

```
create table mydb1  
(id char(4) not null,  
  sname char(32),  
  deptid char(32));
```

增加名称必须取唯一值的约束条件

```
ALTER TABLE mydb1  
  ADD Constraint UQ_name UNIQUE (sname) ;
```

添加表的主键约束：

```
Alter table mydb1 Add Constraint PK_id primary  
key (id)
```

添加表的外约束：

```
Alter table mydb1 Add Constraint FK_deptid  
foreign key (deptid) references Dept (deptid)
```

```
Create table myTB1 (  
id nvarchar(4) not null,  
name nvarchar(32),  
Sex nvarchar(4),  
deptid nvarchar(32))
```

添加表的Check约束:

```
Alter table myTB1 Add Constraint CK_sex  
Check (sex='男' or sex='女');
```

```
Create table myTB1 (  
id nvarchar(4) not null,  
name nvarchar(32),  
Sex nvarchar(4),  
deptid nvarchar(32))
```

添加表的Default约束:

```
Alter table myTB1 add Constraint DF_sex Default ('F')  
for sex
```

删除表的操作语句格式为：

DROP TABLE <表名> [RESTRICT|CASCADE];

MySQL的索引

- 单列索引(普通索引，唯一索引，主键索引)、组合索引、全文索引引擎为MyISAM、空间索引MyISAM

`explain select * from student where sno='1';`
查看是否使用索引

3.3.1 单表查询

- 查询各门课程的选课人数（聚集，分组）
- 查询选修了2门以上课程的学生学号。（分组，条件）
- 查询选修了2号课程的学生学号和成绩，查询结果按成绩从大到小排列（排序）
- 查询学号在1-4之间至少选修了两门课程的学生学号及其选修的课程数,并对课程数降序排序

3.3.2 连接查询

1) 等值与非等值连接查询

[〈表名1〉.] 〈列名〉 〈比较运算符〉 [〈表名2〉.]〈列名〉

2) 自身连接: 一个表与其自己进行连接。

3) 外连接:

右外连接right join : = (*)

左外连接left join : (*) =

4) 复合条件连接: WHERE 子句中可以有多个连接条件。

5) 集合运算

3.3.2 连接查询

- 查询选修了课程的学生及其选修的课程的编号及成绩
- 查询选修了数据库原理课程的学生学号
- 查询每一门课的间接先修课（即先修课的先修课）
- 求选修了2号学生选修的课程的学生学号
- 求同时选修了1号和2号课程的学生学号（自连接）
- 求所有学生的选课的课程号和相应成绩（外连接）
- 求选修了1号课程或2号课程的学生学号（并运算）
- 求同时选修了1号和2号课程的学生学号（交运算）
- 求选修了1号没有选修2号课程的学生学号（差运算）

集合查询

```
SELECT Sno  
FROM SC  
WHERE Cno="1"  
UNION  
SELECT Sno  
FROM SC  
WHERE Cno="2";
```



```
SELECT Sno  
FROM SC  
WHERE Cno="1"
```

INTERSECT

```
SELECT Sno  
FROM SC  
WHERE Cno="2";
```



```
SELECT Sno, Cno, Grade  
FROM SC  
WHERE Cno='1'
```

EXCEPT

```
SELECT Sno, Cno, Grade  
FROM SC  
WHERE Cno='2';
```



3.3.3 嵌套查询

嵌套查询：一个查询块(**SELECT—FROM—WHERE**)嵌套在另一个查询块的条件中。

外层查询（父查询） 内层查询（子查询）

- 1) 带有**IN**谓词的子查询
- 2) 带有比较运算符的子查询
- 3) 带有**ANY**或**ALL**谓词的子查询（使用**ANY**和**ALL**时必须同时使用比较运算符）
- 4) 带有**EXISTS**谓词的子查询（查询结果不返回任何数据，只产生逻辑“真”或“假”）

一、带有IN谓词的子查询

查询与刘晨在同一个学院学习的学生

```
SELECT Student.Sno, Sname  
FROM Student  
WHERE Sdept IN  
( SELECT Sdept  
FROM Student  
WHERE Sname='刘晨' );
```

不相关
子查询

查询选修了数据库课程的学生学号和姓名

```
SELECT Sno, Sname
FROM Student
WHERE Sno IN
    (SELECT Sno
     FROM SC
     WHERE Cno IN
         ( SELECT Cno
           FROM Course
           WHERE Cname='数据库' ));
```

查询同时选修了1号和2号课程的学生学号

交运算的表达方式

```
select Sno
from SC
where Cno='1'
and Sno in(
    select Sno
    from SC
    where Cno='2')
```



查询选修了1号课程而没有选修2号课程的学生
学号

```
select Sno
from SC
where Cno='1'
and Sno NOT IN (
    select Sno
    from SC
    where Cno='2')
```



二、带有比较运算符的子查询

带有比较运算符的子查询是指父查询与子查询之间用比较运算符进行连接。当用户能**确切**知道内层查询返回的是单值时，可以用>,<,<=,>=,!=等比较运算符。

找出每个学生超过他选修课程平均成绩的课程号

```
SELECT Sno,Cno
FROM SC x
WHERE Grade >=
  (SELECT AVG(Grade)
   FROM SC y
   WHERE y.Sno=x.Sno)
```



相关子查询

三、带有ANY或ALL的子查询

子查询返回单值时，可以用比较运算符，但返回多值时，要用ANY或ALL谓词修饰符。使用ANY或ALL谓词时，必须同时使用比较运算符。

查询比计算机学院的学生年龄都小的
学生姓名和年龄

```
SELECT  Sname, Sage
FROM    Student
WHERE   Sage < ALL
        (SELECT  Sage
         FROM    Student
         WHERE   Sdept='计算机' )
```

查询比计算机学院的学生年龄都小的
学生姓名和年龄

```
SELECT  Sname, Sage
FROM    Student
WHERE   Sage<
        (SELECT  MIN(Sage)
         FROM    Student
         WHERE   Sdept='计算机')
```

	=	<>	<	<=	>	>=
ANY	IN	--	<MAX	<=MAX	>MIN	>=MIN
ALL	--	NOT IN	<MIN	<=MIN	>MAX	>=MAX

3.3.3 嵌套查询

不相关子查询

- 查询与刘晨在同一个学院学习的学生 (IN, 自连接)
- 查询选修了数据库课程的学生学号和姓名 (IN, 连接)
- 求同时选修了1号和2号课程的学生学号 (IN交运算, 自连接、除法)
- 求选修了1号没有选修2号课程的学生学号 (IN差运算)

相关子查询

- 找出每个学生超过他选修课程平均成绩的课程号
- 查询比计算机学院的学生年龄都小的学生姓名和年龄 (ANY、ALL、MIN、MAX)

四、帶有EXISTS的子查詢

EXISTS代表存在量詞，帶有**EXISTS**的子查詢不返回任何數據，只產生邏輯真值“true”或邏輯假值“false”

不同形式的查询间的替换

- 一些带**EXISTS**或**NOT EXISTS**谓词的子查询不能被其他形式的子查询等价替换
- 所有带**IN**谓词、比较运算符、**ANY**和**ALL**谓词的子查询都能用带**EXISTS**谓词的子查询等价替换。

查询选修了2号课程的学生姓名和学号

```
SELECT  Sname,Sno  
FROM Student , SC  
WHERE Student.Sno = SC.Sno  
And Cno='2';
```

连接查询

```
SELECT  Sname,Sno  
FROM Student  
WHERE Sno IN  
      (SELECT Sno FROM SC  
       WHERE Cno='2');
```

IN嵌套
查询

EXISTS相关 子查询

```
SELECT   Sname,Sno  
FROM     Student  
WHERE Exists  
      (SELECT *  
        FROM SC  
        WHERE Sno=Student.Sno  
        AND   Cno='2');
```

查询选修了数据库原理课程的学生学号和姓名

```
SELECT Sno, Sname
FROM Student
WHERE Sno IN
      (SELECT Sno
       FROM SC
       WHERE Cno IN
            ( SELECT Cno
              FROM Course
              WHERE Cname='数据库' ));
```

IN嵌套
查询

```
SELECT    Sname,Student.Sno  
FROM      Student,SC  
WHERE     Student.Sno=SC.Sno  
AND       Exists  
    (SELECT *  
      FROM   Course  
      WHERE  Cno = SC.Cno  
      AND    Cname='数据库');
```

求同时选修了1号和2号课程的学生学号

```
select Sno
from SC
where Cno='1'
and Sno in(
    select Sno
    from SC
    where Cno='2')
```

求同时选修了1号和2号课程的学生学号

交运算用exists如何表达？

```
select Sno
from SC as A
where Cno='1'
and exists (
    select *
    from SC as B
    where Cno='2'
    and A.Sno = B.Sno)
```



敲黑板，重点到了

查询没有选修2号课程的学生姓名和学号

```
SELECT  Sname,Sno  
FROM Student  
WHERE  Sno NOT IN  
      (SELECT Sno FROM SC  
       WHERE Cno='2');
```

IN嵌套
查询

查询没有选修2号课程的学生姓名和学号

差运算用exists如何表达？

```
SELECT  Sname,Sno  
FROM Student  
WHERE NOT EXISTS  
    (SELECT *  
      FROM SC  
      WHERE Sno=Student.Sno  
      AND  Cno='2');
```



敲黑板，重点到了