

17장. 테이블 관련 명령어

17.1. CREATE TABLE

17.2. DESC

17.3. RENAME

17.4. ALTER TABLE

17.5. INDEX

17.6. PRIMARY KEY

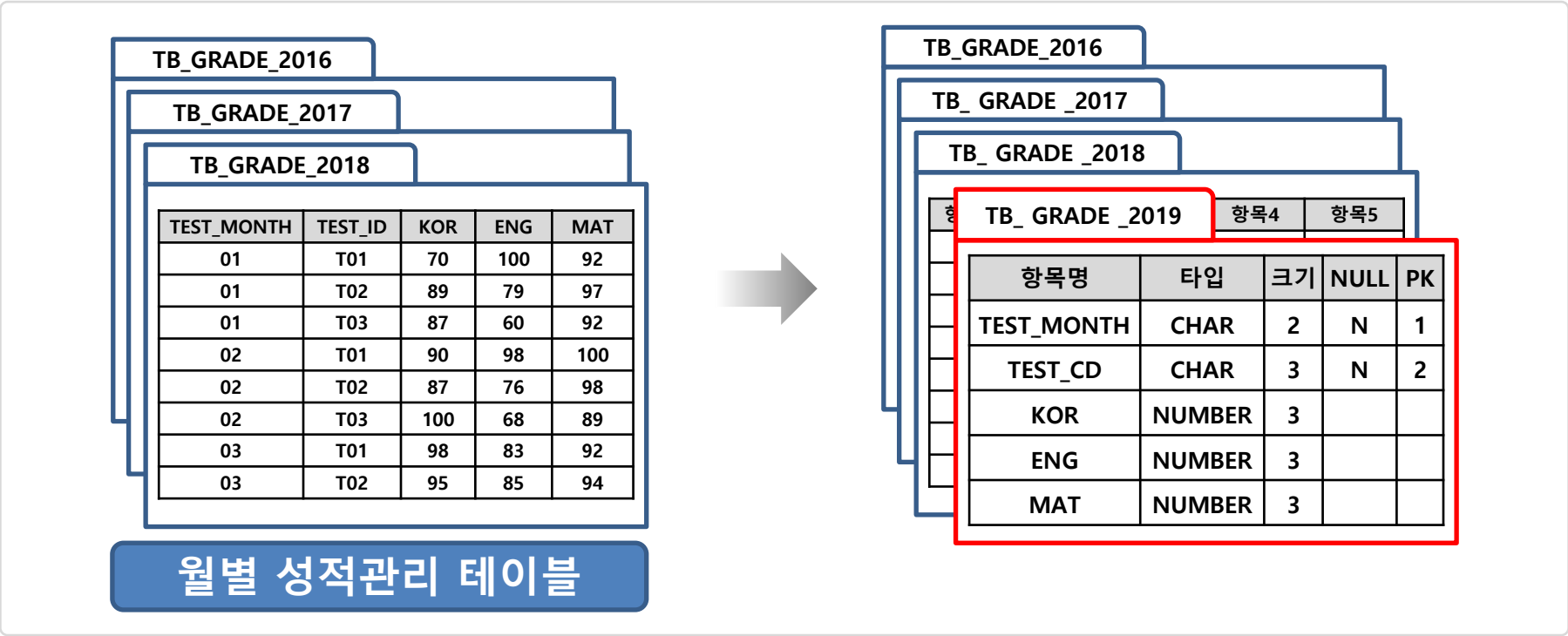
17.7. TRUNCATE TABLE

17.8. DROP TABLE

CREATE TABLE

KEYWORD : 테이블 생성

구 문	<pre>CREATE TABLE 테이블 (필드1 데이터타입, 필드2 데이터타입, 필드3 데이터타입, : 필드n 데이터타입);</pre>
제약조건	<ul style="list-style-type: none">■ PRIMARY KEY: 대표성을 가지는 기본키를 선언한다.<ul style="list-style-type: none">- 기본키는 UNIQUE + NOT NULL 속성이다.■ FOREIGN KEY: 외부 테이블의 필드와 연관하는 참조키를 선언한다.■ UNIQUE KEY: 모든 레코드 행에서 유일값 속성으로 선언한다.■ NOT NULL: 필드 입력값은 NULL 값을 허용하지 않는다.■ CHECK: 필드 입력값의 범위를 선언한다.■ DEFAULT: 필드 입력값이 없을 때 기본적으로 입력하는 값이다.



:: TB_GRADE_2019 테이블 생성

```
CREATE TABLE TB_GRADE_2019
(
    TEST_MONTH      CHAR(2 BYTE)          NOT NULL,    -- 월
    TEST_CD         CHAR(3 BYTE)          NOT NULL,    -- 시험코드
    KOR             NUMBER(3),             -- 국어
    ENG             NUMBER(3),             -- 영어
    MAT             NUMBER(3)              -- 수학
);

CREATE UNIQUE INDEX PK_TB_GRADE_2019 ON TB_GRADE_2019 (TEST_MONTH, TEST_CD);

ALTER TABLE TB_GRADE_2019 ADD (CONSTRAINT PK_TB_GRADE_2019 PRIMARY KEY (TEST_MONTH, TEST_CD));
```

결과

Table TB_GRADE_2019이 (가) 생성되었습니다.
INDEX PK_TB_GRADE_2019이 (가) 생성되었습니다.
Table TB_GRADE_2019이 (가) 변경되었습니다.

```
INSERT INTO TB_GRADE_2019 VALUES ( '01' , 'T01' , 92 , 98 , 84 );  
INSERT INTO TB_GRADE_2019 VALUES ( '01' , 'T02' , 84 , 94 , 90 );  
COMMIT;
```

결과

1 행 이 (가) 삽입되었습니다.
1 행 이 (가) 삽입되었습니다.
커밋 완료.

```
SELECT *  
FROM   TB_GRADE_2019;
```

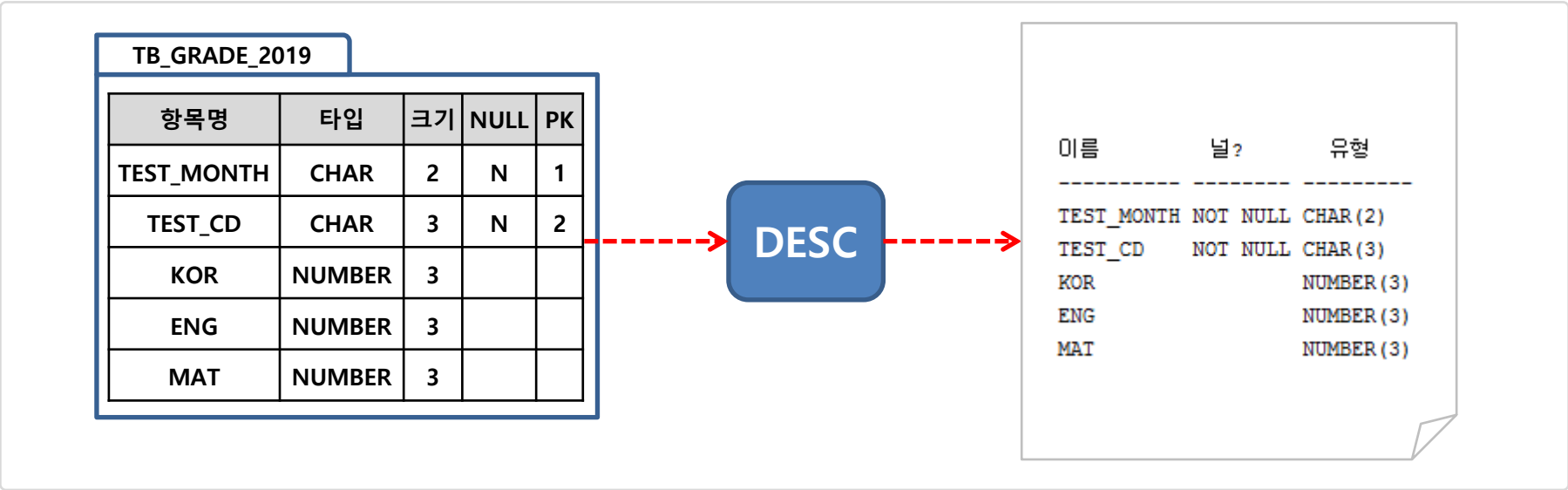
결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90

DESC와 DESCRIBE

KEYWORD : 테이블 구조 보기

구 문	[DESC DESCRIBE] 테이블;
사용 예	DESC TB_CUSTOMER;



:: TB_GRADE_2019 테이블 구조를 검색

```
DESC TB_GRADE_2019;
```

```
DESCRIBE TB_GRADE_2019;
```

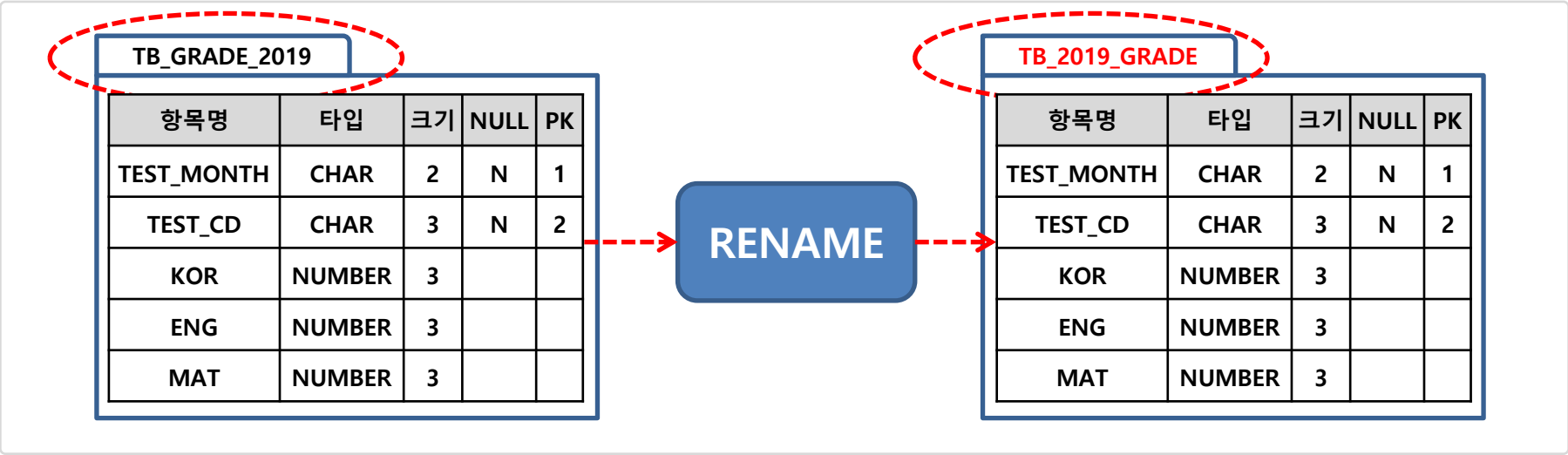
결과

이름	널?	유형
TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)

RENAME

KEYWORD : 테이블명 변경

구 문	RENAME 기존 테이블 TO 신규 테이블;
사용 예	RENAME TB_OLD_NM TO TB_NEW_NM;



:: TB_GRADE_2019 테이블명을 TB_2019_GRADE로 변경

```
SELECT *  
FROM   TB_GRADE_2019;
```

결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90

```
RENAME TB_GRADE_2019 TO TB_2019_GRADE;
```

결과

테이블 이름이 변경되었습니다.

```
SELECT *  
FROM   TB_GRADE_2019;
```

결과

```
ORA-00942: table or view does not exist  
00942, 00000 - "table or view does not exist"  
*Cause:  
*Action:  
1,979행, 8열에서 오류 발생
```

```
SELECT *  
FROM   TB_2019_GRADE;
```

결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90


```
RENAME TB_2019_GRADE TO TB_GRADE_2019;
```

결과

테이블 이름이 변경되었습니다.

```
SELECT *  
FROM   TB_GRADE_2019;
```

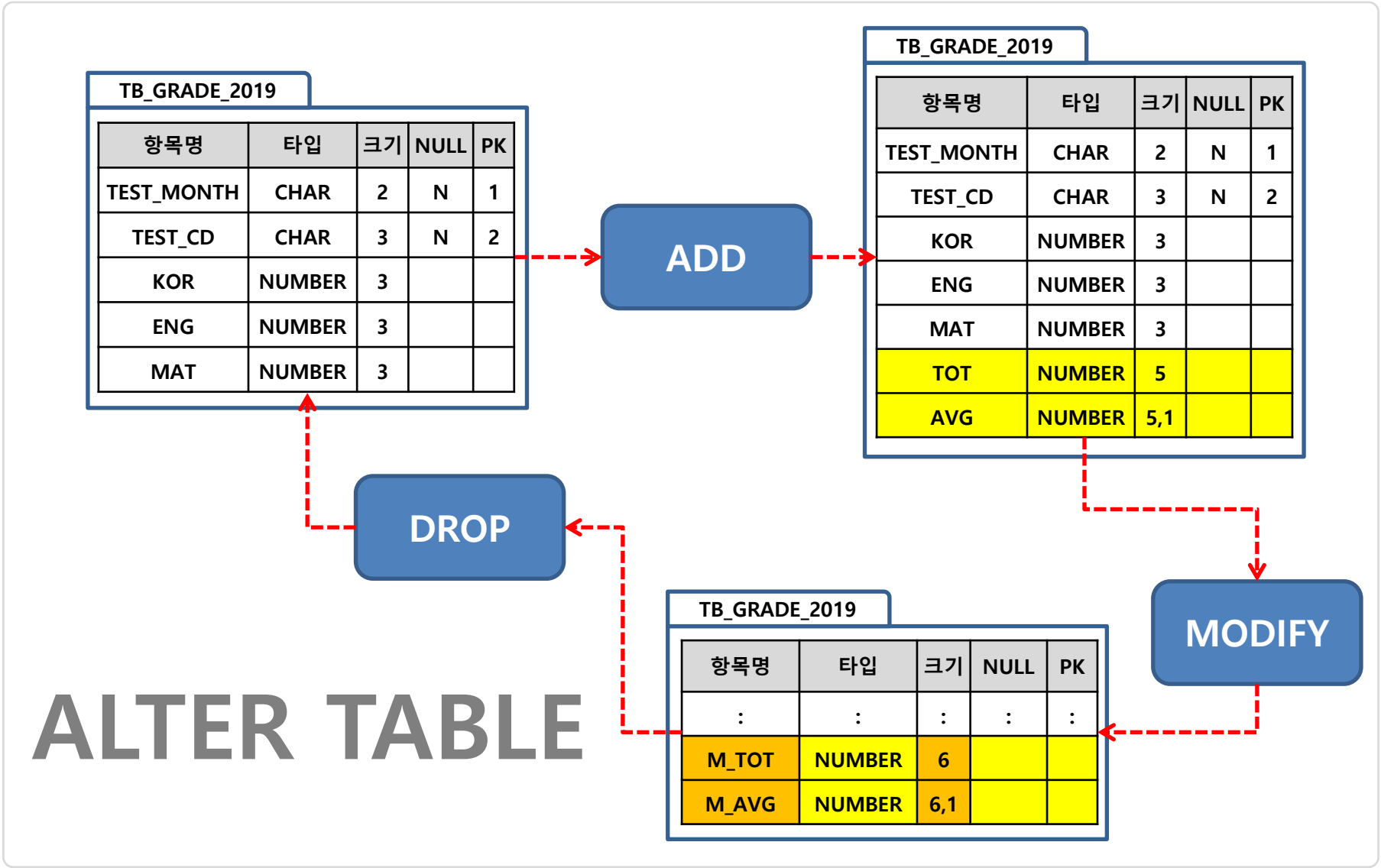
결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90

ALTER TABLE

KEYWORD : 테이블 속성 변경

구문 #1	ALTER TABLE 테이블 ADD (필드 [데이터타입]);
구문 #2	ALTER TABLE 테이블 MODIFY (필드 [데이터타입]);
구문 #3	ALTER TABLE 테이블 RENAME COLUMN 기존 필드 TO 변경 필드;
구문 #4	ALTER TABLE 테이블 DROP COLUMN 필드;
사용 예	ALTER TABLE TB_GRADE MODIFY (TOT NUMBER(4) NULL) ALTER TABLE TB_GRADE ADD(HIS NUMBER(3) NULL) ALTER TABLE TB_GRADE RENAME COLUMN HIS TO HISTORY ALTER TABLE TB_GRADE DROP COLUMN HISTORY



:: TB_GRADE_2019 테이블에 TOT, AVG 항목을 추가하고 속성을 변경

```
SELECT *  
FROM   TB_GRADE_2019;
```

결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90

```
DESC TB_GRADE_2019;
```

결과

이름	널?	유형
TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)

```
ALTER TABLE TB_GRADE_2019 ADD (TOT NUMBER(5) NULL);
```

결과

Table TB_GRADE_2019이 (가) 변경되었습니다.

```
ALTER TABLE TB_GRADE_2019 ADD (AVG NUMBER(5,1) NULL);
```

결과

Table TB_GRADE_2019이 (가) 변경되었습니다.


```
DESC TB_GRADE_2019;
```

결과

이름	널?	유형

TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)
TOT		NUMBER(5)
AVG		NUMBER(5,1)

```
ALTER TABLE TB_GRADE_2019 MODIFY (TOT NUMBER(6));  
ALTER TABLE TB_GRADE_2019 RENAME COLUMN TOT TO M_TOT;
```

결과

Table TB_GRADE_2019이 (가) 변경되었습니다.
Table TB_GRADE_2019이 (가) 변경되었습니다.

```
ALTER TABLE TB_GRADE_2019 MODIFY (AVG NUMBER(6,2));  
ALTER TABLE TB_GRADE_2019 RENAME COLUMN AVG TO M_AVG;
```

결과

Table TB_GRADE_2019이 (가) 변경되었습니다.
Table TB_GRADE_2019이 (가) 변경되었습니다.

```
DESC TB_GRADE_2019;
```

결과

이름	널?	유형
TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)
M_TOT		NUMBER(6)
M_AVG		NUMBER(6, 2)

```
ALTER TABLE TB_GRADE_2019 DROP COLUMN M_TOT;  
ALTER TABLE TB_GRADE_2019 DROP COLUMN M_AVG;
```

결과

Table TB_GRADE_2019이 (가) 변경되었습니다.
Table TB_GRADE_2019이 (가) 변경되었습니다.

```
DESC TB_GRADE_2019;
```

결과

이름	널?	유형
TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)

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

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
2017042	5	드라이기 구매 포인트 적립	3200	20190801234530
2017042	6	면도기 구매 포인트 적립	4500	20190819022005
2019001	1	등록 포인트 적립	500	20190102120720
2019001	2	믹서기 구매 포인트 적립	4600	20190405134554
2019001	3	드라이기 구매 포인트 적립	7820	20190829071234

:: TB_CUSTOMER 테이블의 CUSTOMER_CD와 TB_POINT 테이블의 CUSTOMER_CD를
참조키로 설정하여 1:N 관계를 만들고 CUSTOMER_CD가 '2017042'인 데이터를 삭제한다.

```
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

```
DELETE
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD = '2017042';
```

결과

1 행 이 (가) 삭제되었습니다.

```
ROLLBACK;
```

결과

롤백 완료.


```
ALTER TABLE TB_POINT ADD (CONSTRAINT FK_TB_POINT FOREIGN KEY (CUSTOMER_CD)
                            REFERENCES TB_CUSTOMER (CUSTOMER_CD));
```

결과

Table TB_POINT이 (가) 변경되었습니다.

```
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

```
DELETE
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD = '2017042';
```

결과

명령의 16 행에서 시작하는 중 오류 발생 -

```
DELETE
FROM   TB_CUSTOMER
WHERE  CUSTOMER_CD = '2017042'
```

오류 보고 -

```
ORA-02292: integrity constraint (SQLDB.FK_TB_POINT) violated - child record found
```

- * 두 테이블을 참조키(외래키)로 설정하면 1:N 관계를 가지게 되며 부모:자식 관계를 형성한다.
- * 부모 테이블의 데이터를 삭제하려면 자식 테이블에 키로 연관된 값이 없어야 한다.

```
ALTER TABLE TB_POINT DROP CONSTRAINT FK_TB_POINT;
```

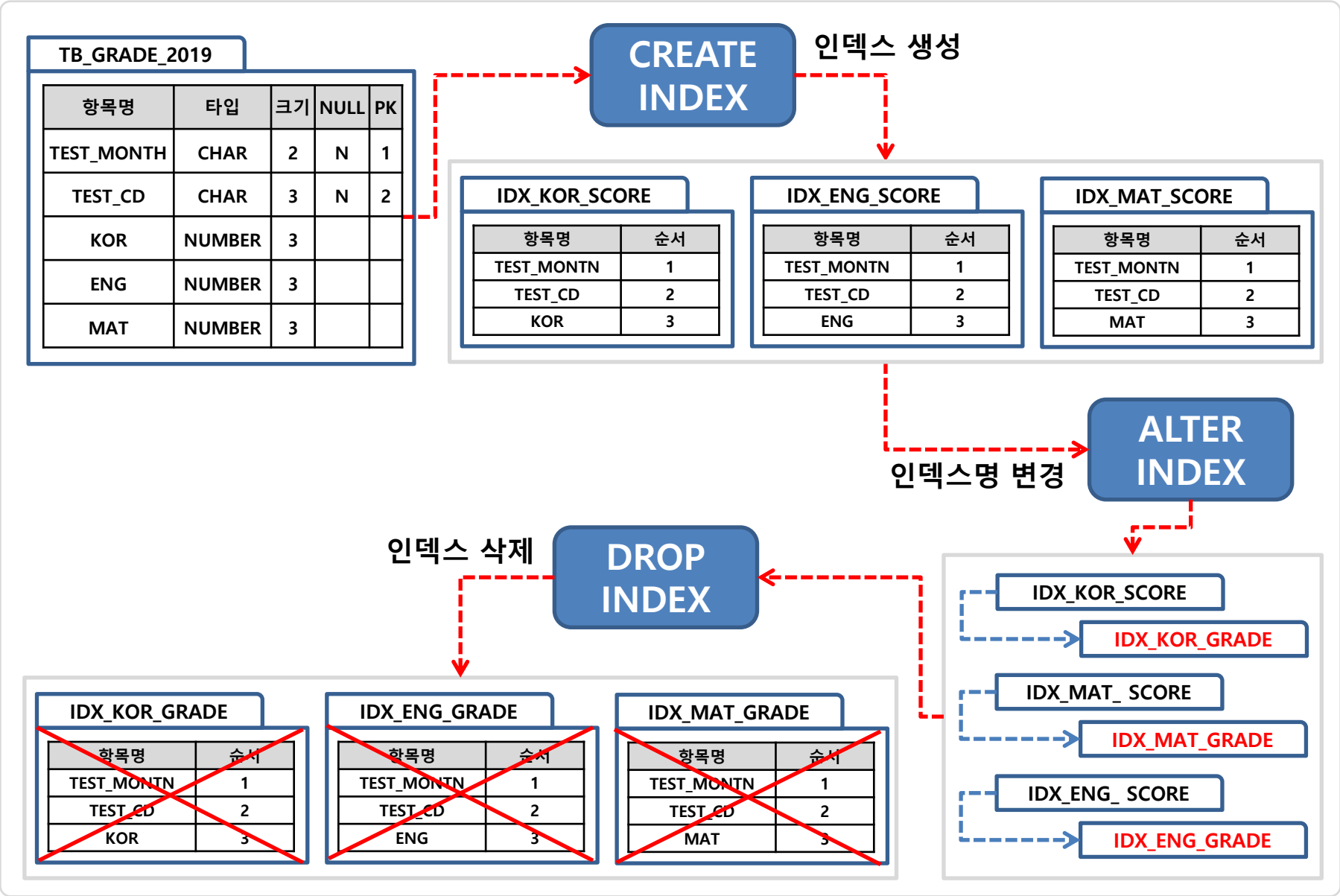
결과

Table TB_POINT이 (가) 변경되었습니다.

INDEX

KEYWORD : 인덱스의 생성, 수정, 삭제 작업

구문 #1	CREATE [UNIQUE] INDEX 인덱스 ON 테이블(필드1, 필드2, 필드3 ..., 필드n);
구문 #2	ALTER INDEX 현재 인덱스 RENAME TO 신규 인덱스;
구문 #3	DROP INDEX 인덱스;
사용 예	CREATE INDEX IDX_TEST ON TB_TEST(TEST_CD, SQNO) ALTER INDEX IDX_TEST TO IDX_EXAM DROP INDEX IDX_EXAM



∴ tb_grade_2019 테이블에 인덱스를 생성하고 인덱스명을 변경하고 삭제

```
SELECT INDEX_NAME,  
       COLUMN_NAME,  
       COLUMN_POSITION,  
       CHAR_LENGTH,  
       DESCEND  
FROM   USER_IND_COLUMNS  
WHERE  TABLE_NAME = 'TB_GRADE_2019';
```

결과

INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	CHAR_LENGTH	DESCEND
PK_TB_GRADE_2019	TEST_MONTH	1	2	ASC
PK_TB_GRADE_2019	TEST_CD	2	3	ASC

```
CREATE INDEX IDX_KOR_SCORE ON TB_GRADE_2019 (TEST_MONTH, TEST_CD, KOR);  
CREATE INDEX IDX_ENG_SCORE ON TB_GRADE_2019 (TEST_MONTH, TEST_CD, ENG);  
CREATE INDEX IDX_MAT_SCORE ON TB_GRADE_2019 (TEST_MONTH, TEST_CD, MAT);
```

결과

Index `IDX_KOR_SCORE`이 (가) 생성되었습니다.

Index `IDX_ENG_SCORE`이 (가) 생성되었습니다.

Index `IDX_MAT_SCORE`이 (가) 생성되었습니다.

```
SELECT INDEX_NAME,  
       COLUMN_NAME,  
       COLUMN_POSITION,  
       CHAR_LENGTH,  
       DESCEND  
FROM   USER_IND_COLUMNS  
WHERE  TABLE_NAME = 'TB_GRADE_2019';
```

결과

INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	CHAR_LENGTH	DESCEND
PK_TB_GRADE_2019	TEST_MONTH	1	2	ASC
PK_TB_GRADE_2019	TEST_CD	2	3	ASC
IDX_KOR_SCORE	TEST_MONTH	1	2	ASC
IDX_KOR_SCORE	TEST_CD	2	3	ASC
IDX_KOR_SCORE	KOR	3	0	ASC
IDX_ENG_SCORE	TEST_MONTH	1	2	ASC
IDX_ENG_SCORE	TEST_CD	2	3	ASC
IDX_ENG_SCORE	ENG	3	0	ASC
IDX_MAT_SCORE	TEST_MONTH	1	2	ASC
IDX_MAT_SCORE	TEST_CD	2	3	ASC
IDX_MAT_SCORE	MAT	3	0	ASC

```
ALTER INDEX IDX_KOR_SCORE RENAME TO IDX_KOR_GRADE;  
ALTER INDEX IDX_ENG_SCORE RENAME TO IDX_ENG_GRADE;  
ALTER INDEX IDX_MAT_SCORE RENAME TO IDX_MAT_GRADE;
```

결과

Index `IDX_KOR_SCORE`이 (가) 변경되었습니다.

Index `IDX_ENG_SCORE`이 (가) 변경되었습니다.

Index `IDX_MAT_SCORE`이 (가) 변경되었습니다.


```
SELECT  INDEX_NAME,
        COLUMN_NAME,
        COLUMN_POSITION,
        CHAR_LENGTH,
        DESCEND
FROM    USER_IND_COLUMNS
WHERE   TABLE_NAME = 'TB_GRADE_2019';
```

결과

INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	CHAR_LENGTH	DESCEND
PK_TB_GRADE_2019	TEST_MONTH	1	2	ASC
PK_TB_GRADE_2019	TEST_CD	2	3	ASC
IDX_KOR_GRADE	TEST_MONTH	1	2	ASC
IDX_KOR_GRADE	TEST_CD	2	3	ASC
IDX_KOR_GRADE	KOR	3	0	ASC
IDX_ENG_GRADE	TEST_MONTH	1	2	ASC
IDX_ENG_GRADE	TEST_CD	2	3	ASC
IDX_ENG_GRADE	ENG	3	0	ASC
IDX_MAT_GRADE	TEST_MONTH	1	2	ASC
IDX_MAT_GRADE	TEST_CD	2	3	ASC
IDX_MAT_GRADE	MAT	3	0	ASC

```
DROP INDEX IDX_KOR_GRADE;  
DROP INDEX IDX_ENG_GRADE;  
DROP INDEX IDX_MAT_GRADE;
```

결과

Index `IDX_KOR_GRADE`이 (가) 삭제되었습니다.

Index `IDX_ENG_GRADE`이 (가) 삭제되었습니다.

Index `IDX_MAT_GRADE`이 (가) 삭제되었습니다.

```
SELECT INDEX_NAME,  
       COLUMN_NAME,  
       COLUMN_POSITION,  
       CHAR_LENGTH,  
       DESCEND  
FROM   USER_IND_COLUMNS  
WHERE  TABLE_NAME = 'TB_GRADE_2019';
```

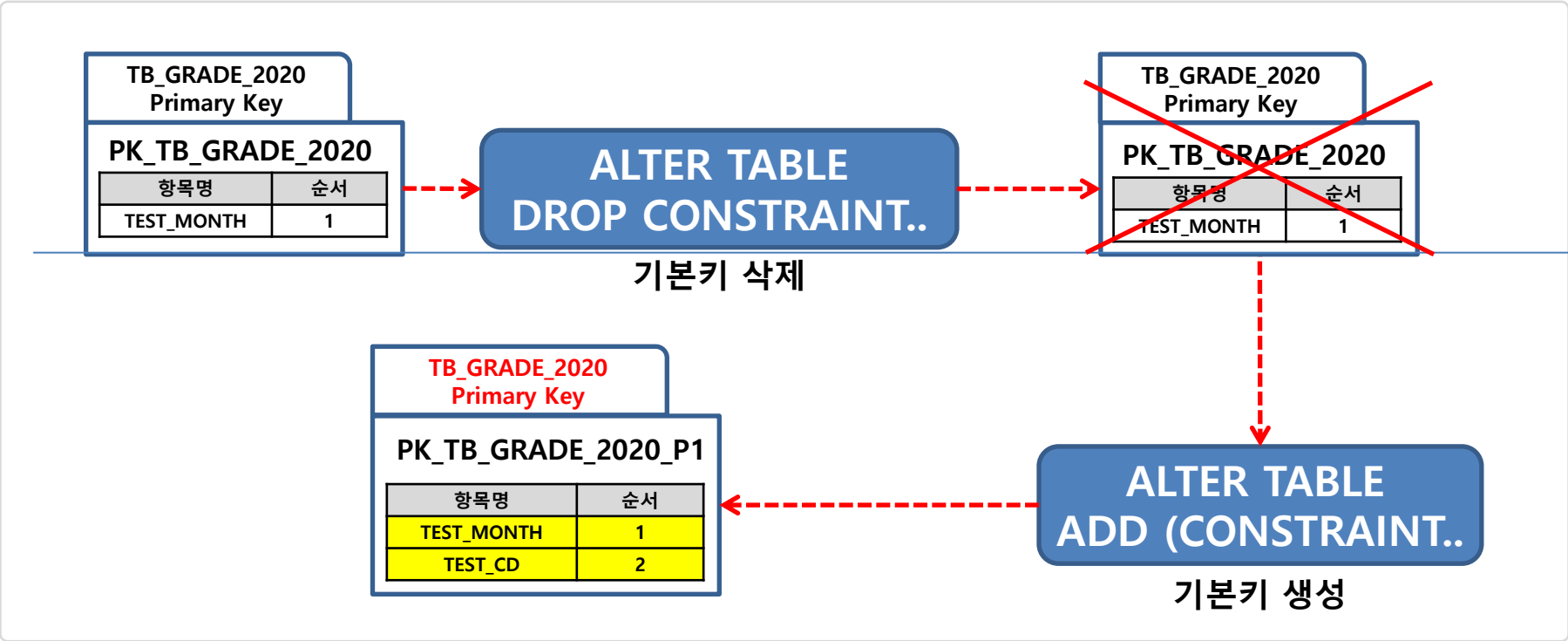
결과

INDEX_NAME	COLUMN_NAME	COLUMN_POSITION	CHAR_LENGTH	DESCEND
PK_TB_GRADE_2019	TEST_MONTH	1	2	ASC
PK_TB_GRADE_2019	TEST_CD	2	3	ASC

PRIMARY KEY

KEYWORD : 기본키의 생성, 삭제 작업

구문 #1	ALTER TABLE 테이블 ADD (CONSTRAINT 기본키 PRIMARY KEY (필드1, 필드2, 필드3 ..., 필드n));
구문 #2	ALTER TABLE 테이블 DROP CONSTRAINT 기본키;
사용 예	ALTER TABLE TB_TEST ADD (CONSTRAINT PK_TEST PRIMARY KEY (TEST_CD, SQNO)) ALTER TABLE TB_TEST DROP CONSTRAINT PK_TEST



:: TB_GRADE_2020 테이블의 기본키를 삭제하고 다시 생성

```
SELECT A.CONSTRAINT_NAME,  
       A.CONSTRAINT_TYPE,  
       A.TABLE_NAME,  
       B.POSITION,  
       B.COLUMN_NAME  
FROM   SYS.USER_CONSTRAINTS A,  
       SYS.USER_CONS_COLUMNS B  
WHERE  A.CONSTRAINT_NAME = B.CONSTRAINT_NAME  
AND    A.TABLE_NAME = 'TB_GRADE_2020'  
AND    A.CONSTRAINT_TYPE = 'P';
```

결과

CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	POSITION	COLUMN_NAME
PK_TB_GRADE_2020	P	TB_GRADE_2020	1	TEST_MONTH

```
ALTER TABLE TB_GRADE_2020 DROP CONSTRAINT PK_TB_GRADE_2020;
```

결과

Table TB_GRADE_2020이 (가) 변경되었습니다.

```
SELECT A.CONSTRAINT_NAME,  
       A.CONSTRAINT_TYPE,  
       A.TABLE_NAME,  
       B.POSITION,  
       B.COLUMN_NAME  
FROM   SYS.USER_CONSTRAINTS A,  
       SYS.USER_CONS_COLUMNS B  
WHERE  A.CONSTRAINT_NAME = B.CONSTRAINT_NAME  
AND    A.TABLE_NAME = 'TB_GRADE_2020'  
AND    A.CONSTRAINT_TYPE = 'P';
```

결과

⚡ CONSTRAINT_NAME	⚡ CONSTRAINT_TYPE	⚡ TABLE_NAME	⚡ POSITION	⚡ COLUMN_NAME
-------------------	-------------------	--------------	------------	---------------


```
ALTER TABLE TB_GRADE_2020 ADD (CONSTRAINT PK_TB_GRADE_2020_P1 PRIMARY KEY (TEST_MONTH,
TEST_CD));
```

결과

Table TB_GRADE_2020이 (가) 변경되었습니다.

```
SELECT A.CONSTRAINT_NAME,  
       A.CONSTRAINT_TYPE,  
       A.TABLE_NAME,  
       B.POSITION,  
       B.COLUMN_NAME  
FROM   SYS.USER_CONSTRAINTS A,  
       SYS.USER_CONS_COLUMNS B  
WHERE  A.CONSTRAINT_NAME = B.CONSTRAINT_NAME  
AND    A.TABLE_NAME = 'TB_GRADE_2020'  
AND    A.CONSTRAINT_TYPE = 'P';
```

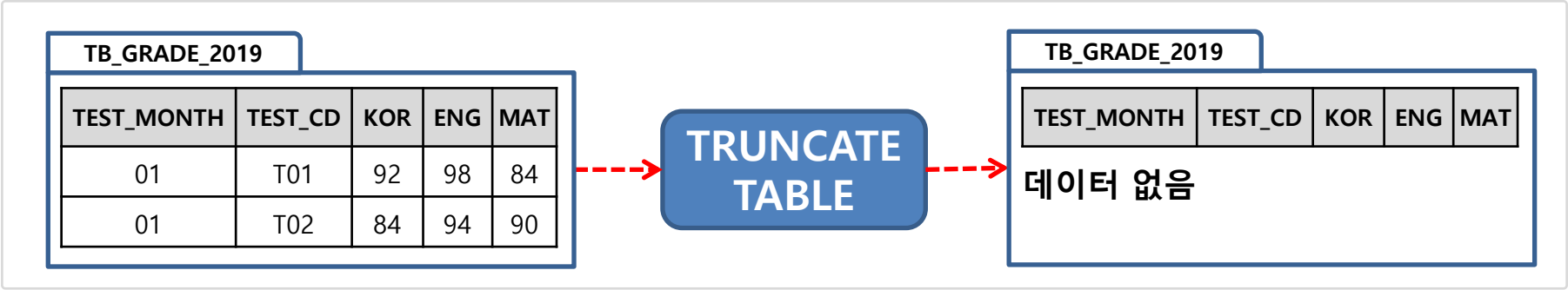
결과

CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	POSITION	COLUMN_NAME
PK_TB_GRADE_2020_P1	P	TB_GRADE_2020	1	TEST_MONTH
PK_TB_GRADE_2020_P1	P	TB_GRADE_2020	2	TEST_CD

TRUNCATE TABLE

KEYWORD : 복구 정보 없이 데이터 전체 삭제

구 문	TRUNCATE TABLE 테이블;
-----	---------------------



:: 복구 정보를 만들지 않고 TB_GRADE_2019 테이블의 데이터를 삭제

```
SELECT *  
FROM   TB_GRADE_2019;
```

결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
01	T01	92	98	84
01	T02	84	94	90

```
TRUNCATE TABLE TB_GRADE_2019;
```

결과

Table TB_GRADE_2019이 (가) 잘렸습니다.

```
SELECT *  
FROM   TB_GRADE_2019;
```

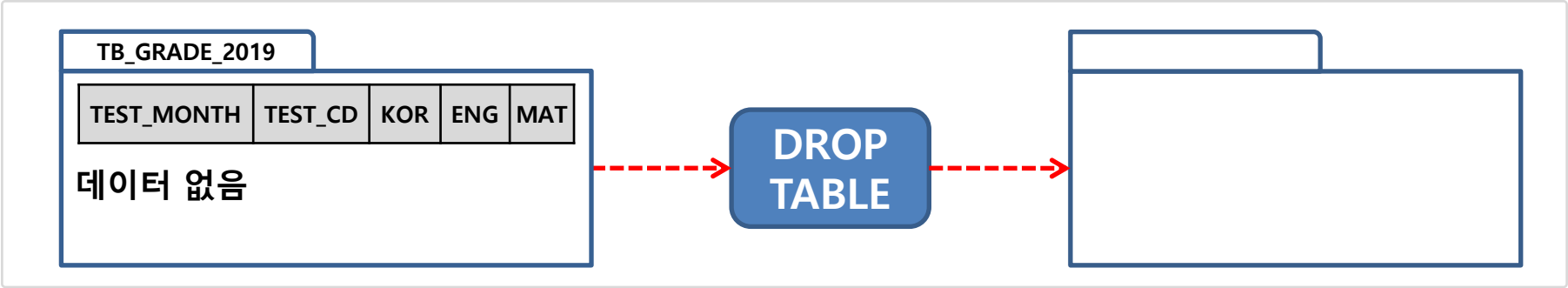
결과

TEST_MONTH	TEST_CD	KOR	ENG	MAT
------------	---------	-----	-----	-----

DROP TABLE

KEYWORD : 테이블 삭제

구문 #1	DROP TABLE 테이블 [CASCADE CONSTRAINTS];
구문 #2	DROP TABLE 테이블 PURGE;
구문 #3	FLASHBACK TABLE 테이블 TO BEFORE DROP;
구문 #4	PURGE RECYCLEBIN;
사용 예	DROP TABLE TB_TEST CASCADE CONSTRAINTS DROP TABLE TB_TEST PURGE FLASHBACK TABLE TB_TEST TO BEFORE DROP PURGE RECYCLEBIN



:: TB_GRADE_2019 테이블을 삭제

```
DROP TABLE TB_GRADE_2019;
```

결과

Table TB_GRADE_2019이 (가) 삭제되었습니다.

```
DESC TB_GRADE_2019;
```

결과

오류:
ORA-04043: tb_grade_2019 객체가 존재하지 않습니다.

```
FLASHBACK TABLE TB_GRADE_2019 TO BEFORE DROP;
```

결과

Flashback을 (를) 성공했습니다.

```
DESC TB_GRADE_2019;
```

결과

이름	널?	유형

TEST_MONTH	NOT NULL	CHAR(2)
TEST_CD	NOT NULL	CHAR(3)
KOR		NUMBER(3)
ENG		NUMBER(3)
MAT		NUMBER(3)

```
DROP TABLE TB_GRADE_2019;
```

결과

Table TB_GRADE_2019이 (가) 삭제되었습니다.

```
PURGE RECYCLEBIN;
```

결과

RECYCLEBIN이 (가) 비워졌습니다.

```
FLASHBACK TABLE TB_GRADE_2019 TO BEFORE DROP;
```

결과

명령의 2,204 행에서 시작하는 중 오류 발생 -

```
FLASHBACK TABLE tb_grade_2019 TO BEFORE DROP
```

오류 보고 -

```
ORA-38305: object not in RECYCLE BIN
```

```
38305. 00000 - "object not in RECYCLE BIN"
```

```
*Cause:      Trying to Flashback Drop an object which is not in RecycleBin.
```

```
*Action:     Only the objects in RecycleBin can be Flashback Dropped.
```



```
DESC TB_GRADE_2019;
```

결과

오류:
ORA-04043: tb_grade_2019 객체가 존재하지 않습니다.