明星关系项目说明文档

深藏blue：白亚茹、丁奕可、陈翊炀、周涛

**一、数据库设计**

**1.图数据库**

首先，我们将每个明星作为一个节点，节点的标签为‘明星’。节点上包含明星的属性信息，其中包括姓名、出生日期、职业、代表作品。其次，用节点之间的边表示明星之间的不同关系，如夫妻、父子、母子、情侣等等。

通过图数据库的查询功能，可以方便地进行明星关系的分析和查询。例如，可以通过查询明星节点的邻居节点来找到与其有关系的其他明星。

**2.关系型数据库**

在关系型数据库中，我们使用关系表来表示明星和关系之间的结构。

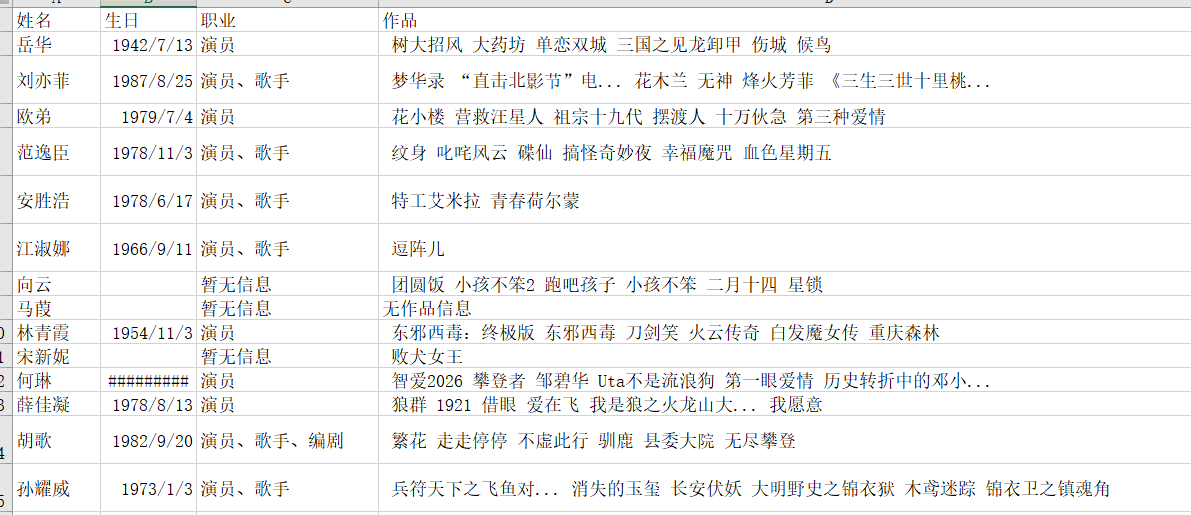
首先，创建一个明星表，其中包含明星的所有属性信息，包括姓名、出生日期、职业、代表作品。每个明星实体可以被表示为表中的一行数据。

其次，需要创建多个关系表，用于存储明星之间的关系信息，如夫妻关系表、母子关系表、合作关系表等等。每张关系表中包含两列，分别为构成该关系的两位明星的姓名。以下是所有关系模式。

1. Stars：（姓名（char），职业（char），生日（Date））
2. 绯闻：（明星1（char），明星2（char））
3. 夫妻：（明星1（char），明星2（char））
4. 父女：（明星1（char），明星2（char））
5. 父子：（明星1（char），明星2（char））
6. 干父女：（明星1（char），明星2（char））
7. 干父子：（明星1（char），明星2（char））
8. 干母子：（明星1（char），明星2（char））
9. 干兄弟：（明星1（char），明星2（char））
10. 干兄妹：（明星1（char），明星2（char））
11. 姑侄：（明星1（char），明星2（char））
12. 好友：（明星1（char），明星2（char））
13. 姐弟：（明星1（char），明星2（char））
14. 姐妹：（明星1（char），明星2（char））
15. 旧爱：（明星1（char），明星2（char））
16. 密友：（明星1（char），明星2（char））
17. 母女：（明星1（char），明星2（char））
18. 母子：（明星1（char），明星2（char））
19. 拍档：（明星1（char），明星2（char））
20. 前任夫妻：（明星1（char），明星2（char））
21. 情侣：（明星1（char），明星2（char））
22. 师徒：（明星1（char），明星2（char））
23. 叔侄：（明星1（char），明星2（char））
24. 同学：（明星1（char），明星2（char））
25. 翁媳：（明星1（char），明星2（char））
26. 兄弟：（明星1（char），明星2（char））
27. 兄妹：（明星1（char），明星2（char））
28. 姻亲：（明星1（char），明星2（char））
29. 祖孙：（明星1（char），明星2（char））

通过关系型数据库的查询功能，可以进行明星关系的分析和查询。例如，可以通过查询明星表和关系表之间的关联来找到与某个明星有关系的其他明星，或者通过查询关系表的条件来找到具有特定关系类型的明星关系。

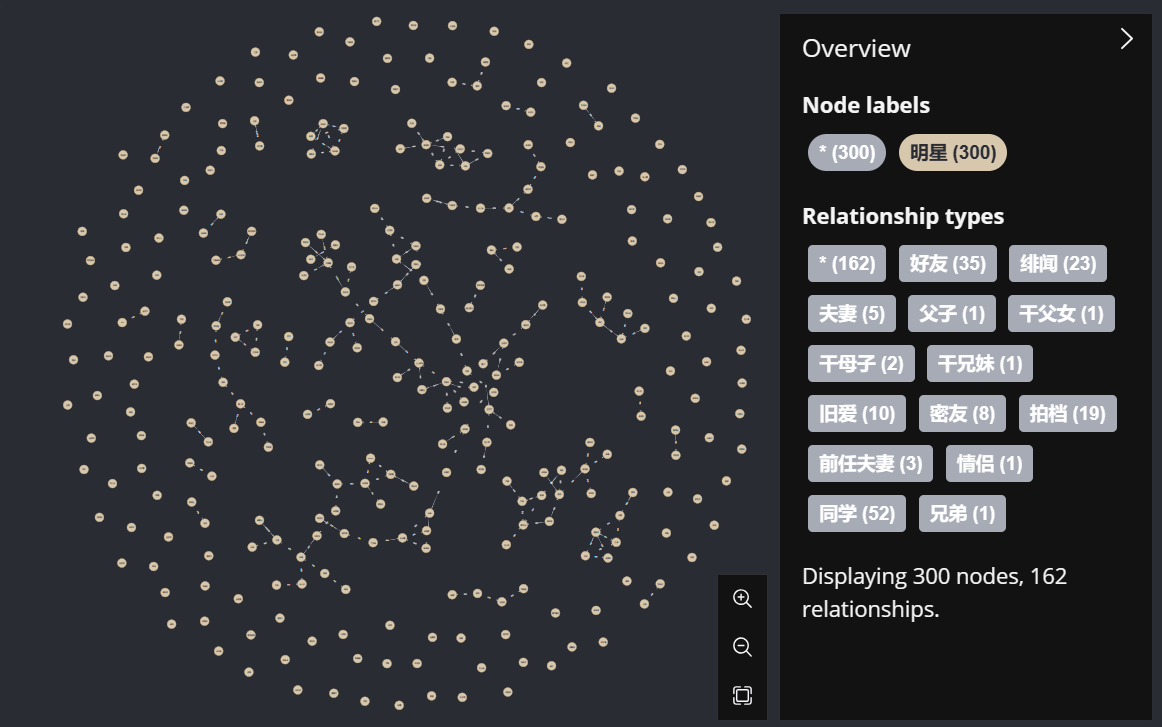
**3.爬取到的部分数据展示**



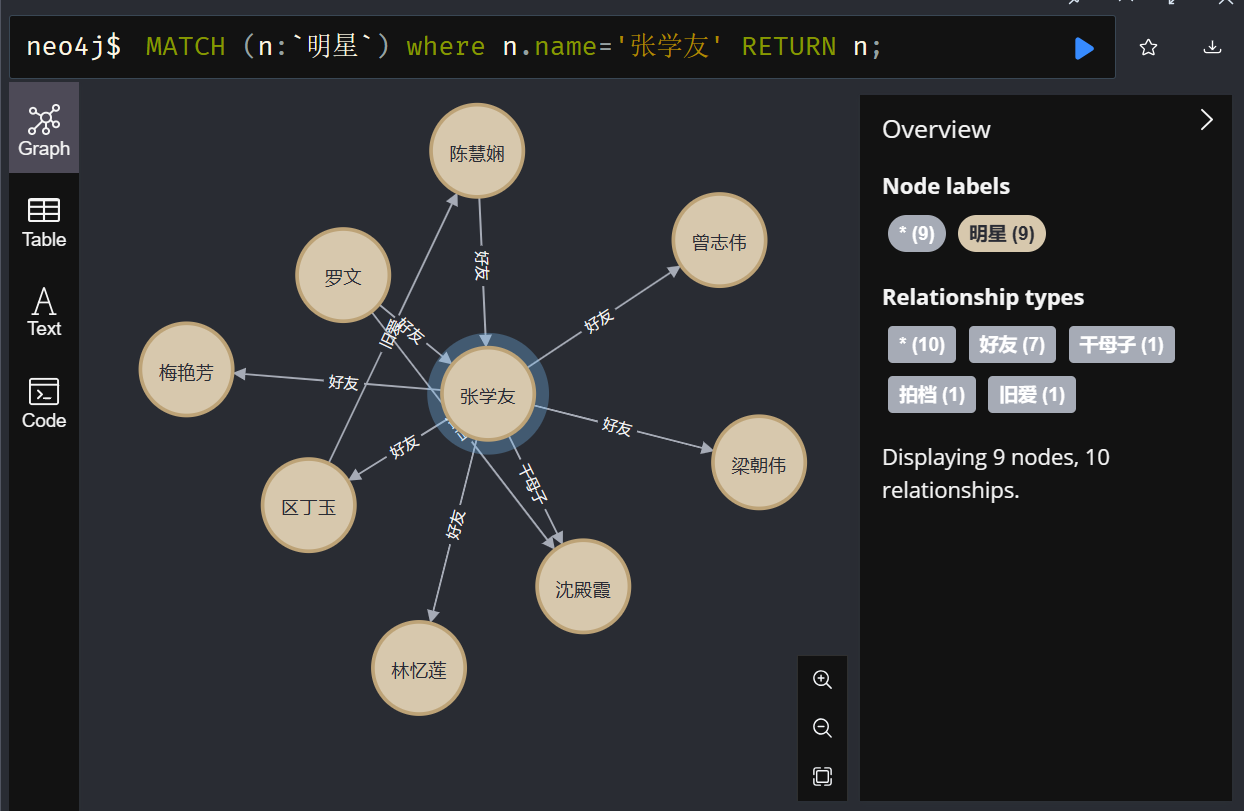


**二、neo4j落地实现**

**1.数据导入后部分节点与关系的展示：**

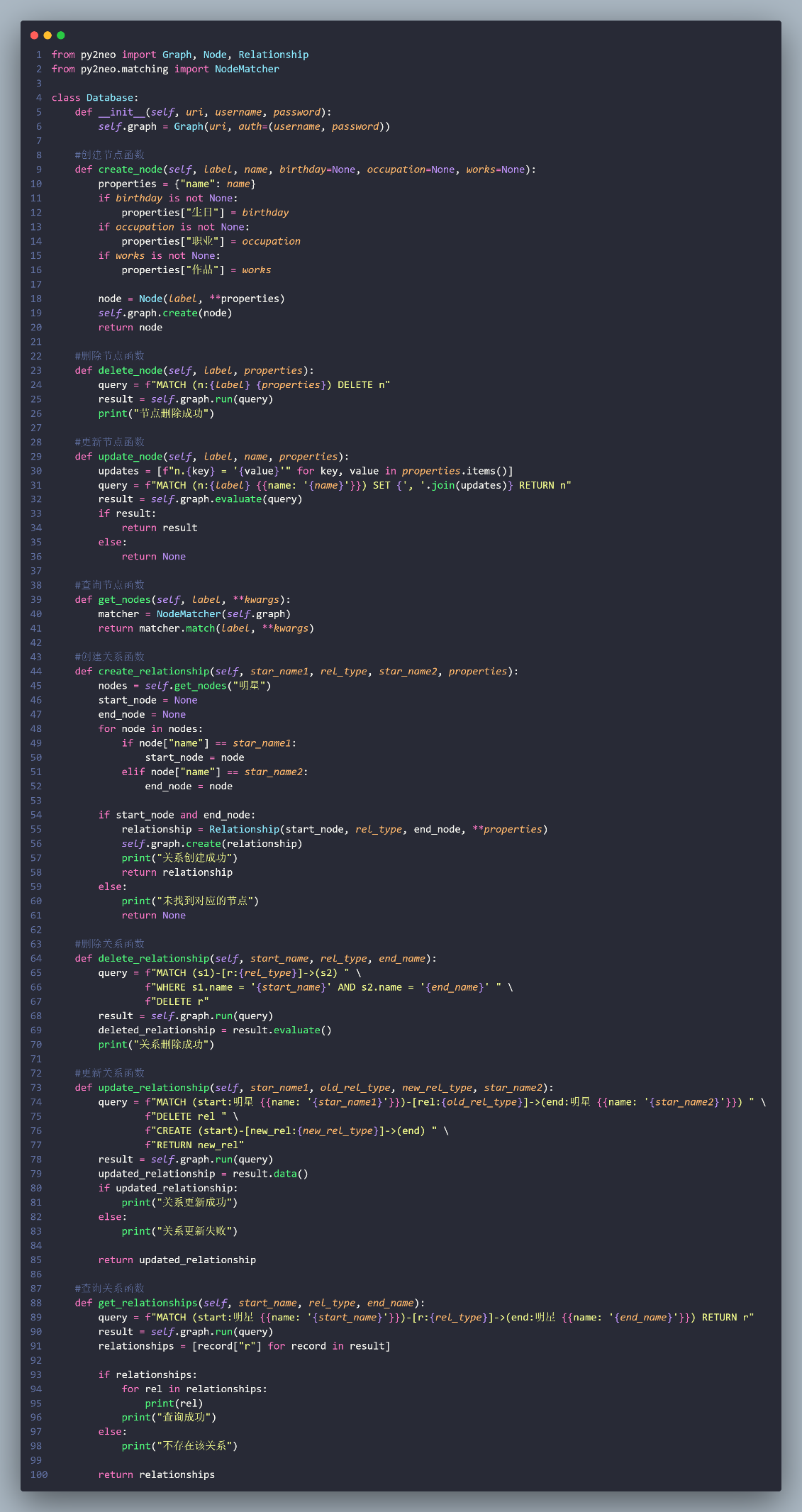


**2.具体节点详细信息和关系表示示例：**



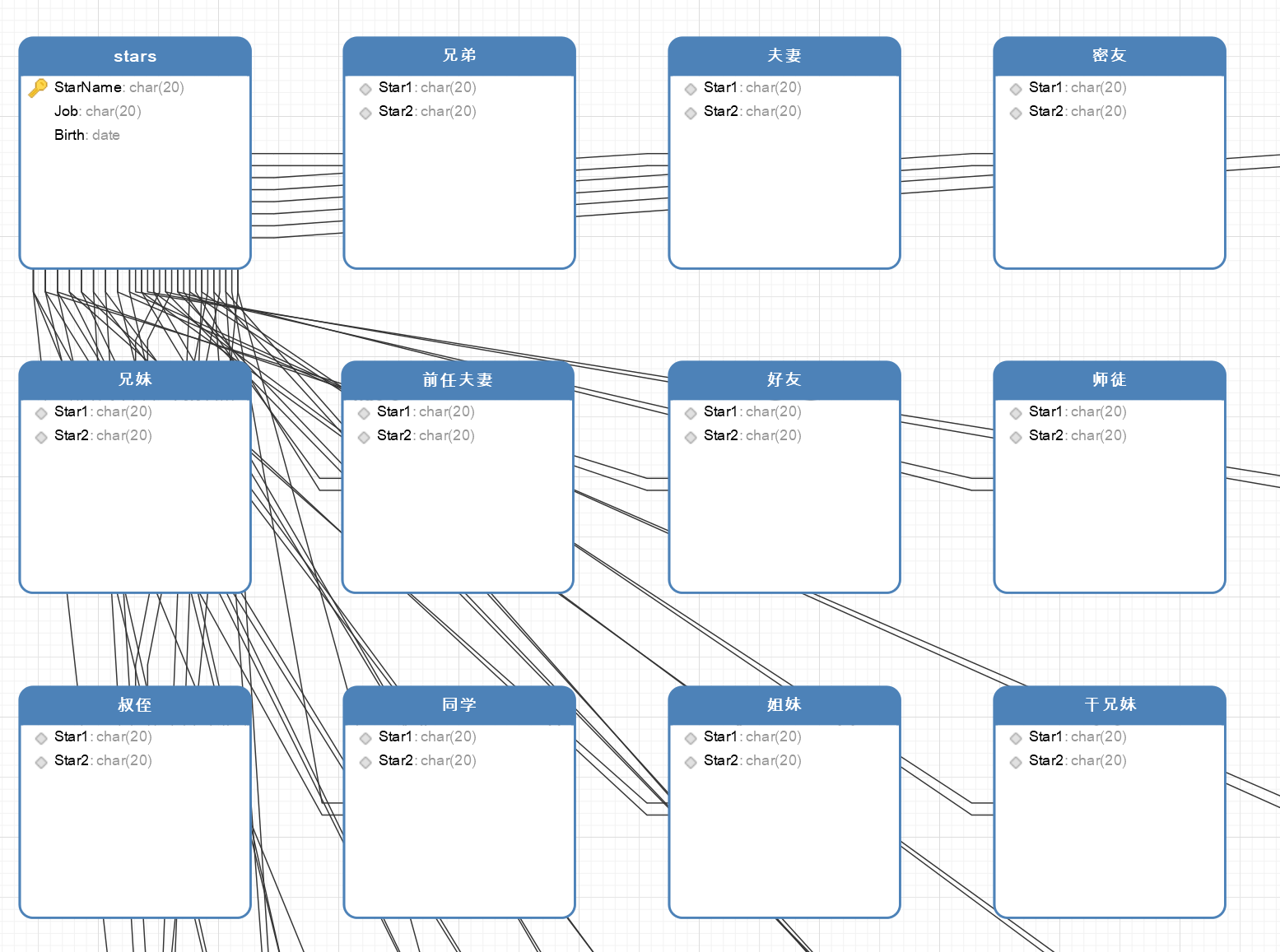
**3.CRUD的函数实现以及封装：**

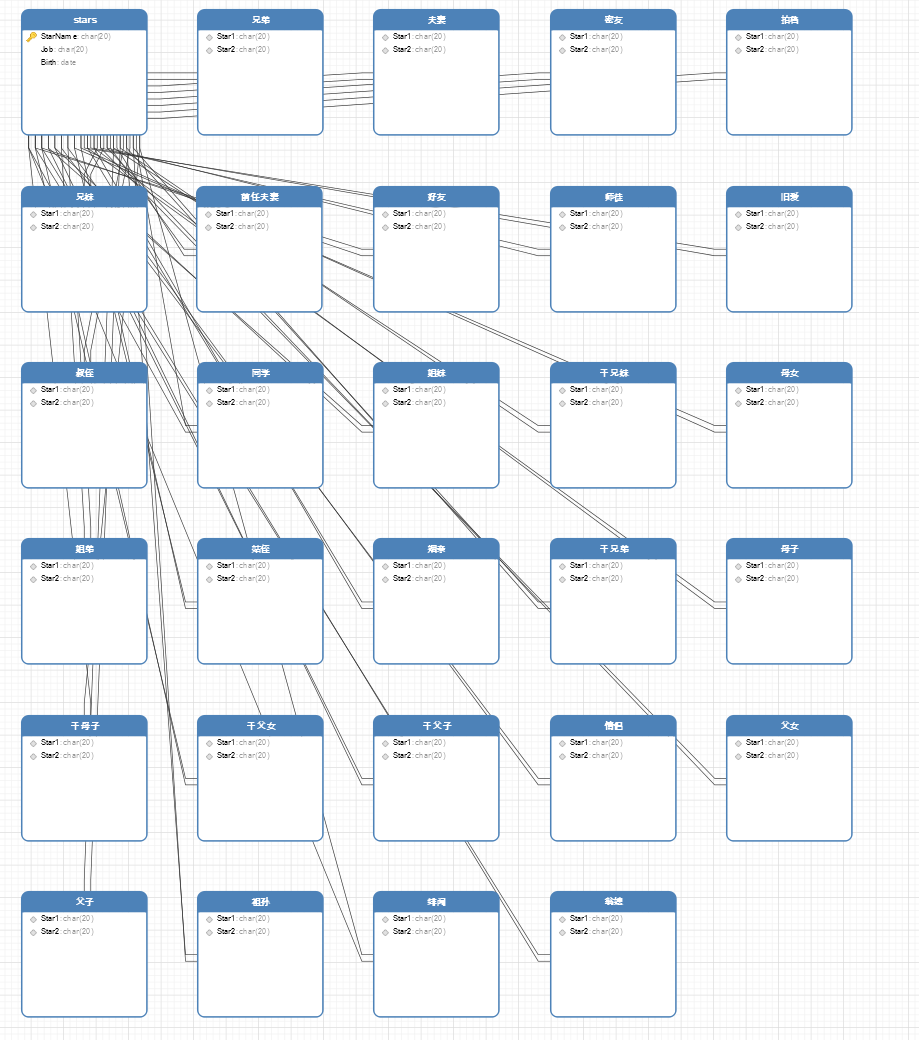
使用py2neo库函数



**三、mySQL落地实现**

**1.各表er图关系：**

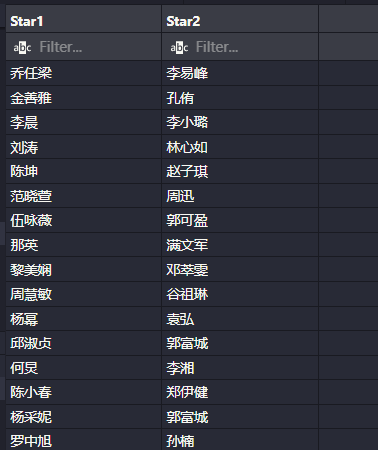
****

****

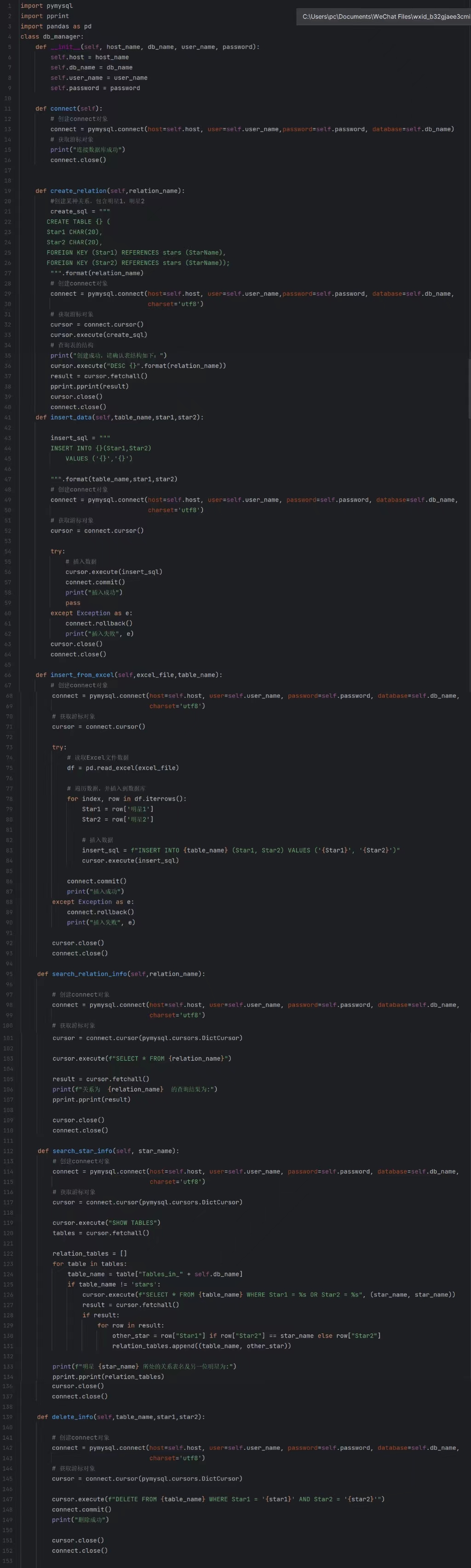
**2.明星信息表部分内容示例：**

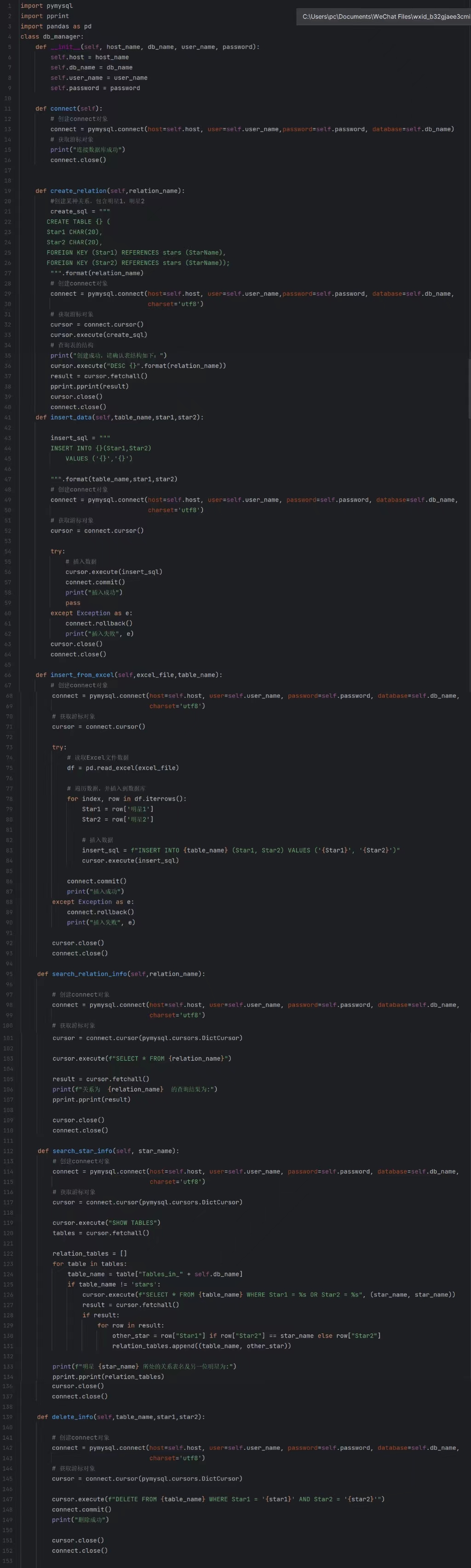
****

**3.好友关系表部分内容示例：**



1. **CRUD函数实现与封装：**

****

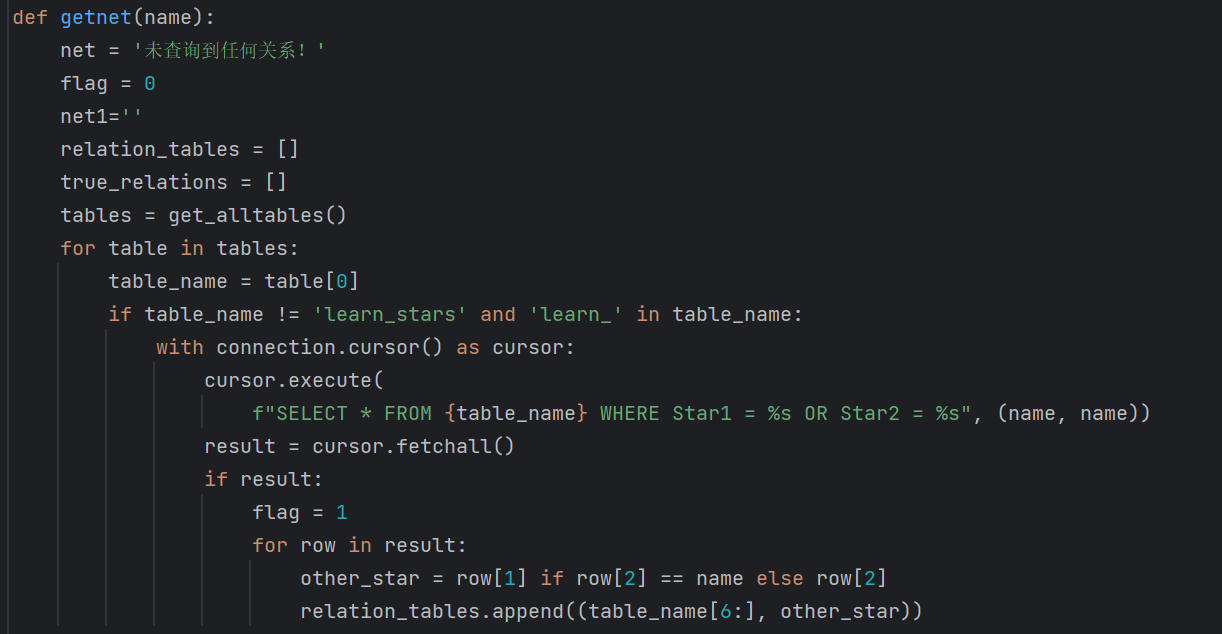
****

**四、扩展功能实现**

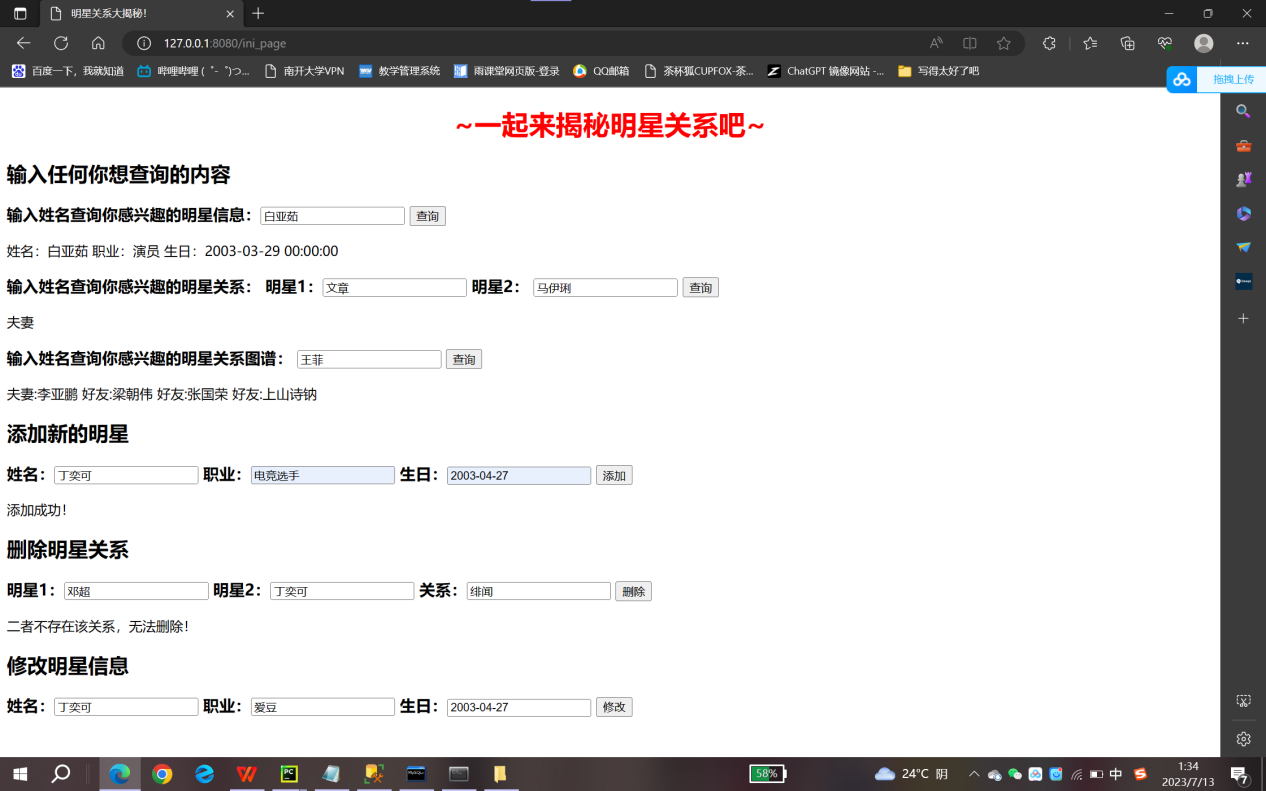
**1.复杂关系查询：**

**查询两个明星之间的关系：利用了connection和cursor工具，获取数据库中的所有表名，之后根据传入的明星姓名去查询是否处于该关系表，若是则将信息保存下来。**

**代码细节：**

****

1. **将部分功能（增删改查）封装到web项目中，展示如图所示：**

****

**具体实现细节：**

1. **创建django项目并完成了数据库star配置，创建了包括stars和28个关系类在内的共29个类。**
2. **建立了多个视图函数，作用如下：**

**getinfo(name)：根据输入姓名查询明星相关信息**

**getrelation(name1,name2)：根据输入的两位明星姓名查询其间关系**

**getnet(name)：根据输入姓名查询该明星关系网**

**insertNew(newname,job,birth)：根据输入明星信息进行插入**

**deleteRelation(name1,name2,relation)：根据输入进行相关删除**

**updateInfo(name,job,birth)：更新该明星相关信息**

**ini\_page(request,method=[‘GET’,’POST’])：主页面视图函数，根据POST请求中参数的不同对不同功能函数进行调用，从而实现信息的呈现和修改。**

**get\_alltables()：辅助函数，用于获取所连数据库中所有表名。**