

数据库原理

第4章 SQL --SELECT

辽东学院 鲁 琴

本节要点

数据库基础概念

数据库原理

数据库新技术

关系数据库

关系数据模型

关系数据语言

数据库设计

数据库管理

关系代数

SQL

DDL

DML

DCL

INDEX

VIEW

嵌入式SQL

SELECT

INSERT

UPDATE

DELETE

查询语句格式

Select A_1, A_2, \dots, A_n
From R_1, R_2, \dots, R_m
Where **condition**
Group By **columns**
Having **condition**
Order by **columns**

两者不同:

1.SELECT是**等值连接**

在 condition中要有等值连接条件

2.SELECT结果不是集合是包, **有重复元组**

$\pi_{A_1, A_2, \dots, A_n} (\sigma_{\text{condition}} (R_1 \bowtie R_2 \dots \bowtie R_m))$

示例数据库

teach数据库

◆ 学生表:

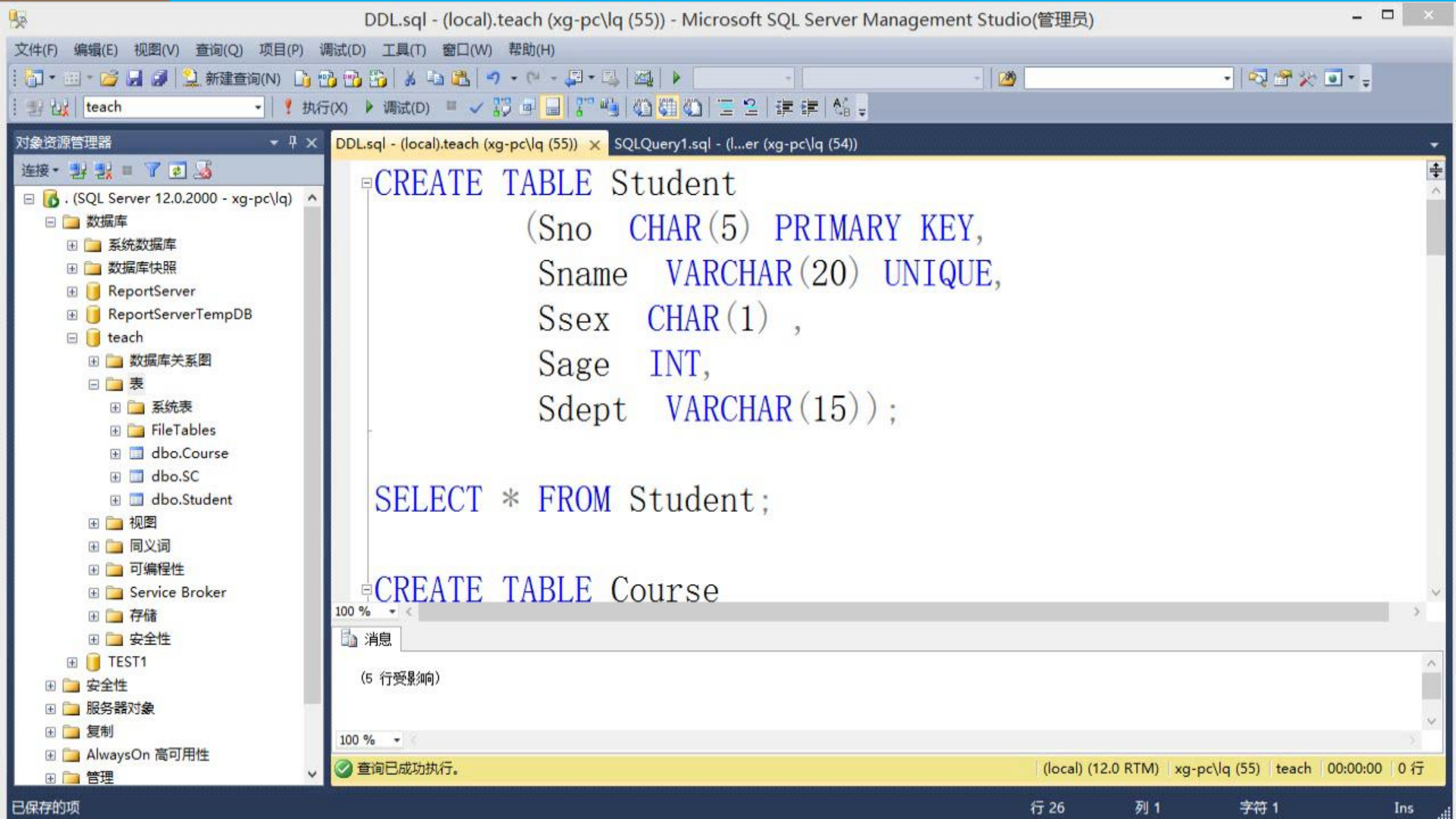
Student(Sno, Sname, Ssex, Sage, Sdept)

◆ 课程表:

Course(Cno, Cname, Cpno, Ccredit)

◆ 学生选课表:

SC(Sno, Cno, Grade)



数据查询

- 1 单表查询
- 2 连接查询
- 3 嵌套查询
- 4 集合查询

2 连接查询

- 同时涉及多个表的查询称为**连接查询**
- 用来连接两个表的条件称为**连接条件**或**连接谓词**
- SQL中连接查询的主要类型
 - (1) 交叉连接（广义笛卡尔积）
 - (2) 等值连接
 - (3) 自身连接
 - (4) 复合条件连接
 - (5) 内连接
 - (6) 外连接

(1) 交叉连接 (广义笛卡尔积)

- 不带连接谓词的连接
- 很少使用

SELECT Student.*, SC.* FROM Student, SC

Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
95001	李勇	M	20	CS	95001	1	92.0
95001	李勇	M	20	CS	95001	2	85.0
95001	李勇	M	20	CS	95001	3	88.0
95001	李勇	M	20	CS	95002	2	90.0
95001	李勇	M	20	CS	95002	3	80.0
95002	刘晨	F	19	IS	95001	1	92.0
95002	刘晨	F	19	IS	95001	2	85.0

功执行。

(local) (12.0 RTM) | xg-pc\lq (54) | teach | 00:00:00

20 行

交叉连接（广义笛卡尔积）

- 很少使用

SELECT Student.*, SC.* FROM Student CROSS JOIN SC

Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
95001	李勇	M	20	CS	95001	1	92.0
95001	李勇	M	20	CS	95001	2	85.0
95001	李勇	M	20	CS	95001	3	88.0
95001	李勇	M	20	CS	95002	2	90.0
95001	李勇	M	20	CS	95002	3	80.0
95002	刘晨	F	19	IS	95001	1	92.0
95002	刘晨	F	19	IS	95001	2	85.0

成功执行。

(local) (12.0 RTM) | xg-pc\lq (54) | teach | 00:00:00 | 20 行

(2) 等值连接

— 连接运算符为 = 的连接操作

- [\langle 表名1 \rangle .] \langle 列名1 \rangle = [\langle 表名2 \rangle .] \langle 列名2 \rangle

— 表名前缀

- 任何子句中引用表1和表2中同名属性时，都必须加表名前缀
- 引用唯一属性名时可以加也可以省略表名前缀

等值连接

[例32] 查询每个学生及其选修课程的情况。

```
SELECT Student.*,SC.* FROM Student,SC  
WHERE Student.Sno=SC.Sno;
```

Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
95001	李勇	M	20	CS	95001	1	92.0
95001	李勇	M	20	CS	95001	2	85.0
95001	李勇	M	20	CS	95001	3	88.0
95002	刘晨	F	19	IS	95002	2	90.0
95002	刘晨	F	19	IS	95002	3	80.0

成功执行。

(local) (12.0 RTM) | xg-pc\lq (54) | teach | 00:00:00

5 行

例： 查询计算机系（CS）学生的学号，姓名，所在系，选修的课程号，课程名和成绩

```
SELECT SC.Sno,Sname,Sdept,SC.Cno,Cname,Grade
FROM Student,SC,Course
WHERE Student.Sno=SC.Sno and Course.Cno=SC.Cno
and Sdept='CS';
```

Sno	Sname	Sdept	cno	Cname	Grade
95001	李勇	CS	1	数据库	92.0
95001	李勇	CS	2	数学	85.0
95001	李勇	CS	3	信息系统	88.0

(3) 自身连接

- 一个表与其自己进行连接，称为表的自身连接
- 表示方法
 - 需要给表起别名以示区别
 - 由于所有属性名都是同名属性，因此必须使用别名前缀

自身连接

[例34] 查询每一门课的直接先修课的课程名

SELECT a1.Cname,a2.Cname

FROM Course **a1**,Course **a2**

WHERE a1.Cpno=a2.Cno

Cname	Cname
数据库	数据结构
信息系统	数据库
操作系统	数据处理
数据结构	C语言
C语言	数据处理

a1

Cno	Cname	Cpno	Ccredit	Cno	Cname	Cpno	Ccredit
1	数据库	5	4	1	数据库	5	4
2	数学	NULL	2	2	数学	NULL	2
3	信息系统	1	4	3	信息系统	1	4
4	操作系统	6	3	4	操作系统	6	3
5	数据结构	7	4	5	数据结构	7	4
6	数据处理	NULL	2	6	数据处理	NULL	2
7	C语言	6	4	7	C语言	6	4

a2

自身连接

[例35] 查询每一门课的间接先修课的课程号（即先修课的先修课）

```
SELECT a1.Cno, a2.Cpno
FROM Course a1, Course a2
WHERE a1.Cpno = a2.Cno;
```

Cno	Cpno
1	7
3	5
4	NULL
5	6
7	NULL

a1

Cno	Cname	Cpno	Ccredit	Cno	Cname	Cpno	Ccredit
1	数据库	5	4	1	数据库	5	4
2	数学	NULL	2	2	数学	NULL	2
3	信息系统	1	4	3	信息系统	1	4
4	操作系统	6	3	4	操作系统	6	3
5	数据结构	7	4	5	数据结构	7	4
6	数据处理	NULL	2	6	数据处理	NULL	2
7	C语言	6	4	7	C语言	6	4

a2

自身连接

[例36] 查询同时选修2号课程
和3号课程学生的学号

```
SELECT a.Sno
FROM SC a,SC b
WHERE a.Sno=b.Sno
      and a.Cno<>b.Cno
      and a.Cno=2
      and b.Cno=3;
```

Sno	Cno	Grade	Sno	Cno	Grade
95001	1	92.0	95001	2	85.0
95001	1	92.0	95001	3	88.0
95001	2	85.0	95001	1	92.0
95001	2	85.0	95001	3	88.0
95001	3	88.0	95001	1	92.0
95001	3	88.0	95001	2	85.0
95002	2	90.0	95002	3	80.0
95002	3	80.0	95002	2	90.0

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

(4) 复合条件连接

- WHERE子句中含多个连接条件时，称为复合条件连接
- 复合条件连接的类型
 - 两表按多个属性连接
 - 自身按多个属性连接
 - 多表连接

复合条件连接[例37]

- 查询选修1号课程且成绩在90分以上的所有学生的姓名, 性别及所在系

SELECT Sname, Ssex, Sdept

FROM Student,SC

WHERE Student.Sno = SC.Sno /* 连接谓词 */

and Cno= 1 /* 其他限定条件 */

and Grade>=90 /* 其他限定条件 */

Sname	Ssex	Sdept
李勇	M	CS

复合条件连接[例38]

- 查询每个学生的学号、姓名、选修的课程名及成绩

```
SELECT Student.Sno,Sname,Cname,Grade  
FROM Student,SC,Course  
WHERE Student.Sno = SC.Sno  
and SC.Cno = Course.Cno;
```

Sno	Sname	Cname	Grade
95001	李勇	数据库	92.0
95001	李勇	数学	85.0
95001	李勇	信息系统	88.0
95002	刘晨	数学	90.0
95002	刘晨	信息系统	80.0

(5) 内连接

- 查询所有的学生信息及其选修课程情况

SELECT Student.*,SC.*

FROM Student,SC

WHERE Student.Sno=SC.Sno;

SELECT Student.*,SC.* FROM Student JOIN SC
ON Student.Sno=SC.Sno;

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	M	20	CS
95002	刘晨	F	19	IS
95003	王名	F	18	MA
95004	张立	M	18	IS

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
95001	李勇	M	20	CS	95001	1	92.0
95001	李勇	M	20	CS	95001	2	85.0
95001	李勇	M	20	CS	95001	3	88.0
95002	刘晨	F	19	IS	95002	2	90.0
95002	刘晨	F	19	IS	95002	3	80.0

(6) 外连接

- 左连接 (**LEFT JOIN**) 或左外连接 (**LEFT OUTER JOIN**)
- 右连接 (**RIGHT JOIN**) 或右外连接 (**RIGHT OUTER JOIN**)
- 全连接 (**FULL JOIN**) 或全外连接 (**FULL OUTER JOIN**)

左连接

- 返回左表中的所有行，如果左表中行在右表中没有匹配行，则结果中右表中的列返回空值

```
SELECT Student.*,SC.*  
FROM Student LEFT JOIN SC  
ON Student.Sno=SC.Sno;
```

Sno	Sname	Ssex	Sage	Sdept	Sno	Cno	Grade
95001	李勇	M	20	CS	95001	1	92.0
95001	李勇	M	20	CS	95001	2	85.0
95001	李勇	M	20	CS	95001	3	88.0
95002	刘晨	F	19	IS	95002	2	90.0
95002	刘晨	F	19	IS	95002	3	80.0
95003	王名	F	18	MA	NULL	NULL	NULL
95004	张立	M	18	IS	NULL	NULL	NULL

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	M	20	CS
95002	刘晨	F	19	IS
95003	王名	F	18	MA
95004	张立	M	18	IS

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

右连接

- 返回右表中的所有行，如果右表中行在左表中没有匹配行，则结果中左表中的列返回空值

```
SELECT SC.*,Course.*  
FROM SC RIGHT JOIN Course  
ON SC.Cno=Course.Cno;
```

Sno	Cno	Grade	Cno	Cname	Cpno	Ccredit
95001	1	92.0	1	数据库	5	4
95001	2	85.0	2	数学	NULL	2
95002	2	90.0	2	数学	NULL	2
95001	3	88.0	3	信息系统	1	4
95002	3	80.0	3	信息系统	1	4
NULL	NULL	NULL	4	操作系统	6	3
NULL	NULL	NULL	5	数据结构	7	4
NULL	NULL	NULL	6	数据处理	NULL	2

Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学	NULL	2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理	NULL	2
7	C语言	6	4

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

全连接

- 返回左表和右表中的所有行。当某行在另一表中没有匹配行，则另一表中的列返回空值

```
SELECT SC.*,Course.*  
FROM SC FULL JOIN Course  
ON SC.Cno=Course.Cno;
```

Sno	Cno	Grade	Cno	Cname	Cpno	Ccredit
95001	1	92.0	1	数据库	5	4
95001	2	85.0	2	数学	NULL	2
95002	2	90.0	2	数学	NULL	2
95001	3	88.0	3	信息系统	1	4
95002	3	80.0	3	信息系统	1	4
NULL	NULL	NULL	4	操作系统	6	3
NULL	NULL	NULL	5	数据结构	7	4
NULL	NULL	NULL	6	数据处理	NULL	2

Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学	NULL	2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理	NULL	2
7	C语言	6	4

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

数据查询

- 1 单表查询
- 2 连接查询
- 3 嵌套查询
- 4 集合查询

嵌套查询概述

- 查询块
 - 一个SELECT语句称为一个查询块
- 嵌套查询
 - 将一个查询块嵌套在另一个查询块的**WHERE**子句或**HAVING**短语的条件中的查询称为嵌套查询
 - **WHERE** condition
 - **HAVING** condition

例. 查询选修2号课程的学生姓名

● **SELECT Sname**

外层查询/父查询

FROM Student

WHERE Sno IN

(SELECT Sno

内层查询/子查询

FROM SC

WHERE Cno = 2);

不相关子查询

由里向外逐层处理。即每个子查询在上一级查询处理之前求解，子查询的结果用于建立其父查询的查找条件。

例. 查询选修2号课程的学生姓名 (方法2)

● **SELECT Sname**

外层查询/父查询

FROM Student

WHERE EXISTS

(SELECT *

内层查询/子查询

FROM SC

相关子查询

WHERE Cno = 2 AND Sno = Student.Sno);

首先取外层查询中表的第一个元组，根据它与内层查询相关的属性值处理内层查询，若WHERE子句返回值为真，则取此元组放入结果表；然后再取外层表的下一个元组；重复这一过程，直至外层表全部检查完为止。

子查询的限制

- ◆ 不能使用**ORDER BY**子句
- ◆ 层层嵌套方式反映了 SQL语言的结构化
- ◆ 有些嵌套查询可以用连接运算替代

嵌套查询分类

– 不相关子查询

- 子查询的查询条件不依赖于父查询

– 相关子查询

- 子查询的查询条件依赖于父查询

嵌套查询的谓词

- (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
- (4) 带有**EXISTS**谓词的子查询

(1) 带有IN谓词的子查询

- 父查询和子查询之间用**IN**进行连接，子查询的结果是一个集合

[例37] 查询与“刘晨”在同一个系学习的学生

- ① 确定“刘晨”所在系名 ② 查找所有在IS系学习的学生

```
SELECT Sdept
FROM Student
WHERE Sname= '刘晨';
```

Sdept
IS

```
SELECT Sno,Sname,Sdept
FROM Student
WHERE Sdept= 'IS';
```

Sno	Sname	Sdept
95002	刘晨	IS
95004	张立	IS

构造带有IN谓词的子查询

- ✓ 将第一步查询嵌入到第二步查询的条件中

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



不相关子查询

带有IN谓词的子查询（续）

用自身连接完成上面的查询要求

--查询与“刘晨”在同一个系学习的学生

```
SELECT S1.Sno,S1.Sname,S1.Sdept
```

```
FROM Student S1,Student S2
```

```
WHERE S1.Sdept = S2.Sdept and S2.Sname = '刘晨';
```

Sno	Sname	Sdept
95002	刘晨	IS
95004	张立	IS

[例38] 查询选修了课程名为“信息系统”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= '信息系统'));

③ 最后在Student关系中

取出Sno和Sname

② 然后在SC关系中找到选

修了3号课程的学生学号

① 首先在Course关系中找到“信

息系统”的课程号，结果为3号

Sno	Sname
95001	李勇
95002	刘晨

带有IN谓词的子查询（续）

等价的连接查询

SELECT SC.Sno,Sname

FROM Student,SC,Course

WHERE Student.Sno = SC.Sno and

SC.Cno = Course.Cno and Cname='信息系统';

Sno	Sname
95001	李勇
95002	刘晨

(2) 带有比较运算符的子查询

- 使用范围

- 当能确切知道内层查询返回单值时，可用比较运算符（>，<，=，>=，<=，!=，<>，!>，!<）。
- 与ANY或ALL谓词配合使用

带有比较运算符的子查询

例：假设一个学生只可能在一个系学习，并且必须属于一个系，则在[例37]可以用 = 代替IN：

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

带有比较运算符的子查询

[例39] 找出每个学生超过他选修课程平均成绩的
的课程号

SELECT Sno,Cno

FROM SC x

WHERE Grade >

(SELECT AVG(Grade)

FROM SC y

WHERE y.Sno = x.SNO);

Sno	Cno
95001	1
95002	2

相关子查询

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

带有比较运算符的子查询

- 子查询一定要跟在比较符之后

错误的例子：

```
SELECT Sno, Sname, Sdept
```

```
FROM Student
```

```
WHERE ( SELECT Sdept
```

```
FROM Student
```

```
WHERE Sname= ' 刘晨 ' ) = Sdept;
```


带有比较运算符的子查询

- 子查询一定要跟在比较符之后

正确的写法:

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

(3) 带有ANY或ALL谓词的子查询

- 谓词语义
 - ANY: 任意一个值
 - ALL: 所有值

需要配合使用比较运算符

> ANY	大于子查询结果中的某个值
> ALL	大于子查询结果中的所有值
< ANY	小于子查询结果中的某个值
< ALL	小于子查询结果中的所有值
>= ANY	大于等于子查询结果中的某个值
>= ALL	大于等于子查询结果中的所有值
<= ANY	小于等于子查询结果中的某个值
<= ALL	小于等于子查询结果中的所有值
= ANY	等于子查询结果中的某个值
= ALL	等于子查询结果中的所有值（通常没有实际意义）
!=（或<>） ANY	不等于子查询结果中的某个值
!=（或<>） ALL	不等于子查询结果中的任何一个值

[例40] 查询其他系中比IS系任一学生年龄小的学生姓名和年龄

```
SELECT Sname,Sage
FROM Student
WHERE Sage < ANY
      (SELECT Sage
       FROM Student
       WHERE Sdept = 'IS')
and Sdept <> 'IS';
```

② 最后在Student关系中
取出Sname和Sage

① 首先在Student关系找出所
有所在系为“IS”人的年龄

Sname	Sage
王名	18

[例41] 查询其他系中比IS系所有学生年龄都大的学生姓名和年龄

SELECT Sname,Sage FROM Student

WHERE Sage > ALL

(SELECT Sage

FROM Student

WHERE Sdept = 'IS')

ORDER BY Sage DESC;

② 最后在Student关系中

取出Sname和Sage

① 首先在Student关系中找到所有所在系为“IS”人的年龄

Sname	Sage
李勇	20

(4) 带有EXISTS谓词的子查询

- 带有EXISTS谓词的子查询不返回任何实际数据
- 它只产生逻辑真值“true”或是逻辑假值“false”

[例43] 查询所有没选修1号课程的学生姓名

SELECT Sname

FROM Student S

WHERE NOT EXISTS

(SELECT *

FROM SC

WHERE S.Sno = Sno AND Cno=1);

[例42] 查询所有选修了1号课程的学生姓名

```
SELECT Sname  
FROM Student S  
WHERE EXISTS  
      (SELECT *  
      FROM SC  
      WHERE S.Sno = Sno AND Cno=1);
```


[例43] 查询选修了全部课程的学生姓名

查询这样的学生姓名，没有一门课程是他不选的

```
SELECT Sname
FROM Student S
WHERE NOT EXISTS
    (SELECT *
     FROM Course C
     WHERE NOT EXISTS
        (SELECT *
         FROM SC
         WHERE S.Sno = Sno AND C.Cno=Cno));
```

[例44] 查询至少选修了学生95002选修的全部课程的学生学号

在不存在这样的课程y，学生95002选修了y，而学生x没有选。

```
SELECT DISTINCT Sno
FROM SC X
WHERE NOT EXISTS
    (SELECT *
    FROM SC Y
    WHERE Sno = '95002' AND
    NOT EXISTS
        (SELECT *
        FROM SC Z
        WHERE Sno = X.Sno AND Cno=Y.Cno));
```

数据查询

- 1 单表查询
- 2 连接查询
- 3 嵌套查询
- 4 集合查询

4 集合查询

- 标准SQL直接支持的集合操作种类
 - 并操作(UNION)
- 一般商用数据库支持的集合操作种类
 - 并操作(UNION)
 - 交操作(INTERSECT)
 - 差操作(MINUS)

(1) 并操作

- 形式

<查询块>

UNION

<查询块>

- 参加**UNION**操作的各结果表的列数必须相同;
- 对应项的数据类型也必须相同

UNION

[例45] 查询选修1号课程和 2号课程学生的学号

SELECT Sno FROM SC WHERE Cno=1

UNION

SELECT Sno FROM SC WHERE Cno=2;

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

Sno
95001
95002

(2) 交操作

◆ 标准SQL中没有提供集合交操作，但可用其他方法间接实现。

[例46] 查询计算机科学系的学生与年龄不大于19岁的学生的交集

本例实际上就是查询计算机科学系中年龄不大于19岁的学生

```
SELECT *
```

```
FROM Student
```

```
WHERE Sdept= 'CS' AND  Sage<=19;
```

交操作（续）

[例47] 查询选修课程1的学生集合与选修课程2的学生集合的交集

本例实际上是查询既选修了课程1又选修了课程2的学生

```
SELECT Sno
FROM SC
WHERE Cno= 1 AND Sno IN
      (SELECT Sno
       FROM SC
       WHERE Cno= 2 );
```

Sno	Cno	Grade
95001	1	92.0
95001	2	85.0
95001	3	88.0
95002	2	90.0
95002	3	80.0

Sno
95001

(3) 差操作

◆ 标准SQL中没有提供集合差操作，但可用其他方法间接实现。

[例48] 查询计算机科学系的学生与年龄不大于19岁的学生的差集。

本例实际上是查询计算机科学系中年龄大于19岁的学生

```
SELECT *
```

```
FROM Student
```

```
WHERE Sdept= 'CS' AND  Sage>19;
```

差操作（续）

[例49] 查询学生姓名与教师姓名的差集

本例实际上是查询学校中未与教师同名的学生姓名

```
SELECT DISTINCT Sname
```

```
FROM Student
```

```
WHERE Sname NOT IN
```

```
(SELECT Tname
```

```
FROM Teacher);
```

(4) 对集合操作结果的排序

- **ORDER BY**子句只能用于对最终查询结果排序，不能对中间结果排序
- 任何情况下，**ORDER BY**子句只能出现在最后
- 对集合操作结果排序时，**ORDER BY**子句中可以用数字指定排序属性

例

SELECT * FROM Student

WHERE Sdept= 'CS'

~~**ORDER BY Sno**~~

UNION

SELECT * FROM Student

WHERE Sage<=19

ORDER BY Sno;

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	M	20	CS
95002	刘晨	F	19	IS
95003	王名	F	18	MA
95004	张立	M	18	IS

查询语句格式

SELECT [ALL|DISTINCT]

指定要显示的属性列

<目标列表表达式> [<别名>]

[, <目标列表表达式> [<别名>]] ...

FROM <表名或视图名> [<别名>]

指定查询对象(基本表或视图)

[, <表名或视图名> [<别名>]] ...

[**WHERE** <条件表达式>]

指定查询条件

[**GROUP BY** <列名> [, <列名>] ...

对查询结果按指定列的值分组

[**HAVING** <条件表达式>]]

对分组后的结果进行筛选

[**ORDER BY** <列名> [, <列名>] ... [ASC|DESC]];

对查询结果表按指定列值的升序或降序排序