**南昌大学实验报告**



学生姓名： 杨孟衡 学 号： 8002118240 专业班级： 软工1809

实验类型：☑ 验证 □ 综合 □ 设计 □ 创新 实验日期： 3.12 实验成绩：

1. 实验项目名称

实验二 数据库查询

1. 实验内容

1. 在studentsdb数据库中使用SELECT语句进行基本查询。

（1）在student\_info表中，查询每个学生的学号、姓名、出生日期信息。

（2）查询student\_info表学号为 0002的学生的姓名和家庭住址。

（3）查询student\_info表所有出生日期在95年以后的女同学的姓名和出生日期。

2. 使用select语句进行条件查询。

（1）在grade表中查询分数在70-80范围内的学生的学号、课程编号和成绩。

（2）在grade表中查询课程编号为0002的学生的平均成绩。

（3）在grade表中查询选修课程编号为0003的人数和该课程有成绩的人数。

（4）查询student\_info的姓名和出生日期，查询结果按出生日期从大到小排序。

（5）查询所有姓名“张”的学生的学号和姓名。

3. 对student\_info表，查询学生的学号、姓名、性别、出生日期及家庭住址，查询结果先按照性别的由小到大排序，性别相同的再按学号由大到小排序。

4. 使用GROUP BY子句查询grade表中各个学生的平均成绩。

5. 使用UNION运算符针student\_info表中姓“刘”的学生的学号、姓名与姓“张”的学生的学号、姓名返回在一个表中。

6. 嵌套查询

（1）在student\_info表中查找与“刘东阳”性别相同的所有学生的姓名、出生日期。

（2）使用IN子查询查找所修课程编号为0002、0005的学生学号、姓名、性别。

（3）使用ANY子查询查找学号为0001的学生的分数比0002号的学生的最低分数高的课程编号和分数。

（4）使用ALL子查询查找学号为0001的学生的分数比学号为0002的学生的最高成绩还要高的课程编号和分数。

7. 连接查询

（1）查询分数在80-90范围内的学生的学号、姓名、分数。

（2）使用INNER JOIN连接方式查询学习“数据库原理及应用”课程的学生学号、姓名、分数。

（3）查询每个学生所选课程的最高成绩，要求列出学号、姓名、最高成绩。

（4）使用左外连接查询每个学生的总成绩，要求列出学号、姓名、总成绩，没有选修课程的学生的总成绩为空。

（5）为grade表添加数据行：学号为0004、课程编号为0006、分数为76。

使用右外连接查询所有课程的选修情况，要求列出课程编号、课程名称、选修人数，curriculum表中没有的课程列值为空。

实验环境：windows10操作系统,，mysql8.0.16，Navicat12.0

1. 实验要求
2. 截图SQL代码及运行结果图，截图清晰。
3. 学习通平台提交实验报告。
4. 文件名：姓名+学号+实验几+班级
5. 实验步骤及处理结果

1.SELECT语句实现基本查询：

* 打开Navicat，连接mysql数据库，登录用户，选择数据库studentsdb，新建一个查询temp；

（1）查询每个学生的学号、姓名、出生日期信息

* 在打开的查询页面中输入以下sql语句：

SELECT 学号,姓名,出生日期 FROM student\_info;

* 运行查询，查询结果如图：

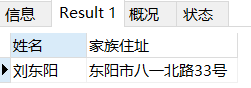


（2）查询0002号学生的姓名和家庭住址：

* 在打开的查询页面中输入以下sql语句：

SELECT 姓名,家族住址 FROM student\_info WHERE 学号='0002';

* 运行查询，查询结果如图：

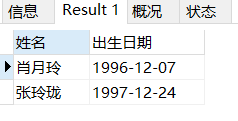


（3）查询所有出生日期在95年以后的女同学的姓名和出生日期：

* 在打开的查询页面中输入以下sql语句：

SELECT 姓名,出生日期 FROM student\_info WHERE 性别='女' AND YEAR(出生日期)>1995;

* 运行查询，查询结果如图：



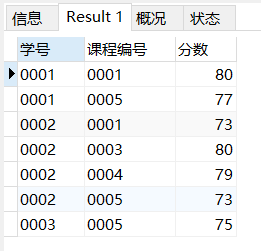
2.SELECT语句实现条件查询：

（1）查询分数在70-80范围内的学生学号、课程编号和成绩：

* 在打开的查询页面中输入以下sql语句：

SELECT 学号,课程编号,分数 FROM grade WHERE 分数 BETWEEN 70 AND 80;

* 运行查询，查询结果如图：

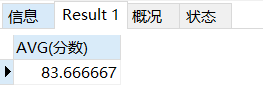


（2）查询课程编号为0002的学生的平均成绩：

* 在打开的查询页面中输入以下sql语句：

SELECT AVG(分数) FROM grade WHERE 课程编号='0002';

* 运行查询，查询结果如图：



（3）查询选修课课程编号为0003号的人数和该课程有成绩的人数：

* 在打开的查询页面中输入以下sql语句：

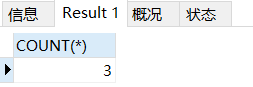
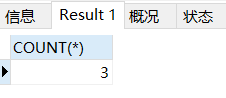
SELECT COUNT(\*) FROM grade WHERE 课程编号='0003';#选修课程编号为0003的人数

SELECT COUNT(\*) FROM grade WHERE 课程编号='0003' AND 课程编号 IS NOT NULL;

#该课程有成绩的人数

* 运行查询，查询结果如图：

选修人数有成绩的人数



（4）查询学生姓名和出生日期，结果按出生日期从大到小排序：

* 在打开的查询页面中输入以下sql语句：

SELECT 姓名,出生日期 FROM student\_info ORDER BY 出生日期 DESC;

* 运行查询，查询结果如图：

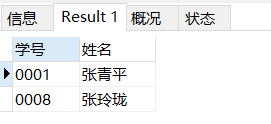


（5）查询所有姓名“张”的学生学号和姓名：

* 在打开的查询页面中输入以下sql语句：

SELECT 学号,姓名 FROM student\_info WHERE 姓名 LIKE '张%';

* 运行查询，查询结果如图：



3查询学生学号、姓名、性别、出生日期及家族住址，查询结果先按照性别的由小到大排序，性别相同的再按学号由大到小排序：

* 在打开的查询页面中输入以下sql语句：

SELECT 学号,姓名,性别,出生日期,家族住址 FROM student\_info ORDER BY 性别 ASC, 学号 DESC;

* 运行查询，查询结果如图：

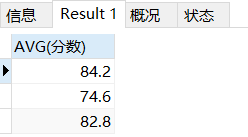


4.使用GROUP BY子句查询grade表中各个学生的平均成绩：

* 在打开的查询页面中输入以下sql语句：

SELECT AVG(分数) FROM grade GROUP BY 学号;

* 运行查询，查询结果如图：



5.使用UNION运算符针student\_info表中姓“刘”的学生学号、姓名与姓“张”学生学号、姓名返回在一个表中：

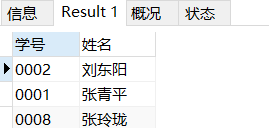
* 在打开的查询页面中输入以下sql语句：

SELECT 学号,姓名 FROM student\_info WHERE 姓名 LIKE '刘%'

UNION

SELECT 学号,姓名 FROM student\_info WHERE 姓名 LIKE '张%';

* 运行查询，查询结果如图：



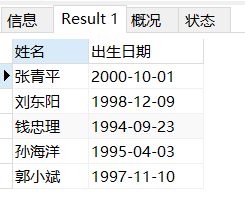
6.SELECT语句实现嵌套查询：

（1）查询与“刘东阳”性别相同的所有学生的姓名、出生日期：

* 在打开的查询页面中输入以下sql语句：

SELECT 姓名,出生日期 FROM student\_info WHERE 性别=(SELECT 性别 FROM student\_info WHERE 姓名='刘东阳');

* 运行查询，查询结果如图：



（2）使用IN子查询查询所修课程编号为0002、0005的学生学号、姓名、性别：

* 在打开的查询页面中输入以下sql语句：

SELECT student\_info.学号,姓名,性别 FROM student\_info, grade WHERE student\_info.`学号`=grade.`学号` AND grade.`课程编号`='0002' AND student\_info.`学号` IN(SELECT 学号 FROM grade WHERE 课程编号='0005');

* 运行查询，查询结果如图：

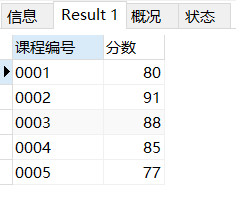


（3）使用ANY子查询查询学号为0001的学生分数比学生为0002学生的最低分数高的课程编号和分数：

* 在打开的查询页面中输入以下sql语句：

SELECT curriculum.`课程编号`,分数 FROM grade, curriculum WHERE grade.`课程编号`=curriculum.`课程编号` AND grade.`学号`='0001' AND grade.分数>ANY(SELECT 分数 FROM grade g1 WHERE g1.学号='0002');

* 运行查询，查询结果如图：

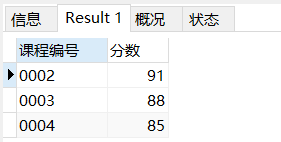


（4）使用ALL子查询查询学号为0001的学生分数比学生为0002学生的最高成绩还要高的课程编号和分数：

* 在打开的查询页面中输入以下sql语句：

SELECT curriculum.`课程编号`,分数 FROM grade, curriculum WHERE grade.`课程编号`=curriculum.`课程编号` AND grade.`学号`='0001' AND grade.分数>ALL(SELECT 分数 FROM grade g1 WHERE g1.学号='0002');

* 运行查询，查询结果如图：



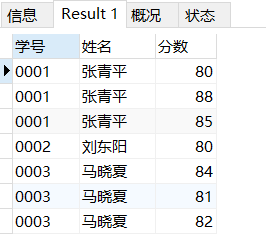
6.SELECT语句实现嵌套查询：

（1）查询分数在80-90范围内的学生学号、姓名、分数：

* 在打开的查询页面中输入以下sql语句：

SELECT S.学号,姓名,分数 FROM student\_info S,grade G WHERE S.`学号`=G.`学号` AND G.`分数` BETWEEN 80 AND 90;

* 运行查询，查询结果如图：

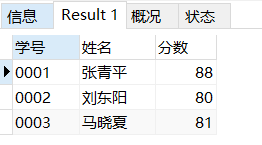


（2）使用INNER JOIN连接方式查询学习“数据库原理及应用”课程的学生学号、姓名、分数：

* 在打开的查询页面中输入以下sql语句：

SELECT student\_info.`学号`,student\_info.姓名,分数 FROM student\_info INNER JOIN grade ON student\_info.`学号`=grade.`学号` WHERE 课程编号=(SELECT 课程编号 FROM curriculum WHERE 课程名称='数据库原理及应用');

* 运行查询，查询结果如图：



（3）查询每个学生所选课程的最高成绩：

* 在打开的查询页面中输入以下sql语句：

SELECT S.学号,S.姓名,MAX(G.分数) 最高成绩 FROM student\_info S, grade G WHERE S.`学号`=G.`学号` GROUP BY S.`学号`;

* 运行查询，查询结果如图：

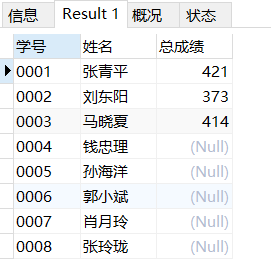


（4）使用左外连接查询每个学生的总成绩：

* 在打开的查询页面中输入以下sql语句：

SELECT S.学号,S.姓名,SUM(G.分数) 总成绩 FROM student\_info S LEFT JOIN grade G ON S.`学号`=G.`学号` GROUP BY S.`学号`;

* 运行查询，查询结果如图：



（5）为grade表添加数据行：学号为0004、课程编号为0006、分数为76。

使用右外连接查询所有课程的选修情况，要求列出课程编号、课程名称、选修人数，curriculum表中没有的课程列值为空：

* 在打开的查询页面中输入以下sql语句：

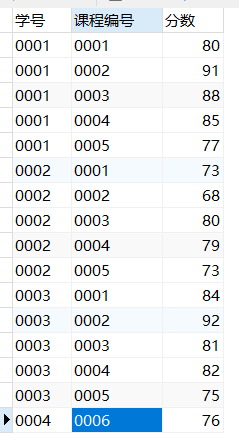
#在grade表中插入数据

INSERT INTO grade VALUES('0004','0006',76);

#若是没有此课程存在，则显示NULL

SELECT C.课程编号,C.课程名称,COUNT(\*) 选修人数 FROM curriculum C RIGHT JOIN grade G ON G.`课程编号`=C.`课程编号` GROUP BY G.`课程编号`;

* 运行查询，查询结果如图：



1. 思考体会或对改进实验的建议

本次实验是本学期数据库学习第二次试验，难度适中，个人感受良好，从这次实验里能感受到

快乐，尤其是查询的快乐，嵌套查询和多表连接查询可以互相转换，只不过语句数量多与少的问题，最重要的是能让我体会到连接关键字的好处！上周已经学习完了嵌套查询，连接查询，所以本次实验也没有花太多的时间，主要还是一个复习巩固以及实操mysql语句的作用，让我又好好的复习了mysql语句与sql语句之间的差别，大同小异，但小细节错一点点都不行，这教会了我作为程序猿的基本素质，抓细节！

花时间花的比较多的就是嵌套查询那里，虽然不是时间也不是很长，但是嵌套查询的确是难度较大的数据库应用！不过多来一点这类题目也可以检验我们自己编写代码，修改代码的能力，培养逻辑思维能力。

1. 参考资料

[1]何玉洁.数据库系统教程（第二版）[M].北京：人民邮电出版社, 978-7-115-40245-5