

#Section3

[실습진행] PL/SQL 딱 이만큼

전체 구조 잡기 (w/커서 For Loop, 변수선언 rowtype)

DECLARE

-- 변수선언 , 서브 프로그램

BEGIN

-- 실행구문

1. 장바구니의 자료를 가져온다.
2. 장바구니의 갯수만큼 Loop 진행을 한다.
3. 주문서에 필요한 기본 정보를 가져온다.
- 3.1 맥주일 경우 미성년자 체크를 한다.
4. 옵션가를 가져온다.
5. 주문금액을 계산한다. (수량*(기준단가+옵션가)
5. 포인트 추가를 한다. 1% 추가
6. 주문서를 생성한다.
7. 개인정보에 포인트를 넣어준다.

EXCEPTION

-- 예외처리

END;

DECLARE

--변수선언, 서브프로그램

-- v_cst_id varchar2(20):='C001';
v_cst_id cst_info.cst_id%type:='C001';
v_MNU_ID tmp_ord.mnu_id%type;
v_MNU_SIZE tmp_ord.mnu_size%type;
-- 주문서 변수
--v_price real_ord.total_price%type:=100;
--v_total_price real_ord.total_price%type:=72000;
--v_point_add real_ord.point_add%type:=0;

r_real_ord real_ord%rowtype;

BEGIN

--실행구문

--1. 장바구니의자료를가져온다.
--2. 장바구니의갯수만큼Loop 진행을한다.
FOR fc IN (SELECT * FROM tmp_ord WHERE cst_id=v_cst_id)
LOOP
v_MNU_SIZE :=fc.mnu_size;
dbms_output.put_line(fc.cst_id);

- 3. 주문서에필요한기본정보를가져온다.
-- 3.1 맥주일경우미성년자체크를한다.
-- 4. 옵션가를가져온다.
-- 5. 주문금액을계산한다. (수량*(기준단가+옵션가)

r_real_ord.price:=200;
r_real_ord.total_price:=9200;
-- 5. 포인트추가를한다. 1% 추가
r_real_ord.point_add:=2000;
-- 6. 주문서를생성한다.

insert into real_ord(ord_seq, cst_id, mnu_id, mnu_size, mnu_ice,
qty, price, total_price, point_use, point_add)
values(1, fc.cst_id, fc.mnu_id, fc.mnu_size, fc.mnu_ice,
fc.qty, r_real_ord.price, r_real_ord.total_price, fc.point_use, r_real_ord.point_add
);

-- 7. 개인정보에포인트를넣어준다.

END LOOP;

END;

메뉴 가격 가져오기 (옵션가 체크) / Function 사용

DECLARE

-- 변수선언 , 서브 프로그램

BEGIN

-- 실행구문

1. 장바구니의 자료를 가져온다.

2. 장바구니의 갯수만큼 Loop 진행을 한다.

3. 주문서에 필요한 기본 정보를 가져온다.

3.1 맥주일 경우 미성년자 체크를 한다.

4. 옵션가를 가져온다.

5. 주문금액을 계산한다. (수량*(기준단가+옵션가)

5. 포인트 추가를 한다. 1% 추가

6. 주문서를 생성한다.

7. 개인정보에 포인트를 넣어준다.

EXCEPTION

-- 예외처리

END;

Function

평선은 어떤 입력 값(없을 수도 있음)에 대해
처리 한 후 Return 값을 반환한다.

```
create or replace FUNCTION F_GET_OPT_PRICE
(
  P_MNU_ID IN VARCHAR2
  , P_MNU_OPT IN VARCHAR2
) RETURN NUMBER AS
-- 변수선언
v_mnu_type menu.mnu_type%type;
v_opt_price menu_opt.opt_price%type;
BEGIN

  -- 메뉴타입 가져오기
  select mnu_type
  into v_mnu_type
  from menu
  where mnu_id=P_MNU_ID
  ;

  -- 옵션가 가져오기
  select opt_price
  into v_opt_price
  from menu_opt
  where mnu_type=v_mnu_type
  and mnu_opt=P_MNU_OPT
  ;

  RETURN v_opt_price;

Exception when others then
  RETURN 0;

END F_GET_OPT_PRICE;
```

Point 계산하기 (w/Function)

```
DECLARE
    -- 변수선언 , 서브 프로그램

BEGIN
    -- 실행구문

    1. 장바구니의 자료를 가져온다.

    2. 장바구니의 갯수만큼 Loop 진행을 한다.

    3. 주문서에 필요한 기본 정보를 가져온다.

    3.1 맥주일 경우 미성년자 체크를 한다.

    4. 옵션가를 가져온다.

    5. 주문금액을 계산한다. (수량*(기준단가+옵션가)

    5. 포인트 추가를 한다. 1% 추가

    6. 주문서를 생성한다.

    7. 개인정보에 포인트를 넣어준다.

EXCEPTION
    -- 예외처리

END;
```



Function

평선은 어떤 입력 값(없을 수도 있음)에 대해
처리 한 후 Return 값을 반환한다.

```
create or replace FUNCTION F_GET_ADDPOINT
(
    P_PRICE IN NUMBER
) RETURN NUMBER
IS
    -- 적립포인트
    v_point cst_info.point%type;
BEGIN

    v_point := Round(P_PRICE*0.1);

    RETURN v_point;

Exception when others then
    RETURN 0;
END F_GET_ADDPOINT;
```

Procedure 생성

Stored Procedure

Declare 문은 실행 가능한 PL/SQL 이지만, 이름을 가지고 있지(익명) 않아 1회성 사용을 하게 된다.

Procedure 문은 명칭을 사용하게 되어 오라클 내에 객체로 저장하게 되어 Stored(저장된) Procedure 라고 불린다.

----- 기본 구문 -----

```
CREATE OR REPLACE PROCEDURE 프로시저명(  
    PARAM1 IN VARCHAR2    -- IN : 입력 파라미터  
, PARAM2 IN OUT VARCHAR2 -- IN OUT : 입력, 리턴값 동시 사용 가능  
, PARAM3 OUT VARCHAR2    -- OUT : 프로시저 리턴값  
) AS  
    -- 변수 선언부  
BEGIN  
    -- 실행부  
END PROCEDURE1;
```

Package 생성

Package Specification

```
CREATE OR REPLACE PACKAGE 패키지명  
AS
```

```
/* TODO enter package declarations  
(types, exceptions, methods etc) here */
```

- **Function , Procedure 명세작성**
- 변수, 커서, 예외

**이곳에 정의된 객체만 외부에서 접근이 가능하다.
(Java interface 와 유사?)**

```
END 패키지명;
```

Package Body

```
CREATE OR REPLACE PACKAGE BODY 패키지명  
AS
```

```
Function 평선이름(  
    파라미터  
) Return 자료형 IS  
BEGIN  
    -- statements  
END 평선이름;
```

```
Procedure 프로시저이름(  
    파라미터  
) IS  
BEGIN  
    -- statements  
END 프로시저이름;
```

```
END 패키지명;
```