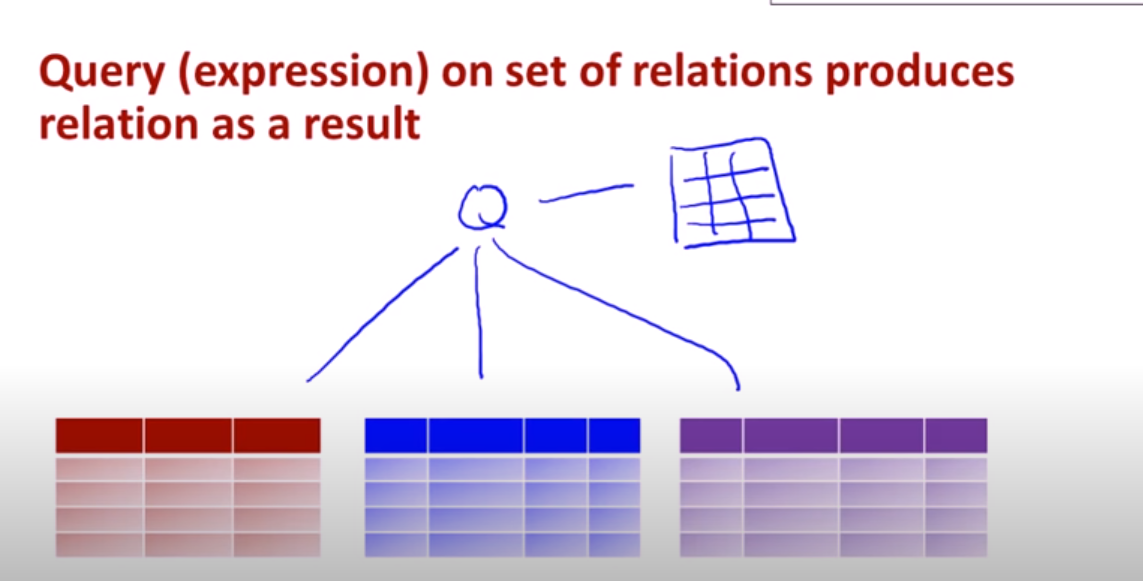


Database Query Language是对DB产生影响的query语言

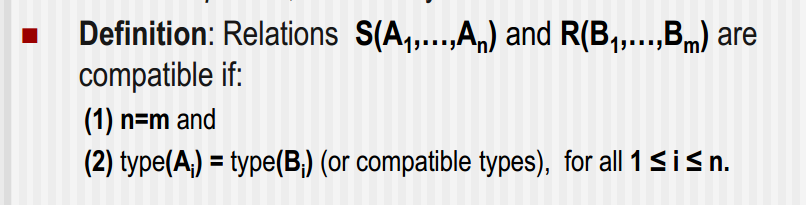
RA是与relational data model(也就是表格) 有关的query语言 //他输入的是表格，输出的也是表格

RA的input output都是relations

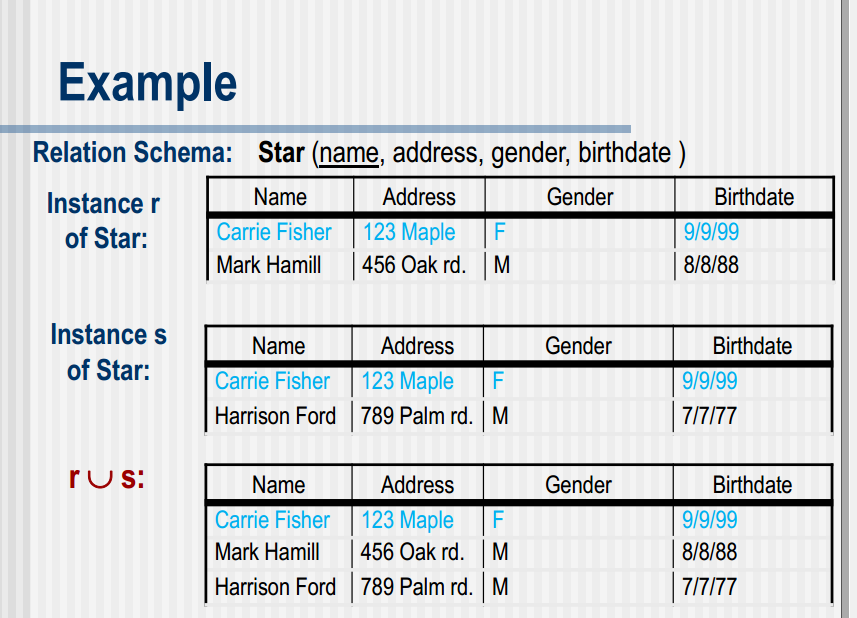


Union,Intersection,Difference

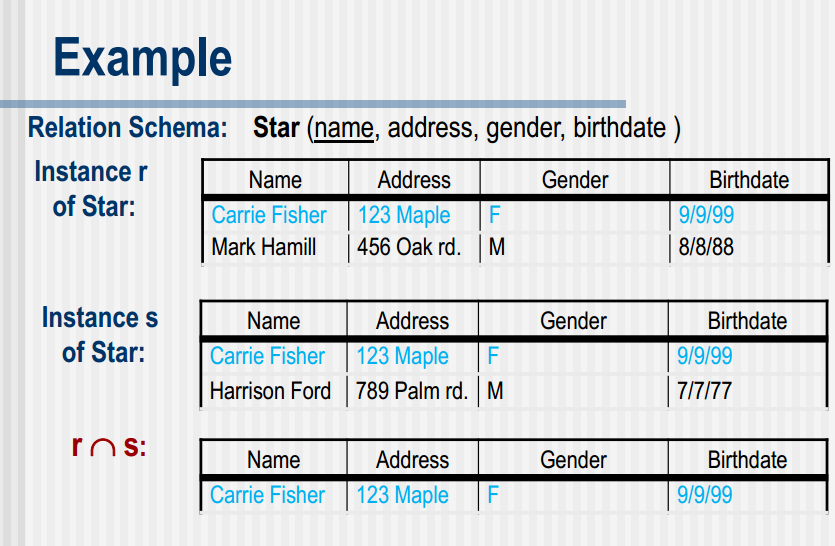
这三兄弟一定要在 relation S 与relation R 在compatible兼容的时候才有意义，换句话说，他们有相同的schema



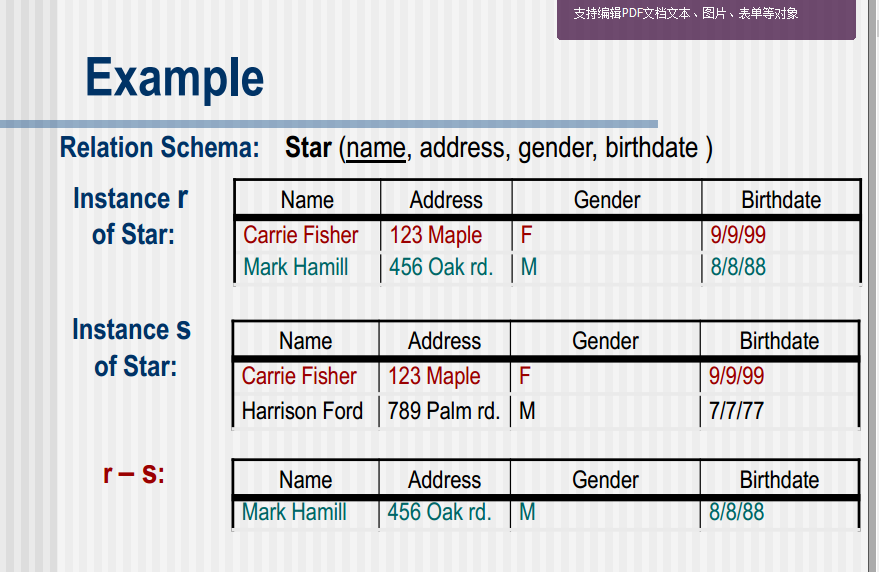
有相同数量的attribute且type都相等

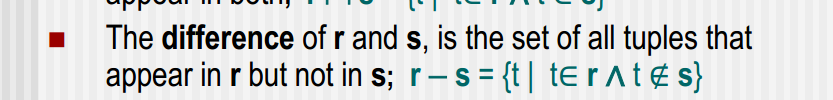


Union就是连接起来，但是是set所以没有重复的



jiaoji

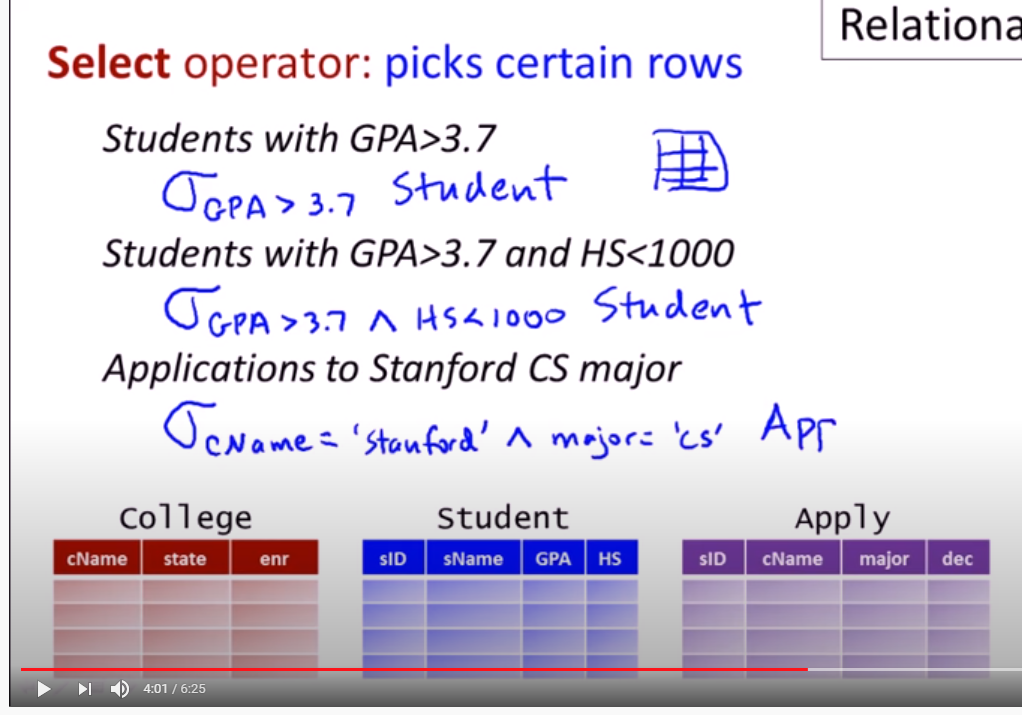




r-s就是不要共同属于的，或者说挑选出属于r单不属于S的



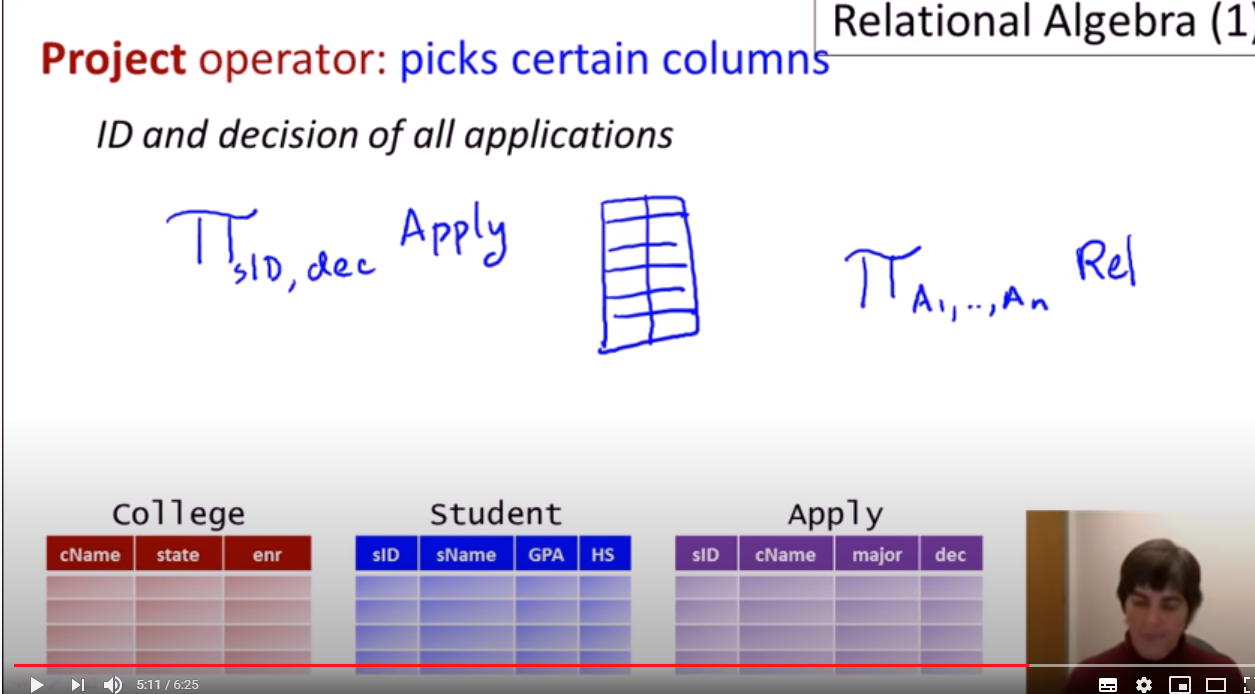
Selection 符号，可以根据条件选出特定的行



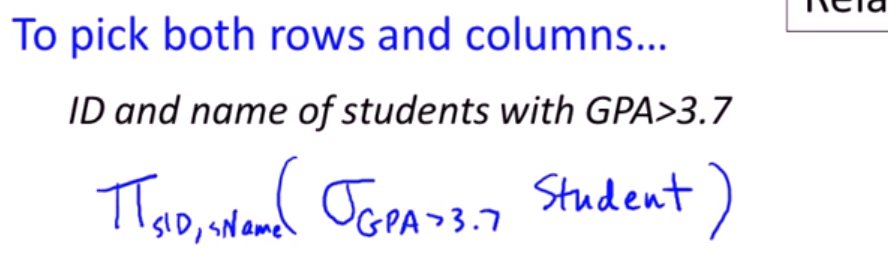
SLECTION右下角写selection判断式， ^就是and 最后加上从哪个table select

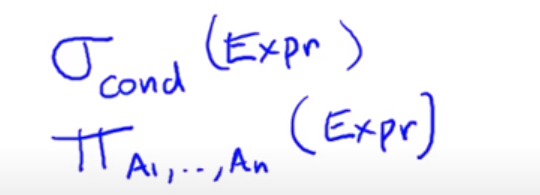


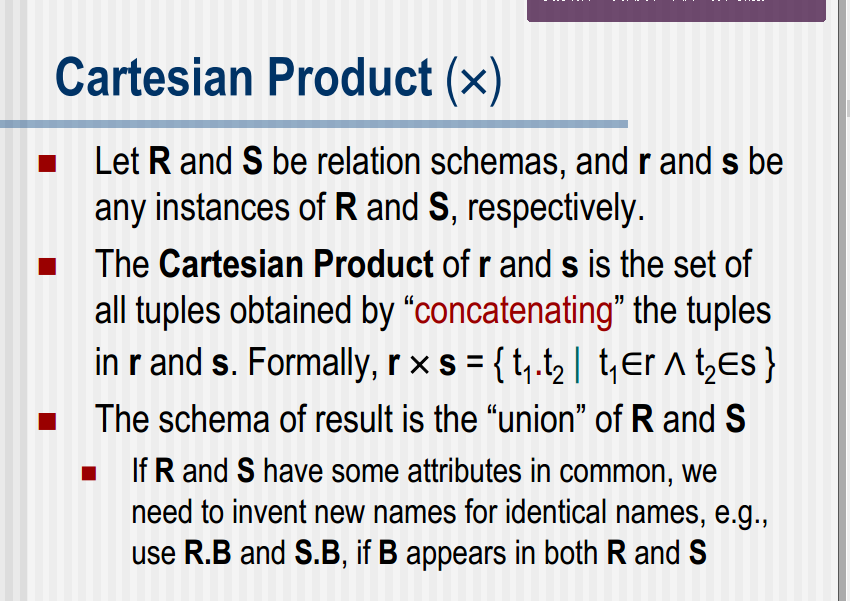
Project，与Selection相对，挑列



怎么同时挑列和行

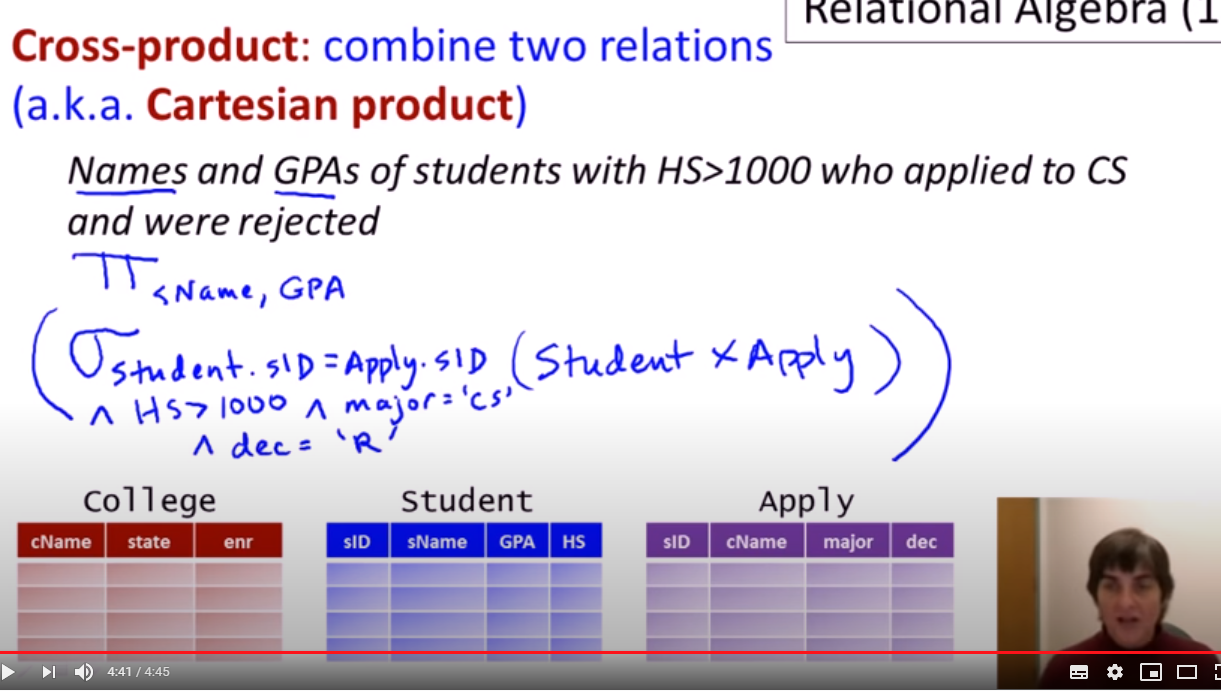
因为RA是基于TABLE，生成table的，所以可以连到一块





X就是把r与s表格接在一起，就是叉乘的组合规律，假设r有a条，s有b条，最后会生成ab条而不是join那样数量不变，

所以为了使用叉乘需要有一些能够连接的attribute



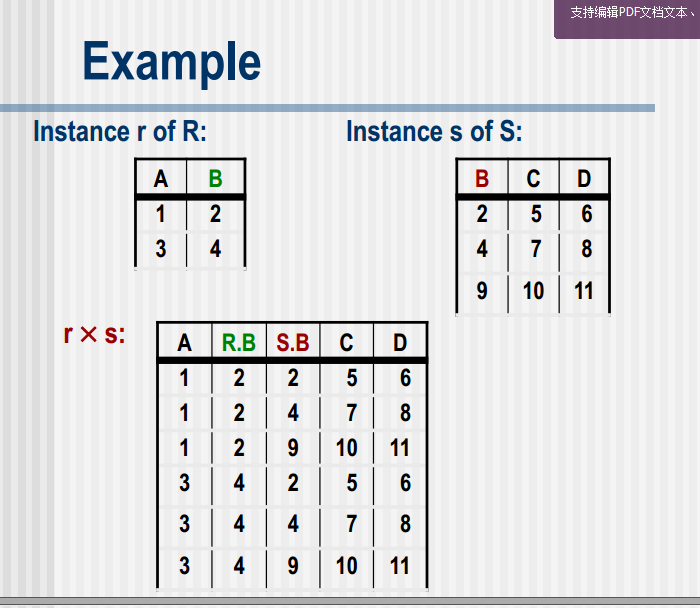
例如这里，用Student与APPLY叉乘以后，会产生16行，但有意义的只有四行，也就是Student.SID=APPLY.SID的时候，

然后就好了

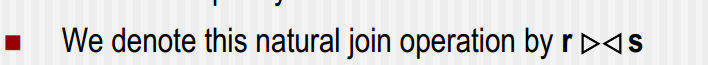
最后挑列挑出来name与GPA

对于R与S那些相同的名字attribute,例如B，

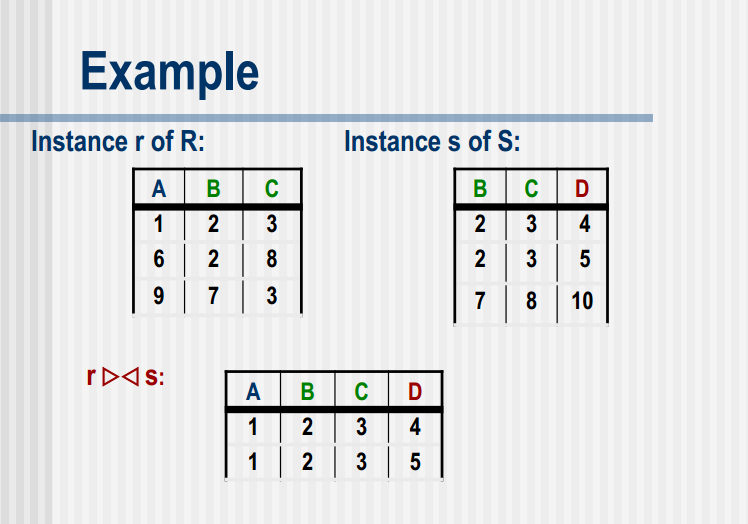
我们使用R.B, S.B

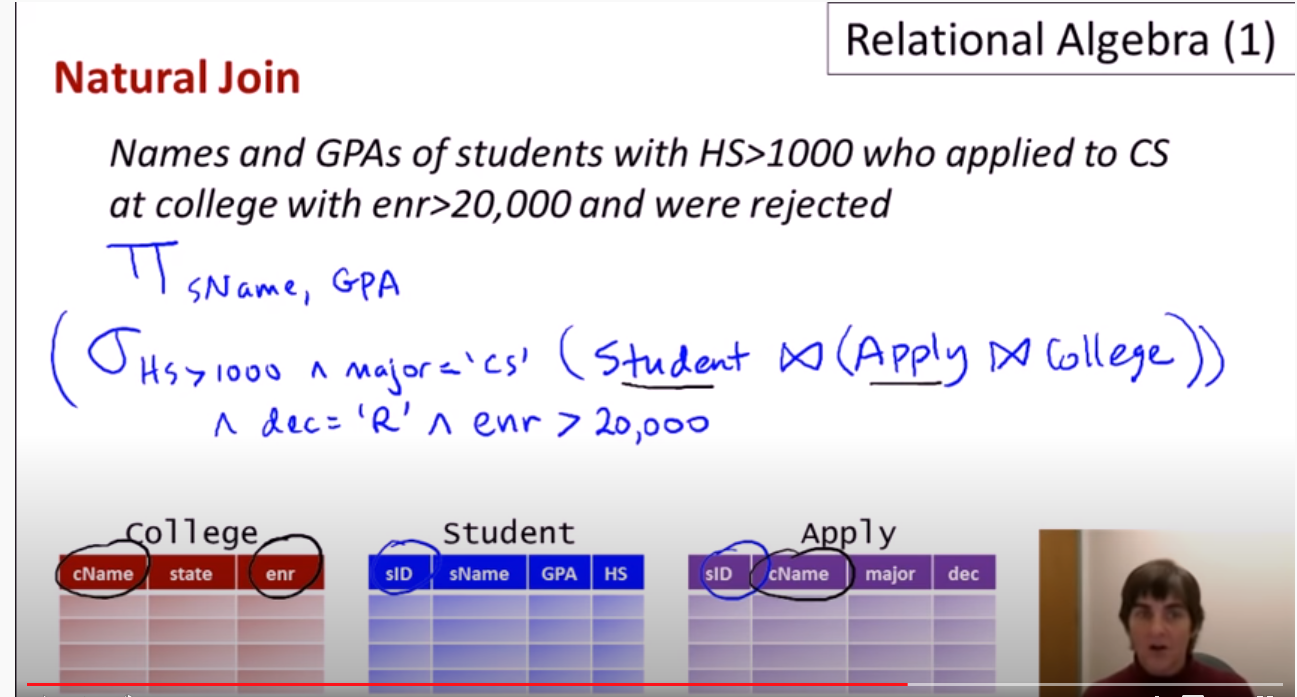


Natural join

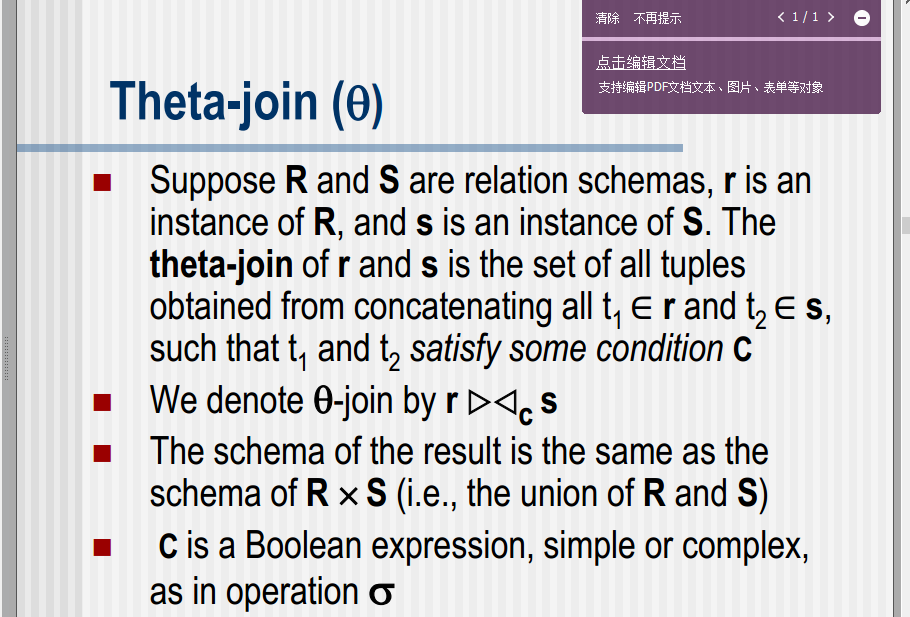


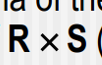
就是找到R与S相等的attribute，因为完全相等，没必要写两遍,如果没有相等的条目，这个自动转化成cartesian product叉乘

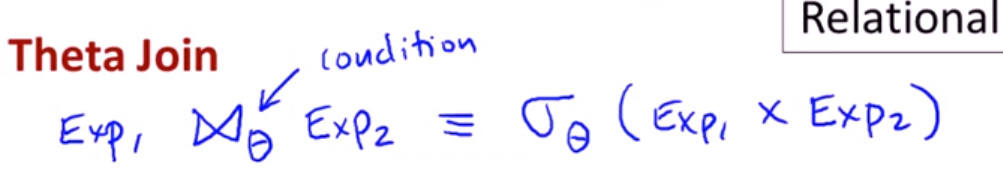


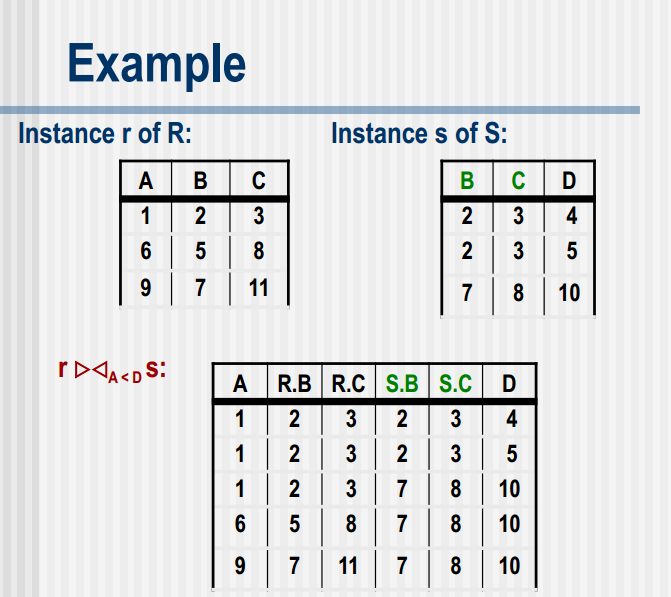


THETA JOIN



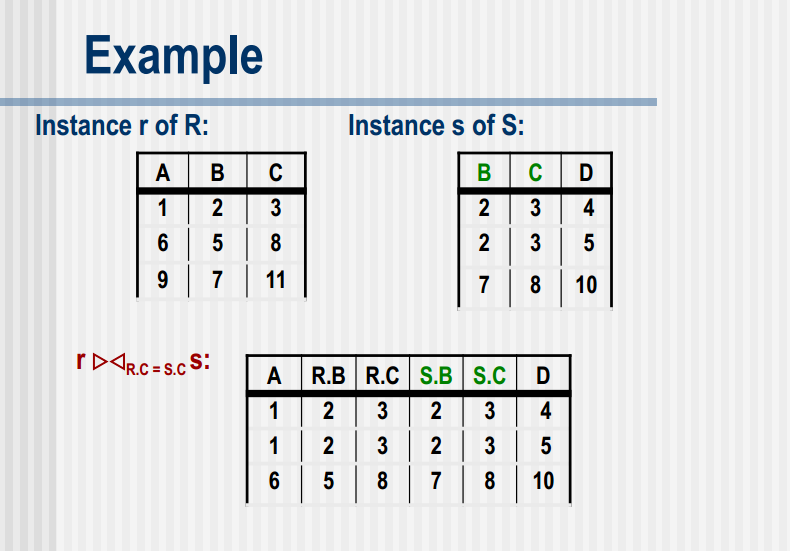
在join符号下面加了一个condition，其实不像join，更像叉乘以后利用这个condition挑行





因为其实实际运行逻辑是叉乘，所以并不会把BC合并

EQUI JOIN， theta join的特殊一种，他的condition只有等于



Renaming符号

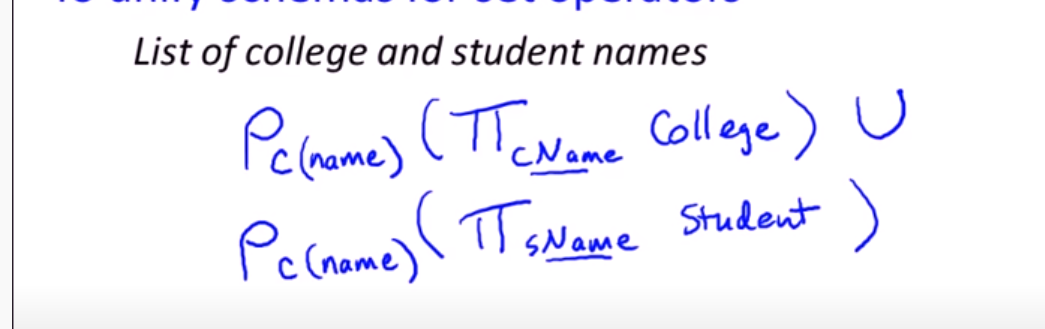
有时候原来的schema不是很好用，所以我们需要额外的在那是rename relation中的某些attribute



下标是要改变的attribute name， r是original input relation instance

结果是一个原来input relation instance的copy，但是attribute名字变了

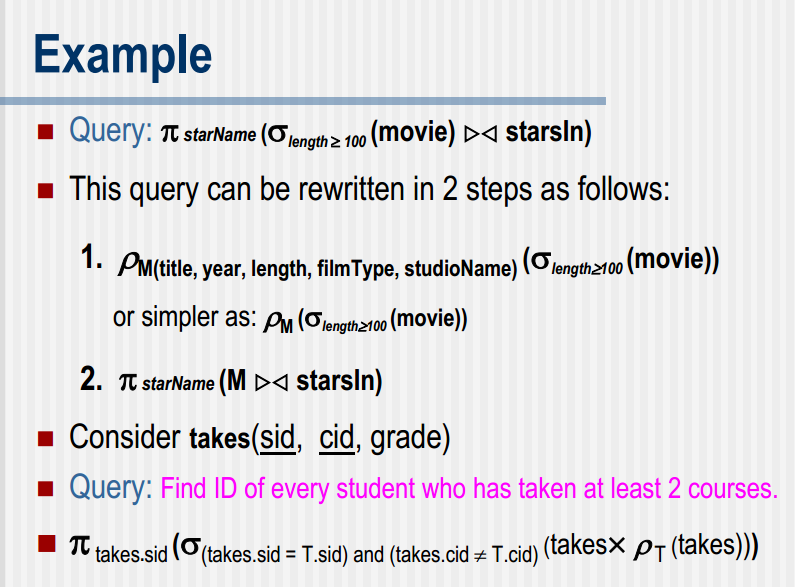
把relation r的名字给一个新名字s



我们想要collecge 与student的 所有学生名字名单，

但是CNAME 与SNAME不能并列，所以重新命名成name

+



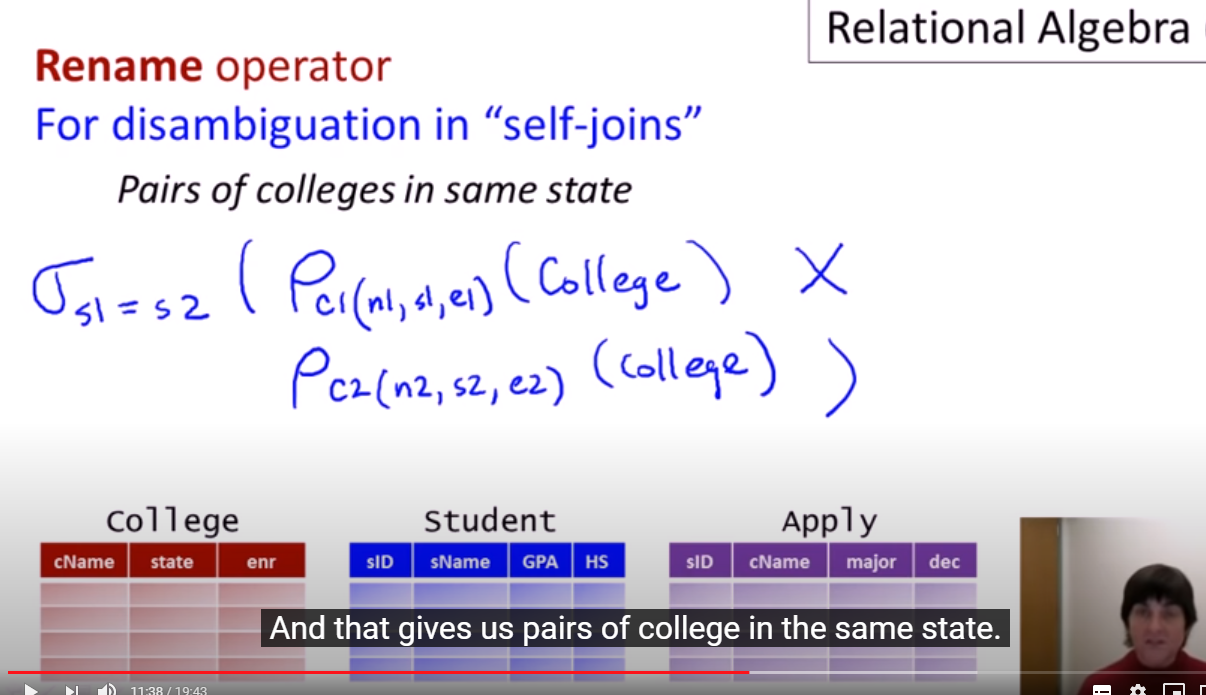
其实老师给的这个例子挺憨的

他就是movie与starsln join一下，挑选出length>=100的行，再调出name

现在先从movie挑出来所有length大于等于100的行，重命名成M，然后相连

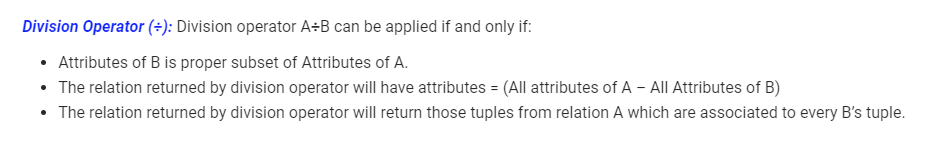
Rename的另一种用法：

self-join，



例如我们希望找到同一洲的大学，那就要自己插自己，但是ATTRIBUTE NAME一样，所以需要重新命名

除以



B的attribute必须是A的子集，

最后return的列是属于A单不属于B的attribute

return 的行是，这些行与B的每一行都有联系，

例如2，他与Badminton有联系，也与Cricket有联系，

所以结果就是2



这里，去掉的列是Support，而1只与Badminton有联系，舍去

