

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică**

**și Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Lucrare de an**

***Sistem informatic privind vânzările de semifabricate***

**Student: Calancea Catalin, gr. MI-222**

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

**Chișinău 2023**

Cuprins

[Introducere în SGBD 3](#_Toc153123752)

[**1.1** **Conceptul de bază de date** 4](#_Toc153123753)

[**1.2** **Gestiunea Bazelor de Date** 4](#_Toc153123759)

[Modelul Conceptual. Modelul Logic. 6](#_Toc153123760)

[**2.1 Scopul și obiectivele lucrării** 6](#_Toc153123761)

[2.2 Descrierea domeniului de studii 7](#_Toc153123762)

[2.3 Schema legăturilor între subdomeniile domeniului de studiu 8](#_Toc153123763)

[2.4 Legături între obiectele Domeniului de Studiu 9](#_Toc153123764)

[2.5 Caracteristicile informaționale ale “ Platformă Web pentru Gestionarea Vânzărilor de Semifabricate” 10](#_Toc153123765)

[2.6 Tabel de acoperire 12](#_Toc153123766)

### Introducere în SGBD

* 1. **Conceptul de bază de date**

Conceptul de bază de date se referă la un set integrat de date structurate, însoțit de o descriere a structurii acestora denumită dicționar de date sau metadate. Aceasta creează o interdependență între date și programe. Datele dintr-o bază de date pot fi organizate pe trei niveluri, în funcție de categoria de personal implicată:

1. Nivelul conceptual (global): Reflectă viziunea administratorului bazei de date asupra datelor și corespunde structurii conceptuale (schema) a bazei de date. Aceasta oferă o descriere a tuturor datelor într-un mod independent de aplicații, facilitând administrarea acestora.

2. Nivelul logic: Reflectă viziunea programatorului de aplicație asupra datelor și furnizează o descriere a datelor corespunzătoare unui anumit program de aplicație.

3. Nivelul fizic: Reflectă viziunea inginerului de sistem asupra datelor și corespunde schemei interne a bazei de date, oferind o descriere a datelor pe suport fizic de memorie.

Există mai multe tipuri de baze de date în funcție de modul de organizare și modul de dispunere pe suport magnetic a informației și a elementelor componente, inclusiv modele primitive, baze de date ierarhice, baze de date în rețea, baze de date relaționale și baze de date distribuite. Baza de date relațională, introdusă de dr. E. F. Codd în 1969, reprezintă o mulțime structurată de date, accesibile prin calculator, care modelează un sistem sau un proces din lumea reală și servește ca suport pentru aplicații informatice. Bazele de date distribuite integrează tehnologia bazelor de date cu cea a rețelelor de calculatoare, fiind logic integrate, dar fizic distribuite pe mai multe sisteme de calcul. Integrarea acestora se realizează prin schemele globale, de fragmentare și de alocare.

* 1. **Gestiunea Bazelor de Date**

Sistemele de gestionare a bazelor de date (SGBD) sunt platforme informatice specializate în stocarea și procesarea eficientă a unui volum mare de date, având un număr relativ redus de operațiuni de prelucrare. Termenul "bază de date" se referă la datele destinate prelucrării și modul în care acestea sunt organizate pe suportul fizic de memorie, în timp ce termenul "gestiune" se referă la operațiile de inserare, ștergere și interogare/extragere a datelor din baza de date.

Pe lângă aceste funcționalități esențiale, un SGBD furnizează și alte servicii, cum ar fi suportul pentru limbaje de programare, interfețe atrăgătoare pentru comunicarea cu utilizatorii și tehnici avansate de stocare a datelor, acoperind astfel totalitatea operațiunilor care se aplică asupra datelor din baza de date.

Principalele funcții pe care un SGBD trebuie să le asigure includ:

**1. Definirea**: Crearea structurii bazei de date prin intermediul unui limbaj specific de definire a datelor (LDD), conform unui model de date anume.

**2. Introducerea datelor**: Adăugarea de informații noi în baza de date prin comenzi în limbajul de manipulare a datelor (LMD).

**3. Modificarea și accesarea datelor**: Actualizarea, organizarea și accesarea datelor deja existente în baza de date, inclusiv utilizarea utilitarelor încorporate pentru gestionarea fișierelor, listelor și tabelelor.

**4. Ajutor pentru utilizator**: Furnizarea de asistență (help) pentru utilizatori în utilizarea eficientă a bazei de date.

Activitățile principale pe care un SGBD trebuie să le desfășoare includ:

**1. Definirea și descrierea structurii bazei de date**: Realizată prin limbajul propriu de definire a datelor (LDD) pentru a stabili atributele, legăturile și criteriile de validare ale datelor.

**2. Încărcarea datelor**: Utilizarea comenzilor în limbajul de manipulare a datelor (LMD) pentru a încărca informațiile în baza de date.

**3. Accesul la date**: Utilizarea unor comenzi specifice din limbajul de manipulare a datelor pentru interogarea și actualizarea datelor, incluzând vizualizarea, consultarea și editarea de rapoarte și liste.

**4. Întreținerea bazei de date**: Utilizarea utilitarelor interne ale SGBD pentru menținerea eficientă a bazei de date.

**5. Reorganizarea bazei de date**: Actualizarea structurii datelor și modificarea strategiilor de acces, efectuate de către administratorul bazei de date.

**6. Securitatea datelor**: Asigurarea confidențialității datelor prin autorizare și controlul accesului, inclusiv criptarea datelor.

Se subliniază următoarele funcții ale unui SGBD:

**1. Funcția de descriere a datelor**: Realizată prin LDD, aceasta definește atributele, legăturile și criteriile de validare, rezultând schema bazei de date.

**2. Funcția de manipulare**: Complexă, realizează actualizarea și regăsirea datelor în cadrul bazei de date.

**3. Funcția de utilizare**: Furnizează interfețe necesare pentru comunicarea eficientă a utilizatorilor cu baza de date.

### Modelul Conceptual. Modelul Logic.

**2.1 Scopul și obiectivele lucrării**

**Scopul**: Dezvoltarea unui sistem informatic privind vânzările de semifabricate, facilitând gestionarea și analiza eficientă a datelor legate de vânzări.

**Obiectivul nr. 1: Analiza performanțelor de vânzare**

- Colectarea și analiza datelor referitoare la vânzările de semifabricate pentru a identifica modelele de vânzare, tendințele și fluctuațiile de cerere.

- Generarea rapoartelor privind performanța vânzărilor, inclusiv evaluarea cifrei de afaceri, a volumului de vânzări și a profitului.

**Obiectivul nr. 2: Prognozarea cererii**

- Utilizarea datelor istorice pentru a dezvolta modele de prognozare a cererii pentru diferite tipuri de semifabricate.

- Implementarea unui sistem de previziune a cererii pentru a ajuta la gestionarea stocurilor și la anticiparea nevoilor de producție.

**Obiectivul nr. 3: Gestionarea inventarului**

- Monitorizarea și actualizarea în timp real a stocurilor de semifabricate disponibile în depozit.

- Generarea automată a alertelor pentru reumplerea stocurilor atunci când acestea ating niveluri critice.

**Obiectivul nr. 4: Optimizarea prețurilor**

- Analiza datelor privind prețurile de vânzare pentru a identifica strategii de preț eficiente și competitive.

- Implementarea unui sistem de ajustare a prețurilor în funcție de cerere, ofertă și condițiile pieței.

**Obiectivul nr. 5: Eficientizarea proceselor operaționale**

- Dezvoltarea unei interfețe intuitive pentru utilizatorii sistemului, facilitând introducerea și accesul la date.

- Automatizarea proceselor repetitive, cum ar fi generarea de facturi și rapoarte periodice.

Aplicația va fi destinată angajaților responsabili cu gestionarea vânzărilor și a stocurilor, administratorilor de baze de date implicați în monitorizarea și întreținerea sistemului, precum și oricărui membru al organizației implicat în procesul de vânzare a semifabricatelor.

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

Scopul sistemului informatic este să eficientizeze și să gestioneze procesul de vânzare a semifabricatelor într-un mod optim, oferindu-le utilizatorilor instrumente pentru monitorizarea, analiza și îmbunătățirea performanțelor în acest domeniu. Aplicația este dezvoltată pentru a sprijini atât personalul responsabil cu vânzările și stocurile, cât și administratorii de baze de date implicați în gestionarea sistemului.

1. **Analiza Performanțelor de Vânzare:**

* Colectarea și analiza datelor referitoare la vânzările de semifabricate pentru a identifica modelele, tendințele și fluctuațiile de cerere.
* Generarea de rapoarte privind cifra de afaceri, volumul de vânzări și profitabilitatea.

1. **Procesul de Prognozare a Cererii:**

* Utilizarea datelor istorice pentru a dezvolta modele de prognozare a cererii pentru diferite tipuri de semifabricate.
* Implementarea unui sistem de previziune pentru gestionarea eficientă a stocurilor și anticiparea nevoilor de producție.

1. **Gestionarea Inventarului:**

* Monitorizarea în timp real și actualizarea stocurilor de semifabricate disponibile în depozit.
* Generarea automată a alertelor pentru reumplerea stocurilor la atingerea nivelurilor critice.

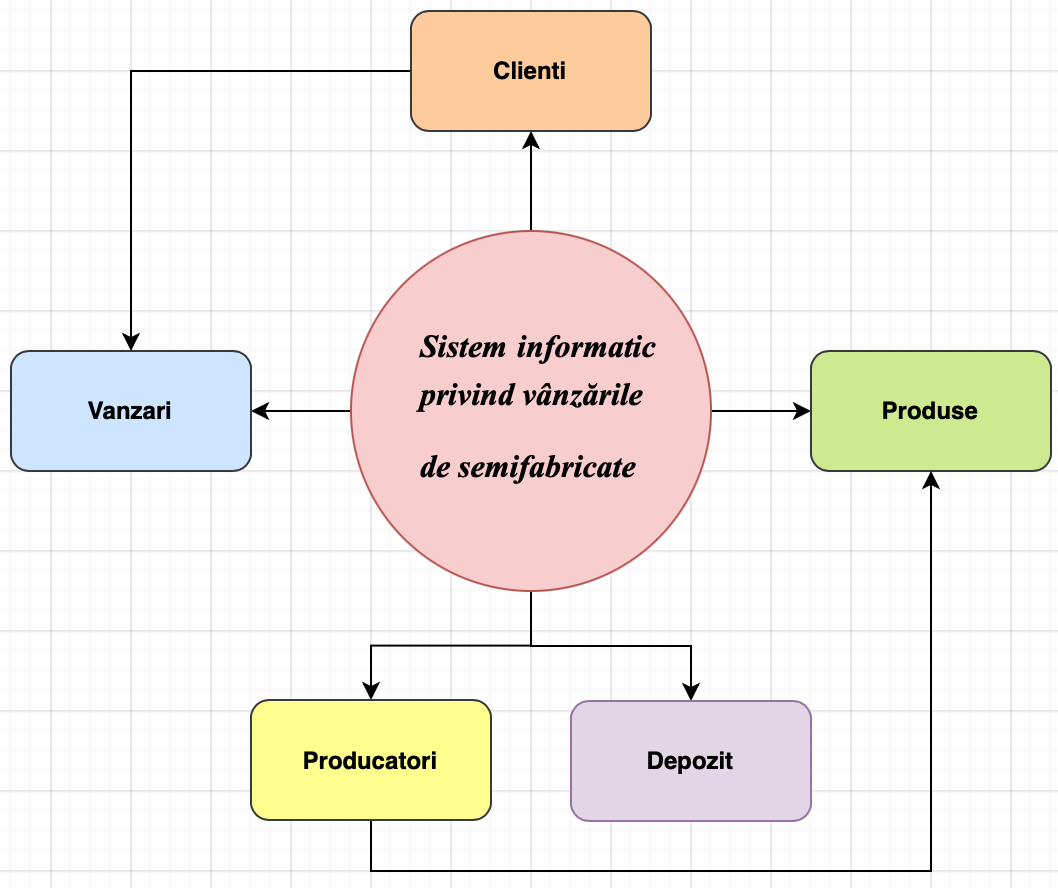
1. **Optimizarea Prețurilor:**

* Analiza datelor privind prețurile de vânzare pentru identificarea strategiilor eficiente și competitive.
* Implementarea unui sistem de ajustare a prețurilor în funcție de cerere, ofertă și condițiile pieței.

1. **Eficientizarea Proceselor Operaționale:**

* Dezvoltarea unei interfețe intuitive pentru utilizatori, facilitând introducerea și accesul la date.
* Automatizarea proceselor repetitive, precum generarea de facturi și rapoarte periodice.

2.3 Schema legăturilor între subdomeniile domeniului de studiu

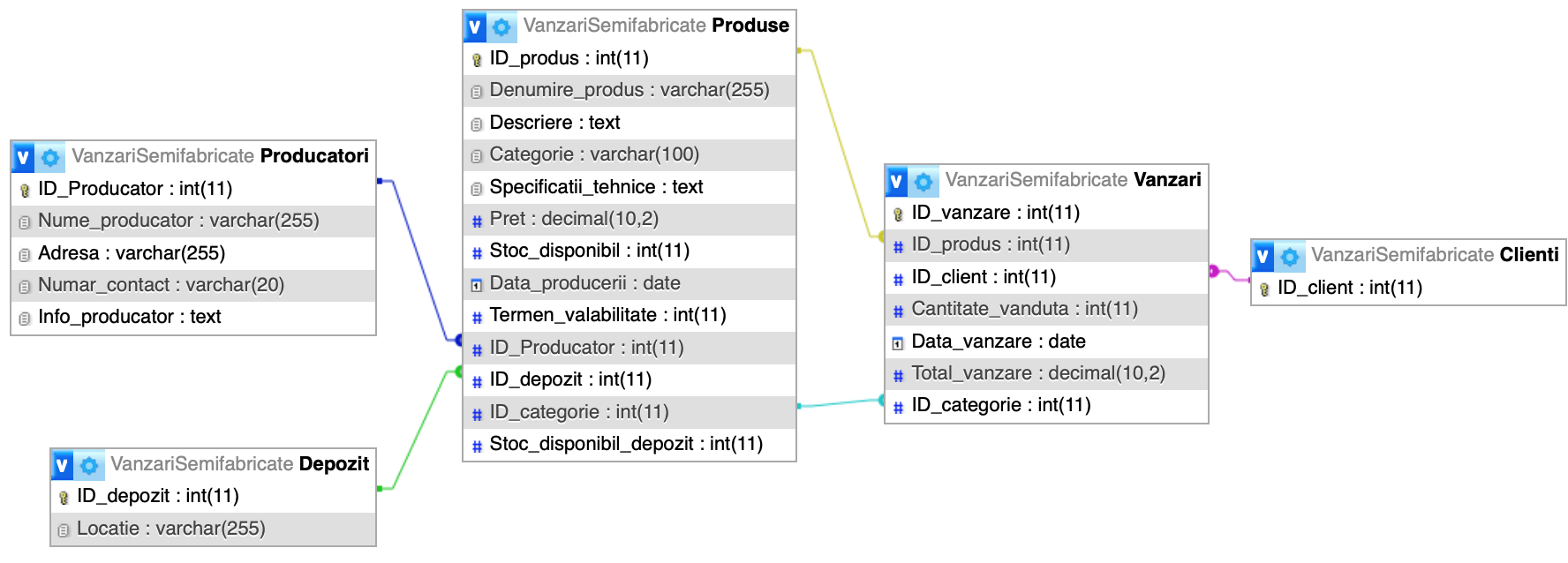


**Figursa 2.3.1 Legaturile intre subdomenii.**

Domeniul de studiu pentru crearea unui SGBD "Platformă Web pentru Gestionarea Vânzărilor de Semifabricate" poate fi structurat în următoarele subdomenii:

* **“Clientii”** – Subdomeniul al bazei de date, în care sunt stocate informații despre clienții implicați în procesul de vânzare a semifabricatelor.;
* **“Vanzari”** – Subdomeniul care cuprinde datele despre produsele achiziționate, cantități, prețuri și totaluri.;
* **“Produse”** – Subdomeniul care reprezintă informații complete despre semifabricatele disponibile pentru vânzare, inclusiv detalii despre stocuri, caracteristici, și prețurile asociate.;
* **“Producatori”** – Subdomeniul dedicat informațiilor despre producătorii sau furnizorii semifabricatelor, cuprinzând date despre companii și relații cu produsele oferite.;
* **“Depozit”** – subdomeniul care reprezintă informatii despre depoziturile de produse.

2.4 Legături între obiectele Domeniului de Studiu



**Figura 2.4.1** Legături între obiectele DS.

2.5 Caracteristicile informaționale ale “ Platformă Web pentru Gestionarea Vânzărilor de Semifabricate”

Ieșiri informaționale:

1. Lista produse:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID produs | Denumire | Descriere | Categorie | Specificatii tehnice | Pret | Stoc disponibil | Data\_producerii | Termen\_valabilitate | ID\_Producator | ID\_depozit | ID\_categorie | Stoc\_disponibil |

1. Lista producatori:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_producator | Nume\_producator | Adresa | Numar\_contact | Info\_producator |

1. Lista vanzari:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID\_vanzare | ID\_produs | ID\_client | Cantitate\_vanduta | Data\_vanzare | Total\_vanzare | ID\_categorie |

1. Lista clienti:

|  |
| --- |
| ID\_clienti |

1. Lista depozit:

|  |  |
| --- | --- |
| ID\_depozit | Locatie |

Interogări:

1.Produsul XXXX cel mai cumparat in categoria YYYY.

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_produs | Categorie | ID\_vanzare | Cantitate\_vanduta |

2.Cantitatea totala XXXX vanduta de un YYYY producator.

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_producator | Nume\_producator | ID\_vanzare | Total\_vanzare |

3. Afișează toate vânzările pentru produsul cu ID-ul "XXXX".

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_produs | Denumire\_produs | ID\_vanzare | Cantitate\_vanduta |

4. Calculează suma totală a vânzărilor pentru luna XXXX.

|  |  |  |
| --- | --- | --- |
| ID\_vanzare | Data\_vanzare | Total\_vanzare |

5. Găsește toate produsele oferite de producătorul cu ID-ul "XXXX":

|  |  |  |  |
| --- | --- | --- | --- |
| ID\_producator | Nume\_producator | ID\_produs | Denumire\_produs |

Servicii:

1. Este dat depozitul XXXX, să se determine numarul de produse YYYY , in categoria ZZZZ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_depozit | Locatie | Stoc\_disponibil | ID\_produs | Denumire\_produs | Categorie |

1. Sa se determine producatorul XXXX , care domina in categoria YYYY:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_producator | Nume\_producator | ID\_Produse | Denumire\_produs | Categorie |

1. Obține informații despre producătorul XXXX și produsul YYYY din categoria ZZZZ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_poducator | Nume\_producator | Info\_producator | ID\_produs | Denumire\_produs | Categorie |

1. Sa se determinare numarul de produse XXXX , in categoria YYYY , disponibile in depozitul ZZZZ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID\_produs | Denumire\_produs | Categorie | ID\_depozit | Stoc\_disponibil | Stoc\_disponibil |

1. Sa se gaseasca produsele XXXX , care corespund specificatiilor tehnice YYYY:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID\_produs | Denumire\_produs | Specificatii\_tehnice | Pret | Stoc\_disponibil |

Restricții:

1. **ID\_producator**, **ID\_depozit**, **ID\_produs**, **ID\_vanzare**, **ID\_clienti** sunt unice și nu coincide, sunt elemente cheie.
2. Un **producator** poate avea mai multe **produse**.
3. Un **depozit** poate să aiba doar o **locatie**.
4. **Pretul , Stoc\_disponibil , Cantiate\_vanduta** nu pot fi negativ.
5. Pentru **producatori Adresa** trebuie sa fie unica.

2.6 Tabel de acoperire

Tabel . Tabel de acoperire

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Informația  Atribute | Ieșiri informaționale | | | | | Interogari | | | | | Servicii | | | | |
|  | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 | 1 | 2 | 3 | 4 | 5 |
| ID\_producator |  | + |  |  |  |  | + |  |  | + |  | + | + |  |  |
| Nume\_producator |  | + |  |  |  |  | + |  |  | + |  | + | + |  |  |
| Adresa |  | + |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Numar\_contact |  | + |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Info\_producator |  | + |  |  |  |  |  |  |  |  |  |  | + |  |  |
| ID\_depozit |  |  |  |  | + |  |  |  |  |  | + |  |  | + |  |
| Locatie |  |  |  |  | + |  |  |  |  |  | + |  |  |  |  |
| Stoc\_disponibil |  |  |  |  | + |  |  |  |  |  | + |  |  | + |  |
| ID\_vanzare |  |  | + |  |  | + | + | + | + |  |  |  |  |  |  |
| Cantitate\_vanduta |  |  | + |  |  | + |  | + |  |  |  |  |  |  |  |
| Data\_vanzare |  |  | + |  |  |  |  |  | + |  |  |  |  |  |  |
| Total\_vanzare |  |  | + |  |  |  | + |  | + |  |  |  |  |  |  |
| ID\_Produs | + | + | + |  | + | + |  | + |  | + | + | + | + | + | + |
| Denumire\_produs | + |  |  |  |  |  |  | + |  | + | + | + | + | + | + |
| Descriere | + |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Categorie | + |  |  |  |  | + |  |  |  |  | + | + | + | + |  |
| Specificatii\_tehnice | + |  |  |  |  |  |  |  |  |  |  |  |  |  | + |
| Pret | + |  |  |  |  |  |  |  |  |  |  |  |  |  | + |
| Stoc\_disponibil | + |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ID\_client |  |  | + | + |  |  |  |  |  |  |  |  |  |  |  |

Tabel . Machetul bazei de date

|  |  |  |  |
| --- | --- | --- | --- |
| Tabel | Câmp | Tip | Comentarii |
| Clienti | Id\_Clienti | INT | PRIMARY KEY, not null |
|  | ID\_producator | INT | PRIMARY KEY, not null |
|  | Nume\_producator | VARCHAR |  |
| Producator | Adresa | VARCHAR |  |
|  | Numar\_contact | INT |  |
|  | Info\_producator | TEXT |  |
|  | ID\_produs | INT | FOREIGN KEY |
|  | ID\_depozit | INT | PRIMARY KEY, not null |
|  | Locatie | VARCHAR |  |
| Depozit | Stoc\_disponibil | INT |  |
|  | ID\_produs | INT | FOREIGN KEY |
|  | ID\_vanzare | INT | PRIMARY KEY, not null |
|  | Cantitate\_vanduta | INT |  |
|  | Data\_vanzare | DATE |  |
| Vanzari | Total\_vanzare | DECIMAL |  |
|  | ID\_produs | INT | FOREIGN KEY |
|  | ID\_client | INT | FOREIGN KEY |
|  | ID\_produs | INT | PRIMARY KEY, not null |
|  | Denumire\_produs | INT |  |
|  | Descriere | TEXT |  |
| Produse | Categorie | INT |  |
|  | Specificatii\_tehnice | TEXT |  |
|  | Pret | DECIMAL |  |
|  | Stoc\_disponibil | INT |  |

### Introducere Programare WEB

3.1 HTML prezentare

### HTML, sau HyperText Markup Language, este un limbaj de marcare folosit pentru a crea pagini web afișabile în browsere (navigatoare). Scopul principal al HTML este prezentarea informațiilor, cum ar fi paragrafe, fonturi și tabele, mai degrabă decât descrierea semanticii documentului. World Wide Web Consortium (W3C) dictează specificațiile HTML, iar limbajul este parte a familiei SGML (Standard Generalized Markup Language).

### Documentele HTML sunt fișiere text salvate cu extensiile .html sau .htm. Structura unui document HTML include elemente precum `<html>`, `<head>` (cap), `<title>` (titlu) și `<body>` (corp). Acestea pot fi create cu un editor de texte sau cu editoare HTML, iar limbajul oferă autorilor posibilitatea de a publica diverse elemente, inclusiv headere, texte, tabele, liste și imagini. De asemenea, HTML permite crearea de link-uri, formulare pentru interacțiunea cu serverele la distanță și includerea diverselor aplicații în documente.

### Limbajul HTML are patru caracteristici principale:

### 1. Utilizează marcaje descriptiv pentru a indica acțiuni specifice în document.

### 2. Definește structuri ierarhice și legături între elementele documentelor.

### 3. Este guvernat de o descriere formală, Document Type Definition (DTD), care stabilește specificațiile formale.

### 4. Este citit atât de oameni, cât și de computere, datorită separării elementelor de marcare de text.

### HTML este bazat pe SGML și moștenește trăsături de la acesta. SGML este un standard internațional care permite reprezentarea structurii documentelor și hyper-documentelor, fiind utilizat pentru a codifica aceste documente pentru interschimb între autori.

### Limbajul de scriptare este esențial în comunicarea dintre servere și clienții web la nivel de aplicație, iar protocolul HTTP (HyperText Transfer Protocol) asigură această comunicare. Un aspect important al protocolului HTTP este independența de platformă și capacitatea de a susține sisteme distribuite pentru managementul obiectelor de date.

### Common Gateway Interface (CGI) este un standard pentru interacțiunea între clienții web și serverele web, asigurând interactivitatea paginilor web. Programele CGI, sau scripturile, pot fi scrise în diverse limbaje precum PHP, Perl, Python sau C, și oferă suport pentru autentificarea utilizatorilor pe server.

### În concluzie, HTML și protocolul HTTP sunt fundamentale în crearea și interacțiunea cu paginile web, iar CGI permite adăugarea de interactivitate și prelucrarea datelor în timp real.

3.3 PHP reprezentare

### PHP, sau Hypertext Preprocessor, cunoscut anterior sub denumirea de PHP/FI (Personal Homepage/Form Interpreter), a fost inițial conceput ca o aplicație CGI simplă pentru interpretarea formularelor definite prin HTML și procesate de un program în limbaj Perl sau shell script pe server. Înainte de PHP, interfața CGI necesita permisiuni pentru a rula programe pe server, ceea ce implica lacune în securitate și separarea programului de procesare a datelor de documentul HTML.

### PHP, în versiunea curentă PHP 5.0, reprezintă un pachet puternic care oferă un limbaj de programare accesibil direct din fișierele HTML. Acesta seamănă cu limbaje precum Perl sau C și oferă suport pentru manipularea bazelor de date în diverse dialecte SQL (dBase, Informix, MySQL, mSQL, Oracle, PostgreSQL, Solid, Sybase, ODBC, etc.). PHP are, de asemenea, suport pentru încărcarea de fișiere de pe calculatorul client și gestionează cookie-urile. Este disponibil gratuit pe internet și se integrează bine cu serverul Apache.

### PHP are un număr estimat de peste un milion de site-uri care îl utilizează în prezent. Există o varietate de aplicații și utilitare dezvoltate în PHP, grupate în cadrul PHP Extension and Add-on Repository (PEAR).

### Diferența cheie dintre PHP și alte limbaje CGI precum Perl sau C constă în faptul că, în loc să scrieți un program cu multe linii de comandă care afișează în cele din urmă o pagină HTML, scrieți direct o pagină HTML cu codul PHP inclus pentru a realiza acțiuni specifice. Codul PHP este inclus între tag-urile speciale care permit trecerea din "modul HTML" în "modul PHP".

### De asemenea, este important de menționat că, spre deosebire de limbaje precum JavaScript, codul PHP se execută pe serverul web, nu în navigatorul web al clientului. Acest lucru oferă PHP acces la fișiere, baze de date și alte resurse inaccesibile pentru JavaScript. PHP poate produce conținut dinamic atractiv pentru vizitatori, iar serverul poate fi configurat pentru a procesa toate fișierele HTML ca fișiere PHP, fără a se distinge între paginile produse dinamic și cele statice.

### PHP oferă aceleași funcții ca alte limbaje de scripting CGI, permițând colectarea și procesarea datelor și generarea paginilor web dinamice sau manipularea cookie-urilor. Un avantaj semnificativ al limbajului PHP este suportul său extins pentru o varietate mare de baze de date, făcând crearea de pagini web dinamice cu bază de date extrem de simplă. PHP are, de asemenea, suport pentru diverse servicii de server folosind protocoale precum IMAP, SNMP, NNTP, POP3 și HTTP.

### Un program PHP include întotdeauna două linii speciale, delimitate de tag-urile `<?php ?>`, care indică serverului PHP că textul cuprins între aceste linii conține instrucțiuni PHP. Operațiile principale ale programelor PHP includ obținerea datelor de la utilizatori, prelucrarea datelor (accesarea și manipularea datelor stocate în fișiere și baze de date) și afișarea datelor pentru vizualizarea utilizatorului.

### În concluzie, PHP oferă o soluție puternică și accesibilă pentru dezvoltarea paginilor web dinamice, oferind suport pentru diverse baze de date și servicii de server.

3.4 Serverul Web Apache

### Serverul web Apache este fundamentul a peste 60% din domeniile web, conform unui studiu Netcraft Web Server. Acesta este ales frecvent de furnizorii de internet și companiile de găzduire de aplicații web datorită suportului extins pentru diverse platforme, facilităților ușor de implementat pentru găzduirea virtuală și modulelor care extind capacitățile sale.

### Caracteristica Apache Portable Routine (APR) optimizează funcționalitățile, precum administrarea proceselor, pentru fiecare sistem de operare. Aceasta permite serverului să ignore diferențele specifice ale platformelor, îmbunătățind astfel performanța și stabilitatea implementărilor pe Windows, eliminând nevoia de emulator. APR este accesibil și dezvoltatorilor care scriu programe multi-platformă în limbajul C, cunoscut pentru viteza sa în comparație cu limbajele interpretative precum PHP și Perl utilizate în mod obișnuit în dezvoltarea web.

### Un alt avantaj al Apache este versatilitatea sa. API-ul deschis permite dezvoltatorilor să scrie diferite module care pot modifica comportamentul serverului. Dacă serverul lipsește o funcționalitate necesară unui site, există șanse mari să existe un modul plug-in disponibil.

### Serverele web au rolul de a primi cereri anonime de la clienți și de a furniza informații într-un mod eficient și rapid. Apache este un daemon care acceptă conexiuni conform protocolului HTTP și răspunde la cererile primite de la clienți. Pentru a oferi servicii HTTP, Apache trebuie să fie instalat în sistem și daemon-ul httpd să fie activat. Apache este un sistem modular, format dintr-un server de bază și module încărcate dinamic, asemănător cu modul de funcționare al modulelor din nucleul Linux.

### Configurarea Apache se poate realiza cu ajutorul interfeței grafice apache-conf (Apache Configuration Tool). Fișierul principal de configurare este httpd-conf și este de obicei localizat în directorul /etc/httpd pentru versiunile de Linux sau Unix.

### Apache oferă posibilitatea de a restricționa accesul la anumite documente prin autentificare cu nume de utilizator și parolă sau în funcție de adresa calculatorului clientului web. Pentru a realiza autentificarea utilizatorilor, se creează un fișier conținând nume și parole, iar serverul se configurează pentru a proteja resursele și a permite accesul doar utilizatorilor autorizați.

### Un aspect remarcabil al Apache este capacitatea sa de a oferi găzduire virtuală, ceea ce înseamnă că poate servi mai multe site-uri web simultan. Există două metode de implementare a găzduirii virtuale: una bazată pe nume și cealaltă bazată pe adrese IP. Găzduirea virtuală bazată pe nume este mai simplu de implementat și recomandată. Aceasta se realizează stabilind adresa IP și portul pentru serverul care va accepta cereri pentru mașina virtuală specifică.

### În concluzie, Apache este un server web puternic și versatil, cu suport pentru diverse platforme și capacitatea de a oferi găzduire virtuală, facându-l alegerea preferată pentru numeroase domenii web și aplicații online.

3.5 Administrarea Bazelor de Date cu MySQl

### MySQL, dezvoltat de compania suedeză MySQL AB, este un server de baze de date gratuit și cu sursă deschisă, cunoscut pentru fiabilitatea și simplitatea sa. Administrarea bazei de date MySQL se realizează prin intermediul unei serii de utilitare în linia de comandă, cu mysql fiind cel mai important - un shell interactiv pentru controlul și interogarea bazei de date. Acesta funcționează eficient pe sistemele Linux, platforma pentru care a fost inițial dezvoltat. Pe platforma Windows, sunt disponibile două utilitare cu sursă deschisă: MySQL Manager, un utilitar grafic pentru interogări similar cu mysql, și WinMySQL Admin, o consolă pentru gestionarea detaliilor de configurare ale MySQL.

### MySQL facilitează importul și exportul diverselor baze de date sub forma fișierelor SQL. Deși tranzacțiile nu sunt parte a tabelelor implicite ISAM (Indexed Sequential Access Method) ale MySQL, acesta include două tipuri noi de tabele, Berkley DB (BDB) și InnoDB, dezvoltate de alte companii. Comunitatea de administratori și dezvoltatori de baze de date apreciază dezvoltarea activă și loialitatea utilizatorilor, consolidând reputația MySQL ca o bază de date sigură.

### MySQL funcționează pe un model client/server, unde orice mașină care dorește să proceseze interogări asupra unei baze de date MySQL trebuie să ruleze serverul MySQL (mysqld). Acesta gestionează tot traficul de intrare/ieșire cu baza de date și ascultă pe un port specific (3306) pentru cererile de conexiune ale clientului. Modelul de securitate al MySQL se bazează pe nume de utilizator, parolă, nume de server sau adresă IP și privilegii, similare cu sistemul Unix.

### Interacțiunea cu MySQL se realizează în principal prin intermediul comenzilor SQL, iar PHP oferă o bibliotecă de funcții pentru a interacționa cu sistemul de gestionare a bazelor de date MySQL. Acest lucru permite programelor PHP să acceseze și să modifice datele dintr-o bază de date MySQL.

### Majoritatea operațiunilor cu o bază de date MySQL urmează un model secvențial simplu în limbajul PHP:

### 1. Se deschide o conexiune cu serverul MySQL.

### 2. Se specifică baza de date la care se dorește accesul.

### 3. Se emit interogări SQL, se obțin rezultatele și se execută operațiuni non-SQL.

### 4. Se închide conexiunea cu serverul MySQL.

### Există două categorii principale de interogări SQL în PHP: interogările SELECT, care returnează rânduri ale unui tabel, și interogările UPDATE, INSERT și DELETE, care nu returnează rânduri ale unui tabel. Ambele categorii de interogări sunt emise cu ajutorul funcției mysql\_query(), iar procesarea rezultatelor diferă între cele două categorii.

### 4.1 Aplicatia Web „Supermarket Analytics”

### Aplicația web pare este asociată cu analiza datelor pentru un supermarket. Există secțiuni legate de vânzări, producători, depozite și un panou de bord general.

### 

### Pagina "sales.html" este dedicată analizei și gestionării vânzărilor în cadrul aplicației web "Supermarket Analytics". Iată o prezentare a componentelor principale ale paginii:

### Navigare și Antet:

### Există un antet care indică titlul paginii ("Sales Page") și include, de asemenea, numele aplicației ("Supermarket Analytics").

### Un meniu de navigare laterală oferă opțiuni pentru a accesa diferitele secțiuni ale aplicației: "Dashboard," "Sales," "Manufacturers," și "Deposits."

### Filtrare și Butoane de Acțiune:

### Pagina include un meniu derulant pentru a selecta o categorie de produse (SelectedCategory).

### Butonul "Get Top Selling Products" inițiază afișarea produselor cele mai bine vândute în categoria selectată.

### Butonul "Sales products" (SalesButton) direcționează utilizatorul către pagina "manageSales.html."

### Top Selling Products in Category:

### Afișează o secțiune care indică cele mai bine vândute produse în categoria selectată.

### Un grafic de bare (salesChart) arată cantitatea vândută pentru fiecare produs.

### Product Sales:

### O secțiune care permite utilizatorilor să selecteze o categorie și un produs specific pentru a obține informații despre vânzări.

### Butonul "Get Product Sales" inițiază afișarea datelor despre vânzări pentru produsul selectat.

### Total Sales for Selected Month:

### Permite utilizatorilor să selecteze un an și o lună specifică pentru a calcula și afișa totalul vânzărilor pentru acea perioadă.

### Un grafic de linii (dailySalesChart) prezintă cantitatea totală vândută în fiecare zi a lunii selectate.

### Pagina(`warehouses.html`):

### - Cuprinde un meniu de navigare cu patru opțiuni: "Dashboard", "Sales", "Manufacturers", și "Deposits" (Depozite).

### - Conține două secțiuni principale: o bară laterală și conținutul principal.

### - Partea principală a conținutului include o titlul "Warehouse Page" și două secțiuni referitoare la produsele depozitelor.

### 2. Secțiunea "Warehouse Products":

### - Permite utilizatorului să selecteze un depozit și o categorie de produse.

### - Include două butoane: "Get Category Product Count" și "Manage deposits".

### 3. Secțiunea "Warehouse Product Count":

### - Permite utilizatorului să selecteze un producător, o categorie și un depozit.

### - Include un buton "Get Product Count" pentru a obține informații despre numărul de produse.

### 4. Scripturile JavaScript asociate (`warehouses.js`):

### - Conțin funcții precum `fetchAndDisplayProductsInfo`, `fetchAndPopulateCategories`, `fetchAndPopulateWarehouses`, și altele, pentru manipularea datelor și actualizarea interfeței.

### 5. Pagina de administrare a depozitelor (`manageWarehouses.html`):

### - Are un meniu de navigare similar cu cel al paginii principale.

### - Conține o secțiune principală cu un titlu "Deposits Management Page" și două butoane: "Back", "Add", și "Edit".

### 6. Scripturile JavaScript asociate (`manageWarehouses.js`):

### - Conțin funcții precum `goBack`, `addTable`, `editTable`, `submitData`, `submitEdit`, `submitDelete`, pentru manipularea datelor și gestionarea interfeței în timpul administrării depozitelor.

### Această aplicație web pare să se concentreze pe gestionarea și analiza datelor legate de depozite, produse și producători într-un mediu de supermarket.

### ### Pagina "Manufacturer - Supermarket Analytics" (`manufacturer.html`):

### 1. Bară de navigare și Meniu lateral:

### - Meniul lateral conține patru opțiuni: "Dashboard", "Sales", "Manufacturers", și "Deposits".

### - Pagina curentă ("Manufacturers") este evidențiată în meniu.

### 2. Titlu și Subsecțiuni:

### - Titlul paginii este "Manufacturer Page".

### - Secțiunea "Manufacturer Statistics" conține un meniu derulant pentru a selecta un producător și două butoane pentru a obține statistici sau pentru a gestiona produsele asociate producătorului.

### - Secțiunea "All products by manufacturer select" permite selectarea unui producător și afișează produsele asociate. Există, de asemenea, un buton pentru a gestiona producătorii.

### - Secțiunea "Product Information" permite introducerea numelui unui produs și afișează informații despre acel produs.

### - Secțiunea "Product Search by Technical Specifications" oferă opțiunea de a căuta produse după specificații tehnice.

### 3. Scripturi JavaScript:

### - Pagina folosește scriptul `manufacturer.js` pentru a gestiona interactivitatea și manipularea datelor pe pagină.

### - Acest script conține funcții precum `getManufacturerStats`, `getallProductByManufacturer`, `getManufacturerProductInfo`, `searchProductsBySpecifications`, și altele.

### ### Pagina "Manufacturers Management Page" (`manageManufacturer.html`):

### 1. Bară de navigare și Meniu lateral:

### - Meniul lateral este similar cu cel al paginii principale.

### - Pagina curentă ("Manufacturers Management Page") este evidențiată în meniu.

### 2. Titlu și Subsecțiuni:

### - Titlul paginii este "Manufacturers Management Page".

### - Secțiunea conține un buton de revenire (`Back`) și două butoane pentru a adăuga și edita date despre producători.

### 3. Script JavaScript:

### - Pagina folosește scriptul `manageManufacturer.js` pentru a gestiona interacțiunile specifice gestionării producătorilor.

### - Acest script conține funcții precum `goBack`, `addTable`, `editTable`, `submitData`, `submitEdit`, `submitDelete`, utilizate pentru manipularea datelor și actualizarea interfeței.

### 4. Formulare pentru Adăugare și Editare:

### - Există funcții precum `addTable` și `editTable`, care permit adăugarea și editarea informațiilor despre producători.

### ### Scripturi JavaScript Comune (`manufacturer.js` și `manageManufacturer.js`):

### - Ambele scripturi folosesc funcții pentru a efectua solicitări AJAX către API-uri (`get\_manufacturers.php`, `get\_manufacturer\_stats.php`, `get\_products\_by\_manufacturer.php`, etc.) pentru a obține și manipula datele.

### - Există funcții pentru afișarea și manipularea datelor, cum ar fi `displayManufacturerStats`, `displayProductsByManufacturer`, `displayProductInfo`, și `displayMatchingProducts`.

### - Pagina de gestionare a producătorilor (`manageManufacturer.html`) conține funcții pentru manipularea datelor prin adăugarea, editarea și ștergerea producătorilor din baza de date (`add\_manufacturers\_databases.php`, `update\_manufacturer.php`, `delete\_manufacturer.php`).

### Pagina pe care ai furnizat-o este o pagină HTML care pare să fie parte a unui sistem de administrare a produselor pentru un supermarket. Această pagină pare să fie asociată cu gestionarea produselor și furnizorilor.

### Iată o scurtă descriere a elementelor principale ale paginii:

### 1. Navigare Sidebar: Oferă link-uri către diverse secțiuni ale sistemului, cum ar fi "Dashboard", "Sales", "Manufacturers" și "Deposits". Acest meniu de navigare este afișat în partea stângă a paginii.

### 2. Titlu Pagină: Titlul paginii este "Products Management Page" și este afișat în partea de sus a conținutului principal.

### 3. Butoane și Funcții:

### - Back Button: Un buton "Back" care, atunci când este apăsat, îl redirecționează pe utilizator înapoi la pagina "Manufacturers".

### - Add Products Button: Un buton "Add" care, atunci când este apăsat, activează o funcție `addTable()` care pare să adauge un tabel pentru introducerea datelor despre un nou produs.

### - Edit Products Button: Un buton "Edit" care, atunci când este apăsat, activează o funcție `editTable()` care pare să creeze un tabel pentru editarea datelor produselor existente.

### 4. Funcția `addTable()`:

### - Acesta adaugă un tabel în care utilizatorii pot introduce date pentru un nou produs. Există câmpuri pentru nume, descriere, categorie, specificații tehnice, preț, stoc disponibil și alte detalii.

### - De asemenea, afișează un buton "Save" care, atunci când este apăsat, declanșează funcția `saveFormData()` pentru a salva datele introduse.

### 5. Funcția `editTable()`:

### - Aceasta pare să adauge un tabel pentru a afișa și edita produsele existente. Există un meniu dropdown pentru selectarea categoriei de produse, iar în funcție de această selecție, produsele corespunzătoare sunt afișate în tabel.

### - Pentru fiecare produs, există o opțiune de ștergere.

### 6. Funcțiile pentru manipularea datelor:

### - Există funcții cum ar fi `fetchAndPopulateCategories`, `fetchAndDisplayProducts`, `deleteProductData`, `populateFormFields`, `updateFormData`, etc., care sunt utilizate pentru a interacționa cu API-uri pentru obținerea, gestionarea și actualizarea datelor legate de produse și producători.

### 

### 

### 

### Concluzie

### 1. Funcționalitatea de Adăugare și Editare a Produselor: Pagina oferă funcționalități pentru adăugarea și editarea produselor. Utilizatorii pot introduce noi produse prin completarea unui formular generat dinamic și pot edita produsele existente, având posibilitatea de a salva modificările.

### 2. Interacțiune cu API-uri: Se pare că aplicația comunică cu API-uri pentru a obține și actualiza datele legate de categorii, producători, depozite și produse. Acest lucru indică o arhitectură de tip client-server, unde partea de frontend interacționează cu un backend pentru gestionarea datelor.

### 3. Gestionarea Categoriilor, Producătorilor și Depozitelor: Există funcții dedicate pentru obținerea și afișarea categoriilor, producătorilor și depozitelor, iar aceste informații sunt utilizate în formularul de adăugare/editare a produselor

### 4. Interfață Utilizator Prietenoasă: Prin intermediul meniului de navigare și a butoanelor intuitive, interfața utilizator pare să fie prietenoasă și ușor de înțeles. Utilizatorii pot naviga între diferitele secțiuni ale aplicației.

### 5. Confirmare Ștergere: Atunci când utilizatorii încearcă să șteargă un produs, există o confirmare pentru a preveni ștergerile accidentale.

### 6. Reîmprospătarea Automată a Paginii: După adăugarea sau ștergerea unui produs, pagina se reîmprospătează automat după o scurtă întârziere. Acest lucru ar putea fi util pentru a reflecta instantaneu schimbările în interfață.

### 7. Potențiale Îmbunătățiri: Există posibilitatea de îmbunătățire a stilizării și organizării codului pentru a face acesta mai modular și ușor de întreținut. De asemenea, gestionarea erorilor și gestionarea răspunsurilor de la API-uri ar putea fi îmbunătățite pentru o experiență mai robustă.

### În ansamblu, această aplicație ofera funcționalități esențiale pentru gestionarea produselor într-un mediu de supermarket, cu accent pe interacțiunea cu datele prin intermediul API-urilor.