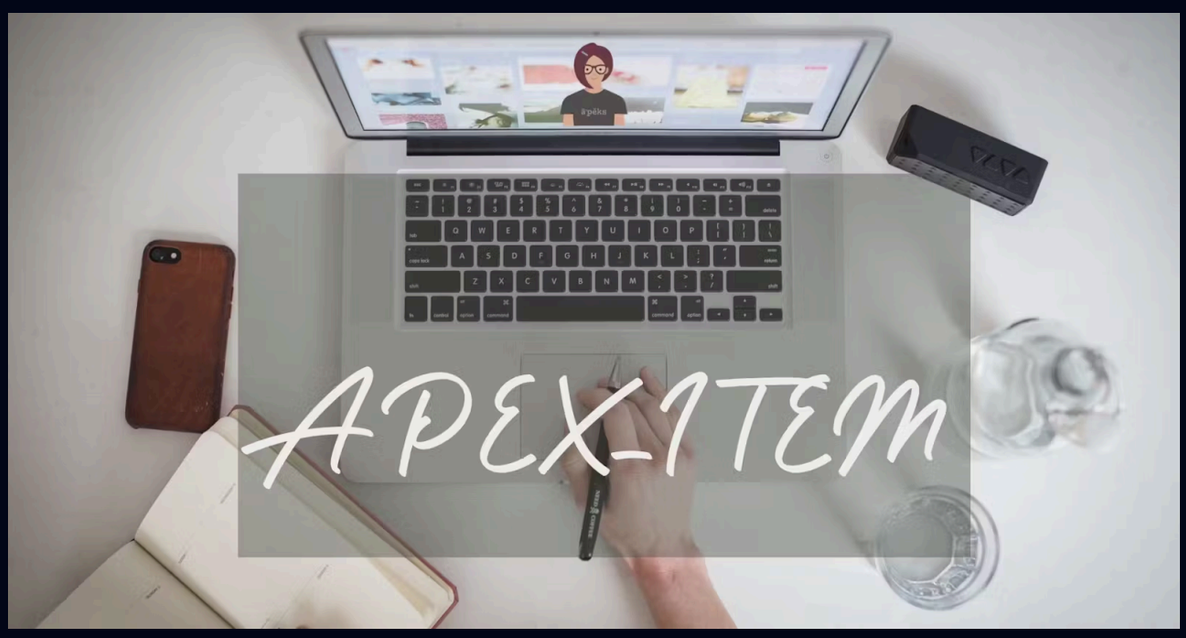


Using APEX_ITEM to Update and Insert a Table with Oracle APEX

Karla Cornejo

Jun 1, 2024 · 4 min read

Oracle Application Express (APEX) provides a number of features, the package `APEX_ITEM` can be used to create form elements dynamically based on an SQL query, rather than creating individual elements page by page. These elements can be used to capture user data and process it in PL/SQL. In this article, we will explain how to use `APEX_ITEM` to manage data entry and how to process this data to insert or update records in a table.

The objective of this [illegible] insert the income of the products in a table `productos_ingresos`. In the product master report,

there is a QUANTITY field, where the user enters the data and at the time of recording goes through line by line with a loop of the report, validating and inserting them into our final table `productos_ingresos`.

Step by Step:

1. We define our query using a Classic Report as follows:

```
Select APEX_ITEM.DISPLAY_AND_SAVE(20, ID) ID,
       IDPRODUCT COD_PRODUCTO,
       APEX_ITEM.DISPLAY_AND_SAVE(30, DESCRIPCTION) DESCRIPCTI
       APEX_ITEM.DISPLAY_AND_SAVE(40, FAMILIA) FAMILIA,
       APEX_ITEM.DISPLAY_AND_SAVE(50, TALLA) TALLA,
       APEX_ITEM.TEXT (10, 0, p_size => 10, p_attributes => '
 from PRODUCTOS
```

APEX_ITEM.DISPLAY_AND_SAVE – Used to display values without the user being able to modify it.
APEX_ITEM.TEXT - Used to create text-type input fields that allow users to enter and modify data dynamically. This feature is essential for capturing user information in forms and reports.
See documentation:
https://docs.oracle.com/en/database/oracle/application-express/20.1/aeapi/TEXT-Function.html#GUID-E63A7E4A-C015-4175-845C-A4501026F9D9

Parameters

Table 21-19 TEXT Parameters

Parameter	Description
p_idx	Number to identify the item you want to generate. The number determines which G_FXX global is populated. **See Also:** "APEX_APPLICATION"
p_value	Value of a text field item.
p_size	Controls HTML tag attributes (such as disabled).
p_maxlength	Maximum number of characters that can be entered in the text box.
p_attributes	Extra HTML parameters you want to add.
p_item_id	HTML attribute ID for the <input> tag.
p_item_label	Invisible label created for the item.

Note: set the *Escape special characters* attribute to OFF in each of the columns.

2. As a result of the query in our report it is the following:

Id	Descripction	Familia	Talla	Cantidad	Cod Producto
1	KID SWEATER COLOR	KIDS	XS	0	P001
10	SWEATER BLACK	SWEATER	M	0	P005
11	HAT	HAT	UNI	0	P006
12	HAT BLACK	HAT	UNI	0	P006
2	KID SWEATER COLOR	KIDS	S	0	P001

1 - 5 Next

3. We create a PLSQL process in Processing where it collects the data entered by the user, validates it and then inserts or updates it in the database. Go through record by record of the classic report created.

COPY

```
DECLARE
  -- Definir el tipo de registro
  TYPE PRODUCTOS_INGRESOS IS RECORD (
    IDPRODUCT     NUMBER,
    DESCRIPCTION  VARCHAR2(250),
    XS            NUMBER,
    S             NUMBER,
    M             NUMBER,
    L             NUMBER,
    XL            NUMBER,
    XXL           NUMBER,
    UNI           NUMBER,
    FAMILIA       VARCHAR2(45)
  );

  -- Definir el tipo de tabla
  TYPE PRODUCTOS_INGRESOS_TABLE IS TABLE OF PRODUCTOS_INGRESOS

  -- Declarar variables de los tipos definidos
  type_productos        PRODUCTOS_INGRESOS;
  type_tabla_productos  PRODUCTOS_INGRESOS_TABLE := PRODUCTOS_
  vw_msg_error          VARCHAR2(500);
BEGIN
  -- Recopilar los datos en el array
  FOR i IN 1 [illegible] COUNT LOOP
    IF (APEX_[illegible] NULL AND TO_NUMBER(AP
        type_productos.IDPRODUCT     := TO_NUMBER(APEX_APPLICA
```

```
        type_productos.DESCRIPCTION  := APEX_APPLICATION.G_F36

        -- Inicializar todas las cantidades a NULL
        type_productos.XS  := NULL;
        type_productos.S   := NULL;
        type_productos.M   := NULL;
        type_productos.L   := NULL;
        type_productos.XL  := NULL;
        type_productos.XXL := NULL;
        type_productos.UNI := NULL;

        -- Asignar el valor correcto basado en el tamaño
        IF APEX_APPLICATION.G_F50(i) = 'XS' THEN
            type_productos.XS := nvl(TO_NUMBER(APEX_APPLICATIO
        ELSIF APEX_APPLICATION.G_F50(i) = 'S' THEN
            type_productos.S := nvl(TO_NUMBER(APEX_APPLICATION
        ELSIF APEX_APPLICATION.G_F50(i) = 'M' THEN
            type_productos.M := nvl(TO_NUMBER(APEX_APPLICATION
        ELSIF APEX_APPLICATION.G_F50(i) = 'L' THEN
            type_productos.L := nvl(TO_NUMBER(APEX_APPLICATION
        ELSIF APEX_APPLICATION.G_F50(i) = 'XL' THEN
            type_productos.XL := nvl(TO_NUMBER(APEX_APPLICATIO
        ELSIF APEX_APPLICATION.G_F50(i) = 'XXL' THEN
            type_productos.XXL := nvl(TO_NUMBER(APEX_APPLICATI
        ELSIF APEX_APPLICATION.G_F50(i) = 'UNI' THEN
            type_productos.UNI := nvl(TO_NUMBER(APEX_APPLICATI
        END IF;

        type_productos.FAMILIA := APEX_APPLICATION.G_F40(i);

        type_tabla_productos.EXTEND;
        type_tabla_productos(type_tabla_productos.COUNT) := ty
    end if;
END LOOP;

-- Insertar o actualizar los datos en la tabla
FOR i IN 1 .. type_tabla_productos.COUNT LOOP
  BEGIN
    UPDATE productos_ingresos
    SET
      DESCRIPCTION = type_tabla_productos(i).DESCRIPCTION,
      XS           = nvl(XS,0) + type_tabla_productos(i).XS,
      S                          tabla_productos(i).S,
      M                          abla_productos(i).M,
      L            = nvl(L,0) + type_tabla_productos(i).L,
```

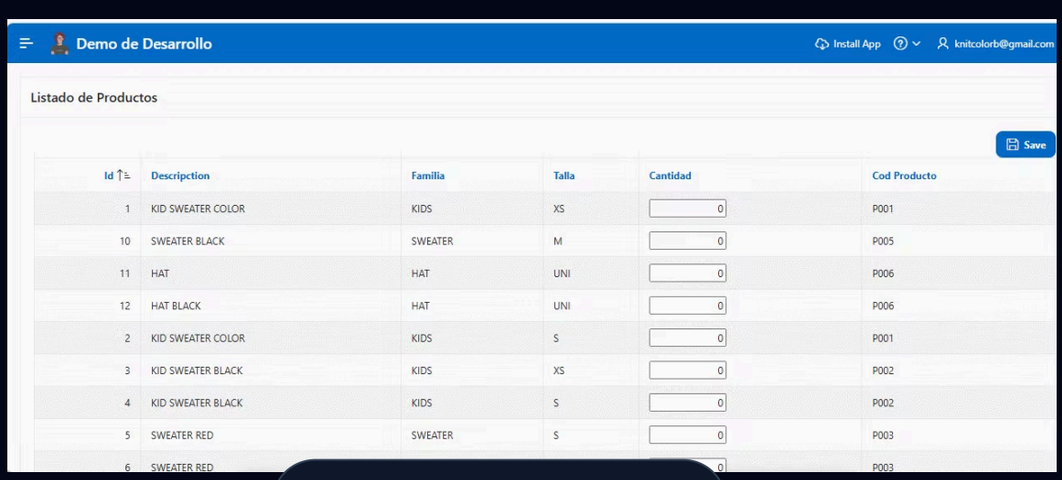
```
          XL           = nvl(XL,0) + type_tabla_productos(i).XL,
          XXL          = nvl(XXL,0) + type_tabla_productos(i).XX
          UNI          = nvl(UNI,0) + type_tabla_productos(i).UN
          FAMILIA      = type_tabla_productos(i).FAMILIA
        WHERE IDPRODUCT = type_tabla_productos(i).IDPRODUCT;

        -- Verificar si se actualizó alguna fila
        IF SQL%ROWCOUNT = 0 THEN
          -- Si no se actualizó ninguna fila, insertar una nueva
          INSERT INTO productos_ingresos (
            IDPRODUCT, DESCRIPCTION, XS, S, M, L, XL, XXL, UNI,
          ) VALUES (
            type_tabla_productos(i).IDPRODUCT,
            type_tabla_productos(i).DESCRIPCTION,
            type_tabla_productos(i).XS,
            type_tabla_productos(i).S,
            type_tabla_productos(i).M,
            type_tabla_productos(i).L,
            type_tabla_productos(i).XL,
            type_tabla_productos(i).XXL,
            type_tabla_productos(i).UNI,
            type_tabla_productos(i).FAMILIA
          );
        END IF;
      EXCEPTION
        WHEN OTHERS THEN
          -- Manejar cualquier error que ocurra durante el inser
          vw_msg_error := 'Error al insertar/actualizar los dato
          APEX_ERROR.ADD_ERROR(
            P_MESSAGE          => vw_msg_error,
            P_DISPLAY_LOCATION => APEX_ERROR.C_INLINE_IN_NOTIFIC
          );
          RETURN;  -- Salir del bloque en caso de error
      END;
    END LOOP;

  EXCEPTION
    WHEN OTHERS THEN
      vw_msg_error := 'Error en el proceso principal: ' || SQLER
      APEX_ERROR.ADD_ERROR(
        P_MESSAGE          => vw_msg_error,
        P_DISPLAY_LOCATION => APEX_ERROR.C_INLINE_IN_NOTIFICATIO
```

```
    );
END;
```

4. Process explanation:
   - **Define Types and Variables:**
     - The types of records and tables necessary to store product data are defined.
     - The variables necessary to handle the data are initialized.
   - **Data collection:**
     - Each user input (`APEX_APPLICATION.G_F10`) is looped and data is collected.
     - The data is assigned to a record `PRODUCTOS_INGRESOS` and then added to the table `PRODUCTOS_INGRESOS_TABLE` using `EXTEND` to increase the size of the collection.
   - **Insertion and Update:**
     For each record in `type_tabla_productos`, an attempt is made to update the table `productos_ingresos`. If the update does not affect any rows (meaning the record does not exist), a new record is inserted. In case of error, an error message is captured and displayed.
5. As a final result we obtain the following, additionally to be able to observe the result when inserting or updating the table, `productos_ingresos,` a region has been created with a classic report based on that table:



Id	Descripction	Familia	Talla	Cantidad	Cod Producto
1	KID SWEATER COLOR	KIDS	XS	0	P001
10	SWEATER BLACK	SWEATER	M	0	P005
11	HAT	HAT	UNI	0	P006
12	HAT BLACK	HAT	UNI	0	P006
2	KID SWEATER COLOR	KIDS	S	0	P001
3	KID SWEATER BLACK	KIDS	XS	0	P002
4	KID SWEATER BLACK	KIDS	S	0	P002
5	SWEATER RED	SWEATER	S	0	P003
6	SWEATER RED			0	P003