**MySQL实验实训手册**

**实验目的：**

通过对《MySQL数据库入门》教材的学习，读者已经掌握了数据库基础知识、数据库和表的基本操作、单表中数据的增删改查、多表中数据的查询操作以及数据库开发的一些高级知识。本实训手册将通过一个综合案例，让读者熟悉并掌握在实际开发中如何灵活应用这些知识。

**实验学时：**

2学时

**考核要点：**创建数据表、单表查询、多表查询

**实验内容：**

已知，有一个学生表student和一个分数表score，请按要求对这两个表进行操作。student表和score分数表的表结构分别如表1-1和表1-2所示。

1. student表结构

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** | **字段描述** |
| stu\_id | INT(10) | 是 | 否 | 是 | 是 | 否 | 学号 |
| stu\_name | VARCHAR(20) | 否 | 否 | 是 | 否 | 否 | 姓名 |
| sex | VARCHAR(2) | 否 | 否 | 否 | 否 | 否 | 性别 |
| birth | YEAR | 否 | 否 | 否 | 否 | 否 | 出生年份 |
| department | VARCHAR(20) | 否 | 否 | 是 | 否 | 否 | 院系 |
| addr | VARCHAR(50) | 否 | 否 | 否 | 否 | 否 | 家庭住址 |

1. score表结构

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **主键** | **外键** | **非空** | **唯一** | **自增** | **字段描述** |
| score\_id | INT(10) | 是 | 否 | 是 | 是 | 是 | 编号 |
| stu\_id | INT(10) | 否 | 否 | 是 | 否 | 否 | 学号 |
| c\_name | VARCHAR(20) | 否 | 否 | 否 | 否 | 否 | 课程名 |
| grade | INT(10) | 否 | 否 | 否 | 否 | 否 | 分数 |

表1-1和表1-2中分别对student表和score表中字段的数据类型、主键、外键、非空、唯一约束、自增和字段描述做了说明，其中这两个表中都有stu\_id字段，用于表示学号，该字段使student表和score表实现了关联。

（1）启动MySQL服务器。

如果MySQL服务器是未启动的状态，可以在cmd窗口中，通过命令来启动服务器，具体命令如下所示：

net start mysql

（2）登录MySQL服务器。

在cmd窗口中登录MySQL服务器，用户名为root，密码为itcast，具体命令如下所示：

mysql -uroot -pitcast

（3）创建数据库。

创建一个名为my\_db的数据库，具体SQL语句如下：

create database my\_db;

（4）切换当前数据库。

将名为my\_db的数据库切换为当前数据库，具体SQL语句如下：

use my\_db;

（5）创建学生表student，插入6条记录。

创建学生表student，SQL语句如下所示：

CREATE TABLE student (

stu\_id INT(10) PRIMARY KEY,

stu\_name VARCHAR(20)NOT NULL,

sex VARCHAR(2),

birth YEAR,

department VARCHAR(20),

addr VARCHAR(50)

);

上述SQL语句执行成功后，表示学生表student创建成功，这时，使用INSERT语句向表中插入数据，SQL语句如下所示：

INSERT INTO student VALUES

( 901,'张飞', '男',1985,'计算机系', '河北省涿州市'),

( 902,'关羽', '男',1986,'中文系', '山西省运城市'),

( 903,'貂蝉', '女',1990,'中文系', '山西省忻州县'),

( 904,'刘备', '男',1990,'英语系', '河北省涿州市'),

( 905,'小乔', '女',1991,'英语系', '安徽省潜山市'),

( 906,'赵云', '男',1988,'计算机系', '河北省正定市');

上述INSERT语句执行成功后，向表中插入了6条记录，分别是学生的学号、姓名、性别、出生年份、院系和出生地址，这时，使用SELECT语句查看stu表中的数据信息，查询结果如下所示：

mysql> select \* from student;

+--------+----------+------+-------+------------+----------+

| stu\_id | stu\_name | sex | birth | department | addr |

+--------+----------+------+-------+------------+----------+

| 901 | 张飞 | 男 | 1985 | 计算机系 | 河北省涿州市 |

| 902 | 关羽 | 男 | 1986 | 中文系 | 山西省运城市 |

| 903 | 貂蝉 | 女 | 1990 | 中文系 | 山西省忻州县 |

| 904 | 刘备 | 男 | 1990 | 英语系 | 河北省涿州市 |

| 905 | 小乔 | 女 | 1991 | 英语系 | 安徽省潜山市 |

| 906 | 赵云 | 男 | 1988 | 计算机系 | 河北省正定市 |

+--------+----------+------+-------+------------+----------+

6 rows in set (0.02 sec)

从查询结果可以看出，在当前的数据库中创建了一个student表，并成功插入了6条记录，student表的主键为id。

（6）创建分数表score，插入10条记录。

首先创建分数表score，SQL语句如下所示：

CREATE TABLE score (

score\_id INT(10) PRIMARY KEY AUTO\_INCREMENT ,

stu\_id INT(10) NOT NULL ,

c\_name VARCHAR(20) ,

grade INT(10)

);

上述SQL语句执行成功后，表示分数表score创建成功，接下来，使用INSERT语句向score表中插入数据，SQL语句如下所示：

INSERT INTO score(stu\_id,c\_name,grade) VALUES(901, '计算机',98),

(901, '英语', 80),(902, '计算机',65),(902, '中文',88),

(903, '中文',95),(904, '计算机',70),(904, '英语',92),

(905, '英语',94),(906, '计算机',90),(906, '英语',85);

上述SQL语句执行成功后，向表中插入了10条记录，分别是学生的学号、姓名、所在学校和报考的学校名称，这时，使用SELECT语句查看score表中的数据信息，查询结果如下所示：

mysql> select \* from score;

+----------+--------+--------+-------+

| score\_id | stu\_id | c\_name | grade |

+----------+--------+--------+-------+

| 1 | 901 | 计算机 | 98 |

| 2 | 901 | 英语 | 80 |

| 3 | 902 | 计算机 | 65 |

| 4 | 902 | 中文 | 88 |

| 5 | 903 | 中文 | 95 |

| 6 | 904 | 计算机 | 70 |

| 7 | 904 | 英语 | 92 |

| 8 | 905 | 英语 | 94 |

| 9 | 906 | 计算机 | 90 |

| 10 | 906 | 英语 | 85 |

+----------+--------+--------+-------+

10 rows in set (0.00 sec)

从查询结果可以看出，score表创建成功，同时向表中插入了10条记录，score表的主键为score\_id。

（7）从student表中查询计算机系和英语系学生的信息。

该操作属于单表查询操作，SQL语句如下所示：

mysql> SELECT \* FROM student

-> WHERE department IN ('计算机系','英语系');

+--------+----------+------+-------+------------+----------+

| stu\_id | stu\_name | sex | birth | department | addr |

+--------+----------+------+-------+------------+----------+

| 901 | 张飞 | 男 | 1985 | 计算机系 | 河北省涿州市 |

| 904 | 刘备 | 男 | 1990 | 英语系 | 河北省涿州市 |

| 905 | 小乔 | 女 | 1991 | 英语系 | 安徽省潜山市 |

| 906 | 赵云 | 男 | 1988 | 计算机系 | 河北省正定市 |

+--------+----------+------+-------+------------+----------+

4 rows in set (0.00 sec)

上述SQL语句执行成功后，顺利查询出了计算机系和英语系的学生的信息，上面的SQL语句使用了in操作符，针对本操作其实还有另外一种查询方式，SQL语句如下所示：

mysql> SELECT \* FROM student

-> WHERE department='计算机系' OR department='英语系';

+--------+----------+------+-------+------------+----------+

| stu\_id | stu\_name | sex | birth | department | addr |

+--------+----------+------+-------+------------+----------+

| 901 | 张飞 | 男 | 1985 | 计算机系 | 河北省涿州市 |

| 904 | 刘备 | 男 | 1990 | 英语系 | 河北省涿州市 |

| 905 | 小乔 | 女 | 1991 | 英语系 | 安徽省潜山市 |

| 906 | 赵云 | 男 | 1988 | 计算机系 | 河北省正定市 |

+--------+----------+------+-------+------------+----------+

4 rows in set (0.00 sec)

从查询结果可以看出，使用or操作符也得到了正确的查询结果。

（8）从student表中查询年龄25~28岁的学生信息。

该操作属于单表查询操作，SQL语句如下所示：

mysql> SELECT stu\_id,stu\_name,sex,

-> 2016-birth AS age,department,addr

-> FROM student

-> WHERE 2016-birth BETWEEN 25 AND 28;

+--------+----------+------+------+------------+----------+

| stu\_id | stu\_name | sex | age | department | addr |

+--------+----------+------+------+------------+----------+

| 903 | 貂蝉 | 女 | 26 | 中文系 | 山西省忻州县 |

| 904 | 刘备 | 男 | 26 | 英语系 | 河北省涿州市 |

| 905 | 小乔 | 女 | 25 | 英语系 | 安徽省潜山市 |

| 906 | 赵云 | 男 | 28 | 计算机系 | 河北省正定市 |

+--------+----------+------+------+------------+----------+

4 rows in set (0.00 sec)

上述SQL语句中，2016-birth表示当前年份减去学生出生年份，得到的就是学生的年龄，并使用as操作符为列取别名为age，BETWEEN... AND...操作符用于表示某个范围之间的数。从查询结果可以看出，顺利得到要求的结果。

针对本操作也可以不使用BETWEEN... AND...操作符，改为使用关系运算符来对年龄范围进行限制，SQL语句如下所示：

mysql> SELECT stu\_id,stu\_name,sex,

-> 2016-birth AS age,department,addr

-> FROM student

-> WHERE 2016-birth>=25 AND 2016-birth<=28;

+--------+----------+------+------+------------+----------+

| stu\_id | stu\_name | sex | age | department | addr |

+--------+----------+------+------+------------+----------+

| 903 | 貂蝉 | 女 | 26 | 中文系 | 山西省忻州县 |

| 904 | 刘备 | 男 | 26 | 英语系 | 河北省涿州市 |

| 905 | 小乔 | 女 | 25 | 英语系 | 安徽省潜山市 |

| 906 | 赵云 | 男 | 28 | 计算机系 | 河北省正定市 |

+--------+----------+------+------+------------+----------+

4 rows in set (0.00 sec)

（9）在student表中统计每个院系各有几个学生。

该操作属于单表查询操作，使用GROUP BY 分组查询，通过院系department来分组， SQL语句如下所示：

mysql> SELECT department AS "院系", COUNT(id) AS "人数"

-> FROM student

-> GROUP BY department;

+----------+------+

| 院系 | 人数 |

+----------+------+

| 中文系 | 2 |

| 英语系 | 2 |

| 计算机系 | 2 |

+----------+------+

3 rows in set (0.00 sec)

从上述查询结果可以看出，每个院系的学生均为2个人。

（10）查询每个院系学生中的最高分。

该操作属于多表查询操作，SQL语句如下所示：

mysql> SELECT MAX(grade),department

-> FROM score

-> LEFT JOIN student ON score.stu\_id = student.st

-> GROUP BY department;

+------------+------------+

| MAX(grade) | department |

+------------+------------+

| 95 | 中文系 |

| 94 | 英语系 |

| 98 | 计算机系 |

+------------+------------+

3 rows in set (0.00 sec)

上述SQL语句执行成功后，顺利查询出了每个院系学生的最高分。

（11）查询学生貂蝉的考试科目（c\_name）和考试成绩（grade）

该操作属于多表查询操作，SQL语句如下所示：

mysql> SELECT stu\_name,c\_name,grade

-> FROM score

-> LEFT JOIN student ON score.stu\_id = student.stu\_id

-> WHERE stu\_name= "貂蝉";

+----------+--------+-------+

| stu\_name | c\_name | grade |

+----------+--------+-------+

| 貂蝉 | 中文 | 95 |

+----------+--------+-------+

1 row in set (0.00 sec)

（12）查询计算机成绩低于95的学生信息。

SQL语句如下：

mysql> SELECT stu\_name,sex,birth,addr,c\_name,grade

-> FROM student

-> LEFT JOIN score ON score.stu\_id = student.stu\_id

-> WHERE c\_name = "计算机" AND grade<95;

+----------+------+-------+--------------+--------+-------+

| stu\_name | sex | birth | addr | c\_name | grade |

+----------+------+-------+--------------+--------+-------+

| 关羽 | 男 | 1986 | 山西省运城市 | 计算机 | 65 |

| 刘备 | 男 | 1990 | 河北省涿州市 | 计算机 | 70 |

| 赵云 | 男 | 1988 | 河北省正定市 | 计算机 | 90 |

+----------+------+-------+--------------+--------+-------+

3 rows in set (0.00 sec)

（13）查询同时参加计算机和英语考试的学生的信息。

SQL语句如下：

mysql> SELECT \*

-> FROM student

-> WHERE stu\_id =ANY(

-> SELECT stu\_id

-> FROM score

-> WHERE stu\_id IN (SELECT stu\_id

-> FROM score WHERE c\_name= '计算机')

-> AND c\_name= '英语' );

+--------+----------+------+-------+------------+----------+

| stu\_id | stu\_name | sex | birth | department | addr |

+--------+----------+------+-------+------------+----------+

| 901 | 张飞 | 男 | 1985 | 计算机系 | 河北省涿州市 |

| 904 | 刘备 | 男 | 1990 | 英语系 | 河北省涿州市 |

| 906 | 赵云 | 男 | 1988 | 计算机系 | 河北省正定市 |

+--------+----------+------+-------+------------+----------+

3 rows in set (0.00 sec)

（14）从student表和score表中分别查询出学生的学号，然后合并查询结果。

SQL语句如下：

mysql> SELECT stu\_id FROM student

-> UNION

-> SELECT stu\_id FROM score;

+--------+

| stu\_id |

+--------+

| 901 |

| 902 |

| 903 |

| 904 |

| 905 |

| 906 |

+--------+

6 rows in set (0.00 sec)

（15）查询姓张或者姓王的同学的个人信息、院系和考试科目及成绩。

SQL语句如下：

mysql> SELECT student.stu\_id, stu\_name,sex,birth,

-> addr, c\_name,grade

-> FROM student, score

-> WHERE (stu\_name LIKE '张%' OR stu\_name LIKE '王%')

-> AND (student.stu\_id=score.stu\_id) ;

+--------+--------+------+-------+----------+--------+-------+

| stu\_id | stu\_name | sex | birth | addr | c\_name | grade |

+--------+--------+------+-------+----------+--------+-------+

| 901 | 张飞 | 男 | 1985 | 河北省涿州市 | 计算机 | 98 |

| 901 | 张飞 | 男 | 1985 | 河北省涿州市 | 英语 | 80 |

+--------+--------+------+-------+----------+--------+-------+

2 rows in set (0.00 sec)

（16）查询山西省的学生的姓名、出生年份、院系、考试科目及成绩。

SQL语句如下：

mysql> SELECT stu\_name,birth,department, c\_name,grade

-> FROM student, score

-> WHERE addr LIKE '山西%'

-> AND student.stu\_id=score.stu\_id;

+----------+-------+------------+--------+-------+

| stu\_name | birth | department | c\_name | grade |

+----------+-------+------------+--------+-------+

| 关羽 | 1986 | 中文系 | 计算机 | 65 |

| 关羽 | 1986 | 中文系 | 中文 | 88 |

| 貂蝉 | 1990 | 中文系 | 中文 | 95 |

+----------+-------+------------+--------+-------+

3 rows in set (0.00 sec)