

嵌套查询

● 嵌套查询概述

- 一个SELECT-FROM-WHERE语句称为一个查询块
- 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

```
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 ' );
```

嵌套查询求解方法

●不相关子查询:

子查询的查询条件不依赖于父查询

➤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。

```
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 ');
```

嵌套查询求解方法（续）

- **相关子查询：** 子查询的查询条件依赖于父查询
 - 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

带有比较运算符的子查询（续）

[例 3.57]找出每个学生不低于他选修课程平均成绩的课程号。

学生-课程 数据库

- 学生-课程模式 S-T :

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)

Student表

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

Course表

课程号 Cno	课程名 Cname	先行课 Cpno	学分 Ccredit
1	数据库	5	4
2	数学		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		2
7	PASCAL语言	6	4

SC表

学 号 Sno	课程号 Cno	成绩 Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

带有比较运算符的子查询（续）

[例 3.57]找出每个学生不低于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

带有EXISTS谓词的子查询

- EXISTS谓词

- 存在量词 \exists

- 带有EXISTS谓词的子查询不返回任何数据，只产生逻辑真值“true”或逻辑假值“false”。

- 若内层查询结果非空，则外层的WHERE子句返回真值

- 若内层查询结果为空，则外层的WHERE子句返回假值

- 由EXISTS引出的子查询，其目标列表表达式通常都用 * ，因为带EXISTS的子查询只返回真值或假值，给出列名无实际意义。

带有EXISTS谓词的子查询（续）

●NOT EXISTS谓词

- 若内层查询结果非空，则外层的WHERE子句返回假
值
- 若内层查询结果为空，则外层的WHERE子句返回真
值

带有EXISTS谓词的子查询（续）

[例 3.60]查询所有选修了1号课程的学生姓名。

思路分析：

- 本查询涉及Student和SC关系
- 在Student中依次取每个元组的Sno值，用此值去检查SC表
- 若SC中存在这样的元组，其Sno值等于此Student.Sno值，并且其Cno= '1'，则取此Student.Sname送入结果表

SELECT Sname

FROM Student

WHERE EXISTS

(SELECT *

FROM SC

WHERE Sno=Student.Sno AND Cno= ' 1 ');

带有EXISTS谓词的子查询（续）

[例 3.61] 查询没有选修1号课程的学生姓名。

```
SELECT Sname
```

```
FROM   Student
```

```
WHERE NOT EXISTS
```

```
  (SELECT *
```

```
   FROM SC
```

```
  WHERE Sno = Student.Sno AND Cno='1');
```

带有EXISTS谓词的子查询（续）

- 不同形式的查询间的替换

- 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
- 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

带有EXISTS谓词的子查询（续）

[例 3.55]查询与“刘晨”在同一个系学习的学生。

可以用带EXISTS谓词的子查询替换：

```
SELECT Sno,Sname,Sdept
FROM Student S1
WHERE EXISTS
    (SELECT *
     FROM Student S2
     WHERE S2.Sdept = S1.Sdept AND
           S2.Sname = '刘晨');
```

带有EXISTS谓词的子查询（续）

- 不同形式的查询间的替换

- 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
- 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

- 用EXISTS/NOT EXISTS实现全称量词（难点）

- SQL语言中没有全称量词 \forall （For all）
- 可以把带有全称量词的谓词转换为等价的带有存在量词的谓词：

$$(\forall x) P \equiv \neg (\exists x (\neg P))$$

带有EXISTS谓词的子查询（续）

[例 3.62] 查询选修了全部课程的学生姓名。

查询这样的学生，没有哪门课是他（她）不选修的。

$(\forall \text{课程}x) \text{该学生选修}x \equiv \neg (\exists \text{课程}x(\neg \text{该学生选修}x))$

```
SELECT Sname
```

```
FROM Student
```

```
WHERE NOT EXISTS
```

不存在这样的课程

```
(SELECT *
```

```
FROM Course
```

```
WHERE NOT EXISTS
```

该学生没有
选该课程

```
(SELECT *
```

```
FROM SC
```

```
WHERE Sno= Student.Sno
```

```
AND Cno= Course.Cno
```

```
)
```

```
);
```

带有EXISTS谓词的子查询（续）

- 用EXISTS/NOT EXISTS实现逻辑蕴涵（难点）

- SQL语言中没有蕴涵（Implication）逻辑运算

- 可以利用谓词演算将逻辑蕴涵谓词等价转换为：

$$p \rightarrow q \equiv \neg p \vee q$$

优先级从大到小： \neg \wedge \vee

p	q	p→q
真	真	真
真	假	假
假	真	真
假	假	真

带有EXISTS谓词的子查询（续）

[例 3.63]查询至少选修了学生201215122选修的全部课程的学生号码。

解题思路：

■用逻辑蕴涵表达：查询学号为x的学生，对所有的课程y，只要201215122学生选修了课程y，则x也选修了y。

■形式化表示：

用p表示谓词 “学生201215122选修了课程y”

用q表示谓词 “学生x选修了课程y”

则上述查询为： $(\forall y) \ p \rightarrow q$

带有EXISTS谓词的子查询（续）

■等价变换：

利用 $(\forall x) T \equiv \neg (\exists x (\neg T))$

$p \rightarrow q \equiv \neg p \vee q$ 代入

$$(\forall y) p \rightarrow q \equiv \neg (\exists y (\neg (p \rightarrow q)))$$

$$\equiv \neg (\exists y (\neg (\neg p \vee q)))$$

$$\equiv \neg \exists y (p \wedge \neg q)$$



T

■变换后语义：不存在这样的课程y，学生201215122选修了y (p)，而学生x没有选 ($\neg q$)。

带有EXISTS谓词的子查询（续）

查找这样的学生x：对学生x来讲，不存在这样的课程y，学生201215122选修了y，而学生x没有选。

```
SELECT DISTINCT Sno  
FROM SC SC1  
WHERE NOT EXISTS
```

不存在课程y

```
(SELECT *
```

$p \wedge \neg q$

```
FROM SC SC2 p:201215122所选的课程y
```

```
WHERE SC2.Sno = '201215122' AND
```

```
NOT EXISTS /*  $\neg q$  */
```

```
(SELECT *
```

```
FROM SC SC3
```

q: 学生x选了y

```
WHERE SC3.Sno=SC1.Sno AND
```

```
SC3.Cno=SC2.Cno));
```

用p表示谓词 “学生201215122选修了课程y”

用q表示谓词 “学生x选修了课程y”

$(\forall y) p \rightarrow q \equiv \neg \exists y (p \wedge \neg q)$

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```
SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
(SELECT *
FROM SC SC2
WHERE SC2.Sno = ' 201215122 ' AND
NOT EXISTS /*内层 not exists */
(SELECT *
FROM SC SC3
WHERE SC3.Sno=SC1.Sno AND
SC3.Cno=SC2.Cno));
```

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

结果

Sno

内层 NOT EXISTS 返回值
false

外层 NOT EXISTS 返回值

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```

SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
  (SELECT *
   FROM SC SC2
   WHERE SC2.Sno = '201215122' AND
        NOT EXISTS /*内层 not exists */
          (SELECT *
           FROM SC SC3
           WHERE SC3.Sno=SC1.Sno AND
                 SC3.Cno=SC2.Cno));
    
```

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

结果

Sno
201215121

内层 NOT EXISTS
返回值

false, false

外层 NOT EXISTS
返回值

true

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```
SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
  (SELECT *
   FROM SC SC2
   WHERE SC2.Sno = ' 201215122 ' AND
    NOT EXISTS /*内层 not exists */
      (SELECT *
       FROM SC SC3
       WHERE SC3.Sno=SC1.Sno AND
        SC3.Cno=SC2.Cno));
```

结果

Sno
201215121
201215121

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

内层 NOT EXISTS
返回值

false, false

外层 NOT EXISTS
返回值

true

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```
SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
  (SELECT *
   FROM SC SC2
   WHERE SC2.Sno = ' 201215122 ' AND
    NOT EXISTS /*内层 not exists */
      (SELECT *
       FROM SC SC3
       WHERE SC3.Sno=SC1.Sno AND
        SC3.Cno=SC2.Cno));
```

结果

Sno
201215121
201215121
201215121

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

内层 NOT EXISTS 返回值
false, false

外层 NOT EXISTS 返回值
true

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```
SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
(SELECT *
FROM SC SC2
WHERE SC2.Sno = ' 201215122 ' AND
NOT EXISTS /*内层 not exists */
(SELECT *
FROM SC SC3
WHERE SC3.Sno=SC1.Sno AND
SC3.Cno=SC2.Cno));
```

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

结果

Sno
201215121
201215121
201215121

内层 NOT EXISTS 返回值
false

外层 NOT EXISTS 返回值

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
(SELECT *
FROM SC SC2
WHERE SC2.Sno = ' 201215122 ' AND
NOT EXISTS /*内层 not exists */
(SELECT *
FROM SC SC3
WHERE SC3.Sno=SC1.Sno AND
SC3.Cno=SC2.Cno));

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

结果

Sno
201215121
201215121
201215121
201215122

内层 NOT EXISTS 返回值
false , false

外层 NOT EXISTS 返回值
true

SC3 最内层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

SC1 最外层SC表

Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

```
SELECT DISTINCT Sno
FROM SC SC1
WHERE NOT EXISTS /*外层 not exists */
(SELECT *
FROM SC SC2
WHERE SC2.Sno = ' 201215122 ' AND
NOT EXISTS /*内层 not exists */
(SELECT *
FROM SC SC3
WHERE SC3.Sno=SC1.Sno AND
SC3.Cno=SC2.Cno));
```

SC2 中间层SC表

Sno	Cno	Grade
201215122	2	90
201215122	3	80

结果

Sno
201215121
201215121
201215121
201215122
201215122

DISTINCT
去重

→

Sno
201215121
201215122

内层 NOT EXISTS 返回值
false, false

外层 NOT EXISTS 返回值
true

带有EXISTS谓词的子查询

- 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。
- 嵌套查询运算分为相关子查询和非相关子查询，各自的运算过程是怎样的？
- EXISTS/NOT EXISTS谓词子查询什么时候返回真值？什么时候返回假值？
- 不同形式子查询的替换，采用IN谓词、比较运算符、ANY和ALL谓词的子查询都能用EXISTS谓词进行替换，而反过来不一定可以。
- 全称量词和逻辑蕴涵采用NOT EXISTS进行转换表述。(难点)