

Proiect

-Baze de Date-

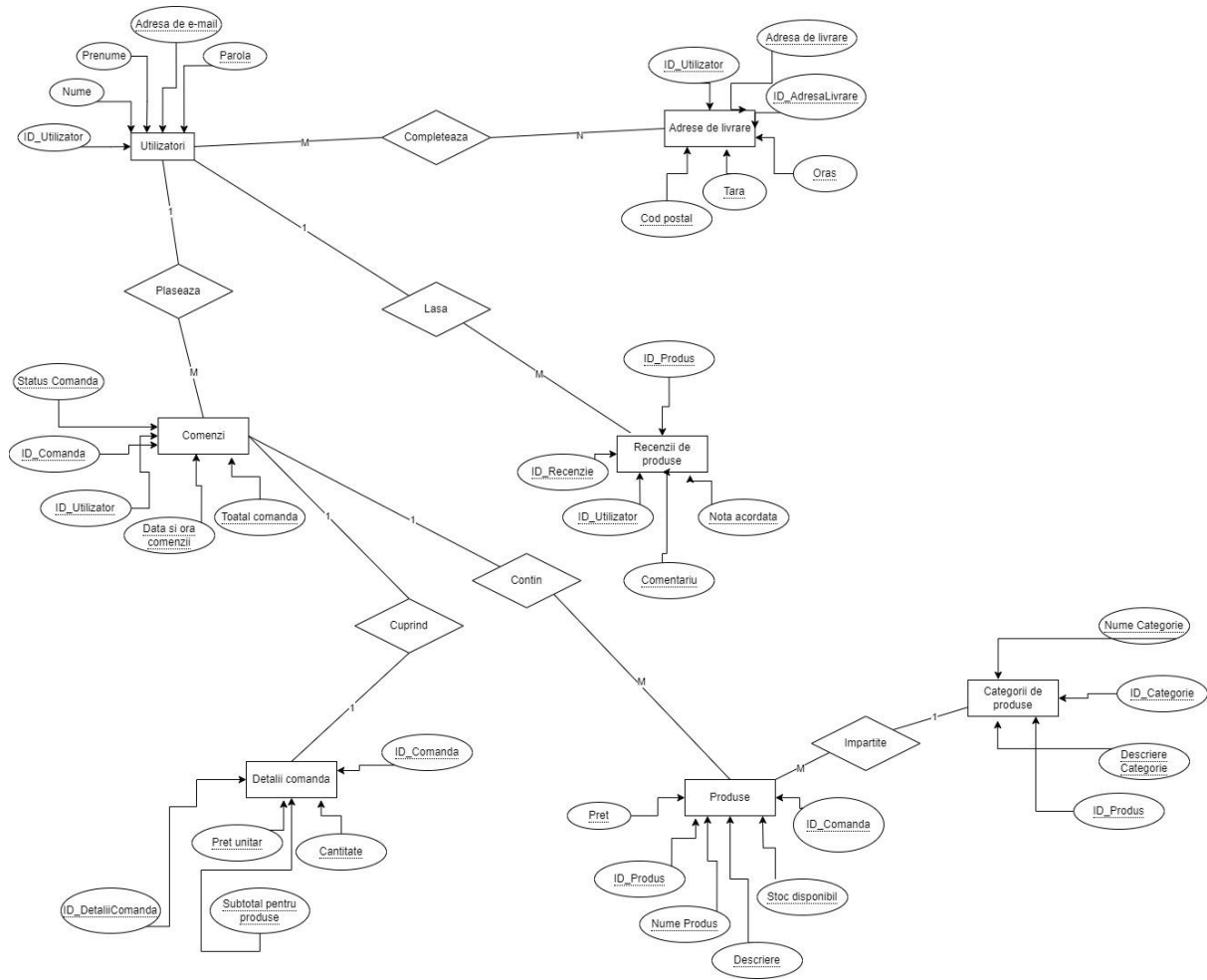
Safciu Alexandru Radu

Descriere tema

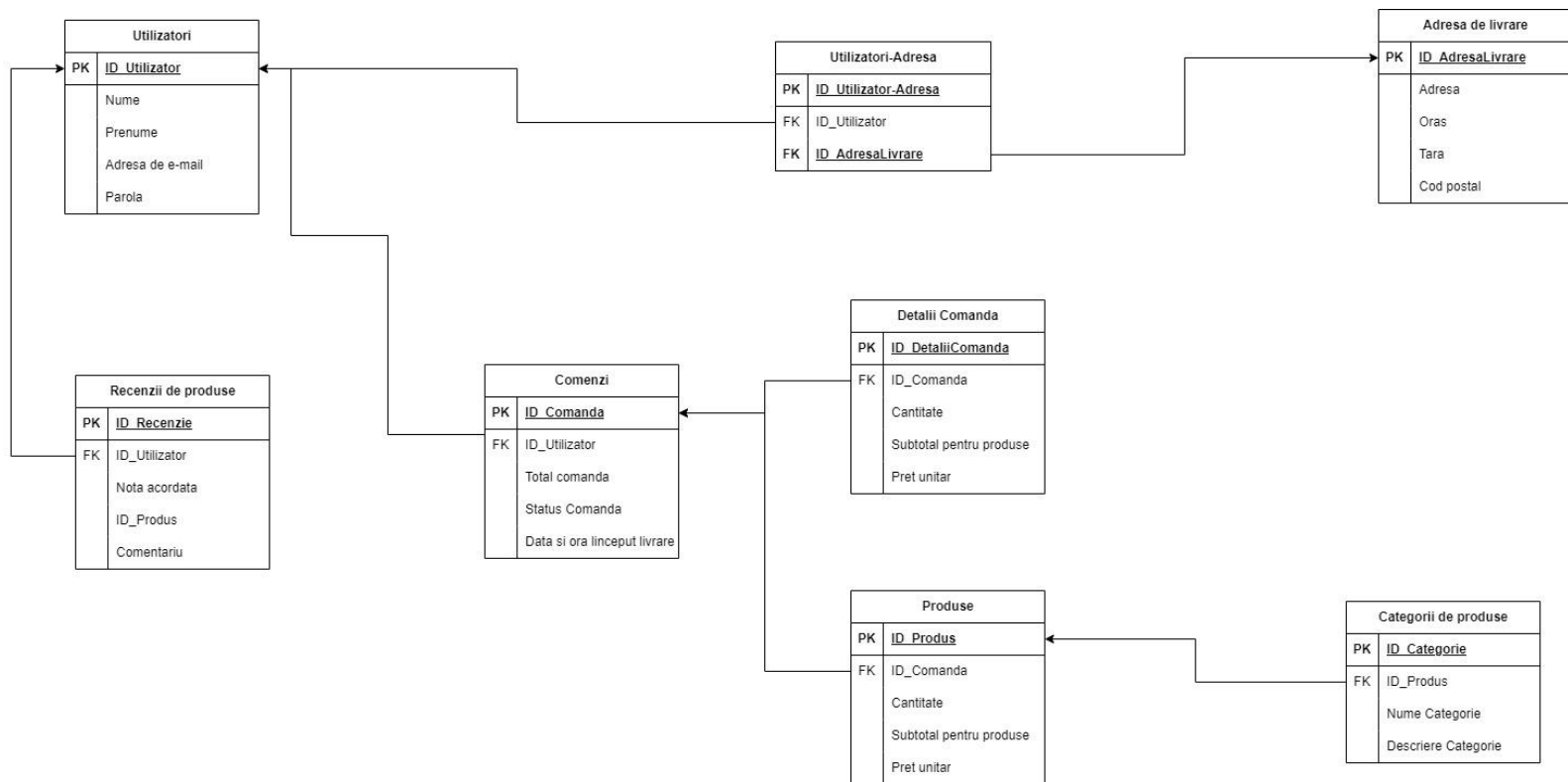
Pentru acest tema de casa am proiectat si implementat o baza de date care descrie un magazine online simplu.

Magazinul are utilizatori care pot comanda produse din diferite categorii care sa fie livrate la domiciliu. De asemenea acestia pot lasa recenzii la produsele pe care le au cumparat.

Diagrama ER



Schema Relatională



Crearea tabelelor

Am creat tabelele descrise in diagrama ER si am pus constrangerile si relatiile dintre tabele cu ajutorul freign keys regasite in Schema Relatională.

Query-urile pentru crearea tabelelor este [pe github-ul meu](#)

Inserarea Datelor

Am continuat prin a popula tabelele, codul este tot [pe github-ul meu](#).

Query

Query History

↗

Scratch Pad ×

1

2

3

4

5

6

SELECT * FROM Utilizatori;

Data Output

Messages

Notifications

⌵

📄

⌵

📋

⌵

🗑

🗄

⬇

📈

	id_utilizator [PK] integer	nume character varying (50)	prenume character varying (50)	adresa_email character varying (100)	parola character varying (255)
1	1	Safciu	Alex	alexradusaf@gmail.com	parola123
2	2	Parker	Peter	peter.parker@gmail.com	parola456
3	3	Prime	Optimus	optimus.prime@yahoo.com	parola789
4	4	Taylor	Emily	emily.taylor@yahoo.com	parolaabc
5	5	Anderson	Michael	michael.anderson@gmail.com	parolaxyz

Query

Query History

1

2

3

4

5

6

SELECT * FROM UtilizatoriAdresa;

Data Output

Messages

Notifications

⌵

📄

⌵

📋

⌵

🗑

🗄

⬇

📈

id_utilizator [PK] integer	id_adresalivrare [PK] integer
-------------------------------	----------------------------------

Query

Query History

Scratch Pad

1

2

3

4

5

6

SELECT * FROM AdreseLivrare ;

Data Output

Messages

Notifications

	id_adresalivrare [PK] integer	oras character varying (50)	tara character varying (50)	cod_postal character varying (20)	adresa_livrare text
1	1	Craiova	Romania	900101	Bulevardul Carol, Nr. 7, Bl D3, ap 12
2	2	Bucuresti	Romania	010101	Strada Florilor, Nr. 1
3	3	Cluj-Napoca	Romania	400001	Bulevardul Principal, Nr. 10, Bl Y, ap 1
4	4	Timisoara	Romania	300101	Aleea Paradisului, Nr. 5, Bl a12, ap 51
5	5	Iasi	Romania	700501	Piata Independentei, Nr. 2

Query

Query History

↗

S

1

2

3

4

5

6

SELECT * FROM RecenziiProduse ;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

	id_recenzie [PK] integer	id_utilizator integer	id_produs integer	comentariu text	nota_acordata integer
1	1	1	154	Produs excelent, sunt foarte mulțumit!	5
2	2	2	366	Calitatea produsului lasă de dorit.	2
3	3	4	288	Recomand cu căldură acest produs!	4
4	4	3	4752	Nu funcționează așa cum mă așteptam.	3
5	5	5	5010	Design modern și performanță remarcabilă!	5



```
1 SELECT * FROM Comenzi ;
2
3
4
5
6
```



	id_comanda [PK] integer	id_utilizator integer	status_comanda character varying (50)	data_ora_livrarii timestamp without time zone	total_comanda numeric (10,2)
1	1	1	In procesare	2024-01-17 12:30:00	250.00
2	2	3	Expediată	2024-01-18 09:45:00	120.50
3	3	2	Livrată	2024-01-19 15:20:00	350.75
4	4	4	In procesare	2024-01-20 10:10:00	80.90
5	5	5	Expediată	2024-01-21 14:00:00	420.25

Query		Query History				
1	SELECT * FROM DetaliiComanda;					
2						
3						
4						
5						
6						

Data Output		Messages		Notifications	
	id_detaliicomanda [PK] integer		id_comanda integer		
	cantitate integer		pret_unitar numeric (10,2)		subtotal numeric (10,2)
	subtotal numeric (10,2)				
1	1		1		250.00
2	2		3		120.50
3	3		2		350.75
4	4		4		80.90
5	5		5		420.25

Query Query History

```
1 SELECT * FROM CategoriProduse;  
2  
3  
4  
5  
6
```

Data Output Messages Notifications



	id_categorie [PK] integer	id_produs integer	descriere_categorie text	nume_categorie character varying (50)
1	1	1	Electronice	Laptopuri
2	2	2	Electronice	Telefoane
3	3	3	Periferice	Mouse-uri
4	4	4	Accesorii	Căști
5	5	5	Electronice	Monitoare

Exemple operatii modificare date

UPDATE Utilizatori

SET Prenume = 'Radu'

WHERE ID_Utilizator = 1;

Query Query History Scratch Pad ✕

1 SELECT * FROM Utilizatori
2 ORDER BY id_utilizator
3
4
5
6
7
8

Data Output Messages Notifications

	id_utilizator [PK] integer	nume character varying (50)	prenume character varying (50)	adresa_email character varying (100)	parola character varying (255)
1	1	Safciu	Radu	alexradusaf@gmail.com	parola123
2	2	Parker	Peter	peter.parker@gmail.com	parola456
3	3	Prime	Optimus	optimus.prime@yahoo.com	parola789
4	4	Taylor	Emily	emily.taylor@yahoo.com	parolaabc
5	5	Anderson	Michael	michael.anderson@gmail.com	parolaxyz

UPDATE Produse

SET Stoc_Disponibil = Stoc_Disponibil + 10

WHERE ID_Produs = 2;

Query

Query History

Scratch Pad

1

2

3

4

5

6

7

8

9

```

SELECT * FROM Produce
ORDER BY id_produc

```

Data Output

Messages

Notifications

+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

	id_produc [PK] integer	id_comanda integer	pret numeric (10,2)	nume_produc character varying (100)	descriere text	stoc_disponibil integer
1	1	[null]	4200.00	Laptop	Laptop performant pentru utilizare zilnica	50
2	2	[null]	550.00	Telefon	Telefon inteligent cu cameră de înaltă rezoluție	40
3	3	[null]	48.20	Mouse	Mouse optic ergonomic pentru gaming	100
4	4	[null]	26.97	Căști Wireless	Căști Bluetooth cu sunet de calitate	80
5	5	[null]	210.12	Monitor	Monitor LED cu rezoluție Full HD	45

Exemple operatii stergere date

DELETE FROM RecenziiProduce

WHERE ID_Recenzie = 3;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

	id_recenzie [PK] integer	id_utilizator integer	id_produc integer	comentariu text	nota_acordata integer
1	1	1	154	Produs excelent, sunt foarte mulțumit!	5
2	2	2	366	Calitatea produsului lasă de dorit.	2
3	4	3	4752	Nu funcționează așa cum mă așteptam.	3
4	5	5	5010	Design modern și performanță remarcabilă!	5

(id_produs din aceasta imagine este gresit, l-am corectat dar nu am mai facut ss-uri pentru problemele prin care deja trecusem.)

Exemple de operații de modificare sau ștergere folosind subinterogări:

Actualizare - Actualizarea statusului comenzii la 'Finalizată' pentru un utilizator anume:

UPDATE Comenzi

SET Status_Comanda = 'Finalizată'

WHERE ID_Utilizator = (SELECT ID_Utilizator FROM Utilizatori
WHERE Nume = 'Safciu' AND Prenume = 'Radu');

Data Output Messages Notifications						
	id_comanda [PK] integer	id_utilizator integer	status_comanda character varying (50)	data_ora_livrarii timestamp without time zone	total_comanda numeric (10,2)	
1	1	1	In procesare	2024-01-17 12:30:00	250.00	
2	2	3	Expediată	2024-01-18 09:45:00	120.50	
3	3	2	Livrată	2024-01-19 15:20:00	350.75	
4	4	4	In procesare	2024-01-20 10:10:00	80.90	
5	5	5	Expediată	2024-01-21 14:00:00	420.25	

Data Output Messages Notifications						
	id_comanda [PK] integer	id_utilizator integer	status_comanda character varying (50)	data_ora_livrarii timestamp without time zone	total_comanda numeric (10,2)	
1	1	1	Finalizată	2024-01-17 12:30:00	250.00	
2	2	3	Expediată	2024-01-18 09:45:00	120.50	
3	3	2	Livrată	2024-01-19 15:20:00	350.75	
4	4	4	In procesare	2024-01-20 10:10:00	80.90	
5	5	5	Expediată	2024-01-21 14:00:00	420.25	

Actualizare - Actualizarea prețului produselor cu o anumită categorie:

UPDATE Produse

SET Pret = Pret * 1.1

WHERE ID_Produs IN (SELECT ID_Produs FROM CategoriProduse
WHERE Nume_Categorie = 'Laptopuri');

Data Output Messages Notifications							
	id_produs [PK] integer	id_comanda integer	pret numeric (10,2)	nume_produs character varying (100)	descriere text	stoc_disponibil integer	
1	1	1	4200.00	Laptop	Laptop performant pentru utilizare zilnica	50	
2	2	2	550.00	Telefon	Telefon inteligent cu cameră de înaltă rezoluție	40	
3	3	3	48.20	Mouse	Mouse optic ergonomic pentru gaming	100	
4	4	4	26.97	Căști Wireless	Căști Bluetooth cu sunet de calitate	80	
5	5	5	210.12	Monitor	Monitor LED cu rezoluție Full HD	45	

	id_produs [PK] integer	id_comanda integer	pret numeric (10,2)	nume_produs character varying (100)	descriere text	stoc_disponibil integer	
1	1	1	4620.00	Laptop	Laptop performant pentru utilizare zilnica	50	
2	2	2	550.00	Telefon	Telefon inteligent cu cameră de înaltă rezoluție	40	
3	3	3	48.20	Mouse	Mouse optic ergonomic pentru gaming	100	
4	4	4	26.97	Căști Wireless	Căști Bluetooth cu sunet de calitate	80	
5	5	5	210.12	Monitor	Monitor LED cu rezoluție Full HD	45	

Ștergere - Ștergerea tuturor recenziilor asociate produselor cu stoc disponibil mai mic de 49:

DELETE FROM RecenziiProduse

WHERE ID_Produs IN (SELECT ID_Produs FROM Produse WHERE
Stoc_Disponibil < 49);

Data Output

Messages

Notifications

	id_recenzie [PK] integer	id_utilizator integer	id_produs integer	comentariu text	nota_acordata integer
1	1	1	1	Produs excelent, sunt foarte mulțumit!	5
2	2	2	2	Calitatea produsului lasă de dorit.	2
3	4	3	4	Nu funcționează așa cum mă așteptam.	3
4	5	5	5	Design modern și performanță remarcabilă!	5

Data Output

Messages

Notifications

	id_recenzie [PK] integer	id_utilizator integer	id_produs integer	comentariu text	nota_acordata integer
1	1	1	1	Produs excelent, sunt foarte mulțumit!	5
2	4	3	4	Nu funcționează așa cum mă aștept...	3

Normalizarea relațiilor prezente

Normalizarea este procesul de proiectare al bazelor de date pentru a minimiza redundanța și a asigura consistența datelor. Există mai multe forme normale, iar fiecare urmărește anumite reguli de organizare a datelor pentru a preveni anumite anomalii.

O situație care nu respectă 1NF ar putea fi dacă ai avea o coloană într-o tabelă care conține o listă de valori separate de virgule, cum ar fi o listă de adrese într-un singur câmp.

-- Nu respectă 1NF

CREATE TABLE Utilizatori (

ID_Utilizator SERIAL PRIMARY KEY,

Nume VARCHAR(50) NOT NULL,

Prenume VARCHAR(50) NOT NULL,

Adrese_Livrare VARCHAR(255) NOT NULL -- Exemplu de câmp cu mai multe valori

);

O situație care nu respectă 2NF ar putea fi dacă ai o cheie primară compusă și există dependențe parțiale.

-- Nu respectă 2NF

```
CREATE TABLE DetaliiComanda (  
    ID_DetaliiComanda SERIAL PRIMARY KEY,  
    ID_Comanda INT,  
    Nume_Produs VARCHAR(100) NOT NULL,  
    Pret_Unitar DECIMAL(10, 2),  
    FOREIGN KEY (ID_Comanda) REFERENCES  
Comenzi(ID_Comanda)  
);
```

În acest caz, Nume_Produs și Pret_Unitar depind doar de o parte a cheii primare (ID_Comanda).

O situație care nu respectă 3NF ar putea fi dacă există dependențe tranzitive.

-- Nu respectă 3NF

```
CREATE TABLE Produse (  
    ID_Produs SERIAL PRIMARY KEY,  
    ID_Comanda INT,  
    Nume_Produs VARCHAR(100) NOT NULL,  
    Descriere TEXT,
```

```
Stoc_Disponibil INT,  
Pret DECIMAL(10, 2),  
FOREIGN KEY (ID_Comanda) REFERENCES  
Comenzi(ID_Comanda)  
);
```

Pret depinde de ID_Comanda, dar și de Nume_Produs. Într-o formă normală corectă, Pret ar trebui să depindă doar de cheia primară.

Din pacate, prima oara cand am citit cerinca credeam ca trebuie sa facem una din cele 3 normalizari, asa ca acest cod este in baza de date:

Creează o nouă tabelă pentru informații despre produse:

```
CREATE TABLE InformatiiProduse (  
ID_Produs SERIAL PRIMARY KEY,  
Nume_Produs VARCHAR(100) NOT NULL,  
Descriere TEXT,  
Stoc_Disponibil INT  
);
```

Actualizează tabela Produse pentru a include doar cheile primare necesare și adaugă o cheie străină către InformatiiProduse:

```
ALTER TABLE Produse  
ADD COLUMN ID_InfoProdus INT,  
ADD FOREIGN KEY (ID_InfoProdus) REFERENCES  
InformatiiProduse(ID_Produs);
```


Transferă datele relevante din Produse la InformatiiProduse:

```
INSERT INTO InformatiiProduse (ID_Produs, Nume_Produs,  
Descriere, Stoc_Disponibil)  
SELECT ID_Produs, Nume_Produs, Descriere, Stoc_Disponibil  
FROM Produse;
```

Elimină coloanele redundante din Produse:

```
ALTER TABLE Produse  
DROP COLUMN Nume_Produs,  
DROP COLUMN Descriere,  
DROP COLUMN Stoc_Disponibil;
```

Cu aceste modificări, am separat informațiile despre produse într-o nouă tabelă InformatiiProduse, iar tabela Produse conține acum doar cheile străine necesare. Astfel, baza de date ar trebui să respecte a doua formă normală (2NF).

Interogari:

Exemple interogari innerjoin+outerjoin:

Inner Join - Obține informații despre produsele dintr-o comandă:

```
SELECT DetaliiComanda.ID_Comanda, DetaliiComanda.Cantitate,  
InformatiiProduse.Nume_Produs  
FROM DetaliiComanda
```

INNER JOIN InformatiiProduse ON DetaliiComanda.ID_Comanda = InformatiiProduse.ID_Produs;

Data Output Messages Notifications			
	id_comanda integer	cantitate integer	nume_produc character varying (100)
1	1	2	Laptop
2	3	5	Mouse
3	2	1	Telefon
4	4	3	Căști Wireless
5	5	2	Monitor

Left Outer Join - Afișează toate adresele de livrare și utilizatorii asociați (chiar și cei fără adrese de livrare):

SELECT Utilizatori.Nume, Utilizatori.Prenume, AdreseLivrare.Oras
FROM Utilizatori

LEFT OUTER JOIN UtilizatoriAdresa ON Utilizatori.ID_Utilizator = UtilizatoriAdresa.ID_Utilizator

LEFT OUTER JOIN AdreseLivrare ON UtilizatoriAdresa.ID_AdresaLivrare = AdreseLivrare.ID_AdresaLivrare;

Data Output Messages Notifications			
	nume character varying (50)	prenume character varying (50)	oras character varying (50)
1	Parker	Peter	[null]
2	Anderson	Michael	[null]
3	Taylor	Emily	[null]
4	Safciu	Radu	[null]
5	Prime	Optimus	[null]

Inner Join - Obține recenziiile și numele utilizatorilor care le-au scris, împreună cu informații despre produse:

```
SELECT RecenziiProduse.Comentariu,  
RecenziiProduse.Nota_Acordata, Utilizatori.Nume, Utilizatori.Prenume,  
InformatiiProduse.Nume_Produs  
  
FROM RecenziiProduse  
  
INNER JOIN Utilizatori ON RecenziiProduse.ID_Utilizator =  
Utilizatori.ID_Utilizator  
  
INNER JOIN InformatiiProduse ON RecenziiProduse.ID_Produs =  
InformatiiProduse.ID_Produs;
```

Data Output Messages Notifications						
	comentariu text	nota_acordata integer	nume character varying (50)	prenume character varying (50)	nume_produs character varying (100)	
1	Produs excelent, sunt foarte mulțumit!	5	Safciu	Radu	Laptop	
2	Nu funcționează așa cum mă aștept...	3	Prime	Optimus	Căști Wireless	

Right Outer Join - Afișează toate categoriile de produse și produsele asociate (chiar și cele fără categorii), împreună cu informații despre produse:

```
SELECT CategoriProduse.Nume_Categorie,  
InformatiiProduse.Nume_Produs  
  
FROM CategoriProduse  
  
RIGHT OUTER JOIN InformatiiProduse ON  
CategoriProduse.ID_Produs = InformatiiProduse.ID_Produs;
```

	nume_categorie character varying (50)	nume_produș character varying (100)
1	Laptopuri	Laptop
2	Telefoane	Telefon
3	Mouse-uri	Mouse
4	Căști	Căști Wireless
5	Monitoare	Monitor

Exemple de interogări cu funcții de agregare și GROUP BY:

Afișează suma totală a comenzilor pentru fiecare utilizator:

(am adăugat încă o comandă la id utiliz 4)

```
SELECT RecenziiProduse.ID_Produs, InformatiiProduse.Nume_Produs,
AVG(RecenziiProduse.Nota_Acordata) AS MedieNota
```

```
FROM RecenziiProduse
```

```
INNER JOIN InformatiiProduse ON RecenziiProduse.ID_Produs =
InformatiiProduse.ID_Produs
```

```
GROUP BY RecenziiProduse.ID_Produs,
InformatiiProduse.Nume_Produs;
```

Data Output Messages Notifications				
	id_utilizator [PK] integer	nume character varying (50)	prenume character varying (50)	totalcomenzi numeric
1	4	Taylor	Emily	230.90
2	2	Parker	Peter	350.75
3	3	Prime	Optimus	120.50
4	1	Safciu	Radu	250.00
5	5	Anderson	Michael	420.25

Afișează media notelor acordate în recenzii pentru fiecare produs:

```
SELECT RecenziiProduse.ID_Produs, InformatiiProduse.Nume_Produs,
AVG(RecenziiProduse.Nota_Acordata) AS MedieNota
```

```
FROM RecenziiProduse
```

```
INNER JOIN InformatiiProduse ON RecenziiProduse.ID_Produs =
InformatiiProduse.ID_Produs
```

```
GROUP BY RecenziiProduse.ID_Produs,
InformatiiProduse.Nume_Produs;
```

Data Output Messages Notifications			
	id_produs integer	nume_produs character varying (100)	medienota numeric
1	4	Căști Wireless	3.0000000000000000
2	1	Laptop	4.0000000000000000

Afișează cantitatea totală de produse comandate pentru fiecare categorie:

```

SELECT CategoriiProduse.Nume_Categorie,
SUM(DetaliiComanda.Cantitate) AS CantitateTotala
FROM CategoriiProduse
INNER JOIN Produse ON CategoriiProduse.ID_Produs =
Produse.ID_Produs
INNER JOIN DetaliiComanda ON Produse.ID_Comanda =
DetaliiComanda.ID_Comanda
GROUP BY CategoriiProduse.Nume_Categorie;

```

Data Output Messages Notifications		
	nume_categorie character varying (50) 🔒	cantitatetotala bigint 🔒
1	Telefoane	1
2	Mouse-uri	5
3	Căști	3
4	Monitoare	2
5	Laptopuri	2

Exemple de subinterogări corelate:

Exemplu cu IN:

Această subinterogare verifică utilizatorii care au comenzi finalizate.

```

SELECT Nume, Prenume
FROM Utilizatori
WHERE ID_Utilizator IN (SELECT ID_Utilizator FROM Comenzi
WHERE Status_Comanda = 'Finalizată');

```

Data Output Messages Notifications		
<div> <div>+</div> <div>▼</div> <div>▼</div> <div>▼</div> <div>▼</div> <div>▼</div> <div>▼</div> <div>▼</div> </div>		
	nume character varying (50)	prenume character varying (50)
1	Safciu	Radu

Exemplu cu ANY:

Această interogare returnează numele și prenumele utilizatorilor care au plasat comenzi cu o cantitate mai mare de 2 pentru orice produs, implicând astfel cele trei tabele menționate.

```

SELECT Nume, Prenume
FROM Utilizatori
WHERE ID_Utilizator IN (
    SELECT DISTINCT ID_Utilizator
    FROM Comenzi
    WHERE ID_Comanda IN (
        SELECT ID_Comanda
        FROM DetaliiComanda
        WHERE Cantitate > 2
    )
);

```

Data Output			Messages	Notifications
	nume character varying (50) 🔒	prenume character varying (50) 🔒		
1	Taylor	Emily		
2	Parker	Peter		

Exemplu cu EXISTS:

Această interogare returnează numele și prenumele utilizatorilor care au comenzi cu o cantitate mai mare de 2 pentru oricare dintre produsele comandate, implicând cele trei tabele menționate.

```

SELECT Nume, Prenume
FROM Utilizatori U
WHERE EXISTS (
    SELECT 1
    FROM Comenzi C
    WHERE EXISTS (
        SELECT 1
        FROM DetaliiComanda DC
        WHERE DC.ID_Comanda = C.ID_Comanda
        AND DC.Cantitate > 2
    )
    AND C.ID_Utilizator = U.ID_Utilizator
);

```


Data Output Messages Notifications		
<div> <div>+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>		
	nume character varying (50) 🔒	prenume character varying (50) 🔒
1	Parker	Peter
2	Taylor	Emily

Exemplu cu ALL:

Această interogare verifică dacă utilizatorul are toate comenzile finalizate cu un total mai mare de 200 și, în același timp, are toate detaliile comenzilor cu o cantitate mai mare de 1.

SELECT Nume

FROM Utilizatori

WHERE ID_Utilizator = ALL (

 SELECT ID_Utilizator

 FROM Comenzi

 WHERE Status_Comanda = 'Finalizată' AND Total_Comanda > 200

 AND ID_Utilizator = ALL (

 SELECT ID_Utilizator

 FROM DetaliiComanda

 WHERE Cantitate > 1

)

);

ANY:

Se selectează numele categoriilor de produse care au cel puțin un produs cu un stoc disponibil mai mare de 10 unități.

```
SELECT Nume_Categorie
FROM CategoriProduce
WHERE ID_Produs = ANY (
    SELECT ID_Produs
    FROM InformatiiProduce
    WHERE Stoc_Disponibil > 10
);
```

Data Output		Messages	Notifications
	nume_categorie character varying (50)		
1	Mouse-uri		
2	Căști		

EXISTS:

Se selectează numele produselor care fac parte din categoria "Electronice" și nu au fost comandate în cantități mai mari de 5.

```
SELECT Nume_Produs
FROM InformatiiProduce
WHERE EXISTS (
    SELECT 1
```

```

FROM CategoriiProduse
WHERE Descriere_Categorie LIKE 'Electronice%'
) AND NOT EXISTS (
    SELECT 1
    FROM DetaliiComanda
    WHERE ID_Produs = InformatiiProduse.ID_Produs AND Cantitate >
5
);

```

Data Output		Messages	Notifications
<div> <div>≡+</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> </div>			
	<div> <div>nume_produs</div> <div>character varying (100)</div> <div>🔒</div> </div>		
1	Mouse		
2	Căști Wireless		
3	Monitor		
4	Telefon		
5	Laptop		

ALL:

Se selectează numele utilizatorilor care au plasat toate comenzile finalizate și au dat note mai mari de 3 în toate recenziile lor.

```

SELECT Nume
FROM Utilizatori
WHERE ID_Utilizator = ALL (
    SELECT ID_Utilizator
    FROM Comenzi
    WHERE Status_Comanda = 'Finalizată'

```

```

) AND ID_Utilizator = ALL (
    SELECT ID_Utilizator
    FROM RecenziiProduse
    WHERE Nota_Acordata > 3
);

```

Data Output	Messages	Notifications
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>		
<div> <div> <div>nume</div> <div>character varying (50)</div> <div></div> </div> </div>		

Exemple de interogări folosind funcții pe șiruri de caractere:

Exemplu de concatenare a numelui și prenumelui utilizatorilor:

```

SELECT Nume || ' ' || Prenume AS Nume_Complet
FROM Utilizatori;

```

Data Output	Messages	Notifications
<div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div>		
<div> <div> <div>nume_complet</div> <div>text</div> <div></div> </div> </div>		
1	Parker Peter	
2	Prime Optimus	
3	Taylor Emily	
4	Anderson Michael	
5	Safciu Radu	

Exemplu de extragere a primei litere din numele produselor:

```

SELECT Nume_Produs, SUBSTRING(Nume_Produs FROM 1 FOR 1)
AS Prima_Litera

```

FROM InformatiiProduse;

Data Output Messages Notifications		
	nume_produș character varying (100)	prima_litera text
1	Mouse	M
2	Căști Wireless	C
3	Monitor	M
4	Telefon	T
5	Laptop	L
Total rows: 5 of 5 Query complete 00:00:00.006		

Exemple de interogări folosind funcții pe date calendaristice:

Exemplu de adăugare a 7 zile la data livrării comenzilor:

```
SELECT ID_Comanda, Data_Ora_Livrării, Data_Ora_Livrării +  
INTERVAL '7 days' AS Noua_Data_Livrare
```

FROM Comenzi;

Data Output Messages Notifications			
	id_comanda [PK] integer	data_ora_livrării timestamp without time zone	noua_data_livrare timestamp without time zone
1	2	2024-01-18 09:45:00	2024-01-25 09:45:00
2	3	2024-01-19 15:20:00	2024-01-26 15:20:00
3	4	2024-01-20 10:10:00	2024-01-27 10:10:00
4	5	2024-01-21 14:00:00	2024-01-28 14:00:00
5	1	2024-01-17 12:30:00	2024-01-24 12:30:00
6	6	2024-01-25 14:30:00	2024-02-01 14:30:00
Total rows: 6 of 6 Query complete 00:00:00.110			

Calcularea zilei săptămânii în care a fost înregistrată fiecare recenzie:

Această interogare utilizează funcția `EXTRACT(DOW FROM Data_Inregistrare)`, care extrage ziua săptămânii (0 pentru duminică, 1 pentru luni, etc.) din data de înregistrare a recenziilor. Rezultatul este afișat în coloana "Zi_Saptamana".

```
SELECT ID_Comanda, Data_Ora_Livrarii,  
       EXTRACT(DOW FROM Data_Ora_Livrarii) AS Zi_Saptamana  
FROM Comenzi;
```

Data Output Messages Notifications			
	id_comanda [PK] integer	data_ora_livrarii timestamp without time zone	zi_saptamana numeric
1	2	2024-01-18 09:45:00	4
2	3	2024-01-19 15:20:00	5
3	4	2024-01-20 10:10:00	6
4	5	2024-01-21 14:00:00	0
5	1	2024-01-17 12:30:00	3
6	6	2024-01-25 14:30:00	4

Exemple de interogări folosind CASE:

Exemplu de utilizare a expresiei CASE pentru a evalua starea comenzilor:

```
SELECT ID_Comanda, Status_Comanda,  
       CASE
```

```

        WHEN Status_Comanda = 'Finalizată' THEN 'Comanda
Finalizata'

        WHEN Status_Comanda = 'In procesare' THEN 'Comanda In
Procesare'

        WHEN Status_Comanda = 'Livrată' THEN 'Comanda a fost
livrată'

        ELSE 'Status necunoscut'

    END AS Descriere_Status
FROM Comenzi;

```

Data Output Messages Notifications			
	id_comanda [PK] integer	status_comanda character varying (50)	descriere_status text
1	2	Expediată	Status necunoscut
2	3	Livrată	Comanda a fost livrată
3	4	In procesare	Comanda In Procesare
4	5	Expediată	Status necunoscut
5	1	Finalizată	Comanda Finalizata
6	6	In procesare	Comanda In Procesare

Exemplu de utilizare a expresiei CASE pentru a clasifica produsele în funcție de stoc:

```

SELECT Nume_Produs, Stoc_Disponibil,
CASE
    WHEN Stoc_Disponibil > 50 THEN 'Stoc Suficient'
    WHEN Stoc_Disponibil > 0 THEN 'Stoc Limitat'
    ELSE 'Stoc Epuizat'

```



```

END AS Stare_Stoc
FROM InformatiiProduce;

```

Data Output Messages Notifications			
	nume_produc character varying (100)	stoc_disponibil integer	stare_stoc text
1	Mouse	100	Stoc Suficient
2	Căști Wireless	80	Stoc Suficient
3	Monitor	45	Stoc Limitat
4	Telefon	40	Stoc Limitat
5	Laptop	50	Stoc Limitat

Vederi:

View pentru Lista Comenzilor:

```
CREATE VIEW ListaComenzi AS
```

```
SELECT Comenzi.ID_Comanda, Utilizatori.Nume, Utilizatori.Prenume,
Comenzi.Status_Comanda
```

```
FROM Comenzi
```

```
JOIN Utilizatori ON Comenzi.ID_Utilizator = Utilizatori.ID_Utilizator;
```

```
SELECT * FROM ListaComenzi;--PERMISA
```

```
UPDATE ListaComenzi
```

```
SET Status_Comanda = 'Finalizată'--NEPERMISA
```

```
WHERE ID_Comanda = 1;
```

View pentru Produsele în Stoc:

```
CREATE VIEW ProduseInStoc AS  
SELECT Nume_Produs, Stoc_Disponibil  
FROM InformatiiProduse  
WHERE Stoc_Disponibil > 0;
```

```
UPDATE ProduseInStoc  
SET Stoc_Disponibil = Stoc_Disponibil - 1 --PERMISA  
WHERE Nume_Produs = 'Laptop';
```

```
INSERT INTO ProduseInStoc (Nume_Produs, Stoc_Disponibil) --  
NEPERMISA  
VALUES ('Cameră Foto', 5);
```

View pentru Detaliile Recenziilor:

```
CREATE VIEW DetaliiRecenziiComenzi AS  
SELECT  
    DC.ID_DetaliiComanda,  
    C.ID_Comanda,  
    C.ID_Utilizator,
```

```

U.Nume AS Nume_Utilizator,
U.Prenume AS Prenume_Utilizator,
C.Status_Comanda,
C.Data_Ora_Livrarii,
C.Total_Comanda,
RP.ID_Recenzie,
RP.Comentariu,
RP.Nota_Acordata
FROM
    DetaliiComanda DC
JOIN Comenzi C ON DC.ID_Comanda = C.ID_Comanda
JOIN RecenziiProduse RP ON C.ID_Utilizator = RP.ID_Utilizator
JOIN Utilizatori U ON C.ID_Utilizator = U.ID_Utilizator;

SELECT * FROM DetaliiRecenziiComenzi;-PERMISA

-- Exemplu de încercare de actualizare a vederii (nepermis)
UPDATE DetaliiRecenziiComenzi
SET Status_Comanda = 'Finalizată'
WHERE ID_Comanda = 1;

```

-- Exemplu de încercare de inserare în vedere (nepermis)

```
INSERT INTO DetaliiRecenziiComenzi (ID_Comanda,  
Status_Comanda)
```

```
VALUES (6, 'Noua Comanda');
```

-- Exemplu de încercare de ștergere din vedere (nepermis)

```
DELETE FROM DetaliiRecenziiComenzi
```

```
WHERE ID_Comanda = 2;
```

Index:

Crearea unui index compus pe coloanele utilizate în condițiile de căutare

```
CREATE INDEX idx_AdreseLivrare_Comenzi
```

```
ON AdreseLivrare (Oras)
```

```
INCLUDE (Cod_Postal);
```

Include Cod_Postal în index, dar nu este parte din cheia de căutare.

Acest index, numit idx_AdreseLivrare_Comenzi, acoperă coloana Oras din tabela AdreseLivrare și include coloana Cod_Postal în index, dar nu este parte a cheii de căutare. Prin includerea coloanei Cod_Postal în index, putem îmbunătăți performanța pentru situațiile în care această coloană este inclusă în SELECT sau în condiții de filtrare, fără a o face o cheie de căutare strictă.