#Section8
Dynamic SQL

Dynamic SQL 개요

Dynamic SQL is a programming methodology for generating and running SQL statements at run time.
동적 SQL은 런타임 시 SQL 문을 생성하고 실행하기 위한 프로그래밍 방법론입니다.

Static SQL vs. Dynamic SQL https://www.oratable.com/static-sql-vs-dynamic-sql/

If it can be done in static SQL, do it in static SQL.

1. Static SQL provides compile time checking. Dynamic SQL does not.
2. Static SQL creates schema object dependencies. Dynamic SQL does not.
3. Dynamic SQL comes with greater security risks
4. Static SQL (usually) performs better than Dynamic SQL
5. Static SQL is easier to read and maintain than Dynamic SQL

Dynamic SQL 개념 맛보기 (vs. Static SQL)

Step 1. 기본 사용 구문 by SQL

▪ Static SQL

```
--  기본구문
declare
 v_birth cst_info.birth%type;
begin

  -- 생년 가져오기
  select birth
  into v_birth
  from cst_info
  where cst_id='C001';
  dbms_output.put_line('v_birth : '||v_birth);
end;
```

▪ Dynamic SQL

```
--  기본구문
declare
 v_birth cst_info.birth%type;
begin

  -- 생년 가져오기
  EXECUTE IMMEDIATE
    ' select birth '
   ||' from cst_info '
   ||' where cst_id=''C001'''
  INTO v_birth
  ;

  dbms_output.put_line('v_birth : '||v_birth);
end;
```

Dynamic SQL 개념 맛보기 (vs. Static SQL)

Step 2. Parameter 사용 방법

▪ Static SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
begin

  -- 생년 가져오기
  select birth
  into v_birth
  from cst_info
  where cst_id=p_cst_id;

  dbms_output.put_line('v_birth : '||v_birth);

end;
```

▪ Dynamic SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
 v_qry varchar2(1000);
begin

  v_qry :=' select birth from cst_info where cst_id= :val ';
  -- 생년 가져오기
  EXECUTE IMMEDIATE
    v_qry
  INTO v_birth
  USING p_cst_id
  ;

  dbms_output.put_line('v_birth : '||v_birth);

end;
```

Dynamic SQL 개념 맛보기 (vs. Static SQL)

Step 3. Runtime 유무 확인

▪ Static SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
begin

  -- 생년 가져오기
  select birth
  into v_birth
  from cst_info
  where cst_id=p_cst_id;

  dbms_output.put_line('v_birth : '||v_birth);

end;
```

*** Static SQL is a PL/SQL feature that allows SQL syntax directly in a PL/SQL statement.**

▪ Dynamic SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
 v_qry varchar2(1000);
begin

  v_qry :=' select birth from cst_info where cst_id= :val ';
  -- 생년 가져오기
  EXECUTE IMMEDIATE
    v_qry
  INTO v_birth
  USING p_cst_id
  ;
  dbms_output.put_line('v_birth : '||v_birth);
end;
```

*** Dynamic SQL** is a programming methodology for generating and running SQL statements at run time.

Dynamic SQL 개념 맛보기 (vs. Static SQL)

Step 4. by Runtime 특징 – 쿼리 재사용

▪ Static SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
begin
  dbms_output.put_line('Start');

  -- 생년 가져오기
  select birth    into v_birth    from cst_info    where cst_id=p_cst_id;
  dbms_output.put_line('v_birth : '||v_birth);

  -- 생년 가져오기
  p_cst_id :='C002';
  select birth    into v_birth    from cst_info    where cst_id=p_cst_id;
  dbms_output.put_line('v_birth : '||v_birth);

end;
```

▪ Dynamic SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
 v_qry varchar2(1000);
begin

  v_qry :=' select birth from cst_info where cst_id= :val ';
  -- 생년 가져오기
  EXECUTE IMMEDIATE v_qry INTO v_birth USING p_cst_id  ;
  dbms_output.put_line('v_birth : '||v_birth);

  p_cst_id :='C002';
  EXECUTE IMMEDIATE v_qry INTO v_birth USING p_cst_id  ;
  dbms_output.put_line('v_birth : '||v_birth);
end;
```

Dynamic SQL 개념 맛보기 (vs. Static SQL)

Step 5. by Runtime 특징 – 컬럼, 테이블명등.. 변수 사용

▪ Static SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
begin
  dbms_output.put_line('Start');

  -- 생년 가져오기
  select birth    into v_birth    from cst_info    where cst_id=p_cst_id;
  dbms_output.put_line('v_birth : '||v_birth);

  -- 생년 가져오기
  p_cst_id :='C002';
  select birth    into v_birth    from cst_info    where cst_id=p_cst_id;
  dbms_output.put_line('v_birth : '||v_birth);

end;
```

▪ Dynamic SQL

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
 v_qry varchar2(1000);
 v_text varchar2(1000);
begin
  v_text :='birth from cst_info ';
  v_qry :=' select '||v_text||' where cst_id= :val ';
  dbms_output.put_line(v_qry);
  -- 생년 가져오기
  EXECUTE IMMEDIATE v_qry INTO v_birth USING p_cst_id  ;
  dbms_output.put_line('v_birth : '||v_birth);

end;
```

Dynamic SQL - DML 실습

▪ Select

```
declare
 v_birth cst_info.birth%type;
 p_cst_id cst_info.cst_id%type:='C001';
 v_qry varchar2(1000);
begin


  v_qry :=' select birth from cst_info where cst_id= :val ';
  -- 생년 가져오기
  EXECUTE IMMEDIATE
    v_qry
  INTO v_birth
  USING p_cst_id
  ;
   dbms_output.put_line('v_birth : '||v_birth);


end;
```

▪ 실습과제 : 인자값으로 Key , 컬럼명 2개를 넘겨서
컬럼의 값을 리턴값으로 받아오는 평션을 생성하고 호출하기
(Table - MENU , Key 파라미터 mnu_id 사용)

```
--  기본구문
declare
 r_val varchar2(100);

 Function f_get_val(p_key in varchar2, p_col in varchar2)
 Return varchar2
 is
  v_qry varchar2(1000);
  v_val varchar2(100);

 Begin
   -- 쿼리 생성 가져오기
    v_qry :=' select '||p_col||' from menu where mnu_id = :p_key ';
   -- 쿼리 실행
  EXECUTE IMMEDIATE v_qry INTO v_val  USING p_key ;
  Return v_val;

 End f_get_val;

begin

  r_val :=f_get_val('M005', 'MNU_NM');
  dbms_output.put_line('r_val : '||r_val);
end;
```

Dynamic SQL - DML 실습

▪ Update

```
declare
  p_mnu_id   menu.mnu_id%type :='M001';
  p_price    menu.mnu_price%type :=9494;
  v_qry varchar2(1000);

begin

  v_qry :=' UPDATE MENU SET mnu_price = :p   where
mnu_id= ''M005'' ';
  dbms_output.put_line(v_qry);
  EXECUTE IMMEDIATE v_qry   USING p_price   ;
  dbms_output.put_line(SQL%ROWCOUNT);

end;
```

▪ 실습과제 : 인자값으로 Key , 컬럼명, 컬럼명 입력값 3개를 넘겨서
컬럼의 값을 변경하는 프로시저를 생성하고 호출하기
(Table - MENU , Key 파라미터 mnu_id 사용)

```
declare

  Procedure sp_menu(p_key in varchar2, p_col in varchar2, p_val in
varchar2)
  is
  v_qry varchar2(1000);
  v_val varchar2(100);

 Begin
   -- 쿼리 생성 가져오기
    v_qry :=' UPDATE MENU SET '||p_col||' = :b where mnu_id= :c ';
   -- 쿼리 실행
  EXECUTE IMMEDIATE v_qry USING p_val, p_key ;

 End sp_menu;

begin

  sp_menu('M005', 'MNU_NM','흑맥주2');

end;
```

Dynamic SQL - DDL 실습

Create , Drop, Alter 등 사용 가능

```
declare
begin

 delete from tmp_ord2;

 insert into tmp_ord2
 select * from tmp_ord;

 commit;
--
end;
```

▪ **Delete -> Truncate**

```
declare
  v_qry varchar2(1000);
begin
  -- 테이블 데이타 삭제
 v_qry := ' Truncate table tmp_ord2 drop storage ';
 EXECUTE IMMEDIATE v_qry;
  -- 인덱스 삭제
 v_qry := ' Drop index idx_tmp_ord2_01  ';
 EXECUTE IMMEDIATE v_qry;
  -- 데이타 입력
 insert into tmp_ord2  select * from tmp_ord;
   -- 인덱스 생성
 v_qry := ' Create index idx_tmp_ord2_01 on
tmp_ord2(cst_id)  ';
 EXECUTE IMMEDIATE v_qry;

 insert into tmp_ord2  select * from tmp_ord;
--
 commit;
end;
```