Normal Forms

给你一个relation schema，我们必须决定他是否是一个good relation schema，或者我们需要把它拆解成更小的

判断条件就是normal forms范式

如果你的relation schema是以下范式的一种，我们知道在某种意义上他是个好relation schema

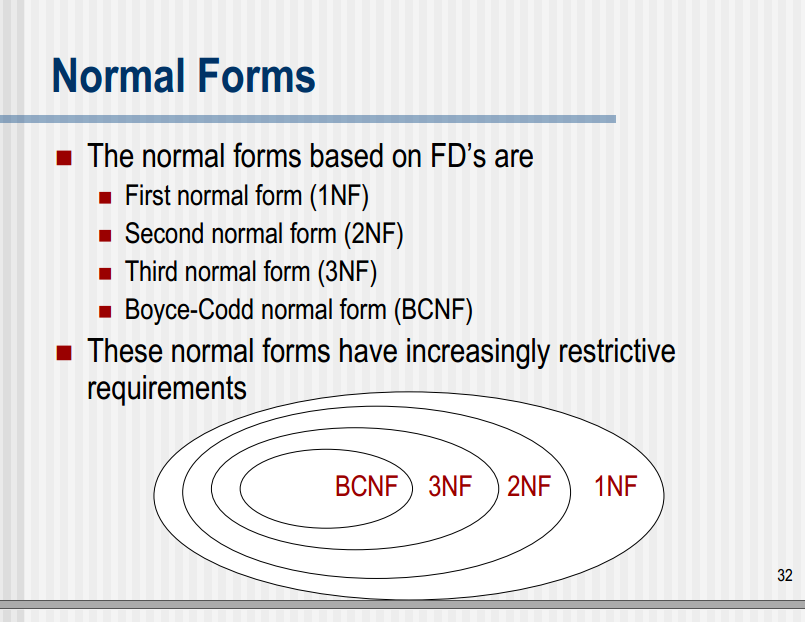
四种：

1NF

2NF

3NFF

BCNF boyce-codd normal form



从上到下，每一个都附加了更严格的条件

原始：

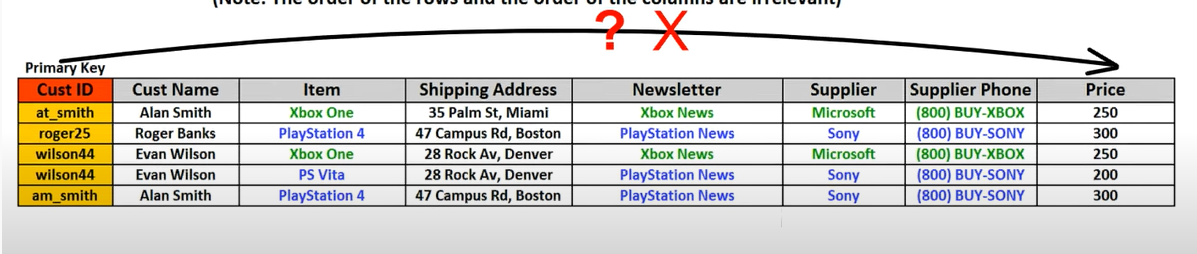


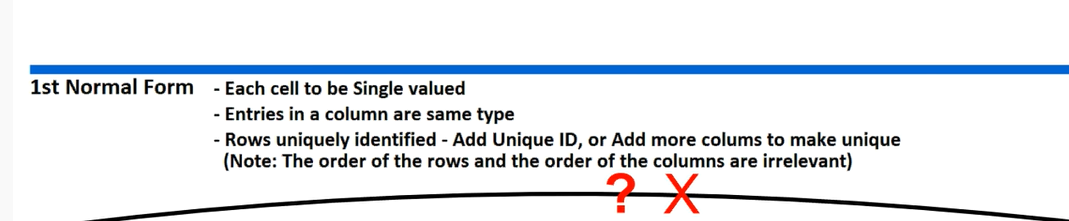
这个不是1NF

第三行，他的item 与newsletter不是**single valued**

他的supplier不是同一类，

1NF:

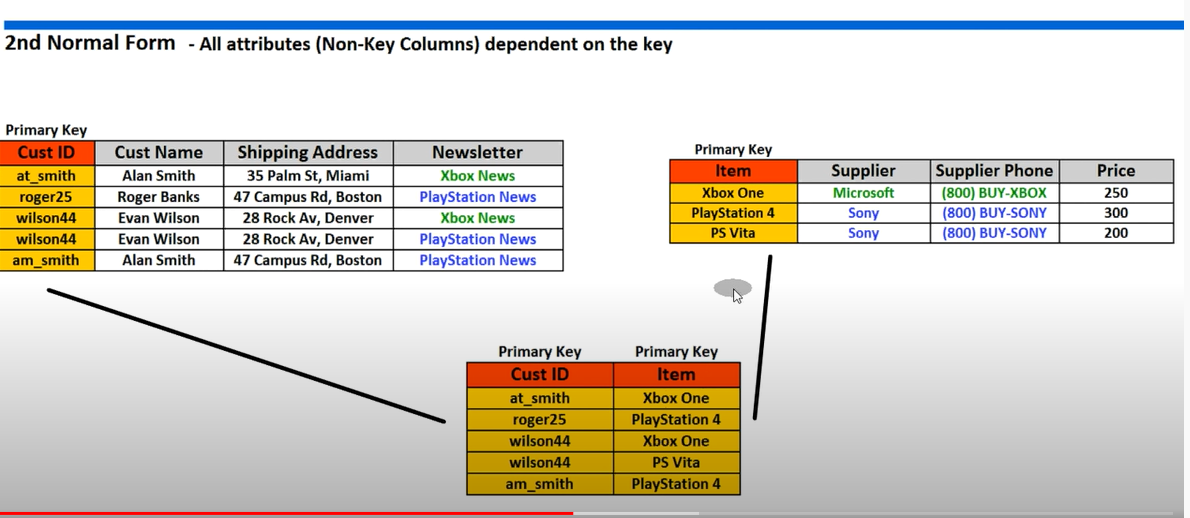




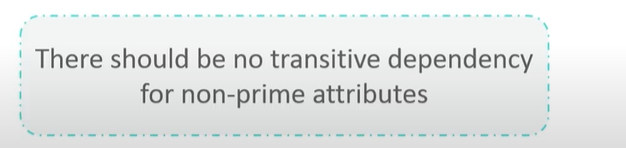
2NF:不存在partial dependency部分依赖

所有attribute依赖于KEY,这里的supplier,item,等信息并不依赖于Custom number,

//所有的东西必须依赖与KEY，3NF是只能依赖于主key



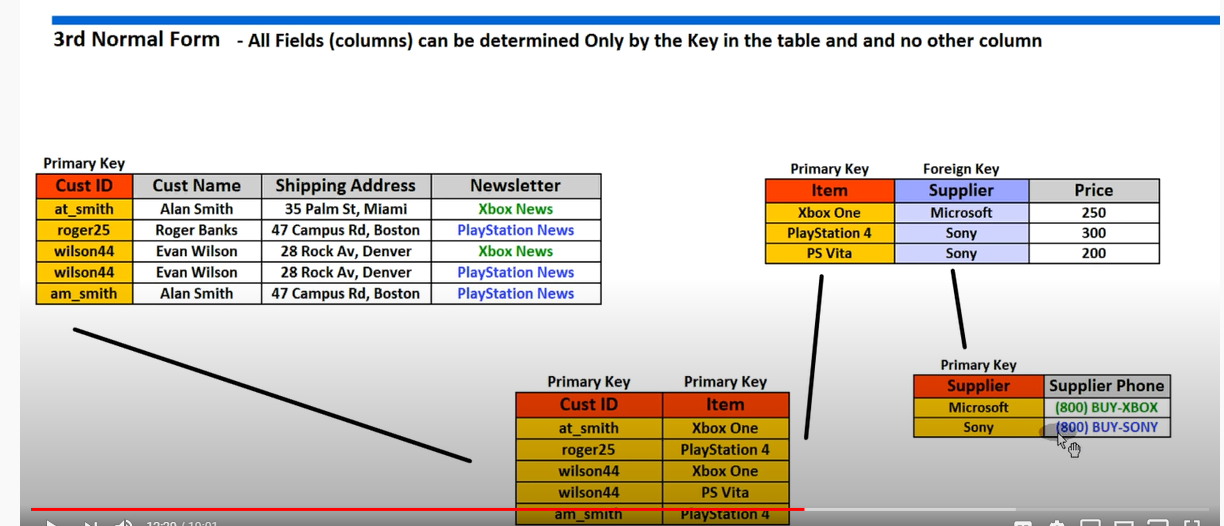
3NF:只能依赖与key



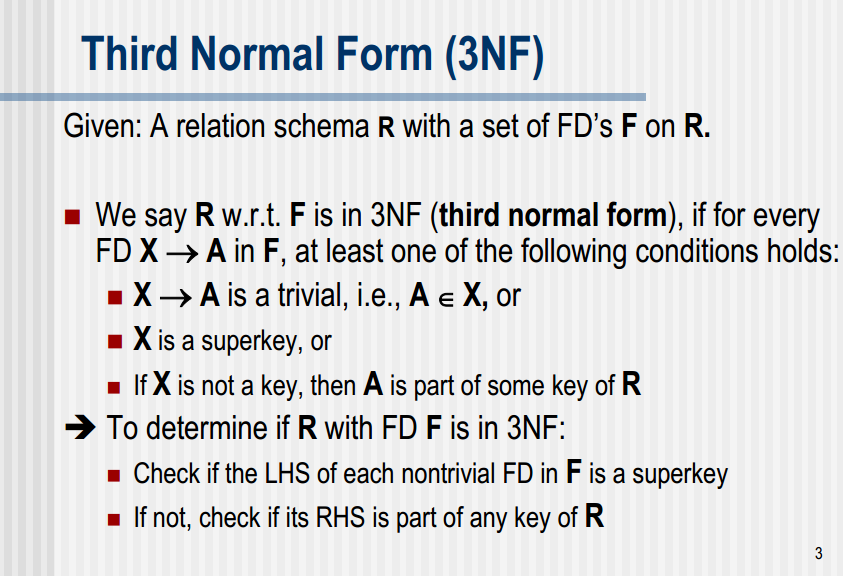
啥叫transitive dependency， ITEM->SUPPLIER, SUPPLIER->Supplier Phone

注意与2nf的区别，2NF你知道item事实上可以得到supplier与supplier phone的，但是

supplier phone也可以由supplier决定，这样就不仅只依靠主key了，因此要新创建一个表



学校的定义：



对于每个FD，至少满足一下一个条件，

X->A是trivial

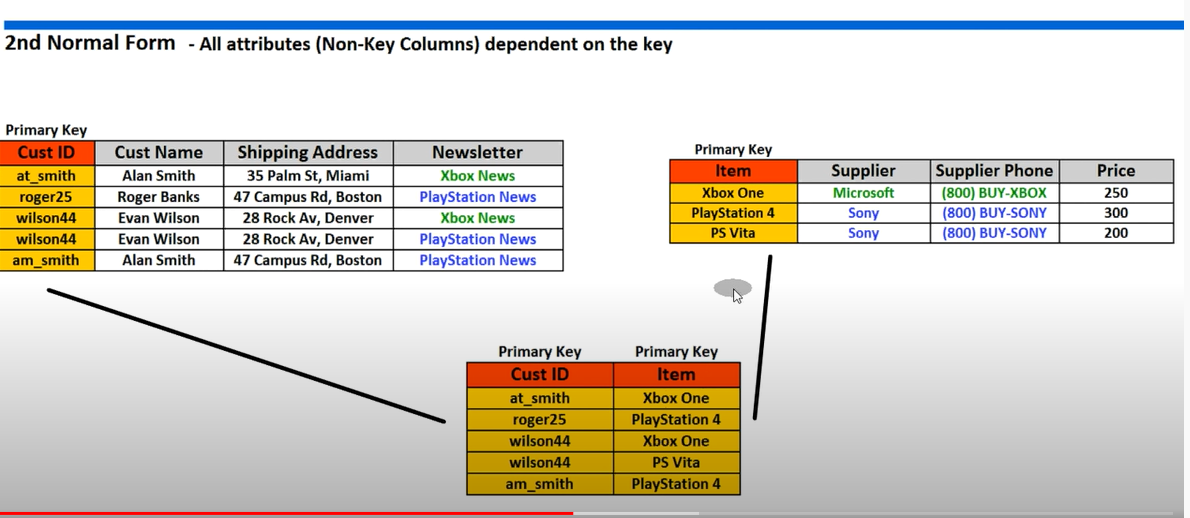
X是superkey(superkey满足key的性质，即决定所有element，但是与candidate key/primary key不同，在于他不用Minimal，他可以加额外信息)

如果X不是key,那么A就是某个key的一部分

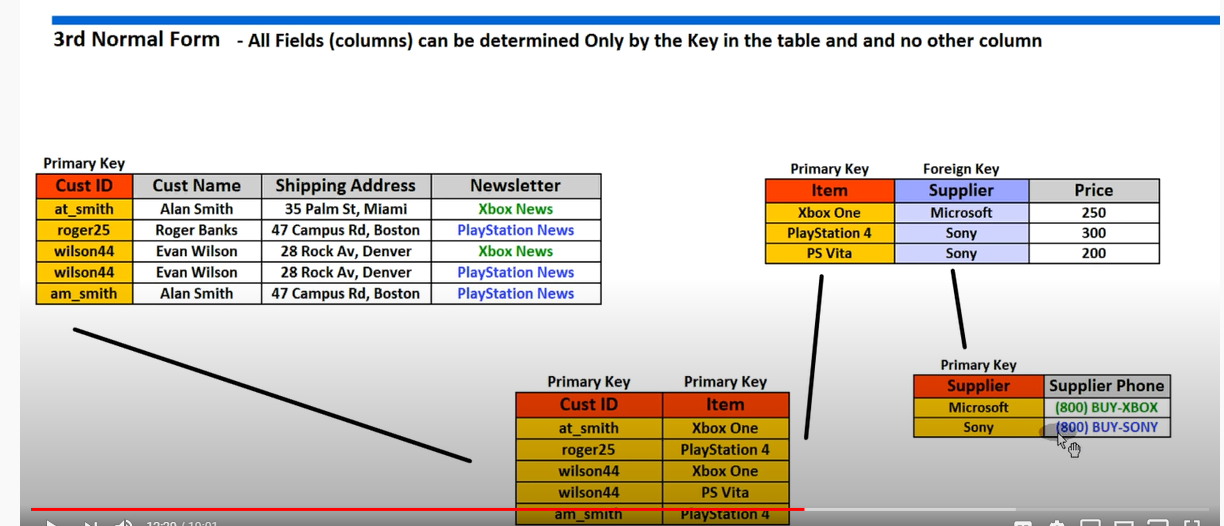
所以为了决定是不是3NF，

看左边是不是superKey，

如果不是，看右边是不是任何Key的一部分



这里Supplier->supplier phone，左边不是superkey，右边不是任何key的一部分



现在supplier左边是superkey了

Boyce-Codd Normal Form

如果每一个FD都满足一下一条

X->A是trivial

或者X是superkey

//去掉了如果X不是key,那么A就是某个key的一部分

所以检查方法，

检查是不是每个左边都是superkey，换句话说，假设左边是X,X+是不是等于R(覆盖所有attribute)

'

拆分成BNCF//一步到位

假设R在1NF状态

我们总能得到一个lossless-join decomposition of R

然而这个decomposition并不总是dependency preserving的

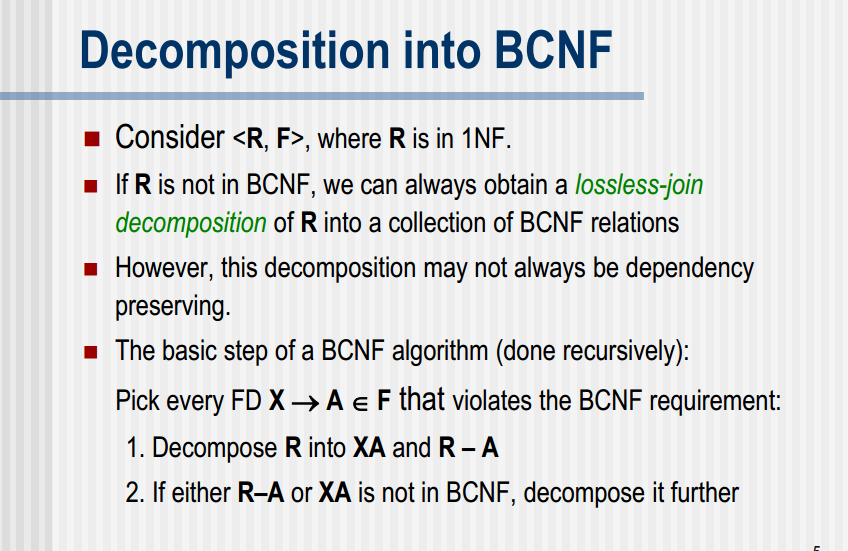
基本算法（recursively,要多重复）

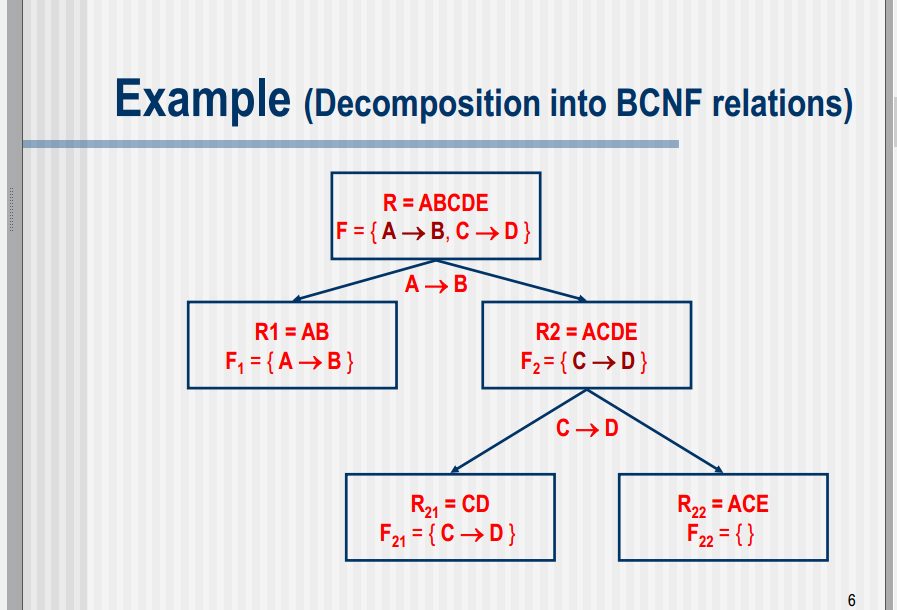
把每一个X->A不符合BNCF的跳出来

把Relation schema拆分成

XA 以及 R-A //还是留下X作为连接的key

如果R-A或是XA不是BCNF，再分解·





A->B,A不是superKey

F是不变的，把AB拆出来作为一个新的R，

这时A是superKey了，

右边去处B，

ACDE

C不是superKey，

CD创建新R，

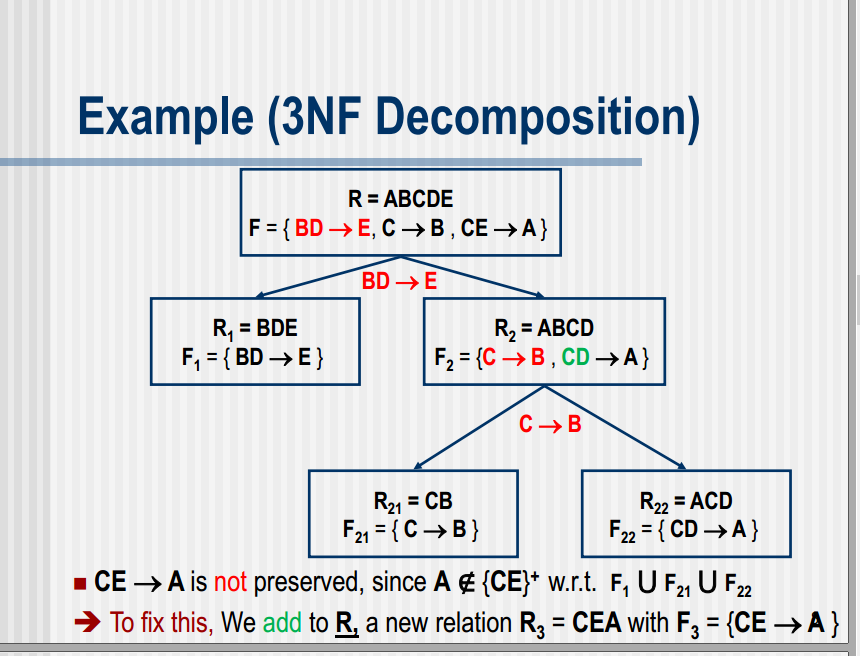
剩下ACE， 没有需要ensure的东西

拆解成3NF

我们总能得到lossless-join,dependency-preserving decomposition 到3NF

第一遍遍历：与BNCF一样

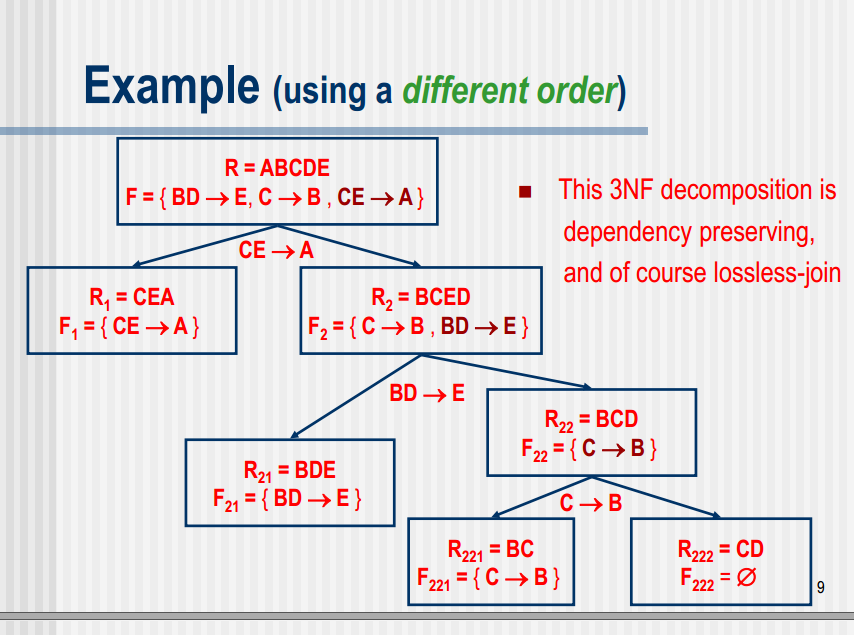
无法确保dependency preserving

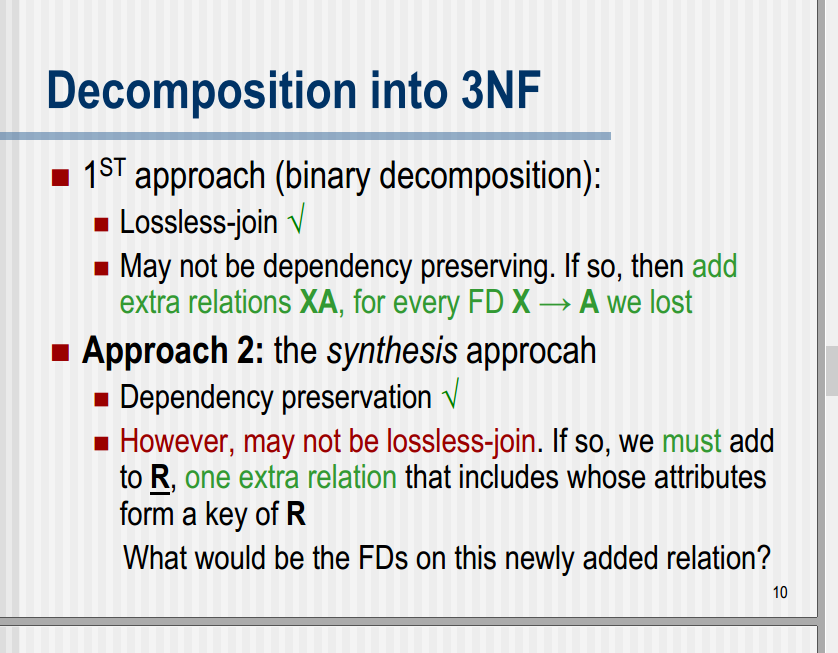


可以看见CE->A没用上

但是好处在于我们能知道是哪个FD没被确保CE->A

第二次遍历先处理CEA

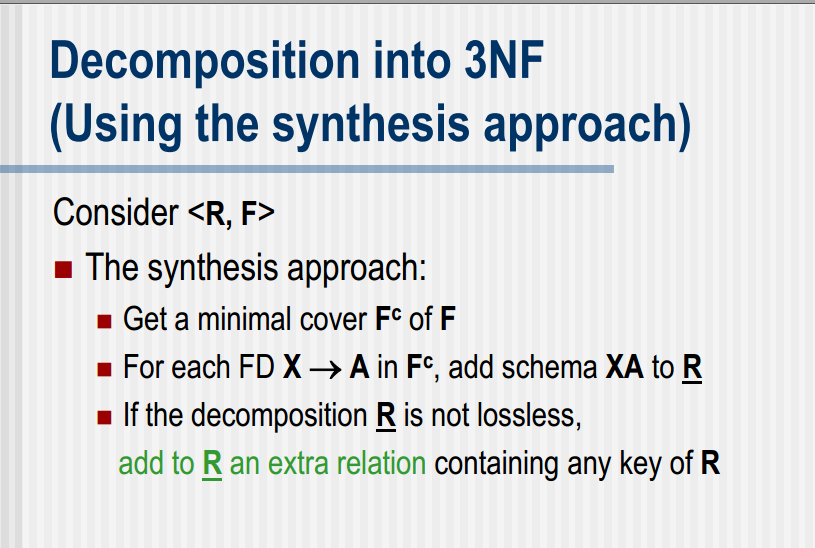




第一次便利只能保证Lossless-join

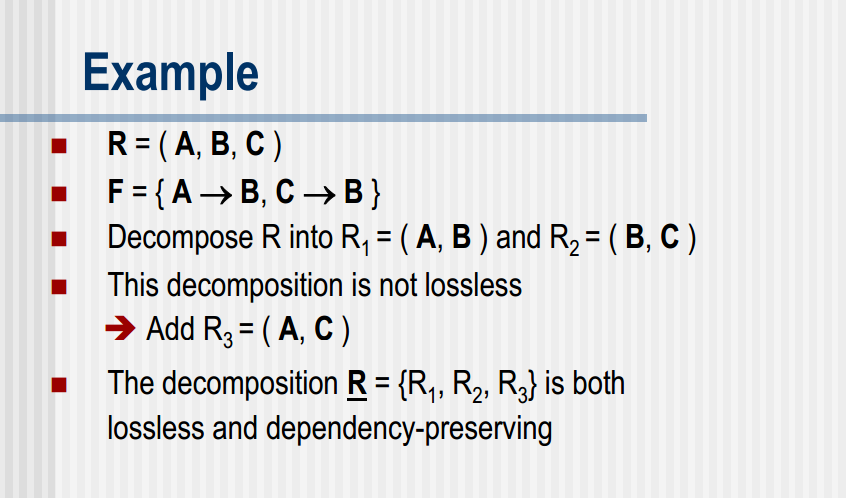
在第二次遍历中，先处理没遍历到的X->A

但是不能保证lossless， 所以我们必须额外加上一个新表格，处理所有为key的attribute



怎么知道要不要加新表格，

看是不是lossless，



这里AB∩BC=B，两个都不能确认，因此要加上新表格，新表格的元素就是原来的KEY AC,这样就能保证AC关系了

Cheeck Lossiness

<https://www.youtube.com/watch?v=6L-AK_Ib17k&t=217s>