基本语法

[WHERE<条件表达式>]

[GROUP BY<列名1>[HAVING<条件表达式>]]

[ORDER BY<列名2>[ASC | DESC]];

[limit N,M];

1.当用户需要查询数据库中符合一定条件的数据时，可以使用WHERE子句对表中的记录进行筛选:

WHERE子句中可以使用多种条件判断符:

|  |  |
| --- | --- |
| **条件判断符** | **说  明** |
| **=** | 相  等 |
| **<** | 小  于 |
| **>** | 大  于 |
| **<>（!=）** | 不相等 |
| **<=** | 小于或者等于 |
| **>=** | 大于或者等于 |
| BETWEEN value1 AND value2 | 在value1和value2之间(包括value1和value2） |
| not BETWEEN value1 AND value2 | 小于value1或者大于value2 |

[例]查询出成绩大于等于90的同学学号

Select studenid from grade where grade>=90;

2.多条件查询

[例]查询成绩大于等于60分，小于等于90的同学学号

select distinct studentid from grade where grade>=60 and grade<=90;

MySQL支持多条件查询：

* 如果条件之间使用AND关键字连接，那么只有符合所有条件的记录才会被返回。
* 如果多条件查询中的条件使用OR关键字连接，表示只需要符合所有条件中的一个条件，此记录就会被返回。
* 使用IN关键字可以查询字段值等于指定集合中任意一个值的记录

[例]查询选修了Dp010001或者Dp010002课程的同学学号

select distinct courseid,studentid

from grade

where courseid in ('Dp010001','Dp010002');

[学习通1]查询没有选修Dp010001或者Dp010002课程的同学学号

select distinct courseid,studentid

from grade

where courseid not in ('Dp010001','Dp010002');

[课堂练习]查找Cs010901班和Cs010902班的女同学

select studentid,sex,classID from student where (classID = 'cs010901' or classID = 'cs010902' ) and sex = '女';

3.查询空值

[例]查找没有班长的班级号

select classid,monitor

from class

where monitor is null;

MySQL提供IS NULL关键字，用于查询字段值为NULL的记录

**不能对NULL使用比较运算符**（=、<>、>、<、>=、<=）例如：...where price = NULL，是查询不到数据的，可以使用IS NULL，反之，希望选取不是NULL 的记录时，需要使用IS NOT NULL

[学习通2]：查找有班长的班级号和班级人数

select classid,studentnum,monitor

from class

where monitor is not null;

4.模糊查询

[例]查找大连市的同学学号和姓名

select studentid,studentname,homeaddr

from student

where HomeAddr like '大连市%';

谓词LIKE可以用来进行字符串的匹配。其一般语法格式如下：

[NOT] LIKE'<匹配串>' [ESCAPE '<换码字符>']

其含义是查找指定的属性列值与<匹配串>相匹配的元组。

<匹配串>可以是一个完整的字符串，也可以含有通配符 % 和 \_ 。其中：

\* %（百分号）代表任意长度（长度可以为0）的字符串。

    例如：a%b 表示以a开头，以b结尾的任意长度的字符串。如acb、addgb、ab等。

\* \_（下划线）代表任意单个字符。

    例如：a\_b表示以a开头，以b结尾的长度为3的任意字符串。如acb、agb等。

[学习通3]查找姓名第二个字为雨的同学

select studentid,studentname

from student

where studentname like '%\_雨\_%';

[例]查找书名中以“数据库技术与应用\_”开头的书

select courseid,bookname

from course

where bookname like '%数据库技术与应用\_%';

5 排序

查询选修了Dp010001且成绩为前三名的同学的学号及对应的成绩。

select courseid,studentid,grade

from grade

where courseid = 'Dp010001'

order by grade desc limit 3;

普通查询查询到的结果是按照记录在表中的默认顺序进行排列的。如果需要将查询结果按照指定的顺序排列，可以使用ORDER BY关键字，语法形式如下：

SELECT {\*|col\_list} FROM table\_name ORDER BY col\_namea [ASC|DESC];

6.限制结果显示数量

LIMIT关键字来限制查询结果的数量。语法形式如下：

SELECT {\*|col\_list} FROM table\_name LIMIT [offset\_start,]row\_count;

[练习]查询选修了Dp010001且成绩为第三名的同学的学号及对应的成绩。

select courseid,studentid,grade

from grade

where courseid = 'Dp010001'

order by grade desc limit 1;

【练习】查询全体学生情况，查询结果按所在系的系号升序排列，同一系中的学生按年龄降序排列。

select studentid,classid,year(birth1),2020-year(birth1)

from student

order by classid,2020-year(birth1) desc;

7聚合函数与分组

查询总共有多少个班级？查询有班长的班级有多少个？通过student表查询每个班级各有多少个人？

实际应用中，聚合函数通常与分组查询一起使用。分组查询就是按照某个字段对数据记录进行分组，比如实验中的商品表，可以按照商品类别对记录进行分组，然后使用聚合函数统计每个类别下的商品数量。

SELECT {\*|col\_list} aggregate\_func FROM table\_name

GROUP BY col\_namea

[学习通]查询每门课程的平均分

select courseid,avg(grade)

from grade

group by courseid;

[例]查询每个班中各年份出生的人数是多少？

select classid as '班级号',year(birth1) as '出生年月',count(\*) as '人数'

from student

group by classid,year(birth1)

order by classid,year(birth1);

结果如图：



[例]查询平均成绩大于70分的课程号

select courseid,avg(grade)

from grade

group by courseid

having avg(grade)>70;

[例]查询平均成绩大于70分的课程（只有及格的同学（大于等于60分)才参与平均成绩计算）

select courseid,avg(grade)

from grade

where grade >= 60

group by courseid

having avg(grade)>70

order by avg(grade) desc;

[例]统计每门课程的学生选修人数（超过2人的课程才统计）。要求输出课程号和选修人数，查询结果按人数降序排列，若人数相同，按课程号升序排列

select courseid as '课程号',count(studentid)

from grade

group by courseid

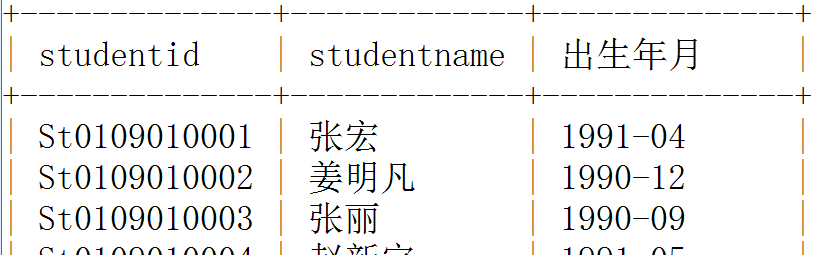
order by count(studentid) asc

实验题：

1.查询所有课程的详细信息。

select \*from course

2.查询所有学生的学号，姓名，出生年月(格式YYYY-MM)，显示时使用别名出生年月，如图。



select studentid,studentname,DATE\_FORMAT(birth1,'%Y-%c') as '出生年月'

from student

3.求姓王的同学和姓张的同学学生号，学生名，班级号

select studentid,studentname,classid

from student

where studentname like '王%' or studentname like '张%'



4.查询每门课程每个学年的平均分，输出课程号，学年，以及平均分数

select courseid,avg(grade)

from grade

group by coursed



5.查询每个学生选择的课程数并显示出这几门课的课程号，以及他的最高分，最低分

select max(grade) as '最高分',min(grade) as '最低分',courseid as '课程号',count(courseid) as '人数'

from grade

group by courseid



6.通过student表计算每个班有多少人，找出大于5人的班级号和其班级人数，并按照班级人数降序排序显示。

select count(studentid) as 班级人数

from student

group by classid

having count(studentid)>5

order by count(studentid) desc



7.筛选出每个同学大于70分的成绩，输出有2门课超过70分的学生号

select studentid

from grade

group by studentid

having count(grade>70)>=2



8.查询Cs010901和Cs010902 班同学的学号

select classid,studentid

from student

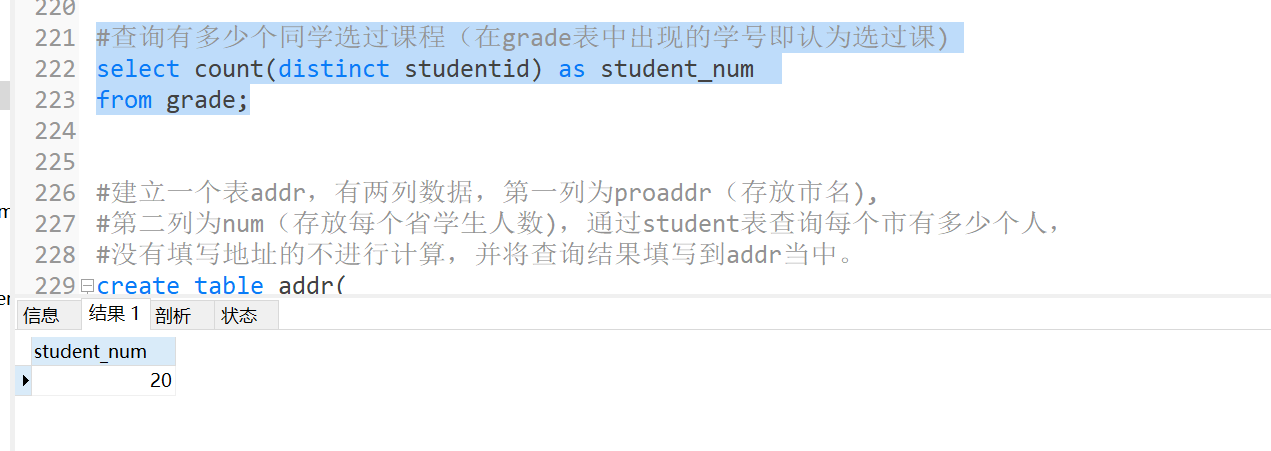
where classid in ('Cs010901','Cs010902')



1. 查询有多少个同学选过课程（在grade表中出现的学号即认为选过课)

select count(distinct studentid) as student\_num

from grade;



1. 建立一个表addr，有两列数据，第一列为proaddr（存放省市名),第二列为num（存放每个省学生人数)，通过student表查询每个市有多少个人，没有填写地址的不进行计算，并将查询结果填写到addr当中。

create table addr(

proaddr varchar(80) comment '省市名',

num int comment '人数'

);

drop table addr;

insert into addr(num,proaddr)

select count(HomeAddr),substring(HomeAddr,1,position('市' in HomeAddr)) as city

from student

where HomeAddr regexp '(佛山市)|(武汉市)|(成都市)|(广州市)|(鞍山市)|(宜宾市)|(大连市)|(汕头市)|(上海市)|(深圳市)'

group by city

select \* from addr;

