

8장. 기본명령어

8.1. SELECT

8.2. WHERE

8.3. AND

8.4. OR

8.5. BETWEEN .. AND

8.6. 비교연산자

8.7. LIKE

8.8. IN

8.9. ORDER BY

8.10. GROUP BY

8.11. DISTINCT

8.12. JOIN

8.13. CASE

8.14. ROWNUM

8.15. NULL

8.16. DUAL

8.17. UPDATE

8.18. INSERT

8.19. MERGE

8.20. DELETE

8.21. COMMIT

8.22. ROLLBACK

8.23. 트랜잭션의 이해

8.24. 서브쿼리(SUB QUERY)의 이해

8.25. EXISTS

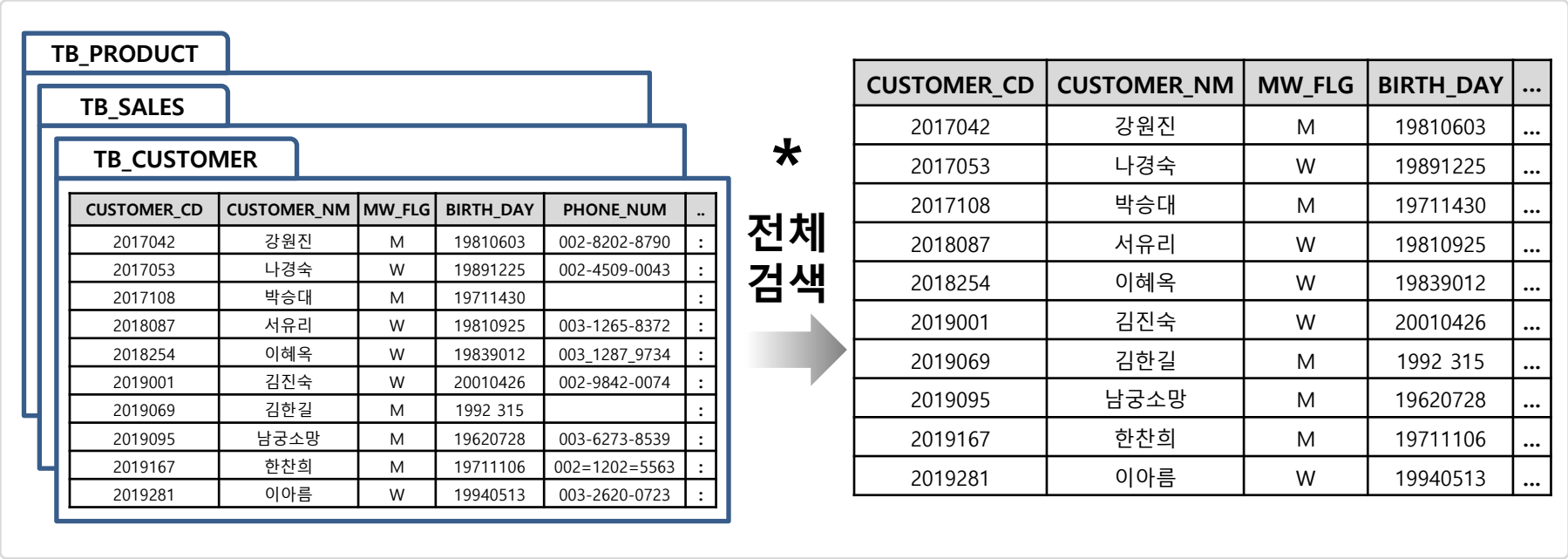
8.26. 검색 조건을 입력해서 SQL을 실행

8.27. 주식 처리

SELECT

KEYWORD : 테이블 검색

구문 #1	SELECT * FROM 테이블;
구문 #2	SELECT 필드1, 필드2 ..., 필드n FROM 테이블;
구문 #3	SELECT 필드1 [AS] 별명1, 필드2 [AS] 별명2, ... , 필드n [AS] 별명n FROM 테이블;
사용 예	SELECT * FROM TB_CUSTOMER; SELECT CUSTOMER_CD, EMAIL FROM TB_CUSTOMER; CUSTOMER_CD AS 고객코드, CUSTOMER_NM 고객명

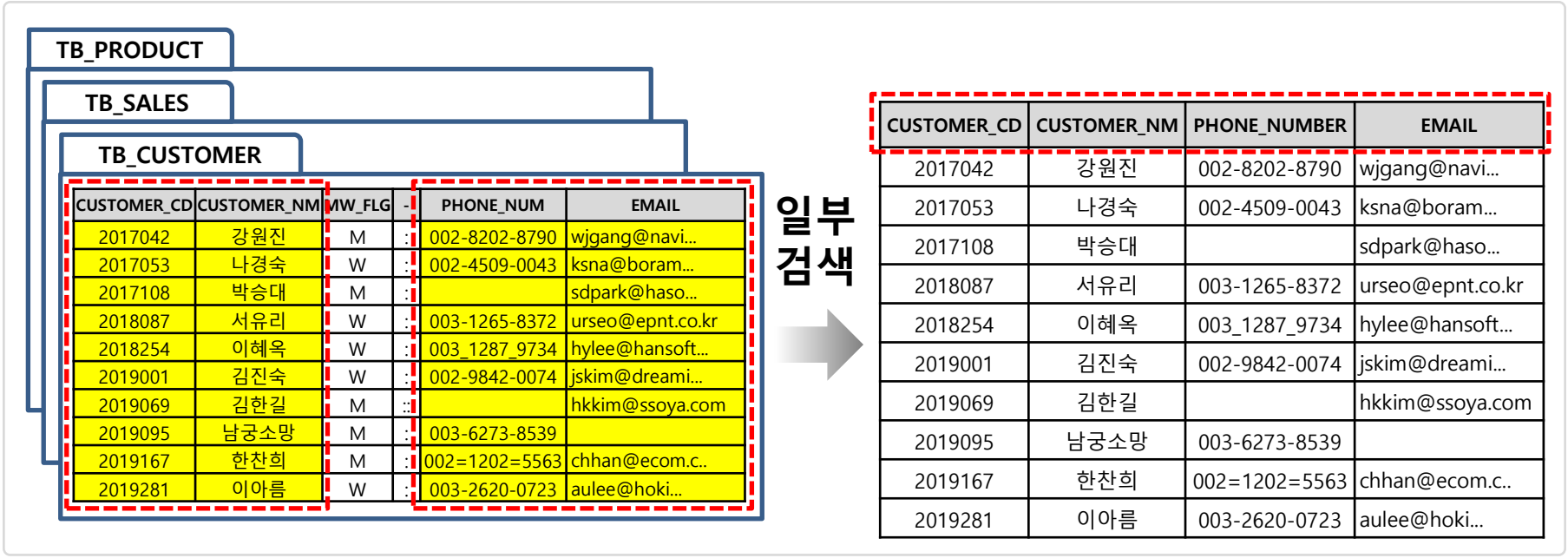


:: TB_CUSTOMER 테이블의 전체 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

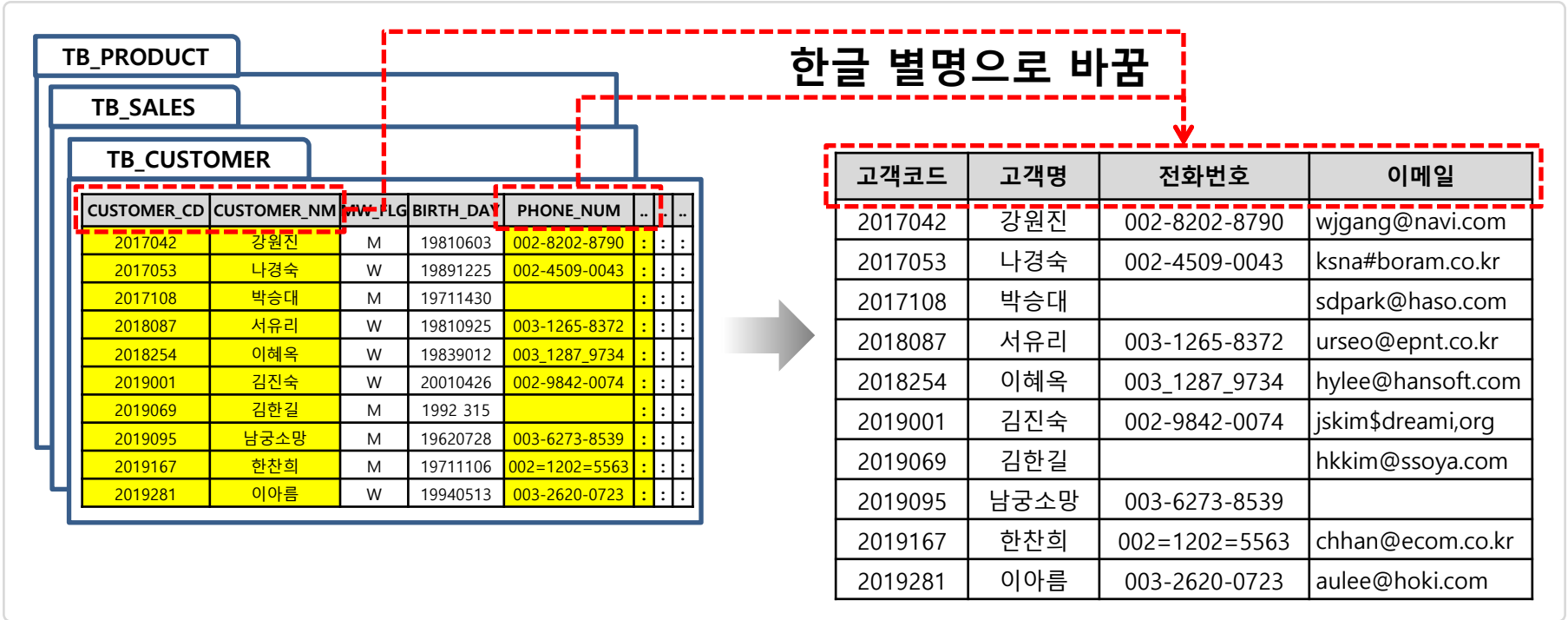


:: TB_CUSTOMER 테이블에서 CUSTOMER_CD, CUSTOMER_NM, PHONE_NUMBER, EMAIL 항목에 해당하는 데이터를 검색


```
SELECT CUSTOMER_CD,  
       CUSTOMER_NM,  
       PHONE_NUMBER,  
       EMAIL  
FROM   TB_CUSTOMER;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ PHONE_NUMBER	⚡ EMAIL
2017042	강원진	002-8202-8790	wjgang@navi.com
2017053	나경숙	002-4509-0043	ksna#boram.co.kr
2017108	박승대		sdpark@haso.com
2018087	서유리	003-1265-8372	urseo@epnt.co.kr
2018254	이혜옥	003_1287_9734	hylee@hansoft.com
2019001	김진숙	002-9842-0074	jskim\$dreami.org
2019069	김한길		hkkim@ssoya.com
2019095	남궁소망	003-6273-8539	
2019167	한찬희	002=1202=5563	chhan@ecom.co.kr
2019281	이아름	003-2620-0723	aulee@hoki.com



:: TB_CUSTOMER 테이블에서 CUSTOMER_CD, CUSTOMER_NM, PHONE_NUMBER, EMAIL 항목명을 한글로 검색

```
SELECT CUSTOMER_CD AS 고객코드 ,  
       CUSTOMER_NM AS 고객명 ,  
       PHONE_NUMBER AS 전화번호 ,  
       EMAIL AS 이메일  
FROM   TB_CUSTOMER;
```

결과

고객코드	고객명	전화번호	이메일
2017042	강원진	002-8202-8790	wjgang@navi.com
2017053	나경숙	002-4509-0043	ksna#boram.co.kr
2017108	박승대		sdpark@haso.com
2018087	서유리	003-1265-8372	urseo@epnt.co.kr
2018254	이혜옥	003_1287_9734	hylee@hansoft.com
2019001	김진숙	002-9842-0074	jskim@dreami.org
2019069	김한길		hkkim@ssoya.com
2019095	남궁소망	003-6273-8539	
2019167	한찬희	002=1202=5563	chhan@ecom.co.kr
2019281	이마름	003-2620-0723	aulee@hoki.com

```
SELECT CUSTOMER_CD AS "Customer Cd",  
       CUSTOMER_NM AS "01_고객명",  
       MW_FLG AS "_성별_",  
       PHONE_NUMBER AS "전화 번호"  
FROM   TB_CUSTOMER;
```

결과

Customer Cd	01_고객명	_성별_	전화 번호
2017042	강원진	M	002-8202-8790
2017053	나경숙	W	002-4509-0043
2017108	박승대	M	
2018087	서유리	W	003-1265-8372
2018254	이혜옥	W	003_1287_9734
2019001	김진숙	W	002-9842-0074
2019069	김한길	M	
2019095	남궁소망	M	003-6273-8539
2019167	한찬희	M	002=1202=5563
2019281	이아름	W	003-2620-0723

□ 테이블 별명의 예

순서	테이블명 (소문자)	테이블명 (대문자)	테이블 설명	별명
1	tb_customer	TB_CUSTOMER	고객	CU
2	tb_add_customer	TB_ADD_CUSTOMER	추가고객	AC
3	tb_point	TB_POINT	포인트	CP
4	tb_point_2017	TB_POINT_2017	2017년도 포인트	PT17
5	tb_point_2018	TB_POINT_2018	2018년도 포인트	PT18
6	tb_point_2019	TB_POINT_2019	2019년도 포인트	PT19
7	tb_etc_info	TB_ETC_INFO	기타정보	EI
8	tb_item_info	TB_ITEM_INFO	품목정보	II
9	tb_sales	TB_SALES	판매	SA
10	tb_sales_09	TB_SALES_09	9월 판매	SA09
11	tb_point_info	TB_POINT_INFO	포인트정보	PI
12	tb_grade	TB_GRADE	성적	GD
13	tb_grade_07	TB_GRADE_07	7월 성적	GD07
14	tb_grade_08	TB_GRADE_08	8월 성적	GD08
15	tb_grade_09	TB_GRADE_09	9월 성적	GD09
16	tb_grade_2019	TB_GRADE_2019	2019년 성적	GD19
17	tb_grade_2020	TB_GRADE_2020	2020년 성적	GD20
18	tb_class_info	TB_CLASS_INFO	반정보	CI
19	tb_train_tm	TB_TRAIN_TM	열차시간표	TT

```
SELECT A.CUSTOMER_CD,
       A.CUSTOMER_NM,
       A.MW_FLG,
       B.POINT
FROM   TB_CUSTOMER A,
       TB_POINT B
WHERE  A.CUSTOMER_CD = '2017042'
AND    A.CUSTOMER_CD = B.CUSTOMER_CD
AND    B.POINT > 10000;
```

```
SELECT CU.CUSTOMER_CD,
       CU.CUSTOMER_NM,
       CU.MW_FLG,
       CP.POINT
FROM   TB_CUSTOMER CU,
       TB_POINT CP
WHERE  CU.CUSTOMER_CD = '2017042'
AND    CU.CUSTOMER_CD = CP.CUSTOMER_CD
AND    CP.POINT > 10000;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	POINT
2017042	강원진	M	120700
2017042	강원진	M	78560
2017042	강원진	M	71540

(1) TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956

(2) TB_POINT

CUSTOMER_CD	POINT_MEMO	POINT
2017042	청소기 구매 포인트 적립	120700
2017042	이벤트 포인트 적립	9500
2017042	냉장고 구매 포인트 적립	78560
2017042	에어컨 구매 포인트 적립	71540
2017053	세탁기 구매 포인트 적립	3500
2017053	드라이기 구매 포인트 적립	2000
2017108	청소기 구매 포인트 적립	14065
2017108	이벤트 포인트 적립	9385
2018087	이벤트 포인트 적립	7800
2018087	냉장고 구매 포인트 적립	10900

(3) TB_POINT

*고객코드	포인트 내용	포인트
2017042	청소기 구매 포인트 적립	120700
2017042	이벤트 포인트 적립	9500
2017042	냉장고 구매 포인트 적립	78560
2017042	에어컨 구매 포인트 적립	71540
2017053	세탁기 구매 포인트 적립	3500
2017053	드라이기 구매 포인트 적립	2000
2017108	청소기 구매 포인트 적립	14065
2017108	이벤트 포인트 적립	9385
2018087	이벤트 포인트 적립	7800
2018087	냉장고 구매 포인트 적립	10900

- :: (1) 고객포인트 테이블의 모든 필드를 검색한다.
- :: (2) 고객포인트 테이블에서 고객코드, 포인트내용, 포인트를 검색한다.
- :: (3) 고객포인트 테이블에서 고객코드, 포인트내용, 포인트 필드 제목을 한글로 출력한다.

WHERE

KEYWORD : 테이블 검색 조건

구 문	SELECT 검색필드 FROM 테이블 WHERE 조건식;
사용 예	WHERE CUSTOMER_NM='나경숙'; WHERE KOR=100;

TB_CUSTOMER				
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUM
2017042	강원진	M	19810603	002-8202-8790
2017053	나경숙	W	19891225	002-4509-0043
2017108	박승대	M	19711430	
2018087	서유리	W	19810925	003-1265-8372
2018254	이혜옥	W	19839012	003_1287_9734
2019001	김진숙	W	20010426	002-9842-0074
2019069	김한길	M	1992 315	
2019095	남궁소망	M	19620728	003-6273-8539
2019167	한찬희	M	19711106	002=1202=5563
2019281	이아름	W	19940513	003-2620-0723

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017108	박승대	M	19711430
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106

:: TB_CUSTOMER 테이블에서 MW_FLG가 'M'인 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  MW_FLG = 'M';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	201902171110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600

TB_CUSTOMER				
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER
2017042	강원진	M	19810603	002-8202-8790
2017053	나경숙	W	19891225	002-4509-0043
2017108	박승대	M	19711430	
2018087	서유리	W	19810925	003-1265-8372
2018254	이혜옥	W	19839012	003_1287_9734
2019001	김진숙	W	20010426	002-9842-0074
2019069	김한길	M	1992 315	
2019095	남궁소망	M	19620728	003-6273-8539
2019167	한찬희	M	19711106	002=1202=5563
2019281	이아름	W	19940513	003-2620-0723

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2019069	김한길	M	1992 315

∴ TB_CUSTOMER 테이블에서 CUSTOMER_NM이 '김한길'인 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  CUSTOMER_NM = '김한길';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992_315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	laulee@hoki.com	35600	20190709201308

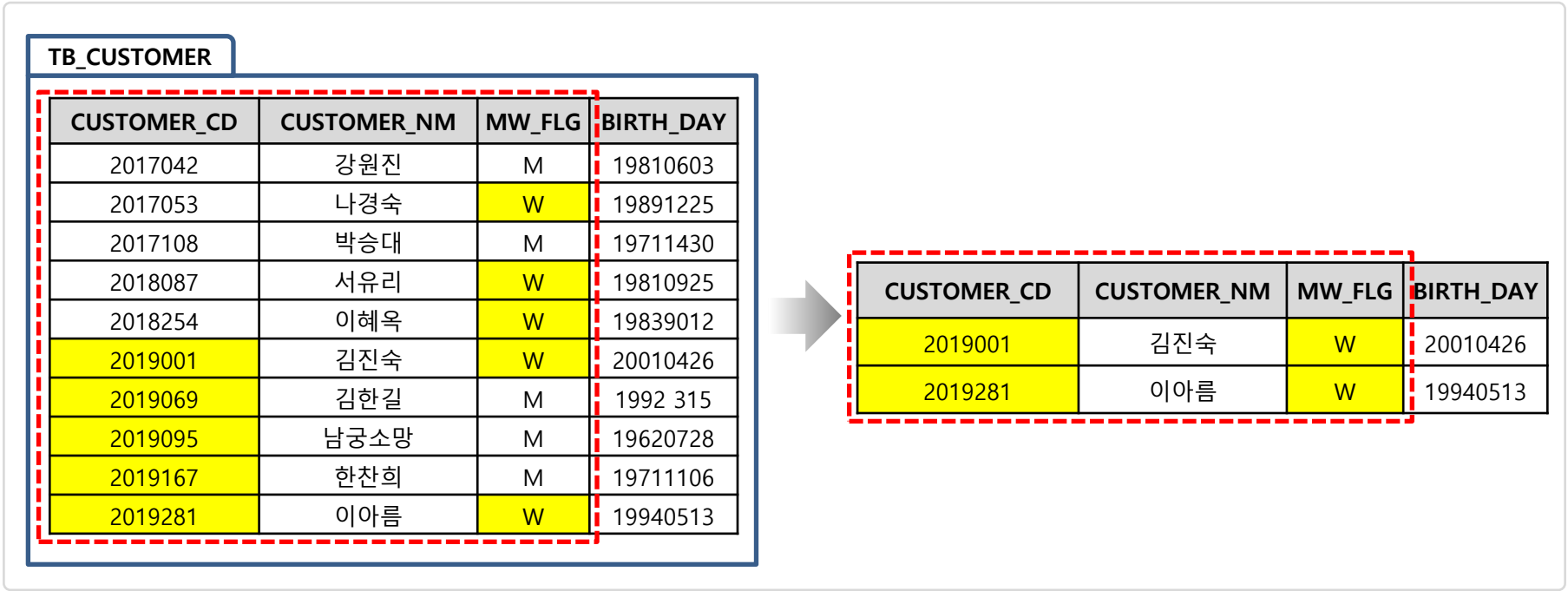
CUSTOMER_CD	CUSTOMER_NM	EMAIL	TOTAL_POINT
2017053	나경숙	ksna#boram.co.kr	4500
2018254	이혜옥	hylee@hansoft.com	570
2019095	남궁소망		890
2019167	한찬희	chhan@ecom.co.kr	6800

∴ 고객관리 테이블에서 누적포인트가 10,000 미만인 데이터의 고객코드, 고객명, 이메일, 누적포인트 필드를 검색한다.

AND

KEYWORD : 모두 만족하는 조건 검색

구 문	SELECT 검색필드 FROM 테이블 WHERE 조건식1 AND 조건식2 AND ..., 조건식n;
사용 예	WHERE CLASS_CD='A' AND KOR=100;



:: TB_CUSTOMER 테이블에서 CUSTOMER_CD가 '2019000' 이후이고 MW_FLG가 'W'인 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  CUSTOMER_CD > '2019000'  
AND    MW_FLG = 'W';
```

결과

↕ CUSTOMER_CD	↕ CUSTOMER_NM	↕ MW_FLG	↕ BIRTH_DAY	↕ PHONE_NUMBER	↕ EMAIL	↕ TOTAL_POINT	↕ REG_DTTM
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017108	박승대	M	19711430
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106

:: TB_CUSTOMER 테이블에서 BIRTH_DAY가 '1990'년 이전이고 MW_FLG가 'M'인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  BIRTH_DAY < '19900101'
AND    MW_FLG = 'M';
```

결과

↕ CUSTOMER_CD	↕ CUSTOMER_NM	↕ MW_FLG	↕ BIRTH_DAY	↕ PHONE_NUMBER	↕ EMAIL	↕ TOTAL_POINT	↕ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017108	박승대	M	197111430		sdpark@haso.com	23450	20170508203450
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956



CUSTOMER_CD	SEQ_NO	POINT
2017053	2	2000

:: 고객포인트 테이블에서 고객코드가 '2017053'이면서 일련번호가 2인 데이터의 고객코드, 일련번호, 포인트 필드를 검색한다.

OR

KEYWORD : 하나라도 만족하는 조건 검색

구 문	SELECT 검색필드 FROM 테이블 WHERE 조건식1 OR 조건식2 OR ..., 조건식n;
사용 예	WHERE MW_FLG='M' OR BIRTH_DAY='20200301';

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2017108	M	19711430	23450
2018087	W	19810925	18700
2018254	W	19839012	570
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

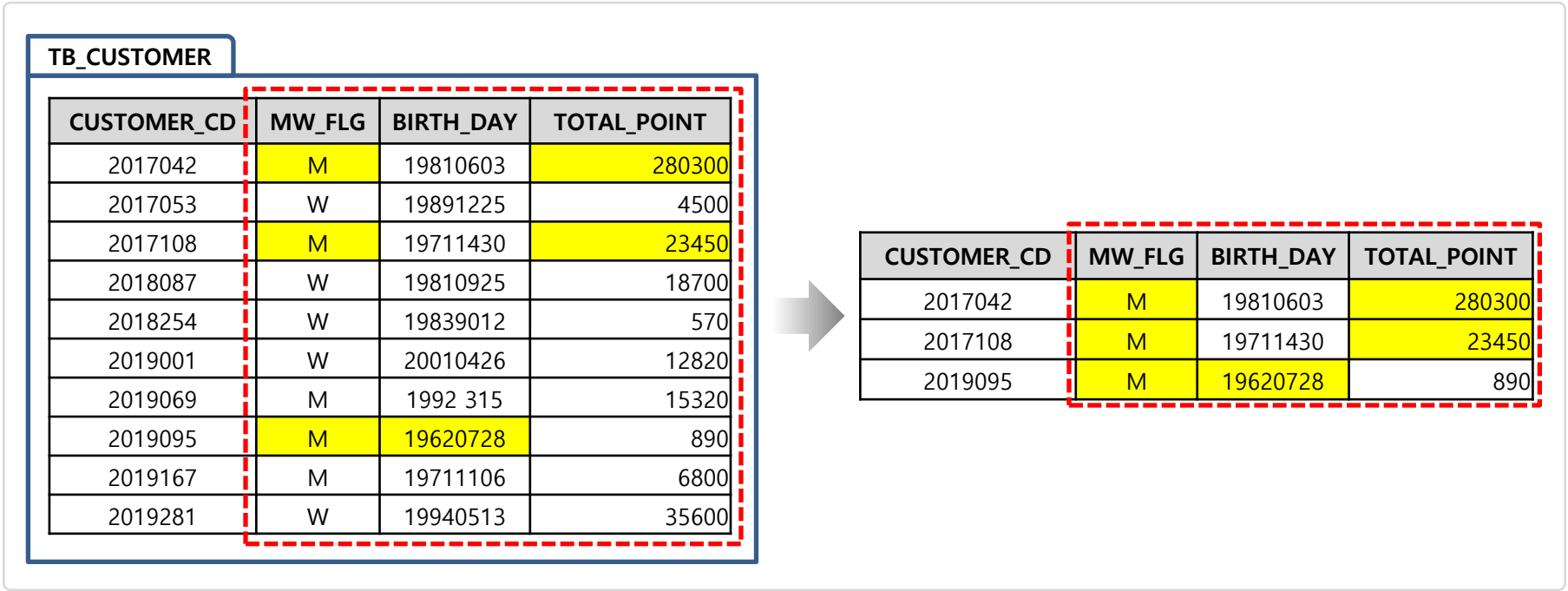
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017108	M	19711430	23450
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019281	W	19940513	35600

:: TB_CUSTOMER 테이블에서 BIRTH_DAY가 '19900101' 이후 이거나 TOTAL_POINT가 20,000 이상인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  BIRTH_DAY >= '19900101'
OR     TOTAL_POINT >= 20000;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019001	김진숙	W	20010426	002-9842-0074	jskim@dreami.org	12820	20190101080518
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308



:: TB_CUSTOMER 테이블에서 MW_FLG가 'M'인 데이터 중 BIRTH_DAY가 '19700101' 이전 이거나 TOTAL_POINT가 20,000 이상을 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  MW_FLG = 'M'
AND    (BIRTH_DAY < '19700101'
OR      TOTAL_POINT >= 20000);
```

결과

↕ CUSTOMER_CD	↕ CUSTOMER_NM	↕ MW_FLG	↕ BIRTH_DAY	↕ PHONE_NUMBER	↕ EMAIL	↕ TOTAL_POINT	↕ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558

TB_GRADE

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0



CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
D	3	이아름	100	87	95	0	0

∴ 성적 테이블에서 반코드가 'A' 또는 'B'이거나 국어, 영어, 수학 점수가 모두 80점 이상인 학생 필드를 검색한다.

BETWEEN .. AND

KEYWORD : 범위에 만족하는 조건 검색

구 문	SELECT 검색필드 FROM 테이블 WHERE 필드 [NOT] BETWEEN 시작값 AND 종료값;
사 용 예	WHERE POINT BETWEEN 200 AND 400 WHERE CLASS_CD BETWEEN 'A' AND 'C' WHERE REG_DTTM BETWEEN SYSDATE AND SYSDATE+60 WHERE POINT NOT BETWEEN 200 AND 400

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2017108	M	19711430	23450
2018087	W	19810925	18700
2018254	W	19839012	570
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2018087	W	19810925	18700
2019001	W	20010426	12820

:: TB_CUSTOMER 테이블에서 MW_FLG가 'W' 이고 TOTAL_POINT가 10,000에서 20,000
까지의 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  MW_FLG = 'W'
AND    TOTAL_POINT BETWEEN 10000 AND 20000;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2017108	M	19711430	23450
2018087	W	19810925	18700
2018254	W	19839012	570
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2018087	W	19810925	18700
2018254	W	19839012	570

:: TB_CUSTOMER 테이블에서 BIRTH_DAY가 '19800101'에서 '19891231' 사이 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  BIRTH_DAY BETWEEN '19800101' AND '19891231';
```

결과

↕ CUSTOMER_CD	↕ CUSTOMER_NM	↕ MW_FLG	↕ BIRTH_DAY	↕ PHONE_NUMBER	↕ EMAIL	↕ TOTAL_POINT	↕ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2017108	M	19711430	23450
2018087	W	19810925	18700
2018254	W	19839012	570
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2018254	W	19839012	570
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

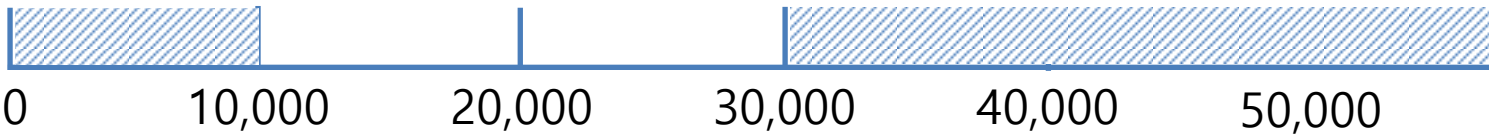
:: TB_CUSTOMER 테이블에서 TOTAL_POINT가 10,000 미만 이거나 30,000 이상인 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  TOTAL_POINT NOT BETWEEN 10000 AND 30000;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

힌트



TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956



CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956

:: 고객포인트 테이블에서 등록일시가 2018년 내에 있고, 포인트가 10,000에서 50,000 포인트 범위의 데이터를 검색한다.

비교연산자

KEYWORD : 크다, 작다, 같다, 다르다 조건 검색

구 문	SELECT 검색필드명 FROM 테이블명 WHERE 조건식;
비교연산자	= : 같다 <>, !=, ^= : 다르다 A > B : A가 B보다 크다 A < B : A가 B보다 작다 A >= B : A가 B보다 크거나 같다 A <= B : A가 B보다 작거나 같다
사용 예	WHERE KOR < 80 WHERE REG_DTTM <> SYSDATE WHERE POINT >= 200 AND POINT <= 400 WHERE POINT < 200 OR POINT > 400

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2017108	M	19711430	23450
2018087	W	19810925	18700
2018254	W	19839012	570
2019001	W	20010426	12820
2019069	M	1992 315	15320
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

CUSTOMER_CD	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	M	19810603	280300
2017053	W	19891225	4500
2018254	W	19839012	570
2019095	M	19620728	890
2019167	M	19711106	6800
2019281	W	19940513	35600

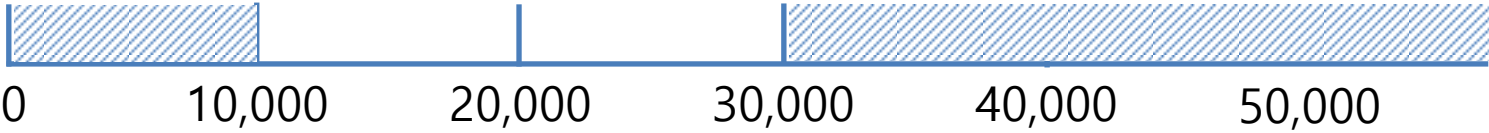
:: TB_CUSTOMER 테이블에서 TOTAL_POINT가 10,000 미만 이거나 30,000 이상인 데이터를 검색

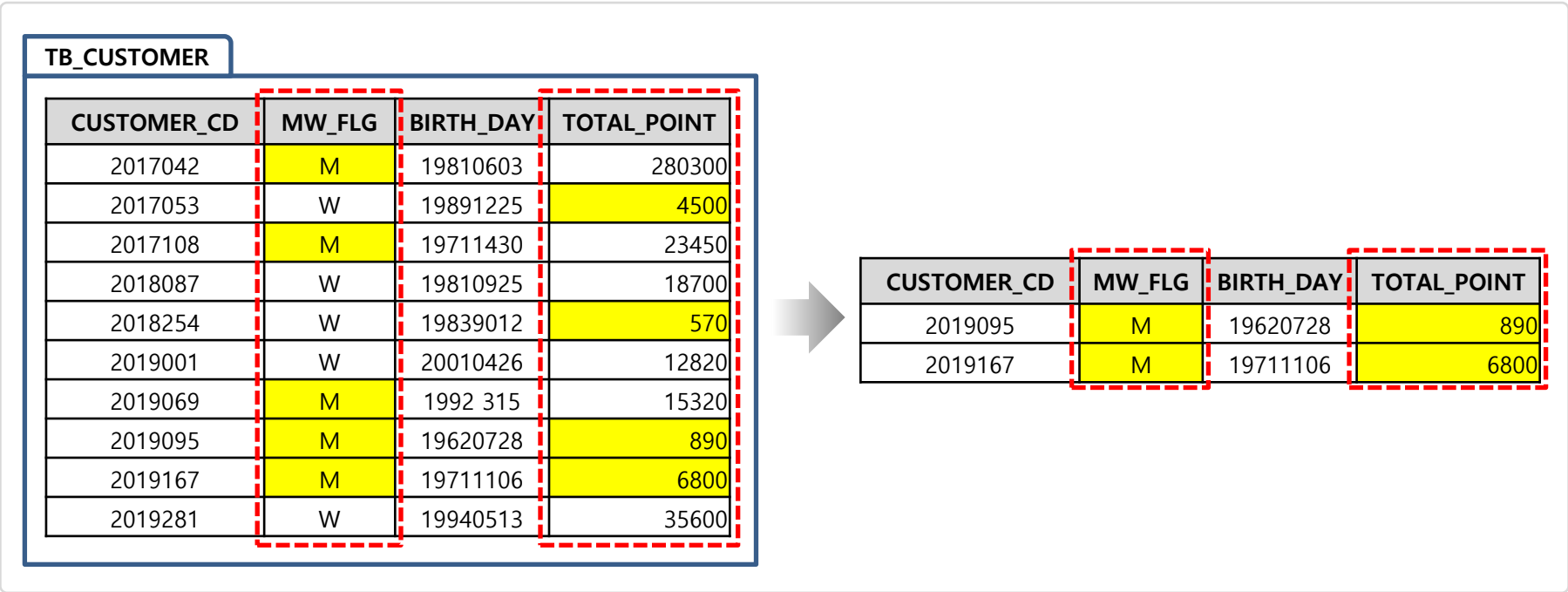
```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  TOTAL_POINT < 10000  
OR     TOTAL_POINT >= 30000;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

힌트





:: TB_CUSTOMER 테이블에서 MW_FLG가 'W'가 아니면서 TOTAL_POINT가 10,000 이하인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  MW_FLG <> 'W'
AND    TOTAL_POINT <= 10000;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600

TB_CUTSOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992_315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308



CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	강원진	M	19810603	280300

:: 고객 테이블에서 누적포인트가 20,000 이상인 1980년대 남성 고객의 고객코드, 고객명, 성별, 생년월일, 누적포인트를 검색한다.

LIKE

KEYWORD : 포함된 문자로 검색하기

구문 #1	SELECT 검색필드 FROM 테이블 WHERE 필드 [NOT] LIKE '%검색값%';
구문 #2	SELECT 필드 FROM 테이블 WHERE 필드 [NOT] LIKE '_검색값_';
참 고	'문자%': 문자로 시작하는 값 '%문자': 문자로 끝나는 값 '%문자%': 문자가 포함된 값 '_': 1자리 문자면 무엇이든 가능

사 용 예	WHERE BIRTH_DAY LIKE '2020%' WHERE RUN_TM LIKE '%: %' WHERE PLN_NO LIKE '_S____' WHERE CLASS_DV_CD NOT LIKE '%T%'
-------	--

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012

:: TB_CUSTOMER 테이블에서 CUSTOMER_CD가 '2018' 로 시작하는 데이터를 검색

```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  CUSTOMER_CD LIKE '2018%';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513



CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017053	나경숙	W	19891225
2019001	김진숙	W	20010426
2019281	이아름	W	19940513

:: TB_CUSTOMER 테이블에서 CUSTOMER_CD가 '2017 또는 '2019'로 시작하는 데이터 중에서 MW_FLG가 'W'인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  (CUSTOMER_CD LIKE '2017%'
OR      CUSTOMER_CD LIKE '2019%')
AND    MW_FLG = 'W';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930

TB_CUSTOMER

CUSTOMER_CD	MW_FLG	BIRTH_DAY	PHONE_NUMBER
2017042	M	19810603	002-8202-8790
2017053	W	19891225	002-4509-0043
2017108	M	19711430	
2018087	W	19810925	003-1265-8372
2018254	W	19839012	003_1287_9734
2019001	W	20010426	002-9842-0074
2019069	M	1992 315	
2019095	M	19620728	003-6273-8539
2019167	M	19711106	002=1202=5563
2019281	W	19940513	003-2620-0723

CUSTOMER_CD	MW_FLG	BIRTH_DAY	PHONE_NUMBER
2018254	W	19839012	003_1287_9734
2019167	M	19711106	002=1202=5563

:: TB_CUSTOMER 테이블에서 PHONE_NUMBER이 ' __-__-__ ' 형식이 아닌 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  PHONE_NUMBER NOT LIKE '____-____-____';
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600

TB_CUTSOMER							
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	미혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992_315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	TOTAL_POINT
2017042	강원진	M	19810603	280300
2019095	남궁소망	M	19620728	890

:: 고객 테이블에서 남성이면서 생년월일 중 월이 5, 6, 7월인 고객의 고객코드, 고객명, 성별, 생년월일, 누적포인트를 검색한다.

TB_CUTSOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992_315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308



CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	TOTAL_POINT
2019001	김진숙	W	20010426	12820
2017108	박승대	M	19711430	23450

∴ 고객 테이블에서 고객코드가 '2017'로 시작하면서 남성인 고객 또는 고객코드가 '2019'로 시작하면서 여성인 고객을 구하고, 그 중 누적포인트가 30000 이하인 데이터를 검색한다.

IN

KEYWORD : 다중 데이터 조건 검색

구 문	SELECT 검색필드 FROM 테이블 WHERE 필드 [NOT] IN (검색값1, 검색값2 ..., 검색값n);
사용 예	WHERE REG_YEAR IN ('2018', '2019', '2020') WHERE REG_YEAR NOT IN ('2018')

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017053	나경숙	W	19891225
2018254	이혜옥	W	19839012
2019069	김한길	M	1992 315
2019281	이아름	W	19940513

'김진철', '강수지'는 해당 데이터가 없어 출력되지 않음

:: TB_CUSTOMER 테이블에서 CUSTOMER_NM이 '나경숙', '이혜옥', '김진철', '김한길', '강수지', '이아름'인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  CUSTOMER_NM IN ('나경숙','이혜옥','김진철','김한길','강수지','이아름');
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017108	박승대	M	19711430
2019167	한찬희	M	19711106

:: TB_CUSTOMER 테이블에서 CUSTOMER_CD가 '2017108', '2018254', '2019167' 이면서 MW_FLG가 'M'인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD IN ( '2017108', '2018254', '2019167' )
AND    MW_FLG = 'M' ;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600

TB_ITEM_INFO

ITEM_CD	ITEM_NM
S01	의류/잡화
S02	뷰티
S03	레저/자동차
S04	식품
S05	생활/건강
S06	가구/인테리어
S07	가전
S08	도서/취미
S09	컴퓨터
S10	브랜드

➔

ITEM_CD	ITEM_NM
S01	의류/잡화
S04	식품
S06	가구/인테리어
S10	브랜드

∴ 품목정보 테이블에서 품목코드가 'S01', 'S04', 'S06', 'S10'인 데이터를 검색한다.

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956



CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956

:: 고객포인트 테이블에서 고객코드가 '2017042' 또는 '2018087' 또는 '2019095' 이면서
포인트내용에 '구매' 문자가 포함된 데이터를 검색한다.

ORDER BY

KEYWORD : 데이터 정렬 방식 지정

구문 #1	SELECT 검색필드 FROM 테이블 WHERE 조건식 ORDER BY 필드1 [ASC DESC], 필드2 [ASC DESC] ..., 필드n [ASC DESC];
구문 #2	SELECT 검색필드 FROM 테이블 WHERE 조건식 ORDER BY 필드번호 [ASC DESC], 필드번호2 [ASC DESC] ..., 필드번호n [ASC DESC];
사용 예	ORDER BY CLASS_CD, CLASS_NO ORDER BY TOTAL_POINT DESC, CUSTOMER_NM ORDER BY 2 DESC, 4

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

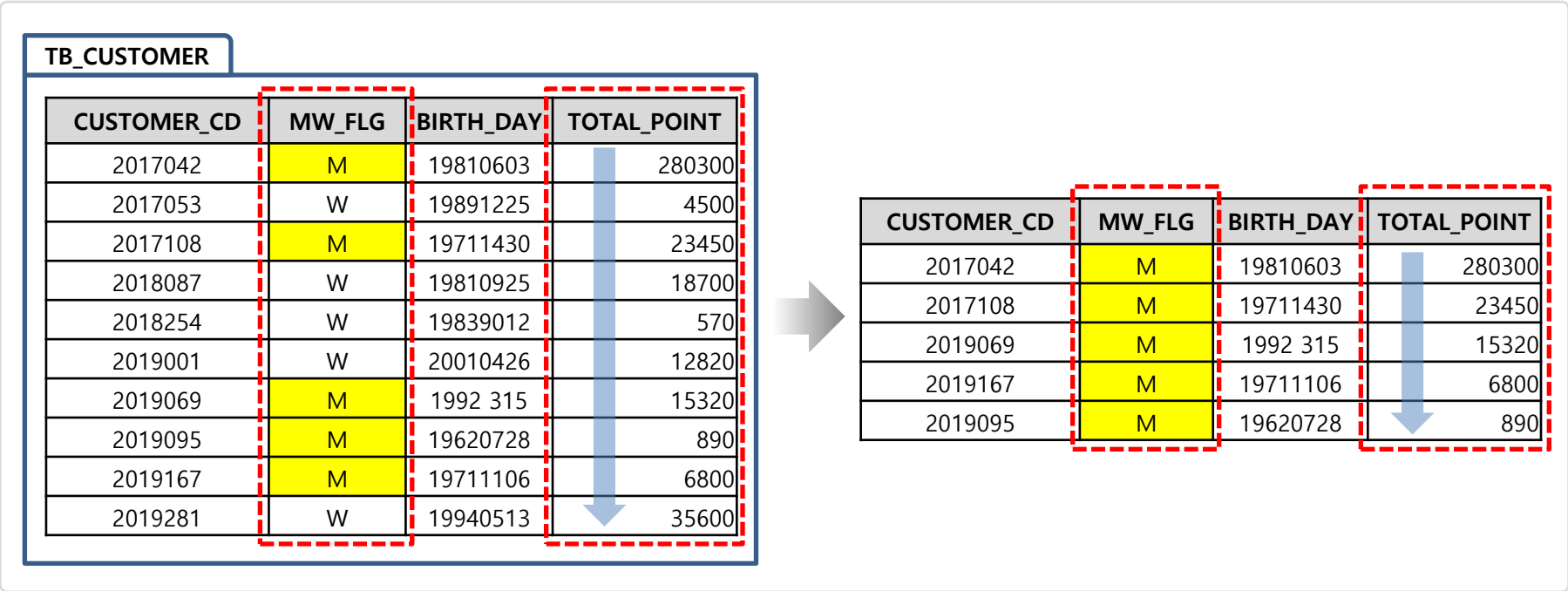
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2019001	김진숙	W	20010426
2017053	나경숙	W	19891225
2018087	서유리	W	19810925
2019281	이아름	W	19940513
2018254	이혜옥	W	19839012
2017042	강원진	M	19810603
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2017108	박승대	M	19711430
2019167	한찬희	M	19711106

:: TB_CUSTOMER 테이블에서 MW_FLG로 내림차순, CUSTOMER_NM으로 오름차순 정렬

```
SELECT *  
FROM   TB_CUSTOMER  
ORDER  BY MW_FLG DESC,  
        CUSTOMER_NM;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600



:: TB_CUSTOMER 테이블에서 MW_FLG가 'M'인 데이터를 TOTAL_POINT로 내림차순 정렬

```
SELECT *
FROM   TB_CUSTOMER
WHERE  MW_FLG = 'M'
ORDER  BY TOTAL_POINT DESC;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956



CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2019281	3	냉장고 구매 포인트 적립	12200	20190712082334
2019281	2	청소기 구매 포인트 적립	8950	20190710200921
2019001	3	드라이기 구매 포인트 적립	7820	20190829071234
2019281	4	드라이기 구매 포인트 적립	7600	20190721134421
2019281	5	믹서기 구매 포인트 적립	6250	20190724022430
2019001	2	믹서기 구매 포인트 적립	4600	20190405134554
2019167	2	전기밥솥 구매 포인트 적립	3600	20190714133422
2019069	3	전기밥솥 구매 포인트 적립	3220	20190620071230

∴ 고객포인트 테이블에서 등록일이 '2019 ' 년이고 포인트내용에 '구매'가 포함된 데이터를
포인트가 큰 순서대로 검색한다.

TB_GRADE

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0



CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
B	2	서유리	96	100	92	0	0
B	1	박승대	90	92	86	0	0
B	3	이혜옥	98	86	78	0	0

:: 성적 테이블에서 'B'반의 국어, 영어, 수학 점수의 합계가 높은 순으로 검색한다.

GROUP BY

KEYWORD : 최소, 최대, 합계, 평균값 등의 취득을 위한 그룹화

구 문	SELECT 검색필드 FROM 테이블 WHERE 조건식 GROUP BY 필드1, 필드2 ..., 필드n HAVING 그룹 내 조건식;
사용 예	GROUP BY CLASS_CD, STUDY_GRP SELECT MAX(KOR) ... GROUP BY CLASS_CD

[참고] 그룹의 통계를 내는 집계함수(MIN, MAX, SUM, AVG 등)와 그룹내 데이터를 구하는 함수를 실습하는 단위에서는 그룹함수로 통칭해 실습을 진행.

TB_GRADE					
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT
A	1	강원진	87	94	98
A	2	나경숙	68	86	78
B	1	박승대	90	92	86
B	2	서유리	96	100	92
B	3	이혜옥	98	86	78
C	1	김진숙	95	77	95
C	2	김한길	73	84	100
D	1	남궁소망	56	68	78
D	2	한찬희	94	90	68
D	3	이아름	100	87	95

반코드 그룹화

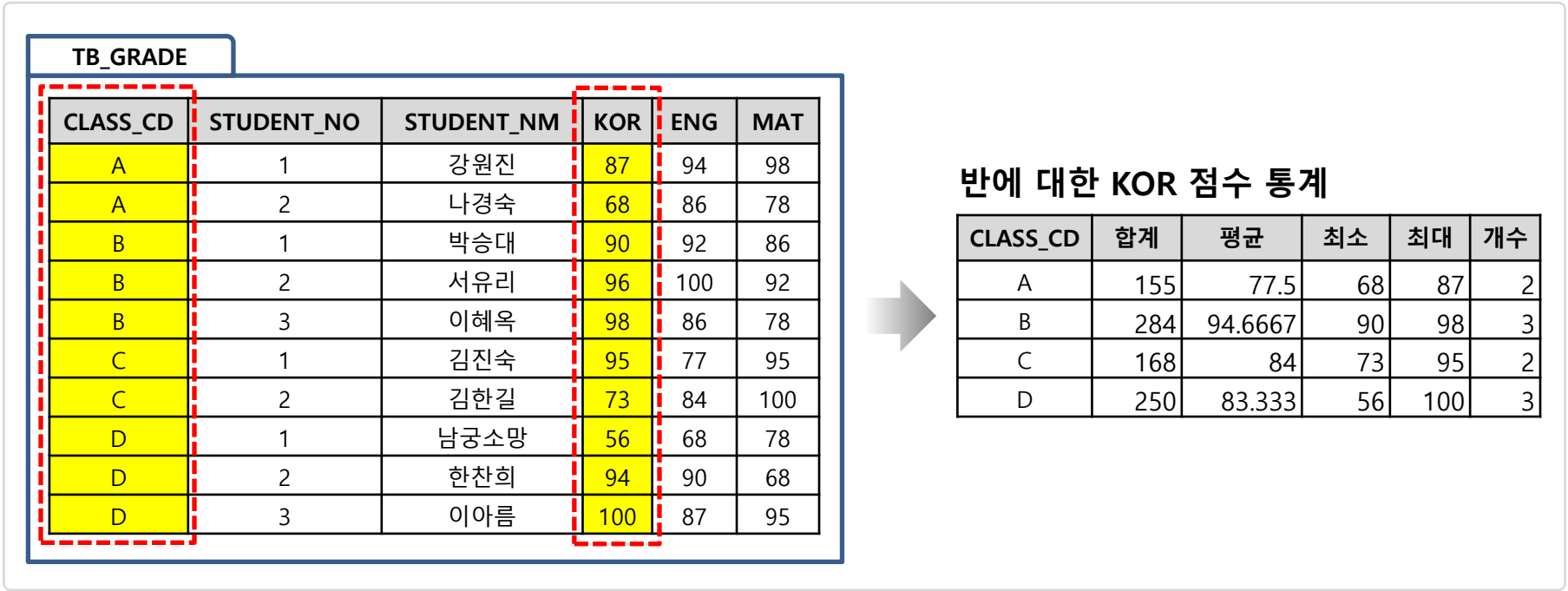
CLASS_CD
A
B
C
D

:: TB_GRADE 테이블에서 CLASS_CD로 그룹화한다.

```
SELECT CLASS_CD
FROM   TB_GRADE
GROUP BY CLASS_CD;
```

결과

↕ CLASS_CD
A
B
C
D



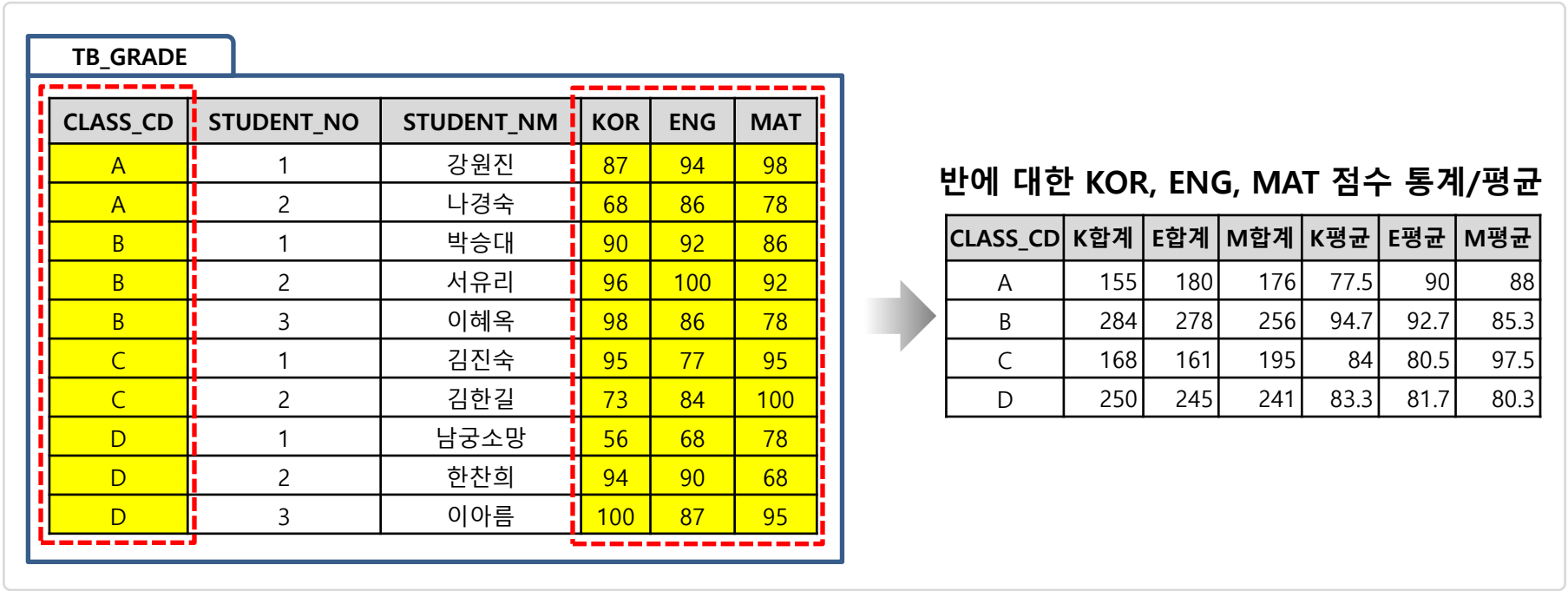
:: TB_GRADE 테이블에서 CLASS_CD로 그룹화해서 KOR 항목의 합계, 평균, 최소, 최대값, 개수 검색

결과

[illegible]

테이블					
필드				SUM(성적) 함수의 결과	
학년	반	번호	성적	학년	학년, 반
1	1	1	86	350	178
1	1	2	92		
1	2	1	90		172
1	2	2	82		
2	1	1	78	370	272
2	1	2	100		
2	1	3	94		
2	2	1	84		182
2	2	2	98		
3	1	1	100	176	100
3	2	1	76		76

:: 그룹화 필드의 SUM(값) 함수 결과

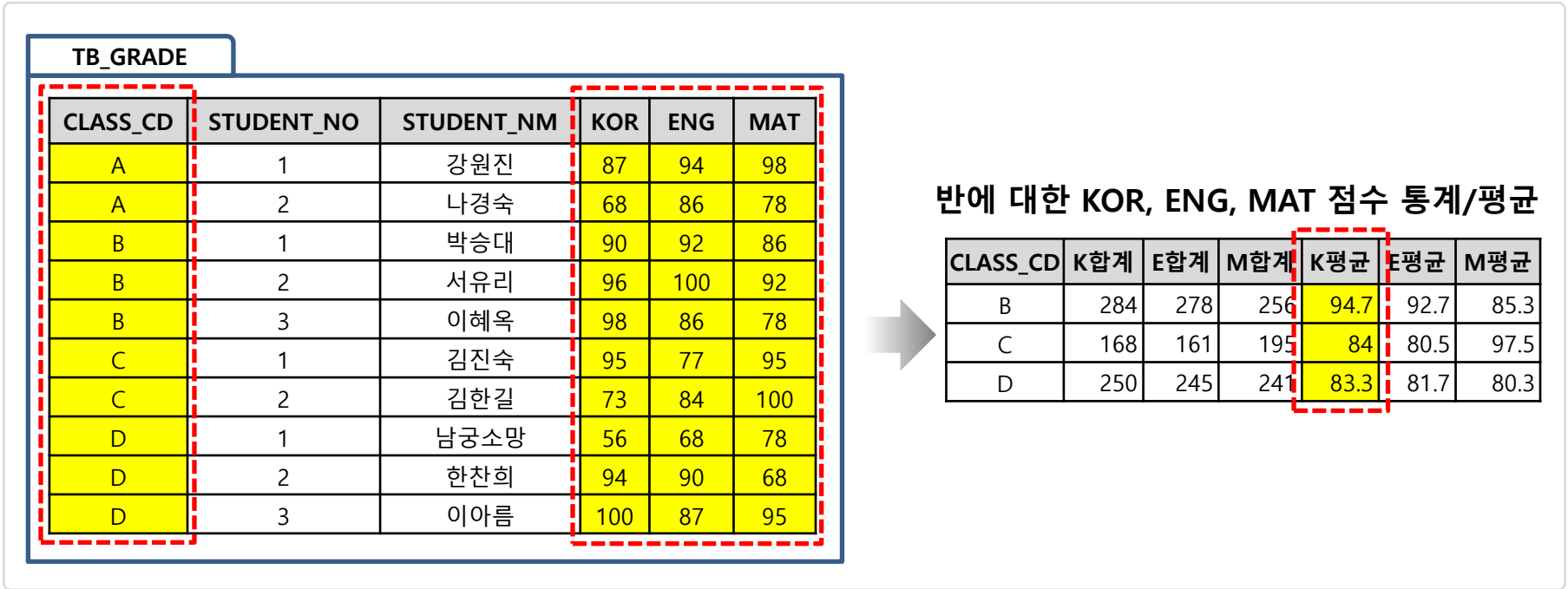


:: TB_GRADE 테이블에서 CLASS_CD로 그룹화해서 KOR, ENG_, MAT 항목의 합계와 평균을 검색

```
SELECT CLASS_CD,  
       SUM(KOR) AS TOT_KOR,  
       SUM(ENG) AS TOT_ENG,  
       SUM(MAT) AS TOT_MAT,  
       ROUND(AVG(KOR),1) AS AVG_KOR,  
       ROUND(AVG(ENG),1) AS AVG_ENG,  
       ROUND(AVG(MAT),1) AS AVG_MAT  
FROM   TB_GRADE  
GROUP  BY CLASS_CD;
```

결과

CLASS_CD	TOT_KOR	TOT_ENG	TOT_MAT	AVG_KOR	AVG_ENG	AVG_MAT
A	155	180	176	77.5	90	88
B	284	278	256	94.7	92.7	85.3
C	168	161	195	84	80.5	97.5
D	250	245	241	83.3	81.7	80.3



:: TB_GRADE 테이블에서 CLASS_CD로 그룹화해서 KOR, ENG, MAT 항목의 합계와 평균을 구하되 KOR 평균이 80 이상인 것만 검색


```
SELECT CLASS_CD,  
       SUM(KOR) AS TOT_KOR,  
       SUM(ENG) AS TOT_ENG,  
       SUM(MAT) AS TOT_MAT,  
       ROUND(AVG(KOR),1) AS AVG_KOR,  
       ROUND(AVG(ENG),1) AS AVG_ENG,  
       ROUND(AVG(MAT),1) AS AVG_MAT  
FROM   TB_GRADE  
GROUP  BY CLASS_CD  
HAVING AVG(KOR) >= 80;
```

결과

CLASS_CD	TOT_KOR	TOT_ENG	TOT_MAT	AVG_KOR	AVG_ENG	AVG_MAT
B	284	278	256	94.7	92.7	85.3
C	168	161	195	84	80.5	97.5
D	250	245	241	83.3	81.7	80.3

TB_SALES

SALES_DT	SEQ_NO	PRODUCT_NM	PRICE	SALES_COUNT
20190801	1	노트북	1045000	4
20190801	2	청소기	545000	2
20190801	3	노트북	1021000	6
20190802	1	선풍기	70000	21
20190802	2	냉장고	1870000	7
20190803	1	선풍기	92000	32
20190803	2	냉장고	2012000	3
20190803	3	선풍기	70000	15
20190803	4	선풍기	92000	24
20190804	1	청소기	980000	5



SALES_DT	PRODUCT_NM	총판매수
20190802	냉장고	7
20190802	선풍기	21
20190803	냉장고	3
20190803	선풍기	71

:: 판매 테이블에서 판매일이 '20190802' 또는 '20190803'을 대상으로 판매일과 상품명으로 그룹화해 총판매수를 구하고 판매일과 상품명은 가나다 순으로 보인다.

DISTINCT

KEYWORD : 출력항목 중복 제거

구 문	SELECT DISTINCT 검색필드 FROM 테이블 WHERE 조건식;
사용 예	DISTINCT SALES_MONTH, PRODUCT_NM

TB_SALES

SALES_DT	SEQ_NO	PRODUCT_NM	PRICE	SALES_COUNT
20190801	1	노트북	1045000	4
20190801	2	청소기	545000	2
20190801	3	노트북	1021000	6
20190802	1	선풍기	70000	21
20190802	2	냉장고	1870000	7
20190803	1	선풍기	92000	32
20190803	2	냉장고	2012000	3
20190803	3	선풍기	70000	15
20190803	4	선풍기	92000	24
20190804	1	청소기	980000	5

노트북 중복

선풍기 중복

SALES_DT	PRODUCT_NM
20190801	노트북
20190801	청소기
20190802	냉장고
20190802	선풍기
20190803	냉장고
20190803	선풍기
20190804	청소기

:: TB_SALES 테이블에서 SALES_DT와 PRODUCT_NM 항목을 중복 없이 검색

```
SELECT DISTINCT SALES_DT,  
                PRODUCT_NM  
FROM    TB_SALES  
ORDER BY SALES_DT,  
        PRODUCT_NM;
```

결과

SALES_DT	PRODUCT_NM
20190801	노트북
20190801	청소기
20190802	냉장고
20190802	선풍기
20190803	냉장고
20190803	선풍기
20190804	청소기

TB_SALES						
SALES_DT	SEQ_NO	PRODUCT_NM	PRICE	SALES_COUNT		
20190801	1	노트북	1045000	4	노트북 중복	PRODUCT_NM
20190801	2	청소기	545000	2		노트북
20190801	3	노트북	1021000	6	선풍기, 냉장고 중복	청소기
20190802	1	선풍기	70000	21		냉장고
20190802	2	냉장고	1870000	7		선풍기
20190803	1	선풍기	92000	32		
20190803	2	냉장고	2012000	3	청소기 중복	
20190803	3	선풍기	70000	15		
20190803	4	선풍기	92000	24		
20190804	1	청소기	980000	5		

:: TB_SALES 테이블에서 PRODUCT_NM 항목을 중복 없이 검색

```
SELECT DISTINCT PRODUCT_NM
FROM    TB_SALES
ORDER  BY PRODUCT_NM;
```

결과

PRODUCT_NM
냉장고
노트북
선풍기
청소기

TB_TRAIN_TM

TRAIN_NO	RUN_ORDR	STATION_NM	ARV_TM	DPT_TM
101	1	서울	092000	092000
101	2	수원	095400	095600
101	3	대전	103800	104100
101	4	대구	112500	112730
101	5	부산	130500	130500
103	1	서울	130800	130800
103	2	대전	142100	142300
103	3	대구	151200	151430
103	4	부산	160400	160400
106	1	부산	083500	083500
106	2	대전	100200	100530
106	3	서울	112800	112800



TRAIN_NO
101
103
106

:: 열차시각표 테이블에서 열차번호를 중복없이 검색한다.

TB_SALES					
SALES_DT	SEQ_NO	PRODUCT_NM	PRICE	SALES_COUNT	
20190801	1	노트북	1045000	4	
20190801	2	청소기	545000	2	
20190801	3	노트북	1021000	6	
20190802	1	선풍기	70000	21	
20190802	2	냉장고	1870000	7	
20190803	1	선풍기	92000	32	
20190803	2	냉장고	2012000	3	
20190803	3	선풍기	70000	15	
20190803	4	선풍기	92000	24	
20190804	1	청소기	980000	5	

PRODUCT_NM

냉장고

노트북

선풍기

청소기

:: 판매 테이블에서 '20190801 ' 에서 '20190802' 기간에 판매한 상품명을 가나다 순으로 중복없이 검색한다.

JOIN

KEYWORD : 테이블의 연관 정보 검색

구문 #1	SELECT 검색필드 FROM 테이블1 JOIN 테이블2 ON 테이블1(별명1).필드1 = 테이블2(별명2).필드1 AND 테이블1(별명1).필드2 = 테이블2(별명2).필드2 WHERE 조건식;
구문 #2	SELECT 검색필드 FROM 테이블1 (별명1), 테이블2 (별명2) WHERE 테이블1(별명1).필드1 = 테이블2(별명2).필드1(+);

사 용 예	<pre>SELECT A.TEST_CD, B.TEST_NM FROM TB_GRADE A JOIN TB_CLASS_CD B ON A.TEST_CD = B.TEST_CD;</pre>
-------	---

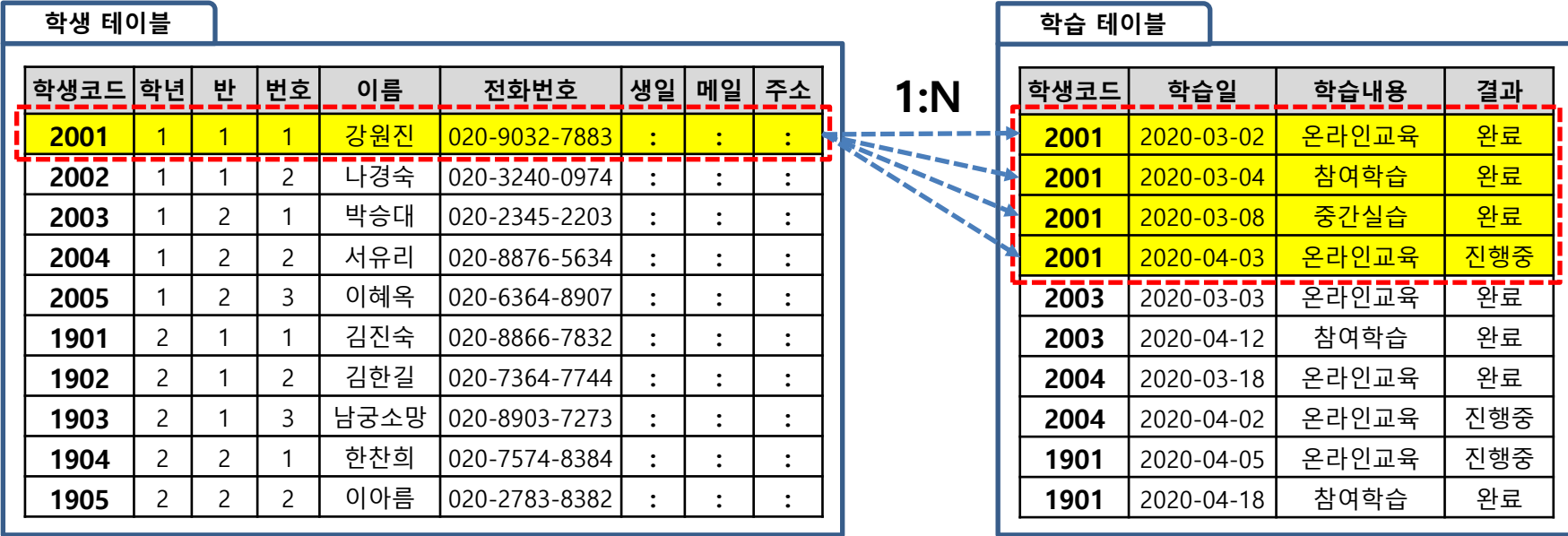
학생 테이블

학생코드	학년	반	번호	이름	전화번호	생일	메일	주소
2001	1	1	1	강원진	020-9032-7883	:	:	:
2002	1	1	2	나경숙	020-3240-0974	:	:	:
2003	1	2	1	박승대	020-2345-2203	:	:	:
2004	1	2	2	서유리	020-8876-5634	:	:	:
2005	1	2	3	이혜옥	020-6364-8907	:	:	:
1901	2	1	1	김진숙	020-8866-7832	:	:	:
1902	2	1	2	김한길	020-7364-7744	:	:	:
1903	2	1	3	남궁소망	020-8903-7273	:	:	:
1904	2	2	1	한찬희	020-7574-8384	:	:	:
1905	2	2	2	이아름	020-2783-8382	:	:	:

학습 테이블

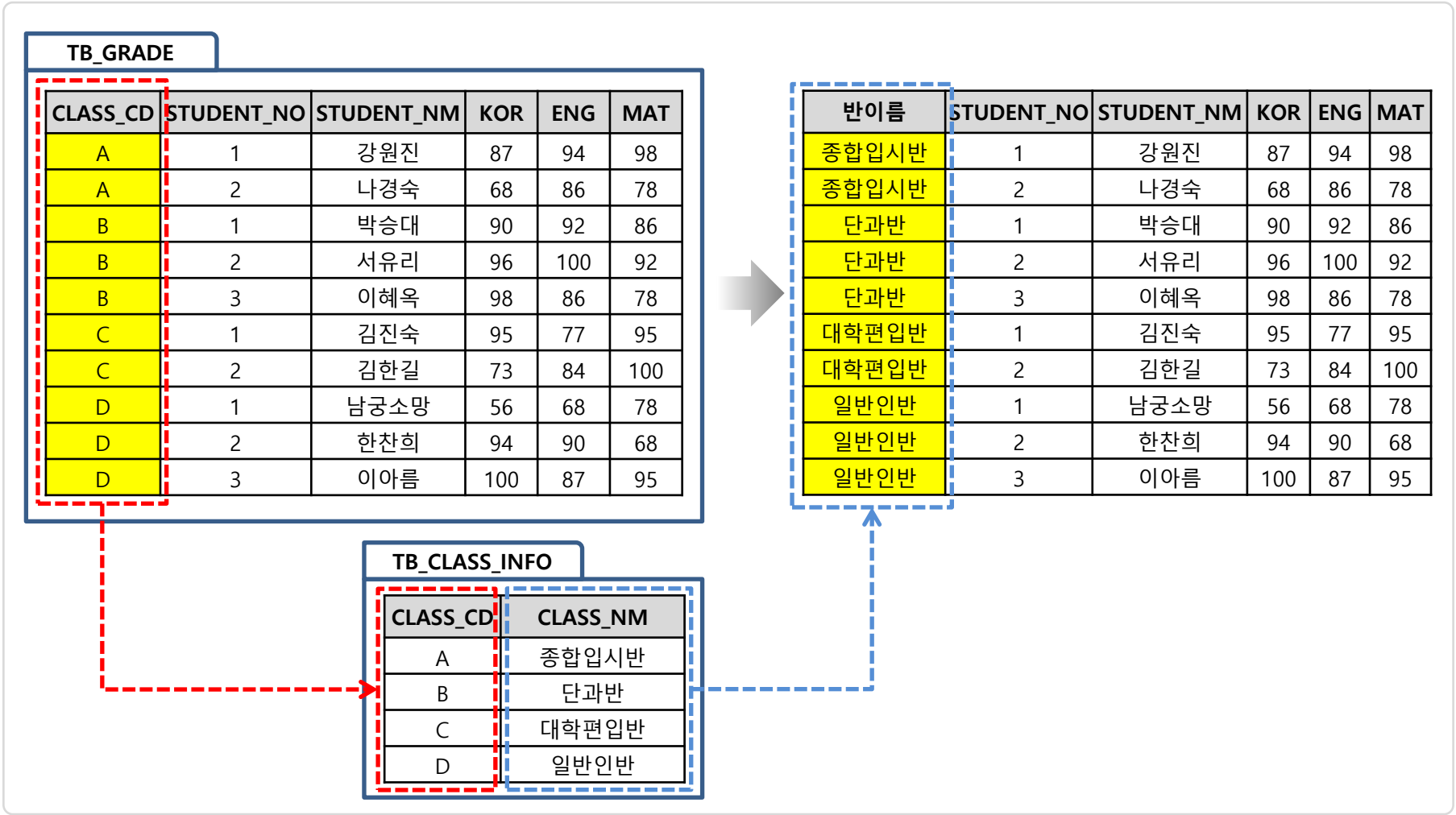
학생코드	학습일	학습내용	결과
2001	2020-03-02	온라인교육	완료
2001	2020-03-04	참여학습	완료
2001	2020-03-08	중간실습	완료
2001	2020-04-03	온라인교육	진행중
2003	2020-03-03	온라인교육	완료
2003	2020-04-12	참여학습	완료
2004	2020-03-18	온라인교육	완료
2004	2020-04-02	온라인교육	진행중
1901	2020-04-05	온라인교육	진행중
1901	2020-04-18	참여학습	완료

```
SELECT *
FROM 학생테이블 A
JOIN 학습테이블 B
ON A.학생코드 = B.학생코드;
```



학생 테이블의 학생코드를 학습 테이블의 학생코드와 JOIN

학생코드	학년	반	번호	이름	전화번호	생일	메일	주소	학생코드	학습일	학습내용	결과
2001	1	1	1	강원진	020-9032-7883	:	:	:	2001	2020-03-02	온라인교육	완료
2001	1	1	1	강원진	020-9032-7883	:	:	:	2001	2020-03-04	참여학습	완료
2001	1	1	1	강원진	020-9032-7883	:	:	:	2001	2020-03-08	중간실습	완료
2001	1	1	1	강원진	020-9032-7883	:	:	:	2001	2020-04-03	온라인교육	진행중
2003	1	2	1	박승대	020-2345-2203	:	:	:	2003	2020-03-03	온라인교육	완료
2003	1	2	1	박승대	020-2345-2203	:	:	:	2003	2020-04-12	참여학습	완료
2004	1	2	2	서유리	020-8876-5634	:	:	:	2004	2020-03-18	온라인교육	완료
2004	1	2	2	서유리	020-8876-5634	:	:	:	2004	2020-04-02	온라인교육	진행중
1901	2	1	1	김진숙	020-8866-7832	:	:	:	1901	2020-04-05	온라인교육	진행중
1901	2	1	1	김진숙	020-8866-7832	:	:	:	1901	2020-04-18	참여학습	완료

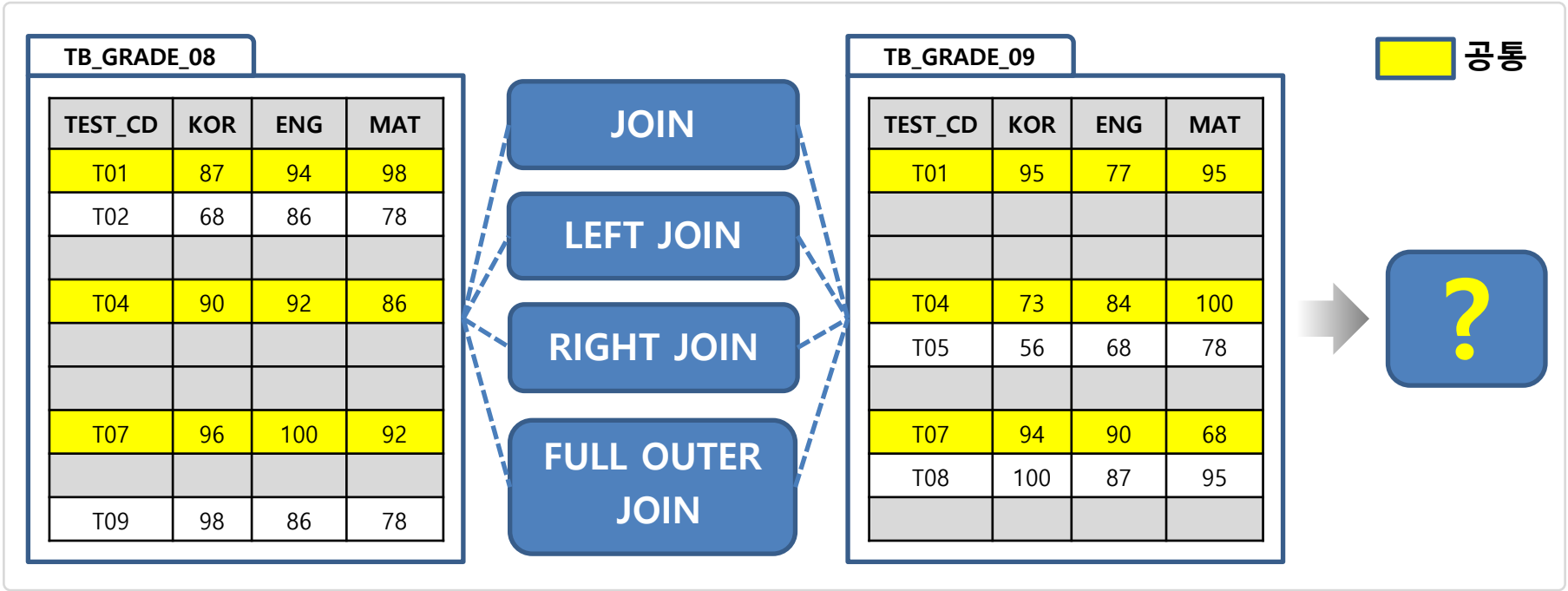


∴ TB_GRADE 테이블의 CLASS_CD와 TB_CLASS_CD 테이블의 CLASS_CD 항목을 연관 검색하여 CLASS_CD를 CLASS_NM으로 변경

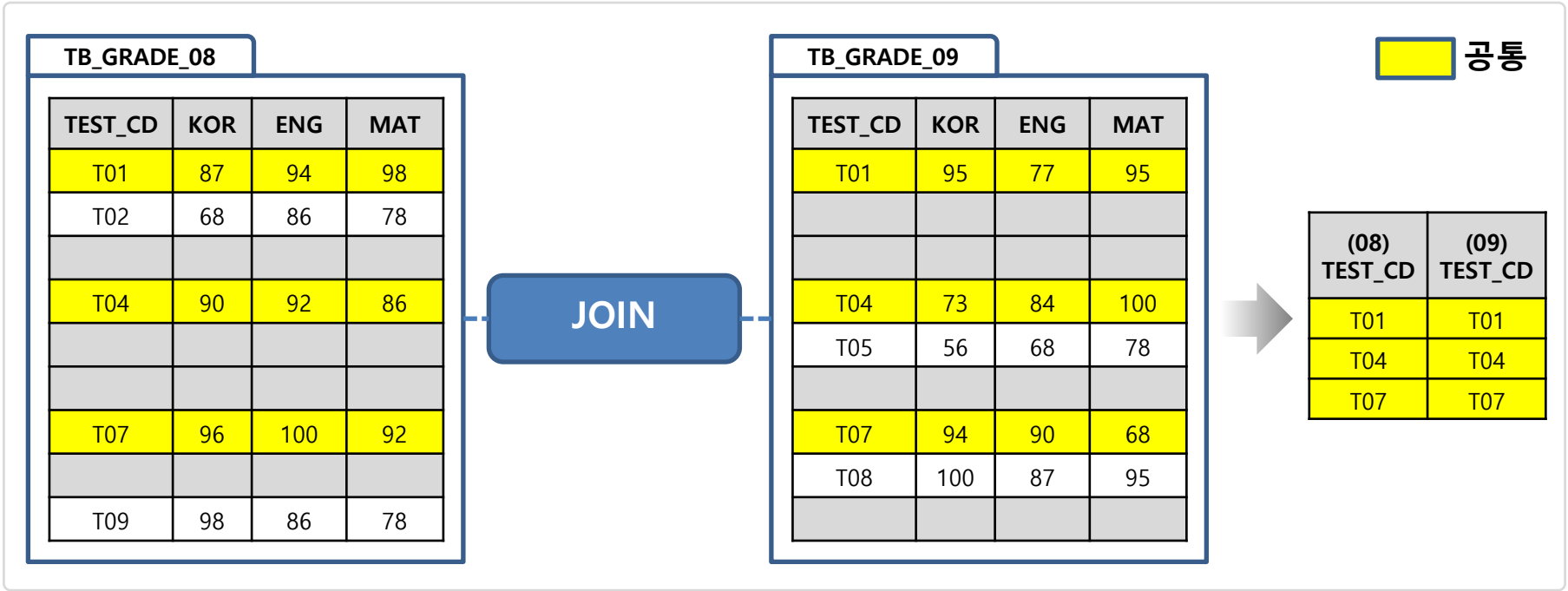
```
SELECT B.CLASS_NM,  
       A.STUDENT_NO,  
       A.STUDENT_NM,  
       A.KOR,  
       A.ENG,  
       A.MAT,  
       A.TOT,  
       A.AVG  
FROM   TB_GRADE A,  
       TB_CLASS_INFO B  
WHERE  A.CLASS_CD = B.CLASS_CD;
```

결과

CLASS_NM	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
종합입시반	1	강원진	87	94	98	0	0
종합입시반	2	나경숙	68	86	78	0	0
단과반	1	박승대	90	92	86	0	0
단과반	2	서유리	96	100	92	0	0
단과반	3	이혜옥	98	86	78	0	0
대학편입반	1	김진숙	95	77	95	0	0
대학편입반	2	김한길	73	84	100	0	0
일반인반	1	남궁소망	56	68	78	0	0
일반인반	2	한찬희	94	90	68	0	0
일반인반	3	이마름	100	87	95	0	0



:: TB_GRADE_08 테이블과 TB_GRADE_09 테이블을 다양한 방법으로 연관 검색하여 결과를 검색



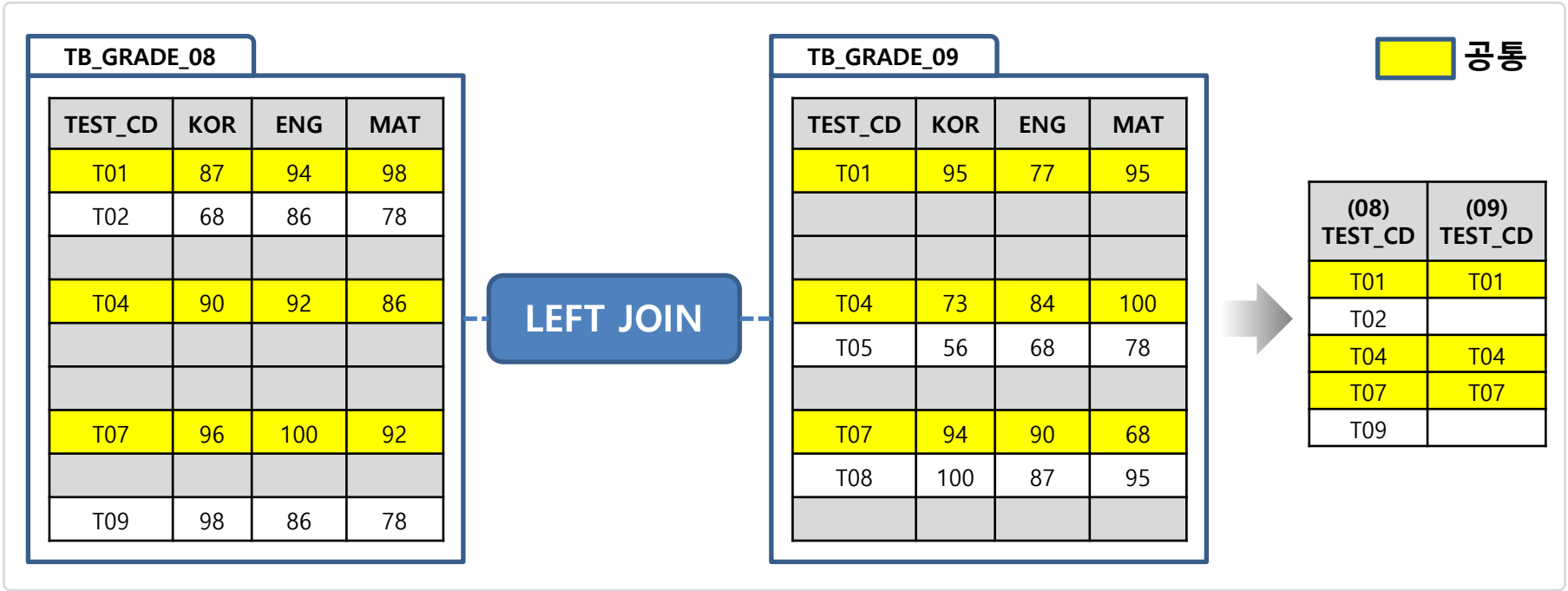
:: TB_GRADE_08 테이블과 TB_GRADE_09 테이블을 JOIN으로 연관 검색

```
SELECT A.TEST_CD AS "08_TEST_CD",  
       A.KOR AS "08_KOR",  
       A.ENG AS "08_ENG",  
       A.MAT AS "08_MAT",  
       B.TEST_CD AS "09_TEST_CD",  
       B.KOR AS "09_KOR",  
       B.ENG AS "09_ENG",  
       B.MAT AS "09_MAT"  
FROM   TB_GRADE_08 A  
JOIN   TB_GRADE_09 B  
ON     A.TEST_CD = B.TEST_CD;
```

JOIN

결과

08_TEST_CD	08_KOR	08_ENG	08_MAT	09_TEST_CD	09_KOR	09_ENG	09_MAT
T01	87	94	98	T01	95	77	95
T04	90	92	86	T04	73	84	100
T07	96	100	92	T07	94	90	68



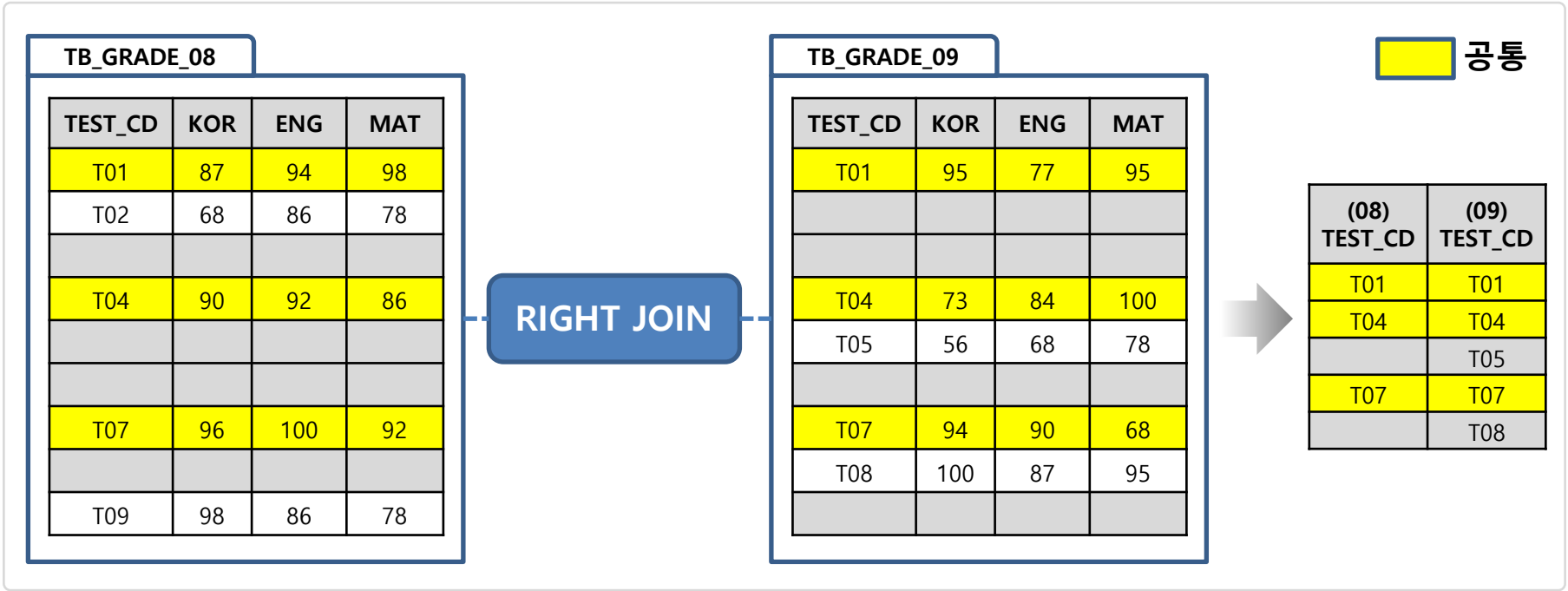
:: TB_GRADE_08 테이블과 TB_GRADE_09 테이블을 LEFT JOIN으로 연관 검색

```
SELECT A.TEST_CD AS "08_TEST_CD",
       A.KOR AS "08_KOR",
       A.ENG AS "08_ENG",
       A.MAT AS "08_MAT",
       B.TEST_CD AS "09_TEST_CD",
       B.KOR AS "09_KOR",
       B.ENG AS "09_ENG",
       B.MAT AS "09_MAT"
FROM   TB_GRADE_08 A
LEFT JOIN TB_GRADE_09 B
ON     A.TEST_CD = B.TEST_CD;
```

```
SELECT A.TEST_CD AS "08_TEST_CD",
       A.KOR AS "08_KOR",
       A.ENG AS "08_ENG",
       A.MAT AS "08_MAT",
       B.TEST_CD AS "09_TEST_CD",
       B.KOR AS "09_KOR",
       B.ENG AS "09_ENG",
       B.MAT AS "09_MAT"
FROM   TB_GRADE_08 A,
       TB_GRADE_09 B
WHERE  A.TEST_CD = B.TEST_CD(+);
```

결과

08_TEST_CD	08_KOR	08_ENG	08_MAT	09_TEST_CD	09_KOR	09_ENG	09_MAT
T01	87	94	98	T01	95	77	95
T02	68	86	78				
T04	90	92	86	T04	73	84	100
T07	96	100	92	T07	94	90	68
T09	98	86	78				



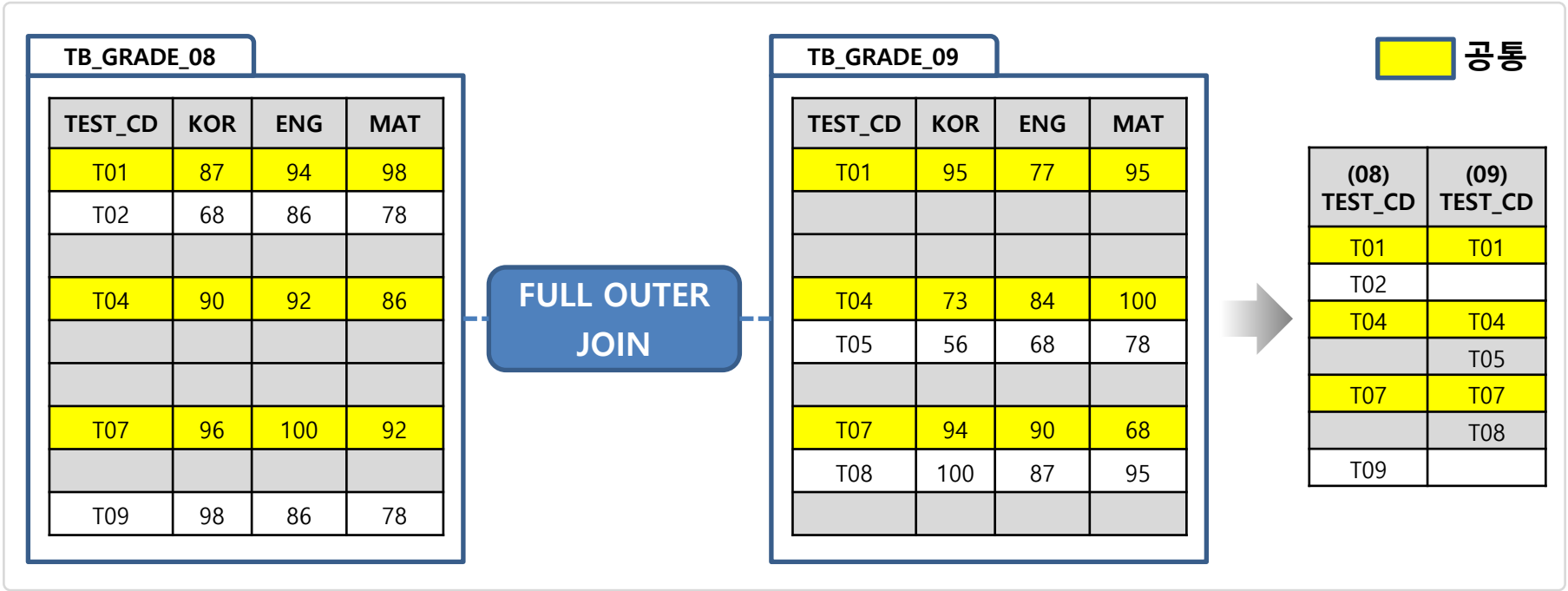
:: TB_GRADE_08 테이블과 TB_GRADE_09 테이블을 RIGHT JOIN으로 연관 검색

```
SELECT A.TEST_CD AS "08_TEST_CD",
       A.KOR AS "08_KOR",
       A.ENG AS "08_ENG",
       A.MAT AS "08_MAT",
       B.TEST_CD AS "09_TEST_CD",
       B.KOR AS "09_KOR",
       B.ENG AS "09_ENG",
       B.MAT AS "09_MAT"
FROM   TB_GRADE_08 A
RIGHT JOIN TB_GRADE_09 B
ON     A.TEST_CD = B.TEST_CD;
```

```
SELECT A.TEST_CD AS "08_TEST_CD",
       A.KOR AS "08_KOR",
       A.ENG AS "08_ENG",
       A.MAT AS "08_MAT",
       B.TEST_CD AS "09_TEST_CD",
       B.KOR AS "09_KOR",
       B.ENG AS "09_ENG",
       B.MAT AS "09_MAT"
FROM   TB_GRADE_08 A,
       TB_GRADE_09 B
WHERE  A.TEST_CD(+) = B.TEST_CD;
```

결과

08_TEST_CD	08_KOR	08_ENG	08_MAT	09_TEST_CD	09_KOR	09_ENG	09_MAT
T01	87	94	98	T01	95	77	95
T04	90	92	86	T04	73	84	100
				T05	56	68	78
T07	96	100	92	T07	94	90	68
				T08	100	87	95



:: TB_GRADE_08 테이블과 TB_GRADE_09 테이블을 FULL OUTER JOIN으로 연관 검색

```
SELECT A.TEST_CD AS "08_TEST_CD",  
       A.KOR AS "08_KOR",  
       A.ENG AS "08_ENG",  
       A.MAT AS "08_MAT",  
       B.TEST_CD AS "09_TEST_CD",  
       B.KOR AS "09_KOR",  
       B.ENG AS "09_ENG",  
       B.MAT AS "09_MAT"  
FROM   TB_GRADE_08 A  
FULL   OUTER JOIN TB_GRADE_09 B  
ON      A.TEST_CD = B.TEST_CD;
```

FULL OUTER JOIN

결과

08_TEST_CD	08_KOR	08_ENG	08_MAT	09_TEST_CD	09_KOR	09_ENG	09_MAT
T01	87	94	98	T01	95	77	95
T04	90	92	86	T04	73	84	100
				T05	56	68	78
T07	96	100	92	T07	94	90	68
				T08	100	87	95
T09	98	86	78				
T02	68	86	78				

TB_CUSTOMER				TB_POINT				
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	RE
2017042	강원진	M	19810603	2019001	1	등록 포인트 적립	500	2019년
2017053	나경숙	W	19891225	2019001	2	믹서기 구매 포인트 적립	4600	2019년
2017108	박승대	M	19711430	2019001	3	드라이기 구매 포인트 적립	7820	2019년
2018087	서유리	W	19810925	2019069	1	이벤트 포인트 적립	8900	2019년
2018254	이혜옥	W	19839012	2019069	2	면도기 구매 포인트 적립	3200	2019년
2019001	김진숙	W	20010426	2019069	3	전기밥솥 구매 포인트 적립	3220	2019년
2019069	김한길	M	1992 315	2019095	1	등록 포인트 적립	500	2019년
2019095	남궁소망	M	19620728	2019095	2	이벤트 포인트 적립	390	2019년
2019167	한찬희	M	19711106	2019167	1	드라이기 구매 포인트 적립	3200	2019년
2019281	이아름	W	19940513	2019167	2	전기밥솥 구매 포인트 적립	3600	2019년

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	SEQ_NO	POINT_MEMO	POINT
2019095	남궁소망	M	1	등록 포인트 적립	500
2019095	남궁소망	M	2	이벤트 포인트 적립	390

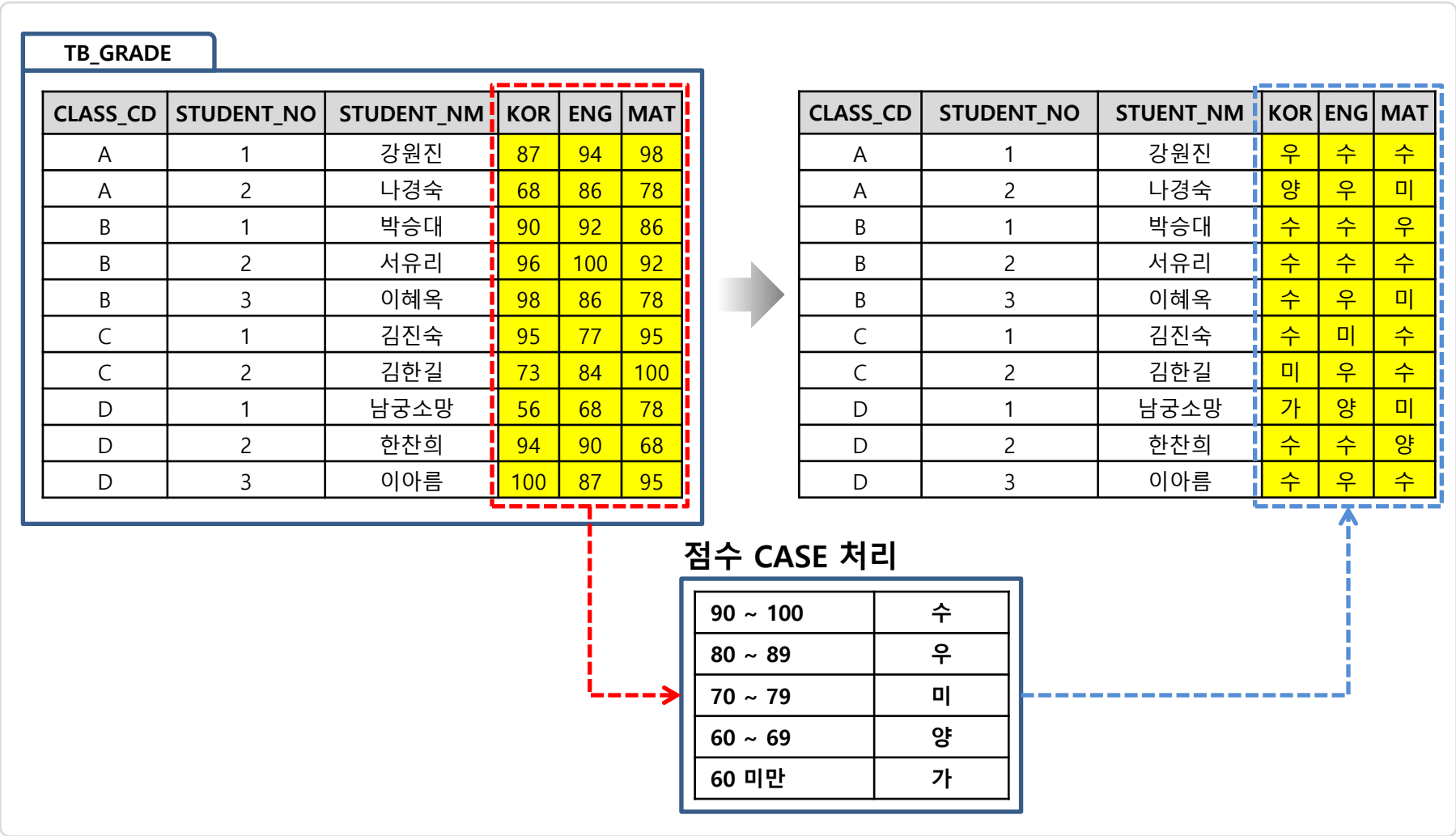
∴ 고객 테이블의 고객코드가 '2019095'인 데이터를 고객포인트 테이블과 연관 검색하여 고객 테이블에서는 고객코드, 고객명, 성별을 검색한 후 고객포인트 테이블에서는 일련번호, 포인트내용, 포인트를 검색한다.

CASE

KEYWORD : 경우에 따른 처리 및 값 설정

구문 #1	<pre>CASE 조건대상 WHEN 비교값1 THEN 결과1 WHEN 비교값2 THEN 결과2 : WHEN 비교값n THEN 결과n ELSE 결과가 없을 때 END</pre>
구문 #2	<pre>CASE WHEN 조건식1 THEN 결과1 WHEN 조건식2 THEN 결과2 : WHEN 조건식n THEN 결과n ELSE 결과가 없을 때 END</pre>

사 용 예	<pre>CASE DAY_CD WHEN '1' THEN '일요일' WHEN '2' THEN '평일' ELSE 'X' END</pre>
-------	--

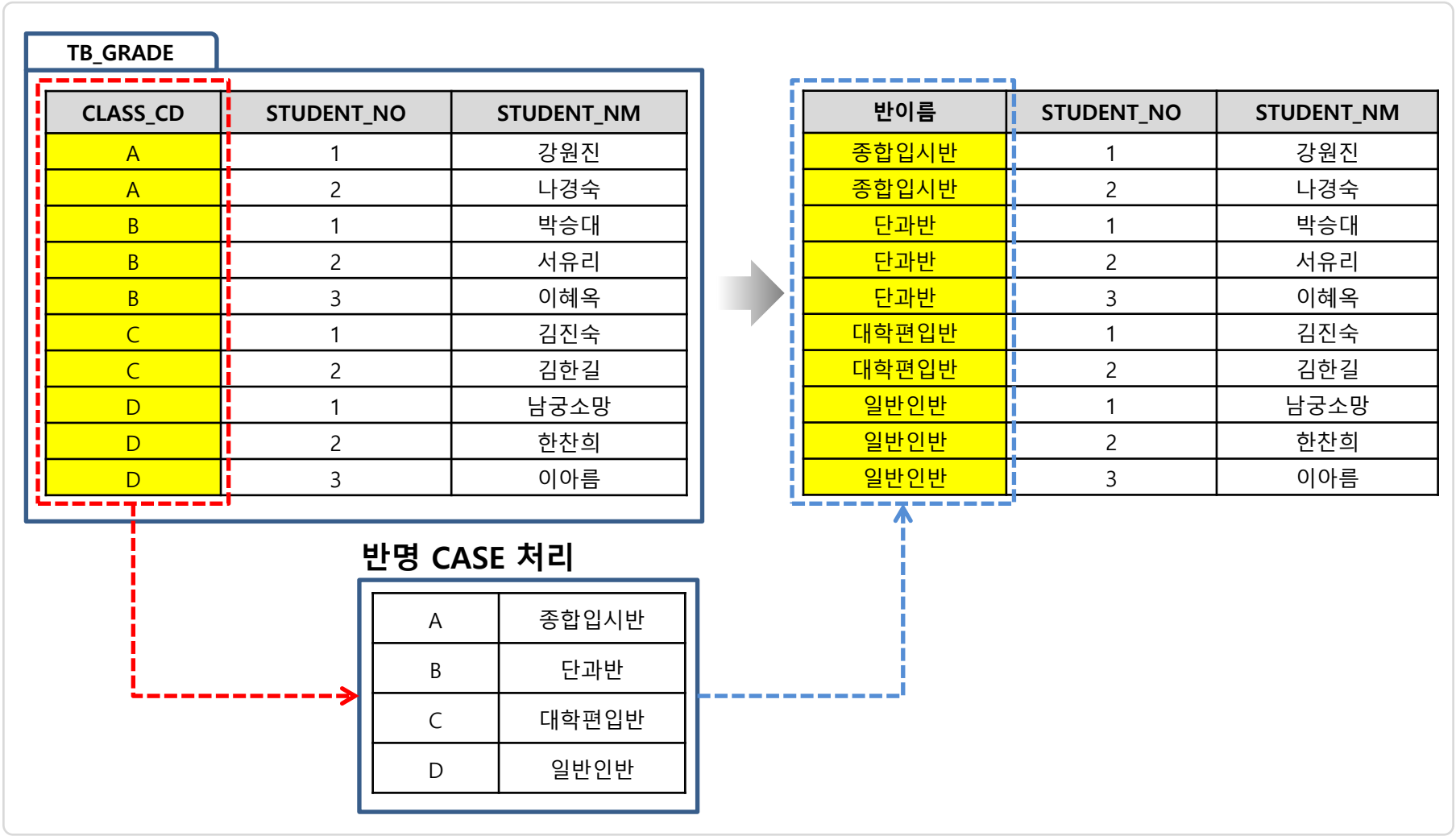


:: TB_GRADE 테이블의 KOR, ENG, MAT 점수에 따라 '수', '우', '미', '양', '가'로 변경

```
SELECT CLASS_CD,  
       STUDENT_NO,  
       STUDENT_NM,  
       CASE WHEN KOR >= 90 AND KOR <= 100 THEN '수'  
            WHEN KOR >= 80 AND KOR <= 89 THEN '우'  
            WHEN KOR >= 70 AND KOR <= 79 THEN '미'  
            WHEN KOR >= 60 AND KOR <= 69 THEN '양'  
            WHEN KOR < 60 THEN '가'  
       END AS KOR,  
       CASE WHEN ENG BETWEEN 90 AND 100 THEN '수'  
            WHEN ENG BETWEEN 80 AND 89 THEN '우'  
            WHEN ENG BETWEEN 70 AND 79 THEN '미'  
            WHEN ENG BETWEEN 60 AND 69 THEN '양'  
            ELSE '가'  
       END AS ENG,  
       CASE WHEN MAT IN (90,91,92,93,94,95,96,97,98,99,100) THEN '수'  
            WHEN MAT IN (80,81,82,83,84,85,86,87,88,89) THEN '우'  
            WHEN MAT IN (70,71,72,73,74,75,76,77,78,79) THEN '미'  
            WHEN MAT IN (60,61,62,63,64,65,66,67,68,69) THEN '양'  
            ELSE '가'  
       END AS MAT,  
       TOT,  
       AVG  
FROM   TB_GRADE;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	무	수	수	0	0
A	2	나경숙	양	무	미	0	0
B	1	박승대	수	수	무	0	0
B	2	서유리	수	수	수	0	0
B	3	이혜옥	수	무	미	0	0
C	1	김진숙	수	미	수	0	0
C	2	김한길	미	무	수	0	0
D	1	남궁소망	가	양	미	0	0
D	2	한찬희	수	수	양	0	0
D	3	이아름	수	무	수	0	0



:: TB_GRADE 테이블의 CLASS_CD가 'A' 이면 '종합입시반', 'B' 이면 '단과반', 'C' 이면 '대학편입반', 'D' 이면 '일반인반' 으로 변경

```
SELECT CASE CLASS_CD WHEN 'A' THEN '종합입시반'
                WHEN 'B' THEN '단과반'
                WHEN 'C' THEN '대학편입반'
                WHEN 'D' THEN '일반인반'
        END AS CLASS_NM,
        STUDENT_NO,
        STUDENT_NM
FROM    TB_GRADE;
```

결과

CLASS_NM	STUDENT_NO	STUDENT_NM
종합입시반	1	강원진
종합입시반	2	나경숙
단과반	1	박승대
단과반	2	서유리
단과반	3	이혜옥
대학편입반	1	김진숙
대학편입반	2	김한길
일반인반	1	남궁소망
일반인반	2	한찬희
일반인반	3	이아름

TB_CUSTOMER			TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	TOTAL_POINT	CUSTOMER_CD	CUSTOMER_NM	TOTAL_POINT	고객 등급
2017042	강원진	280300	2017042	강원진	280300	VIP
2017053	나경숙	4500	2017053	나경숙	4500	실버
2017108	박승대	23450	2017108	박승대	23450	골드
2018087	서유리	18700	2018087	서유리	18700	실버
2018254	이혜옥	570	2018254	이혜옥	570	일반
2019001	김진숙	12820	2019001	김진숙	12820	실버
2019069	김한길	15320	2019069	김한길	15320	실버
2019095	남궁소망	890	2019095	남궁소망	890	일반
2019167	한찬희	6800	2019167	한찬희	6800	실버
2019281	이마름	35600	2019281	이마름	35600	골드

∴ 고객 테이블에서 누적포인트가 1,000에서 20,000미만이면 “실버”, 20,000에서 50,000미만이면 “골드”, 50,000이상이면 “VIP” 등급을 보이고 위 조건에 해당 없으면 “일반” 등급을 보인다.

ROWNUM

KEYWORD : 행 번호 취득

구 문	ROWNUM
결과값	수치형

ROWNUM	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	조회
1	2017042	강원진	M	○
2	2017053	나경숙	W	
3	2017108	박승대	M	
4	2018087	서유리	W	○
5	2018254	이혜옥	W	○
6	2019001	김진숙	W	
7	2019069	김한길	M	
8	2019095	남궁소망	M	○
9	2019167	한찬희	M	○
10	2019281	이아름	W	○



ROWNUM	CUSTOMER_CD	CUSTOMER_NM	MW_FLG
1	2017042	강원진	M
2	2018087	서유리	W
3	2018254	이혜옥	W
4	2019095	남궁소망	M
5	2019167	한찬희	M
6	2019281	이아름	W

:: 일부 레코드를 조회 후 ROWNUM의 변경

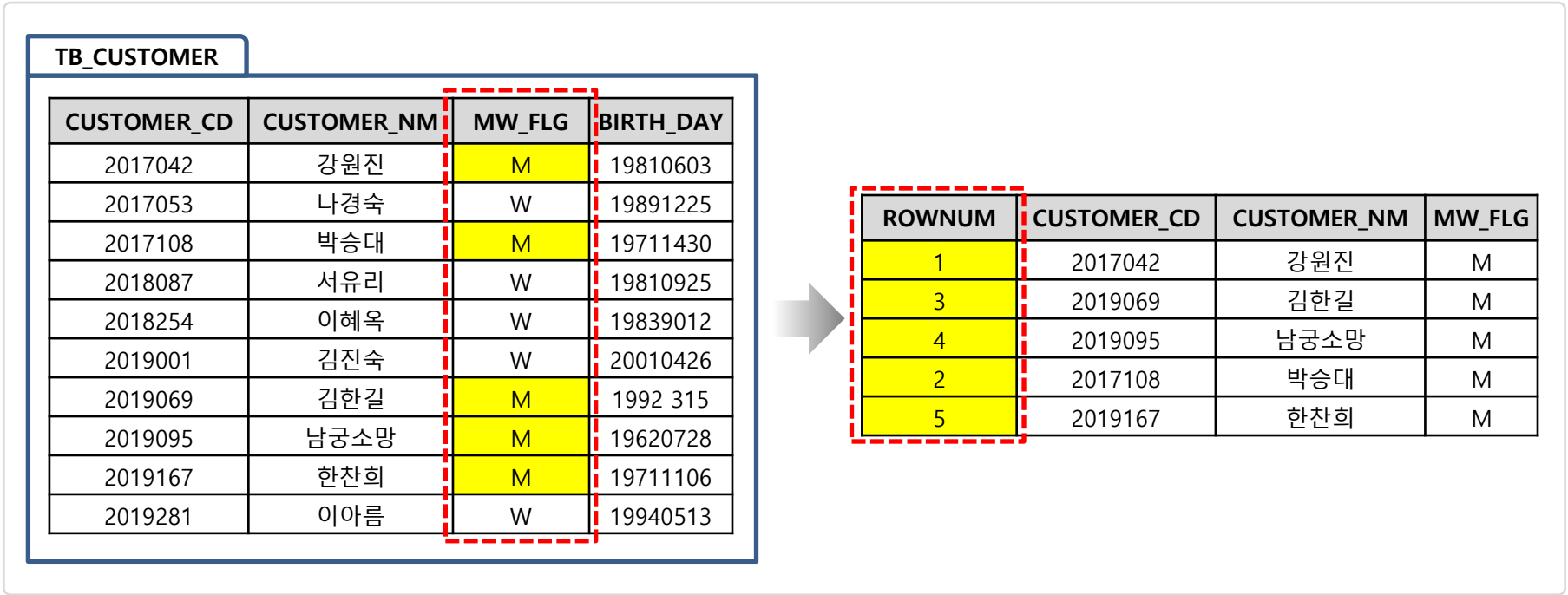
TB_CUSTOMER							
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	ROWNUM	CUSTOMER_CD	CUSTOMER_NM	MW_FLG
2017042	강원진	M	19810603	1	2017042	강원진	M
2017053	나경숙	W	19891225	2	2017053	나경숙	W
2017108	박승대	M	19711430	3	2017108	박승대	M
2018087	서유리	W	19810925	4	2018087	서유리	W
2018254	이혜옥	W	19839012	5	2018254	이혜옥	W
2019001	김진숙	W	20010426	6	2019001	김진숙	W
2019069	김한길	M	1992 315	7	2019069	김한길	M
2019095	남궁소망	M	19620728	8	2019095	남궁소망	M
2019167	한찬희	M	19711106	9	2019167	한찬희	M
2019281	이아름	W	19940513	10	2019281	이아름	W

:: TB_CUSTOMER 테이블의 전체 데이터의 행번호를 검색

```
SELECT ROWNUM,  
       CUSTOMER_CD,  
       CUSTOMER_NM,  
       MW_FLG,  
       BIRTH_DAY,  
       PHONE_NUMBER  
FROM   TB_CUSTOMER;
```

결과

ROWNUM	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER
1	2017042	강원진	M	19810603	002-8202-8790
2	2017053	나경숙	W	19891225	002-4509-0043
3	2017108	박승대	M	19711430	
4	2018087	서유리	W	19810925	003-1265-8372
5	2018254	미혜옥	W	19839012	003_1287_9734
6	2019001	김진숙	W	20010426	002-9842-0074
7	2019069	김한길	M	1992 315	
8	2019095	남궁소망	M	19620728	003-6273-8539
9	2019167	한찬희	M	19711106	002=1202=5563
10	2019281	이아름	W	19940513	003-2620-0723



:: TB_CUSTOMER 테이블에서 MW_FLG가 'M'인 데이터의 ROWNUM을 검색하고
CUSTOMER_NM으로 정렬

```
SELECT ROWNUM,  
       CUSTOMER_CD,  
       CUSTOMER_NM,  
       MW_FLG,  
       BIRTH_DAY,  
       PHONE_NUMBER  
FROM   TB_CUSTOMER  
WHERE  MW_FLG = 'M'  
ORDER  BY CUSTOMER_NM;
```

결과

ROWNUM	CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER
1	2017042	강원진	M	19810603	002-8202-8790
3	2019069	김한길	M	1992 315	
4	2019095	남궁소망	M	19620728	003-6273-8539
2	2017108	박승대	M	19711430	
5	2019167	한찬희	M	19711106	002-1202-5563

TB_GRADE

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0



ROWNUM	CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
1	A	1	강원진	87	94	98	0	0
2	A	2	나경숙	68	86	78	0	0
3	C	1	김진숙	95	77	95	0	0
4	C	2	김한길	73	84	100	0	0

:: 성적 테이블에서 반코드가 'A' 또는 'C'반의 학생을 대상으로 모든 필드와 순차적인 행번호를 검색한다.

NULL

KEYWORD : 값이 설정되어 있지 않음

구 문	SELECT 검색필드 FROM 테이블 WHERE 필드 IS [NOT] NULL;
사용 예	WHERE EMAIL IS NULL WHERE EMAIL NOT IS NULL

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	PHONE_NUMBER	EMAIL
2017042	M	002-8202-8790	wjgang@navi...
2017053	W	002-4509-0043	ksna@boram...
2017108	M		sdpark@haso...
2018087	W	003-1265-8372	urseo@epnt.co.kr
2018254	W	003_1287_9734	hylee@hansoft...
2019001	W	002-9842-0074	jskim@dreami...
2019069	M		hkkim@ssoya.com
2019095	M	003-6273-8539	
2019167	M	002=1202=5563	chhan@ecom.c..
2019281	W	003-2620-0723	aulee@hoki...

CUSTOMER_CD	MW_FLG	PHONE_NUMBER	EMAIL
2017108	M		sdpark@haso...
2019069	M		hkkim@ssoya.com
2019095	M	003-6273-8539	

∴ TB_CUSTOMER 테이블에서 PHONE_NUMBER가 NULL 이거나 EMAIL이 NULL인 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  PHONE_NUMBER IS NULL
OR     EMAIL IS NULL;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558

TB_CUSTOMER			
CUSTOMER_CD	MW_FLG	PHONE_NUMBER	EMAIL
2017042	M	002-8202-8790	wjgang@navi...
2017053	W	002-4509-0043	ksna@boram...
2017108	M		sdpark@haso...
2018087	W	003-1265-8372	urseo@epnt.co.kr
2018254	W	003_1287_9734	hylee@hansoft...
2019001	W	002-9842-0074	jskim@dreami...
2019069	M		hkkim@ssoya.com
2019095	M	003-6273-8539	
2019167	M	002=1202=5563	chhan@ecom.c..
2019281	W	003-2620-0723	aulee@hoki...

CUSTOMER_CD	MW_FLG	PHONE_NUMBER	EMAIL
2017053	W	002-4509-0043	ksna@boram...
2018087	W	003-1265-8372	urseo@epnt.co.kr
2018254	W	003_1287_9734	hylee@hansoft...
2019001	W	002-9842-0074	jskim@dreami...
2019281	W	003-2620-0723	aulee@hoki...
2019095	M	003-6273-8539	

:: TB_CUSTOMER 테이블에서 MW_FLG가 'M' 이면서 EMAIL이 NULL 이거나 MW_FLG가 'W'이면서 EMAIL이 NULL이 아닌 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  (MW_FLG = 'M'
AND    EMAIL IS NULL)
OR     (MW_FLG = 'W'
AND    EMAIL IS NOT NULL)
ORDER BY MW_FLG DESC;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER	⚡ EMAIL	⚡ TOTAL_POINT	⚡ REG_DTTM
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2019281	이마름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	PHONE_NUMBER
2017042	강원진	M	002-8202-8790
2017053	나경숙	W	002-4509-0043
2017108	박승대	M	
2018087	서유리	W	003-1265-8372
2018254	이혜옥	W	003_1287_9734
2019001	김진숙	W	002-9842-0074
2019069	김한길	M	
2019095	남궁소망	M	003-6273-8539
2019167	한찬희	M	002=1202=5563
2019281	이아름	W	003-2620-0723

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	PHONE_NUMBER
2017042	강원진	M	002-8202-8790
2018254	이혜옥	W	003_1287_9734
2019095	남궁소망	M	003-6273-8539
2019167	한찬희	M	002=1202=5563

∴ TB_CUSTOMER 테이블에서 CUSTOMER_NM이 '강원진', '박승대', '이혜옥', '남궁소망', '한찬희'이면서 PHONE_NUMBER가 NULL이 아닌 데이터를 검색

```
SELECT *
FROM   TB_CUSTOMER
WHERE  CUSTOMER_NM IN ( '강원진', '박승대', '이혜옥', '남궁소망', '한찬희' )
AND    PHONE_NUMBER IS NOT NULL;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992_315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308



CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308

∴ 고객 테이블에서 고객코드가 '2018' 또는 '2019'로 시작하고, 생일이 1990년 또는 2000년
대인 고객 중 전화번호가 설정되어 있는 데이터를 검색한다.


DUAL

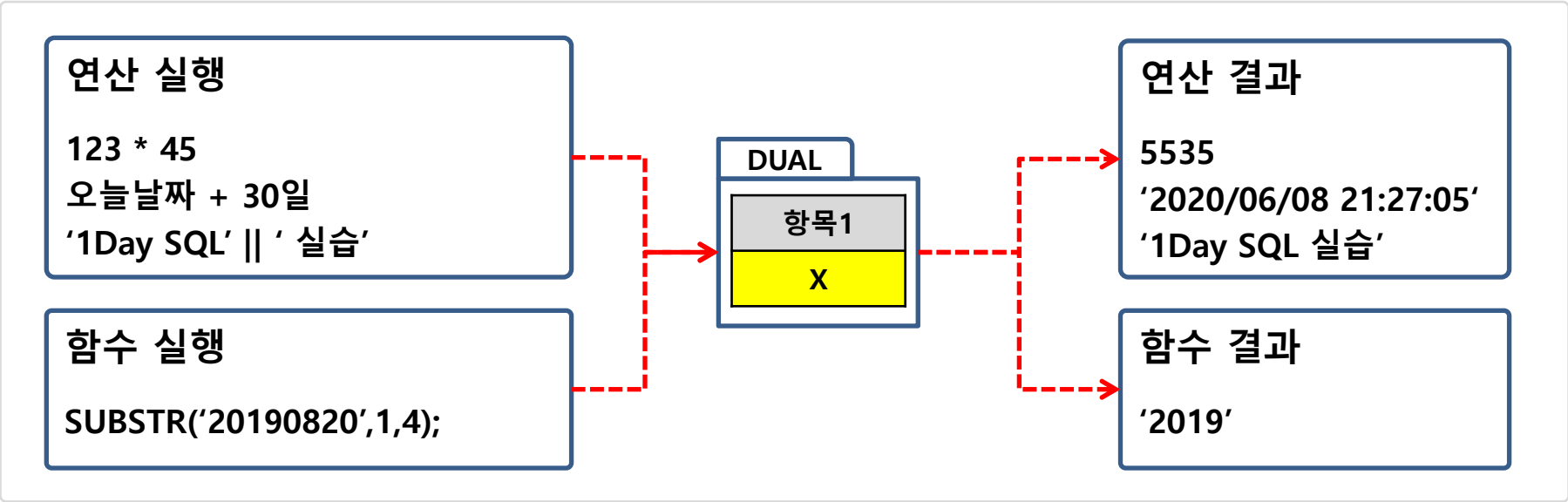
KEYWORD : 연산 값 및 기능 검색용 1행 테이블

구문 #1	SELECT 연산, 함수기능, 출력값 등 FROM DUAL;
구문 #2	SELECT 필드, 연산, 함수기능, 출력값 등 FROM DUAL CONNECT BY LEVEL <= n;
사용 예	SELECT 20*3 FROM DUAL; SELECT SUBSTR('20190820',1,4) FROM DUAL; SELECT ROWNUM FROM DUAL CONNECT BY LEVEL <= 3;


```
SELECT *  
FROM   DUAL ;
```

결과

 DUMMY
X



:: DUAL 테이블을 사용한 출력 방법

```
SELECT 123 * 45,  
       SYSDATE + 30,  
       '1Day SQL' || ' 연습',  
       SUBSTR( '20190820' ,1,4)  
FROM    DUAL;
```

결과

123*45	SYSDATE+30	'1DaySQL' '연습'	SUBSTR('20190820',1,4)
5535	2019/12/18 11:34:35	1Day SQL 연습	2019

```
SELECT '2019-' || ROWNUM
FROM   DUAL
CONNECT BY LEVEL <= 5;
```

결과

↕ '2019-' ROWNUM
2019-1
2019-2
2019-3
2019-4
2019-5

< 출력결과 >

300/60	오늘 날짜	30일 후 날짜
5	2020-08-14	2020-09-13

:: DUAL 테이블로 300÷60과 오늘 날짜와 30일 이후의 날짜를 'YYYY-MM-DD' 형식으로 보인다. 참고로 'YYYY-MM-DD' 형식은 CHAR() 함수에 현재날짜와 'YYYY-MM-DD' 형식을 옵션으로 입력해 보인다.

< 출력결과 >

DAY_CD	DAY_NM
1	일요일
2	월요일
3	화요일
4	수요일
5	목요일
6	금요일
7	토요일

:: DUAL 테이블로 요일코드와 요일명을 생성한다. 요일코드는 수치를 문자열로 바꾸는 TO_CHAR() 함수를 사용한다.

TO_CHAR() 함수 형식 : TO_CHAR(수치) → 문자열

UPDATE

KEYWORD : 테이블 항목 값 변경

구 문	UPDATE 테이블 SET 필드1 = 값1, 필드2 = 값2 ..., 필드n = 값n WHERE 조건식;
사용 예	UPDATE TB_GRADE SET KOR = 89, ENG = 92 WHERE STUDENT_NO = 2045;

```
SELECT *  
FROM   TB_GRADE;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	0	0
A	2	나경숙	68	86	78	0	0
B	1	박승대	90	92	86	0	0
B	2	서유리	96	100	92	0	0
B	3	이혜옥	98	86	78	0	0
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0

:: TB_GRADE 테이블에서 CLASS_CD가 'A', 'B'인 데이터의 KOR, ENG, MAT 점수의 합계, 평균을 계산하여 각각 TOT, AVG 항목을 수정

```
UPDATE TB_GRADE  
SET     TOT = KOR + ENG + MAT,  
        AVG = ROUND((KOR + ENG + MAT) / 3, 1)  
WHERE  CLASS_CD IN ('A', 'B');
```

결과

5개 행 이 (가) 업데이트되었습니다.

```
SELECT *  
FROM   TB_GRADE;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	0	0
C	2	김한길	73	84	100	0	0
D	1	남궁소망	56	68	78	0	0
D	2	한찬희	94	90	68	0	0
D	3	이아름	100	87	95	0	0

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이아름	100	87	95	282	94

:: TB_GRADE 테이블에서 KOR, ENG, MAT 점수의 합계, 평균을 계산하여 각각 TOT, AVG 항목을 수정

```
UPDATE TB_GRADE
SET    TOT = KOR + ENG + MAT,
      AVG = ROUND((KOR + ENG + MAT) / 3,1);
```

결과

10개 행 이 (가) 업데이트되었습니다.

```
SELECT *  
FROM   TB_GRADE;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이아름	100	87	95	282	94

```
COMMIT;
```

결과

커밋 완료.

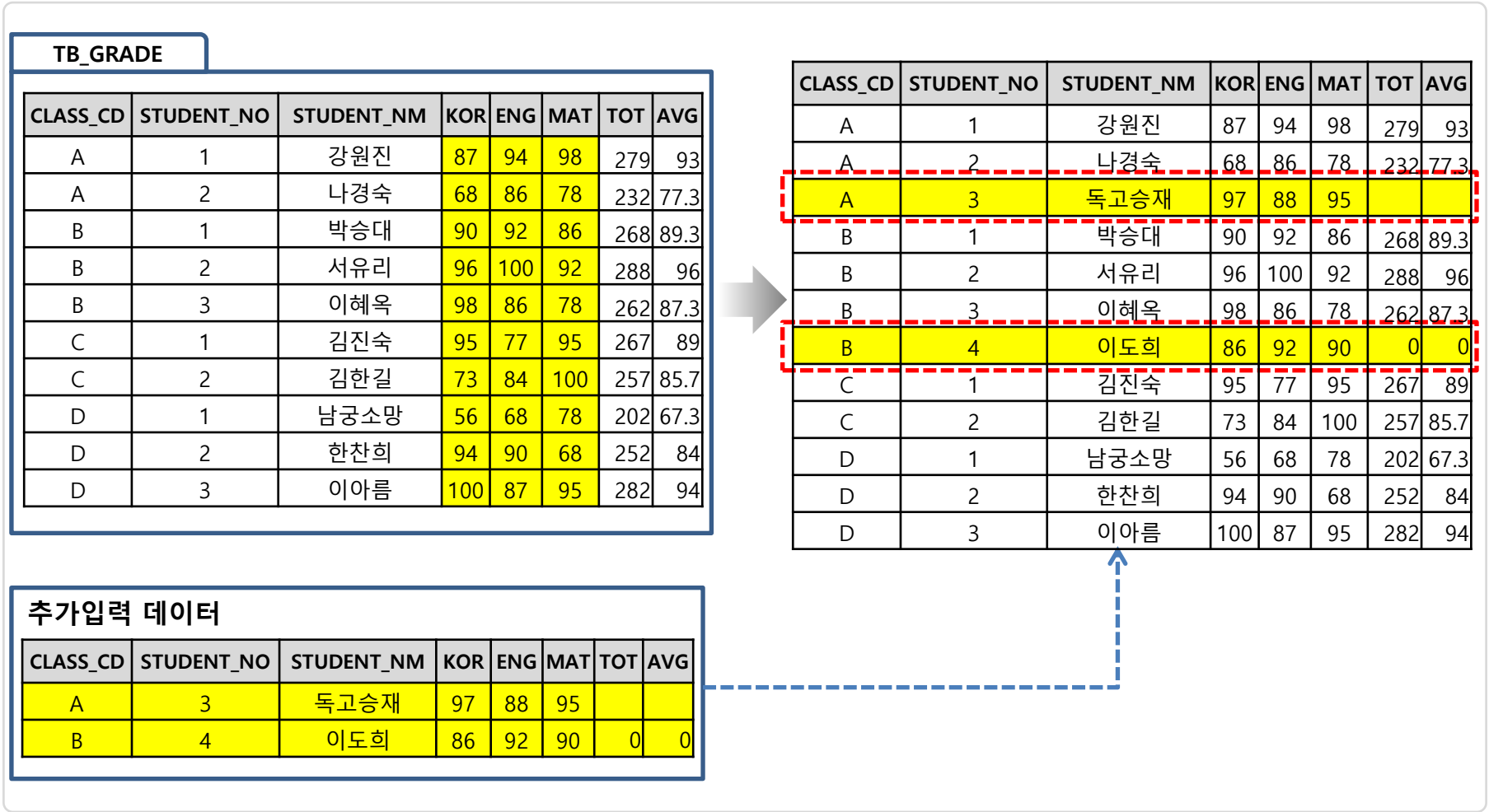
INSERT

KEYWORD : 테이블에 데이터 입력

구문 #1	INSERT INTO 테이블 VALUES (값1, 값2 ..., 값n);
구문 #2	INSERT INTO 테이블 (필드1, 필드2 ..., 필드n) VALUES (값1, 값2 ..., 값n);
구문 #3	INSERT INTO 테이블 SELECT 필드1, 필드2 ..., 필드n FROM 테이블 WHERE 조건식;

구문 #4	<pre>INSERT [ALL FIRST] WHEN 조건식1 THEN INTO 테이블1 WHEN 조건식2 THEN INTO 테이블2 : WHEN 조건식n THEN INTO 테이블n ELSE INTO 테이블x SELECT 필드1, 필드2, 필드3, 필드n FROM 테이블 WHERE 조건식;;</pre>
구문 #5	<pre>INSERT [ALL FIRST] INTO 테이블1 VALUES (입력필드1, 입력필드2, ... 입력필드n) INTO 테이블2 VALUES (입력필드1, 입력필드2, ... 입력필드n) : INTO 테이블n SELECT 필드1, 필드2, 필드3 ... 필드n FROM 테이블 WHERE 조건식;</pre>

사용 예	<div>INSERT INTO 테이블 VALUES (입력값, ...)</div> <div>INSERT INTO 테이블 (필드, ...) VALUES (입력값, ...)</div> <div>INSERT INTO 테이블 SELECT 검색필드1, 검색필드2, ...</div> <div>INSERT [ALL FIRST] WHEN 조건식 THEN INTO 테이블</div>
------	--



:: TB_GRADE 테이블에 A반3번 '독고승재' 학생과 B반4번 '이도희' 학생의 데이터를 입력

```
SELECT *  
FROM   TB_GRADE;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이아름	100	87	95	282	94

```
INSERT INTO TB_GRADE (CLASS_CD, STUDENT_NO, STUDENT_NM, KOR, ENG, MAT)
VALUES ('A',3, '독고승재',97,88,95);
```

결과

1 행 이 (가) 삽입되었습니다.

```
INSERT INTO TB_GRADE VALUES ( 'B' ,4, '이도희' ,86,92,90,0,0);
```

결과

1 행 이 (가) 삽입되었습니다.

```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

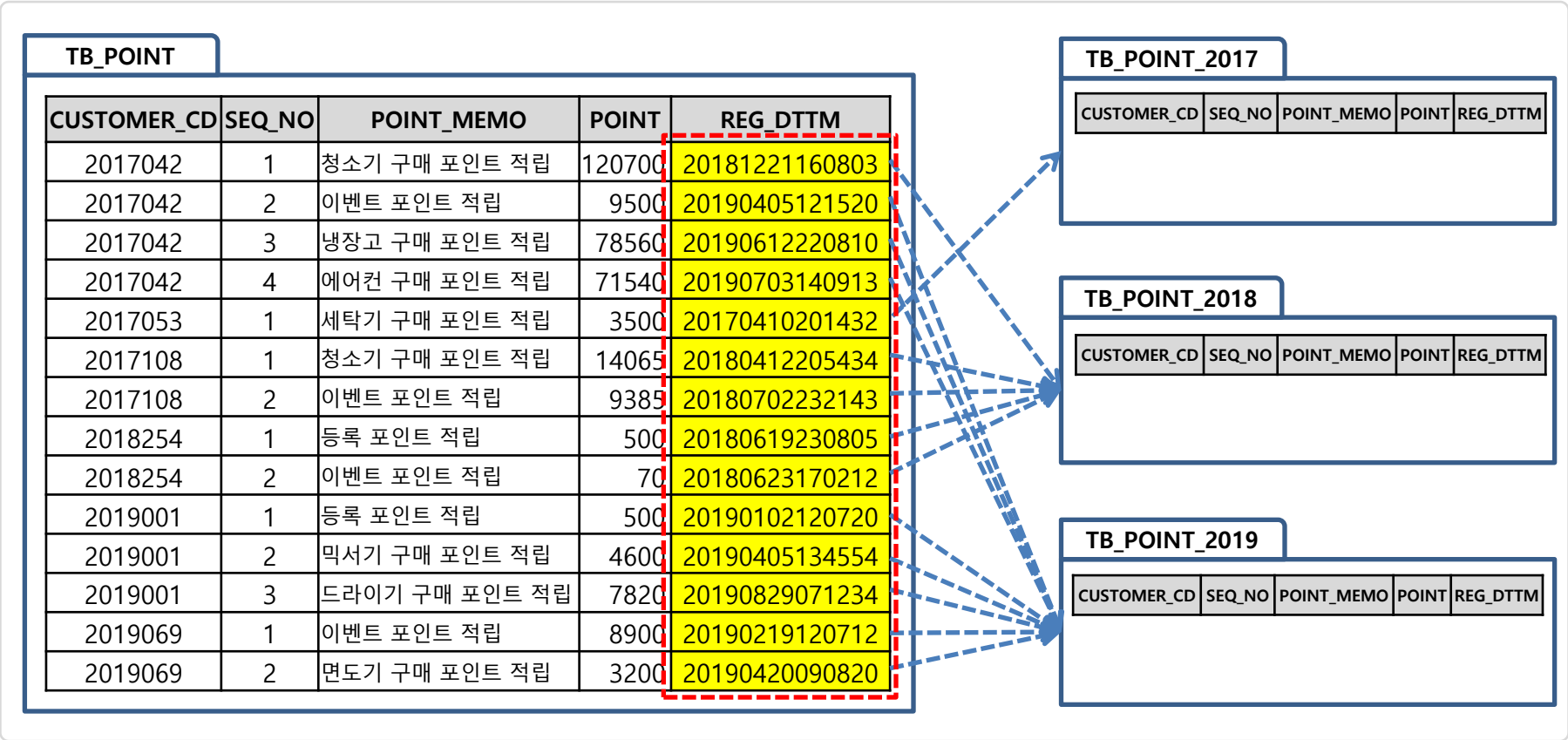
결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
A	3	독고승재	97	88	95		
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이아름	100	87	95	282	94

```
COMMIT;
```

결과

커밋 완료.



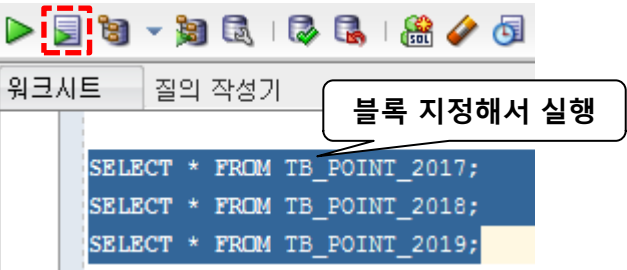
∴ TB_POINT 테이블에서 REG_DTTM의 4자리 연도에 따라 TB_POINT_2017 ~ 2019 테이블에 입력

```
SELECT * FROM TB_POINT_2017;  
SELECT * FROM TB_POINT_2018;  
SELECT * FROM TB_POINT_2019;
```

결과

선택된 행 없음
선택된 행 없음
선택된 행 없음

참고



```
INSERT ALL
  WHEN SUBSTR(REG_DTTM,1,4) = '2017' THEN INTO TB_POINT_2017
  WHEN SUBSTR(REG_DTTM,1,4) = '2018' THEN INTO TB_POINT_2018
  WHEN SUBSTR(REG_DTTM,1,4) = '2019' THEN INTO TB_POINT_2019
  ELSE INTO TB_POINT_2019
SELECT CUSTOMER_CD,
       SEQ_NO,
       POINT_MEMO,
       POINT,
       REG_DTTM
FROM   TB_POINT
WHERE  REG_DTTM >= '20170101000000';
```

결과

27개 행 이 (가) 삽입되었습니다.

```
SELECT * FROM TB_POINT_2017;
SELECT * FROM TB_POINT_2018;
SELECT * FROM TB_POINT_2019;
```

결과

CUSTOMER	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017053	1	세탁기 구매 포인트 적립	3500	20170410201432

} TB_POINT_2017 테이블 내용

CUSTOMER	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017053	2	드라이기 구매 포인트 적립	2000	20181216171040
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956
2018254	1	등록 포인트 적립	500	20180619230805
2018254	2	이벤트 포인트 적립	70	20180623170212

} TB_POINT_2018 테이블 내용

8개 행이 선택되었습니다.

CUSTOMER	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2019001	1	등록 포인트 적립	500	20190102120720
2019001	2	믹서기 구매 포인트 적립	4600	20190405134554
2019001	3	드라이기 구매 포인트 적립	7820	20190829071234
2019069	1	이벤트 포인트 적립	8900	20190219120712
2019069	2	면도기 구매 포인트 적립	3200	20190420090820
2019069	3	전기밥솥 구매 포인트 적립	3220	20190620071230
2019095	1	등록 포인트 적립	500	2019031212456
2019095	2	이벤트 포인트 적립	390	20190510072345

} TB_POINT_2019 테이블 내용

CUSTOMER	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2019167	1	드라이기 구매 포인트 적립	3200	20190612042450
2019167	2	전기밥솥 구매 포인트 적립	3600	20190714133422
2019281	1	등록 포인트 적립	500	20190709201308
2019281	2	청소기 구매 포인트 적립	8950	20190710200921
2019281	3	냉장고 구매 포인트 적립	12200	20190712082334
2019281	4	드라이기 구매 포인트 적립	7600	20190721134421
2019281	5	믹서기 구매 포인트 적립	6250	20190724022430

18개 행이 선택되었습니다.

```
COMMIT;
```

결과

커밋 완료.

MERGE

KEYWORD : 테이블의 내용 삽입 or 변경 처리

구 문	<pre>MERGE INTO 입력/수정할 테이블 USING (테이블 뷰 서브쿼리) ON 조건식 WHEN MATCHED THEN UPDATE SET 필드1 = 값1, 필드2 = 값2, ..., 필드n = 값n WHEN NOT MATCHED THEN INSERT .. VALUES 구문;</pre>
사용 예	<pre>MERGE INTO TB_POINT USING TB_POINT_NEW ON 조건식 WHEN 매칭 THEN UPDATE ... WHEN 언매칭 THEN INSERT ...;</pre>

TB_CUSTOMER			
CUSTOMER_CD	CUSTOMER_NM	PHONE_NUMBER	...
2017042	강원진	002-8202-8790	
2017053	나경숙	002-4509-0043	
2017108	박승대		
2018087	서유리	003-1265-8372	
2018254	이혜옥	003_1287_9734	
2019001	김진숙	002-9842-0074	
2019069	김한길		
2019095	남궁소망	003-6273-8539	
2019167	한찬희	002=1202=5563	
2019281	이아름	003-2620-0723	



CUSTOMER_CD	CUSTOMER_NM	MW_FLG	PHONE_NUMBER	...
2017042	강원진	M	002-8202-8790	
2017053	나경숙	W	002-4509-0043	
2017108	박승대	M	002-2580-9919	
2018087	서유리	W	003-1265-8372	
2018254	이혜옥	W	003_1287_9734	
2019001	김진숙	W	002-9842-0074	
2019069	김한길	M	003-0089-2000	
2019095	남궁소망	M	003-6273-8539	
2019167	한찬희	M	002=1202=5563	
2019281	이아름	W	003-2620-0723	
2019302	전미래	W	002-8864-0232	

TB_ADD_CUSTOMER				
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER
2017108	박승대	M	19711430	002-2580-9919
2019302	전미래	W	19740812	002-8864-0232

동일 데이터면 UPDATE
신규 데이터면 INSERT

:: TB_ADD_CUSTOMER 테이블의 내용을 TB_CUSTOMER 테이블과 비교하여 데이터가 있을 경우 업데이트 하고 데이터가 없을 경우 입력

```
SELECT *  
FROM   TB_CUSTOMER;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430		adpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim\$dreami.org	12820	20190101080518
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002=1202=5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308


```
SELECT *  
FROM   TB_ADD_CUSTOMER;
```

결과

⚡ CUSTOMER_CD	⚡ CUSTOMER_NM	⚡ MW_FLG	⚡ BIRTH_DAY	⚡ PHONE_NUMBER
2017108	박승대	M	19711430	002-2580-9919
2019302	전미래	W	19740812	002-8864-0232

```
MERGE INTO TB_CUSTOMER CU
  USING TB_ADD_CUSTOMER NC
  ON    (CU.CUSTOMER_CD = NC.CUSTOMER_CD)
  WHEN MATCHED THEN
    UPDATE SET CU.CUSTOMER_NM = NC.CUSTOMER_NM,
               CU.MW_FLG = NC.MW_FLG,
               CU.BIRTH_DAY = NC.BIRTH_DAY,
               CU.PHONE_NUMBER = NC.PHONE_NUMBER
  WHEN NOT MATCHED THEN
    INSERT (CU.CUSTOMER_CD, CU.CUSTOMER_NM, CU.MW_FLG, CU.BIRTH_DAY,
           CU.PHONE_NUMBER, CU.EMAIL, CU.TOTAL_POINT, CU.REG_DTTM)
    VALUES (NC.CUSTOMER_CD, NC.CUSTOMER_NM, NC.MW_FLG, NC.BIRTH_DAY,
            NC.PHONE_NUMBER, ' ', 0, TO_CHAR(SYSDATE, 'YYYYMMDDHHMISS'));
```

결과

2개 행 이 (가) 병합되었습니다.

```
SELECT *  
FROM   TB_CUSTOMER;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432
2017053	나경숙	W	19891225	002-4509-0043	ksna#boram.co.kr	4500	20170210180930
2017108	박승대	M	19711430	002-2580-9919	sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019001	김진숙	W	20010426	002-9842-0074	jskim@dreami.org	12820	20190101080518
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558
2019167	한찬희	M	19711106	002-1202-5563	chhan@ecom.co.kr	6800	20190508155600
2019281	이아름	W	19940513	003-2620-0723	aulee@hoki.com	35600	20190709201308
2019302	전미래	W	19740812	002-8864-0232		0	20191118121320

```
COMMIT;
```

결과

커밋 완료.

DELETE

KEYWORD : 테이블의 데이터 삭제

구 문	DELETE FROM 테이블 WHERE 조건식;
사용 예	DELETE FROM TB_POINT WHERE CUSTOMER_CD = '190013'

```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
A	3	독고승재	97	88	95		
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이마름	100	87	95	282	94

TB_GRADE

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	2	나경숙	68	86	78	232	77.3
A	3	독고승재	97	88	95		
B	1	박승대	90	92	86	268	89.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84
D	3	이아름	100	87	95	282	94

삭제

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95		
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

:: TB_GRADE 테이블에서 STUDENT_NM이 '나경숙', '박승대', '이아름' 데이터를 삭제

```
DELETE
FROM    TB_GRADE
WHERE   STUDENT_NM IN ('나경숙','박승대','이아름');
```

결과

3개 행 이 (가) 삭제되었습니다.


```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95		
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

```
COMMIT;
```

결과

커밋 완료.

COMMIT

KEYWORD : 데이터베이스 변경 내용을 반영

구 문	COMMIT;
-----	---------

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95		
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

계산

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	3	독고승재	97	88	95	280	93.3
B	4	이도희	86	92	90	268	89.3

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

Commit으로 반영 처리

:: TB_GRADE 테이블에서 STUDENT_NM이 '독고승재', '이도희' 학생의 KOR_ ENG_ MAT에 대한 TOT와 AVG를 업데이트 하고 COMMIT하여 데이터베이스에 반영

```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95		
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	0	0
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

```
UPDATE TB_GRADE
SET    TOT = KOR + ENG + MAT,
      AVG = ROUND((KOR + ENG + MAT) / 3,1)
WHERE  STUDENT_NM IN ('독고승재','이도희');
```

결과

2개 행 이 (가) 업데이트되었습니다.

```
SELECT *
FROM   TB_GRADE
ORDER BY CLASS_CD,
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

```
COMMIT;
```

결과

커밋 완료.

ROLLBACK

KEYWORD : 데이터베이스 변경 내용을 취소

구 문	ROLLBACK ;
-----	------------

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	98.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

삭제

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	98.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7

TB_GRADE							
CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	98.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

Rollback으로 복원 처리

∴ TB_GRADE 테이블에서 CLASS_CD가 'B', 'D'인 데이터를 삭제하고 ROLLBACK으로 원래 데이터 상태로 복구

```
SELECT *  
FROM   TB_GRADE  
ORDER  BY CLASS_CD,  
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

```
DELETE
FROM    TB_GRADE
WHERE   CLASS_CD IN ('B','D');
```

결과

5개 행 이 (가) 삭제되었습니다.

```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7

```
ROLLBACK;
```

결과

롤백 완료.

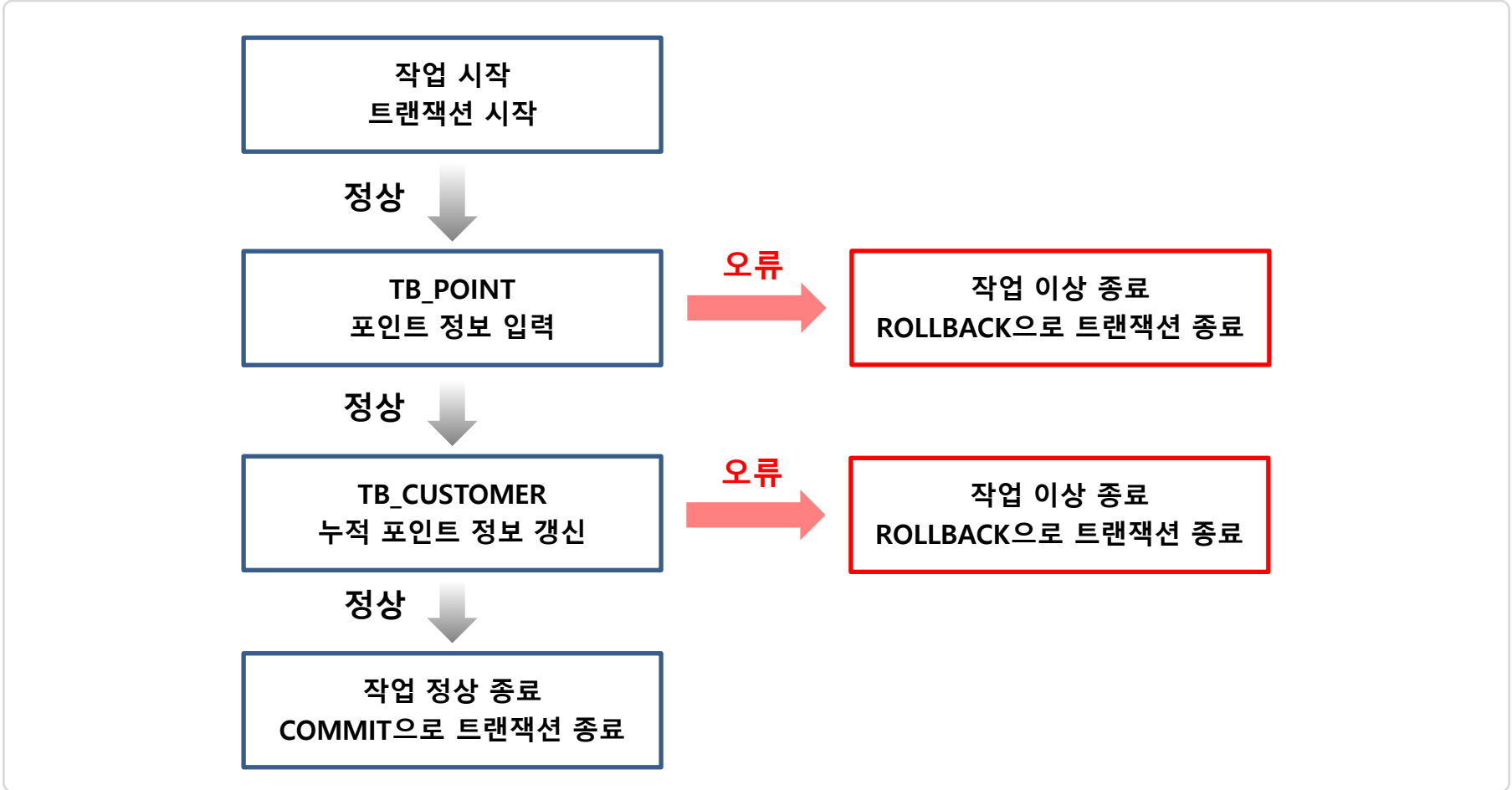
```
SELECT *
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        STUDENT_NO;
```

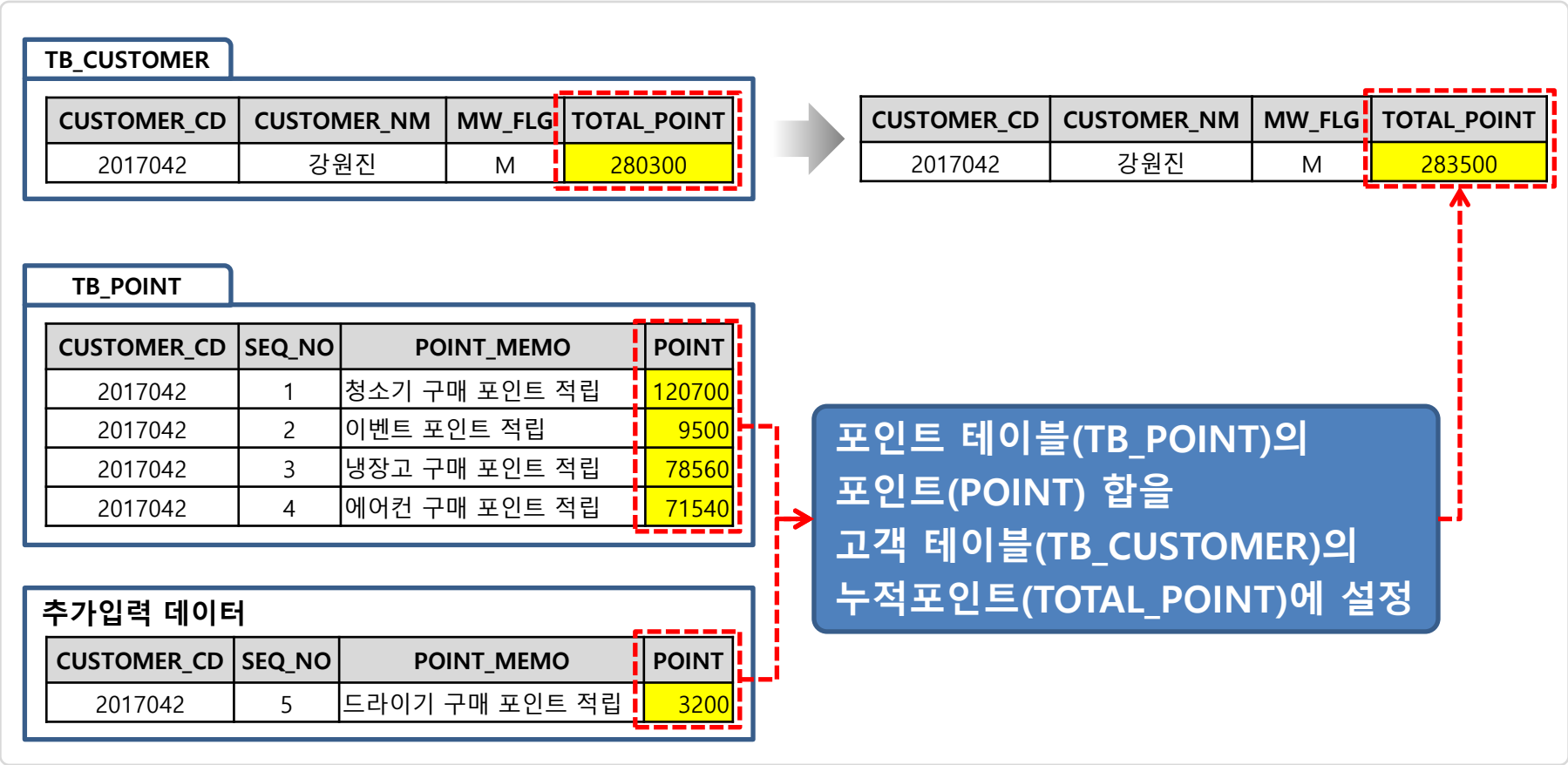
결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

트랜잭션의 이해

KEYWORD : 여러 개의 단위 작업 배치 시 처리 방법





**:: TB_POINT 테이블에 포인트 이력을 추가하고 TB_CUSTOMER 테이블의 TOTAL_POINT에
누적 포인트를 변경**

```
SELECT *
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD = '2017042';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	280300	20170123132432

```
SELECT *  
FROM   TB_POINT  
WHERE  CUSTOMER_CD = '2017042';
```

결과

⚡ CUSTOMER_CD	⚡ SEQ_NO	⚡ POINT_MEMO	⚡ POINT	⚡ REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913

```
INSERT INTO TB_POINT VALUES ( '2017042' ,5, '드라이기 구매 포인트 적립' ,3200, '20190801234530' );
```

결과

1 행 이 (가) 삽입되었습니다.

```
SELECT *
FROM   TB_POINT
WHERE  CUSTOMER_CD = '2017042';
```

결과

↕ CUSTOMER_CD	↕ SEQ_NO	↕ POINT_MEMO	↕ POINT	↕ REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017042	5	드라이기 구매 포인트 적립	3200	20190801234530

```
UPDATE TB_CUSTOMER CU
SET    CU.TOTAL_POINT = (SELECT SUM(CP.POINT)
                        FROM    TB_POINT CP
                        WHERE   CP.CUSTOMER_CD = CU.CUSTOMER_CD
                        GROUP  BY CP.CUSTOMER_CD
                        )
WHERE  CP.CUSTOMER_CD = '2017042';
```

결과

오류 발생 명령행: 824 열: 9
오류 보고 -
SQL 오류: ORA-00904: "CP"."CUSTOMER_CD": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:

```
ROLLBACK;
```

결과

롤백 완료.

*** 예제#1 ~ 예제#4 까지 다시 실행하고 예제#7을 실행한다.**

```
UPDATE TB_CUSTOMER CU
SET    CU.TOTAL_POINT = (SELECT SUM(CP.POINT)
                        FROM    TB_POINT CP
                        WHERE   CP.CUSTOMER_CD = CU.CUSTOMER_CD
                        GROUP  BY CP.CUSTOMER_CD
                        )
WHERE  CU.CUSTOMER_CD = '2017042';
```

결과

1 행 이 (가) 업데이트되었습니다.


```
SELECT *  
FROM   TB_CUSTOMER  
WHERE  CUSTOMER_CD = '2017042';
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	283500	20170123132432

```
COMMIT;
```

결과

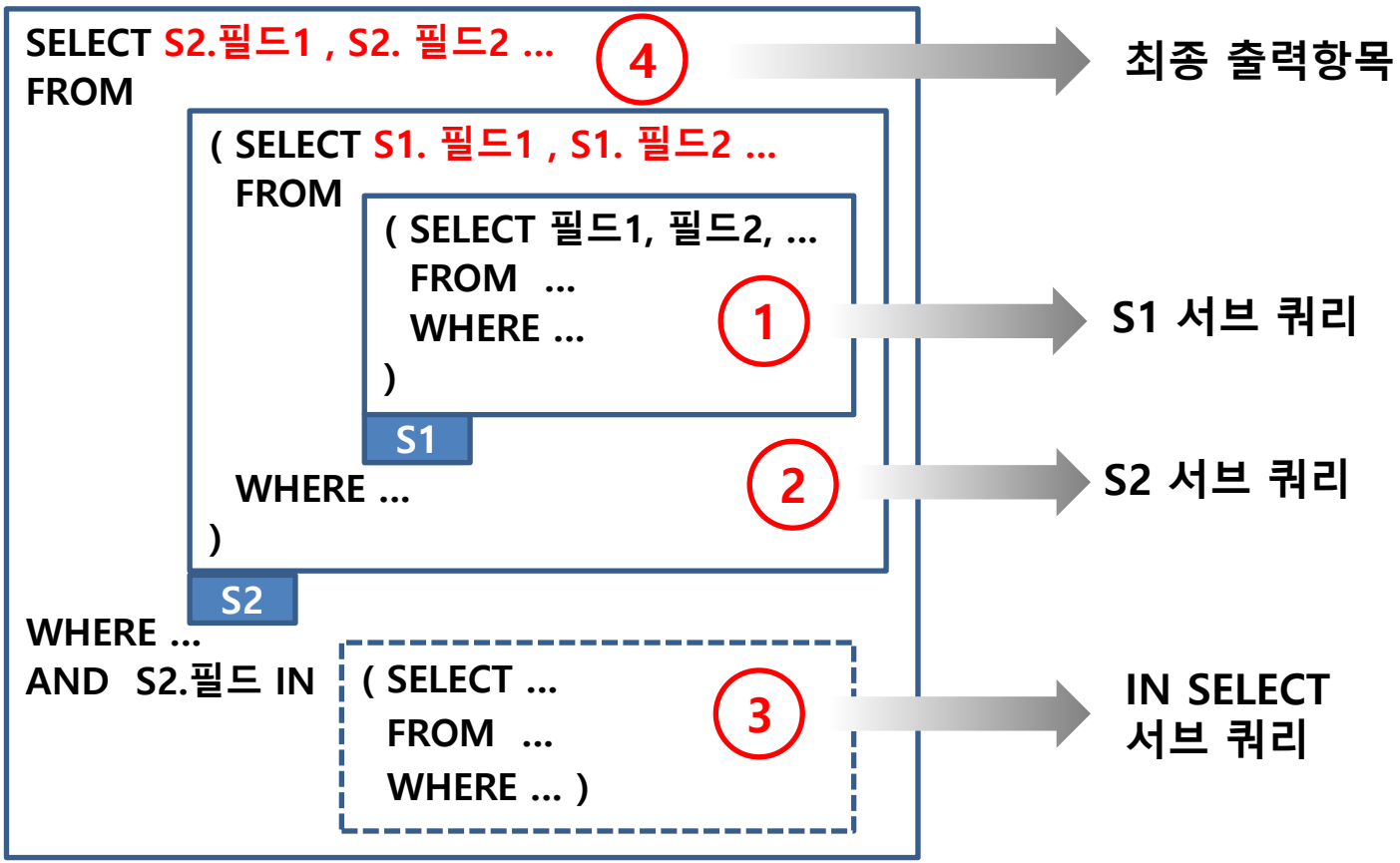
커밋 완료.

서브쿼리(SUB QUERY)의 이해

KEYWORD : SELECT 구문 내 SELECT 작업

구문 #1	<pre>SELECT S1.필드명1, S1.필드명2 ..., S1.필드명n FROM (SELECT 필드명1, 필드명2 ..., 필드명n FROM 테이블명 WHERE 조건식) S1 WHERE 조건식;</pre>
구문 #2	<pre>SELECT 필드명1, 필드명2 ..., 필드명n FROM 테이블명 WHERE 필드명 [IN =] (SELECT 필드명 FROM 테이블명 WHERE 조건식)</pre>

사용 예	<pre>SELECT S1.X, S1.Y FROM (SELECT CUSTOMER_CD AS X, POINT AS Y FROM TB_POINT) S1 WHERE B.CUSTOMER_CD IN (SELECT A.CUSTOMER_CD FROM TB_POINT A WHERE A.CUSTOMER_CD = B.CUSTOMER_CD)</pre>
------	---



```
SELECT  S2.CUSTOMER_CD,  
        (SELECT CT.CUSTOMER_NM  
          FROM   TB_CUSTOMER CT  
          WHERE  CT.CUSTOMER_CD = S2.CUSTOMER_CD  
          ) AS CUSTOMER_NM,  
        S2.TOT_POINT  
FROM    (  
  SELECT  S1.CUSTOMER_CD,  
          S1.TOT_POINT  
  FROM    (  
    SELECT CUSTOMER_CD,  
           SUM(POINT) AS TOT_POINT  
    FROM   TB_POINT  
    WHERE  CUSTOMER_CD >= '2019000'  
    GROUP BY CUSTOMER_CD  
  ) S1  
  WHERE  S1.TOT_POINT > 10000  
  ) S2  
WHERE   S2.CUSTOMER_CD IN (  
        SELECT CUSTOMER_CD  
        FROM   TB_CUSTOMER  
        WHERE  MW_FLG = 'W'  
        );
```

결과

CUSTOMER_CD	CUSTOMER_NM	TOT_POINT
2019001	김진숙	12920
2019281	미아름	35500

S1 결과

S2 결과

S2 결과에 WHERE 조건 결과

CUSTOMER_CD	TOT_POINT
2019001	12920
2019069	15320
2019095	890
2019167	6800
2019281	35500



CUSTOMER_CD	TOT_POINT
2019001	12920
2019069	15320
2019281	35500



CUSTOMER_CD	CUSTOMER_NM	TOT_POINT
2019001	김진숙	12920
2019281	미아름	35500

TB_GRADE

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3
B	2	서유리	96	100	92	288	96
B	3	이혜옥	98	86	78	262	87.3
B	4	이도희	86	92	90	268	89.3
C	1	김진숙	95	77	95	267	89
C	2	김한길	73	84	100	257	85.7
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84



등수	CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
1	A	3	독고승재	97	88	95	280	93.3
2	A	1	강원진	87	94	98	279	93
3	C	1	김진숙	95	77	95	267	89
4	C	2	김한길	73	84	100	257	85.7

∴ 성적 테이블에서 반코드가 'A' 또는 'C'반의 학생을 대상으로 합계가 많은 순으로 행번호를 보인다. (가장 높은 점수가 1등이다.)

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	TOTAL_POINT
2017042	강원진	M	283500
2017053	나경숙	W	4500
2017108	박승대	M	13450
2018087	서유리	W	18700
2018254	이혜옥	W	570
2019001	김진숙	W	12820
2019069	김한길	M	15320
2019095	남궁소망	M	890
2019167	한찬희	M	16800
2019281	이마름	W	35600
2019302	전미래	W	0

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT
2017042	1	청소기 구매 포인트 적립	120700
2017042	2	이벤트 포인트 적립	9500
2017042	3	냉장고 구매 포인트 적립	78560
2017042	4	에어컨 구매 포인트 적립	71540
2017053	1	세탁기 구매 포인트 적립	3500
2017053	2	드라미기 구매 포인트 적립	2000
2017108	1	청소기 구매 포인트 적립	14065
2017108	2	이벤트 포인트 적립	9385
2018087	1	이벤트 포인트 적립	7800
2018087	2	냉장고 구매 포인트 적립	10900

나경숙 고객의 상세 포인트 합은 5,500인데 고객 테이블의 누적포인트는 4,500으로 잘못 설정되어 있음



CUSTOMER_CD	CUSTOMER_NM	TOTAL_POINT	체크 누적합
2017053	나경숙	4500	5500
2019001	김진숙	12820	12920
2019281	이마름	35600	35500

:: 고객포인트 테이블의 고객 포인트 합이 고객 테이블에 누적포인트와 다른 것을 검색한다.

EXISTS

KEYWORD : 다중 데이터 조건 검색

구 문	<pre>SELECT 검색필드 FROM 테이블 WHERE [NOT] EXISTS (SELECT [필드 의미없는 문자] FROM 테이블 WHERE 조건식);</pre>
사용 예	<pre>EXISTS (SELECT CUSTOMER_CD FROM TB_POINT WHERE REG_DTTM LIKE '2019%')</pre>

TB_CUSTOMER								
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY					
2017042	강원진	M	19810	TB_POINT				
2017053	나경숙	W	19891	CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	...
2017108	박승대	M	19711	2017042	1	청소기 구매 포인트 적립	120700	...
2018087	서유리	W	19810	2017042	2	이벤트 포인트 적립	9500	...
2018254	이혜옥	W	19839	2017042	3	냉장고 구매 포인트 적립	78560	...
2019001	김진숙	W	20010	2017042	4	에어컨 구매 포인트 적립	71540	...
2019069	김한길	M	1992 3	2017053	1	세탁기 구매 포인트 적립	3500	...
2019095	남궁소망	M	19620	2017053	2	드라이기 구매 포인트 적립	2000	...
2019167	한찬희	M	19711	2017108	1	청소기 구매 포인트 적립	14065	...
2019281	이아름	W	19940	2017108	2	이벤트 포인트 적립	9385	...
				2018087	1	이벤트 포인트 적립	7800	...
				2018087	2	냉장고 구매 포인트 적립	10900	...
				2018254	1	등록 포인트 적립	500	...
				2018254	2	이벤트 포인트 적립	70	...
				:	:	:	:	...
				:	:	:	:	...

:: TB_CUSTOMER 테이블에서 MW_FLG가 'M'인 데이터의 CUSTOMER_CD가 TB_POINT 테이블의 CUSTOMER_CD와 같으며 TB_POINT 테이블의 내용을 검색

```
SELECT *
FROM   TB_POINT CP
WHERE  EXISTS (SELECT 'A'
                FROM   TB_CUSTOMER CU
                WHERE  CU.CUSTOMER_CD = CP.CUSTOMER_CD
                AND    CU.MW_FLG = 'M');
```

결과

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	1	청소기 구매 포인트 적립	120700	20181221160803
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017042	3	냉장고 구매 포인트 적립	78560	20190612220810
2017042	4	에어컨 구매 포인트 적립	71540	20190703140913
2017108	1	청소기 구매 포인트 적립	14065	20180412205434
2017108	2	이벤트 포인트 적립	9385	20180702232143
2019069	1	이벤트 포인트 적립	8900	20190219120712
2019069	2	면도기 구매 포인트 적립	3200	20190420090820
2019069	3	전기밥솥 구매 포인트 적립	3220	20190620071230
2019095	1	등록 포인트 적립	500	2019031212456
2019095	2	이벤트 포인트 적립	390	20190510072345
2019167	1	드라이기 구매 포인트 적립	3200	20190612042450
2019167	2	전기밥솥 구매 포인트 적립	3600	20190714133422

TB_POINT				
CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2017042	2	이벤트 포인트 적립	9500	20190405121520
2017108	2	이벤트 포인트 적립	9385	20180702232143
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018254	2	이벤트 포인트 적립	70	20180623170212
2019069	1	이벤트 포인트 적립	8900	20190219120712
2019095	2	이벤트 포인트 적립	390	20190510072345

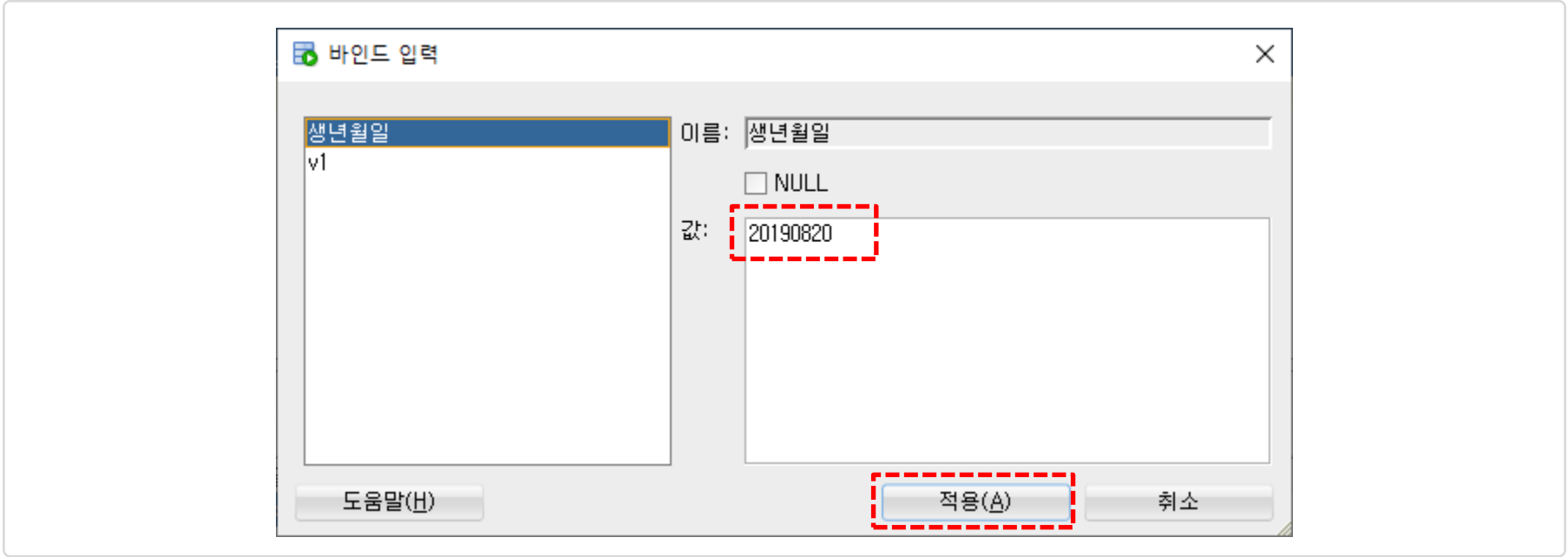
TB_CUSTOMER							
CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2017042	강원진	M	19810603	002-8202-8790	wjgang@navi.com	283500	20170123132432
2017108	박승대	M	19711430	002-2580-9919	sdpark@haso.com	23450	20170508203450
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903
2018254	이혜옥	W	19839012	003_1287_9734	hylee@hansoft.com	570	20180619230805
2019069	김한길	M	1992 315		hkkim@ssoya.com	15320	20190217110704
2019095	남궁소망	M	19620728	003-6273-8539		890	20190312124558

:: 고객포인트 테이블에서 포인트 내용에 “이벤트”가 포함된 고객을 찾아 고객 테이블에 같은 고객코드가 존재한다면 검색한다.

검색 조건을 입력해서 SQL을 실행

KEYWORD : 변수로 실행

구문 #1	필드 = :검색조건명
구문 #2	함수(:검색조건명)
사용 예	WHERE CLASS_CD = :반코드 WHERE A필드 = TO_DATE(:생년월일)



:: 바인드 입력 화면에서 해당 변수의 값을 입력하고 적용

```
SELECT *  
FROM TB_GRADE  
WHERE CLASS_CD = :v1  
ORDER BY CLASS_CD,  
        STUDENT_NO;
```

바인드 입력

v1

이름: v1

☐ NULL

값:

A

검색 조건값 입력창

도움말(H)

적용(A)

취소

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3

:v1 (A)

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

:v1 (D)

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG

:v1 (F)

:: SQL 구문을 :v1 변수를 사용하여 실행합니다.


```
SELECT *
FROM   TB_GRADE
WHERE  CLASS_CD = :v1
ORDER BY CLASS_CD,
        STUDENT_NO;
```

결과 (:v1 = 'A')

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
A	1	강원진	87	94	98	279	93
A	3	독고승재	97	88	95	280	93.3

결과 (:v1 = 'D')

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
D	1	남궁소망	56	68	78	202	67.3
D	2	한찬희	94	90	68	252	84

결과 (:v1 = 'F')

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	ENG	MAT	TOT	AVG
----------	------------	------------	-----	-----	-----	-----	-----

```
SELECT *
FROM   TB_CUSTOMER
WHERE  CUSTOMER_NM LIKE '%' || :고객명 || '%'
AND    MW_FLG = :성별;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2018254	이혜옥	W	19830912
2019281	이마름	W	19940513

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2019302	전미래	W	19740812
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

바인드 입력

고객명

이름: 고객명

☐ NULL

값: 이

도움말(H)

적용(A)

취소

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2018254	1	등록 포인트 적립	500	20180619230805
2018254	2	이벤트 포인트 적립	70	20180623170212
2019281	1	등록 포인트 적립	500	20190709201308
2019281	2	청소기 구매 포인트 적립	8950	20190710200921
2019281	3	냉장고 구매 포인트 적립	12200	20190712082334
2019281	4	드라이기 구매 포인트 적립	7600	20190721134421
2019281	5	믹서기 구매 포인트 적립	6250	20190724022430

:: 고객 테이블의 고객명을 조건으로 입력해 고객명에 포함된 고객코드가 고객포인트 테이블에 있다면 검색한다. (위의 예는 조건을 “이”로 입력했을 때이다.)

TB_CUSTOMER

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY
2019302	전미래	W	19740812
2017042	강원진	M	19810603
2017053	나경숙	W	19891225
2017108	박승대	M	19711430
2018087	서유리	W	19810925
2018254	이혜옥	W	19839012
2019001	김진숙	W	20010426
2019069	김한길	M	1992 315
2019095	남궁소망	M	19620728
2019167	한찬희	M	19711106
2019281	이아름	W	19940513

바인드 입력

고객명

이름: 고객명

☐ NULL

값: 이

도움말(H)

적용(A)

취소

TB_POINT

CUSTOMER_CD	SEQ_NO	POINT_MEMO	POINT	REG_DTTM	CUSTOMER_NM	MW_FLG
2018254	1	등록 포인트 적립	500	20180619230805	이혜옥	W
2018254	2	이벤트 포인트 적립	70	20180623170212	이혜옥	W
2019281	1	등록 포인트 적립	500	20190709201308	이아름	W
2019281	2	청소기 구매 포인트 적립	8950	20190710200921	이아름	W
2019281	3	냉장고 구매 포인트 적립	12200	20190712082334	이아름	W
2019281	4	드라이기 구매 포인트 적립	7600	20190721134421	이아름	W
2019281	5	믹서기 구매 포인트 적립	6250	20190724022430	이아름	W

:: 고객 테이블의 고객명을 조건으로 입력해 고객명에 포함된 고객코드가 고객포인트 테이블에 있다면 검색하면서 고객 테이블의 고객명과 성별을 같이 보인다.

주식 처리

KEYWORD : SQL 구문 내 미실행 문장

구문 #1	REM 설명
구문 #2	-- 설명
구문 #3	/* 설명 */
사용 예	REM :: 고객정보 검색 -- 고객코드 /* 누적포인트 계산 */

```
REM *****
REM :: CUSTOMER_CD가 '2018000' 이상을 검색
REM *****
SELECT CUSTOMER_CD,      — 고객코드
      CUSTOMER_NM,      — 고객명
      MW_FLG,           — 성별
      BIRTH_DAY,        — 생일
      /*
      PHONE_NUMBER,      전화번호
      EMAIL,             이메일
      */
      TOTAL_POINT,      — 포인트
      REG_DTTM          — 등록일
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD >= '2018000';
```



실행하는 SQL 구문

```
SELECT CUSTOMER_CD,
      CUSTOMER_NM,
      MW_FLG,
      BIRTH_DAY,
      TOTAL_POINT,
      REG_DTTM
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD >= '2018000';
```

:: SQL 구문 내 주식 처리

```
REM *****
REM :: CUSTOMER_CD가 '2018000' 이상을 검색
REM *****

SELECT CUSTOMER_CD,      -- 고객코드
        CUSTOMER_NM,     -- 고객명
        MW_FLG,          -- 성별
        BIRTH_DAY,       -- 생일
        /*
        PHONE_NUMBER,     전화번호
        EMAIL,            이메일
        */
        TOTAL_POINT,     -- 포인트
        REG_DTTM          -- 등록일
FROM    TB_CUSTOMER
WHERE   CUSTOMER_CD >= '2018000';
```

결과

↕ CUSTOMER_CD	↕ CUSTOMER_NM	↕ MW_FLG	↕ BIRTH_DAY	↕ TOTAL_POINT	↕ REG_DTTM
2018087	서유리	W	19810925	18700	20180204160903
2018254	이혜옥	W	19839012	570	20180619230805
2019001	김진숙	W	20010426	12820	20190101080518
2019069	김한길	M	1992 315	15320	20190217110704
2019095	남궁소망	M	19620728	890	20190312124558
2019167	한찬희	M	19711106	6800	20190508155600
2019281	이아름	W	19940513	35600	20190709201308
2019302	전미래	W	19740812	0	20191118121320