武汉大学计算机学院

本科生实验报告

数据库系统实验

专 业 名 称 ：计算机科学与技术

课 程 名 称 ：数据库系统

指 导 教 师 ：安杨 副教授

学 生 学 号 ：

学 生 姓 名 ：

二〇二四年五月

一、 目录

二、 实验目的

本实验旨在设计并实现一个数据库系统，通过所提供的上市公司信息，高管信息，和学校信息等表格来提取我国上市公司高管的简历信息，建立主要大学的校友数据库。该系统用于分析和展示校友在各上市公司中的分布情况，支持校友查询、统计和比较分析。

三、 实验内容

本次实验的内容主要包括通过给定的上市公司信息，高管信息，和学校信息等表格来得出他们之间所蕴含的关系。并根据这个关系分析需求，设计 ER 模型和关系模型。

分析完本次实验涉及的关系模型后使用使用数据库管理系统（本此实验中使用的是 MySQL 系统）创建数据库并实现模型。即使用 MySQL 系统建立表格设置主键外键等约束信息，在使用 python 的 pandas 库将所给的数据导入到建立的表格中。

随后需要进行编写关系代数和 SQL 语句实现各种查询功能。再对于进行数据库范式分析和优化，确保数据完整性和一致性。最后对数据库进行备份，确保数据安全。

四、 数据的实体关联以及 E-R 模型设计

首先对于大学的表中，检查了序号属性是否有重复的项：（此处的 Universities 表是用来检查数据的，没有添加主键和外键等约束，并非后文中的大学表）

```
mysql> USE work
Database changed
mysql> SELECT 序号, COUNT(*)
    -> FROM Universities
    -> GROUP BY 序号
    -> HAVING COUNT(*) > 1;
Empty set (0.00 sec)
```

因此本实验中选择大学的序号作为大学表格的主键，大学表格的属性和键情况如图所示：

```
mysql> DESC Universities;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| 序号     | int          | NO   | PRI | NULL    |       |
| 学校名称 | varchar(100) | NO   |     | NULL    |       |
| 省份     | varchar(100) | NO   |     | NULL    |       |
| 城市     | varchar(100) | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

其次对于上市公司表中，检查了 SECUCODE 属性是否有重复的项：（此处的 Companies 表也是用来检查数据的，没有添加主键和外键等约束，并非后文中的上市公司表）

```
mysql> SELECT SECUCODE, COUNT(*)
    -> FROM Companies
    -> GROUP BY SECUCODE
    -> HAVING COUNT(*) > 1;
Empty set (0.06 sec)
```

因此可以选择 SECUCODE 属性作为上市公司表的主键，其表中的键和属性情况如图所示：

```
mysql> DESC Companies;
+--------------------+---------------+------+-----+---------+-------+
| Field              | Type          | Null | Key | Default | Extra |
+--------------------+---------------+------+-----+---------+-------+
| SECUCODE           | varchar(120)  | NO   | PRI | NULL    |       |
| SECURITY_CODE      | varchar(120)  | NO   |     | NULL    |       |
| SECURITY_NAME_ABBR | varchar(150)  | NO   |     | NULL    |       |
| ORG_CODE           | varchar(120)  | NO   |     | NULL    |       |
| ORG_NAME           | varchar(200)  | NO   |     | NULL    |       |
| ORG_NAME_EN        | varchar(200)  | YES  |     | NULL    |       |
| FORMERNAME         | varchar(200)  | YES  |     | NULL    |       |
| STR_CODEA          | varchar(120)  | YES  |     | NULL    |       |
| STR_NAMEA          | varchar(150)  | YES  |     | NULL    |       |
| STR_CODEB          | varchar(120)  | YES  |     | NULL    |       |
| STR_NAMEB          | varchar(150)  | YES  |     | NULL    |       |
| STR_CODEH          | varchar(120)  | YES  |     | NULL    |       |
| STR_NAMEH          | varchar(150)  | YES  |     | NULL    |       |
| SECURITY_TYPE      | varchar(150)  | YES  |     | NULL    |       |
| EM2016             | varchar(150)  | YES  |     | NULL    |       |
| TRADE_MARKET       | varchar(150)  | YES  |     | NULL    |       |
| INDUSTRYCSRC1      | varchar(150)  | YES  |     | NULL    |       |
| PRESIDENT          | varchar(200)  | YES  |     | NULL    |       |
| LEGAL_PERSON       | varchar(150)  | YES  |     | NULL    |       |
| SECRETARY          | varchar(150)  | YES  |     | NULL    |       |
| CHAIRMAN           | varchar(150)  | YES  |     | NULL    |       |
| SECPRESENT         | varchar(150)  | YES  |     | NULL    |       |
| INDEDIRECTORS      | varchar(200)  | YES  |     | NULL    |       |
| ORG_TEL            | varchar(200)  | YES  |     | NULL    |       |
| ORG_EMAIL          | varchar(200)  | YES  |     | NULL    |       |
| ORG_FAX            | varchar(150)  | YES  |     | NULL    |       |
| ORG_WEB            | varchar(200)  | YES  |     | NULL    |       |
| ADDRESS            | varchar(355)  | YES  |     | NULL    |       |
| REG_ADDRESS        | varchar(200)  | YES  |     | NULL    |       |
| PROVINCE           | varchar(150)  | YES  |     | NULL    |       |
| ADDRESS_POSTCODE   | varchar(150)  | YES  |     | NULL    |       |
| REG_CAPITAL        | decimal(18,2) | YES  |     | NULL    |       |
| REG_NUM            | varchar(150)  | YES  |     | NULL    |       |
| EMP_NUM            | int           | YES  |     | NULL    |       |
| TATOLNUMBER        | varchar(120)  | YES  |     | NULL    |       |
| LAW_FIRM           | varchar(200)  | YES  |     | NULL    |       |
| ACCOUNTFIRM_NAME   | varchar(200)  | YES  |     | NULL    |       |
| ORG_PROFILE        | text          | YES  |     | NULL    |       |
| BUSINESS_SCOPE     | text          | YES  |     | NULL    |       |
| TRADE_MARKETT      | varchar(150)  | YES  |     | NULL    |       |
| EMP_NUM            | int           | YES  |     | NULL    |       |
| TATOLNUMBER        | varchar(120)  | YES  |     | NULL    |       |
| LAW_FIRM           | varchar(200)  | YES  |     | NULL    |       |
| ACCOUNTFIRM_NAME   | varchar(200)  | YES  |     | NULL    |       |
| ORG_PROFILE        | text          | YES  |     | NULL    |       |
| BUSINESS_SCOPE     | text          | YES  |     | NULL    |       |
| TRADE_MARKETT      | varchar(150)  | YES  |     | NULL    |       |
| TRADE_MARKET_CODE  | varchar(120)  | YES  |     | NULL    |       |
| SECURITY_TYPEE     | varchar(150)  | YES  |     | NULL    |       |
| SECURITY_TYPE_CODE | varchar(120)  | YES  |     | NULL    |       |
| EXPAND_NAME_ABBR   | varchar(150)  | YES  |     | NULL    |       |
| EXPAND_NAME_PINYIN | varchar(200)  | YES  |     | NULL    |       |
+--------------------+---------------+------+-----+---------+-------+
45 rows in set (0.00 sec)
```

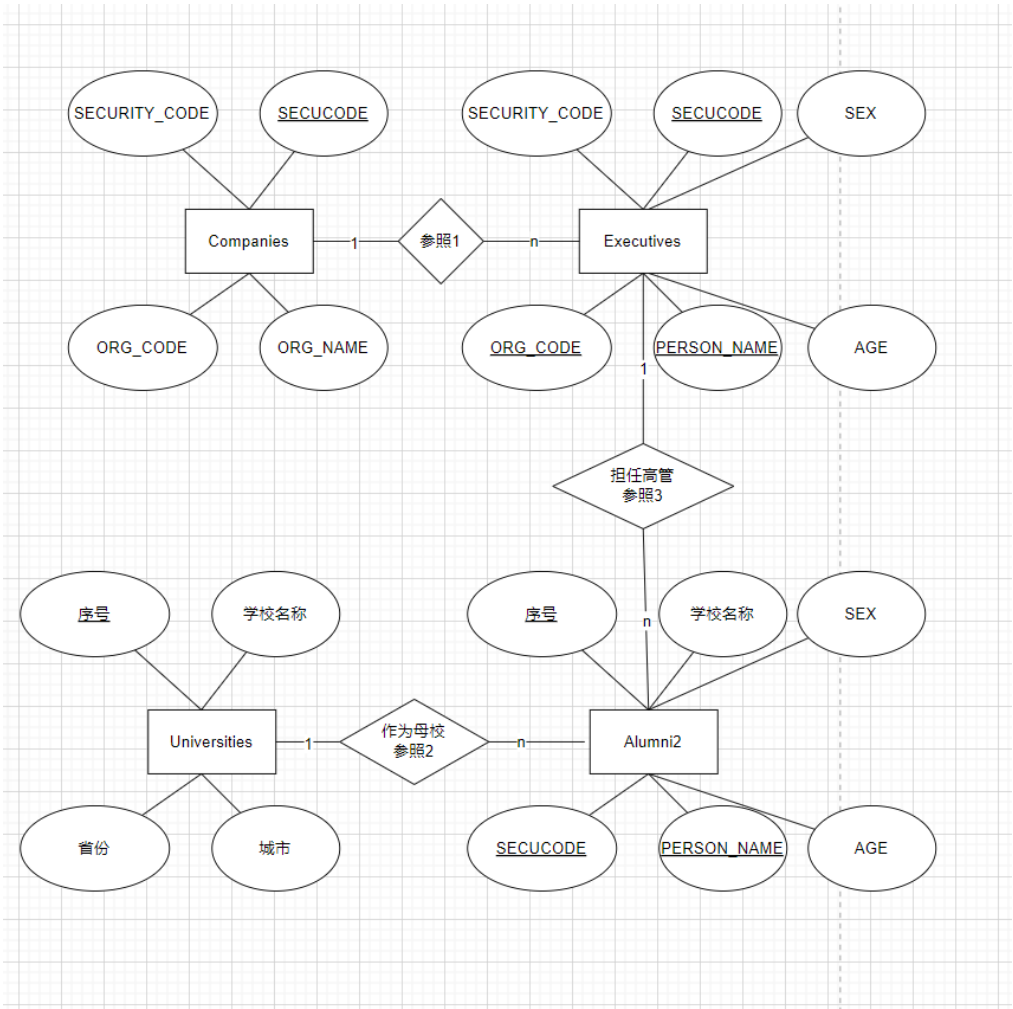
对于高管表中的主键选择，由于很多属性都无法唯一决定高管表中的表项，因此高管表中设置了联合主键（SECUCODE,ORG_CODE,PERSON_NAME）先对于其是否能够唯一决定任意一个表项进行检查：

```
mysql> SELECT SECUCODE, ORG_CODE,PERSON_NAME,COUNT(*)
    -> FROM Executives
    -> GROUP BY SECUCODE,ORG_CODE,PERSON_NAME
    -> HAVING COUNT(*)>1;
Empty set (0.04 sec)
```

由上述检查结果可知可以将该联合主键设置为高管列表的主键，故高管表中的键和属性情况如图所示：

```
mysql> DESC Executives;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| SECUCODE       | varchar(120) | NO   | PRI | NULL    |       |
| SECURITY_CODE  | varchar(120) | YES  |     | NULL    |       |
| ORG_CODE       | varchar(120) | NO   | PRI | NULL    |       |
| PERSON_NAME    | varchar(200) | NO   | PRI | NULL    |       |
| POSITION       | varchar(150) | YES  |     | NULL    |       |
| SEX            | char(1)      | YES  |     | NULL    |       |
| HIGH_DEGREE    | varchar(150) | YES  |     | NULL    |       |
| AGE            | int          | YES  |     | NULL    |       |
| RESUME         | text         | YES  |     | NULL    |       |
| INCUMBENT_TIME | varchar(355) | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

以上就是给出的数据集中相应的关系模型的设计，其 E-R 图如下所示：



E-R 图中已经加入了后续需要假如的学校-高管关联表。由于 E-R 图的篇幅有限，所有的表中都只列举了每张表中的重要主键和属性，包括外键。

五、 实验环境介绍

5.1 数据库管理系统

本实验中使用的是 MySQL 数据库管理系统，由瑞典公司 MySQL AB 开发，目前属于 Oracle 公司。它以其快速、可靠和易于使用而闻名，广泛应用于 Web 应用程序开发中。MySQL 支持多种操作系统，包括 Windows、Linux 和 macOS，能够处理大量数据并提供强大的查询功能。其基于结构化查询语言（SQL），支持事务处理、存储过程、视图和触发器等高级特性，确保数据的完整性和安全性。此外，MySQL 的灵活性和扩展性使其成为各类应用的理想选择，从小型应用程序到大型数据仓库和分布式系统均能胜任。在本实验中，将使用 MySQL 来设计和实现校友数据库系统，进行数据的存储、管理和查询。

MySQL 版本：8.4.0 MySQL Community Server

5.2 数据导入

本实验中使用的是 PyCharm 软件进行数据导入。PyCharm 是一款由 JetBrains 开发的专业集成开发环境（IDE），广泛用于 Python 编程。其丰富的功能和友好的用户界面，使其成为数据科学家和开发人员的首选工具之一。在数据导入实验环境中，PyCharm 提供了许多有用的特性来提高工作效率和代码质量。

具体来说，使用 pandas 库中的 read_excel()函数来读取 excel 文件中的信息并导入到数据框中，在使用 sqlalchemy 库中的 create_engine() 函数创建一个数据库引擎用于连接 MySQL 数据库。具体导入数据和连接数据库的代码如图所示：

导入高管信息：

```
from sqlalchemy import create_engine
import pandas as pd
df=pd.read_excel(r'C:\Users [redacted] Desktop\数据库系统大作业\所有高管.xlsx')
engine =create_engine('mysql+pymysql [redacted] )
df.to_sql( name: 'executives',con=engine,if_exists='append',index=False)
```

导入大学信息：

```
from sqlalchemy import create_engine
import pandas as pd
df=pd.read_excel(r'C:\Users [redacted] Desktop\数据库系统大作业\全国985大学.xlsx')
engine =create_engine('mysql+pymysql [redacted]
df.to_sql( name: 'Universities',con=engine,if_exists='append',index=False)
```

导入上市公司：

```
main.py
1  from sqlalchemy import create_engine
2  import pandas as pd
3  df=pd.read_excel(r'C:\Users [redacted] Desktop\数据库系统大作业\上市公司列表.xlsx')
4  engine =create_engine('mysql+pymy [redacted] )
5  df.to_sql( name: 'Companies',con=engine,if_exists='append',index=False)
```

使用 PyCharm 版本：2023.3 Community；Python 解释器版本：Python3.11

六、 关系代数实现

本实验中要求完成五个关系代数的实现：

6.1 查询员工人数超过 500 的所有上市公司的公司代码、公司名称、注册地址：

首先根据给定的数据集可以知道这些属性和数据在上市公司表中都有体现，因此具体的关系代数实现如下：

$$\pi\ \text{SECUCODE},\ \text{ORG\_NAME},\ \text{REG\_ADDRESS}\ (\sigma\ \text{EMP\_NUM}\ >\ 500\ (\text{Companies}))$$

6.2 查询“星光农机”公司的所有高管的基本信息：

首先需要找到名为星光农机的公司，因此需要再上市公司表中找，其次又要找该公司的所有高管的基本信息，因此需要将高管表和上市公司表做连接后再进行查询，具体的关系代数如下：

$$\pi\ \text{SECUCODE},\ \text{PERSON\_NAME},\ \text{POSITION},\ \text{SEX},\ \text{HIGH\_DEGREE},\ \text{AGE},\ \text{RESUME},\ \text{INCUMBENT\_TIME}\ (\pi\ \text{SECUCODE}\ (\sigma\ \text{ORG\_NAME}\ =\ \text{'星光农机'}\ (\text{Companies}))\ \bowtie\ \text{Executives})$$

6.3 查询注册地址在湖北和湖南的所有上市公司

需要找注册地在湖北和湖南的上市公司，因此只需要在上市公司列表中找注册地省份属性是湖南和湖北的即可，具体的关系代数如下：

$$\pi\ \text{SECUCODE},\ \text{ORG\_NAME},\ \text{REG\_ADDRESS}\ (\sigma\ \text{PROVINCE}\ =\ \text{'湖北'}\ \vee\ \text{PROVINCE}\ =\ \text{'湖南'}\ (\text{Companies}))$$

6.4 查询至少在 2 家以上上市公司任职“独立董事”的高管姓名。（不考虑重名）：

本问中要求至少两家以上公司任职独立董事，因此只需要在高管表中查询，从 Executives 表中选择任职“独立董事”的记录，投影出高管姓名和公司代码，然后按高管姓名分组计算每个高管任职公司的数量，最后选择任职公司数量大于等于 2 的高管姓名。具体的关系代数如下：

$$\pi\ \text{PERSON\_NAME}\left(\begin{array}{c}\gamma\ \text{PERSON\_NAME};\ \text{COUNT}(\text{SECUCODE})\ \text{AS CompanyCount}\\ \left(\begin{array}{c}\pi\ \text{PERSON\_NAME},\ \text{SECUCODE}\\ (\sigma\ \text{POSITION}\ =\ \text{'独立董事'}\ (\text{Executives}))\end{array}\right)\ \bowtie\ \sigma\ \text{CompanyCount}\ \geqslant\ 2\end{array}\right)$$

6.5 查询至少和“何德军”所在所有公司的相同的所有高管姓名。

本问中需要的是何德军所在公司相同的高管，因此肯定要先找到何德军所在公司，在进行除法。首先选择“何德军”所在公司的记录，投影出这些公司的公司代码，然后选择在这些公司任职的所有高管，最后进行除法操作，以确定至少在“何德军”所在所有公司任职的高管姓名。具体的关系代数如下：

$$\pi\ \text{PERSON\_NAME}\ (\text{Executives}) \div\ \pi\ \text{SECUCODE}\ (\sigma\ \text{PERSON\_NAME}\ =\ \text{'何德军'}\ (\text{Executives}))$$

七、SQL 语句实现

在本次实验中共需要实现 17 个 sql 语句的查询与实现操作，因此本部分将逐句分析：

7.1 查询员工人数超过 500 的所有上市公司的公司代码、公司名称、注册地址

本问中需要从 Companies 表中筛选出所有员工人数超过 500 的公司，并返回这些公司的公司代码（SECUCODE）、公司名称（ORG_NAME）和注册地址（REG_ADDRESS）。在 MySQL 中的操作和得到的部分结果如下图所示，具体的查询完整结果见结果的 excel 文件。

```
mysql> SELECT SECUCODE, ORG_NAME, REG_ADDRESS
    -> FROM Companies
    -> WHERE EMP_NUM > 500;
| SECUCODE  | ORG_NAME                 | REG_ADDRESS
| 000001.SZ | 平安银行股份有限公司     | 中华人民共和国广东省深圳市罗湖区深南东路5047号
| 000002.SZ | 万科企业股份有限公司     | 中国深圳市盐田区大梅沙环梅路33号万科中心
| 000008.SZ | 神州高铁技术股份有限公司 | 北京市海淀区高梁桥斜街59号院1号楼16层1606
```

```
| 000055.SZ | 方大集团股份有限公司 | 中华人民共和国深圳市高新技术产业园南区科技南十二路方大科技大厦
| 000056.SZ | 深圳市皇庭国际企业股份有限公司 | 深圳市福田区福田街道岗厦社区福华路350号岗厦皇庭大厦28A01单元
| 000058.SZ | 深圳赛格股份有限公司 | 深圳市福田区华强北路群星广场A座31楼
| 000059.SZ | 北方华锦化学工业股份有限公司 | 辽宁省盘锦市双台子区红旗大街
| 000060.SZ | 深圳市中金岭南有色金属股份有限公司 | 深圳市罗湖区清水河街道清水河社区清水河一路112号深业进元大厦塔楼2座303C
| 000061.SZ | 深圳市农产品集团股份有限公司 | 深圳市光明区马田街道根竹园社区公明南环大道1688号海吉星农产品光明物流园2栋8层
| 000062.SZ | 深圳华强实业股份有限公司 | 深圳市福田区华强北路华强广场A座5楼
| 000063.SZ | 中兴通讯股份有限公司 | 中国广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦
| 000065.SZ | 北方国际合作股份有限公司 | 北京市丰台区南四环西路188号12区47号楼3层(301、302)
| 000066.SZ | 中国长城科技集团股份有限公司 | 中国深圳南山区科技工业园长城计算机大厦
| 000068.SZ | 深圳华控赛格股份有限公司 | 深圳市大工业区兰竹大道以北CH3主厂房
| 000069.SZ | 深圳华侨城股份有限公司 | 深圳市南山区华侨城指挥部大楼103、105、107、111、112室
```

7.2 查询“星光农机”公司的所有高管的基本信息

本问中需要从 Executives 表和 Companies 表中筛选出所有在公司名称为“星光农机”的公司任职的高管，并返回这些高管的姓名（PERSON_NAME）、职位（POSITION）、年龄（AGE）和性别（SEX）。在 MySQL 中的操作和查询结果（查询结果为空，即不存在星光农机这个公司的信息）如图：

```
mysql> SELECT e.PERSON_NAME, e.POSITION, e.AGE, e.SEX
    -> FROM Executives e
    -> JOIN Companies c ON e.SECUCODE = c.SECUCODE
    -> WHERE c.ORG_NAME = '星光农机';
Empty set (0.01 sec)
```

7.3 查询注册地址在湖北和湖南的所有上市公司

本问中需要从 Companies 表中筛选出注册地址中包含“湖北”或“湖南”的公司，并返回这些公司的公司代码（SECUCODE）、证券代码（SECURITY_CODE）、公司名称（ORG_NAME）和注册地址（REG_ADDRESS）。具体的 MySQL 操作和查询结果如图，完整的查询结果见查询结果 excel 表中。

```
mysql> SELECT SECUCODE, SECURITY_CODE, ORG_NAME, REG_ADDRESS
    -> FROM Companies
    -> WHERE REG_ADDRESS LIKE '%湖北%' OR REG_ADDRESS LIKE '%湖南%';
+-----------+---------------+-----------------------------------------+-------------------------------
-------------------------------------------------+
| SECUCODE  | SECURITY_CODE | ORG_NAME                                | REG_ADDRESS
                                                 |
+-----------+---------------+-----------------------------------------+-------------------------------
-------------------------------------------------+
| 000157.SZ | 157           | 中联重科股份有限公司                    | 湖南省长沙市银盆南路361号
                                                 |
| 000419.SZ | 419           | 长沙通程控股股份有限公司                | 中国湖南长沙市劳动西路589号
```

```
| 688799.SH | 688799        | 湖南华纳大药厂股份有限公司              | 湖南浏阳生物医药园区
                                                 |
| 832978.BJ | 832978        | 湖北开特汽车电子电器系统股份有限公司 | 湖北省武汉市武昌区长江路36附25号3楼
                                                 |
| 833874.BJ | 833874        | 十堰市泰祥实业股份有限公司              | 湖北省十堰市经济开发区吉林路258号
                                                 |
| 836942.BJ | 836942        | 武汉恒立工程钻具股份有限公司            | 湖北省武汉市东湖新技术开发区财富二路5号
                                                 |
| 838030.BJ | 838030        | 湖南德众汽车销售服务股份有限公司        | 湖南省怀化市鹤城区怀化工业园鹤城分园鸭嘴岩物
区                                               |
| 838670.BJ | 838670        | 恒进感应科技(十堰)股份有限公司         | 湖北省十堰市普林工业园普林一路6号
                                                 |
| 839273.BJ | 839273        | 湖北一致魔芋生物科技股份有限公司        | 湖北省宜昌市长阳经济开发区长阳大道438号
                                                 |
| 839946.BJ | 839946        | 湖北华阳汽车变速系统股份有限公司        | 湖北省十堰市郧县城关镇大桥南路2号
                                                 |
| 873305.BJ | 873305        | 荆州九菱科技股份有限公司                | 湖北省荆州市沙市区关沮工业园西湖路129号
                                                 |
+-----------+---------------+-----------------------------------------+-------------------------------
-------------------------------------------------+
190 rows in set (0.01 sec)
```

7.4 查询至少在 2 家以上上市公司任职“独立董事”的高管姓名。（不考虑重名）

本问中需要从 Executives 表中筛选出至少在 2 家以上公司担任“独立董事”职位的高管，并返回这些高管的姓名。具体的 MySQL 操作和查询结果如下图，完整的查询结果见 excel 表中：

```
mysql> SELECT PERSON_NAME
    -> FROM Executives
    -> WHERE POSITION LIKE '%独立董事%'
    -> GROUP BY PERSON_NAME
    -> HAVING COUNT(DISTINCT SECUCODE) >= 2;
+-----------------------+
| PERSON_NAME           |
+-----------------------+
| Kirsch Christoph      |
| ZHANG TAO TAO(张涛涛) |
```

```
| 高飞                  |
| 魏强                  |
| 魏琼                  |
| 鲍丽娜                |
| 黄亮                  |
| 黄伟                  |
| 黄峰                  |
| 黄明良                |
| 黄曼行                |
| 黄森                  |
| 黄田阳                |
| 黄继章                |
| 黄静                  |
| 黄颖健                |
+-----------------------+
379 rows in set (0.20 sec)
```

7.5 查询至少和“何德军”所在所有公司的相同的所有高管姓名。

本问中需要从 Executives 表中找到与“何德军”在相同公司任职的所有其他高管，并返回这些高管的姓名。通过连接 Executives 表的两个实例，确保查询结果中包含所有与“何德军”同事的其他高管。查询结果和操作如下：

```
mysql> SELECT DISTINCT e2.PERSON_NAME
    -> FROM Executives e1
    -> JOIN Executives e2 ON e1.SECUCODE = e2.SECUCODE
    -> WHERE e1.PERSON_NAME = '何德军'
    -> AND e2.PERSON_NAME != '何德军';
Empty set (0.01 sec)
```

7.6 列出所有上市公司，统计每个上市公司的高管人数，并按照人数排序倒序排序

本问中需要统计每个公司拥有的高管数量，并按高管数量从多到少排序。通过连接 Companies 表和 Executives 表，将每个公司的高管数量计算出来，并按降序排列，以便找出高管最多的公司。具体的操作和部分查询结果如下：

```
mysql> SELECT c.SECUCODE, c.ORG_NAME, COUNT(e.SECUCODE) AS ExecCount
    -> FROM Companies c
    -> JOIN Executives e ON c.SECUCODE = e.SECUCODE
    -> GROUP BY c.SECUCODE, c.ORG_NAME
    -> ORDER BY ExecCount DESC;
```

```
| 838670.BJ | 恒进感应科技(干堰)股份有限公司                |         1 |
| 838701.BJ | 浙江豪声电子科技股份有限公司                  |         1 |
| 839273.BJ | 湖北一致魔芋生物科技股份有限公司              |         1 |
| 839371.BJ | 苏州欧福蛋业股份有限公司                      |         1 |
| 839680.BJ | 深圳市广道数字技术股份有限公司                |         1 |
| 839729.BJ | 广东永顺生物制药股份有限公司                  |         1 |
| 839792.BJ | 辽宁东和新材料股份有限公司                    |         1 |
| 870204.BJ | 南京沪江复合材料股份有限公司                  |         1 |
| 870436.BJ | 南通大地电气股份有限公司                      |         1 |
| 870976.BJ | 广州视声智能股份有限公司                      |         1 |
| 871478.BJ | 宁夏巨能机器人股份有限公司                    |         1 |
| 871970.BJ | 山西大禹生物工程股份有限公司                  |         1 |
| 872190.BJ | 青岛雷神科技股份有限公司                      |         1 |
| 872351.BJ | 华光源海国际物流集团股份有限公司              |         1 |
| 872374.BJ | 深圳云里物里科技股份有限公司                  |         1 |
| 872541.BJ | 上海铁大电信科技股份有限公司                  |         1 |
| 873152.BJ | 浙江天宏锂电股份有限公司                      |         1 |
| 873223.BJ | 浙江荣亿精密机械股份有限公司                  |         1 |
| 873527.BJ | 浙江夜光明光电科技股份有限公司                |         1 |
| 873570.BJ | 浙江坤博精工科技股份有限公司                  |         1 |
| 873703.BJ | 北京广厦环能科技股份有限公司                  |         1 |
+-----------+-----------------------------------------------+-----------+
4871 rows in set (0.08 sec)
```

7.7 列出所有上市公司，统计每个上市公司不同类型职务（董事长、总经理、法定代表人、董事、副总经理、董事会秘书、独立董事）的高管人数，并按照注册资金倒序排序

本问中需要从 Companies 表和 Executives 表中筛选出每个公司的董事长、总经理、法定代表

人、董事、副总经理、董事会秘书和独立董事的人数，以及公司的注册资本，并按注册资本从高到低排序。通过左连接 Companies 表和 Executives 表，确保每个公司即使没有高管记录也会包含在结果中。需要注意的是为区分董事长和副董事长两个都含有董事长字段的两个不同职位，使用的是两种不同的字符串匹配方式董事长%和%副董事长%的方式来进行区分。具体的查询操作和部分查询结果如下图：

```
mysql> SELECT
    ->     c.SECUCODE,
    ->     c.ORG_NAME as 公司名称,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '董事长%' THEN e.PERSON_NAME ELSE NULL END) AS 董事长人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%总经理%' THEN e.PERSON_NAME ELSE NULL END) AS 总经理人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%法定代表人%' THEN e.PERSON_NAME ELSE NULL END) AS 法定代表人人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%董事%' THEN e.PERSON_NAME ELSE NULL END) AS 董事人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%副总经理%' THEN e.PERSON_NAME ELSE NULL END) AS 副总经理人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%董事会秘书%' THEN e.PERSON_NAME ELSE NULL END) AS 董事会秘书人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%独立董事%' THEN e.PERSON_NAME ELSE NULL END) AS 独立董事人数,
    ->     c.REG_CAPITAL as 注册资本
    -> FROM Companies c
    -> LEFT JOIN Executives e ON c.SECUCODE = e.SECUCODE
    -> GROU
+-----------+-----------------------------------------------+------------+------------+----------------+----------+----
----------+----------------+--------------+-------------+
| 838227.BJ | 杭州美登科技股份有限公司                      |          0 |          0 |              0 |        0 |
        0 |              0 |            0 |     3894.15 |
| 430425.BJ | 成都乐创自动化技术股份有限公司                |          1 |          1 |              1 |       10 |
        0 |              1 |            3 |     3619.21 |
| 301578.SZ | 广东辰奕智能科技股份有限公司                  |          1 |          1 |              1 |        2 |
        1 |              0 |            0 |     3600.00 |
| 688981.SH | 中芯国际集成电路制造有限公司                  |          0 |          0 |              0 |        4 |
        0 |              1 |            0 |     3178.62 |
| 300728.SZ | 江苏天常复合材料有限公司                      |          1 |          1 |              1 |        3 |
        0 |              1 |            0 |     3150.00 |
| 873570.BJ | 浙江坤博精工科技股份有限公司                  |          1 |          0 |              1 |        1 |
        0 |              0 |            0 |     3140.03 |
| 688235.SH | 百济神州有限公司                              |          0 |          0 |              0 |        2 |
        0 |              0 |            0 |       13.54 |
| 688728.SH | 格科微有限公司                                |          1 |          0 |              1 |        4 |
        0 |              1 |            3 |        2.60 |
| 688428.SH | 诺诚健华医药有限公司                          |          0 |          0 |              0 |        5 |
        0 |              0 |            0 |        0.35 |
+-----------+-----------------------------------------------+------------+------------+----------------+----------+----
----------+----------------+--------------+-------------+
5591 rows in set (0.19 sec)
```

7.8 查询所有“武汉大学”校友的高管列表以及其所属公司代码、公司名称

本问中需要从 Executives 表和 Companies 表中筛选出简历中提到“武汉大学”的高管信息，并返回这些高管的公司代码、姓名、组织代码以及公司名称。通过连接 Executives 表和 Companies 表，确保每个高管的公司信息也包含在结果中。具体的查询操作和部分查询结果如图：

```
mysql> SELECT
    ->     e.SECUCODE,
    ->     e.PERSON_NAME,
    ->     e.ORG_CODE,
    ->     c.ORG_NAME AS 公司名称
    -> FROM
    ->     Executives e
    -> JOIN
    ->     Companies c ON e.ORG_CODE = c.ORG_CODE
    -> WHERE
    ->     e.RESUME LIKE '%武汉大学%';
```

7.9 查询所有高管来自于“武汉大学”校友的公司列表

本问中需要从 Executives 表和 Companies 表中筛选出简历中提到“武汉大学”的高管所在的公司，并返回这些公司的组织代码和公司名称。通过连接 Executives 表和 Companies 表，确保查询结果包含所有符合条件的公司的信息，并通过 DISTINCT 关键字去除重复的公司记录。具体的查询操作和部分结果如图：

```
mysql> SELECT DISTINCT
    ->     c.ORG_CODE,
    ->     c.ORG_NAME AS 公司名称
    -> FROM
    ->     Executives e
    -> JOIN
    ->     Companies c ON e.ORG_CODE = c.ORG_CODE
    -> WHERE
    ->     e.RESUME LIKE '%武汉大学%';
+-------------+----------------------------------------+
| ORG_CODE    | 公司名称                               |
+-------------+----------------------------------------+
| 10004086    | 万科企业股份有限公司                   |
| 10000007187 | 聚辰半导体股份有限公司                 |
| 10511851    | 上海皓元医药股份有限公司               |
| 10105765    | 青岛海尔生物医疗股份有限公司           |
| 10176325    | 钜泉光电科技(上海)股份有限公司         |
| 10259248    | 凌云光技术股份有限公司                 |
| 10000016116 | 金科环境股份有限公司                   |
| 10000015845 | 深圳威迈斯新能源股份有限公司           |
| 10000015524 | 江苏浩欧博生物医药股份有限公司         |
| 10627962    | 四方光电股份有限公司                   |
| 10399282    | 武汉市蓝电电子股份有限公司             |
| 10487346    | 宁波球冠电缆股份有限公司               |
| 10506563    | 珠海市派特尔科技股份有限公司           |
| 10000023649 | 荆州九菱科技股份有限公司               |
+-------------+----------------------------------------+
96 rows in set (0.07 sec)
```

7.10 查询每一所学校的校友的高管列表以及其所属公司代码、公司名称

本问中需要从 Universities 表、Executives 表和 Companies 表中筛选出在简历中提到学校名称的高管信息，并返回这些高管的学校名称、高管姓名、公司代码和公司名称。通过连接这三个表，确保查询结果包含符合条件的高管及其公司信息，并按学校名称进行排序。具体的操作和查询结果如下图：

```
mysql> SELECT
    ->     u.学校名称,
    ->     e.PERSON_NAME AS 高管姓名,
    ->     e.ORG_CODE AS 公司代码,
    ->     c.ORG_NAME AS 公司名称
    -> FROM
    ->     Universities u
    -> JOIN
    ->     Executives e ON e.RESUME LIKE CONCAT('%', u.学校名称, '%')
    -> JOIN
    ->     Companies c ON e.ORG_CODE = c.ORG_CODE
    -> ORDER BY
    ->     u.学校名称;
+------------------+--------------------------+-------------+--------------------------------
| 学校名称         | 高管姓名                 | 公司代码    | 公司名称
+------------------+--------------------------+-------------+--------------------------------
| 上海交通大学     | 程娴                     | 10141732    | 江苏中利集团股份有限公司
```

```
| 重庆大学         | 王俭                     | 10475013    | 北京流金岁月传媒科技股份有限公司
| 重庆大学         | 林衍                     | 10002489    | 国家电投集团远达环保股份有限公司
| 重庆大学         | 罗正英                   | 10526165    | 浙江东尼电子股份有限公司
| 重庆大学         | 万鑫铭                   | 10090345    | 中国汽车工程研究院股份有限公司
| 重庆大学         | 廖成林                   | 10002489    | 国家电投集团远达环保股份有限公司
| 重庆大学         | 董雅姝                   | 10000015607 | 上海新炬网络信息技术股份有限公司
| 重庆大学         | 李岩                     | 10064635    | 建设工业集团(云南)股份有限公司
| 重庆大学         | 吉杏丹                   | 10325746    | 金龙羽集团股份有限公司
| 重庆大学         | 王红卫                   | 10002425    | 浙江医药股份有限公司
| 重庆大学         | 黄建军                   | 10002521    | 四川宏达股份有限公司
| 重庆大学         | 刘先成                   | 10000015315 | 深圳普门科技股份有限公司
| 重庆大学         | 马微                     | 10375316    | 中设工程咨询(重庆)股份有限公司
| 重庆大学         | 蒋茜                     | 10002351    | 重庆太极实业(集团)股份有限公司
| 重庆大学         | 李蓉                     | 10008350    | 中信银行股份有限公司
| 重庆大学         | 方伟华                   | 10000016108 | 杭州格林达电子材料股份有限公司
| 重庆大学         | 刘定群                   | 10109188    | 四川安控科技股份有限公司
+------------------+--------------------------+-------------+----------------------------------------+
3019 rows in set (2.96 sec)
```

7.11 基于上述的批量查询，优化查询结果，将查询结果批量插入到“高管-学校的校友关联”的表中。

本问中主要是基于第 10 个查询的结果进行插入的，需要从 Universities 表、Executives 表和 Companies 表中筛选出在简历中提到学校名称的高管信息，并将这些信息插入到 Alumni2 表中。插入的信息包括高管的公司代码、姓名、性别、年龄、学校的序号和学校名称。通过连接这三个表，确保插入到 Alumni2 表的数据是符合条件的高管及其学校信息。插入操作和结果如图所示：

```
mysql> INSERT INTO Alumni2 (SECUCODE, PERSON_NAME, 性别, 年龄, 序号, 学校名称)
    -> SELECT
    ->     e.SECUCODE,
    ->     e.PERSON_NAME,
    ->     e.SEX AS 性别,
    ->     e.AGE AS 年龄,
    ->     u.序号,
    ->     u.学校名称
    -> FROM
    ->     Universities u
    -> JOIN
    ->     Executives e ON e.RESUME LIKE CONCAT('%', u.学校名称, '%')
    -> JOIN
    ->     Companies c ON e.ORG_CODE = c.ORG_CODE;
Query OK, 3019 rows affected (1.84 sec)
Records: 3019  Duplicates: 0  Warnings: 0
```

7.12 基于“高管-学校的校友关联”表，查询武汉大学的所有高管校友列表。

本问中需要从 Alumni2 表和 Executives 表中筛选出毕业于“武汉大学”的高管信息，并返回这些高管的公司代码、姓名、职位和组织代码。通过连接 Alumni2 表和 Executives 表，确保查询结果包含符合条件的高管的详细职位信息。具体的查询操作和部分查询结果如图所示：

```
mysql> SELECT
    ->     a.SECUCODE,
    ->     a.PERSON_NAME,
    ->     e.POSITION,
    ->     e.ORG_CODE
    -> FROM
    ->     Alumni2 a
    -> JOIN
    ->     Executives e ON a.SECUCODE = e.SECUCODE AND a.PERSON_NAME = e.PERSON_NAME
    -> WHERE
    ->     a.学校名称 = '武汉大学';
+-----------+-------------+-----------------------------------------------------------------+-------------+
| SECUCODE  | PERSON_NAME | POSITION                                                        | ORG_CODE    |
+-----------+-------------+-----------------------------------------------------------------+-------------+
| 000002.SZ | 王蕴        | 职工代表董事                                                    | 10004086    |
| 000014.SZ | 王凡        | 董事会秘书                                                      | 10004098    |
| 000021.SZ | 钟彦        | 董事会秘书                                                      | 10004105    |
| 000100.SZ | 梁伟华      | 副董事长,董事                                                   | 10007178    |
| 300332.SZ | 汪芳敏      | 董事,董事会秘书                                                 | 10193245    |
| 300683.SZ | 陈亚        | 董事长,总经理,法定代表人,非独立董事                            | 10024976    |
| 600699.SH | 李俊彧      | 副总裁,非独立董事,财务总监                                     | 10004009    |
| 002548.SZ | 代伊博      | 非独立董事                                                      | 10163072    |
| 688612.SH | 李莹莹      | 副总经理                                                        | 10000015845 |
| 300199.SZ | 张敏        | 副总裁                                                          | 10174604    |
| 002909.SZ | 吴珈宜      | 副总经理,董事会秘书                                             | 10506165    |
| 300380.SZ | 孙丹        | 非职工代表监事                                                  | 10181918    |
| 301187.SZ | 罗刚        | 董事会秘书                                                      | 10000119711 |
| 301211.SZ | 张葵莉      | 副总经理                                                        | 10026233    |
| 300579.SZ | 吴舜皋      | 监事会主席,职工代表监事                                         | 10140413    |
+-----------+-------------+-----------------------------------------------------------------+-------------+
110 rows in set (0.01 sec)
```

7.13 假设可以通过姓名、性别、年龄三项来唯一确定一个不重复的人，基于“高管-学校的校友关联”表，去重查询武汉大学的所有高管校友列表，同时输出该校友任职公司数量。

本问中需要实现去重，因此从 Alumni2 表中筛选出毕业于“武汉大学”的高管，并返回这些高管的姓名、性别、年龄，以及他们任职的不同公司数量。通过 GROUP BY 子句，将结果按高管的姓名、性别和年龄进行分组，并通过 COUNT(DISTINCT a.SECUCODE) 计算每个高管任职的不同公司数量。具体的操作和部分查询结果如图所示：

```
mysql> SELECT
    ->     a.PERSON_NAME,
    ->     a.性别,
    ->     a.年龄,
    ->     COUNT(DISTINCT a.SECUCODE) AS 任职公司数量
    -> FROM
    ->     Alumni2 a
    -> WHERE
    ->     a.学校名称 = '武汉大学'
    -> GROUP BY
    ->     a.PERSON_NAME,
    ->     a.性别,
    ->     a.年龄;
+-------------+------+------+--------------+
| PERSON_NAME | 性别 | 年龄 | 任职公司数量 |
+-------------+------+------+--------------+
| 丁国清      | 男   |   49 |            2 |
| 陈作涛      | 男   |   53 |            2 |
| 陈利        | 女   |   56 |            1 |
| 陈文静      | 女   |   50 |            1 |
| 陈锋        | 男   |   46 |            1 |
| 雷鹤        | 女   |   52 |            1 |
| 韩胜利      | 男   |   44 |            1 |
| 项焱        | 女   |   52 |            1 |
| 高敏        | 女   |   42 |            1 |
| 高洁芬      | 女   |   53 |            1 |
| 魏琼        | 女   |   47 |            1 |
| 黄晓华      | 女   |   49 |            1 |
| 黄晓辉      | 女   |   36 |            1 |
| 黄静        | 女   |   59 |            1 |
| 龚雯雯      | 女   |   43 |            2 |
+-------------+------+------+--------------+
103 rows in set (0.00 sec)
```

7.14 列出所有高校，统计每个高校的高管人数，并根据高管人数进行倒序排序。

本问中需要从 Alumni2 表和 Executives 表中筛选出每个学校的高管人数，并按高管人数从多到少排序。通过连接 Alumni2 表和 Executives 表，确保查询结果包含在公司中担任高管职位的校友信息。通过 GROUP BY 子句，将结果按学校名称进行分组，并通过 COUNT(DISTINCT a.PERSON_NAME) 计算每个学校的高管人数。具体的操作和部分查询结果如图：

```
mysql> SELECT
    ->     a.学校名称,
    ->     COUNT(DISTINCT a.PERSON_NAME) AS 高管人数
    -> FROM
    ->     Alumni2 a
    -> JOIN
    ->     Executives e ON a.SECUCODE = e.SECUCODE
    -> GROUP BY
    ->     a.学校名称
    -> ORDER BY
    ->     高管人数 DESC;
+------------------+----------+
| 学校名称         | 高管人数 |
| 重庆大学         |       37 |
| 东北大学         |       32 |
| 华东师范大学     |       31 |
| 北京师范大学     |       30 |
| 兰州大学         |       25 |
| 西北工业大学     |       23 |
| 大连理工大学     |       22 |
| 中国农业大学     |       20 |
| 中国海洋大学     |       19 |
| 国防科技大学     |       12 |
| 中央民族大学     |        7 |
| 西北农林科技大学 |        1 |
+------------------+----------+
39 rows in set (0.05 sec)
```

7. 15 列出所有高校，统计每个高校的每一类高管人数，并根据总的高管人数进行倒序排序。

本问中需要统计每个高管的数量，因此从 Alumni2 表和 Executives 表中筛选出每个学校的不同职位的高管人数和总高管人数，并按总高管人数从多到少排序。通过连接 Alumni2 表和 Executives 表，确保查询结果包含在公司中担任高管职位的校友信息。通过 GROUP BY 子句，将结果按学校名称进行分组，并通过 COUNT(DISTINCT CASE WHEN ... THEN e.PERSON_NAME ELSE NULL END) 计算每个学校的不同职位的独立高管人数。同样需要注意的是，为区别董事长和副董事长两个均含有董事长字段的职务，使用了董事长%和%副董事长%两种不同的字符匹配方式来匹配职务信息。具体的操作和部分的查询结果如图所示：

```
mysql> SELECT
    ->     a.学校名称,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '董事长%' THEN e.PERSON_NAME ELSE NULL END) AS 董事长人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%总经理%' THEN e.PERSON_NAME ELSE NULL END) AS 总经理人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%法定代表人%' THEN e.PERSON_NAME ELSE NULL END) AS 法定代表人人数
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%副总经理%' THEN e.PERSON_NAME ELSE NULL END) AS 副总经理人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%董事%' AND e.POSITION NOT LIKE '%董事长%' THEN e.PERSON_NAME ELS
NULL END) AS 董事人数,
    ->     COUNT(DISTINCT CASE WHEN e.POSITION LIKE '%独立董事%' THEN e.PERSON_NAME ELSE NULL END) AS 独立董事人数,
    ->     COUNT(DISTINCT e.PERSON_NAME) AS 总高管人数
    -> FROM
    ->     Alumni2 a
    -> JOIN
    ->     Executives e ON a.SECUCODE = e.SECUCODE
    -> GROUP BY
    ->     a.学校名称
    -> ORDER BY
    -> 总高管人数 DESC;
+------------------+------------+------------+----------------+--------------+----------+--------------+------------+
| 学校名称         | 董事长人数 | 总经理人数 | 法定代表人人数 | 副总经理人数 | 董事人数 | 独立董事人数 | 总高管人数 |
+------------------+------------+------------+----------------+--------------+----------+--------------+------------+
| 清华大学         |        258 |        399 |            248 |          239 |      718 |          667 |       1607 |
| 东南大学         |         36 |         47 |             32 |           27 |       92 |           93 |        235 |
| 华东师范大学     |         21 |         30 |             22 |           18 |       95 |           57 |        228 |
| 北京航空航天大学 |         33 |         52 |             33 |           30 |       84 |           82 |        203 |
| 中国科学技术大学 |         31 |         53 |             29 |           35 |       87 |           83 |        193 |
| 重庆大学         |         28 |         34 |             28 |           23 |       84 |           72 |        193 |
| 东北大学         |         29 |         36 |             28 |           19 |       71 |           58 |        178 |
| 北京师范大学     |         22 |         37 |             22 |           24 |       82 |           63 |        168 |
| 兰州大学         |         22 |         31 |             16 |           21 |       39 |           33 |        119 |
| 大连理工大学     |         19 |         24 |             19 |           14 |       49 |           54 |        119 |
| 西北工业大学     |         18 |         27 |             17 |           19 |       46 |           37 |        115 |
| 中国海洋大学     |         14 |         16 |             14 |            9 |       52 |           43 |        112 |
| 中国农业大学     |         17 |         28 |             16 |           17 |       26 |           38 |         94 |
| 国防科技大学     |         13 |         17 |             12 |            9 |       33 |           30 |         85 |
| 中央民族大学     |          7 |          7 |              7 |            4 |        4 |           11 |         22 |
| 西北农林科技大学 |          1 |          1 |              1 |            1 |        2 |            2 |          3 |
+------------------+------------+------------+----------------+--------------+----------+--------------+------------+
39 rows in set (0.10 sec)
```

7.16 创建一个视图，表示武汉大学高管校友。

本问中需要创建一个表示武汉大学高管校友的视图，因此需要创建一个名为 Wuhan_University_Alumni 的视图，这个视图包含了所有简历中提到“武汉大学”的高管信息。视图中包含的信息包括高管的公司代码、姓名、性别、年龄、学校序号和学校名称。通过连接 Executives 表和 Universities 表，确保视图中包含符合条件的高管及其学校信息。具体的创建操作和验证视图存在的方式如图所示：

```
mysql> CREATE VIEW Wuhan_University_Alumni AS
    -> SELECT
    ->     e.SECUCODE,
    ->     e.PERSON_NAME,
    ->     e.SEX AS 性别,
    ->     e.AGE AS 年龄,
    ->     u.序号,
    ->     u.学校名称
    -> FROM
    ->     Executives e
    -> JOIN
    ->     Universities u ON e.RESUME LIKE CONCAT('%', u.学校名称, '%')
    -> WHERE
    ->     u.学校名称 = '武汉大学';
```

```
mysql> SELECT * FROM Wuhan_University_Alumni;
+-----------+-------------+------+------+------+----------+
| SECUCODE  | PERSON_NAME | 性别 | 年龄 | 序号 | 学校名称 |
+-----------+-------------+------+------+------+----------+
| 000002.SZ | 王蕴        | 女   |   48 |   15 | 武汉大学 |
| 000014.SZ | 王凡        | 男   |   51 |   15 | 武汉大学 |
| 000021.SZ | 钟彦        | 女   |   40 |   15 | 武汉大学 |
| 000100.SZ | 李东生      | 男   |   66 |   15 | 武汉大学 |
| 000100.SZ | 梁伟华      | 男   |   42 |   15 | 武汉大学 |
| 000415.SZ | 张灿        | 男   |   42 |   15 | 武汉大学 |
| 000526.SZ | 朱晋丽      | 女   |   51 |   15 | 武汉大学 |
| 000966.SZ | 罗丹        | 男   |   52 |   15 | 武汉大学 |
| 002129.SZ | 李东生      | 男   |   66 |   15 | 武汉大学 |
| 002130.SZ | 易华蓉      | 女   |   43 |   15 | 武汉大学 |
| 002153.SZ | 李仲初      | 男   |   60 |   15 | 武汉大学 |
```

7. 17 创建一个用户，授权所有的查询权限，但没有修改权限。

本问中创建了一个名为 new_user 的用户，密码为 password。@'localhost' 指定该用户只能从本地主机登录。并且给 new_user 用户在所有数据库和所有表上的 SELECT 权限。*.* 表示所有数据库和表。但是没有赋予 INSERT 插入（修改）权限。具体的用户创建操作和权限展示如图所示：

```
mysql> CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT SELECT ON *.* TO 'new_user'@'localhost';
Query OK, 0 rows affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT User, Host FROM mysql.user WHERE User = 'new_user';
+----------+-----------+
| User     | Host      |
+----------+-----------+
| new_user | localhost |
+----------+-----------+
1 row in set (0.00 sec)

mysql> SHOW GRANTS FOR 'new_user'@'localhost';
+-----------------------------------------------+
| Grants for new_user@localhost                 |
+-----------------------------------------------+
| GRANT SELECT ON *.* TO `new_user`@`localhost` |
+-----------------------------------------------+
1 row in set (0.00 sec)

mysql>
```

八、 数据库范式分析

本数据库中一共创建了四张表格，分别是上市公司表（Companies），高管表（Executives），高校列表（Universities）和校友-学校关联表（Alumni2）。因此在这个部分中将会逐一分析四张表的依赖关系来得到数据库整体的范式情况。

8.1 Companies 表

上市公司表的各个属性如图所示：

SECUCODE	varchar(120)	NO	PRI	NULL	
SECURITY_CODE	varchar(120)	NO		NULL	
SECURITY_NAME_ABBR	varchar(150)	NO		NULL	
ORG_CODE	varchar(120)	NO		NULL	
ORG_NAME	varchar(200)	NO		NULL	
ORG_NAME_EN	varchar(200)	YES		NULL	
FORMERNAME	varchar(200)	YES		NULL	
STR_CODEA	varchar(120)	YES		NULL	
STR_NAMEA	varchar(150)	YES		NULL	
STR_CODEB	varchar(120)	YES		NULL	
STR_NAMEB	varchar(150)	YES		NULL	
STR_CODEH	varchar(120)	YES		NULL	
STR_NAMEH	varchar(150)	YES		NULL	
SECURITY_TYPE	varchar(150)	YES		NULL	
EM2016	varchar(150)	YES		NULL	
TRADE_MARKET	varchar(150)	YES		NULL	
INDUSTRYCSRC1	varchar(150)	YES		NULL	
PRESIDENT	varchar(200)	YES		NULL	
LEGAL_PERSON	varchar(150)	YES		NULL	
SECRETARY	varchar(150)	YES		NULL	
CHAIRMAN	varchar(150)	YES		NULL	
SECPRESENT	varchar(150)	YES		NULL	
INDEDIRECTORS	varchar(200)	YES		NULL	
ORG_TEL	varchar(200)	YES		NULL	
ORG_EMAIL	varchar(200)	YES		NULL	
ORG_FAX	varchar(150)	YES		NULL	
ORG_WEB	varchar(200)	YES		NULL	
ADDRESS	varchar(355)	YES		NULL	
REG_ADDRESS	varchar(200)	YES		NULL	
PROVINCE	varchar(150)	YES		NULL	
ADDRESS_POSTCODE	varchar(150)	YES		NULL	
REG_CAPITAL	decimal(18,2)	YES		NULL	
REG_NUM	varchar(150)	YES		NULL	
EMP_NUM	int	YES		NULL	
TATOLNUMBER	varchar(120)	YES		NULL	
LAW_FIRM	varchar(200)	YES		NULL	
ACCOUNTFIRM_NAME	varchar(200)	YES		NULL	
ORG_PROFILE	text	YES		NULL	
BUSINESS_SCOPE	text	YES		NULL	
TRADE_MARKETT	varchar(150)	YES		NULL	
TRADE_MARKET_CODE	varchar(120)	YES		NULL	
SECURITY_TYPEE	varchar(150)	YES		NULL	
SECURITY_TYPE_CODE	varchar(120)	YES		NUL	

可以看出所有的属性都直接依赖于主键 SECUCODE，没有任何部分依赖或传递依赖。因此可以得出 Companies 表中的依赖关系至少满足第三范式 (3NF)。

8.2 Executives 表

高管表中的各个属性和键情况如图所示：

```
mysql> DESC Executives;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| SECUCODE       | varchar(120) | NO   | PRI | NULL    |       |
| SECURITY_CODE  | varchar(120) | YES  |     | NULL    |       |
| ORG_CODE       | varchar(120) | NO   | PRI | NULL    |       |
| PERSON_NAME    | varchar(200) | NO   | PRI | NULL    |       |
| POSITION       | varchar(150) | YES  |     | NULL    |       |
| SEX            | char(1)      | YES  |     | NULL    |       |
| HIGH_DEGREE    | varchar(150) | YES  |     | NULL    |       |
| AGE            | int          | YES  |     | NULL    |       |
| RESUME         | text         | YES  |     | NULL    |       |
| INCUMBENT_TIME | varchar(355) | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

可以得到 POSITION, SEX, AGE, HIGH_DEGREE, RESUME 等字段都直接依赖于(SECUCODE, ORG_CODE, PERSON_NAME)复合主键，没有部分依赖。因此本表中的依赖关系任然至少满足第三范式 (3NF)。

8.3 Universities 表

学校表中的各个属性和主键情况如图所示：

```
mysql> DESC Universities;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| 序号     | int          | NO   | PRI | NULL    |       |
| 学校名称 | varchar(100) | NO   |     | NULL    |       |
| 省份     | varchar(100) | NO   |     | NULL    |       |
| 城市     | varchar(100) | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

可以看到学校表中除了主键序号之外其他的所有属性(学校名称, 省份, 城市) 直接依赖于主键，没有部分依赖或传递依赖。因此大学表中的依赖关系任然至少满足第三范式 (3NF)。

8.4 Alumni2 表

本表中储存的是校友的学校和他们的公司之间的关联信息，各个属性如下图所示：

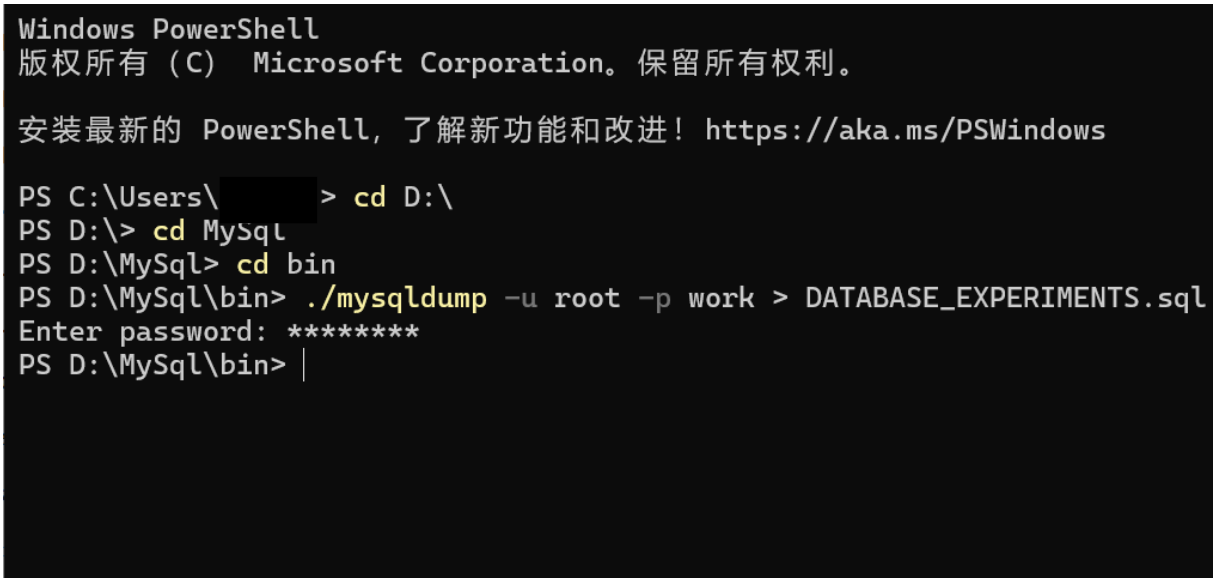
```
mysql> DESC Alumni2;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| SECUCODE    | varchar(120) | NO   | PRI | NULL    |       |
| PERSON_NAME | varchar(200) | NO   | PRI | NULL    |       |
| 性别        | varchar(10)  | YES  |     | NULL    |       |
| 年龄        | int          | YES  |     | NULL    |       |
| 序号        | int          | NO   | PRI | NULL    |       |
| 学校名称    | varchar(100) | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

由上图可知：每个表列都具有单一值（无重复组或数组），Alumni 表的每个字段值都是原子的，满足第一范式。并且非主键列完全依赖于整个主键（不仅仅是主键的一部分）。在 Alumni 表中，所有非主键列（如 学校名称）都依赖于完整的联合主键（SECUCODE, PERSON_NAME, 序号），因此满足第二范式。并且所有非主键列只直接依赖于主键，而不依赖于其他非主键列。在 Alumni 表中，没有任何非主键列依赖于其他非主键列的情况，这意味着它也至少满足第三范式。

由以上分析可以得出：本实验中所涉及的数据库模型满足且至少满足第三范式。

九、 数据库备份与导出

本次实验中，使用的是 MySQL 中自带的 mysqldump 命令进行导出。由于 MySQL 在本机上的安装位置不是默认的 C 盘路径，因此直接命令行的操作无法实现。因此以此进入 MySQL 安装包的位置并且进入 bin 目录，在执行命令后导出的数据库文件被保存在 bin 目录下的 DATABASE_EXPERIMENT.sql 文件中。具体的操作代码如下图所示：



十、 实验总结

本次实验中完成了从原始数据中提取关系模型，设计 E-R 模型和关系模型，到数据库的建立，数据表的建立以及对于关系代数的编写和 sql 查询语句的编写，再到视图的创建，用户的创建以及数据库的备份和导出等一系列的任务。不仅帮助我完成了对于《数据库系统》这门课上所学的有关关系数据库的相关知识和 sql 查询语句的知识的融会贯通；也让我亲自实践了一个数据库从需求分析到模型设计到创建数据库再到进行查询和插入删除等等一系列的操作。

通过本次实验，我加深了对数据库系统设计和实现的理解，掌握了从数据需求分析到数据库创建和优化的完整过程。这些经验将对我未来在数据管理和应用开发方面的工作有很大的帮助。同时，我也认识到，在实际项目中，良好的数据库设计和优化是确保系统性能和数据一致性的关键。

教师评语评分

评语：_____

_____

_____

_____

_____

_____

_____

_____

评分：_____

评阅人：

年　　月　　日

（备注：对该实验报告给予优点和不足的评价，并给出百分之评分。）