

데이터베이스시스템

14. 고급 질의어

나 홍 석 교수



14
LESSON

고급 질의어

학습 목표

- 1 연산자를 이용하여 조건식을 구성할 수 있다.
- 2 서브쿼리를 사용할 수 있다.
- 3 조인을 이용하여 원하는 결과를 추출할 수 있다.

학습 내용

1 연산자와 서브쿼리(Sub Query)

2 조인(Join)

3 질의문 연습

Chapter 01 연산자와 서브쿼리(Sub Query)

1 회사 데이터베이스 상태

1 사원, 부양가족 테이블

SELECT * FROM employee;

	ename	ssn	address	salary	sex	bdate	dno	superssn
▶	김창현	123456789	서울 광진구 구의동	4000	M	1975-01-09 00:00:00	5	666884444
	강명석	333445555	서울 송파구 풍납동	5000	M	1965-12-08 00:00:00	5	NULL
	박현식	444889999	서울 강동구 암사동	4000	M	1975-10-09 00:00:00	1	NULL
	이혜진	453453453	경기 용인시 동천동	4000	F	1975-09-12 00:00:00	5	666884444
	한명숙	666884444	경기 용인시 풍덕천동	4300	F	1972-07-05 00:00:00	5	333445555
	이지은	888665555	서울 강동구 고덕동	3800	F			
	전명희	987654321	서울 종로구 안국동	3500	F			
	김유현	987987987	서울 강남구 압구정동	4800	M			
	남석훈	999887777	경기 구리시 인창동	4000	M			
*	NULL	NULL	NULL	NULL	NULL			

SELECT * FROM dependent;

	essn	dpntname	sex	bdate	relation
▶	333445555	강동진	M	1995-04-05 00:00:00	아들
	333445555	강수진	F	1998-05-05 00:00:00	딸
	333445555	김명희	F	1967-10-13 00:00:00	배우자
	444889999	한명숙	F	1975-04-06 00:00:00	배우자
	987987987	김다래	F	2000-02-21 00:00:00	딸
	987987987	남희정	F	1970-07-09 00:00:00	배우자
*	NULL	NULL	NULL	NULL	NULL

1 회사 데이터베이스 상태

2 부서, 부서위치 테이블

SELECT * FROM department;

	dnumber	dname	mgrssn	mgrstartdate
▶	1	영업부	333445555	2015-04-18 00:00:00
	4	총무부	987987987	2017-03-21 00:00:00
	5	연구소	444889999	2018-09-11 00:00:00
*	NULL	NULL	NULL	NULL

SELECT * FROM dept_locations;

	dnumber	dlocations
▶	1	대구
	1	부산
	1	서울
	4	서울
	4	인천
	5	대전
*	NULL	NULL

1 회사 데이터베이스 상태

3 프로젝트, 참여 테이블

SELECT * FROM project;

	pnumber	pname	plocation	dnum
▶	10	차세대 정보 시스템 도입	서울	5
	20	효과적 조직 개편안 연구	서울	4
	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	40	보안 강화를 위한 인프라 구축	대전	5
•	NULL	NULL	NULL	NULL

SELECT * FROM works_on;

	essn	pno	hours
▶	123456789	10	25.0
	333445555	30	20.0
	444889999	10	20.0
	444889999	40	10.5
	453453453	10	20.0
	666884444	40	20.0
	888665555	30	25.0
	987654321	10	5.0
	987654321	40	5.0
	987987987	10	5.5
	987987987	20	20.0
	987987987	40	5.5
	999887777	10	5.0
	999887777	20	10.5
•	NULL	NULL	NULL

2 연산자

1 연산자의 역할

- ☑ SELECT 문의 **WHERE 절**은 조회하려는 데이터에 특정 조건을 부여할 목적으로 사용
- ☑ **비교연산자, SQL연산자, 논리연산자**를 사용하여 조건식을 구성

```
SELECT [DISTINCT] <컬럼리스트>  
FROM <테이블 리스트>  
[WHERE <조건>]  
[GROUP BY <컬럼명> [HAVING <그룹 조건>]]  
[ORDER BY <컬럼명> [ASC | DESC]]
```

2 연산자

2 연산자의 종류

구분	연산자	연산자의 의미
비교 연산자	=	같다.
	>, >=	크다, 크거나 같다.
	<, <=	작다, 작거나 같다.
	!=	같지 않다.
SQL 연산자	BETWEEN a AND b	a와 b의 값 사이의 값을 갖는다.(a와 b값이 포함된다)
	IN(list)	리스트(집합)에 있는 값 중에서 어느 하나라도 일치한다.
	LIKE '비교문자열'	비교문자열과 형태가 일치한다. (% , _ 사용)
	IS NULL	NULL 값인지를 비교한다.
논리 연산자	a AND b	조건식 a와 b를 모두 만족하는 경우 (참인경우) 참이된다.
	a OR b	조건식 a와 b중 어느 하나라도 만족하면 (참인경우) 참이된다.
	NOT a	조건식 a의 반대되는 결과(참 또는 거짓)을 반환한다.

2 연산자

3 단순 비교연산자

질의문 연봉이 4000이하인 사원들의 이름과 연봉, 부서번호를 검색하세요.

SQL **SELECT** ename, salary, dno **FROM** employee
WHERE salary <= 4000;

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

결과

	ename	salary	dno
▶	이지은	3800	1
	전명희	3850	4

2 연산자

4 BETWEEN a AND b

질의문 연봉이 4000이상 5000이하인 사원들의 이름과 연봉, 부서번호를 검색하세요.

SQL **SELECT** ename, salary, dno **FROM** employee
WHERE salary >= 4000 AND salary <=5000;

SELECT ename, salary, dno **FROM** employee
WHERE salary **BETWEEN** 4000 **AND** 5000;

결과

	ename	salary	dno
▶	김창현	4400	5
	강명석	5000	5
	박현식	4400	1
	이혜진	4400	5
	한명숙	4300	5
	남석훈	4400	4

2 연산자

5 LIKE

질의문

성이 '이' 씨인 사원을 검색하세요.

단어 '마케팅'이 포함된 프로젝트를 검색하세요.

SQL

```
SELECT * FROM employee WHERE ename LIKE '이%';
```

```
SELECT * FROM project WHERE pname LIKE '%마케팅%';
```

결과

	ename	ssn	address	salary	sex	bdate	dno	superssn
▶	이혜진	453453453	경기 용인시 동천동	4400	F	1975-09-12 00:00:00	5	666884444
	이지은	888665555	서울 강동구 고덕동	3800	F	1977-12-11 00:00:00	1	444889999
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	pnumber	pname	plocation	dnum
▶	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
*	NULL	NULL	NULL	NULL

2

6

질의문

SQL

결과

[illegible]

2 연산자

7 IN #1

질의문 1번 또는 5번 부서에 속한 사원을 검색하세요.

SQL

```
SELECT * FROM employee WHERE dno =1 or dno = 5;  
SELECT * FROM employee WHERE dno IN (1,5);  
SELECT * FROM employee WHERE dno NOT IN (1,5);
```

결과

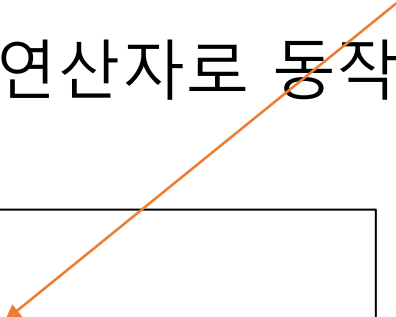
[illegible]

3 서브쿼리(Sub Query)

1 개요

- ✓ SQL 문 안에 포함된 또 다른 SQL 문
- ✓ 포함된 SQL문의 결과(릴레이션)이 SQL문의 피연산자로 동작

```
SELECT ename, salary FROM employee  
WHERE salary > (SELECT avg(salary) FROM employee);
```



Sub Query

	ename	salary
▶	강명석	5000
	김유현	5280

3 서브쿼리(Sub Query)

2 단일값을 반환하는 서브쿼리

질의문 박현식 사원과 같은 부서에서 근무하는 사원의 이름, 주소, 부서번호를 검색하세요.

SQL **SELECT** ename, address, dno **FROM** employee
WHERE dno = (**SELECT** dno **FROM** employee **WHERE** ename = '박현식');

결과

	ename	address	dno
▶	박현식	서울 강동구 암사동	1
	이지은	서울 강동구 고덕동	1

3 서브쿼리(Sub Query)

3 다중 값을 반환하는 서브쿼리

질의문 20번 프로젝트에 참여하는 사원의 이름과 주소를 검색하세요.

SQL **SELECT** ename, address **FROM** employee
WHERE ssn **IN** (SELECT essn FROM works_on WHERE pno = 20);

결과

	ename	address
▶	김유현	서울 강남구 압구정동
	남석훈	경기 구리시 인창동

3 서브쿼리(Sub Query)

4 NOT IN

질의문

20번 프로젝트에 **참여하지 않는** 사원의 이름과 주소를 검색하세요.

SQL

```
SELECT ename, address FROM employee  
WHERE ssn NOT IN (SELECT essn FROM works_on WHERE pno = 20);
```

결과

	ename	address
▶	김창헌	서울 광진구 구의동
	강명석	서울 송파구 풍납동
	박현식	서울 강동구 암사동
	이혜진	경기 용인시 동천동
	한명숙	경기 용인시 풍덕천동
	이지은	서울 강동구 고덕동
	전명희	서울 종로구 안국동

3 서브쿼리(Sub Query)

5 SELECT 구문에 들어가는 서브쿼리

질의문 사원의 이름과 부양가족 수를 함께 보여주세요.

SQL **SELECT** **ename**, **address**, (**SELECT** **count(essn)** **FROM** **dependent** **WHERE** **essn** = **ssn**)
FROM **employee**;

dependent	essn	dpntname	sex	bdate	relation
-----------	------	----------	-----	-------	----------

결과

	ename	address	(SELECT count(essn) FROM dependent WHERE essn = ssn)
▶	김창현	서울 광진구 구의동	0
	강명석	서울 송파구 풍납동	3
	박현식	서울 강동구 암사동	1
	이혜진	경기 용인시 동천동	0
	한명숙	경기 용인시 풍덕천동	0
	이지은	서울 강동구 고덕동	0

Chapter 02 조인(Join)

1 개요

1

조인 정의

- ☑ 두 개 이상의 테이블을 서로 묶어서 하나의 결과집합으로 만들어 내는 것
- ☑ 정규화(normalization) 과정을 통해 분할된 테이블들로부터 통합적인 정보를 얻기 위해서 조인 과정이 필요함

**카티션
프로덕트**
(크로스조인)

내부조인
(동등조인,
자연조인)

외부조인

자체조인

2 카디션 프로젝트

1 정의

- ☑ 두 테이블의 모든 로우를 조인하여 새로운 테이블 생성
- ☑ 주로 테스트 용도로 대용량의 테이블 생성 시 사용

1. **SELECT * FROM department CROSS JOIN project;**
2. **SELECT * FROM department, project ;**




	dnumber	dname	mgrssn	mgrstartdate
▶	1	영업부	333445555	2015-04-18 00:00:00
	4	총무부	987987987	2017-03-21 00:00:00
	5	연구소	444889999	2018-09-11 00:00:00
*	NULL	NULL	NULL	NULL

	pnumber	pname	plocation	dnum
▶	10	차세대 정보 시스템 도입	서울	5
	20	효과적 조직 개편안 연구	서울	4
	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	40	보안 강화를 위한 인프라 구축	대전	5
*	NULL	NULL	NULL	NULL

2 카디션 프로젝트

2 예제

☑ `SELECT * FROM department CROSS JOIN project;`

Result Grid  Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 								
	dnumber	dname	mgrssn	mgrstartdate	pnumber	pname	plocation	dnum
▶	5	연구소	444889999	2018-09-11 00:00:00	10	차세대 정보 시스템 도입	서울	5
	4	총무부	987987987	2017-03-21 00:00:00	10	차세대 정보 시스템 도입	서울	5
	1	영업부	333445555	2015-04-18 00:00:00	10	차세대 정보 시스템 도입	서울	5
	5	연구소	444889999	2018-09-11 00:00:00	20	효과적 조직 개편안 연구	서울	4
	4	총무부	987987987	2017-03-21 00:00:00	20	효과적 조직 개편안 연구	서울	4
	1	영업부	333445555	2015-04-18 00:00:00	20	효과적 조직 개편안 연구	서울	4
	5	연구소	444889999	2018-09-11 00:00:00	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	4	총무부	987987987	2017-03-21 00:00:00	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	1	영업부	333445555	2015-04-18 00:00:00	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	5	연구소	444889999	2018-09-11 00:00:00	40	보안 강화를 위한 인프라 구축	대전	5
	4	총무부	987987987	2017-03-21 00:00:00	40	보안 강화를 위한 인프라 구축	대전	5
	1	영업부	333445555	2015-04-18 00:00:00	40	보안 강화를 위한 인프라 구축	대전	5

3 내부조인(Inner Join)

1 정의

- ☑ 두 테이블을 대상으로 조인 조건에 맞는 row들을 결합하여 새로운 테이블 생성
- ☑ 일반적인 조인은 내부조인을 말함

세타조인

동등조인

자연조인

3 내부조인(Inner Join)

2 형식

✓ **SELECT** <열 목록> **FROM** <테이블1> **[INNER] JOIN** <테이블2> **ON** <조인 조건>

✓ **SELECT** <열 목록> **FROM** <테이블1>, <테이블2> **WHERE** <조인 조건>

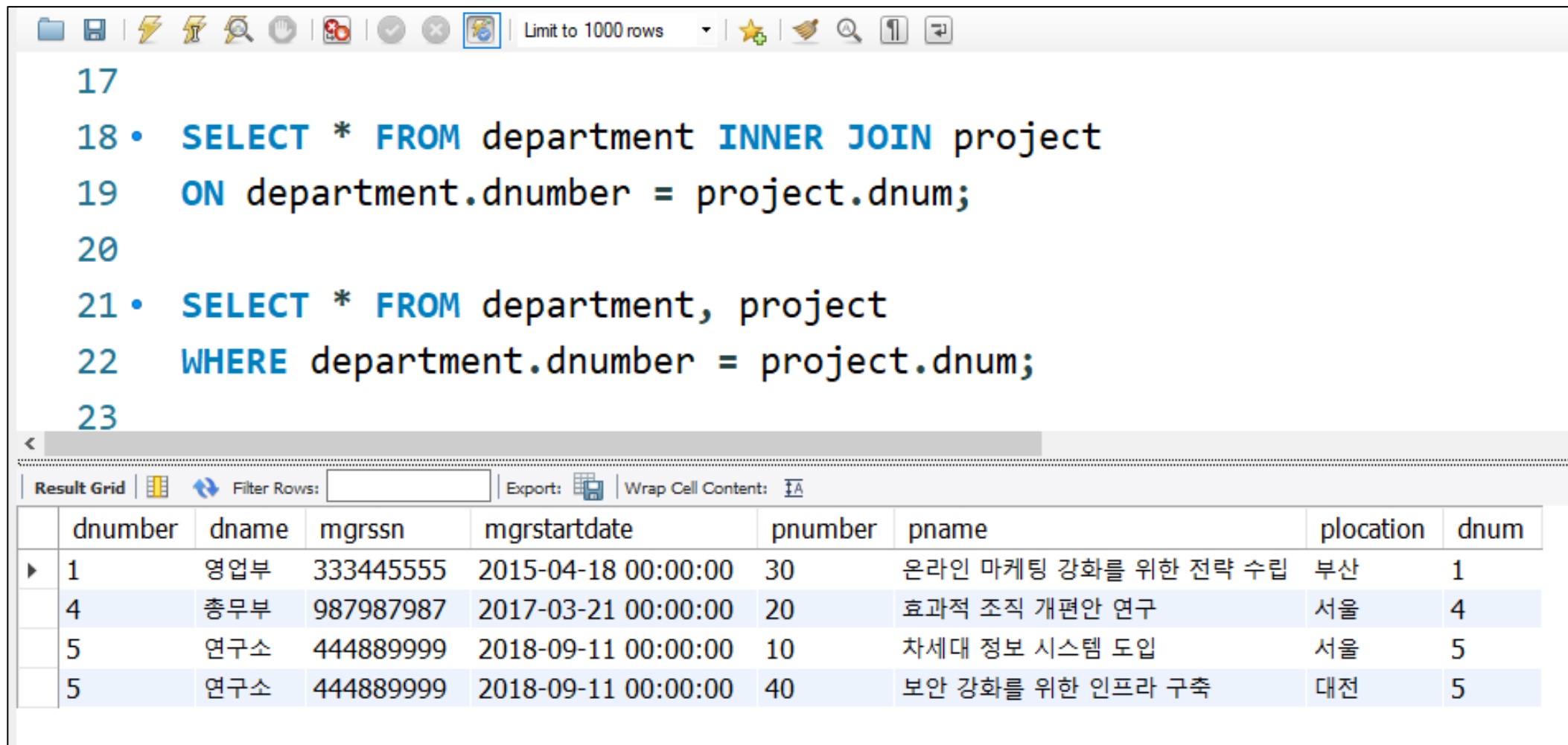
SELECT * FROM department INNER JOIN project
ON department.dnumber = project.dnum

	dnumber	dname	mgrssn	mgrstartdate
▶	1	영업부	333445555	2015-04-18 00:00:00
	4	총무부	987987987	2017-03-21 00:00:00
	5	연구소	444889999	2018-09-11 00:00:00
*	NULL	NULL	NULL	NULL

	pnumber	pname	plocation	dnum
▶	10	차세대 정보 시스템 도입	서울	5
	20	효과적 조직 개편안 연구	서울	4
	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	40	보안 강화를 위한 인프라 구축	대전	5
*	NULL	NULL	NULL	NULL

3 내부조인(Inner Join)

3 실행결과 #1



```
17
18 • SELECT * FROM department INNER JOIN project
19   ON department.dnumber = project.dnum;
20
21 • SELECT * FROM department, project
22   WHERE department.dnumber = project.dnum;
23
```

	dnumber	dname	mgrssn	mgrstartdate	pnumber	pname	plocation	dnum
▶	1	영업부	333445555	2015-04-18 00:00:00	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	4	총무부	987987987	2017-03-21 00:00:00	20	효과적 조직 개편안 연구	서울	4
	5	연구소	444889999	2018-09-11 00:00:00	10	차세대 정보 시스템 도입	서울	5
	5	연구소	444889999	2018-09-11 00:00:00	40	보안 강화를 위한 인프라 구축	대전	5

3 내부조인(Inner Join)

3 실행결과 #2

☑ 조인 결과에 조건을 줄 수 있음(WHERE 절)

```
24 • SELECT * FROM department INNER JOIN project
25   ON department.dnumber = project.dnum
26   WHERE plocation IN ('서울', '부산');
```

Result Grid Filter Rows: Export: Wrap Cell Content:								
	dnumber	dname	mgrssn	mgrstartdate	pnumber	pname	plocation	dnum
▶	5	연구소	444889999	2018-09-11 00:00:00	10	차세대 정보 시스템 도입	서울	5
	4	총무부	987987987	2017-03-21 00:00:00	20	효과적 조직 개편안 연구	서울	4
	1	영업부	333445555	2015-04-18 00:00:00	30	온라인 마케팅 강화를 위한 전략 수립	부산	1

4 외부조인(Outer Join)

1 정의

- ☑ 조인 조건을 만족하지 않는 행까지 결과 집합에 포함하여 돌려주는 특수한 조인
- ☑ 집계 등에 종종 사용됨
- ☑ 왼쪽(Left) 외부 조인, 오른쪽(Right) 외부 조인, 완전(Full) 외부 조인

(질의)

사원별 부양가족의 이름을 사원이름과 함께 보여주세요.

4 외부조인(Outer Join)

2

예제1 - 내부조인

질의

사원별 부양가족의 이름을 사원이름과 함께 보여주세요.

```
SELECT ssn, ename  
FROM employee;
```

	ssn	ename
▶	123456789	김창헌
	333445555	강명석
	444889999	박현식
	453453453	이혜진
	666884444	한명숙
	888665555	이지은
	987654321	전명희
	987987987	김유현
	999887777	남석훈
•	NULL	NULL

```
SELECT essn, dpntname  
FROM dependent;
```

	essn	dpntname
▶	333445555	강동진
	333445555	강수진
	333445555	김명희
	444889999	한명숙
	987987987	김다래
	987987987	남희정
•	NULL	NULL

4 외부조인(Outer Join)

2

예제1 - 내부조인

질의

사원별 부양가족의 이름을 사원이름과 함께 보여주세요.

```
SELECT ssn, ename  
FROM employee;
```

	ssn	ename
▶	123456789	김창현
	333445555	강명석
	444889999	박현식
	453453453	이혜진
	666884444	한명숙
	888665555	이지은
	987654321	전명희
	987987987	김유현
	999887777	남석훈
•	NULL	NULL

```
SELECT essn, dpntname  
FROM dependent;
```

	essn	dpntname
▶	333445555	강동진
	333445555	강수진
	333445555	김명희
	444889999	한명숙
	987987987	김다래
	987987987	남희정
•	NULL	NULL

```
SELECT e.ssn, e.ename, d.essn, d.dpntname  
FROM employee e JOIN dependent d  
ON e.ssn = d.essn;
```

	ssn	ename	essn	dpntname
▶	333445555	강명석	333445555	강동진
	333445555	강명석	333445555	강수진
	333445555	강명석	333445555	김명희
	444889999	박현식	444889999	한명숙
	987987987	김유현	987987987	김다래
	987987987	김유현	987987987	남희정

4 외부조인(Outer Join)

2 예제1 - 왼쪽 외부조인

질의

사원별 부양가족의 이름을 사원이름과 함께 보여주세요.
- 부양가족이 없는 사원도 함께 보여주세요.

```
SELECT e.ssn, e.ename, d.essn, d.dpntname  
FROM employee e LEFT OUTER JOIN dependent d  
ON e.ssn = d.essn;
```

Result Grid	Filter Rows:	Export:	Wrap C
ssn	ename	essn	dpntname
123456789	김창현	NULL	NULL
333445555	강명석	333445555	강동진
333445555	강명석	333445555	강수진
333445555	강명석	333445555	김명희
444889999	박현식	444889999	한명숙
453453453	이혜진	NULL	NULL
666884444	한명숙	NULL	NULL
888665555	이지은	NULL	NULL
987654321	전명희	NULL	NULL
987987987	김유현	987987987	김다래
987987987	김유현	987987987	남희정
999887777	남석훈	NULL	NULL

4 외부조인(Outer Join)

3 예제2 - 조인 및 집계함수 사용 #1

질의 사원별 부양가족 수를 보여주세요.

```
SELECT essn, count(essn) FROM dependent GROUP BY essn;
```

Result Grid			Filter Rows:
	essn	count(essn)	
▶	333445555	3	
	444889999	1	
	987987987	2	

4 외부조인(Outer Join)

3 예제2 - 조인 및 집계함수 사용 #2

질의 사원명과 부양가족 수를 구하세요. (사원명이 나옵니다.)

```
SELECT e.ssn, e.ename, count(d.essn)
FROM employee e JOIN dependent d ON e.ssn = d.essn
GROUP BY e.ssn;
```

Result Grid	Filter Rows:	Export:
ssn	ename	count(d.essn)
▶ 333445555	강명석	3
444889999	박현식	1
987987987	김유현	2

4 외부조인(Outer Join)

4

예제3

질의

사원명과 부양가족 이름을 구하세요. (사원명이 나옵니다.)
- 부양가족이 없는 사원도 보여주세요.

```
SELECT e.ssn, e.ename, count(d.essn)
FROM employee e LEFT OUTER JOIN dependent d ON e.ssn = d.essn
GROUP BY e.ssn;
```

Result Grid	Filter Rows:	Exp
ssn	ename	count(d.essn)
123456789	김창현	0
333445555	강명석	3
444889999	박현식	1
453453453	이혜진	0
666884444	한명숙	0
888665555	이지은	0
987654321	전명희	0
987987987	김유현	2
999887777	남석훈	0

5 유니온(UNION)

1 정의

- ☑ 둘 이상의 집합을 합성해서 하나의 결과 집합으로 만들어주는 연산자
- ☑ 비슷한 결과 집합을 가지지만 하나의 SELECT 문으로 만들기 힘들 경우에 유니온 연산자를 사용
- ☑ 복잡한 단일 SELECT 문보다 유니온 연산자를 이용하여 여러 개의 단순한 SELECT 문으로 분리하면 성능 개선도 가능

5 유니온(UNION)

2 예제1

- ☒ 연봉 4천이상인 직원들의 평균 연봉과,
연봉 4천미만인 직원들의 평균 연봉을 구해서 보여주세요.

```
SELECT avg(salary)
FROM employee
WHERE salary >= 4000;
```

Result Grid		Filter Rows:
	avg(salary)	
▶	4300.0000	

```
SELECT avg(salary)
FROM employee
WHERE salary < 4000;
```

Result Grid		Filter Rows:
	avg(salary)	
▶	3650.0000	

5 유니온(UNION)

3 예제2

✓ 결과 합쳐서 보여주기

```
SELECT avg(salary)
FROM employee
WHERE salary >= 4000
UNION
SELECT avg(salary)
FROM employee
WHERE salary < 4000;
```

Result Grid	
	avg(salary)
▶	4300.0000
	3650.0000

✓ 앞에 설명 넣기

```
SELECT '연봉 4천이상', avg(salary)
FROM employee
WHERE salary >= 4000
UNION
SELECT '연봉 4천미만', avg(salary)
FROM employee
WHERE salary < 4000;
```

Result Grid		
	Filter Rows:	
	연봉 4천 이상	avg(salary)
▶	연봉 4천이상	4300.0000
	연봉 4천미만	3650.0000

Chapter 03 질의문 연습

1 회사 데이터베이스 검색(1)

1

단일 테이블 검색

질의문 이름이 '김유현'인 사원(들)의 생년월일(BDATE)과 주소(ADDRESS)를 검색하라.

SQL `SELECT bdate, address FROM employee WHERE ename = '김유현';`

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

결과

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
bdate	address			
▶ 1968-12-12 00:00:00	서울 강남구 압구정동			

1 회사 데이터베이스 검색(1)

2 조인 검색

질의문 모든 사원의 사원번호(SSN)와 소속된 부서명(DNAME)을 출력하시오.

SQL `SELECT ssn, dname
FROM employee JOIN department ON employee.dno = department.dnumber;`

employee
department

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

dnumber	dname	mgrssn	mgrstartdate
---------	-------	--------	--------------

결과

ssn	dname
123456789	연구소
333445555	연구소
453453453	연구소
666884444	연구소
444889999	영업부
888665555	영업부
987654321	총무부
987987987	총무부
999887777	총무부

1 회사 데이터베이스 검색(1)

3 조인 검색 - 조건 추가

질의문 '연구소' 에 근무하는 모든 사원의 이름(NAME)과 주소(ADDRESS)를 검색하라.

SQL **SELECT** ename, address **FROM** employee **JOIN** department
ON employee.dno = department.dnumber
WHERE department.dname = '연구소';

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

department

dnumber	dname	mgrssn	mgrstartdate
---------	-------	--------	--------------

결과

Result Grid		Filter Rows:
▶	ename	address
	김창현	서울 광진구 구의동
	강명석	서울 송파구 풍납동
	이혜진	경기 용인시 동천동
	한명숙	경기 용인시 풍덕천동

1 회사 데이터베이스 검색(1)

3 조인 검색 - 별명 사용

질의문 '연구소'에 근무하는 모든 사원의 이름(NAME)과 주소(ADDRESS)를 검색하라.

SQL **SELECT** ename, address **FROM** employee **JOIN** department
ON employee.dno = department.dnumber
WHERE department.dname = '연구소';



SELECT e.ename, e.address **FROM** employee e **JOIN** department d
ON e.dno = d.dnumber
WHERE d.dname = '연구소';

1 회사 데이터베이스 검색(1)

4 같은 테이블 조인

검색
질의문

각 사원에 대해 사원의 이름(NAME)과 연봉(SALARY),
직속 상사의 이름(NAME)과 연봉(SALARY)을 검색하라.

SQL

```
SELECT e.ename, e.salary, s.ename, s.salary  
FROM employee e JOIN employee s ON e.superssn = s.ssn;
```

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

결과

	ename	salary	ename	salary
▶	김창현	4000	한명숙	4300
	이혜진	4000	한명숙	4300
	한명숙	4300	강명석	5000
	이지은	3800	박현식	4000
	전명희	3500	김유현	4800
	남석훈	4000	김유현	4800

```
SELECT e.ename, e.salary, s.ename, s.salary  
FROM employee e LEFT OUTER JOIN employee s  
ON e.superssn = s.ssn;
```

	ename	salary	ename	salary
▶	김창현	4000	한명숙	4300
	강명석	5000	NULL	NULL
	박현식	4000	NULL	NULL
	이혜진	4000	한명숙	4300
	한명숙	4300	강명석	5000
	이지은	3800	박현식	4000
	전명희	3500	김유현	4800
	김유현	4800	NULL	NULL
	남석훈	4000	김유현	4800

1 회사 데이터베이스 검색(1)

5

중복값

제거

질의문

모든 사원의 급여(SALARY)를 검색하되 중복되는 값이 없도록 검색하라.

SQL

SELECT salary **FROM** employee;

SELECT DISTINCT salary **FROM** employee;

결과

	salary
▶	4000
	5000
	4000
	4000
	4300
	3800
	3500
	4800
	4000

	salary
▶	4000
	5000
	4300
	3800
	3500
	4800

2 회사 데이터베이스 검색(2)

1 문자열 비교 검색

질의문 '용인시'에 사는 모든 사원을 검색하세요. (번호, 이름, 주소)

SQL **SELECT** ssn, ename, address **FROM** employee
WHERE address like '%용인%';

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

결과

	ssn	ename	address
▶	453453453	이혜진	경기 용인시 동천동
	666884444	한명숙	경기 용인시 풍덕천동
*	NULL	NULL	NULL

2 회사 데이터베이스 검색(2)

2 서브쿼리

질의문

10번 프로젝트에 참여하는 모든 사원의 급여를 10% 올리세요.

SQL

```
SELECT ename, salary FROM employee  
WHERE employee.ssn IN (SELECT essn FROM works_on WHERE pno = '10');
```

```
UPDATE employee SET salary = salary * 1.1  
WHERE employee.ssn IN (SELECT essn FROM works_on WHERE pno = '10');
```

결과

	ename	salary
▶	김창헌	4000
	박현식	4000
	이혜진	4000
	전명희	3500
	김유현	4800
	남석훈	4000

	ename	salary
▶	김창헌	4400
	박현식	4400
	이혜진	4400
	전명희	3850
	김유현	5280
	남석훈	4400

2 회사 데이터베이스 검색(2)

3 조건의 나열

질의문 연봉이 4000이상인 5번 부서의 모든 사원을 검색하라.

SQL **SELECT * FROM employee WHERE salary>=4000 AND dno=5 ;**

employee	ename	ssn	address	salary	sex	bdate	dno	superssn
----------	-------	-----	---------	--------	-----	-------	-----	----------

결과

[illegible]

2 회사 데이터베이스 검색(2)

4 2개 이상 테이블의 조인(결과 정렬)

질의문

사원의 이름과 각 사원이 참여하는 프로젝트들의 리스트를 검색하시오.
이때 사원이름의 오름차순으로 정렬하세요.

SQL

```
SELECT e.ename, q.pname FROM employee e
JOIN (SELECT * FROM project p JOIN works_on w ON p.pnumber = w.pno) q
ON e.ssn = q.essn
ORDER BY e.ename;
```

project

pnumber	pname	plocation	dn
---------	-------	-----------	----

employee

ename	ssn	address	salary	sex	bi
-------	-----	---------	--------	-----	----

works_on

essn	pno	hours
------	-----	-------

결과

ename	pname
강명석	온라인 마케팅 강화를 위한 전략 수립
김유현	보안 강화를 위한 인프라 구축
김유현	차세대 정보 시스템 도입
김유현	효과적 조직 개편안 연구
김창현	차세대 정보 시스템 도입
남석훈	차세대 정보 시스템 도입
남석훈	효과적 조직 개편안 연구
박현식	보안 강화를 위한 인프라 구축
박현식	차세대 정보 시스템 도입
이지은	온라인 마케팅 강화를 위한 전략 수립
이혜진	차세대 정보 시스템 도입
전명희	보안 강화를 위한 인프라 구축
전명희	차세대 정보 시스템 도입
한명숙	보안 강화를 위한 인프라 구축

2 회사 데이터베이스 검색(2)

5 2개 이상 테이블의 조인 - 조건추가

질의문

'서울'에 위치한 모든 프로젝트에 대해서 프로젝트 번호(PNUMBER), 담당 부서 번호(DNUM), 부서 관리자의 이름(NAME), 주소(ADDRESS), 생년월일(BDATE)을 검색하라.

SQL

```
SELECT p.pnumber, p.dnum, q.ename, q.address, q.bdate FROM project p
JOIN (SELECT * FROM employee e JOIN department d ON e.ssn = d.mgrssn) q
ON p.dnum = q.dnumber WHERE p.plocation = '서울';
```

project

pnumber	pname	plocation	dnum
---------	-------	-----------	------

employee

ename	ssn	address	salary	sex	bdate	dno	superssn
-------	-----	---------	--------	-----	-------	-----	----------

department

dnumber	dname	mgrssn	mgrstartdate
---------	-------	--------	--------------

결과

Result Grid Filter Rows: Export: Wrap Cell Content:					
	pnumber	dnum	ename	address	bdate
▶	10	5	박현식	서울 강동구 암사동	1975-10-09 00:00:00
	20	4	김유현	서울 강남구 압구정동	1968-12-12 00:00:00

학습 정리



연산자

구분	연산자	연산자의 의미
비교 연산자	=	같다.
	>, >=	크다, 크거나 같다.
	<, <=	작다, 작거나 같다.
	!=	같지 않다.
SQL 연산자	BETWEEN a AND b	a와 b의 값 사이의 값을 갖는다.(a와 b값이 포함된다)
	IN(list)	리스트(집합)에 있는 값 중에서 어느 하나라도 일치한다.
	LIKE '비교문자열'	비교문자열과 형태가 일치한다. (% , _ 사용)
	IS NULL	NULL 값인지를 비교한다.
논리 연산자	a AND b	조건식 a와 b를 모두 만족하는 경우 (참인경우) 참이된다.
	a OR b	조건식 a와 b중 어느 하나라도 만족하면 (참인경우) 참이된다.
	NOT a	조건식 a의 반대되는 결과(참 또는 거짓)을 반환한다.

학습 정리



서브쿼리

- SQL 문 안에 포함된 또 다른 SQL 문
- 포함된 SQL문의 결과(릴레이션)이 SQL문의 피연산자로 동작

```
SELECT ename, salary FROM employee  
WHERE salary > (SELECT avg(salary) FROM employee);
```

학습 정리



조인

- 두 개 이상의 테이블을 서로 묶어서 하나의 결과집합으로 만들어 내는 것
- 정규화(normalization) 과정을 통해 분할된 테이블들로부터 통합적인 정보를 얻기 위해서 조인 과정이 필요함

카티션
프로덕트
(크로스조인)

내부조인
(동등조인,
자연조인)

외부조인

자체조인

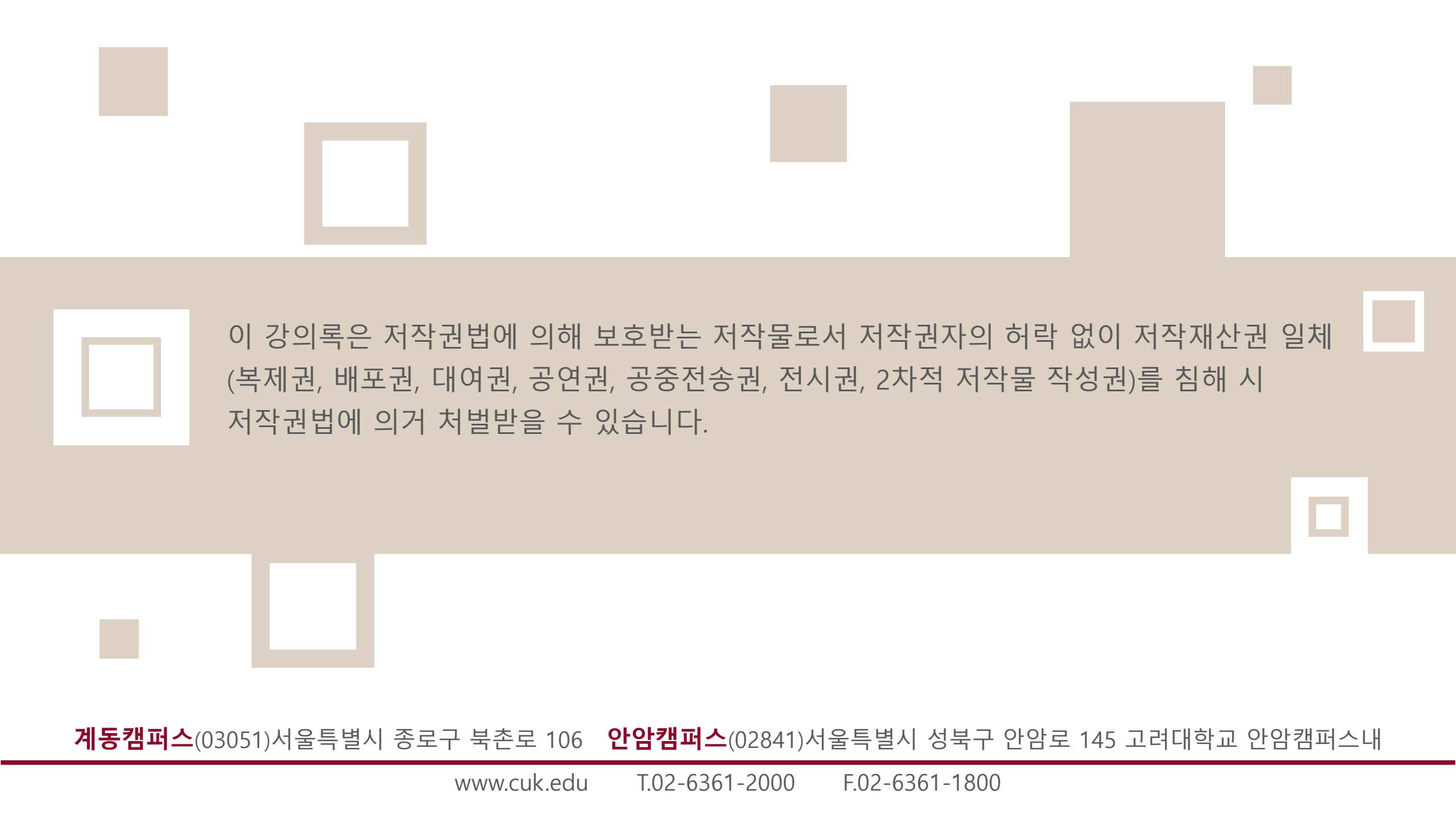
참고 문헌



데이터베이스 시스템 7판,
Ramez Elmasri , Shamkant B. Navathe 지음,
황규영 등 옮김, 홍릉과학출판사,
2018년 8월



www.wikipedia.org



이 강의록은 저작권법에 의해 보호받는 저작물로서 저작권자의 허락 없이 저작권재산권 일체 (복제권, 배포권, 대여권, 공연권, 공중전송권, 전시권, 2차적 저작물 작성권)를 침해 시 저작권법에 의거 처벌받을 수 있습니다.