**DYNAMIC PLSQL ON EXAMPLES**

**EXAMPLE 1. HARD PARSING**

CREATE OR REPLACE PROCEDURE fetch\_sales\_dynamic (

s\_orderid NUMBER,

s\_cust\_id NUMBER) AS

sale\_rec sales%rowtype;

sql\_stmt VARCHAR2(500) := 'SELECT SALES\_DATE, ORDER\_ID, PRODUCT\_ID, CUSTOMER\_ID, SALESPERSON\_ID, QUANTITY, UNIT\_PRICE, SALES\_AMOUNT, TAX\_AMOUNT, TOTAL\_AMOUNT FROM SALES WHERE 1 = 1' ;

BEGIN

IF s\_orderid IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = ' || s\_orderid;

END IF;

IF s\_cust\_id IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = ' || s\_cust\_id;

END IF;

dbms\_output.put\_line(sql\_stmt);

EXECUTE IMMEDIATE sql\_stmt INTO sale\_rec;

dbms\_output.put\_line(sale\_rec.order\_id);

dbms\_output.put\_line(sale\_rec.customer\_id);

dbms\_output.put\_line(sale\_rec.product\_id);

EXCEPTION

WHEN no\_data\_found THEN

dbms\_output.put\_line('NO SUCH ORDER');

WHEN too\_many\_rows THEN

dbms\_output.put\_line('YOU GOT MORE THAN 1 ROW!');

WHEN OTHERS THEN

dbms\_output.put\_line('ERROR');

END;

**EXAMPLE 2. SOFT PARSING**

CREATE OR REPLACE PROCEDURE fetch\_sales\_dynamic (

s\_orderid NUMBER,

s\_cust\_id NUMBER) AS

sale\_rec sales%rowtype;

sql\_stmt VARCHAR2(500) := 'SELECT SALES\_DATE, ORDER\_ID, PRODUCT\_ID, CUSTOMER\_ID, SALESPERSON\_ID, QUANTITY, UNIT\_PRICE, SALES\_AMOUNT, TAX\_AMOUNT, TOTAL\_AMOUNT FROM SALES WHERE 1 = 1' ;

BEGIN

IF s\_orderid IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = :var1';

END IF;

IF s\_cust\_id IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = :var2';

END IF;

dbms\_output.put\_line(sql\_stmt);

IF s\_orderid IS NOT NULL AND s\_cust\_id IS NULL THEN

EXECUTE IMMEDIATE sql\_stmt INTO sale\_rec USING s\_orderid;

ELSIF s\_orderid IS NULL AND s\_cust\_id IS NOT NULL THEN

EXECUTE IMMEDIATE sql\_stmt INTO sale\_rec USING s\_cust\_id;

ELSIF s\_orderid IS NOT NULL AND s\_cust\_id IS NOT NULL THEN

EXECUTE IMMEDIATE sql\_stmt INTO sale\_rec USING s\_orderid, s\_cust\_id;

END IF;

dbms\_output.put\_line(sale\_rec.order\_id);

dbms\_output.put\_line(sale\_rec.customer\_id);

dbms\_output.put\_line(sale\_rec.product\_id);

EXCEPTION

WHEN no\_data\_found THEN

dbms\_output.put\_line('NO SUCH ORDER');

WHEN too\_many\_rows THEN

dbms\_output.put\_line('YOU GOT MORE THAN 1 ROW!');

WHEN OTHERS THEN

dbms\_output.put\_line('ERROR');

END;

**EXAMPLE 3. WITH CURSOR**

CREATE OR REPLACE PROCEDURE fetch\_sales\_dynamic (

s\_orderid NUMBER,

s\_cust\_id NUMBER) AS

type sales\_cur is ref cursor;

s\_cursor sales\_cur;

sale\_rec sales%rowtype;

sql\_stmt VARCHAR2(500) := 'SELECT SALES\_DATE, ORDER\_ID, PRODUCT\_ID, CUSTOMER\_ID, SALESPERSON\_ID, QUANTITY, UNIT\_PRICE, SALES\_AMOUNT, TAX\_AMOUNT, TOTAL\_AMOUNT FROM SALES WHERE 1 = 1' ;

BEGIN

IF s\_orderid IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = :var1';

END IF;

IF s\_cust\_id IS NOT NULL THEN

sql\_stmt := sql\_stmt || ' AND ORDER\_ID = :var2';

END IF;

dbms\_output.put\_line(sql\_stmt);

IF s\_orderid IS NOT NULL AND s\_cust\_id IS NULL THEN

OPEN s\_cursor FOR sql\_stmt USING s\_orderid;

ELSIF s\_orderid IS NULL AND s\_cust\_id IS NOT NULL THEN

OPEN s\_cursor FOR sql\_stmt USING s\_cust\_id;

ELSIF s\_orderid IS NOT NULL AND s\_cust\_id IS NOT NULL THEN

OPEN s\_cursor FOR sql\_stmt USING s\_orderid, s\_cust\_id;

END IF;

LOOP

FETCH s\_cursor INTO sale\_rec;

EXIT WHEN s\_cursor%notfound;

dbms\_output.put\_line(sale\_rec.order\_id);

dbms\_output.put\_line(sale\_rec.customer\_id);

dbms\_output.put\_line(sale\_rec.product\_id);

END LOOP;

-- no exception needed as cursor is capable of getting multiple rows if looped

END;