

Proiect Baze de Date

(Platforma logistica)

Realizat de Dan Dragos-Andrei

Grupa 133

Cuprins:

- 1.Descrierea modelului real,a utilității acestuia și a regulilor de funcționare.
- 2.Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului.
- 3.Descrierea entităților, incluzând precizarea cheiiprimare.
- 4.Descrierea relațiilor, incluzând precizareacardinalității acestora.
- 5.Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
- 6.Realizarea diagrameientitate-relație corespunzătoare descrierii de la punctele 3-5.
- 7.Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație .
- 8.Enumerarea schemelorrelaționalecorespunzătoare diagramei conceptuale.
- 9.Realizarea normalizării până la forma normală 3(FN1-FN3).
- 10.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
- 11.Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe care vor utiliza, în ansamblul lor, următoarele elemente:
 - subcereri sincronizateîn care intervin cel puțin 3 tabele
 - subcereri nesincronizateîn clauza FROM
 - grupări de datecu subcereri nesincronizate in care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel degrupuri(in cadrul aceleiasi cereri)
 - ordonărisi utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)
 - utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
 - utilizarea a cel puțin 1 bloc de cerere(clauza WITH)
- 13.Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.
- 15.Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele,o cerere ce utilizează operația division și o cerere care implementează analiza top-n.

1.Descrierea modelului real,a utilității acestuia și a regulilor de funcționare:

Platforma logistică este o platformă online care permite clienților să achiziționeze produse dintr-o gamă variată de categorii, să plaseze comenzi, să aleagă metode de plată și transport și să furnizeze recenzii pentru produsele achiziționate. Fiecare client are un cont personal, cu opțiuni de a-și adăuga produsele preferate, metodele preferate de plată și informații de livrare. Producătorii pot să-și promoveze produsele, iar serviciul de transport asigură livrarea sigură și la timp a comenzilor.

Utilitate:

- **Achiziționare ușoară:** Utilizatorii pot naviga prin diverse categorii de produse, vizualiza detalii și descrieri, apoi să adauge produse în coșul de cumpărături.
- **Gestionarea preferințelor:** Utilizatorii pot să-și seteze producătorii preferați și să-și stocheze informațiile de plată și de livrare pentru a face procesul de cumpărături mai rapid și mai convenabil.
- **Metode de plată flexibile:** Platforma acceptă diferite metode de plată, inclusiv carduri de credit, transferuri bancare și opțiuni de plată la livrare.
- **Monitorizarea transportului:** Utilizatorii pot urmări statusul comenzilor lor și estimarea livrării prin intermediul serviciului de transport.
- **Recenzii și feedback:** Clienții pot lăsa recenzii pentru produsele achiziționate, ceea ce ajută alți clienți să ia decizii informate în privința achizițiilor.

Reguli de funcționare:

- **Înregistrare și autentificare:** Utilizatorii trebuie să se înregistreze pentru a avea acces la funcționalitățile platformei. Autentificarea asigură securitatea și confidențialitatea datelor personale.
- **Relații și restricții:** Tabelele și relațiile dintre ele definesc modul în care datele sunt stocate și gestionate. De exemplu, o adresă trebuie să fie asociată cu un utilizator valid.
- **Integritatea datelor:** Datele precum metodele de plată și adresele trebuie să fie corecte și actualizate pentru a asigura tranzacții fără probleme și livrări corecte.
- **Securitatea datelor:** Informațiile sensibile, cum ar fi datele cardurilor de credit, trebuie să fie stocate într-un mod sigur și conform regulamentelor de securitate.
- **Managementul comenzilor:** Procesarea comenzilor trebuie să fie eficientă, de la confirmarea comenzii, până la livrare și actualizări de stoc.
- **Recenzii și rating:** Recenziile trebuie să fie autentice și să respecte regulile de limbaj și conținut. Acestea oferă feedback util producătorilor și clienților.
- **Serviciul de transport:** Procesul de livrare trebuie să fie organizat și eficient, asigurând livrarea în termen și comunicarea cu clienții în cazul întârzierilor.

2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului:

- **Chei Primare și Străine:**
 - Tabelele utilizează chei primare pentru a identifica în mod unic înregistrările. De exemplu, **id_utilizator** în tabelul **client** și **id_metoda** în tabelul **metode_plata**. Relațiile dintre tabele sunt definite prin chei străine. De exemplu, **id_client** din tabelul **producator_preferat** este o cheie străină care face referire la **id_utilizator** din tabelul **client**.
- **Validarea Datelor:**
 - Datele introduse în tabele trebuie să corespundă tipurilor și regulilor definite. De exemplu, **cvv** în tabelul **date_card** trebuie să aibă exact 4 cifre.
- **Integritate Referențială:**
 - Tabelele care au relații străine trebuie să respecte integritatea referențială. Înregistrările nu pot fi eliminate dacă sunt folosite în alte înregistrări prin cheile străine. De exemplu, un client nu poate fi șters dacă are asociată o metodă de plată în tabelul **metode_plata**.
- **Unicitatea Datelor:**
 - Anumite atribute trebuie să fie unice. De exemplu, **username** sau **email** în tabelul **client** trebuie să fie unice pentru fiecare înregistrare.
- **Gestionarea Stocurilor:**
 - Tabelul **produse** poate fi asociat cu tabelul **transport_produse** pentru a gestiona stocurile și transportul. Un produs poate fi asociat cu mai multe transporturi, dar trebuie să fie gestionat astfel încât stocurile să nu fie depășite.
- **Securitatea Datelor:**
 - Informațiile sensibile, cum ar fi datele cardurilor de credit în tabelul **date_card**, trebuie să fie protejate în mod corespunzător pentru a preveni accesul neautorizat.
- **Integritatea Datelor Personale:**
 - Informațiile personale ale clienților, cum ar fi adresele din tabelul **adresa**, trebuie să fie gestionate în mod confidențial și utilizate doar în scopul serviciilor logistice.
- **Comunicarea cu Clienții:**
 - Pentru a menține transparența, clienții trebuie să primească actualizări despre starea comenzilor și estimările de livrare prin intermediul tabelului **comanda** și a serviciului de transport.

3. Descrierea entităților, incluzând precizarea cheii primare

Entitatea client:

- Descriere: Reprezintă informațiile despre clienții care utilizează platforma de logistică.
- Cheie Primară: **id_utilizator**

Entitatea producator_preferat:

- Descriere: Păstrează legătura dintre clienți și producătorii preferați de către aceștia.

- Cheie Primară: **id_producator#**
- Cheie Străină: **id_client** referențiază **client.id_utilizator**

Entitatea metode_plata:

- Descriere: Stochează informații despre metodele de plată asociate clienților.
- Cheie Primară: **id_metoda**
- Cheie Străină: **id_utilizator** referențiază **client.id_utilizator**

Entitatea date_card:

- Descriere: Conține datele cardurilor de credit ale clienților.
- Cheie Primară: **id_card**
- Cheie Străină: **id_utilizator** referențiază **client.id_utilizator**
- **Entitatea transfer_bancar:**
- Descriere: Reține informații despre transferurile bancare disponibile.
- Cheie Primară: **id_transfer**

Entitatea ramburs:

- Descriere: Descrie opțiunile de ramburs disponibile.
- Cheie Primară: **id_ramburs**

Entitatea adresa:

- Descriere: Găzduiește datele de livrare asociate clienților.
- Cheie Primară: **id_adresa**
- Cheie Străină: **id_utilizator** referențiază **client.id_utilizator**

Entitatea transport:

- Descriere: Conține informații despre serviciile de transport disponibile.
- Cheie Primară: **id_transport**
- Cheie Străină: **id_utilizator** referențiază **client.id_utilizator**

Entitatea produse:

- Descriere: Stochează informații despre produsele disponibile pentru achiziționare.
- Cheie Primară: **id_produs**

Entitatea categorie:

- Descriere: Defineste categoriile din care fac parte produsele.
- Cheie Primară: **id_categorie#**

Entitatea producator:

- Descriere: Reține informații despre producătorii implicați în oferirea produselor.
- Cheie Primară: **id_produs#**
- Cheie Străină: **id_producator_preferat** referențiază **producator_preferat.id_producator#**

Entitatea categorie_produse:

- Descriere: Menține legătura între produse și categoriile din care fac parte.
- Cheie Primară: **id_categorie#, id_produs**

- Cheie Străină: **id_categorie#** referențiază **categorie.id_categorie#**, **id_produs** referențiază **produse.id_produs**

Entitatea **transport_produce**:

- Descriere: Asociază produsele cu serviciile de transport disponibile.
- Cheie Primară: **id_produs#**, **id_transport**
- Cheie Străină: **id_produs#** referențiază **produse.id_produs**, **id_transport** referențiază **transport.id_transport**

Entitatea **comanda**:

- Descriere: Păstrează detaliile comenzilor plasate de clienți.
- Cheie Primară: **id_comanda**
- Cheie Străină: **id_metoda** referențiază **metode_plata.id_metoda**, **id_transport** referențiază **transport.id_transport**, **id_adresa** referențiază **adresa.id_adresa**

Entitatea **recenzii**:

- Descriere: Conține recenzii ale clienților pentru produsele achiziționate.
- Cheie Primară: **id_recenzie**
- Cheie Străină: **id_produs** referențiază **produse.id_produs**, **id_client** referențiază **client.id_utilizator**

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora

Relația între **client** și **producator_preferat**:

- Cardinalitate: Un client poate avea zero sau mai mulți producători preferați. Un producător preferat poate fi asociat cu un singur client.
- Această relație permite fiecărui client să-și aleagă producătorii preferați.

Relația între **client** și **metode_plata**:

- Cardinalitate: Un client poate avea zero sau mai multe metode de plată asociate. O metodă de plată poate fi asociată cu un singur client.
- Această relație permite fiecărui client să-și gestioneze metodele preferate de plată.

Relația între **client** și **date_card**:

- Cardinalitate: Un client poate avea zero sau mai multe date de card de credit asociate. O dată de card de credit poate fi asociată cu un singur client.
- Această relație permite clienților să-și stocheze informațiile de card de credit pentru plată.

Relația între **client** și **adresa**:

- Cardinalitate: Un client poate avea zero sau mai multe adrese asociate. O adresă poate fi asociată cu un singur client.

- Această relație permite clienților să-și stocheze și să-și gestioneze adresele de livrare.

Relația între **client** și **transport**:

- Cardinalitate: Un client poate avea zero sau mai multe servicii de transport asociate. Un serviciu de transport poate fi asociat cu un singur client.
- Această relație permite clienților să-și aleagă serviciile preferate de transport.

Relația între **categorie** și **categorie_produce**:

- Cardinalitate: O categorie poate fi asociată cu zero sau mai multe produse. Un produs poate fi asociat cu o singură categorie.
- Această relație permite organizarea produselor în categorii.

Relația între **produse** și **categorie_produce**:

- Cardinalitate: Un produs poate fi asociat cu zero sau mai multe categorii. O categorie poate fi asociată cu un singur produs.
- Această relație permite etichetarea produselor cu categoriile corespunzătoare.

Relația între **produse** și **transport_produce**:

- Cardinalitate: Un produs poate fi asociat cu zero sau mai multe servicii de transport. Un serviciu de transport poate fi asociat cu unul sau mai multe produse.
- Această relație permite gestionarea stocurilor și a transportului pentru fiecare produs.

Relația între **client**, **comanda**, și **transport**:

- Cardinalitate: Un client poate plasa zero sau mai multe comenzi. O comandă poate fi asociată cu un singur client și un singur serviciu de transport.
- Această relație leagă comenzile clienților cu detaliile de transport aferente.

Relația între **produse**, **comanda** și **recenzii**:

- Cardinalitate: Un produs poate avea zero sau mai multe recenzii. O comandă poate avea zero sau mai multe produse asociate. O recenzie poate fi asociată cu un singur produs și un singur client.
- Această relație permite clienților să lase recenzii pentru produsele achiziționate.

5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.

Tabela client:

- **id_utilizator**: NUMBER (Cheie Primară)
- **nume**: VARCHAR2(255)
- **prenume**: VARCHAR2(255)
- **username**: VARCHAR2(255) (Unic)
- **email**: VARCHAR2(255) (Unic)
- **parola**: VARCHAR2(255)

Tabela producator_preferat:

- **id_producator#**: NUMBER (Cheie Primară)

- **id_client**: NUMBER (Cheie Străină către **client.id_utilizator**)

Tabela metode_plata:

- **id_metoda**: NUMBER (Cheie Primară)
- **id_utilizator**: NUMBER (Cheie Străină către **client.id_utilizator**)
- **data_asociere**: DATE

Tabela date_card:

- **id_card**: NUMBER (Cheie Primară)
- **id_utilizator**: NUMBER (Cheie Străină către **client.id_utilizator**)
- **nr_card**: VARCHAR2(16)
- **cvv**: VARCHAR2(4)
- **data_expirare**: DATE

Tabela transfer_bancar:

- **id_transfer**: NUMBER (Cheie Primară)
- **nume_banca**: VARCHAR2(255)
- **limita_transfer**: NUMBER(10, 2)

Tabela ramburs:

- **id_ramburs**: NUMBER (Cheie Primară)
- **firma_preferata**: VARCHAR2(255)
- **verificare_livrare**: NUMBER(1)

Tabela adresa:

- **id_adresa**: NUMBER (Cheie Primară)
- **id_utilizator**: NUMBER (Cheie Străină către **client.id_utilizator**)
- **judet**: VARCHAR2(255)
- **oras**: VARCHAR2(255)
- **strada**: VARCHAR2(255)
- **numar**: VARCHAR2(20)

Tabela transport:

- **id_transport**: NUMBER (Cheie Primară)
- **id_utilizator**: NUMBER (Cheie Străină către **client.id_utilizator**)

Tabela produse:

- **id_produs**: NUMBER (Cheie Primară)
- **nume**: VARCHAR2(255)
- **descriere**: CLOB
- **pret**: NUMBER(10, 2)

Tabela categorie:

- **id_categorie#**: NUMBER (Cheie Primară)
- **tip**: VARCHAR2(255)

Tabela producator:

- **id_produs#**: NUMBER (Cheie Primară)

- **id_producator_preferat**: NUMBER (Cheie Străină către **producator_preferat.id_producator#**)

Tabela categorie_produce:

- **id_categorie#**: NUMBER (Cheie Străină către **categorie.id_categorie#**)
- **id_produs**: NUMBER (Cheie Străină către **produse.id_produs**)
- **nume**: VARCHAR2(255) (Cheie Primară împreună cu **id_categorie#** și **id_produs**)

Tabela transport_produce:

- **id_produs#**: NUMBER (Cheie Străină către **produse.id_produs**)
- **id_transport**: NUMBER (Cheie Străină către **transport.id_transport**)
- Cheie Primară: **id_produs#, id_transport**

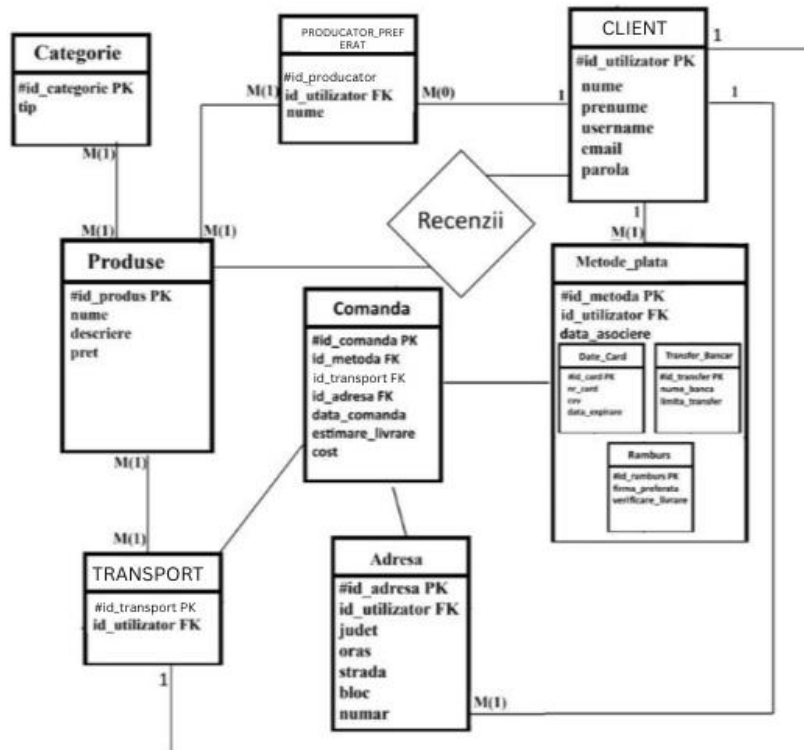
Tabela comanda:

- **id_comanda**: NUMBER (Cheie Primară)
- **id_metoda**: NUMBER (Cheie Străină către **metode_plata.id_metoda**)
- **id_transport**: NUMBER (Cheie Străină către **transport.id_transport**)
- **id_adresa**: NUMBER (Cheie Străină către **adresa.id_adresa**)
- **data_comanda**: DATE
- **estimare_livrare**: DATE
- **cost**: NUMBER(10, 2)

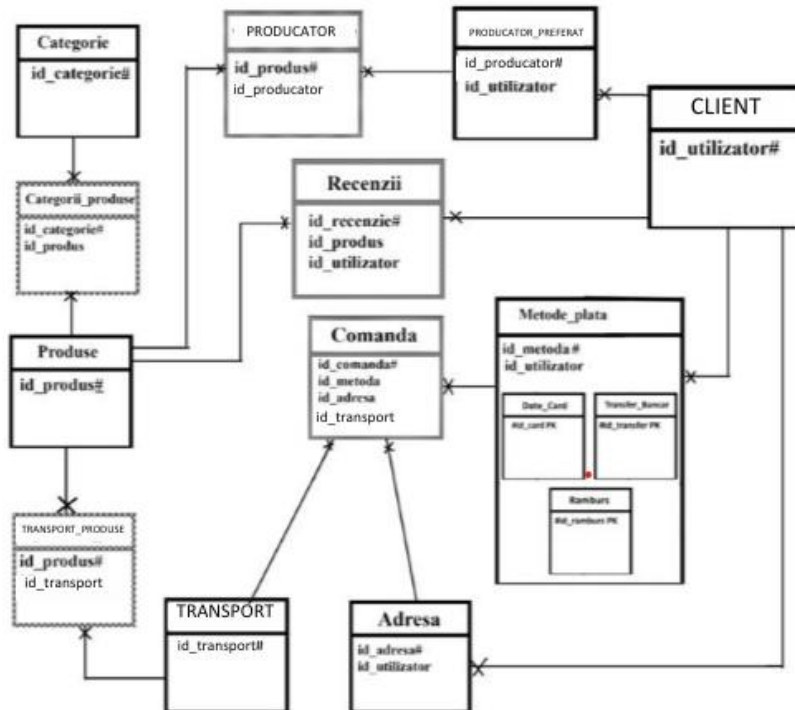
Tabela recenzii:

- **id_recenzie**: NUMBER (Cheie Primară)
- **id_produs**: NUMBER (Cheie Străină către **produse.id_produs**)
- **id_client**: NUMBER (Cheie Străină către **client.id_utilizator**)

6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.



8.Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

Tabela **client**:

- **id_utilizator** (Primary Key)
- **nume**
- **prenume**
- **username**
- **email**
- **parola**

Tabela **producator_preferat**:

- **id_producator#** (Primary Key)
- **id_client** (Foreign Key către **client.id_utilizator**)

Tabela **metode_plata**:

- **id_metoda** (Primary Key)
- **id_utilizator** (Foreign Key către **client.id_utilizator**)

- **data_asociere**

Tabela **date_card**:

- **id_card** (Primary Key)
- **id_utilizator** (Foreign Key către **client.id_utilizator**)
- **nr_card**
- **cvv**
- **data_expirare**

Tabela **transfer_bancar**:

- **id_transfer** (Primary Key)
- **nume_banca**
- **limita_transfer**

Tabela **ramburs**:

- **id_ramburs** (Primary Key)
- **firma_preferata**
- **verificare_livrare**

Tabela **adresa**:

- **id_adresa** (Primary Key)
- **id_utilizator** (Foreign Key către **client.id_utilizator**)
- **judet**
- **oras**
- **strada**
- **numar**

Tabela **transport**:

- **id_transport** (Primary Key)
- **id_utilizator** (Foreign Key către **client.id_utilizator**)

Tabela **produse**:

- **id_produs** (Primary Key)
- **nume**
- **descriere**
- **pret**

Tabela **categorii**:

- **id_categorie#** (Primary Key)
- **tip**

Tabela **producator**:

- **id_produs#** (Primary Key)
- **id_producator_preferat** (Foreign Key către **producator_preferat.id_producator#**)

Tabela **categorii_produse**:

- **id_categorie#** (Foreign Key către **categorii.id_categorie#**)
- **id_produs** (Foreign Key către **produse.id_produs**)

- **nume** (Primary Key împreună cu **id_categorie#** și **id_produc**)

Tabela **transport_produce**:

- **id_produc#** (Foreign Key către **produse.id_produc**)
- **id_transport** (Foreign Key către **transport.id_transport**)
- Primary Key format din **id_produc#** și **id_transport**

Tabela **comanda**:

- **id_comanda** (Primary Key)
- **id_metoda** (Foreign Key către **metode_plata.id_metoda**)
- **id_transport** (Foreign Key către **transport.id_transport**)
- **id_adresa** (Foreign Key către **adresa.id_adresa**)
- **data_comanda**
- **estimare_livrare**
- **cost**

Tabela **recenzii**:

- **id_recenzie** (Primary Key)
- **id_produc** (Foreign Key către **produse.id_produc**)
- **id_client** (Foreign Key către **client.id_utilizator**)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3):

Exemplu și transformarea în Forma Normală 1 (FN1):

În acest exemplu, vom presupune că un client poate avea mai multe metode de plată.

Tabelul Non-FN1: metode_plata

id_metoda	id_utilizator	data_asociere
1	101	2023-08-01
2	101	2023-08-02
3	102	2023-08-03

Transformarea în FN1: metode_plata

id_metoda	id_utilizator
1	101
2	101
3	102

În Forma Normală 1, am eliminat coloana `data_asociere` din tabelul `metode_plata`, deoarece nu depinde funcțional de `id_metoda`. Datele repetitive au fost îndepărtate.

Exemplu și transformarea în Forma Normală 2 (FN2):

În acest exemplu, vom presupune că un produs poate aparține mai multor categorii.

Tabelul Non-FN2: **categorie_produce**

id_categorie#	id_produc	nume
1	201	Categorie A
1	202	Categorie A
2	201	Categorie B

Transformarea în FN2: **categorie_produce**

Tabelul **categorie**

id_categorie#	tip
1	Categorie A
2	Categorie B

Tabelul **produse_categorii**

id_produc	id_categorie#
201	1
202	1
201	2

În Forma Normală 2, am separat informațiile despre categorii într-un tabel distinct

(**categorie**) și am creat o legătură de tip mulți-la-mulți între **produse** și **categorie** folosind tabelul intermediar **produse_categorii**.

Exemplu și transformarea în Forma Normală 3 (FN3):

În acest exemplu, vom presupune că o adresă poate fi folosită pentru mai multe transporturi.

Tabelul Non-FN3: transport

id_transport	id_utilizator
1	101
2	102
3	101

Tabelul Non-FN3: adresa

id_adresa	id_utilizator	judet	oras
1	101	Judet A	Oras A
2	102	Judet B	Oras B

Transformarea în FN3: transport

id_transport	id_adresa
1	1
2	2
3	1

În Forma Normală 3, am separat informațiile despre adrese într-un tabel distinct (**`adresa`**) și am creat o legătură între **`transport`** și **`adresa`**, eliminând astfel redundanța din schema.

10.Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). Am creat cate o secventa pentru fiecare tabel dupa cum urmeaza:

-- Secvența pentru tabela "producator_preferat"

```
CREATE SEQUENCE producator_preferat_seq START WITH 1 INCREMENT BY 1;
```

-- Secvența pentru tabela "producator"

```
CREATE SEQUENCE producator_seq START WITH 1 INCREMENT BY 1;
```

-- Secvența pentru tabela "categorie"


```
CREATE SEQUENCE categorie_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "produse"
```

```
CREATE SEQUENCE produse_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "transport"
```

```
CREATE SEQUENCE transport_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "transport_produse"
```

```
CREATE SEQUENCE transport_produse_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "comanda"
```

```
CREATE SEQUENCE comanda_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "adresa"
```

```
CREATE SEQUENCE adresa_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "client"
```

```
CREATE SEQUENCE client_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "metode_plata"
```

```
CREATE SEQUENCE metode_plata_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "date_card"
```

```
CREATE SEQUENCE date_card_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "transfer_bancar"
```

```
CREATE SEQUENCE transfer_bancar_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "ramburs"
```

```
CREATE SEQUENCE ramburs_seq START WITH 1 INCREMENT BY 1;
```

```
-- Secvența pentru tabela "recenzii"
```

```
CREATE SEQUENCE recenzii_seq START WITH 1 INCREMENT BY 1;
```

	SEQUENCE_NAME
1	ADRESA_SEQ
2	CATEGORIE_SEQ
3	CLIENT_SEQ
4	COMANDA_SEQ
5	DATE_CARD_SEQ
6	METODE_PLATA_SEQ
7	PRODUCATOR_PREFERAT_SEQ
8	PRODUCATOR_SEQ
9	PRODUSE_SEQ
10	RAMBURS_SEQ
11	RECENZII_SEQ
12	TRANSFER_BANCAR_SEQ
13	TRANSPORT_PRODUSE_SEQ
14	TRANSPORT_SEQ

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative)

COD SQL:

```
CREATE TABLE client (
    id_utilizator NUMBER PRIMARY KEY,
    nume VARCHAR2(255),
    prenume VARCHAR2(255),
    username VARCHAR2(255),
    email VARCHAR2(255),
    parola VARCHAR2(255)
);
```

```
CREATE TABLE producator_preferat (
    id_producator# NUMBER PRIMARY KEY,
    id_client NUMBER,
    FOREIGN KEY (id_client) REFERENCES client(id_utilizator)
);
```

```
CREATE TABLE metode_plata (
    id_metoda NUMBER PRIMARY KEY,
```

```
id_utilizator NUMBER,  
data_asociere DATE,  
FOREIGN KEY (id_utilizator) REFERENCES client(id_utilizator)  
);
```

```
CREATE TABLE date_card (  
id_card NUMBER PRIMARY KEY,  
id_utilizator NUMBER,  
nr_card VARCHAR2(16),  
cvv VARCHAR2(4),  
data_expirare DATE,  
FOREIGN KEY (id_utilizator) REFERENCES client(id_utilizator)  
);
```

```
CREATE TABLE transfer_bancar (  
id_transfer NUMBER PRIMARY KEY,  
nume_banca VARCHAR2(255),  
limita_transfer NUMBER(10, 2)  
);
```

```
CREATE TABLE ramburs (  
id_ramburs NUMBER PRIMARY KEY,  
firma_preferata VARCHAR2(255),  
verificare_livrare NUMBER(1)  
);
```

```
CREATE TABLE adresa (  
id_adresa NUMBER PRIMARY KEY,  
id_utilizator NUMBER,  
judet VARCHAR2(255),  
oras VARCHAR2(255),  
strada VARCHAR2(255),  
numar VARCHAR2(20),  
FOREIGN KEY (id_utilizator) REFERENCES client(id_utilizator)  
);
```

```
CREATE TABLE transport (  
id_transport NUMBER PRIMARY KEY,  
id_utilizator NUMBER,  
FOREIGN KEY (id_utilizator) REFERENCES client(id_utilizator)  
);
```

```
CREATE TABLE produse (  
    id_produs NUMBER PRIMARY KEY,  
    nume VARCHAR2(255),  
    descriere CLOB,  
    pret NUMBER(10, 2)  
);
```

```
CREATE TABLE categorii (  
    id_categorie# NUMBER PRIMARY KEY,  
    tip VARCHAR2(255)  
);
```

```
CREATE TABLE producatori (  
    id_produs# NUMBER PRIMARY KEY,  
    id_producator_preferat NUMBER,  
    FOREIGN KEY (id_producator_preferat) REFERENCES producatori(id_producator#)  
);
```

```
CREATE TABLE categorii_produce (  
    id_categorie# NUMBER,  
    id_produs NUMBER,  
    nume VARCHAR2(255),  
    PRIMARY KEY (id_categorie#, id_produs),  
    FOREIGN KEY (id_categorie#) REFERENCES categorii(id_categorie#),  
    FOREIGN KEY (id_produs) REFERENCES produse(id_produs)  
);
```

```
CREATE TABLE transport_produce (  
    id_produs# NUMBER,  
    id_transport NUMBER,  
    PRIMARY KEY (id_produs#, id_transport),  
    FOREIGN KEY (id_produs#) REFERENCES produse(id_produs),  
    FOREIGN KEY (id_transport) REFERENCES transport(id_transport)  
);
```

```
CREATE TABLE comanda (  
    id_comanda NUMBER PRIMARY KEY,  
    id_metoda NUMBER,  
    id_transport NUMBER,
```

```
id_adresa NUMBER,  
data_comanda DATE,  
estimare_livrare DATE,  
cost NUMBER(10, 2),  
FOREIGN KEY (id_metoda) REFERENCES metode_plata(id_metoda),  
FOREIGN KEY (id_transport) REFERENCES transport(id_transport),  
FOREIGN KEY (id_adresa) REFERENCES adresa(id_adresa)  
);
```

```
CREATE TABLE recenzii (  
    id_recenzie NUMBER PRIMARY KEY,  
    id_produus NUMBER,  
    id_client NUMBER,  
    FOREIGN KEY (id_produus) REFERENCES produse(id_produus),  
    FOREIGN KEY (id_client) REFERENCES client(id_utilizator)  
);
```

-- INSERARE IN TABELA CLIENT

```
INSERT INTO client (id_utilizator, nume, prenume, username, email, parola)  
VALUES (client_seq.NEXTVAL, 'Popescu', 'Ana', 'ana.popescu', 'ana.popescu@email.com',  
'parola123');
```

```
INSERT INTO client (id_utilizator, nume, prenume, username, email, parola)  
VALUES (client_seq.NEXTVAL, 'Ionescu', 'Alex', 'alex.ionescu', 'alex.ionescu@email.com',  
'parola456');
```

```
INSERT INTO client (id_utilizator, nume, prenume, username, email, parola)  
VALUES (client_seq.NEXTVAL, 'Mihai', 'Andreea', 'andreea.mihai', 'andreea.mihai@email.com',  
'parola789');
```

```
INSERT INTO client (id_utilizator, nume, prenume, username, email, parola)  
VALUES (client_seq.NEXTVAL, 'Dumitru', 'Mihai', 'mihai.dumitru', 'mihai.dumitru@email.com',  
'parolaabc');
```

```
INSERT INTO client (id_utilizator, nume, prenume, username, email, parola)  
VALUES (client_seq.NEXTVAL, 'Georgescu', 'Elena', 'elena.georgescu',  
'elena.georgescu@email.com', 'parolaxyz');
```

-- INSERARE IN TABELA PRODUCATOR PREFERAT:

```
INSERT INTO producator_preferat (id_producator#, id_client)
VALUES (producator_preferat_seq.NEXTVAL, 1);
```

```
INSERT INTO producator_preferat (id_producator#, id_client)
VALUES (producator_preferat_seq.NEXTVAL, 2);
```

```
INSERT INTO producator_preferat (id_producator#, id_client)
VALUES (producator_preferat_seq.NEXTVAL, 3);
```

```
INSERT INTO producator_preferat (id_producator#, id_client)
VALUES (producator_preferat_seq.NEXTVAL, 4);
```

```
INSERT INTO producator_preferat (id_producator#, id_client)
VALUES (producator_preferat_seq.NEXTVAL, 5);
```

```
-- INSERARE IN TABELA METODE_PLATA:
```

```
INSERT INTO metode_plata (id_metoda, id_utilizator, data_asociere)
VALUES (metode_plata_seq.NEXTVAL, 1, TO_DATE('2023-08-28', 'YYYY-MM-DD'));
```

```
INSERT INTO metode_plata (id_metoda, id_utilizator, data_asociere)
VALUES (metode_plata_seq.NEXTVAL, 2, TO_DATE('2023-08-29', 'YYYY-MM-DD'));
```

```
INSERT INTO metode_plata (id_metoda, id_utilizator, data_asociere)
VALUES (metode_plata_seq.NEXTVAL, 3, TO_DATE('2023-08-30', 'YYYY-MM-DD'));
```

```
INSERT INTO metode_plata (id_metoda, id_utilizator, data_asociere)
VALUES (metode_plata_seq.NEXTVAL, 4, TO_DATE('2023-08-31', 'YYYY-MM-DD'));
```

```
INSERT INTO metode_plata (id_metoda, id_utilizator, data_asociere)
VALUES (metode_plata_seq.NEXTVAL, 5, TO_DATE('2023-09-01', 'YYYY-MM-DD'));
```

```
-- INSERARE IN TABELA DATE_CARD:
```

```
INSERT INTO date_card (id_card, nr_card, cvv, data_expirare)
VALUES (date_card_seq.NEXTVAL, '1234567890123456', '123', TO_DATE('2025-12-31', 'YYYY-MM-DD'));
```

```
INSERT INTO date_card (id_card, nr_card, cvv, data_expirare)
```

```
VALUES (date_card_seq.NEXTVAL, '9876543210987654', '456', TO_DATE('2024-09-30', 'YYYY-MM-DD'));
```

```
INSERT INTO date_card (id_card, nr_card, cvv, data_expirare)  
VALUES (date_card_seq.NEXTVAL, '4567890123456789', '789', TO_DATE('2023-11-30', 'YYYY-MM-DD'));
```

```
INSERT INTO date_card (id_card, nr_card, cvv, data_expirare)  
VALUES (date_card_seq.NEXTVAL, '7890123456789012', '234', TO_DATE('2024-06-30', 'YYYY-MM-DD'));
```

```
INSERT INTO date_card (id_card, nr_card, cvv, data_expirare)  
VALUES (date_card_seq.NEXTVAL, '5678901234567890', '567', TO_DATE('2023-08-31', 'YYYY-MM-DD'));
```

-- INSERARE IN TABELA TRANSFER_BANCAR:

```
INSERT INTO transfer_bancar (id_transfer, nume_banca, limita_transfer)  
VALUES (transfer_bancar_seq.NEXTVAL, 'BCR', 1000.00);
```

```
INSERT INTO transfer_bancar (id_transfer, nume_banca, limita_transfer)  
VALUES (transfer_bancar_seq.NEXTVAL, 'BRD', 1500.00);
```

```
INSERT INTO transfer_bancar (id_transfer, nume_banca, limita_transfer)  
VALUES (transfer_bancar_seq.NEXTVAL, 'ING', 2000.00);
```

```
INSERT INTO transfer_bancar (id_transfer, nume_banca, limita_transfer)  
VALUES (transfer_bancar_seq.NEXTVAL, 'Raiffeisen', 1200.00);
```

```
INSERT INTO transfer_bancar (id_transfer, nume_banca, limita_transfer)  
VALUES (transfer_bancar_seq.NEXTVAL, 'Banca Transilvania', 1800.00);
```

-- INSERARE IN TABELA RAMBURS:

```
INSERT INTO ramburs (id_ramburs, firma_preferata, verificare_livrare)  
VALUES (ramburs_seq.NEXTVAL, 'Fan Courier', 1);
```

```
INSERT INTO ramburs (id_ramburs, firma_preferata, verificare_livrare)  
VALUES (ramburs_seq.NEXTVAL, 'DPD', 1);
```

```
INSERT INTO ramburs (id_ramburs, firma_preferata, verificare_livrare)
VALUES (ramburs_seq.NEXTVAL, 'Cargus', 0);
```

```
INSERT INTO ramburs (id_ramburs, firma_preferata, verificare_livrare)
VALUES (ramburs_seq.NEXTVAL, 'Urgent Cargus', 1);
```

```
INSERT INTO ramburs (id_ramburs, firma_preferata, verificare_livrare)
VALUES (ramburs_seq.NEXTVAL, 'FAN', 1);
```

-- INSERARE IN TABELA ADRESA:

```
INSERT INTO adresa (id_adresa, id_utilizator, judet, oras, strada, numar)
VALUES (adresa_seq.NEXTVAL, 1, 'București', 'București', 'Strada Victoriei', '10A');
```

```
INSERT INTO adresa (id_adresa, id_utilizator, judet, oras, strada, numar)
VALUES (adresa_seq.NEXTVAL, 2, 'Cluj', 'Cluj-Napoca', 'Strada Mihai Viteazu', '25');
```

```
INSERT INTO adresa (id_adresa, id_utilizator, judet, oras, strada, numar)
VALUES (adresa_seq.NEXTVAL, 3, 'Timiș', 'Timișoara', 'Strada Libertății', '8B');
```

```
INSERT INTO adresa (id_adresa, id_utilizator, judet, oras, strada, numar)
VALUES (adresa_seq.NEXTVAL, 4, 'Iași', 'Iași', 'Strada Palat', '15');
```

```
INSERT INTO adresa (id_adresa, id_utilizator, judet, oras, strada, numar)
VALUES (adresa_seq.NEXTVAL, 5, 'Constanța', 'Constanța', 'Bulevardul Tomis', '50C');
```

-- INSERARE IN TABELA TRANSPORT:

```
INSERT INTO transport (id_transport, id_utilizator)
VALUES (transport_seq.NEXTVAL, 1);
```

```
INSERT INTO transport (id_transport, id_utilizator)
VALUES (transport_seq.NEXTVAL, 2);
```

```
INSERT INTO transport (id_transport, id_utilizator)
VALUES (transport_seq.NEXTVAL, 3);
```

```
INSERT INTO transport (id_transport, id_utilizator)
VALUES (transport_seq.NEXTVAL, 4);
```



```
INSERT INTO transport (id_transport, id_utilizator)
VALUES (transport_seq.NEXTVAL, 5);
```

-- INSERARE IN TABELA PRODUSE:

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Televizor Smart', 'Televizor LED Smart Full HD, diagonala 55
inch.', 1500.00);
```

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Bluză cu Glugă', 'Bluză confortabilă cu glugă, material de
calitate.', 50.00);
```

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Pâine Integrală', 'Pâine proaspătă integrală, făcută în casă.',
3.50);
```

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Lapte Demachiant', 'Lapte demachiant delicat pentru toate
tipurile de ten.', 12.99);
```

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Călătorie în Tărâmul Minunilor', 'Carte de aventuri pentru
copii.', 15.00);
```

```
INSERT INTO produse (id_produc, nume, descriere, pret)
VALUES (produse_seq.NEXTVAL, 'Servetele umede', 'Servetele', NULL);
```

-- INSERARE IN TABELA CATEGORIE:

```
INSERT INTO categorie (id_categorie#, tip)
VALUES (categorie_seq.NEXTVAL, 'Electronice');
```

```
INSERT INTO categorie (id_categorie#, tip)
VALUES (categorie_seq.NEXTVAL, 'Îmbrăcăminte');
```

```
INSERT INTO categorie (id_categorie#, tip)
VALUES (categorie_seq.NEXTVAL, 'Alimente');
```

```
INSERT INTO categorie (id_categorie#, tip)
VALUES (categorie_seq.NEXTVAL, 'Cosmetice');
```

```
INSERT INTO categorie (id_categorie#, tip)
VALUES (categorie_seq.NEXTVAL, 'Cărți');
```

-- INSERARE IN TABELA PRODUCATOR:

```
INSERT INTO producator (id_produș#, id_producator_preferat)
VALUES (producator_seq.NEXTVAL, 1);
INSERT INTO producator (id_produș#, id_producator_preferat)
VALUES (producator_seq.NEXTVAL, 2);
INSERT INTO producator (id_produș#, id_producator_preferat)
VALUES (producator_seq.NEXTVAL, 3);
INSERT INTO producator (id_produș#, id_producator_preferat)
VALUES (producator_seq.NEXTVAL, 4);
INSERT INTO producator (id_produș#, id_producator_preferat)
VALUES (producator_seq.NEXTVAL, 5);
```

-- INSERARE IN TABELA CATEGORIE_PRODUSE:

```
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (1, 1, 'Electronice');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (2, 2, 'Modă');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (3, 3, 'Cosmetice');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (4, 4, 'Cărți');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (5, 5, 'Jucării');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (1, 2, 'Electronice');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
VALUES (2, 3, 'Modă');
INSERT INTO categorie_produce (id_categorie#, id_produș, nume)
```

```
VALUES (3, 4, 'Cosmetice');
INSERT INTO categorie_produce (id_categorie#, id_produc, nume)
VALUES (4, 5, 'Cărți');
INSERT INTO categorie_produce (id_categorie#, id_produc, nume)
VALUES (5, 1, 'Jucării');
```

-- INSERARE IN TABELA TRANSPORT PRODUSE:

```
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (1, 1);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (2, 2);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (3, 3);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (4, 4);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (5, 5);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (1, 2);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (2, 3);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (3, 4);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (4, 5);
INSERT INTO transport_produce (id_produc#, id_transport)
VALUES (5, 1);
```

-- INSERARE IN TABELA COMANDA:

```
INSERT INTO comanda (id_comanda, id_metoda, id_transport, id_adresa, data_comanda,
estimare_livrare, cost)
VALUES (comanda_seq.NEXTVAL, 1, 1, 1, TO_DATE('2023-08-28', 'YYYY-MM-DD'),
TO_DATE('2023-09-04', 'YYYY-MM-DD'), 200.00);
INSERT INTO comanda (id_comanda, id_metoda, id_transport, id_adresa, data_comanda,
estimare_livrare, cost)
```

```
VALUES (comanda_seq.NEXTVAL, 2, 2, 2, TO_DATE('2023-08-29', 'YYYY-MM-DD'),
TO_DATE('2023-09-05', 'YYYY-MM-DD'), 150.00);
INSERT INTO comanda (id_comanda, id_metoda, id_transport, id_adresa, data_comanda,
estimare_livrare, cost)
VALUES (comanda_seq.NEXTVAL, 3, 3, 3, TO_DATE('2023-08-30', 'YYYY-MM-DD'),
TO_DATE('2023-09-06', 'YYYY-MM-DD'), 50.00);
INSERT INTO comanda (id_comanda, id_metoda, id_transport, id_adresa, data_comanda,
estimare_livrare, cost)
VALUES (comanda_seq.NEXTVAL, 4, 4, 4, TO_DATE('2023-08-31', 'YYYY-MM-DD'),
TO_DATE('2023-09-07', 'YYYY-MM-DD'), 80.00);
INSERT INTO comanda (id_comanda, id_metoda, id_transport, id_adresa, data_comanda,
estimare_livrare, cost)
VALUES (comanda_seq.NEXTVAL, 5, 5, 5, TO_DATE('2023-09-01', 'YYYY-MM-DD'),
TO_DATE('2023-09-08', 'YYYY-MM-DD'), 300.00);
```

-- INSERARE IN TABELA RECENZII:

```
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 1, 1);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 2, 2);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 3, 3);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 4, 4);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 5, 5);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 1, 2);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 2, 3);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 3, 4);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 4, 5);
INSERT INTO recenzii (id_recenzie, id_produs, id_client)
VALUES (recenzii_seq.NEXTVAL, 5, 1);
```

Tables (Filtered)	
+	ADRESA
+	CATEGORIE
+	CATEGORIE_PRODUSE
+	CLIENT
+	COMANDA
+	DATE_CARD
+	METODE_PLATA
+	PRODUCATOR
+	PRODUCATOR_PREFERAT
+	PRODUSE
+	RAMBURS
+	RECENZII
+	TRANSFER_BANCAR
+	TRANSPORT
+	TRANSPORT_PRODUSE

DATE DIN TABELE:

ID_ADRESA	ID_UTILIZATOR	JUDET	ORAS	STRADA	NUMAR
1	1	Bucureşti	Bucureşti	Strada Victoriei	10A
2	2	Cluj	Cluj-Napoca	Strada Mihai Viteazu	25
3	3	Timiş	Timişoara	Strada Libertăţii	8B
4	4	Iaşi	Iaşi	Strada Palat	15
5	5	Constanţa	Constanţa	Bulevardul Tomis	50C

ADRRESA:

ID_CATEGORIE#	TIP
1	Electronice
2	Îmbrăcămintे
3	Alimente
4	Cosmetice
5	Cărţi

CATEGORIE

ID_CATEGORIE#	ID_PRODUS	NUME
1	1	Electronice
2	2	Modă
3	3	Cosmetice
4	4	Cărți
5	5	Jucării
1	2	Electronice
2	3	Modă
3	4	Cosmetice
4	5	Cărți
5	1	Jucării

CATEGORIE_PRODUSE

ID_UTILIZATOR	NUME	PRENUME	USERNAME	EMAIL	PAROLA
1	Popescu	Ana	ana.popescu	ana.popescu@email.com	parola123
2	Ionescu	Alex	alex.ionescu	alex.ionescu@email.com	parola456
3	Mihai	Andreea	andreea.mihai	andreea.mihai@email.com	parola789
4	Dumitru	Mihai	mihai.dumitru	mihai.dumitru@email.com	parolaabc
5	Georgescu	Elena	elena.georgescu	elena.georgescu@email.com	parolaxyz

CLIENT:

ID_COMANDA	ID_METODA	ID_TRANSPORT	ID_ADRESA	DATA_COMANDA	ESTIMARE_LIVRARE	COST
1	1	1	1	28-AUG-23	04-SEP-23	200
2	2	2	2	29-AUG-23	05-SEP-23	150
3	3	3	3	30-AUG-23	06-SEP-23	50
4	4	4	4	31-AUG-23	07-SEP-23	80
5	5	5	5	01-SEP-23	08-SEP-23	300

COMANDA

ID_METODA	ID_UTILIZATOR	DATA_ASOCIERE
1	1	28-AUG-23
2	2	29-AUG-23
3	3	30-AUG-23
4	4	31-AUG-23
5	5	01-SEP-23

METODE PLATA

ID_PRODUS#	ID_PRODUCATOR_PREFERAT
1	1
2	2
3	3
4	4
5	5

PRODUCATOR

ID_PRODUCATOR#	ID_CLIENT
1	1
2	2
3	3
4	4
5	5

PRODUCATOR_PREFERAT

ID_PRODUS	NUME	DESCRIERE	PRET
1	Televizor Smart	Televizor LED Smart Full HD, diagonala 55 inch.	1650
2	Bluză cu Glugă	Bluză confortabilă cu glugă, material de calitate.	55
3	Pâine Integrală	Pâine proaspătă integrală, făcută în casă.	3.5
4	Lapte Demachiant	Lapte demachiant delicat pentru toate tipurile de ten.	12.99
5	Călătorie în Tărâmul Minunilor	Carte de aventuri pentru copii.	15
6	Servetele umede	Servetele	(null)

PRODUSE

ID_RAMBURS	FIRMA_PREFERATA	VERIFICARE_LIVRARE
1	Fan Courier	1
2	DPD	1
3	Cargus	0
4	Urgent Cargus	1
5	FAN	1

RAMBURS

ID_RECENZIE	ID_PRODUS	ID_CLIENT
2	2	2
3	3	3
4	4	4
5	5	5
6	1	2
7	2	3
8	3	4
9	4	5

RECENZII

ID_TRANSFER	NUME_BANCA	LIMITA_TRANSFER
1	BCR	1000
2	BRD	1500
3	ING	2000
4	Raiffeisen	1200
5	Banca Transilvania	1800

TRANSFER_BANCAR

ID_TRANSPORT	ID_UTILIZATOR
1	1
2	2
3	3
4	4
5	5

TRANSPORT

ID_PRODUS#	ID_TRANSPORT
1	1
2	2
3	3
4	4
5	5
1	2
2	3
3	4
4	5
5	1

TRANSPORT_PRODUSE

12. Formulați în limbaj natural și implementați 5 cereri SQL complexe care vor utiliza, în ansamblul lor, următoarele elemente:

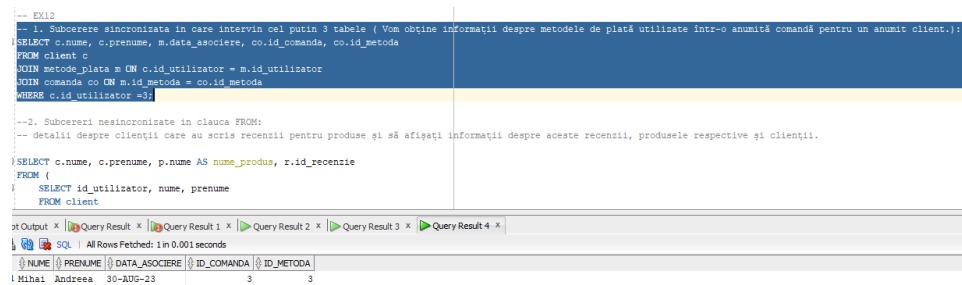
- subcereri sincronizate în care intervin cel puțin 3 tabele
- subcereri nesincronizate în clauza FROM
- grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)
- ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
- utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
- utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

COD SQL:

-- EX12

-- 1. Subcerere sincronizata in care intervin cel putin 3 tabele (Vom obține informații despre metodele de plată utilizate într-o anumită comandă pentru un anumit client.):

```
SELECT c.num, c.prenume, m.data_asociere, co.id_comanda, co.id_metoda
FROM client c
JOIN metode_plata m ON c.id_utilizator = m.id_utilizator
JOIN comanda co ON m.id_metoda = co.id_metoda
WHERE c.id_utilizator =3;
```



--2. Subcereri nesincronizate in clauca FROM:

-- detalii despre clienții care au scris recenzii pentru produse și să afișați informații despre aceste recenzii, produsele respective și clienții.

```
SELECT c.num, c.prenume, p.num AS nume_produs, r.id_recenzie
FROM (
    SELECT id_utilizator, nume, prenume
    FROM client
    WHERE id_utilizator IN (
        SELECT DISTINCT id_client
        FROM recenzii
    )
) c
JOIN recenzii r ON c.id_utilizator = r.id_client
JOIN produse p ON r.id_produs = p.id_produs;
```

```

SELECT c.num, c.prenume, p.num AS nume_produs, r.id_recenzie
FROM (
    SELECT id_utilizator, nume, prenume
    FROM client
    WHERE id_utilizator IN (
        SELECT DISTINCT id_client
        FROM recenzii
    )
) c
JOIN recenzii r ON c.id_utilizator = r.id_client
JOIN produse p ON r.id_produs = p.id_produs;

```

--3. grupări de date cu subcereri nesincronizate în care intervin cel puțin 3
--grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)

Output	Query Result	Query Result 1	Query Result 2	Query Result 3	Query
SQL All Rows Fetched: 8 in 0.003 seconds					
NUME	PRENUME	NUME_PRODUS	ID_RECENZIE		
Ionescu	Alex	Televizor Smart	6		
Mihai	Andreea	Bluză cu Glugă	7		
Ionescu	Alex	Bluză cu Glugă	2		
Dumitru	Mihai	Pâine Integrală	8		
Mihai	Andreea	Pâine Integrală	3		
Georgescu	Elena	Lapte Demachiant	9		
Dumitru	Mihai	Lapte Demachiant	4		
Georgescu	Elena	Călătorie în Tărâmul Minunilor	5		

--3. grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții
--grup, filtrare la nivel de grupuri (în cadrul aceleiași cereri)
--câți clienți unici au plasat comenzi în fiecare județ și să calculăm valoarea medie a comenzilor
pentru fiecare județ

```

SELECT
    a.judet,
    COUNT(DISTINCT c.id_utilizator) AS numar_clienti_unici,
    AVG(co.cost) AS valoare_medie_comenzi
FROM
    adresa a
JOIN
    comanda co ON a.id_adresa = co.id_adresa
JOIN
    client c ON a.id_utilizator = c.id_utilizator
GROUP BY

```

```
--câți clienți unici au plasat comenzi în fiecare județ și să calculăm valoarea  
SELECT  
    a.judet,  
    COUNT(DISTINCT c.id_utilizator) AS numar_clienti_unici,  
    AVG(co.cost) AS valoare_medie_comenzi  
FROM  
    adresa a  
JOIN  
    comanda co ON a.id_adresa = co.id_adresa  
JOIN  
    client c ON a.id_utilizator = c.id_utilizator  
GROUP BY  
    a.judet  
HAVING  
    COUNT(DISTINCT c.id_utilizator) >= 1  
ORDER BY  
    valoare_medie_comenzi DESC;
```

Query Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4

All Rows Fetched: 5 in 0.002 seconds

JUDET	NUMAR_CLIENTI_UNICI	VALOARE_MEDIE_COMENZI
1 Constanța	1	300
2 București	1	200
3 Cluj	1	150
4 Iași	1	80
5 Timiș	1	50

```
SELECT
  p.id_produ,
  p.nume,
  NVL(p.pret, 0) AS pret_actual,
  DECODE(p.pret, NULL, 0, p.pret) AS pret_effectiv
```

```

FROM
    produse p
ORDER BY
    pret_efectiv DESC;

```

```

SELECT
    p.id_produc,
    p.nume,
    NVL(p.pret, 0) AS pret_actual,
    DECODE(p.pret, NULL, 0, p.pret) AS pret_efectiv
FROM
    produse p
ORDER BY
    pret_efectiv DESC;

```

--5. • utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

--• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

--presupunem că vrem să obținem o listă a comenzilor împreună cu detaliile corespunzătoare, iar pentru comenzile care au depășit data estimată de livrare, dorim să afișăm un mesaj special.

```

WITH Livrari AS (

```

Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

SQL | All Rows Fetched: 6 in 0.001 seconds

ID_PRODUS	NUME	PRET_ACTUAL	PRET_EFECTIV
1	Televizor Smart	1650	1650
2	Bluză cu Glugă	55	55
5	Călătorie în Tărâmul Minunilor	15	15
4	Lapte Demachiant	12.99	12.99
3	Pâine Integrală	3.5	3.5
6	Servetele umede	0	0

--5. • utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

--• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

--presupunem că vrem să obținem o listă a comenzilor împreună cu detaliile corespunzătoare, iar pentru comenzile care au depășit data estimată de livrare, dorim să afișăm un mesaj special.

```

WITH Livrari AS (
    SELECT
        c.id_comanda,
        c.data_comanda,
        c.estimare_livrare,
        CASE

```

```

        WHEN c.estimate_livrare < SYSDATE THEN 'Livrat intarziat'
        ELSE 'In transit'
    END AS status_livrare
FROM
    comanda c
)
SELECT
    l.id_comanda,
    TO_CHAR(l.data_comanda, 'DD-MON-YYYY HH24:MI') AS data_comanda_formatata,
    TO_CHAR(l.estimate_livrare, 'DD-MON-YYYY HH24:MI') AS estimate_livrare_formatata,
    l.status_livrare
FROM
    Livrari l;

```

WITH Livrari AS (

```

    SELECT
        c.id_comanda,
        c.data_comanda,
        c.estimate_livrare,
        CASE
            WHEN c.estimate_livrare < SYSDATE THEN 'Livrat intarziat'
            ELSE 'In transit'
        END AS status_livrare
    FROM
        comanda c
)
SELECT
    l.id_comanda,
    TO_CHAR(l.data_comanda, 'DD-MON-YYYY HH24:MI') AS data_comanda_formatata,
    TO_CHAR(l.estimate_livrare, 'DD-MON-YYYY HH24:MI') AS estimate_livrare_formatata,
    l.status_livrare
FROM
    Livrari l;

```

--EXERCITIUL 13:

Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x | Query Result 4 x | Query Result 5

SQL | All Rows Fetched: 5 in 0.001 seconds

ID_COMANDA	DATA_COMANDA_FORMATATA	ESTIMARE_LIVRARE_FORMATATA	STATUS_LIVRARE
1	28-AUG-2023 00:00	04-SEP-2023 16:48	In transit
2	29-AUG-2023 00:00	05-SEP-2023 16:48	In transit
3	30-AUG-2023 00:00	06-SEP-2023 16:48	In transit
4	31-AUG-2023 00:00	07-SEP-2023 16:48	In transit
5	01-SEP-2023 00:00	08-SEP-2023 16:48	In transit

13.Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

--EXERCITIUL 13:

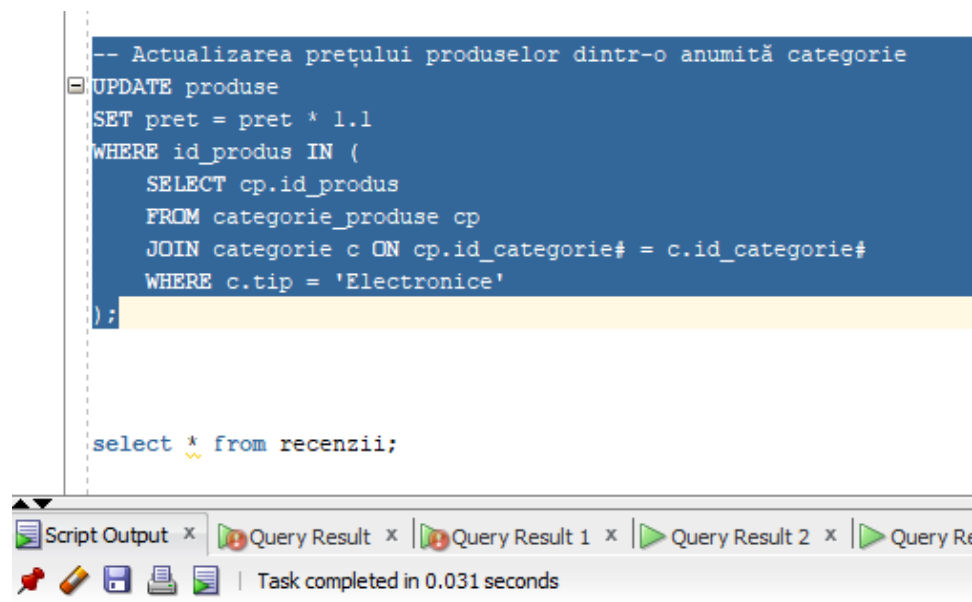
-- Actualizarea prețului produselor dintr-o anumită categorie

```

UPDATE produse
SET pret = pret * 1.1
WHERE id_produs IN (
    SELECT cp.id_produs
    FROM categorie_produce cp
    JOIN categorie c ON cp.id_categorie# = c.id_categorie#
    WHERE c.tip = 'Electronice'
);

```

```
select * from recenzii;
```



```

-- Actualizarea prețului produselor dintr-o anumită categorie
UPDATE produse
SET pret = pret * 1.1
WHERE id_produs IN (
    SELECT cp.id_produs
    FROM categorie_produce cp
    JOIN categorie c ON cp.id_categorie# = c.id_categorie#
    WHERE c.tip = 'Electronice'
);

select * from recenzii;

```

Script Output x | Query Result x | Query Result 1 x | Query Result 2 x | Query Result 3 x

Task completed in 0.031 seconds

2 rows updated.

-- actualizeaza timpul de livrare cu 10% pentru comenzile care au fost plasate dupa 20 august 2022

```

UPDATE comanda
SET estimare_livrare = estimare_livrare + (estimare_livrare - data_comanda) * 0.1
WHERE data_comanda > TO_DATE('2022-08-20', 'YYYY-MM-DD');

```

```
-- actualizeaza timpul de livrare cu 10% pentru comenzile care au fost plasate dupa 20 august 2022
UPDATE comanda
SET estimare_livrare = estimare_livrare + (estimare_livrare - data_comanda) * 0.1
WHERE data_comanda > TO DATE('2022-08-20', 'YYYY-MM-DD');

--EX 15:

--Cerere cu Outer-Join pe 4 tabele:
SELECT c.num, c.prenume, mp.data_asociere, a.judet
FROM client c
LEFT OUTER JOIN metode_plata mp ON c.id_utilizator = mp.id_utilizator
LEFT OUTER JOIN adresa a ON c.id_utilizator = a.id_utilizator
LEFT OUTER JOIN recenzii r ON c.id_utilizator = r.id_client;
--Cerere cu Operația Division:
-- Această cerere va selecta clienții care au recenzii pentru toate produsele dintr-o anumită categori
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x Query Result

Task completed in 0.044 seconds

2 rows updated.

5 rows updated.

-- operatie de suprimare:

```
DELETE FROM recenzii WHERE id_client = <id_client>;
DELETE FROM comanda WHERE id_client = <id_client>;
DELETE FROM transport_produce WHERE id_produc# IN (SELECT id_produc FROM produse WHERE
id_client = <id_client>);
DELETE FROM produse WHERE id_client = <id_client>;
DELETE FROM adresa WHERE id_utilizator = <id_client>;
DELETE FROM metode_plata WHERE id_utilizator = <id_client>;
DELETE FROM client WHERE id_utilizator = <id_client>;
```

15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n

--EX 15:

--Cerere cu Outer-Join pe 4 tabele:

```
SELECT c.num, c.prenume, mp.data_asociere, a.judet
FROM client c
LEFT OUTER JOIN metode_plata mp ON c.id_utilizator = mp.id_utilizator
LEFT OUTER JOIN adresa a ON c.id_utilizator = a.id_utilizator
LEFT OUTER JOIN recenzii r ON c.id_utilizator = r.id_client;
```

```
--Cerere cu Outer-Join pe 4 tabele:
SELECT c.num, c.prenume, mp.data_asociere, a.judet
FROM client c
LEFT OUTER JOIN metode_plata mp ON c.id_utilizator = mp.id_utilizator
LEFT OUTER JOIN adresa a ON c.id_utilizator = a.id_utilizator
LEFT OUTER JOIN recenzii r ON c.id_utilizator = r.id_client;

--Cerere cu Operația Division:
-- Această cerere va selecta clienții care au recenzii pentru toate produse:

SELECT
    c.id_utilizator,
    c.num,
    c.prenume
FROM
```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x

SQL | All Rows Fetched: 9 in 0.002 seconds

	NUME	PRENUME	DATA_ASOCIERE	JUDET
1	Ionescu	Alex	29-AUG-23	Cluj
2	Mihai	Andreea	30-AUG-23	Timiș
3	Dumitru	Mihai	31-AUG-23	Iași
4	Georgescu	Elena	01-SEP-23	Constanța
5	Ionescu	Alex	29-AUG-23	Cluj
6	Mihai	Andreea	30-AUG-23	Timiș
7	Dumitru	Mihai	31-AUG-23	Iași
8	Georgescu	Elena	01-SEP-23	Constanța
9	Popescu	Ana	28-AUG-23	București

--Cerere cu Operația Division:
 -- Această cerere va selecta clienții care au recenzii pentru toate produsele dintr-o anumită categorie.

```
SELECT
    c.id_utilizator,
    c.num,
    c.prenume
FROM
    client c
WHERE NOT EXISTS (
    SELECT p.id_produș
    FROM categorie_produș cp
    JOIN produș p ON cp.id_produș = p.id_produș
    JOIN categorie cat ON cp.id_categorie# = cat.id_categorie#
    WHERE cat.tip = 'Electronice' -- Categoria dorită
    MINUS
    SELECT r.id_produș
    FROM recenzii r
```


WHERE r.id_client = c.id_utilizator

);

```
SELECT
    c.id_utilizator,
    c.nume,
    c.prenume
FROM
    client c
WHERE NOT EXISTS (
    SELECT p.id_produș
    FROM categorie_produș cp
    JOIN produse p ON cp.id_produș = p.id_produș
    JOIN categorie cat ON cp.id_categorie# = cat.id_categorie#
    WHERE cat.tip = 'Electronice' -- Categoria dorită
    MINUS
    SELECT r.id_produș
    FROM recenzii r
    WHERE r.id_client = c.id_utilizator
);

--cerere care implementează analiza top-n
-- Cerere pentru a obține N produse împreună cu recenziile lor:

SELECT
    p.id_produș,
```

--cerere care implementează analiza top-n

-- Cerere pentru a obține N produse împreună cu recenziile lor:

```
SELECT
    p.id_produș,
    p.nume AS nume_produș,
    r.id_recenzie,
    r.id_client
FROM
    produse p
LEFT JOIN
    recenzii r ON p.id_produș = r.id_produș
WHERE ROWNUM <= 3; -- N
```

```

SELECT
  p.id_produc,
  p.num AS nume_produc,
  r.id_recenzie,
  r.id_client
FROM
  produse p
LEFT JOIN
  recenzii r ON p.id_produc = r.id_produc
WHERE ROWNUM <= 3; -- N

select * from client;

```

Script Output x Query Result x Query Result 1 x Query Result 2 x Query Result 3 x Query Result 4 x

SQL | All Rows Fetched: 5 in 0.001 seconds

ID_UTILIZATOR	NUME	PRENUME	USERNAME	EMAIL	PAROLA
1	1 Popescu	Ana	ana.popescu	ana.popescu@email.com	parola123
2	2 Ionescu	Alex	alex.ionescu	alex.ionescu@email.com	parola456
3	3 Mihai	Andreea	andreea.mihai	andreea.mihai@email.com	parola789
4	4 Dumitru	Mihai	mihai.dumitru	mihai.dumitru@email.com	parolaabc
5	5 Georgescu	Elena	elena.georgescu	elena.georgescu@email.com	parolaxyz