

L6

Crearea principalelor tipuri de formulare si rapoarte

1. Obiective

- Crearea unui raport
- Crearea unui raport pornind de la un formular simplu (Blank Page)

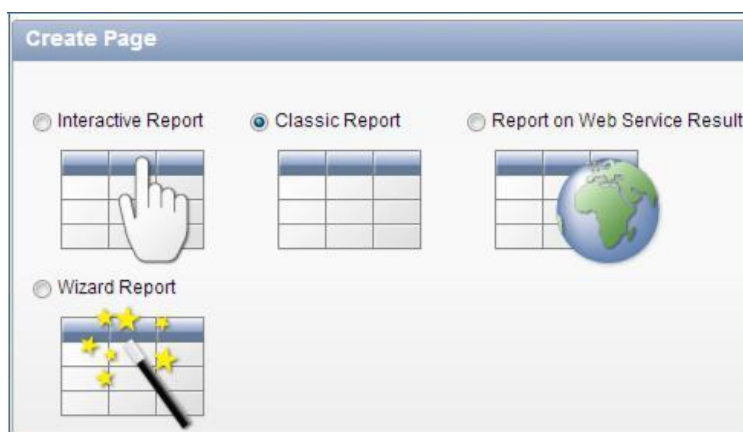
2. Considerații teoretice

f. Crearea unui raport

Un raport este bazat de obicei pe o comandă SQL "SELECT". Pentru a genera un astfel de raport se inițiază adăugarea unei noi pagini aplicației curente și în prima fereastră afișată se selectează "Report".



În fereastra următoare se selectează tipul de raport ("Classic Report").



Create SQL ReportCancelNext >

Identify page number and name. If the page number you specify does not exist, the wizard creates the page for you.

Application: **101 - PERSO1**

* Page Number

* Page Name

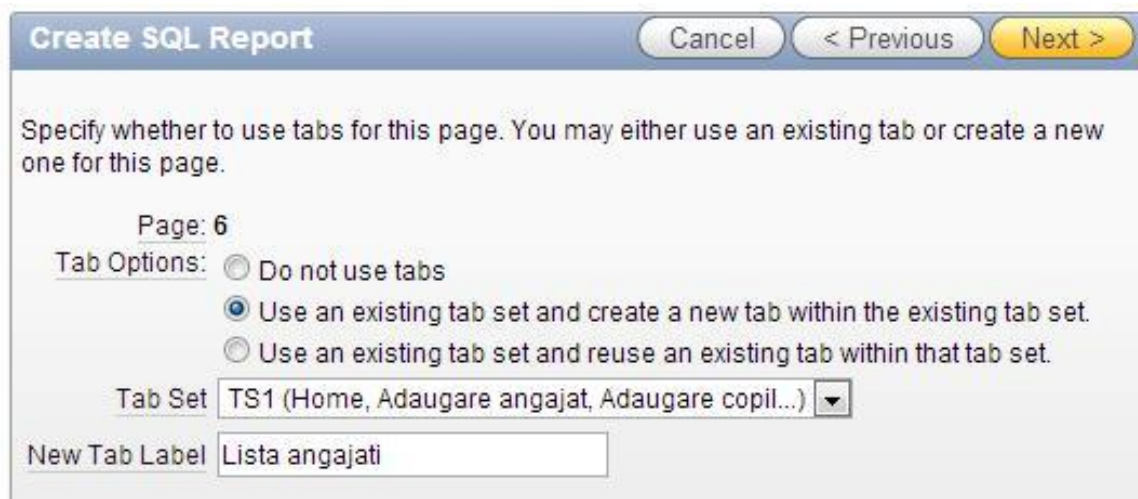
Breadcrumb ▼

Create Breadcrumb Entry

Entry Name

Parent Entry

În continuare se definește un tab a cărui selectare va genera raportul.



Create SQL Report [Cancel] [< Previous] [Next >]

Specify whether to use tabs for this page. You may either use an existing tab or create a new one for this page.

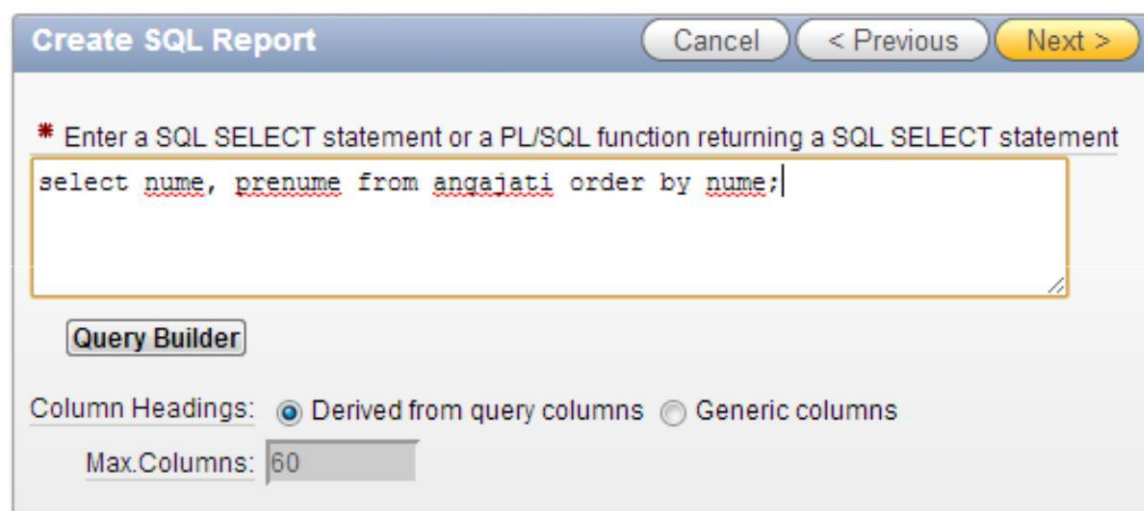
Page: 6

Tab Options: ☐ Do not use tabs
☒ Use an existing tab set and create a new tab within the existing tab set.
☐ Use an existing tab set and reuse an existing tab within that tab set.

Tab Set: TS1 (Home, Adaugare angajat, Adaugare copil...) ▼

New Tab Label: Lista angajati

În fereastra următoare se introduce comanda SQL "SELECT" care va produce liniile raportului.



Create SQL Report [Cancel] [< Previous] [Next >]

* Enter a SQL SELECT statement or a PL/SQL function returning a SQL SELECT statement

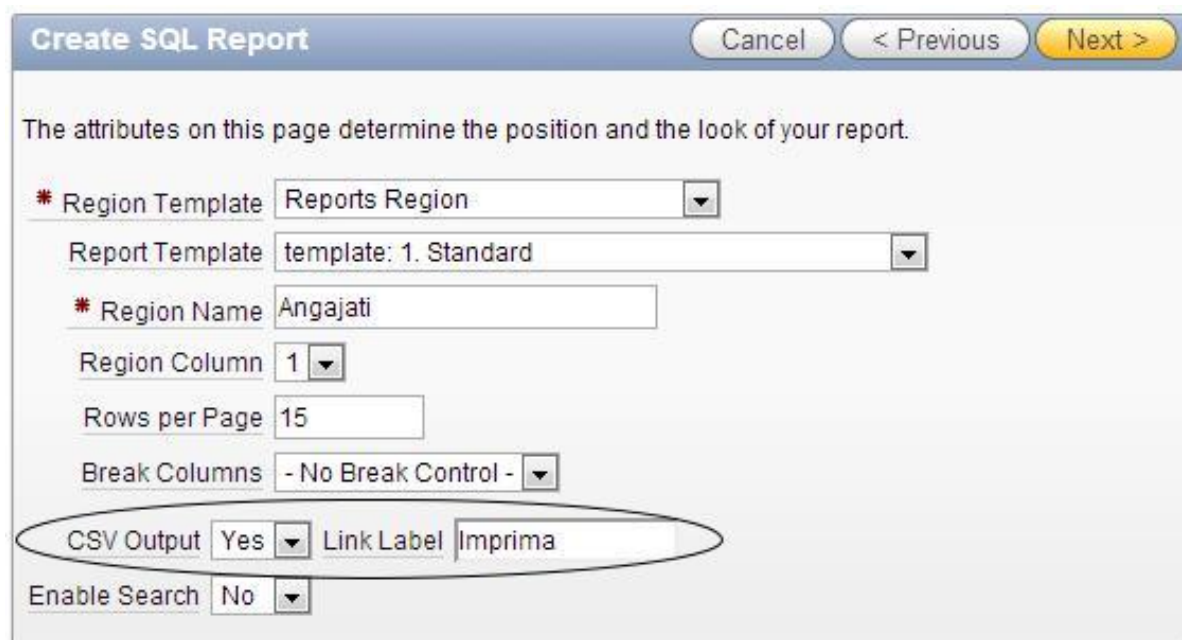
```
select nume, prenume from angajati order by nume;
```

[Query Builder]

Column Headings: ☒ Derived from query columns ☐ Generic columns

Max. Columns: 60

În fereastra următoare se configurează raportul (aspect, număr de coloane, număr de linii pe pagină) și se impune exportarea sa în format CSV (Comma Separated Values, format acceptat de Excel).



Create SQL Report [Cancel] [< Previous] [Next >]

The attributes on this page determine the position and the look of your report.

* Region Template: Reports Region ▼

Report Template: template: 1. Standard ▼

* Region Name: Angajati

Region Column: 1 ▼

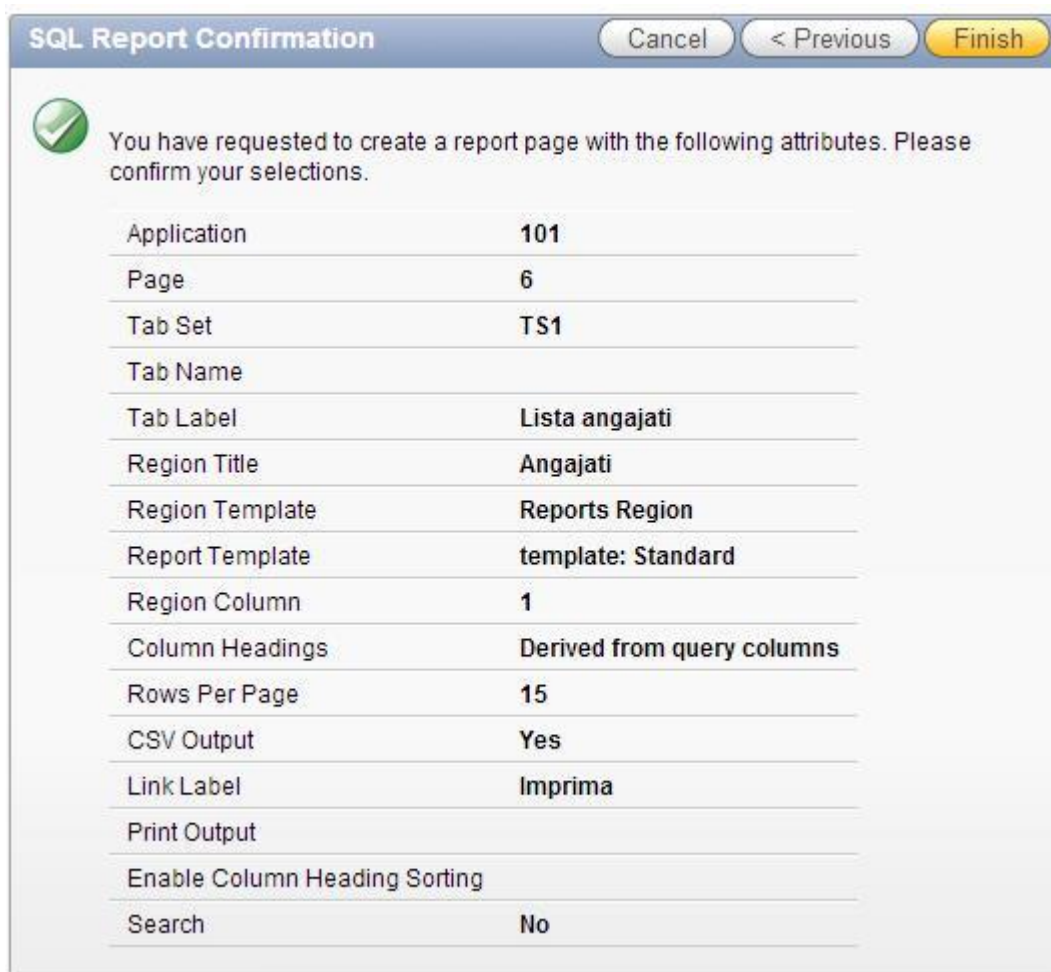
Rows per Page: 15

Break Columns: - No Break Control - ▼

CSV Output: Yes ▼ Link Label: Imprima

Enable Search: No ▼

Ultima fereastră afișată conține o recapitulare finală a opțiunilor impuse.



The image shows a 'SQL Report Confirmation' dialog box. It has a title bar with 'Cancel', '< Previous', and 'Finish' buttons. A green checkmark icon is next to the text: 'You have requested to create a report page with the following attributes. Please confirm your selections.' Below this is a table of attributes and their values.

Application	101
Page	6
Tab Set	TS1
Tab Name	
Tab Label	Lista angajati
Region Title	Angajati
Region Template	Reports Region
Report Template	template: Standard
Region Column	1
Column Headings	Derived from query columns
Rows Per Page	15
CSV Output	Yes
Link Label	Imprima
Print Output	
Enable Column Heading Sorting	
Search	No

Rezultatul este realizarea raportului pe baza comenzii SQL date. În partea inferioară a paginii apare referința "Imprima" a cărei selectare provoacă crearea fișierului "angajati.csv" care poate fi deschis și prelucrat în continuare în Excel.

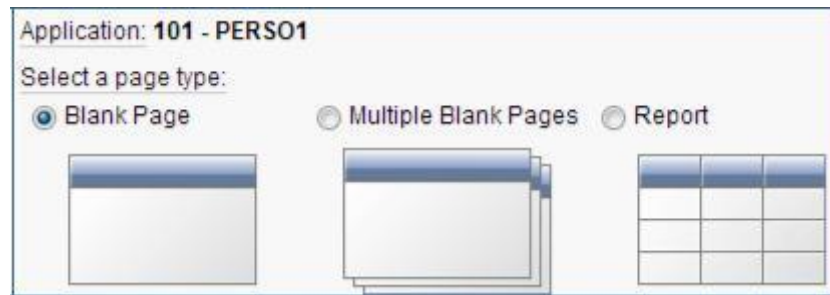


The image is a screenshot of a web application. At the top, there is a navigation bar with buttons: 'Home', 'Adaugare angajat', 'Adaugare copil', 'Copii angajatilor', and 'Lista angajati'. Below this is a breadcrumb trail: 'Home > lista personal'. The main content area is titled 'Angajati' and contains a table with two columns: 'NUME' and 'PRENUME'. The table has four rows of data: 'Avram Lucian', 'Ionescu Valeriu', 'Pop Vasile', and 'Tanase Maria'. Below the table, there is a button labeled 'Imprima' which is circled in red. At the bottom right of the table area, it says '1 - 4'.

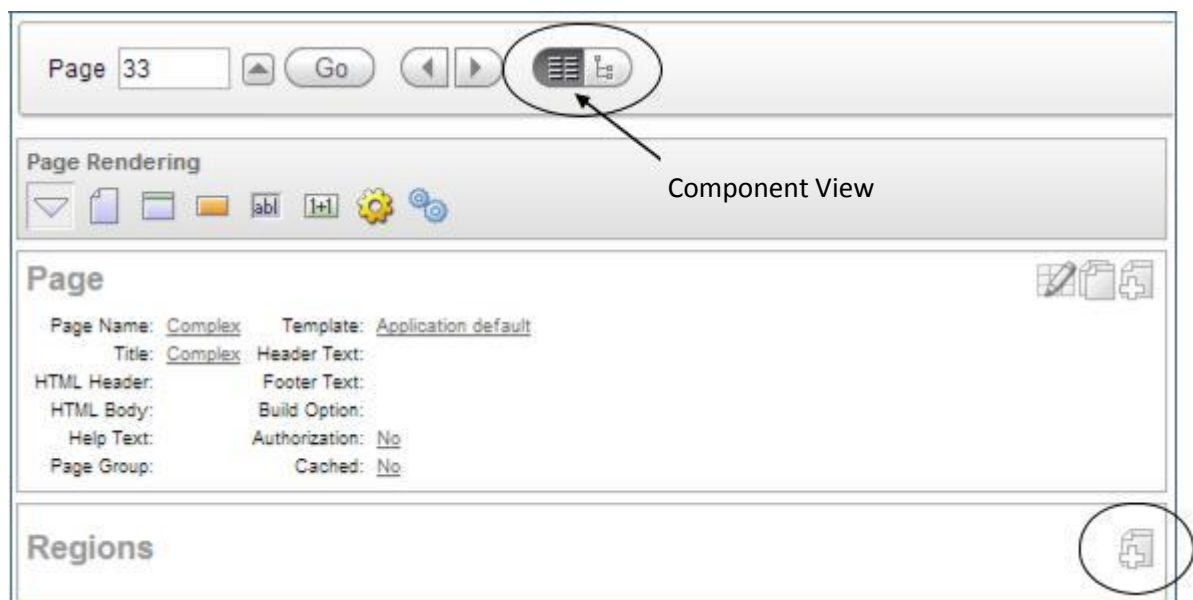
g, Crearea unui raport pornind de la un formular simplu (Blank Page)

În APEX există posibilitatea realizării unor pagini având structuri complexe, adaptate unor cerințe neacoperite de variantele expuse de aplicația expert. Un exemplu este realizarea unei pagini conținând un raport al cărui conținut depinde de elementul selectat dintr-un control de tip listă aflat pe aceeași pagină.

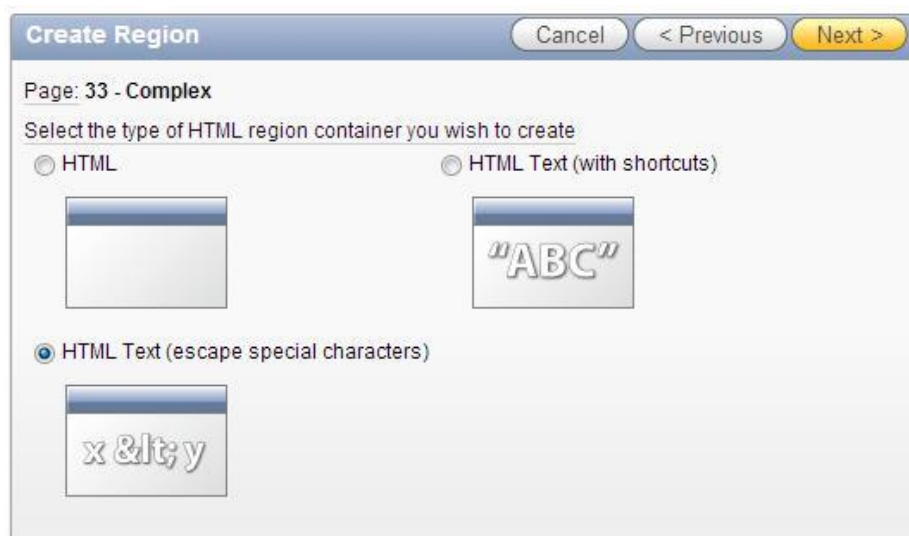
Pentru crearea unei astfel de pagini în aplicația dezvoltată va fi adăugată o pagină goală ("Blank Page").



După adăugarea paginii se va adăuga o primă regiune. Pentru a opera mai ușor modificările impuse pagina se va afișa selectând butonul "Component View".



Ca și tip de regiune se poate alege "HTML Text".



În continuare se configurează regiunea adăugată. I se dă un titlu și ca și tip se selectează "Button Region without Title", ca în imagine.

Create Region Cancel < Previous Next >

Page: 33 - Complex
Region Source Type: HTML Text (escape special characters)

* Title ←

Region Template ←

Parent Region

Display Point ↗

[Body] [Pos.1] [Pos.2] [Pos.3] [Pos.4] [Pos.5]

* Sequence Column

Pentru a adăuga controale Windows unei regiuni se va selecta "Create" din zon "Items". În exemplul dat se va crea un control de tip "Select List" care se va adăuga regiunii "Selectare Angajat". Numele dat controlului va fi "P33_selectare".

Page Rendering

Page

Page Name: [Complex](#) Template: [Application default](#)
 Title: [Complex](#) Header Text:
 HTML Header: Footer Text:
 HTML Body: Build Option:
 Help Text: Authorization: [No](#)
 Page Group: Cached: [No](#)

Regions

Display Point: Page Template Body (3)
 10 ☐ [Selectare angajat](#) HTML ESC

Buttons

Items

Create Item

Page: 33 - Complex

Select Item Type:

☐ Check Box ☐ Date Picker ☐ Display Image ☐ Display Only

☒ Select List ☐ Shuttle ☐ Stop and start table ☐ Text Area

Create Item Cancel < Previous Next >

Page: 33 - Complex
Display As: **Select List**

* Item Name

* Sequence

* Region

Create Item [Cancel] [< Previous] [Next >]

Page: 33 - Complex
 Item Name: P33_SELECTARE
 Display As: Select List
 Label: Selectare angajat [Clear]
 Label Alignment: Right
 Height: 1
 Field Alignment: Left center
 Label Template: Optional with help
 Begin On New Line: Yes

Pentru a provoca refacerea conținutului raportului care va fi adăugat paginii, Pentru "Page Action when Value Changed" se selectează "Submit Page".

Create Item [Cancel] [< Previous] [Next >]

Page: 33 - Complex
 Item Name: P33_SELECTARE
 Display As: Select List
 Value Required: No
 Page Action when Value Changed: Submit Page
 Execute Validations: Yes

În fereastra următoare se va selecta lista dinamică de valori cu care se populează lista noului control. În aplicație există deja lista "Angajati" care este exact lista necesară.

Identify List of Values [Cancel] [< Previous] [Next >]

Use this page to define the list of values. Either construct a SQL statement with the number of columns required by the item type, or use the STATIC syntax. See the List Of Values Examples section for examples.

Application/Page: 101/33
 Item Name: P33_SELECTARE
 Display As: Select List
 Named LOV: ANGAJATI
 Display Null Value: Yes
 Null Display Value:
 Null Return Value:
 Cascading LOV Parent Item(s):

În ultimul pas se crează elementul din pagină și se testează pagina.

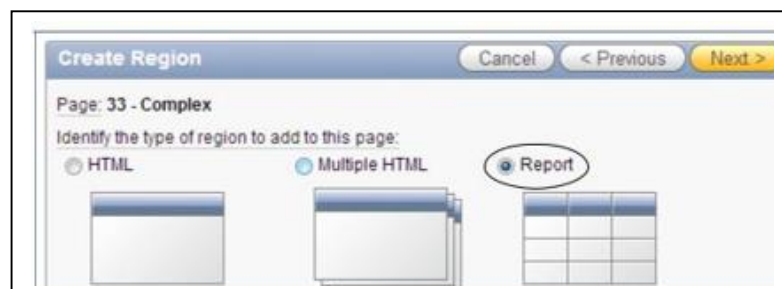
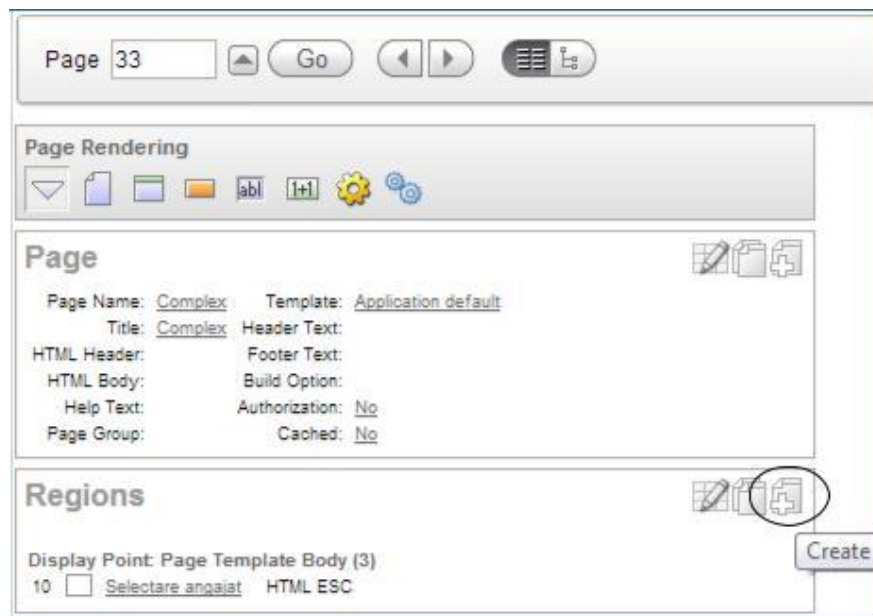
Create Item [Cancel] [< Previous] [Create Item]

Identify the source of the item. If the item source is null the default value will be used.

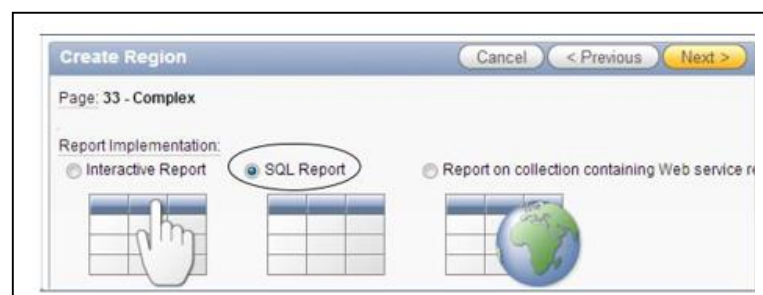
Page: 33 - Complex
 Item Name: P33_SELECTARE
 Display As: Select List
 Source Used: Only when current value in session state is null
 Source Type: Static Assignment (value equals source attribute)



În continuare se va adăuga paginii o regiune nouă conținând un raport. În comanda SELECT utilizată la crearea mulțimii de selecție a datelor din raport va interveni ca parametru controlul "P33_SELECTARE" creat deja.



Ca tip de raport se alege fie "Interactive Report", fie "SQL Report".



Create Region Cancel < Previous Next >

Page: 33 - Complex
Region Source Type: SQL Query

* Title: Lista copiilor

Region Template: Reports Region

Parent Region: - Select a Parent -

Display Point: Page Template Body (3. items above region content)

[Body] [Pos.1] [Pos.2] [Pos.3] [Pos.4] [Pos.5]

* Sequence: 20 Column: 1

Create Region Cancel < Previous Next > Create Region

Page: 33 - Complex
Region Title: Lista copiilor

Enter SQL Query or PL/SQL function returning a SQL Query:

```
select nume, prenume from copii where id_angajat = :P33_SELECTARE
```

În comanda scrisă intervine parametrul: "P33+SELECTARE". Valoarea acestuia este chiar valoarea cheii primare corespunzătoare angajatului selectat, așa cum s-a impus la crearea acestei liste de valori.

Rezultat:

Home Adaugare angajat Adaugare copil

Selectare Pop Vasile

Lista copiilor

NUME	PRENUME
Pop	Traian
Pop	Alexandru

1 - 2

În regiunea "Buttons" se selectează "New Item".

Page Rendering

Page

Page Name: Complex Template: Application default

Title: Complex Header Text:

HTML Header: Footer Text:

HTML Body: Build Option:

Help Text: Authorization: No

Page Group: Cached: No

Regions

Display Point: Page Template Body (3)

10 ☐ Selectare angajat HTML ESC

20 ☒ Lista copiilor Report

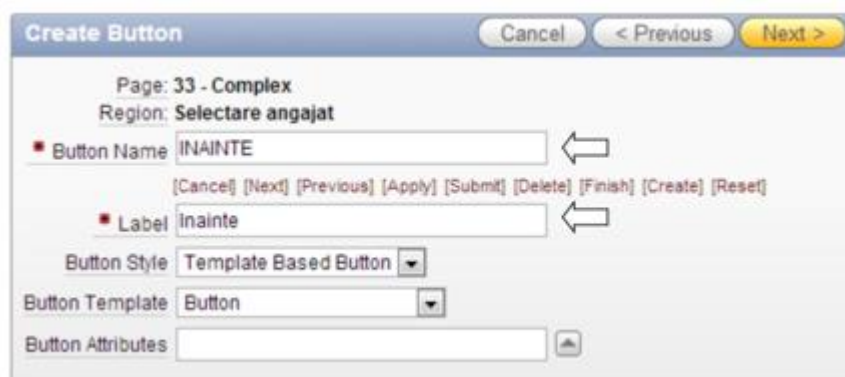
Buttons

New Item

Ca regiune în care să se afișeze noul buton se selectează "Selectare angajat".



În continuare se precizează numele butonului și eticheta cu care va fi inscripționat.

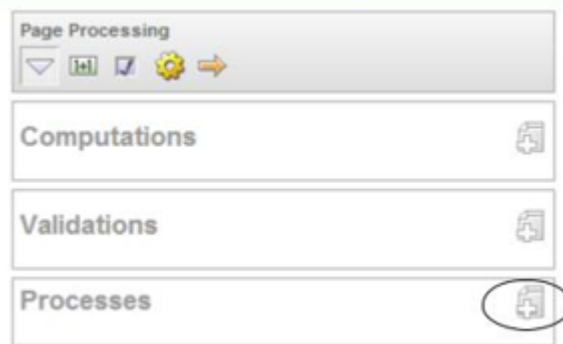


La apăsarea butonului se va realiza reîncărcarea paginii.

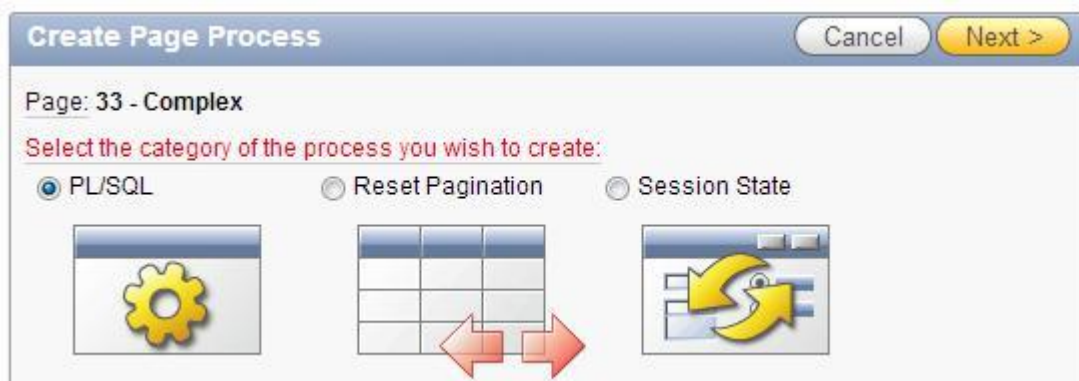


Efectul selectării butonului adăugat este reîncărcarea paginii curente. Pentru ca la reîncărcare angajatul selectat în listă să fie următorul angajat trebuie adăugată o secvență de cod PL/SQL care impune aceasta. Secvența de cod se adaugă astfel:

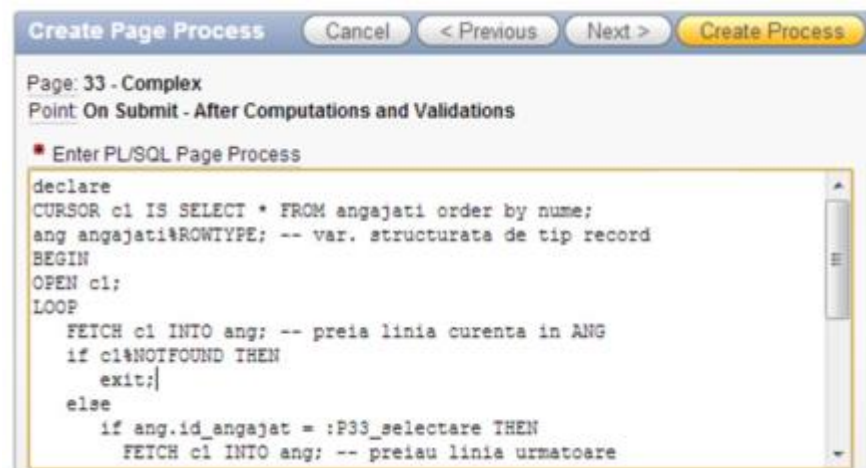
Pagina curentă fiind selectată, în regiunea "Processes" din panoul "Page Processing" se selectează butonul "Create".



Ca tip de proces se selectează PL/SQL.



Se dă apoi procesului un nume și se introduce codul PL/SQL corespunzător.



```
declare
CURSOR c1 IS SELECT * FROM angajati order by nume;
ang angajati%ROWTYPE; -- var. structurata de tip record
BEGIN
OPEN c1;
```


LOOP

```
FETCH c1 INTO ang; -- preia linia curenta in ANG
if c1%NOTFOUND THEN
    exit;
else
    if ang.id_angajat = :P33_selectare THEN
        FETCH c1 INTO ang; -- preiau linia urmatoare
        if c1%NOTFOUND THEN
            exit;
        else
            :P33_selectare := ang.id_angajat;
            exit;
        end if;
    end if;
end if;
END LOOP;
CLOSE c1;
end;
```

Secvența de cod este un bloc anonim PL/SQL care crează și parcurge un cursor având un conținut similar listei de valori ANGAJATI. După găsirea angajatului selectat în lista ANGAJATI se mai citește o linie din cursor și se impune noua valoare ca valoare curentă în controlul cu listă.

După crearea procesului acesta trebuie editat deoarece execuția sa trebuie condiționată de apăsarea butonului creat anterior.

The screenshot displays the Oracle APEX 'Page Processing' configuration page. The 'Processes' tab is selected, showing a table with the following data:

Edit	Sequence	Point	Name
	10	After Page Submit - After Computations and Validations	p_inainte

The 'Edit' button for the process 'p_inainte' is circled in red. The 'Page' number is 33, and the 'Go' button is visible.



Pentru butonul care realizează parcurgerea listei de angajați în sens invers, de la poziția curentă spre înapoi, codul PL/SQL poate fi următorul:

```

declare
  CURSOR c1 IS SELECT * FROM angajati order by nume;
  ang angajati%ROWTYPE; -- var. structurata de tip record
  angurm angajati%ROWTYPE; -- var. structurata de tip record
BEGIN
  OPEN c1;
  FETCH c1 INTO ang; -- preia linia curenta in ANG
  IF ang.id_angajat != :P33_selectare THEN
    LOOP
      FETCH c1 INTO angurm; -- preia linia curenta in
      ANGURM if c1%NOTFOUND THEN
        exit;
      else
        if angurm.id_angajat = :P33_selectare THEN
          :P33_selectare := ang.id_angajat;
          exit;
        else
          ang := angurm;
        end if;
      end if;
    END LOOP;
  end if;
  CLOSE c1;
end;
```

3. Desfășurarea lucrării

Se vor aplica notiunile prezentate în consideratii teoretice utilizand aplicatia individuala aleasa

Probleme propuse spre rezolvare:

1. Verificați folosind SQL Developer dacă la Dv. utilizatorul BIBLIO are drepturile de acces corect impuse și creați în APEX spațiul de lucru BIBLIO cu parola biblio definit peste schema BIBLIO.
2. Creați o aplicație de administrare a schemei BIBLIO având un singur rând de taburi și paginile următoare:
 - i. O pagină denumită "Home" de tip "Blank" conținând o singură regiune. În regiunea inclusă tastați o descriere a aplicației.
 - ii. O pagină de tip "Form" / "Form on a Table or View" destinată editării datelor din tabelului "Edituri".
 - iii. O pagină de tip "Form" / "Form on a Table or View" destinată editării datelor din tabelului "Autori".
 - iv. O pagină de tip "Form" / "Form on a Table or View" destinată editării datelor din tabelului "Cititori".
 - v. O pagină de tip "Form" / "Form on a Table or View" destinată editării datelor din tabelului "Carti". Pentru introducerea codului editurii se va defini o listă de valori de tip "Dynamic" (conținând editurile în ordine alfabetică) și se va modifica tipul controlului Windows corespunzător din "Number Field" în "Select List".
 - vi. O pagină de tip "Form" / "Form on a Table or View" destinată editării datelor din tabelul de legătură "Autcarti". Pentru a putea realiza selectarea valorilor din două controale Windows cu listă se va defini o listă suplimentară de valori de tip "Dynamic" care va conține autorii în ordinea alfabetică, procedându-se apoi ca la punctul precedent.
 - vii. O pagină care va afișa cărțile de la o anumită editură. Pentru realizarea acesteia se va porni de la un formular gol ("Blank Page") și se vor adăuga două regiuni destinate selectării editurii respectiv afișării cărților de la editura selectată.
3. Adăugați aplicației Dv. alte pagini pe care le considerați necesare.
4. Adăugați spațiului de lucru BIBLIO un utilizator simplu și verificați modul de operare a aplicației în cazul conectării folosind noul cont.