

## 4주차 실습 과제

20185109 길홍배

1.1 2 주차 실습 PPT 와 영상에 나오는 SQL Query 를 직접 입력하여 실행해 보고 해당 쿼리에 대한 설명과 결과를 Word 나 한글 파일로 정리해 제출하세요.

(1) 데이터 입력

dept_name	building	budget	id	name	dept_name	salary	id	name	dept_name	tot_cred
Biology	Watson	90000	10101	Srinivasan	Comp.Sci.	65000	12345	Shankar	Comp.Sci.	32
Comp.Sci	Taylor	100000	12121	Wu	Finance	90000	128	Zhang	Comp.Sci.	102
Elec.Eng.	Taylor	85000	15151	Mozart	Music	40000	19991	Brandt	HISTORY	80
Finance	Painter	120000	22222	Einstein	Physics	95000	23121	Chavez	Finance	110
History	Painter	50000	32343	El Said	History	60000	44553	Peltier	Physics	56
Music	Packard	80000	33456	Gold	Physics	87000	45678	Levy	Physics	46
Physics	Watson	70000	45565	Katz	Comp.Sci.	75000	54321	Williams	Comp.Sci.	54
			58583	Clifieri	History	62000	55739	Sanchez	Music	38
			76543	Singh	Finance	80000	70557	Snow	Physics	0
			76766	Crick	Biology	72000	76543	Brown	Comp.Sci.	58
			83821	Brandt	Comp.Sci.	92000	76653	Aoi	Elec.Eng.	60
			98345	Kim	Elec.Eng.	80000	98765	Bourikas	Elec.Eng.	98

(2) SELECT, FROM, WHERE 실습

```
SELECT * FROM department, instructor;
SELECT name FROM student;
```

```
SELECT * FROM instructor
WHERE salary > 60000;
```

(2) - 1 SELECT

name
Shankar
Zhang
Brandt
Chavez
Peltier
Levy
Williams
Sanchez
Snow
Brown
Aoi
Bourikas

Student 테이블의 이름만 가져온다.

(2) - 2 FROM

dept_name	building	budget	id	name	dept_name	salary
Physics	Watson	70000	10101	Srinivasan	Comp.Sci.	65000
Music	Packard	80000	10101	Srinivasan	Comp.Sci.	65000
History	Painter	50000	10101	Srinivasan	Comp.Sci.	65000
Finance	Painter	120000	10101	Srinivasan	Comp.Sci.	65000
Elec.Eng.	Taylor	85000	10101	Srinivasan	Comp.Sci.	65000
Comp.Sci	Taylor	100000	10101	Srinivasan	Comp.Sci.	65000
Biology	Watson	90000	10101	Srinivasan	Comp.Sci.	65000
Physics	Watson	70000	12121	Wu	Finance	90000
Music	Packard	80000	12121	Wu	Finance	90000
History	Painter	50000	12121	Wu	Finance	90000
Finance	Painter	120000	12121	Wu	Finance	90000
Elec.Eng.	Taylor	85000	12121	Wu	Finance	90000
Comp.Sci	Taylor	100000	12121	Wu	Finance	90000
Biology	Watson	90000	12121	Wu	Finance	90000
Physics	Watson	70000	15151	Mozart	Music	40000
Music	Packard	80000	15151	Mozart	Music	40000
History	Painter	50000	15151	Mozart	Music	40000
Finance	Painter	120000	15151	Mozart	Music	40000
Elec.Eng.	Taylor	85000	15151	Mozart	Music	40000
Comp.Sci	Taylor	100000	15151	Mozart	Music	40000
Biology	Watson	90000	15151	Mozart	Music	40000
Physics	Watson	70000	22222	Einstein	Physics	95000

두 테이블의 모든 값을 합쳐서 보여준다. (카티션 곱과 같다.)

(2) - 3 WHERE

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

instructor의 테이블에서 salary의 값이 60000이 넘는 것을 보여줍니다.

### (3) 집합 연산

#### (3) – 1 UNION

두 결과의 합을 보여준다.

name	Brandt
	Kim
Srinivasan	Shankar
Wu	Zhang
Mozart	Chavez
Einstein	Peltier
El Said	Levy
Gold	Williams
Katz	Sanchez
Clifieri	Snow
Singh	Brown
Crick	Aoi
	Bourikas

#### (3) – 2 INTERSECT

```
SELECT i.name FROM instructor AS i
INNER JOIN
student AS s ON i.`name` = s.`name`;
```

Student와 instructor테이블의 이름이 같은 멤버 하나만 출력

name
Brandt

#### (3) – 3 EXCEPT

```
SELECT i.name
FROM instructor AS i LEFT JOIN student AS s ON i.`name` = s.`name`
WHERE s.`name` IS NULL;
```

Instructor와 student 테이블의 name 필드 중 겹치는 name값을 instructor의 name 필드에서 제외하고 출력

name
Sriniv...
Wu
Mozart
Einstein
El Said
Gold
Katz
Clifieri
Singh
Crick
Brandt
Kim

### (4) JOIN 연산

#### (4) – 1 INNER JOIN

```
SELECT *
FROM instructor AS i INNER JOIN department AS d
ON i.dept_name = d.dept_name;
```

Instructor와 department의 dept\_name이 같은 데이터들만 출력

#### (4) – 2 OUTER JOIN

```
SELECT *
FROM instructor AS i LEFT JOIN department AS d
ON i.dept_name = d.dept_name;
```

instructor를 기준으로 dept\_name이 같은 데이터를 출력

Department테이블에 값이 없을 경우 null로 출력

id	name	dept_...	salary	dept_name	building	budget
10101	Srinivasan	Comp...	65000	NULL	NULL	NULL
12121	Wu	Finance	90000	Finance	Painter	120000
15151	Mozart	Music	40000	Music	Packard	80000
22222	Einstein	Physics	95000	Physics	Watson	70000
32343	El Said	History	60000	History	Painter	50000
33456	Gold	Physics	87000	Physics	Watson	70000
45565	Katz	Comp...	75000	NULL	NULL	NULL
58583	Clifieri	History	62000	History	Painter	50000
76543	Singh	Finance	80000	Finance	Painter	120000
76766	Crick	Biology	72000	Biology	Watson	90000
83821	Brandt	Comp...	92000	NULL	NULL	NULL
98345	Kim	Elec....	80000	Elec.Eng.	Taylor	85000

1.2 3 주차 실습 PPT 에 나오는 SQL Query 를 직접 입력하여 실행해 보고 해당 쿼리에 대한 설명과 결과를 Word 나 한글 파일로 정리해 제출하세요.

#### (1) – 1 AVG, MIN, MAX, SUM, COUNT

<pre>SELECT AVG(salary) FROM instructor</pre>	<pre>SELECT MIN(salary) FROM instructor</pre>	<pre>SELECT MAX(salary) FROM instructor</pre>	<pre>SELECT SUM(salary) FROM instructor</pre>	<pre>SELECT COUNT(salary) FROM instructor</pre>
AVG(salary)	MIN(salary)	MAX(salary)	SUM(salary)	COUNT(salary)
74833.3333	40000	95000	898000	12

#### (1) – 2 GROUP BY, HAVING

<pre>SELECT dept_name FROM instructor GROUP BY dept_name HAVING dept_name LIKE "_i%";</pre>	dept_name
	Finance
	History
	Biology

SELECT dept\_name FROM instructor의 dept\_name들을 가져온 뒤, GROUP BY를 통해 dept\_name으로 그룹화를 합니다. 이후 HAVING dept\_name LIKE "\_i%"를 통해 dept\_name으로 그룹화 된 값에 두번째 문자가 i인 경우만을 출력합니다.

#### (2) – 1 중첩 부질의 IN, NOT IN

```
SELECT name, salary, dept_name
FROM instructor
WHERE dept_name NOT IN
(
    SELECT dept_name
    FROM student
    WHERE dept_name LIKE "F%"
);
```

name	salary	dept_name
Srinivasan	65000	Comp.Sci.
Mozart	40000	Music
Einstein	95000	Physics
El Said	60000	History
Gold	87000	Physics
Katz	75000	Comp.Sci.
Cliffier	62000	History
Crick	72000	Biology
Brandt	92000	Comp.Sci.
Kim	80000	Elec.Eng.

```
SELECT name, salary, dept_name
FROM instructor
WHERE dept_name IN
(
    SELECT dept_name
    FROM student
    WHERE dept_name LIKE "F%"
);
```

name	salary	dept_name
Wu	90000	Finance
Singh	80000	Finance

NOT IN 의 경우 NOT IN 안의 코드로 인해 dept\_name이 Finance인 데이터가 검색됩니다. 이후 NOT IN을 통해 dept\_name값이 Finance이 아닌 값들을 출력해줍니다. 반대로 IN의 경우 dept\_name이 Finance인 데이터들만 찾아줍니다.

#### (2) – 2 SOME, ALL

```
SELECT name, salary
FROM instructor
WHERE salary < ALL
(
    SELECT salary
    FROM instructor
    WHERE name LIKE "%i%"
);
```

name	salary
Mozart	40000

```
SELECT name, salary
FROM instructor
WHERE salary < SOME
(
    SELECT salary
    FROM instructor
    WHERE name LIKE "%i%"
);
```

name	salary
Srinivasan	65000
Wu	90000
Mozart	40000
El Said	60000
Gold	87000
Katz	75000
Cliffier	62000
Singh	80000
Crick	72000
Brandt	92000
Kim	80000

ALL의 경우 salary의 값이 이름 사이에 i가 들어가는 데이터의 값들의 모든 salary보다 작은 값을 가진 데이터를 출력합니다. SOME의 경우 salary의 값이 모든 salary중 하나보다 작기만 하면 그 값들을 출력합니다.

## (2) - 3 EXIST, NOT EXIST

```
SELECT *
FROM instructor
WHERE EXISTS
(
    SELECT dept_name
    FROM student
    WHERE tot_cred > 20
);
```

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

쿼리문의 결과가 빈테이블인지 아닌지 검사하는 키워드

```
SELECT *
FROM instructor
WHERE NOT EXISTS
(
    SELECT dept_name
    FROM student
    WHERE tot_cred > 20
);
```

## (2) - 4 FROM절 부질의

```
SELECT *
FROM
(
    SELECT *
    FROM student
) as tmp
WHERE tmp.dept_name LIKE "__i%";
```

id	name	dept_name	tot_cred
19991	Brandt	HISTORY	80
23121	Chavez	Finance	110

WHERE절의 중첩 부질의 이외에도 FROM절에 사용하여 쿼리문을 실행 할 수 있습니다.

## 과제 2.

다음 슬라이드의 테이블을 참고하여 아래의 데이터를 검색하는 SQL 쿼리를 작성하고 결과를 파일로 정리해 제출하세요.

1. section 테이블을 이용하여 2009 년도에 열렸던 모든 과목의 코드 (c\_id)를 중복없이 검색하세요.

```
SELECT DISTINCT c_id
FROM section
WHERE years = "2009";
```

c_id
BIO-101
CS-101
CS-190
CS-347
EE-181
PHY-101

distinct키워드를 이용하여 중복을 제거하고, years가 2009인 레코드만 출력

2. student와 takes 테이블을 이용하여 'Levy'이라는 학생이 들었던 과목의 코드(c\_id)와 성적(grade)을 모두 검색하세요. (join 사용)

```
SELECT c_id, grade
FROM takes AS t INNER JOIN student AS s
ON s.id = t.s_id
WHERE name = "Levy";
```

c_id	grade
CS-101	F
CS-101	B+
CS-319	B

3. section 과 teaches 테이블을 이용하여 'Taylor' 건물에서 열린 수업을 진행한 강사의 코드(i\_id)를 모두 검색하세요. (중첩 부질의 사용)

```
SELECT i_id
FROM teaches
WHERE c_id IN
(
SELECT c_id
FROM section
WHERE building = "Taylor"
);
```

i_id
10101
45565
83821
83821
83821
98345

4. takes 와 course 테이블을 이용하여 각 학생들이 들었던 모든 과목에 대해서 학생(s\_id) 별로 취득한 총 학점수(누적된 학점수)를 구하시오.

```
SELECT s_id, sum(credits) AS "모든학점"
FROM takes INNER JOIN course
ON c_id = id
GROUP BY s_id
```

s_id	모든학점
00128	7
12345	14
19991	3
23121	3
44553	4
45678	11
54321	8