

18장. 사용자 기능

18.1. 함수(FUNCTION)

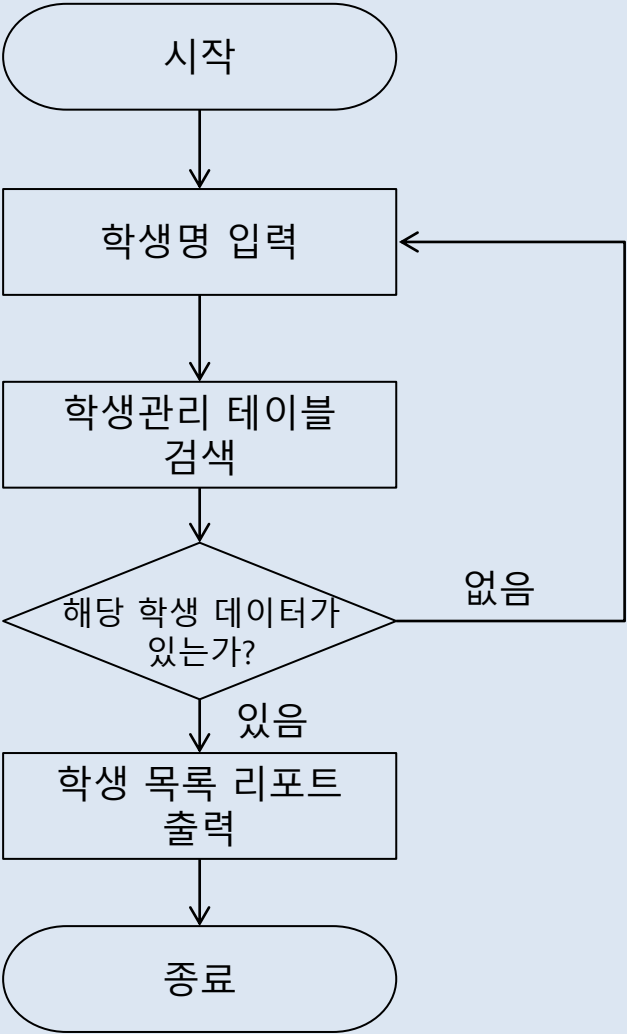
18.2. 프로시저(PRODECURE)

18.3. 트리거(TRIGGER)

18.4. 뷰(VIEW)

18.5. 시퀀스(SEQUENCE)

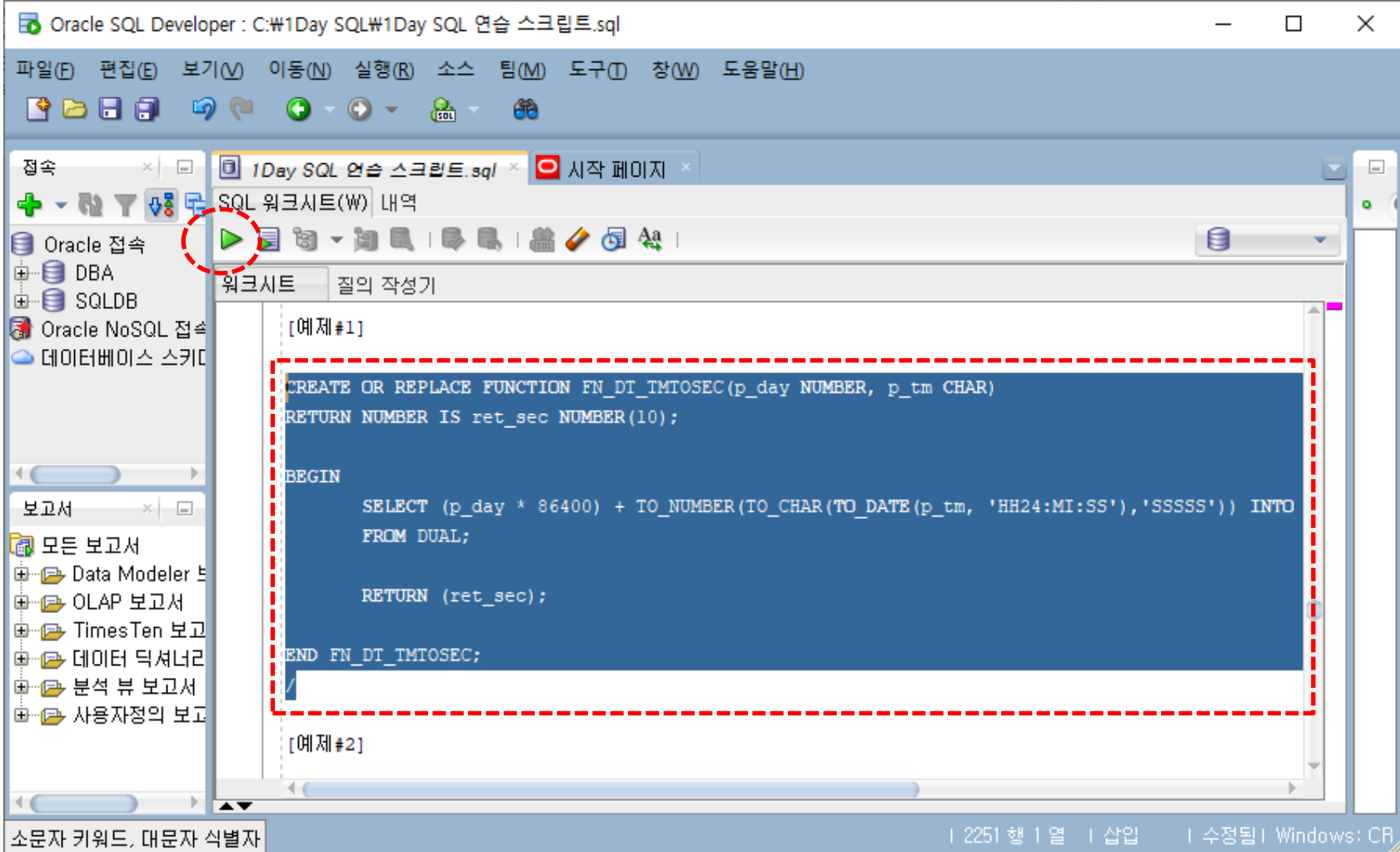
학생 목록 리포트를 출력하는 절차

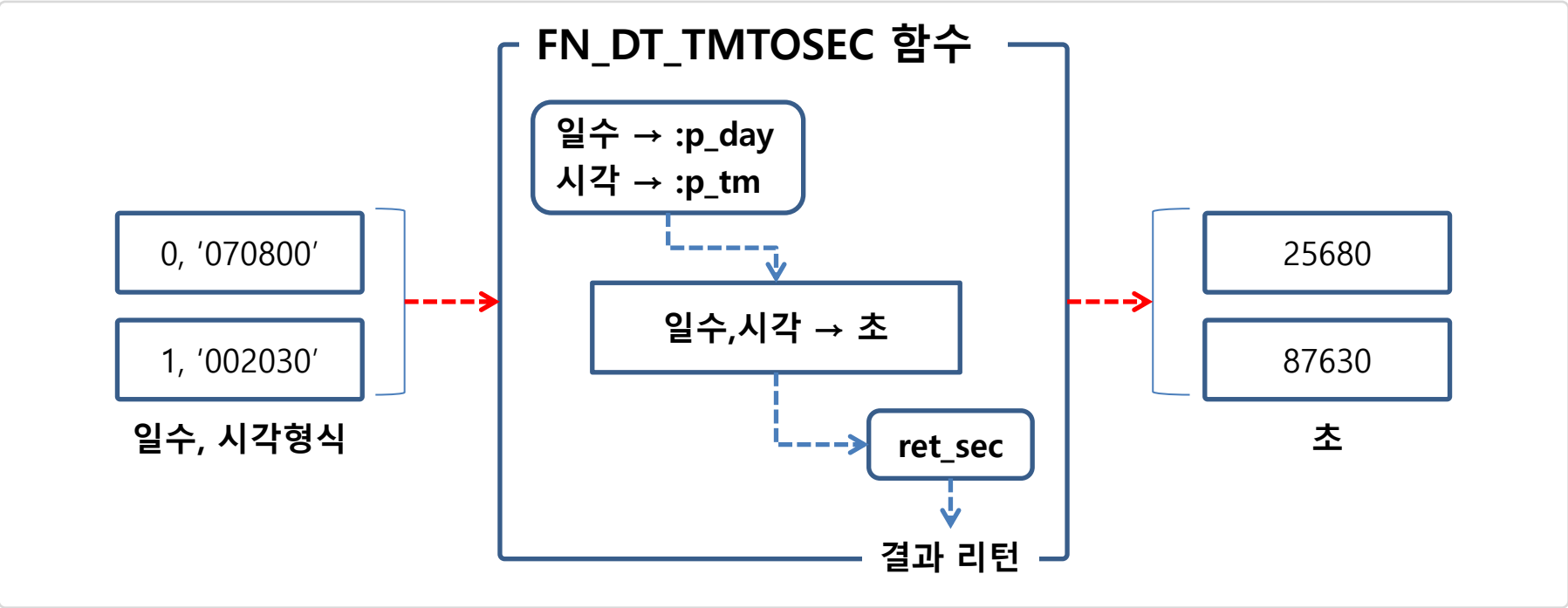


함수 (FUNCTION)

KEYWORD : 결과 값을 가지는 기능 정의

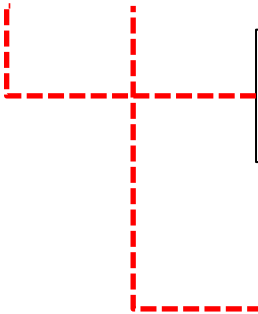
구문 #1	<pre>CREATE [OR REPLACE] FUNCTION 함수명(매개변수1 타입1, 매개변수2 타입2 ...,) RETURN 반환타입 IS 변수 변수타입(크기); BEGIN 실행 SQL 구문 INTO 변수; RETURN (변수); [EXCEPTION 예외처리;] END 함수명; /</pre>
구문 #2	<pre>DROP FUNCTION 함수명;</pre>





:: 수치형 일수와 시각형식의 6자리 문자열을 입력하여 초를 구하는 함수를 생성합니다.

FN_DT_TMTOSEC(1, '002030')



1 * 86400 = 86400
결과 : 86,400

TO_DATE('002030') → 시간: 20분30초
TO_CHAR(20분30초,'SSSSS') → 문자열: '01230'
TO_NUMBER('01230') → 수치: 1230
결과: 1,230

86,400 + 1,230 = 87,630(초)

:: FN_DT_TMTOSEC() 함수에 1, '002030' 매개변수를 입력한 결과

```
CREATE OR REPLACE FUNCTION FN_DT_TMTOSEC(p_day NUMBER, p_tm CHAR)
RETURN NUMBER IS ret_sec NUMBER(10);

BEGIN
    SELECT (p_day * 86400) + TO_NUMBER(TO_CHAR(TO_DATE(p_tm,
        'HH24:MI:SS'), 'SSSSS')) INTO ret_sec
    FROM DUAL;

    RETURN (ret_sec);

END FN_DT_TMTOSEC;
/
```

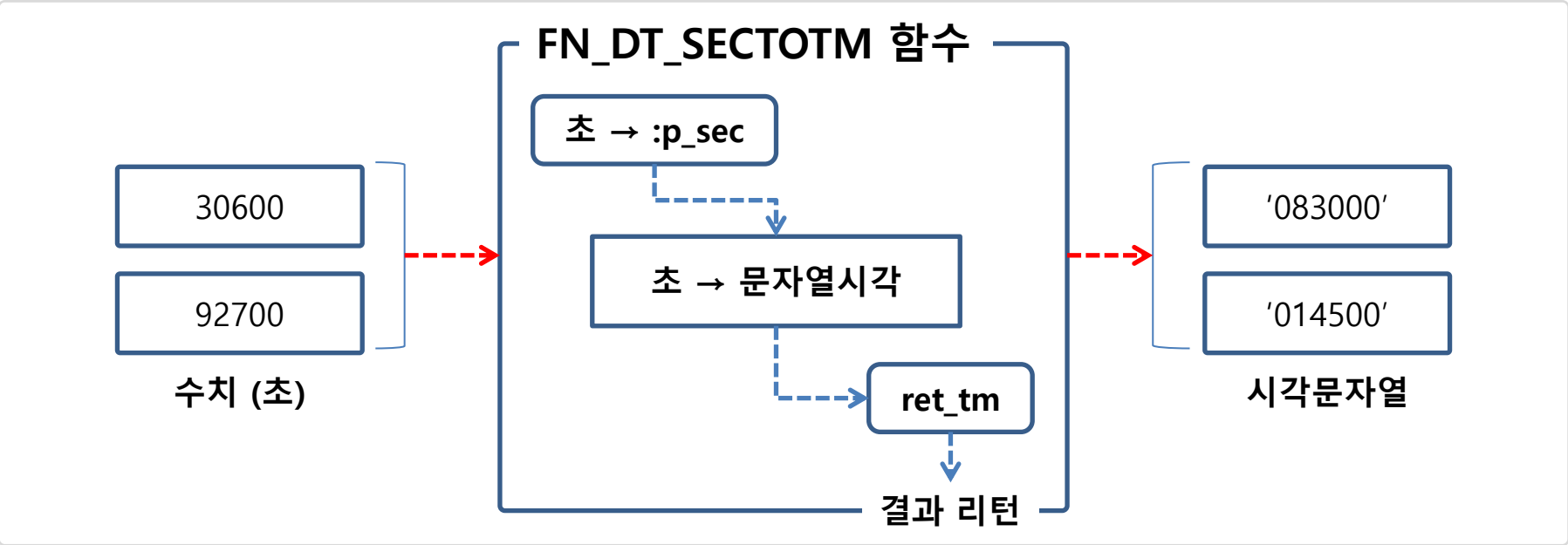
결과

Function **FN_DT_TMTOSEC**이 (가) 컴파일되었습니다.


```
SELECT FN_DT_TMTOSEC(0, '070800'),  
       FN_DT_TMTOSEC(1, '002030'),  
       TO_CHAR(SYSDATE, 'HH24:MI:SS') AS CUR_TM,  
       FN_DT_TMTOSEC(0, TO_CHAR(SYSDATE, 'HH24MISS')) AS CUR_SEC  
FROM   DUAL;
```

결과

FN_DT_TMTOSEC(0,'070800')	FN_DT_TMTOSEC(1,'002030')	CUR_TM	CUR_SEC
25680	87630	15:45:56	56756



:: 수치(초)를 입력하여 6자리 시각형식의 문자열을 취득하는 함수를 생성합니다.

FN_DT_SECTOTM(92700)

1.75시간

45.0분

0.0초

MOD(92700,86400) → 수치: 6300 (몫은 1)

FLOOR(6300/3600) → 수치: 1 (1.75에서 버림함)

LPAD(1,'0') → 문자열: '01'

결괏값: '01'

3600으로 나눠 시간을 구함

MOD(92700,86400) → 수치: 6300

MOD(6300,3600) → 수치: 2700

FLOOR(2700/60) → 수치: 45 (45.0에서 버림함)

LPAD(45,'0') → 문자열: '45'

결괏값: '45'

3600으로 나눈 나머지 값을 60으로 나눠 분을 구함

MOD(92700,86400) → 수치: 6300

MOD(6300,3600) → 수치: 2700

MOD(2700,60) → 수치: 0

FLOOR(0) → 수치: 0 (0.0에서 버림함)

LPAD(0,'0') → 문자열: '00'

결괏값: '00'

3600으로 나눈 나머지 값을 다시 60으로 나눈 나머지 값을 60으로 나눠 초를 구함

'01' || '45' || '00' = '014500' (문자열)

:: FN_DT_SECTOTM() 함수에 92700 매개변수를 입력한 결과

```
CREATE OR REPLACE FUNCTION FN_DT_SECTOTM(p_sec NUMBER)
RETURN CHAR IS ret_tm CHAR(6);

BEGIN

    SELECT (
        LPAD(FLOOR(MOD(p_sec,86400)/3600),2,'0') ||
        LPAD(FLOOR(MOD(MOD(p_sec,86400),3600)/60),2,'0') ||
        LPAD(FLOOR(MOD(MOD(MOD(p_sec,86400),3600),60)),2,'0')
    ) INTO ret_tm
FROM    DUAL;

RETURN (ret_tm);

END FN_DT_SECTOTM;
/
```

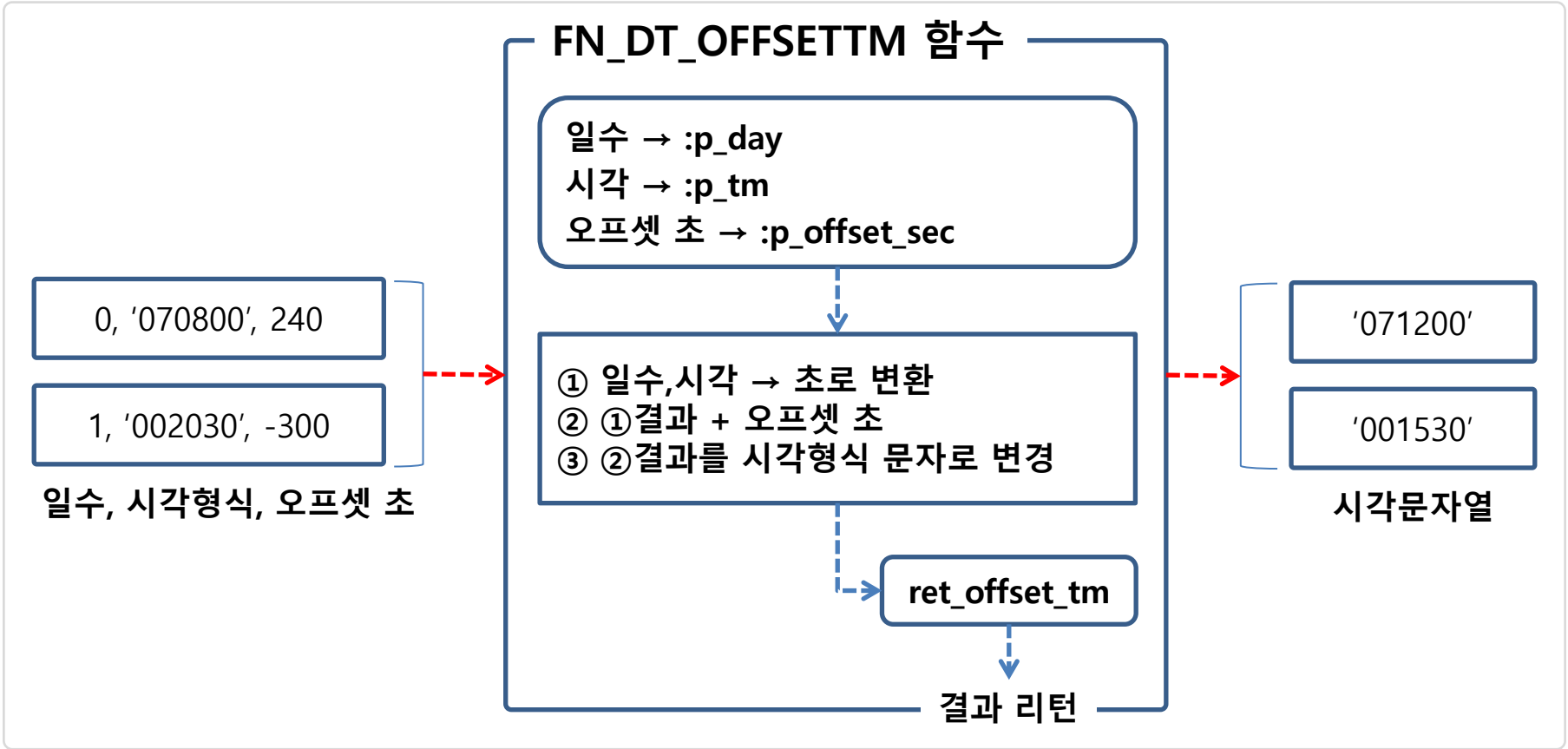
결과

Function FN_DT_TMTIOSEC이 (가) 컴파일되었습니다.

```
SELECT TO_CHAR(SYSDATE, 'HH24:MI:SS') AS CUR_TM,  
       FN_DT_SECTOTM(FN_DT_TMTOSEC(0, TO_CHAR(SYSDATE, 'HH24MISS'))-300)  
       AS MINUS_SEC,  
       FN_DT_SECTOTM(FN_DT_TMTOSEC(0, TO_CHAR(SYSDATE, 'HH24MISS'))+300)  
       AS PLUS_SEC  
FROM   DUAL;
```

결과

CUR_TM	MINUS_SEC	PLUS_SEC
16:34:17	162917	163917



:: 기준시간에서 이동한 초를 계산하여 6자리 시각형식의 문자열로 취득하는 함수를 생성합니다.

FN_DT_OFFSETTM(1,'002030',-300)

1 * 86400 = 86400
결과 : 86,400

TO_DATE('002030') → 시간: 20분30초
TO_CHAR(20분30초,'SSSSS') → 문자열: '01230'
TO_NUMBER('01230') → 수치: 1230
결과: 1,230

-300

86,400 + 1,230 - 300 = 87,330(초)



87,300초를 'HHMMDD' 형식 문자열로 변환 = '001530' (문자열)

:: FN_DT_OFFSETM() 함수에 1, '002030', -300 매개변수를 입력한 결과

```
CREATE OR REPLACE FUNCTION FN_DT_OFFSETTM(p_day NUMBER,
                                           p_tm CHAR,
                                           p_offset_sec NUMBER
                                           )
RETURN CHAR IS ret_offset_tm CHAR(6);

BEGIN

    SELECT (
        LPAD(FLOOR(MOD(S1.SEC,86400)/3600),2,'0') ||
        LPAD(FLOOR(MOD(MOD(S1.SEC,86400),3600)/60),2,'0') ||
        LPAD(FLOOR(MOD(MOD(MOD(S1.SEC,86400),3600),60)),2,'0')
    ) INTO ret_offset_tm
FROM    (
    SELECT (p_day * 86400) + TO_NUMBER(TO_CHAR(TO_DATE(p_tm,
        'HH24:MI:SS'),'SSSSS')) + (p_offset_sec) AS SEC
    FROM    DUAL
    )S1;

    RETURN (ret_offset_tm);

END FN_DT_OFFSETTM;
/
```

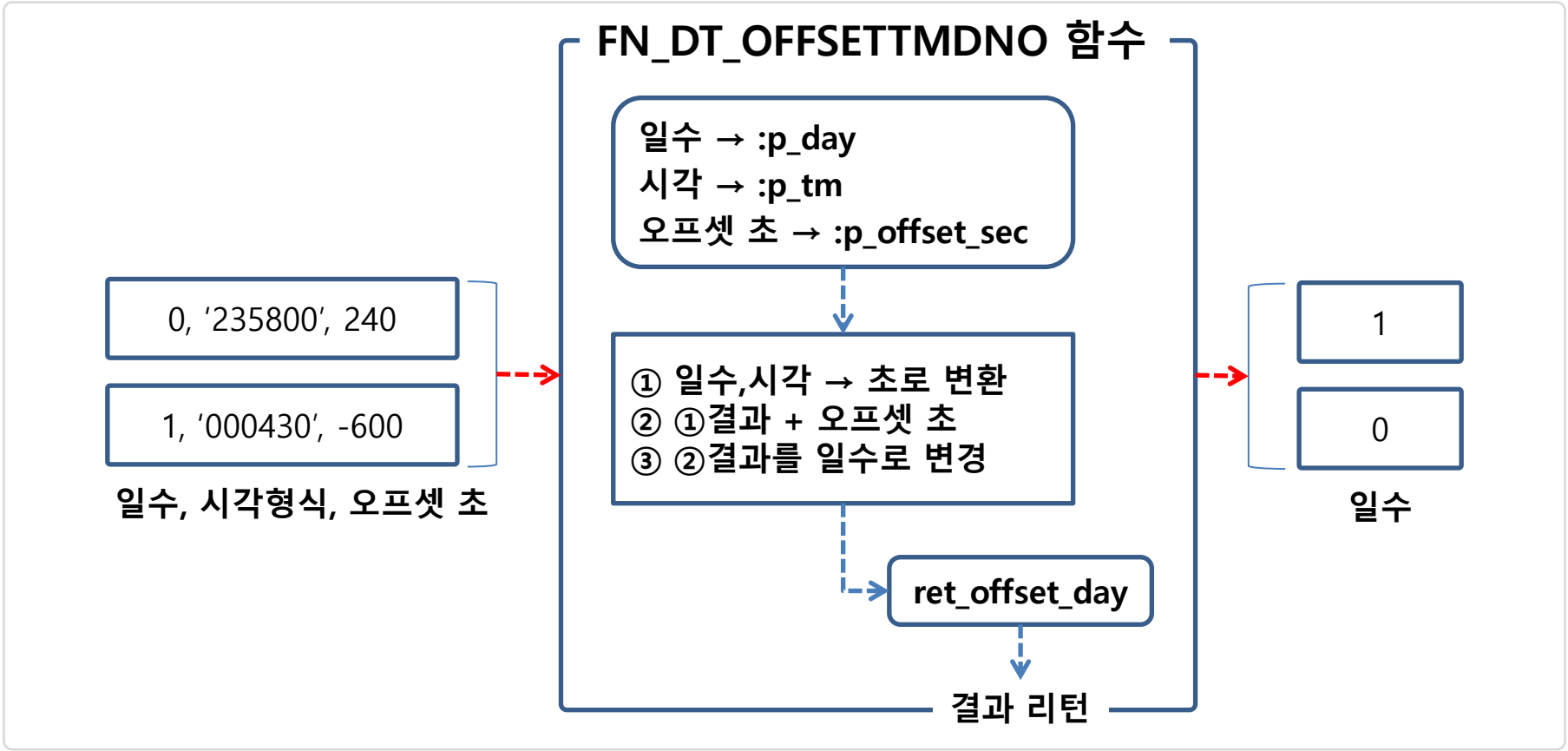

결과

Function `FN_DI_OFFSETTM`이 (가) 컴파일되었습니다.

```
SELECT TO_CHAR(SYSDATE, 'HH24:MI:SS') AS CUR_TM,  
       FN_DT_OFFSETTM(0, TO_CHAR(SYSDATE, 'HH24MISS'), -300)  
       AS MINUS_SEC,  
       FN_DT_OFFSETTM(0, TO_CHAR(SYSDATE, 'HH24MISS'), 300)  
       AS PLUS_SEC  
FROM   DUAL;
```

결과

CUR_TM	MINUS_SEC	PLUS_SEC
17:22:15	171715	172715



∴ 기준시간에서 이동한 초의 일수를 취득하는 함수를 생성합니다.

FN_DT_OFFSETTMDNO (0,'235800',240)

0 * 86400 = 0
결과 : 0

TO_DATE('235800') → 시간: 23시58분00초
TO_CHAR(4분30초,'SSSSS') → 문자열: '86280'
TO_NUMBER('86,280') → 수치: 86,280
결과: 86,280

240

0 + 86,280 + 240 = 86,520 (초)



86,520초 1일에 해당하는 86,400초로 나눈값을 소수점 버림 = 1 (일)

:: FN_DT_OFFSETTMDNO() 함수에 0, '235800', 200 매개변수를 입력한 결과

```
CREATE OR REPLACE FUNCTION FN_DT_OFFSETTMDNO(p_day NUMBER,  
                                              p_tm CHAR,  
                                              p_offset_sec NUMBER  
                                              )  
RETURN NUMBER IS ret_offset_dno NUMBER(3);  
  
BEGIN  
  
    SELECT FLOOR(S1.SEC/86400) INTO ret_offset_dno  
    FROM    (  
        SELECT (p_day * 86400) + TO_NUMBER(TO_CHAR(TO_DATE(p_tm,  
            'HH24:MI:SS'),'SSSSS')) + (p_offset_sec) AS SEC  
        FROM    DUAL  
    )S1;  
  
    RETURN (ret_offset_dno);  
  
END FN_DT_OFFSETTMDNO;  
/
```

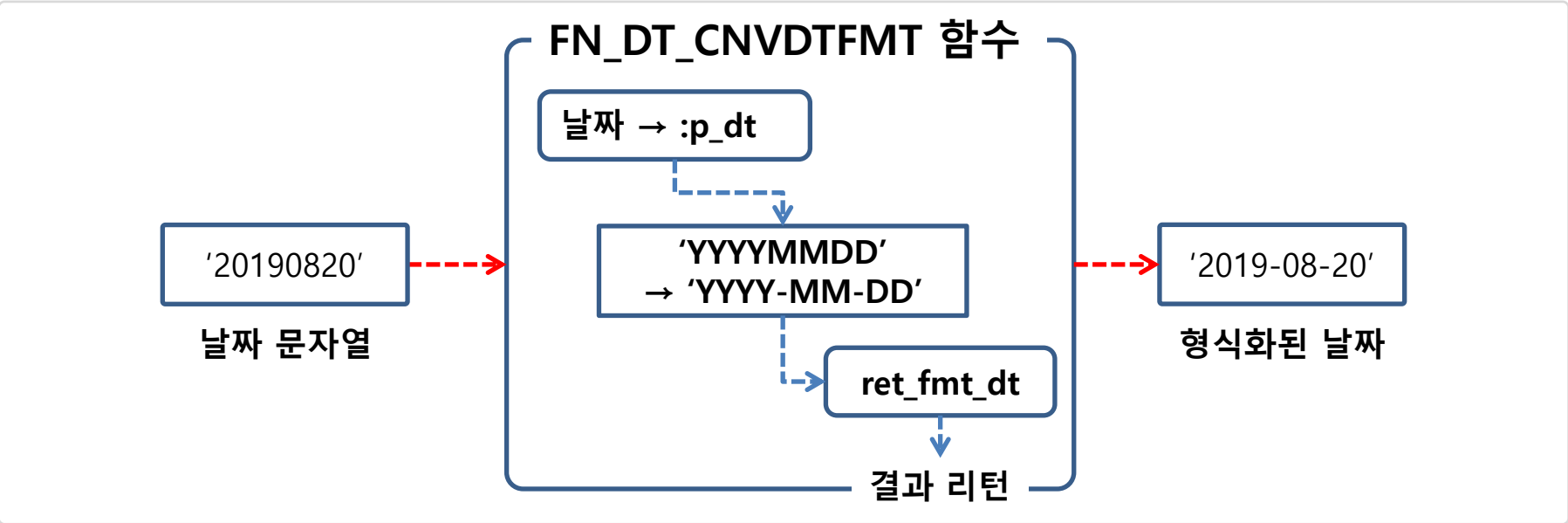
결과

Function FN_DT_OFFSETTMDNO이 (가) 컴파일되었습니다.

```
SELECT TO_CHAR(SYSDATE, 'HH24:MI:SS') AS CUR_TM,  
       FN_DT_OFFSETTMDNO(0, TO_CHAR(SYSDATE, 'HH24MISS'), -43200)  
       AS MINUS_DNO,  
       FN_DT_OFFSETTMDNO(0, TO_CHAR(SYSDATE, 'HH24MISS'), 43200)  
       AS PLUS_DNO  
FROM   DUAL;
```

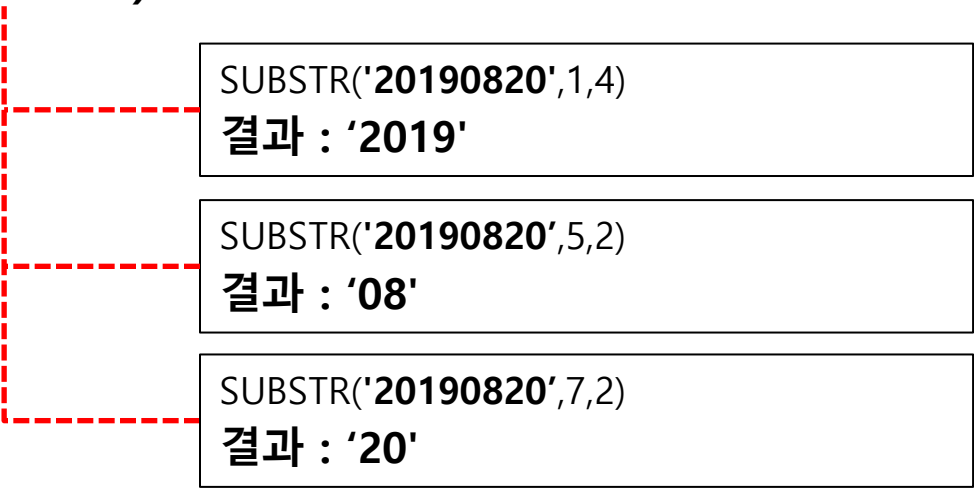
결과

CUR_TM	MINUS_DNO	PLUS_DNO
17:53:20	0	1



:: 'YYYYMMDD' 날짜 문자열을 입력하여 'YYYY-MM-DD' 형식으로 변환하는 함수를 생성합니다.

FN_DT_CNVDTFMT ('20190820')



'2019' || '-' || '08' || '-' || '20' = '2019-08-20'

:: FN_DT_CNVDTFMT() 함수에 '20190820' 매개변수를 입력한 결과


```
CREATE OR REPLACE FUNCTION FN_DT_CNVDTFMT(p_dt CHAR)
RETURN CHAR IS ret_fmt_dt CHAR(10);

BEGIN

    SELECT SUBSTR(p_dt,1,4) || '-' || SUBSTR(p_dt,5,2) || '-' ||
           SUBSTR(p_dt,7,2) INTO ret_fmt_dt
    FROM    DUAL;

    RETURN (ret_fmt_dt);

END FN_DT_CNVDTFMT;
/
```

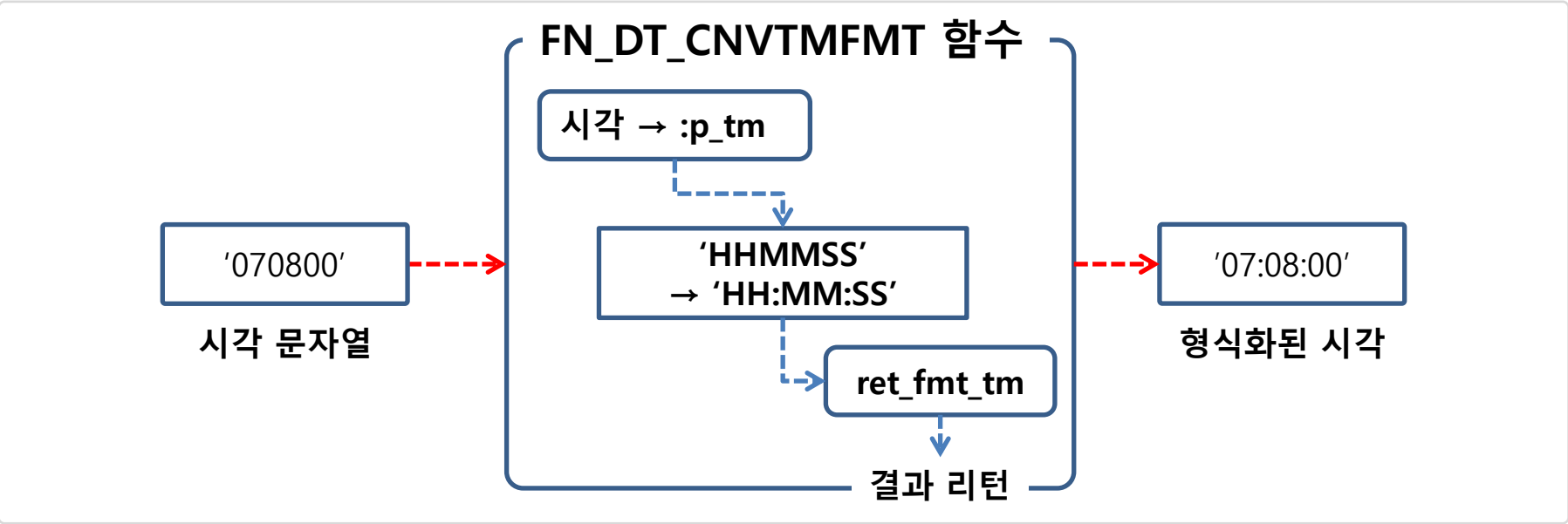
결과

Function **FN_DT_CNVDTFMT**이 (가) 컴파일되었습니다.

```
SELECT FN_DT_CNVDTFMT('20190820')  
FROM   DUAL;
```

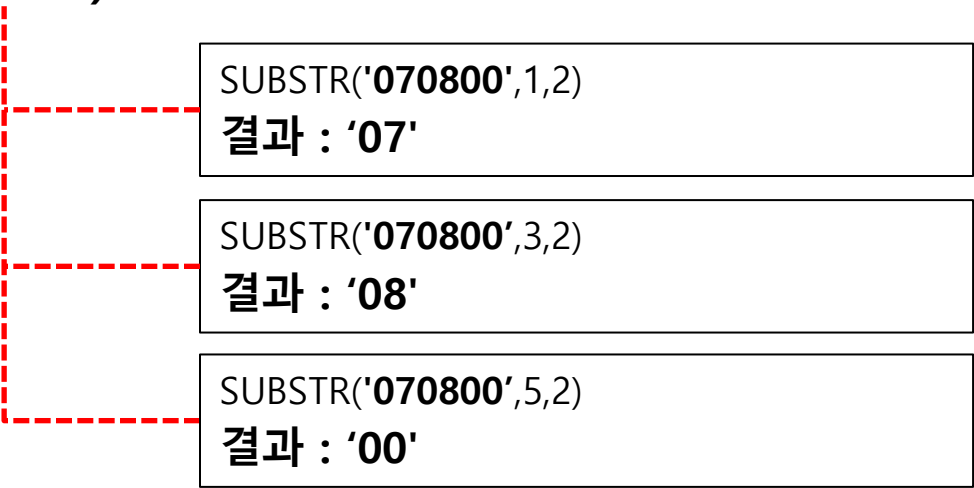
결과

FN_DT_CNVDTFMT('20190820')
2019-08-20



:: 'HHMMSS' 시각 문자열을 'HH:MM:SS' 형식으로 변환하는 함수를 생성합니다.

FN_DT_CNVTFMT ('070800')



'07' || ':' || '08' || ':' || '00' = '07:08:00'

:: FN_DT_CNVTFMT() 함수에 '070800' 매개변수를 입력한 결과

```
CREATE OR REPLACE FUNCTION FN_DT_CNVTFMT(p_tm CHAR)
RETURN CHAR IS ret_fmt_tm CHAR(8);

BEGIN

    SELECT DECODE(p_tm,NULL,NULL,SUBSTR(p_tm,1,2) || ':' ||
                SUBSTR(p_tm,3,2) || ':' || SUBSTR(p_tm,5,2)) INTO ret_fmt_tm
    FROM    DUAL;

    RETURN (ret_fmt_tm);

END FN_DT_CNVTFMT;
/
```

결과

Function **FN_DT_CNVTFMT**이 (가) 컴파일되었습니다.


```
SELECT FN_DT_CNVTFMT( ' 164530' )  
FROM   DUAL;
```

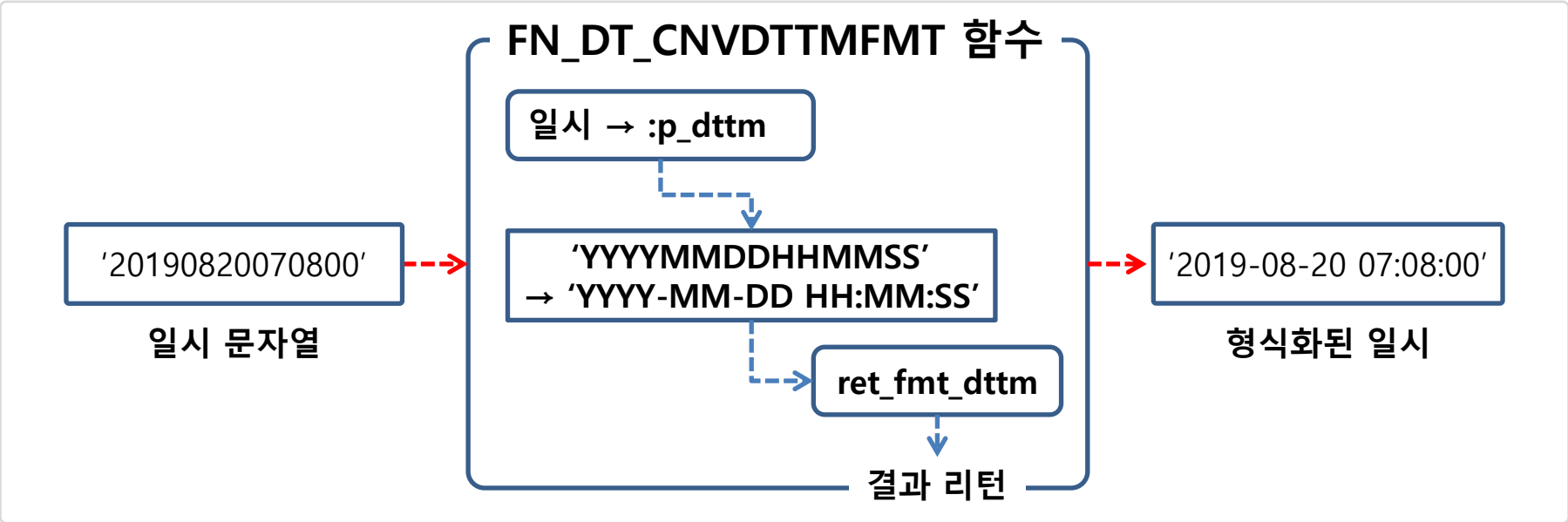
결과

 FN_DT_CNVTFMT('164530')
16:45:30

```
SELECT FN_DT_CNVTFMT(FN_DT_OFFSETTM(0, ' 234500' , 3600))  
FROM   DUAL;
```

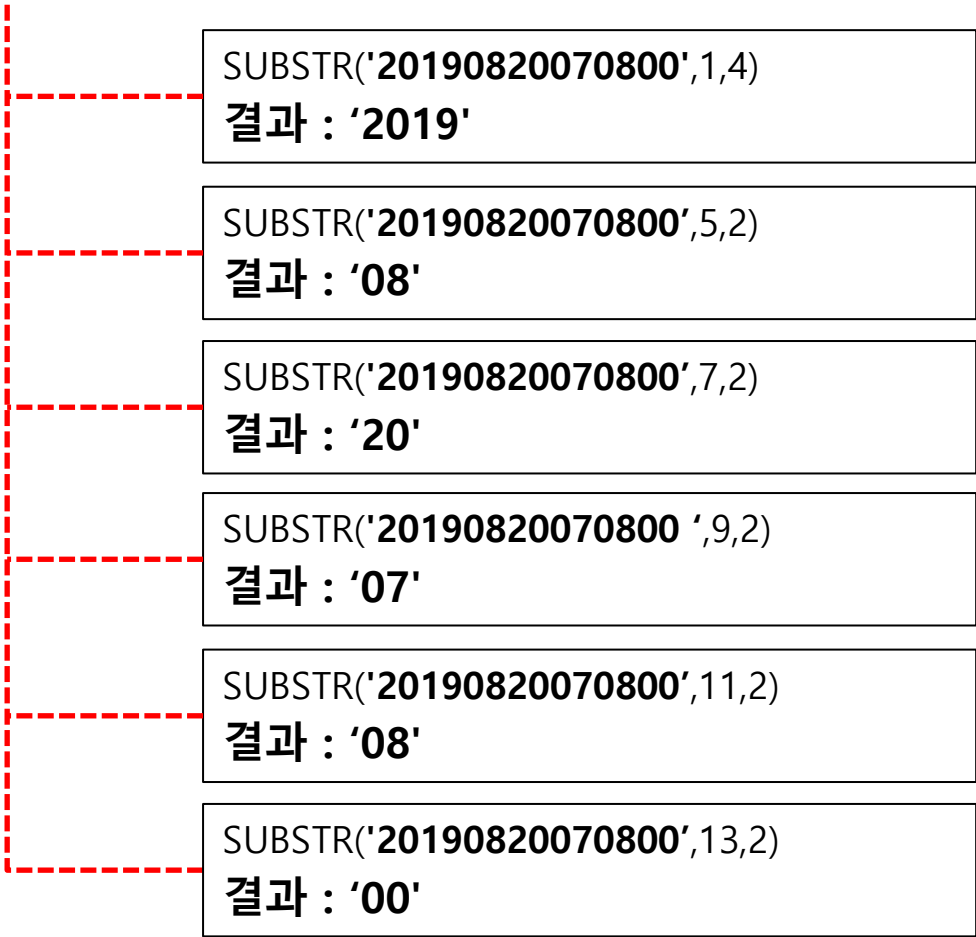
결과

 FN_DT_CNVTFMT(FN_DT_OFFSETTM(0,'234500',3600))
00:45:00



:: 'YYYYMMDDHHMMSS' 문자열을 'YYYY-MM-DD HH:MM:SS' 형식으로 변환하는 함수를 생성합니다.

FN_DT_CNVDTTMFMT ('20190820070800')



'2019' || '-' || '08' || '-' || '20' || ' ' ||
'07' || ':' || '08' || ':' || '00' = '07:08:00'

:: FN_DT_CNVDTTMFMT() 함수에 '070800' 매개변수를 입력한 결과


```
CREATE OR REPLACE FUNCTION FN_DT_CNVDTTMFMT(p_dttm CHAR)
RETURN CHAR IS ret_fmt_dttm CHAR(23);

BEGIN

    SELECT SUBSTR(p_dttm,1,4) || '-' || SUBSTR(p_dttm,5,2) || '-' ||
           SUBSTR(p_dttm,7,2) || ' ' ||
           SUBSTR(p_dttm,9,2) || ':' || SUBSTR(p_dttm,11,2) || ':' ||
           SUBSTR(p_dttm,13,2) INTO ret_fmt_dttm
    FROM    DUAL;

    RETURN (ret_fmt_dttm);


END FN_DT_CNVDTTMFMT;
/
```

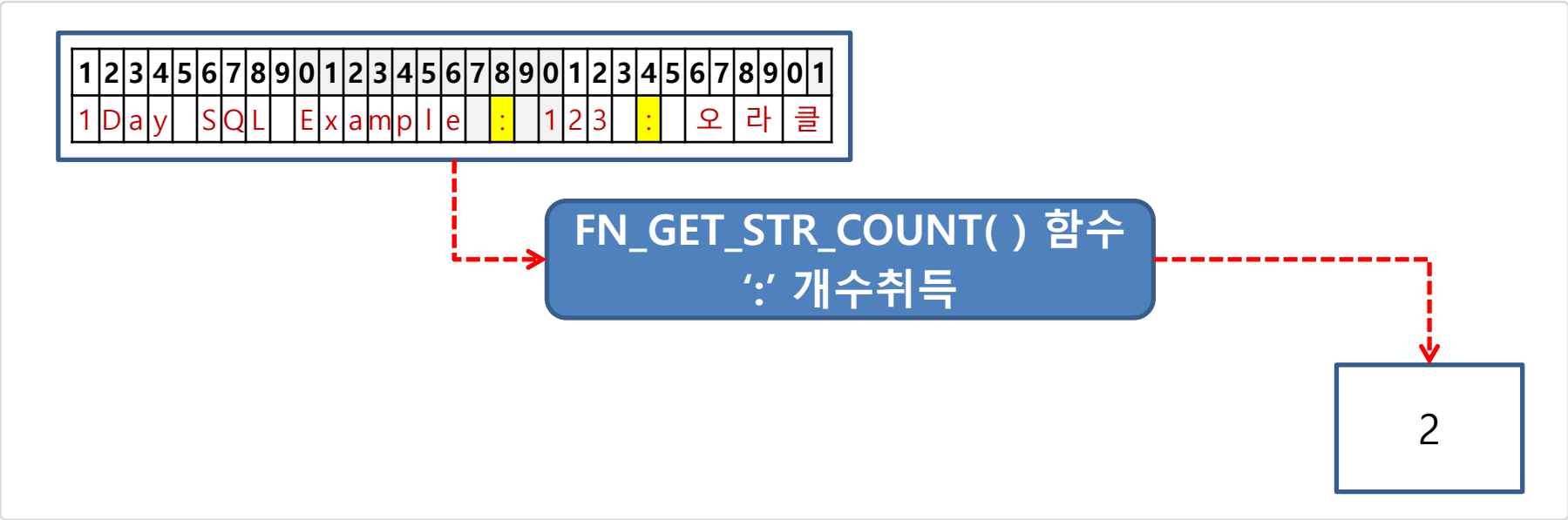
결과

Function `FN_DT_CNVDTTMFMT`이 (가) 컴파일되었습니다.

```
SELECT FN_DT_CNVDTTMFMT( '20190820070800' )  
FROM    DUAL;
```

결과

 FN_DT_CNVDTTMFMT('20190820070800')
2019-08-20 07:08:00

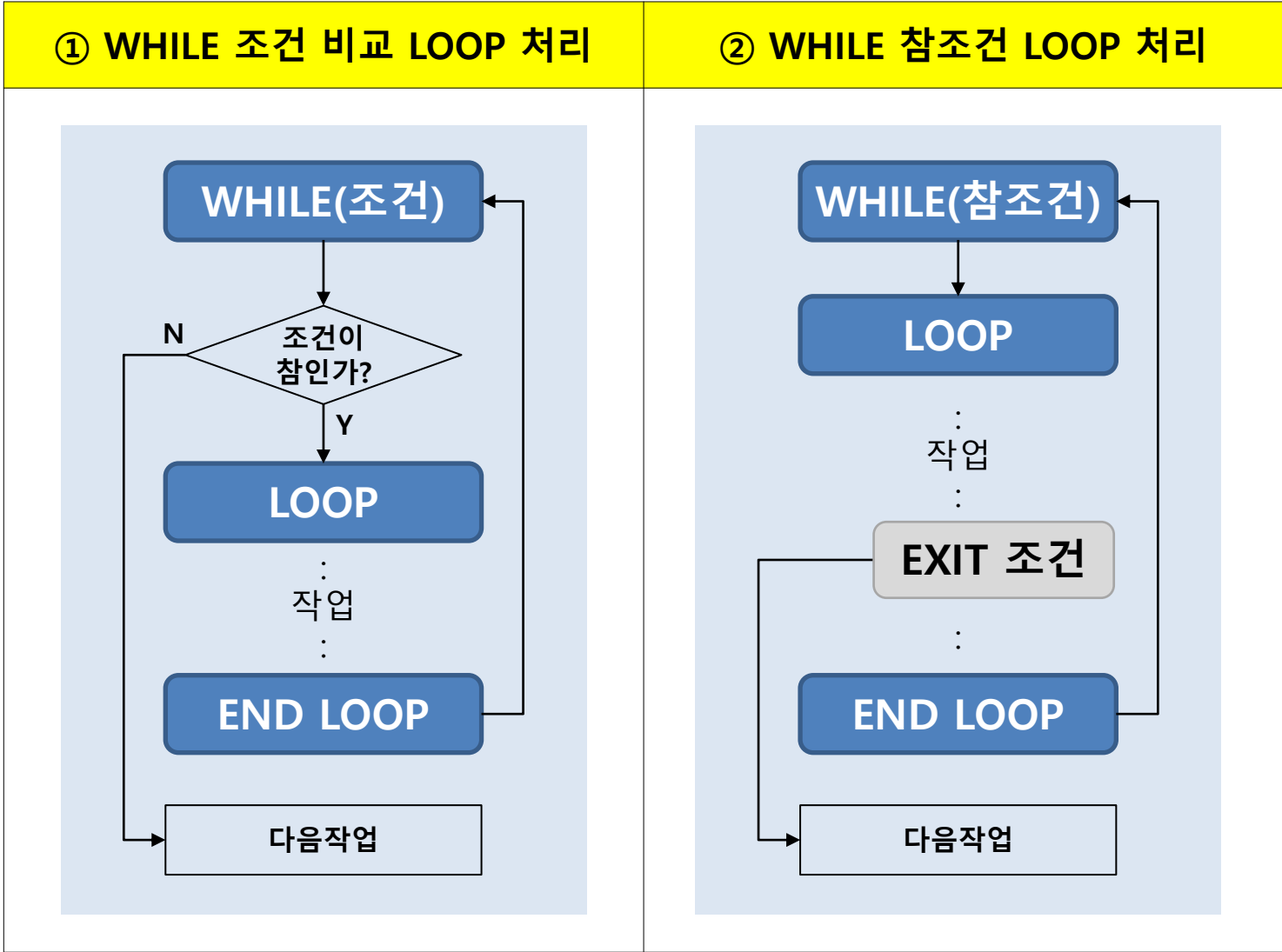


:: 지정문자열에서 특정문자의 개수를 취득하는 함수를 생성합니다.

```
CREATE OR REPLACE FUNCTION FN_GET_STR_COUNT(p_str CHAR,  
                                             p_find_str CHAR)  
  
RETURN NUMBER  
IS  
    ret_sales_count NUMBER(3);  
    loc NUMBER(3);  
BEGIN  
  
    ret_sales_count := 0;  
    loc := 1;  
  
    WHILE (1=1)  
    LOOP  
        loc := INSTR(p_str, p_find_str, loc);  
        EXIT WHEN loc = 0;  
  
        loc := loc + LENGTH(p_find_str);  
        ret_sales_count := ret_sales_count + 1;  
    END LOOP;  
  
    RETURN (ret_sales_count);  
END FN_GET_STR_COUNT;  
/
```

결과

Function `FN_GET_STR_COUNT`이 (가) 컴파일되었습니다.



:: 루프에서 조건 지정 방법

```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'FUNCTION';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
FN_DT_TMTIOSEC	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_SECTOTM	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_OFFSETTM	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_OFFSETTMDNO	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVDTFMT	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVTIMFMT	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVDTTMFMT	FUNCTION	2019/08/02 12:31:09	VALID
CUSTOM_HASH	FUNCTION	2019/07/29 10:29:49	VALID
CUSTOM_AUTH	FUNCTION	2019/07/29 10:29:49	VALID
FN_GET_STR_COUNT	FUNCTION	2019/08/08 15:06:19	VALID

```
SELECT FN_GET_STR_COUNT('1Day SQL Example : 123 : 오라클',':')  
FROM DUAL;
```

결과

<div>FN_GET_STR_COUNT('1DAYSQLEXAMPLE:123:오라클',':')</div>
2


```
DROP FUNCTION FN_GET_STR_COUNT;
```

결과

Function `FN_GET_STR_COUNT`이 (가) 삭제되었습니다.

```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'FUNCTION';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
FN_DT_TMTOSEC	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_SECTOTM	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_OFFSETTM	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_OFFSETTMDNO	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVDTFMT	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVTFMT	FUNCTION	2019/08/02 12:31:09	VALID
FN_DT_CNVDTTMFMT	FUNCTION	2019/08/02 12:31:09	VALID
CUSTOM_HASH	FUNCTION	2019/07/29 10:29:49	VALID
CUSTOM_AUTH	FUNCTION	2019/07/29 10:29:49	VALID

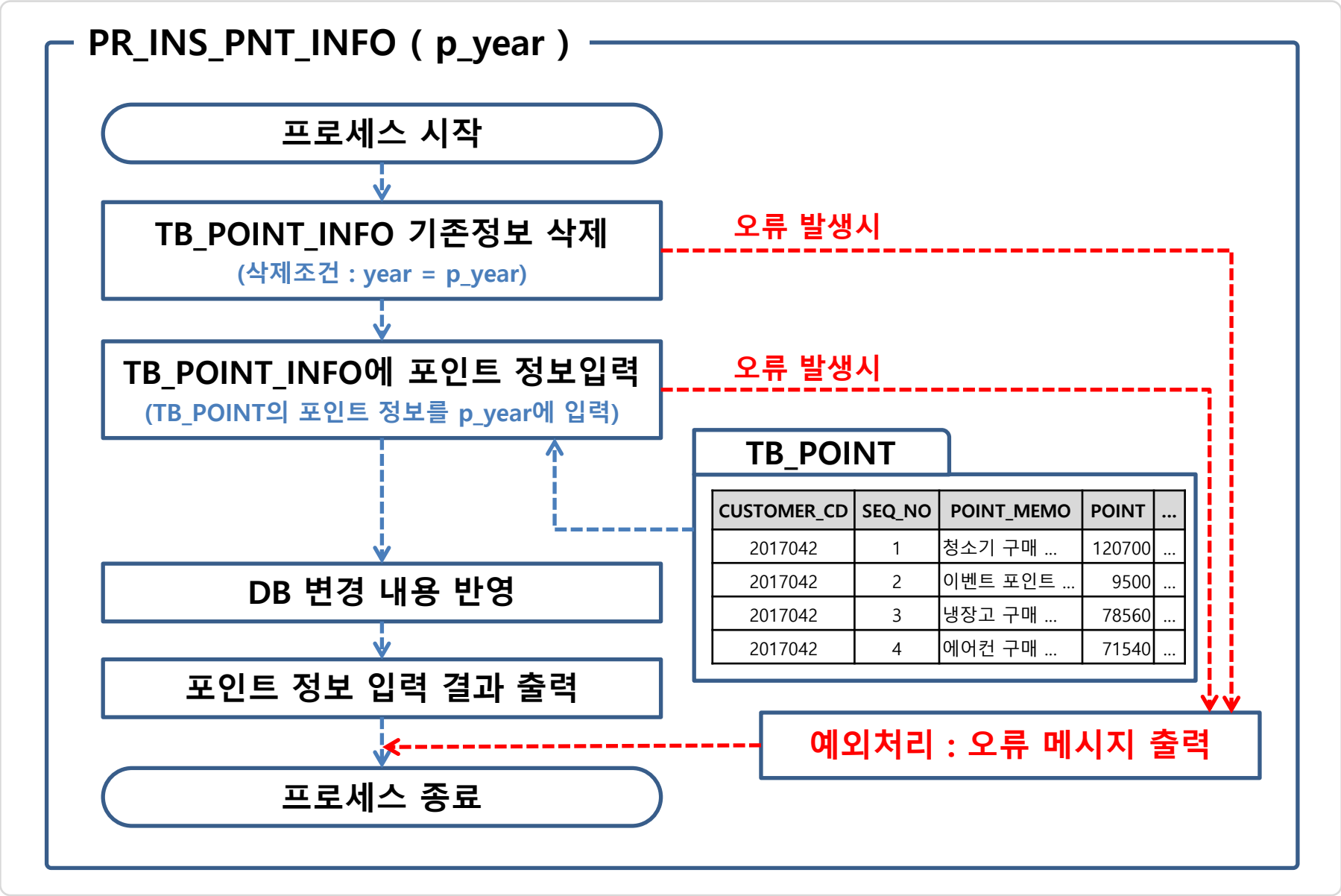
프로시저 (PROCEDURE)

KEYWORD : 데이터베이스 업무 기능 정의

구문 #1	<pre>CREATE [OR REPLACE] PROCEDURE 프로시저명(매개변수1 타입1, 매개변수2 타입2 ...,) IS 변수 변수타입(크기); BEGIN SQL 구문; [EXCEPTION 예외처리;] END 프로시저명; /</pre>
-------	---

구문 #2	<pre>CREATE [OR REPLACE] PROCEDURE 프로시저명(매개변수1 타입1, 매개변수2 타입2 ...,) IS 변수; CURSOR 커서명 IS SELECT SQL구문; BEGIN OPEN 커서명; FETCH 커서명 INTO 변수; 처리; CLOSE 커서명; [EXCEPTION 예외처리;] END 프로시저명; /</pre>
구문 #3	<pre>EXEC 프로시저명;</pre>
구문 #4	<pre>DROP PROCEDURE 프로시저명;</pre>

CURSOR 속성	설명
%FOUND	커서의 결과에 값이 존재할 경우 참입니다.
%NOTFOUND	커서의 결과에 값이 존재하지 않으면 참입니다.
%ROWCOUNT	커서에서 취득한 레코드의 수를 반환합니다.
%ISOPEN	커서가 열려있는 상태라면 참입니다.



∴ TB_POINT_INFO 테이블에 포인트 정보를 입력하는 프로시저를 구성

```
CREATE OR REPLACE PROCEDURE PR_INS_PNT_INFO(p_year CHAR)

IS
    tot_sales_count NUMBER;

BEGIN

    /* 기존 포인트 정보 삭제 */
    DELETE
    FROM    TB_POINT_INFO
    WHERE   YEAR = p_year;

    /* 포인트 정보 입력 */
    INSERT INTO TB_POINT_INFO
    SELECT p_year AS YEAR,
           RANK() OVER (ORDER BY SUM(POINT) DESC) AS RANK,
           CUSTOMER_CD,
           SUM(POINT) AS POINT,
           SUBSTR(MAX(REG_DTTM),1,8) AS LAST_PNT_DT
    FROM    TB_POINT
    WHERE   SUBSTR(REG_DTTM,1,4) = p_year
    GROUP BY CUSTOMER_CD;
```

```
/* 데이터베이스에 적용 */  
COMMIT;  
  
/* 출력 메시지 */  
  
SELECT COUNT(YEAR) INTO tot_sales_count  
FROM    TB_POINT_INFO  
WHERE   YEAR = p_year;  
  
DBMS_OUTPUT.PUT_LINE('결과 : ' || p_year ||  
    ' : ' || tot_sales_count || ' 개 포인트 정보 입력 완료');  
  
/* 예외 처리 */  
EXCEPTION WHEN OTHERS THEN  
  
ROLLBACK;  
DBMS_OUTPUT.PUT_LINE('결과 : ' || p_year ||  
    ' 포인트 입력 오류!!!');  
  
END PR_INS_PNT_INFO;  
/
```


결과

Procedure PR_INS_PNT_INFO이 (가) 컴파일되었습니다.

```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'PROCEDURE';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
PR_INS_PNT_INFO	PROCEDURE	2019/08/08 15:08:13	VALID

```
SET SERVEROUT ON;  
EXEC PR_INS_PNT_INFO('2018')  
SET SERVEROUT OFF;
```

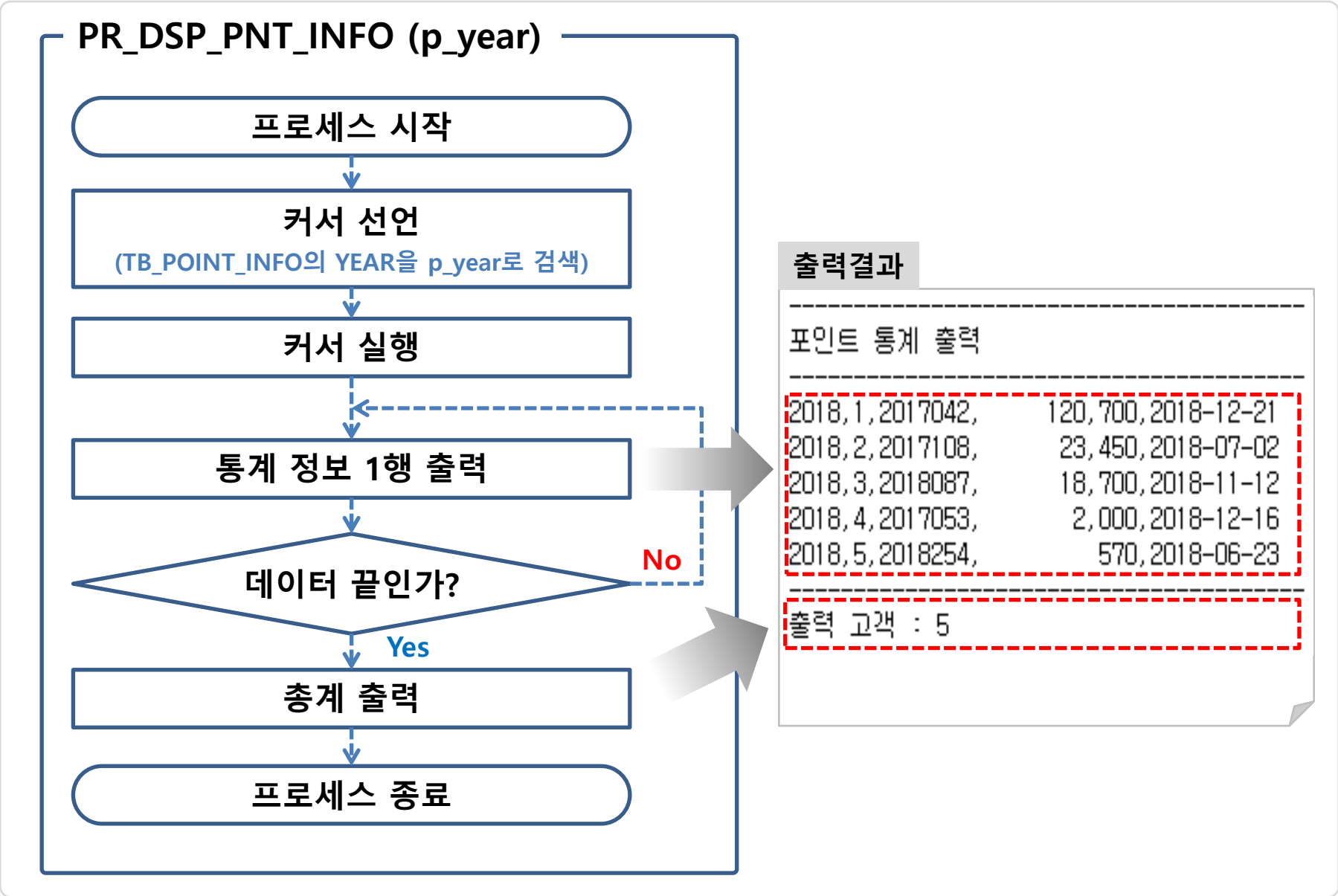
결과

결과 : 2018 : 5 개 포인트 정보 입력 완료
PL/SQL 프로시저가 성공적으로 완료되었습니다.

```
SELECT *  
FROM   TB_POINT_INFO  
WHERE  YEAR = '2018';
```

결과

YEAR	RANK	CUSTOMER_CD	POINT	LAST_POINT_DT
2018	1	2017042	120700	20181221
2018	2	2017108	23450	20180702
2018	3	2018087	18700	20181112
2018	4	2017053	2000	20181216
2018	5	2018254	570	20180623



∴ TB_POINT_INFO 테이블의 포인트 통계정보를 출력하는 프로시저를 구성

```
CREATE OR REPLACE PROCEDURE PR_DSP_PNT_INFO(p_year CHAR)

IS
    /* 커서 선언 */
    CURSOR cur_pnt_info(c_year IN VARCHAR2)
    IS
        SELECT YEAR,
               RANK,
               CUSTOMER_CD,
               POINT,
               LAST_POINT_DT
        FROM    TB_POINT_INFO
        WHERE   YEAR = c_year;

    /* 프로시저 변수 */
    n_tot_sales_count NUMBER;

BEGIN

    n_tot_sales_count := 0;
```

```
DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('포인트 통계 출력');
DBMS_OUTPUT.PUT_LINE('-----');

FOR row_info IN cur_pnt_info(p_year)
LOOP
    DBMS_OUTPUT.PUT_LINE(ROW_INFO.YEAR || ',' ||
        ROW_INFO.RANK || ',' || ROW_INFO.CUSTOMER_CD || ',' ||
        TO_CHAR(ROW_INFO.POINT, '999,999,999') || ',' ||
        FN_DT_CNVDTFMT(ROW_INFO.LAST_POINT_DT));
    n_tot_sales_count := n_tot_sales_count + 1;
END LOOP;

IF (n_tot_sales_count > 0) THEN
    DBMS_OUTPUT.PUT_LINE('-----');
    DBMS_OUTPUT.PUT_LINE('출력 고객:' || n_tot_sales_count);
ELSE
    DBMS_OUTPUT.PUT_LINE('검색된 데이터가 없습니다.');
```

```
END PR_DSP_PNT_INFO;
/
```

결과

Procedure PR_DSP_PNT_INFO이 (가) 컴파일되었습니다.


```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'PROCEDURE';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
PR_INS_PNT_INFO	PROCEDURE	2019/08/08 15:08:13	VALID
PR_DSP_PNT_INFO	PROCEDURE	2019/08/08 15:11:15	VALID

```
SET SERVEROUT ON;  
EXEC PR_DSP_PNT_INFO('2018')  
SET SERVEROUT OFF;
```

결과

포인트 통계 출력

2018,1,2017042,	120,700,2018-12-21
2018,2,2017108,	23,450,2018-07-02
2018,3,2018087,	18,700,2018-11-12
2018,4,2017053,	2,000,2018-12-16
2018,5,2018254,	570,2018-06-23

출력 고객 : 5

PL/SQL 프로시저가 성공적으로 완료되었습니다.

```
DROP PROCEDURE PR_INS_PNT_INFO;  
DROP PROCEDURE PR_DSP_PNT_INFO;  
  
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'PROCEDURE';
```

결과

Procedure PR_INS_PNT_INFO이 (가) 삭제되었습니다.
Procedure PR_DSP_PNT_INFO이 (가) 삭제되었습니다.

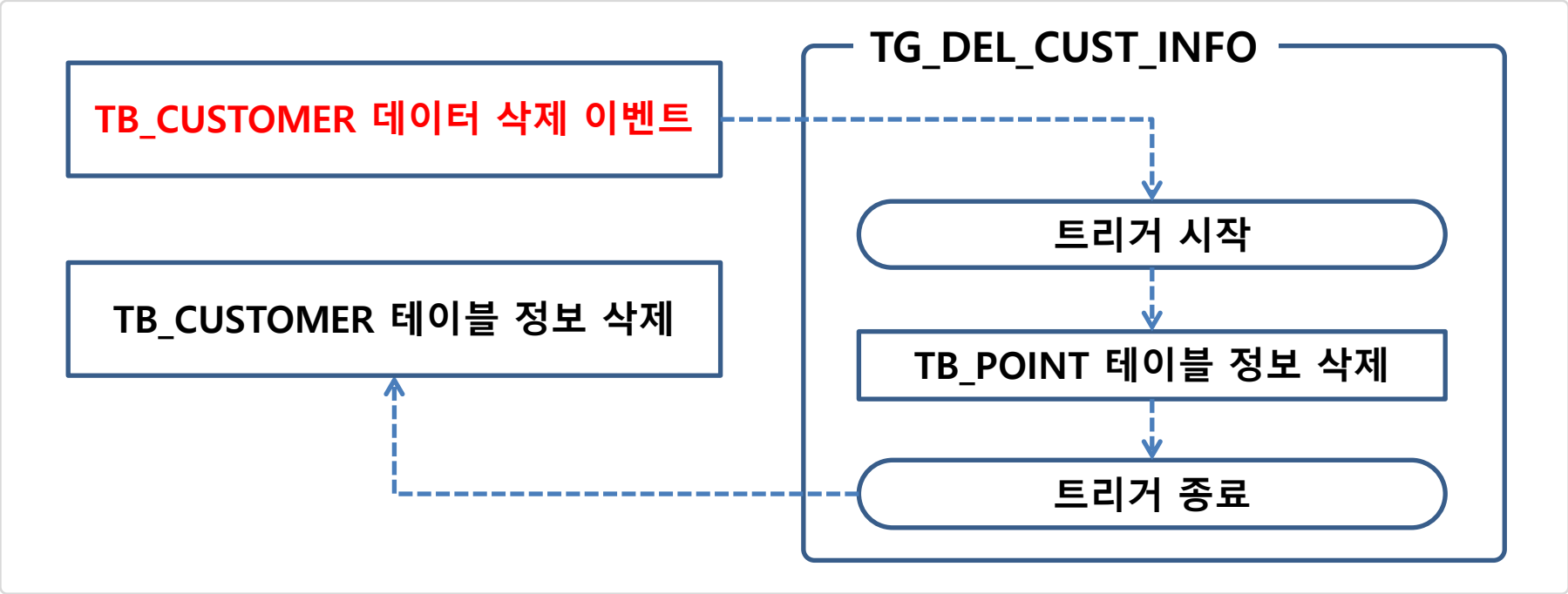
결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
-------------	-------------	---------	--------

트리거 (TRIGGER)

KEYWORD : 데이터베이스 변경 시 자동 처리

구문 #1	<pre>CREATE [OR REPLACE] TRIGGER 트리거명 BEFORE AFTER INSERT UPDATE DELETE ON 테이블 [FOR EACH ROW] [WHEN 조건] DECLARE 변수 변수타입(크기); BEGIN 실행 SQL 구문; [EXCEPTION 예외처리;] END 트리거명; /</pre>
구문 #2	<pre>DROP TRIGGER 트리거명;</pre>



:: TB_CUSTOMER 테이블의 데이터 삭제 시 관련 정보를 일괄 삭제하는 트리거를 구성

```
CREATE OR REPLACE TRIGGER TG_DEL_CUST_INFO  
  
BEFORE DELETE ON TB_CUSTOMER  
  
FOR EACH ROW  
  
BEGIN  
  
DELETE TB_POINT  
WHERE  CUSTOMER_CD = :OLD.CUSTOMER_CD;  
  
END TG_DEL_CUST_INFO;  
/
```

결과

Trigger TG_DEL_CUST_INFO이 (가) 컴파일되었습니다.

```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'TRIGGER'  
AND    OBJECT_NAME = 'TG_DEL_CUST_INFO';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
TG_DEL_CUST_INFO	TRIGGER	2019/08/08 15:13:34	VALID

```
SELECT * FROM TB_CUSTOMER WHERE CUSTOMER_CD = '2018087';  
SELECT * FROM TB_POINT WHERE CUSTOMER_CD = '2018087';
```

결과

CUSTOME	CUSTOMER_N	M	BIRTH_DA	PHONE_NUMBER	EMAIL	TOTAL_POINT	REG_DTTM
2018087	서유리	W	19810925	003-1265-8372	urseo@epnt.co.kr	18700	20180204160903

CUSTOME	SEQ_NO	POINT_MEMO	POINT	REG_DTTM
2018087	1	이벤트 포인트 적립	7800	20180421161903
2018087	2	냉장고 구매 포인트 적립	10900	20181112161956


```
DELETE
FROM    TB_CUSTOMER
WHERE   CUSTOMER_CD = '2018087';
```

결과

1 행 이 (가) 삭제되었습니다.

```
SELECT * FROM TB_CUSTOMER WHERE CUSTOMER_CD = '2018087';  
SELECT * FROM TB_POINT WHERE CUSTOMER_CD = '2018087';
```

결과

선택된 행 없음
선택된 행 없음

```
DROP TRIGGER TG_DEL_CUST_INFO;

SELECT OBJECT_NAME,
       OBJECT_TYPE,
       CREATED,
       STATUS
FROM   USER_OBJECTS
WHERE  OBJECT_TYPE = 'TRIGGER'
AND    OBJECT_NAME = 'TG_DEL_CUST_INFO';
```

결과

Trigger TG_DEL_CUST_INFO이 (가) 삭제되었습니다.

결과

↕ OBJECT_NAME	↕ OBJECT_TYPE	↕ CREATED	↕ STATUS
---------------	---------------	-----------	----------

```
COMMIT;
```

결과

커밋 완료.

뷰 (VIEW)

KEYWORD : 관련 테이블 변경 시 자동 반영되는 가상 테이블

구문 #1	CREATE [OR REPLACE] [FORCE NOFORCE] VIEW 뷰명 AS 실행 SQL 구문 [WHERE 조건 WITH CHECK OPTION WITH READ ONLY];
구문 #2	DROP VIEW 뷰명;

VW_PNT_STAT_INFO 뷰

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	SEQ_NO	POINT_MEMO	POINT
2017042	강원진	남	19810603	00282028790	1	청소기 구매 포인트 적립	120700
2017042	강원진	남	19810603	00282028790	2	이벤트 포인트 적립	9500
2017042	강원진	남	19810603	00282028790	3	냉장고 구매 포인트 적립	78560
2017042	강원진	남	19810603	00282028790	4	에어컨 구매 포인트 적립	71540
2017042	강원진	남	19810603	00282028790	5	냉장고 구매 포인트 적립	78560
2017053	나경숙	여	19891225	00245090043	1	세탁기 구매 포인트 적립	3500
2017053	나경숙	여	19891225	00245090043	2	드라이기 구매 포인트 적립	2000
2017108	박승대	남	19711430	00225809919	1	냉장고 구매 포인트 적립	14065
2017108	박승대	남	19711430	00225809919	2	이벤트 포인트 적립	9385

포인트 통계 정보

자동 데이터 변경

반영

TB_CUSTOMER
고객 테이블 변경

반영

TB_POINT
포인트 테이블 변경

:: 고객코드별 포인트 통계 정보를 관리하는 뷰를 구성

```
CREATE OR REPLACE FORCE VIEW VW_PNT_STAT_INFO

AS

SELECT CU.CUSTOMER_CD,
       CU.CUSTOMER_NM,
       DECODE(CU.MW_FLG, 'M', '남', '여') AS MW_FLG,
       CU.BIRTH_DAY,
       REPLACE(CU.PHONE_NUMBER, '-', '') AS PHONE_NUMBER,
       CP.SEQ_NO,
       CP.POINT_MEMO,
       CP.POINT
FROM   TB_CUSTOMER CU,
       TB_POINT CP
WHERE  CU.CUSTOMER_CD = CP.CUSTOMER_CD
ORDER BY CU.CUSTOMER_CD,
        CP.SEQ_NO;

WITH READ ONLY;

/
```

결과

View VW_PNT_STAT_INFO이 (가) 생성되었습니다.

```
SELECT OBJECT_NAME,  
       OBJECT_TYPE,  
       CREATED,  
       STATUS  
FROM   USER_OBJECTS  
WHERE  OBJECT_TYPE = 'VIEW'  
AND    OBJECT_NAME = 'VW_PNT_STAT_INFO';
```

결과

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
VW_PNT_STAT_INFO	VIEW	2019/08/08 15:17:27	VALID


```
SELECT *  
FROM   VW_PNT_STAT_INFO;
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	SEQ_NO	POINT_MEMO	POINT
2017042	강원진	남	19810603	00282028790	1	청소기 구매 포인트 적립	120700
2017042	강원진	남	19810603	00282028790	2	이벤트 포인트 적립	9500
2017042	강원진	남	19810603	00282028790	3	냉장고 구매 포인트 적립	78560
2017042	강원진	남	19810603	00282028790	4	에어컨 구매 포인트 적립	71540
2017042	강원진	남	19810603	00282028790	5	드라이기 구매 포인트 적립	3200
2017053	나경숙	여	19891225	00245090043	1	세탁기 구매 포인트 적립	3500
2017053	나경숙	여	19891225	00245090043	2	드라이기 구매 포인트 적립	2000
2017108	박승대	남	19711430	00225809919	1	청소기 구매 포인트 적립	14065
2017108	박승대	남	19711430	00225809919	2	이벤트 포인트 적립	9385
2018254	이혜옥	여	19839012	003_1287_9734	1	등록 포인트 적립	500
2018254	이혜옥	여	19839012	003_1287_9734	2	이벤트 포인트 적립	70
2019001	김진숙	여	20010426	00298420074	1	등록 포인트 적립	500
2019001	김진숙	여	20010426	00298420074	2	믹서기 구매 포인트 적립	4600
2019001	김진숙	여	20010426	00298420074	3	드라이기 구매 포인트 적립	7820
2019069	김한길	남	1992 315		1	이벤트 포인트 적립	8900

```
SELECT *
FROM   VW_PNT_STAT_INFO
WHERE  CUSTOMER_NM IN ('이혜옥','한찬희','나경숙');
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	SEQ_NO	POINT_MEMO	POINT
2017053	나경숙	여	19891225	00245090043	1	세탁기 구매 포인트 적립	3500
2017053	나경숙	여	19891225	00245090043	2	드라이기 구매 포인트 적립	2000
2018254	이혜옥	여	19839012	003_1287_9734	1	등록 포인트 적립	500
2018254	이혜옥	여	19839012	003_1287_9734	2	이벤트 포인트 적립	70
2019167	한찬희	남	19711106	002=1202=5563	1	드라이기 구매 포인트 적립	3200
2019167	한찬희	남	19711106	002=1202=5563	2	전기밥솥 구매 포인트 적립	3600

```
UPDATE TB_CUSTOMER  
SET    PHONE_NUMBER = '003-1287-9734'  
WHERE  CUSTOMER_CD = '2018254';
```

[SQL1]

결과

1 행 이 (가) 업데이트되었습니다.

```
UPDATE TB_CUSTOMER  
SET    PHONE_NUMBER = '002-1202-5563'  
WHERE  CUSTOMER_CD = '2019167';
```

[SQL2]

결과

1 행 이 (가) 업데이트되었습니다.

```
DELETE  
FROM  TB_POINT  
WHERE CUSTOMER_CD = '2017053'  
AND   SEQ_NO = 2;
```

[SQL3]

결과

1 행 이 (가) 삭제되었습니다.

```
COMMIT;
```

결과

커밋 완료.

```
SELECT *
FROM   VW_PNT_STAT_INFO
WHERE  CUSTOMER_NM IN ('이혜옥','한찬희','나경숙');
```

결과

CUSTOMER_CD	CUSTOMER_NM	MW_FLG	BIRTH_DAY	PHONE_NUMBER	SEQ_NO	POINT_MEMO	POINT
2017053	나경숙	여	19891225	00245090043	1	세탁기 구매 포인트 적립	3500
2018254	이혜옥	여	19839012	00312879734	1	등록 포인트 적립	500
2018254	이혜옥	여	19839012	00312879734	2	이벤트 포인트 적립	70
2019167	한찬희	남	19711106	00212025563	1	드라이기 구매 포인트 적립	3200
2019167	한찬희	남	19711106	00212025563	2	전기밥솥 구매 포인트 적립	3600

```
DROP VIEW VW_PNT_STAT_INFO;

SELECT OBJECT_NAME,
       OBJECT_TYPE,
       CREATED,
       STATUS
FROM   USER_OBJECTS
WHERE  OBJECT_TYPE = 'VIEW'
AND    OBJECT_NAME = 'VW_PNT_STAT_INFO';
```

결과

View VW_PNT_STAT_INFO이 (가) 삭제되었습니다.

결과

↕ OBJECT_NAME	↕ OBJECT_TYPE	↕ CREATED	↕ STATUS
---------------	---------------	-----------	----------

시퀀스 (SEQUENCE)

KEYWORD : 번호 자동 생성

구문 #1	<pre>CREATE SEQUENCE 시퀀스명 [START WITH 초기값] [INCREMENT BY 증가값] [MAXVALUE 최댓값 NOMAXVALUE] [MINVALUE 최솟값 NOMINVALUE] [CYCLE NOCYCLE] [CACHE NOCACHE];</pre>
-------	--

구문 #2	ALTER SEQUENCE 시퀀스명 [INCREMENT BY 증가값] [MAXVALUE 최댓값 NOMAXVALUE] [MINVALUE 최솟값 NOMINVALUE] [CYCLE NOCYCLE] [CACHE NOCACHE];
구문 #3	시퀀스.CURRVAL
구문 #4	시퀀스.NEXTVAL
구문 #5	DROP SEQUENCE 시퀀스명;

SEQ_PAN_SEQ_NO 시퀀스

초깃값 : 1
증가값 : 1
최댓값 : 9,999,999,999

현재값

현재값 + 1

:: 1부터 9,999,999,999까지의 번호를 자동 증가하는 시퀀스를 생성

```
CREATE SEQUENCE SEQ_PAN_SEQ_NO  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 9999999999  
  NOMINVALUE  
  NOCYCLE  
  NOCACHE;
```

결과

Sequence SEQ_PAN_SEQ_NO이 (가) 생성되었습니다.

```
SELECT SEQ_PAN_SEQ_NO.CURRVAL
FROM   DUAL;

SELECT SEQ_PAN_SEQ_NO.NEXTVAL
FROM   DUAL;
```

결과

ORA-08002: sequence SEQ_PAN_SEQ_NO.CURRVAL is not yet defined in this session
08002, 00000 - "sequence %s.CURRVAL is not yet defined in this session"
*Cause: sequence CURRVAL has been selected before sequence NEXTVAL
*Action: select NEXTVAL from the sequence before selecting CURRVAL

결과

NEXTVAL
1

```
SELECT SEQ_PAN_SEQ_NO.CURRVAL
FROM   DUAL;
```

결과

↕ CURRVAL
1

TB_SALES_09			
SEQ_NO	PRODUCT_NM	CUSTOMER_CD	SALES_COUNT

SEQ_NO	PRODUCT_NM	CUSTOMER_CD	SALES_COUNT
자동입력	냉장고	2018254	4
자동입력	세탁기	2019001	10
자동입력	세탁기	2019167	2
자동입력	세탁기	2019069	6
자동입력	냉장고	2018254	3
자동입력	선풍기	2019167	4
자동입력	냉장고	2019001	2
자동입력	냉장고	2019069	1
자동입력	선풍기	2018254	3
자동입력	선풍기	2019001	5

:: TB_SALES_90 테이블에 자동 입력기능을 사용하여 데이터를 입력

```
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '냉장고', '2018254', 4);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '세탁기', '2019001', 10);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '세탁기', '2019167', 2);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '세탁기', '2019069', 6);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '냉장고', '2018254', 3);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '선풍기', '2019167', 4);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '냉장고', '2019001', 2);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '냉장고', '2019069', 1);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '선풍기', '2018254', 3);
INSERT INTO TB_SALES_09 VALUES (SEQ_PAN_SEQ_NO.NEXTVAL, '선풍기', '2019001', 5);
```

결과

[illegible]

```
SELECT *  
FROM   TB_SALES_09;
```

결과

SEQ_NO	PRODUCT_NM	CUSTOMER_CD	SALES_COUNT
2	냉장고	2018254	4
3	세탁기	2019001	10
4	세탁기	2019167	2
5	세탁기	2019069	6
6	냉장고	2018254	3
7	선풍기	2019167	4
8	냉장고	2019001	2
9	냉장고	2019069	1
10	선풍기	2018254	3
11	선풍기	2019001	5

```
COMMIT;
```

결과

커밋 완료.


```
DROP SEQUENCE SEQ_PAN_SEQ_NO;
```

결과

Sequence SEQ_PAN_SEQ_NO이 (가) 삭제되었습니다.