Project 1

Yutong Liu N16773429 & SiyunWang N10409344

# Introduction

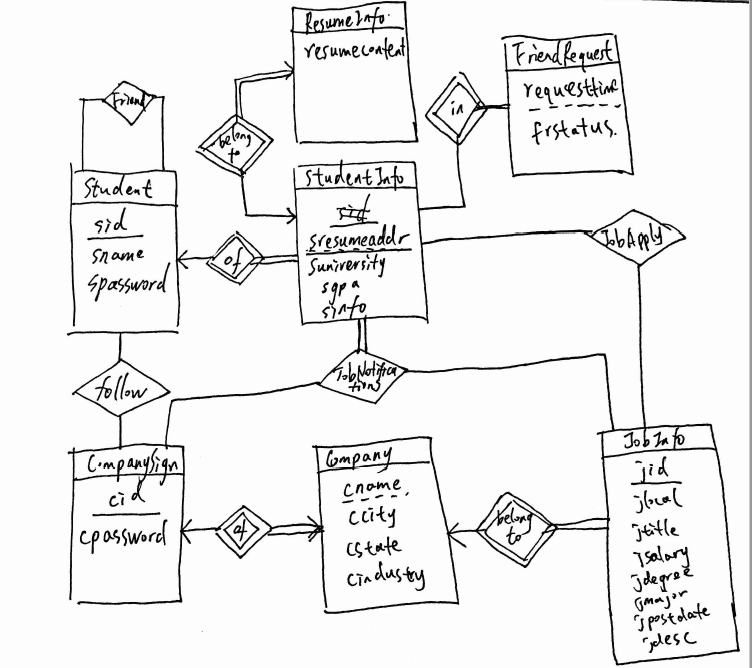
Here, we want to build a database system which is used in the Job Hunting website. There are two user group: Student and Company. The website is a bridge between them which helps student find the great employee and helps company find the great employers. There are some main functions we want to achieve:

1. Sign Up and Sign In for Student and Company. Different Sign Up methods for Different Groups.
2. Allow Student and Company submit their information and allow the information can be searched. For Students, information including their name, university, major, degree, and a resume in a pdf format. For Company, information including their name, address, industry, and Job information they posted.
3. Student can send friend request to the other students. And if the other one agreed, they become friends and can share some job or company information.
4. Student can follow their favorite company and receive the job notifications from their followed company.
5. Student can apply the job they like.
6. Company can post their job announcement to the specific student group even that they are not followed by these students.
7. For the Student and Company, they both can search some information by keywords.

Based on these main functions, we start to design the database system and achieve it in frontend in the second project. The ER diagram is shown as below:

# (a)

**ER diagram**



## (b)

**Student** (sid, sname, spassword)

Sid is login name.

**StudentInfo** (sid, sresumeaddr, suniversity, sdegree, smajor, sgpa, sinfo)

Foreign Key: sid refers Student.sid

**ResumeInfo** (sid, sresumeaddr, resumecontent)

Foreign Key: (sid, sresumeaddr) refers StudentInfo.(sid, sresumeaddr)

**CompanySign** (cid, cpassword)

**Company** (cid, cname, ccity, cstate, cindustry)

Foreign Key: cid refers CompanySign.cid

**JobInfo** (jid, cid, jlocal, jtitle, jsalary, jdegree, jmajor, jpostdate, jdesc)

**Friends** (sid, fid)

Foreign Key: sid, fid refers Students.sid

**Follow** (sid, cid, followStatus)

Foreign Key: sid refers Students.sid; cid refers Companys.cid

**JobApply** (sid, jid, applyStatus)

Foreign Key: sid refers Students.sid; jid refers JobInfo.jid

**JobNotifications** (sid, jid, cid)

Foreign Key: sid refers Students.sid; cid refers Companys.cid; jid refers JobInfo.jid

**FriendRequest** (sid, rid, requesttime, frstatus)

Foreign Key: sid, rid refers Student.sid; Status = (‘Ignored’,’Agreed’,’Rejected’)

**Explanation:**

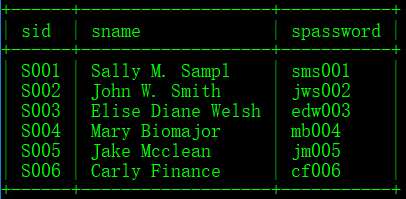
Student is for user log in. sname is the real name and sid is login name, here login name is unique. So sid is primary key for Student. After sign in, student need to fill their information, which stores in StudentInfo table. In the StudentInfo, we only store the resume address not the content. Based on this, we create a new table to store the resume content, called ResumeInfo. CompanySign is used for company sign up. And also, the information of company will be filled in Company Table. JobInfo is a list of job information, including the unique id jid. Friends table represents the relationship between two sid is friends. Follow table will record the status if the student follow the company. JobApply is for recording the apply status and JobNotifications is for store the information which student will be notificated. FriendRequest is stored the action of friend request.

## (c)

In the third question, we write the query and use sample data test them. Below is our SQL query and Result in MySQL.

### 1.

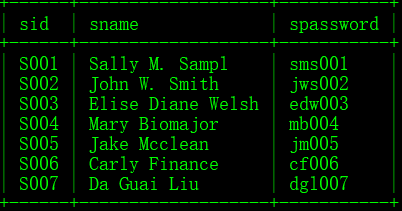
**Original Table**: (sid means loginname)



**SQL Query:**

INSERT INTO `Student` VALUES('S007','Da Guai Liu','dgl007');

**Updated Table:**



### 2.

For example, for the student which loginname = ‘S001’, list the name of all friends of him.

**SQL Query:**

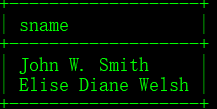
SELECT Student.sname

FROM Friends, Student

WHERE Student.sid = Friends.fid

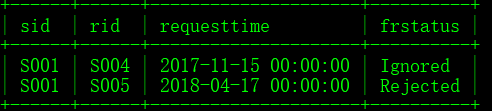
AND Friends.sid = "S001"

**Result:**



### 3.

**Original FriendRequest:**



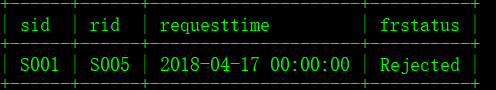
**SQL Query:**

DELETE FROM FriendRequest

WHERE TIMESTAMPDIFF(MONTH, requesttime, CURDATE()) > 1

AND frstatus != 'Agreed'

**After Delete Operation:**



### 4.

**SQL Query:**

SELECT StudentInfo.sid

FROM Follow, StudentInfo, Company

WHERE Follow.sid = StudentInfo.sid AND Follow.cid = Company.cid

AND StudentInfo.suniversity = 'New York University'

AND Company.cname = 'Microsoft'

AND Follow.followstatus = 'Followed'

**Result:**

Output the unique loginname



### 5.

**SQL Query:**

SELECT jid

FROM JobInfo

WHERE TIMESTAMPDIFF(DAY, jpostdate, CURDATE()) >= 7

AND TIMESTAMPDIFF(DAY, jpostdate, CURDATE()) < 14

AND jdegree = 'MS' AND jmajor = 'Computer Science'

**Result:**



### 6.

**First, find the student which satisfy the condition**

**SQL Query:**

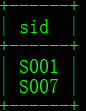
SELECT StudentInfo.sid

FROM StudentInfo, ResumeInfo

WHERE StudentInfo.sid = ResumeInfo.sid AND StudentInfo.sresumeaddr = ResumeInfo.sresumeaddr

AND sgpa > '3.5' AND resumecontent like '%database%' collate utf8\_general\_ci

**Result:**



**Second, create a notification, here for the company C05 and job J009**

**SQL Query in PHP:**

$sqlNew = "SELECT Student.sid

FROM StudentInfo, ResumeInfo, Student

WHERE StudentInfo.sid = ResumeInfo.sid AND StudentInfo.sresumeaddr = ResumeInfo.sresumeaddr AND Student.sid = StudentInfo.sid

AND sgpa > '3.5' AND resumecontent like '%database%' collate utf8\_general\_ci";

$result = $conn->query($sqlNew);

if ($result->num\_rows > 0) {

while($row = $result->fetch\_assoc()){

$sid = $row["sid"];

echo "$sid";

// insert new data to the JobNotifications Table

mysqli\_query($conn,"INSERT INTO JobNotifications (`sid`, `jid`, `cid`) VALUES ('$sid', 'J009','C05')")

or die(mysqli\_error($conn));}}

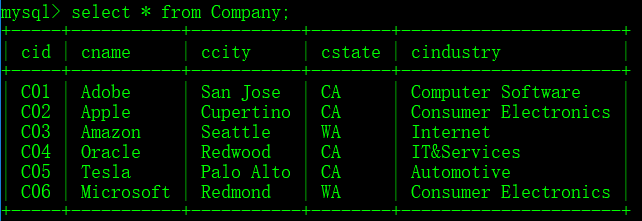
**Result:**



## (d)

**Sample Data We design:**

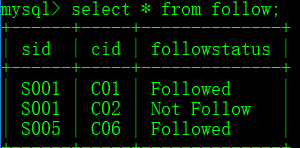
Company:



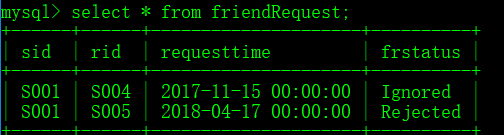
CompanySign:



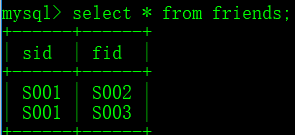
Follow:



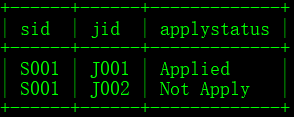
FriendRequest:

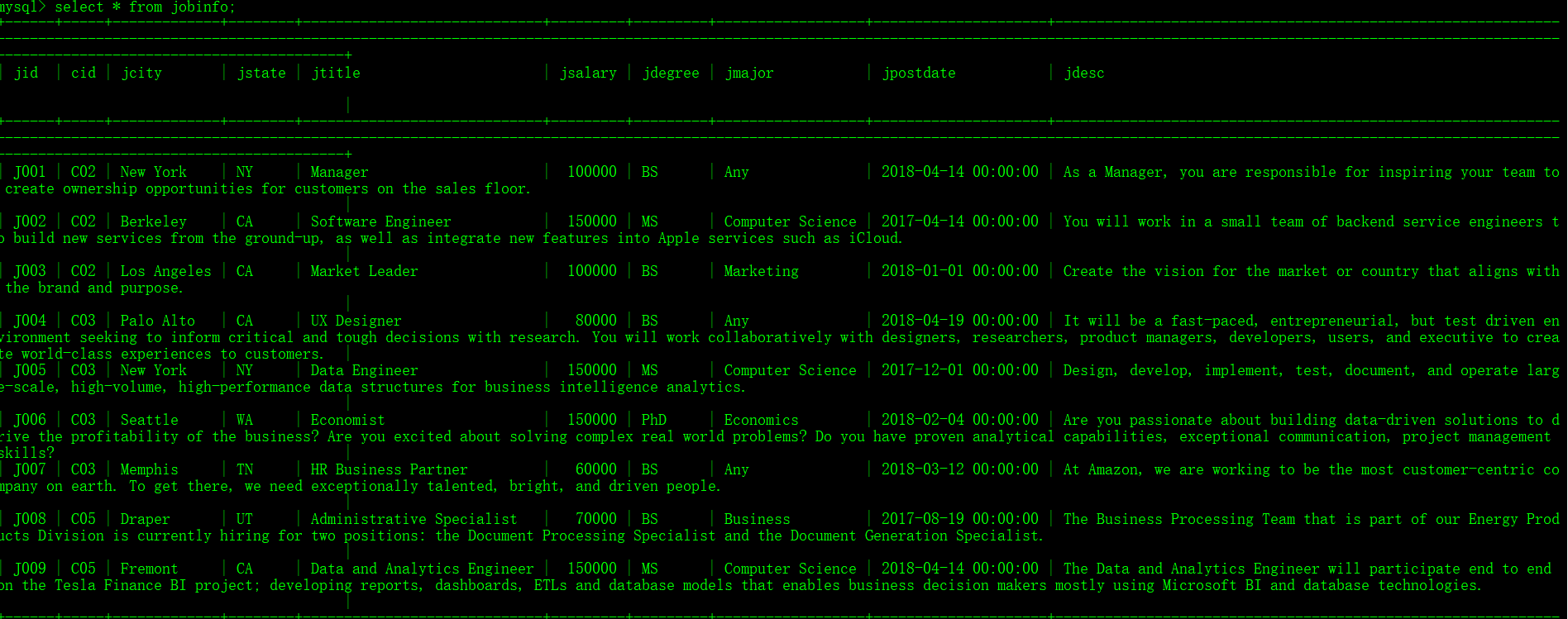


Friends

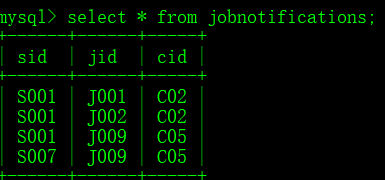


JobApply

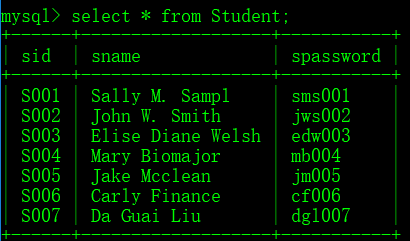


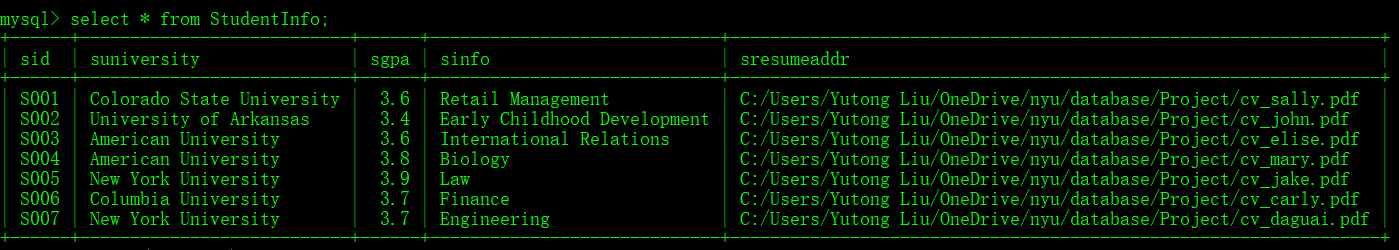
JobInfo

JobNotifications



Student



StudentInfo.

To sum up, these are data we have for now. We will add them in the second project. Also, the sql file is in the attachment, which called jonhunter.sql.