

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică**

**și Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Teză de an**

***Platforma Web pentru Diagnostic si Tratament***

**Student: Anna Groza, gr. MI-201**

**Coordonator: Mihail Perebinos, doctor, conf. univ.**

**Chișinău 2021**

Cuprins

[Introducere în SGBD 3](#_Toc90218340)

[1.1 Conceptul de Bază de Date 3](#_Toc90218341)

[1.2 Gestiunea Bazelor de Date 4](#_Toc90218342)

[Modelul Conceptual. Modelul Logic. 6](#_Toc90218343)

[2.1 Scopul și obiectivele lucrării 6](#_Toc90218344)

[2.2 Descrierea domeniului de studii 6](#_Toc90218345)

[2.3 Schema legăturilor între subdomeniile domeniului de studiu 7](#_Toc90218346)

[2.4 Legături între obiectele Domeniului de Studiu 8](#_Toc90218347)

[2.5 Caracteristicile informaționale ale SAPI “Platformă Web pentru Diagnostic și Tratament” 8](#_Toc90218348)

[2.6 Tabel de acoperire 9](#_Toc90218349)

[2.7 Legături între atributele obictelor Domeniului de Studiu ER – schema BD 11](#_Toc90218350)

[2.8 Modelul Logic 12](#_Toc90218351)

[2.9 Machetul bazei de date 14](#_Toc90218352)

[Introducere Programare WEB 15](#_Toc90218353)

[3.1 HTML prezentare 15](#_Toc90218354)

[3.2 Limbajul de scriptare 17](#_Toc90218355)

[3.4 Serverul Web Apache 20](#_Toc90218356)

[3.5 Administrarea Bazelor de Date cu MySQl 21](#_Toc90218357)

[Structura Bazei de date 23](#_Toc90218358)

[4.1 Designer-ul Bazei de Date 23](#_Toc90218359)

### Introducere în SGBD

* 1. Conceptul de Bază de Date

O bază de date reprezintă un ansamblu de date integrat, anume structurat şi dotat cu o descriere a acestei structuri. Descrierea structurii poartă numele de dicţionar de date sau metadate şi crează o interdependenţă între datele propriu-zise şi programe.

Datele dintr-o bază de date pot fi structurate pe 3 niveluri, în funcţie de categoria de personal implicată:

* nivelul conceptual (global) – exprimă viziunea administratorului bazei de date asupra datelor. Acestui nivel îi corespunde structura conceptuală (schema) a bazei de date, prin care se realizează o descriere a tuturor datelor, într-un mod independent de aplicaţii, ce face posibilă administrarea datelor.
* nivelul logic – exprimă viziunea programatorului de aplicaţie asupra datelor. La acest nivel se realizează o descriere a datelor corespunzătoare unui anumit program de aplicaţie.
* nivelul fizic – care exprimă viziunea inginerului de sistem asupra datelor. Corespunde schemei interne a bazei de date prin care se realizează o descriere a datelor pe suport fizic de memorie.

Se cunosc mai multe tipuri de baze de date după modul de organizare, modul de dispunere pe suport magnetic a informaţiei şi a elementelor componente:

* modele primitive – datele sunt organizate la nivel logic în fişiere, structura de bază este înregistrarea, mai multe înregistrări fiind grupate în structuri de tip fişier;
* baze de date ierarhice – legăturile dintre date sunt ordonate unic, accesul se face numai prin vârful ierarhiei, un subordonat nu poate avea decât un singur superior direct şi nu se poate ajunge la el decât pe o singură cale;
* baze de date în reţea – datele sunt reprezentate ca într-o mulţime de ierarhii, în care un membru al ei poate avea oricâţi superiori, iar la un subordonat se poate ajunge pe mai multe căi;
* baze de date relaţionale – structura de bază a datelor este aceea de relaţie – tabel, limbajul SQL (Structured Query Language) este specializat în comenzi de manipulare la nivel de tabel. Termenul relaţional a fost introdus de un cercetător al firmei IBM dr. E. F. Codd în 1969 cel care a enunţat cele 13 reguli de bază necesare pentru definerea unei baze de date relaţionale. Baza de date relaţională reprezintă o mulţime structurată de date, accesibile prin calculator, care pot satisface în timp minim şi într-o manieră selectivă mai mulţi utilizatori. Această mulţime de date modelează un sistem sau un proces din lumea reală şi serveşte ca suport unei aplicaţii informatice;
* baze de date distribuite – sunt rezultatul integrării tehnologiei bazelor de date cu cea a reţelelor de calculatoare. Sunt baze de date logic integrate, dar fizic distribuite pe mai multe sisteme de calcul. Integrarea bazei de date distribuite se face cu ajutorul celor 3 tipuri de scheme care sunt implementate:

1. schema globală – defineşte şi descrie toate informaţiile din baza de date distribuită în reţea;
2. schema de fragmentare – descrie legăturile dintre o colecţie globală şi fragmentele sale. Ea este de tipul unu la mai mulţi şi are forma unei ierarhii;
3. schema de alocare – descrie modul de distribuire a segmentelor pe calculatoarele (nodurile) din reţea. Fiecare segment va avea o alocare fizică pe unul sau mai multe calculatoare. Schema de alocare introduce o redundanţă minimă şi controlată: un anumit segment se poate regăsi fizic pe mai multe calculatoare.
   1. Gestiunea Bazelor de Date

Sistemele de gestiune a bazalor de date (SGBD) sunt sisteme informatice specializate în stocarea şi prelucrarea unui volum mare de date, numărul prelucrărilor fiind relativ mic.

Termenul de bază de date se va referi la datele de prelucrat, la modul de organizare a acestora pe suportul fizic de memorare, iar termenul de gestiune se va referi la:

* inserarea în baza de date;
* ştergerea datelor din baza de date;
* consultarea bazei de date – interogare/extragerea datelor.

În plus un SGBD mai asigură şi alte servicii:

* suport pentru limbaj de programare;
* interfaţă cât mai atractivă pentru comunicare cu utilizatorul;
* tehnici avansate de memorare semnifică totalitatea operaţiilor ce se aplică asupra datelor din baza de date.

Un SGBD trebuie să asigure următoarele funcţii:

* definirea – crearea bazei de date;
* introducerea (adăugarea) datelor în baza de date;
* modificarea unor date deja existente, organizare, accesare a datelor din baza de date;
* utilitare încorporate: sistem de gestiune a fişierelor, liste, tabele etc.;
* “help” pentru ajutarea utilizatorului în lucrul cu baza de date.

Un SGBD trebuie să asigure următoarele activităţi:

* definirea şi descrierea structurii bazei de date – se realizează printr-un limbaj propriu, limbaj de definire a datelor (LDD), conform unui anumit model de date;
* încărcarea datelor în baza de date – se realizează prin comenzi în limbaj propriu, limbaj de manipulare a datelor (LMD);
* accesul la date – se realizează prin comenzi specifice din limbajul de manipulare a datelor. Accesul la date se referă la operaţiile de interogare şi actualizare.

Interogarea este complexă şi presupune vizualizarea, consultarea, editarea de situaţii de ieşire (rapoarte, liste, regăsiri punctuale).

Actualizarea presupune 3 operaţiuni: adăugare, modificare efectuate prin respectarea restricţiilor de integritate ale BD şi ştergere;

* întreţinerea bazei de date – se realizează prin utilitare proprii ale SGBD;
* reorganizarea bazei de date – se face prin facilităţi privind actualizarea structurii de date şi modificarea strategiei de acces. Se execută de către administratorul bazei de date;
* securitatea datelor – se referă la asigurarea confidenţialităţii datelor prin autorizarea şi controlul accesului la date, criptarea datelor.

Evidenţiem următoarele funcţii ale unui SGBD.

* funcţia de descriere a datelor – se face cu ajutorul LDD, realizându-se descrierea atributelor din cadrul structurii BD, legăturile dintre entităţile BD, se definesc eventualele criterii de validare a datelor, metode de acces la date, integritatea datelor. Concretizarea acestei funcţii este schema BD.
* funcţia de manipulare – este cea mai complexă şi realizează actualizarea şi regăsirea datelor.
* funcţia de utilizare – asigură mulţimea interfeţelor necesare pentru comunicare a tuturor utilizatorilor cu BD.

### Modelul Conceptual. Modelul Logic.

2.1 Scopul și obiectivele lucrării

Scopul:

Crearea unui SGBD privind gestionarea datelor care va permite utilizatorilor să-și facă diagnosticul sinestătător.

Obiectivul nr. 1

De a analiza simptomatica utilizatorilor.

Obiectivul nr. 2:

În baza simptomaticii, de a prezice posibila îmbolnăvire.

Obiectivul nr. 3:

De a analiza simptomatica, imbolnăvirile posibile și de a propune tratamentul.

Aplicația va fi destinată administratorilor, administratorilor ale bazelor de date, sefilor care gestionează afacerea cât și publicului larg

Notă : XXXX, YYYY se observa de odata un domeniu se refera la gestionarea datelor şi altul la generarea informaţiei.

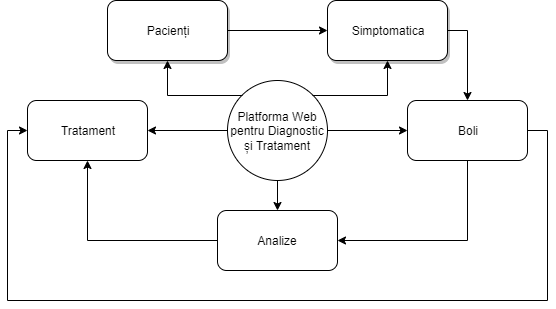
2.2 Descrierea domeniului de studii

La moment nu este o aplicație, destinată cetățenilor care doresc să-și facă un diagnostic sinestătător. Această întrebare este una actuală, deoarece mulți dintre cetățeni nu au posibilitatea să consulteze un medic, nu au surse necesare, au frică sau neîncredere în medicina de astăzi.

Una dintre căile de rezolvare ale acestor probleme este crearea unei aplicații web, cu ajutorul căruia fiecare utilizator să-și poată face un diagnostic independent fără a depune un efort. Unicul lucru care se așteaptă de la el să fie sincer, în primul rând cu sine însuși și să-și poată analiza corect starea organizmului său.

Toate datele introduse de utilizatori ar putea servi ca o sursă pentru crearea unei statistici referitor la starea sănătății cetățenilor, aceasta ar fi de folos în dezvoltarea medicinii, care utilizează din ce în ce mai mult domeniul IT și ar contribui la o schimbare a mentalității cetățenilor despre propria sănătate.

2.3 Schema legăturilor între subdomeniile domeniului de studiu

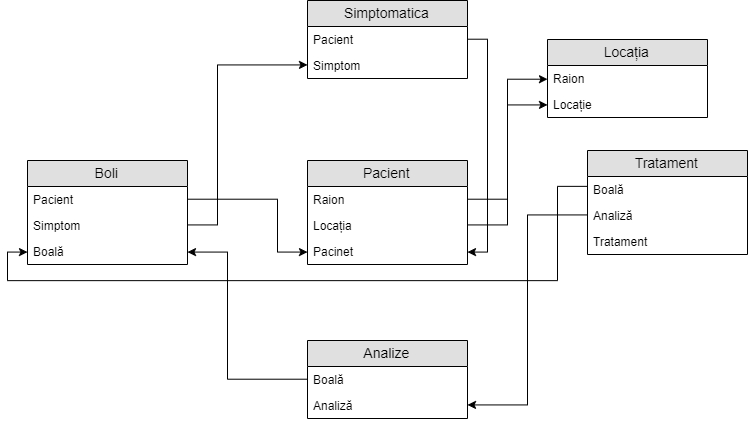


**Figura .** Schema legăturilor între subdomeniile DS.

Domeniul de studiu a sarcinii de creare a unui SGBD “Platformă Web pentru Diagnostic și Tratament” poate fi prezentat prin următoarele componente, care pot fi precăutate în calitate de subdomenii:

* **“Pacienți”** – acesta este subdomeniul principal al bazei de date și reprezintă o colecție de date despre pacienți (utilizatori);
* **“Simptomatica”** – subdomeniul dat reprezintă o colecție de date referitoare la simptomele introduce de către utilizatori;
* **“Boli”** – subdomeniul care reprezintă boala depistată pe baza la simptomele introduse;
* **“Analize”** – subdomeniul care reprezintă un șir de analize care sunt necesare pentru a alege tratamentul necesar;
* **“Tratament”** – subdomeniul care reprezintă tratamentul necesar pentru boala depistată.

2.4 Legături între obiectele Domeniului de Studiu



**Figura .** Legături între obiectele DS.

2.5 Caracteristicile informaționale ale SAPI “Platformă Web pentru Diagnostic și Tratament”

Ieșiri informaționale:

1. Lista pacienților

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cod | Nume | Prenume | IdLocația | Vârsta | Sex | Înălțime | Greutate | PatologiiCronice |

1. Lista simptomaticii:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdSimptom | Descriere | Cauză | DataApariției | Cod |

1. Lista bolilor:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdBoală | IdTipBoală | NumeBoală | IdSimptom | Cod |

1. Lista analizelor:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdAnaliză | IdTipAnaliză | Caracteristici | Valoare | IdBoală |

1. Lista tratamentelor:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdTratament | IdTipTratament | NumeTratament | IdBoală | IdAnaliză |

Interogări:

1. Să se determine care boală XXXX este cel mai des întâlnită în raionul YYYY:

|  |  |  |  |
| --- | --- | --- | --- |
| IdBoală | IdTipBoală | NumeBoală | IdLocație |

1. Să se determine vârsta XXXX cea mai des întalnită pentru tipul de boală YYYY:

|  |  |
| --- | --- |
| Vârsta | NumeBoală |

1. Să se determine genderul XXXX cel mai des întâlnit pentru tipul de boală YYYY:

|  |  |
| --- | --- |
| Sex | TipBoală |

Servicii:

1. Este dată boala XXXX, să se determine simptomatica YYYY:

|  |  |
| --- | --- |
| NumeBoală | NumeSimptom |

1. Este dată boala XXXX, să se determine analizele YYYY:

|  |  |  |
| --- | --- | --- |
| NumeBoală | IdAnaliză | Caracteristici |

1. Este dată boala XXXX, să se determine tratamentul YYYY:

|  |  |  |  |
| --- | --- | --- | --- |
| IdBoală | IdAnaliză | IdTipTratament | NumeTratament |

Restricții:

1. Cod, IdSimptom, IdBoală, IdAnaliză, Idtratament sunt unice și nu coincide, sunt elemente cheie.
2. Mai multe boli pot avea aceleași simptome.
3. O boală poate avea mai multe simptome.
4. O boală poate să necesite mai multe analize.
5. O boală poate să aibă mai multe tipuri de tratament.

2.6 Tabel de acoperire

Tabel . Tabel de acoperire

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Informația  Atribute | Ieșiri informaționale | | | | | Interogări | | | Servicii | | |
|  | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 1 | 2 | 3 |
| Cod | + | + |  |  |  |  |  |  |  |  |  |
| Nume | + |  |  |  |  |  |  |  |  |  |  |
| Prenume | + |  |  |  |  |  |  |  |  |  |  |
| IdLocația | + |  |  |  |  | + |  |  |  |  |  |
| Vârsta | + |  |  |  |  |  | + |  |  |  |  |
| Sex | + |  |  |  |  |  |  | + |  |  |  |
| Înălțime | + |  |  |  |  |  |  |  |  |  |  |
| Greutate | + |  |  |  |  |  |  |  |  |  |  |
| PatologiiCronice | + |  |  |  |  |  |  |  |  |  |  |
| IdSimptom |  | + | + |  |  |  |  |  | + |  |  |
| Descriere |  | + |  |  |  |  |  |  |  |  |  |
| Cauză |  | + |  |  |  |  |  |  |  |  |  |
| DataApariției |  | + |  |  |  |  |  |  |  |  |  |
| IdBoală |  |  | + | + | + | + |  |  | + | + | + |
| IdTipBoală |  |  | + |  |  | + |  | + |  |  |  |
| NumeBoală |  |  | + |  |  | + | + |  |  |  |  |
| IdAnaliză |  |  |  | + | + |  |  |  |  | + | + |
| IdTipAnaliză |  |  |  | + |  |  |  |  |  |  |  |
| Caracteristici |  |  |  | + |  |  |  |  |  | + |  |
| Valoare |  |  |  | + |  |  |  |  |  |  |  |
| IdTratament |  |  |  |  | + |  |  |  |  |  | + |
| IdTipTratament |  |  |  |  | + |  |  |  |  |  |  |
| NumeTratament |  |  |  |  | + |  |  |  |  |  |  |

2.7 Legături între atributele obictelor Domeniului de Studiu ER – schema BD

Pacienți

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cod | Nume | Prenume | IdLocație | Vârsta | Sex | Înălțime | Greutate | PatologiiCronice |

Locații

|  |  |
| --- | --- |
| IdLocație | NumeLocație |

Simptomatica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdSimptom | Descriere | Cauză | DataApariției | Cod |

Boli

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdBoală | IdTipBoală | NumeBoală | IdSimptom | Cod |

TipBoli

|  |  |
| --- | --- |
| IdTipBoală | NumeTipBoală |

Analize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdAnaliză | IdTipAnaliză | Caracteristici | Valoare | IdBoală |

TipAnalize

|  |  |
| --- | --- |
| IdTipAnaliză | NumeTipAnaliză |

Tratament

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdTratament | IdTipTratament | NumeTratament | IdBoală | IdAnaliză |

TipTratament

|  |  |
| --- | --- |
| IdTipTratament | NumeTipTratament |

2.8 Modelul Logic

Pacienți

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cod | Nume | Prenume | IdLocație | Varsta | Sex | Înălțime | Greutate | PatologiiCronice |

Locații

|  |  |
| --- | --- |
| IdLocație | NumeLocație |

Simptomatica

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdSimptom | Descriere | Cauză | DataApariției | Cod |

Boli

|  |  |  |  |
| --- | --- | --- | --- |
| IdBoală | IdTipBoală | NumeBoală | IdSimptom |

Analize

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdAnaliză | IdTipAnaliză | Caracteristici | Valoare | IdBoală |

Tratament

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IdTratament | IdTipTratament | NumeTratament | IdBoală | IdAnaliză |

TipBoli

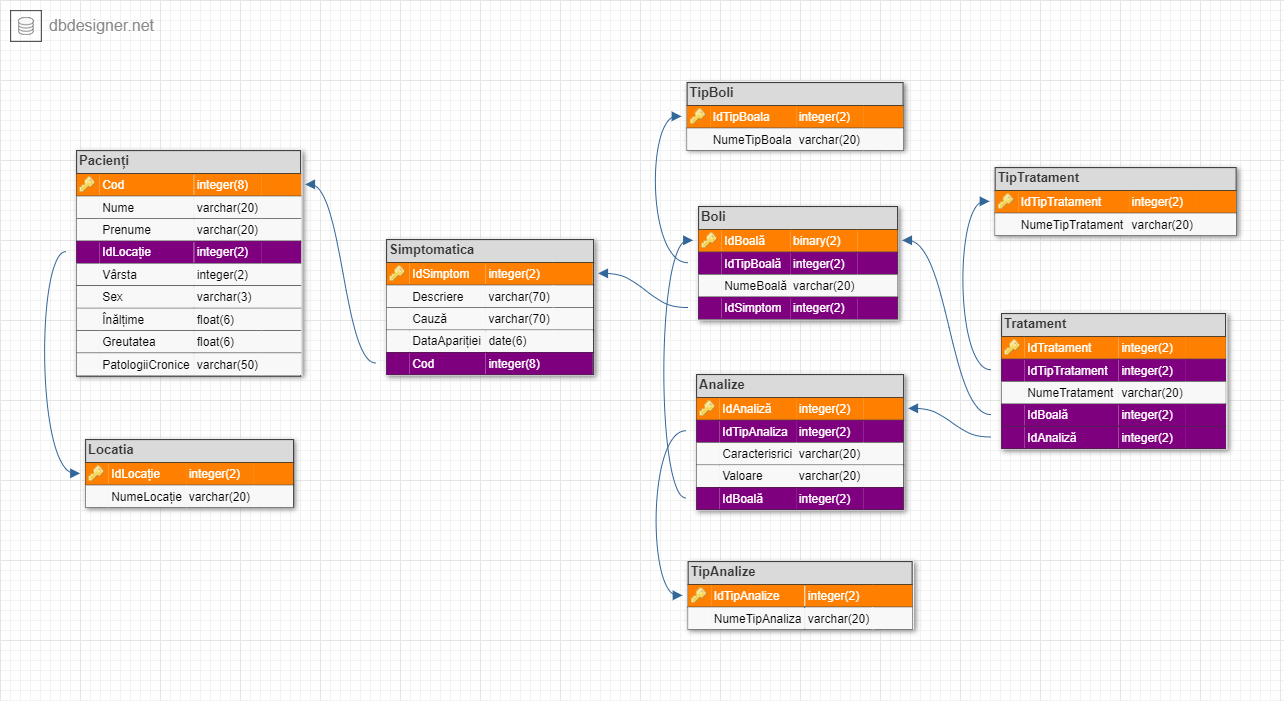
|  |  |
| --- | --- |
| IdTipBoală | NumeTipBoală |

TipAnalize

|  |  |
| --- | --- |
| IdTipAnaliză | NumeTipAnaliză |

TipTratament

|  |  |
| --- | --- |
| IdTipTratament | NumeTipTratament |



**Figura .** Schema ER a Modelului Logic.

2.9 Machetul bazei de date

Tabel . Machetul bazei de date

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel | Câmp | Tip(Lungime) | Comentarii |
| Locații | IdLocație | INT(2) | PRIMARY KEY, not null |
|  | NumeLocație | VARCHAR(20) |  |
| Pacienți | Cod | CHAR(8) | PRIMARY KEY, not null |
|  | Nume | VARCHAR(20) |  |
|  | Prenume | VARCHAR(20) |  |
|  | IdLocația | INT(2) | FOREIGN KEY |
|  | Vârsta | INT(2) |  |
|  | Sex | CHAR(1) |  |
|  | Înălțimea | INT(3) |  |
|  | Greutatea | INT(3) |  |
|  | PatologiiCronice | VARCHAR(50) |  |
| Simptomatica | IdSimptom | INT(2) | PRIMARY KEY, not null |
|  | Descriere | VARCHAR(70) |  |
|  | Cauză | VARCHAR(70) |  |
|  | DataApariției | DATE |  |
|  | Cod | CHAR(8) | FOREIGN KEY |
| TipBoli | IdTipBoală | INT(2) | PRIMARY KEY, not null |
|  | NumeTipBoală | VARCHAR(20) |  |
| Boli | IdBoală | INT(2) | PRIMARY KEY, not null |
|  | IdTipBoală | INT(2) | FOREIGN KEY |
|  | NumeBoală | VARCHAR(20) |  |
|  | IdSimptom | INT(2) | FOREIGN KEY |
| TipAnalize | IdTipAnaliză | INT(2) | PRIMARY KEY, not null |
|  | NumeTipAnaliză | VARCHAR(20) |  |
| Analize | IdAnaliză | INT(2) | PRIMARY KEY, not null |
|  | IdTipAnaliza | INT(2) | FOREIGN KEY |
|  | Caracteristici | VARCHAR(20) |  |
|  | Valoare | VARCHAR(20) |  |
|  | IdBoală | INT(2) | FOREIGN KEY |
| TipTratament | IdTipTratament | INT(2) | PRIMARY KEY, not null |
|  | NumeTipTratament | VARCHAR(20) |  |
| Tratament | IdTratament | INT(2) | PRIMARY KEY, not null |
|  | IdTipTratament | INT(2) | FOREIGN KEY |
|  | NumeTratament | VARCHAR(50) |  |
|  | IdBoală | INT(2) | FOREIGN KEY |
|  | IdAnaliză | INT(2) | FOREIGN KEY |

### Introducere Programare WEB

3.1 HTML prezentare

**HyperText Markup Language (HTML)** este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser (navigator). Scopul HTML este mai degrabă prezentarea informațiilor – paragrafe, fonturi, tabele ș.a.m.d. – decât descrierea semanticii documentului.

Specificațiile HTML sunt dictate de World Wide Web Consortium (W3C).

HTML este un limbaj de marcare din familia SGML.

SGML - Standard Generalized Markup Language este un ISO-standard pentru definirea limbajeor de marcare a documentelor.

**Documentele HTML** sunt fişerele textuale şi se salveză în fişiere cu extensia .html sau .htm

Limbajul de marcare HTML nu utilizeaza instructiuni, ci etichete, acestea fiind numite tag-uri.

Structura unui document HTML

<html>

<head>

<title>

Titlul paginii

</title>

</head>

<body>

Continutul paginii

</body>

</html>

Documentele HTML pot fi create cu un editor de texte sau cu editoare HTML care permit crearea vizuală (WYSIWYG) rezultînd însă tot documente în format text-pur. Limbajul HTML dă autorilor posibilitatea :

Să publice documente cu headere, texte, tabele, liste, fotografii etc;

* Să regasească on-line informații prin intermediul hiperlink-urilor accesate printr-un simplu click de mouse;
* Să proiecteze formulare pentru realizarea tranzacțiilor cu servere aflate la distanță pentru căutari de informație sau pentru activități specifice comerțului;
* Să includă foi de calcul tabelar, clipuri video, sunete și alte aplicații direct în documente.

Limbajul HTML are patru caracteristici principale:

1. Folosește un marcaj descriptiv pentru a indica diversele acțiuni ce trebuie executate. Aceasta înseamnă că părți ale documentului descris de HTML sunt marcate cu nume descriptive, ca de exemplu <CHAPTER> sau <TITLE> care sunt aplicabile oricărei porțiuni de date corespunzătoare din document.
2. Definește structuri de documente ierarhice și legături întra- și inter-documente. O legătură este o relație unară între două elemente ale unui document. Structura unui document este însoțită de astfel de legături între elementele sale.
3. Limbajul HTML este guvernat de o descriere formală. HTML are o descriere a tipului documentului (Document Type Definition DTD) care stabilește specificațiile formale ale limbajului. DTD stabileste sintaxa limbajului, descrie fiecare element individual al unui document scris în limbajul HTML, definește atributele permise pentru fiecare element și descrie modelul datelor conținute în fiecare element. În corelație cu informațiile despre elemente, DTD oferă definiții pentru entitățile externe ce pot fi referite în HTML.
4. Atât specificațiile limbajului cât și limbajul însuși pot fi citite și de om dar și de computer. Datorită faptului că elementele de marcare sunt separate de text prin șiruri de delimitare alcătuite din caractere tiparibile, textul și marcajele pot coexista.

HTML este un limbaj bazat pe SGML (Standard Generalized Markup Language). SGML este un standard internațional (ISO-8879) aprobat în 1986.

SGML oferă o modalitate de a reprezenta structura documentelor și hyper-documentelor. Este totodată și o cale de a codifica hyper-documentele astfel ca acesta să poată fi interschimbate asemanator procesului de interschimbare a unor documente în cazul mai multor autori care colaborează utilizând platforme diferite aflate la distanță.

SGML este un sistem complex de descriere a documentelor. Este utilizat pentru descrierea structurii generale a diferitelor tipuri de documente fără să fie un limbaj de descriere a paginii. Principala preocupare a SGML se răsfrânge asupra conținutului documentului, nu asupra aspectului sau.

Deci, în virtutea trăsăturilor moștenite de la SGML, HTML este un limbaj pentru descrierea documentelor structurate. Teoria din spatele acestui limbaj se bazează pe faptul că majoritatea documentelor au elemente comune și că dacă definești un set de elemente, poți marca elementele documentului cu etichetele corespunzătoare.

Majoritatea etichetelor HTML arată sub forma:

<NUME ETICHET{> Textul afectat de eticheta </NUME ETICHET{> și indica navigatorului elemente de structură a documentului, formatare, hypertext sau alte elemente. Documentele HTML conțin doar textul propriu-zis și etichetele HTML iar sursa lor poate fi ușor văzută din orice navigator.

3.2 Limbajul de scriptare

Baza comunicării între servere și clienții Web, la nivel de aplicație este asigurată de protocolul HTTP (HyperText Transfer Protocol). Protocolul HTTP este folosit în special pentru hipertext, dar este un protocol generic, putând susține un sistem distribuit pentru managementul obiectelor de date. O caracteristică importantă a protocolului este independența de platformă, diferitele calculatoare care comunică prin HTTP putând folosi diverse sisteme de operare și aplicații hipertext.

Principalele concepte cu care lucrează protocolul sunt cererea și raspunsul: un client Web trimite un mesaj(cererea) la un server. Mesajul conține identificatorul resursei dorite, dat sub forma unui URI (Uniform Resource Identifier), metoda de acces folosită, precum și o serie de meta-informații care pot fi utile serverului. Răspunsul serverului cuprinde un cod indicând starea serverului după interpretarea cererii, un mesaj explicativ pentru codul de stare transmis, meta-informațiile care vor fi procesate de către client și, eventual, un conținut(resursa solicitată).

Standard pentru interacțiunea clienților Web cu serverele Web, Common Gateway Interface se află în prezent la versiunea 1.1. Un program CGI, denumit în mod uzual script, se execută pe serverul WWW fie în mod explicit, apelat din cadrul paginii printr-o directivă specială, fie la preluarea informațiilor aflate în cadrul câmpurilor unui formular interactiv sau coordonatelor unei zone senzitive. CGI conferă interactivitate paginilor Web, documentele HTML putând să-și modifice în mod dinamic conținutul și să permită prelucrări sofisticate de date. Programele CGI pot oferi suport și la autentificarea utilizatorilor pe partea de server.

Programele CGI pot fi scrise în orice limbaj, fiind interpretate în cazul limbajelor PHP, Perl, Python, TCL sau compilate cum se întimplă pentru limbajul C și Delphi.

În conceperea unui script CGI trebuie respectate anumite reguli de bază:

* programul scrie datele spre a fi transmise navigatorului Web la ieșirea standard
* programul generează anteturi care permit serverului Web să interpreteze corect ieșirea scriptului

Cele mai multe scripturi CGI sunt concepute pentru a procesa datele introduse în formulare. Un formular se definește în HTML folosindu-se marcatori specifici pentru afisarea conținutului și introducerea datelor către clienți, iar scriptul invocat și executat de serverul de Web, va prelua conținutul acelui formular și-l va prelucra, returnând eventual rezultatele către navigator.

3.3 PHP reprezentare

PHP (PHP: Hypertext Prepocessor), cunoscut în versiunile mai vechi şi sub numele de PHP/FI (Personal Homepage/Form Interpreter), iniţial a fost gândit a fi o simplă aplicaţie CGI pentru interpretarea formularelor definte prin HTML şi procesate de un program scris într-un limbaj Perl, script shell, executat pe server. În cazul interfeţei CGI era necesară permisiunea de a rula programe pe server, ceea ce ducea la lacune în securitate şi în plus la disocierea de documentul HTML a programului care procesa datele.

PHP (în versiunea curentă PHP 5.0) reprezintă un pachet puternic care oferă un limbaj de programare accesibil din cadrul fişierelor HTML, limbaj asemănător cu Perl sau C, plus suport pentru manipularea bazelor de date într-un dialect SQL (dBase, Informix, MySQL, mSQL, Oracle, PostgreSQL, Solid, Sybase, ODBC etc.) şi acces la sisteme hipermedia precum Hyperwave. De asemeni, PHP suportă incărcarea fişierelor de pe calculatorul client: upload (standard propus de E. Nebel şi L. Masinter de la Xerox, descris în RFC 1867) şi oferă suport pentru cookies (mecanism de stocare a datelor în navigatorul client pentru identificarea utilizatorilor, propus de Netscape). Această aplicaţie este disponibilă gratuit pe Internet, pentru medii Unix şi mai nou pentru medii Windows (inclusiv sursele), integriându-se în popularul server Apache.

Se estimează că numărul site-urilor care folosesc în prezent PHP este de peste un milion. Deja, pe Web, exista o multitudine de aplicaţii şi utilitare concepute în PHP, care se regăsesc grupate şi în aşa-numitul PEAR (PHP Extension and Add-on Repository).

PHP este un limbaj de script care funcţionează alături de un server Web.

Este de notat diferenţa cu alte limbaje script CGI scrise în limbaje precum Perl sau C: în loc de a scrie un program cu o multitudine de linii de comanda afişate în final într-o pagină HTML, veţi scrie o pagină HTML cu codul inclus pentru a realiza o acţiune precisă.

Codul PHP este inclus între tag-urile speciale de început şi de sfârşit care permit utilizatorului să treacă din "modul HTML" în "modul PHP". Faţă de alte limbaje script, precum Javascript, la PHP codul se execută pe serverul Web, nu in navigator(client Web). In consecinta, PHP poate obtine accesul la fisiere , baze de date si alte resurse inaccesibile programului JavaScript. Acestea contin bogate surse de continut dinamic care atrag vizitatorii. Dacă pe serverul Web se află un script similar, clientul nu va primi decât rezultatul execuţiei scriptului, fără a avea nici o posibilitate de acces la codul care a produs rezultatul. Vă puteţi configura serverul de Web să prelucreze (analizeze) toate fişierele HTML ca fişiere PHP. Astfel nu există nici un mijloc de a distinge paginile care sunt produse dinamic de paginile statice.

Limbajul PHP posedă aceleaşi funcţii ca alte limbaje permiţâind să se scrie scripturi CGI, să colecteze date şi să genereze dinamic pagini web sau să trimită şi să primească cookies. Marea calitate şi cel mai mare avantaj al limbajului PHP este suportul cu un număr mare de baze de date. A realiza o pagină web dinamic cu o baza de date este extrem de simplu. PHP suporta foarte multe tipuri de baze de date, ceea ce este un avantaj foarte mare pentru el. Limbajul PHP are deasemenea suport pentru diverse servicii server utilizând protocoale precum IMAP, SNMP, NNTP, POP3 şi HTTP. Fiecare program PHP include două linii speciale, care indică serverului PHP că textul cuprins între cele doua linii este alcătuit din instrucțiuni PHP.

<?php ?>

Practic aceste linii pot fi asimilate copertelor unei cărți, care păstrează unitatea programului PHP. Programele PHP execută trei categorii de operații elementare:

1. obțin date de la un utilizator;
2. execută prelucrări ale datelor, respectiv obțin accesul la datele stocate în fișiere și baze de date și le manipulează;
3. afisează date astfel încât un utilizator să le poată vizualiza.

Pentru executarea unui script trebuie mai întâi încărcat scriptul într-un server unde este instalat PHP. Accesul la distanță la un server Linux sau Unix se poate face prin intermediul protocoalelor Telnet sau SSH, dacă în prealabil se primește autorizare de folosire a acestor protocoale de la administatorul de sistem pentru un anumit utilizator. Dacă se folosește un server Microsoft Windows situat în aceeași rețea locală ca și stația de lucru atunci este posibilă alocarea de către adimistratorul de rețea a unei partiții de fișiere în acest scop. În vederea încărcării scriptului prin intermediul protocolului FTP adimistratorul de sistem trebuie să furnizeze utilizatorilor autorizați următoarele informații:

* numele serverului gazda;
* numele de utilizator și parola pentru deschiderea sesiunii de lucru prin intermediul protocolului FTP;
* catalogul în care trebuie să se găsească scripturile PHP;
* localizatorul uniform de resurse(URL), pe care trebuie să-l folosiți pentru a obține acces la scripturile PHP.

După ce încărcarea scriptului PHP a fost realizată cu succes acesta se lansează în execuție prin dechiderea unui navigator Web(client) cu adresa URL asociată scriptului PHP.

3.4 Serverul Web Apache

Serverul de Web Apache stă la baza a peste 60% din domenii, potrivit unui studiu Netcraft Web Server. Providerii de Internet și companiile de găzduire de aplicații pe Web apelează deseori la Apache din cauza suportului sau pentru numeroase platforme, capabilităților de găzduire virtuală ușor de implementat și modulelor sale care îi extind capabilitățile.

Caracteristica Apache Portable Routine(APR) optimizează capabilitățile precum administrarea proceselor pentru fiecare sistem de operare, dar permite serverului propriu-zis să ignore distincțiile specifice fiecărei platforme. Acest lucru a îmbunătățit performanța și stabilitatea implementărilor Windows, prin eliminarea emulatorului. Mai mult APR este accesibil și dezvoltatorilor Web care scriu programe multi-platforme în C, care este mai rapid decît limbajele interpretative, precum PHP și Perl, folosite de obicei pentru dezvoltare Web.

La acest lucru se adaugă faptul că serverul propriu-zis este independent de protocoale. Cu toate că este în principal un server HTTP(Web), este proiectat să suporte alternative cum ar fi FTP. O astfel de structură simplifică administrarea și reduce riscurile de securitate. O parte însemnată din atractivitatea Apache o constituie versabilitatea sa. API-ul său deschis a permis dezvoltatorilor să scrie diferite module care i-au schimbat comportamentul. Dacă serverului îi lipsește o funcție de care are nevoie un site, sunt mari șanse ca undeva să fie disponibil un modul plug-in.

Serverele de Web au ca funcționalitate de baza recepționarea de cereri anonime de la clienți și furnizarea de informații într-o maniera dorită a fi eficientă și rapidă. De fapt un server Web este un daemon care accepta conexiuni conforme protocolului HTTP, răspunzând cererilor recepționate de la clienți. Pentru a asigura servicii HTTP, serverul Apache trebuie să fie instalat în sistem(în mod uzual, fiind vorba de un pachet RPM în Linux sau de un program executabil .exe în Windows), iar daemon-ul httpd pornit. Apache este un sistem modular, alcătuit dintr-un server de baza și mai multe module care sunt încărcate dinamic într-un mod similar cu funcționarea modulelor din nucleul Linux.

Apache poate fi configurat cu ajutorul interfeței grafice apache-conf(Apache Configuration Tool). Fișierul de configurare principal este http-conf și este de obicei localizat în directorul /etc/httpd (în versiunile de Linux sau Unix).

În anumite cazuri, este necesar să se restrictioneze accesul la anumite documente, prin intermediul autentificării prin nume de utilizator și parolă sau în funcție de adresa calculatorului clientului Web.

Pentru autentificarea utilizatorilor, vom parcurge doi pași:

1. se creeaza un fișier conținând numele și parolele utilizatorilor care vor avea acces la anumite date de pe serverul Web (în particular Apache);
2. se configurează serverul pentru a seta care resurse vor fi protejate și care sunt utilizatorii având permisiunea accesării lor, după introducerea unei parole valide.

Configurarea serverului se poate realiza fie prin fișierul httpd-conf, fie prin .htacces, indicând o zona protejată, de obicei în funcție de directoarele dorite a fi accesate pe bază de autentificare. Fișierul .htaccess va fi stocat în directorul asupra căruia dorim să modificăm comportamentul implicit al serverului Web. Înainte de a modifica maniera de autentificare din fișierul .htaccess, administratorul serverului Apache va specifica în httpd.conf că autentificările să se realizeze via .htaccess.

De asemenea, Apache oferă posibilitatea de a servi mai multe situri Web simultan, altfel spus, găzduire virtuală (virtual hosting). Există două metode de implementare a găzduirii virtuale: prima bazată pe nume și a doua bazată pe adrese IP. Mașinile virtuale bazate pe adresa utilizează adresa IP a conexiunii pentru a determina mașina virtuală corectă. Astfel pentru fiecare găzduire virtuală bazată pe nume, determinarea mașinii virtuale se face pe baza numelui acestuia.

Găzduirea virtuala bazată pe nume este mai simplu de implementat, și este recomandată utilizarea acesteia. Pentru a utiliza serviciul de găzduire virtuala, trebuie mai întâi stabilite adresa IP și portul pentru serverul care va accepta cereri pentru respectiva mașina virtuală.

3.5 Administrarea Bazelor de Date cu MySQl

Dezvoltat de firma suedeză MySQL AB, MySQL este un server de baze de date disponibil gratuit, cu sursa deschisă(open-source) care oferă fiabilitate și avantaje reale. Frumusețea serverului MySQL este simplitatea sa fundamentală. Administrarea bazei de date se face folosind peste o duzină de utilitare în linia de comandă, cel mai important fiind mysql, un shell interactiv pentru controlul și interogarea bazei de date. Utilitarele rulează cel mai bine pe sistemul Linux, platforma pe care MySQL a fost dezvoltat inițial. Alte două utilitare cu sursa deschisă, oferite pe platforma Windows, care oferă un set de comenzi de administrare sunt MySql Manager, un utilitar de interogare în mod grafic similar cu mysql și WinMySQL admin, o consolă pentru administrarea detaliilor configurării lui MySQL.MySQL realizează cu ușurință importarea diverselor baze de date și exportarea sub forma fisierelor sql. Tranzacțiile nu sunt parte a tabelelor implicite(ISAM) ale lui MySQL, dar sunt incluse două tipuri noi de tabele. Berkley DB(BDB) și InnoDB, care au fost dezvoltate de alte firme. Administratorii de firme și dezvoltatorii de baze de date au ajuns la concluzia că MySQL are o dezvoltare energică și loială, ca să nu mai menționăm reputația câștigată cu trudă, de baza de date sigură.

MySQL operează în baza unui model client/server. Orice mașina care dorește să proceseze interogări asupra unei baze de date MySQL trebuie să ruleze MySQL server(mysqld), care este responsabil de tot traficul de tip intrari/iesiri(incoming/outgoing) cu baza de date. Ca orice server, mysqld "ascultă" pe un port particular (3306) eventualele cereri de conexiune ale unui "client" - orice aplicație care trimite cereri către o bază de date via mysqld. Acest client poate fi un simplu script PHP care poate trimite o cerere către baza de date prin intermediul serverului MySQL sau chiar clientului mysql. Clientul mysql este o interfață interactiva pentru transmiterea de comenzi către server.

Modelul de securitate folosit de MySQL se bazează pe nume de utilizator, parola, nume server(hostname) sau adresa de IP și privilegii, fiind similar celui generic folosit de sistemele Unix. Prin privilegii se înteleg în cazul MySQL operațiunile ce vor fi permise asupra bazei/bazelor de date, tabelelor sau indecsilor, cum sunt de exemplu SELECT, INSERT, UPDATE, DELETE, CREATE, DROP.

Datele sunt obiectul celor mai multe operații de prelucrare, iar sistemele de gestiune a bazelor de date furnizează cele mai complexe și puternice facilități pentru lucrul cu datele. PHP include o biblioteca de functii care furnizeaza o interfață cu sistemul MySQL de gestiune a bazelor de date. Folosind aceste funcții, un program PHP poate obține accesul la datele rezidente într-o bază de date MySQL și le poate modifica.

Majoritatea interacțiunilor cu o bază de date MySQL se poate desfășura după un model secvențial simplu:

* se deschide o conexiune cu serverul MySQL;
* se specifica baza de date la care se obține accesul;
* se emit interogări SQL, se obține accesul la rezultatele interogării și se execută operații non-SQL
* se închide conexiunea cu serverul MySQL

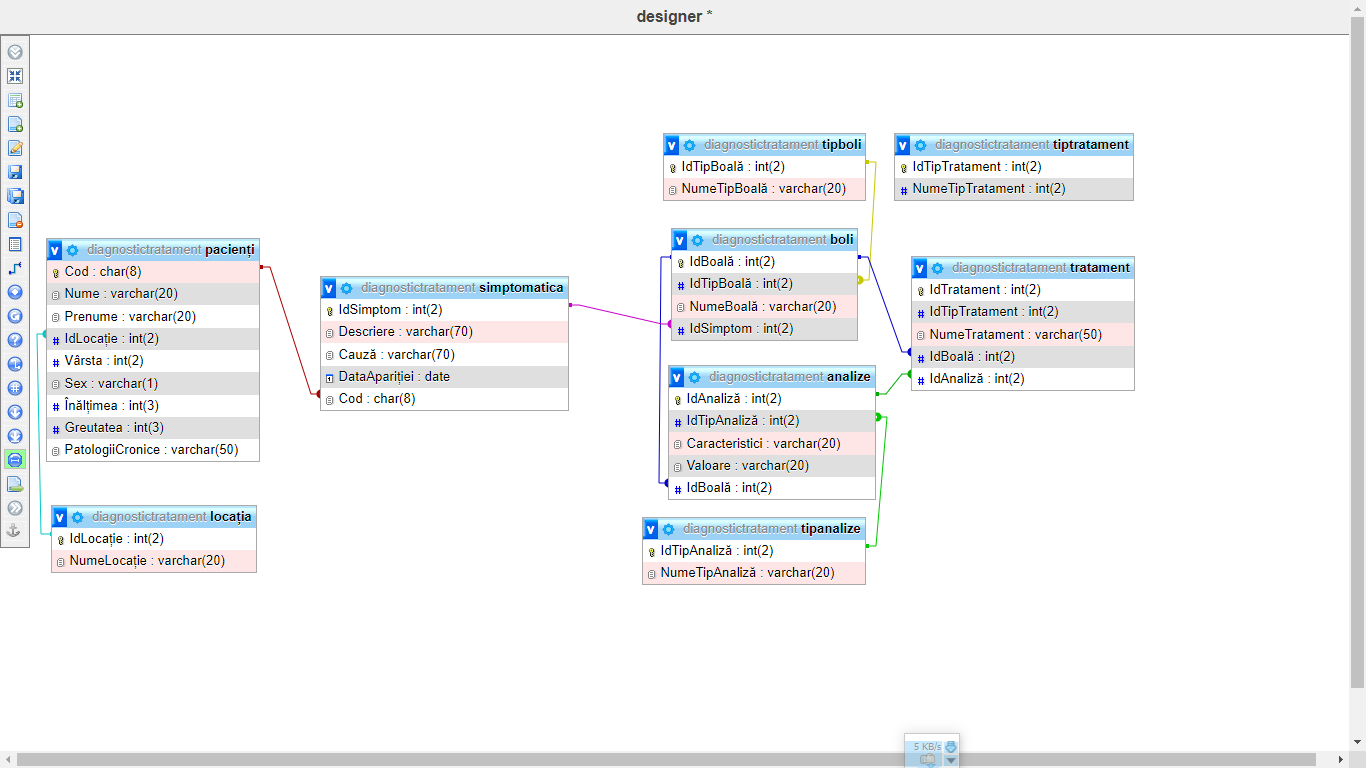
Din punct de vedere al limbajului PHP, există două categorii de interogări SQL:

1. interogările SELECT, care returnează rânduri ale unui tabel,
2. interogările UPDATE, INSERT și DELETE, care nu returnează rânduri ale unui tabel.

Ambele categorii de interogari sunt emise folosind funcția mysql\_query(), dar verificarea și prelucrarea celor două categorii de rezultate ale interogărilor sunt procesate destul de diferit.

### Structura Bazei de date

4.1 Designer-ul Bazei de Date



**Figura .** Designer-ul Bazei de Date