



고급 SQL: 과제



Database Laboratory

과제

▶ 과제 1 - 1.

- ▶ 2주차 실습 PPT와 영상에 나오는 SQL Query를 직접 입력하여 실행해 보고 해당 쿼리에 대한 설명과 결과를 Word나 한글 파일로 정리해 제출하세요.
 - ▶ 12p. 하단. 재명명 연산
 - ▶ 13p. 스트링 연산
 - ▶ 14p. 데이터 출력의 순서화
 - ▶ 16p. UNION, UNION ALL (합집합)
 - ▶ 17p. INTERSECT (교집합)
 - ▶ 18p. EXCEPT (차집합)
 - ▶ 20p. 카티전 곱
 - ▶ 21p. INNER JOIN (WHERE 절을 이용)
 - ▶ 22p. INNER JOIN (INNER JOIN과 ON을 이용)
 - ▶ 23p. OUTER JOIN(LEFT)
 - ▶ 24p. OUTER JOIN(RIGHT)

과제

▶ 과제 1 - 2.

- ▶ 3주차 실습 PPT에 나오는 SQL Query를 직접 입력하여 실행해 보고 해당 쿼리에 대한 설명과 결과를 Word나 한글 파일로 정리해 제출하세요.
 - ▶ 4p. ~ 6p. 집계함수 (AVG, MIN, MAX, SUM, COUNT)
 - ▶ 7p. GROUP BY, HAVING 절
 - ▶ 9p. IN & NOT IN
 - ▶ 10p. SOME & ALL
 - ▶ 11p. EXISTS & NOT EXISTS
 - ▶ 12p. FROM 절의 부질의

과제

▶ 과제 2.

- ▶ 다음 슬라이드의 테이블을 참고하여 아래의 데이터를 검색하는 SQL 쿼리를 작성하고 결과를 파일로 정리해 제출하세요.
 - ▶ 1. section 테이블을 이용하여 2009년도에 열렸던 모든 과목의 코드(c_id)를 중복없이 검색하세요.
 - ▶ 2. student와 takes 테이블을 이용하여 'Levy'이라는 학생이 들었던 과목의 코드(c_id)와 성적(grade)을 모두 검색하세요. (join 사용)
 - ▶ 3. section과 teaches 테이블을 이용하여 'Taylor' 건물에서 열린 수업을 진행한 강사의 코드(i_id)를 모두 검색하세요. (중첩 부질의 사용)
 - ▶ 4. takes 와 course 테이블을 이용하여 각 학생들이 들었던 모든 과목에 대해서 학생(s_id) 별로 취득한 총 학점수(누적된 학점수)를 구하십시오. (다음 그림 참조)

결과		메시지
	s_id	모든학점
1	00128	7
2	12345	14
3	19991	3
4	23121	3
5	44553	4
6	45678	11
7	54321	8

예시 데이터

▶ department

```
CREATE TABLE department (  
    dept_name    VARCHAR(20),  
    building     VARCHAR(15),  
    budget       INT,  
    PRIMARY KEY(dept_name)  
);
```

dept_name	building	budget
Comp.Sci.	Taylor	100000
Biology	Watson	90000
Elec.Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

예시 데이터

▶ instructor

```
CREATE TABLE instructor (  
    id          VARCHAR(5),  
    name        VARCHAR(20) NOT NULL,  
    dept_name    VARCHAR(20),  
    salary       INT,  
    PRIMARY KEY(id)  
);
```

id	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp.Sci.	75000
98345	Kim	Elec.Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp.Sci.	65000
58583	Clifieri	History	62000
83821	Brandt	Comp.Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

예시 데이터

▶ student

```
CREATE TABLE student (  
    id          VARCHAR(5),  
    name        VARCHAR(20) NOT NULL,  
    dept_name   VARCHAR(20),  
    tot_cred    INT,  
    PRIMARY KEY(id)  
);
```

id	name	dept_name	tot_cred
00128	Zhang	Comp.Sci.	102
12345	Shankar	Comp.Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp.Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp.Sci.	58
76653	Aoi	Elec.Eng.	60
98765	Bourikas	Elec.Eng.	98

과제

▶ course

```
CREATE TABLE course (  
    id VARCHAR(7),  
    title VARCHAR(100),  
    dept_name VARCHAR(20),  
    credits INT,  
    PRIMARY KEY(id)  
);
```

id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Computational Biology	Biology	4
BIO-399	Genetics	Biology	3
CS-101	Intro. to Computer Science	Comp.Sci.	4
CS-190	Game Design	Comp.Sci.	4
CS-315	Robotics	Comp.Sci.	3
CS-319	Image Processing	Comp.Sci.	3
CS-347	Database System Concepts	Comp.Sci.	3
EE-181	Intro. to Digital System	Ekec.Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physycal Principles	Physics	4

과제

▶ section

sec_id	c_id	semester	years	building	room_num	time_slot_id
1	BIO-101	Summer	2009	Painter	514	B
1	BIO-301	Summer	2010	Painter	514	A
1	CS-101	Fall	2009	Packard	101	H
1	CS-101	Spring	2010	Packard	101	F
1	CS-190	Spring	2009	Taylor	3128	E
2	CS-190	Spring	2009	Taylor	3128	A
1	CS-315	Spring	2010	Watson	120	D
1	CS-319	Spring	2010	Watson	100	B
2	CS-319	Spring	2010	Taylor	3128	C
1	CS-347	Fall	2009	Taylor	3128	A
1	EE-181	Spring	2009	Taylor	3128	C
1	FIN-201	Spring	2010	Packard	101	B
1	HIS-351	Spring	2010	Painter	514	C
1	MU-199	Spring	2010	Packard	101	D
1	PHY-101	Fall	2009	Watson	100	A

```
CREATE TABLE section (  
    sec_id          VARCHAR(1),  
    c_id            VARCHAR(7),  
    semester        VARCHAR(10),  
    years           VARCHAR(4),  
    building         VARCHAR(15),  
    room_num        VARCHAR(5),  
    time_slot_id    VARCHAR(1)  
    [PRIMARY KEY 선행...]  
);
```

과제

▶ teaches

i_id	c_id	sec_id	semester	years
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

```
CREATE TABLE teaches (  
    i_id      VARCHAR(5),  
    c_id      VARCHAR(7),  
    sec_id    VARCHAR(1),  
    semester  VARCHAR(10),  
    year      VARCHAR(4)  
    [PRIMARY KEY 생략...]  
);
```

과제

▶ takes

s_id	c_id	sec_id	semester	years	grade
00128	CS-101	1	Fall	2009	A
00128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+

```
CREATE TABLE takes (  
    s_id      VARCHAR(5),  
    c_id      VARCHAR(7),  
    sec_id    VARCHAR(1),  
    semester  VARCHAR(10),  
    year      VARCHAR(4),  
    grade     VARCHAR(2)  
    [PRIMARY KEY 생략...]  
);
```