

Baza de date a brandului Oysho

Sisteme de baze de date

2024-2025

Aldea Alexia, Master IS

Grupa 406

Cuprins

1. Prezentarea bazei de date aleasă
2. Realizarea diagramei ERD
3. Realizarea diagramei conceptuale
4. Transformarea sistemului conceptual într-un design logic
5. Convertirea design-ului logic într-un design fizic în FN3
6. Implementarea tabelor în Oracle, adaugand chei primare, constrangeri si date
7. Ilustrarea noțiunilor de PL/SQL învățate

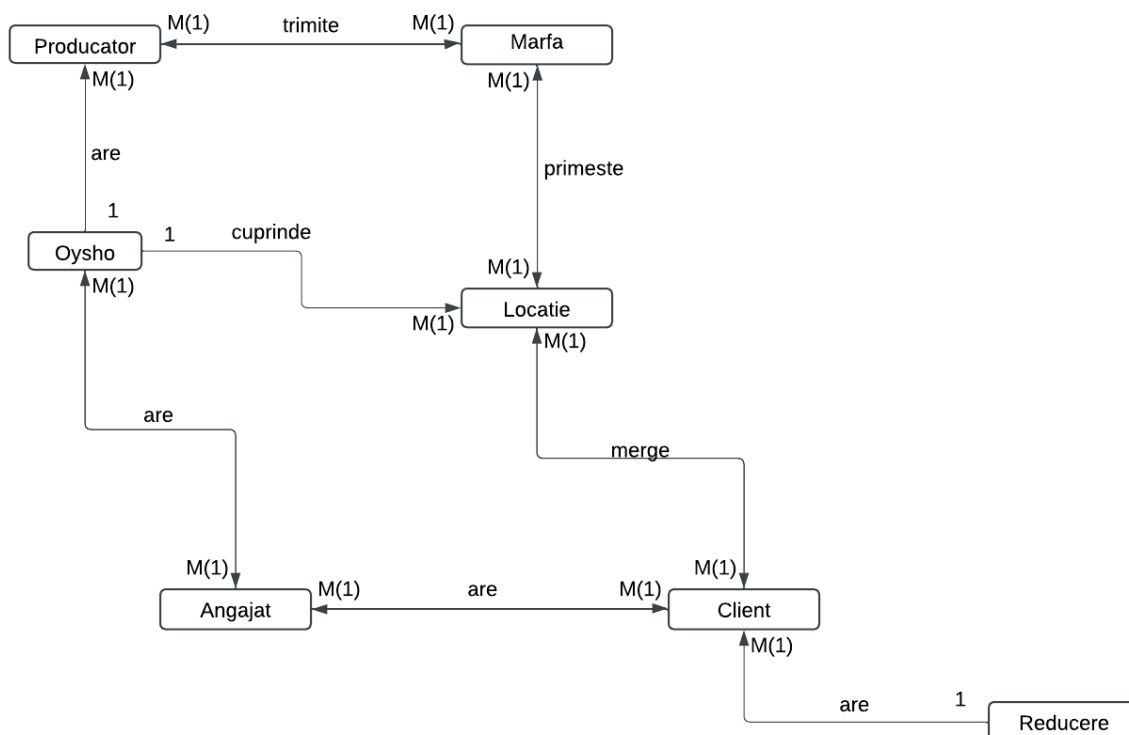
1. Prezentarea bazei de date aleasă

În acest proiect voi realiza baza de date a lanțului de magazine Oysho. În primul rând, acesta are unul sau mai mulți producători, ce îi livrează haine. Desigur, unele haine pot veni cu un defect din fabricație la marfă, urmând ca acestea să fie trimise înapoi la producător de către magazine.

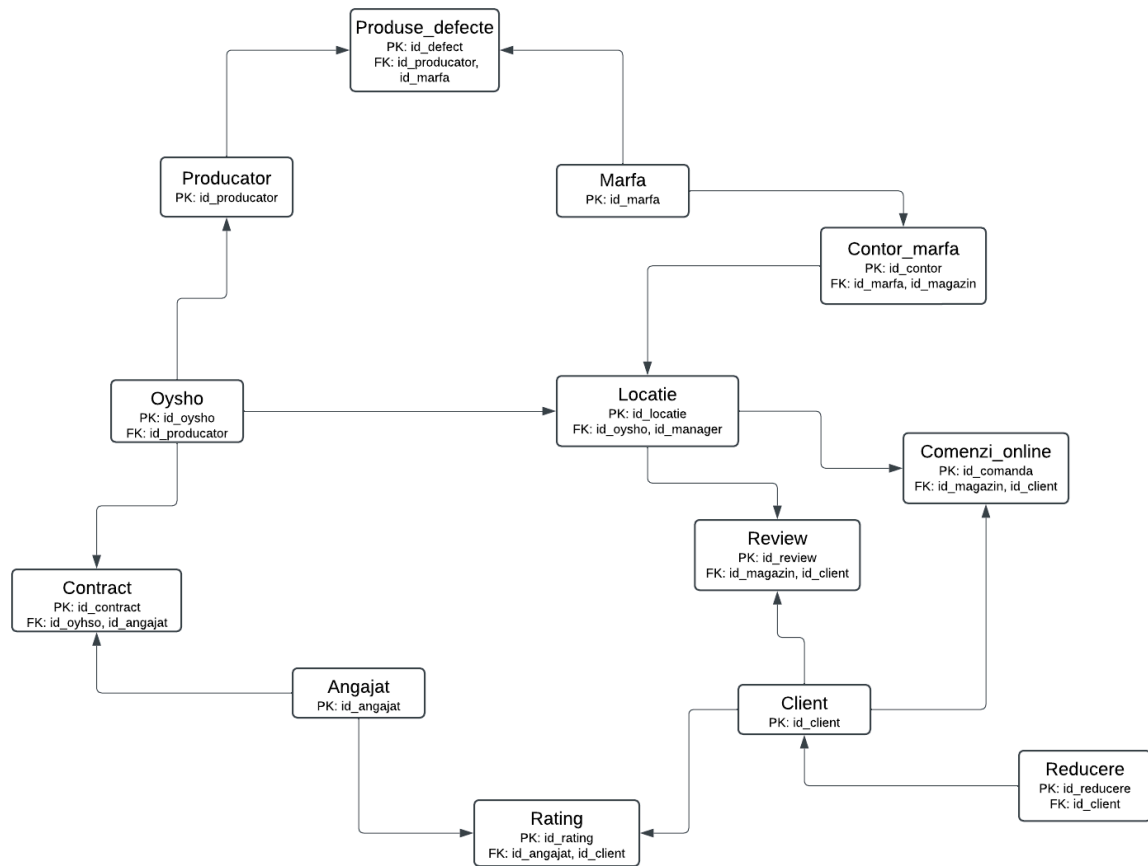
Brandul Oysho are unul sau mai multe contracte cu mai mulți angajați. Pentru fiecare angajat vom reține câteva date de bază despre el, iar în contract, la fel. Un angajat lucrează la un singur mall, dintr-o singură locație. Brandul Oysho are magazine în diferite locuri ale lumii, în mai multe mall-uri din orașe. Fiecare magazin primește marfă, printr-un contor, care îi spune câte cutii se vor livra în ziua X. De asemenea, un magazin Oysho poate primi și comenzi online, de care trebuie să se ocupe (să le împacheteze).

Clienții care vizitează un magazin Oysho îi pot lăsa un review, iar angajaților, o notă, pentru a-i ajuta pe viitorii clienți să își creeze o părere despre experiența din fiecare magazin Oysho și pentru a-l vizita pe cel potrivit pentru ei. De asemenea, un client poate avea și o reducere, începând cu o anumită dată și cu o perioadă de X ani (decisă de Oysho, în funcție de cât de multe cumpărături face la acest brand).

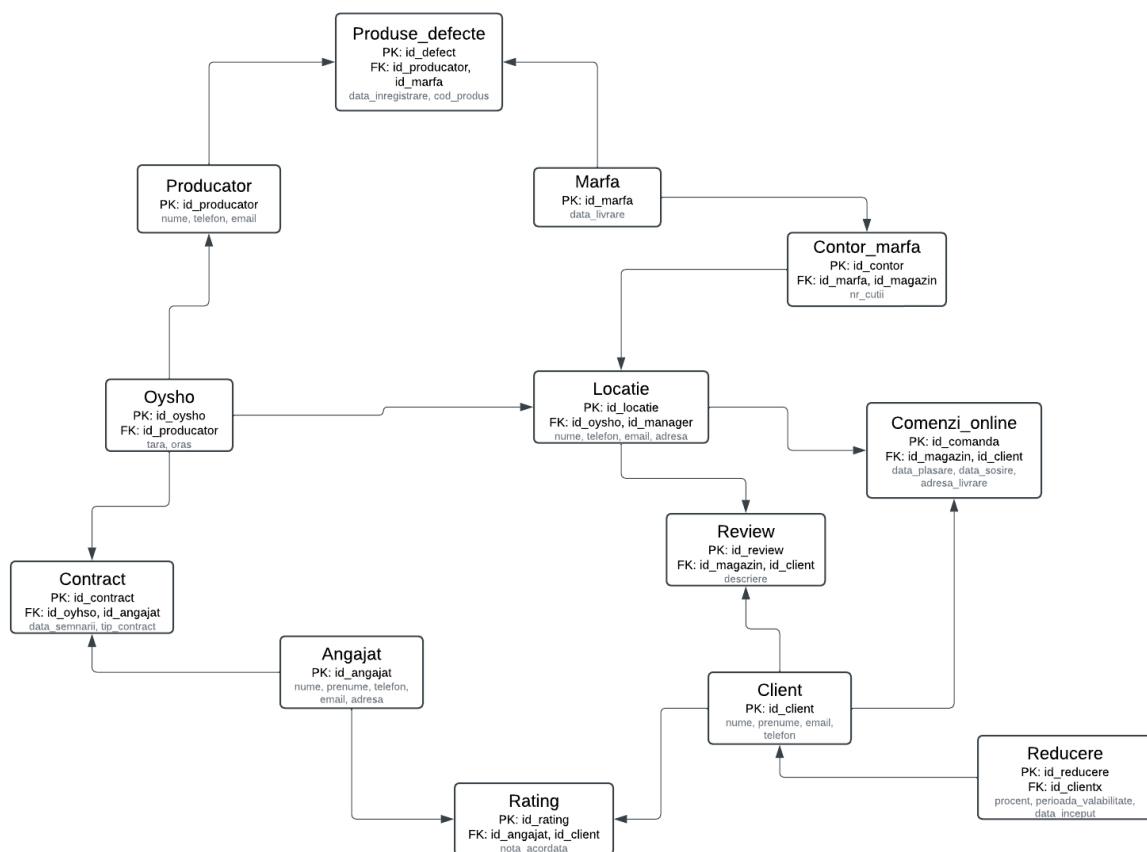
2. Realizarea diagramei ERD



3. Realizarea diagramei conceptuale



4. Transformarea sistemului conceptual într-un design logic



5. Convertirea design-ului logic într-un design fizic în FN3

-> un exemplu de atribut repetitiv

În versiunea inițială a tabelului "Locatie", atributul "review" putea conține valori multivaloare (de exemplu, mai multe recenzii pentru aceeași locație, toate stocate într-o singură celulă). Acest lucru încalcă regulile Primei Forme Normale (1NF), care specifică faptul că fiecare coloană trebuie să conțină valori atomice, adică fiecare celulă trebuie să conțină o singură valoare indivizibilă.

Pentru a respecta 1NF, am decis să creez un tabel separat, denumit "Review", în care fiecare recenzie individuală este înregistrată pe o linie separată. Am realizat o relație „one-to-many” între tabelul "Locatie" și tabelul "Review", unde fiecare locație poate avea mai multe recenzii asociate, dar fiecare recenzie este stocată ca o valoare unică.

Prin această abordare:

- Am eliminat attributele multivaloare din tabelul "Locatie".
- Am asigurat că fiecare înregistrare respectă regulile 1NF, în care toate valorile sunt atomice.

-> un exemplu de tabel care este în FN1, dar nu în FN2

În versiunea inițială a tabelului "Client", atributul "reducere" era inclus, permițând stocarea informațiilor despre reduceri în același tabel cu datele clientului. Deși această structură

respectă Prima Formă Normală (1NF), ea nu respectă regulile celei de-a Doua Forme Normale (2NF), care impune ca toate atributele să depindă de cheia primară și să nu existe dependențe parțiale.

Pentru a aduce tabela în 2NF, am creat un tabel separat denumit `Reducere`, în care fiecare reducere individuală este înregistrată separat și asociată printr-o relație „one-to-many” cu tabela `Client`. Astfel, fiecare reducere a unui client este stocată independent și este ușor de gestionat și actualizat.

Prin această abordare:

- Am eliminat dependențele parțiale dintre atributele din `Client` și `Reducere`.
- Am adus tabelul în conformitate cu regulile 2NF, păstrând datele despre reduceri într-un tabel separat pentru o organizare mai clară și eficientă.

-> un exemplu de tabel relațional care este în FN2, dar nu în FN3.

În versiunea inițială a tabelii “Producator”, anumite atribute conțineau informații ce nu depindeau direct de cheia primară, ci de alți factori din contextul Oysho, încălcând regulile celei de-a Treia Forme Normale (3NF). A treia formă normală impune ca toate atributele non-cheie să fie determinate în mod direct și exclusiv de cheia primară, fără dependențe tranzitive.

Pentru a aduce tabela în 3NF, am creat un tabel separat, denumit `Oysho`, în care fiecare detaliu legat de brandul Oysho este înregistrat distinct. Acest tabel are o relație clară cu `Producator`, eliminând astfel dependențele tranzitive și asigurând că fiecare atribut este determinat direct doar de cheia primară a tabelului în care se află.

Prin această abordare:

- Am eliminat dependențele tranzitive dintre tabelele `Producator` și `Oysho`.
- Am adus structura bazei de date în conformitate cu regulile 3NF, făcând baza de date mai clară, mai eficientă și mai ușor de întreținut.

6. Implementarea tabelelor în Oracle, adaugand chei primare, constrangeri si date

```
CREATE TABLE producator (  
    id_producator NUMBER(5) CONSTRAINT pk_producator PRIMARY KEY,  
    nume VARCHAR(20) CONSTRAINT nume_producator NOT NULL,  
    telefon VARCHAR(15) CONSTRAINT telefon_producator NOT NULL,  
    email VARCHAR(20) CONSTRAINT email_producator NOT NULL  
);  
  
INSERT INTO producator  
VALUES (1, 'BrandMarck', '0345678322', 'contact@gmail.com');  
INSERT INTO producator  
VALUES (2, 'ClothesForEveryone', '0789345655',  
'clothes_1@gmail.com');
```

```

INSERT INTO producator
VALUES (3, 'MadeInMorocco', '0345687732', 'callus@yahoo.com');
INSERT INTO producator
VALUES (4, 'HommyTilfinger', '034544377', 'hommy_t@gmail.com');
INSERT INTO producator
VALUES (5, 'BugoHoss', '0456788124', 'bugo12@gmail.com');

COMMIT;

SELECT * FROM producator;

```

```

1 ✓ CREATE TABLE producator (
2     id_producator NUMBER(5) CONSTRAINT pk_producator PRIMARY KEY,
3     nume VARCHAR(20) CONSTRAINT nume_producator NOT NULL,
4     telefon VARCHAR(15) CONSTRAINT telefon_producator NOT NULL,
5     email VARCHAR(20) CONSTRAINT email_producator NOT NULL
6 );
7
8 ✓ INSERT INTO producator
9 VALUES (1, 'BrandMarck', '0345678322', 'contact@gmail.com');

```

Statement processed.

ID_PRODUCATOR	NUME	TELEFON	EMAIL
1	BrandMarck	0345678322	contact@gmail.com
2	ClothesForEveryone	0789345655	clothes_1@gmail.com
3	MadeInMorocco	0345687732	callus@yahoo.com
4	HommyTilfinger	034544377	hommy_t@gmail.com
5	BugoHoss	0456788124	bugo12@gmail.com

```

CREATE TABLE marfa (
    id_marfa NUMBER(5) CONSTRAINT pk_marfa PRIMARY KEY,
    data_livrare DATE CONSTRAINT data_marfa NOT NULL
);

INSERT INTO marfa
VALUES (6, TO_DATE('15-03-2023', 'DD-MM-YYYY'));
INSERT INTO marfa
VALUES (7, TO_DATE('01-03-2023', 'DD-MM-YYYY'));
INSERT INTO marfa
VALUES (8, TO_DATE('15-04-2023', 'DD-MM-YYYY'));
INSERT INTO marfa

```

```
VALUES (9, TO_DATE('01-04-2023', 'DD-MM-YYYY'));
INSERT INTO marfa
VALUES (10, TO_DATE('01-05-2023', 'DD-MM-YYYY'));

COMMIT;

SELECT * FROM marfa;
```

```
24
25 ✓ CREATE TABLE marfa (
26     id_marfa NUMBER(5) CONSTRAINT pk_marfa PRIMARY KEY,
27     data_livrare DATE CONSTRAINT data_marfa NOT NULL
28 );|
29
30 ✓ INSERT INTO marfa
31 VALUES (6, TO_DATE('15-03-2023', 'DD-MM-YYYY'));
32 ✓ INSERT INTO marfa
33 VALUES (7, TO_DATE('01-03-2023', 'DD-MM-YYYY'));
```

Statement processed.

ID_MARFA	DATA_LIVRARE
6	15-MAR-23
7	01-MAR-23
8	15-APR-23
9	01-APR-23
10	01-MAY-23

```
CREATE TABLE produse_defecte (
    id_defect NUMBER(5) CONSTRAINT pk_prod_defect PRIMARY KEY,
    data_inregistrare DATE CONSTRAINT data_prod_defect NOT NULL,
    cod_producat NUMBER(15) CONSTRAINT cod_prod_defect NOT NULL,
    id_producator NUMBER(5),
    CONSTRAINT fk_prod_def FOREIGN KEY (id_producator) REFERENCES
    producator (id_producator),
    id_marfa NUMBER(5),
    CONSTRAINT fk_prod_def FOREIGN KEY (id_marfa) REFERENCES
```



```
marfa (id_marfa)
);

INSERT INTO produse_defecte
VALUES (11, TO_DATE('15-01-2023', 'DD-MM-YYYY'), 1234670800, 1,
7);
INSERT INTO produse_defecte
VALUES (12, TO_DATE('03-04-2022', 'DD-MM-YYYY'), 2345674456, 4,
10);
INSERT INTO produse_defecte
VALUES (13, TO_DATE('16-02-2023', 'DD-MM-YYYY'), 1456239485, 5,
8);
INSERT INTO produse_defecte
VALUES (14, TO_DATE('17-12-2022', 'DD-MM-YYYY'), 3456233466, 3,
9);
INSERT INTO produse_defecte
VALUES (15, TO_DATE('18-01-2023', 'DD-MM-YYYY'), 4332456800, 2,
6);
INSERT INTO produse_defecte
VALUES (16, TO_DATE('20-01-2023', 'DD-MM-YYYY'), 2331441500, 2,
7);
INSERT INTO produse_defecte
VALUES (17, TO_DATE('14-12-2022', 'DD-MM-YYYY'), 2230450678, 3,
10);
INSERT INTO produse_defecte
VALUES (18, TO_DATE('01-02-2023', 'DD-MM-YYYY'), 4523900800, 5,
9);
INSERT INTO produse_defecte
VALUES (19, TO_DATE('22-11-2022', 'DD-MM-YYYY'), 4567890957, 1,
8);
INSERT INTO produse_defecte
VALUES (20, TO_DATE('13-01-2023', 'DD-MM-YYYY'), 3453230567, 3,
9);

COMMIT;

SELECT * FROM produse_defecte;
```

```

46 CREATE TABLE produse_defecte (
47     id_defect NUMBER(5) CONSTRAINT pk_prod_defect PRIMARY KEY,
48     data_inregistrare DATE CONSTRAINT data_prod_defect NOT NULL,
49     cod_produs NUMBER(15) CONSTRAINT cod_prod_defect NOT NULL,
50     id_producator NUMBER(5),
51     CONSTRAINT fk_prod_def FOREIGN KEY (id_producator) REFERENCES producator (id_producator),
52     id_marfa NUMBER(5),
53     CONSTRAINT fk_prod_deff FOREIGN KEY (id_marfa) REFERENCES marfa (id_marfa)
54 );

```

ID_DEFECT	DATA_INREGISTRARE	COD_PRODUS	ID_PRODUCATOR	ID_MARFA
11	15-JAN-23	1234670800	1	7
12	03-APR-22	2345674456	4	10
13	16-FEB-23	1456239485	5	8
14	17-DEC-22	3456233466	3	9
15	18-JAN-23	4332456800	2	6
16	20-JAN-23	2331441500	2	7
17	14-DEC-22	2220450678	2	10

```

CREATE TABLE oysho (
    id_oysho NUMBER(5) CONSTRAINT pk_oysho PRIMARY KEY,
    tara VARCHAR(20) CONSTRAINT tara_oysho NOT NULL,
    oras VARCHAR(20) CONSTRAINT oras_oysho NOT NULL,
    id_producator NUMBER(5),
    CONSTRAINT fk_oysho FOREIGN KEY (id_producator) REFERENCES
producator (id_producator)
);

```

```

INSERT INTO oysho
VALUES (21, 'Romania', 'Bucuresti', 1);
INSERT INTO oysho
VALUES (22, 'Spania', 'Madrid', 4);
INSERT INTO oysho
VALUES (23, 'Italia', 'Roma', 5);
INSERT INTO oysho
VALUES (24, 'Franta', 'Paris', 2);
INSERT INTO oysho
VALUES (25, 'Portugalia', 'Lisabona', 3);

```

```
COMMIT;
```

```
SELECT * FROM oysho;
```

```

83 CREATE TABLE oysho (
84     id_oysho NUMBER(5) CONSTRAINT pk_oysho PRIMARY KEY,
85     tara VARCHAR(20) CONSTRAINT tara_oysho NOT NULL,
86     oras VARCHAR(20) CONSTRAINT oras_oysho NOT NULL,
87     id_producator NUMBER(5),
88     CONSTRAINT fk_oysho FOREIGN KEY (id_producator) REFERENCES producator (id_producator)
89 );
90
91 INSERT INTO oysho
92 VALUES (21, 'Romania', 'Bucuresti', 1);
93 INSERT INTO oysho

```

Statements processed:

ID_OYSHO	TARA	ORAS	ID_PRODUCATOR
21	Romania	Bucuresti	1
22	Spania	Madrid	4
23	Italia	Roma	5
24	Franta	Paris	2
25	Portugalia	Lisabona	3

```

CREATE TABLE angajati (
    id_angajat NUMBER(5) CONSTRAINT pk_angajat PRIMARY KEY,
    nume_angajat VARCHAR(50) CONSTRAINT nume_angajat NOT NULL,
    prenume_angajat VARCHAR(50) CONSTRAINT prenume_angajat NOT
NULL,
    telefon_angajat VARCHAR(50) CONSTRAINT telefon_angajat NOT
NULL,
    email_angajat VARCHAR(50) CONSTRAINT email_angajat NOT NULL,
    adresa_angajat VARCHAR(50) CONSTRAINT adresa_angajat NOT NULL
);

INSERT INTO angajati
VALUES (26, 'Florea', 'Catalin', '0756345786',
'florea@gmail.com', 'Florilor 6');
INSERT INTO angajati
VALUES (27, 'Adam', 'Antonio', '0746330454', 'andy_ad@gmail.com',
'Tepes Voda 25');
INSERT INTO angajati
VALUES (28, 'Vulcanescu', 'Mircea', '0745555676',
'mirciulical23@gmail.com', 'Rezervelor 4');
INSERT INTO angajati
VALUES (29, 'Ceausu', 'Radu', '0789888543', 'radulish@yahoo.com',
'Anton Pann 32');
INSERT INTO angajati
VALUES (30, 'Craciun', 'Alexandru', '0734565723',
'christmas_alex@gmail.com', 'Mircea Voievod 67');
INSERT INTO angajati
VALUES (36, 'Aldea', 'Alexia', '0756888187', 'alexia@gmail.com',
'Constructorilor 29');
INSERT INTO angajati

```

```

VALUES (37, 'Rosu', 'Cristina', '0731456777',
'cris_rosu@gmail.com', 'Teiul Doamnnei 12');
INSERT INTO angajati
VALUES (38, 'Luta', 'Alexandra', '0733454558', 'kiwi@gmail.com',
'Grozavesti 34');
INSERT INTO angajati
VALUES (39, 'Pentu', 'Miruna', '0735464889',
'mirus.spirus@gmail.com', 'Bucurestii Noi 1');
INSERT INTO angajati
VALUES (40, 'Bozeanu', 'Mirela', '0763888176',
'miree_boze@gmail.com', 'Morarilor 90');

COMMIT;

SELECT * FROM angajati;

```

```

109 v CREATE TABLE angajati (
110     id_angajat NUMBER(5) CONSTRAINT pk_angajat PRIMARY KEY,
111     nume_angajat VARCHAR(50) CONSTRAINT nume_angajat NOT NULL,
112     prenume_angajat VARCHAR(50) CONSTRAINT prenume_angajat NOT NULL,
113     telefon_angajat VARCHAR(50) CONSTRAINT telefon_angajat NOT NULL,
114     email_angajat VARCHAR(50) CONSTRAINT email_angajat NOT NULL,
115     adresa_angajat VARCHAR(50) CONSTRAINT adresa_angajat NOT NULL
116 );
117
118 v INSERT INTO angajati

```

ID_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	TELEFON_ANGAJAT	EMAIL_ANGAJAT	ADRESA_ANGAJAT
26	Florea	Catalin	0756345786	florea@gmail.com	Florilor 6
27	Adam	Antonio	0746330454	andy_ad@gmail.com	Tepes Voda 25
28	Vulcanescu	Mircea	0745555676	mirciulica123@gmail.com	Rezervelor 4
29	Ceausu	Radu	0789888543	radulish@yahoo.com	Anton Pann 32
30	Craciun	Alexandru	0734565723	christmas_alex@gmail.com	Mircea Voievod 67
36	Aldea	Alexia	0756888187	alexia@gmail.com	Constructorilor 29

```

CREATE TABLE locatie (
    id_magazin NUMBER(5) CONSTRAINT pk_locatie PRIMARY KEY,
    nume_magazin VARCHAR(20) CONSTRAINT nume_locatie NOT NULL,
    telefon VARCHAR(20) CONSTRAINT telefon_magazin NOT NULL,
    email VARCHAR(50) CONSTRAINT email_magazin NOT NULL,
    adresa VARCHAR(100) CONSTRAINT adresa_magazin NOT NULL,
    id_oysho NUMBER(5),
    CONSTRAINT fk_magazin FOREIGN KEY (id_oysho) REFERENCES oysho
(id_oysho),
    id_manager NUMBER(5),
    CONSTRAINT fk_magg FOREIGN KEY (id_manager) REFERENCES
angajati (id_angajat)
);

INSERT INTO locatie
VALUES (31, 'Oysho Promenada', '031455678',
'oysho_prome@gmail.com', 'Barbu Vacarescu 8', 21, 30);

```

```

INSERT INTO locatie
VALUES (32, 'Oysho SunPlaza', '0314567890',
'oysho_sunplaza@gmail.com', 'Vacaresti 32', 22, 26);
INSERT INTO locatie
VALUES (33, 'Oysho Cotroceni', '031400500',
'oysho_afi@gmail.com', 'Cotroceni 35', 23, 29);
INSERT INTO locatie
VALUES (34, 'Oysho Baneasa', '031444556',
'oysho_baneasa@gmail.com', 'Bucuresti-Ploiesti 14', 24, 28);
INSERT INTO locatie
VALUES (35, 'Oysho ParkLake', '031446673',
'oysho_parklake@gmail.com', 'Liviu Rebreanu 5', 25, 27);

COMMIT;

SELECT * FROM locatie;

```

```

145 CREATE TABLE locatie (
146     id_magazin NUMBER(5) CONSTRAINT pk_locatie PRIMARY KEY,
147     nume_magazin VARCHAR(20) CONSTRAINT nume_locatie NOT NULL,
148     telefon VARCHAR(20) CONSTRAINT telefon_magazin NOT NULL,
149     email VARCHAR(50) CONSTRAINT email_magazin NOT NULL,
150     adresa VARCHAR(100) CONSTRAINT adresa_magazin NOT NULL,
151     id_oysho NUMBER(5),
152     CONSTRAINT fk_magazin FOREIGN KEY (id_oysho) REFERENCES oysho (id_oysho),
153     id_manager NUMBER(5),
154     CONSTRAINT fk_magg FOREIGN KEY (id_manager) REFERENCES angajati (id_angajat)
155 );
156
157 INSERT INTO locatie

```

ID_MAGAZIN	NUME_MAGAZIN	TELEFON	EMAIL	ADRESA	ID_OYSHO	ID_MANAGER
31	Oysho Promenada	031455678	oysho_prome@gmail.com	Barbu Vacarescu 8	21	30
32	Oysho SunPlaza	0314567890	oysho_sunplaza@gmail.com	Vacaresti 32	22	26
33	Oysho Cotroceni	031400500	oysho_afi@gmail.com	Cotroceni 35	23	29
34	Oysho Baneasa	031444556	oysho_baneasa@gmail.com	Bucuresti-Ploiesti 14	24	28
35	Oysho ParkLake	031446673	oysho_parklake@gmail.com	Liviu Rebreanu 5	25	27

```

CREATE TABLE contract (
    id_contract NUMBER(5) CONSTRAINT pk_contr PRIMARY KEY,
    data_semnarii DATE CONSTRAINT data_contract NOT NULL,
    tip_contract VARCHAR(20) CONSTRAINT tip_contract NOT NULL,
    id_oysho NUMBER(5),
    CONSTRAINT fk_contr_oysho FOREIGN KEY (id_oysho) REFERENCES
oysho (id_oysho),
    id_angajat NUMBER(5),
    CONSTRAINT fk_contr_angajat FOREIGN KEY (id_angajat)
REFERENCES angajati (id_angajat)
);

INSERT INTO contract
VALUES (41, TO_DATE('15-01-2023', 'DD-MM-YYYY'), 'Full-time', 21,
29);

```

```
INSERT INTO contract
VALUES (42, TO_DATE('01-02-2023', 'DD-MM-YYYY'), 'Part-time', 22,
26);
INSERT INTO contract
VALUES (43, TO_DATE('03-03-2023', 'DD-MM-YYYY'), 'Full-time', 23,
26);
INSERT INTO contract
VALUES (44, TO_DATE('15-04-2023', 'DD-MM-YYYY'), 'Full-time', 24,
27);
INSERT INTO contract
VALUES (45, TO_DATE('20-04-2023', 'DD-MM-YYYY'), 'Part-time', 25,
30);
INSERT INTO contract
VALUES (46, TO_DATE('18-02-2023', 'DD-MM-YYYY'), 'Full-time', 22,
31);
INSERT INTO contract
VALUES (47, TO_DATE('31-01-2020', 'DD-MM-YYYY'), 'Part-time', 25,
40);
INSERT INTO contract
VALUES (48, TO_DATE('16-10-2021', 'DD-MM-YYYY'), 'Part-time', 24,
36);
INSERT INTO contract
VALUES (49, TO_DATE('25-09-2021', 'DD-MM-YYYY'), 'Part-time', 21,
37);
INSERT INTO contract
VALUES (50, TO_DATE('19-07-2022', 'DD-MM-YYYY'), 'Part-time', 23,
38);

COMMIT;

SELECT * FROM contract;
```

```

174 CREATE TABLE contract (
175     id_contract NUMBER(5) CONSTRAINT pk_contr PRIMARY KEY,
176     data_semnarii DATE CONSTRAINT data_contract NOT NULL,
177     tip_contract VARCHAR(20) CONSTRAINT tip_contract NOT NULL,
178     id_oyshe NUMBER(5),
179     CONSTRAINT fk_contr_oyshe FOREIGN KEY (id_oyshe) REFERENCES oyshe (id_oyshe),
180     id_angajat NUMBER(5),
181     CONSTRAINT fk_contr_angajat FOREIGN KEY (id_angajat) REFERENCES angajati (id_angajat)
182 );
183
184 INSERT INTO contract
185 VALUES (41, TO_DATE('15-01-2023', 'DD-MM-YYYY'), 'Full-time', 21, 29);

```

ID_CONTRACT	DATA_SEMNARII	TIP_CONTRACT	ID_OYSHO	ID_ANGAJAT
41	15-JAN-23	Full-time	21	29
42	01-FEB-23	Part-time	22	26
43	03-MAR-23	Full-time	23	26
44	15-APR-23	Full-time	24	27
45	20-APR-23	Part-time	25	30

```

CREATE TABLE contor_marfa (
    id_contor NUMBER(5) CONSTRAINT pk_contor PRIMARY KEY,
    nr_cutii NUMBER(5) CONSTRAINT cutii_contor NOT NULL,
    id_marfa NUMBER(5),
    CONSTRAINT fk_contor FOREIGN KEY (id_marfa) REFERENCES marfa
(id_marfa),
    id_magazin NUMBER(5),
    CONSTRAINT fk_contorr FOREIGN KEY (id_magazin) REFERENCES
locatie (id_magazin)
);

```

```

INSERT INTO contor_marfa
VALUES (51, 156, 6, 34);
INSERT INTO contor_marfa
VALUES (52, 144, 8, 33);
INSERT INTO contor_marfa
VALUES (53, 200, 9, 35);
INSERT INTO contor_marfa
VALUES (54, 45, 10, 31);
INSERT INTO contor_marfa
VALUES (55, 57, 8, 32);
INSERT INTO contor_marfa
VALUES (56, 230, 7, 35);
INSERT INTO contor_marfa
VALUES (57, 400, 6, 33);
INSERT INTO contor_marfa
VALUES (58, 245, 9, 32);
INSERT INTO contor_marfa
VALUES (59, 120, 7, 35);
INSERT INTO contor_marfa

```

```
VALUES (60, 89, 9, 34);

COMMIT;

SELECT * FROM contor_marfa;
```

```
211 ✓ CREATE TABLE contor_marfa (
212     id_contor NUMBER(5) CONSTRAINT pk_contor PRIMARY KEY,
213     nr_cutii NUMBER(5) CONSTRAINT cutii_contor NOT NULL,
214     id_marfa NUMBER(5),
215     CONSTRAINT fk_contor FOREIGN KEY (id_marfa) REFERENCES marfa (id_marfa),
216     id_magazin NUMBER(5),
217     CONSTRAINT fk_contorr FOREIGN KEY (id_magazin) REFERENCES locatie (id_magazin)
218 );
219
220 ✓ INSERT INTO contor_marfa
221 VALUES (51, 156, 6, 34);
222 ✓ INSERT INTO contor_marfa
223 VALUES (52, 144, 8, 33);
```

ID_CONTOR	NR_CUTII	ID_MARFA	ID_MAGAZIN
51	156	6	34
52	144	8	33
53	200	9	35
54	45	10	31
55	57	8	32

```
CREATE TABLE client (
    id_client NUMBER(5) CONSTRAINT pk_client PRIMARY KEY,
    nume_client VARCHAR(20) CONSTRAINT ck_nume_client NOT NULL,
    prenume_client VARCHAR(20) CONSTRAINT ck_prenume_client NOT
NULL,
    email_client VARCHAR(50) CONSTRAINT ck_email_client NOT NULL,
    telefon_client VARCHAR(20) CONSTRAINT ck_telefon_client NOT
NULL
);

INSERT INTO client
VALUES (61, 'Pirinei', 'Ionut', 'ionut@gmail.com', '0756789900');
INSERT INTO client
VALUES (62, 'Predeanu', 'Mariana', 'mary12@gmail.com',
'0745677899');
INSERT INTO client
VALUES (63, 'Ciocos', 'Bianca', 'bye_eu@gmail.com',
'0765444143');
INSERT INTO client
VALUES (64, 'Teodoroiu', 'Cristina', 'cris_teo@gmail.com',
'0746767888');
```



```

INSERT INTO client
VALUES (65, 'Mircea', 'Carina', 'cary_cary@gmail.com',
'0745883464');

COMMIT;

SELECT * FROM client;

```

```

247 ✓ CREATE TABLE client (
248     id_client NUMBER(5) CONSTRAINT pk_client PRIMARY KEY,
249     nume_client VARCHAR(20) CONSTRAINT ck_nume_client NOT NULL,
250     prenume_client VARCHAR(20) CONSTRAINT ck_prenume_client NOT NULL,
251     email_client VARCHAR(50) CONSTRAINT ck_email_client NOT NULL,
252     telefon_client VARCHAR(20) CONSTRAINT ck_telefon_client NOT NULL
253 );
254
255 ✓ INSERT INTO client
256 VALUES (61, 'Pirinei', 'Ionut', 'ionut@gmail.com', '0756789900');
257 ✓ INSERT INTO client
258 VALUES (62, 'Predeanu', 'Mariana', 'mary12@gmail.com', '0745677899');
259 INSERT INTO client

```

ID_CLIENT	NUME_CLIENT	PRENUME_CLIENT	EMAIL_CLIENT	TELEFON_CLIENT
61	Pirinei	Ionut	ionut@gmail.com	0756789900
62	Predeanu	Mariana	mary12@gmail.com	0745677899
63	Ciocos	Bianca	bye_eu@gmail.com	0765444143
64	Teodoroiu	Cristina	cris_teo@gmail.com	0746767888
65	Mircea	Carina	cary_cary@gmail.com	0745883464

```

CREATE TABLE comenzi_online (
    id_comanda NUMBER(5) CONSTRAINT pk_comanda PRIMARY KEY,
    data_plasare DATE CONSTRAINT ck_data_comanda NOT NULL,
    data_sosire DATE CONSTRAINT ck_data_sosire NOT NULL,
    adresa_livrare VARCHAR(100) CONSTRAINT ck_adresa_comanda NOT
NULL,
    id_magazin NUMBER(5),
    CONSTRAINT fk_comanda_magazin FOREIGN KEY (id_magazin)
REFERENCES locatie (id_magazin),
    id_client NUMBER(5),
    CONSTRAINT fk_comanda_client FOREIGN KEY (id_client)
REFERENCES client (id_client)
);

INSERT INTO comenzi_online
VALUES (66, TO_DATE('16-01-2023', 'DD-MM-YYYY'),
TO_DATE('24-02-2023', 'DD-MM-YYYY'), 'Barbu Vacarescu 70', 31,
64);

```

```
INSERT INTO comenzi_online
VALUES (67, TO_DATE('31-01-2023', 'DD-MM-YYYY'),
TO_DATE('14-02-2023', 'DD-MM-YYYY'), 'Liviu Rebreanu 13', 33,
61);
INSERT INTO comenzi_online
VALUES (68, TO_DATE('14-02-2023', 'DD-MM-YYYY'),
TO_DATE('15-03-2023', 'DD-MM-YYYY'), 'Magheru 13', 32, 65);
INSERT INTO comenzi_online
VALUES (69, TO_DATE('22-02-2023', 'DD-MM-YYYY'),
TO_DATE('18-03-2023', 'DD-MM-YYYY'), 'Nicolae Grigorescu 17', 34,
64);
INSERT INTO comenzi_online
VALUES (70, TO_DATE('29-03-2023', 'DD-MM-YYYY'),
TO_DATE('14-04-2023', 'DD-MM-YYYY'), 'Vlad Tepes 13', 35, 63);
INSERT INTO comenzi_online
VALUES (71, TO_DATE('15-04-2023', 'DD-MM-YYYY'),
TO_DATE('16-05-2023', 'DD-MM-YYYY'), 'Crizantemelor 50', 33, 65);
INSERT INTO comenzi_online
VALUES (72, TO_DATE('20-03-2023', 'DD-MM-YYYY'),
TO_DATE('17-04-2023', 'DD-MM-YYYY'), 'Decebal 20', 31, 61);
INSERT INTO comenzi_online
VALUES (73, TO_DATE('14-02-2023', 'DD-MM-YYYY'),
TO_DATE('18-04-2023', 'DD-MM-YYYY'), 'Lascar Catargiu 12', 32,
65);
INSERT INTO comenzi_online
VALUES (74, TO_DATE('15-01-2023', 'DD-MM-YYYY'),
TO_DATE('13-02-2023', 'DD-MM-YYYY'), 'Regina Elisabeta 3', 34,
65);
INSERT INTO comenzi_online
VALUES (75, TO_DATE('22-03-2023', 'DD-MM-YYYY'),
TO_DATE('14-04-2023', 'DD-MM-YYYY'), 'Sfanta Vineri 34', 32, 62);

COMMIT;

SELECT * FROM comenzi_online;
```

```

274 CREATE TABLE comenzi_online (
275     id_comanda NUMBER(5) CONSTRAINT pk_comanda PRIMARY KEY,
276     data_plasare DATE CONSTRAINT ck_data_comanda NOT NULL,
277     data_sosire DATE CONSTRAINT ck_data_sosire NOT NULL,
278     adresa_livrare VARCHAR(100) CONSTRAINT ck_adresa_comanda NOT NULL,
279     id_magazin NUMBER(5),
280     CONSTRAINT fk_comanda_magazin FOREIGN KEY (id_magazin) REFERENCES locatie (id_magazin),
281     id_client NUMBER(5),
282     CONSTRAINT fk_comanda_client FOREIGN KEY (id_client) REFERENCES client (id_client)
283 );

```

```

284
285 INSERT INTO comenzi_online
286 VALUES (66, TO_DATE('16-01-2023', 'DD-MM-YYYY'), TO_DATE('14-02-2023', 'DD-MM-YYYY'), 'Barbu Vacarescu 70', 31, 64);

```

ID_COMANDA	DATA_PLASARE	DATA_SOSIRE	ADRESA_LIVRARE	ID_MAGAZIN	ID_CLIENT
66	16-JAN-23	24-FEB-23	Barbu Vacarescu 70	31	64
67	31-JAN-23	14-FEB-23	Liviu Rebreanu 13	33	61
68	14-FEB-23	15-MAR-23	Magheru 13	32	65
69	22-FEB-23	18-MAR-23	Nicolae Grigorescu 17	34	64
70	29-MAR-23	14-APR-23	Vlad Tepes 13	35	63

```

CREATE TABLE rating (
    id_rating NUMBER(5) CONSTRAINT pk_rating PRIMARY KEY,
    nota_acordata NUMBER(5) CONSTRAINT ck_nota_rating NOT NULL,
    id_angajat NUMBER(5),
    CONSTRAINT fk_rating_angajat FOREIGN KEY (id_angajat)
REFERENCES angajati (id_angajat),
    id_client NUMBER(5),
    CONSTRAINT fk_rating_client FOREIGN KEY (id_client) REFERENCES
client (id_client)
);

```

```

INSERT INTO rating
VALUES (76, 8, 26, 65);
INSERT INTO rating
VALUES (77, 10, 27, 61);
INSERT INTO rating
VALUES (78, 5, 37, 64);
INSERT INTO rating
VALUES (79, 7, 30, 61);
INSERT INTO rating
VALUES (80, 9, 38, 65);
INSERT INTO rating
VALUES (91, 3, 36, 63);
INSERT INTO rating
VALUES (92, 7, 39, 64);
INSERT INTO rating
VALUES (93, 2, 28, 61);
INSERT INTO rating
VALUES (94, 10, 26, 65);
INSERT INTO rating

```

```
VALUES (95, 9, 38, 63);
```

```
COMMIT;
```

```
SELECT * FROM rating;
```

```
313 CREATE TABLE rating (  
314     id_rating NUMBER(5) CONSTRAINT pk_rating PRIMARY KEY,  
315     nota_acordata NUMBER(5) CONSTRAINT ck_nota_rating NOT NULL,  
316     id_angajat NUMBER(5),  
317     CONSTRAINT fk_rating_angajat FOREIGN KEY (id_angajat) REFERENCES angajati (id_angajat),  
318     id_client NUMBER(5),  
319     CONSTRAINT fk_rating_client FOREIGN KEY (id_client) REFERENCES client (id_client)  
320 );  
321  
322 INSERT INTO rating  
323 VALUES (76, 8, 26, 65);
```

ID_RATING	NOTA_ACORDATA	ID_ANGAJAT	ID_CLIENT
76	8	26	65
77	10	27	61
78	5	37	64
79	7	30	61
80	9	38	65
91	3	36	63

```
CREATE TABLE review (  
    id_review NUMBER(5) CONSTRAINT pk_review PRIMARY KEY,  
    descriere VARCHAR(100) CONSTRAINT ck_descriere_review NOT  
NULL,  
    id_magazin NUMBER(5),  
    CONSTRAINT fk_review_magazin FOREIGN KEY (id_magazin)  
REFERENCES locatie (id_magazin),  
    id_client NUMBER(5),  
    CONSTRAINT fk_review_client FOREIGN KEY (id_client) REFERENCES  
client (id_client)  
);
```

```
INSERT INTO review  
VALUES (81, 'angajati foarte prietensosi', 33, 64);
```

```
INSERT INTO review  
VALUES (82, 'haine calitative', 31, 65);
```

```
INSERT INTO review  
VALUES (83, 'personal priceput', 32, 63);
```

```
INSERT INTO review  
VALUES (84, 'ador acest brand', 34, 61);
```

```
INSERT INTO review  
VALUES (85, 'mai revin aici', 35, 62);
```

```
INSERT INTO review
```

```

VALUES (96, 'nu mi-a placut', 31, 65);
INSERT INTO review
VALUES (97, 'haine calitative', 33, 63);
INSERT INTO review
VALUES (98, 'personal priceput', 34, 62);
INSERT INTO review
VALUES (99, 'ador acest brand', 31, 61);
INSERT INTO review
VALUES (100, 'mai revin aici', 34, 62);

COMMIT;

SELECT * FROM review;

```

```

349 CREATE TABLE review (
350     id_review NUMBER(5) CONSTRAINT pk_review PRIMARY KEY,
351     descriere VARCHAR(100) CONSTRAINT ck_descriere_review NOT NULL,
352     id_magazin NUMBER(5),
353     CONSTRAINT fk_review_magazin FOREIGN KEY (id_magazin) REFERENCES locatie (id_magazin),
354     id_client NUMBER(5),
355     CONSTRAINT fk_review_client FOREIGN KEY (id_client) REFERENCES client (id_client)
356 );
357
358 INSERT INTO review
359 VALUES (81, 'angajati foarte prietenosi', 33, 64);
360 INSERT INTO review

```

ID_REVIEW	DESCRIERE	ID_MAGAZIN	ID_CLIENT
81	angajati foarte prietenosi	33	64
82	haine calitative	31	65
83	personal priceput	32	63
84	ador acest brand	34	61
85	mai revin aici	35	62
96	nu mi-a placut	31	65

```

CREATE TABLE reducere (
    id_reducere NUMBER(5) CONSTRAINT pk_reducere PRIMARY KEY,
    procent NUMBER(5) CONSTRAINT ck_procent_reducere NOT NULL,
    perioada_valabilitate NUMBER(5) CONSTRAINT
ck_perioada_reducere NOT NULL,
    data_inceput DATE CONSTRAINT ck_data_inceput_reducere NOT
NULL,
    id_client NUMBER(5),
    CONSTRAINT fk_reducere_client FOREIGN KEY (id_client)
REFERENCES client (id_client)
);

INSERT INTO reducere
VALUES (86, 10, 3, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 61);
INSERT INTO reducere

```

```
VALUES (87, 25, 5, TO_DATE('23-07-2022', 'DD-MM-YYYY'), 63);
INSERT INTO reducere
VALUES (88, 40, 7, TO_DATE('24-11-2021', 'DD-MM-YYYY'), 65);
INSERT INTO reducere
VALUES (89, 35, 2, TO_DATE('01-02-2023', 'DD-MM-YYYY'), 62);
INSERT INTO reducere
VALUES (90, 25, 1, TO_DATE('15-01-2023', 'DD-MM-YYYY'), 64);

COMMIT;

SELECT * FROM reducere;
```

```
388 CREATE TABLE reducere (
389     id_reducere NUMBER(5) CONSTRAINT pk_reducere PRIMARY KEY,
390     procent NUMBER(5) CONSTRAINT ck_procent_reducere NOT NULL,
391     perioada_valabilitate NUMBER(5) CONSTRAINT ck_perioada_reducere NOT NULL,
392     data_inceput DATE CONSTRAINT ck_data_inceput_reducere NOT NULL,
393     id_client NUMBER(5),
394     CONSTRAINT fk_reducere_client FOREIGN KEY (id_client) REFERENCES client (id_client)
395 );
396
397 INSERT INTO reducere
398 VALUES (86, 10, 3, TO_DATE('01-01-2023', 'DD-MM-YYYY'), 61);
399 INSERT INTO reducere
400 VALUES (87, 25, 5, TO_DATE('23-07-2022', 'DD-MM-YYYY'), 63);
```

ID_REUCERE	PROCENT	PERIOADA_VALABILITATE	DATA_INCEPUT	ID_CLIENT
86	10	3	01-JAN-23	61
87	25	5	23-JUL-22	63
88	40	7	24-NOV-21	65
89	35	2	01-FEB-23	62
90	25	1	15-JAN-23	64

7. Ilustrarea noțiunilor de PL/SQL învățate

- a) Pentru marfă, o vom lua pe cea care a avut data livrării în luna martie. Apoi, pentru fiecare locație, calculăm cate cutii s-au primit la marfă și de asemenea, salvăm data livrării pentru fiecare cutie de la marfă.

```
CREATE OR REPLACE PROCEDURE proceduraex9a AS
    TYPE cutii_comanda IS TABLE OF contor_marfa.nr_cutii%TYPE; --
nested table
    TYPE id_magazin IS TABLE OF contor_marfa.id_magazin%TYPE INDEX
BY PLS_INTEGER; -- tabela indexata
    TYPE marfa_info IS VARRAY(100) OF VARCHAR2(100);
    v_cutii cutii_comanda;
    v_locatie_id id_magazin;
    v_marfa_info marfa_info;
```

```

    CURSOR c_marfa_martie IS
        SELECT id_marfa, data_livrare
        FROM marfa
        WHERE EXTRACT(MONTH FROM data_livrare) = 3;
BEGIN
    v_marfa_info := marfa_info();
    v_marfa_info.extend(100);

    FOR m IN c_marfa_martie LOOP
        SELECT contor_marfa.nr_cutii, contor_marfa.id_magazin,
        TO_CHAR(m.data_livrare, 'DD-MON-YYYY')
        BULK COLLECT INTO v_cutii, v_locatie_id, v_marfa_info
        FROM contor_marfa
        WHERE contor_marfa.id_marfa = m.id_marfa;

        FOR i IN 1..v_locatie_id.COUNT LOOP
            -- Calculam nr total de cutii pt fiecare locatie
            DBMS_OUTPUT.PUT_LINE('Magazinul ' || v_locatie_id(i) ||
            ' a primit ' || v_cutii(i) || ' cutii de marfa pe ' ||
            v_marfa_info(i) || '.');
        END LOOP;
    END LOOP;
END proceduraex9a;
/

BEGIN
    proceduraex9a;
END;
/

```

```

413 CREATE OR REPLACE PROCEDURE proceduraex9a AS
414     TYPE cutii_comanda IS TABLE OF contor_marfa.nr_cutii%TYPE; -- nested table
415     TYPE id_magazin IS TABLE OF contor_marfa.id_magazin%TYPE INDEX BY PLS_INTEGER; -- tabela indexata
416     TYPE marfa_info IS VARRAY(100) OF VARCHAR2(100);
417     v_cutii cutii_comanda;
418     v_locatie_id id_magazin;
419     v_marfa_info marfa_info;
420 CURSOR c_marfa_martie IS
421     SELECT id_marfa, data_livrare
422     FROM marfa
423     WHERE EXTRACT(MONTH FROM data_livrare) = 3;
424 BEGIN
425     v_marfa_info := marfa_info();
426     v_marfa_info.extend(100);
427
428     FOR m IN c_marfa_martie LOOP
429         SELECT contor_marfa.nr_cutii, contor_marfa.id_magazin, TO_CHAR(m.data_livrare, 'DD-MON-YYYY')
430         BULK COLLECT INTO v_cutii, v_locatie_id, v_marfa_info

```

Procedure created.

Statement processed.

Magazinul 34 a primit 156 cutii de marfa pe 15-MAR-2023.
 Magazinul 33 a primit 400 cutii de marfa pe 15-MAR-2023.
 Magazinul 35 a primit 230 cutii de marfa pe 01-MAR-2023.
 Magazinul 35 a primit 120 cutii de marfa pe 01-MAR-2023.

b) Pentru fiecare angajat, afisam media notelor de la rating primite.

```

CREATE OR REPLACE PROCEDURE proceduraex9b IS
    -- Cursor simplu pentru a itera prin fiecare angajat
    CURSOR c_angajati IS
        SELECT id_angajat,
               nume_angajat,
               prenume_angajat
        FROM angajati;
    -- Cursor parametrizat pentru a calcula media notelor primite de
    -- fiecare angajat
    CURSOR c_rating (p_id_angajat NUMBER) IS
        SELECT AVG(nota_acordata) AS media
        FROM rating
        WHERE id_angajat = p_id_angajat;
    -- Variabile pentru a stoca media notelor primite de fiecare
    -- angajat
    v_id_angajat    angajati.id_angajat%TYPE;
    v_nume           angajati.nume_angajat%TYPE;
    v_prenume        angajati.prenume_angajat%TYPE;
    v_media_rating   rating.nota_acordata%TYPE;
BEGIN
    -- Iteram prin fiecare angajat
    FOR r_angajati IN c_angajati LOOP
        v_id_angajat := r_angajati.id_angajat;
        v_nume       := r_angajati.nume_angajat;
        v_prenume    := r_angajati.prenume_angajat;
        -- Calculam media notelor primite de fiecare angajat
        OPEN c_rating(v_id_angajat);
        FETCH c_rating INTO v_media_rating;
    -- Verificăm dacă a fost returnată o medie
        IF v_media_rating IS NOT NULL THEN
            -- Afisam media notelor primite de fiecare angajat
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_nume || ' ' ||
v_prenume || ' are media notelor primite: ' || v_media_rating);
        ELSE
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_nume || ' ' ||
v_prenume || ' nu are note primite. ');
        END IF;
    CLOSE c_rating;
    END LOOP;
END proceduraex9b;
/
BEGIN
    proceduraex9b;
END;
/

```



```

449 CREATE OR REPLACE PROCEDURE proceduraex9b IS
450     -- Cursor simplu pentru a itera prin fiecare angajat
451     CURSOR c_angajati IS
452         SELECT id_angajat,
453                nume_angajat,
454                prenume_angajat
455         FROM angajati;
456
457     -- Cursor parametrizat pentru a calcula media notelor primite de fiecare angajat
458     CURSOR c_rating (p_id_angajat NUMBER) IS
459         SELECT AVG(nota_acordata) AS media
460         FROM rating
461         WHERE id_angajat = p_id_angajat;
462

```

Procedure created.

Statement processed.

Angajatul Florea Catalin are media notelor primite: 9
 Angajatul Adam Antonio are media notelor primite: 10
 Angajatul Vulcanescu Mircea are media notelor primite: 2
 Angajatul Ceausu Radu nu are note primite.
 Angajatul Craciun Alexandru are media notelor primite: 7
 Angajatul Aldea Alexia are media notelor primite: 3
 Angajatul Rosu Cristina are media notelor primite: 5
 Angajatul Luta Alexandra are media notelor primite: 9
 Angajatul Pentu Miruna are media notelor primite: 7
 Angajatul Bozeanu Mirela nu are note primite.

- c) Toate comenzile online care se incadreaza in perioada de valabilitate a reducerii (data_plasare > data_inceput_reducere).

```

CREATE OR REPLACE FUNCTION functiex9c (
    p_id_client IN client.id_client%TYPE
) RETURN comenzi_online.id_comanda%TYPE
IS
    v_data_sfarsit DATE;
    v_rezultat      NUMBER;
    v_count_orders  NUMBER;
    -- Definirea exceptiilor personalizate
    exceptie1 EXCEPTION;
    PRAGMA exception_init(exceptie1, -20001);
    exceptie2 EXCEPTION;
    PRAGMA exception_init(exceptie2, -20002);
    exceptie3 EXCEPTION;
    PRAGMA exception_init(exceptie3, -20003);
BEGIN
    -- Verificarea existenței clientului
    SELECT COUNT(*)
    INTO v_rezultat
    FROM client
    WHERE id_client = p_id_client;
    IF v_rezultat = 0 THEN
        RAISE exceptie3; -- Clientul nu există
    ELSIF v_rezultat > 1 THEN
        RAISE exceptie2; -- Există mai mulți clienți cu același ID
    END IF;

```

```

-- Calcularea datei de sfârșit a perioadei de reducere
SELECT ADD_MONTHS(data_inceput, perioada_valabilitate * 12)
INTO v_data_sfarsit
FROM reducere
WHERE id_client = p_id_client;
-- Verificare dacă există o reducere validă
IF v_data_sfarsit IS NULL THEN
    RAISE exceptie1; -- Nu există reducere
END IF;
-- Extrage comenzile online pentru client în perioada de
reducere
SELECT COUNT(co.id_comanda)
INTO v_count_orders
FROM comenzi_online co
JOIN reducere r ON co.id_client = r.id_client
WHERE co.id_client = p_id_client
    AND co.data_plasare BETWEEN r.data_inceput AND
v_data_sfarsit;
IF v_count_orders = 0 THEN
    RAISE exceptie1; -- Nu există comenzi online
END IF;
-- Returnează ID-ul comenzii (poate fi ajustat în funcție de
nevoile tale)
RETURN v_count_orders;
EXCEPTION
    WHEN exceptie1 THEN
        DBMS_OUTPUT.PUT_LINE('Nu exista nicio comanda plasata
pentru clientul dat');
        RETURN 0;
    WHEN exceptie2 THEN
        DBMS_OUTPUT.PUT_LINE('Exista mai multi clienti cu acelasi
id in baza de date');
        RETURN 0;
    WHEN exceptie3 THEN
        DBMS_OUTPUT.PUT_LINE('Clientul dat nu exista in baza de
date');
        RETURN 0;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
        RETURN 0;
END functieex9c;
/
-- Apelarea funcției cu un ID de client existent și care are
comenzi
SELECT functieex9c(61) FROM dual; -- Ar trebui să returneze
numărul comenzilor
-- Apelarea funcției cu un ID de client care nu există
SELECT functieex9c(1) FROM dual; -- Ar trebui să afișeze mesajul
că nu există
-- Apelarea funcției cu un ID de client care nu are comenzi
SELECT functieex9c(101) FROM dual; -- Ar trebui să afișeze
mesajul că nu există comenzi

```

```

499
500 CREATE OR REPLACE FUNCTION functieex9c (
501     p_id_client IN client.id_client%TYPE
502 ) RETURN comenzi_online.id_comanda%TYPE
503 IS
504     v_data_sfarsit DATE;
505     v_rezultat     NUMBER;
506     v_count_orders NUMBER;
507
508     -- Definirea excepțiilor personalizate
509     exceptie1 EXCEPTION;
510     PRAGMA exception_init(exceptie1, -20001);
511     exceptie2 EXCEPTION;
512     PRAGMA exception_init(exceptie2, -20002);
513     -- ...

```

Function created.

```

571 -- Apelarea funcției cu un ID de client existent și care are comenzi
572 SELECT functieex9c(61) FROM dual; -- Ar trebui să returneze numărul comenzilor

```

FUNCTIEEX9C(61)	
	2

```

574 -- Apelarea funcției cu un ID de client care nu există
575 SELECT functieex9c(1) FROM dual; -- Ar trebui să afișeze mesajul că nu există

```

FUNCTIEEX9C(1)	
	0

Download CSV

Clientul dat nu exista in baza de date

- d) Trigger care permite lucrul asupra tabelului CONTRACT doar în intervalul 8-18, de luni pana sambata.

```

CREATE OR REPLACE TRIGGER triggerex9d
BEFORE INSERT OR UPDATE OR DELETE ON contract

```

```

BEGIN
    IF (TO_CHAR(SYSDATE, 'D') = 7) OR
       (TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')) NOT BETWEEN '08' AND
'18') THEN
        RAISE_APPLICATION_ERROR(
            -20001,
            'Tabelul nu poate fi actualizat in afara orelor de
lucru!'
        );
    END IF;
END;
/

UPDATE contract
SET tip_contract = 'angajat';

```

```

581 ✓ CREATE OR REPLACE TRIGGER triggerex9d
582 BEFORE INSERT OR UPDATE OR DELETE ON contract
583 BEGIN
584     IF (TO_CHAR(SYSDATE, 'D') = 7) OR
585        (TO_NUMBER(TO_CHAR(SYSDATE, 'HH24')) NOT BETWEEN '08' AND '18') THEN
586        RAISE_APPLICATION_ERROR(
587            -20001,
588            'Tabelul nu poate fi actualizat in afara orelor de lucru!'
589        );
590    END IF;
591 END;
592 ✓ /
593
594 UPDATE contract
595 SET tip_contract = 'angajat';
596

```

Trigger created.

ORA-20001: Tabelul nu poate fi actualizat in afara orelor de lucru! ORA-06512: at "SQL_MNWWJAQRALXUTNVSICFJBWJUP.TRIGGEREX9D", line 4
ORA-06512: at "SYS.DBMS_SQL", line 1721

e) Trigger care sa nu permita ca reducerea unui client sa depaseasca 50%.

```

CREATE OR REPLACE TRIGGER triggerex9e
BEFORE UPDATE OF procent ON reducere
FOR EACH ROW
WHEN (NEW.procent > 50)
BEGIN
    RAISE_APPLICATION_ERROR(
        -20002,
        'Procentul de reducere nu poate depasi 50%!'
    );
END;
/
UPDATE reducere

```

```
SET procent = procent + 50;
```

```
600 ✓ CREATE OR REPLACE TRIGGER triggerex9e
601 BEFORE UPDATE OF procent ON reducere
602 FOR EACH ROW
603 WHEN (NEW.procent > 50)
604 BEGIN
605     RAISE_APPLICATION_ERROR(
606         -20002,
607         'Procentul de reducere nu poate depasi 50%! '
608     );
609 END;
610 ✓ /
611 UPDATE reducere
612 SET procent = procent + 50;
613
```

Trigger created.

ORA-20002: Procentul de reducere nu poate depasi 50%! OR
ORA-06512: at "SYS.DBMS_SQL", line 1721

f) Trigger dintr-o comanda LDD

```
CREATE TABLE erori (
    nume_bd VARCHAR2(50),
    user_logat VARCHAR2(30),
    data_occurenta TIMESTAMP(3),
    mesaj_eroare VARCHAR2(2000)
);

CREATE OR REPLACE TRIGGER triggerex9f
AFTER SERVERERROR ON SCHEMA
DECLARE
    v_error_msg VARCHAR2(2000);
BEGIN
    -- Captăm ultimul mesaj de eroare generat
    v_error_msg := DBMS_UTILITY.FORMAT_ERROR_STACK;
    INSERT INTO erori (nume_bd, user_logat, data_occurenta,
    mesaj_eroare)
    VALUES (
        SYS_CONTEXT('USERENV', 'DB_NAME'), -- Folosim SYS_CONTEXT
        pentru a obține numele bazei de date
```

```

        SYS_CONTEXT('USERENV', 'SESSION_USER'), -- Folosim
SYS_CONTEXT pentru a obține utilizatorul logat
        SYSTIMESTAMP, -- Data și ora
la care a apărut eroarea
        v_error_msg -- Mesajul de
eroare capturat
    );
END;
/

BEGIN
    EXECUTE IMMEDIATE 'ALTER TABLE a DROP COLUMN
nonexistent_column';
END;

SELECT * FROM erori;

```

```

616 CREATE TABLE erori (
617     nume_bd VARCHAR2(50),
618     user_logat VARCHAR2(30),
619     data_occurenta TIMESTAMP(3),
620     mesaj_eroare VARCHAR2(2000)
621 );
622
623 CREATE OR REPLACE TRIGGER triggerex9f
624 AFTER SERVERERROR ON SCHEMA
625 DECLARE
626     v_error_msg VARCHAR2(2000);
627 BEGIN
628     -- Captăm ultimul mesaj de eroare generat
629     v_error_msg := DBMS_UTILITY.FORMAT_ERROR_STACK;
630 INSERT INTO erori (nume_bd, user_logat, data_occurenta, mesaj_eroare)
631 VALUES (
632     SYS_CONTEXT('USERENV', 'DB_NAME'), -- Folosim SYS_CONTEXT pentru a obține numele bazei de date
633     SYS_CONTEXT('USERENV', 'SESSION_USER'), -- Folosim SYS_CONTEXT pentru a obține utilizatorul logat
634     SYSTIMESTAMP, -- Data și ora la care a apărut eroarea
635     v_error_msg -- Mesajul de eroare capturat

```

Table created.

Trigger created.

- g) Un pachet care să conțină toate procedurile, funcțiile și trigger-ii creați/creați mai sus.

```

CREATE OR REPLACE PACKAGE pachet_marfa AS
    PROCEDURE proceduraex9a;
    PROCEDURE proceduraex9b;
    FUNCTION functieex9c(p_id_client IN client.id_client%TYPE)
RETURN comenzi_online.id_comanda%TYPE;
END pachet_marfa;
/

CREATE OR REPLACE PACKAGE BODY pachet_marfa AS
    -- Procedura a)
    PROCEDURE proceduraex9a AS
        TYPE cutii_comanda IS TABLE OF contor_marfa.nr_cutii%TYPE;

```

```

-- nested table
    TYPE id_magazin IS TABLE OF contor_marfa.id_magazin%TYPE
INDEX BY PLS_INTEGER; -- tabela indexata
    TYPE marfa_info IS VARRAY(100) OF VARCHAR2(100);
    v_cutii cutii_comanda;
    v_locatie_id id_magazin;
    v_marfa_info marfa_info;
    CURSOR c_marfa_martie IS
        SELECT id_marfa, data_livrare
        FROM marfa
        WHERE EXTRACT(MONTH FROM data_livrare) = 3;

BEGIN
    v_marfa_info := marfa_info();
    v_marfa_info.extend(100);

    BEGIN
        -- Actualizare pentru a declanșa triggerul în afara orelor
de lucru
        UPDATE contract
        SET tip_contract = 'nou_tip'
        WHERE id_contract = 1; -- înlocuiește cu un ID valid
existent în tabelul `contract`
    EXCEPTION
        WHEN OTHERS THEN
            DBMS_OUTPUT.PUT_LINE('Eroare: ' || SQLERRM); --
Afișează mesajul de eroare din trigger
    END;

    FOR m IN c_marfa_martie LOOP
        SELECT contor_marfa.nr_cutii, contor_marfa.id_magazin,
TO_CHAR(m.data_livrare, 'DD-MON-YYYY')
        BULK COLLECT INTO v_cutii, v_locatie_id, v_marfa_info
        FROM contor_marfa
        WHERE contor_marfa.id_marfa = m.id_marfa;

        FOR i IN 1..v_locatie_id.COUNT LOOP
            -- Calculam nr total de cutii pt fiecare locatie
            DBMS_OUTPUT.PUT_LINE('Magazinul ' || v_locatie_id(i)
|| ' a primit ' || v_cutii(i) || ' cutii de marfa pe ' ||
v_marfa_info(i) || '.');
        END LOOP;
    END LOOP;
END proceduraex9a;
-- Procedura b)
PROCEDURE proceduraex9b IS
    -- Cursor simplu pentru a itera prin fiecare angajat
    CURSOR c_angajati IS
        SELECT id_angajat,
            nume_angajat,
            prenume_angajat
        FROM angajati;

    -- Cursor parametrizat pentru a calcula media notelor
primite de fiecare angajat
    CURSOR c_rating (p_id_angajat NUMBER) IS

```

```

        SELECT AVG(nota_acordata) AS media
        FROM rating
        WHERE id_angajat = p_id_angajat;
    -- Variabile pentru a stoca media notelor primite de
    fiecare angajat
    v_id_angajat    angajati.id_angajat%TYPE;
    v_numa          angajati.numa_angajat%TYPE;
    v_prenume       angajati.prenume_angajat%TYPE;
    v_media_rating  rating.nota_acordata%TYPE;
BEGIN
    -- Iteram prin fiecare angajat
    FOR r_angajati IN c_angajati LOOP
        v_id_angajat := r_angajati.id_angajat;
        v_numa := r_angajati.numa_angajat;
        v_prenume := r_angajati.prenume_angajat;

        -- Calculam media notelor primite de fiecare angajat
        OPEN c_rating(v_id_angajat);
        FETCH c_rating INTO v_media_rating;

        -- Verificăm dacă a fost returnată o medie
        IF v_media_rating IS NOT NULL THEN
            -- Afisam media notelor primite de fiecare angajat
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_numa || ' '
|| v_prenume || ' are media notelor primite: ' ||
v_media_rating);
        ELSE
            DBMS_OUTPUT.PUT_LINE('Angajatul ' || v_numa || ' '
|| v_prenume || ' nu are note primite. ');
        END IF;
    CLOSE c_rating;
    END LOOP;
END proceduraex9b;

--Funcția c)
FUNCTION functieex9c (
    p_id_client IN client.id_client%TYPE
) RETURN comenzi_online.id_comanda%TYPE
IS
    v_data_sfarsit DATE;
    v_rezultat      NUMBER;
    v_count_orders  NUMBER;

    -- Definirea excepțiilor personalizate
    exceptie1 EXCEPTION;
    PRAGMA exception_init(exceptie1, -20001);
    exceptie2 EXCEPTION;
    PRAGMA exception_init(exceptie2, -20002);
    exceptie3 EXCEPTION;
    PRAGMA exception_init(exceptie3, -20003);

BEGIN
    UPDATE reducere SET procent = 60 WHERE id_client = 62;
    -- Verificarea existenței clientului
    SELECT COUNT(*)

```



```

        INTO v_rezultat
        FROM client
        WHERE id_client = p_id_client;

        IF v_rezultat = 0 THEN
            RAISE exceptie3; -- Clientul nu există
        ELSIF v_rezultat > 1 THEN
            RAISE exceptie2; -- Există mai mulți clienți cu același
ID
        END IF;

        -- Calcularea datei de sfârșit a perioadei de reducere
        SELECT ADD_MONTHS(data_inceput, perioada_valabilitate *
12)

        INTO v_data_sfarsit
        FROM reducere
        WHERE id_client = p_id_client;

        -- Verificare dacă există o reducere validă
        IF v_data_sfarsit IS NULL THEN
            RAISE exceptie1; -- Nu există reducere
        END IF;

        -- Extrage comenzile online pentru client în perioada de
        reducere
        SELECT COUNT(co.id_comanda)
        INTO v_count_orders
        FROM comenzi_online co
        JOIN reducere r ON co.id_client = r.id_client
        WHERE co.id_client = p_id_client
        AND co.data_plasare BETWEEN r.data_inceput AND
        v_data_sfarsit;

        IF v_count_orders = 0 THEN
            RAISE exceptie1; -- Nu există comenzi online
        END IF;

        -- Returnează ID-ul comenzii (poate fi ajustat în funcție
        de nevoile tale)
        RETURN v_count_orders;

    EXCEPTION
        WHEN exceptie1 THEN
            DBMS_OUTPUT.PUT_LINE('Nu exista nicio comanda plasata
        pentru clientul dat');
            RETURN 0;
        WHEN exceptie2 THEN
            DBMS_OUTPUT.PUT_LINE('Exista mai multi clienti cu
        acelasi id in baza de date');
            RETURN 0;
        WHEN exceptie3 THEN
            DBMS_OUTPUT.PUT_LINE('Clientul dat nu exista in baza de
        date');
            RETURN 0;
        WHEN OTHERS THEN

```

```

        DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
        RETURN 0;
    END functieex9c;
END;
/

BEGIN
    pachet_marfa.proceduraex9a;
    pachet_marfa.proceduraex9b;
    DBMS_OUTPUT.PUT_LINE('Numărul comenzilor online: ' ||
pachet_marfa.functieex9c(63));
END;
/

```

```

650 CREATE OR REPLACE PACKAGE pachet_marfa AS
651     PROCEDURE proceduraex9a;
652     PROCEDURE proceduraex9b;
653     FUNCTION functieex9c(p_id_client IN client.id_client%TYPE) RETURN comenzi_online.id_comanda%TYPE;
654 END pachet_marfa;

```

Package created.

Package Body created.

Statement processed.

Eroare: ORA-20001: Tabelul nu poate fi actualizat in afara orelor de lucru!

ORA-06512: at "SQL_MNWVJAQRALXUTNVSICFJBWJUP.TRIGGEREX9D", line 4

ORA-04088: error during execution of trigger 'SQL_MNWVJAQRALXUTNVSICFJBWJUP.TRIGGEREX9D'

Magazinul 34 a primit 156 cutii de marfa pe 15-MAR-2023.

Magazinul 33 a primit 400 cutii de marfa pe 15-MAR-2023.

Magazinul 35 a primit 230 cutii de marfa pe 01-MAR-2023.

Magazinul 35 a primit 120 cutii de marfa pe 01-MAR-2023.

Angajatul Florea Catalin are media notelor primite: 9

Angajatul Adam Antonio are media notelor primite: 10

Angajatul Vulcanescu Mircea are media notelor primite: 2

Angajatul Ceausu Radu nu are note primite.

Angajatul Craciun Alexandru are media notelor primite: 7

Angajatul Aldea Alexia are media notelor primite: 3

Angajatul Rosu Cristina are media notelor primite: 5

Angajatul Luta Alexandra are media notelor primite: 9

Angajatul Pentu Miruna are media notelor primite: 7

Angajatul Bozeanu Mirela nu are note primite.

Numărul comenzilor online: 1