

TechFlow

-PROIECT SGBD-

CUPRINS

Cerința 1.	3
Cerința 2.	3
Cerința 3.	4
Cerința 4.	5
Cerința 5.	9
Cerința 6.	19
Cerința 7.	22
Cerința 8.	26
Cerința 9.	31
Cerința 10.	35
Cerința 11.	38
Cerința 12.	44
Cerința 13.	46
Cerința 14.	58

Cerință 1.

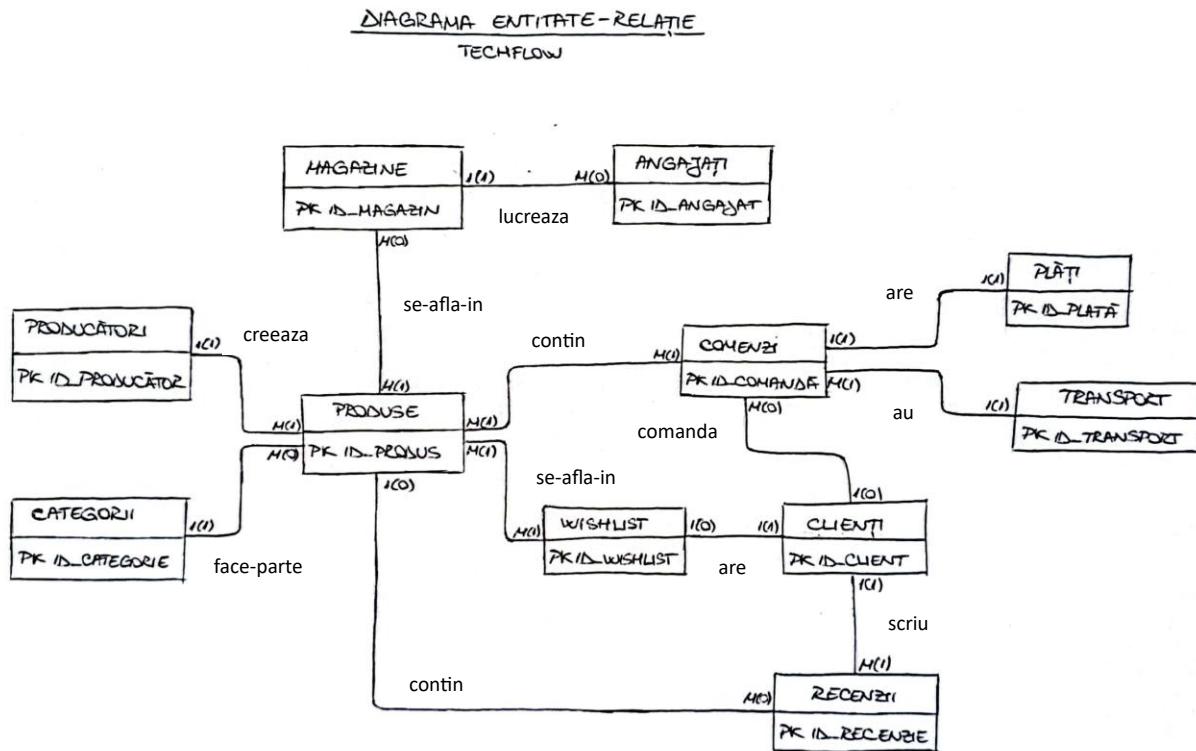
Prezentați pe scurt baza de date (utilitatea ei).

TechFlow este o afacere specializată în vânzarea de produse IT, cum ar fi componente hardware, accesorii și echipamente IT, prin intermediul magazinelor fizice și a site-ului lor de comerț electronic.

Prin intermediul bazei de date TechFlow, se realizează gestiunea detaliată a produselor, categoriilor de produse, comenziilor, plășilor și recenziilor. Aceasta facilitează procesul de vânzare, stocare, gestionare a comenziilor și interacțiune cu clienții. De asemenea, baza de date permite generarea de rapoarte și analize pentru a obține o înțelegere mai profundă a activitășilor de vânzare și a preferințelor clienților, contribuind astfel la **luarea deciziilor strategice și la îmbunătățirea experienței clienților**.

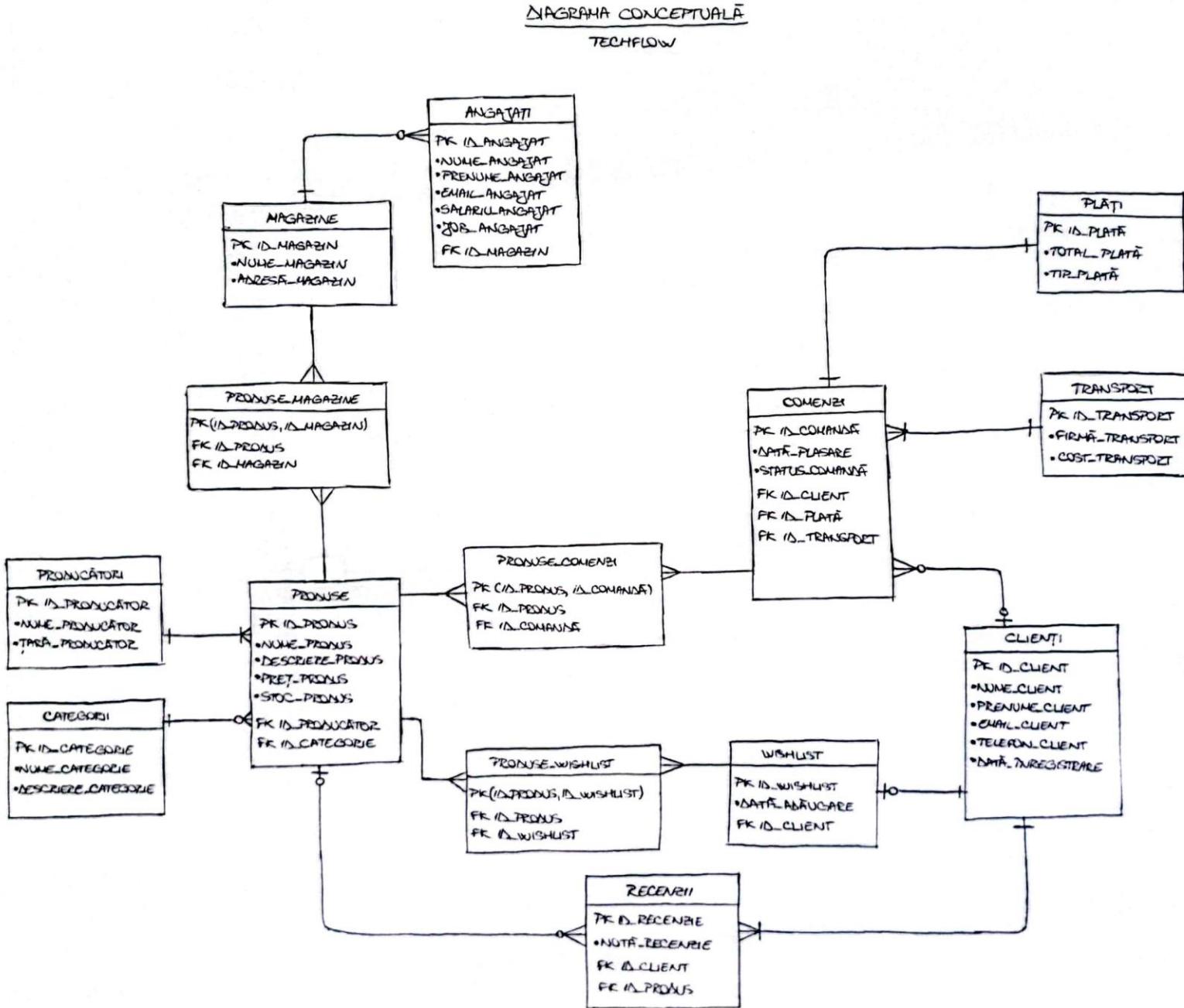
Cerință 2.

Realizați diagrama entitate-relație (ERD).



Cerință 3.

Realizați diagrama conceptuală a modelului propus.



Cerința 4.

Definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).

Table MAGAZINE created.	Table PRODUSE_MAGAZINE created.	Table PRODUSE_COMENZI created.
Table ANGAJATI created.	Table PLATI created.	Table WISHLIST created.
Table PRODUCATORI created.	Table TRANSPORT created.	Table PRODUSE_WISHLIST created.
Table CATEGORII created.	Table CLIENTI created.	Table RECENZII created.
Table PRODUSE created.	Table COMENZI created.	

```
-- Cerinta 4
-- Crearea secventei pentru cheile primare.
create sequence secventa_techflow
start with 1
increment by 1
maxvalue 10000
minvalue 1
nocycle
nocache;

-- 1 tabel MAGAZINE
create table magazine (
    id_magazin number(4),
    nume_magazin varchar2(30) constraint nn_nume_magazin not null,
    adresa_magazin varchar2(40) constraint nn_adresa_magazin not null,
    constraint pk_id_magazin primary key(id_magazin)
);
-- 2 tabel ANGAJATI
create table angajati (
    id_angajat number(4),
    nume_angajat varchar2(50) constraint nn_nume_angajat not null,
    prenume_angajat varchar2(50) constraint nn_prenume_angajat not null,
    email_angajat varchar2(50),
    salariu_angajat number(10,2) constraint nn_salariu_angajat not null,
    job_angajat varchar2(50),
    id_magazin number(4),
```

```
constraint pk_id_angajat primary key(id_angajat),
constraint fk_id_magazin foreign key(id_magazin)
    references magazine(id_magazin)
);

-- 3 tabel PRODUCATORI
create table producatori (
    id_producator number(4),
    nume_producator varchar2(50) constraint nn_nume_producator not null,
    tara_producator varchar(50),
    constraint pk_id_producator primary key(id_producator)
);

-- 4 tabel CATEGORII
create table categorii (
    id_categorie number(4),
    nume_categorie varchar2(50) constraint nn_nume_categorie not null,
    descriere_categorie varchar2(200),

    constraint pk_id_categorie primary key(id_categorie)
);

-- 5 tabel PRODUSE
create table produse (
    id_produs number(4),
    nume_produs varchar2(50) constraint nn_nume_produs not null,
    descriere_produs varchar2(200),
    pret_produs number(10,2),
    stoc_produs number(3),
    id_producator number(4),
    id_categorie number(4),
    constraint pk_id_produs primary key(id_produs),
    constraint fk_id_producator foreign key(id_producator)
        references producatori(id_producator),
    constraint fk_id_categorie foreign key(id_categorie)
        references categorii(id_categorie)
);

-- 6 tabel PRODUSE_MAGAZINE
create table produse_magazine (
    id_produs number(4),
    id_magazin number(4),
```

```
constraint pk_id_produs_id_magazin primary key(id_produs,id_magazin),
constraint fk_id_produs_pm foreign key(id_produs)
references produse(id_produs),
constraint fk_id_magazin_pm foreign key(id_magazin)
references magazine(id_magazin)
);

-- 7 tabel PLATI
create table plati (
    id_plata number(4),
    total_plata number(10,2) constraint nn_total_plata not null,
    tip_plata varchar2(50) constraint nn_tip_plata not null,
    constraint pk_id_plata primary key(id_plata)
);

-- 8 tabel TRANSPORT
create table transport (
    id_transport number(4),
    firma_transport varchar2(50) constraint nn_firma_transport not null,
    cost_transport number(10,2) constraint nn_cost_transport not null,
    constraint pk_id_transport primary key (id_transport)
);

-- 9 tabel CLIENTI
create table clienti (
    id_client number(4),
    nume_client varchar2(50),
    prenume_client varchar2(50),
    email_client varchar2(35),
    telefon_client varchar2(20),
    data_inregistrare date,
    constraint pk_id_client primary key (id_client)
);

-- 10 tabel COMENZI
create table comenzi (
    id_comanda number(4),
    data_plasare date,
    status_comanda varchar2(50),
    id_client number(4),
    id_plata number(4),
    id_transport number(4),
    constraint pk_id_comanda primary key (id_comanda),
```

```
constraint fk_id_client foreign key (id_client)
references clienti(id_client),
constraint fk_id_plata foreign key (id_plata)
references plati(id_plata),
constraint fk_id_transport foreign key (id_transport)
references transport(id_transport)
);

-- 11 tabel PRODUSE_COMENZI
create table produse_comenzi (
    id_produs number(4),
    id_comanda number(4),
    constraint pk_id_produs_id_comanda primary key(id_produs,id_comanda),
    constraint fk_id_produs_pc foreign key(id_produs)
        references produse(id_produs),
    constraint fk_id_comanda_pc foreign key(id_comanda)
        references comenzi(id_comanda)
);

-- 12 tabel WISHLIST
create table wishlist (
    id_wishlist number(4),
    data_adaugare date,
    id_client number(4),
    constraint pk_id_wishlist primary key (id_wishlist),
    constraint fk_id_client_wl foreign key (id_client)
        references clienti(id_client)
);

-- 13 tabel PRODUSE_WISHLIST
create table produse_wishlist (
    id_produs number(4),
    id_wishlist number(4),
    constraint pk_id_produs_id_wishlist primary key(id_produs,id_wishlist),
    constraint fk_id_produs_pw foreign key(id_produs)
        references produse(id_produs),
    constraint fk_id_wishlist_pw foreign key(id_wishlist)
        references wishlist(id_wishlist)
);

-- 14 tabel RECENZII
create table recenzii (
```

```

id_recenzie number(4),
nota_recenzie number(2,1),
id_client number(4),
id_produs number(4),

constraint pk_id_recenzie primary key (id_recenzie),
constraint fk_id_client_rz foreign key (id_client)
references clienti(id_client),
constraint fk_id_produs_rz foreign key (id_produs)
references produse(id_produs)
);

```

Cerință 5.

Adăugați informații coerente în tabelele create.

ID_ANGAJAT	NUME_ANGAJAT	PRENUME_ANGAJAT	EMAIL_ANGAJAT	SALARIU_ANGAJAT	JOB_ANGAJAT	ID_MAGAZIN
1	6 Popescu	Ion	ion.popescu@example.com	2500	Casier	1
2	7 Ionescu	Maria	maria.ionescu@example.com	3000	Manager	1
3	8 Popa	Andrei	andrei.popa@example.com	2200	Casier	2
4	9 Dragomir	Elena	elena.dragomir@example.com	2800	Manager	2
5	10 Georgescu	Alexandru	alexandru.georgescu@example.com	2400	Casier	3
6	11 Gheorghe	Mihai	mihai.gheorghe@example.com	3100	Manager	3
7	12 Pop	Adrian	adrian.pop@example.com	2300	Casier	4
8	13 Vasile	Ioana	ioana.vasile@example.com	2900	Manager	4
9	14 Stan	Daniel	daniel.stan@example.com	2600	Casier	5
10	15 Nistor	Cristina	cristina.nistor@example.com	3300	Manager	5

ID_CATEGORIE	NUME_CATEGORIE	DESCRIERE_CATEGORIE
1	21 Placi de baza	Categorie dedicata placilor de baza
2	22 Procesoare	Categorie dedicata procesoarelor
3	23 Placi video	Categorie dedicata placilor video
4	24 Memorii RAM	Categorie dedicata modulelor de memorie RAM
5	25 Stocare ROM	Categorie dedicata solutiilor de stocare

	ID_CLIENT	NUME_CLIENT	PRENUME_CLIENT	EMAIL_CLIENT	TELEFON_CLIENT	DATA_INREGISTRARE
1	46	Avram	Gabriela	gabriela.avram@example.com	0776123456	12-MAY-23
2	47	Constantin	Mihai	mihai.constantin@example.com	0787123456	15-MAR-23
3	48	Dumitru	Adriana	adriana.dumitru@example.com	0798123456	14-JAN-23
4	49	Florescu	Valentin	valentin.florescu@example.com	0709123456	13-FEB-23
5	50	Gheorghe	Cristina	cristina.gheorghe@example.com	0710123456	12-MAR-23

	ID_COMANDA	DATA_PLASARE	STATUS_COMANDA	ID_CLIENT	ID_PLATA	ID_TRANSPORT
1	51	15-MAY-23	Noua	46	36	41
2	52	10-MAY-23	Confirmata	47	37	41
3	53	15-APR-23	Anulata	46	38	41
4	54	15-MAY-23	Noua	48	39	42
5	55	09-MAY-23	Confirmata	49	40	44

	ID_MAGAZIN	NUME_MAGAZIN	ADRESA_MAGAZIN
1	1	TechFlow Red	Bacau
2	2	TechFlow Blue	Bucuresti
3	3	TechFlow Black	Cluj
4	4	TechFlow Green	Iasi
5	5	TechFlow Yellow	Brasov

	ID_COMANDA	DATA_PLASARE	STATUS_COMANDA	ID_CLIENT	ID_PLATA	ID_TRANSPORT
1	51	15-MAY-23	Noua	46	36	41
2	52	10-MAY-23	Confirmata	47	37	41
3	53	15-APR-23	Anulata	46	38	41
4	54	15-MAY-23	Noua	48	39	42
5	55	09-MAY-23	Confirmata	49	40	44

	ID_PLATA	TOTAL_PLATA	TIP_PLATA
1	36	2598.99	Card
2	37	1699.97	Numerar
3	38	4199.98	Transfer bancar
4	39	5899.98	Card
5	40	2499.98	Numerar

	ID_PRODUS	NUME_PRODUS	DESCRIERE_PRODUS	PRET_PRODUS	STOC_PRODUS	ID_PRODUCATOR	ID_CATEGORIE
1	26	ASUS ROG Strix X570-E Gaming	Placă de bază ATX pentru gaming, compatibilă cu procesoare AMD Ryzen	1599.99	10	16	21
2	27	Intel Core i9-10900K	Procesor Intel de ultimă generație cu 10 nucle și 20 fire de execuție	2899.99	5	16	22
3	28	NVIDIA GeForce RTX 3080	Placă video high-end pentru gaming, cu 10 GB memorie GDDR6X	5299.99	3	17	23
4	29	Corsair Vengeance RGB Pro	Modul de memorie DDR4 cu iluminare RGB și capacitate de 16 GB	699.99	20	17	24
5	30	Samsung 970 EVO Plus	Solid State Drive NVMe cu capacitate de 1 TB și viteză de transfer ridicată	999.99	8	17	25
6	31	MSI B450 TOMAHAWK MAX	Placă de bază ATX pentru procesoare AMD Ryzen, suportă memorie DDR4	599.99	15	18	21
7	32	AMD Ryzen 7 5800X	Procesor AMD cu 8 nucle și 16 fire de execuție, frecvență de bază 3.8 GHz	1999.99	7	18	22
8	33	Gigabyte GeForce GTX 1660 SUPER OC	Placă video performantă pentru gaming, 6 GB memorie GDDR6	1799.99	10	19	23
9	34	RAM Kingston HyperX Fury RGB	Modul de memorie DDR4 cu iluminare RGB și capacitate de 32 GB	1299.99	12	19	24
10	35	SSD Western Digital Blue	Solid State Drive SATA cu capacitate de 500 GB și viteză de transfer mare	399.99	25	20	25

	ID_PRODUS	ID_COMANDA	ID_PRODUS	ID_MAGAZIN	ID_PRODUS	ID_WISHLIST
1	26	51	1	26	1	1
2	30	51	2	27	1	2
3	29	52	3	26	2	3
4	35	52	4	29	2	4
5	31	52	5	30	3	5
6	27	53	6	27	3	6
7	34	53	7	32	4	7
8	28	54	8	33	4	8
9	31	54	9	34	5	9
10	29	55	10	33	5	10
11	33	55			11	26

	ID_RECENZIE	NOTA_RECENZIE	ID_CLIENT	ID_PRODUS	ID_TRANS...	FIRMA_TRANSPORT	COST_TRANSPORT
1	66	4.5	46	26	1	41 Cargus	100
2	67	3.8	47	27	2	42 Fan Courier	150.5
3	68	4.2	48	28	3	43 Same Day	200.75
4	69	5	49	29	4	44 DPD	180.25
5	70	3.5	50	30	5	45 DHL	220

	ID_WISHLIST	DATA_ADAUGARE	ID_CLIENT
1	56	14-MAY-23	46
2	57	13-MAY-23	47
3	58	12-MAY-23	48
4	59	11-MAY-23	49
5	60	10-MAY-23	50

```
-- 1 tabel MAGAZINE
insert into magazine
values (secventa_techflow.nextval, 'TechFlow Red', 'Bacau');
insert into magazine
values (secventa_techflow.nextval, 'TechFlow Blue', 'Bucuresti');
insert into magazine
values (secventa_techflow.nextval, 'TechFlow Black', 'Cluj');
insert into magazine
values (secventa_techflow.nextval, 'TechFlow Green', 'Iasi');
insert into magazine
values (secventa_techflow.nextval, 'TechFlow Yellow', 'Brasov');
select * from magazine;

-- 2 tabel ANGAJATI
insert into angajati
values (secventa_techflow.nextval, 'Popescu', 'Ion', 'ion.popescu@example.com', 2500.00,
'Casier', 1);
insert into angajati
values (secventa_techflow.nextval, 'Ionescu', 'Maria', 'maria.ionescu@example.com', 3000.00,
'Manager', 1);
insert into angajati
values (secventa_techflow.nextval, 'Popa', 'Andrei', 'andrei.popa@example.com', 2200.00,
'Casier', 2);
insert into angajati
values (secventa_techflow.nextval, 'Dragomir', 'Elena', 'elena.dragomir@example.com', 2800.00,
'Manager', 2);
insert into angajati
values (secventa_techflow.nextval, 'Georgescu', 'Alexandru',
'alexandru.georgescu@example.com', 2400.00, 'Casier', 3);
insert into angajati
values (secventa_techflow.nextval, 'Gheorghe', 'Mihai', 'mihai.gheorghe@example.com',
3100.00, 'Manager', 3);
insert into angajati
values (secventa_techflow.nextval, 'Pop', 'Adrian', 'adrian.pop@example.com', 2300.00, 'Casier',
4);
insert into angajati
values (secventa_techflow.nextval, 'Vasile', 'Ioana', 'ioana.vasile@example.com', 2900.00,
'Manager', 4);
insert into angajati
values (secventa_techflow.nextval, 'Stan', 'Daniel', 'daniel.stan@example.com', 2600.00, 'Casier',
5);
```

```
insert into angajati
values (secventa_techflow.nextval, 'Nistor', 'Cristina', 'cristina.nistor@example.com', 3300.00,
'Manager', 5);
select * from angajati;

-- 3 tabel PRODUCATORI (inserare explicită)
insert into producatori (id_producator, nume_producator, tara_producator)
values (secventa_techflow.nextval, 'Stentor', 'Romania');
insert into producatori (id_producator, nume_producator, tara_producator)
values (secventa_techflow.nextval, 'PC Master', 'Germania');
insert into producatori (id_producator, nume_producator, tara_producator)
values (secventa_techflow.nextval, 'HardwareCO', 'Franta');
insert into producatori (id_producator, nume_producator, tara_producator)
values (secventa_techflow.nextval, 'VideoMaster', 'Italia');
insert into producatori (id_producator, nume_producator, tara_producator)
values (secventa_techflow.nextval, 'SiSoft', 'Spania');
select * from producatori;

-- 4 tabel CATEGORII
insert into categorii
values (secventa_techflow.nextval, 'Placi de baza', 'Categorie dedicată placilor de baza');
insert into categorii
values (secventa_techflow.nextval, 'Procesoare', 'Categorie dedicată procesoarelor');
insert into categorii
values (secventa_techflow.nextval, 'Placi video', 'Categorie dedicată placilor video');
insert into categorii
values (secventa_techflow.nextval, 'Memorii RAM', 'Categorie dedicată modulelor de memorie RAM');
insert into categorii
values (secventa_techflow.nextval, 'Stocare ROM', 'Categorie dedicată soluțiilor de stocare');
select * from categorii;

-- 5 tabel PRODUSE
insert into produse
values (secventa_techflow.nextval, 'ASUS ROG Strix X570-E Gaming', 'Placă de bază ATX pentru gaming, compatibilă cu procesoare AMD Ryzen', 1599.99, 10, 16, 21);
insert into produse
values (secventa_techflow.nextval, 'Intel Core i9-10900K', 'Procesor Intel de ultimă generație cu 10 nucle și 20 fire de execuție', 2899.99, 5, 16, 22);
insert into produse
values (secventa_techflow.nextval, 'NVIDIA GeForce RTX 3080', 'Placă video high-end pentru gaming, cu 10 GB memorie GDDR6X', 5299.99, 3, 17, 23);
```

```
insert into produse
values (secventa_techflow.nextval, 'Corsair Vengeance RGB Pro', 'Modul de memorie DDR4 cu
iluminare RGB și capacitate de 16 GB', 699.99, 20, 17, 24);
insert into produse
values (secventa_techflow.nextval, 'Samsung 970 EVO Plus', 'Solid State Drive NVMe cu
capacitate de 1 TB și viteză de transfer ridicată', 999.99, 8, 17, 25);
insert into produse
VALUES (secventa_techflow.nextval, 'MSI B450 TOMAHAWK MAX', 'Placă de bază ATX
pentru procesoare AMD Ryzen, suportă memorie DDR4', 599.99, 15, 18, 21);
insert into produse
VALUES (secventa_techflow.nextval, 'AMD Ryzen 7 5800X', 'Procesor AMD cu 8 nuclee și 16
fire de execuție, frecvență de bază 3.8 GHz', 1999.99, 7, 18, 22);
insert into produse
VALUES (secventa_techflow.nextval, 'Gigabyte GeForce GTX 1660 SUPER OC', 'Placă video
performantă pentru gaming, 6 GB memorie GDDR6', 1799.99, 10, 19, 23);
insert into produse
VALUES (secventa_techflow.nextval, 'RAM Kingston HyperX Fury RGB', 'Modul de memorie
DDR4 cu iluminare RGB și capacitate de 32 GB', 1299.99, 12, 19, 24);
insert into produse
VALUES (secventa_techflow.nextval, 'SSD Western Digital Blue', 'Solid State Drive SATA cu
capacitate de 500 GB și viteză de transfer mare', 399.99, 25, 20, 25);
select * from produse;

-- 6 tabel PRODUSE_MAGAZINE
-- Magazinul - TechFlow Red
-- Produs - ASUS ROG Strix X570-E Gaming
insert into produse_magazine
values (26, 1);
-- Produs - Intel Core i9-10900K
insert into produse_magazine
values (27, 1);
-- Magazinul - TechFlow Blue
-- Produs - ASUS ROG Strix X570-E Gaming
insert into produse_magazine
values (26,2);
-- Produs - Corsair Vengeance RGB Pro
insert into produse_magazine
values (29,2);
-- Magazinul - TechFlow Black
-- Produs - Samsung 970 EVO Plus
insert into produse_magazine
values (30,3);
```

```
-- Produs - Intel Core i9-10900K
insert into produse_magazine
values (27,3);
-- Magazinul - TechFlow Green
-- Produs - AMD Ryzen 7 5800X
insert into produse_magazine
values (32,4);
-- Produs - Gigabyte GeForce GTX 1660 SUPER OC
insert into produse_magazine
values (33,4);
-- Magazinul - TechFlow Yellow
-- Produs - RAM Kingston HyperX Fury RGB
insert into produse_magazine
values (34,5);
-- Produs - Gigabyte GeForce GTX 1660 SUPER OC
insert into produse_magazine
values (33,5);
select * from produse_magazine;

-- 7 tabel PLATI
-- total_plata id_produs : 26 + 30
insert into plati (id_plata, total_plata, tip_plata)
values (secventa_techflow.nextval, 2598.99, 'Card');
-- total_plata id_produs : 29 + 35 + 31
insert into plati (id_plata, total_plata, tip_plata)
values (secventa_techflow.nextval, 1699.97, 'Numerar');
-- total_plata id_produs : 27 + 34
insert into plati (id_plata, total_plata, tip_plata)
values (secventa_techflow.nextval, 4199.98, 'Transfer bancar');
-- total_plata id_produs : 28 + 31
insert into plati (id_plata, total_plata, tip_plata)
values (secventa_techflow.nextval, 5899.98, 'Card');
-- total_plata id_produs : 29 + 33
insert into plati (id_plata, total_plata, tip_plata)
values (secventa_techflow.nextval, 2499.98, 'Numerar');
select * from plati;

-- 8 tabel TRANSPORT
insert into transport
values (secventa_techflow.nextval, 'Cargus', 100.00);
insert into transport
values (secventa_techflow.nextval, 'Fan Courier', 150.50);
```

```
insert into transport
values (secventa_techflow.nextval, 'Same Day', 200.75);
insert into transport
values (secventa_techflow.nextval, 'DPD', 180.25);
insert into transport
values (secventa_techflow.nextval, 'DHL', 220.00);
select * from transport;

-- 9 tabel CLIENTI
insert into clienti
VALUES (secventa_techflow.nextval, 'Avram', 'Gabriela', 'gabriela.avram@example.com',
'0776123456', TO_DATE('2023-05-12', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Constantin', 'Mihai', 'mihai.constantin@example.com',
'0787123456', TO_DATE('2023-03-15', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Dumitru', 'Adriana', 'adriana.dumitru@example.com',
'0798123456', TO_DATE('2023-01-14', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Florescu', 'Valentin', 'valentin.florescu@example.com',
'0709123456', TO_DATE('2023-02-13', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Gheorghe', 'Cristina', 'cristina.gheorghe@example.com',
'0710123456', TO_DATE('2023-03-12', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Popescu', 'Mihai', 'popescu.mihai@example.com',
'0710121245', TO_DATE('2024-01-02', 'YYYY-MM-DD'));
insert into clienti
VALUES (secventa_techflow.nextval, 'Pop', 'Alexandra', 'pop.alexandra@example.com',
'0710131245', TO_DATE('2024-01-04', 'YYYY-MM-DD'));
select * from clienti;

-- 10 tabel COMENZI
insert into comenzi
values (secventa_techflow.nextval, to_date('2023-05-15', 'YYYY-MM-DD'), 'Noua', 46, 36, 41);
insert into comenzi
values (secventa_techflow.nextval, to_date('2023-05-10', 'YYYY-MM-DD'), 'Confirmata', 47,
37, 41);
insert into comenzi
values (secventa_techflow.nextval, to_date('2023-04-15', 'YYYY-MM-DD'), 'Anulata', 46, 38,
41);
insert into comenzi
```

```
values (seventa_techflow.nextval, to_date('2023-05-15', 'YYYY-MM-DD'), 'Noua', 48, 39, 42);
insert into comenzi
values (seventa_techflow.nextval, to_date('2023-05-09', 'YYYY-MM-DD'), 'Confirmata', 49,
40, 44);
select * from comenzi;

-- 11 tabel PRODUSE_COMENZI
insert into produse_comenzi
values (26,51);
insert into produse_comenzi
values (30,51);
insert into produse_comenzi
values (29,52);
insert into produse_comenzi
values (35,52);
insert into produse_comenzi
values (31,52);
insert into produse_comenzi
values (27,53);
insert into produse_comenzi
values (34,53);
insert into produse_comenzi
values (28,54);
insert into produse_comenzi
values (31,54);
insert into produse_comenzi
values (29,55);
insert into produse_comenzi
values (33,55);
select * from produse_comenzi;

-- 12 tabel WISHLIST
insert into wishlist
values (seventa_techflow.nextval, to_date('2023-05-14', 'YYYY-MM-DD'), 46);
insert into wishlist
values (seventa_techflow.nextval, to_date('2023-05-13', 'YYYY-MM-DD'), 47);
insert into wishlist
values (seventa_techflow.nextval, to_date('2023-05-12', 'YYYY-MM-DD'), 48);
insert into wishlist
values (seventa_techflow.nextval, to_date('2023-05-11', 'YYYY-MM-DD'), 49);
insert into wishlist
values (seventa_techflow.nextval, to_date('2023-05-10', 'YYYY-MM-DD'), 50);
```

```
select * from wishlist;

-- 13 tabel PRODUSE_WISHLIST
insert into produse_wishlist
values (27,56);
insert into produse_wishlist
values (28,56);
insert into produse_wishlist
values (29,57);
insert into produse_wishlist
values (31,57);
insert into produse_wishlist
values (31,58);
insert into produse_wishlist
values (27,58);
insert into produse_wishlist
values (34,59);
insert into produse_wishlist
values (27,59);
insert into produse_wishlist
values (31,60);
insert into produse_wishlist
values (28,60);
insert into produse_wishlist
values (26,60);
select * from produse_wishlist;

-- 14 tabel RECENZII
insert into recenzii
values (seventa_techflow.nextval, 4.5, 46, 26);
insert into recenzii
values (seventa_techflow.nextval, 3.8, 47, 27);
insert into recenzii
values (seventa_techflow.nextval, 4.2, 48, 28);
insert into recenzii
values (seventa_techflow.nextval, 5.0, 49, 29);
insert into recenzii
values (seventa_techflow.nextval, 3.5, 50, 30);
select * from recenzii;
```

Cerința 6.

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

Pentru primele 5 produse (varray) în ordine alfabetica după nume sa se afiseze codul, numele, pretul, stocul, în ce magazine se află (t indexat) și recenziile lasate de clienti (t imbr).

The screenshot shows the Oracle SQL Developer interface. At the top, the code for the PL/SQL procedure `PRODUSE_INFO` is displayed:

```
35
36 begin
37
38     dbms_output.put_line('##### Informatii despre produse - magazine - recenzii #####');
39     dbms_output.new_line();
40
41     -- atribuire produse
42     select id_produs,nume_produs,pret_produs,stoc_produs bulk collect into t_prod
43     from produse
44     order by nume_produs
45     fetch first 5 rows only;
46
47     for i in t_prod.first..t_prod.last loop
48         -- afisare info produs
49         dbms_output.put_line('Produsul ' || i);
50         dbms_output.put_line('#####');
51         dbms_output.put_line('Id: ' || t_prod(i).cod_prod || ' Nume: ' || t_prod(i).nume);
```

Below the code, the output window shows the procedure was compiled successfully and completed in 0.169 seconds:

Procedure `PRODUSE_INFO` compiled
PL/SQL procedure successfully completed.

The bottom section shows the execution results for the first five products:

Produsul	Id	Nume	Pret	Stoc	Magazine	Recenziile
1	29	Corsair Vengeance RGB Pro	699.99	20	TechFlow Green, TechFlow Red	5
2	33	Gigabyte GeForce GTX 1660 SUPER OC	1799.99	10	TechFlow Yellow	0
3	27	Intel Core i9-10900K	2899.99	5	TechFlow Black	0
4	32	AMD Ryzen 7 5800X	1999.99	7		
5	26	ASUS ROG Strix X570-E Gaming	1599.99	10		

```
create or replace procedure produse_info as
    -- varray de record-uri pentru produse
    type prod_rec is record (
        cod_prod produse.id_produs%type,
        nume produse.nume_produs%type,
        pret produse.pret_produs%type,
        stoc produse.stoc_produs%type
    );
    type prod_vec is varray(5) of prod_rec;
    t_prod prod_vec := prod_vec();

    -- tablou indexat pentru magazine
    type mag_rec is record (
        nume magazine.nume_magazin%type,
        adresa magazine.adresa_magazin%type
    );
    type mag_ind is table of mag_rec index by binary_integer;
    t_mag mag_ind;

    -- tablou imbricat pentru recenzii
    type recenzii_rec is record (
        nota recenzii.nota_recenzie%type,
        nume_client clienti.nume_client%type
    );
    type rec_imbr is table of recenzii_rec;
    t_recenzii rec_imbr := rec_imbr();

begin
    dbms_output.put_line('##### Informatii despre produse - magazine - recenzii #####');
    dbms_output.new_line();
    -- atribuire produse
    select id_produs, nume_produs, pret_produs, stoc_produs bulk collect into t_prod
    from produse
    order by nume_produs
    fetch first 5 rows only;

    for i in t_prod.first..t_prod.last loop
        -- afisare info produs
        dbms_output.put_line('Produsul ' || i);
        dbms_output.put_line('#####');
        dbms_output.put_line('Id: ' || t_prod(i).cod_prod || ' Nume: ' || t_prod(i).nume);
        dbms_output.put_line('Pret: ' || t_prod(i).pret || ' Stoc: ' || t_prod(i).stoc);
    end loop;
end;
```

```

dbms_output.put_line('#####');
-- atribuire magazine
select m.nume_magazin,m.adresa_magazin bulk collect into t_mag
from magazine m
join produse_magazine pm on (m.id_magazin = pm.id_magazin)
where pm.id_produs = t_prod(i).cod_prod;
-- afisare info magazin
if t_mag.count = 0 then
    dbms_output.put_line('Produsul nu se afla intr-un magazin!');
    dbms_output.new_line;
else
    dbms_output.put_line('Magazine: ');
    for j in t_mag.first..t_mag.last loop
        dbms_output.put_line('Magazinul ' || j);
        dbms_output.put_line('Nume: ' || t_mag(j).nume || ' Adresa: ' || t_mag(j).adresa);
        dbms_output.new_line();
    end loop;
end if;
dbms_output.put_line('#####');
-- atribuire recenzii
select r.nota_recenzie,c.nume_client bulk collect into t_recenzii
from recenzii r
join clienti c on (r.id_client = c.id_client)
where r.id_produs = t_prod(i).cod_prod;
-- afisare recenzii
if t_recenzii.count = 0 then
    dbms_output.put_line('Produsul nu are recenzii!');
    dbms_output.new_line;
else
    dbms_output.put_line('Recenzii: ');
    for k in t_recenzii.first..t_recenzii.last loop
        dbms_output.put_line('Recenzia ' || k);
        dbms_output.put_line('Nota: ' || t_recenzii(k).nota || ' Nume Client: ' ||
t_recenzii(k).nume_client);
        dbms_output.new_line();
    end loop;
end if;
dbms_output.new_line();
end loop;
end produse_info;
/
execute produse_info;

```

Cerință 7.

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.

Pentru fiecare categorie să se determine primele 2 produse ordonate descrescător după stoc (cursor dinamic). Sa se afiseze detalii despre comenzi care contin produsele determinate anterior (cursor parametrizat).

The screenshot shows the Oracle SQL Developer interface. The top half displays a PL/SQL procedure named 'COMENZI_INFO' with line numbers 174 to 190. The code uses DBMS_OUTPUT.PUT_LINE to print information about products and orders. The bottom half shows the 'Script Output' tab with the message 'Procedure COMENZI_INFO compiled' and 'PL/SQL procedure successfully completed.'

```
174      dbms_output.put_line('Nu există produse!');  
175  else  
176    for i in v_prod.first..v_prod.last loop  
177      dbms_output.put_line('Produs '||i||'. '|| v_prod(i).nume);  
178      dbms_output.put_line('#####');  
179      open com_info(v_prod(i).id_produs);  
180    loop  
181      fetch com_info into v_com;  
182      exit when com_info%notfound;  
183      dbms_output.put_line('Comanda '||contor);  
184      dbms_output.put_line('Data Plasare: ' || v_com.data_plasare || ' Status Comanda: ' || v_com.status_  
185      dbms_output.put_line('Total Plata: ' || v_com.total_plata || ' Tip Plata: ' || v_com.tip_plata);  
186      dbms_output.new_line;  
187      contor := contor + 1;  
188    end loop;  
189  
190    if com_info%rowcount = 0 then
```

Script Output | Query Result | Task completed in 0.104 seconds

Procedure COMENZI_INFO compiled

PL/SQL procedure successfully completed.

```
docker_project x | docker_project x |
##### Informatii despre categorii - produse - comenzi - plati #####
Categoria: Placi de baza cu id-ul: 21
#####
Produs 1. MSI B450 TOMAHAWK MAX
#####
Comanda 1
Data Plasare: 15-MAY-23 Status Comanda: Noua
Total Plata: 5899.98 Tip Plata: Card

Comanda 2
Data Plasare: 10-MAY-23 Status Comanda: Confirmata
Total Plata: 1699.97 Tip Plata: Numerar

#####
Produs 2. ASUS ROG Strix X570-E Gaming
#####
Comanda 1
Data Plasare: 15-MAY-23 Status Comanda: Noua
Total Plata: 2598.99 Tip Plata: Card
#####

docker_project x | docker_project x |
Categoria: Procesoare cu id-ul: 22
#####
Produs 1. AMD Ryzen 7 5800X
#####
Nu are comenzi!
#####

Produs 2. Intel Core i9-10900K
#####
Comanda 1
Data Plasare: 15-APR-23 Status Comanda: Anulata
Total Plata: 4199.98 Tip Plata: Transfer bancar
#####

docker_project x | docker_project x |
Categoria: Placi video cu id-ul: 23
#####
Produs 1. Gigabyte GeForce GTX 1660 SUPER OC
#####
Comanda 1
Data Plasare: 09-MAY-23 Status Comanda: Confirmata
Total Plata: 2499.98 Tip Plata: Numerar

#####
Produs 2. NVIDIA GeForce RTX 3080
#####
Comanda 1
Data Plasare: 15-MAY-23 Status Comanda: Noua
Total Plata: 5899.98 Tip Plata: Card
#####

Categoria: Memori RAM cu id-ul: 24
#####
Produs 1. Corsair Vengeance RGB Pro
#####
Comanda 1
Data Plasare: 10-MAY-23 Status Comanda: Confirmata
Total Plata: 1699.97 Tip Plata: Numerar

#####
Produs 2. RAM Kingston HyperX Fury RGB
#####
Comanda 1
Data Plasare: 09-MAY-23 Status Comanda: Confirmata
Total Plata: 2499.98 Tip Plata: Numerar
#####

Categoria: Stocare ROM cu id-ul: 25
#####
Produs 1. SSD Western Digital Blue
#####
Comanda 1
Data Plasare: 10-MAY-23 Status Comanda: Confirmata
Total Plata: 1699.97 Tip Plata: Numerar

#####
Produs 2. Samsung 970 EVO Plus
#####
Comanda 1
Data Plasare: 15-MAY-23 Status Comanda: Noua
Total Plata: 2598.99 Tip Plata: Card
#####

Categoria: Microfoane cu id-ul: 71
#####
Nu exista produse!
```

```
create or replace procedure comenzi_info as
```

```
type com_rec is record (
    data_plasare comenzi.data_plasare%type,
    status_comanda comenzi.status_comanda%type,
    total_plata plati.total_plata%type,
    tip_plata plati.tip_plata%type
);

type prod_rec is record (
    nume_produse.nume_produs%type,
    id_produse.produse.id_produs%type
);

type t_prod is table of prod_rec
index by binary_integer;
v_prod t_prod;

v_com com_rec;

-- cursor dinamic/ expresie cursor
-- pentru fiecare categorie se determina produsul cu cel mai mare stoc
cursor categorii_info is
    select id_categoria,nume_categoria,
        cursor(select max(nume_produs),max(id_produs)
            from produse p
            where c.id_categoria = p.id_categoria
            group by p.id_produs
            order by max(stoc_produs) desc)
    from categorii c;

-- cursor parametrizat pentru comenzi
cursor com_info(v_produs produse.id_produs%type) is
    select c.data_plasare,c.status_comanda,p.total_plata,p.tip_plata
    from comenzi c
    join plati p on (c.id_plata = p.id_plata)
    join produse_comenzi pc on (c.id_comanda = pc.id_comanda)
    where pc.id_produs = v_produs
    order by c.data_plasare desc;

v_cursor sys_refcursor;
```

```
v_id_categorie categorii.id_categorie%type;
v_nume_categorie categorii.nume_categorie%type;
contor number := 1;
begin
    dbms_output.put_line('##### Informatii despre categorii - produse - comenzi - plati');
    open categorii_info;
    loop
        fetch categorii_info into v_id_categorie,v_nume_categorie,v_cursor;

        exit when categorii_info%notfound;
        dbms_output.put_line('Categoria: ' || v_nume_categorie || ' cu id-ul: ' || v_id_categorie);
        dbms_output.put_line('#####');
        fetch v_cursor bulk collect into v_prod limit 2;
        dbms_output.new_line;

        if v_prod.count = 0 then
            dbms_output.put_line('Nu exista produse!');
        else
            for i in v_prod.first..v_prod.last loop
                dbms_output.put_line('Produs '||i||'. '|| v_prod(i).nume);
                dbms_output.put_line('#####');
                open com_info(v_prod(i).id_produs);
                loop
                    fetch com_info into v_com;
                    exit when com_info%notfound;
                    dbms_output.put_line('Comanda'||contor);
                    dbms_output.put_line('Data Plasare: '|| v_com.data_plasare || ' Status Comanda: ' ||
v_com.status_comanda);
                    dbms_output.put_line('Total Plata: '|| v_com.total_plata || ' Tip Plata: ' ||
v_com.tip_plata);
                    dbms_output.new_line;
                    contor := contor + 1;
                end loop;

                if com_info%rowcount = 0 then
                    dbms_output.put_line('Nu are comenzi!');
                end if;
                close com_info;

                contor := 1;
                dbms_output.put_line('#####');
```

```
dbms_output.new_line;
end loop;
end if;

dbms_output.new_line;
end loop;
close categorii_info;
end comenzi_info;
/
execute comenzi_info;
```

Cerință 8.

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite și tratate

Sa se returneze comenziile plasate într-o anumita perioadă care nu depasesc un anumit cost de transport și un anumit cost total.

```
214 create or replace function filtrare_comenzi
215   (v_data_min date, v_data_max date, v_transport_maxim transport.cost_transport%type,
216  return t_id_com is
217    v_id_com t_id_com;
218
219    data_eroare exception;
220    transport_eroare exception;
221    cost_total_eroare exception;
222    no_comenzi exception;
223 begin
224
225    if v_data_min > v_data_max then
226      raise data_eroare;
227    end if;
228
229    if v_transport_maxim < 0 then
230      raise transport eroare;
```

Script Output x Query Result x
| Task completed in 0.075 seconds

Function FILTRARE_COMENZI compiled

```

create or replace type t_id_com is table of number;
/
-- returneaza un tablou de id-uri
create or replace function filtrare_comenzi
    (v_data_min date, v_data_max date, v_transport_maxim transport.cost_transport%type,
v_cost_total plati.total_plata%type)
return t_id_com is
    v_id_com t_id_com;
    data_eroare exception;
    transport_eroare exception;
    cost_total_eroare exception;
    no_comenzi exception;
begin
    if v_data_min > v_data_max then
        raise data_eroare;
    end if;
    if v_transport_maxim < 0 then
        raise transport_eroare;
    end if;
    if v_cost_total < 0 then
        raise cost_total_eroare;
    end if;
    -- folosirea a 3 table
    select distinct c.id_comanda bulk collect into v_id_com
    from comenzi c
    join plati p on (c.id_plata = p.id_plata)
    join transport t on (c.id_transport = t.id_transport)
    where c.data_plasare >= v_data_min and c.data_plasare <= v_data_max
    and t.cost_transport <= v_transport_maxim
    and p.total_plata <= v_cost_total;
    if v_id_com.count = 0 then
        raise no_comenzi;
    end if;
    return v_id_com;
exception
    when no_comenzi then
        raise_application_error(-20000,'Nu exista comanda cu aceste conditii!');
    when data_eroare then
        raise_application_error(-20001,'Intervalul nu este introdus corect!');
    when transport_eroare then
        raise_application_error(-20002,'Costul transportului nu poate fi negativ!');
    when cost_total_eroare then

```

```
raise_application_error(-20003,'Costul total nu poate fi negativ!');  
when others then  
    raise_application_error(-20004,'Alta eroare!');  
end filtrare_comenzi;  
/  
-- caz 1 apel corect  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 5000');  
    dbms_output.put_line('si costul de transport mai mic de 190.');
```

solutie := filtrare_comenzi(to_date('10-may-2023', 'dd-mon-yyyy'),to_date('15-may-2023', 'dd-mon-yyyy'),190,5000);
dbms_output.new_line;
dbms_output.put_line('Id-uri comenzi care respecta criteriul:');
for i in solutie.first..solutie.last loop
 dbms_output.put_line(solutie(i)|| ' ');
end loop;
end;
/
-- caz2 exceptie no_comenzi
-- caz in care nu exista comenzi care sa respecte conditiile
declare
 solutie t_id_com;
begin
 dbms_output.put_line('Comenzi intre 10-15 mai 2022 cu totalul de plata mai mic de 3000');
 dbms_output.put_line('si costul de transport mai mic de 100.');

dbms_output.new_line;
solutie := filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),100,3000);
exception
 when others then
 dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul ' || SQLERRM);
end;
/
-- caz3 exceptie data_eroare
-- caz in care data minima este mai mare decat data maxima
declare
 solutie t_id_com;
begin
 dbms_output.put_line('Comenzi intre 15-10 mai 2022 cu totalul de plata mai mic de 3000');
 dbms_output.put_line('si costul de transport mai mic de 100.');

```
dbms_output.new_line;
solutie := filtrare_comenzi(to_date('15-may-2022', 'dd-mon-yyyy'),to_date('10-may-2022', 'dd-mon-yyyy'),100,3000);
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul ' || SQLERRM);
end;
/
-- caz4 exceptie transport_eroare
-- caz in care costul pentru transport maximal este negativ
declare
    solutie t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 3000');
    dbms_output.put_line('si costul de transport mai mic de -10.');
    dbms_output.new_line;
    solutie := filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),-10,3000);
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul ' || SQLERRM);
end;
/
-- caz5 exceptie cost_total_eroare
-- caz in care costul total maximal este negativ
declare
    solutie t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de -50');
    dbms_output.put_line('si costul de transport mai mic de 200.');
    dbms_output.new_line;
    solutie := filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),200,-50);
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul ' || SQLERRM);
end;
/
```

```
/  
  
-- caz 1 apel corect  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 500  
si costul de transport mai mic de 190.');  
    solutie := filtrare_comenzi(to_date('10-may-2023', 'dd-mon-yyyy'), to_date('15-may-2023', 'dd-mon-yyyy'));  
    dbms_output.new_line;  
    dbms_output.put_line('Id-uri comenzi care respecta criteriul:');  
    for i in solutie.first..solutie.last loop  
        dbms_output.put_line(solutie(i)|| ' ');  
    end loop;  
end;  
  
-- caz2 exceptie no_comenzi  
-- caz in care nu exista comenzi care sa respecte conditiile  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 10-15 mai 2022 cu totalul de plata mai mic de 300  
si costul de transport mai mic de 100.');  
    dbms_output.new_line;  
    solutie := filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'), to_date('15-may-2022', 'dd-mon-yyyy'));  
exception  
    when others then  
        dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);  
end;  
  
-- caz3 exceptie data_eroare  
-- caz in care data minima este mai mare decat data maxima  
  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 15-10 mai 2022 cu totalul de plata mai mic de 300  
si costul de transport mai mic de 100.');  
    dbms_output.new_line;  
    solutie := filtrare_comenzi(to_date('15-may-2022', 'dd-mon-yyyy'), to_date('10-may-2022', 'dd-mon-yyyy'));  
exception  
    when others then  
        dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);  
end;  
  
-- caz4 exceptie transport_eroare  
-- caz in care costul pentru transport maximal este negativ  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 300  
si costul de transport mai mic de -10.');  
    dbms_output.new_line;  
    solutie := filtrare_comenzi(to_date('10-may-2023', 'dd-mon-yyyy'), to_date('15-may-2023', 'dd-mon-yyyy'));  
exception  
    when others then  
        dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);  
end;  
  
-- caz5 exceptie cost_total_eroare  
-- caz in care costul total maximal este negativ  
declare  
    solutie t_id_com;  
begin  
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de -50  
si costul de transport mai mic de 200.');  
    dbms_output.new_line;  
    solutie := filtrare_comenzi(to_date('10-may-2023', 'dd-mon-yyyy'), to_date('15-may-2023', 'dd-mon-yyyy'));  
exception  
    when others then  
        dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);  
end;
```

Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 500
si costul de transport mai mic de 190.
Id-uri comenzi care respecta criteriul:
51
52

Comenzi intre 10-15 mai 2022 cu totalul de plata mai mic de 300
si costul de transport mai mic de 100.
Eroare cu codul -20000 si mesajul ORA-20000: Nu exista comanda cu aceste conditii!

Comenzi intre 15-10 mai 2022 cu totalul de plata mai mic de 300
si costul de transport mai mic de 100.
Eroare cu codul -20001 si mesajul ORA-20001: Intervalul nu este introdus corect!

Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 300
si costul de transport mai mic de -10.
Eroare cu codul -20002 si mesajul ORA-20002: Costul transportului nu poate fi negativ!

Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de -50
si costul de transport mai mic de 200.
Eroare cu codul -20003 si mesajul ORA-20003: Costul total nu poate fi negativ!

Cerință 9.

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate exceptiile care pot apărea, inclusiv exceptiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Definiti un subprogram stocat de tip procedura care sa afiseze numarul de recenzii, numarul de produse in wishlist, numarul de comenzi si numarul de produse din comenzile neanulate ale clientului al caror nume este dat.

De asemenea, sa se afiseze numele si prenumele clientului.

```
373 if t_nume_client.count = 0 then
374     raise no_data_found;
375 elsif t_nume_client.count > 1 then
376     raise too_many_rows;
377 else
378     dbms_output.put_line('Informatii client: ');
379     dbms_output.new_line;
380
381 -- 6 table intr-o comanda sql
382 select
383     (select count(*) from recenzii r
384      join clienti cl on (cl.id_client = r.id_client)
385      where cl.numere_client like v_nume_client
386      )as nr_recenzii,
387     (select count(*) from wishlist w
388      join clienti cl on (cl.id_client = w.id_client)
389      where cl.numere_client like v_nume_client) as nr_wishlist,
390     (select count(*) from comenzi c
391      join clienti cl on (cl.id_client = c.id_client)
392      where cl.numere_client like v_nume_client)as nr_comenzi,
393     (select count(*) from produse p
394      join produse_comenzi pc on (pc.id_produs = p.id_produs)
395      join comenzi c on (c.id_comanda = pc.id_comanda)
396      join clienti cl on (cl.id_client = c.id_client)
397      where cl.numere_client like v_nume_client and c.status_comanda <> 'Anulata'
398      ) as nr_produse_neanulate,
399     cl.numere_client,cl.prenume_client into l.info
```

```

create or replace procedure clienti_info(v_nume_client clienti.nume_client%type)
is
type info_rec is record (
    nr_recenzii integer,
    nr_wishlist integer,
    nr_comenzi integer,
    nr_produse_neanulate integer,
    nume_client clienti.nume_client%type,
    prenume_client clienti.prenume_client%type
);
l_info info_rec;
type nume_client_imbr is table of clienti.nume_client%type;
t_nume_client nume_client_imbr := nume_client_imbr();
begin
select nume_client bulk collect into t_nume_client
from clienti
where nume_client like v_nume_client || '%';
if t_nume_client.count = 0 then
    raise no_data_found;
elsif t_nume_client.count > 1 then
    raise too_many_rows;
else
    dbms_output.put_line('Informatii client: ');
    dbms_output.new_line;
    -- 6 table intr-o comanda sql
    select
        (select count(*) from recenzii r
         join clienti cl on (cl.id_client = r.id_client)
         where cl.nume_client like v_nume_client
        )as nr_recenzii,
        (select count(*) from wishlist w
         join clienti cl on (cl.id_client = w.id_client)
         where cl.nume_client like v_nume_client) as nr_wishlist,
        (select count(*) from comenzi c
         join clienti cl on (cl.id_client = c.id_client)
         where cl.nume_client like v_nume_client)as nr_comenzi,
        (select count(*) from produse p
         join produse_comenzi pc on (pc.id_produs = p.id_produs)
         join comenzi c on (c.id_comanda = pc.id_comanda)
         join clienti cl on (cl.id_client = c.id_client)
         where cl.nume_client like v_nume_client and c.status_comanda <> 'Anulata'
        ) as nr_produse_neanulate,

```

```

cl.nume_client,cl.prenume_client into l_info
from clienti cl
where cl.nume_client like v_nume_client;
dbms_output.put_line('Nume: ' || l_info.nume_client);
dbms_output.put_line('Prenume: ' || l_info.prenume_client);
dbms_output.new_line;
dbms_output.put_line('Numar recenzii: ' || l_info.nr_recenzii);
dbms_output.put_line('Numar produse wishlist: ' || l_info.nr_wishlist);
dbms_output.put_line('Numar comenzi: ' || l_info.nr_comenzi);
dbms_output.put_line('Numar produse din comenziile neanulate: ' ||
l_info.nr_produse_neanulate); end if;
exception
when no_data_found then
    raise_application_error(-20000,'Nu exista client cu acest nume!');
when too_many_rows then
    raise_application_error(-20001,'Prea multi clienti cu acest string!');
when others then
    raise_application_error(-20002,'Alta eroare!');
end clienti_info;
/
-- caz 1 apel corect
begin
    clienti_info('Avram');
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz 2 no_data_found (nu exista cu client cu acest nume)
begin
    clienti_info('Huma');
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz 3 too_many_rows (pentru Pop => exista Popa si Popescu)
begin clienti_info('Pop');
exception
when others then
    dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;

```

```

418  'end clienti_info;
419  /
420
421 -- caz 1 apel corect
422 begin
423   clienti_info('Avram');
424 exception
425   when others then
426     dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);
427 end;
428 /

```

Informatii client:

Nume: Avram
 Prenume: Gabriela
 Numar recenzii: 1
 Numar produse wishlist: 1
 Numar comenzi: 2
 Numar produse din comenzile neanulate: 2

```

430 -- caz 2 no_data_found (nu exista cu client cu acest nume)
431 begin
432   clienti_info('Huma');
433 exception
434   when others then
435     dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);
436 end;
437 /

```

Eroare cu codul -20000 si mesajul ORA-20000: Nu exista client cu acest nume!

```

439 -- caz 3 too_many_rows (pentru Pop => exista Popa si Popescu)
440 begin
441   clienti_info('Pop');
442 exception
443   when others then
444     dbms_output.put_line('Eroare cu codul'||SQLCODE|| ' si mesajul '|| SQLERRM);
445 end;
446 /

```

Eroare cu codul -20001 si mesajul ORA-20001: Prea multi clienti cu acest string!

Cerință 10.

Definiți un trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

Trigger LMD la nivel de comandă care nu permite executarea unei comenzi pe tabelul angajati intre orele 01:00 - 03:00 in ziua de 8 si 9 a lunilor ianuarie - septembrie.

The screenshot shows the Oracle SQL Developer interface with the code editor and a message bar. The code editor contains PL/SQL code for a trigger named 'check_angajati'. The message bar at the bottom indicates the task was completed in 0.513 seconds.

```
454 create or replace trigger check_angajati
455   before insert or delete or update on angajati
456 declare
457   valid number := 1;
458 begin
459   for i in 1..9 loop
460     if to_char(sysdate,'dd/mm') = '08/0' || i or to_char(sysdate,'dd/mm') = '07/0' || i then
461       valid := 0;
462     end if;
463   end loop;
464   if to_char(sysdate, 'hh24') between 1 and 3
465     then valid := 0;
466   end if;
467
468   if valid = 0 then
469     if inserting then
470       raise_application_error(-20000,'Inserati date mai tarziu!');
471     elsif deleting then
472       raise_application_error(-20001,'Stergeti date mai tarziu!');
473     else
474       raise_application_error(-20002,'Modificati date mai tarziu!');
475     end if;
476   end if;

```

Script Output x | Query Result x
Task completed in 0.513 seconds

Trigger CHECK_ANGAJATI compiled

```
create or replace trigger check_angajati
  before insert or delete or update on angajati
declare
  valid number := 1;
begin
  for i in 1..9 loop
    if to_char(sysdate,'dd/mm') = '08/0' || i or to_char(sysdate,'dd/mm') = '07/0' || i then
      valid := 0;
    end if;
  end loop;
  if to_char(sysdate, 'hh24') between 1 and 3
    then valid := 0;
  end if;

  if valid = 0 then
    if inserting then
      raise_application_error(-20000,'Inserati date mai tarziu!');
    elsif deleting then
      raise_application_error(-20001,'Stergeti date mai tarziu!');
    else
      raise_application_error(-20002,'Modificati date mai tarziu!');
    end if;
  end if;
end;
/
-- inserare
insert into angajati
values (secenta_techflow.nextval, 'Gabroveanu', 'Ion', 'gabroveanu.ion@example.com', 2500.00,
'Casier', 1);

-- stergere
delete from angajati where id_angajat = 6;

-- modificare
update angajati
set id_angajat = 1000
where id_angajat = 7;
```

```
480 | -- inserare
481 | insert into angajati
482 | values (secventa_techflow.nextval, 'Gabroveanu', 'Ion', 'gabroveanu.example.com', 2500.00, 'Casier', 1);
483 |
Script Output x
| Task completed in 0.213 seconds
Error at Command Line : 481 Column : 13
Error report -
SQL Error: ORA-20000: Inserati date mai tarziu!
ORA-06512: at "PROIECT.CHECK_ANAJATI", line 15
ORA-04088: error during execution of trigger 'PROIECT.CHECK_ANAJATI'
20000. 00000 - "%s"
```

```
484 | -- stergere
485 | delete from angajati where id_angajat = 6;
486 |
```

```
Script Output x
| Task completed in 0.083 seconds
```

```
Error starting at line : 485 in command -
delete from angajati where id_angajat = 6
Error at Command Line : 485 Column : 13
Error report -
SQL Error: ORA-20001: Stergeti date mai tarziu!
ORA-06512: at "PROIECT.CHECK_ANAJATI", line 17
ORA-04088: error during execution of trigger 'PROIECT.CHECK_ANAJATI'
```

```
487 | -- modificare
488 | update angajati
489 | set id_angajat = 1000
490 | where id_angajat = 7;
```

```
Script Output x
| Task completed in 0.079 seconds
```

```
Error starting at line : 488 in command -
update angajati
set id_angajat = 1000
where id_angajat = 7
Error at Command Line : 488 Column : 8
Error report -
SQL Error: ORA-20002: Modificati date mai tarziu!
ORA-06512: at "PROIECT.CHECK_ANAJATI", line 19
ORA-04088: error during execution of trigger 'PROIECT.CHECK_ANAJATI'
```

Cerință 11.

Definiți un trigger de tip LMD la nivel de linie. Declanșați trigger-ul.

■ Varianta 1

(se va folosi o procedura de actualizare a sumei totale în trigger)

Trigger LMD la nivel de linie care va actualiza automat campul suma_totala din tabelul info_categorii cand se introduce un nou produs, se sterge un produs sau se modifica pretul produsului.

The screenshot shows the Oracle SQL Developer interface with two main sections: a code editor and a script output window.

Code Editor:

```
1065 | where c.id_categorie = p.id_categorie();
1066 | group by c.id_categorie,c.numere_categorie;
1067 |
1068 | select * from info_categorii;
1069 |
1070 | -- procedura care actualizeaza suma_totala din info_categorii
1071 | create or replace procedure modificare_suma_categorii
1072 |   ( v_id info_categorii.id%type,
1073 |     v_suma_totala info_categorii.suma_totala%type
1074 |   ) as
1075 | begin
1076 |   update info_categorii
1077 |   set suma_totala = nvl(suma_totala,0) + v_suma_totala
1078 |   where id = v_id;
1079 | end;
1080 |
1081 |
1082 | -- trigger LMD
1083 | create or replace trigger lmd_pret_produc
1084 |   after delete or update or insert of pret_produc
```

Script Output:

Table INFO_CATEGORII created.
6 rows inserted.
Procedure MODIFICARE_SUMA_CATEGORII compiled
Trigger LMD_PRET_PRODUS compiled

```
-- Tabel info_categorii in care suma_totala reprezinta suma preturilor
-- produselor din categoria respectiva
create table info_categorii
(id number(3) primary key,
nume_categorie varchar2(50),
suma_totala number);

-- adaugare date in tabel
insert into info_categorii
select c.id_categorie,c.nume_categorie,nvl(sum(p.pret_produs),0)
from categorii c, produse p
where c.id_categorie = p.id_categorie(+)
group by c.id_categorie,c.nume_categorie;

select * from info_categorii;

-- procedura care actualizeaza suma_totala din info_categorii
create or replace procedure modificare_suma_categorii
( v_id info_categorii.id%type,
  v_suma_totala info_categorii.suma_totala%type
 ) as
begin
  update info_categorii
  set suma_totala = nvl(suma_totala,0) + v_suma_totala
  where id = v_id;
end;
/
-- trigger LMD
create or replace trigger lmd_pret_produs
  after delete or update or insert of pret_produs
  on produse
  for each row – la nivel de linie
begin
  if deleting then
    -- stergerea unui produs
    modificare_suma_categorii(:old.id_categorie,-1*:old.pret_produs);
  elsif updating then
    -- modificarea unui produs
    modificare_suma_categorii(:old.id_categorie,:new.pret_produs-:old.pret_produs);
  else
    -- adaugarea unui produs
    modificare_suma_categorii(:new.id_categorie,:new.pret_produs);
```

```

end if;
end;
/
select * from info_categorii;
select * from produse;

-- stergere produs
delete from produse where id_produs = 75;

-- modificare produs
update produse
set pret_produs = 1999.99
where id_produs = 27;

-- adaugare produs
insert into produse
values (seventa_techflow.nextval, 'Microfon Test', '-', 1799.99, 25, 16, 71);

```

stergere

inainte de trigger

ID	NUME_CATEGORIE	SUMA_TOTALA
1	21 Placi de baza	3599.97

```

1104 -- stergere produs
1105 delete from produse where id_produs = 76;

```

Script Output x | Query Result x
Task completed in 0.057 seconds

1 row deleted.

dupa trigger

ID	NUME_CATEGORIE	SUMA_TOTALA
1	21 Placi de baza	2199.98

modificare

inainte de trigger

2	22 Procesoare	4899.98
---	---------------	---------

```

1107 -- modificare produs
1108 update produse
1109 set pret_produs = 1999.99
1110 where id_produs = 27;
1111

```

Script Output x | Query Result x
Task completed in 0

1 row updated.

dupa trigger

2	22 Procesoare	3999.98
---	---------------	---------

inserare

inainte de trigger

6	71 Microfoane	0
---	---------------	---

```

1111 -- adaugare produs
1112 insert into produse|
1113 values (seventa_techflow.nextval, 'Microfon Test', '-', 1799.99, 25, 16, 71);

```

Script Output x | Query Result x
Task completed in 0.062 seconds

1 row inserted.

dupa trigger

6	71 Microfoane	1799.99
---	---------------	---------

■ Varianta 2

(trigger pe tabela mutating) Trigger LMD la nivel de linie care sa nu permita ca un client sa aiba mai mult de 2 comenzi.

The screenshot shows the SQL Developer interface with a code editor containing the following PL/SQL code:

```
1120 create or replace trigger check_comenzi
1121   before insert or update of id_client on comenzi
1122   for each row
1123 declare
1124   numer_comenzi number;
1125 begin
1126   select count(*) into numer_comenzi
1127   from comenzi
1128   where id_client = :new.id_client;
1129
1130   if numer_comenzi = 2 then
1131     raise_application_error(-20005,'Clientul are deja 2
1132   end if;
1133 end;
```

Below the code editor, the status bar displays "Trigger CHECK COMENZI compiled".

-- Varianta 2
-- (trigger pe tabela mutating)
-- Trigger LMD la nivel de linie care sa nu permita ca un client sa aiba mai mult
-- de 2 comenzi.

```
create or replace trigger check_comenzi
  before insert or update of id_client on comenzi
  for each row
declare
  numer_comenzi number;
begin
  select count(*) into numer_comenzi
  from comenzi
  where id_client = :new.id_client;

  if numer_comenzi = 2 then
    raise_application_error(-20005,'Clientul are deja 2 comenzi in desfasurare!');
  end if;
end;
/
select * from clienti;
```

```
select * from comenzi;

-- nu se declanseaza triggerul deoarece clientul are 0 comenzi
insert into comenzi
values (secventa_techflow.nextval, sysdate, 'Noua', 72, 36, 42);

-- clientul are numarul maxim de comenzi
insert into comenzi
values (secventa_techflow.nextval, sysdate, 'Noua', 46, 38, 44);

-- eroare mutating
-- clientul are 0 comenzi dar insert into + select genereaza eroare
insert into comenzi
select secventa_techflow.nextval, sysdate, 'Noua', 73, 37, 42
from dual;
```

```
1138 -- nu se declanseaza triggerul deoarece clientul are 0 comenzi
1139 insert into comenzi
1140 values (secventa_techflow.nextval, sysdate, 'Noua', 72, 36, 42);
1141
```

```
Script Output X | Query Result X
| Task completed in 0.045 seconds
```

1 row inserted.

```
1142 -- clientul are numarul maxim de comenzi
1143 insert into comenzi
1144 values (secventa_techflow.nextval, sysdate, 'Noua', 46, 38, 44);
1145
```

```
Script Output X | Query Result X
| Task completed in 0.062 seconds
```

```
Error starting at line : 1,143 in command -
insert into comenzi
values (secventa_techflow.nextval, sysdate, 'Noua', 46, 38, 44)
Error at Command Line : 1,144 Column : 9
Error report -
SQL Error: ORA-20005: Clientul are deja 2 comenzi in desfasurare!
ORA-06512: at "PROIECT.CHECK_COMENZI", line 9
ORA-04088: error during execution of trigger 'PROIECT.CHECK_COMENZI'
```

```
1146 -- eroare mutating
1147 -- clientul are 0 comenzi dar insert into + select genereaza eroare
1148 insert into comenzi
1149 select secventa_techflow.nextval, sysdate, 'Noua', 73, 37, 42
1150 from dual;
1151
```

```
Script Output X | Query Result X
| Task completed in 0.074 seconds
```

```
Error at Command Line : 1,149 Column : 8
Error report -
SQL Error: ORA-04091: table PROIECT.COMENZI is mutating, trigger/function may not
ORA-06512: at "PROIECT.CHECK_COMENZI", line 4
ORA-04088: error during execution of trigger 'PROIECT.CHECK_COMENZI'
04091. 00000 - "table %s.%s is mutating, trigger/function may not see it"
*Cause: A trigger (or a user defined plsql function that is referenced in
this statement) attempted to look at (or modify) a table that was
in the middle of being modified by the statement which fired it.
*Action: Rewrite the trigger (or function) so it does not read that table.
```

```
-- rezolvarea erorii de tabel mutating
create or replace package pachet_check_comenzi
as
    type com_rec_pachet is record ( id_client clienti.id_client%type,
                                    numar_comenzi number(1));
    type com_idx is table of com_rec_pachet index by pls_integer;
    comenzi com_idx;
    contor number(1) :=0;
end;
/
create or replace trigger check_comenzi_comanda
    before insert or update of id_client on comenzi
begin
    pachet_check_comenzi.contor := 0;

    select id_client, count(*) bulk collect into pachet_check_comenzi.comenzi
    from comenzi
    group by id_client;
end;
/
create or replace trigger check_comenzi_linie
    before insert or update of id_client on comenzi
    for each row
begin
    for i in 1..pachet_check_comenzi.comenzi.last loop
        if pachet_check_comenzi.comenzi(i).numar_comenzi +
            pachet_check_comenzi.contor = 2
            and pachet_check_comenzi.comenzi(i).id_client = :new.id_client
        then
            raise_application_error(-20005,'Clientul are deja 2 comenzi in desfasurare!');
        end if;
    end loop;
    pachet_check_comenzi.contor := pachet_check_comenzi.contor + 1;
end;
/
alter trigger proiect.check_comenzi disable;

-- acum se declanseaza triggerul
insert into comenzi
select secventa_techflow.nextval, sysdate, 'Noua', 73, 37, 42
from dual;
```



```

1194 | -- acum se declanseaza triggerul
1195 | insert into comenzi
1196 | select secventa_techflow.nextval, sysdate, 'Noua', 73, 37, 42
1197 | from dual;

```

Script Output x | Query Result x

Task completed in 0.037 seconds

```

Error starting at line : 1,195 in command -
insert into comenzi
select secventa_techflow.nextval, sysdate, 'Noua', 73, 37, 42
from dual
Error report -
ORA-20005: Clientul are deja 2 comenzi in desfasurare!
ORA-06512: at "PROIECT.CHECK_COMENZI_LINIE", line 7
ORA-04088: error during execution of trigger 'PROIECT.CHECK_COMENZI_LINIE'

```

Cerință 12.

Definiți un trigger de tip LDD. Declanșați trigger-ul.

Trigger LDD care blochează adăugarea/stergerea/modificarea tabelelor în zilele de sămbată și duminică. Dacă se realizează o operatie atunci informațiile sunt inserate în tabelul audit_user

```

create table audit_user
(
    nume_bd varchar2(50),
    user_logat varchar2(30),
    eveniment varchar2(20),
    tip_object_referit varchar2(30),
    nume_object_referit varchar2(30),
    data timestamp(3));

```

```

create or replace trigger check_schema
    after create or drop or alter on schema
begin
    if to_char(sysdate,'day') = 'sunday' or to_char(sysdate,'day') = 'saturday'
    then
        raise_application_error(-20000,'Nu se pot face operatii LDD in weekend!');
    end if;

```

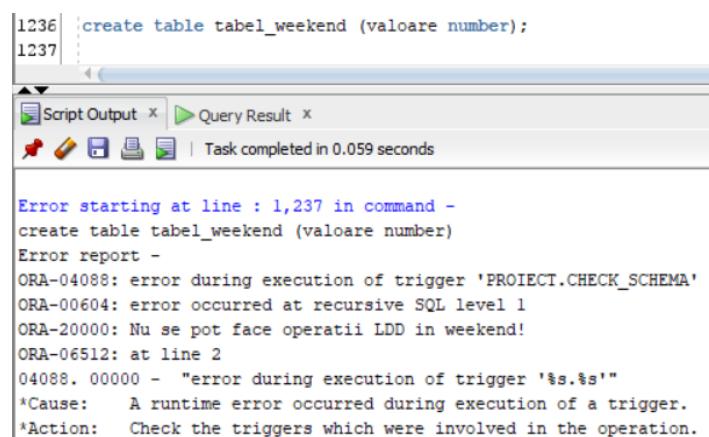
```

    insert into audit_user
    values (sys.database_name, sys.login_user,

```

```
sys.sysevent, sys.dictionary_obj_type,  
sys.dictionary_obj_name, systimestamp(3));  
end;  
/  
  
create table tabel_test (valoare number);  
  
alter table tabel_test add (alta_valoare number);  
  
drop table tabel_test;  
  
select * from audit_user;
```

NUME_BD	USER_LOGAT	EVENIMENT	TIP_OBIECT_REFERIT	NUME_OBIECT_REFERIT	DATA
1 ORCLPDB1	PROIECT	CREATE	TABLE	TABEL_TEST	08-JAN-24 05.17.55.420000000 PM
2 ORCLPDB1	PROIECT	ALTER	TABLE	TABEL_TEST	08-JAN-24 05.18.19.748000000 PM
3 ORCLPDB1	PROIECT	DROP	TABLE	TABEL_TEST	08-JAN-24 05.19.22.059000000 PM



The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there is a script editor window containing two lines of SQL code:

```
1236 | create table tabel_weekend (valoare number);
1237 |
```

In the bottom-right pane, there is a "Query Result" tab showing the output of the executed query:

```
Error starting at line : 1,237 in command -
create table tabel_weekend (valoare number)
Error report -
ORA-04088: error during execution of trigger 'PROIECT.CHECK_SCHEMA'
ORA-00604: error occurred at recursive SQL level 1
ORA-20000: Nu se pot face operatii LDD in weekend!
ORA-06512: at line 2
04088. 00000 -  "error during execution of trigger '%s.%s'"
*Cause: A runtime error occurred during execution of a trigger.
*Action: Check the triggers which were involved in the operation.
```

Cerință 13.

Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

```
1247 type t_id_com is table of number;
1248 function filtrare_comenzi
1249   (v_data_min date, v_data_max date,
1250    v_transport_maxim transport.cost_transport%type, v_cost_total plati.total_plata%type)
1251   return t_id_com;
1252   --ex 9
1253   procedure clienti_info(v_numa_client clienti.nume_client%type);
1254 end pachet_proiect;
1255 /
1256 |
1257 create or replace package body pachet_proiect as
1258   --ex 6
1259   procedure produse_info as
1260     -- varray de record-uri pentru produse
1261     type prod_rec is record (
1262       cod_prod produse.id_produs%type,
1263       nume_produse.nume_produs%type,
1264       pret_produse.pret_produs%type,
1265       stoc_produse.stoc_produs%type
1266     );
1267     type prod_vec is varray(5) of prod_rec;
1268     t_prod prod_vec := prod_vec();
1269
1270     -- tablou indexat pentru magazine
1271     type mag_rec is record (
1272       id_magazine mag.magazine%type,
1273       nume_magazine mag.nume_magazine%type,
1274       cantitate_magazine mag.cantitate_magazine%type
1275     );
1276     type mag_vec is varray(5) of mag_rec;
1277     t_mag mag_vec := mag_vec();
1278
1279   begin
1280     for i in 1..5 loop
1281       for j in 1..5 loop
1282         if t_prod(i).cod_prod = t_mag(j).id_magazine then
1283           t_mag(j).cantitate_magazine := t_mag(j).cantitate_magazine + t_prod(i).stoc_produs;
1284         end if;
1285       end loop;
1286     end loop;
1287   end;
1288
1289   procedure comenzi_info as
1290   begin
1291   end;
1292
1293   procedure clienti_info as
1294   begin
1295   end;
1296
1297   procedure produse_info as
1298   begin
1299   end;
1300
1301   procedure comenzi_info as
1302   begin
1303   end;
1304
1305   procedure filtrare_comenzi as
1306   begin
1307   end;
1308
1309   procedure clienti_info as
1310   begin
1311   end;
1312
1313   procedure produse_info as
1314   begin
1315   end;
1316
1317   procedure comenzi_info as
1318   begin
1319   end;
1320
1321   procedure clienti_info as
1322   begin
1323   end;
1324
1325   procedure produse_info as
1326   begin
1327   end;
1328
1329   procedure comenzi_info as
1330   begin
1331   end;
1332
1333   procedure filtrare_comenzi as
1334   begin
1335   end;
1336
1337   procedure clienti_info as
1338   begin
1339   end;
1340
1341   procedure produse_info as
1342   begin
1343   end;
1344
1345   procedure comenzi_info as
1346   begin
1347   end;
1348
1349   procedure clienti_info as
1350   begin
1351   end;
1352
1353   procedure produse_info as
1354   begin
1355   end;
1356
1357   procedure comenzi_info as
1358   begin
1359   end;
1360
1361   procedure filtrare_comenzi as
1362   begin
1363   end;
1364
1365   procedure clienti_info as
1366   begin
1367   end;
1368
1369   procedure produse_info as
1370   begin
1371   end;
1372
1373   procedure comenzi_info as
1374   begin
1375   end;
1376
1377   procedure clienti_info as
1378   begin
1379   end;
1380
1381   procedure produse_info as
1382   begin
1383   end;
1384
1385   procedure comenzi_info as
1386   begin
1387   end;
1388
1389   procedure filtrare_comenzi as
1390   begin
1391   end;
1392
1393   procedure clienti_info as
1394   begin
1395   end;
1396
1397   procedure produse_info as
1398   begin
1399   end;
1400
1401   procedure comenzi_info as
1402   begin
1403   end;
1404
1405   procedure clienti_info as
1406   begin
1407   end;
1408
1409   procedure produse_info as
1410   begin
1411   end;
1412
1413   procedure comenzi_info as
1414   begin
1415   end;
1416
1417   procedure filtrare_comenzi as
1418   begin
1419   end;
1420
1421   procedure clienti_info as
1422   begin
1423   end;
1424
1425   procedure produse_info as
1426   begin
1427   end;
1428
1429   procedure comenzi_info as
1430   begin
1431   end;
1432
1433   procedure clienti_info as
1434   begin
1435   end;
1436
1437   procedure produse_info as
1438   begin
1439   end;
1440
1441   procedure comenzi_info as
1442   begin
1443   end;
1444
1445   procedure filtrare_comenzi as
1446   begin
1447   end;
1448
1449   procedure clienti_info as
1450   begin
1451   end;
1452
1453   procedure produse_info as
1454   begin
1455   end;
1456
1457   procedure comenzi_info as
1458   begin
1459   end;
1460
1461   procedure clienti_info as
1462   begin
1463   end;
1464
1465   procedure produse_info as
1466   begin
1467   end;
1468
1469   procedure comenzi_info as
1470   begin
1471   end;
1472
1473   procedure filtrare_comenzi as
1474   begin
1475   end;
1476
1477   procedure clienti_info as
1478   begin
1479   end;
1480
1481   procedure produse_info as
1482   begin
1483   end;
1484
1485   procedure comenzi_info as
1486   begin
1487   end;
1488
1489   procedure clienti_info as
1490   begin
1491   end;
1492
1493   procedure produse_info as
1494   begin
1495   end;
1496
1497   procedure comenzi_info as
1498   begin
1499   end;
1500
1501   procedure filtrare_comenzi as
1502   begin
1503   end;
1504
1505   procedure clienti_info as
1506   begin
1507   end;
1508
1509   procedure produse_info as
1510   begin
1511   end;
1512
1513   procedure comenzi_info as
1514   begin
1515   end;
1516
1517   procedure clienti_info as
1518   begin
1519   end;
1520
1521   procedure produse_info as
1522   begin
1523   end;
1524
1525   procedure comenzi_info as
1526   begin
1527   end;
1528
1529   procedure filtrare_comenzi as
1530   begin
1531   end;
1532
1533   procedure clienti_info as
1534   begin
1535   end;
1536
1537   procedure produse_info as
1538   begin
1539   end;
1540
1541   procedure comenzi_info as
1542   begin
1543   end;
1544
1545   procedure clienti_info as
1546   begin
1547   end;
1548
1549   procedure produse_info as
1550   begin
1551   end;
1552
1553   procedure comenzi_info as
1554   begin
1555   end;
1556
1557   procedure filtrare_comenzi as
1558   begin
1559   end;
1560
1561   procedure clienti_info as
1562   begin
1563   end;
1564
1565   procedure produse_info as
1566   begin
1567   end;
1568
1569   procedure comenzi_info as
1570   begin
1571   end;
1572
1573   procedure clienti_info as
1574   begin
1575   end;
1576
1577   procedure produse_info as
1578   begin
1579   end;
1580
1581   procedure comenzi_info as
1582   begin
1583   end;
1584
1585   procedure filtrare_comenzi as
1586   begin
1587   end;
1588
1589   procedure clienti_info as
1590   begin
1591   end;
1592
1593   procedure produse_info as
1594   begin
1595   end;
1596
1597   procedure comenzi_info as
1598   begin
1599   end;
1600
1601   procedure clienti_info as
1602   begin
1603   end;
1604
1605   procedure produse_info as
1606   begin
1607   end;
1608
1609   procedure comenzi_info as
1610   begin
1611   end;
1612
1613   procedure filtrare_comenzi as
1614   begin
1615   end;
1616
1617   procedure clienti_info as
1618   begin
1619   end;
1620
1621   procedure produse_info as
1622   begin
1623   end;
1624
1625   procedure comenzi_info as
1626   begin
1627   end;
1628
1629   procedure clienti_info as
1630   begin
1631   end;
1632
1633   procedure produse_info as
1634   begin
1635   end;
1636
1637   procedure comenzi_info as
1638   begin
1639   end;
1640
1641   procedure filtrare_comenzi as
1642   begin
1643   end;
1644
1645   procedure clienti_info as
1646   begin
1647   end;
1648
1649   procedure produse_info as
1650   begin
1651   end;
1652
1653   procedure comenzi_info as
1654   begin
1655   end;
1656
1657   procedure clienti_info as
1658   begin
1659   end;
1660
1661   procedure produse_info as
1662   begin
1663   end;
1664
1665   procedure comenzi_info as
1666   begin
1667   end;
1668
1669   procedure filtrare_comenzi as
1670   begin
1671   end;
1672
1673   procedure clienti_info as
1674   begin
1675   end;
1676
1677   procedure produse_info as
1678   begin
1679   end;
1680
1681   procedure comenzi_info as
1682   begin
1683   end;
1684
1685   procedure clienti_info as
1686   begin
1687   end;
1688
1689   procedure produse_info as
1690   begin
1691   end;
1692
1693   procedure comenzi_info as
1694   begin
1695   end;
1696
1697   procedure filtrare_comenzi as
1698   begin
1699   end;
1700
1701   procedure clienti_info as
1702   begin
1703   end;
1704
1705   procedure produse_info as
1706   begin
1707   end;
1708
1709   procedure comenzi_info as
1710   begin
1711   end;
1712
1713   procedure clienti_info as
1714   begin
1715   end;
1716
1717   procedure produse_info as
1718   begin
1719   end;
1720
1721   procedure comenzi_info as
1722   begin
1723   end;
1724
1725   procedure filtrare_comenzi as
1726   begin
1727   end;
1728
1729   procedure clienti_info as
1730   begin
1731   end;
1732
1733   procedure produse_info as
1734   begin
1735   end;
1736
1737   procedure comenzi_info as
1738   begin
1739   end;
1740
1741   procedure clienti_info as
1742   begin
1743   end;
1744
1745   procedure produse_info as
1746   begin
1747   end;
1748
1749   procedure comenzi_info as
1750   begin
1751   end;
1752
1753   procedure filtrare_comenzi as
1754   begin
1755   end;
1756
1757   procedure clienti_info as
1758   begin
1759   end;
1760
1761   procedure produse_info as
1762   begin
1763   end;
1764
1765   procedure comenzi_info as
1766   begin
1767   end;
1768
1769   procedure clienti_info as
1770   begin
1771   end;
1772
1773   procedure produse_info as
1774   begin
1775   end;
1776
1777   procedure comenzi_info as
1778   begin
1779   end;
1780
1781   procedure filtrare_comenzi as
1782   begin
1783   end;
1784
1785   procedure clienti_info as
1786   begin
1787   end;
1788
1789   procedure produse_info as
1790   begin
1791   end;
1792
1793   procedure comenzi_info as
1794   begin
1795   end;
1796
1797   procedure clienti_info as
1798   begin
1799   end;
1800
1801   procedure produse_info as
1802   begin
1803   end;
1804
1805   procedure comenzi_info as
1806   begin
1807   end;
1808
1809   procedure filtrare_comenzi as
1810   begin
1811   end;
1812
1813   procedure clienti_info as
1814   begin
1815   end;
1816
1817   procedure produse_info as
1818   begin
1819   end;
1820
1821   procedure comenzi_info as
1822   begin
1823   end;
1824
1825   procedure clienti_info as
1826   begin
1827   end;
1828
1829   procedure produse_info as
1830   begin
1831   end;
1832
1833   procedure comenzi_info as
1834   begin
1835   end;
1836
1837   procedure filtrare_comenzi as
1838   begin
1839   end;
1840
1841   procedure clienti_info as
1842   begin
1843   end;
1844
1845   procedure produse_info as
1846   begin
1847   end;
1848
1849   procedure comenzi_info as
1850   begin
1851   end;
1852
1853   procedure clienti_info as
1854   begin
1855   end;
1856
1857   procedure produse_info as
1858   begin
1859   end;
1860
1861   procedure comenzi_info as
1862   begin
1863   end;
1864
1865   procedure filtrare_comenzi as
1866   begin
1867   end;
1868
1869   procedure clienti_info as
1870   begin
1871   end;
1872
1873   procedure produse_info as
1874   begin
1875   end;
1876
1877   procedure comenzi_info as
1878   begin
1879   end;
1880
1881   procedure clienti_info as
1882   begin
1883   end;
1884
1885   procedure produse_info as
1886   begin
1887   end;
1888
1889   procedure comenzi_info as
1890   begin
1891   end;
1892
1893   procedure filtrare_comenzi as
1894   begin
1895   end;
1896
1897   procedure clienti_info as
1898   begin
1899   end;
1900
1901   procedure produse_info as
1902   begin
1903   end;
1904
1905   procedure comenzi_info as
1906   begin
1907   end;
1908
1909   procedure clienti_info as
1910   begin
1911   end;
1912
1913   procedure produse_info as
1914   begin
1915   end;
1916
1917   procedure comenzi_info as
1918   begin
1919   end;
1920
1921   procedure filtrare_comenzi as
1922   begin
1923   end;
1924
1925   procedure clienti_info as
1926   begin
1927   end;
1928
1929   procedure produse_info as
1930   begin
1931   end;
1932
1933   procedure comenzi_info as
1934   begin
1935   end;
1936
1937   procedure clienti_info as
1938   begin
1939   end;
1940
1941   procedure produse_info as
1942   begin
1943   end;
1944
1945   procedure comenzi_info as
1946   begin
1947   end;
1948
1949   procedure filtrare_comenzi as
1950   begin
1951   end;
1952
1953   procedure clienti_info as
1954   begin
1955   end;
1956
1957   procedure produse_info as
1958   begin
1959   end;
1960
1961   procedure comenzi_info as
1962   begin
1963   end;
1964
1965   procedure clienti_info as
1966   begin
1967   end;
1968
1969   procedure produse_info as
1970   begin
1971   end;
1972
1973   procedure comenzi_info as
1974   begin
1975   end;
1976
1977   procedure filtrare_comenzi as
1978   begin
1979   end;
1980
1981   procedure clienti_info as
1982   begin
1983   end;
1984
1985   procedure produse_info as
1986   begin
1987   end;
1988
1989   procedure comenzi_info as
1990   begin
1991   end;
1992
1993   procedure clienti_info as
1994   begin
1995   end;
1996
1997   procedure produse_info as
1998   begin
1999   end;
1999
2000   procedure comenzi_info as
2001   begin
2002   end;
2003
2004   procedure filtrare_comenzi as
2005   begin
2006   end;
2007
2008   procedure clienti_info as
2009   begin
2010   end;
2011
2012   procedure produse_info as
2013   begin
2014   end;
2015
2016   procedure comenzi_info as
2017   begin
2018   end;
2019
2020   procedure clienti_info as
2021   begin
2022   end;
2023
2024   procedure produse_info as
2025   begin
2026   end;
2027
2028   procedure comenzi_info as
2029   begin
2030   end;
2031
2032   procedure filtrare_comenzi as
2033   begin
2034   end;
2035
2036   procedure clienti_info as
2037   begin
2038   end;
2039
2040   procedure produse_info as
2041   begin
2042   end;
2043
2044   procedure comenzi_info as
2045   begin
2046   end;
2047
2048   procedure clienti_info as
2049   begin
2050   end;
2051
2052   procedure produse_info as
2053   begin
2054   end;
2055
2056   procedure comenzi_info as
2057   begin
2058   end;
2059
2060   procedure filtrare_comenzi as
2061   begin
2062   end;
2063
2064   procedure clienti_info as
2065   begin
2066   end;
2067
2068   procedure produse_info as
2069   begin
2070   end;
2071
2072   procedure comenzi_info as
2073   begin
2074   end;
2075
2076   procedure clienti_info as
2077   begin
2078   end;
2079
2080   procedure produse_info as
2081   begin
2082   end;
2083
2084   procedure comenzi_info as
2085   begin
2086   end;
2087
2088   procedure filtrare_comenzi as
2089   begin
2090   end;
2091
2092   procedure clienti_info as
2093   begin
2094   end;
2095
2096   procedure produse_info as
2097   begin
2098   end;
2099
2100   procedure comenzi_info as
2101   begin
2102   end;
2103
2104   procedure clienti_info as
2105   begin
2106   end;
2107
2108   procedure produse_info as
2109   begin
2110   end;
2111
2112   procedure comenzi_info as
2113   begin
2114   end;
2115
2116   procedure filtrare_comenzi as
2117   begin
2118   end;
2119
2120   procedure clienti_info as
2121   begin
2122   end;
2123
2124   procedure produse_info as
2125   begin
2126   end;
2127
2128   procedure comenzi_info as
2129   begin
2130   end;
2131
2132   procedure clienti_info as
2133   begin
2134   end;
2135
2136   procedure produse_info as
2137   begin
2138   end;
2139
2140   procedure comenzi_info as
2141   begin
2142   end;
2143
2144   procedure filtrare_comenzi as
2145   begin
2146   end;
2147
2148   procedure clienti_info as
2149   begin
2150   end;
2151
2152   procedure produse_info as
2153   begin
2154   end;
2155
2156   procedure comenzi_info as
2157   begin
2158   end;
2159
2160   procedure filtrare_comenzi as
2161   begin
2162   end;
2163
2164   procedure clienti_info as
2165   begin
2166   end;
2167
2168   procedure produse_info as
2169   begin
2170   end;
2171
2172   procedure comenzi_info as
2173   begin
2174   end;
2175
2176   procedure filtrare_comenzi as
2177   begin
2178   end;
2179
2180   procedure clienti_info as
2181   begin
2182   end;
2183
2184   procedure produse_info as
2185   begin
2186   end;
2187
2188   procedure comenzi_info as
2189   begin
2190   end;
2191
2192   procedure clienti_info as
2193   begin
2194   end;
2195
2196   procedure produse_info as
2197   begin
2198   end;
2199
2200   procedure comenzi_info as
2201   begin
2202   end;
2203
2204   procedure filtrare_comenzi as
2205   begin
2206   end;
2207
2208   procedure clienti_info as
2209   begin
2210   end;
2211
2212   procedure produse_info as
2213   begin
2214   end;
2215
2216   procedure comenzi_info as
2217   begin
2218   end;
2219
2220   procedure clienti_info as
2221   begin
2222   end;
2223
2224   procedure produse_info as
2225   begin
2226   end;
2227
2228   procedure comenzi_info as
2229   begin
2230   end;
2231
2232   procedure filtrare_comenzi as
2233   begin
2234   end;
2235
2236   procedure clienti_info as
2237   begin
2238   end;
2239
2240   procedure produse_info as
2241   begin
2242   end;
2243
2244   procedure comenzi_info as
2245   begin
2246   end;
2247
2248   procedure clienti_info as
2249   begin
2250   end;
2251
2252   procedure produse_info as
2253   begin
2254   end;
2255
2256   procedure comenzi_info as
2257   begin
2258   end;
2259
2260   procedure filtrare_comenzi as
2261   begin
2262   end;
2263
2264   procedure clienti_info as
2265   begin
2266   end;
2267
2268   procedure produse_info as
2269   begin
2270   end;
2271
2272   procedure comenzi_info as
2273   begin
2274   end;
2275
2276   procedure clienti_info as
2277   begin
2278   end;
2279
2280   procedure produse_info as
2281   begin
2282   end;
2283
2284   procedure comenzi_info as
2285   begin
2286   end;
2287
2288   procedure filtrare_comenzi as
2289   begin
2290   end;
2291
2292   procedure clienti_info as
2293   begin
2294   end;
2295
2296   procedure produse_info as
2297   begin
2298   end;
2299
2300   procedure comenzi_info as
2301   begin
2302   end;
2303
2304   procedure clienti_info as
2305   begin
2306   end;
2307
2308   procedure produse_info as
2309   begin
2310   end;
2311
2312   procedure comenzi_info as
2313   begin
2314   end;
2315
2316   procedure filtrare_comenzi as
2317   begin
2318   end;
2319
2320   procedure clienti_info as
2321   begin
2322   end;
2323
2324   procedure produse_info as
2325   begin
2326   end;
2327
2328   procedure comenzi_info as
2329   begin
2330   end;
2331
2332   procedure clienti_info as
2333   begin
2334   end;
2335
2336   procedure produse_info as
2337   begin
2338   end;
2339
2340   procedure comenzi_info as
2341   begin
2342   end;
2343
2344   procedure filtrare_comenzi as
2345   begin
2346   end;
2347
2348   procedure clienti_info as
2349   begin
2350   end;
2351
2352   procedure produse_info as
2353   begin
2354   end;
2355
2356   procedure comenzi_info as
2357   begin
2358   end;
2359
2360   procedure clienti_info as
2361   begin
2362   end;
2363
2364   procedure produse_info as
2365   begin
2366   end;
2367
2368   procedure comenzi_info as
2369   begin
2370   end;
2371
2372   procedure filtrare_comenzi as
2373   begin
2374   end;
2375
2376   procedure clienti_info as
2377   begin
2378   end;
2379
2380   procedure produse_info as
2381   begin
2382   end;
2383
2384   procedure comenzi_info as
2385   begin
2386   end;
2387
2388   procedure clienti_info as
2389   begin
2390   end;
2391
2392   procedure produse_info as
2393   begin
2394   end;
2395
2396   procedure comenzi_info as
2397   begin
2398   end;
2399
2400   procedure filtrare_comenzi as
2401   begin
2402   end;
2403
2404   procedure clienti_info as
2405   begin
2406   end;
2407
2408   procedure produse_info as
2409   begin
2410   end;
2411
2412   procedure comenzi_info as
2413   begin
2414   end;
2415
2416   procedure clienti_info as
2417   begin
2418   end;
2419
2420   procedure produse_info as
2421   begin
2422   end;
2423
2424   procedure comenzi_info as
2425   begin
2426   end;
2427
2428   procedure filtrare_comenzi as
2429   begin
2430   end;
2431
2432   procedure clienti_info as
2433   begin
2434   end;
2435
2436   procedure produse_info as
2437   begin
2438   end;
2439
2440   procedure comenzi_info as
2441   begin
2442   end;
2443
2444   procedure clienti_info as
2445   begin
2446   end;
2447
2448   procedure produse_info as
2449   begin
2450   end;
2451
2452   procedure comenzi_info as
2453   begin
2454   end;
2455
2456   procedure filtrare_comenzi as
2457   begin
2458   end;
2459
2460   procedure clienti_info as
2461   begin
2462   end;
2463
2464   procedure produse_info as
2465   begin
2466   end;
2467
2468   procedure comenzi_info as
2469   begin
2470   end;
2471
2472   procedure clienti_info as
2473   begin
2474   end;
2475
2476   procedure produse_info as
2477   begin
2478   end;
2479
2480   procedure comenzi_info as
2481   begin
2482   end;
2483
2484   procedure filtrare_comenzi as
2485   begin
2486   end;
2487
2488   procedure clienti_info as
2489   begin
2490   end;
2491
2492   procedure produse_info as
2493   begin
2494   end;
2495
2496   procedure comenzi_info as
2497   begin
2498   end;
2499
2500   procedure clienti_info as
2501   begin
2502   end;
2503
2504   procedure produse_info as
2505   begin
2506   end;
2507
2508   procedure comenzi_info as
2509   begin
2510   end;
2511
2512   procedure filtrare_comenzi as
2513   begin
2514   end;
2515
2516   procedure clienti_info as
2517   begin
2518   end;
2519
2520   procedure produse_info as
2521   begin
2522   end;
2523
2524   procedure comenzi_info as
2525   begin
2526   end;
2527
2528   procedure clienti_info as
2529   begin
2530   end;
2531
2532   procedure produse_info as
2533   begin
2534   end;
2535
2536   procedure comenzi_info as
2537   begin
2538   end;
2539
2540   procedure filtrare_comenzi as
2541   begin
2542   end;
2543
2544   procedure clienti_info as
2545   begin
2546   end;
2547
2548   procedure produse_info as
2549   begin
2550   end;
2551
2552   procedure comenzi_info as
2553   begin
2554   end;
2555
2556   procedure clienti_info as
2557   begin
2558   end;
2559
2560   procedure produse_info as
2561   begin
2562   end;
2563
2564   procedure comenzi_info as
2565   begin
2566   end;
2567
2568   procedure filtrare_comenzi as
2569   begin
2570   end;
2571
2572   procedure clienti_info as
2573   begin
2574   end;
2575
2576   procedure produse_info as
2577   begin
2578   end;
2579
2580   procedure comenzi_info as
2581   begin
2582   end;
2583
2584   procedure clienti_info as
2585   begin
2586   end;
2587
2588   procedure produse_info as
2589   begin
2590   end;
2591
2592   procedure comenzi_info as
2593   begin
2594   end;
2595
2596   procedure filtrare_comenzi as
2597   begin
2598   end;
2599
2600   procedure clienti_info as
2601   begin
2602   end;
2603
2604   procedure produse_info as
2605   begin
2606   end;
2607
2608   procedure comenzi_info as
2609   begin
2610   end;
2611
2612   procedure clienti_info as
2613   begin
2614   end;
2615
2616   procedure produse_info as
2617   begin
2618   end;
2619
2620   procedure comenzi_info as
2621   begin
2622   end;
2623
2624   procedure filtrare_comenzi as
2625   begin
2626   end;
2627
2628   procedure clienti_info as
2629   begin
2630   end;
2631
2632   procedure produse_info as
2633   begin
2634   end;
2635
2636   procedure comenzi_info as
2637   begin
2638   end;
2639
2640   procedure clienti_info as
2641   begin
2642   end;
2643
2644   procedure produse_info as
2645   begin
2646   end;
2647
2648   procedure comenzi_info as
2649   begin
2650   end;
2651
2652   procedure filtrare_comenzi as
2653   begin
2654   end;
2655
2656   procedure clienti_info as
2657   begin
2658   end;
2659
2660   procedure produse_info as
2661   begin
2662   end;
2663
2664   procedure comenzi_info as
2665   begin
2666   end;
2667
2668   procedure clienti_info as
2669   begin
2670   end;
2671
2672   procedure produse_info as
2673   begin
2674   end;
2675
2676   procedure comenzi_info as
2677   begin
2678   end;
2679
2680   procedure filtrare_comenzi as
2681   begin
2682   end;
2683
268
```

```
--ex 7
execute pachet_proiect.comenzi_info();

--ex 8
-- caz 1 apel corect
declare
    solutie pachet_proiect.t_id_com;
```

Informatii despre categorii - produse - comenzi
Categorie: Placi de baza cu id-ul: 21

Produs 1. Produs Test1

Nu are comenzi!

```
solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 3000');
    dbms_output.put_line('si costul de transport mai mic de -10.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date
exception
    when others then
        dbms_output.put_line('Eroare cu codul '||SQLCODE|| ' si mesajul '|| SQLERRM);
end;
/

-- caz5 exceptie cost_total_eroare
-- caz in care costul total maximal este negativ
declare
    solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de -50');
    dbms_output.put_line('si costul de transport mai mic de 200.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date
exception
    when others then
        dbms_output.put_line('Eroare cu codul '||SQLCODE|| ' si mesajul '|| SQLERRM);
end;
```

Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 5000
si costul de transport mai mic de 190.
Id-uri comenzi care respecta criteriul:
51
52
Comenzi intre 10-15 mai 2022 cu totalul de plata mai mic de 3000
si costul de transport mai mic de 100.
Eroare cu codul -20000 si mesajul ORA-20000: Nu exista comanda cu aceste conditii!
Comenzi intre 15-10 mai 2022 cu totalul de plata mai mic de 3000
si costul de transport mai mic de 100.
Eroare cu codul -20001 si mesajul ORA-20001: Intervalul nu este introdus corect!
Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 3000
si costul de transport mai mic de -10.
Eroare cu codul -20002 si mesajul ORA-20002: Costul transportului nu poate fi negativ
Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de -50
si costul de transport mai mic de 200.
Eroare cu codul -20003 si mesajul ORA-20003: Costul total nu poate fi negativ!

```
-- ex 9
-- caz 1 apel corect
begin
    pachet_proiect.clienti_info('Avram');
exception
    when others then
        dbms_output.put_line('Eroare cu codul '||SQLCODE|| ' si mesajul '|| SQLERRM);
end;
/

-- caz 2 no_data_found (nu exista client cu acest nume)
begin
    -- code here
end;
```

Informatii client:
Nume: Avram
Prenume: Gabriela
Numar recenzii: 1
Numar produse wishlist: 1
Numar comenzi: 3
Numar produse din comenziile neanulate: 2
Eroare cu codul -20000 si mesajul ORA-20000: Nu exista client cu acest nume!
Eroare cu codul -20001 si mesajul ORA-20001: Prea multi clienti cu acest string!

```
create or replace package pachet_proiect as
    --ex 6
    procedure produse_info;
    --ex 7
    procedure comenzi_info;
    --ex 8
    type t_id_com is table of number;
    function filtrare_comenzi
        (v_data_min date, v_data_max date,
        v_transport_maxim transport.cost_transport%type, v_cost_total plati.total_plata%type)
        return t_id_com;
    --ex 9
    procedure clienti_info(v_nume_client clienti.nume_client%type);
end pachet_proiect;
/
create or replace package body pachet_proiect as
    --ex 6
    procedure produse_info as
        -- varray de record-uri pentru produse
        type prod_rec is record (
            cod_prod produse.id_produs%type,
            nume_produse.nume_produs%type,
            pret_produse.pret_produs%type,
            stoc_produse.stoc_produs%type
        );
        type prod_vec is varray(5) of prod_rec;
        t_prod prod_vec := prod_vec();
        -- tablou indexat pentru magazine
        type mag_rec is record (
            nume_magazine.nume_magazin%type,
            adresa_magazine.adresa_magazin%type
        );
        type mag_ind is table of mag_rec index by binary_integer;
        t_mag mag_ind;
        -- tablou imbricat pentru recenzii
        type recenzii_rec is record (
            nota_recenzii.nota_recenzie%type,
            nume_client.clienti.nume_client%type
        );
    end;
```

```
type rec_imbr is table of recenzii_rec;
t_recenzii rec_imbr := rec_imbr();
begin
    dbms_output.put_line('##### Informatii despre produse - magazine - recenzii
#####');
    dbms_output.new_line();
    -- atribuire produse
    select id_produs,nume_produs,pret_produs,stoc_produs bulk collect into t_prod
    from produse
    order by nume_produs
    fetch first 5 rows only;

    for i in t_prod.first..t_prod.last loop
        -- afisare info produs
        dbms_output.put_line('Produsul ' || i);
        dbms_output.put_line('#####');
        dbms_output.put_line('Id: ' || t_prod(i).cod_prod || ' Nume: ' || t_prod(i).nume);
        dbms_output.put_line('Pret: ' || t_prod(i).pret || ' Stoc: ' || t_prod(i).stoc);
        dbms_output.put_line('#####');

        -- atribuire magazine
        select m.nume_magazin,m.adresa_magazin bulk collect into t_mag
        from magazine m
        join produse_magazine pm on (m.id_magazin = pm.id_magazin)
        where pm.id_produs = t_prod(i).cod_prod;

        -- afisare info magazin
        if t_mag.count = 0 then
            dbms_output.put_line('Produsul nu se afla intr-un magazin!');
            dbms_output.new_line();
        else
            dbms_output.put_line('Magazine: ');
            for j in t_mag.first..t_mag.last loop
                dbms_output.put_line('Magazinul ' || j);
                dbms_output.put_line('Nume: ' || t_mag(j).nume || ' Adresa: ' || t_mag(j).adresa);
                dbms_output.new_line();
            end loop;
        end if;
        dbms_output.put_line('#####');

        -- atribuire recenzii
        select r.nota_recenzie,c.nume_client bulk collect into t_recenzii
```

```
from recenzi r
join clienti c on (r.id_client = c.id_client)
where r.id_produs = t_prod(i).cod_prod;

-- afisare recenzi
if t_recenzi.count = 0 then
    dbms_output.put_line('Produsul nu are recenzi!');
    dbms_output.new_line;
else
    dbms_output.put_line('Recenzi: ');
    for k in t_recenzi.first..t_recenzi.last loop
        dbms_output.put_line('Recenzia ' || k);
        dbms_output.put_line('Nota: ' || t_recenzi(k).nota || ' Nume Client: ' ||
t_recenzi(k).nume_client);
        dbms_output.new_line();
    end loop;
end if;
dbms_output.new_line();
end loop;
end produse_info;

--ex 7
procedure comenzi_info as

type com_rec is record (
    data_plasare comenzi.data_plasare%type,
    status_comanda comenzi.status_comanda%type,
    total_plata plati.total_plata%type,
    tip_plata plati.tip_plata%type
);

type prod_rec is record (
    nume_produse.nume_produs%type,
    id_produs produse.id_produs%type
);

type t_prod is table of prod_rec
index by binary_integer;
v_prod t_prod;

v_com com_rec;
```

```
-- cursor dinamic/ expresie cursor
-- pentru fiecare categorie se determina produsul cu cel mai mare stoc
cursor categorii_info is
    select id_categorie,nume_categorie,
        cursor(select max(nume_produs),max(id_produs)
            from produse p
            where c.id_categorie = p.id_categorie
            group by p.id_produs
            order by max(stoc_produs) desc)
    from categorii c;

-- cursor parametrizat pentru comenzi
cursor com_info(v_produs produse.id_produs%type) is
    select c.data_plasare,c.status_comanda,p.total_plata,p.tip_plata
    from comenzi c
    join plati p on (c.id_plata = p.id_plata)
    join produse_comenzi pc on (c.id_comanda = pc.id_comanda)
    where pc.id_produs = v_produs
    order by c.data_plasare desc;

v_cursor sys_refcursor;

v_id_categorie categorii.id_categorie%type;
v_nume_categorie categorii.nume_categorie%type;
contor number := 1;
begin
    dbms_output.put_line('##### Informatii despre categorii - produse - comenzi - plati');
    #####);
    open categorii_info;
    loop
        fetch categorii_info into v_id_categorie,v_nume_categorie,v_cursor;
        exit when categorii_info%notfound;
        dbms_output.put_line('Categoria: ' || v_nume_categorie || ' cu id-ul: ' || v_id_categorie);
        dbms_output.put_line('#####');
        fetch v_cursor bulk collect into v_prod limit 2;
        dbms_output.new_line;
        if v_prod.count = 0 then
            dbms_output.put_line('Nu exista produse!');
        else
            for i in v_prod.first..v_prod.last loop
```

```

dbms_output.put_line('Produs'||i||'.'|| v_prod(i).nume);
dbms_output.put_line('#####');
open com_info(v_prod(i).id_produs);
loop
    fetch com_info into v_com;
    exit when com_info%notfound;
    dbms_output.put_line('Comanda'||contor);
    dbms_output.put_line('Data Plasare: '||v_com.data_plasare||' Status Comanda: '||v_com.status_comanda);
    dbms_output.put_line('Total Plata: '||v_com.total_plata||' Tip Plata: '||v_com.tip_plata);
    dbms_output.new_line;
    contor := contor + 1;
end loop;

if com_info%rowcount = 0 then
    dbms_output.put_line('Nu are comenzi!');
end if;
close com_info;

contor := 1;
dbms_output.put_line('#####');
dbms_output.new_line;
end loop;
end if;

dbms_output.new_line;
end loop;
close categorii_info;
end comenzi_info;

--ex 8
function filtrare_comenzi
    (v_data_min date, v_data_max date, v_transport_maxim transport.cost_transport%type,
v_cost_total plati.total_plata%type)
return t_id_com is
    v_id_com t_id_com;

    data_eroare exception;
    transport_eroare exception;
    cost_total_eroare exception;
    no_comenzi exception;

```

```
begin

    if v_data_min > v_data_max then
        raise data_eroare;
    end if;

    if v_transport_maxim < 0 then
        raise transport_eroare;
    end if;

    if v_cost_total < 0 then
        raise cost_total_eroare;
    end if;

    -- folosirea a 3 tabele
    select distinct c.id_comanda bulk collect into v_id_com
    from comenzi c
    join plati p on (c.id_plata = p.id_plata)
    join transport t on (c.id_transport = t.id_transport)
    where c.data_plasare >= v_data_min and c.data_plasare <= v_data_max
    and t.cost_transport <= v_transport_maxim
    and p.total_plata <= v_cost_total;

    if v_id_com.count = 0 then
        raise no_comenzi;
    end if;
    return v_id_com;
exception
    when no_comenzi then
        raise_application_error(-20000,'Nu exista comanda cu aceste conditii!');
    when data_eroare then
        raise_application_error(-20001,'Intervalul nu este introdus corect!');
    when transport_eroare then
        raise_application_error(-20002,'Costul transportului nu poate fi negativ!');
    when cost_total_eroare then
        raise_application_error(-20003,'Costul total nu poate fi negativ!');
    when others then
        raise_application_error(-20004,'Alta eroare!');
end filtrare_comenzi;

--ex 9
procedure clienti_info(v_num_client clienti.nume_client%type)
```

```
is
type info_rec is record (
    nr_recenzi integer,
    nr_wishlist integer,
    nr_comenzi integer,
    nr_produse_neanulate integer,
    nume_client clienti.nume_client%type,
    prenume_client clienti.prenume_client%type
);
l_info info_rec;
type nume_client_imbr is table of clienti.nume_client%type;
t_nume_client nume_client_imbr := nume_client_imbr();
begin
select nume_client bulk collect into t_nume_client
from clienti
where nume_client like v_nume_client || '%';
if t_nume_client.count = 0 then
    raise no_data_found;
elsif t_nume_client.count > 1 then
    raise too_many_rows;
else
    dbms_output.put_line('Informatii client: ');
    dbms_output.new_line;
-- 6 tabele intr-o comanda sql
select
(select count(*) from recenzi r
join clienti cl on (cl.id_client = r.id_client)
where cl.nume_client like v_nume_client
)as nr_recenzi,
(select count(*) from wishlist w
join clienti cl on (cl.id_client = w.id_client)
where cl.nume_client like v_nume_client) as nr_wishlist,
(select count(*) from comenzi c
join clienti cl on (cl.id_client = c.id_client)
where cl.nume_client like v_nume_client)as nr_comenzi,
(select count(*) from produse p
join produse_comenzi pc on (pc.id_produs = p.id_produs)
join comenzi c on (c.id_comanda = pc.id_comanda)
join clienti cl on (cl.id_client = c.id_client)
where cl.nume_client like v_nume_client and c.status_comanda <> 'Anulata'
) as nr_produse_neanulate,
cl.nume_client,cl.prenume_client into l_info
```

```
from clienti cl
where cl.numere_client like v_numere_client
dbms_output.put_line('Nume: ' || l_info.numere_client);
dbms_output.put_line('Prenume: ' || l_info.prenume_client);
dbms_output.new_line;
dbms_output.put_line('Numar recenzii: ' || l_info.nr_recenzii);
dbms_output.put_line('Numar produse wishlist: ' || l_info.nr_wishlist);
dbms_output.put_line('Numar comenzi: ' || l_info.nr_comenzi);
dbms_output.put_line('Numar produse din comenziile neanulate: ' ||
l_info.nr_produse_neanulate);
end if;
exception
when no_data_found then
raise_application_error(-20000,'Nu exista client cu acest nume!');
when too_many_rows then
raise_application_error(-20001,'Prea multi clienti cu acest string!');
when others then
raise_application_error(-20002,'Alta eroare!');
end clienti_info;
end pachet_proiect;
/
--ex 6
execute pachet_proiect.produse_info();
--ex 7
execute pachet_proiect.comenzi_info();
--ex 8
-- caz 1 apel corect
declare
solutie pachet_proiect.t_id_com;
begin
dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 5000');
dbms_output.put_line('si costul de transport mai mic de 190.');
solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2023', 'dd-mon-yyyy'),to_date('15-may-2023', 'dd-mon-yyyy'),190,5000);
dbms_output.new_line;
dbms_output.put_line('Id-uri comenzi care respecta criteriul:');
for i in solutie.first..solutie.last loop
dbms_output.put_line(solutie(i)|| ' ');
end loop;
end;
/
-- caz2 exceptie no_comenzi
```

```
-- caz in care nu exista comenzi care sa respecte conditiile
declare
    solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2022 cu totalul de plata mai mic de 3000');
    dbms_output.put_line('si costul de transport mai mic de 100.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),100,3000);
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz3 exceptie data_eroare
-- caz in care data minima este mai mare decat data maxima
declare
    solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 15-10 mai 2022 cu totalul de plata mai mic de 3000');
    dbms_output.put_line('si costul de transport mai mic de 100.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('15-may-2022', 'dd-mon-yyyy'),to_date('10-may-2022', 'dd-mon-yyyy'),100,3000);
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz4 exceptie transport_eroare
-- caz in care costul pentru transport maximal este negativ
declare
    solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totalul de plata mai mic de 3000');
    dbms_output.put_line('si costul de transport mai mic de -10.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),-10,3000);
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
```

```
end;
/
-- caz5 exceptie cost_total_eroare
-- caz in care costul total maximal este negativ
declare
    solutie pachet_proiect.t_id_com;
begin
    dbms_output.put_line('Comenzi intre 10-15 mai 2023 cu totatul de plata mai mic de -50');
    dbms_output.put_line('si costul de transport mai mic de 200.');
    dbms_output.new_line;
    solutie := pachet_proiect.filtrare_comenzi(to_date('10-may-2022', 'dd-mon-yyyy'),to_date('15-may-2022', 'dd-mon-yyyy'),200,-50);
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- ex 9
-- caz 1 apel corect
begin
    pachet_proiect.clienti_info('Avram');
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz 2 no_data_found (nu exista cu client cu acest nume)
begin
    pachet_proiect.clienti_info('Huma');
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
-- caz 3 too_many_rows (pentru Pop => exista Popa si Popescu)
begin
    pachet_proiect.clienti_info('Pop');
exception
    when others then
        dbms_output.put_line('Eroare cu codul'||SQLCODE||' si mesajul '||SQLERRM);
end;
/
```

Cerința 14.

Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite.

Pachetul angajati_magazine se ocupa cu realizarea unui bilant (suma salariilor angajatilor) pe un anumit magazin sau pe toata compania (overload de functii si metode) pe un anumit numar de luni. De asemenea pachetul cuprinde proceduri standard care determina bilantul anual si semestrial.

- 4 functii, 4 proceduri, 2 cursoare

The screenshot shows the Oracle SQL Developer interface. In the main editor window, there is PL/SQL code for a package named PACHET_ANGAJATI_MAGAZINE. The code includes several procedures and functions, such as cursor ang_mag, function bilant_magazin, and a loop for calculating a sum. Below the code, the output pane shows the results of the compilation process:

```
1302     cursor ang_mag(v_num_magazin magazine.nume_magazin%type)
1303     return ang_rec
1304     is select a.id_angajat, a.num_angajat, a.prenume_angajat,a.salariu_angajat,a.job_ang
1305         from angajati a join magazine m on (a.id_magazin =m.id_magazin)
1306         where m.nume_magazin = v_num_magazin;
1307
1308     cursor nume_magazine return mag_rec
1309     is select id_magazin,nume_magazin,adresa_magazin
1310         from magazine;
1311
1312     function bilant_magazin(v_num_magazin magazine.nume_magazin%type,v_luni number) retu
1313     is suma number;
1314     begin
1315         suma := 0;
1316         open ang_mag(v_num_magazin);
1317         loop
1318             fetch ang_mag into rec_angajati;
1319             exit when ang_mag%notfound;
```

Script Output | Query Result | Task completed in 0.073 seconds

```
Package PACHET_ANGAJATI_MAGAZINE compiled

Package Body PACHET_ANGAJATI_MAGAZINE compiled

PL/SQL procedure successfully completed.
```

```
create or replace package pachet_angajati_magazine
as
-- record info angajati
type ang_rec is record
  (id_angajati.id_angajat%type,
  nume_angajati.nume_angajat%type,
  prenume_angajati.prenume_angajat%type,
  salariu_angajati.salariu_angajat%type,
  job_angajati.job_angajat%type);
-- record info magazine
type mag_rec is record
  (id_magazine.id_magazin%type,
  nume_magazine.nume_magazin%type,
  adresa_magazine.adresa_magazin%type
 );
-- tablou angajati
type t_ang is table of ang_rec
index by binary_integer;
vec_angajati t_ang;
rec_angajati ang_rec;
rec_magazine mag_rec;
-- cursor pentru angajati
cursor ang_mag(v_numemagazin magazine.nume_magazin%type) return ang_rec;
-- cursor magazine
cursor numemagazine return mag_rec;
-- returneaza suma salariilor angajatilor dintr-un magazin dat
-- pe un numar de luni dat
function bilant_magazin(v_numemagazin magazine.nume_magazin%type,v_luni number)
  return number;
-- returneaza suma salariilor angajatilor din toate magazinele
-- pe un numar de luni dat
function bilant_magazin(v_luni number)
  return number;
-- afiseaza suma salariilor angajatilor din toate magazinele
-- pe 12 luni
procedure afisare_bilant_anual;
```

```
-- afiseaza suma salariilor angajatilor dintr-un magazin dat
-- pe 12 luni
procedure afisare_bilant_anual(v_numemagazin magazine.nume_magazin%type);

-- afiseaza suma salariilor angajatilor din toate magazinele
-- pe 6 luni
procedure afisare_bilant_semestrial;

-- afiseaza suma salariilor angajatilor dintr-un magazin dat
-- pe 6 luni
procedure afisare_bilant_semestrial(v_numemagazin magazine.nume_magazin%type);

end pachet_angajati_magazine;
/
create or replace package body pachet_angajati_magazine
as
    cursor ang_mag(v_numemagazin magazine.nume_magazin%type)
    return ang_rec
    is select a.id_angajat, a.nume_angajat, a.prenume_angajat,a.salariu_angajat,a.job_angajat
        from angajati a join magazine m on (a.id_magazin =m.id_magazin)
        where m.nume_magazin = v_numemagazin;

    cursor nume_magazine return mag_rec
    is select id_magazin,nume_magazin,adresa_magazin
        from magazine;

    function bilant_magazin(v_numemagazin magazine.nume_magazin%type,v_luni number)
    return number
    is suma number;
    begin
        suma := 0;
        open ang_mag(v_numemagazin);
        loop
            fetch ang_mag into rec_angajati;
            exit when ang_mag%notfound;
            suma := suma + rec_angajati.salariu * v_luni;
        end loop;
        close ang_mag;
        return suma;
    end bilant_magazin;

    function bilant_magazin(v_luni number) return number
```

```
is suma number;
begin
    suma := 0;
    open nume_magazine;
    loop
        fetch nume_magazine into rec_magazine;
        exit when nume_magazine%notfound;
        suma := suma + bilant_magazin(rec_magazine.nume,v_luni);
    end loop;
    close nume_magazine;
    return suma;
end bilant_magazin;

procedure afisare_bilant_anual is
    bilant number := bilant_magazin(12);
begin
    dbms_output.put_line('Bilantul anual pentru Compania TechFlow este: ');
    dbms_output.put_line(bilant || '$');
end afisare_bilant_anual;

procedure afisare_bilant_semestrial is
    bilant number := bilant_magazin(6);
begin
    dbms_output.put_line('Bilantul semestrial pentru Compania TechFlow este: ');
    dbms_output.put_line(bilant || '$');
end afisare_bilant_semestrial;

procedure afisare_bilant_anual(v_nume_magazin magazine.nume_magazin%type) is
    bilant number := bilant_magazin(v_nume_magazin,12);
begin
    dbms_output.put_line('Bilantul anual pentru magazinul ' || v_nume_magazin || ' este: ');
    dbms_output.put_line(bilant || '$');
end afisare_bilant_anual;

procedure afisare_bilant_semestrial(v_nume_magazin magazine.nume_magazin%type) is
    bilant number := bilant_magazin(v_nume_magazin,6);
begin
    dbms_output.put_line('Bilantul semestrial pentru magazinul ' || v_nume_magazin || ' este: ');
    dbms_output.put_line(bilant || '$');
end afisare_bilant_semestrial;

end pachet_angajati_magazine;
```

```
/  
  
begin  
    pachet_angajati_magazine.afisare_bilant_anual();  
    pachet_angajati_magazine.afisare_bilant_semestrial();  
  
    for i in pachet_angajati_magazine.nume_magazine loop  
        pachet_angajati_magazine.afisare_bilant_anual(i.nume);  
        pachet_angajati_magazine.afisare_bilant_semestrial(i.nume);  
    end loop;  
end;
```

```
begin  
    dbms_output.put_line('Bilantul semestrial pentru magazinul ' || v_nume_magazin ||  
    dbms_output.put_line(bilant || ' $');  
end afisare_bilant_semestrial;  
  
end pachet_angajati_magazine;  
/  
  
begin  
    pachet_angajati_magazine.afisare_bilant_anual();  
    pachet_angajati_magazine.afisare_bilant_semestrial();  
  
    for i in pachet_angajati_magazine.nume_magazine loop  
        pachet_angajati_magazine.afisare_bilant_anual(i.nume);  
        pachet_angajati_magazine.afisare_bilant_semestrial(i.nume);  
    end loop;  
end;
```

Bilantul anual pentru Compania TechFlow este: 325200 \$
Bilantul semestrial pentru Compania TechFlow este: 162600 \$
Bilantul anual pentru magazinul TechFlow Red este: 66000 \$
Bilantul semestrial pentru magazinul TechFlow Red este: 33000 \$
Bilantul anual pentru magazinul TechFlow Blue este: 60000 \$
Bilantul semestrial pentru magazinul TechFlow Blue este: 30000 \$
Bilantul anual pentru magazinul TechFlow Black este: 66000 \$
Bilantul semestrial pentru magazinul TechFlow Black este: 33000 \$
Bilantul anual pentru magazinul TechFlow Green este: 62400 \$
Bilantul semestrial pentru magazinul TechFlow Green este: 31200 \$
Bilantul anual pentru magazinul TechFlow Yellow este: 70800 \$
Bilantul semestrial pentru magazinul TechFlow Yellow este: 35400 \$
Bilantul anual pentru magazinul test este: 0 \$
Bilantul semestrial pentru magazinul test este: 0 \$