

数据库原理

The Theory of Database System

第三章 关系数据库标准语言

SQL



中国矿业大学计算机学院



中国矿业大学数据库原理精品课程

本讲主要内容

数据查询——嵌套查询（集合运算
和除运算）



引导子查询的谓词

- 带有IN谓词的子查询
- 带有比较运算符的子查询
- 带有ANY或ALL谓词的子查询
- 带有EXISTS谓词的子查询



带有EXISTS谓词的子查询

■ 相关子查询

- 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EXISTS谓词的子查询(续)

■ 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，则返回真值
- 若内层查询结果为空，则返回假值



带有EXISTS谓词的子查询(续)

- NOT EXISTS谓词
 - 若内层查询结果非空，则返回假值
 - 若内层查询结果为空，则返回真值
- 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)

【例】查询选修了180102号课程的学生学号和姓名。

```
SELECT 学号,姓名  
FROM 学生  
WHERE EXISTS  
    (SELECT *  
     FROM 学习  
     WHERE 学生.学号=学习.学号  
     AND 学习.课程号='180102');
```



带有EXISTS谓词的子查询(续)

表 3-2 学生表

| 学号 | 姓名 | 性别 | 籍贯 | 出生年份 | 学院 |
|--------|-----|-----|----|------|-----|
| 091501 | 王英 | 女 | 河北 | 1997 | 计算机 |
| 091502 | 王小梅 | 女 | 江苏 | 2000 | 信电 |
| 091503 | 张小飞 | 男 | 江西 | 1996 | 计算机 |
| 091504 | 孙志鹏 | 男 | | | |
| 091505 | 徐颖 | 女 | | | |
| 091506 | 钱易蒙 | 男 | | | |
| ... | ... | ... | | | |

学习表

| 学号 | 课程号 | 成绩 |
|--------|--------|-----|
| 091501 | 180101 | 78 |
| 091501 | 180102 | 80 |
| 091501 | 180103 | 77 |
| 091503 | 180101 | 89 |
| 091503 | 180102 | 78 |
| 091503 | 180103 | 70 |
| 091503 | 180104 | 90 |
| 091504 | 180101 | 59 |
| 091504 | 180102 | 50 |
| 091504 | 180103 | |
| ... | ... | ... |

结果集

| 姓名 | 学号 |
|-----|--------|
| 王英 | 091501 |
| 张小飞 | 091503 |
| ... | ... |



带有EXISTS谓词的子查询(续)

表 3-2 学生表

| 学号 | 姓名 | 性别 | 籍贯 | 出生年份 | 学院 |
|--------|-----|-----|----|------|-----|
| 091501 | 王英 | 女 | 河北 | 1997 | 计算机 |
| 091502 | 王小梅 | 女 | 江苏 | 2000 | 信电 |
| 091503 | 张小飞 | 男 | 江西 | 1996 | 计算机 |
| 091504 | 孙志鹏 | 男 | | | |
| 091505 | 徐颖 | 女 | | | |
| 091506 | 钱易蒙 | 男 | | | |
| ... | ... | ... | | | |

学习表

| 学号 | 课程号 | 成绩 |
|--------|--------|-----|
| 091501 | 180101 | 78 |
| 091501 | 180102 | 80 |
| 091501 | 180103 | 77 |
| 091503 | 180101 | 89 |
| 091503 | 180102 | 78 |
| 091503 | 180103 | 70 |
| 091503 | 180104 | 90 |
| 091504 | 180101 | 59 |
| 091504 | 180102 | 50 |
| 091504 | 180103 | |
| ... | ... | ... |

结果集

| 姓名 | 学号 |
|-----|--------|
| 王英 | 091501 |
| 张小飞 | 091503 |
| ... | ... |



带有EXISTS谓词的子查询(续)

【例】查询没有选修数据库原理课程的学生
的学号和姓名。

```
SELECT 学号,姓名
```

```
FROM 学生
```

```
WHERE NOT EXISTS
```

```
(SELECT *
```

```
FROM 学习 JOIN 课程 ON 学习.课程号=课程.课程号
```

```
WHERE 课程名='数据库原理'
```

```
AND 学习.学号=学生.学号);
```



集合运算

- 集合运算：
 - 并操作 (UNION)
 - 交操作 (INTERSECT)
 - 差操作 (EXCEPT)
- 运算的对象：集合
- 双目运算：两个查询块代表两个集合



(1) 并运算

- 形式

<查询块>

UNION

<查询块>

- 参加UNION操作的结果集的列数必须相同，对应项的数据类型也必须相同。



(1) 并运算

【例】查询选修了180101号或180102号课程的学生学号。

```
(SELECT 学号  
FROM 学习  
WHERE 课程号='180101')  
UNION  
(SELECT 学号  
FROM 学习  
WHERE 课程号='180102');
```



(1) 并运算

- UNION: 去除重复
- UNION ALL: 保留重复

【例】查询选修了180101号或180102号课程的学生学号。

```
(SELECT 学号  
FROM 学习  
WHERE 课程号='180101'  
OR 课程号='180102');
```



(2) 交运算

- 标准SQL中没有提供集合交操作，但可用其他方法间接实现。
- 有些DBMS支持交运算：

形式

<查询块>

INTERSECT

<查询块>



(2) 交运算

【例】查询同时选修了180101和180102号课程的学生学号。

```
(SELECT 学号  
FROM 学习  
WHERE 课程号='180101')  
INTERSECT  
(SELECT 学号  
FROM 学习  
WHERE 课程号='180102');
```



(2) 交运算

【例】查询同时选修了180101和180102号课程的学生学号。

通过**IN**实现交运算：

```
SELECT 学号  
FROM 学习  
WHERE 课程号='180101' AND 学号 IN  
    ( SELECT 学号  
      FROM 学习  
      WHERE 课程号='180102');
```



(2) 交运算

【例】查询同时选修了180101和180102号课程的学生学号。

通过**EXISTS**实现交运算：

```
SELECT S1.学号
FROM 学习 as S1
WHERE S1.课程号='180101' AND EXISTS
( SELECT *
  FROM 学习 as S2
  WHERE S2.课程号='180102'
    AND S2.学号=S1.学号);
```



(3) 差运算

- 标准SQL中没有提供集合交操作，但可用其他方法间接实现。
- 有些DBMS支持交运算：
形式

<查询块>

EXCEPT

<查询块>



(3) 差运算

【例】查询只选修了180101号课程而没有选修180102号课程的学生学号。

```
(SELECT 学号  
FROM 学习  
WHERE 课程号='180101')  
EXCEPT  
(SELECT 学号  
FROM 学习  
WHERE 课程号='180102');
```



(3) 差运算

【例】查询只选修了180101号课程而没有选修180102号课程的学生学号。

通过**NOT IN**实现差运算：

```
SELECT 学号  
FROM 学习  
WHERE 课程号='180101' AND 学号 NOT IN  
( SELECT 学号  
  FROM 学习  
  WHERE 课程号='180102');
```



(3) 差运算

【例】查询只选修了180101号课程而没有选修180102号课程的学生学号、课程号和成绩。

通过**NOT EXISTS**实现差运算：

```
SELECT S1.学号  
FROM 学习 as S1  
WHERE S1.课程号='180101'  
AND NOT EXISTS  
( SELECT *  
FROM 学习 as S2  
WHERE S2.课程号='180102'  
AND S2.学号=S1.学号);
```



除运算

(1) 除运算适用什么场合？

查询条件中：

体现“一个集合包含另一个集合”的概念

【例】选修过全部课程的学生们的学号和姓名。

查询的条件：某个学生的选课集合大于等于全部课程的集合。



除运算(续)

(2) SQL中如何表示除运算？

关键点：证明一个集合包含了另一个集合

设有两个集合A和B，证明 $A \supseteq B$

- 正向证明：穷举法
- 反向证明： $\neg (B-A)$

不存在
NOT EXISTS

差运算
EXCEPT



除运算(续)

【例】选修过全部课程的学生们的学号和姓名。

- 分析:

该题的查询条件是一个集合包含另一集合，即某一个学生的选课集合 \supseteq 全部课程的集合。

- 思路:

大集合：某个学生所选修的课程集合

小集合：全部的课程集合

$\neg(B-A)$: NOT EXISTS (小集合 EXCEPT 大集合)



除运算(续)

(SELECT 课程号
FROM 课程)

所有的课程

EXCEPT

(SELECT DISTINCT 课程号
FROM 学习
WHERE 学号=XXX)

某个学生所
选的课程



除运算(续)

SELECT ?

FROM ?

WHERE NOT EXISTS

((SELECT 课程号
FROM 课程)

EXCEPT

(SELECT DISTINCT 课程号
FROM 学习
WHERE 学号=XXX));



除运算(续)

```
SELECT 姓名  
FROM 学生  
WHERE NOT EXISTS  
( (SELECT 课程号  
    FROM 课程 )  
  EXCEPT  
(SELECT DISTINCT 课程号  
  FROM 学习  
  WHERE 学号=学生.学号) );
```



除运算(续)

```
SELECT 姓名  
FROM 学生  
WHERE NOT EXISTS  
( (SELECT 课程号  
    FROM 课程  
    WHERE 课程号 NOT IN  
        (SELECT DISTINCT 课程号  
         FROM 学习  
         WHERE 学号=学生.学号) ) );
```



除运算(续)

SELECT 姓名

FROM 学生

WHERE NOT EXISTS

(SELECT *

FROM 课程

WHERE NOT EXISTS

(SELECT *

FROM 学习

WHERE 学习.学号=学生.学号

AND 学习.课程号=课程.课程号));



除运算(续)

除运算的执行过程:

学生

| 学号 | 姓名 | |
|-----|----|-------|
| s01 | 张三 | |
| s02 | 李四 | |
| s03 | 王五 | |

课程

| 课程号 | 课程名 | ... |
|-----|-----|-----|
| c01 | C++ | ... |
| c02 | AI | ... |

学习

| 学号 | 课程号 | 成绩 |
|-----|-----|----|
| s01 | c01 | 80 |
| s01 | c02 | 85 |
| s02 | c01 | 70 |
| s03 | c02 | 78 |



除运算(续)

除运算的执行过程:

学生

| 学号 | 姓名 | |
|-----|----|-------|
| s01 | 张三 | |
| s02 | 李四 | |
| s03 | 王五 | |

课程

| 课程号 | 课程名 | ... |
|-----|-----|-----|
| c01 | C++ | ... |
| c02 | AI | ... |

学习

| 学号 | 课程号 | 成绩 |
|-----|-----|----|
| s01 | c01 | 80 |
| s01 | c02 | 85 |
| s02 | c01 | 70 |
| s03 | c02 | 78 |

ϕ





除运算(续)

除运算的执行过程:

学生

| 学号 | 姓名 | |
|--|----|-------|
| s01 | 张三 | |
|  s02 | 李四 | |
| s03 | 王五 | |

课程

| 课程号 | 课程名 | ... |
|---|-----|-----|
|  c01 | C++ | ... |
|  c02 | AI | ... |

学习

| 学号 | 课程号 | 成绩 |
|-----|-----|----|
| s01 | c01 | 80 |
| s01 | c02 | 85 |
| s02 | c01 | 70 |
| s03 | c02 | 78 |



除运算(续)

除运算的执行过程:

学生

| 学号 | 姓名 | |
|------|----|-------|
| ✓s01 | 张三 | |
| ✗s02 | 李四 | |
| ✗s03 | 王五 | |

课程

| 课程号 | 课程名 | ... |
|------|-----|-----|
| ✓c01 | C++ | ... |
| ✗c02 | AI | ... |

学习

| 学号 | 课程号 | 成绩 |
|-----|-----|----|
| s01 | c01 | 80 |
| s01 | c02 | 85 |
| s02 | c01 | 70 |
| s03 | c02 | 78 |



除运算(续)

【例】 查询至少选修了091501号学生选修的全部课程的学生学号。

分析：

- B集合：091501号同学的选课集合
- A集合：某个学生的选课集合
- $A \supseteq B$



除运算(续)

【例】查询至少选修了091501号学生选修的全部课程的学生学号。

```
SELECT 学号
FROM 学生
WHERE NOT EXISTS
( SELECT *
  FROM 学习 AS First
 WHERE First.学号='091501' AND NOT EXISTS
    ( SELECT *
      FROM 学习 AS Second
     WHERE Second.学号=学生.学号
       AND Second.课程号= First.课程号));
```



除运算(续)

【例】 查询被所有学生都选修的课程名称。

分析：

- 小集合： 所有学生
- 大集合： 某门课程的选课学生



除运算(续)

【例】 查询被所有学生都选修的课程名称。

```
SELECT 课程名  
FROM 课程  
WHERE NOT EXISTS  
  (SELECT *  
   FROM 学生  
   WHERE NOT EXISTS  
     ( SELCET *  
      FROM 学习  
      WHERE 学号=学生.学号  
      AND   课程号=课程.课程号) );
```

