



Universitatea Tehnică de Construcții București

**Facultatea
de Hidrotehnică**

Bază de date pentru un magazin de bricolaje

-PROIECT-

Student: Tene Radu-Stefan

Specializare: AIA

An: 3

Grupa: 2

1.Introducere

Descrierea temei alese

Proiectul propus vizează dezvoltarea unei baze de date pentru un magazin de bricolaj, cu scopul de a organiza și gestiona informațiile legate de produsele comercializate, clienți, furnizori, comenzi, stocuri și plăți. Alegerea acestei teme se bazează pe necesitatea de a îmbunătăți gestionarea eficientă a activităților magazinului și de a facilita procesele interne, cum ar fi urmărirea produselor și optimizarea stocurilor. O bază de date bine proiectată va permite o mai bună gestionare a produselor, relațiilor cu clienții și furnizorii, precum și o mai bună monitorizare a comenzilor și vânzărilor.

Obiective propuse

- Crearea unei baze de date structurate pentru a gestiona informațiile despre produse, comenzi, clienți, stocuri și furnizori.
- Implementarea unor măsuri de integritate a datelor, prin utilizarea de constrângeri și tipuri de date corespunzătoare.
- Elaborarea de interogări care să permită o gestionare rapidă și eficientă a produselor, comenzilor și stocurilor.
- Generarea de rapoarte utile în analiza performanțelor magazinului și în luarea unor decizii informate.

Structura proiectului

Lucrarea este organizată în trei capitole. Primul capitol introduce tema proiectului, obiectivele acestuia și structura generală a lucrării. În al doilea capitol vom găsi prezentate tehnologiile și instrumentele utilizate pentru dezvoltarea optimă a bazei de date, iar în capitolul trei vom regăsi detalii cu privire la procesul de realizare a proiectului, incluzând designul bazei de date și abordările privind reprezentarea și gestionarea datelor.

2. Tehnologii utilizate

SQL

SQL (Structured Query Language) este un limbaj de programare standardizat, utilizat pentru gestionarea bazelor de date relaționale. SQL permite crearea, modificarea și interogarea bazelor de date, precum și definirea și modificarea structurii tabelelor și a altor obiecte asociate cu baza de date.

SQL a fost inițial dezvoltat la compania IBM de către Donald D. Chamberlin și Raymond F. Boyce în anul 1974. La început, numele era SEQUEL (tot de la Structured QueryLanguage) și avea ca scop manipularea și regăsirea datelor stocate în bazele de date relaționale IBM.

SQL este un limbaj declarativ, ceea ce înseamnă că utilizatorii specifică ceea ce doresc să realizeze, și nu modul în care trebuie să fie realizate aceste operațiuni. Acesta se bazează pe conceptul de relații, astfel încât datele sunt organizate în tabele care sunt interconectate prin intermediul cheilor primare și a celor străine.

Prin intermediul SQL, se pot realiza o gamă largă de operațiuni, cum ar fi crearea, citirea, actualizarea și ștergerea datelor dintr-o bază de date. De asemenea, SQL permite interogarea și agregarea datelor, sortarea, filtrarea și gruparea acestora într-o varietate de moduri.

Există mai multe dialecte de SQL, dar majoritatea interogărilor sunt portabile între diferite sisteme de gestiune a bazelor de date, deoarece majoritatea producătorilor de baze de date au implementat același standard SQL. Cu toate acestea, există și unele diferențe semnificative între diferitele implementări ale standardului SQL, ceea ce poate duce la probleme de compatibilitate între sistemele de gestiune a bazelor de date.

IntelliJ IDEA

IntelliJ IDEA este un mediu de dezvoltare integrat (IDE) dezvoltat de JetBrains, cunoscut pentru suportul său extins pentru diverse limbaje de programare, inclusiv Java. Este un instrument puternic și flexibil, care include funcționalități avansate pentru dezvoltare, refactorizare, debugging și gestionarea proiectelor.

Pe lângă suportul său robust pentru programarea în Java, IntelliJ IDEA oferă și un set de unelte dedicate pentru gestionarea bazelor de date. Cu ajutorul plugin-ului integrat Database Tools and SQL, acest IDE permite conectarea la baze de date Oracle, executarea interogărilor SQL, vizualizarea și editarea tabelor, precum și gestionarea obiectelor bazei de date. Interfața sa prietenoasă și funcțiile inteligente, precum completarea automată a codului SQL și evidențierea sintaxei, fac din IntelliJ un instrument ideal pentru lucrul cu bazele de date.

Am ales să utilizez IntelliJ IDEA pentru acest proiect deoarece sunt familiarizat cu acest mediu de dezvoltare din experiența mea anterioară cu limbajul de programare Java. Această familiaritate îmi oferă o eficiență crescută și o mai bună înțelegere a funcționalităților oferite de IDE, ceea ce facilitează lucrul atât cu codul, cât și cu baza de date. Personal, prefer IntelliJ IDEA datorită interfeței intuitive, integrării avansate cu baze de date și flexibilității pe care o oferă în gestionarea proiectelor complexe.

Modul de descriere a datelor

Baza de date este structurată pentru un magazin de bricolaj care gestionează informații despre produse, angajați, clienți, furnizori, recenzii, comenzi, retururi, promoții și stocuri.

Aceasta conține 10 tabele: **Magazin, Clienti, Furnizori, Produse, Angajati, Recenzii, Comenzi, Retururi, Promotii și Stoc.**

Fiecare tabel are coloane cu tipuri de date specifice, iar cheile primare și cheile externe sunt definite pentru a asigura integritatea referențială a datelor.

Atributele tabelelor sunt definite cu tipuri de date precum NUMBER, VARCHAR, CHAR și DATE. Parametrul NOT NULL indică faptul că un atribut nu poate avea valoarea NULL, iar PRIMARY KEY definește cheia primară a fiecărui tabel, care este utilizată pentru a identifica în mod unic fiecare înregistrare. Atributele care leagă tabelele între ele sunt definite drept chei externe (FOREIGN KEY)

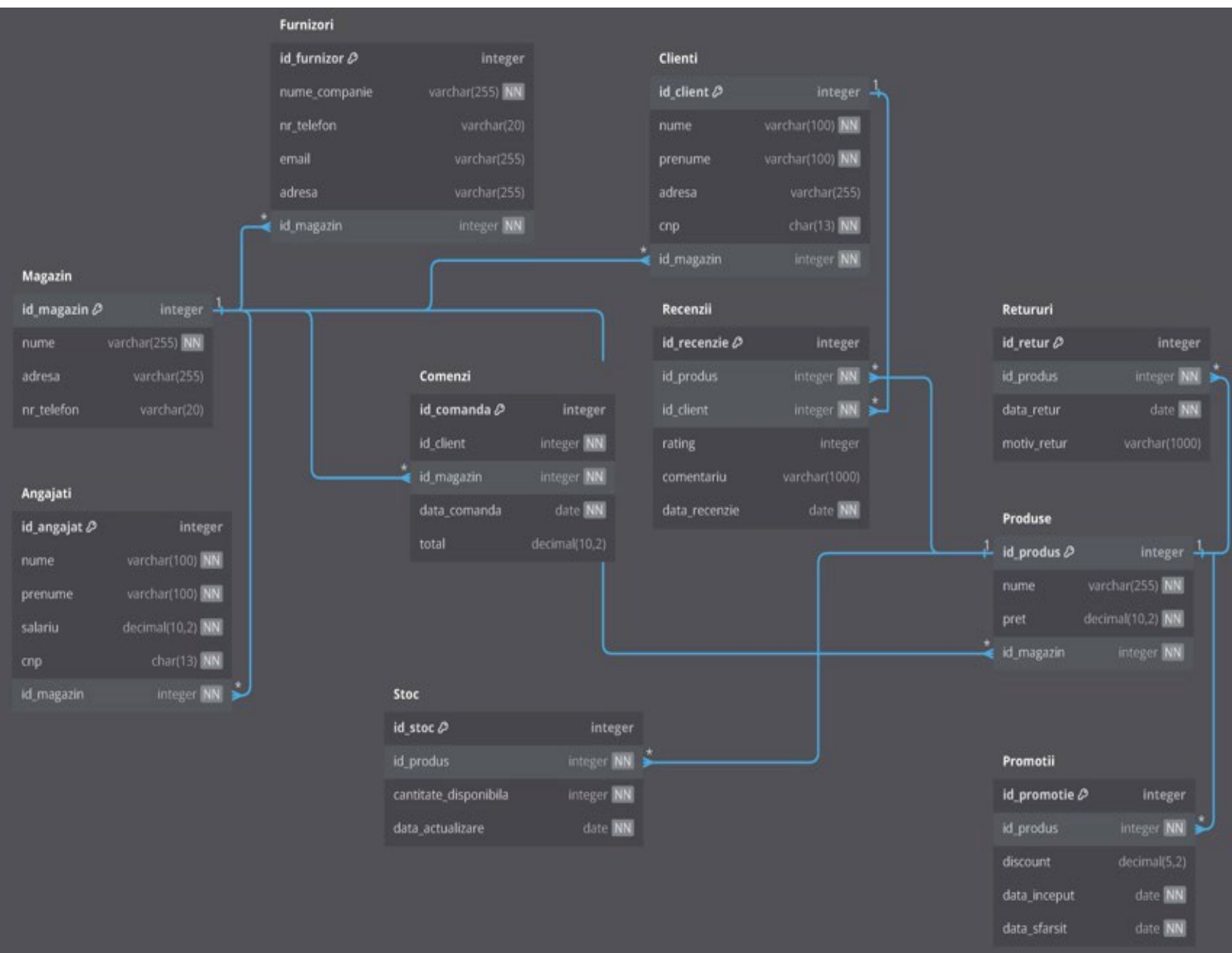


Figura 1.

Diagrama entitate-asociere (EA)

Descrierea diagramei, a relațiilor, a entităților și a atributelor

Schema bazei de date conține următoarele tabele: **CLIENTI, MAGAZIN, FURNIZORI, PRODUSE, ANGAJATI, RECENZII, COMENZI, RETURURI, PROMOTII și STOC**, care sunt legate între ele prin chei primare și străine. Tabelele descriu diferite aspecte ale unui magazin.

MAGAZIN

Tabela **MAGAZIN** stochează informațiile despre magazinele din sistem, inclusiv id_magazin (cheie primară), care este un identificator unic pentru fiecare magazin, nume (numele magazinului), adresa (adresa fizică) și nr_telefon (numărul de telefon). Această tabelă este centrală și este referențiată de tabelele **CLIENTI, FURNIZORI, PRODUSE, ANGAJATI și COMENZI** prin coloana id_magazin.

CLIENTI

Tabela **CLIENTI** stochează informații despre clienți, precum id_client (cheie primară), nume (numele clientului), prenume (prenumele clientului), adresa (adresa acestuia) și cnp (cod numeric personal, unic și obligatoriu). De asemenea, include o cheie străină, id_magazin, care indică magazinul asociat clientului. Relația dintre clienți și magazin este de tip many-to-one.

FURNIZORI

Tabela **FURNIZORI** conține detalii despre furnizorii fiecărui magazin, incluzând id_furnizor (cheie primară), nume_companie (numele companiei furnizoare), nr_telefon (număr de telefon), email (adresa de e-mail) și adresa (adresa sediului). Cheia străină id_magazin leagă fiecare furnizor de un magazin specific, formând o relație many-to-one.

PRODUSE

Tabela **PRODUSE** înregistrează informații despre produsele disponibile, cum ar fi id_produș (cheie primară), nume (numele produsului), pret (prețul acestuia) și id_magazin (cheie străină către magazin). Aceasta este referențiată de mai multe alte tabele, inclusiv **RECENZII**, **RETURURI**, **PROMOȚII** și **STOC**, pentru a gestiona detalii suplimentare despre produse.

ANGAJATI

Tabela **ANGAJATI** conține informații despre angajații magazinelor, incluzând id_angajat (cheie primară), nume (numele angajatului), prenume (prenumele acestuia), salariu (salariul lunar), cnp (cod numeric personal, unic și obligatoriu) și id_magazin (cheie străină care leagă angajatul de magazin). Relația este de tip many-to-one cu tabela **MAGAZIN**.

RECENZII

Tabela **RECENZII** gestionează feedback-ul clienților despre produse. Informațiile stocate includ id_recenzie (cheie primară), id_produș (cheie străină către tabela **PRODUSE**), id_client (cheie străină către tabela **CLIENTI**), rating (o valoare între 1 și 5), comentariu (text opțional) și data_recenzie (data la care a fost lăsată recenzia). Relațiile sunt de tip many-to-one cu tabelele **PRODUSE** și **CLIENTI**.

COMENZI

Tabela **COMENZI** înregistrează comenzile plasate de clienți. Aceasta include id_comanda (cheie primară), id_client (cheie străină către tabela CLIENTI), id_magazin (cheie străină către tabela MAGAZIN), data_comanda (data plasării comenzii) și total (valoarea totală a comenzii). Relațiile sunt de tip many-to-one cu tabelele **CLIENTI** și **MAGAZIN**.

RETURURI

Tabela **RETURURI** înregistrează returnările produselor. Coloanele includ id_retur (cheie primară), id_produș (cheie străină către tabela PRODUSE), data_retur (data la care a fost returnat produsul) și motiv_retur (motivul returnării). Relația este de tip many-to-one cu tabela **PRODUSE**.

PROMOȚII

Tabela **PROMOȚII** stochează detalii despre reducerile oferite pentru produse, cum ar fi id_promotie (cheie primară), id_produș (cheie străină către tabela PRODUSE), discount (valoarea procentuală a reducerii), data_inceput (data începerii promoției) și data_sfarsit (data încheierii). Relația este de tip many-to-one cu tabela **PRODUSE**.

STOC

Tabela **STOC** monitorizează cantitățile disponibile pentru produse. Informațiile includ id_stoc (cheie primară), id_produș (cheie străină către tabela PRODUSE), cantitate_disponibila (numărul de unități disponibile) și data_actualizare (data ultimei actualizări a stocului). Relația este de tip many-to-one cu tabela **PRODUSE**.

Această structură de baze de date permite o gestionare eficientă și bine organizată a datelor pentru un sistem de management al magazinelor, integrând toate elementele esențiale într-un mod logic și scalabil.

4.Codul SQL

Crearea tabelelor

Tabelele vor fi create dupa modelul

urmator: CREATE TABLE Magazin (
 id_magazin NUMBER PRIMARY KEY,
 nume VARCHAR(255) NOT NULL,
 adresa VARCHAR(255),
 nr_telefon VARCHAR(20)

);

```
C##RADU> CREATE TABLE Magazin (
    id_magazin NUMBER PRIMARY KEY,
    nume VARCHAR(255) NOT NULL,
    adresa VARCHAR(255),
    nr_telefon VARCHAR(20)
)
[2025-01-12 14:14:37] completed in 12 ms
[2025-01-12 14:14:37] transaction committed: @proiect radu tene.sql
```

Figura

2. Crearea tabelului

Magazin

Inserarea datelor

În tabel vom adauga date folosind urmatorul model:

```
INSERT INTO Magazin (id_magazin, nume, adresa, nr_telefon)
VALUES (1, 'Magazin Bricolaj', 'Soseaua Giurgiului 31',
'024192665');
```

```
C##RADU> INSERT INTO Magazin (id_magazin, nume, adresa, nr_telefon)
VALUES (1, 'Magazin Bricolaj', 'Soseaua Giurgiului 31', '024192665')
[2025-01-12 14:16:26] 1 row affected in 34 ms
[2025-01-12 14:16:26] transaction committed: @proiect radu tene.sql
```

Figura 3

Interogari de date:

Interogari de baza

1. Inserarea unui client nou in baza de date

```
INSERT INTO Clienti (id_client, nume, prenume, adresa, cnp, id_magazin)
VALUES (1, 'Chesnoiu', 'Bogdan', 'Str. Calusei, Bucuresti',
'1780822166912', 1);
```

ID_CLIENT	NUME	PRENUME	ADRESA	CNP	ID_MAGAZIN
1	1 Chesnoiu	Bogdan	Str. Calusei, Bucuresti	1780822166912	1

'1780822166912', 1);

Figura 4.

2. Afișează tot conținutul din tabela
Magazin: `SELECT * FROM Magazin;`

ID_MAGAZIN	NUME	ADRESA	NR_TELEFON
1	1 Magazin Bricolaj	Soseaua Giurgiului 31	024192665

Figura 5.

3. Afișează toate detaliile despre produsele dintr-un anumit magazin (exemplu: ID-ul magazinului este 1):

```
SELECT * FROM Produse WHERE id_magazin = 1;
```

	ID_PRODUS	NUME	PRET	ID_MAGAZIN
1	1	Ciocan	35.99	1
2	2	Feronerie	12.50	1
3	3	Vopsea lavabila	45.80	1
4	4	Rola vopsea	9.99	1
5	5	Surubelnita	25.60	1
6	6	Lampa de birou	89.90	1

Figura 6.

4. Afișează intreg numele clienților:

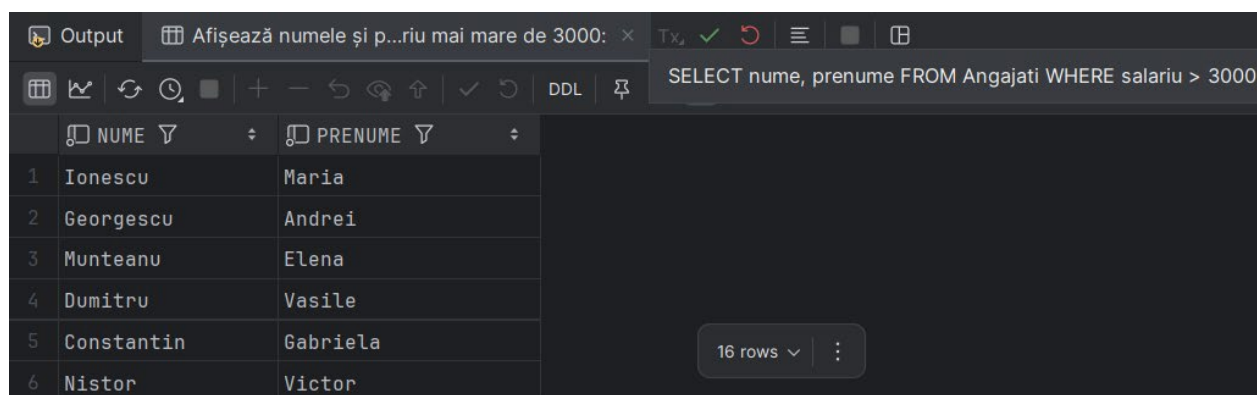
```
SELECT nume, prenume FROM  
Cienti;
```

	NUME	PRENUME
1	Chesnoiu	Bogdan
2	Ionescu	Maria
3	Vasilescu	George
4	Georgescu	Ana
5	Popa	Mihai
6	Dumitru	Elena

Figura 7.

5. Afișează numele și prenumele angajaților care au un salariu mai mare de 3000:

```
SELECT nume,  
prenume FROM  
Angajati WHERE  
salariu > 3000;
```



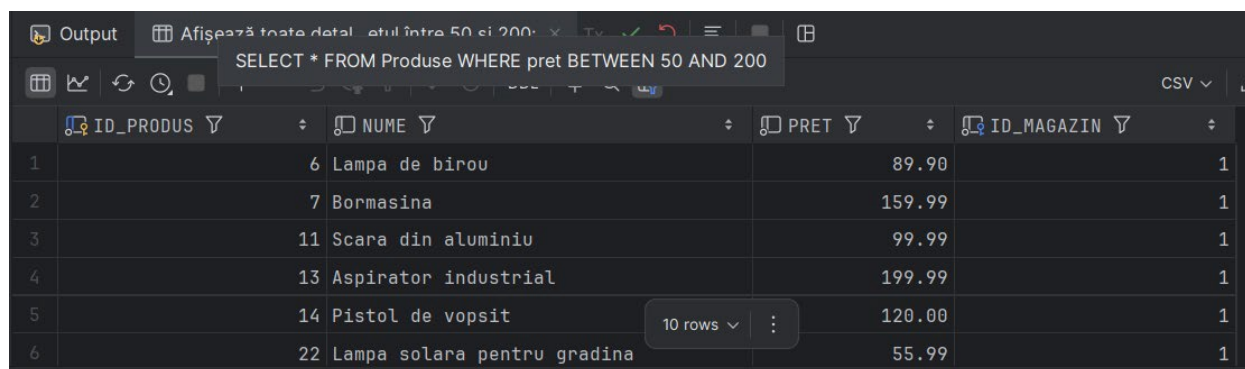
The screenshot shows a database query interface with a toolbar and a results table. The query is: `SELECT nume, prenume FROM Angajati WHERE salariu > 3000`. The results table has two columns: NUME and PRENUME. It displays 6 rows of data.

	NUME	PRENUME
1	Ionescu	Maria
2	Georgescu	Andrei
3	Munteanu	Elena
4	Dumitru	Vasile
5	Constantin	Gabriela
6	Nistor	Victor

Figura 8.

6. Afișează toate detaliile produselor care au prețul între 50 și 200: `SELECT *`

```
FROM Produse  
WHERE pret BETWEEN 50 AND 200;
```



The screenshot shows a database query interface with a toolbar and a results table. The query is: `SELECT * FROM Produse WHERE pret BETWEEN 50 AND 200`. The results table has four columns: ID_PRODUS, NUME, PRET, and ID_MAGAZIN. It displays 6 rows of data.

	ID_PRODUS	NUME	PRET	ID_MAGAZIN
1	6	Lampa de birou	89.90	1
2	7	Bormasina	159.99	1
3	11	Scara din aluminiu	99.99	1
4	13	Aspirator industrial	199.99	1
5	14	Pistol de vopsit	120.00	1
6	22	Lampa solara pentru gradina	55.99	1

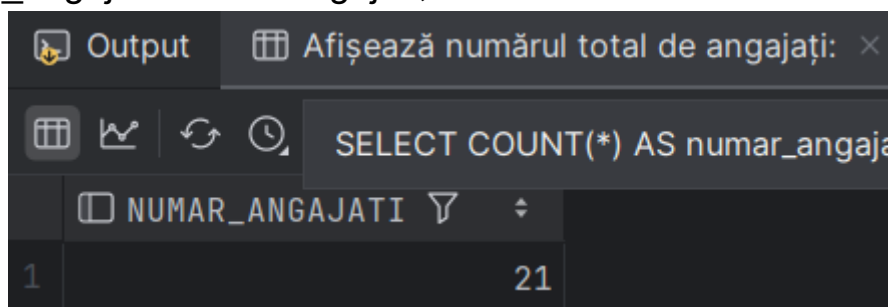
Figura 9.

Interogari cu functii speciale

7. Afișează numărul total de angajați:

```
SELECT COUNT(*) AS
```

```
numar_angajati FROM Angajati;
```



The screenshot shows a database query result in a dark-themed interface. The title bar of the window reads "Afișează numărul total de angajați: ×". The query text in the editor is "SELECT COUNT(*) AS numar_angajati FROM Angajati;". The result table has one column named "NUMAR_ANGAJATI" and one row with the value 21.

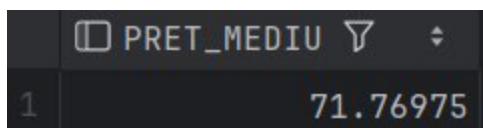
	NUMAR_ANGAJATI
1	21

Figura 10.

8. Afișează prețul mediu al

```
produselor: SELECT AVG(pret)
```

```
AS pret_mediu FROM Produse;
```



The screenshot shows a database query result in a dark-themed interface. The result table has one column named "PRET_MEDIU" and one row with the value 71.76975.

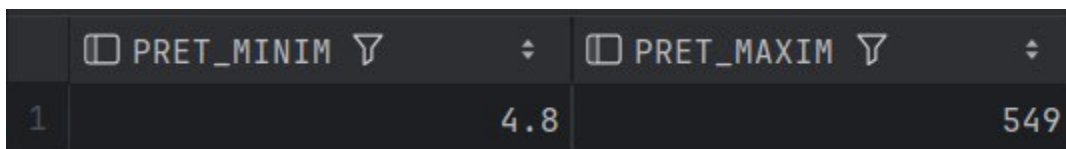
	PRET_MEDIU
1	71.76975

Figura 11.

9. Afișează prețul minim și maxim al produselor:

```
SELECT MIN(pret) AS pret_minim, MAX(pret) AS
```

```
pret_maxim FROM Produse;
```

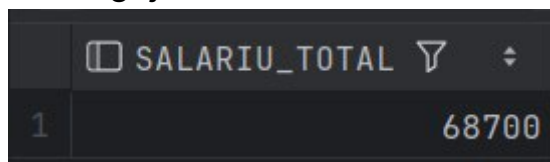


The screenshot shows a database query result in a dark-themed interface. The result table has two columns: "PRET_MINIM" and "PRET_MAXIM". The first row shows the values 4.8 and 549.

	PRET_MINIM	PRET_MAXIM
1	4.8	549

Figura 12.

10. Afișează suma totală a salariilor angajaților: `SELECT SUM(salariu) AS salariu_total FROM Angajati;`

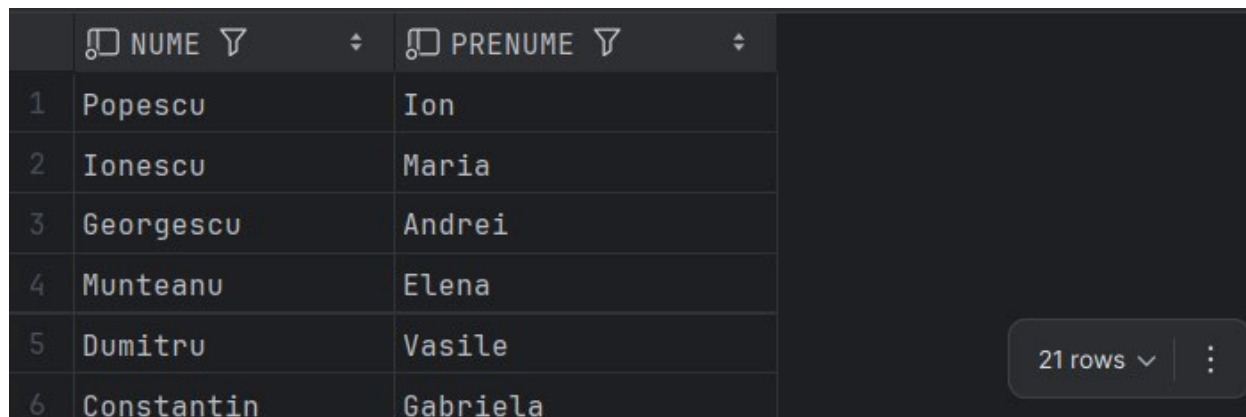


	SALARIU_TOTAL
1	68700

Figura 13.

Interogari cu conditii complexe:

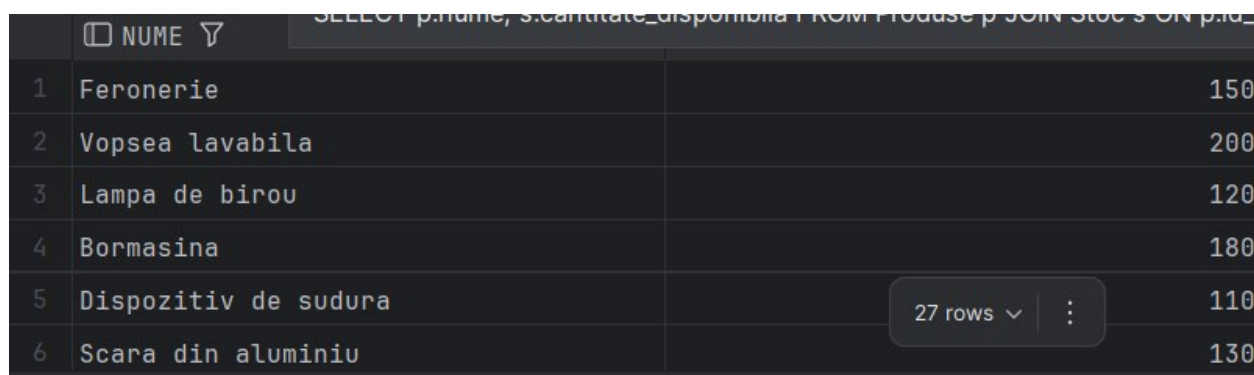
11. Afișează angajații care lucrează în magazinul cu ID-ul 1: `SELECT nume, prenume FROM Angajati WHERE id_magazin = 1;`



	NUME	PRENUME
1	Popescu	Ion
2	Ionescu	Maria
3	Georgescu	Andrei
4	Munteanu	Elena
5	Dumitru	Vasile
6	Constantin	Gabriela

Figura 14.

12. Afișează produsele care au stoc disponibil mai mare de 100: `SELECT p.num, s.cantitate_disponibila`
`FROM Produse p`
`JOIN Stoc s ON p.id_produ =`
`s.id_produ WHERE s.cantitate_disponibila >`
`100;`

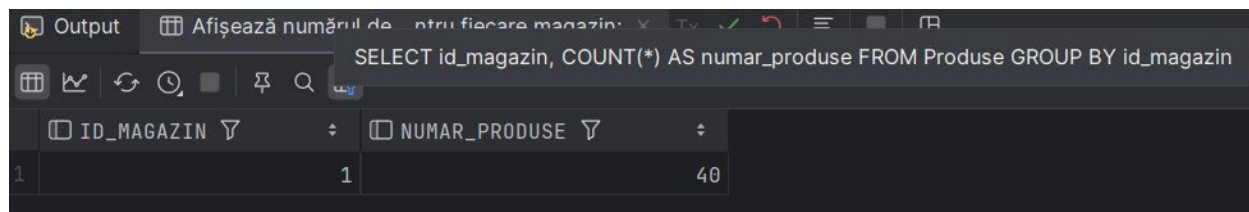


	NUME	
1	Feronerie	150
2	Vopsea lavabila	200
3	Lampa de birou	120
4	Bormasina	180
5	Dispozitiv de sudura	110
6	Scara din aluminiu	130

Figura 15.

Interogări cu **GROUP BY** și **HAVING**:

13. Afișează numărul de produse pentru fiecare magazin: `SELECT id_magazin, COUNT(*) AS`
`numar_produce FROM Produse`
`GROUP BY id_magazin;`



ID_MAGAZIN	NUMAR_PRODUSE
1	40

Figura 16.

Interogari distinct.

14. Afișează toate numele distincte ale companiilor furnizorilor: `SELECT DISTINCT nume_companie FROM Furnizori;`

	NUME_COMPANIE
1	Tech Solutions
2	Green Energy
3	Furnizor IT
4	Tech Distributors
5	ElektroCom
6	SoftWareHub

Figura 17.

Interogari cu JOIN

15. Afișează produsele împreună cu stocurile lor: `SELECT p.nume, s.cantitate_disponibila FROM Produse p JOIN Stoc s ON p.id_produș = s.id_produș;`

	NUME	CANTITATE_DISPONIBILA
1	Ciocan	100
2	Feronerie	150
3	Vopsea lavabila	200
4	Rola vopsea	50
5	Surubelnita	75
6	Lampa de birou	120

Figura 18.

Interogari cu UNION

16. Afișează salariile tuturor angajaților și prețurile produselor: `SELECT salariu AS valoare
FROM Angajati
UNION
SELECT pret
FROM Produse;`

	VALOARE
1	3000.00
2	3500.00
3	3200.00
4	4200.00
5	3100.00
6	3400.00

Figura 19.

17. Creează un VIEW pentru a vizualiza clienții și magazinele lor: CREATE VIEW ClientiSiMagazine AS
SELECT c.num AS nume_client, c.prenume AS prenume_client,
m.num AS nume_magazin
FROM Clienti c
JOIN Magazin m ON c.id_magazin = m.id_magazin;

Vizualizează VIEW-ul creat:

SELECT * FROM ClientiSiMagazine;

	NUME_CLIENT	PRENUME_CLIENT	NUME_MAGAZIN
1	Chesnoiu	Bogdan	Magazin Bricolaj
2	Petrescu	Loredana	Magazin Bricolaj
3	Vasilescu	George	Magazin Bricolaj
4	Georgescu	Ana	Magazin Bricolaj
5	Popa	Mihai	Magazin Bricolaj
6	Dumitru	Elena	Magazin Bricolaj

Figura 20.

18. Afișează numele clienților în ordine
alfabetică: `SELECT nume, prenume`
`FROM Clienti`
`ORDER BY nume ASC, prenume ASC;`

	NUME	PRENUME
1	Balan	Vasile
2	Chesnoiu	Bogdan
3	Cojocaru	Simona
4	Constantin	Ioana
5	Cozma	Delia
6	Craciun	Ioana

Figura 21.

Instructiunea pentru realizarea produsului cartezian

19. Afișează clienții împreună cu furnizorii (produs cartezian):
`SELECT c.nume AS nume_client, c.prenume AS`
`prenume_client, f.nume_companie AS furnizor`
`FROM Clienti c, Furnizori f;`

	NUME_CLIENT	PRENUME_CLIENT	FURNIZOR
1	Chesnoiu	Bogdan	Tech Solutions
2	Ionescu	Maria	Tech Solutions
3	Vasilescu	George	Tech Solutions
4	Georgescu	Ana	Tech Solutions
5	Popa	Mihai	
6	Dumitru	Elena	Tech Solutions

Figura 22.

Join natural

20. Afișează detaliile produselor și stocurilor unde valorile coloanei comune coincid:

```
SELECT *
FROM Produse
NATURAL JOIN Stoc;
```

	ID_PRODUS	NUME	PRET	ID_MAGAZIN	ID_STOC	CANTITATE_DIS
1	1	Ciocan	35.99	1	1	
2	ID_PRODUS: number		12.50	1	2	
3	3	Vopsea lavabila	45.80	1	3	
4	4	Rola vopsea	9.99	1	4	
5	5	Surubelnita		1	5	
6	6	Lampa de birou		1	6	

Figura 23.

Left Join

21. Afișează produsele și, dacă există, stocurile lor (unde stocurile lipsesc, coloanele vor avea NULL):

```
SELECT p.nume AS produs,
s.cantitate_disponibila FROM Produse p
LEFT JOIN Stoc s ON p.id_produs = s.id_produs;
```

	PRODUS	CANTITATE_DISPONIBILA
1	Ciocan	100
2	Feronerie	150
3	Vopsea lavabila	200
4	Rola vopsea	50
5	Surubelnita	75
6	Lampa de birou	120

Figura 24.

Concatenare text

22. Afișează detalii despre clienți și magazinul lor într-o propoziție: `SELECT 'Clientul ' || c.numc || ' ' || c.prenume || ' a cumpărat din magazinul ' || m.numc AS detalii`
`FROM Clienti c`
`JOIN Magazin m ON c.id_magazin = m.id_magazin;`

	DETAII
1	Clientul Chesnoiu Bogdan a cumpărat din magazinul Magazin Bricolaj
2	Clientul Petrescu Loredana a cumpărat din magazinul Magazin Bricolaj
3	Clientul Vasilescu George a cumpărat din magazinul Magazin Bricolaj
4	Clientul Georgescu Ana a cumpărat din magazinul Magazin Bricolaj
5	Clientul Popa Mihai a cumpărat din magazinul Magazin
6	Clientul Dumitru Elena a cumpărat din magazinul Magazin Bricolaj

Figura 25.



Bibliografie

<https://docs.oracle.com/en/database/oracle/oracle-database/21/sqlrf/index.html>

https://en.wikipedia.org/wiki/Oracle_SQL_Developer

<https://www.jetbrains.com/idea/>