领航，航空图上标有等磁偏线和[磁力异常区](https://baike.baidu.com/item/%E7%A3%81%E5%8A%9B%E5%BC%82%E5%B8%B8%E5%8C%BA/0?fromModule=lemma_inlink)（点）。

**（3）空中特区与垂直障碍物[5]**

[](https://baike.baidu.com/pic/%E8%88%AA%E7%A9%BA%E5%9B%BE/4951315/0/738b4710b912c8fc7752cfecf5039245d6882169?fr=lemma&fromModule=lemma_content-image)空中特区与障碍物符号

一个国家根据某种考虑，在领土或领海上空划定的各种特定空域，叫做空中特区。图上用紫色绘出边线，中间加绘晕线(点)并注明特区类型、编号和所属国家，如右图所示。

以机场为中心，半径15km范围内独立、高大，对飞行安全有潜在危险的建筑物叫垂直障碍物。图上亦用紫色特定符号表示，并注有比高和顶端高程，如右图所示。

**导航台的数据要素:**

导航台的数据要素主要包括以下几类：

位置信息：导航台的位置信息是关键的数据要素之一，包括经纬度坐标、海拔高度、地面高程等。这些信息用于确定导航台在空间中的位置，以及与其他导航台或目标之间的相对位置关系。

信号参数：导航台的信号参数包括信号频率、信号功率、信号传播路径等。这些参数用于描述导航台发射的信号特性，以及信号在空间中的传播方式和范围。

时间信息：导航台的时间信息包括系统时间、授时信号等，用于提供准确的时间信息，以及与其他导航台或目标进行时间同步。

交通信息：导航台的交通信息包括车辆位置、速度、方向等，用于实时监测和显示交通状况，以及为车辆提供导航和路线规划。

气象信息：导航台的气象信息包括温度、湿度、气压、风速等，用于提供实时的气象数据，以及为飞行器提供气象预警和航路规划。

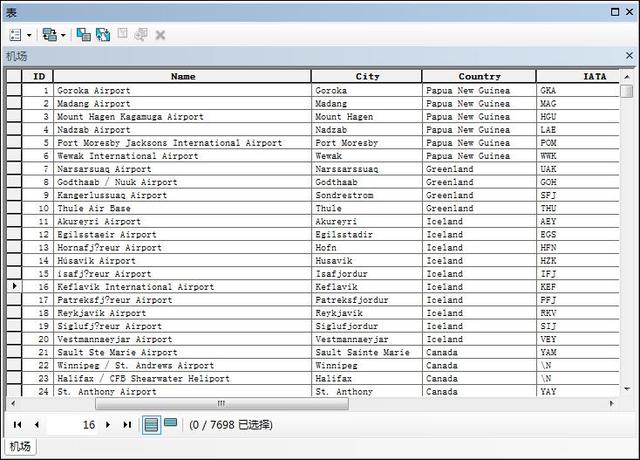
设备状态：导航台的设备状态包括工作状态、故障状态、维护状态等，用于监测设备的运行状况，以及及时发现和处理设备故障。

用户信息：导航台的用户信息包括用户身份、权限等级等，用于管理用户登录和操作权限，确保系统的安全性和稳定性。

这些数据要素在导航台的运营和管理中发挥着重要的作用，例如通过监测交通信息和气象信息，可以及时调整交通管制措施和航路规划方案，提高交通运营的效率和安全性；通过管理设备状态和用户信息，可以确保系统的稳定性和安全性，避免因设备故障或非法操作导致的风险。

**机场的数据要素：**

打开属性表，可以看到详细的字段信息，如下图所示[2]。



各个字段的含义如下：

|  |
| --- |
| ID 此机场唯一的标识符  Name 机场名称，可能包含也可能不包含城市名称。  City 主要城市由机场服务，可能与Name拼写不同。  Country 机场所在的国家或地区。  IATA 3个字母的IATA代码，如果未分配/未知，则为空。  ICAO 4个字母的ICAO代码，如果未分配，则为空。  Latitude 纬度，十进制度，通常为六位有效数字，负为南，正为北。  Longitude 经度，十进制度，通常为六位有效数字，负为西，正为东。  Altitude 海拔，单位英尺。  Timezone 从 UTC 偏移的小时数，小数小时表示为小数。  DST 夏令时。  Tz database time “tz”（奥尔森）格式的时区。  Type 机场类型。 |

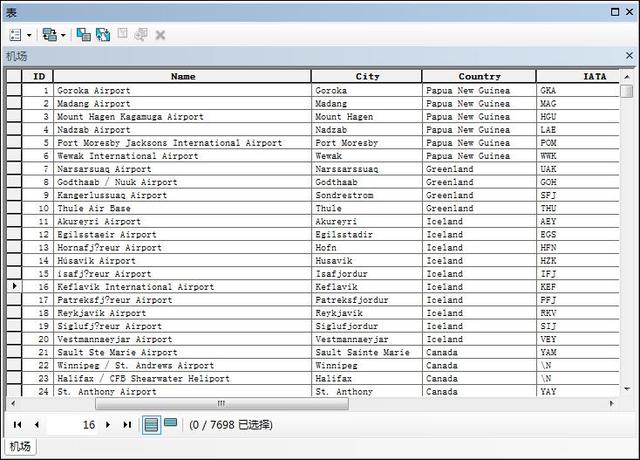
**跑道的数据要素：**

跑道是一个机场的重要组成部分。它决定了机场的等级标准 ，跑道及其相关设施的修建、标识等是有严格规定的[3]。

|  |  |
| --- | --- |
| 方向和跑道号 | 主跑道的方向一般和当地的主风向一致，跑道号按照跑道中心线的磁方向以10度为单位；四舍五人用两位数表示。 |
| 基本尺寸 | 指跑道的长度、宽度和坡度。 |
| 道面 | 跑道道面分为刚性和非刚性道面 |
| 强度 | PCN数是由道面的性质，道面基础的承载强度经技术评估而得出的，每条跑道都有一个PCN值。 |
| 跑道附属区域 | 滑行道 |
| 净空道 |
| 跑道道肩 |
| 跑道安全带 |
| 侧安全地带 |
| 道端安全地带 |

**相关图片**

1,打开属性表，可以看到详细的字段信息，如下图所示[2]



**图表样式**

航空要素的图表样式通常包括以下几种：

折线图：折线图可以清晰地展示飞行数据随时间的变化趋势，例如飞行高度、速度、航向等参数的变化情况。通过将折线图与时间轴结合，可以更直观地了解飞行过程中各个参数的变化趋势。

柱状图：柱状图是一种用于比较航空数据的图表样式，可以用于比较不同航班、不同机型、不同时间段等各个类别之间的差异。通过柱状图的排列和对比，可以更直观地了解各个类别之间的差异情况。

饼图：饼图是一种用于表示航空数据比例关系的图表样式，可以用于表示航班各个舱位、各个乘客类型等比例情况。通过饼图的展示，可以更直观地了解各个类别所占的比例情况。

散点图：散点图是一种用于展示航空数据相关关系的图表样式，可以用于展示两个变量之间的相关关系。通过散点图的展示，可以更直观地了解两个变量之间的相关关系情况。

气泡图：气泡图是一种多变量的统计图表，由笛卡尔坐标系（直角坐标系）和大小不一的圆组成，可以看作是散点图的变形。通常用于展示和比较飞行数据中两个变量的变化情况以及它们之间的相关性。

雷达图：雷达图是一种显示多变量数据的图形方法。通常从同一中心点开始等角度间隔地射出三个以上的轴，每个轴代表一个定量变量。可以用来在变量间进行对比,或者查看变量中有没有异常值。

堆积柱状图：堆积柱状图可以用于展示航空数据的总量和各分类数据量的关系。例如，可以用来表示某个机场各个航班的总起飞次数、各航班的起飞次数以及各航班所占的比例关系。

这些图表样式可以根据实际需要和数据特点进行选择和调整，例如调整颜色、形状、标签等视觉元素，添加图例、标注、网格线等辅助元素，优化图表的交互效果等。同时，不同的图表样式也适用于不同的数据类型和展示需求，需要根据具体情况进行选择和调整。

**参考链接：**

[1] [民航数据分类分级标准 - 百度文库 (baidu.com)](https://wenku.baidu.com/view/9cb3f3540422192e453610661ed9ad51f11d5446.html?_wkts_=1698042963683&bdQuery=%E8%88%AA%E7%A9%BA%E8%A6%81%E7%B4%A0%E7%9A%84%E6%95%B0%E6%8D%AE%E7%B1%BB%E5%9E%8B)

[2] [全球机场数据分享 (baidu.com)](https://baijiahao.baidu.com/s?id=1729871989737670740&wfr=spider&for=pc)

数据展示 字段含义 机场热力图 结语

[3] [航空知识解读：跑道的组成、标准和参数 (qq.com)](https://mp.weixin.qq.com/s?src=3&timestamp=1698043957&ver=1&signature=3q7PI6TKpO04F0y36bZHJcU6I3pYfQ9mOvtCvMYUCqD9d9iZwArzs1Re7VjG0qv7MNKDzk4S6Fab6iFTPpMz3qZUd1QdBCRUrouJ0aFk*JFxZHPHrc2ffA3Lk5VmnK1hoR0TdMKX5G6QNWmIxpV4ew==)

跑道的基本参数 方向和跑道号

[4] [【短小科普】机载导航数据库 - 哔哩哔哩 (bilibili.com)](https://www.bilibili.com/read/cv11554635/)

一、导航数据库内的数据分类 二、导航数据库生效当日的周期切换 三、两类典型的公司数据 四、航径代码的含义 五、导航数据库与地形数据库区别

[5] [航空图\_百度百科 (baidu.com)](https://baike.baidu.com/item/%E8%88%AA%E7%A9%BA%E5%9B%BE/4951315#2-3)

1航空图基本知识 航空图的种类与用途 航空图的投影 航空图的分幅、编号与图名

2航空图的识别 地物的表示与识别 地貌的表示与识别 **航空要素的表示与识别**

3航空图的一般应用 点位确定 标绘航线 量算航线角 量算航线距离

**地理信息**

**定义：**

1,地理信息：地理信息可以包括地形、地貌、水文、气象、城市规划、道路网络等数据，通常以地图、地理信息系统（GIS）的形式存在。

2，地理信息是有关地理实体和地理现象的性质、特征和运动状态的表征和一切有用的知识，它是对表达地理特征与地理现象之间关系的地理数据的解释[1]

3, 地理数据是各种地理特征和现象间关系的数字化表示。地理特征和现象的数据描述包括空间位置﹑属性特征(简称属性)及时域特征三部分[1]。

4,GIS中的数据来源和数据类型繁多，概括起来主要有以下几种来源[2]:

⑴地图数据。来源于各种类型的普通地图和专题地图，这些地图的内容丰富，图上实体间的空间关系直观，实体的类别或属性清晰，实测地形图还具有很高的精度，是地理信息的 主要载体，同时也是地理信息系统最重要的信息源。

⑵影像数据。主要来源于卫星遥感和航空遥感，包括多平台、多层面、多种传感器、多时相、多光谱、多角度和多种分辨率的遥感影像数据，构成多源海量数据，也是GIS的最 有效的数据源之一。

⑶地形数据。来源于地形等高线图的数字化，已建立的数字高程模型（DEM）和其他实测的地形数据等。

⑷属性数据。来源于各类调查报告、实测数据、文献资料、解译信息等。

⑸元数据。来源于由各类纯数据通过调查、推理、分析和总结得到的有关数据的数据，例如数据来源、数据权属、数据产生的时间、数据精度、数据分辨率、源数据比例尺、数据 转换方法等。

**数据类型**

矢量数据和栅格数据[6]:

矢量数据在GIS之中主要由几何形状组成，包括点、线和多边形（Points, lines and polygons），优势在于可以较为精确地表达大陆、河流、海洋或是其他类型区域的形状 和轮廓，结构干净没有冗余。矢量数据储存空间小、对计算机的要求较低。

栅格数据结构则是以像素格、也就是栅格为基础的。每个栅格会储存相关的数值，并且 连成一个完整的平面。

栅格数据的最大优势在于叠加，相对应的栅格也可以进行数值的加减乘除

宏观数据类型：

1. 地理数据：包括各种地理特征和现象间的关系，如地理环境要素的数量、质量、分布特征及其规律等，通常以数字、文字、图像等形式表示。
2. 地理信息流：是指地理信息从现实世界到概念世界，再到数字世界（GIS），最后到应用领域的过程。

根据表示对象的不同，空间数据又具体分为七种类型，表示的具体内容如下[2]:

类型数据:例如考古地点、道路线、土壤类型的分布等。

面域数据:例如随机多边形的中心点，行政区域界线、行政单元等。

网络数据:例如道路交点、街道、街区等。

样本数据:例如气象站、航线、野外样方分布区等。

曲面数据:例如高程点、等高线、等值区域等。

文本数据:例如地名、河流名称、区域名称等。

符号数据:例如点状符号、线状符号、面状符号(晕线)等。

所有这些不同类型的数据都可以分为点、线、面三种不同的图形，并可以分别采用x、y平面坐标，地理经纬度a、b，或者格网法表示。

GIS数据来源[5]:

1地图数据

主要表达居民地、交通网、水系、地貌、境界、土质、植被等

2遥感图像

遥感数据用于提取线划数据和生成数字正射影像数据(DOM)、DEM数据等

3测量数据

野外控制测量可使用传统的测量仪器，如经纬仪(theodolite)、水准仪(1evel)和视距仪 (stadia)，测量方位、角度和距离，测量的数据记录在笔记本上，然后输入计算机，用程 序将方位、角度相距离转换成平面坐标和高程，这类程序称为COGO(coordinate geometry)。

4数字资料

各种数据形式的原始资料，包括社会经济数据、人口普查数据、野外调查或监测数据， 例如环境污染监测数据，地质钻井数据，磁力、重力、地震等地球物理数据，气象、水 文观测数据等

5文字报告

在土地资源管理信息系统、灾害监测信息系统、水质信息系统、森林资源管理 信息系统等专题信息系统中，各种文字说明资料，对确定专题内容的属性特征 起着重要的作用

**数据要素**

数据要素：

1. 空间位置数据：描述地理对象所在的位置，包括地理要素的绝对位置（如大地经纬度坐标）和相对位置关系（如空间上的相邻、包含等）。
2. 属性特征数据：描述特定地理要素特征的定性或定量指标，如公路的等级、宽度、起点、终点等。
3. 时域特征数据：记录地理数据采集或地理现象发生的时刻或时段，对于环境模拟分析非常重要。
4. 空间关系数据：描述地理要素间的空间关系，如拓扑关系、方位关系、距离关系等。
5. 地貌特征数据：描述地形的起伏变化、地貌类型等特征。
6. 气象数据：包括气温、降水、气压、风速、风向等气象参数。
7. 水文数据：包括水位、流速、流量等水文参数。
8. 社会经济数据：如人口分布、经济指标、交通情况等社会经济数据的空间分布特征。

这些数据要素构成了地理空间分析的三大基本要素，即空间位置、属性及时域特征[3]，对于地理信息系统具有重要意义。

要素的数据分类[4]:

1.1.点 表示过小而无法表示为线或面的要素。 例如GPS定位点，或者地图中的POI点，每个省的行政驻地等。 图中点代表中国各个市的行政驻地。 ​

1.2.线 表示形状和位置过窄而无法表示为区域的地理对象（如，街道中心线与河流）。 也使用线来表示具有长度但没有面积的要素，如等值线和边界线。 图中表示中国主要河流信息。 ​

1.3.面 一组具有多个边的面要素，表示同类要素类型（如州、县、宗地、土壤类型和土地使用区域）的形状和位置。 图中表示安徽省轮廓面。 ​

1.4.要素之间的拓扑关系 拓扑关系描述的是基本的空间目标点、线、面之间的邻接、关联和包含关系。 如：点与点的邻接关系、点与面的包含关系、线与面的相离关系、面与面的重合关系等。 ​ 拓扑关系指图形在保持连续状态下的变形（缩放、旋转和拉伸等）但图形关系不变的性质[4]。

**相关图片**

**图表样式**

地理信息的图表样式通常包括以下几种：

1. 区域地图：区域地图是展示地理信息最常用的图表之一，它以行政区划或地理区域为单位，展示各区域的地理信息、人口分布、经济状况等数据。区域地图通常采用不同颜色或符号来表示不同类型的数据，例如用深浅不同的颜色表示不同的人口密度或经济发展水平。

2. 散点地图：散点地图多用于展示地理区域内各个点的位置和相关数据。这些点可以是实际的地标点，也可以是根据一定规则生成的虚拟点。散点地图通常在地理坐标系中使用，以展示地理区域内的空间分布和各点的属性信息。

3. 热力地图：热力地图是在地理坐标系中用颜色深浅表示数据大小的一种图表样式。它通过将数据点着色，并按照数据大小调整颜色的深浅，来展示数据的分布情况和相对大小。热力地图通常用于显示人口分布、交通流量、气候变化等空间分布情况。

4. 条形图：条形图是一种以条形为主要表现形式的图表样式，常用于展示分类数据的分布情况。在地理信息的图表样式中，条形图可以用来比较不同地区或不同时间段的分类数据，例如比较不同城市的人口数量、经济发展水平等。

5. 饼图：饼图是一种以圆形为表现形式的图表样式，常用于展示数据的比例关系。在地理信息的图表样式中，饼图可以用来展示不同地区或不同类别的比例关系，例如展示不同城市的人口比例、不同产业的产值比例等。

这些图表样式可以根据实际需要和数据特点进行选择和调整，例如调整颜色、符号、标签等视觉元素，添加图例、标注、网格线等辅助元素，优化图表的交互效果等。同时，不同的图表样式也适用于不同的数据类型和展示需求，需要根据具体情况进行选择和调整。

**参考链接：**

[1] [专题一：GIS基本概念 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/381969743)

1、数据 2、信息 3、地理信息 4、地理数据 5、地理信息的特征 1．信息系统的概念 1、地理信息系统的定义 2、地理信息系统的内涵 3、地理信息系统基本特征 1.1.4地理信息系统外延

[2] [GIS空间数据的来源与类型 开源地理空间基金会中文分会 开放地理空间实验室 (osgeo.cn)](https://www.osgeo.cn/post/7401g#:~:text=%E7%A9%BA%E9%97%B4%E6%95%B0%E6%8D%AE%E6%A0%B9%E6%8D%AE%E8%A1%A8%E7%A4%BA%E5%AF%B9%E8%B1%A1%E7%9A%84%E4%B8%8D%E5%90%8C%EF%BC%8C%E5%8F%88%E5%85%B7%E4%BD%93%E5%88%86%E4%B8%BA%E4%B8%83%E7%A7%8D%E7%B1%BB%E5%9E%8B%EF%BC%8C%E5%AE%83%E4%BB%AC%E5%90%84%E8%A1%A8%E7%A4%BA%E7%9A%84%E5%85%B7%E4%BD%93%E5%86%85%E5%AE%B9%E5%A6%82%E4%B8%8B%3A%201%20%E7%B1%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E8%80%83%E5%8F%A4%E5%9C%B0%E7%82%B9%E3%80%81%E9%81%93%E8%B7%AF%E7%BA%BF%E3%80%81%E5%9C%9F%E5%A3%A4%E7%B1%BB%E5%9E%8B%E7%9A%84%E5%88%86%E5%B8%83%E7%AD%89%E3%80%82%202%20%E9%9D%A2%E5%9F%9F%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E9%9A%8F%E6%9C%BA%E5%A4%9A%E8%BE%B9%E5%BD%A2%E7%9A%84%E4%B8%AD%E5%BF%83%E7%82%B9%EF%BC%8C%E8%A1%8C%E6%94%BF%E5%8C%BA%E5%9F%9F%E7%95%8C%E7%BA%BF%E3%80%81%E8%A1%8C%E6%94%BF%E5%8D%95%E5%85%83%E7%AD%89%E3%80%82%203%20%E7%BD%91%E7%BB%9C%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E9%81%93%E8%B7%AF%E4%BA%A4%E7%82%B9%E3%80%81%E8%A1%97%E9%81%93%E3%80%81%E8%A1%97%E5%8C%BA%E7%AD%89%E3%80%82%204,%E6%A0%B7%E6%9C%AC%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E6%B0%94%E8%B1%A1%E7%AB%99%E3%80%81%E8%88%AA%E7%BA%BF%E3%80%81%E9%87%8E%E5%A4%96%E6%A0%B7%E6%96%B9%E5%88%86%E5%B8%83%E5%8C%BA%E7%AD%89%E3%80%82%205%20%E6%9B%B2%E9%9D%A2%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E9%AB%98%E7%A8%8B%E7%82%B9%E3%80%81%E7%AD%89%E9%AB%98%E7%BA%BF%E3%80%81%E7%AD%89%E5%80%BC%E5%8C%BA%E5%9F%9F%E7%AD%89%E3%80%82%206%20%E6%96%87%E6%9C%AC%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E5%9C%B0%E5%90%8D%E3%80%81%E6%B2%B3%E6%B5%81%E5%90%8D%E7%A7%B0%E3%80%81%E5%8C%BA%E5%9F%9F%E5%90%8D%E7%A7%B0%E7%AD%89%E3%80%82%207%20%E7%AC%A6%E5%8F%B7%E6%95%B0%E6%8D%AE%3A%E4%BE%8B%E5%A6%82%E7%82%B9%E7%8A%B6%E7%AC%A6%E5%8F%B7%E3%80%81%E7%BA%BF%E7%8A%B6%E7%AC%A6%E5%8F%B7%E3%80%81%E9%9D%A2%E7%8A%B6%E7%AC%A6%E5%8F%B7%20%28%E6%99%95%E7%BA%BF%29%E7%AD%89%E3%80%82)

地理数据的来源 空间数据的类型

[3] [地理信息系统教程--第1章 概论 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/603055052#SnippetTab)

1.GIS基本概念 2.GIS的组成 3.GIS的功能 4.地理信息系统的研究内容 5.GIS与相关学科与技术的关系 6.GIS的发展情况

[4] [GIS理论知识（五）之地理要素的概念 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/64856177#:~:text=1.%E8%A6%81%E7%B4%A0%E7%9A%84%E6%95%B0%E6%8D%AE%E5%88%86%E7%B1%BB%201%201.1.%E7%82%B9%20%E8%A1%A8%E7%A4%BA%E8%BF%87%E5%B0%8F%E8%80%8C%E6%97%A0%E6%B3%95%E8%A1%A8%E7%A4%BA%E4%B8%BA%E7%BA%BF%E6%88%96%E9%9D%A2%E7%9A%84%E8%A6%81%E7%B4%A0%E3%80%82%20%E4%BE%8B%E5%A6%82GPS%E5%AE%9A%E4%BD%8D%E7%82%B9%EF%BC%8C%E6%88%96%E8%80%85%E5%9C%B0%E5%9B%BE%E4%B8%AD%E7%9A%84POI%E7%82%B9%EF%BC%8C%E6%AF%8F%E4%B8%AA%E7%9C%81%E7%9A%84%E8%A1%8C%E6%94%BF%E9%A9%BB%E5%9C%B0%E7%AD%89%E3%80%82%20%E5%9B%BE%E4%B8%AD%E7%82%B9%E4%BB%A3%E8%A1%A8%E4%B8%AD%E5%9B%BD%E5%90%84%E4%B8%AA%E5%B8%82%E7%9A%84%E8%A1%8C%E6%94%BF%E9%A9%BB%E5%9C%B0%E3%80%82%20%E2%80%8B%202,%E2%80%8B%204%201.4.%E8%A6%81%E7%B4%A0%E4%B9%8B%E9%97%B4%E7%9A%84%E6%8B%93%E6%89%91%E5%85%B3%E7%B3%BB%20%E6%8B%93%E6%89%91%E5%85%B3%E7%B3%BB%E6%8F%8F%E8%BF%B0%E7%9A%84%E6%98%AF%E5%9F%BA%E6%9C%AC%E7%9A%84%E7%A9%BA%E9%97%B4%E7%9B%AE%E6%A0%87%E7%82%B9%E3%80%81%E7%BA%BF%E3%80%81%E9%9D%A2%E4%B9%8B%E9%97%B4%E7%9A%84%E9%82%BB%E6%8E%A5%E3%80%81%E5%85%B3%E8%81%94%E5%92%8C%E5%8C%85%E5%90%AB%E5%85%B3%E7%B3%BB%E3%80%82%20%E5%A6%82%EF%BC%9A%E7%82%B9%E4%B8%8E%E7%82%B9%E7%9A%84%E9%82%BB%E6%8E%A5%E5%85%B3%E7%B3%BB%E3%80%81%E7%82%B9%E4%B8%8E%E9%9D%A2%E7%9A%84%E5%8C%85%E5%90%AB%E5%85%B3%E7%B3%BB%E3%80%81%E7%BA%BF%E4%B8%8E%E9%9D%A2%E7%9A%84%E7%9B%B8%E7%A6%BB%E5%85%B3%E7%B3%BB%E3%80%81%E9%9D%A2%E4%B8%8E%E9%9D%A2%E7%9A%84%E9%87%8D%E5%90%88%E5%85%B3%E7%B3%BB%E7%AD%89%E3%80%82%20%E2%80%8B%20%E6%8B%93%E6%89%91%E5%85%B3%E7%B3%BB%E6%8C%87%E5%9B%BE%E5%BD%A2%E5%9C%A8%E4%BF%9D%E6%8C%81%E8%BF%9E%E7%BB%AD%E7%8A%B6%E6%80%81%E4%B8%8B%E7%9A%84%E5%8F%98%E5%BD%A2%EF%BC%88%E7%BC%A9%E6%94%BE%E3%80%81%E6%97%8B%E8%BD%AC%E5%92%8C%E6%8B%89%E4%BC%B8%E7%AD%89%EF%BC%89%E4%BD%86%E5%9B%BE%E5%BD%A2%E5%85%B3%E7%B3%BB%E4%B8%8D%E5%8F%98%E7%9A%84%E6%80%A7%E8%B4%A8%E3%80%82%20)

1.要素的数据分类 1.4.要素之间的拓扑关系 2.要素的构成

[5] [4.1. GIS数据来源 — 地理信息系统原理在线教程——GIS在线教程 (osgeo.cn)](https://www.osgeo.cn/gis-tutorial/ch04/04_1.html)

[6] [【GIS数据】地理信息中常用的数据分类 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/27802871)

**图表样式**

**定义：**

图表样式是指图表在视觉上的表现形式和风格，

包括图表的布局、颜色、字体、标签、提示信息等元素的设置。图表样式的作用是使图表更加清晰、易读和美观，同时也可以帮助读者更好地理解和分析数据。

不同的图表类型和数据特点需要采用不同的图表样式，例如柱状图通常采用水平排列的矩形条形来表示数据，折线图则用点来表示数据的变化趋势等。

图表样式可以根据个人喜好和实际需要进行调整和定制，例如修改颜色、字体、添加图例、调整坐标轴范围等。

**数据类型**

图表样式的数据类型通常包括以下几种：

1. 数值型数据：用于表示数量或测量值，例如销售额、人口数量、温度等。
2. 文本型数据：用于表示文本信息，例如类别、标签、名称等。
3. 日期型数据：用于表示日期或时间信息，例如日期、时间、年月日等。
4. 布尔型数据：用于表示真或假、开或关等二元状态。
5. 缺失数据：表示数据缺失或不完整。

图表样式可以根据所使用的数据类型进行选择和调整，例如折线图通常用于显 示数值型数据的趋势和变化，柱状图通常用于比较不同类别的数值型数据，饼 图通常用于显示各部分在整体中所占的比例等。同时，图表样式还可以根据数 据的特点进行优化和美化，例如添加数据标签、调整颜色和字体、添加图例 等。

**数据要素**

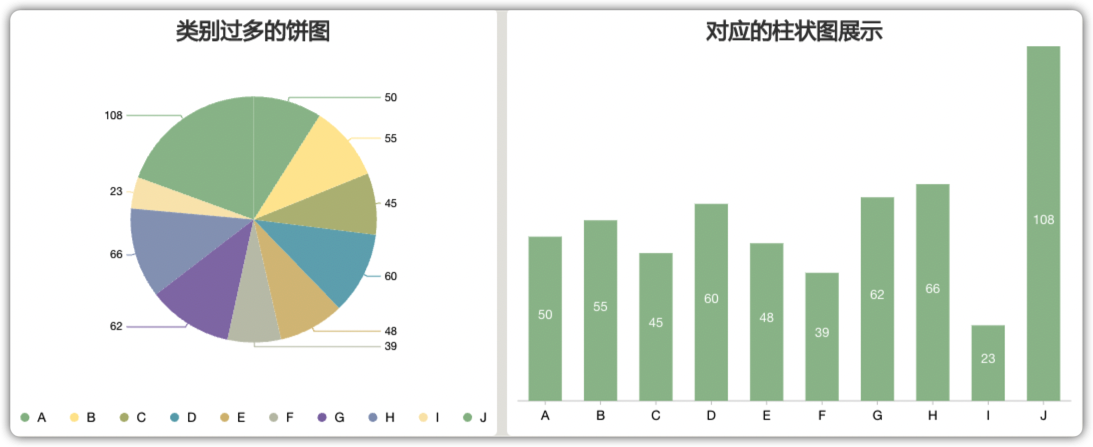
图表样式的数据要素主要包括以下几类：

1. 图表标题：通常位于图表的顶部中央，用于概括图表的主题或说明图表的目的。
2. 坐标轴：包括横坐标轴和纵坐标轴，用于定义数据在平面上的位置和映射关系。
3. 数据点：在图表中表示实际数据的点或符号，如折线图中的点、柱状图中的矩形等。
4. 数据标签：通常用于显示数据点的具体值，可以以悬停提示、图例等方式展示。
5. 图例：对图形本身的概括，用来区分图表中不同类别代表的数据含义。
6. 背景层：通常为白色卡片，是整个图表中视觉层级最弱的部分。
7. 提示信息：当鼠标悬停在某个数据点时，以提示信息的形式展示该点的数据详情，鼠标离开时，则提示信息消失。

这些数据要素可以根据实际需要和数据特点进行个性化设置和调整，例如调整颜色、字体、大小等视觉属性，添加网格线、标记等辅助元素，优化图表的交互效果等。

**相关图片**

1、倾向于精确的表达时，比起饼图更建议使用柱状图[1]



**参考链接：**

[1] [图表样式解析 - DataEase 文档](https://dataease.io/docs/chart_style_analysis/)

## 任务总结：

1、形成全球的国家、省州份、市和区县以及山川河流等特殊地方的geom的点线面（POINT类型、LINESTRING和MULTIPOLYGON等类型）的区域、线和点数据，并能在QGIS中成像。并按照九个原则，规范字段为uuid、name、type、parentUUID（全球区域数据表worldGeo对应的国家uuid）、parentName（全球区域数据表worldGeo对应的国家name）、ffname、geom、country、name\_en、nickname（一些海峡channel数据具有别名）。最后用SQLite或者Spatialite数据库中的表保存起来，并能在QGIS中成像。

2、完成了编写了以下三个mbtiles文件格式规范代码程序的任务,即：从文件夹->实现创建->直接创建（地址）输出.db数据库,从左上角.db数据库变为左下角.db数据库,传入非标准的数据库但表是符合标准的,变为mbtiles文件。最后在geoserver上发布mbtiles文件进行验证，解决发布时候图片混乱的问题。

3、完成了整理资料的任务，即将以下数据：知识图谱、航空要素、地理信息、图表样式的数据类型、数据要素和相关图片整理，并形成文档，要求相关的都要包含，有图表和参考文献，形成了万字的整理文档，和Xmind框架图。

这些项目让我展示了自己与团队成员和顾问保持良好工作关系的能力。我还发现了团队内部的独立动机和动力对项目成功的重要性。期间我们持续每周一次或多次的在会议室等进行项目进度的总结，和项目难度的攻克，摊开重点和难点使得项目计划得到制定和解决。