

[点这里，有很多篇《用关系代数表示数据查询的典型例子》](#)

在线阅读本文：http://m.3y.uu456.com/mbp_7tmvq2vucj0fluh9bb2i_1.html

用关系代数表示数据查询的典型例子

用关系代数表示数据查询的典型例子

[例]设 教学数据库中有 3 个关系：

学生关系 S(SNO,SNAME,AGE,SEX)

学习关系 SC(SNO,CNO,GRADE)

课程关系 C(CNO,CNAME,TEACHER)

下面用关系代数表达式和 SQL 语句分别表达每个查询语句。(1) 检索学习课程号为 C2 的学生学号与成绩。

$\pi_{SNO, GRADE}(\sigma_{CNO='C2'}(SC))$

SELECT SNO, GRADE FROM SC WHERE CNO='C2'

(2) 检索学习课程号为 C2 的学生学号与姓名

$\pi_{SNO, SNAME}(\sigma_{CNO='C2'}(SSC))$

由于这个查询涉及到两个关系 S 和 SC，因此先对这两个关系进行自然连接，同一位学生的有关的信息，然后再执行选择投影操作。此查询亦可等价地写成： $\pi_{SNO, SNAME}(S)$

$(\pi_{SNO}(\sigma_{CNO='C2'}(SC)))$

这个表达式中自然连接的右分量为”学了 C2 课的学生学号的集合”。这个表达式比前一个表达式优化，执行起来要省时间，省空间。

SELECT SNO, SNAME FROM S WHERE EXISTS (SELECT * FROM SC WHERE SNO=S.SNO AND CNO='C2')

(3) 检索选修课程名为 MATHS 的学生学号与姓名。 $\pi_{SNO, SNAME}(\sigma_{CNAME='MATHS'}(S \bowtie SCC))$

SELECT S.SNO, SNAME FROM S, SC, C

WHERE S.SNO=SC.SNO AND SC.CNO=C.CNO AND CNAME='MATHS'

(4) 检索选修课程号为 C2 或 C4 的学生学号。

$\pi_{SNO}(\sigma_{CNO='C2' \vee CNO='C4'}(SC))$

SELECT S.SNO, SNAME FROM S, SC, C

WHERE S.SNO=SC.SNO AND SC.CNO=C.CNO AND CNAME='MATHS'

(5) 检索至少选修课程号为 C2 或 C4 的学生学号。 $\pi_1(\sigma_{1=4 \wedge 2='C2' \wedge 5='C4'}(SC \times SC))$

SELECT SNO

FROM SC AS X, SC AS Y

WHERE X.SNO=Y.SNO AND X.CNO='C2'

AND Y.CNO='C4'

这里 (SC×SC) 表示关系 SC 自身相乘的乘积操作，其中数字 1, 2, 4, 5 都为它的结果关系中的属性序号。比较这一题与上一题的差别。

(6) 检索不学 C2 课的学生姓名与年龄。

$\pi_{SNAME, AGE}(S) - \pi_{SNAME, AGE}(\sigma_{CNO='C2'}(S$

SC))

这个表达式用了差运算，差运算的左分量为“全体学生的姓名和年龄”，右分量为“学了 C2 课的学生姓名与年龄”。 SELECT SNAME, AGE FROM S

WHERE SNO NOT IN (SELECT SNO FROM SC

WHERE CNO='C2') 或 SELECT SNAME, AGE FROM S

WHERE NOT EXISTS (SELECT * FROM SC

WHERE SC.SNO=S.SNO AND CNO='C2')

(7) 检索学习全部课程的学生姓名。编写这个查询语句的关系代数过程如下：

(a) 学生选课情况可用 $\pi_{SNO, CNO}(SC)$ 表示； (b) 全部课程可用 $\pi_{CNO}(C)$ 表示；

(c) 学了全部课程的学生学号可用除法操作表示。

操作结果为学号 SNO 的集合，该集合中每个学生（对应 SNO）与 C 中任一门课程号 CNO 配在一起都在 $\pi_{SNO, CNO}(SC)$ 中出现（即 SC 中出现），所以结果中每个学生都学了全部的课程（这是“除法”操作的含义）： $\pi_{SNO, CNO}(SC) \div \pi_{CNO}(C)$

(d) 从 SNO 求学生姓名 SNAME，可以用自然连结和投影操作组合而成： $\pi_{SNAME}(S$

$(\pi_{SNO, CNO}(SC) \div \pi_{CNO}(C)))$

这就是最后得到的关系代数表达式。SELECT SNAME FROM S

WHERE NOT EXISTS (SELECT * FROM C

WHERE NOT EXISTS (SELECT * FROM SC

WHERE SC.SNO=S.SNO AND SC.CNO=C.CNO)) (8) 检索所学课程包含 S3 所学课程的学生学号。

注意：学生 S3 可能学多门课程，所以要用到除法操作来表达此查询语句。学生选课情况可用操作 $\pi_{SNO, CNO}(SC)$ 表示；

所学课程包含学生 S3 所学课程的学生学号，可以用除法操作求得：

$\pi_{SNO, CNO}(SC) \div \pi_{CNO}(\sigma_{SNO='S3'}(SC))$

SELECT DISTINCT SNO FROM SC X

WHERE NOT EXISTS

(SELECT * FROM SC Y WHERE Y.SNO='S3' AND NOT EXISTS

(SELECT * FROM SC Z

WHERE Z.SNO=X.SNO AND Z.CNO=Y.CNO))

(9) 将新课程元组 (' C10','PHYSICS','YU') 插入到关系 C 中 (CU('C10','PHYSICS','YU'))

INSERT INTO C

VALUES('C10','PHYSICS','YU')

(10) 将学号 S4 选修课程号为 C4 的成绩改为 85 分。 (SC- (' S4','C4',?)U('S4','C4',85))

修改操作代数表示分两步实现：先删去原元组，再插入新元组。题目中未给出原先的成绩，所以用”？”代替，检索时可忽略。

UPDATE SC SET GRADE=85 WHERE SNO='S4' AND CNO='C4'

三亿文库 <http://m.3y.uu456.com/>

免费文档下载