# How to Populate  Grid column value automatically without Page submitting

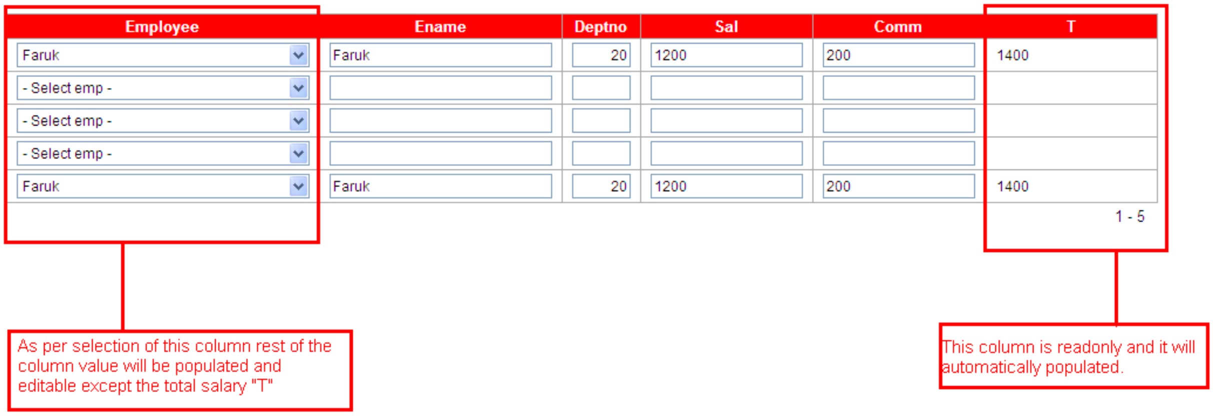
Prepared  by :

M. Kamal Hossain

Emaill : dbhossain@yahoo.com

Solution :

At the first stage let's we take a look how our complete solution will look like.



| Employee | Ename | Deptno | Sal | Comm | T |
|---|---|---|---|---|---|
| Faruk | Faruk | 20 | 1200 | 200 | 1400 |
| - Select emp - | | | | | |
| - Select emp - | | | | | |
| - Select emp - | | | | | |
| Faruk | Faruk | 20 | 1200 | 200 | 1400 |

1 - 5

As per selection of this column rest of the column value will be populated and editable except the total salary "T"

This column is readonly and it will automatically populated.

Fig : 1 Displaying the Final Stage

Now lets we proceed step by step how to do that

## Step : 1

You have to create two  Application Items. Now below showing how to create Application Items.

Go to Shared components > Application Items  : Click Application Items as shown in the Fig-2 below

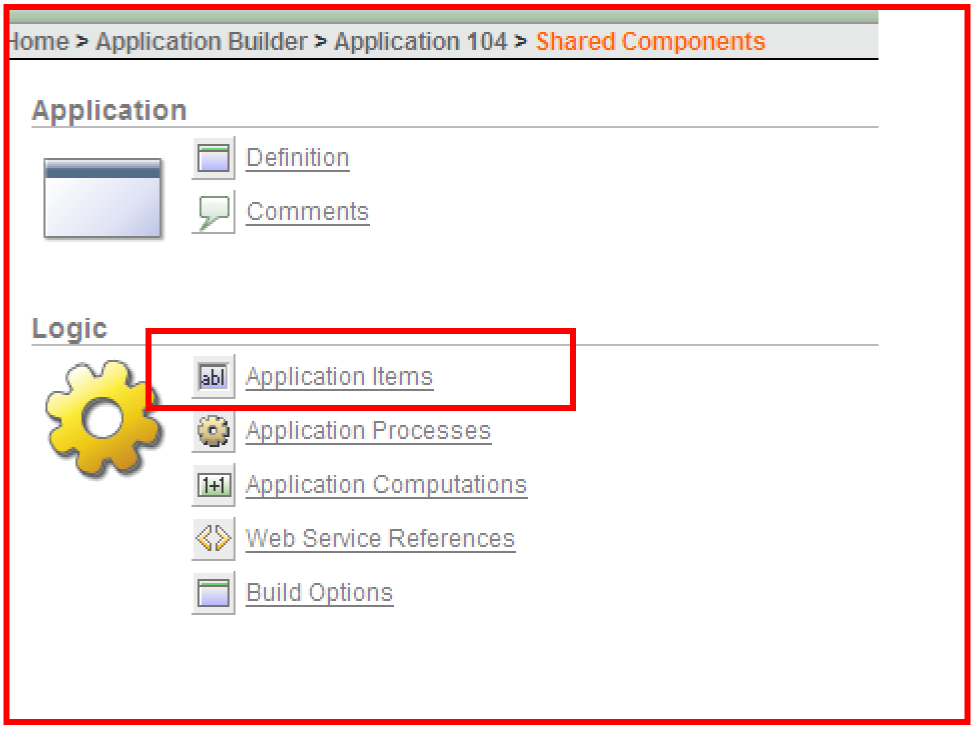


Fig : 2 Application Items creating

Now Create Two appication items Named : G_ID and G_NUM

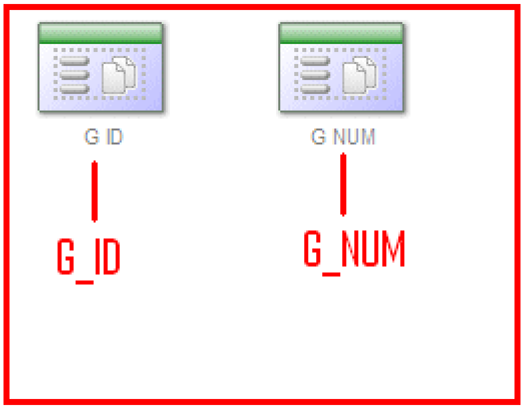
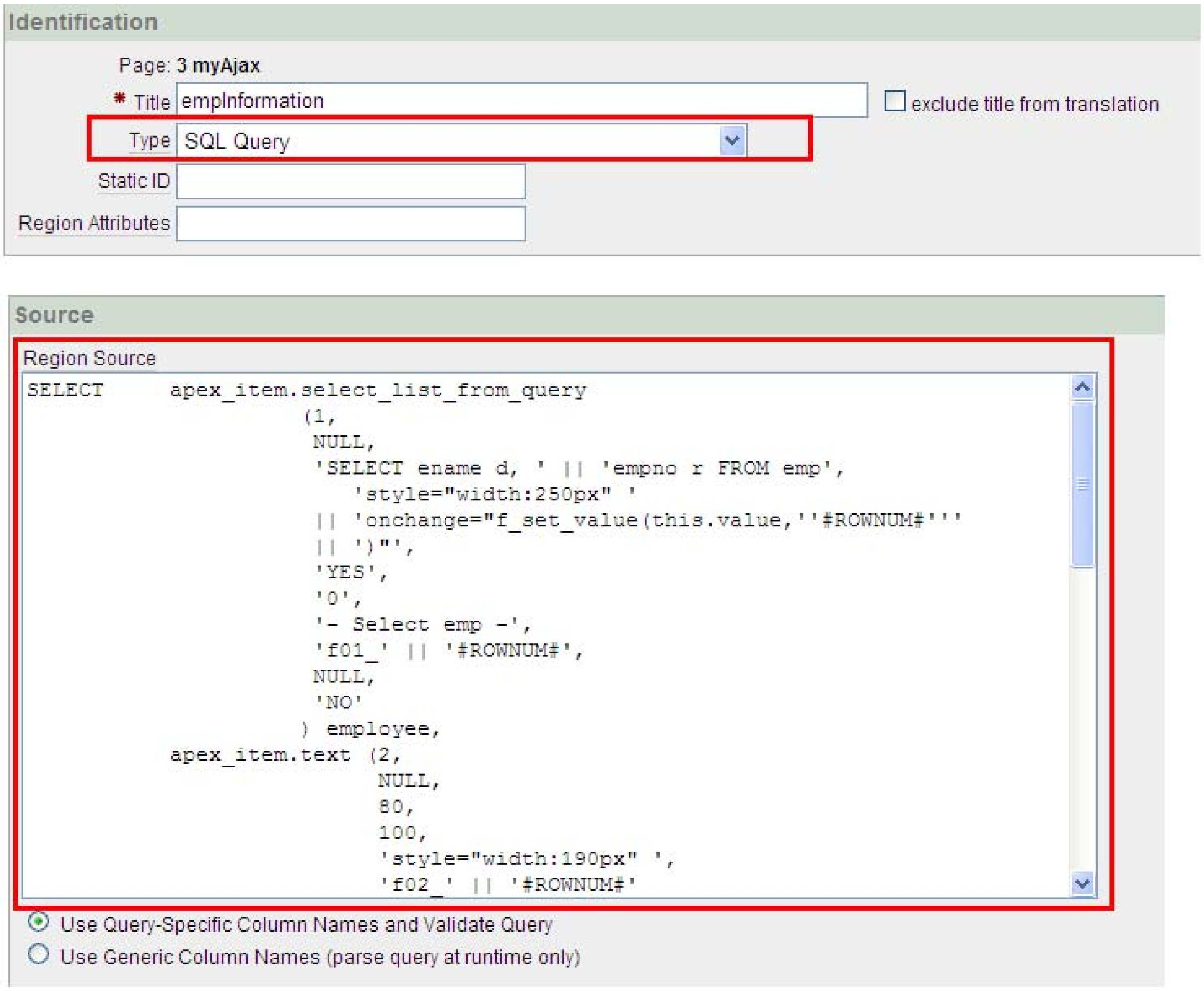


Fig : 3 Application Item List

## Step : 2

**Create a SQL Report as shown in Fig : 4 and the Query source below :**



Page: 3 myAjax

* Title: empInformation

□ exclude title from translation

Type: SQL Query

Static ID:

Region Attributes:

Region Source

```
SELECT     apex_item.select_list_from_query
                      (1,
                       NULL,
                       'SELECT ename d, ' || 'empno r FROM emp',
                          'style="width:250px" '
                       || 'onchange="f_set_value(this.value,''#ROWNUM#'''
                       || ')"',
                       'YES',
                       '0',
                       '- Select emp -',
                       'f01_' || '#ROWNUM#',
                       NULL,
                       'NO'
                      ) employee,
           apex_item.text (2,
                            NULL,
                            80,
                            100,
                            'style="width:190px" ',
                            'f02_' || '#ROWNUM#'
```

● Use Query-Specific Column Names and Validate Query

○ Use Generic Column Names (parse query at runtime only)

**Fig: 4 Showing SQL Report**

**SQL Query Source:**

-------------------------------------------------------------------------------------------------------------------------------------

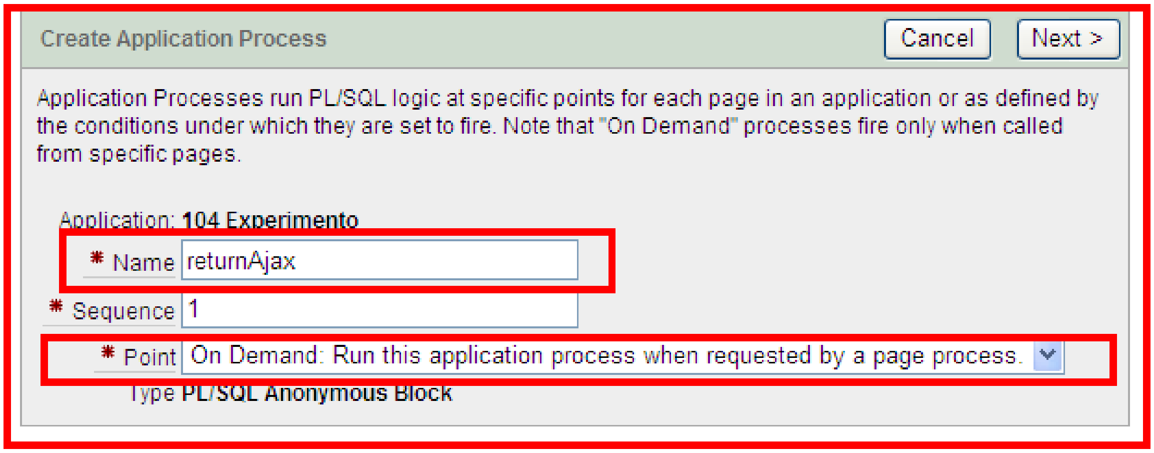
```
SELECT     apex_item.select_list_from_query

                (1,

                 NULL,

                 'SELECT ename d, ' || 'empno r FROM emp',
                   'style="width:120px" '
                 || 'onchange="f_set_value(this.value,''#ROWNUM#'''
                 || ')"',
                 'YES',
                 '0',
                 '- Select emp -',
                 'f01_' || '#ROWNUM#',
                 NULL,
                 'NO'
                ) employee,
            apex_item.text (2,
                    NULL,
                    80,
                    100,
                    'style="width:100px" ',
                    'f02_' || '#ROWNUM#'
                   ) ename,
            apex_item.text (3,
                    NULL,
                    80,
                    100,
                    'style="width:50px;text-align:right" ',
                    'f03_' || '#ROWNUM#'
                   ) deptno,
            apex_item.text (4,
                    NULL,
                    80,
                    100,
                    'style="width:130px" ',
                    'f04_' || '#ROWNUM#'
                   ) sal,

            apex_item.text (5,
                    NULL,
                    80,
                    100,
                    'style="width:130px" ',
                    'f05_' || '#ROWNUM#'
                   ) comm,

            apex_item.text (6,
                    NULL,
                    80,
                    100,
                    'style="width:130px;border:none;background:none;" '
                    ||'readonly="readonly"',
                    'f06_' || '#ROWNUM#'
                   ) t

         FROM emp
```

**Note: you can directly copy the above Query Text and paste into your SQL Query source of the Report.**

## Step : 3

**You have to create an Application Process: Go to Shared Components > Application Processes.**

Create a process Named : returnAjax as shown in Fig : 5



Create Application Process

Application Processes run PL/SQL logic at specific points for each page in an application or as defined by the conditions under which they are set to fire. Note that "On Demand" processes fire only when called from specific pages.

Application: **104 Experimento**

* Name: returnAjax

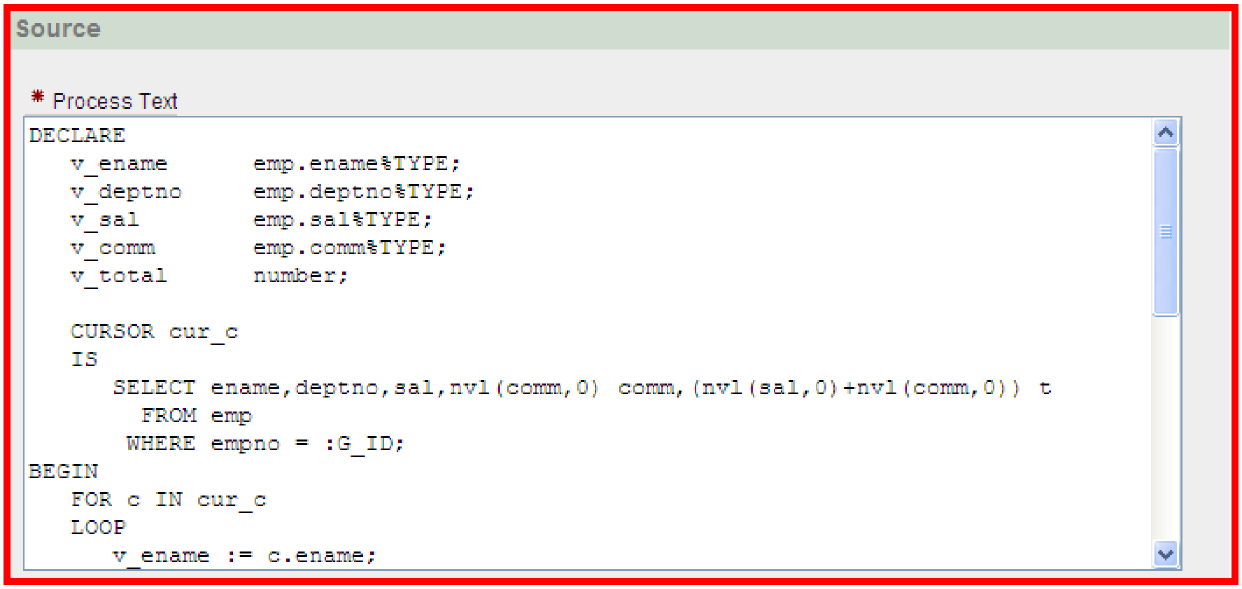
* Sequence: 1

* Point: On Demand: Run this application process when requested by a page process.

Type: **PL/SQL Anonymous Block**

Fig: 5 showing How to create an Application Process named: returnAjax

Now type the Process Text in the Source as shown in Fig : 6 and Source Text below :



Source

* Process Text

```
DECLARE
   v_ename      emp.ename%TYPE;
   v_deptno     emp.deptno%TYPE;
   v_sal        emp.sal%TYPE;
   v_comm       emp.comm%TYPE;
   v_total      number;

   CURSOR cur_c
   IS
      SELECT ename,deptno,sal,nvl(comm,0)  comm,(nvl(sal,0)+nvl(comm,0))  t
        FROM emp
       WHERE empno = :G_ID;
BEGIN
   FOR c IN cur_c
   LOOP
      v_ename := c.ename;
```

Fig : 6 Showing Source Text of the Process named " returnAjax"

## Source Text :

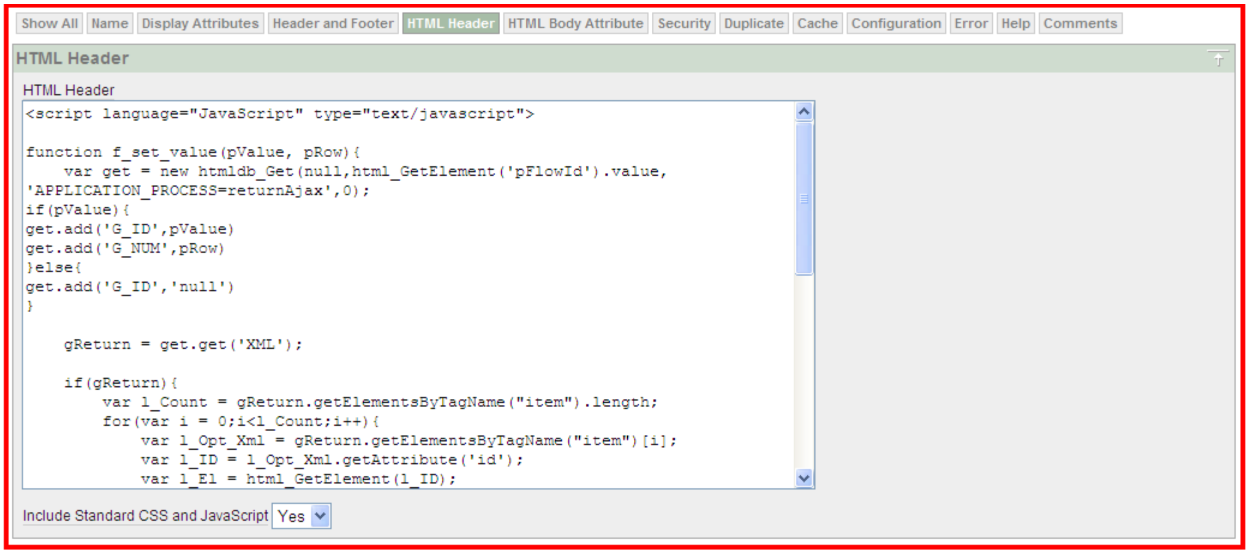
```
------------------------------------------------------------------------------------------------------------------------------------------
DECLARE
  v_ename      emp.ename%TYPE;
  v_deptno     emp.deptno%TYPE;
  v_sal        emp.sal%TYPE;
  v_comm       emp.comm%TYPE;
  v_total      number;

  CURSOR cur_c
  IS
    SELECT ename,deptno,sal,nvl(comm,0) comm,(nvl(sal,0)+nvl(comm,0)) t
      FROM emp
     WHERE empno = :G_ID;
BEGIN
  FOR c IN cur_c
  LOOP
    v_ename := c.ename;
    v_deptno := c.deptno;
    v_sal := c.sal;
    v_comm := c.comm;
    v_total := c.t;

  END LOOP;

  OWA_UTIL.mime_header ('text/xml', FALSE);
  HTP.p ('Cache-Control: no-cache');

  HTP.p ('Pragma: no-cache');

  OWA_UTIL.http_header_close;

  HTP.prn ('<body>');

  HTP.prn ('<desc>this xml genericly sets multiple items</desc>');

  HTP.prn ('<item id="f02_' || :G_NUM || '">' || v_ename || '</item>');

  HTP.prn ('<item id="f03_' || :G_NUM || '">' || v_deptno || '</item>');

  HTP.prn ('<item id="f04_' || :G_NUM || '">' || v_sal || '</item>');

  HTP.prn ('<item id="f05_' || :G_NUM || '">' || v_comm || '</item>');

  HTP.prn ('<item id="f06_' || :G_NUM || '">' || v_total || '</item>');


  HTP.prn ('</body>');

END;
```

---

**Note: you can copy the text above and paste into the Source Text body**

## Step : 4

**Create a JavaScript in your application page. Your page > HTML header**



HTML Header

```
<script language="JavaScript" type="text/javascript">

function f_set_value(pValue, pRow){
    var get = new htmldb_Get(null,html_GetElement('pFlowId').value,
'APPLICATION_PROCESS=returnAjax',0);
if(pValue){
get.add('G_ID',pValue)
get.add('G_NUM',pRow)
}else{
get.add('G_ID','null')
}

    gReturn = get.get('XML');

    if(gReturn){
        var l_Count = gReturn.getElementsByTagName("item").length;
        for(var i = 0;i<l_Count;i++){
            var l_Opt_Xml = gReturn.getElementsByTagName("item")[i];
            var l_ID = l_Opt_Xml.getAttribute('id');
            var l_El = html_GetElement(l_ID);
```

Include Standard CSS and JavaScript: Yes

**Fig: 7 Showing JavaScript Code in the Page HTML header**

# JavaScript Source:

---

```
<script language="JavaScript" type="text/javascript">


function f_set_value(pValue, pRow){

    var get = new htmldb_Get(null,html_GetElement('pFlowId').value,

'APPLICATION_PROCESS=returnAjax',0);

if(pValue){

get.add('G_ID',pValue)

get.add('G_NUM',pRow)

}else{

get.add('G_ID','null')

}


    gReturn = get.get('XML');
    if(gReturn){
        var l_Count = gReturn.getElementsByTagName("item").length;
        for(var i = 0;i<l_Count;i++){
            var l_Opt_Xml = gReturn.getElementsByTagName("item")[i];
            var l_ID = l_Opt_Xml.getAttribute('id');
            var l_El = html_GetElement(l_ID);
            if(l_Opt_Xml.firstChild){
                var l_Value = l_Opt_Xml.firstChild.nodeValue;
            }else{
                var l_Value = '';
            }

            if(l_El){
                if(l_El.tagName == 'INPUT'){
                    l_El.value = l_Value;
                }else if(l_El.tagName == 'SPAN' && l_El.className == 'grabber'){
                    l_El.parentNode.innerHTML = l_Value;
                    l_El.parentNode.id = l_ID;
                }else{
                    l_El.innerHTML = l_Value;
                }
            }
        }
    }
    get = null;
}
```

```
</script>
```

**Note: you can directly copy the above JavaScript Code and Paste into the HTML header.**

**Now you can run your application pages and Change the Employee from the Grid  and you will notice rest of the column information coming automation as per selection without submitting / Refreshing Page.**

**Table Structure:**

**For above example below is the table definition**

```
CREATE TABLE  "EMP"
   (    "EMPNO" NUMBER(10,0),
        "ENAME" VARCHAR2(40),
        "DEPTNO" NUMBER(2,0),
        "SAL" NUMBER(10,2),
        "COMM" NUMBER(10,2)
   )
```

**Below is the data :**

**7369,Kamal,10,12000,1200**

**4562,Faruk,20,1200,200**

**8942,Saiful,10,3200,100**

**8941,Mamun,10,2500,1200**

**7125,Nazrul,20,2500,200**

**** End ****

**Thank you**