2015-2016 年度

中山大学本科生实验报告

科目：数据库实验　　**教师：**郑贵锋

年级	13 级	专业（方向）	移动信息工程
学号	13354485	姓名	朱琳
电话	13726231932	Email	280273861@qq.com

一 实验目的

熟悉数据库基本的链接操作，如 INNER JOIN，LEFT OUTER JOIN ,RIGHT OUTER JOIN,UNION 等

二 实验内容及结果

1.建立一个名称为 SYSU 的数据库

```
mysql> CREATE DATABASE SYSU;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| artbase            |
| labfour            |
| mysql              |
| performance_schema |
| school             |
| sysu               |
| test               |
+--------------------+
8 rows in set (0.00 sec)
```

2.建立以下三个表格。

① lab5.sql 文档

```
CREATE TABLE Students(
    SID int,
    name CHAR(20),
    category CHAR(20)
);

CREATE TABLE Works(
    SID int,
    CID CHAR(20),
    Type CHAR(20)
);

CREATE TABLE Corporations(
    CID CHAR(20),
    Cname CHAR(20)
);


insert into Students values(10330000,"James","graduate");
insert into Students values(11330000,"Maria","undergraduate");
insert into Students values(12330000,"Zack","undergraduate");

insert into Works values(10330000,"PRC001","Employee");
insert into Works values(12330000,"PRC001","Intern");
insert into Works values(16330000,"USA001","Intern");

insert into Corporations values("PRC001","Alibaba");
insert into Corporations values("PRC002","Tencent");
insert into Corporations values("USA001","Google");
```

②导入数据库

```
E:\Desktop>mysql -h localhost -u root -p sysu < lab5.sql
Enter password: ******
```

③显示已经建立的表格

```
mysql> use sysu
Database changed
mysql> select * from Students;
+----------+-------+---------------+
| SID      | name  | category      |
+----------+-------+---------------+
| 10330000 | James | graduate      |
| 11330000 | Maria | undergraduate |
| 12330000 | Zack  | undergraduate |
+----------+-------+---------------+
3 rows in set (0.00 sec)

mysql> select * from Works;
+----------+--------+----------+
| SID      | CID    | Type     |
+----------+--------+----------+
| 10330000 | PRC001 | Employee |
| 12330000 | PRC001 | Intern   |
| 16330000 | USA001 | Intern   |
+----------+--------+----------+
3 rows in set (0.00 sec)

mysql> select * from Corporations;
+--------+---------+
| CID    | Cname   |
+--------+---------+
| PRC001 | Alibaba |
| PRC002 | Tencent |
| USA001 | Google  |
+--------+---------+
3 rows in set (0.00 sec)
```

3.完成以下问题(规定只能使用InnerJoin ，Left Join，Join Right Join和Union这几种方式来连接不同的表格):

(1)找出所有进入过大公司的学生的学号、姓名以及公司的ID

```
mysql> SELECT Students.SID,Students.name,Works.CID
    -> FROM Students INNER JOIN Works ON
    -> Students.SID=Works.SID;
+----------+-------+--------+
| SID      | name  | CID    |
+----------+-------+--------+
| 10330000 | James | PRC001 |
| 12330000 | Zack  | PRC001 |
+----------+-------+--------+
2 rows in set (0.00 sec)
```

【分析】首先确定需要找到的三个属性分别可以在 Students 表格和 Works 表格找到，并且两个表格具有一个公共属性就是 SID，那么我就可以通过 SID 对两个表格进行内联，这样的就得到具有相同 SID 的一个表格了，然后我再选出需要的三列（SID，name，CID）即可

(2)找出所有的学生的学号、姓名，以及他们进入过的大公司的名字，如果有的话

```
mysql> SELECT Students.SID,Students.name,Corporations.Cname FROM
    -> (Students LEFT OUTER JOIN Works ON Students.SID=Works.SID)
    -> LEFT OUTER JOIN Corporations ON Works.CID=Corporations.CID;
+----------+-------+---------+
| SID      | name  | Cname   |
+----------+-------+---------+
| 10330000 | James | Alibaba |
| 12330000 | Zack  | Alibaba |
| 11330000 | Maria | NULL    |
+----------+-------+---------+
3 rows in set (0.00 sec)
```

【分析】首先找到所有的学生->需要得到学生的全集->需要左连接（或者右连接）。初步确认为要使用左（右）连接。然后需要找到的三个属性分别在 Students 表格和 Corporations 表格，但是两个表格又没有公共属性，所以还需要一个中间表格 Works。也就是说我需要先将 Students 左连 Works（通过属性 SID），然后再将这个整体左连 Corporations，这样就可以得到含有 students 全集的一张表了。

(3)找出所有的大公司名称，以及进入过大公司的学生学号和姓名，如果有的话

```
mysql> SELECT Students.SID,Students.name,Corporations.Cname FROM
    -> (Students RIGHT OUTER JOIN Works ON Students.SID=Works.SID)
    -> RIGHT OUTER JOIN Corporations ON Works.CID=Corporations.CID;
+----------+-------+---------+
| SID      | name  | Cname   |
+----------+-------+---------+
| 10330000 | James | Alibaba |
| 12330000 | Zack  | Alibaba |
|     NULL | NULL  | Google  |
|     NULL | NULL  | Tencent |
+----------+-------+---------+
4 rows in set (0.00 sec)
```

【分析】与(2)类似，只不过现在是需要找到 Corporations 全集，那么我只需要将上述命令的左连改为右连即可。PS：其实关于 students 和 works 的连接，使用左连接和右连接的效果是一样的。但是左连就必须全部使用左连。

(4)找出所有的学生，以及他们进过的大公司，以及所有的大公司，以及进入过它们的学生。

(提示：这道题就是把第(2)问和第(3)问得到的2个表,做一个union操作就可以得到了)

```
mysql> SELECT Students.SID,Students.name,Corporations.Cname FROM
    -> (Students LEFT OUTER JOIN Works ON Students.SID=Works.SID)
    -> LEFT OUTER JOIN Corporations ON Works.CID=Corporations.CID
    -> union
    -> SELECT Students.SID,Students.name,Corporations.Cname FROM
    -> (Students RIGHT OUTER JOIN Works ON Students.SID=Works.SID)
    -> RIGHT OUTER JOIN Corporations ON Works.CID=Corporations.CID;
+----------+-------+---------+
| SID      | name  | Cname   |
+----------+-------+---------+
| 10330000 | James | Alibaba |
| 12330000 | Zack  | Alibaba |
| 11330000 | Maria | NULL    |
|     NULL | NULL  | Google  |
|     NULL | NULL  | Tencent |
+----------+-------+---------+
5 rows in set (0.00 sec)
```

【分析】按照提示，现在我只需要将(2)(3)中得出的表进行取交集，即进行 UNION 的操作，即只需在两个 SELECT 之间进行取交集即可（前提是所选择的列的属性相同）

三 实验心得

1. 这次实验主要运用 INNER JOIN,LEFT JOIN,RIGHT JOIN 和 UNION.主要考察的也是三张表的连接。内容比较简单，只要理解了各个语句的意义，写出来相应的语句是完全不成问题的。另外还有一个小技巧，就是只要出现“所有的 XXX”几个字，那么基本上所有的连接都是倾向于他，就是说如果他在表的左边，就需要使用左连接，如果他在表的右边，就使用右连接。如果在中间（如果你真的闲的没事干的话），就先用右连接再使用左连接——为了验证一下，我做了如下测试：以（2）题为例，“**找出所有的学生的学号、姓名，以及他们进入过的大公司的名字**”。

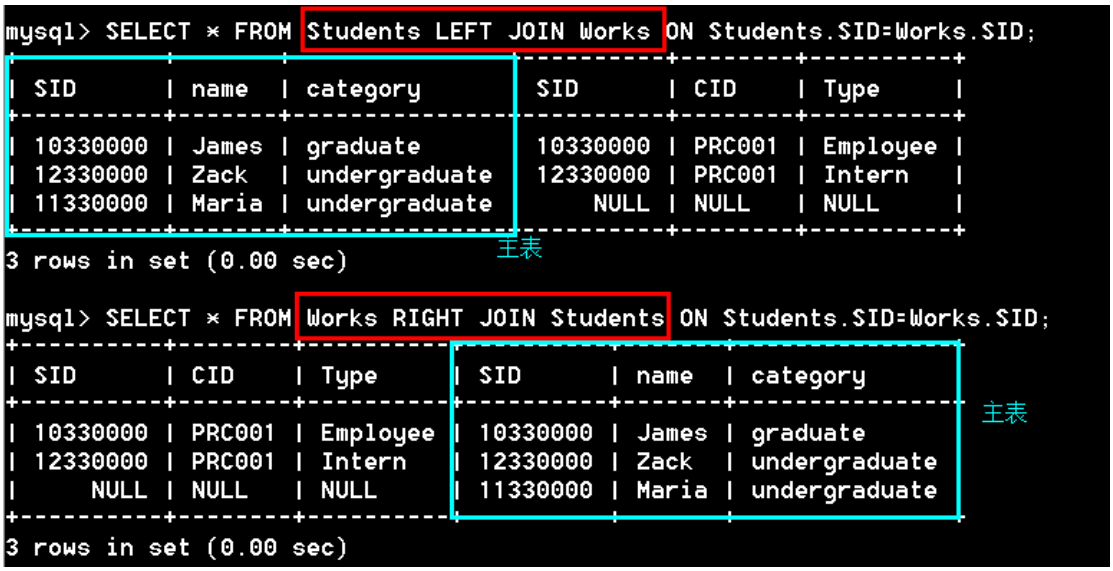
```
mysql> SELECT Students.SID,Students.name,Corporations.Cname FROM
    -> (works RIGHT OUTER JOIN Students ON Students.SID=Works.SID)
    -> LEFT OUTER JOIN Corporations ON Works.CID=Corporations.CID;
+----------+-------+---------+
| SID      | name  | Cname   |
+----------+-------+---------+
| 10330000 | James | Alibaba |
| 12330000 | Zack  | Alibaba |
| 11330000 | Maria | NULL    |
+----------+-------+---------+
3 rows in set (0.00 sec)
```

以上就是 Works->right join->Students->left join->Corporations.总之就是始终让目标保持全集即可。

2.关于左外连接和右外连接的区别，左外连接是以左表为主表，右表为辅表，显示的是左表的全部内容和右表中与之匹配的内容。而右外连接正好相反，他是以右表为主，左表为辅，显示的是

右表的全部内容和左表的匹配内容。

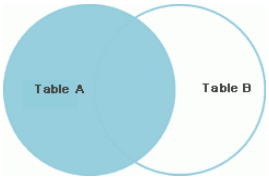
此外还有一个不值一提的区别就是当全选择列的时候(*)，如果是两个表格是按照出现的顺序进行排列的而并非是按照主表和辅表进行排列的。如下图，都是以 Students 为主表，第一个是左连接，一个是右连接：



3.mysql 写命令行的格式一定要规范，这样不仅有利于代码的解读，也非常有利于命令的复用，比如我想使用之前打过的命令，一长串的命令行肯定会带来很多麻烦，如果我把命令行分成几段，那么再用“上箭头”进行命令复用的时候就容易修改多了。So，规范的格式可以带来很多意想不到的好处。

4.关于上述命令的使用，现总结如下：

交叉连接	作用	交叉连接返回左表中的所有行，左表中的每一行与右表中的所有行组合。（笛卡尔积）
	语法	语句1：隐式的交叉连接，没有显示的声明cross join。 select * from table1, table2; 语句2：显式的交叉连接，使用cross join。 select * from table1 cross join table2;
内连接	作用	内连接返回链接表中符合连接条件和查询条件的数据行。（所谓的链接表就是数据库在做查询形成的中间表）。**INNER JOIN 产生的结果是 AB 的交集**
	语法	SELECT * FROM TableA INNER JOIN TableB ON TableA.name = TableB.name

<table>
<tr><td rowspan="2">左连接</td><td>作用</td><td>LEFT [OUTER] JOIN 产生表A的完全集，而B表中匹配的则有值，没有匹配的则以null值取代。</td></tr>
<tr><td>语法</td><td>SELECT * FROM TableA LEFT OUTER JOIN TableB ON TableA.name = TableB.name
  </td></tr>
<tr><td rowspan="2">右连接</td><td>作用</td><td>RIGHT [OUTER] JOIN 产生表B的完全集，而A表中匹配的则有值，没有匹配的则以null值取代。</td></tr>
<tr><td>语法</td><td>SELECT * FROM TableA RIGHT OUTER JOIN TableB ON TableA.name = TableB.name</td></tr>
<tr><td rowspan="2">全连接</td><td>作用</td><td>FULL OUTER JOIN 关键字只要左表（table1）和右表（table2）其中一个表中存在匹配，则返回行.
FULL OUTER JOIN 关键字结合了 LEFT JOIN 和 RIGHT JOIN 的结果。</td></tr>
<tr><td>语法</td><td>FULL [OUTER] JOIN 产生A和B的并集。对于没有匹配的记录，则会以null做为值。
SELECT * FROM TableA FULL OUTER JOIN TableB ON TableA.name = TableB.name

【注意】mysql 不支持 FULL JOIN</td></tr>
<tr><td rowspan="2">并集</td><td>作用</td><td>UNION 操作符用于合并两个或多个 SELECT 语句的结果集。
请注意，UNION 内部的 SELECT 语句必须拥有相同数量的列。列也必须拥有相似的数据类型。同时，每条 SELECT 语句中的列的顺序必须相同。</td></tr>
<tr><td>语法</td><td>select
column_name(s) from table_name1
union
select
column_name(s) from table_name2</td></tr>
</table>