

一、

1. **A** 可以举出以下反例

表 1: R	
A	F
1	2
3	4

表 2: S	
A	F
1	2
3	4

可以看到，对于左式 $\Pi_F(R_1 \cap R_2) = \emptyset$ ；而对于右式 $\Pi_F(R_1) \cap \Pi_F(R_2) = \{2\}$ 。因而 **A** 错误。

2. **A** 对于 $R \times S$ 的前 r 列为 R 的一个元组，后 s 列为 S 的一个元组，故原来比较中使用的为 R 的第 i 列和 S 的第 j 列，此处应该换为 $R \times S$ 的第 i 列和第 $j+r$ 列。

二、

1. $\Pi_{SNO,PNO}(SPJ) \div \Pi_{SNO}(\sigma_{CITY=LONDON}(S))$
2. $\Pi_{PNO}(\sigma_{S.CITY=J.CITY}(SPJ \bowtie S \bowtie J))$
3. $\Pi_{JNAME}(\sigma_{SNO=S1}(SPJ) \bowtie J)$
4. $\Pi_{JNAME}((SPJ \bowtie J) \bowtie (\Pi_{PNO}(\sigma_{COLOR=red}(P))))$
5. $\Pi_{JNAME,PNO}(SPJ \bowtie J) \div \Pi_{PNO}(P)$
6. $\Pi_{SNAME,PNO}(SPJ \bowtie S) \div \sigma_{PNO=P1 \vee PNO=P2}(\Pi_{PNO}(P))$
7. $\Pi_{PNAME}(P \bowtie \Pi_{COLOR}(\sigma_{PNAME="TV"}(P)))$

三、

- (1) $\{t | \exists u \in R \wedge t[B] = u[B] \wedge t[C] = u[C]\}$
- (2) $\{t | t \in R \wedge t[B] > t[C]\}$
- (3) $\{t | t \in R \vee t \in S\}$
- (4) $\{t | t \in R \wedge t \in S\}$
- (5) $\{t | t \in R \wedge t \notin S\}$
- (6) $\{t | \exists r \in R \wedge \exists s \in S \wedge r[B] = s[B] \wedge t[A] = r[A] \wedge t[B] = r[B] \wedge t[C] = s[C]\}$
- (7) $\{t | t \in \{u | \exists r \in R \wedge u[A] = r[A] \wedge u[B] = r[B]\} \wedge (\forall s \in \{v | \exists p \in S \wedge v[C] = p[C]\})(\exists r \in R \wedge r[A] = t[A] \wedge r[B] = t[B] \wedge s[C] = r[C])\}$

四、

- 联系：限制在安全表达式范围内的元组关系演算，限制在安全表达式范围内的域关系演算以及关系代数的表达能力等价。
- 区别：
 - **关系代数**是一种抽象的查询语言，它用对关系的运算来表达查询；并且有限关系上的关系代数操作结果都不会导致无限关系和无穷验证，因此关系代数系统是安全的。
 - **元组关系演算**是非过程化的查询语言，只描述所需信息，而不给出获得该信息的具体过程；并且元组关系演算系统不是安全的。
 - **域关系演算**是另一种关系演算的另一种形式，使用从属性域中取值的域变量，而不是整个元组的值；并且关系演算系统也不是安全的。