数据库系统概论

An Introduction to Database System

第三章 关系数据库标准语言SQL

中国人民大学信息学院

数据查询

（嵌套查询）

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（续）

- 上层的查询块称为外层查询或父查询
- 下层查询块称为内层查询或子查询
- SQL语言允许多层嵌套查询
- 子查询的限制
  - 不能使用ORDER BY子句

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。

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

3.4.3 嵌套查询

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（SOME）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

[例 3.55] 查询与“刘晨”在同一个系学习的学生。

此查询要求可以分步来完成

① 确定“刘晨”所在系名

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

结果为： CS

带有IN谓词的子查询（续）

② 查找所有在CS系学习的学生。

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

结果为：

Sno	Sname	Sdept
201215121	李勇	CS
201215122	刘晨	CS

带有IN谓词的子查询（续）

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 ');
```

此查询为不相关子查询。

带有IN谓词的子查询（续）

用自身连接完成[例 3.55]查询要求

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

带有IN谓词的子查询（续）

[例 3.56]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关系中找出选修了3号课程的学生学号

① 首先在Course关系中找出“信息系统”的课程号，为3号

带有IN谓词的子查询（续）

用连接查询实现[例 3.56]：

```
SELECT Sno,Sname
FROM   Student,SC,Course
WHERE Student.Sno = SC.Sno  AND
          SC.Cno = Course.Cno AND
          Course.Cname='信息系统';
```

3.4.3 嵌套查询

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（SOME）或ALL谓词的子查询

4.带有EXISTS谓词的子查询

2. 带有比较运算符的子查询

- 当能确切知道内层查询返回单值时，可用比较运算符（>，<，=，>=，<=，!=或< >）。

在[例 3.55]中，

```
SELECT Sno,Sname,Sdept
FROM   Student
WHERE Sdept  =  /*由于一个学生只可能在一个系学习，用=代替*/
      (SELECT Sdept
       FROM   Student
       WHERE Sname= '刘晨');
```

带有比较运算符的子查询（续）

[例 3.57]找出每个学生超过他选修课程平均成绩的课程号。

相关子查询

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

带有比较运算符的子查询（续）

- 可能的执行过程
  - 从外层查询中取出SC的一个元组x，将元组x的Sno值（201215121）传送给内层查询。

```
SELECT AVG(Grade)
FROM SC y
WHERE y.Sno='201215121';
```

带有比较运算符的子查询（续）

- 可能的执行过程（续）
  - 执行内层查询，得到值88（近似值），用该值代替内层查询，得到外层查询：

```
SELECT Sno,Cno
FROM   SC x
WHERE  Grade >=88;
```

  - 执行该查询，得到结果：

（201215121,1）
（201215121,3）

带有比较运算符的子查询（续）

- 然后外层查询取出下一个元组重复做上述①至③步骤，直到外层的SC元组全部处理完毕。

结果为:

（201215121,1）

（201215121,3）

（201215122,2）

3.4.3 嵌套查询

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

4.带有EXISTS谓词的子查询

带有ANY（SOME）或ALL谓词的子查询（续）

使用ANY或ALL谓词时必须同时使用比较运算

语义为：

> 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

带有ANY（SOME）或ALL谓词的子查询 （续）

[例 3.58] 查询非计算机科学系中比计算机科学系任意一个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= ' CS ')
 AND Sdept <> 'CS ';        /*父查询块中的条件 */
```

带有ANY（SOME）或ALL谓词的子查询（续）

结果：

Sname	Sage
王敏	18
张立	19

执行过程：

（1）首先处理子查询，找出CS系中所有学生的年龄，构成一个集合（20,19）

（2）处理父查询，找所有不是CS系且年龄小于20 或 19的学生

带有ANY（SOME）或ALL谓词的子查询 （续）

- 用聚集函数实现[例 3.58]

```
SELECT Sname,Sage
FROM   Student
WHERE Sage <
                    (SELECT MAX（Sage）
                      FROM Student
                      WHERE Sdept= 'CS ')
  AND Sdept <> ' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

[例 3.59] 查询非计算机科学系中比计算机科学系所有学生年龄都小的学生姓名及年龄。

方法一：用ALL谓词

```
SELECT Sname,Sage
FROM Student
WHERE Sage < ALL
            (SELECT Sage
             FROM Student
             WHERE Sdept= ' CS ')
  AND Sdept <> ' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

方法二：用聚集函数

```
SELECT Sname,Sage
FROM Student
WHERE Sage <
        (SELECT MIN(Sage)
         FROM Student
         WHERE Sdept= ' CS ')
  AND Sdept <>' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

表3.7 ANY（或SOME），ALL谓词与聚集函数、IN谓词的等价转换关系

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

3.4.3 嵌套查询

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（SOME）或ALL谓词的子查询

4.带有EXISTS谓词的子查询

带有EXISTS谓词的子查询

❖ EXISTS谓词

- 存在量词∃
- 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

带有EXISTS谓词的子查询（续）

```
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
  - 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

- 用EXISTS/NOT EXISTS实现全称量词（难点）
  - SQL语言中没有全称量词∀（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：

    （∀x）P ≡ ¬ （∃ x（¬ P））

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谓词的相关子查询替换：

```
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）

[例 3.62] 查询选修了全部课程的学生姓名。

```
SELECT Sname
FROM Student
WHERE NOT EXISTS
        (SELECT *
         FROM Course
         WHERE NOT EXISTS
                 (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：

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

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：

- 用逻辑蕴涵表达该查询：

  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：（∀y） $p \rightarrow q$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）

带有EXISTS谓词的子查询（续）

- 等价变换：

  （∀y）$p \rightarrow q$ ≡ ¬（∃y（¬（$p \rightarrow q$））

  ≡ ¬（∃y（¬（¬p∨ q））））

  ≡ ¬∃y（p∧¬q）

- 变换后语义：不存在这样的课程y，学生201215122选修了y，而学生x没有选。

带有EXISTS谓词的子查询（续）

```
SELECT DISTINCT SCX.Sno
  FROM SC SCX
  WHERE NOT EXISTS
          (SELECT *
           FROM SC SCY
           WHERE SCY.Sno = ' 201215122 '  AND
                    NOT EXISTS
                    (SELECT *
                     FROM SC SCZ
                     WHERE SCZ.Sno=SCX.Sno AND
                              SCZ.Cno=SCY.Cno));
```

数据查询

（嵌套查询2）

3.4.3 嵌套查询

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

4.带有EXISTS谓词的子查询

带有ANY（SOME）或ALL谓词的子查询（续）

在谓词逻辑中，还有存在量词和全称量词的概念，在SQL中并没有对应的表达，统一采用“谓词”来表达

方法一：引入ANY和ALL谓词，其对象为某个查询结果，表示其中任意一个值或者全部值

方法二：引入EXIST谓词，其对象也是某个查询结果，但表示这个查询结果是否为空，返回真值。

带有ANY（SOME）或ALL谓词的子查询（续）

使用ANY或ALL谓词时必须同时使用比较运算

语义为：

> 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

带有ANY（SOME）或ALL谓词的子查询（续）

使用ANY或ALL谓词时必须同时使用比较运算

语义为（续）

<= 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

带有ANY（SOME）或ALL谓词的子查询 （续）

[例 3.58] 查询非计算机科学系中比计算机科学系任意一个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= ' CS ')
 AND Sdept <> 'CS ';        /*父查询块中的条件 */
```

带有ANY（SOME）或ALL谓词的子查询（续）

结果：

Sname	Sage
王敏	18
张立	19

执行过程：

（1）首先处理子查询，找出CS系中所有学生的年龄，构成一个集合（20,19）

（2）处理父查询，找所有不是CS系且年龄小于20 或 19的学生

带有ANY（SOME）或ALL谓词的子查询 （续）

- 用聚集函数实现[例 3.58]

```
SELECT Sname,Sage
FROM   Student
WHERE Sage <
                    (SELECT MAX（Sage）
                      FROM Student
                      WHERE Sdept= 'CS ')
  AND Sdept <> ' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

[例 3.59] 查询非计算机科学系中比计算机科学系所有学生年龄都小的学生姓名及年龄。

方法一：用ALL谓词

```
SELECT Sname,Sage
FROM Student
WHERE Sage < ALL
            (SELECT Sage
             FROM Student
             WHERE Sdept= ' CS ')
  AND Sdept <> ' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

方法二：用聚集函数

```
SELECT Sname,Sage
FROM Student
WHERE Sage <
        (SELECT MIN(Sage)
         FROM Student
         WHERE Sdept= ' CS ')
  AND Sdept <>' CS ';
```

带有ANY（SOME）或ALL谓词的子查询（续）

表3.7 ANY（或SOME），ALL谓词与聚集函数、IN谓词的等价转换关系

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

3.4.3 嵌套查询

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（SOME）或ALL谓词的子查询

4.带有EXISTS谓词的子查询

带有EXISTS谓词的子查询

❖ EXISTS谓词

- 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）

[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

带有EXISTS谓词的子查询（续）

```
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）

- 用EXISTS代替其他谓词
  - 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
- 用EXISTS/NOT EXISTS实现全称量词（难点）
- 用EXISTS实现逻辑蕴含
  - 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：

    （∀x）P ≡ ¬ （∃ x（¬ P））

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谓词的相关子查询替换：

```
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）

[例 3.62] 查询选修了全部课程的学生姓名。

```
SELECT Sname
FROM Student
WHERE NOT EXISTS
        (SELECT *
          FROM Course
          WHERE NOT EXISTS
                  (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：

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

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：

- 用逻辑蕴涵表达该查询：

  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：（∀y） $p \rightarrow q$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）

带有EXISTS谓词的子查询（续）

- 等价变换：

  （∀y）$p \rightarrow q$ ≡ ¬（∃y（¬（$p \rightarrow q$））

  ≡ ¬（∃y（¬（¬p∨ q））））

  ≡ ¬∃y（p∧¬q）

- 变换后语义：不存在这样的课程y，学生201215122选修了y，而学生x没有选。

带有EXISTS谓词的子查询（续）

- SELECT DISTINCT SCX.Sno
  FROM SC SCX
  WHERE NOT EXISTS

```
(SELECT *                                          P & Not(Q)
 FROM SC SCY
 WHERE SCY.Sno = ' 201215122 '  AND
            NOT EXISTS
            (SELECT *
             FROM SC SCZ
             WHERE SCZ.Sno=SCX.Sno AND
                        SCZ.Cno=SCY.Cno));
```