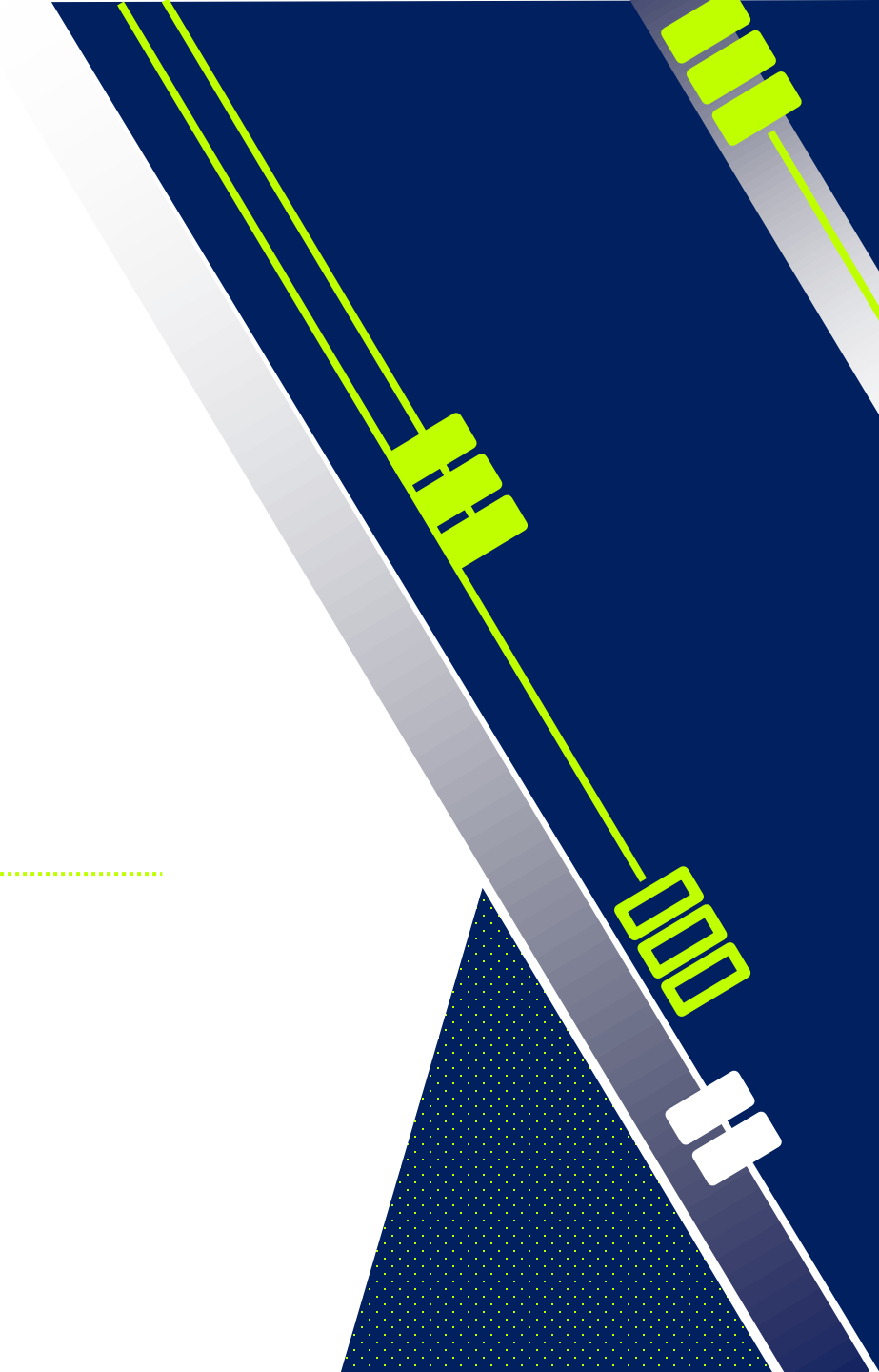


6 강 SQL (3)

■ ■ ■ ■ 컴퓨터과학과 정재화



목 차



01. 데이터 검색 (2)

02. 뷰의 사용



01

■ ■ ■ ■ 6강 SQL (3)

데이터 검색 (2)

- 그룹 질의
- 내부 조인
- 외부 조인

1. 특정 컬럼에 집계함수를 통해 다양한 통계 연산을 수행할 수 있는 기능
2. SELECT 절 또는 HAVING절에 기술
3. 집계 함수의 종류
 - ▶ COUNT: 컬럼에 있는 값들의 개수
 - ▶ SUM: 컬럼에 있는 값들의 합
 - ▶ AVG: 컬럼에 있는 값들의 평균
 - ▶ MAX: 컬럼에서 가장 큰 값
 - ▶ MIN: 컬럼에서 가장 작은 값

? 질의문

단과대학의 개수를 출력하시오.

학과이름	단과대학	주소	전화번호	졸업학점
국어국문학과	인문과학대학	...	02-3668-4550	120
법학과	사회과학대학	...	02-3668-4590	130
생활과학과	자연과학대학	...	02-3668-4640	125
유아교육과	교육과학대학	...	02-3668-4670	125
컴퓨터과학과	자연과학대학	...	02-3668-4650	130
행정학과	사회과학대학	...	02-3668-4600	120
영어영문학과	인문과학대학	...	02-3668-4510	135
자연과학대학	자연과학대학	...	02-3668-4600	140



SQL

```
SELECT COUNT(단과대학)
FROM 학과
```



COUNT(단과대학)
8



SQL

```
SELECT COUNT(DISTINCT 단과대학)  
FROM 학과
```



COUNT(DISTINCT 단과대학)

4



SQL

```
SELECT COUNT(DISTINCT 단과대학) AS 단과대학수  
FROM 학과
```



단과대학수
4

1. 특정 기준으로 레코드를 그룹화하고 각 레코드 그룹에 대해 집계 함수를 적용하는 질의

구문형식

SELECT 질의
GROUP BY 컬럼

2. [주의] SELECT 절에 그룹의 기준과 집계 함수 이외의 컬럼은 포함될 수 없음

? 질의문

소속학과별 교수의 수를 출력하시오.

교수번호	교수이름	직위	소속학과	연봉
186432-760829	최우성	조교수	생활과학과	52000000
201547-634895	현경석	정교수	생활과학과	66000000
189414-790829	한용운	조교수	법학과	45000000
191924-730620	이동휘	부교수	행정학과	51000000
194634-810228	김규식	정교수	컴퓨터과학과	70000000
194834-760517	정재화	부교수	컴퓨터과학과	53000000
201216-158465	정용제	조교수	국어국문학과	55000000
210315-549413	황지수	부교수	유아교육과	52000000



SQL

```
SELECT 소속학과, COUNT(*) AS 교수수  
FROM 교수  
GROUP BY 소속학과
```



교수번호	교수이름	직위	소속학과	연봉
186432-760829	최우성	조교수	생활과학과	52000000
201547-634895	현경석	정교수	생활과학과	66000000
189414-790829	한용운	조교수	법학과	45000000
191924-730620	이동휘	부교수	행정학과	51000000
194634-810228	김규식	정교수	컴퓨터과학과	70000000
194834-760517	정재화	부교수	컴퓨터과학과	53000000
201216-158465	정용제	조교수	국어국문학과	55000000
210315-549413	황지수	부교수	유아교육과	52000000



SQL

```
SELECT 소속학과, COUNT(*) AS 교수수  
FROM 교수  
GROUP BY 소속학과
```



소속학과	교수수
생활과학과	2
법학과	1
행정학과	1
컴퓨터과학과	1
국어국문학과	1
유아교육과	1

SQL

```
SELECT 소속학과, COUNT(*) AS 교수수, 교수이름  
FROM 교수  
GROUP BY 소속학과
```



소속학과	교수수	교수이름
생활과학과	2	
법학과	1	
행정학과	1	
컴퓨터과학과	1	
국어국문학과	1	
유아교육과	1	

1. 그룹 질의의 결과 레코드에 대해 출력 조건을 기술하기 위한 절

구문형식 1

SELECT 질의
GROUP BY 컬럼
HAVING 조건

2. 조건 기술 절

- ▶ WHERE: 레코드에 대한 조건을 기술
- ▶ HAVING: 집계 결과 레코드에 대한 조건을 기술

? 질의문

2개 이상의 전공을 신청한 학생의 학생번호와 신청 전공수를 출력하시오.

학생번호	학과이름	이수학점
201831-331215	국어국문학과	9
201831-331215	컴퓨터과학과	39
201834-021216	국어국문학과	81
201926-880215	생활과학과	48
201931-781109	유아교육과	93
201934-790902	유아교육과	87
201978-610408	법학과	87
⋮	⋮	⋮



SQL

```
SELECT 학생번호, COUNT(*) AS 신청_전공수  
FROM 전공  
GROUP BY 학생번호
```



학생번호	학과이름	이수학점
201831-331215	국어국문학과	9
201831-331215	컴퓨터과학과	39
201834-021216	국어국문학과	81
201926-880215	생활과학과	48
201931-781109	유아교육과	93
201934-790902	유아교육과	87
201978-610408	법학과	87
⋮	⋮	⋮



SQL

```
SELECT 학생번호, COUNT(*) AS 신청_전공수  
FROM 전공  
GROUP BY 학생번호
```



학생번호	신청_전공수
201831-331215	2
201834-021216	1
201926-880215	1
201931-781109	1
201934-790902	1
201978-610408	2
⋮	⋮



SQL

```
SELECT 학생번호, COUNT(*) AS 신청_전공수  
FROM 전공  
GROUP BY 학생번호  
HAVING 신청_전공수 >= 2
```



학생번호	신청_전공수
201831-331215	2
201978-610408	2
202026-590930	2

1. SELECT 문 내부에서 독립적으로 실행 가능한
또 다른 SELECT 문이 내포되어 있는 질의
2. 일반적으로 내부 질의의 처리결과를 외부 질의에서
재사용

3. 중첩 질의의 종류

- ▶ FROM 절에서의 중첩 질의 활용
 - FROM 절에서의 결과 집합을 SELECT에서 재검색
- ▶ WHERE 절에서의 중첩 질의 활용
 - WHERE 절에서의 결과 집합을 활용하여 외부 질의에서 레코드의 출력 여부를 결정
 - IN, NOT IN, EXISTS, NOT EXISTS 사용

구문형식 1

```
SELECT 컬럼1, 컬럼2, ..., 컬럼n  
  FROM (SELECT 컬럼1, 컬럼2, ..., 컬럼m  
        FROM 테이블  
        WHERE 조건)  
WHERE 조건
```

구문형식 2

```
SELECT 컬럼1, 컬럼2, ..., 컬럼n  
  FROM 테이블1  
  WHERE 컬럼i 연산자 (SELECT 컬럼j  
                      FROM 테이블2  
                      WHERE 조건)
```

? 질의문

학과별 교수의 평균 연봉이 70,000,000 미만인 학과 중 가장 높은 평균 연봉을 출력하시오.

교수번호	교수이름	직위	소속학과	연봉
186432-760829	최우성	조교수	생활과학과	52000000
201547-634895	현경석	정교수	생활과학과	66000000
189414-790829	한용운	조교수	법학과	45000000
191924-730620	이동휘	부교수	행정학과	51000000
194634-810228	김규식	정교수	컴퓨터과학과	70000000
194834-760517	정재화	부교수	컴퓨터과학과	53000000
201216-158465	정용제	조교수	국어국문학과	55000000
210315-549413	황지수	부교수	유아교육과	52000000



SQL

```
SELECT 소속학과, AVG(연봉) AS 평균연봉  
FROM 교수  
GROUP BY 소속학과
```



소속학과	평균연봉
국어국문학과	55000000.00
법학과	45000000.00
생활과학과	59000000.00
유아교육과	52000000.00
컴퓨터과학과	61500000.00



SQL

```
SELECT MAX(d.평균연봉) AS 평균연봉  
FROM (SELECT 소속학과, AVG(연봉) AS 평균연봉  
      FROM 교수  
      GROUP BY 소속학과) AS d  
WHERE d.평균연봉 < 70000000
```



평균연봉

61500000.00

? 질의문

'생활과학과' 소속 학생 중 수강신청을 하지 않은 학생의
학생번호를 출력하시오.

학생번호	학과이름	이수학점
201831-331215	국어국문학과	9
201831-331215	컴퓨터과학과	39
201834-021216	국어국문학과	81
201926-880215	생활과학과	48
201931-781109	유아교육과	93
201934-790902	유아교육과	87
201978-610408	법학과	87
⋮	⋮	⋮



SQL

```
SELECT 학생번호  
FROM 전공  
WHERE 학과이름 = '생활과학과'
```



학생번호

201926-880215

202031-354516



SQL

```
SELECT A.학생번호  
FROM 전공 AS A  
WHERE A.학과이름 = '생활과학과' AND  
      NOT EXISTS (SELECT B.학생번호 FROM 수강 B  
                  WHERE A.학생번호 = B.학생번호)
```



학생번호

202031-354516

? 질의문

나이가 30세 이상인 학생의 학생이름과 나이, 그리고 그 학생이 소유한 계좌의 계좌번호, 잔액을 출력하시오.

1. 테이블 간의 관련성을 이용하여 두 개 이상의 테이블에서 데이터를 검색하는 질의 기법
2. ER 모델링 및 정규화 기법으로 여러 테이블로 분리된 정보를 통합하여 검색 시 유용
3. 조인 질의의 종류
 - ▶ 내부조인
 - ▶ 외부조인

1. 두 개 이상의 테이블에서 조인 조건을 만족하는 레코드 만 **결합하여 출력** 결과에 포함시키는 연산
2. 조인 조건은 WHERE 절이 아닌 ON 절에 기록
3. ANSI SQL 표준과 사실상의 표준인 Oracle사가 제안한 조인 형식이 사용

구문형식

```
SELECT 컬럼1, 컬럼2, ..., 컬럼m,  
FROM 테이블1 INNER JOIN 테이블2  
ON 조인조건1  
[WHERE 조건]
```

? 질의문

나이가 30세 이상인 학생의 학생이름과 나이, 그리고 그 학생이 소유한 계좌의 계좌번호, 잔액을 출력하시오.





SQL

```
SELECT 학생이름, 나이, 학생번호
FROM 학생
WHERE 나이 >= 30
```



학생이름	나이	...	학생번호
유관순	118	...	201834-021216
지청천	32	...	201926-880215
안창호	42	...	201931-781109
박은식	61	...	201934-080621
안중근	41	...	201934-790902
손병희	59	...	201978-610408
윤봉길	112	...	202031-816515

학생번호	...	잔액
201831-331215	...	800000
201834-021216	...	600000
201978-610408	...	400000
201931-781109	...	400000
201926-880215	...	300000
201934-790902	...	100000
201934-080621	...	300000
202034-596541	...	1200000
⋮	⋮	⋮





SQL

```
SELECT 학생.학생이름, 학생.나이  
       계좌.계좌번호, 계좌.잔액  
FROM 학생 INNER JOIN 계좌  
ON 학생.학생번호 = 계좌.계좌번호  
WHERE 학생.나이 >= 30
```



학생이름	나이	...	학생번호	...	잔액
유관순	118	...	201834-021216	...	600000
지청천	32	...	201926-880215	...	300000
안창호	42	...	201931-781109	...	400000
박은식	61	...	201934-080621	...	300000
안중근	41	...	201934-790902	...	100000
손병희	59	...	201978-610408	...	400000
윤봉길	112	...	202031-816515	...	150000



SQL(오라클)

```
SELECT 학생.학생이름, 학생.나이  
       계좌.계좌번호, 계좌.잔액  
FROM 학생, 계좌  
WHERE 학생.학생번호 = 계좌.계좌번호  
      AND 학생.나이 >= 30
```



학생이름	나이	...	학생번호	...	잔액
유관순	118	...	201834-021216	...	600000
지청천	32	...	201926-880215	...	300000
안창호	42	...	201931-781109	...	400000
박은식	61	...	201934-080621	...	300000
안중근	41	...	201934-790902	...	100000
손병희	59	...	201978-610408	...	400000
윤봉길	112	...	202031-816515	...	150000

1. 두개 이상의 테이블을 하나의 테이블로 결합하는 내부 조인과 매우 유사한 기능
2. 두 테이블에 **동일한 이름의 컬럼**에 대해 값이 같은 레코드를 결합하는 내부 조인

구문형식

```
SELECT 컬럼1, 컬럼2, ..., 컬럼m,  
FROM 테이블1 NATURAL JOIN 테이블2  
[WHERE 조건]
```

1. 내부 조인(inner join)은 조인조건에 일치하는 레코드만 결합하여 결과를 생성

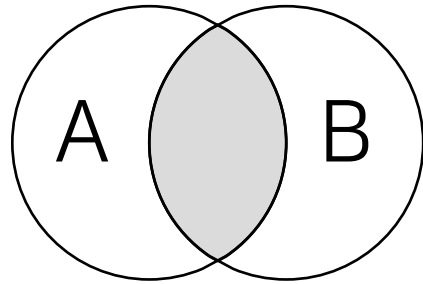
- ▶ 조인 결과에 정보의 손실이 발생

2. 외부 조인은 조인조건에 맞지 않는 레코드도 질의의 결과에 포함시키는 질의

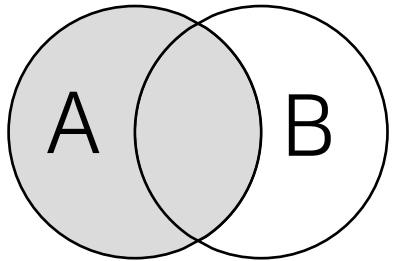
3. 외부 조인의 종류

- ▶ 왼쪽 외부 조인(left outer join)
- ▶ 오른쪽 외부 조인(right outer join)
- ▶ 완전 외부 조인(full outer join)

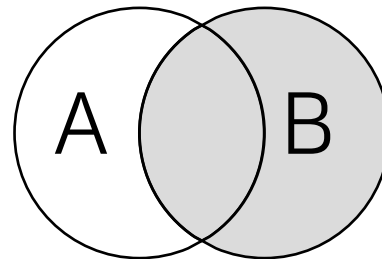
내부 조인



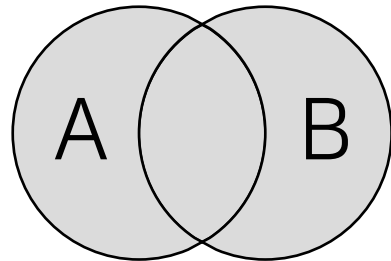
왼쪽 외부 조인



오른쪽 외부 조인



완전 외부 조인



구문형식

```
SELECT 별칭1.컬럼1, 별칭1.컬럼2, ..., 별칭1.컬럼m,  
      별칭2.컬럼1, 별칭2.컬럼2, ..., 별칭2.컬럼n,  
FROM 테이블1 AS 별칭1  
      LEFT|RIGHT [OUTER] JOIN  
      테이블2 AS 별칭2  
ON 별칭1.컬럼i=별칭2.컬럼j  
[WHERE 절]
```

? 질의문

학생의 학생번호, 학생이름과 그 학생이 수강신청한 과목의 과목코드, 신청시각을 출력하시오.
단, 수강신청을 하지 않은 학생도 결과에 포함시키고 과목코드를 기준으로 오름차순 정렬한다.

과목코드	학생번호	신청시각
COM11	201934-790902	2019-02-28 08:32:54
COM11	202026-590930	2019-02-20 16:00:21
COM11	202078-080621	2019-02-21 15:21:54
COM12	201831-331215	2019-08-21 23:25:25
COM12	201931-781109	2018-08-02 03:25:16
COM12	201978-610408	2015-02-24 10:25:40
⋮	⋮	⋮



SQL

```
SELECT A.학생번호, A.학생이름, B.과목코드, B.신청시각  
FROM 학생 AS A LEFT OUTER JOIN 수강 AS B  
ON A.학생번호 = B.학생번호  
ORDER BY B.과목코드 ASC
```

과목코드	학생번호	신청시각
COM11	201934-790902	2019-02-28 08:32:54
COM11	202026-590930	2019-02-20 16:00:21
COM11	202078-080621	2019-02-21 15:21:54
COM12	201831-331215	2019-08-21 23:25:25
COM12	201931-781109	2018-08-02 03:25:16
COM12	201978-610408	2015-02-24 10:25:40
⋮	⋮	⋮



학생

학생이름	...	학생번호
강신영	...	202078-080621
안중근	...	201934-790902
조중대	...	202031-354516
⋮		⋮
정용호	...	202034-596541

수강

학생번호	...	과목코드
202078-080621	...	COM11
201934-790902	...	COM11
201931-781109	...	COM12
⋮		⋮
202034-596541	...	COM24

강신영	...	202078-080621	...	COM11
안중근	...	201934-790902	...	COM11
조중대	...	202031-354516	...	NULL
⋮		⋮		⋮
정용호	...	202034-596541	...	COM24





SQL

```
SELECT A.학생번호, A.학생이름, B.과목코드, B.신청시각  
FROM 학생 AS A LEFT OUTER JOIN 수강 AS B  
ON A.학생번호 = B.학생번호  
ORDER BY B.과목코드 ASC
```



학생번호	학생이름	과목코드	신청시각
202031-354516	조중대	NULL	NULL
201934-080621	박은식	NULL	NULL
201934-790902	안중근	COM11	2019-02-28 08:32:54
202078-080621	강신영	COM11	2019-02-21 15:21:54
202026-590930	정용민	COM11	2019-02-20 16:00:21
201931-781109	안창호	COM12	2018-08-02 03:25:16
⋮	⋮	⋮	⋮

1. 한 테이블이 자기 자신과 조인되는 형태
2. 동일한 이름의 테이블에 대한 조인이므로 반드시 테이블 이름에 대한 **별칭이 의무적으로 사용**

구문형식

```
SELECT 별칭1.컬럼1, 별칭1.컬럼2, ..., 별칭1.컬럼m,  
       별칭2.컬럼1, 별칭2.컬럼2, ..., 별칭2.컬럼n,  
FROM 테이블1 AS 별칭1  
      INNER|OUTER JOIN 테이블2 AS 별칭2  
ON 조인조건  
[WHERE 절]
```

? 질의문

과목의 과목코드, 과목명 그리고 그 과목의 선수과목의 과목코드, 과목명을 모두 출력하시오.
단, 선수과목이 없는 과목도 결과에 포함시킨다.

과목코드	과목명	학점	선수과목	이수구분	교수번호
COM11	컴퓨터의 이해	3		교양	...
COM12	인터넷과 정보사회	3		교양	...
COM24	자료구조	3	COM12	전공필수	...
COM31	데이터베이스 시스템	3	COM24	전공필수	...
COM34	알고리즘	3	COM24	일반선택	...
COM44	클라우드 컴퓨팅	3		전공필수	...
⋮	⋮	⋮	⋮	⋮	⋮



SQL

```
SELECT B.과목명, B.과목코드  
       A.과목명 AS 선수과목명, A.과목코드 AS 선수과목코드  
FROM 과목 AS A RIGHT OUTER JOIN 과목 AS B  
ON A.과목코드 = B.선수과목
```



과목명	과목코드	선수과목명	선수과목코드
컴퓨터의 이해	COM11		
인터넷과 정보사회	COM12		
자료구조	COM24	인터넷과 정보사회	COM12
데이터베이스 시스템	COM31	자료구조	COM24
알고리즘	COM34	자료구조	COM24
클라우드 컴퓨팅	COM44		
⋮	⋮	⋮	⋮

02

6강 SQL (3)

뷰의 사용

- 뷰의 개념
- 뷰의 생성, 수정, 삭제
- 뷰를 이용한 검색 및 수정

1. 데이터를 저장하고 있는 하나 이상의 테이블을 유도하여 생성하는 가상의 테이블(virtual table)

- ▶ 데이터 독립성: 원본 테이블의 구조가 바뀌어도 뷰를 이용한 작업은 정의만 변경되어 응용 프로그램에 영향이 없음
- ▶ 데이터 보안: 사용자에게 원본 테이블의 일부 컬럼에 대한 접근을 허용하여 보안 효과를 향상
- ▶ 다양한 구조의 테이블 사용: 사용자의 요구사항에 맞는 테이블의 구조를 제공
- ▶ 작업의 단순화: 복잡한 질의문을 뷰로 단순화
- ▶ 데이터 무결성: WITH CHECK OPTION을 이용하여 뷰 생성에 위배되는 수정작업을 거부

1. 생성되는 뷰의 구조는 SELECT 문의 결과로 결정

구문형식

```
CREATE VIEW 뷰이름 AS  
  ( SELECT 컬럼1, 컬럼2, ..., 컬럼n  
    FROM 테이블  
    [WHERE 조건] )  
[WITH CHECK OPTION]
```

1. 뷰의 수정은 생성과 동일하게 새로운 SELECT 문의 결과로 변경

구문형식

```
ALTER VIEW 뷰이름(컬럼1, 컬럼2, ..., 컬럼n) AS  
    ( SELECT 컬럼1, 컬럼2, ..., 컬럼n  
      FROM 테이블  
      [WHERE 조건] )
```

구문형식

```
DROP VIEW 뷰이름
```



SQL

```
SELECT 학생.*, 전공.학과이름, 전공.이수학점  
FROM 학생 NATURAL JOIN 전공  
WHERE 전공.학과이름 = '컴퓨터과학과'
```



학생번호	학생이름	성별	생년월일	나이	전화번호	학과이름	이수학점
201831-331215	김마리아	여	1991-06-18	30	010-0000-0002	컴퓨터과학과	39
201934-790902	안중근	남	1979-09-02	42	010-0000-0006	컴퓨터과학과	
202026-590930	정용민	남	2003-05-19	18	010-0000-0012	컴퓨터과학과	137
202031-816515	윤봉길	남	1908-06-21	113	010-0000-0009	컴퓨터과학과	12
202034-596541	정용호	남	2000-01-23	21	010-0000-0008	컴퓨터과학과	117
202078-080621	강신영	남	1991-06-26	30	010-0000-0011	컴퓨터과학과	96

1. 뷰는 가상의 테이블이므로 데이터 조작은 테이블 조작과 동일하게 수행

구문형식

```
SELECT 컬럼1, ..., 컬럼2 FROM 뷰이름  
WHERE 조건
```



실행형식

```
SELECT 컬럼1, ..., 컬럼2 FROM 뷰이름  
WHERE 조건 AND 뷰정의조건
```

1. 뷰에 대한 INSERT 문은 원본 테이블에서 실행

- ▶ PRIMARY KEY, NOT NULL 등의 제약사항이 위배되는 경우 삽입이 불가능
- ▶ 원본 테이블에 존재하는 컬럼이지만 뷰에는 없는 컬럼에 삽입하는 경우 실행 불가능
- ▶ 조인 질의 또는 그룹 질의가 적용된 뷰는 데이터 삽입 및 수정이 불가능
- ▶ WITH CHECK OPTION이 적용된 뷰는 위배되는 사항은 없지만 뷰에 맞지 않는 조건일 경우 실행 불가능



다음 시간에는

7강 저장 객체를

학습하겠습니다.

