**2020春《数据库系统》作业1**

主讲教师: 邹兆年(znzou@hit.edu.cn)

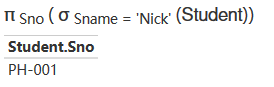
姓名: \_\_\_陶飞宇\_\_\_ 学号:\_\_1170300720\_ 班号: \_\_\_1737301\_\_\_\_

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题目 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 总分 |
| 得分 |  |  |  |  |  |  |  |  |  |  |  |

1. 回答
   1. 成立，因为与两者因选择的串接定律分别成立，所以该式成立。
   2. 不成立，由投影的串接定律可知，最终结果由最外侧的投影操作条件决定，即

，而两者不一定相等

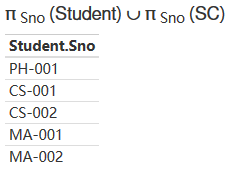
c) 不成立，前者是包含后者的关系，后者因为先进行投影操作，而选择操作中的一些条件判断所需的属性可能被这次投影操作所滤去，使得某些项被缺少，例如：



交换后，则直接报错，因为投影后Sname被过滤



d) 不成立，并操作要求两者具有相同的属性个数且各属性相容，该等式右侧先进行了相同的投影操作，故而能够保证并操作的正常执行；而左侧的部分无法保证这一点，故而两者不一定相等。例如：

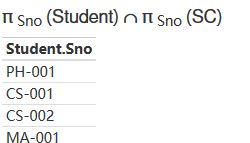


而两者直接做并集则报错：



如果两者有相同的属性，那么该命题成立

e) 不成立，结果与原理同上题，实例如下：

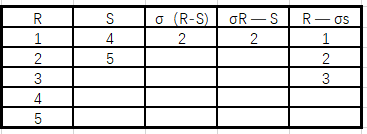




f) 成立，首先考虑R与S不相容的情况，那么等式的每一项都不相容，成立。

若R与S相容，那么无论是对R于S的交集进行选择，还是只对R、S的一项进行选择，两者相交部分都进行了选择操作。如果先选择再交，交运算过滤了不在两者交集之外选择出的部分，等同于交后在进行选择，故而等式成立。

g) 不成立，举例如下：



假设σ会从12345中选出2，则因为减数被选择操作过滤为空，导致最后结果不一致。

h) 成立，实际上相当于三者同时做连接，即使将括号去掉三者也一样。

i）成立，实际上相当于三者同时做自然连接，即使将括号去掉三者也一样。

j）成立, 无论是自然连接还是交运算，两个相同集合做运算，得到的还是这个集合。

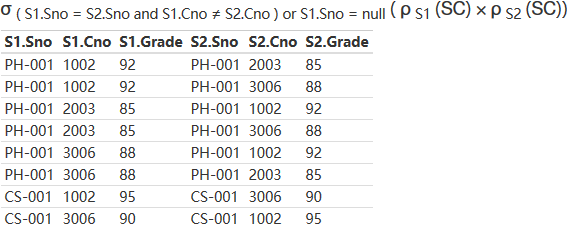
1. 设R的一个非主键关系为P，则关系代数式为：



用法：如果结果集合有任何元组，则表明该关系R不满足实体完整性约束：

原理：对R进行重命名后对其自身进行笛卡尔乘积，将其中K相同且P不同的元组选择出，这类元组由原表中主键相同的两个不同元组组成。如果这类元组存在，则说明有两个主键相同的不同元组。再加上主键为空的元组。整个集合说明了一旦其中有元组，则R不满足实体完整性约束。

如上节课所用数据库中的SC，将其Sno属性视为主键，则该表达式的结果为：



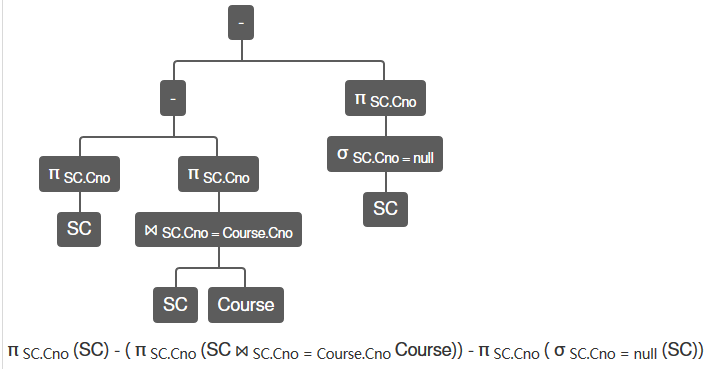
可以说明此时SC不满足实体完整性约束。

3. 

用法：如果结果集合有任何元组，则表明该关系R不满足参照完整性约束

原理：选出F与K中有两者相同情况的部分F中的值（关系式第二部分），从F中去除，剩下的就是在F中出现而在K中不出现的，F中的值；再从其中将F为空的元组去除，剩下的就是在F中出现而在K中不出现的且F不为空的元组，这样的元组违反了参照完整性约束。

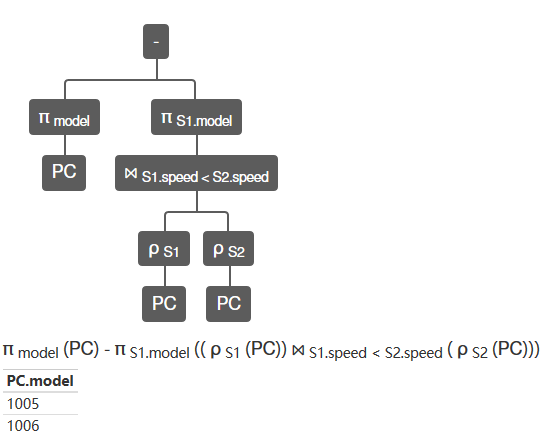
举例：视SC的Cno为F，Course的Cno为K，则有：



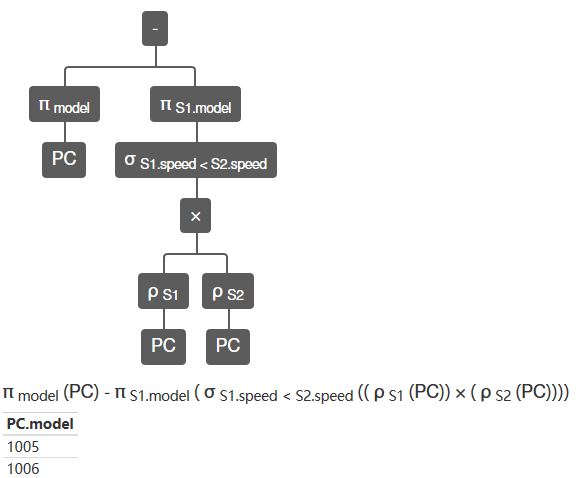
可见最后没有结果，SC与Course符合参照完整性约束。

4.

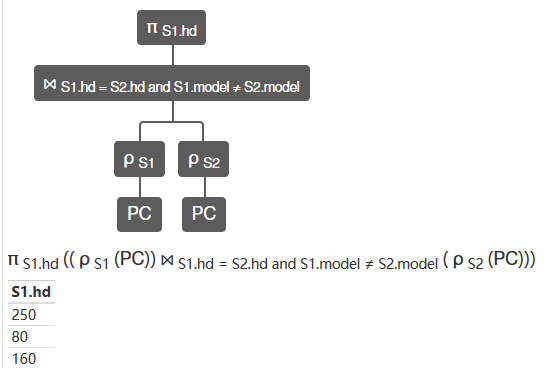
a) 1. πmodel (PC) - πS1.model ((ρS1 (PC)) ⨝S1.speed<S2.speed (ρS2 (PC)))



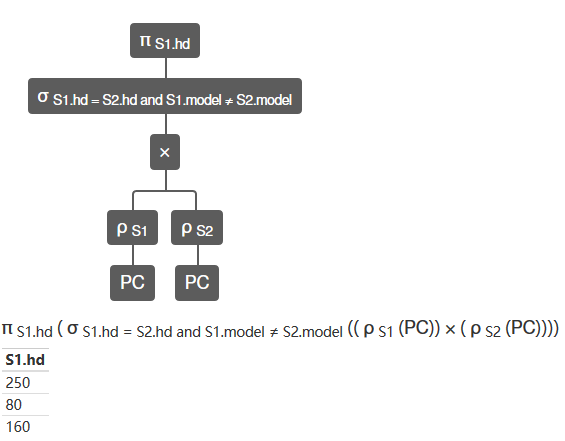
2. πmodel (PC) - πS1.model (σS1.speed<S2.speed ((ρS1 (PC)) ⨯ (ρS2 (PC))))



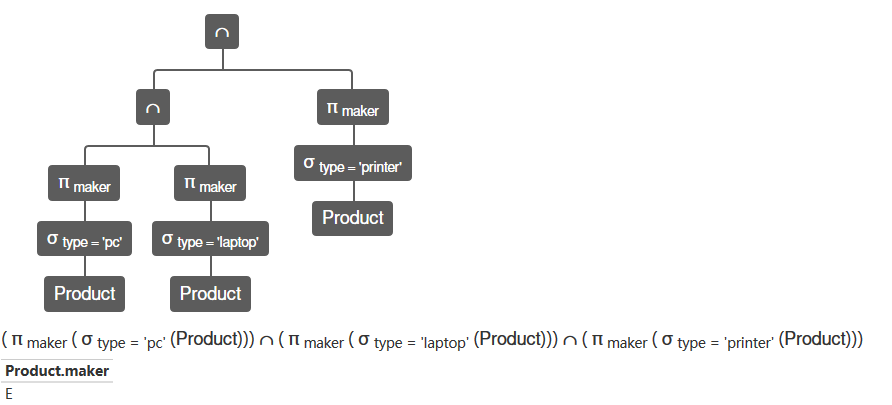
b)1. πS1.hd ((ρS1 (PC)) ⨝S1.hd=S2.hd∧S1.model≠S2.model (ρS2 (PC)))



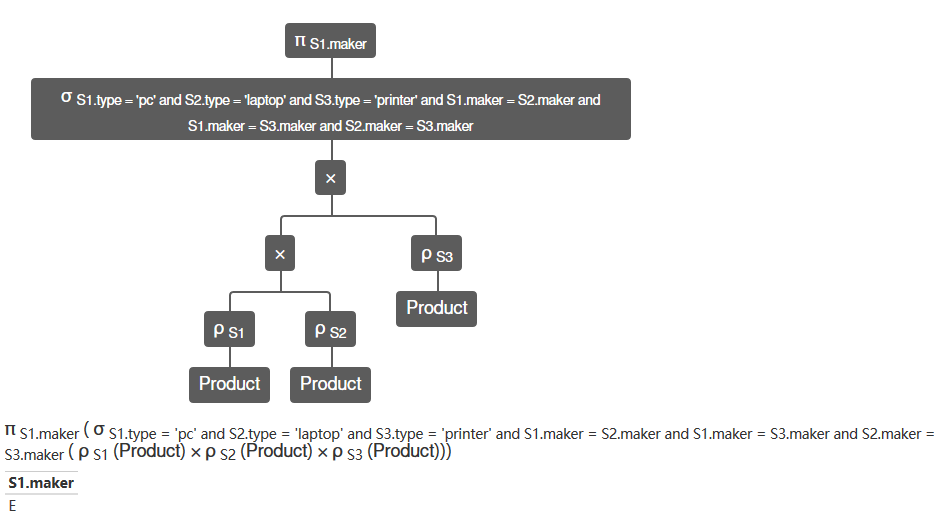
2. πS1.hd (σS1.hd=S2.hd∧S1.model≠S2.model ((ρS1 (PC)) ⨯ (ρS2 (PC))))



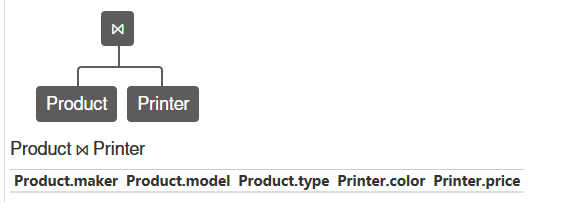
c)1. (πmaker (σtype='pc' (Product))) ∩ (πmaker (σtype='laptop' (Product))) ∩ (πmaker (σtype='printer' (Product)))



2. πS1.maker (σS1.type='pc'∧S2.type='laptop'∧S3.type='printer'∧S1.maker=S2.maker∧S1.maker=S3.maker∧S2.maker=S3.maker (ρS1 (Product)⨯ρS2 (Product)⨯ρS3 (Product)))



d）结果为空：



因为两者同名的属性为model与type，虽然model有相同的部分，但是对于type，Product.type的取值为{‘pc’,’product’,’laptop’,’printer’}而Printer.type的取值为{'ink-jet

,’ laser’ }故而两者在model和type上没有完全相同的一对元组，则结果为空。

将Printer的type过滤，则可以得到model相同的结果：

