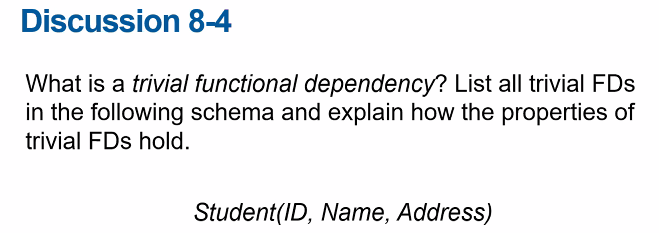
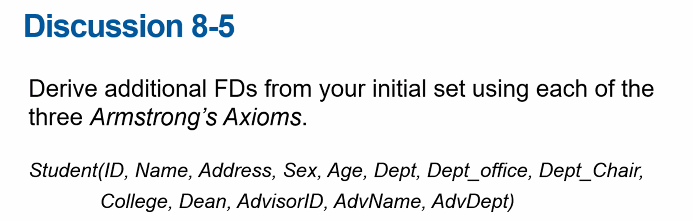
**Discussion 05/04**



넘어감



Dept -> Dept\_office, Dept\_Chair, College

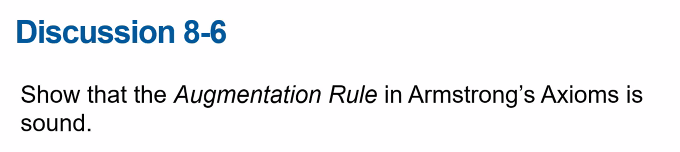
College -> Dean

AdvisorID -> AdvName, AdvDept

Dept Dept\_office -> Dept (by reflexivity)

College Dept -> College Dept\_Chair (by augmentation)

Dept -> Dean (by transitivity)

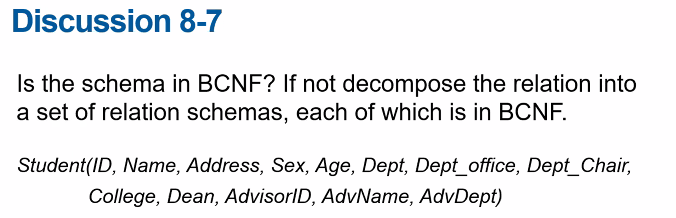


Augmentation Rule: a -> b일 때 ra -> rb.

ra -> rb에서 reflexivity에 의해 ra -> a가 성립하고, rb -> b가 성립한다.

그 다음엔..?

=> FD의 정의에 의해 증명. t1(a) = t2(a) => t1(b) = t2(b) 를 보여야 함.



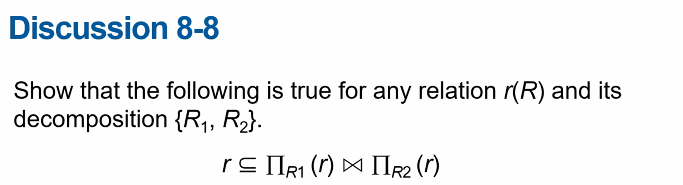
No.

R1 = {ID, Name, Address, Sex, Age, Dept, AdvisorID}

R2 = {Dept, Dept\_Office, Dept\_Chair, College}

R3 = {College, Dean}

R4 = {AdvisorID, AdvName, AdvDept}



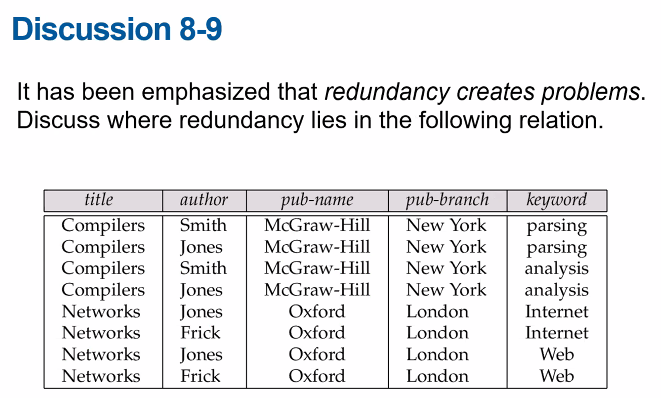
Join 연산은 공유하는 attribute가 없을 경우 cartesian product가 되므로, 모든 경우의 수를 다 포함한다. 따라서 r을 포함한다.

반대로 공유 attribute가 있을 때는 공유하고 있는 attribute에 대해서 모든 경우의 수를 생성하므로, r을 포함한다.

따라서 두 경우 모두 join의 결과가 r을 포함하므로 위의 명제가 성립한다.

=> 교수님 설명: r의 임의의 tuple t가 join 연산에 존재한다는 것을 증명해야 함.

t(R1) t(R2) -> t(R) = t.



Title, author -> title, pub-name

Keyword -> title.

Pub-name -> pub-branch

책과 출판사가 같은 relation에 들어 있으므로 pub-name에만 dependent한 pub-branch가 반복해서 나오는 문제가 생겨 redundancy가 생긴다. 또한 keyword도 다른 relation으로 분리 가능.

=> 교수님 설명: title ->> author, title ->> keyword. (MVD. Multi-valued dependency)

FD가 모든 dependency를 설명하는 건 아님. 하지만 이번 학기에는 MVD는 공부 ㄴㄴ