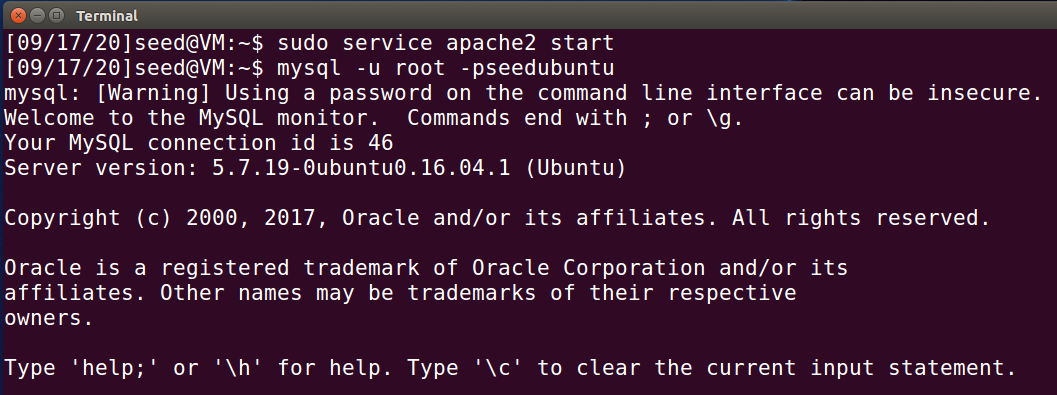
**Lab6-report**

**57118132 吴嘉琪**

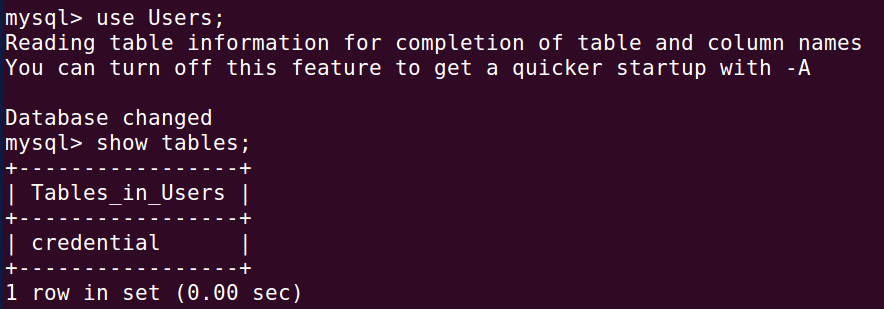
**Task1:** **Get Familiar with SQL Statements**

启动apache2并且进入MySQL控制台：

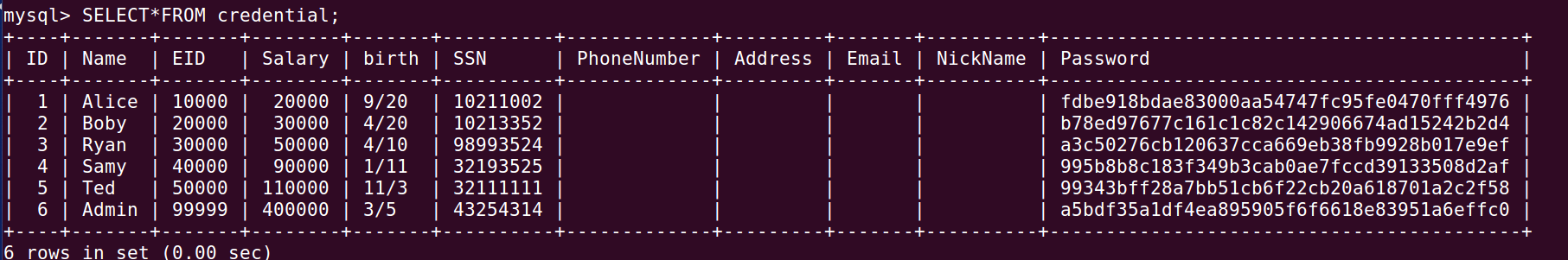


使用use Users; 进入到Users的数据库

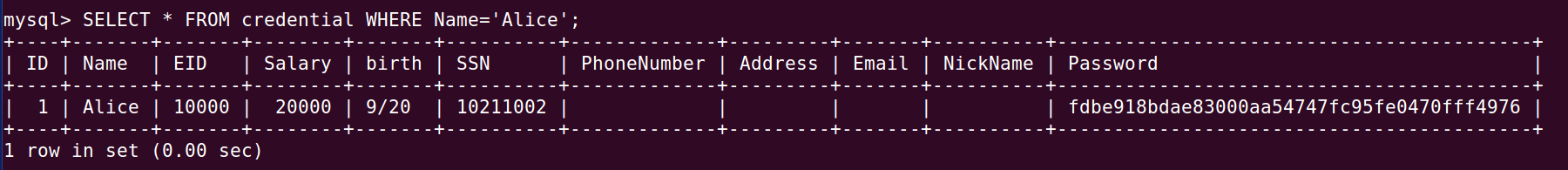
使用show tables; 查看Users数据库下的表格：



可以发现，表格名字叫做credential，所以使用SELECT \* FROM credential;命令查看表格中的内容：



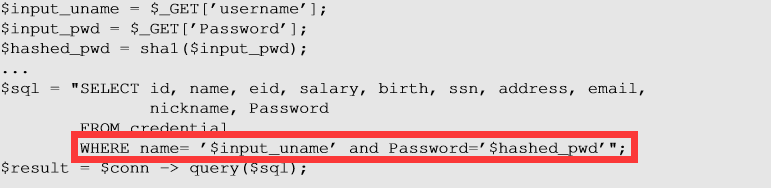
使用SELECT \* FROM credential WHERE Name='Alice';命令查看Alice的信息：



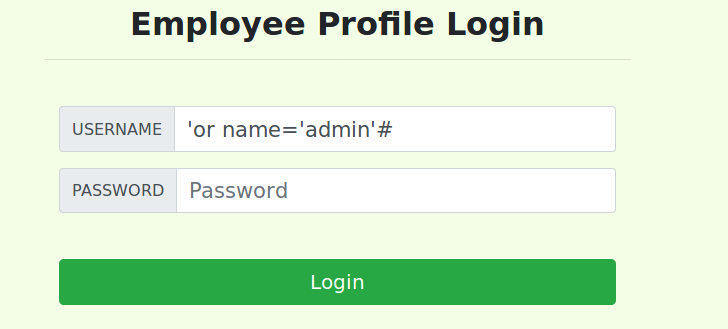
**Task2:** **SQL Injection Attack on SELECT Statement**

**Task 2.1: SQL Injection Attack from webpage**

由代码可知，可以知道select语句存在SQL注入漏洞，数据和代码是混合在一起的，所以我们可以通过在数据中将代码注释的方法绕过password的验证，当登入用户为admin时,会显示所有员工的信息

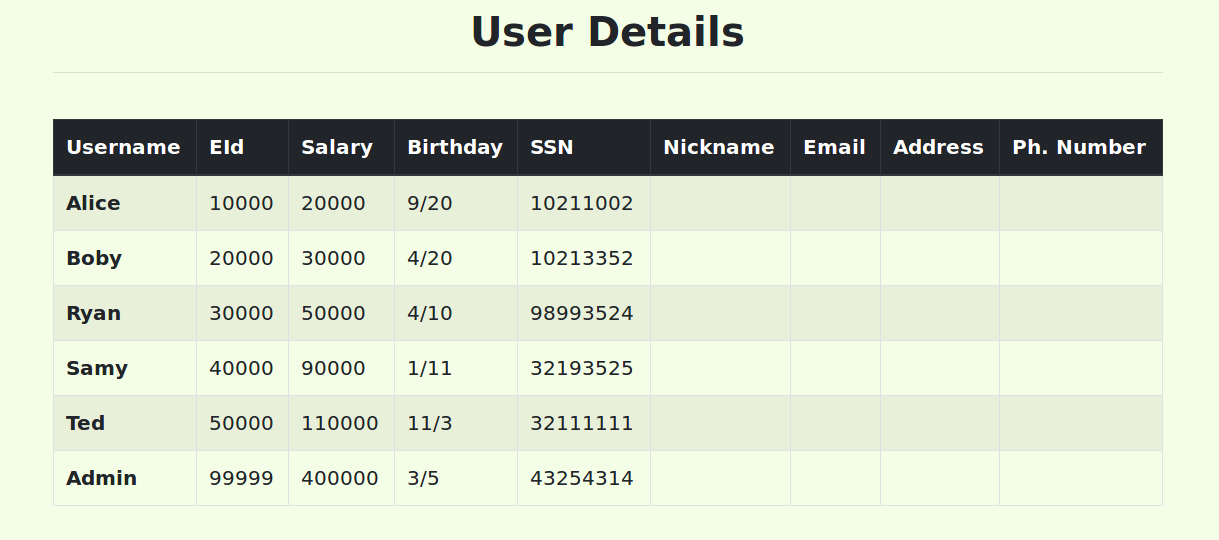


输入'or name='admin'#:



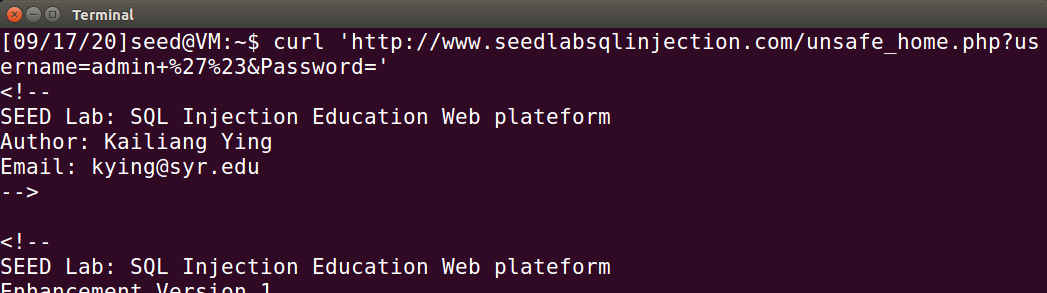
点击Login，发现进入了admin账户：

由代码分析可以知道,在ID栏输入’or name=’admin’#就可以成功以admin的身份登入.原因是,第一个单引号会闭合eid=’$input\_eid’,使条件变成了eid=’’or name=’admin’#,但是#会把后面的语句都注释掉,所以相当于直接登入admin

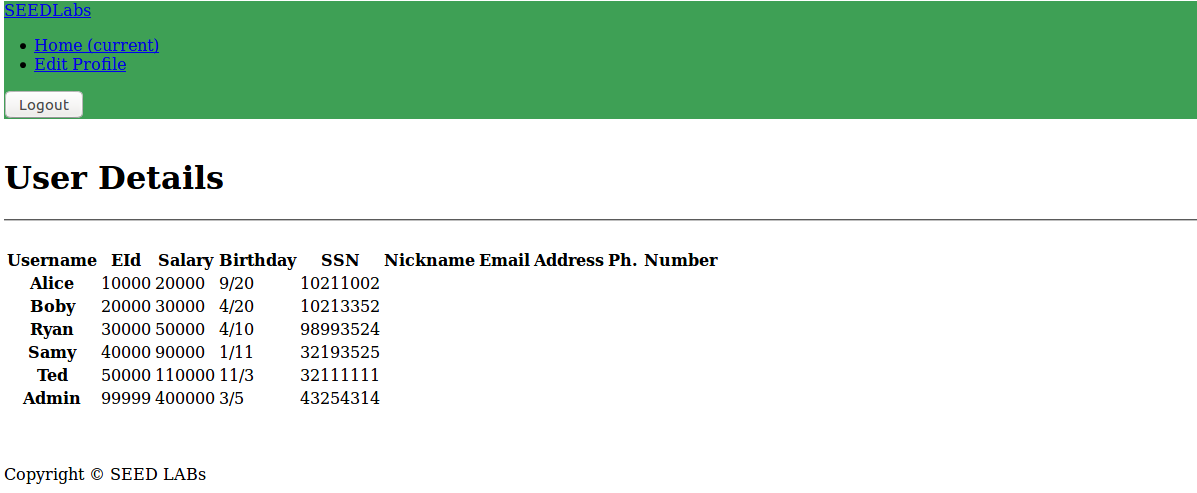


**Task 2.2: SQL Injection Attack from command line**

使用curl 'http://www.seedlabsqlinjection.com/unsafe\_home.php?username=admin+%27%23&Password=' 命令进行攻击。其中因为使用 # ’ 等符号会被程序改变为其他含义，所以使用这些符号的编码：%23 %27，因此得到上述的url，运行截图如下：

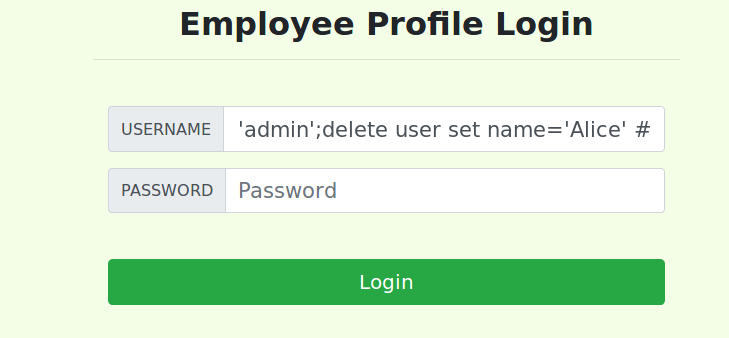


因为输出的是网页的所有代码，所以可以复制到一个html文件中并使用浏览器打开，则得到了排版好的数据：

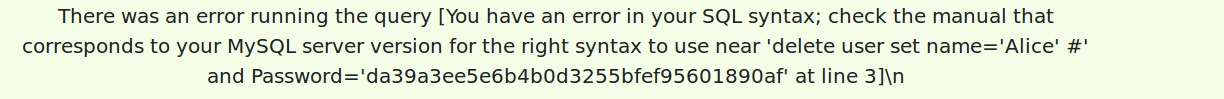


**Task 2.3: Append a new SQL statement**

输入'or name='admin';delete user set name='Alice' #:



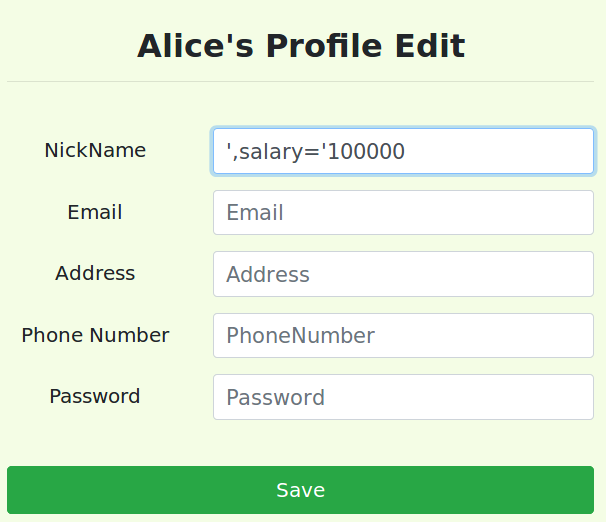
发现失败了,原因是, 在MySQL中实现了一种特殊的保护机制，mysq\_query不允许提交多个请求，导致我们两个连续的请求就会报错。



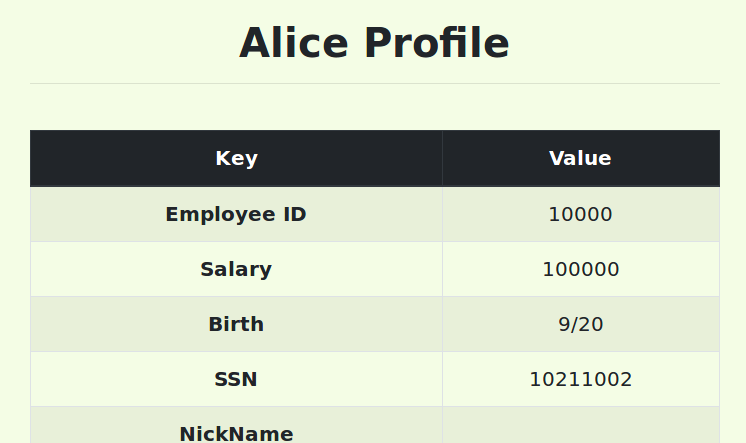
**Task3:** **SQL Injection Attack on UPDATE Statement**

**Task 3.1: Modify your own salary**

更改salary：

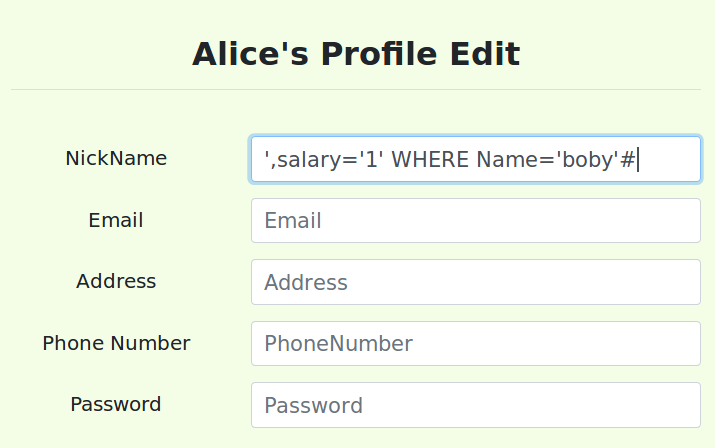


保存之后可以看到alice的工资改变了：

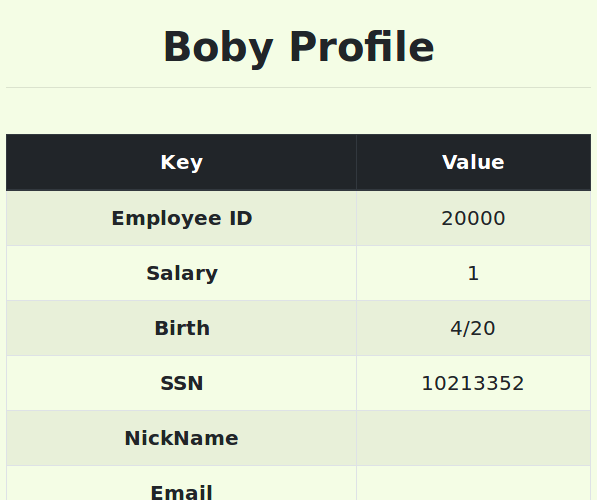


**Task 3.2: Modify other people’salary**

同样进入编辑界面，更改代码为 ',salary='1' WHERE Name='boby'# ：



保存后发现boby的工资被更改：



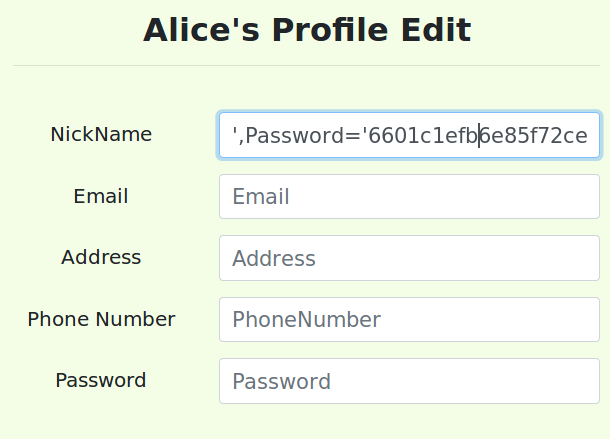
**Task 3.3: Modify other people’ password**

因为数据库存储的password是以SHA1加密的，所以我们先将想要设的密码的hash值计算出来：

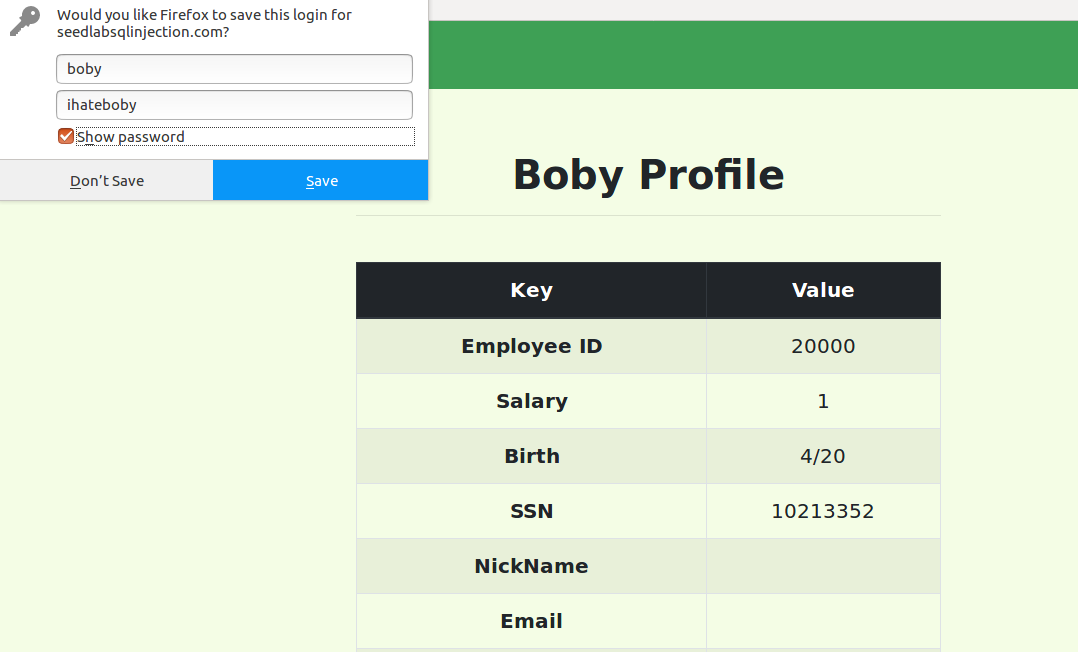
ihateboby :6601c1efb6e85f72ce74cc3dbaf62c6c9900cff5

使用代码更改boby的密钥：

',Password='6601c1efb6e85f72ce74cc3dbaf62c6c9900cff5' where Name='boby'#

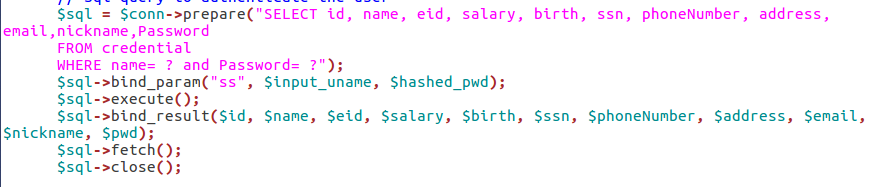


保存后尝试登录boby账号：

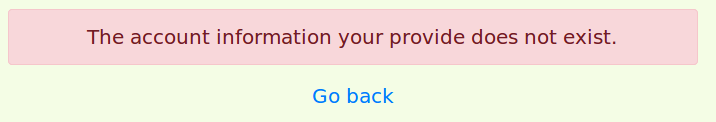


**Task4:** **Countermeasure—Prepared Statement**

**Unsafe\_home.php代码更改后如下**

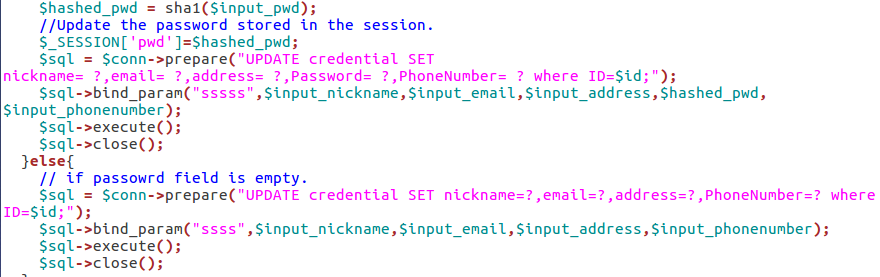


修改后再次攻击，发现攻击失败：



使用命令行进行攻击，同样失败

**Unsafe\_edit\_backend.php代码更改后如下：**



修改后再次攻击，发现攻击失败：

