

L1

Noțiuni fundamentale

1. Obiective

- Proiectarea de ansamblu a sistemului informatic
- Proiectarea de detaliu a aplicației informatice

2. Considerații teoretice

I. Să se proiecteze o aplicație după scenariul de mai jos, folosind ORACLE DBMS:

Scenariul afacerii

Unei firme ce se ocupă de modelarea și proiectarea aplicațiilor, i se cere să realizeze o aplicație pentru un client care dorește optimizarea activității în centrul medical al cărui proprietar este.

În acest centru medical funcționează mai multe cabinete medicale pe diverse specializări. Pentru specializările solicitate frecvent, pot funcționa mai multe cabinete cu același profil. Centrul medical oferă și servicii de analize medicale, precum și tratamente specializate pentru anumite afecțiuni (de ex. tratamente pentru reumatism, astm bronșic, alergii etc.). Centrul medical nu oferă servicii de chirurgie și nu are posibilitatea de a interna pacienți. Fiind un centru medical ce oferă servicii de calitate, pentru o gamă largă de afecțiuni, mulți dintre pacienții care apelează la serviciile acestuia, sunt pacienți vechi. Pentru a îmbunătăți serviciile oferite clienților, se dorește urmărirea afecțiunilor unui pacient precum și a tratamentelor recomandate în timp acestuia.

Pentru fiecare pacient se realizează, la prima vizită în centrul medical, o fișă a pacientului în care sunt păstrate informații despre pacient și antecedentele medicale ale acestuia.

Medicii folosesc și completează această fișă la fiecare vizită a pacientului.

Medicii doresc să aibă la dispoziție și informații despre toate medicamentele care se găsesc pe piață cu indicațiile și contraindicațiile lor.

II. Să se implementeze aplicația folosind utilitarul ORACLE APEX

III. Să se realizeze următoarele interogări în cadrul aplicației:

1. Afecțiunile unui pacient cu codul dat
2. Ultimele afecțiuni ale unui pacient cu codul dat
3. Afecțiunile unui pacient dat dintr-un an dat
4. Medicii pacientului cu codul dat
5. Tratamentele unui pacient cu un cod dat
6. Tratamentele unui pacient de la o anumită dată
7. Ultimul tratament al pacientului cu codul dat
8. Analizele unui pacient
9. Rezultatele analizelor unui pacient
10. Ultimele analize ale pacientului cu codul dat

11. Rezultatele ultimelor analize ale unui pacient si medicul care le-a prescris
12. Pacienții unui medic dat
13. Numărul de pacienți ai fiecărui medic, precum și un grafic asociat raportului respectiv

Răspunsuri posibile:

1. Proiectarea aplicației

Pentru proiectarea bazei de date ce va implementa cerințele afacerii, trebuie parcurse câteva etape:

1. stabilirea necesităților afacerii
2. stabilirea regulilor afacerii
3. presupunerile și ipotezele de la care pornim
4. stabilirea regulilor structurale(indică ce informații vor fi stocate și legăturile dintre acestea)
5. stabilirea regulilor procedurale(modul de lucru sau descrierea procesului afacerii respective)
6. stabilirea regulilor programabile(ceea ce nu se poate descrie în diagrama entitate-relație, se implementează prin programare)
7. stabilirea entităților și a atributelor acestora
8. elaborarea diagramei entitate-relație inițială(ERD inițial)
9. elaborarea diagramei entitate-relație finală(ERD final)
10. elaborarea tabelelor și a legăturilor dintre acestea
11. realizarea aplicației respective.

Descrieți pentru scenariul de mai sus, fiecare etapă din proiectarea aplicației

Răspunsuri posibile:

1. Necesitățile afacerii

- Centrul medical se dezvoltă rapid, iar numărul clienților este în creștere.
- Este necesar să se rețină într-o bază de date, informațiile despre un pacient, precum și istoricul afecțiunilor sale, al analizelor efectuate și al tratamentelor primite. Toate aceste informații sunt actualizate permanent.
- Un medic are posibilitatea să acceseze informații despre un anumit medicament sau altă terapie.

2. Regulile afacerii

- În acest centru medical se găsesc cabinete de diferite specialități
- Pentru specialitățile solicitate frecvent, pot funcționa mai multe cabinete cu același profil.
- Centrul medical oferă și servicii de analize medicale, precum și tratamente specializate pentru anumite afecțiuni
- Centrul medical nu oferă servicii de chirurgie și nu are posibilitatea de a interna pacienți.
- Pentru fiecare pacient se realizează, la prima vizită în centrul medical, o fișă a pacientului în care sunt păstrate informații despre pacient și antecedentele medicale ale acestuia
- Medicii folosesc și completează această fișă la fiecare vizită a pacientului.

3. Presupunerile și ipotezele de la care pornim

- Acest model este conceput pentru un singur centru medical, dar el poate fi extins.
- Un pacient este consultat la un moment dat de către un singur medic de o anumită specialitate.

- Pacienții sunt conștienți că informațiile furnizate centrului medical sunt confidențiale și sunt de acord cu acest lucru.
- Datele despre un pacient vor fi introduse într-o bază de date ce va furniza istoricul medical al acestuia

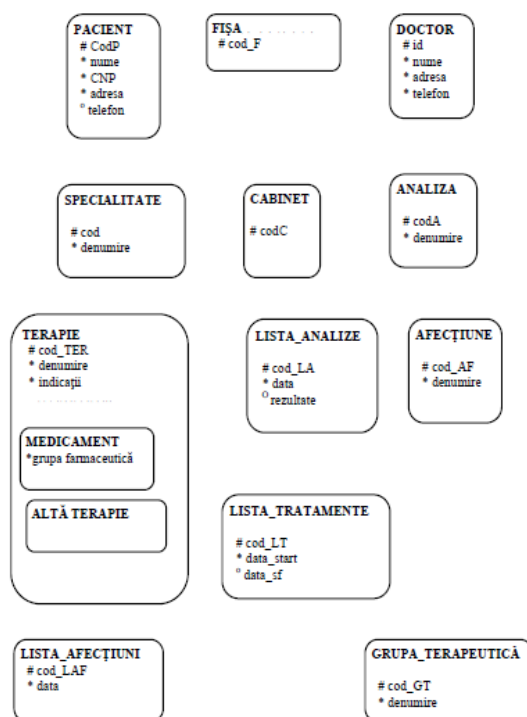
4. Reguli structurale

- Un pacient este o persoană ce are o fișă medicală în centrul medical respectiv.
- Pentru fiecare pacient se rețin datele sale personale (nume, CNP, adresa și eventual numărul de telefon) .
- În fișă medicală a unui pacient este reținut istoricul său medical (lista analizelor efectuate, afecțiunile pe care le-a avut, tratamentele pe care le-a urmat, precum și datele când s-au efectuat)
- Pentru fiecare tratament se rețin data începerii și data încheierii tratamentului respectiv (în cazul în care se cunoaște această dată).
- Cabinetele sunt de diferite specialități.
- Fiecare doctor este de o anumită specialitate.
- Pentru fiecare doctor se rețin datele sale personale (numele, adresa și numărul de telefon)
- Pentru fiecare medicament se rețin denumirea, grupa terapeutică de care aparține, indicațiile și contraindicațiile acestuia.
- Pentru fiecare terapie nemedicamentoasă se rețin denumirea, indicațiile și contraindicațiile acesteia.
- Fiecare listă de analize conține una sau mai multe analize, data când au fost făcute, medicul care a cerut a autorizat această listă precum și fișă medicală a pacientului căruia i-au fost prescrise analizele respective.
- O listă de afecțiuni conține una sau mai multe afecțiuni diagnosticate de către un medic unui pacient, codul medicului respectiv, precum și numărul fișei pacientului în cauză.
- O listă de tratamente conține unul sau mai multe tratamente prescrise de către un medic unui pacient, codul medicului respectiv, precum și numărul fișei pacientului în cauză.
- Un tratament poate fi format dintr-un medicament sau altă terapie.

5. Reguli procedurale

- O cerere de analiză este emisă pentru un singur pacient, la recomandarea unui medic, pentru o anumită analiză specificată într-o listă de analize
- Un diagnostic este autorizat de către un singur medic, pentru un singur pacient, pentru o anumită afecțiune a acestuia, specificată într-o listă de afecțiuni.
- Fiecare tratament este autorizat de către un singur medic, pentru un singur pacient, pentru o anumită terapie a acestuia, specificată într-o listă de tratamente.
- Numai un medic poate autoriza o listă de tratamente, poate diagnostica o afecțiune și poate recomanda o listă de analize pentru unul sau mai mulți pacienți.
- O listă de analize poate fi recomandată numai de către un singur medic pentru un singur pacient.
- O fișă medicală aparține unui singur pacient și fiecare pacient are o fișă medicală.

7. Stabilirea entităților și a atributelor acestora



8. Elaborarea diagramei entitate-relație inițială(ERD inițial)

DIAGRAMA ENTITAE-RELATIE INIȚIALĂ

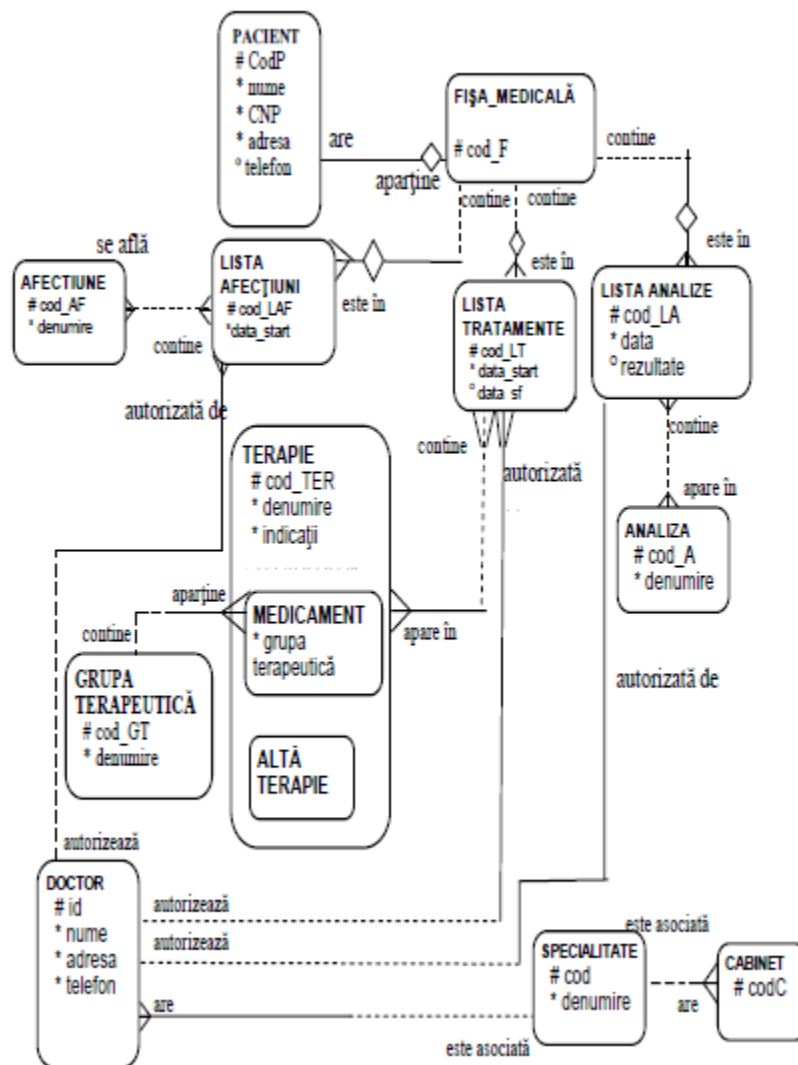
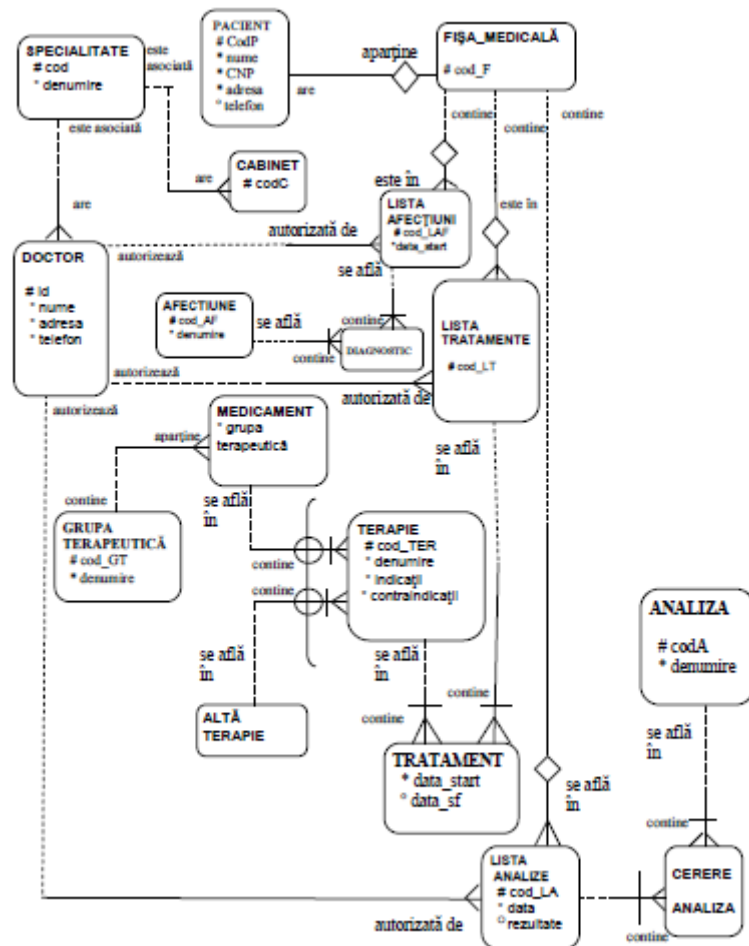
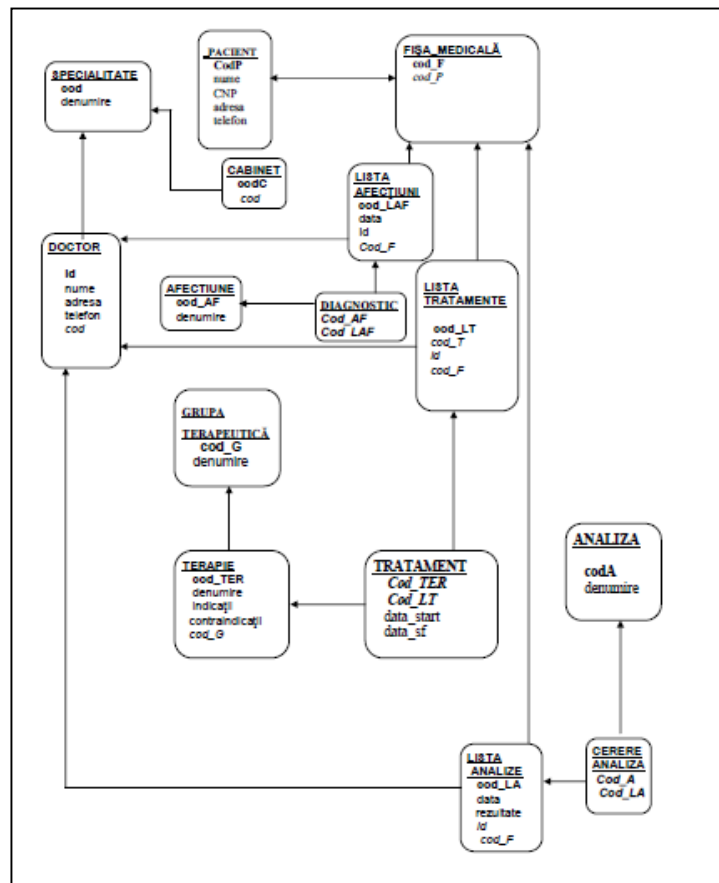


DIAGRAMA ENTITAE-RELATIE FINALĂ



10. Diagrama tablele-legături



II Implementarea aplicației folosind mediul ORACLE APEX.

Răspunsuri posibile: Se vor crea tabelele folosind *Object Browser* sau *SQL Command* din aplicația *SQL Workshop*.

CREAREA TABELELOR

Notații folosite:

CodP – codul pacientului

Cod_F – codul fișei pacientului

CodC - codul cabinetului

Cod – codul unei specialități

ID – codul unui medic

CodA – codul unei analize

Cod_LA – codul unei liste de analize

Cod_AF – codul unei afecțiuni

Cod_LAF – codul unei liste de afecțiuni

Cod_G – codul unei grupe terapeutice

Cod_TER – codul unei terapii

Cod_LT – codul unei liste de tratamente

3. CABINET

Column Name	Data Type	Nullable	Default	Primary Key
CODC	NUMBER	No	-	1
COD	NUMBER	No	-	-

```
CREATE TABLE CABINET
(
    CODC NUMBER NOT NULL,
    COD NUMBER NOT NULL,
    PRIMARY KEY (CODC),
    FOREIGN KEY (COD)
    REFERENCES SPECIALITATE(COD) ON DELETE CASCADE)
```

4. CERERE_ANALIZĂ

Column Name	Data Type	Nullable	Default	Primary Key
CODA	NUMBER(3,0)	No	-	1
COD_LA	NUMBER(3,0)	No	-	2

```
CREATE TABLE CERERE_ANALIZA
(
    CODA NUMBER(3,0) NOT NULL,
    COD_LA NUMBER(3,0) NOT NULL,
    PRIMARY KEY (CODA, COD_LA),
    FOREIGN KEY (CODA)
    REFERENCES ANALIZA(CODA),
    FOREIGN KEY (COD_LA)
    REFERENCES LISTA_ANALIZE(COD_LA) ON DELETE CASCADE)
```

6. DOCTOR

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER	No	-	1
NUME	VARCHAR2(50)	No	-	-
ADRESA	VARCHAR2(100)	No	-	-
TELEFON	VARCHAR2(15)	No	-	-
COD	NUMBER	No	-	-

```
CREATE TABLE DOCTOR
(
    ID NUMBER NOT NULL,
    NUME VARCHAR2(50) NOT NULL,
    ADRESA VARCHAR2(100) NOT NULL,
    TELEFON VARCHAR2(15) NOT NULL,
    COD NUMBER NOT NULL,
    PRIMARY KEY (ID),
    FOREIGN KEY (COD)
    REFERENCES SPECIALITATE (COD) ON DELETE CASCADE )
```

1. ANALIZA

Column Name	Data Type	Nullable	Default	Primary Key
CODA	NUMBER	No	-	1
DENUMIRE	VARCHAR2(100)	No	-	-

```
CREATE TABLE ANALIZA
(
    CODA NUMBER NOT NULL,
    DENUMIRE VARCHAR2(100) NOT NULL,
    CONSTRAINT ANALIZA_PK PRIMARY KEY (CODA),
    CONSTRAINT ANALIZA_UK1 UNIQUE (DENUMIRE)
)
```

2. AFECȚIUNE

Column Name	Data Type	Nullable	Default	Primary Key
COD_AF	NUMBER(3,0)	No	-	1
DENUMIRE	VARCHAR2(50)	No	-	-

```
CREATE TABLE AFECTIUNE
(
    COD_AF NUMBER(3,0),
    DENUMIRE VARCHAR2(50) NOT NULL,
    PRIMARY KEY (COD_AF)
```

5 DIAGNOSTIC

Column Name	Data Type	Nullable	Default	Primary Key
COD_AF	NUMBER(3,0)	No	-	1
COD_LAF	NUMBER(3,0)	No	-	2

```
CREATE TABLE DIAGNOSTIC
(
    COD_AF NUMBER(3,0) NOT NULL,
    "COD_LAF" NUMBER(3,0) NOT NULL,
    PRIMARY KEY (COD_AF, COD_LAF),
    FOREIGN KEY (COD_AF)
    REFERENCES AFECTIUNE(COD_AF),
    FOREIGN KEY (COD_LAF)
    REFERENCES LISTA_AFECTIUNI(COD_LAF) ON DELETE CASCADE)
```

7. FIȘA_MEDICALĂ

Column Name	Data Type	Nullable	Default	Primary Key
COD_F	NUMBER(3,0)	No	-	1
CODP	NUMBER(3,0)	No	-	-

```
CREATE TABLE FISA_MEDICALA
(
    COD_F NUMBER(3,0),
    CODP NUMBER(3,0) NOT NULL,
    PRIMARY KEY (COD_F),
    FOREIGN KEY (CODP)
    REFERENCES PACIENT (CODP))
```

8. GRUPA_TERAPEUTICĂ

Column Name	Data Type	Nullable	Default	Primary Key
COD_G	NUMBER(3,0)	No	-	1
DENUMIRE	VARCHAR2(100)	No	-	-

```
CREATE TABLE GRUPA_TERAPEUTICA
(
    COD_G NUMBER(3,0),
    DENUMIRE VARCHAR2(100) NOT NULL,
    PRIMARY KEY (COD_G),
    UNIQUE (DENUMIRE)
)
```

9. LISTA_AFECTIUNI

Column Name	Data Type	Nullable	Default	Primary Key
COD_LAF	NUMBER(3,0)	No	-	1
DATA	DATE	No	-	-
ID	NUMBER(3,0)	No	-	-
COD_F	NUMBER(3,0)	No	-	-

```
CREATE TABLE LISTA_AFECTIUNI
(
    COD_LAF NUMBER(3,0),
    DATA DATE NOT NULL,
    ID NUMBER(3,0) NOT NULL,
    COD_F NUMBER(3,0) NOT NULL,
    PRIMARY KEY (COD_LAF),
    FOREIGN KEY (COD_F)
        REFERENCES FISA_MEDICALA (COD_F) ON DELETE CASCADE
    FOREIGN KEY (ID)
        REFERENCES DOCTOR (ID) )
```

11. LISTA_TRATAMENTE

Column Name	Data Type	Nullable	Default	Primary Key
COD_LT	NUMBER(3,0)	No	-	1
ID	NUMBER	No	-	-
COD_F	NUMBER	No	-	-

```
CREATE TABLE LISTA_TRATAMENTE
(
    COD_LT NUMBER(3,0),
    ID NUMBER(3,0) NOT NULL,
    COD_F NUMBER(3,0) NOT NULL,
    PRIMARY KEY (COD_LT),
    FOREIGN KEY (ID)
        REFERENCES DOCTOR (ID),
    FOREIGN KEY (COD_F)
        REFERENCES FISA_MEDICALA (COD_F) ON DELETE CASCADE )
```

13. SPECIALITATE

Column Name	Data Type	Nullable	Default	Primary Key
COD	NUMBER	No	-	1
DENUMIRE	VARCHAR2(50)	No	-	-

```
CREATE TABLE SPECIALITATE
(
    COD NUMBER NOT NULL,
    DENUMIRE VARCHAR2(50) NOT NULL,
    PRIMARY KEY (COD),
    UNIQUE (DENUMIRE)
)
```

14. TERAPIE

Column Name	Data Type	Nullable	Default	Primary Key
COD_TER	NUMBER(3,0)	No	-	1
DENUMIRE	VARCHAR2(100)	No	-	-
INDICATII	VARCHAR2(100)	No	-	-
CONTRAINDICATII	VARCHAR2(100)	No	-	-
COD_G	NUMBER(3,0)	Yes	-	-

(Numai pentru medicamente avem grupa farmaceutică, pe care o găsim prin codul grupei cod_G ce este cheie străină în acest tabel și cheie primară în tabelul GRUPA_FARMACEUTICA)

```
CREATE TABLE TERAPIE
(
    COD_TER NUMBER(3,0),
    DENUMIRE VARCHAR2(100) NOT NULL,
    INDICATII VARCHAR2(100) NOT NULL,
    CONTRAINDICATII VARCHAR2(100) NOT NULL,
    COD_G NUMBER(3,0),
    PRIMARY KEY (COD_TER),
    UNIQUE (DENUMIRE),
    FOREIGN KEY (COD_G)
        REFERENCES GRUPA_TERAPEUTICA (COD_G) )
```

10. LISTA_ANALIZE

Column Name	Data Type	Nullable	Default	Primary Key
COD_LA	NUMBER(3,0)	No	-	1
DATA	DATE	No	-	-
REZULTATE	VARCHAR2(100)	Yes	-	-
ID	NUMBER(3,0)	No	-	-
COD_F	NUMBER(3,0)	No	-	-

```
CREATE TABLE LISTA_ANALIZE
(
    COD_LA NUMBER(3,0),
    DATA DATE NOT NULL,
    REZULTATE VARCHAR2(100),
    ID NUMBER(3,0) NOT NULL,
    COD_F NUMBER(3,0) NOT NULL,
    PRIMARY KEY (COD_LA),
    FOREIGN KEY (ID)
        REFERENCES DOCTOR (ID),
    FOREIGN KEY (COD_F)
        REFERENCES FISA_MEDICALA (COD_F) ON DELETE CASCADE )
```

12. PACIENT

Column Name	Data Type	Nullable	Default	Primary Key
CODP	NUMBER(3,0)	No	-	1
NUME	VARCHAR2(50)	No	-	-
CNP	VARCHAR2(14)	No	-	-
ADRESA	VARCHAR2(50)	No	-	-
TELEFON	VARCHAR2(15)	Yes	-	-
COD_F	NUMBER(3,0)	No	-	-

```
CREATE TABLE PACIENT
(
    CODP NUMBER(3,0),
    NUME VARCHAR2(50) NOT NULL,
    CNP VARCHAR2(14) NOT NULL,
    ADRESA VARCHAR2(50) NOT NULL,
    TELEFON VARCHAR2(15),
    COD_F NUMBER(3,0) NOT NULL,
    PRIMARY KEY ("CODP")
)
```

15. TRATAMENT

Column Name	Data Type	Nullable	Default	Primary Key
COD_TER	NUMBER	No	-	1
COD_LT	NUMBER	No	-	2
DATA_START	DATE	No	-	-
DATA_SF	DATE	Yes	-	-

```
CREATE TABLE TRATAMENT
(
    DATA_START DATE NOT NULL,
    DATA_SF DATE,
    COD_TER NUMBER NOT NULL,
    COD_LT NUMBER NOT NULL,
    PRIMARY KEY (COD_TER, COD_LT),
    FOREIGN KEY (COD_TER) REFERENCES TERAPIE (COD_TER),
    FOREIGN KEY (COD_LT) REFERENCES LISTA_TRATAMENTE (COD_LT)
    ON DELETE CASCADE )
```

Popularea tabelelor create anterior

Răspunsuri posibile:

Tabelele se vor popula din *Object Browser* sau direct din aplicația creată cu *Application Builder*

Tabela PACIENT

CODP	NUME	CNP	ADRESA	TELEFON
1	IONESCU MIHAI	1780405126354	PITESTI, LIVEZILOR NR 7	0248212911
2	MARIN IOANA	2680204254879	BUCURESTI, LAHOVARI NR 45	0212256283
3	IONESCU LIDIA	2800904235648	PITESTI, EXERCITIU NR 56	-
4	ENACHE IONELA	2600201124578	PITESTI, DUMBRAVEI NR 14	-
5	PETRESCU MIRCEA	15630201054556	PITESTI, I.C. BRATIANU NR 45	-

Tabela SPECIALITATE

COD	DENUMIRE
1	ORL
2	OFTALMOLOGIE
3	MEDICINA GENERALA
4	DERMATOLOGIE
5	PSIHIATRIE
6	REUMATOLOGIE
7	ALERGOLOGIE
8	STOMATOLOGIE
9	FIZIOTERAPIE
10	PEDIATRIE

Tabela CABINET

CABINET

Reset

Search

Display

15

Go

COD CABINET	Denumire
1	dermatologia
2	ori
3	dermatologie
4	medicina generala
1 - 4	
Spread Sheet	
Modific	Evidenta Cabinetelor

Tabela DOCTOR

ID	NUME	ADRESA	TELEFON	COD
1	PASCU ION	STR NICOLAE BALCESCU NR 13 PITESTI	0745667908	11
2	STAN EMIL	STR NICOLAE BALCESCU NR 7 PITESTI	0723445006	8
3	DUTA ION	STR LIVEZILOR NR 3 PITESTI	0786908678	3
4	SMEUREANU NICOLETA	STR EXERCITIU NR 4 PITESTI	0723456678	5
5	PENESCU VALENTIN	STR I C BRATIANU NR 23 PITESTI	0786908678	1
6	GORBANESCU RODICA	STR 1 MAI NR 6 PITESTI	0723456678	1
7	HERMAN MIHAELA	STR I C BRATIANU NR 45 PITESTI	0786967678	3
8	VANCEA PETRE	STR FRASINULUI NR45 BUCURESTI	0723456789	2
9	SORESCU MARIA	STR 1 MAI NR 46 PITESTI	078696678	4
10	GIURCANEANU EMIL	B-DUL EROILOR NR 7 BUCURESTI	0786967543	7
11	ION MARIN	B-DUL PETROCHIMISTILOR NR 54 PITESTI	0786966421	6
12	PASCU EMIL	STR LIVEZILOR NR 7 PITESTI	0745334607	7
13	IONESCU DAN	B-DUL EROILOR NR 45 BUCURESTI	0746567897	9

Tabela DIAGNOSTIC

SELECT cod_laf "lista afectiune",cod_af "codul afectiunii"

FROM diagnostic

ORDER BY cod_laf

Tabela AFECTIUNE

COD_AF	DENUMIRE
1	REUMATISM
2	RINITA ALERGICA
3	AMIGDALITA
4	AFTE
5	PNEUMONIE
6	MIGRENA
7	ARSURI SOLARE
8	BRONSITA
9	OTITA
10	DIABET

Tabela GRUPA_TERAPEUTICA

COD_G	DENUMIRE
1	ANTIANEMICE
2	ANTIACIDE
3	ANTIASTMATICE
4	TOPICE OROFARINGIENE
5	TOPICE NAZALE
6	TOPICE OTICE
7	VITAMINE
8	ANTIIDIABETICE
9	ANALGEZICE
10	ANTIREFUMATICE

Tabela TERAPIE

COD TER	DENUMIRE	INDICATII	CONTRAINDICATII	COD G
1	ANTINEVRALGIC	CEFALEE,NEVRALGII, DURERI, VIROZE RESPIRATORII	PRUDENTA IN CAZ DE HEPATITA, ULCER GASTROINTESTINAL SI ANEMIE	9
2	ACID ACETIL SALICILIC	NEVRALGII, REUMATISM, VIROZE RESPIRATORII, DURERI, ARSURI SOLARE	ALERGIE LA SALICILATI, ULCER GASTRO-DUODENAL, PRUDENTA IN ASTM BRONSIAC	9
3	MASAJ	REUMATISM	IRITATII SAU INFLAMATII ALE TESUTULUI CUTANAT	-
4	EXERCITII RECUPERATORII	APECTIUNI FIZIOLOGICE ALE MEMBRELOR INFERIOARE SI SUPERIOARE	PRUDENTA LA PERSOANELE CARDIACE	-
5	AMPICILINA	OTITE, PNEUMONII, BRONSITE	ALERGIE LA PENICILINE	12
6	PENICILINA	BRONSITE, PNEUMONII, INFECTII	ALERGIE LA PENICILINE	12
7	MINIDIAB	DIABET NON-INSULINODEPENDENT	DIABET INSULINODEPENDENT	8
8	ELECTOVIT	ANEMII	NU POSEDA	7
9	FENOSEPT	FARINGITE, AMIGDALITE, AFTE, STOMATITE	NU SE ADMINISTREAZA COPILOR SUB 2 ANI	4
10	RHINXYL	RINITE, SINUZITE	ALERGIE LA UNA DIN COMPONENTELE SOLUTIEI	5

Tabela LISTA_TRATAMENTE

SELECT cod_lt "LISTA TRATAMENTELOR", COD_TER "CODUL TERAPIEI",DATA_START,DATA_SF

FROM tratament

ORDER BY cod_lt

Tabela ANALIZA

CODA	DENUMIRE
1	HEMOLEUCOGRAMA
2	HEMOGLOBINA
3	GLICEMIE
4	ELECTROFOREZA PROTEINELOR SERICE
5	FACTOR REUMATOID
6	VSH
7	ANTIBIOGRAMA
9	EXUDAT FARINGIAN
10	ECHOGRAFIE
11	EXUDAT NAZAL

III. Interogări în cadrul aplicației:

Răspunsuri posibile:

1. AFECTIUNILE UNI PACIENT CU CODUL DAT(Ex:1)

SELECT b.DENUMIRE "AFECTIUNE", c.DATA, d.num "MEDIC"

FROM pacient a, afectiune b, lista_afectiuni c, doctor d, fisa_medicala e, diagnostic f

WHERE (a.codp=e.codp) and
(e.cod_f=c.cod_f)and(c.cod_laf=f.cod_laf)and
(f.cod_af=b.cod_af) and(c.id=d.id)and (a.codp=1)
order by c.data

AFECTIUNE	DATA	MEDIC
REUMATISM	12-03-2006	ION MARIN
AFTE	12-03-2006	ION MARIN
MIGRENA	12-03-2006	ION MARIN
RINITA ALERGICA	03-04-2006	GORBANESCU RODICA
AMIGDALITA	03-04-2006	GORBANESCU RODICA
REUMATISM	15-03-2007	ION MARIN

2. ULTIMELE AFECTIUNI ALE UNUI PACIENT CU CODUL DAT(Ex:1)

SELECT b.DENUMIRE "AFECTIUNE",c.data,d.num "MEDIC"

FROM pacient a, afectiune b, lista_afectiuni c, doctor d, fisa_medicala e, diagnostic f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_laf=f.cod_laf)and (f.cod_af=b.cod_af)
and(c.id=d.id)and (a.codp=1)and

(c.data=
(**SELECT** MAX(c.data)

FROM pacient a, afectiune b, lista_afectiuni c, doctor d, fisa_medicala e, diagnostic f

WHERE (a.codp=e.codp) and
(e.cod_f=c.cod_f)and(c.cod_laf=f.cod_laf)and
(f.cod_af=b.cod_af) and(c.id=d.id)and (a.codp=1)))

AFECTIUNE	DATA	MEDIC
REUMATISM	15-03-2007	ION MARIN

3. AFECTIUNILE UNUI PACIENT CU CODUL DAT(1) DINTR-UN AN DAT (EX: 2006)

SELECT b.DENUMIRE "AFECTIUNE"

FROM pacient a, afectiune b, lista_afectiuni c, doctor d, fisa_medicala e, diagnostic f

WHERE (a.codp=e.codp) and
(e.cod_f=c.cod_f)and(c.cod_laf=f.cod_laf)and (f.cod_af=b.cod_af)
and(c.id=d.id)and (a.codp=1)and (to_char(c.data,'yyyy')='2006')

AFECTIUNE	DATA
REUMATISM	12-03-2006
AFTE	12-03-2006
MIGRENA	12-03-2006
RINITA ALERGICA	03-04-2006
AMIGDALITA	03-04-2006

4. MEDICII PACIENTULUI CU CODUL DAT (Ex:1)

SELECT DISTINCT d.num, g.denumire

FROM pacient a, afectiune b, lista_afectiuni c, doctor d, fisa_medicala e, diagnostic f, specialitate g

WHERE (a.codp=e.codp) and
(e.cod_f=c.cod_f)and(c.cod_laf=f.cod_laf)and
(f.cod_af=b.cod_af) and(c.id=d.id)and
(a.codp=1)and(g.cod=d.cod)

ORDER BY d.num

NUME	DENUMIRE
GORBANESCU RODICA	ORL
ION MARIN	REUMATOLOGIE

5. TRATAMENTELE UNUI PACIENT CU UN COD DAT(Ex: 1)

SELECT b.DENUMIRE "TRATAMENT",f.data_start,f.data_sf

FROM pacient a, terapie b, lista_tratamente c, doctor d, fisa_medicala e, tratament f

WHERE (a.codp=e.codp) and
(e.cod_f=c.cod_f)and(c.cod_lt=f.cod_lt)and (f.cod_ter=b.cod_ter)
and(c.id=d.id)and (a.CODP=1)

ORDER BY f.data_start

TRATAMENT	DATA_START	DATA_SF
ACID ACETIL SALICILIC	14-03-2006	14-04-2006
VOLTAREN	14-03-2006	24-03-2006
MASAJ	14-03-2006	24-03-2006
FENOSEPT	03-04-2006	08-04-2006
RHINXYL	03-04-2006	08-04-2006
PARACETAMOL	03-04-2006	08-04-2006
ACID ACETIL SALICILIC	17-03-2007	26-03-2007
FASTUM GEL	20-03-2007	30-03-2007
MASAJ	20-03-2007	30-03-2007

6. TRATAMENTELE UNUI PACIENT(CU CODUL 1) DE LA O ANUMITA DATA

(Ex:14.03.06)

SELECT b.denumire "TRATAMENT",f.data_sf "SFARSITUL TRATAMENTULUI", d.num "MEDIC"

FROM pacient a, terapie b,lista_tratamente c, doctor d, fisa_medicala e,tratament f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_lt=f.cod_lt)and (f.cod_ter=b.cod_ter) and(c.id=d.id)and (a.codp=1)and (f.data_start='14.03.06')

TRATAMENT	SFARSITUL TRATAMENTULUI	MEDIC
VOLTAREN	24-03-2006	ION MARIN
MASAJ	24-03-2006	ION MARIN
ACID ACETIL SALICILIC	14-04-2006	ION MARIN

7. ULTIMUL TRATAMENT AL PACIENTULUI CU CODUL DAT(Ex:1)

SELECT b.denumire "TRATAMENT",f.data_start "INCEPUTUL TRATAMENTULUI",f.data_sf "SFARSITUL TRATAMENTULUI", d.num "MEDIC"

FROM pacient a, terapie b,lista_tratamente c, doctor d, fisa_medicala e,tratament f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_lt=f.cod_lt)and (f.cod_ter=b.cod_ter) and(c.id=d.id)and (a.codp=1)and (f.data_start = (SELECT MAX(F.DATA_START)

FROM pacient a, terapie b,lista_tratamente c, doctor d, fisa_medicala e,tratament f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_lt=f.cod_lt)and (f.cod_ter=b.cod_ter) and(c.id=d.id)and (a.codp=1)))

TRATAMENT	INCEPUTUL TRATAMENTULUI	SFARSITUL TRATAMENTULUI	MEDIC
MASAJ	20-03-2007	30-03-2007	ION MARIN
FASTUM GEL	20-03-2007	30-03-2007	ION MARIN

8. ANALIZELE UNUI PACIENT(Ex:1)

SELECT b.DENUMIRE "ANALIZA", c.DATA

FROM pacient a, analiza b, lista_analize c, doctor d, fisa_medicala e,cerere_analiza f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_la=f.cod_la)and (f.coda=b.coda)

and(c.id=d.id)and (a.CODP=1)

ORDER BY c.data

ANALIZA	DATA
FACTOR REUMATOID	12-03-2006
HEMOLEUCOGRAMA	12-03-2006
VSH	12-03-2006
EXUDAT NAZAL	03-04-2006
EXUDAT FARINGIAN	03-04-2006
ANTIBIOGRAMA	03-04-2006
FACTOR REUMATOID	15-03-2007
VSH	15-03-2007

9. REZULTATELE ANALIZELOR UNUI PACIENT(Ex:1)

SELECT DISTINCT c.DATA, C.REZULTATE, D.NUME

FROM pacient a, analiza b, lista_analize c, doctor d, fisa_medicala e,cerere_analiza f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_la=f.cod_la)and (f.coda=b.coda) and(c.id=d.id)and (a.CODP=1)

ORDER BY c.data

DATA	REZULTATE	NUME
12-03-2006	VSH PESTE LIMITA VALORILOR NORMALE, FACTOR REUMATOID MARE	ION MARIN
03-04-2006	RINITA ALERGICA, BACTERII	GORBANESCU RODICA
15-03-2007	FACTORUL REUMATOID DEPASESTE LIMITELE ADMISE, VSH DEPASESTE LIMITA VALORILOR NORMALE	ION MARIN

10. ULTIMELE ANALIZE ALE PACIENTULUI CU CODUL DAT(Ex:1)

SELECT b.denumire „ANALIZA”,c.data

FROM pacient a, analiza b,lista_analize c, doctor d, fisa_medicala e,cerere_analiza f

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_LA=f.cod_LA)and (f.codA=b.codA) and(c.id=d.id)and (a.codp=1)and (c.data= (SELECT MAX(c.data)

FROM pacient a, analiza b,lista_analize c, doctor d, fisa_medicala e,cerere_analiza f

ANALIZA	DATA
VSH	15-03-2007
FACTOR REUMATOID	15-03-2007

WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_LA=f.cod_LA)and (f.codA=b.codA) and(c.id=d.id)and (a.codp=1)))

11. REZULTATELE ULTIMELOR ANALIZE ALE UNUI PACIENT SI MEDICUL CARE LE-A PRESCRIS

SELECT DISTINCT c.data ,c.rezultate "REZULTATE",d.numa "MEDIC"
FROM pacient a, analiza b,lista_analize c, doctor d, fisa_medicala e,cerere_analiza f
WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_LA=f.cod_LA)and (f.codA=b.codA) and(c.id=d.id)and (a.codp=1)and (c.data=(**SELECT MAX**(c.data)
FROM pacient a, analiza b,lista_analize c, doctor d, fisa_medicala e,cerere_analiza f
WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.cod_LA=f.cod_LA)and (f.codA=b.codA) and(c.id=d.id)and (a.codp=1)))

DATA	REZULTATE	MEDIC
15-03-	FACTORUL REUMATOID DEPASESTE LIMITELE ADMISE, VSH DEPASESTE LIMITA	ION
2007	VALORILOR NORMALE	MARIN

12. PACIENȚII UNUI MEDIC DAT(Ex: Id=7)

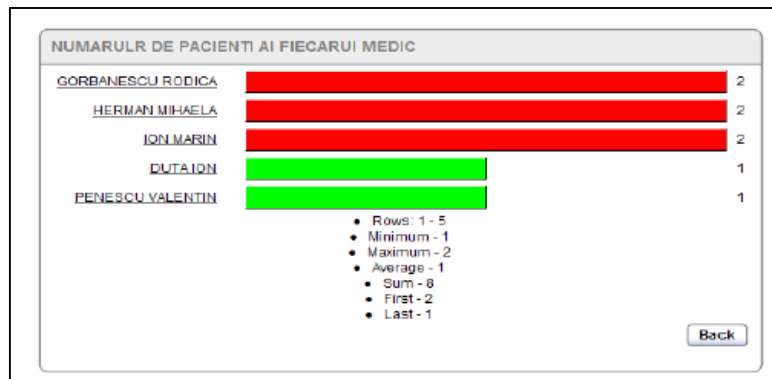
SELECT a.numa
FROM pacient a, lista_afectiuni c, doctor d, fisa_medicala e
WHERE (a.codp=e.codp) and (e.cod_f=c.cod_f)and(c.id=d.id)and (d.id=7)

NUME PACIENT
PETRESCU MIRCEA
ENACHE IONELA

13. NUMĂRUL DE PACIENȚI AI FIECĂRUI MEDIC

SELECT a.id, a.numa "MEDIC",count(DISTINCT b.cod_f)
"NUMARUL PACIENTILOR"
FROM doctor a, lista_afectiuni b, fisa_medicala c
WHERE (a.id=b.id)and(b.cod_f=c.cod_f)
GROUP BY a.id,a.numa
ORDER BY count(DISTINCT b.cod_f)DESC

ID	MEDIC	NUMARUL PACIENTILOR
6	GORBANESCU RODICA	2
7	HERMAN MIHAELA	2
11	ION MARIN	2
3	DUTA ION	1
5	PENESCU VALENTIN	1



ID	MEDIC	NUMARUL PACIENTILOR
6	GORBANESCU RODICA	2
7	HERMAN MIHAELA	2
11	ION MARIN	2
3	DUTA ION	1
5	PENESCU VALENTIN	1
1 - 6		

NUMĂRUL DE PACIENȚI AI FIECĂRUI MEDIC

În figura de mai sus se observă o hartă și un raport bazate pe interogarea anterioară(cu numărul 13) ce selectează medici care au pacienți, precum și numărul de pacienți ai acestora, în ordine descrescătoare.

3. Desfășurarea lucrării

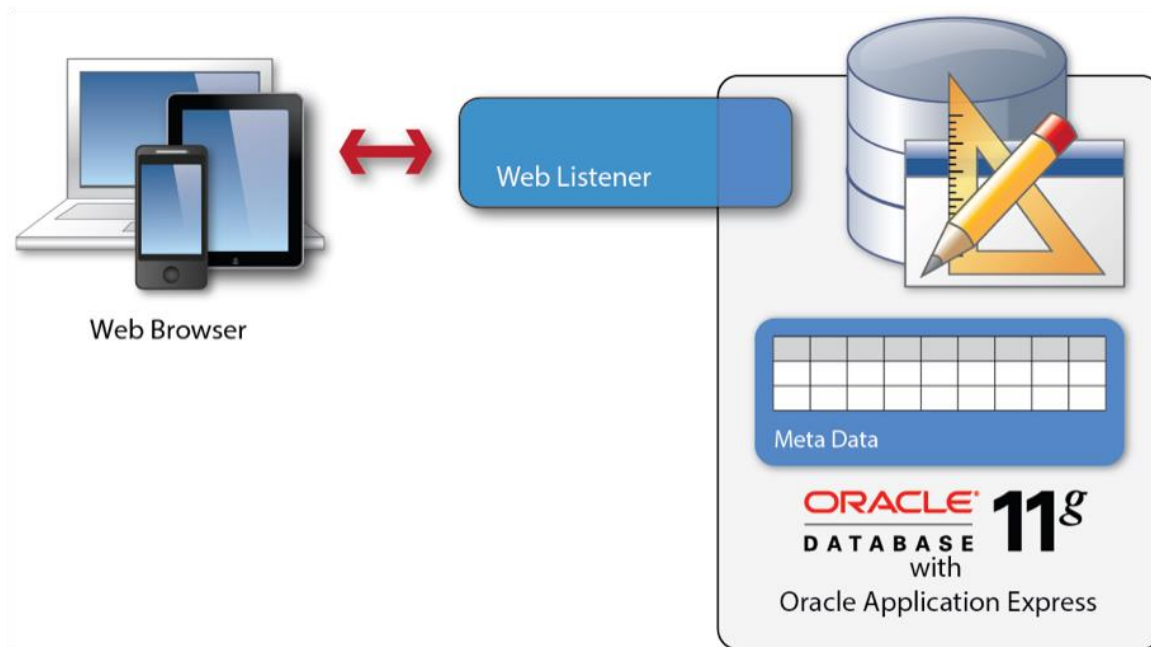
Oracle Application Express, cunoscut ca APEX, este *tool* de Rapid Application Development (RAD) care a atins un nivel de maturitate o dată cu lansarea versiunii 4.0 în iunie 2010. APEX combină ciclurile de dezvoltare rapidă pentru aplicații web, în jurul unei baze de date Oracle. Tehnica de programare este foarte declarativă și se desfășoară într-un mediu *web-based*, investind un efort de codare minim.

APEX folosește un concept unic care poate fi considerat opusul trendurilor actuale din dezvoltare aplicațiilor web. În timp ce acum majoritatea aplicațiilor web ar trebui să fie cuplate cât mai "loose" de bazele de date din *backend*, cu o emfază pe interacțiunea *client-side*, APEX are o abordare radicală, în care totul este stocat în bazele de date, de la datele până la meta-datele folosite pentru generarea paginilor web. Un *web server* al RDBMS-ului Oracle este folosit pentru a genera pagini HTML direct din baza de date, unde atât datele folosite de aplicație cât și meta-datele care descriu paginile aplicațiilor. Cu toate că APEX este un produs gratis, acesta funcționează doar cu bazele de date Oracle, marea parte a procesărilor de *backend*, dar și *frontend* fiind realizate de procedurile stocate.

O aplicație web Oracle APEX este dezvoltată folosindu-se SQL și PL-SQL, cu toate că marea parte a efortului de codare poate fi realizat într-un mod declarativ, folosind interfața de dezvoltare din *web browser*. APEX este un *tool database-centric*, adică necesită și rulează doar cu o bază de date Oracle. Istoria APEX începe în anul 2004, pe când era doar un *tool* intern al Oracle numit HTML DB. În 2006 a fost redenumit în Oracle Application Express (versiunea 2.1). În acest moment versiunea stabilă este 4.2.4 și deja este lansată și versiunea *early adopter* 5.0 (<https://apexea.oracle.com/i/index.html>).

Pentru a utiliza APEX în cadrul unei instanțe a unei baze de date Oracle, chiar și cu varianta *free* de baze de date Oracle XE, nu este nevoie de *licensing* adițional, pentru că numărul de dezvoltatori, aplicații și end-users nu este restricționat. Suportă toate versiunile de baze de date Oracle începând cu 10gR2 și poate fi folosit și cu *setup-uri* Exadata, ORA și RAC. În mod implicit, Oracle APEX este distribuit cu toate edițiile bazelor de date Oracle.

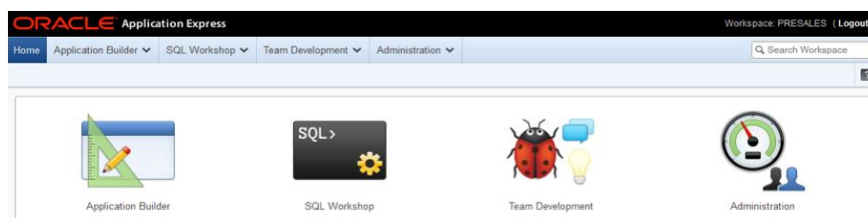
Din punct de vedere arhitectural, APEX folosește o arhitectură simplă de tip 2-tier. Paginile web sunt generate dinamic folosindu-se metadata stocată în baza de date și nu se generează cod compilabil sub forma unor fișiere. De fapt, APEX rulează o dată cu baza de date. APEX folosește un principiu de *multitenant hosting*, organizând paginile web în aplicații și *workspace-uri*, care pot folosi la rândul lor baze de date distincte sau *shared*.



Cu toate că marea parte a codului din spate este scris în PLSQL, pentru a începe să lucrezi cu APEX nu ai nevoie de cunoștințe avansate de programare, cu excepția unor elemente de HTML. Fiind un *tool web-based*, procesul de *development* consistă dintr-o serie de pagini și obiecte predefinite, de la formulare și rapoarte până la grafice. Toate paginile și componentele sunt stocate în obiecte din baza de date Oracle, de obicei tabele și *view*-uri, așa că este pus la dispoziție și un *tool de management* a schemei bazei de date. Crearea de tabele, *views* și proceduri stocate se poate face direct în APEX, așa încât întregul proces de dezvoltare se face în același *environment*, cel de dezvoltare *web-based*.

APEX se poate accesa prin intermediul unui URL într-un *browser web*, fie că folosiți o instanță instalată local, o instanță de *private cloud* (SaaS) sau serviciul Oracle Database Cloud Service, produsul *cloud* al Oracle care folosește APEX pentru dezvoltare de aplicații web (<http://cloud.oracle.com>).

Cu toate acestea, Oracle APEX nu este un *tool* care este potrivit oricărui proiect. Cazurile tipice unde APEX poate fi folosit sunt cele de aplicații *data-driven* (aplicații de productivitate la nivel de departament sau aplicații ad-hoc), reporting online bazat de *query*-uri SQL, transformarea *spreadsheet*-urilor Excel sau de alt tip în aplicații web sau pentru centralizarea accesului la date (unde APEX se poate fi folosit ca un central point of access pentru scheme multiple într-una sau mai multe baze de date Oracle).



Principalele componente ale mediului de dezvoltare APEX sunt:

Application Builder, unde sunt construite paginile și aplicațiile *web* în mod declarativ prin folosirea unor *wizards*. Fiecare aplicație este compusă din una sau mai multe pagini, fiecare pagină este împărțită în regiuni. Fiecare regiune a unei pagini poate să conțină test, cod PLSQL, rapoarte, grafice, hărți, calendare, formulare sau rezultate aduse prin intermediul unor servicii web. De asemenea sunt disponibile obiecte care sunt specifice nu doar paginilor, ci întregii aplicații, cum ar fi *application items*, *processes*, *computations*, scheme de autentificare și autorizare sau obiecte de navigare ca *tabs*, *lists* sau *breadcrumbs*.

SQL Workshop, un *tool* care permite managementul obiectelor din baza de date Oracle. *Query-uri* SQL ad-hoc, *wizards* pentru crearea de tabele, *view-uri*, proceduri stocate și alte obiecte de baze de date pot fi utilizate de developer pentru a face management-ul schemei Oracle din acest *tool browser-based*.

Team Development, o unealtă de *team management* pentru development pentru urmărirea *feature*-urilor, *bugs* și *milestones*. Acest *tool* este legat direct de paginile APEX.

Administration, pentru administrarea contului, a *workspace*-urilor și a *dashbord*-urilor.

Article	Date	Author Email	Author URL	Author IP
XP4maintenance: simplified and customisable maintenance work order system	11.02.2013 13:18:46		mail.blogspot.com	78.96.65.20
XP4maintenance: simplified and customisable maintenance work order system	25.08.2012 19:43:00			201.153.242.5
XP4maintenance: simplified and customisable maintenance work order system	21.04.2012 04:06:29			188.24.122.203
The Bucovina Shepherd Dog	19.04.2012 14:08:08		www.apexninja.com	188.24.122.203
The Bucovina Shepherd Dog	22.11.2011 00:00:00	John Doe	john@john.com	127.0.0.1

Ca majoritate uneltelor RAD, Oracle APEX facilitează dezvoltarea de aplicații într-un mod declarativ, folosind item-uri de pagini web deja existente cum ar fi: rapoarte, formulare, grafice, calendare, *template-uri* de UI, navigație, validări, procese de pagină și aplicație, servicii web, servicii de e-mail și localizare (traducere), autentificare, autorizare, *logging* și monitorizare.

Bibliografie:

Pentru mai multe detalii legate de dezvoltare aplicațiilor APEX, consultați:

"Oracle APEX Reporting Tips&Tricks": <http://www.apexninja.com/blog/2013/06/oracle-apex-reporting-tips-tricks-out-now/>