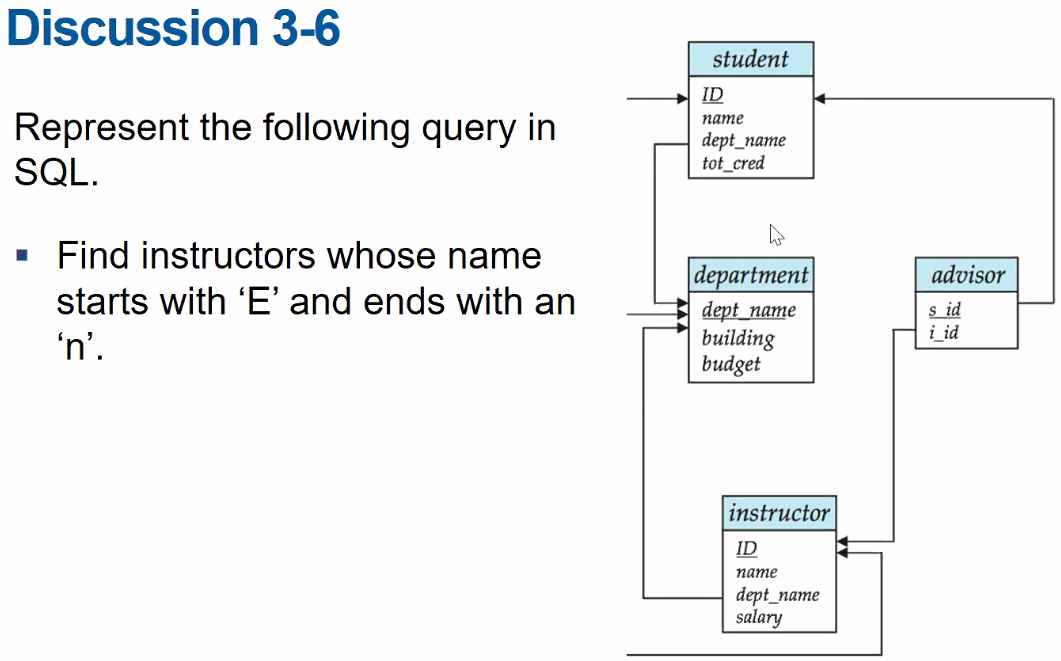
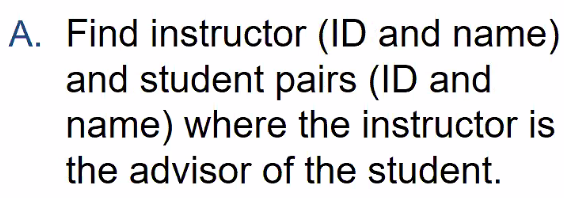
**Discussion 03**



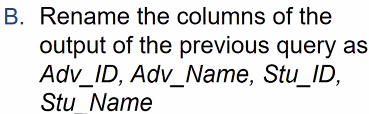
select name from instructor where name like “E%” and name like “%n”

**3-7**

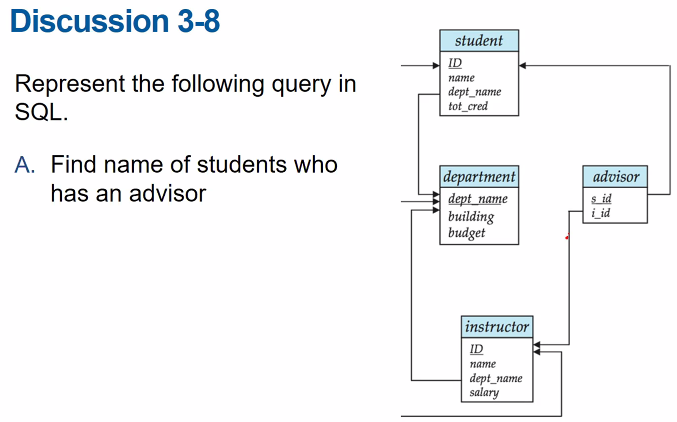


A. Select i.id, i.name, s.id, s.name from student as s, instructor as i, advisor as a where

i.id = a.i\_id and s.id and a.s\_id



B. Select i.id as Adv\_id, i.name as Adv\_Name, s.id as Stu\_id, s.name as Stu\_name 그리고 from 절부터는 같음.

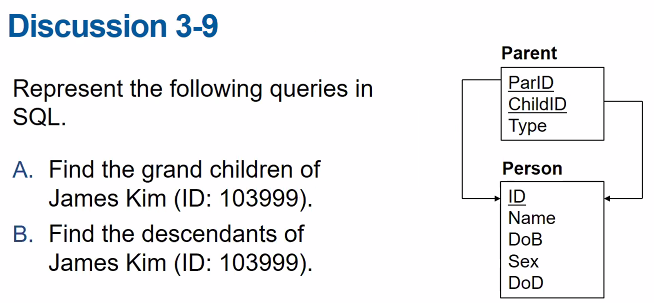


A. select s.name from student as s, advisor as a where a.s\_id = s.id



B. (select id from student) – (select s\_id from advisor)

=> select s.id from student as s where not exists (select \* from advisor where s.id = s\_id) 라는 방법도 있다고 한다.



A. select c.name from person as c, parent as p where p.ParID=103999 and p.ChildID = c.ID

=> 교수님 풀이: select g.ID, g.name from parent p1, parent p2, person g where p1.ParID = 103999 and p1.ChildID = p2.ParID and p2.ChildID = g.ID

**B. 4대손까지를 먼저 구해보고, 그 다음에 모두 구하면 어떻게 하면 될지 생각해보기**

1대손은

Select c.ID, c.name from parent as p, person as c where p.ParID=103999 and p.ChildID = c.ID

N = 1 => p = p1

2대손은

select c.ID, c.name from (select p1.ParID, p2.ChildID from p1, p2 where p1.ChildID = p2.ParID) as p, person as c where p.ParID = 103999 and p.ChildID = c.id

n = 2 => (select p1.ParID, p2.ChildID from p1, p2 where p1.ChildID = p2.ParID) = p1

3대손은

Select c.ID, c.name from (select p1.ParID, p2.ChildID from p1, p2 where p1.ChildID = p2.ParID) as p, person as c where p.ParID=103999 and p.ChildID=c.ID

N = 3 => n이 2일때 계산 값을 p1으로 주고 p2 붙이고..

즉 이전 테이블의 from절 계산 결과를 p1이라고 하고 parent 테이블을 하나 더 붙여서 재귀적으로 계산하면 될 것 같음.

🡺 이걸 SQL 자체로 구할 순 없다고 한다. Transitive closure를 구하려면 programming language의 도움을 받아야 함.