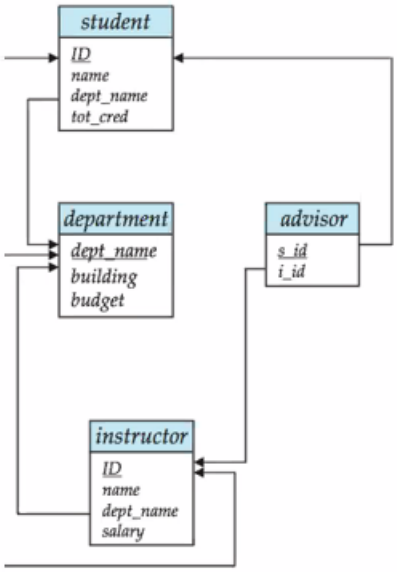
Discussion 3-14

- Represent the following query in SQL.

➢ Raise the budget of departments in building '301' by 20%.



Update department set budget = budget * 1.2 where building = '301'

3-15

For each student who doesn't have an advisor, assign instructor '10101' as her/his advisor.

Insert into advisor (select ID, '10101' from (select ID from student - select s_id from advisor))

3-16

SQL 의 무슨 특징이 DB 언어의 사실상 표준으로 만들었는가?

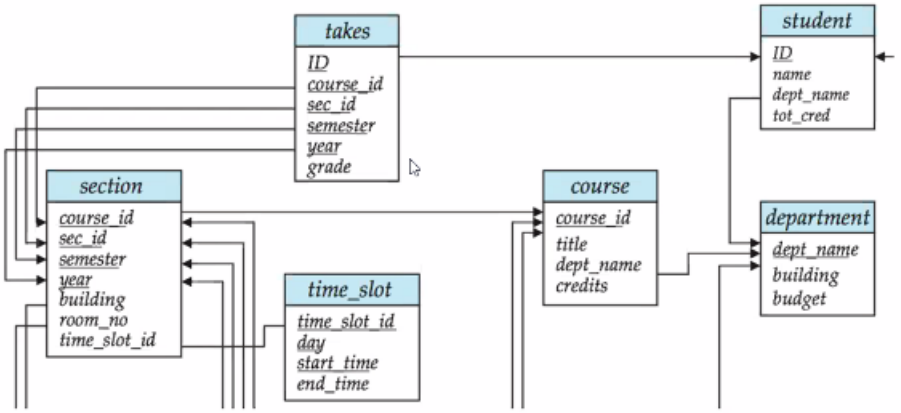
SQL 은 관계를 표현하는데 적합한 특징을 가지고 있어 현실세계의 데이터에 적용하기 편했을 것 같다.

Discussion 4-1

Write the following query in SQL using the **Join** operator.

A. *Students (ID & name) who took a course in 2014.*

B. *Students (ID & name) who took a course in 2014 held in building '301'.*



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

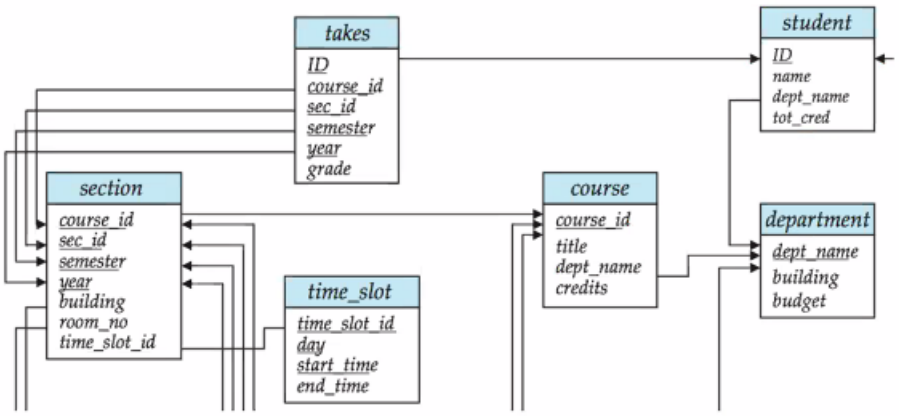
=〉 Natural join 을 사용해서 select distinct ID, name from student natural join takes where year=2014. 로 해도 됨.

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'.

Discussion 4-2

Write the following query in SQL using the **Join** operator.

A. *Students (ID, name) who took courses offered by the 'CS' dept.*
B. *List student IDs with title and credits of courses they took. Include students who haven't taken any courses.*



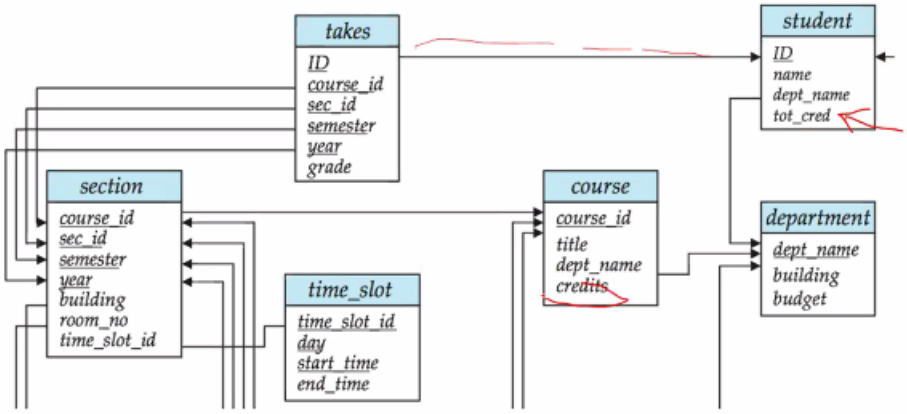
A. select distinct ID, name from student natural join takes natural join section join on (course.course_id = section.course_id) where course.dept_name = 'CS'

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

Discussion 4-3

Write the following query in SQL.

Update the tot_cred of each student to the sum of credits that she/he has taken with a grade other than 'F' and null.



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 record 는 record 도 될 수 있고 일반 value 도 될 수 있음.

```
update student S
set tot_cred = ( select sum (credits)
                 from takes natural join course
                 where S.ID=takes.ID
                       and grade <> 'F'
                       and grade is not null )
```

답은 이렇다고 한다 =>

이렇게 correlation variable 을 가져와야 함. 위의 답 같은 경우 모든 학생의 credit 의 sum 이 구해지게 됨. Correlation variable 을 설정하지 않고 student.ID 를 사용할 경우 에러 발생.