

15장. 그룹함수

15.1. MAX()

15.2. MIN()

15.3. SUM()

15.4. AVG()

15.5. COUNT()

15.6. ROLLUP()

15.7. ROW_NUMBER()

15.8. RANK()

15.9. DENSE_RANK()

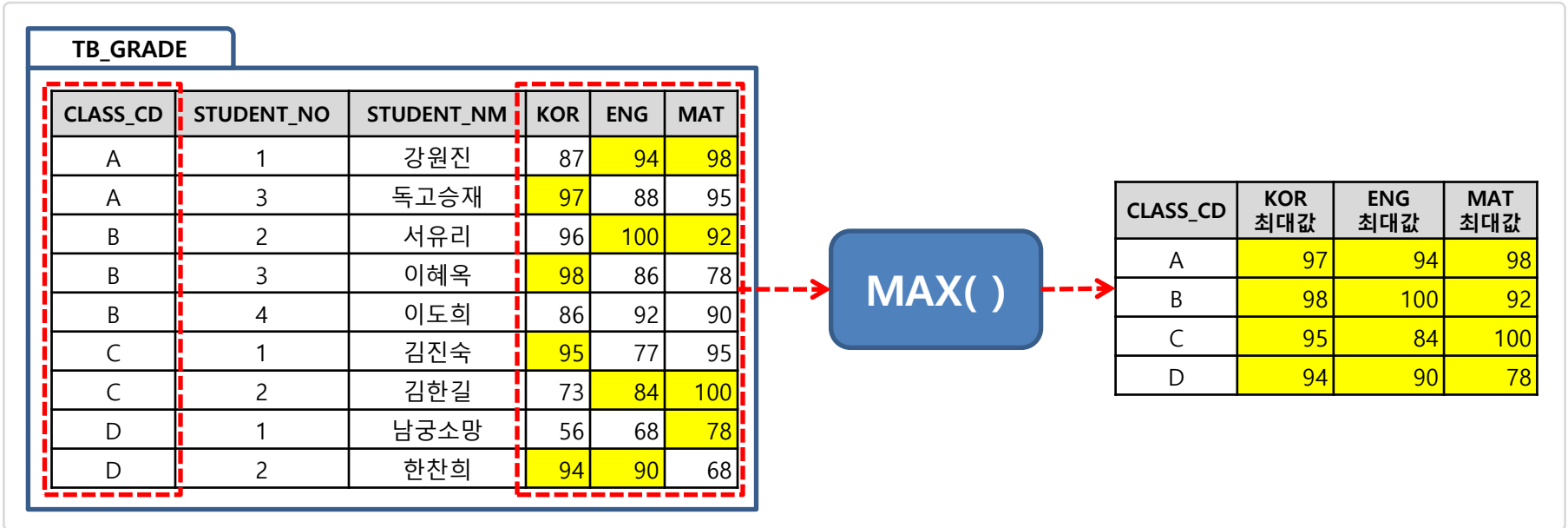
15.10. RANK() WITHIN

15.11. FIRST_VALUE()

MAX()

KEYWORD : 그룹 항목의 최대값 취득

구문 #1	MAX(필드)
구문 #2	MAX(필드) OVER (PARTITION BY 그룹 필드)
결과값	지정 값에 따라 다름
사용 예	MAX(KOR) ... GROUP BY CLASS_CD PARTITION BY CLASS_CD MAX(KOR) OVER (PARTITION BY CLASS_CD)

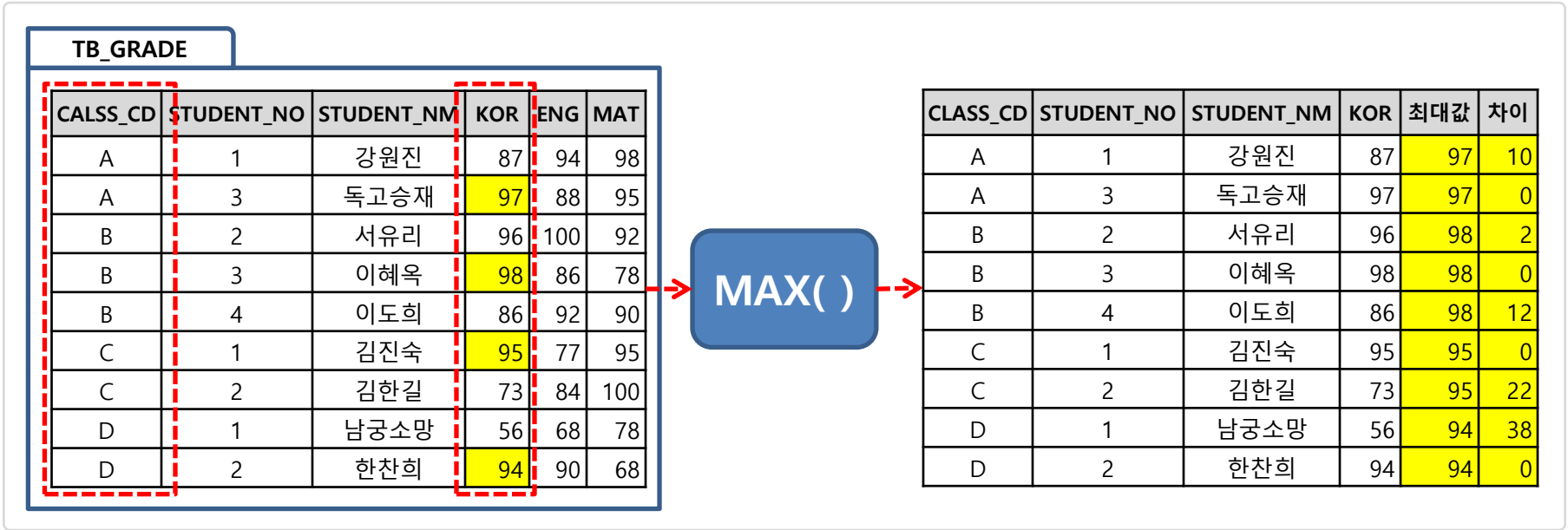


:: TB_GRADE 테이블에서 CLASS_CD를 그룹화하여 KOR, ENG, MAT의 최대값을 검색

```
SELECT CLASS_CD,  
       MAX(KOR) AS KOR_최대값,  
       MAX(ENG) AS ENG_최대값,  
       MAX(MAT) AS MAT_최대값  
FROM   TB_GRADE  
GROUP  BY CLASS_CD  
ORDER  BY CLASS_CD;
```

결과

CLASS_CD	KOR_최대값	ENG_최대값	MAT_최대값
A	97	94	98
B	98	100	92
C	95	84	100
D	94	90	78



:: TB_GRADE 테이블에서 CLASS_CD에 대한 KOR 최대값, KOR 최대값과 KOR 값의 차이를 검색

```
SELECT CLASS_CD,
       STUDENT_NO,
       STUDENT_NM,
       KOR,
       MAX(KOR) OVER (PARTITION BY CLASS_CD) AS 최대값,
       MAX(KOR) OVER (PARTITION BY CLASS_CD) - KOR AS 차이
FROM   TB_GRADE
ORDER  BY CLASS_CD,
         STUDENT_NO;
```

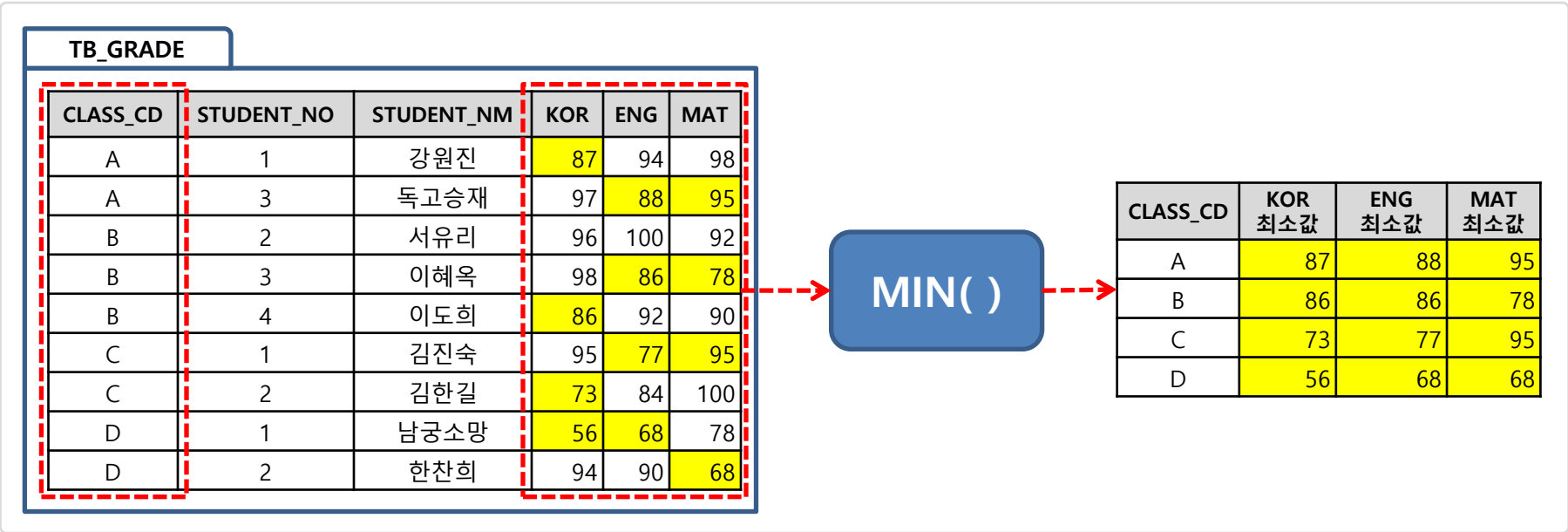
결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	최대값	차이
A	1	강원진	87	97	10
A	3	독고승재	97	97	0
B	2	서유리	96	98	2
B	3	이혜옥	98	98	0
B	4	이도희	86	98	12
C	1	김진숙	95	95	0
C	2	김한길	73	95	22
D	1	남궁소망	56	94	38
D	2	한찬희	94	94	0

MIN()

KEYWORD : 그룹 항목의 최소값 취득

구문 #1	MIN(필드)
구문 #2	MIN(필드) OVER (PARTITION BY 그룹 필드)
결과값	지정 값에 따라 다름
사용 예	MIN(KOR) OVER (PARTITION BY CLASS_CD)

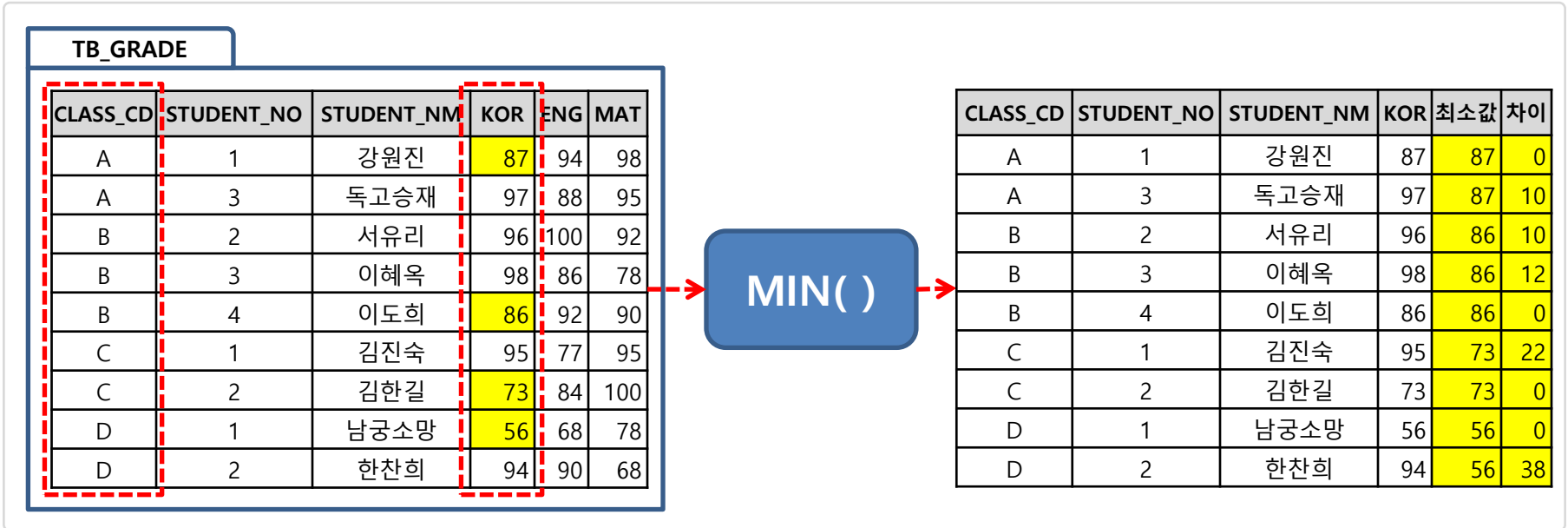


:: TB_GRADE 테이블에서 CLASS_CD를 그룹화하여 KOR, ENG, MAT의 최소값을 검색

```
SELECT CLASS_CD,  
       MIN(KOR) AS KOR_최소값,  
       MIN(ENG) AS ENG_최소값,  
       MIN(MAT) AS MAT_최소값  
FROM   TB_GRADE  
GROUP  BY CLASS_CD  
ORDER  BY CLASS_CD;
```

결과

CLASS_CD	KOR_최소값	ENG_최소값	MAT_최소값
A	87	88	95
B	86	86	78
C	73	77	95
D	56	68	68



:: TB_GRADE 테이블에서 CLASS_CD에 대한 KOR의 최소값, KOR 값과 KOR의 최소값의 차이 검색

```
SELECT CLASS_CD,
       STUDENT_NO,
       STUDENT_NM,
       KOR,
       MIN(KOR) OVER (PARTITION BY CLASS_CD) AS 최소값,
       KOR - MIN(KOR) OVER (PARTITION BY CLASS_CD) AS 차이
FROM   TB_GRADE
ORDER  BY CLASS_CD,
         STUDENT_NO;
```

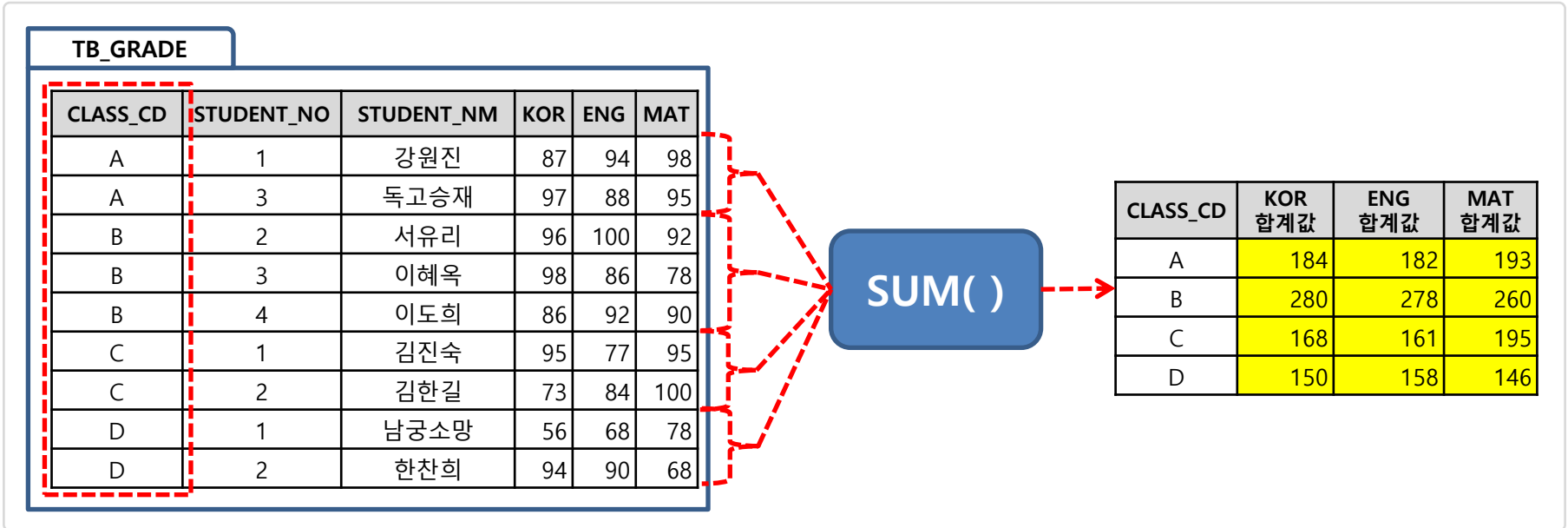
결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	최소값	차이
A	1	강원진	87	87	0
A	3	독고승재	97	87	10
B	2	서유리	96	86	10
B	3	이혜옥	98	86	12
B	4	이도희	86	86	0
C	1	김진숙	95	73	22
C	2	김한길	73	73	0
D	1	남궁소망	56	56	0
D	2	한찬희	94	56	38

SUM()

KEYWORD : 그룹 항목의 합계 취득

구문 #1	SUM(필드)
구문 #2	SUM(필드) OVER (PARTITION BY 그룹 필드)
결과값	수치
사용 예	SUM(KOR) OVER (PARTITION BY CLASS_CD)



:: TB_GRADE 테이블에서 CLASS_CD를 그룹화하여 KOR, ENG, MAT 항목의 합계를 검색

```
SELECT CLASS_CD,  
       SUM(KOR) AS KOR_합계값,  
       SUM(ENG) AS ENG_합계값,  
       SUM(MAT) AS MAT_합계값  
FROM   TB_GRADE  
GROUP  BY CLASS_CD  
ORDER  BY CLASS_CD;
```

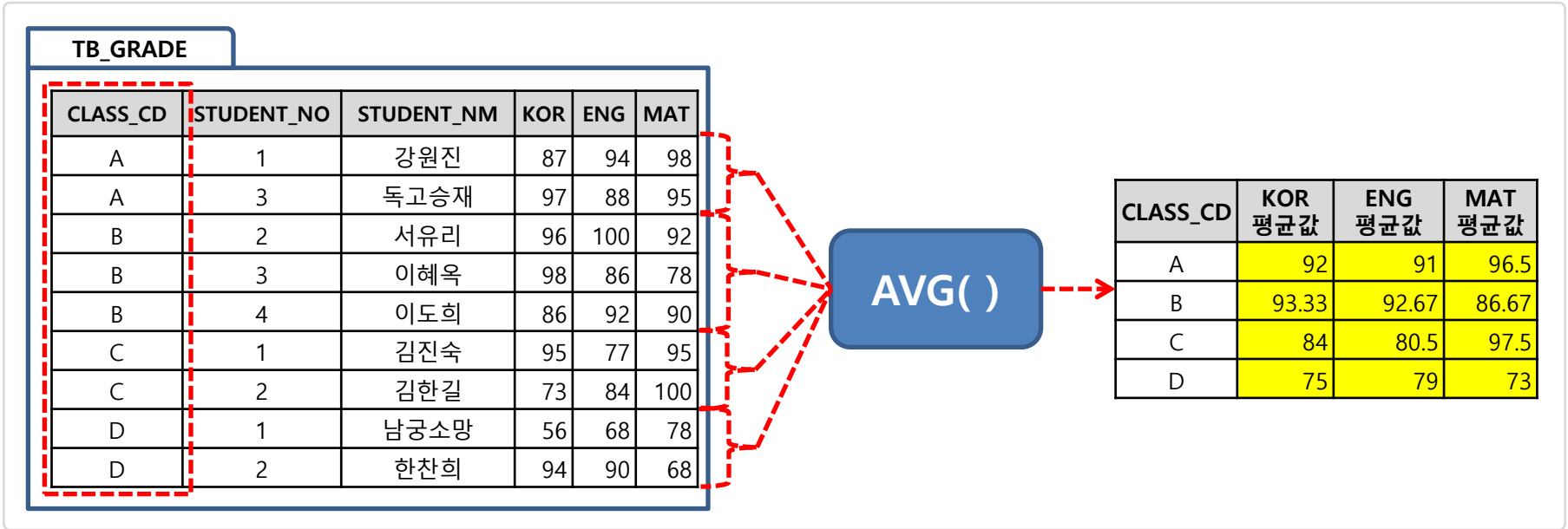
결과

CLASS_CD	KOR_합계값	ENG_합계값	MAT_합계값
A	184	182	193
B	280	278	260
C	168	161	195
D	150	158	146

AVG()

KEYWORD : 그룹 항목의 평균 취득

구문 #1	AVG(필드)
구문 #2	AVG(필드) OVER (PARTITION BY 그룹 필드)
결과값	수치형
사용 예	AVG(KOR) OVER (PARTITION BY CLASS_CD)



:: TB_GRADE 테이블에서 CLASS_CD를 그룹화하여 KOR, ENG, MAT 항목의 평균을 검색

```
SELECT CLASS_CD,  
       ROUND(AVG(KOR),2) AS KOR_평균값,  
       ROUND(AVG(ENG),2) AS ENG_평균값,  
       ROUND(AVG(MAT),2) AS MAT_평균값  
FROM   TB_GRADE  
GROUP  BY CLASS_CD  
ORDER  BY CLASS_CD;
```

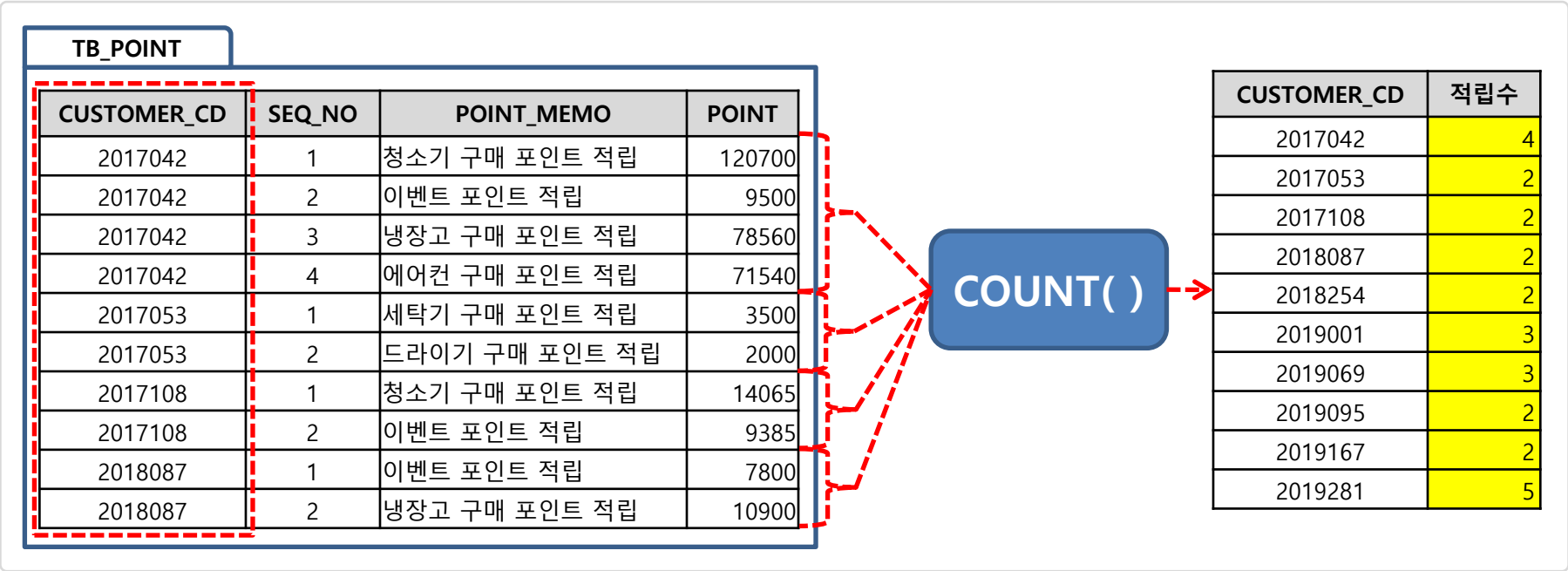
결과

CLASS_CD	KOR_평균값	ENG_평균값	MAT_평균값
A	92	91	96.5
B	93.33	92.67	86.67
C	84	80.5	97.5
D	75	79	73

COUNT()

KEYWORD : 그룹 항목의 개수 취득

구문 #1	COUNT(필드)
구문 #2	COUNT(필드) OVER (PARTITION BY 그룹 필드)
결과값	수치형
사용 예	COUNT(STUDENT_NM) OVER (PARTITION BY CLASS_CD)



:: TB_POINT 테이블에서 CUSTOMER_CD를 그룹화하여 SEQ_NO에 대한 개수를 검색

```
SELECT CUSTOMER_CD,  
       COUNT(SEQ_NO) AS 적립수  
FROM   TB_POINT  
GROUP  BY CUSTOMER_CD  
ORDER  BY CUSTOMER_CD;
```

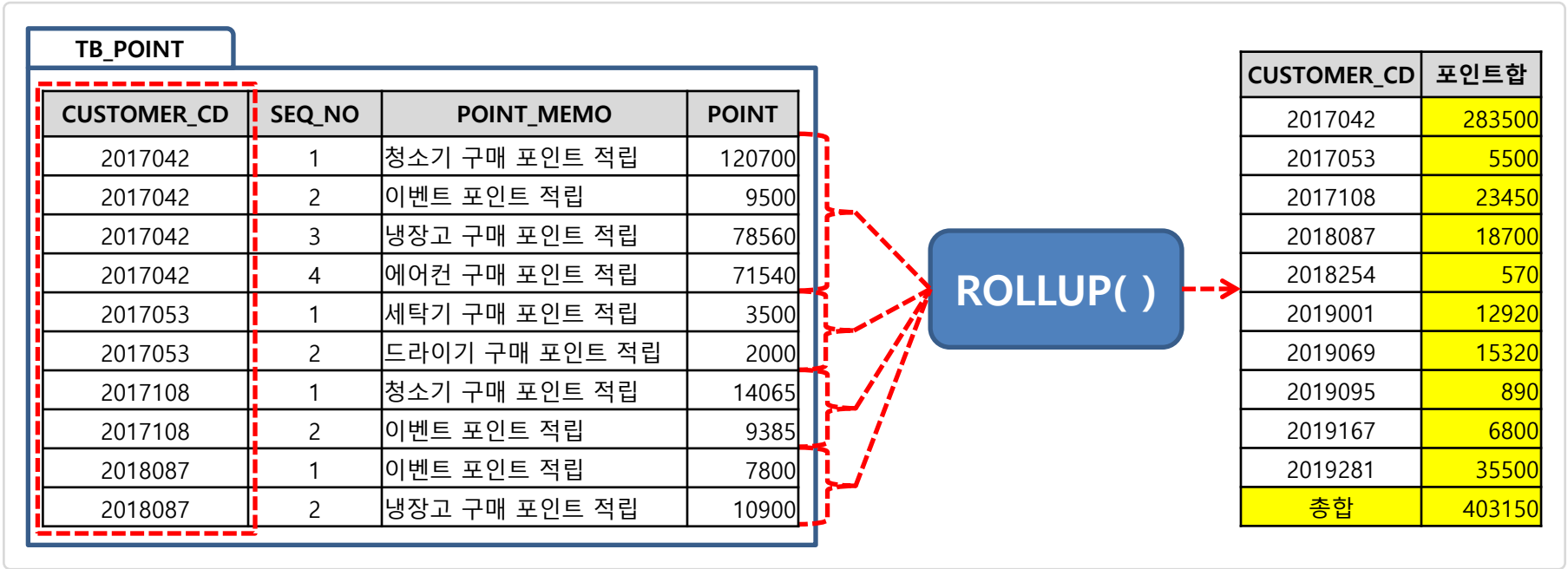
결과

CUSTOMER_CD	적립수
2017042	5
2017053	2
2017108	2
2018087	2
2018254	2
2019001	3
2019069	3
2019095	2
2019167	2
2019281	5

ROLLUP()

KEYWORD : 그룹 항목의 소계 및 합계 취득

구 문	ROLLUP(필드1, 필드2, ... 필드n)
결과값	수치
사용 예	SUM(KOR) ... GROUP BY ROLLUP(CLASS_CD)

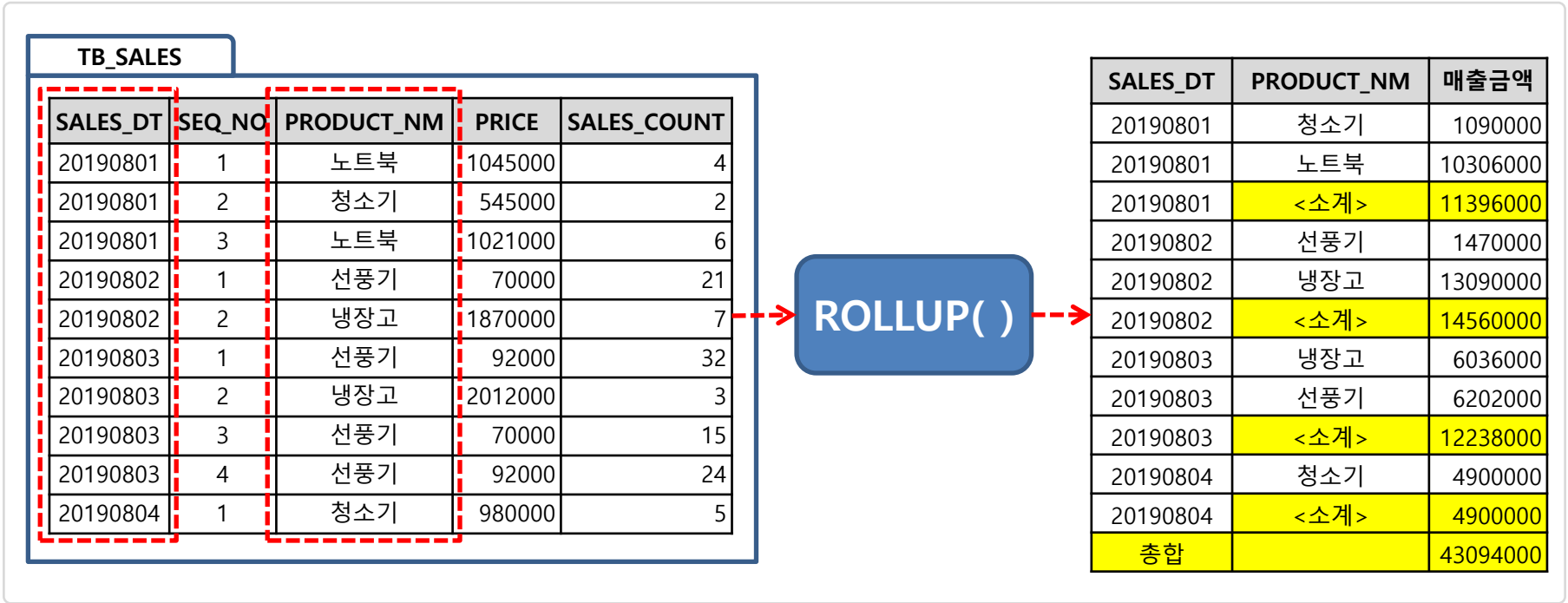


:: TB_POINT 테이블에서 CUSTOMER_CD를 그룹화하여 POINT 항목의 누적합계를 검색


```
SELECT NVL(CUSTOMER_CD, '총합') AS CUSTOMER_CD,  
       SUM(POINT) AS 포인트합  
FROM   TB_POINT  
GROUP  BY ROLLUP(CUSTOMER_CD)  
ORDER  BY CUSTOMER_CD;
```

결과

CUSTOMER_CD	포인트합
2017042	283500
2017053	5500
2017108	23450
2018087	18700
2018254	570
2019001	12920
2019069	15320
2019095	890
2019167	6800
2019281	35500
총합	403150



:: TB_SALES 테이블에서 SALES_DT와 PRODUCT_NM 항목의 소계 및 합계를 검색
매출금액 : 금액(PRICE) * 판매개수(SALES_COUNT)

```
SELECT NVL(SALES_DT, '총합') AS SALES_DT,
       CASE WHEN SALES_DT IS NOT NULL AND PRODUCT_NM IS NULL THEN
           '<소계>'
       ELSE PRODUCT_NM
       END AS PRODUCT_NM,
       SUM(PRICE * SALES_COUNT) AS 매출금액
FROM   TB_SALES
GROUP  BY ROLLUP(SALES_DT, PRODUCT_NM)
ORDER  BY SALES_DT;
```

결과

SALES_DT	PRODUCT_NM	매출금액
20190801	청소기	1090000
20190801	노트북	10306000
20190801	<소계>	11396000
20190802	선풍기	1470000
20190802	냉장고	13090000
20190802	<소계>	14560000
20190803	냉장고	6036000
20190803	선풍기	6202000
20190803	<소계>	12238000
20190804	청소기	4900000
20190804	<소계>	4900000
총합		43094000

```
SELECT SALES_DT,
       PRODUCT_NM,
       SUM(PRICE * SALES_COUNT) AS 매출금액
FROM   TB_SALES
GROUP BY ROLLUP(SALES_DT, PRODUCT_NM)
ORDER BY SALES_DT,
         PRODUCT_NM;
```

```
SELECT SALES_DT,
       PRODUCT_NM,
       SUM(PRICE * SALES_COUNT) AS 매출금액
FROM   TB_SALES
GROUP BY CUBE(SALES_DT,PRODUCT_NM)
ORDER BY SALES_DT,
         PRODUCT_NM;
```

결과

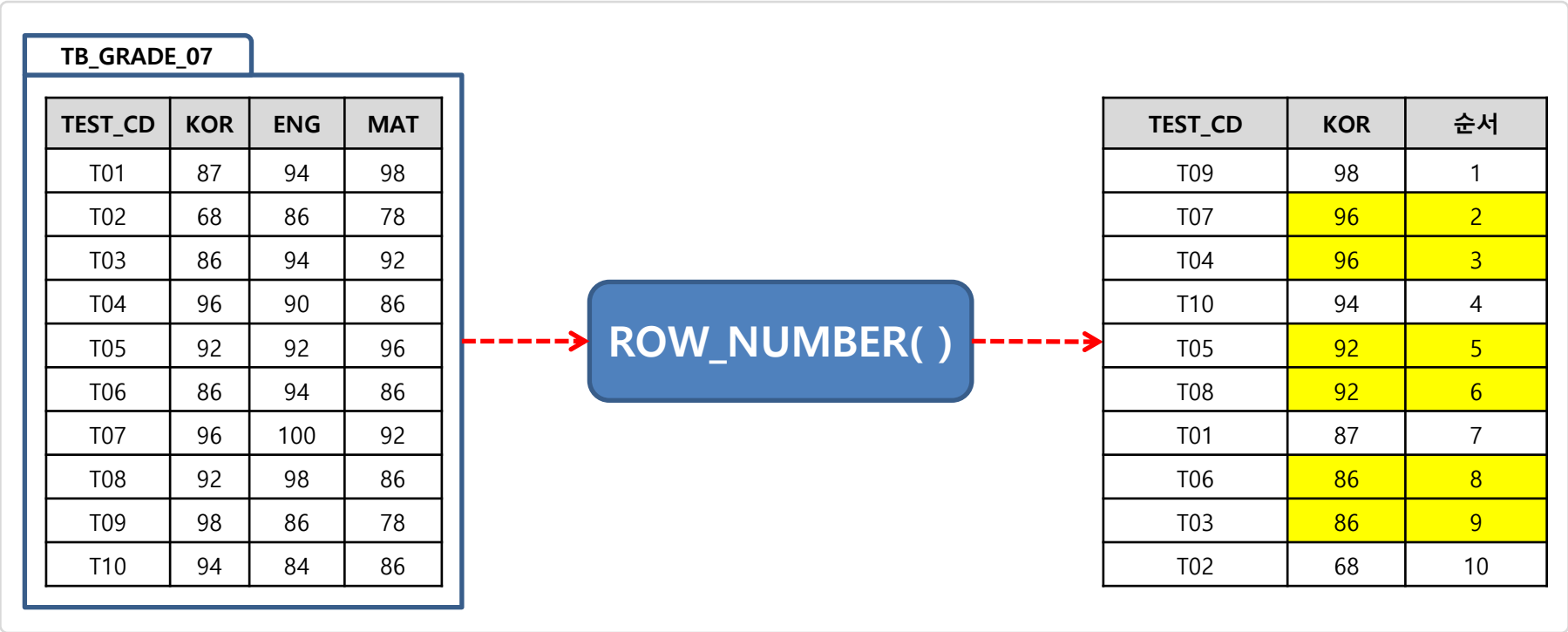
SALES_DT	PRODUCT_NM	매출금액
20190801	노트북	10306000
20190801	청소기	1090000
20190801		11396000
20190802	냉장고	13090000
20190802	선풍기	1470000
20190802		14560000
20190803	냉장고	6036000
20190803	선풍기	6202000
20190803		12238000
20190804	청소기	4900000
20190804		4900000
		43094000

SALES_DT	PRODUCT_NM	매출금액
20190801	노트북	10306000
20190801	청소기	1090000
20190801		11396000
20190802	냉장고	13090000
20190802	선풍기	1470000
20190802		14560000
20190803	냉장고	6036000
20190803	선풍기	6202000
20190803		12238000
20190804	청소기	4900000
20190804		4900000
	냉장고	19126000
	노트북	10306000
	선풍기	7672000
	청소기	5990000
		43094000

ROW_NUMBER()

KEYWORD : 중복 없는 순차 순번 부여

구 문	ROW_NUMBER() OVER (PARTITION BY 그룹필드 ORDER BY 정렬필드)
결과값	수치
사용 예	ROW_NUMBER() OVER (PARTITION BY CLASS_CD ORDER BY READING_CNT)



:: TB_GRADE_07 테이블에서 KOR 항목을 내림차순으로 정렬한 후 중복 없는 순서를 검색

```
SELECT TEST_CD,  
       KOR,  
       ROW_NUMBER() OVER (ORDER BY KOR DESC) AS 순위  
FROM   TB_GRADE_07  
ORDER  BY KOR DESC;
```

결과

TEST_CD	KOR	순위
T09	98	1
T07	96	2
T04	96	3
T10	94	4
T05	92	5
T08	92	6
T01	87	7
T06	86	8
T03	86	9
T02	68	10

RANK()

KEYWORD : 동일 값 중복 순번 후 순위만큼 이동한 순번 부여

구 문	RANK() OVER (PARTITION BY 그룹필드 ORDER BY 정렬필드)
결과값	수치
사용 예	RANK() OVER (PARTITION BY CLASS_CD ORDER BY READING_CNT)



∴ TB_GRADE_07 테이블에서 KOR 항목을 내림차순으로 정렬한 후 동일 값에 중복 순서를 부여 후 순번만큼 이동한 순서를 검색

```
SELECT TEST_CD,  
       KOR,  
       RANK() OVER (ORDER BY KOR DESC) AS 순위  
FROM   TB_GRADE_07  
ORDER  BY KOR DESC;
```

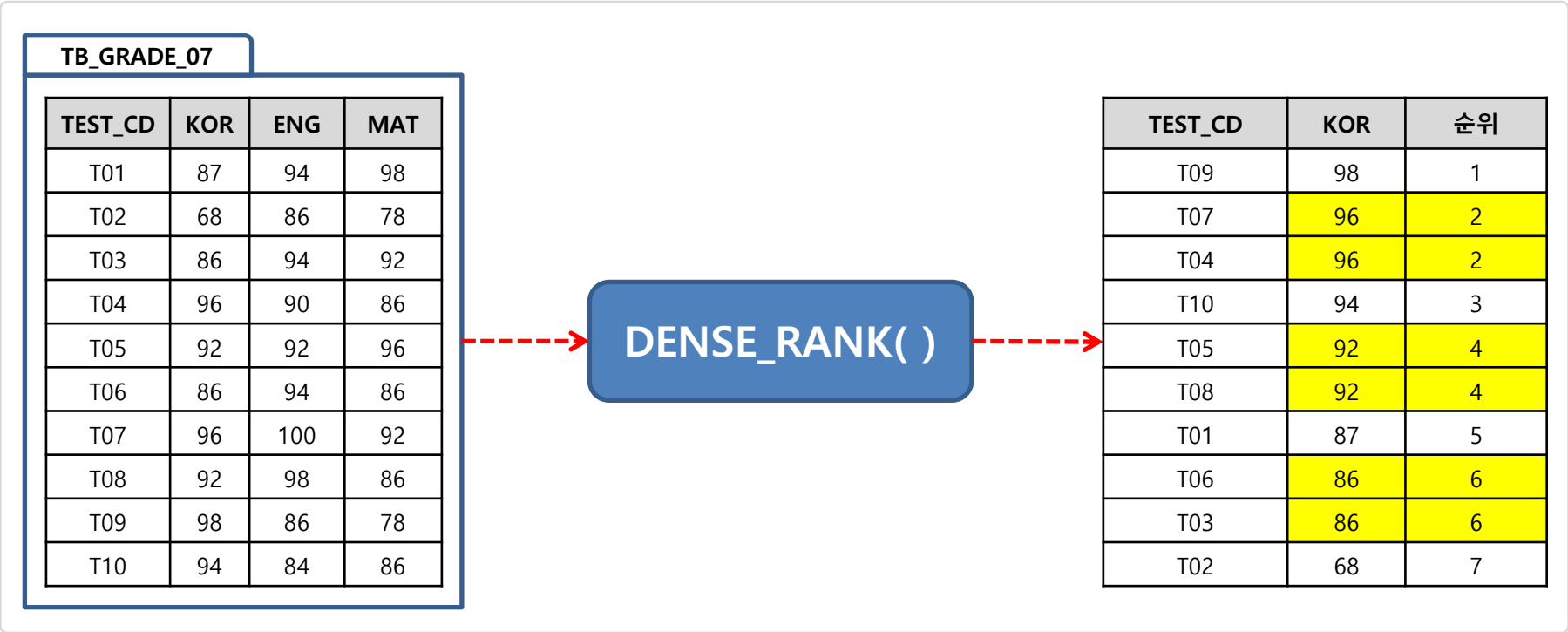
결과

TEST_CD	KOR	순서
T09	98	1
T07	96	2
T04	96	2
T10	94	4
T05	92	5
T08	92	5
T01	87	7
T06	86	8
T03	86	8
T02	68	10

DENSE_RANK()

KEYWORD : 동일 값 중복 순번 후 순차 순번 부여

구 문	DENSE_RANK() OVER (PARTITION BY 그룹필드 ORDER BY 정렬필드)
결과값	수치
사용 예	DENSE_RANK() OVER (PARTITION BY CLASS_CD ORDER BY READING_CNT)



∴ TB_GRADE_07 테이블에서 KOR 항목을 내림차순으로 정렬한 후 동일 값에 중복 순서 부여 후 순차적인 순서를 검색

```
SELECT TEST_CD,  
       KOR,  
       DENSE_RANK() OVER (ORDER BY KOR DESC) AS 순위  
FROM   TB_GRADE_07  
ORDER  BY KOR DESC;
```

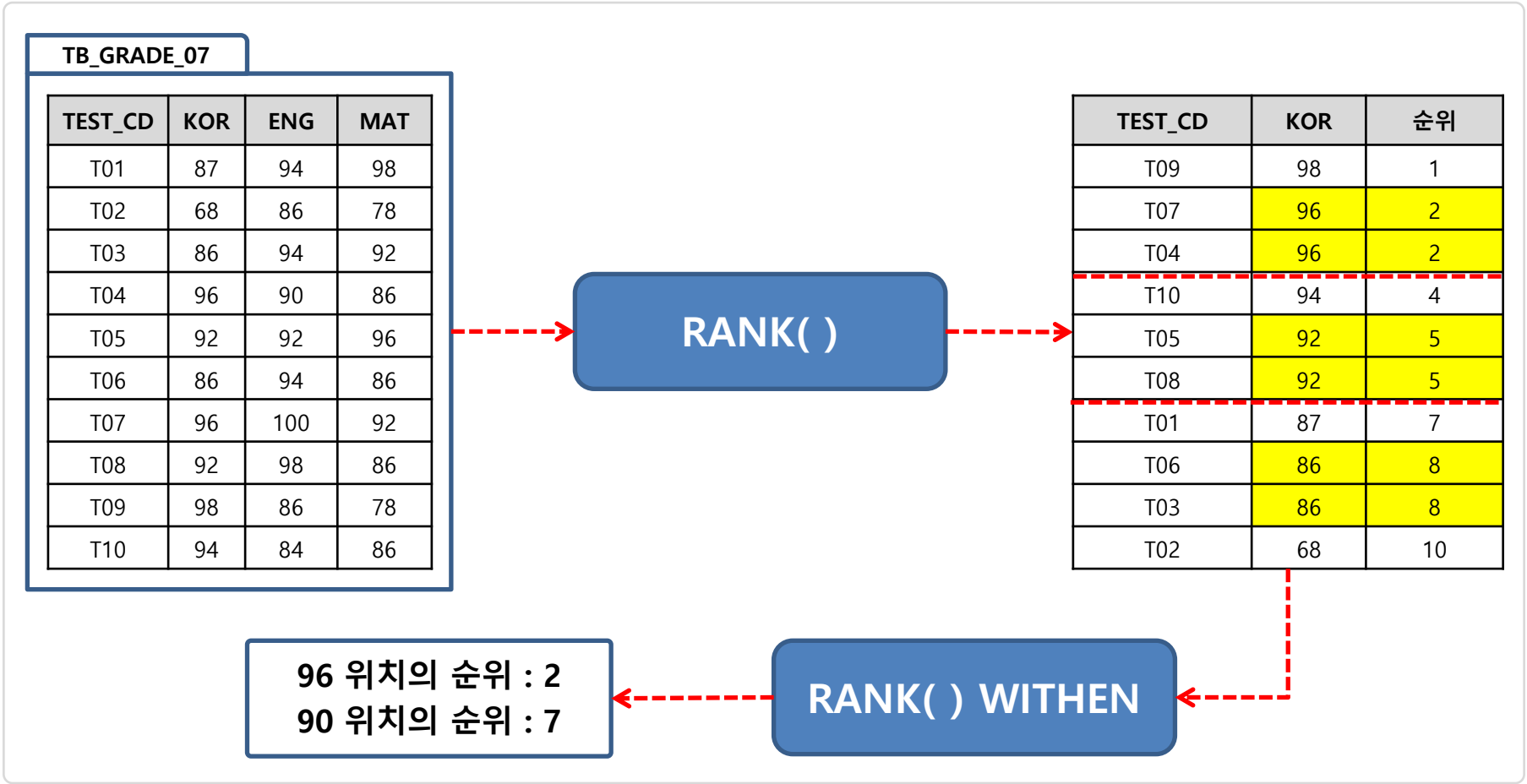
결과

TEST_CD	KOR	순위
T09	98	1
T07	96	2
T04	96	2
T10	94	3
T05	92	4
T08	92	4
T01	87	5
T06	86	6
T03	86	6
T02	68	7

RANK() WITHIN

KEYWORD : RANK() 순번 처리 후 지정 값의 위치 취득

구 문	RANK(값) WITHIN GROUP(정렬필드 [ASC DESC])
결과값	수치
사용 예	RANK(15) WITHIN GROUP(ORDER BY SQNO)



:: tb_grade_07 테이블에서 kor 항목을 중복 있는 내림차순으로 정렬 후 96과 90의 위치를 취득

```
SELECT RANK(96) WITHIN GROUP(ORDER BY KOR DESC) AS "96_위치",  
       RANK(90) WITHIN GROUP(ORDER BY KOR DESC) AS "90_위치"  
FROM   TB_GRADE_07;
```

결과

96_위치	90_위치
2	7


```
SELECT DENSE_RANK(96) WITHIN GROUP(ORDER BY KOR DESC) AS "96_위치",  
       DENSE_RANK(90) WITHIN GROUP(ORDER BY KOR DESC) AS "90_위치"  
FROM   TB_GRADE_07;
```

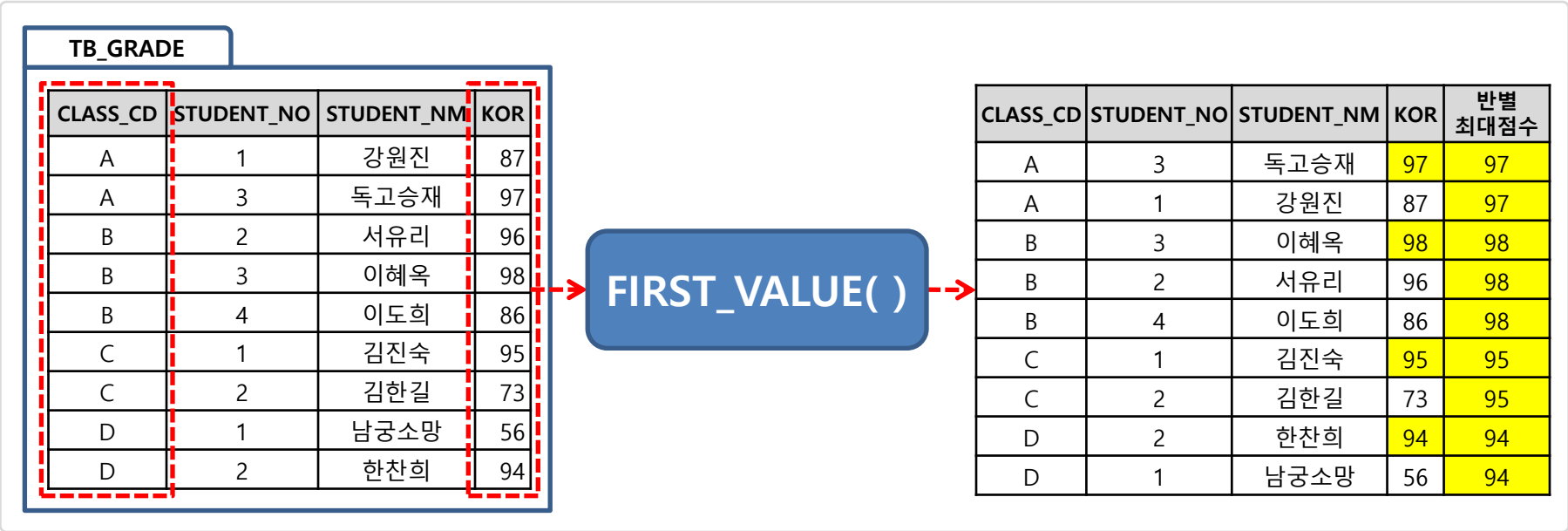
결과

96_위치	90_위치
2	5

FIRST_VALUE()

KEYWORD : RANK() 순번 처리 후 지정 값의 위치 취득

구 문	FIRST_VALUE(필드) OVER (PARTITION BY 그룹필드 ORDER BY 정렬필드)
결과값	지정 값에 따라 다름
사용 예	FIRST_VALUE(KOR) OVER (PARTITION BY CLASS_CD ORDER BY KOR DESC)



:: TB_GRADE 테이블에서 CLASS_CD를 오름차순, KOR을 내림차순 정렬한 후 KOR의 첫번째 값을 검색

```
SELECT CLASS_CD,
       STUDENT_NO,
       STUDENT_NM,
       KOR,
       FIRST_VALUE(KOR) OVER (PARTITION BY CLASS_CD ORDER BY KOR
                             DESC) AS "반별 최대점수"
FROM   TB_GRADE
ORDER  BY CLASS_CD,
        KOR DESC;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	반별 최대점수
A	3	독고승재	97	97
A	1	강원진	87	97
B	3	이혜옥	98	98
B	2	서유리	96	98
B	4	이도희	86	98
C	1	김진숙	95	95
C	2	김한길	73	95
D	2	한찬희	94	94
D	1	남궁소망	56	94

```
SELECT CLASS_CD,
       STUDENT_NO,
       STUDENT_NM,
       KOR,
       LAST_VALUE(KOR) OVER (PARTITION BY CLASS_CD) AS "반별 최소점수"
FROM   TB_GRADE
ORDER  BY CLASS_CD,
       KOR DESC;
```

결과

CLASS_CD	STUDENT_NO	STUDENT_NM	KOR	반별 최소점수
A	3	독고승재	97	87
A	1	강원진	87	87
B	3	미혜옥	98	86
B	2	서유리	96	86
B	4	이도희	86	86
C	1	김진숙	95	73
C	2	김한길	73	73
D	2	한찬희	94	72
D	1	남궁소망	72	72