

REPORT

“ 데이터베이스시스템 2차과제 ”



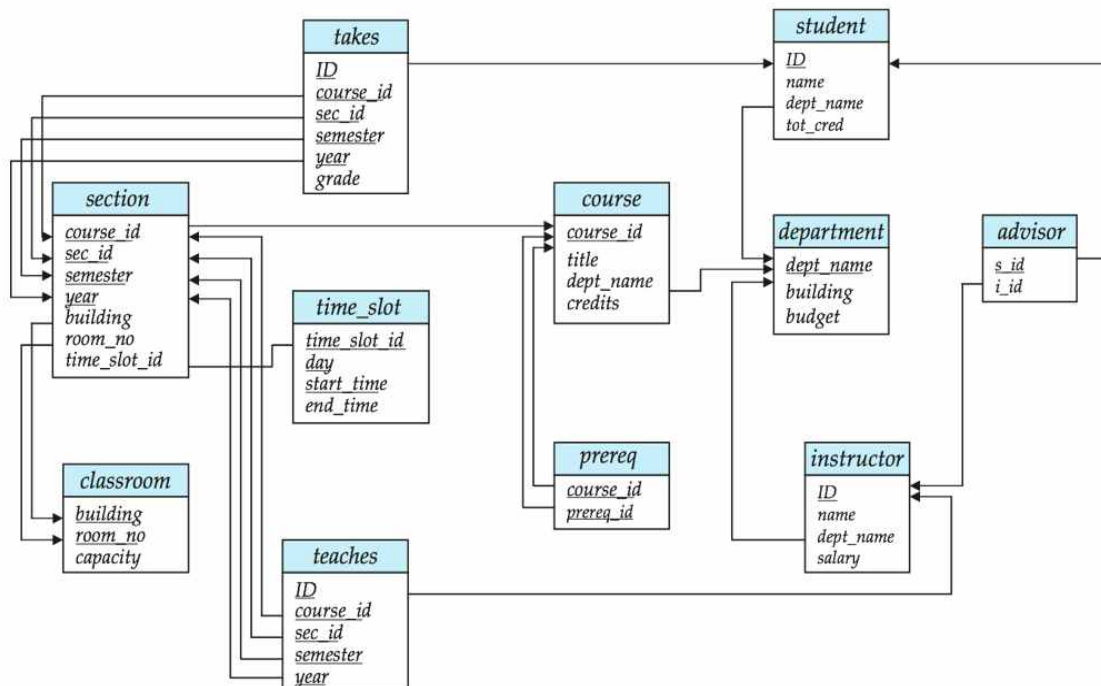
과 목 명	데이터베이스시스템
담당교수	손용락 교수님
학 과	컴퓨터공학과
학 번	2016305078
이 름	최영환
제 출 일	2021.04.22

- ✓ 교재의 University Database 스키마를 참조할 것.
- ✓ 검색 요구를 처리하는 SQL문과 MS SQLSERVER에서 실행한 결과 화면을 제시하시오.
LargeDB를 사용하시오. 검색결과 화면이 한 페이지를 넘을 경우 한 페이지 분량만 제시하시오.

```

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

```



1. 'Comp. Sci.' 학과에 소속된 어떤 학생보다 많은 학점을 이수한 학생의 이름을 검색.

```
-- 1. Comp. Sci. 학과에 소속된 어떤 학생보다 많은 학점을 이수한 학생의 이름을 검색.
-- distinct name 의 경우, 동명이인이 삭제되는 문제점이 존재함.
select distinct S1.ID, S1.name
from student as S1, student as S2
where (S1.tot_cred > S2.tot_cred) and (S2.dept_name = 'Comp. Sci.')
```

	ID	name
1	1000	Manber
2	10033	Zelty
3	10076	Duan
4	1018	Colin
5	10204	Mediratta
6	10267	Rzecz
7	10269	Hilberg
8	10454	Ugarte
9	10481	Grosch
10	10527	Kieras
11	10556	Reed
12	10663	Okaf
13	10693	Zabary
14	107	Shabuno
15	10705	Terauchi

- ▶ distinct S1.name 의 실행 결과에서는 동명이인인 학생들은 중복으로 판단이 되어 삭제됨.
- ▶ 동명이인의 결과도 얻기 위하여, S1.ID를 함께 검색하였음.
- ▶ 결과는 1980개의 행이 출력되었음.

2. 'Comp. Sci. 학과에 소속된 어떤 학생보다 많은 학점을 이수한 학생들 가운데 Comp. Sci.' 학과에 소속되지 않은 학생들의 수를 학과별로 검색.

```
-- 2. Comp. Sci. 학과에 소속된 어떤 학생보다 많은 학점을 이수한 학생들 가운데
-- Comp. Sci. 학과에 소속되지 않은 학생들의 수를 학과별로 검색.
select S1.dept_name, count(distinct S1.ID) as student_cnt
from student as S1, student as S2
where (S1.tot_cred > S2.tot_cred) and (S2.dept_name = 'Comp. Sci.')
group by S1.dept_name
having S1.dept_name != 'Comp. Sci.'
```

	dept_name	student_cnt
1	Elec. Eng.	97
2	Finance	97
3	Astronomy	105
4	Cybernetics	86
5	Accounting	97
6	Marketing	85
7	Math	90
8	Statistics	83
9	Athletics	91
10	Biology	98
11	English	95
12	Psychology	97
13	Pol. Sci.	109
14	History	117
15	Mech. Eng.	105
16	Physics	95
17	Civil Eng.	118
18	Geology	92
19	Languages	116

3. 'Accounting' 학과에 소속된 학생들 중 이름이 'Ch'로 시작하는 학생들에 대한 모든 정보를 검색.

-- 3. Accounting 학과에 소속된 학생들 중 이름이 Ch 로 시작하는 학생들에 대한 모든 정보를 검색.

```
select *
from student
where (dept_name = 'Accounting') and (name like 'Ch%')
```

	ID	name	dept_name	tot_cred
1	15249	Cheah	Accounting	1
2	62373	Cheed	Accounting	60
3	67021	Chenu	Accounting	73
4	83314	Chow	Accounting	40
5	95099	Chien	Accounting	81

4. 예산이 600000 이상 800000 이하인 학과의 학과명과 예산을 예산값 내림차순 기준으로 검색

-- 4. 예산이 600000 이상 800000 이하인 학과의 학과명과 예산을 예산값 내림차순 기준으로 검색.

```
select dept_name, budget
from department
where budget between 600000 and 800000
order by budget desc
```

	dept_name	budget
1	Cybernetics	794541.46
2	Math	777605.11
3	Athletics	734550.70
4	History	699140.86
5	Biology	647610.55
6	Astronomy	617253.94
7	English	611042.66
8	Languages	601283.60

5. 2006학년도 봄학기에는 강의를 맡았지만 2006학년도 가을학기에는 강의를 맡지 않은 교수들의 교수번호를 검색

-- 5. 2006학년도 봄학기에는 강의를 맡았지만 2006학년도 가을학기에는 강의를 맡지 않은 교수들의 교수번호를 검색.

```
select ID
from teaches
where semester = 'Spring' and year = 2006
except
select ID
from teaches
where semester = 'Fall' and year = 2009
```

	ID
1	22591

6번 문제를 풀기위하여 아래 update SQL을 작성, 실행하였음.

(‘Pol. Sci.’ 학과 교수들의 연봉을 null로 변경하는 작업)

```
update instructor
set salary = null
where dept_name = 'Pol. Sci.' or dept_name = 'Comp. Sci.'
```

6. 아직 연봉이 정해지지 않은(연봉이 NULL을 가지는) 교수들의 수를 학과별로 검색

-- 6. 아직 연봉이 정해지지 않은(연봉이 NULL을 가지는) 교수들의 수를 학과별로 검색

```
select dept_name, count(*) as instructor_cnt
from instructor
where salary is null
group by dept_name
```

	dept_name	instructor_cnt
1	Comp. Sci.	2
2	Pol. Sci.	3

7. 학년도, 학기 기준으로 수강신청하여 이수한 학생 수를 검색. (수강신청 후 아직 이수완료하지 않은 경우 grade는 null을 가짐. grade가 'F'인 경우 해당 강좌를 이수하지 않은 것이 됨.)

-- 7. 학년도, 학기 기준으로 수강신청하여 이수한 학생 수를 검색
-- (수강 신청 후 아직 이수완료하지 않은 경우 grade는 null을 가짐.
-- grade가 'F'인 경우 해당 강좌를 이수하지 않은 것이 됨.)

```
select year, semester, count(distinct ID) as std_cnt
from takes
where (grade is not null) and (grade != 'F')
group by year, semester
order by year, semester desc
```

	year	semester	std_cnt
1	2001	Spring	774
2	2001	Fall	560
3	2002	Spring	898
4	2002	Fall	1562
5	2003	Spring	1257
6	2003	Fall	1278
7	2004	Spring	966
8	2004	Fall	747
9	2005	Spring	950
10	2005	Fall	998
11	2006	Spring	1107
12	2006	Fall	1464
13	2007	Spring	1252
14	2007	Fall	1239
15	2008	Spring	1573
16	2008	Fall	291
17	2009	Spring	533
18	2009	Fall	1379
19	2010	Spring	1249
20	2010	Fall	1067

8. 학과별 지도학생 수를 검색

-- 8. 학과별 지도학생 수를 검색

```
select instructor.dept_name, count(s_id) as std_cnt
from advisor, instructor
where i_id = ID
group by instructor.dept_name
```

	dept_name	std_cnt
1	Accounting	159
2	Astronomy	34
3	Athletics	213
4	Biology	83
5	Comp. Sci.	65
6	Cybernetics	141
7	Elec. Eng.	171
8	English	170
9	Finance	46
10	Geology	54
11	Languages	97
12	Marketing	159
13	Mech. Eng.	86
14	Physics	70
15	Pol. Sci.	130
16	Psychology	94
17	Statistics	228

9. 50명 이상 학생들을 지도하는 교수들의 교수번호와 지도학생 수를 검색.

-- 9. 50명 이상 학생들을 지도하는 교수들의 교수번호와 지도학생 수를 검색

```
select i_id, count(s_id) as s_cnt
from advisor, instructor
where i_id = ID
group by i_id
having count(s_id) >= 50
```

	i_id	s_cnt
1	35579	54
2	97302	50
3	37687	50
4	77346	54

10. 학년도, 강의실(building, room_number) 기준으로 해당 학년도, 강의실에서 진행된 강좌에 참여한 학생 수를 검색

```
-- 10. 학년도, 강의실(building, room_number) 기준으로 해당 학년도, 강의실에서 진행된 강좌에 참여한 학생 수를 검색
select takes.year, building, room_number, count(distinct ID) as std_cnt
from section, takes
where (takes.course_id = section.course_id) and (takes.semester = section.semester) and (section.sec_id = takes.sec_id)
group by takes.year, building, room_number, takes.course_id, takes.sec_id, takes.semester
order by takes.course_id
```

	year	building	room_number	std_cnt
1	2002	Taylor	183	307
2	2009	Chandler	375	327
3	2002	Fairchild	145	304
4	2008	Taylor	812	286
5	2008	Whitman	434	291
6	2002	Drown	757	280
7	2007	Gates	314	300
8	2002	Polya	808	338
9	2002	Chandler	375	292
10	2007	Saucon	180	299
11	2008	Power	717	302
12	2009	Fairchild	145	311
13	2006	Taylor	183	328
14	2009	Fairchild	145	295
15	2007	Main	45	294