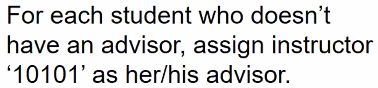


Update department set budget = budget \* 1.2 where building = ‘301’

**3-15**

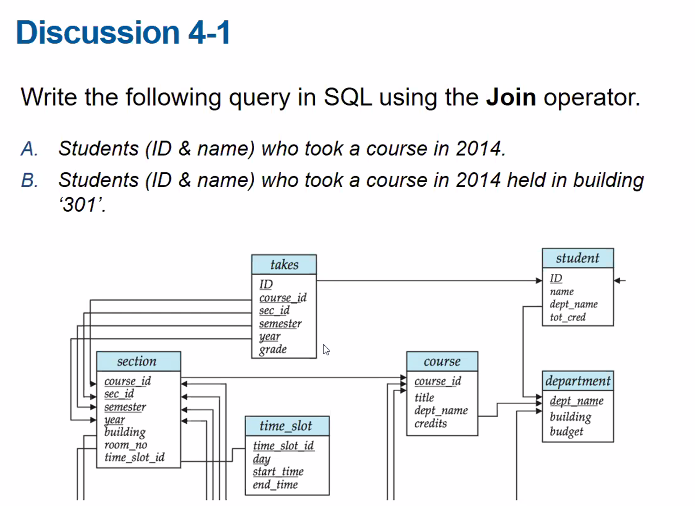
****

Insert into advisor (select ID, ‘10101’ from (select ID from student – select s\_id from advisor))

**3-16**

SQL의 무슨 특징이 DB 언어의 사실상 표준으로 만들었는가?

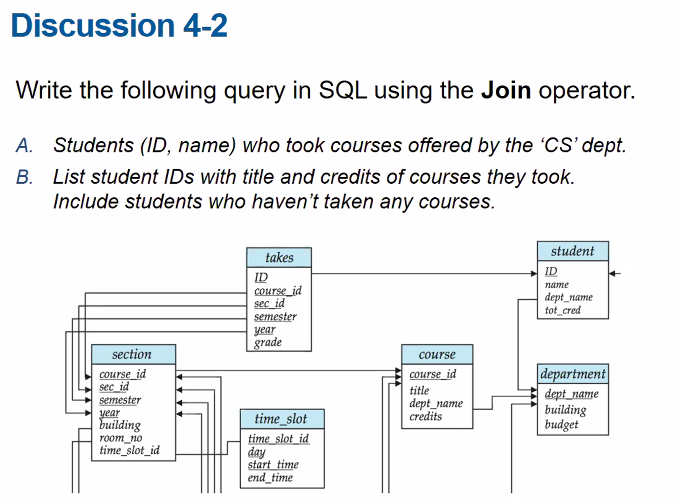
SQL은 관계를 표현하는데 적합한 특징을 가지고 있어 현실세계의 데이터에 적용하기 편했을 것 같다.



A. select ID, name from student join takes on (takes.ID = student.ID) where year=2014.

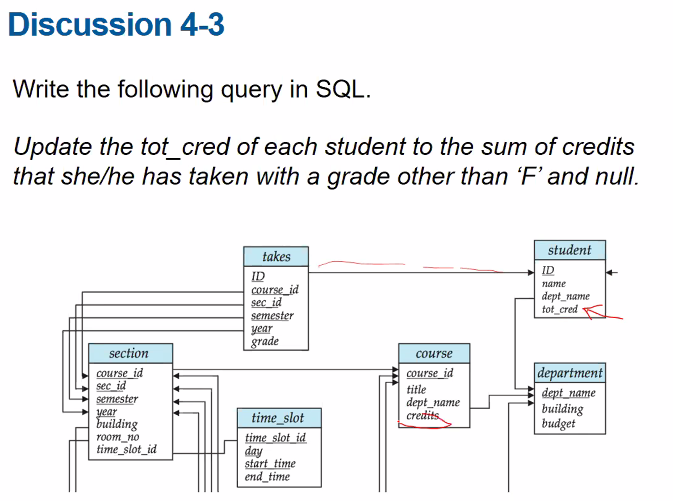
=> Natural join을 사용해서 select distinct ID, name from student natural join takes where year=2014. 로 해도 됨.

B. select ID, name from student (join takes on (takes.ID = student.ID)) join section using (course\_id, sec\_id, semester, year) where building = ‘301’.



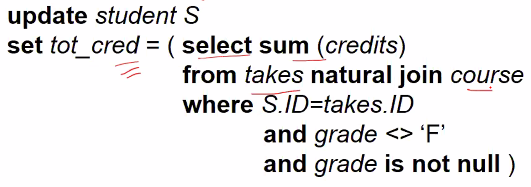
A. select distinct ID, name from student natural join takes natural join section join on (course.course\_id = section.course\_id) where course.dept\_name = ‘CS’

B. select distinct ID, title, credits from student ((natural left outer join takes) natural join section) join using (course\_id) where course.course\_id=section.course\_id



Update student set tot\_cred = select sum(credits) from student natural join takes join course using (course\_id) where grade is not null and grade != ‘F’

=> 중요한 점: sum 등 aggregate function 들은 select 문에만 나옴. 그리고 single record는 record도 될 수 있고 일반 value도 될 수 있음.

답은 이렇다고 한다 => 

이렇게 correlation variable을 가져와야 함. 위의 답 같은 경우 모든 학생의 credit의 sum이 구해지게 됨. Correlation variable을 설정하지 않고 student.ID를 사용할 경우 에러 발생.