

Created by Ahsan Arif

## Procedure with cursor

---

**Question:** In Oracle, I have a table called "wine" and a stored procedure that outputs a cursor based on the "wine" table.

I've created an HTML Form where the user can enter any combination of three values to retrieve results from the "wine" table. My problem is that I need a general "select" statement that will work no matter what value(s), the user enters.

**Example:**

```
parameter_1= "Chianti"  
parameter_2= "10"  
parameter_3= wasn't entered by the user but I have to use in the select statement. And  
this is my problem. How to initialize this parameter to get all rows for column3?
```

```
SELECT * FROM wine  
WHERE column1 = parameter_1  
AND column2 = parameter_2  
AND column3 = parameter_3;
```

The output of my stored procedure must be a cursor.

**Answer:** To solve your problem, you will need to output a dynamic PLSQL cursor in Oracle.

Let's take a look at how we can do this. We've divided this process into 3 steps.

### Step 1 - Table Definition

First, we need a table created in Oracle called "wine". Below is the create statement for the wine table.

```
create table wine  
( col1 varchar2(40),  
  col2 varchar2(40),  
  col3 varchar2(40)  
);
```

We've made this table definition very simple, for demonstration purposes.

## Step 2 - Create package

Next, we've created a package called "winepkg" that contains our cursor definition. This needs to be done so that we can use a cursor as an **output** parameter in our stored procedure.

```
create or replace PACKAGE winepkg
IS
  /* Define the REF CURSOR type. */
  TYPE wine_type IS REF CURSOR RETURN wine%ROWTYPE;
END winepkg;
```

This cursor will accept all fields from the "wine" table.

## Step 3 - Create stored procedure

Our final step is to create a stored procedure to return the cursor. It accepts three parameters (entered by the user on the HTML Form) and returns a cursor (c1) of type "wine\_type" which was declared in Step 2.

The procedure will determine the appropriate cursor to return, based on the value(s) that have been entered by the user (input parameters).

```
create or replace procedure find_wine2
(col1_in in varchar2,
 col2_in in varchar2,
 col3_in in varchar2,
 c1 out winepkg.wine_type)
as

BEGIN

  /* all columns were entered */
  IF (length(col1_in) > 0) and (length(col2_in) > 0) and (length(col3_in) > 0)
  THEN
    OPEN c1 FOR
    select *
    from wine
    where wine.col1 = col1_in
    and wine.col2 = col2_in
    and wine.col3 = col3_in;

  /* col1 and col2 were entered */
  ELSIF (length(col1_in) > 0) and (length(col2_in) > 0) and (length(col3_in) = 0)
  THEN
    OPEN c1 FOR
```

```

select *
from wine
where wine.col1 = col1_in
and wine.col2 = col2_in;

/* col1 and col3 were entered */
ELSIF (length(col1_in) > 0) and (length(col2_in) = 0) and (length(col3_in) > 0)
THEN
  OPEN c1 FOR
  select *
  from wine
  where wine.col1 = col1_in
  and wine.col3 = col3_in;

/* col2 and col3 where entered */
ELSIF (length(col1_in) = 0) and (length(col2_in) > 0) and (length(col3_in) > 0)
THEN
  OPEN c1 FOR
  select *
  from wine
  where wine.col2 = col2_in
  and wine.col3 = col3_in;

/* col1 was entered */
ELSIF (length(col1_in) > 0) and (length(col2_in) = 0) and (length(col3_in) = 0)
THEN
  OPEN c1 FOR
  select *
  from wine
  where wine.col1 = col1_in;

/* col2 was entered */
ELSIF (length(col1_in) = 0) and (length(col2_in) > 0) and (length(col3_in) = 0)
THEN
  OPEN c1 FOR
  select *
  from wine
  where wine.col2 = col2_in;

/* col3 was entered */
ELSIF (length(col1_in) = 0) and (length(col2_in) = 0) and (length(col3_in) > 0)
THEN
  OPEN c1 FOR
  select *
  from wine
  where wine.col3 = col3_in;

```

END IF;

END find\_wine2;