

CREAREA ȘI DEZVOLTAREA APLICAȚIILOR UTILIZÂND ORACLE APPLICATION EXPRESS

DIMA NARCISA, ENESCU CĂTĂLINA



CREAREA ȘI DEZVOLTAREA APLICAȚIILOR UTILIZÂND ORACLE APPLICATION EXPRESS

Copyright © 2020
Autori: DIMA NARCISA, ENESCU CĂTĂLINA

Toate drepturile rezervate.

ISBN 978-606-94690-8-8

Editura Evomind, 2020

<https://evomind.org/>

CUPRINS

1. CE ESTE ORACLE APEX?.....	4
2. CREAREA UNEI APLICAȚII CU BAZE DE DATE ÎN ORACLE APEX .	24
3. CREAREA UNEI APLICAȚII WEB MOBILE ÎN ORACLE APEX	38
4. GRAFICE ȘI RAPOARTE BAZATE PE INTEROGĂRI ÎN APLICAȚII ORACLE APEX	43
5. SHUTTLE ITEM IN ORACLE APEX 4.2	56
BIBLIOGRAFIE.....	62

1. CE ESTE ORACLE APEX?

Oracle Application Express (prescurtat APEX, denumit anterior Oracle HTML DB) este un mediu de dezvoltare software bazat pe web care rulează pe o bază de date Oracle. Poate fi utilizat pentru a construi aplicații web complexe care pot rula în majoritatea browserelor web moderne. Mediul de dezvoltare APEX este, de asemenea, bazat pe browser.

APEX permite construirea cu ușurință a aplicațiilor web fără cod. Acolo unde cerințele sunt mai complexe, APEX permite extinderea obiectelor Low Code printr-un cadru declarativ.

Limbajul SQL este utilizat pentru a defini rapoarte, diagrame și alte obiecte ale bazei de date, deci poate fi utilă cunoșterea acestui limbaj.

Dacă este necesară logica procedurală, puteți scrie fragmente de cod folosind PL/SQL. Codul declarativ produce mai puține diferențe între dezvoltatori, iar aceasta face ca aplicațiile Application Express să fie ușor de întreținut și de gestionat.

Motorul Application Express redă aplicațiile în timp real din datele stocate în tabelele bazei de date. Când creați sau extindeți o aplicație, Oracle Application Express creează sau modifică metadate. Când aplicația rulează, motorul Application Express afișează pagina sau procesează trimiterea paginii pe baza metadatelor. Aceasta

înseamnă că mai mulți dezvoltatori pot edita aceeași pagină, iar modificările pot fi văzute imediat la rularea aplicației. Nu este necesară nicio compilare sau implementare.

Exemple de programare declarativă în Oracle Application Express:

Formulare

Folosind vrăjitori, puteți crea cu ușurință formulare pe tabele sau pe o procedură stocată. Când creați un formular, acești vrăjitori oferă gestionarea automată a inserării, actualizării și ștergerii, precum și detectarea actualizărilor pierdute. Odată ce creați un formular, puteți rearanja câmpurile formularului (numite elemente de formular) folosind o reprezentare vizuală, această facilitate permițându-vă să realizați rapid aspectul dorit.

Rapoarte interactive

Pentru a crea un raport interactiv, dezvoltatorii fie selectează un tabel, fie furnizează o instrucțiune SQL. Apoi pot restricționa capacitățile disponibile utilizatorilor finali (cum ar fi dezactivarea descărcării, nepermiterea ascunderii unei coloane), dacă se dorește. Funcționalitatea este aceeași în toate rapoartele din aplicație (de exemplu, cum se filtrează, se evidențiază, se abonează și se salvează vizualizări personalizate).

Starea sesiunii

Oracle Application Express gestionează în mod transparent starea sesiunii în baza de date. Nu există SET și GET, Oracle Application Express gestionează totul pentru dvs. Dezvoltatorii de aplicații pot accesa starea sesiunii folosind substituții simple, precum și sintaxa variabilelor legate SQL standard. Crearea unui proces pentru ștergerea stării sesiunii pentru o pagină se poate face prin câteva clicuri de mouse și specificarea paginii dorite.

Versiuni și instalare

Oracle Application Express poate fi instalat pe orice bază de date Oracle de la versiunea 9.2 sau o versiune ulterioară și, începând de la Oracle 11g, este instalat în mod implicit cu baza de date.

APEX 4.0 și versiunile ulterioare pot fi instalate pe o bază de date Oracle 10.2.0.3 sau mai mare.

APEX 5.0 și versiunile ulterioare pot fi instalate pe toate edițiile (SE1, SE și EE) ale bazei de date Oracle, 11.1.0.7 sau versiuni ulterioare, cu un acord de asistență tehnică Oracle Database valid.

De la APEX 18.1, versiunea minimă a bazei de date este 11.2.0.4. Poate fi utilizat și cu Oracle Database 11g Express Edition (XE), dar este acceptat prin forumul de discuții Oracle Technology Network și nu prin Oracle Support Services. Deși funcționalitatea APEX nu este limitată intenționat atunci când rulează pe XE, limitările motorului bazei de date pot împiedica funcționarea unor caracteristici APEX. De asemenea, Oracle XE are limite pentru utilizarea procesorului, a memoriei și a discului.

Avantajele utilizării APEX

- Dezvoltatorii de baze de date familiarizați cu PL / SQL își pot folosi setul de competențe pentru a crea și aplicații web.
- Folosind teme pre-construite se pot crea cu ușurință machete.
- Este ușor de implementat (utilizatorul final accesează o adresă URL pentru a accesa o aplicație APEX);
- Este scalabil (poate fi implementat pe laptopuri, stand - servere individuale sau instalații Oracle RAC).
- Procesarea și validările datelor se fac pe partea de server.
- Există o comunitate puternică și suport permanent.
- Se oferă suport de bază pentru dezvoltatorii de aplicații;
- Se oferă găzduirea gratuită pentru aplicațiile demo oferite de Oracle.
- Aplicațiile APEX pot rula gratuit pe Oracle Express Edition (XE).
- Componentele individuale ale unei aplicații pot fi redactate sau identificate folosind SQL, facilitând astfel crearea de rapoarte personalizate.

Crearea și dezvoltarea aplicațiilor utilizând ORACLE APPLICATION EXPRESS

- Aderă cu ușurință la modelul de dezvoltare/testare/producție SQA (fără a expune parolele DB).
- La crearea unei aplicații atenția se concentrează în mod deosebit pe modelul DB care stă la baza soluției (comparativ cu codarea în Java, .NET sau PHP - aveți nevoie doar de JavaScript).
- Suportă cu ușurință o temă standardizată pentru toate seturile de aplicații (și schimbarea temei respective).
- Chiar și persoanele cu puține cunoștințe tehnice își pot construi propriile pagini web și rapoarte.

Dezavantajele utilizării APEX

- Aplicațiile APEX sunt create folosind instrumentele proprii Oracle și pot fi găzduite doar într-o bază de date Oracle, generând astfel o dependență a dezvoltatorului și a beneficiarului de serviciile ORACLE.
- Foarte puțini furnizori de găzduire web oferă APEX (baza de date Oracle) în pachetul lor de servicii de găzduire (majoritatea oferă PHP + MySQL sau ASP + Microsoft SQL Server). Ca urmare, aplicațiile APEX sunt limitate la alegerea soluției de găzduire web. Cu toate acestea, deoarece APEX poate rula pe ediția Express gratuită a bazei de date Oracle (Oracle XE), este posibil să instalați pachetul software necesar (baza de date, cadrul APEX și gateway-ul web) pe orice sistem de operare standard gazdă (Linux sau Windows).
- Proiectele care necesită ca mai mulți dezvoltatori să acceseze aceeași pagină web își vor comunica intențiile între ele. Nu există un control al versiunilor încorporat și toate componentele trebuie editate prin intermediul interfeței web. Nu există suport pentru îmbinarea a două versiuni.

Blocarea paginilor poate ajuta însă la protejarea împotriva conflictelor.

Securitatea aplicațiilor APEX

Aplicațiile APEX au cam aceleași probleme de securitate ca și alte aplicații web bazate pe tehnologii mai directe, cum ar fi PHP, ASP.net și Java.

Principalele clase de vulnerabilități care afectează aplicațiile APEX sunt:

- injecția SQL,
- Cross-site scripting (XSS),
- Controlul accesului.

Aplicațiile APEX utilizează construcțiile PL/SQL ca limbaj de bază pentru server. Pe lângă accesarea datelor prin blocuri PL/ QL, o aplicație APEX va utiliza PL/SQL și pentru a implementa autorizarea accesului și pentru a afișa elementele paginii web în funcție de anumite condiții. Astfel, aplicațiile APEX pot suferi injecții SQL atunci când aceste blocuri PL/SQL nu validează corect și complet datele și nu gestionează intrările rău intenționate.

Oracle a implementat un tip de variabilă special pentru APEX numit Variabile de substituție (cu o sintaxă de tipul & NAME). Acest variabile nu sunt sigure și pot duce la SQL Injection.

În cazul în care injecția are loc într-un bloc PL/SQL, un atacator poate injecta un număr arbitrar de interogări sau instrucțiuni de executat.

Pentru a evita vulnerabilitățile de tip injecție XSS și SQL, se recomandă tratarea caracterelor speciale și utilizarea variabilelor legate în scrierea codului.

Vulnerabilitățile Cross-Site Scripting apar în aplicațiile APEX la fel ca în alte limbaje de creare a aplicațiilor web. Oracle oferă funcția `htf.escape_sc ()` pentru a scăpa de datele utilizatorului care sunt afișate printr-o interpretare de cod HTML.

Rapoartele generate de APEX oferă, de asemenea, protecție împotriva XSS prin setarea *Afișare ca*. Definițiile coloanelor pot fi interogate programatic pentru a verifica coloanele care nu scapă de valoare.

Pentru a controla accesul la resurse în cadrul unei aplicații APEX, un dezvoltator poate atribui scheme de autorizare resurselor (cum ar fi paginile și articolele). Acestea trebuie aplicate în mod consecvent pentru a se asigura că resursele sunt protejate în mod corespunzător. Un exemplu tipic de aplicare a unui control de acces inconsecvent este acela în care este setată o schemă de autorizare pentru un element Buton, dar nu și pe Procesul asociat care se efectuează când se face clic pe buton. Un utilizator rău intenționat

poate efectua procesul (prin JavaScript) fără a necesita accesarea butonului propriu-zis.

De la APEX 4.0, interfața Application Builder oferă posibilitatea unei evaluări a aplicației prin intermediul utilitarului Advisor.

Oracle Application Express Advisor funcționează ca un compilator sau LINT și semnalează comportamentul suspect sau erorile. Running Advisor verifică integritatea aplicației pe baza metadatelor de bază.

Arhitectura APX

Oracle APEX utilizează o arhitectură simplă pe 3 niveluri în care cererile sunt trimise de la browser, printr-un server web, la baza de date. Toate prelucrările, manipularea datelor și logica de afaceri sunt executate în baza de date. Această arhitectură garantează accesul rapid la date, performanțe superioare și scalabilitate.

O cerere web de la browserul web este trimisă la **Oracle REST Data Services (ORDS)** unde este predată **Oracle Database** pentru a fi executată. În baza de date, cererea este procesată de **Oracle APEX**. După finalizarea procesării, rezultatul este trimis înapoi prin **ORDS** către browser.

Oracle REST Data Services (ORDS), **Oracle APEX** și **Oracle Database** sunt împachetate împreună în Stiva **Oracle RAD**. Această stivă oferă toate componentele necesare pentru a dezvolta și implementa aplicații performante, scalabile și cu design atractiv.

Atât **Oracle APEX**, cât și **ORDS** sunt caracteristici gratuite ale bazei de date Oracle, ceea ce înseamnă că, dacă aveți baza de date Oracle, aveți deja această stivă Oracle RAD.

La crearea sau extinderea unei aplicație, Oracle Application Express creează sau modifică metadatele stocate în tabelele sale din baze de date. Când aplicația este rulată, motorul Oracle Application Express citește apoi metadatele și afișează pagina solicitată sau procesează

trimiterile de pagini. Pentru a oferi un comportament de stare în cadrul unei aplicații, Oracle Application Express gestionează în mod transparent starea sesiunii în baza de date. Dezvoltatorii de aplicații pot obține și seta starea sesiunii folosind substituții simple, precum și sintaxa variabilelor legate SQL standard.

Oracle Application Express poate fi implementat pentru a permite utilizatorilor să solicite un spațiu de lucru. Un spațiu de lucru oferă acces la un mediu de dezvoltare a aplicațiilor Oracle Application Express, precum și acces la una sau mai multe scheme de baze de date.

Modelul de autoservire permite consolidarea proiectelor de dezvoltare a aplicațiilor din mai multe grupuri de lucru într-o singură bază de date Oracle.

Oracle Application Express oferă trei moduri de lucru:

- Manual - Un administrator creează manual fiecare spațiu de lucru
- Solicitare - Utilizatorii urmează un link afișat pe pagina de autentificare pentru a solicita spații de lucru pe care administratorii le revizuiesc și aprobă sau refuză

Crearea și dezvoltarea aplicațiilor utilizând ORACLE APPLICATION EXPRESS

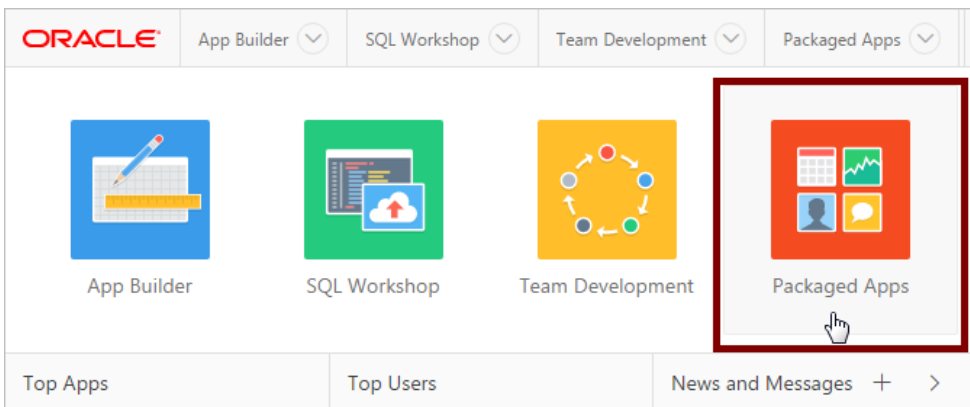
- Verificarea prin e-mail - Utilizatorii trimit în continuare cereri, dar spațiile de lucru se creează numai după verificarea prin e-mail de către utilizator.

Componentele Application Express, Versiunea 5.1¹

Dacă accesați Oracle Application Express într-un mediu găzduit, solicitarea unui cont creează un spațiu de lucru și un cont. Dacă gestionați local Oracle Application Express, veți crea manual un spațiu de lucru și un cont. În ambele situații, contul dvs. Oracle Application Express trebuie să aibă privilegiile de dezvoltator.

Packaged Apps – *Sample Database Application*

Acest pachet conține exemple de aplicații cu baze de date.



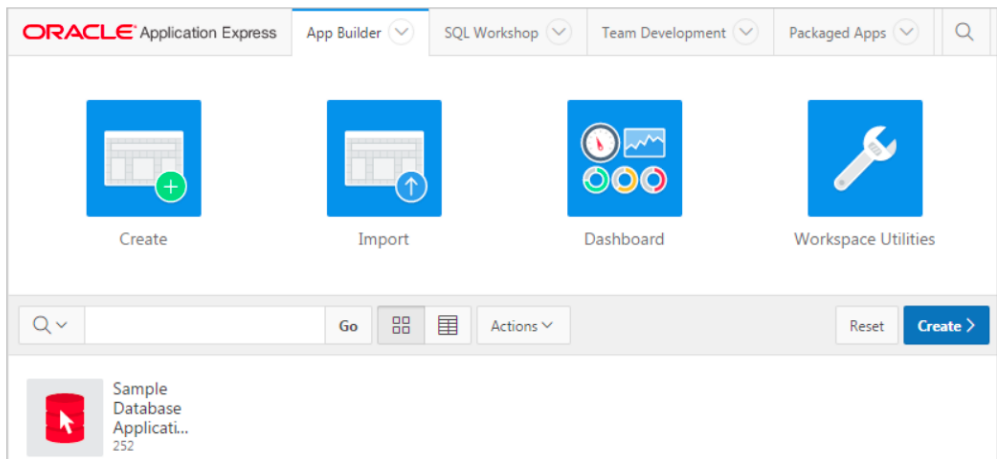
Pentru a verifica dacă pachetul *Sample Database Application* este instalat în spațiul dvs. de lucru, puteți parcurge următoarele etape:

¹ <https://docs.oracle.com/database/apex-5.1/AETUT/toc.htm>

Crearea și dezvoltarea aplicațiilor utilizând ORACLE APPLICATION EXPRESS

- Vă conectați la spațiul de lucru Oracle Application Express.
- Click pe **Packaged Apps**.
- Click pe **Sample Database Application**.
- Dacă pe ecran apare mesajul **Install Packaged App**, se pornește procesul de instalare. Altfel, pachetul este deja instalat.

App Builder



App Builder se poate folosi pentru a crea aplicații noi, pentru a selecta o aplicație pentru editare, pentru a importa aplicații

exportate anterior, pentru a vizualiza și accesa utilitățile spațiului de lucru.

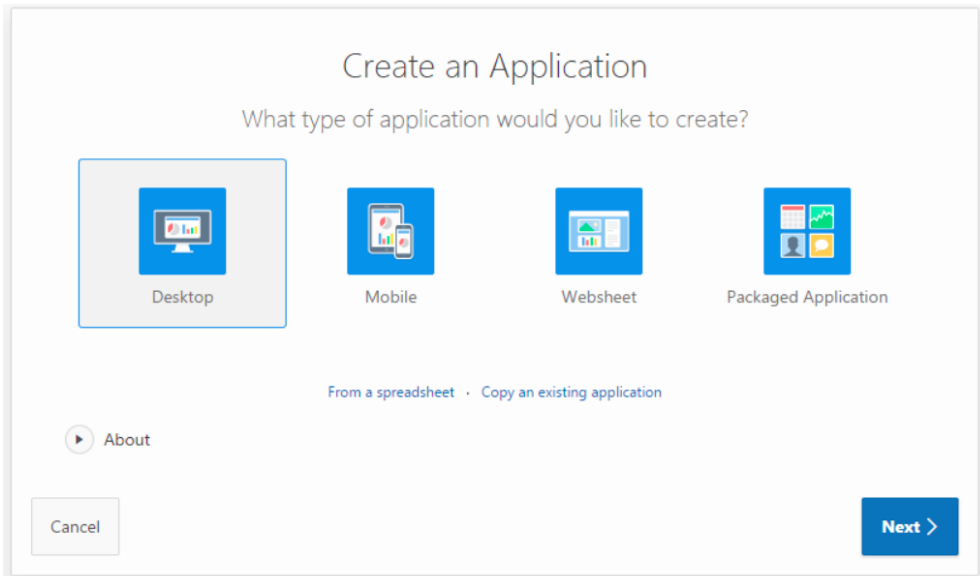
Aplicațiile sunt organizate în pagini. Conținutul paginii este organizat în regiuni. Regiunile pot conține text, PL/SQL personalizat, rapoarte, diagrame, hărți, calendare, referințe de servicii web sau formulare.

Formularele sunt alcătuite din câmpuri (numite articole) care pot fi selectate din multitudinea de tipuri încorporate (cum ar fi câmpurile de text, zonele de text, butoanele radio, listele de selecție, casetele de selecție, selectorul de date și lista pop-up de valori) sau un dezvoltator își poate crea propriile tipuri folosind suportul pentru plug-in-uri. Funcționalitatea de actualizare a tabelelor este încorporată și PL/SQL poate fi utilizat pentru procesarea datelor.

Starea sesiunii (sau contextul aplicației) este gestionată în mod transparent, iar prezentarea interfeței cu utilizatorul este separată de logica aplicației, astfel încât aspectul unei aplicații să poată fi schimbat pur și simplu prin selectarea unei teme diferite.

App Builder - Worksheet Applications

Când rulați **Create Application Wizard**, acesta vă solicită să alegeți tipul de aplicație pe care doriți să o creați.



Aplicațiile de tip **Websheet Applications** permit utilizatorilor să construiască aplicații centrate pe date chiar și fără cunoștințe de programare SQL. Sunt pagini web interactive care combină textul cu datele. Aceste aplicații sunt extrem de dinamice și sunt definite de utilizatorii lor. Includ controale de navigare, funcții de căutare și posibilitatea de a adăuga adnotări, cum ar fi fișiere, note și etichete. Pot fi securizate utilizând reguli de control al accesului și mai multe modele de autentificare încorporate.

Utilizatorii finali pot gestiona date structurate și nestructurate fără asistența dezvoltatorului. Secțiunile de pagină conțin date nestructurate și sunt editate folosind un editor WYSIWYG.

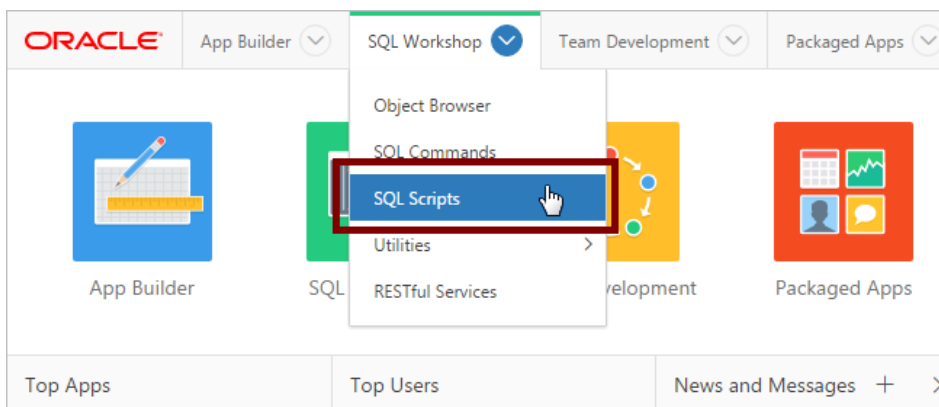
Rapoartele oferă acces la datele bazei de date scriind SQL.

Grilele de date pot gestiona date structurate fără a scrie SQL. Adăugarea coloanelor, redenumirea coloanei și validările sunt definite utilizând dialoguri de execuție. Fiecare pagină și rând de date ale grilei de date pot fi adnotate cu fișiere, etichete, note și linkuri.

Paginile pot conține secțiuni, precum și rapoarte și grile de date și toate pot fi legate împreună, oferind navigare.

Toate informațiile pot fi căutate și controlate complet de către utilizatorul final.

SQL Workshop



SQL Workshop oferă instrumente care vă permit să vizualizați și să gestionați obiectele bazei de date.

Object Browser vă permite să utilizați o structură arborescentă pentru a vizualiza proprietățile obiectului și a crea obiecte noi.

Instrumentul **SQL Command** vă permite să introduceți cod.

SQL Scripts vă permit să stocați și să rulați scripturi.

Team Development

Team Development ajută la gestionarea ciclului de viață al unei aplicații. Acesta permite urmărirea și gestionarea caracteristicilor aplicației, pentru a efectua intrări, a identifica erori și a primi feedback-ul utilizatorului final.

2. CREAREA UNEI APLICAȚII CU BAZE DE DATE ÎN ORACLE APEX

O aplicație construită cu utilitarul *ORACLE Application Express* este o colecție de pagini legate între ele, navigarea printre ele făcându-se prin intermediul unor butoane, tab-uri sau link-uri hypertext. Pentru crearea unei aplicații trebuie să parcurgem următorii pași:

Denumirea aplicației

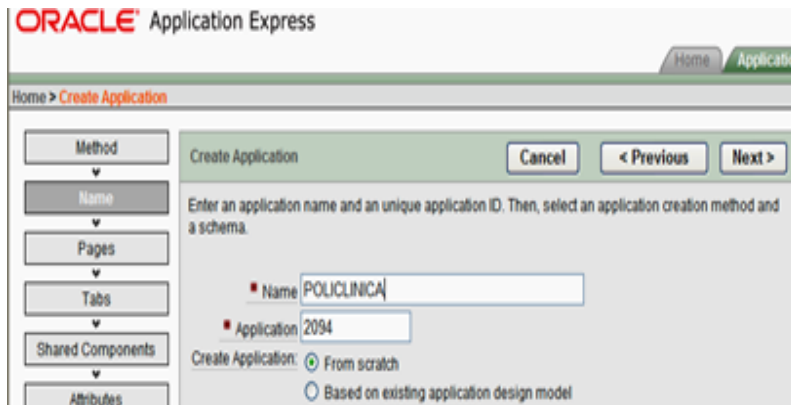


Figura 2-1 Crearea unei aplicații

Adăugarea unei pagini în aplicație

Paginile au la bază tabele sau interogări, având ca formă finală afișarea unor rapoarte, rezultatul interogărilor sau actualizări ale tabelelor. Pentru adăugarea unei pagini trebuie să selectăm tipul acesteia:

- fără funcționalitate (**Blank**)

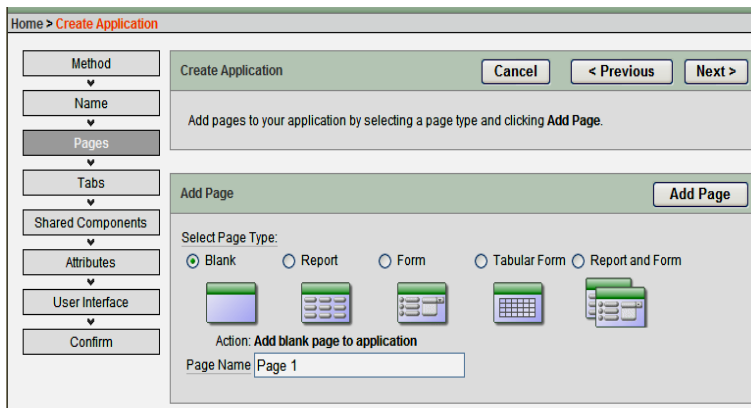


Figura 2-2 Adăugarea unei pagini la aplicație

- **Form** reprezintă actualizarea unui singure linii dintr-o tabelă selectată.

- **Report** reprezintă rezultatul unei interogări SQL. Această interogare poate fi specificată explicit (scrisă direct într-o casetă specifică) sau construită cu ajutorul opțiunii *Query Builder*

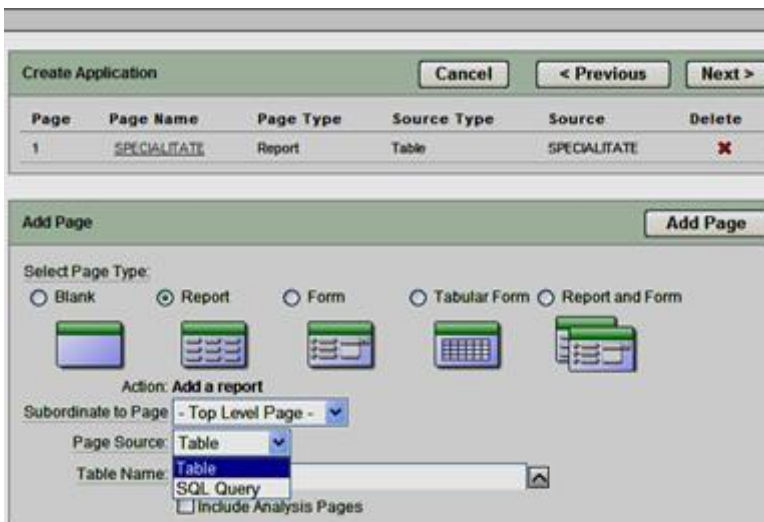


Figura 2-3 Crearea unui raport pe baza unei table

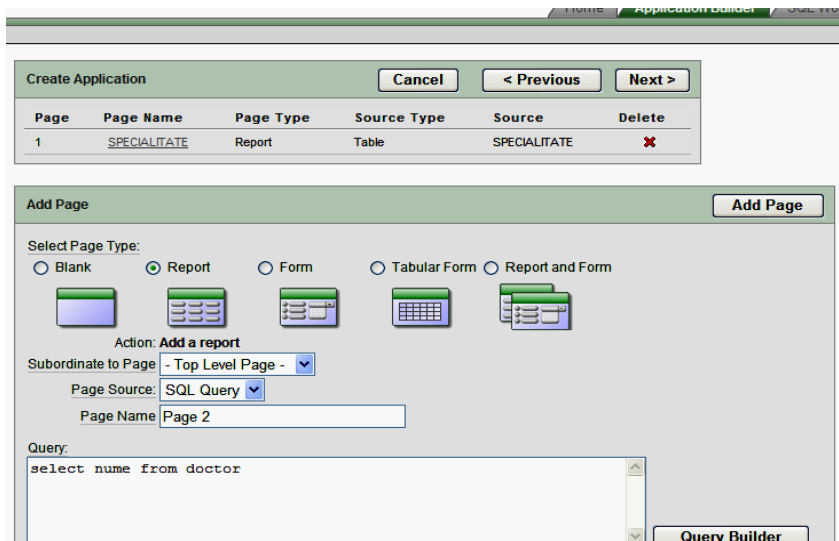


Figura 2-4 Crearea unui raport pe baza unei interogări

- **Tabular form** permite actualizarea, inserarea și ștergerea datelor dintr-o tabelă.
- **Report and Form** este o combinație între form și report, având ca efect crearea a două pagini: una pentru form (permite actualizarea unui singur rând), iar cealaltă pentru raport.

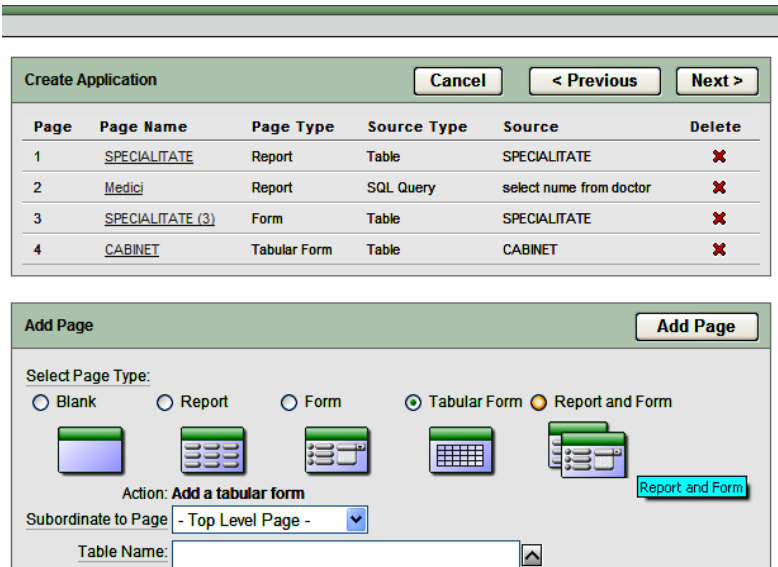


Figura 2-5 Adăugarea mai multor pagini într-o aplicație

Stabilirea nivelelor pentru tab-uri

- fără tab-uri
- tab-uri pe un singur nivel
- tab-uri pe două nivele

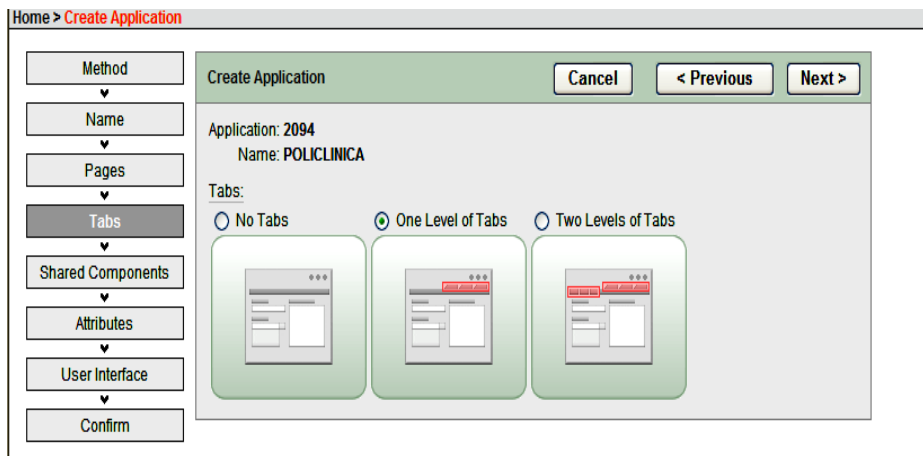


Figura 2-6 Stabilirea nivelelor pentru tab-uri

The screenshot shows the 'Create Application' wizard in Oracle APEX, specifically the 'Shared Components' step. The left sidebar contains a vertical list of steps: Method, Name, Pages, Tabs, Shared Components (highlighted), Attributes, User Interface, and Confirm. The main content area is titled 'Create Application' and includes 'Cancel', '< Previous', and 'Next >' buttons. Below the title, there is explanatory text: 'Shared components are common application elements that can be displayed or applied across multiple pages in an application. To save time or maintain consistency between applications, it is possible to copy the shared components from an existing application.' Underneath, the option 'Copy Shared Components from Another Application:' is shown with two radio buttons: 'Yes' (unselected) and 'No' (selected).

Figura 2-7 Shared Components

Copierea elementelor comune. Stabilirea atributelor

Elementele comune din cadrul paginilor unei aplicații existente, se pot copia pentru noua aplicație.

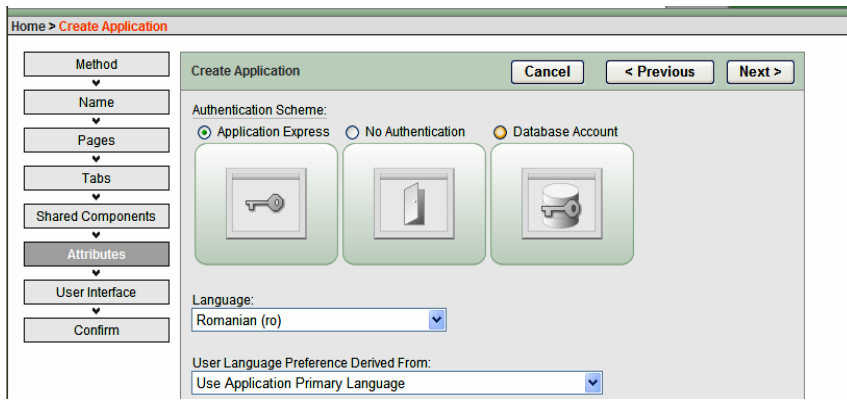


Figura 2-8

Stabilirea atributelor aplicației

Alegând *Application Express*, în continuare, după crearea aplicației, pentru a vizualiza, modifica sau executa această aplicație, se vor cere numele user-ului ce a creat aplicația respectivă, precum și parola acestuia.

Alegerea unei interfețe grafice pentru aplicație (theme)

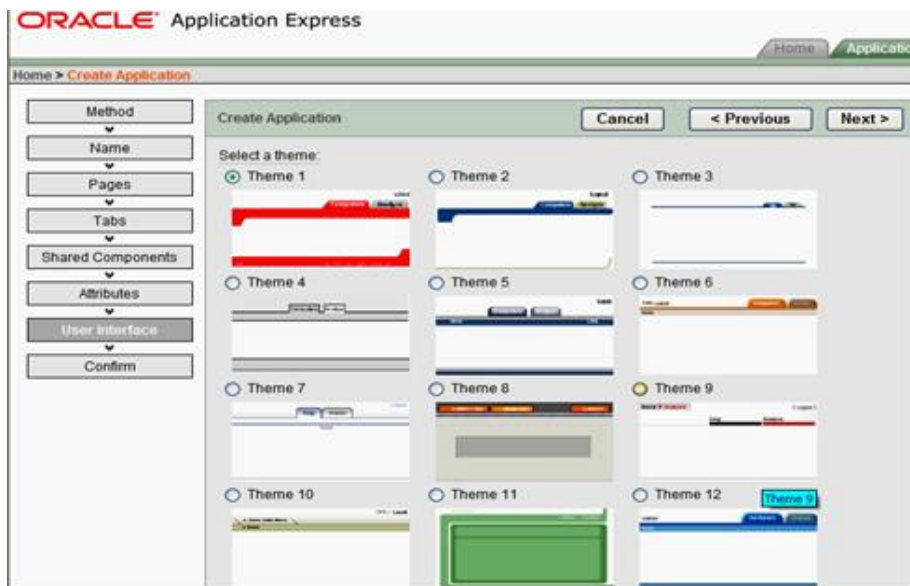


Figura 2-9

Selectarea unei interfețe pentru aplicație

Confirmarea creării aplicației

Home > Create Application

Method
Name
Pages
Tabs
Shared Components
Attributes
User Interface
Confirm

Create Application

You have requested to create an application with the following attributes. Please confirm your selections.

Application	2094
Name	POLICLINICA
Parsing Schema	RO_INST31_SQL03_S13
Default Language	ro
Tabs	One Level of Tabs
Default Authentication Scheme	Application Express Authentication

UI Theme

Save this definition as a design model for reuse

Figura 2-10

Confirmarea creării aplicației

Lansarea în execuție a aplicației create

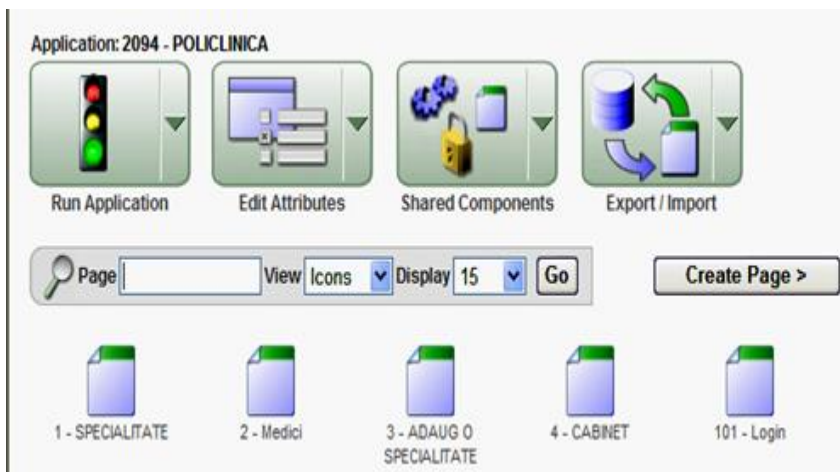


Figura 2-11

Lansarea în execuție a aplicației create

Execuția aplicației

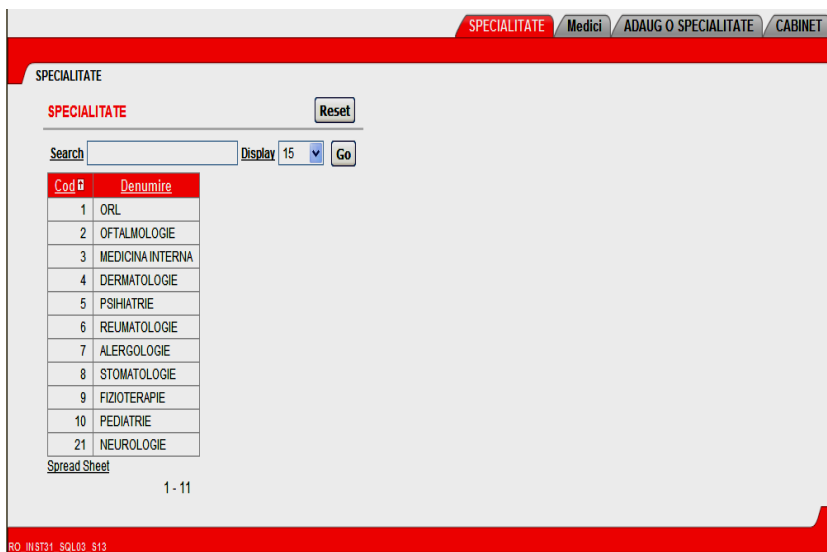


Figura 2-12 Execuția aplicației – Pagina 1

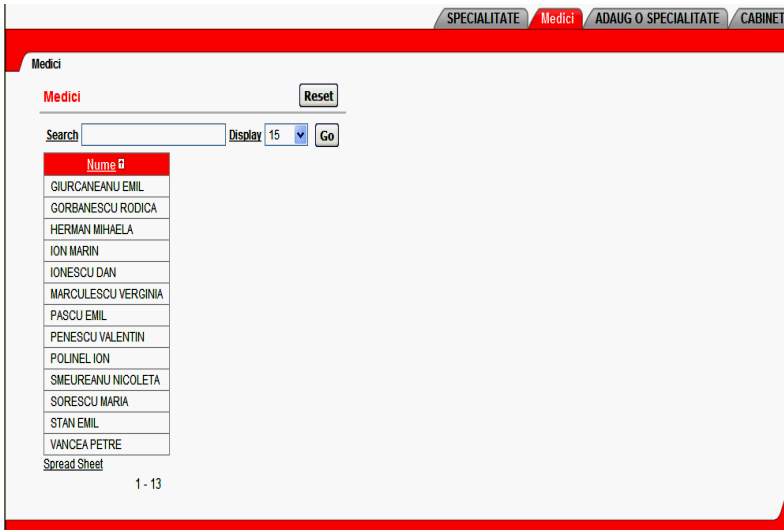


Figura 2-13 Execuția aplicației – Pagina 2



Figura 2-14 Execuția aplicației – Pagina 3

SPECIALITATE Medici ADAUG O SPECIALITATE CABINET

CABINET

Cancel Delete Submit

<input type="checkbox"/>	CODC_DISPLAY	Cod
<input type="checkbox"/>	1	2
<input type="checkbox"/>	3	1
<input type="checkbox"/>	6	4
<input type="checkbox"/>	7	6
<input type="checkbox"/>	10	5
<input type="checkbox"/>	12	9
<input type="checkbox"/>	13	8
<input type="checkbox"/>	2	2
<input type="checkbox"/>	4	1
<input type="checkbox"/>	8	3

row(s) 1 - 10 of 11 Next

Add Row

DUSTRII_S01.03_813

Figura 2-15 Execuția aplicației – Pagina 4

3. CREAREA UNEI APLICAȚII WEB MOBILE ÎN ORACLE APEX

Versiunea 4.2 a utilitarului Oracle Application Express introduce posibilitatea de a defini declarativ aplicații web mobile folosind jQuery Mobile. *Application Builder* a fost îmbunătățit pentru a sprijini interfețe multiple, atât desktop cât și mobile, într-o singură aplicație. Prin urmare, dezvoltatorii pot alege să construiască o aplicație fie numai pentru desktop, fie doar pentru mobil, sau pentru ambele. Paginile aplicației web mobile folosesc *jQuery Mobile* pentru a reda conținutul pe marea majoritate a dispozitivelor mobile.

În continuare, vom crea o aplicație web mobile și o vom executa pentru a vizualiza interfața cu utilizatorul de telefonie mobilă. Aplicația va ține evidența elevilor dintr-o școală, folosind pentru aceasta două tabele, *Clasa* și *Elevii*. În figura 1 creăm aplicația *Scoala* cu interfață *jQuery Mobile Smartphone*. Vom introduce în aplicație două pagini de tip *List View and Form*, *Clasa* și *Elevii* ce vor conține fiecare un formular de introducere a datelor, precum și o lista a valorilor ce poate fi modificată (Fig.2.2).

Crearea și dezvoltarea aplicațiilor utilizând ORACLE APPLICATION EXPRESS

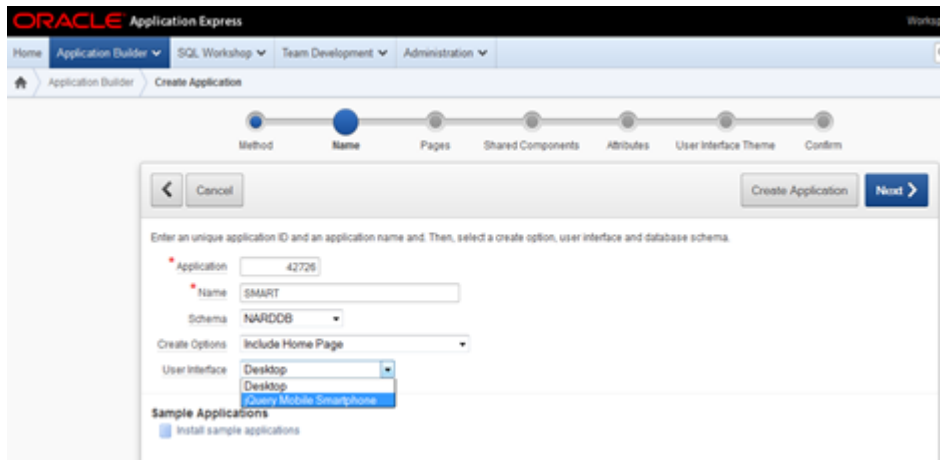


Figura 3-1

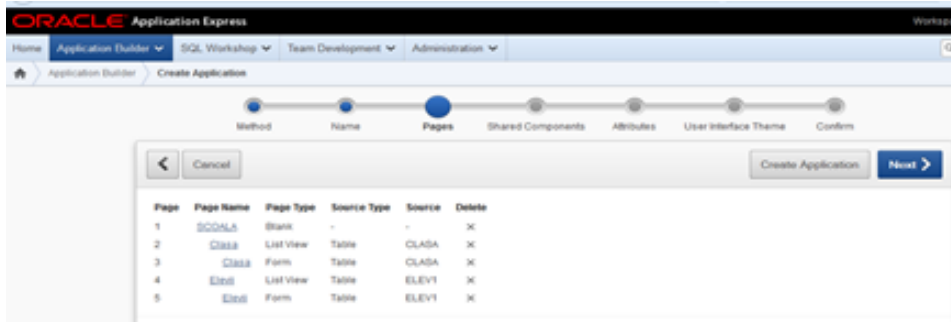


Figura 3-2

Execuția aplicației pe desktop



Figura 3-3 Execuția aplicației

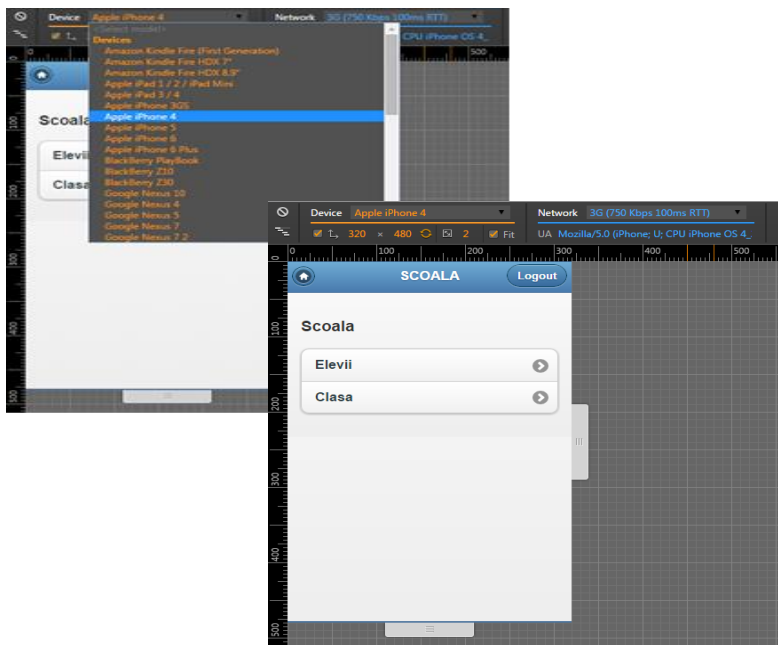


Figura 3-4 Alegerea dispozitivului mobil



Figura 3-5

Lista cu formular pentru introducerea datelor în tabela Elevi

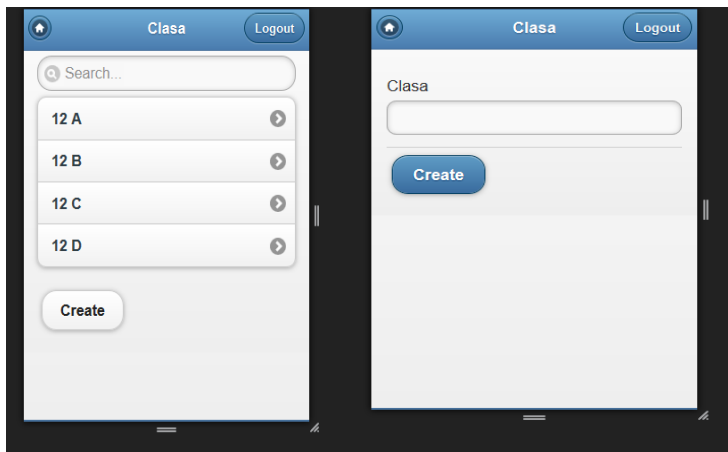


Figura 3-6

Lista cu formular pentru introducerea datelor în tabela Clasa

4. GRAFICE ȘI RAPOARTE BAZATE PE INTEROGĂRI ÎN APLICAȚII ORACLE APEX

Ne propunem în aplicația descrisă mai jos, să construim un raport bazat pe o interogare a două tabele cu legături între ele, precum și un grafic al acestuia.

Se consideră o bază de date în care se reține evidența medicilor dintr-o unitate medicală. Pentru fiecare medic se cunosc următoarele informații: numele și prenumele, telefonul, precum și specialitatea acestuia. Ne propunem să realizăm o aplicație în ORACLE APEX prin care să putem gestiona și vizualiza datele respective.

Se vor crea tabelele `MEDIC` și `SPECIALITATE` în care vom reține informațiile de care avem nevoie.

MEDIC							
Table	Data	Indexes	Model Constraints	Grants	Statistics	UI Defaults	Triggers
Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	
Column Name	Data Type	Nullable	Default	Primary Key			
COD	NUMBER	No	-	1			
NUME	VARCHAR2(4000)	No	-	-			
TELEFON	NUMBER	Yes	-	-			
CODSPEC	NUMBER	No	-	-			
				1 - 4			

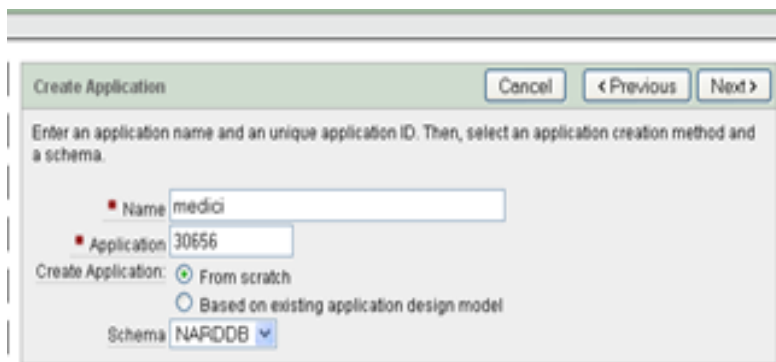
Figura 4.1 Tabela Medic

SPECIALITATE							
Table	Data	Indexes	Model Constraints	Grants	Statistics	UI Defaults	Triggers
Add Column	Modify Column	Rename Column	Drop Column	Rename	Copy	Drop	T
Column Name	Data Type	Nullable	Default	Primary Key			
COD	NUMBER	No	-	1			
DENUMIRE	VARCHAR2(4000)	No	-	-			
				1 - 2			
Download		Print					

Figura 4.2 Tabela Specialitate

În continuare vom construi aplicația MEDICI ce va include trei pagini:

- O pagină de tip **Blank** ce va conține intrările către celelalte două pagini
- O pagină de tip **Report** prin intermediul căreia vom vizualiza informațiile din tabela **Specialitate**
- O pagină de tip **Report** ce va conține un raport bazat pe o interogare, prin intermediul căruia vom vizualiza informațiile din tabellele **Specialitate și Medic**.



Create Application

Cancel < Previous Next >

Enter an application name and an unique application ID. Then, select an application creation method and a schema.

Name medici

Application 30656

Create Application: From scratch Based on existing application design model

Schema NAPDDB

Figura 4.3 Crearea aplicației MEDICI

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

Page	Page Name	Page Type	Source Type	Source	Delete
1	medici	Blank	-	-	X

Add Page [Add Page]

Select Page Type:

Blank Report Form

Tabular Form Master Detail Report and Form

Action: **Add a report**

Subordinate to Page: - Top Level Page -

Page Source: SQL Query

Page Name: Page 2

Implementation: Interactive

Query:

Query Builder

Figura 4.4

Crearea unui raport pe baza unei interogări

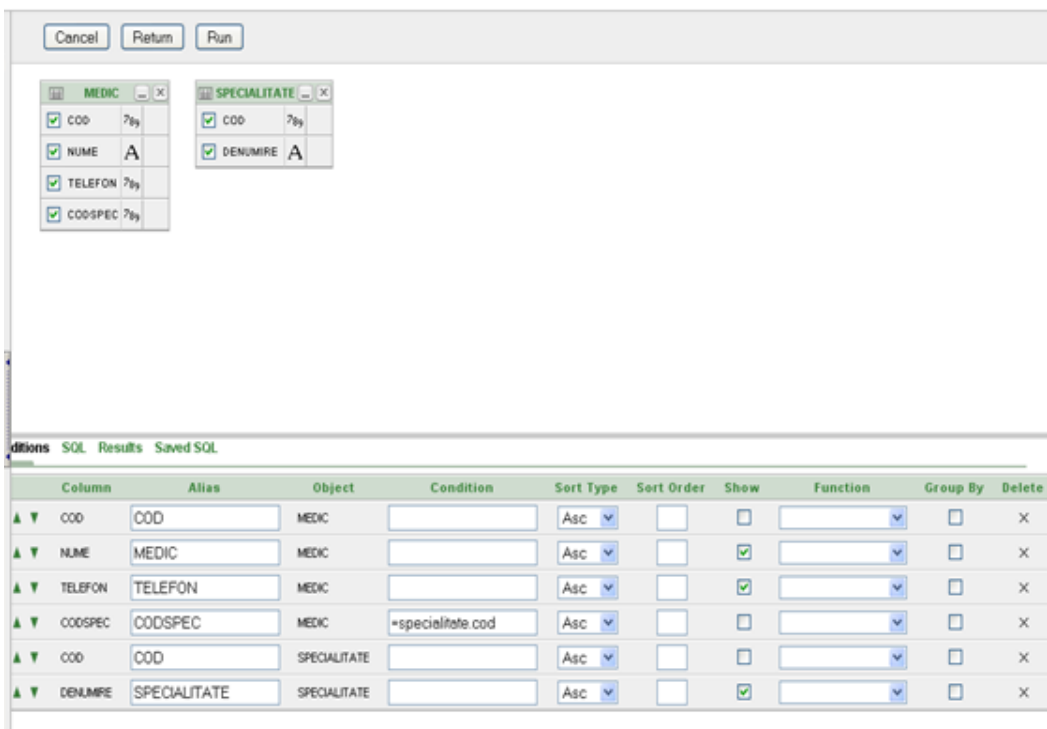


Figura 4.5

Specificarea legăturilor dintre cele două tabele, în cadrul raportului

Cancel Return Run

MEDIC	
<input checked="" type="checkbox"/> COD	789
<input checked="" type="checkbox"/> NUME	A
<input checked="" type="checkbox"/> TELEFON	789
<input checked="" type="checkbox"/> CODSPEC	789

SPECIALITATE	
<input checked="" type="checkbox"/> COD	789
<input checked="" type="checkbox"/> DENUMIRE	A

Conditions SQL **Results** Saved SQL

MEDIC	TELEFON	SPECIALITATE
Popescu Ilie	748998689	medicina generala
Herman Mihaela	742335689	medicina generala
Patrascu Viorel	724568912	dermatologie
Panescu Ion	748998698	orl
Ilie Marin	76898709	orl
Gorbanescu Ana	748894689	dermatologie

row(s) 1 - 6 of 6

[Download](#)

Figura 4.6

Execuția raportului Medici

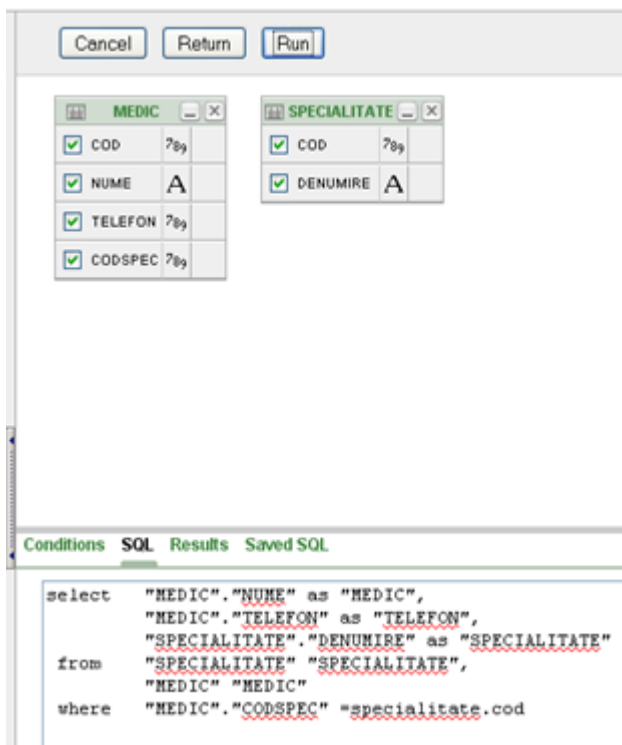


Figura 4.7 Comanda SQL pentru crearea raportului Medici

Create Application Cancel < Previous Next > Create

Page	Page Name	Page Type	Source Type	Source	Delete
1	medici	Blank	-	-	X

Add Page Add Page

Select Page Type:

Blank
 Report
 Form

Tabular Form
 Master Detail
 Report and Form

Action: **Add a report**

Subordinate to Page: medici (1)

Page Source: SQL Query

Page Name: MEDICI

Implementation: Interactive

Query:

```

select  "MEDIC"."NUME" as "MEDIC",
        "MEDIC"."TELEFON" as "TELEFON",
        "SPECIALITATE"."DENUMIRE" as "SPECIALITATE"
from    "SPECIALITATE" "SPECIALITATE",
        "MEDIC" "MEDIC"
where   "MEDIC"."CODSPEC" =specialitate.cod
  
```

Query Builder

Figura 4.8 Crearea raportului Medici în cadrul aplicației

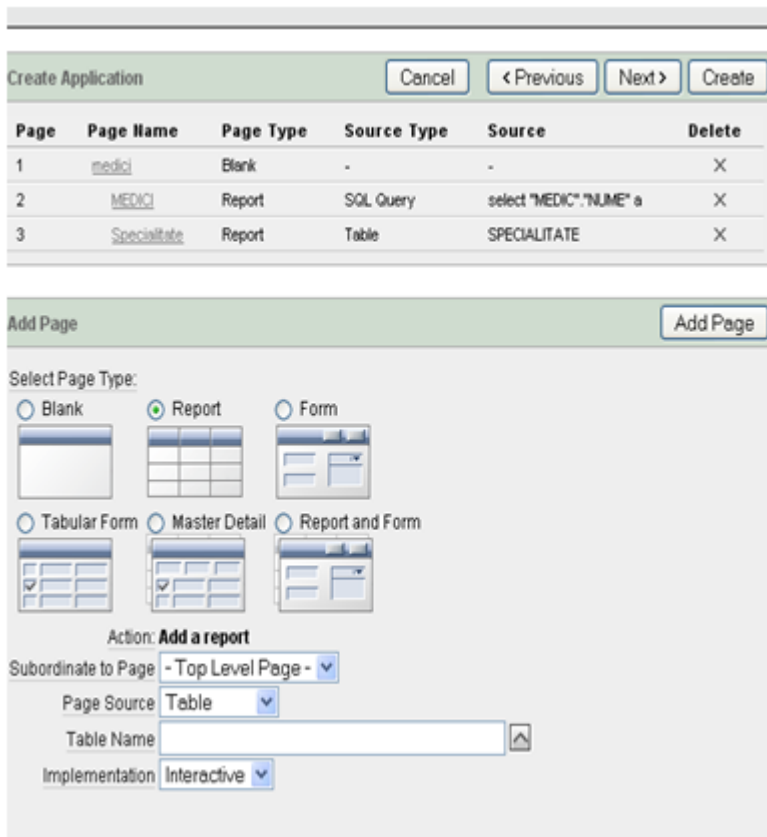


Figura 4.9 Adăugarea raportului Medici în cadrul aplicației

În figura 9 se observă, în cadrul raportului apariția denumirii specialității fiecărui medic, în locul codului specialității respective.

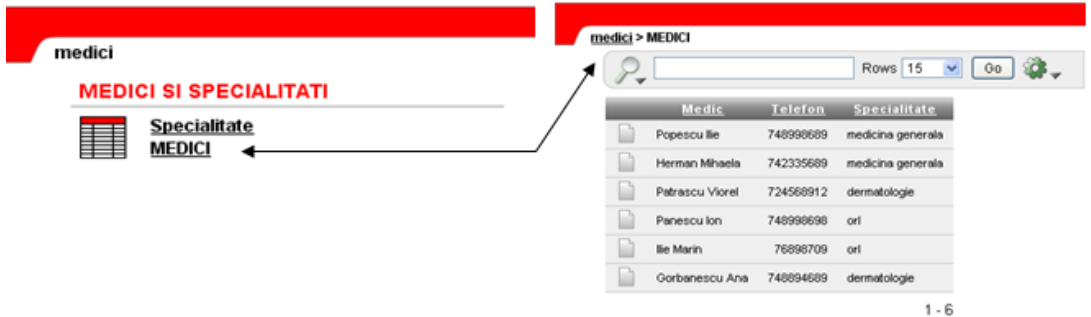


Figura 4.10 Execuția aplicației Medici

Ne propunem în continuare să atașăm un grafic la raportul bazat pe interogarea creată anterior. În graficul respectiv dorim să afișăm numărul de medici din fiecare specialitate prezentă în tabela *Specialități* (Fig.4.10).

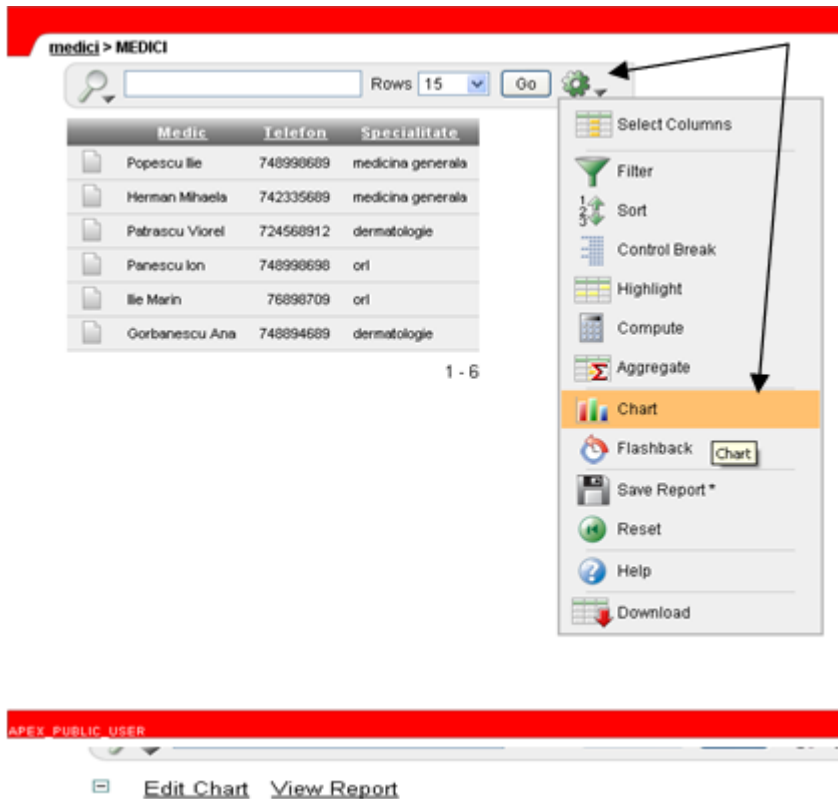


Figura 4.11 Crearea unui grafic



The screenshot displays the Oracle APEX interface for a report named 'medici'. At the top, there is a search bar and a 'Rows' dropdown set to 15. Below this is a 'Chart' configuration dialog box. The dialog box has the following settings:

- Chart Type: Bar chart (selected)
- Label: Specialitate
- Value: Telefon
- Function: Count
- Sort: Value - Ascending

Below the dialog box is a table with the following data:

Medic	Telefon	Specialitate
Popescu Ilie	748998889	medicina generala
Herman Mihaela	742335689	medicina generala
Patrascu Viorel	724568912	dermatologie
Panescu Ion	748998698	ori
Ilie Marin	76898709	ori
Gorbanescu Ana	748894689	dermatologie

Figura 4.12 Atașarea unui grafic la raportul Medici

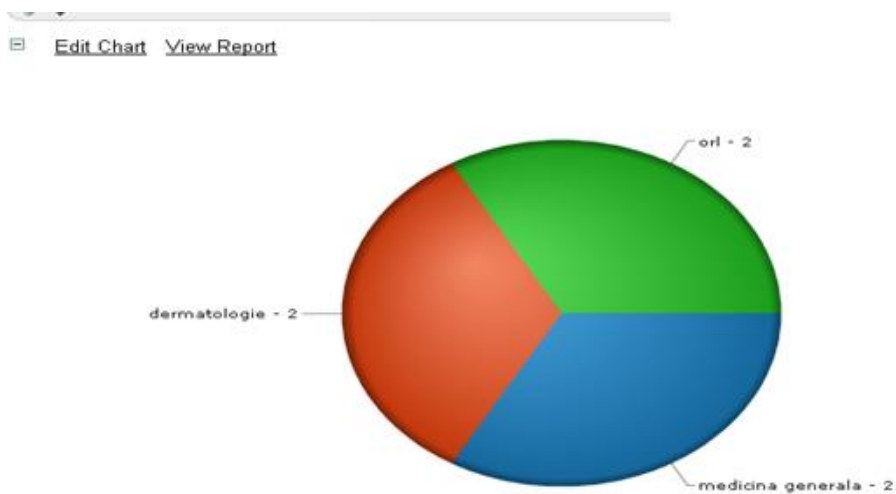


Figura 4.13 Vizualizarea graficului

5. SHUTTLE ITEM IN ORACLE APEX 4.2

De multe ori, în aplicațiile cu baze de date este necesară folosirea filtrelor în interiorul rapoartelor acestora. Utilitarul *ORACLE APEX 4.2* pune la dispoziția utilizatorului atât posibilitatea folosirii filtrelor standard cât și crearea și utilizarea propriilor filtre în aplicații.

În cele ce urmează, ne propunem să arătăm un exemplu de folosire a filtrelor definite de utilizator în rapoartele unei baze de date, cărora le vom atașa o acțiune dinamică de tip *Refresh*, prin intermediul unei aplicații realizată în *ORACLE APEX 4.2*.

Pentru exemplificare, vom crea o aplicație pentru gestiunea elevilor unei școli, în care vom reține informații despre numele elevilor și clasa de unde provine fiecare. Aplicația va folosi două tabele: *Scoala* care va reține denumirea claselor și *Scoala_E* care va reține elevii pe clase.

SCOALA					SCOALA_E				
Column Name	Data Type	Nullable	Default	Primary Key	Column Name	Data Type	Nullable	Default	Primary Key
COD	NUMBER	No	-	1	ID	NUMBER	No	-	1
CLASA	VARCHAR2(5)	Yes	-	-	NUME	VARCHAR2(40)	No	-	-
					CLASA	NUMBER	No	-	-

Figura 5.1Tabela Scoala

Figura 5.2Tabela Scoala_E

Vom crea aplicația după structura prezentă în Fig 5.3, în care *Clasele* reprezintă intrarea către un tabular form pentru gestionarea claselor școlii(Fig.5.4), iar *Elevii* reprezintă intrarea către un tabular form pentru gestionarea elevilor școlii(Fig.5.5).

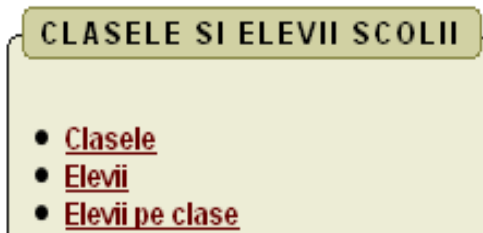


Figura 5.3 Aplicația

The screenshot shows the "Clasele" tabular form. It has a title bar "Clasele" and buttons for "Cancel", "Delete", and "Submit". The table has columns for "Clasa" and "Cod".

<input type="checkbox"/>	Clasa	Cod
<input type="checkbox"/>	12 A	1
<input type="checkbox"/>	12 B	2
<input type="checkbox"/>	12 C	3
<input type="checkbox"/>	12 D	4

1 - 4
Add Row

Figura 5.4 Clasele

The screenshot shows the "Elevii" tabular form. It has a title bar "Elevii" and buttons for "Cancel", "Delete", and "Submit". The table has columns for "Id", "Nume", and "Clasa".

<input type="checkbox"/>	Id	Nume	Clasa
<input type="checkbox"/>	1	Ion Maria	12 A
<input type="checkbox"/>	3	Bica Camelia	12 A
<input type="checkbox"/>	5	Boncea Ovidiu	12 A
<input type="checkbox"/>	2	Ana Constantin	12 B
<input type="checkbox"/>	4	Dita Aurel	12 B
<input type="checkbox"/>	6	Cristescu Amalia	12 C
<input type="checkbox"/>	8	Dumitrescu Diana	12 C
<input type="checkbox"/>	7	Mares Gina	12 D
<input type="checkbox"/>	9	Oancea Dumitru	12 D

1 - 9
Add Row

Figura 5.5 Elevii pe clase

Intrarea *Elevii pe clase* din aplicație ne conduce către un raport interactivă căruia îi vom atașa o filtrare a elevilor pe clase. Pentru aceasta, vom contrui un item de tip shuttle.

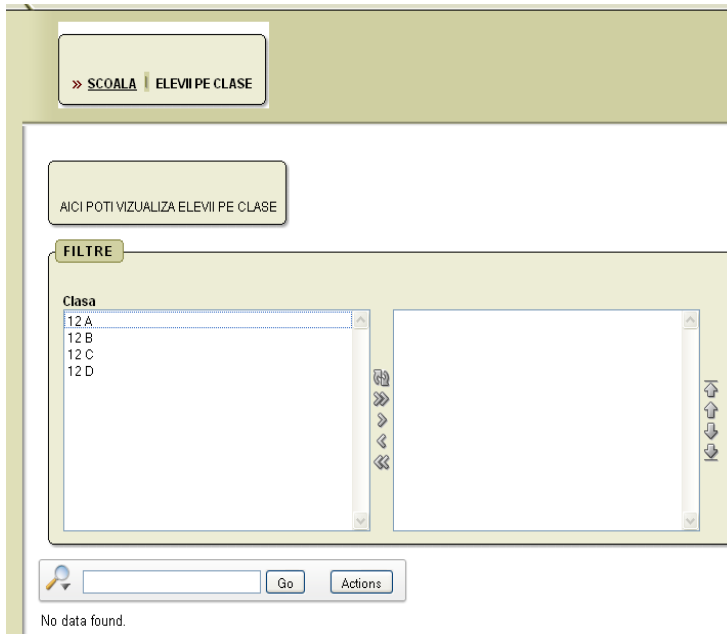


Figura 5.6 Raport interactiv cu filtre(1)

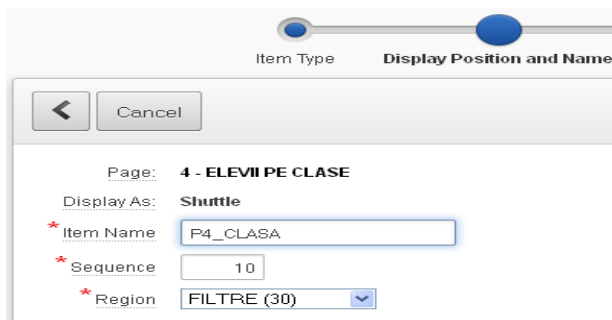


Figura 5.7 Item de tip Shuttle

Acestui item îi vom atașa o acțiune dinamică de tip *Refresh* cu numele *Filter Refresh*

Page: 4 - ELEVII PE CLASE

Name: Filter Refresh

Sequence: 10

Page: 4 - ELEVII PE CLASE

Name: Filter Refresh

Action: Refresh

[Show](#), [Hide](#), [Enable](#), [Disable](#), [Set Value](#)

Fire On Page Load

Identify when you would like the Dynamic Action to fire.

Page: 4 - ELEVII PE CLASE

Name: Filter Refresh

Event: Change

[Change](#), [Click](#), [Page Load](#)

Selection Type: Item(s)

[Item](#), [Button](#), [Region](#)

Item(s): P4_CLASA

Condition: - No Condition -

[equal to](#), [is null](#), [in list](#), [JavaScript Expression](#)

Figura 5.8 Crearea acțiunii dinamice de tip Refresh

*** Enter a SQL SELECT statement**

```
select SCOALA_E.ID as ID,
       SCOALA_E.NUME as NUME,
       SCOALA.CLASA as CLASA
from SCOALA_E SCOALA_E,
     SCOALA SCOALA
where SCOALA_E.CLASA =scoala.cod
```

Page Items to Submit: P4_CLASA

Link to Single Row View: No

Uniquely Identify Rows by: ROWID

Unique Column:

Figura 5.9 Crearea raportului interactiv

Filtre

Clasa

12 B
12 C
12 D

12 A

Go Actions

Id	Nume	Clasa
1	Ion Maria	12 A
3	Bica Camelia	12 A
5	Boncea Ovidiu	12 A

1 - 3

Figura 5.10 Raport interactiv cu filtre(2)

La crearea raportului interactiv asociat intrării *Elevii pe clase* din aplicație, vom face legătura dintre acesta și itemul de tip shuttle creat anterior. În raportul interactiv, la selectarea unei clase din panoul cu filtre *Clasa*, se afișează elevii corespunzători acelei clase.

BIBLIOGRAFIE

1. <http://www.oracle.com> - Creating a Mobile Web Application Using Oracle Application Express 4.2, 2014
2. Riaz Ahmed- Create Rapid Web Applications Using Oracle Application Express - Second Edition, 2013
3. **Roel Hartman, Christian Rokitta, David Peake** - *Oracle Application Express for Mobile Web Applications, 2013*
4. ORACLE ACADEMY- 2011, Database Programming with PL\SQL , <https://academy.ORACLE.com/>
5. ORACLE ACADEMY - 2011 Database Programming with SQL, <https://academy.ORACLE.com/>
6. ORACLE® APPLICATION EXPRESS 4.1, 2011, <http://apex.oracle.com/>
7. **ORACLE ACADEMY**- *Database design*, <https://academy.ORACLE.com/>, 2007
8. **ORACLE ACADEMY** - *Database Programming with SQL*, <https://academy.ORACLE.com/>, 2007
9. **ORACLE® Database** SQL Reference 10g Release 1 (10.1) , 2003

<http://bd.ase.ro/Suport%20de%20curs/Instalari/SQL%20Reference.pdf>

10. **PL/SQL** User's Guide and Reference 10g Release 1 (10.1),
decembrie 2003
11. Luers, Tom- 2001, ORACLE PL/SQL, Editura TEORA
12. Hernandez, Michael J- 2000 Proiectarea bazelor de date,
Editura TEORA
13. **ORACLE® Database** Application Express User's Guide Release
3.0,
http://www.ORACLE.com/technology/products/database/application_express/html/doc.html
14. <https://docs.oracle.com/database/apex-5.1/AETUT/toc.htm>