

# 数据库实验报告 实验三、数据查询（单表查询）

姓名	学号	班级	课室
熊明	20305055	计科5班	D503

## 一、实验目的

熟悉 SQL 语句的数据查询语言，能够使用 SQL 语句对数据库进行单表查询。

## 二、实验环境

数据库：Mysql

图形化工具：Navicat Premium 16

## 三、实验内容

本节实验的主要内容包括：

- 查询的目标表达式为所有列、指定列或指定列的运算。
- 使用 DISTINCT 保留字消除重复行。
- 对查询结果排序和分组。
- 集合分组使用集函数进行各项统计。

## 四、课内实验

以 school 数据库为例(可在本实验单元的课时材料中下载)，在该数据库中存在 4 张表格，分别为：

STUDENTS(sid,sname,email,grade)

TEACHERS(tid,tname,email,salary)

COURSES(cid,cname,hour)

CHOICES(no,sid,tid,cid,score)

在数据库中，存在这样的关系：学生可以选择课程。一个课程对应一个教师。在表 CHOICES 中保存学生的选课记录。

1. 查询年级为 2001 的所有学生的名称，按编号升序排列。

```
1 SELECT sid,sname,grade
2 FROM students
3 WHERE grade = 2001
4 ORDER BY sid;
```

部分结果如下：

sid	sname	grade
800028044	ztozk	2001
800041569	pgmrkdhh	2001
800070739	nkdnfq	2001
800152632	qtxblqzsv	2001
800166448	ctvxn	2001
800169970	airnnfv	2001
800177146	vaesalave	2001
800202438	xiraegdlg	2001

2. 查询学生的选课成绩合格的课程成绩，并把成绩换算为积点(60 分对应积点为 1,每增加 1 分，积点增加 0.1)。

```

1 SELECT sid,score as pre_score,(score-60)*0.1+1 as fina_score
2 FROM choices
3 WHERE score > 60;

```

部分结果如下：

sid	pre_score	fina_score
823069829	76	2.6
829348273	87	3.7
847061074	92	4.2
860635914	82	3.2
829785562	77	2.7
822137137	67	1.7
826310502	90	4.0
801967882	70	2.0

3. 查询课时是 48 或 64 的课程名称。

```

1 SELECT cname,hour
2 FROM courses
3 WHERE hour = 48 OR hour = 64;

```

cname	hour
computer graphics	48
java	48
design pattern	48
real-time system	48
c	48
computer interface	48

4. 查询所有课程名称中含有 data 的课程编号。

```

1 SELECT cid,cname
2 FROM courses
3 WHERE cname LIKE '%data%';

```

cid	cname
10001	database
10008	data structure
10016	data mining
10048	data warehouse

5. 查询所有选课记录的课程号(不重复显示)。

```

1 SELECT distinct no
2 FROM choices;

```

部分结果如下：

no
500040525
500077492
500096659
500100279
500126184
500140784
500159131
500163876

6. 统计所有老师的平均工资。

一位老师只教一门课，所有老师的平均工资为：

```

1 SELECT AVG(salary) as avg_salary
2 FROM teachers;

```

avg_salary
2917.3280

7. 查询所有学生的编号，姓名和平均成绩，按总平均成绩降序排列

```

1 SELECT students.*,cc.avg_score
2 FROM students
3 LEFT OUTER JOIN(
4 SELECT sid,avg(any_value(score)) as avg_score
5 FROM choices
6 GROUP BY sid
7 ) as cc
8 on students.sid = cc.sid
9 ORDER BY cc.avg_score DESC;

```

MySQL对于GROUP BY聚合操作，如果在SELECT中的列，没有在GROUP BY中出现，那么这个SQL是不合法的，因为列不在GROUP BY语句中，也就是说查出来的列必须是GROUP BY之后的字段，或者这个字段出现在聚合函数里面。

解决方法就是除了group by的列其余都加any\_value函数。具体参考了以下博客：

[Mysql使用Group by之Error 1055之坑](#)

[MySQL 错误代码:1055 解决方案 \(推荐!!\)](#)

后面都用了any\_value的解决方法，就不在赘述。

因为是左外连接并降序排列，故最底下的都是null，但不意味着这些学生是零分，而是他们没有选课所以平均成绩是null。

部分结果如下：

sid	sname	email	grade	avg_score
896002290	jnkevd	9o0sa1z@hdgl.gov	2002	99.0000
896168867	ukiysvexw	fsi1@wtp.gov	1998	99.0000
896205003	dwgolzqc	u7c7@uwekg.org	1995	99.0000
896313795	rvhyf	qzzlu@ymi.edu	1992	99.0000
896383091	dmyjf	t7a7_z@hejfb.org	2002	99.0000
897265208	sdzfyn	agqr@arsx.com	2004	99.0000
897332193	jgqak	c8nxg@rze.gov	1994	99.0000
897502548	cyziuf	wzb8p5s@tku.gov	2001	99.0000

#### 8. 统计各个课程的选课人数和平均成绩

```
1 SELECT courses.cid,cname,cc.nums,cc.avg_score
2 FROM courses
3 LEFT OUTER JOIN(
4 SELECT cid,any_value(COUNT(cid)) as nums,avg(any_value(score)) as
  avg_score
5 FROM choices
6 GROUP BY cid) as cc
7 on courses.cid = cc.cid;
```

部分结果如下：

cid	cname	nums	avg_score
10001	database	5898	75.9724
10002	operating system	6013	75.8944
10003	computer graphics	5975	75.9430
10004	java	6110	76.1287
10005	c++	6031	76.0171
10006	design pattern	6090	76.0700
10007	uml	5965	75.9119
10008	data structure	5985	75.6591

#### 9. 查询至少选修了三门课程的学生编号。

```

1 SELECT *
2 FROM
3 (SELECT sid,any_value(COUNT(cid)) as cid_nums
4 FROM choices
5 GROUP BY sid) as cc
6 WHERE cc.cid_nums >= 3;

```

部分结果如下：

sid	cid_nums
800001216	3
800002933	4
800006682	3
800006941	5
800007595	3
800009026	3
800009099	3
800009249	3

自我检测：

#### 1. 查询全部课程的详细记录

```

1 SELECT *
2 FROM courses;

```

部分结果如下：

cid	cname	hour
10001	database	96
10002	operating system	88
10003	computer graphics	48
10004	java	48
10005	c++	60
10006	design pattern	48
10007	uml	30

#### 2. 查询所有有选修课的学生的编号；

```

1 SELECT DISTINCT sid
2 FROM choices;

```

部分结果如下：

sid
800001216
800002933
800005753
800006682
800006941
800007595
800008565

3. 查询课时<88(小时)的课程编号;

```
1 SELECT cid, hour
2 FROM courses
3 WHERE hour < 88;
```

部分结果如下:

cid	hour
10003	48
10004	48
10005	60
10006	48
10007	30
10008	60
10009	36

4. 请找出总分超过 400 分的学生

```
1 SELECT *
2 FROM
3 (SELECT sid, sum(any_value(score)) as score_sum
4 FROM choices
5 GROUP BY sid) as cc
6 WHERE cc.score_sum > 400;
```

部分结果如下:

sid	score_sum
800031798	411
800034166	407
800087520	408
800215306	404
800216402	413
800233504	417
800249234	403

5. 查询课程的总数;

```

1 SELECT SUM(cc.courses_count) as courses_sum
2 FROM (
3 SELECT COUNT(courses.cid) as courses_count
4 FROM courses
5 ) as cc;

```

courses_sum
50

6. 查询所有课程和选修该课程的学生总数;

```

1 SELECT courses.cid,courses.cname,cc.student_nums
2 FROM courses
3 LEFT OUTER JOIN
4 (SELECT cid,count(any_value(sid)) as student_nums
5 FROM choices
6 GROUP BY cid) as cc
7 on courses.cid = cc.cid;

```

部分结果如下:

cid	cname	student_nums
10001	database	5898
10002	operating system	6013
10003	computer graphics	5975
10004	java	6110
10005	c++	6031
10006	design pattern	6090
10007	uml	5965

7. 查询选修成绩合格的课程超过两门的学生编号。

```

1 SELECT *
2 FROM
3 (SELECT sid,count(any_value(score)) as count
4 FROM choices
5 WHERE score >= 60
6 GROUP BY sid) as cc
7 WHERE cc.count > 2;

```

部分结果如下:

sid	count
800001216	3
800002933	3
800006682	3
800006941	3
800007595	3
800013889	4
800016416	3

#### 8. 统计各个学生的选修课程数目和平均成绩

```

1 SELECT students.sid,sname,cc.count,cc.avg_score
2 FROM students
3 LEFT OUTER JOIN
4 (SELECT sid,count(any_value(score)) as count,avg(any_value(score)) as
   avg_score
5 FROM choices
6 GROUP BY sid) as cc
7 on students.sid = cc.sid;

```

部分结果如下：

sid	sname	count	avg_score
800001216	gfxrgs	3	62.3333
800002933	vnbqz%svv	3	73.6667
800005753	waqcj	1	66.0000
800006682	fiiluommh	3	89.0000
800006941	ogvmu	4	71.0000
800007595	uxqqbkjn	3	74.0000
800008565	ehlycg	1	76.0000