**Baza de date a brandului Oysho**

Sisteme de baze de date

2024-2025

Aldea Alexia, Master IS

Grupa 406

**Cuprins**

1. [Prezentarea bazei de date aleasă](#21wm17aojk7l)
2. [Realizarea diagramei ERD](#4gdsilcspcx3)
3. [Realizarea diagramei conceptuale](#bmb7a0et40i)
4. [Transformarea sistemului conceptual într-un design logic](#toqzrd8ejhoc)
5. [Convertirea design-ului logic într-un design fizic în FN3](#pxa8siubufjb)
6. [Implementarea tabelelor în Oracle, adaugand chei primare, constrangeri si date](#jlz6b5rzpjp6)
7. [Ilustrarea noțiunilor de PL/SQL învățate](#564x9pbvhr6c)

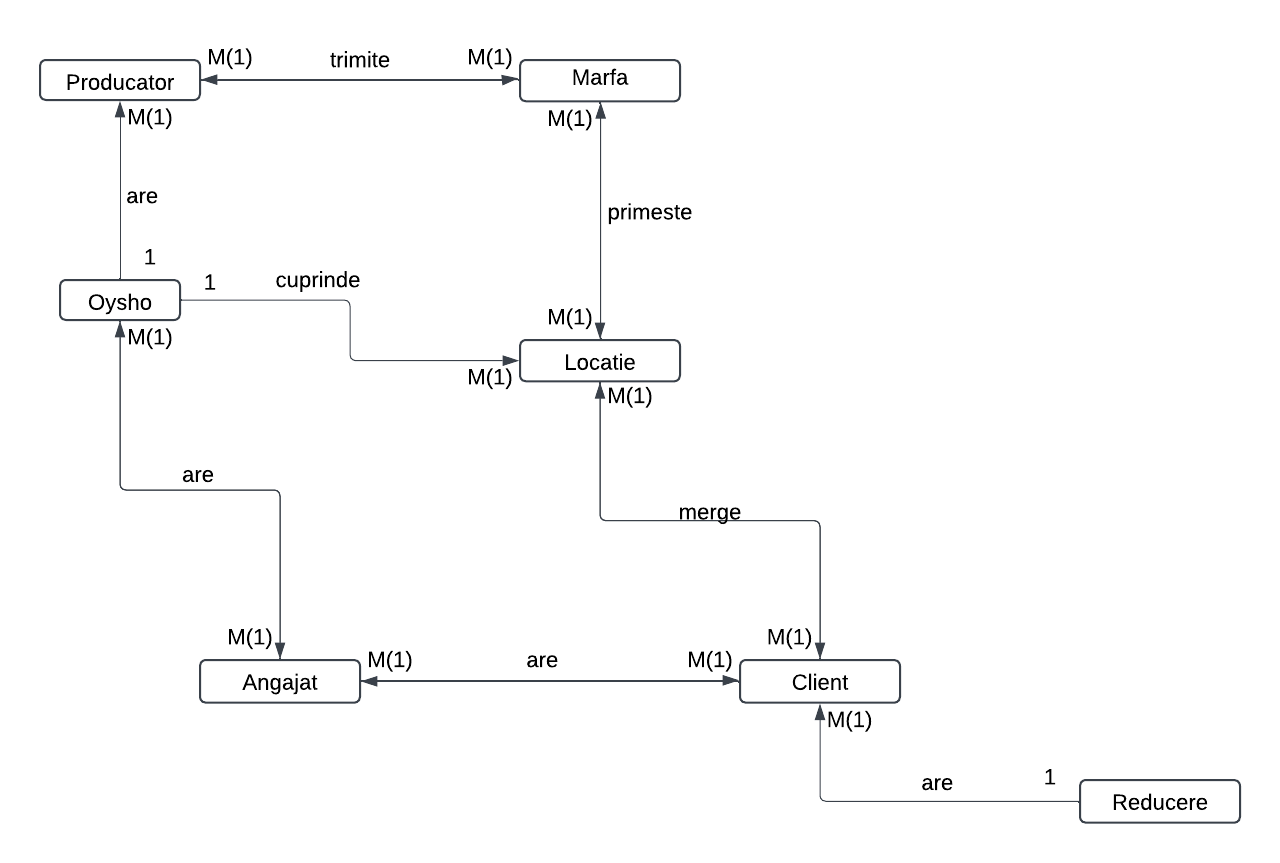
1. **Prezentarea bazei de date aleasă**

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

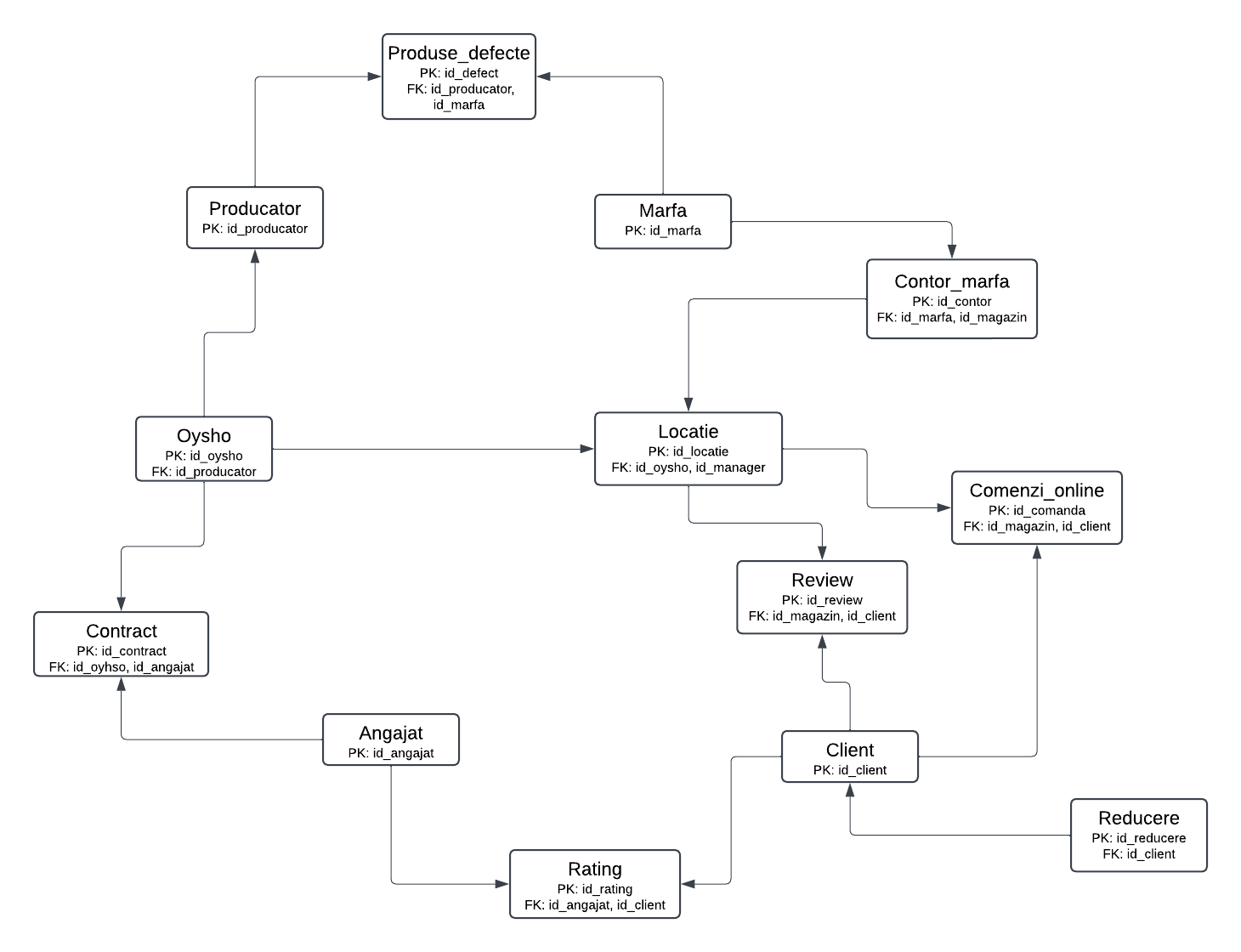
Brandul Oysho are unul sau mai multe contracte cu mai mulți angajați. Pentru fiecare angajat vom reține câteva date de bază despre el, iar în contract, la fel. Un angajat lucrează la un singur mall, dintr-o singură locație. Brandul Oysho are magazine în diferite locuri ale lumii, în mai multe mall-uri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

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

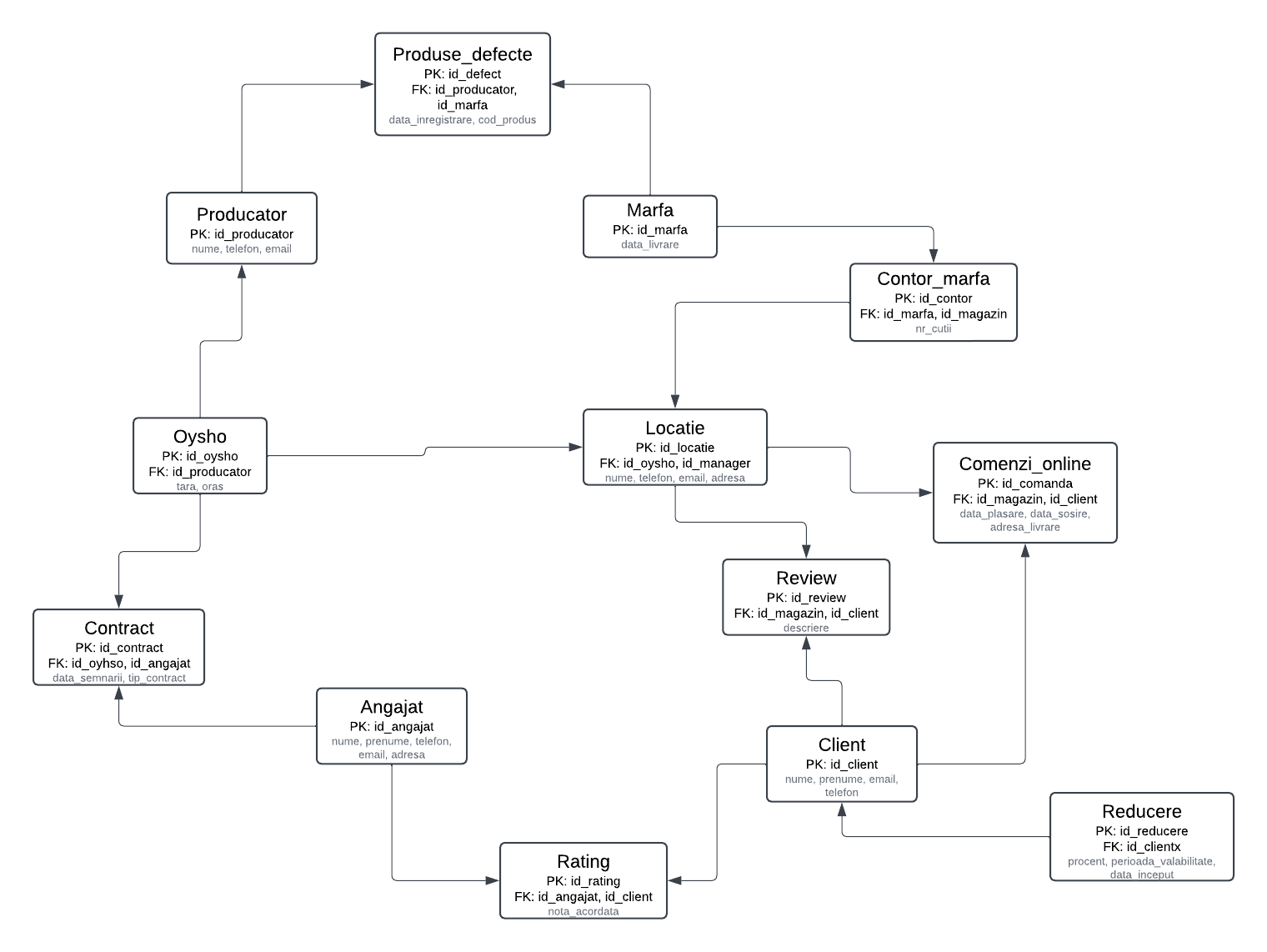
1. **Realizarea diagramei ERD**



1. **Realizarea diagramei conceptuale**



1. **Transformarea sistemului conceptual într-un design logic**



1. **Convertirea design-ului logic într-un design fizic în FN3**

-> un exemplu de atribut repetitiv

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

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

Prin această abordare:

* Am eliminat atributele multivaloare din tabelul `Locatie`.
* Am asigurat că fiecare înregistrare respectă regulile 1NF, în care toate valorile sunt atomice.

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

În versiunea inițială a tabelei “Client”, atributul `reducere` era inclus, permițând stocarea informațiilor despre reduceri în același tabel cu datele clientului. Deși această structură respectă Prima Formă Normală (1NF), ea nu respectă regulile celei de-a Doua Forme Normale (2NF), care impune ca toate atributele să depindă de cheia primară și să nu existe dependențe parțiale.

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

Prin această abordare:

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

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

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

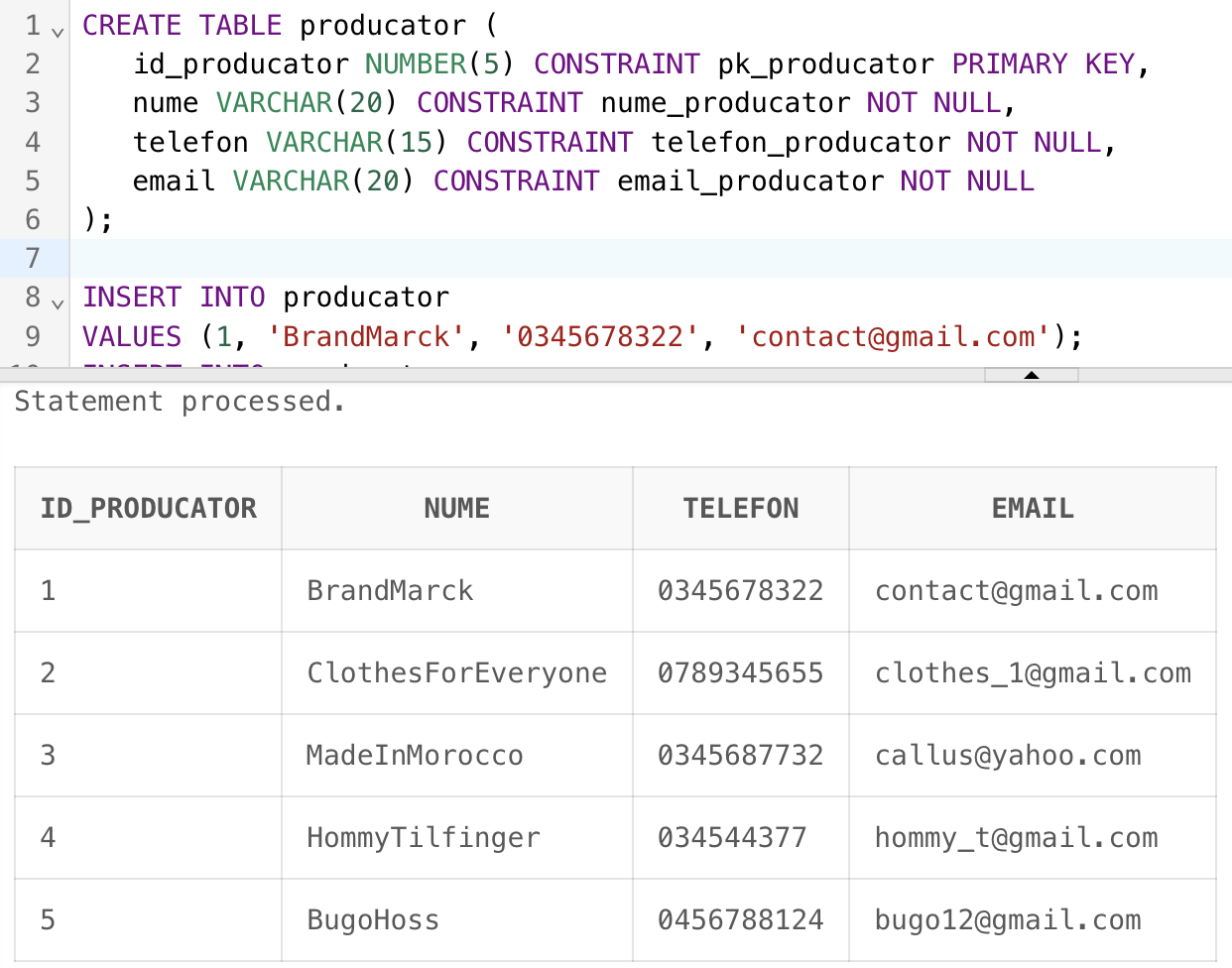
Pentru a aduce tabela în 3NF, am creat un tabel separat, denumit `Oysho`, în care fiecare detaliu legat de brandul Oysho este înregistrat distinct. Acest tabel are o relație clară cu `Producator`, eliminând astfel dependențele tranzitive și asigurând că fiecare atribut este determinat direct doar de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

Prin această abordare:

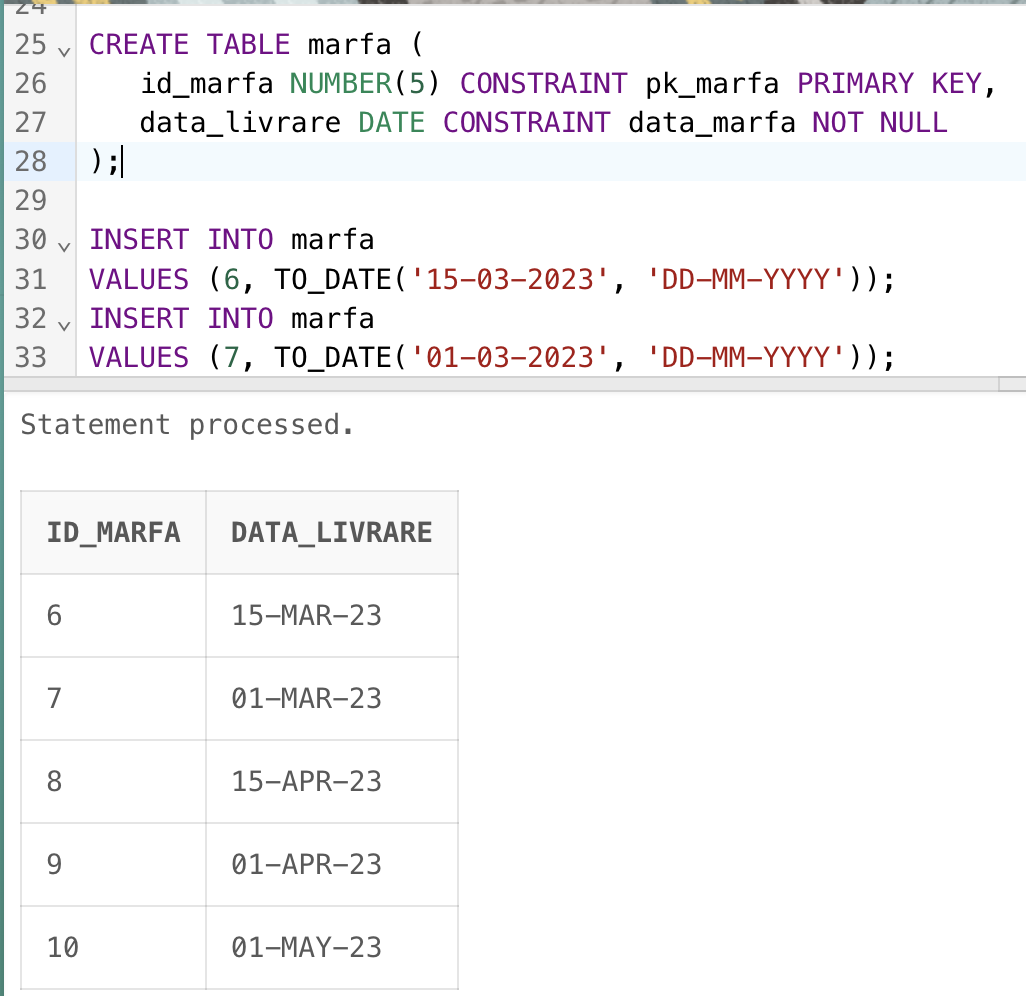
* Am eliminat dependențele tranzitive dintre tabelele `Producator` și `Oysho`.
* Am adus structura bazei de date în conformitate cu regulile 3NF, făcând baza de date mai clară, mai eficientă și mai ușor de întreținut.

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

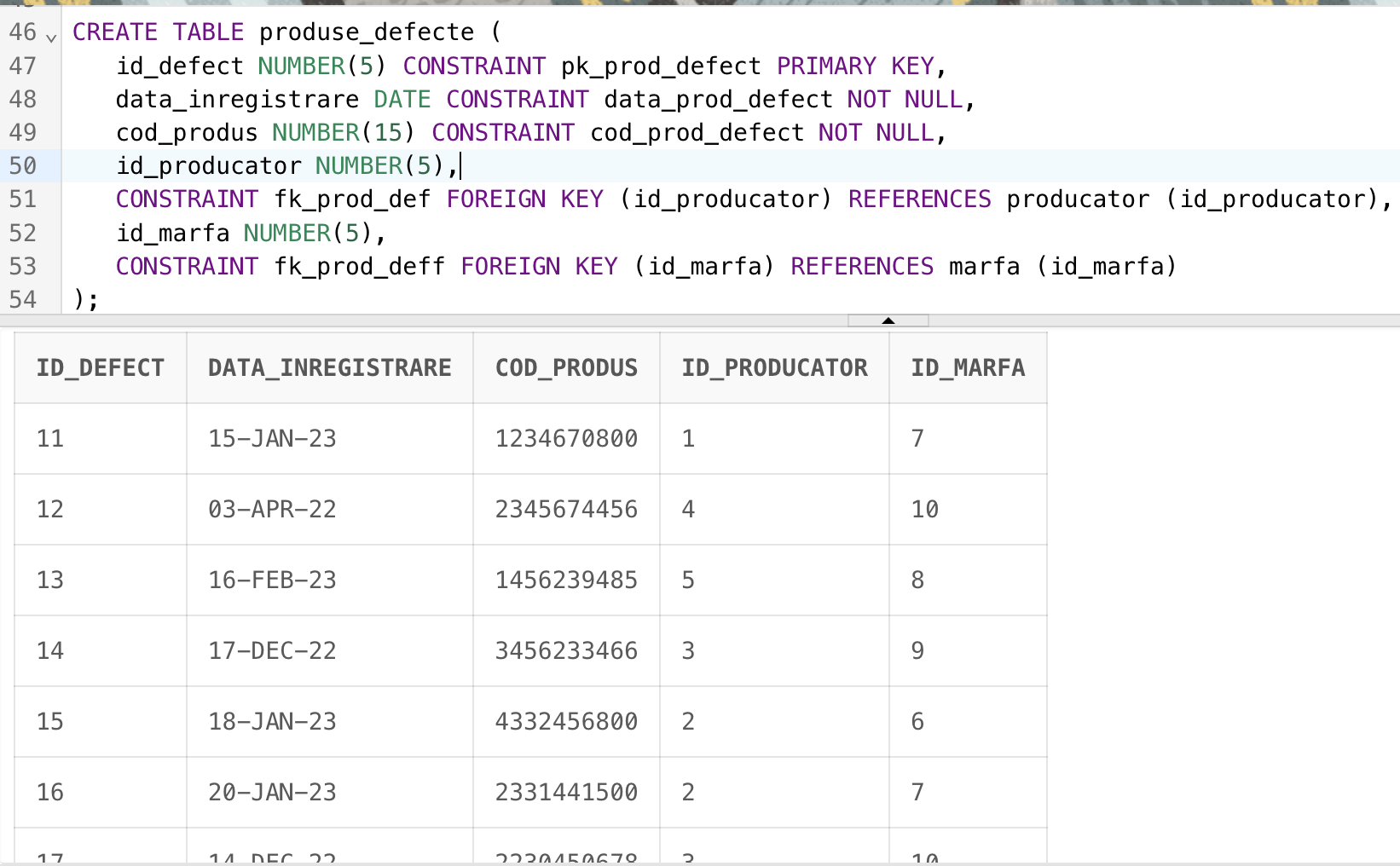
| CREATE TABLE producator (  id\_producator NUMBER(5) CONSTRAINT pk\_producator PRIMARY KEY,  nume VARCHAR(20) CONSTRAINT nume\_producator NOT NULL,  telefon VARCHAR(15) CONSTRAINT telefon\_producator NOT NULL,  email VARCHAR(20) CONSTRAINT email\_producator NOT NULL  );  INSERT INTO producator  VALUES (1, 'BrandMarck', '0345678322', 'contact@gmail.com');  INSERT INTO producator  VALUES (2, 'ClothesForEveryone', '0789345655', 'clothes\_1@gmail.com');  INSERT INTO producator  VALUES (3, 'MadeInMorocco', '0345687732', 'callus@yahoo.com');  INSERT INTO producator  VALUES (4, 'HommyTilfinger', '034544377', 'hommy\_t@gmail.com');  INSERT INTO producator  VALUES (5, 'BugoHoss', '0456788124', 'bugo12@gmail.com');  COMMIT;  SELECT \* FROM producator; |
| --- |

****

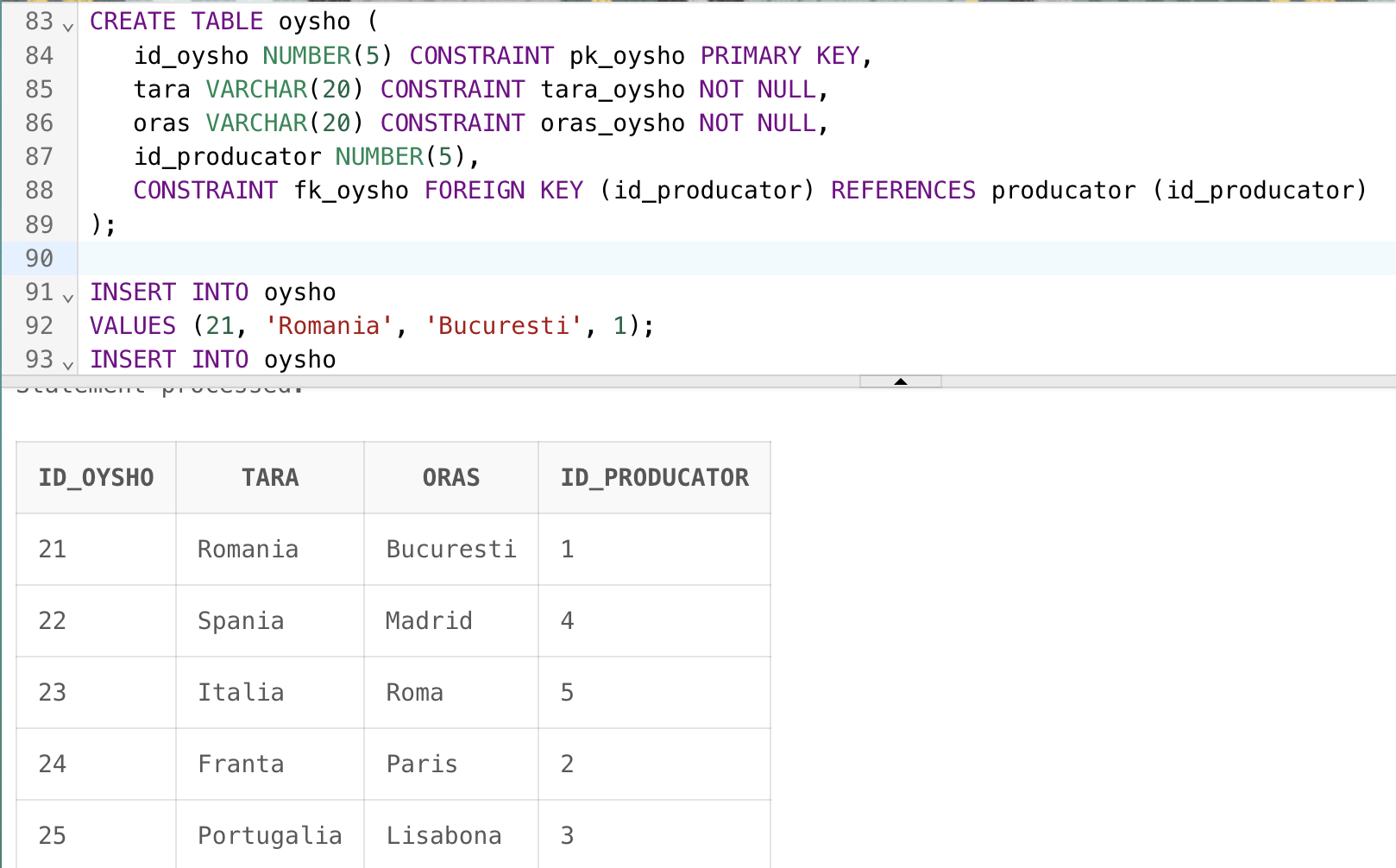
| CREATE TABLE marfa (  id\_marfa NUMBER(5) CONSTRAINT pk\_marfa PRIMARY KEY,  data\_livrare DATE CONSTRAINT data\_marfa NOT NULL  );  INSERT INTO marfa  VALUES (6, TO\_DATE('15-03-2023', 'DD-MM-YYYY'));  INSERT INTO marfa  VALUES (7, TO\_DATE('01-03-2023', 'DD-MM-YYYY'));  INSERT INTO marfa  VALUES (8, TO\_DATE('15-04-2023', 'DD-MM-YYYY'));  INSERT INTO marfa  VALUES (9, TO\_DATE('01-04-2023', 'DD-MM-YYYY'));  INSERT INTO marfa  VALUES (10, TO\_DATE('01-05-2023', 'DD-MM-YYYY'));  COMMIT;  SELECT \* FROM marfa; |
| --- |

****

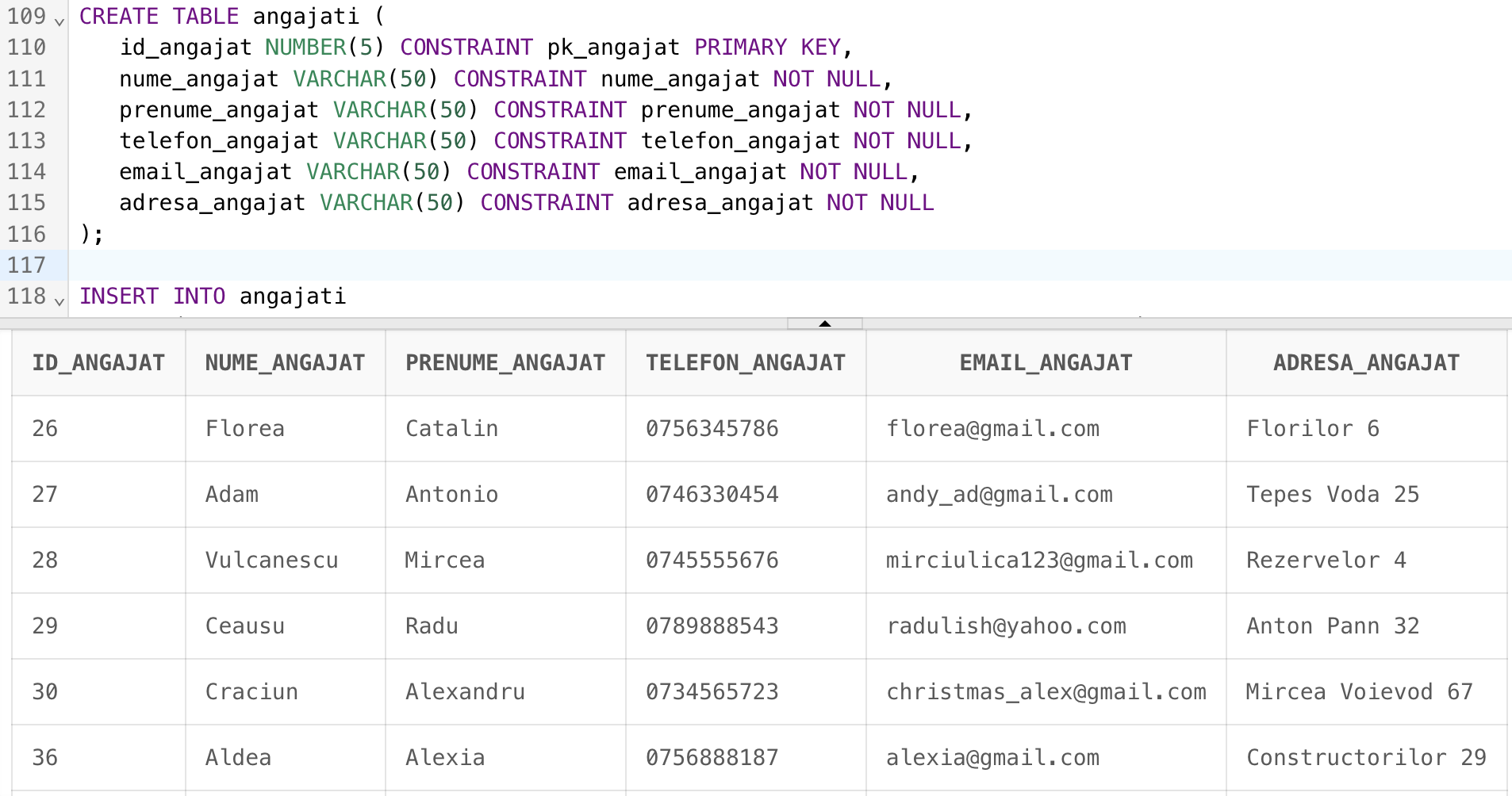
| CREATE TABLE produse\_defecte (  id\_defect NUMBER(5) CONSTRAINT pk\_prod\_defect PRIMARY KEY,  data\_inregistrare DATE CONSTRAINT data\_prod\_defect NOT NULL,  cod\_produs NUMBER(15) CONSTRAINT cod\_prod\_defect NOT NULL,  id\_producator NUMBER(5),  CONSTRAINT fk\_prod\_def FOREIGN KEY (id\_producator) REFERENCES producator (id\_producator),  id\_marfa NUMBER(5),  CONSTRAINT fk\_prod\_deff FOREIGN KEY (id\_marfa) REFERENCES marfa (id\_marfa)  );  INSERT INTO produse\_defecte  VALUES (11, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 1234670800, 1, 7);  INSERT INTO produse\_defecte  VALUES (12, TO\_DATE('03-04-2022', 'DD-MM-YYYY'), 2345674456, 4, 10);  INSERT INTO produse\_defecte  VALUES (13, TO\_DATE('16-02-2023', 'DD-MM-YYYY'), 1456239485, 5, 8);  INSERT INTO produse\_defecte  VALUES (14, TO\_DATE('17-12-2022', 'DD-MM-YYYY'), 3456233466, 3, 9);  INSERT INTO produse\_defecte  VALUES (15, TO\_DATE('18-01-2023', 'DD-MM-YYYY'), 4332456800, 2, 6);  INSERT INTO produse\_defecte  VALUES (16, TO\_DATE('20-01-2023', 'DD-MM-YYYY'), 2331441500, 2, 7);  INSERT INTO produse\_defecte  VALUES (17, TO\_DATE('14-12-2022', 'DD-MM-YYYY'), 2230450678, 3, 10);  INSERT INTO produse\_defecte  VALUES (18, TO\_DATE('01-02-2023', 'DD-MM-YYYY'), 4523900800, 5, 9);  INSERT INTO produse\_defecte  VALUES (19, TO\_DATE('22-11-2022', 'DD-MM-YYYY'), 4567890957, 1, 8);  INSERT INTO produse\_defecte  VALUES (20, TO\_DATE('13-01-2023', 'DD-MM-YYYY'), 3453230567, 3, 9);  COMMIT;  SELECT \* FROM produse\_defecte; |
| --- |

****

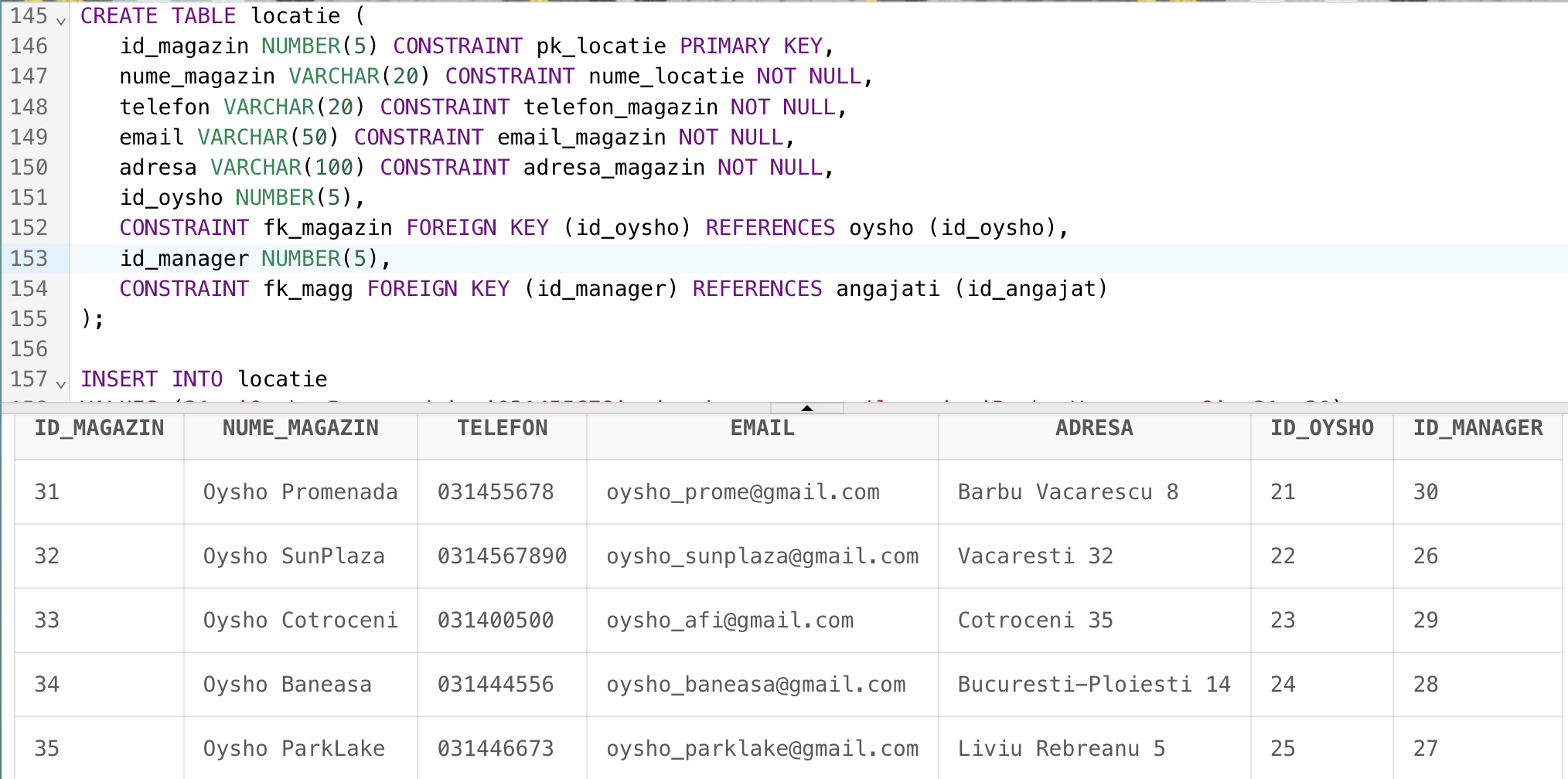
| CREATE TABLE oysho (  id\_oysho NUMBER(5) CONSTRAINT pk\_oysho PRIMARY KEY,  tara VARCHAR(20) CONSTRAINT tara\_oysho NOT NULL,  oras VARCHAR(20) CONSTRAINT oras\_oysho NOT NULL,  id\_producator NUMBER(5),  CONSTRAINT fk\_oysho FOREIGN KEY (id\_producator) REFERENCES producator (id\_producator)  );  INSERT INTO oysho  VALUES (21, 'Romania', 'Bucuresti', 1);  INSERT INTO oysho  VALUES (22, 'Spania', 'Madrid', 4);  INSERT INTO oysho  VALUES (23, 'Italia', 'Roma', 5);  INSERT INTO oysho  VALUES (24, 'Franta', 'Paris', 2);  INSERT INTO oysho  VALUES (25, 'Portugalia', 'Lisabona', 3);  COMMIT;  SELECT \* FROM oysho; |
| --- |

****

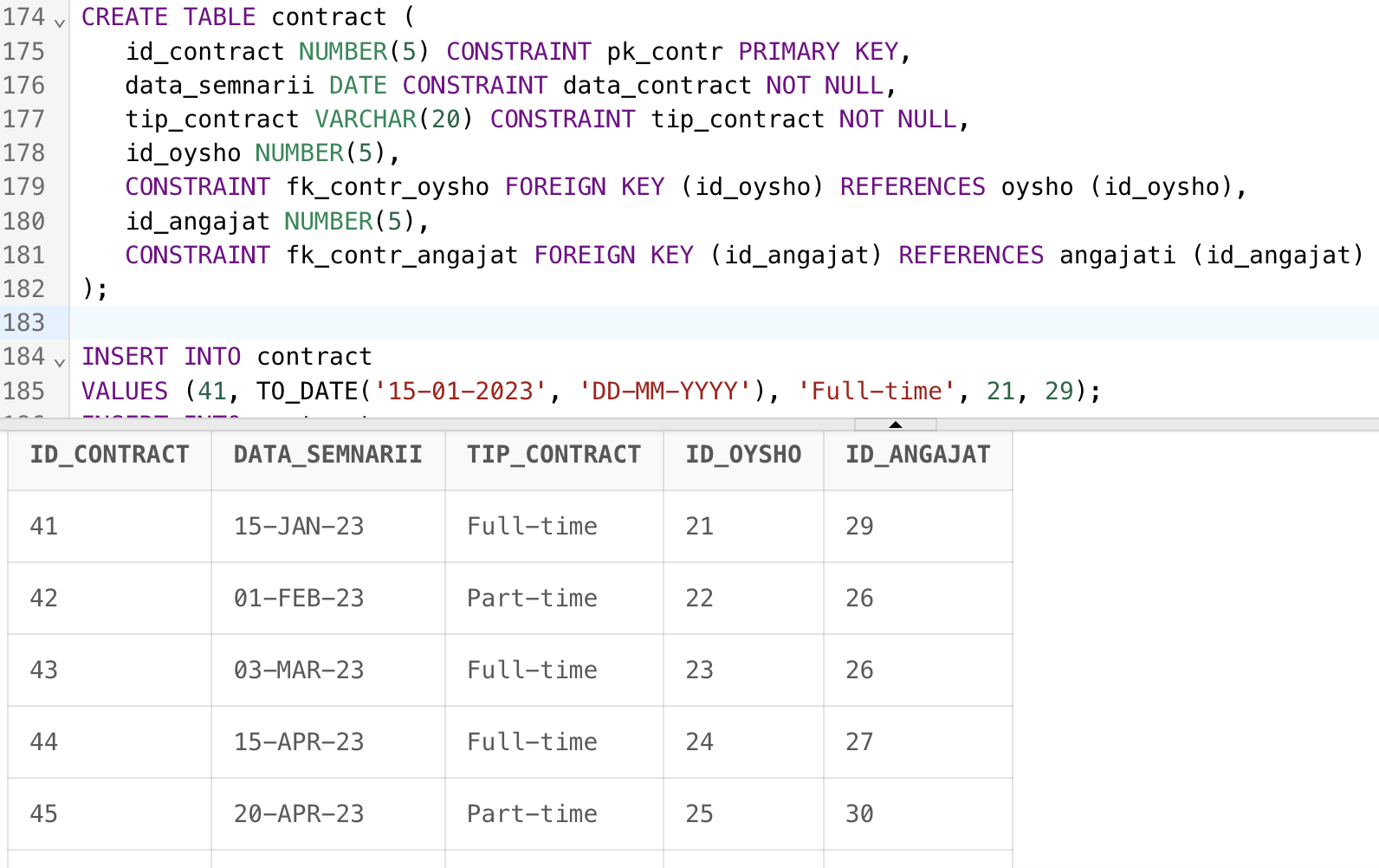
| CREATE TABLE angajati (  id\_angajat NUMBER(5) CONSTRAINT pk\_angajat PRIMARY KEY,  nume\_angajat VARCHAR(50) CONSTRAINT nume\_angajat NOT NULL,  prenume\_angajat VARCHAR(50) CONSTRAINT prenume\_angajat NOT NULL,  telefon\_angajat VARCHAR(50) CONSTRAINT telefon\_angajat NOT NULL,  email\_angajat VARCHAR(50) CONSTRAINT email\_angajat NOT NULL,  adresa\_angajat VARCHAR(50) CONSTRAINT adresa\_angajat NOT NULL  );  INSERT INTO angajati  VALUES (26, 'Florea', 'Catalin', '0756345786', 'florea@gmail.com', 'Florilor 6');  INSERT INTO angajati  VALUES (27, 'Adam', 'Antonio', '0746330454', 'andy\_ad@gmail.com', 'Tepes Voda 25');  INSERT INTO angajati  VALUES (28, 'Vulcanescu', 'Mircea', '0745555676', 'mirciulica123@gmail.com', 'Rezervelor 4');  INSERT INTO angajati  VALUES (29, 'Ceausu', 'Radu', '0789888543', 'radulish@yahoo.com', 'Anton Pann 32');  INSERT INTO angajati  VALUES (30, 'Craciun', 'Alexandru', '0734565723', 'christmas\_alex@gmail.com', 'Mircea Voievod 67');  INSERT INTO angajati  VALUES (36, 'Aldea', 'Alexia', '0756888187', 'alexia@gmail.com', 'Constructorilor 29');  INSERT INTO angajati  VALUES (37, 'Rosu', 'Cristina', '0731456777', 'cris\_rosu@gmail.com', 'Teiul Doamnnei 12');  INSERT INTO angajati  VALUES (38, 'Luta', 'Alexandra', '0733454558', 'kiwi@gmail.com', 'Grozavesti 34');  INSERT INTO angajati  VALUES (39, 'Pentu', 'Miruna', '0735464889', 'mirus.spirus@gmail.com', 'Bucurestii Noi 1');  INSERT INTO angajati  VALUES (40, 'Bozeanu', 'Mirela', '0763888176', 'miree\_boze@gmail.com', 'Morarilor 90');  COMMIT;  SELECT \* FROM angajati; |
| --- |

****

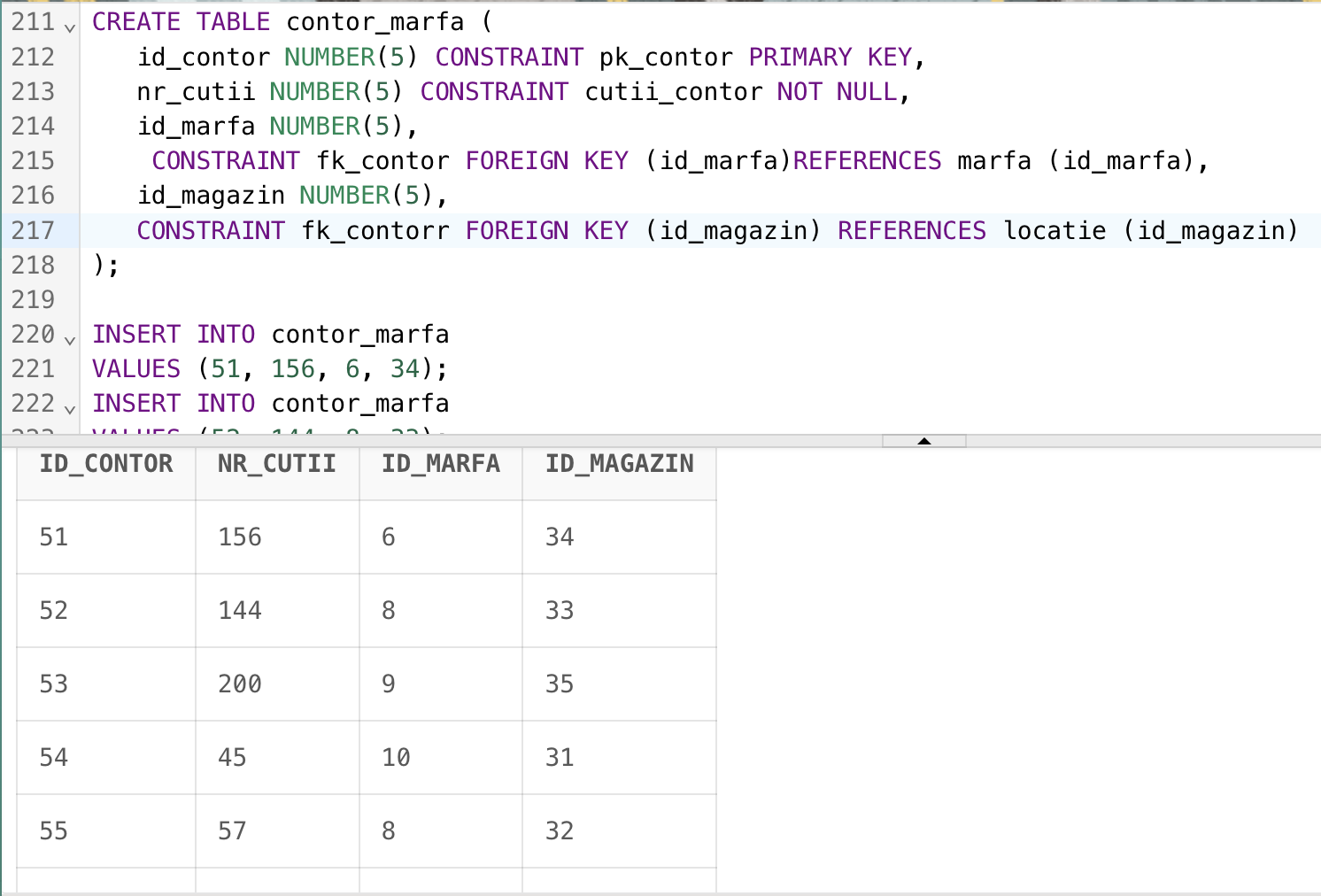
| CREATE TABLE locatie (  id\_magazin NUMBER(5) CONSTRAINT pk\_locatie PRIMARY KEY,  nume\_magazin VARCHAR(20) CONSTRAINT nume\_locatie NOT NULL,  telefon VARCHAR(20) CONSTRAINT telefon\_magazin NOT NULL,  email VARCHAR(50) CONSTRAINT email\_magazin NOT NULL,  adresa VARCHAR(100) CONSTRAINT adresa\_magazin NOT NULL,  id\_oysho NUMBER(5),  CONSTRAINT fk\_magazin FOREIGN KEY (id\_oysho) REFERENCES oysho (id\_oysho),  id\_manager NUMBER(5),  CONSTRAINT fk\_magg FOREIGN KEY (id\_manager) REFERENCES angajati (id\_angajat)  );  INSERT INTO locatie  VALUES (31, 'Oysho Promenada', '031455678', 'oysho\_prome@gmail.com', 'Barbu Vacarescu 8', 21, 30);  INSERT INTO locatie  VALUES (32, 'Oysho SunPlaza', '0314567890', 'oysho\_sunplaza@gmail.com', 'Vacaresti 32', 22, 26);  INSERT INTO locatie  VALUES (33, 'Oysho Cotroceni', '031400500', 'oysho\_afi@gmail.com', 'Cotroceni 35', 23, 29);  INSERT INTO locatie  VALUES (34, 'Oysho Baneasa', '031444556', 'oysho\_baneasa@gmail.com', 'Bucuresti-Ploiesti 14', 24, 28);  INSERT INTO locatie  VALUES (35, 'Oysho ParkLake', '031446673', 'oysho\_parklake@gmail.com', 'Liviu Rebreanu 5', 25, 27);  COMMIT;  SELECT \* FROM locatie; |
| --- |

****

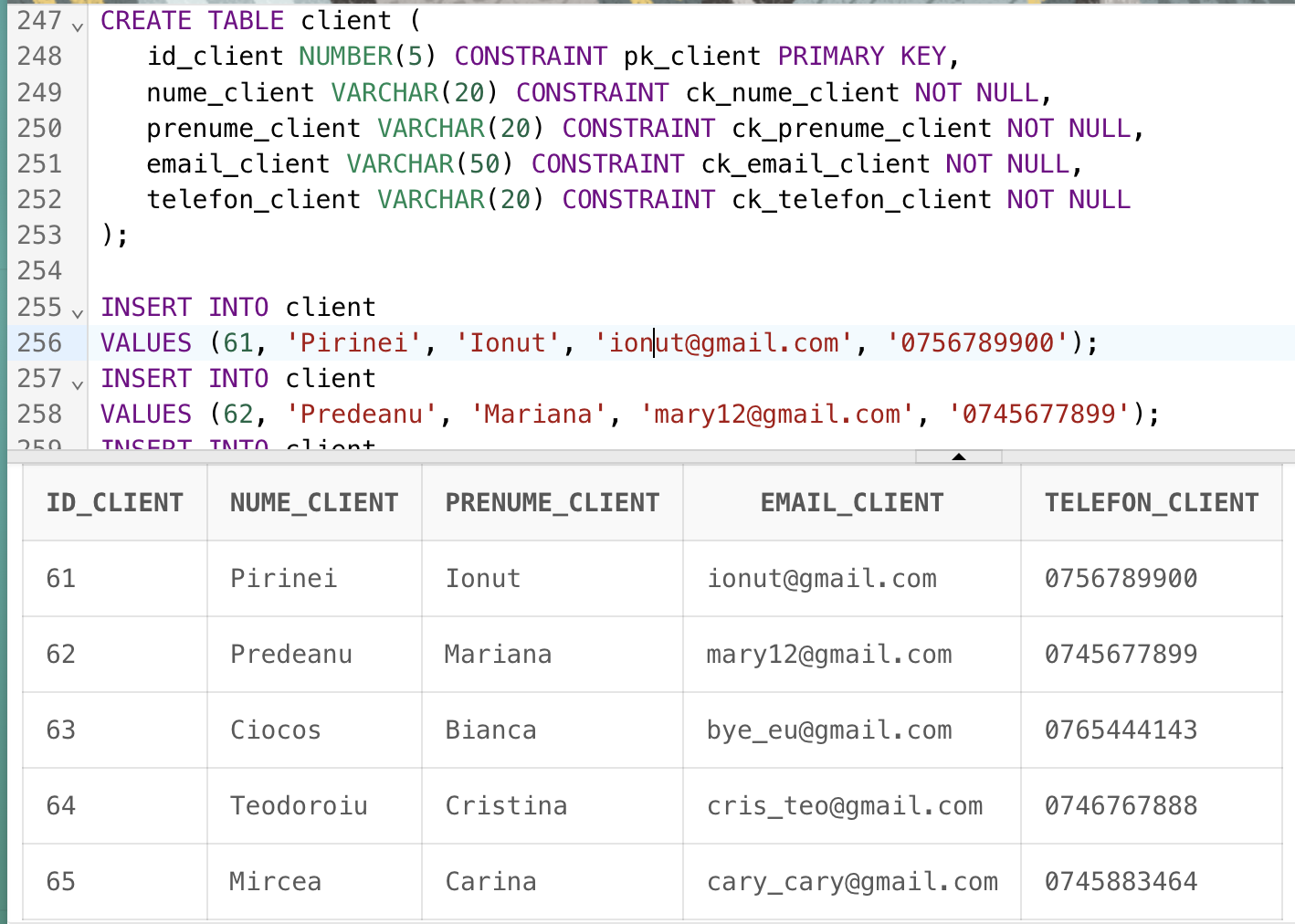
| CREATE TABLE contract (  id\_contract NUMBER(5) CONSTRAINT pk\_contr PRIMARY KEY,  data\_semnarii DATE CONSTRAINT data\_contract NOT NULL,  tip\_contract VARCHAR(20) CONSTRAINT tip\_contract NOT NULL,  id\_oysho NUMBER(5),  CONSTRAINT fk\_contr\_oysho FOREIGN KEY (id\_oysho) REFERENCES oysho (id\_oysho),  id\_angajat NUMBER(5),  CONSTRAINT fk\_contr\_angajat FOREIGN KEY (id\_angajat) REFERENCES angajati (id\_angajat)  );  INSERT INTO contract  VALUES (41, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 'Full-time', 21, 29);  INSERT INTO contract  VALUES (42, TO\_DATE('01-02-2023', 'DD-MM-YYYY'), 'Part-time', 22, 26);  INSERT INTO contract  VALUES (43, TO\_DATE('03-03-2023', 'DD-MM-YYYY'), 'Full-time', 23, 26);  INSERT INTO contract  VALUES (44, TO\_DATE('15-04-2023', 'DD-MM-YYYY'), 'Full-time', 24, 27);  INSERT INTO contract  VALUES (45, TO\_DATE('20-04-2023', 'DD-MM-YYYY'), 'Part-time', 25, 30);  INSERT INTO contract  VALUES (46, TO\_DATE('18-02-2023', 'DD-MM-YYYY'), 'Full-time', 22, 31);  INSERT INTO contract  VALUES (47, TO\_DATE('31-01-2020', 'DD-MM-YYYY'), 'Part-time', 25, 40);  INSERT INTO contract  VALUES (48, TO\_DATE('16-10-2021', 'DD-MM-YYYY'), 'Part-time', 24, 36);  INSERT INTO contract  VALUES (49, TO\_DATE('25-09-2021', 'DD-MM-YYYY'), 'Part-time', 21, 37);  INSERT INTO contract  VALUES (50, TO\_DATE('19-07-2022', 'DD-MM-YYYY'), 'Part-time', 23, 38);  COMMIT;  SELECT \* FROM contract; |
| --- |

****

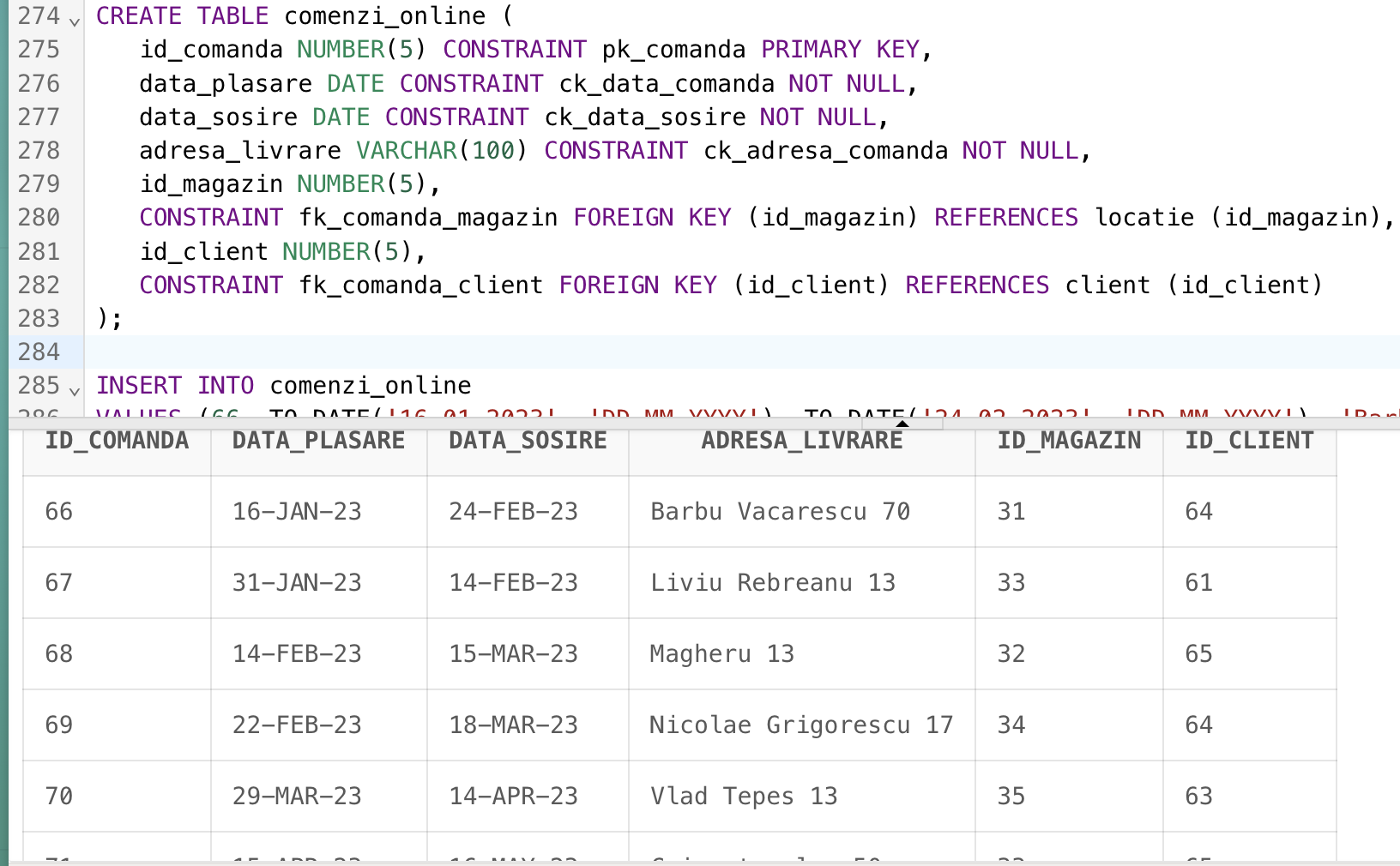
| CREATE TABLE contor\_marfa (  id\_contor NUMBER(5) CONSTRAINT pk\_contor PRIMARY KEY,  nr\_cutii NUMBER(5) CONSTRAINT cutii\_contor NOT NULL,  id\_marfa NUMBER(5),  CONSTRAINT fk\_contor FOREIGN KEY (id\_marfa)REFERENCES marfa (id\_marfa),  id\_magazin NUMBER(5),  CONSTRAINT fk\_contorr FOREIGN KEY (id\_magazin) REFERENCES locatie (id\_magazin)  );  INSERT INTO contor\_marfa  VALUES (51, 156, 6, 34);  INSERT INTO contor\_marfa  VALUES (52, 144, 8, 33);  INSERT INTO contor\_marfa  VALUES (53, 200, 9, 35);  INSERT INTO contor\_marfa  VALUES (54, 45, 10, 31);  INSERT INTO contor\_marfa  VALUES (55, 57, 8, 32);  INSERT INTO contor\_marfa  VALUES (56, 230, 7, 35);  INSERT INTO contor\_marfa  VALUES (57, 400, 6, 33);  INSERT INTO contor\_marfa  VALUES (58, 245, 9, 32);  INSERT INTO contor\_marfa  VALUES (59, 120, 7, 35);  INSERT INTO contor\_marfa  VALUES (60, 89, 9, 34);  COMMIT;  SELECT \* FROM contor\_marfa; |
| --- |

****

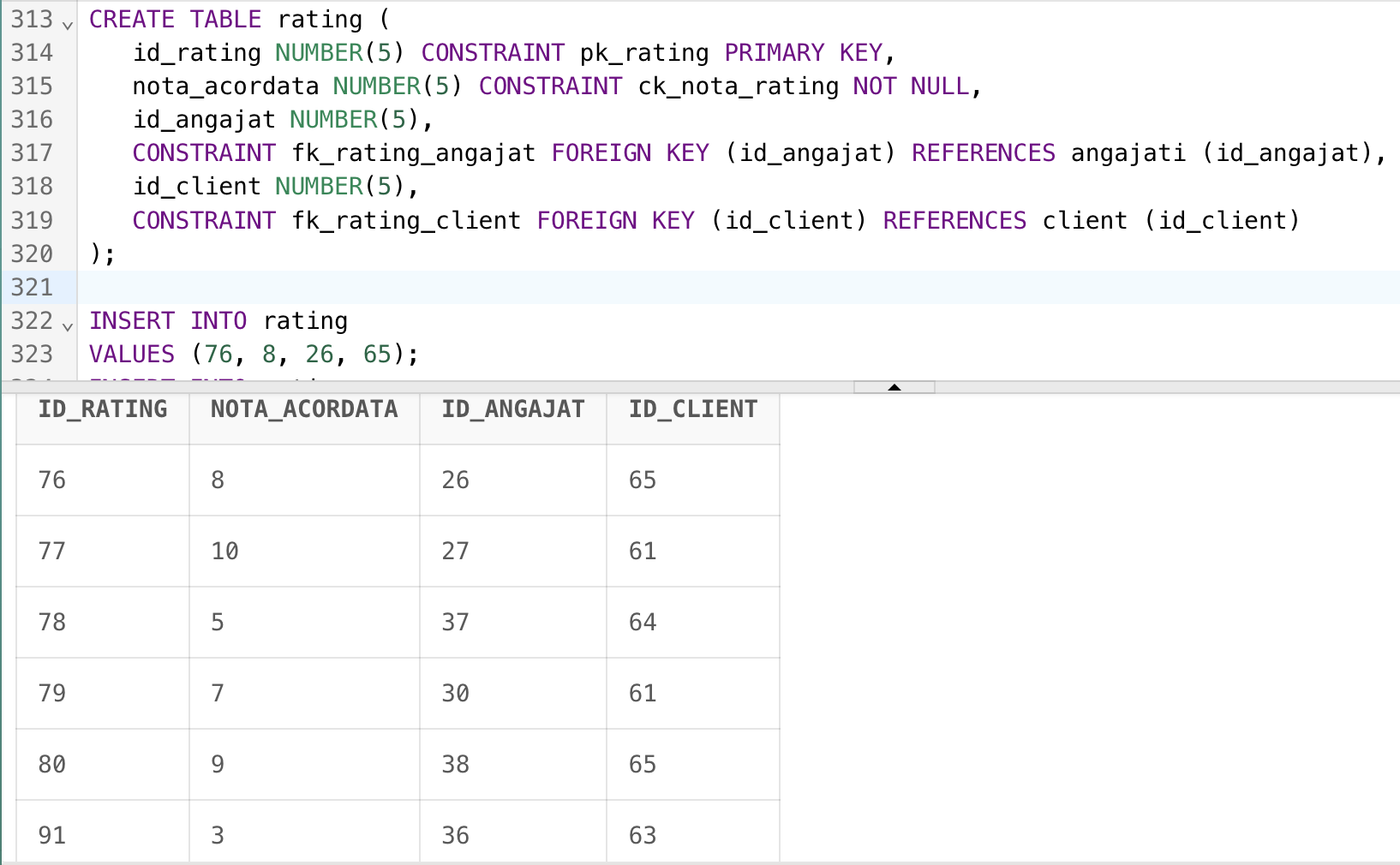
| CREATE TABLE client (  id\_client NUMBER(5) CONSTRAINT pk\_client PRIMARY KEY,  nume\_client VARCHAR(20) CONSTRAINT ck\_nume\_client NOT NULL,  prenume\_client VARCHAR(20) CONSTRAINT ck\_prenume\_client NOT NULL,  email\_client VARCHAR(50) CONSTRAINT ck\_email\_client NOT NULL,  telefon\_client VARCHAR(20) CONSTRAINT ck\_telefon\_client NOT NULL  );  INSERT INTO client  VALUES (61, 'Pirinei', 'Ionut', 'ionut@gmail.com', '0756789900');  INSERT INTO client  VALUES (62, 'Predeanu', 'Mariana', 'mary12@gmail.com', '0745677899');  INSERT INTO client  VALUES (63, 'Ciocos', 'Bianca', 'bye\_eu@gmail.com', '0765444143');  INSERT INTO client  VALUES (64, 'Teodoroiu', 'Cristina', 'cris\_teo@gmail.com', '0746767888');  INSERT INTO client  VALUES (65, 'Mircea', 'Carina', 'cary\_cary@gmail.com', '0745883464');  COMMIT;  SELECT \* FROM client; |
| --- |

****

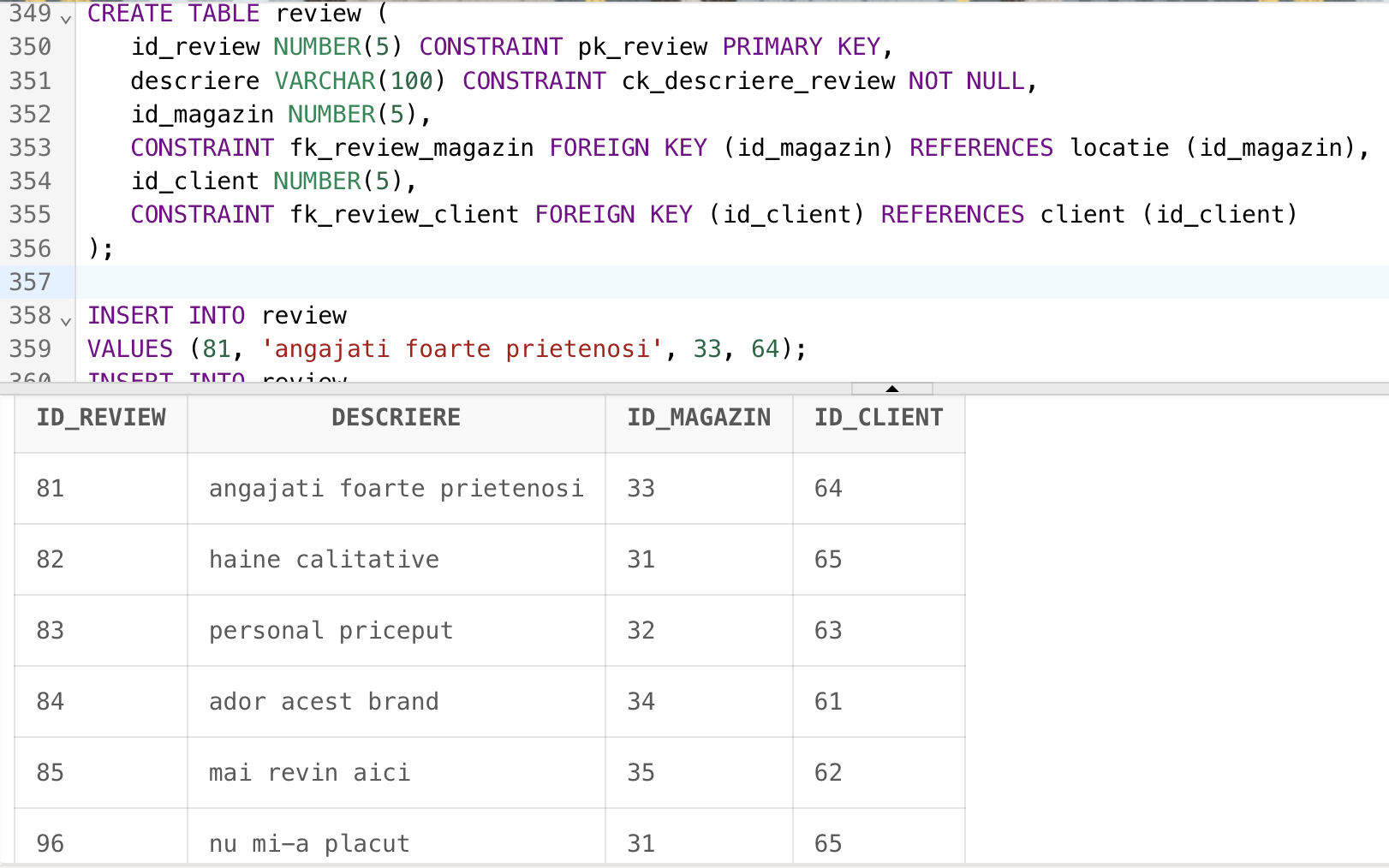
| CREATE TABLE comenzi\_online (  id\_comanda NUMBER(5) CONSTRAINT pk\_comanda PRIMARY KEY,  data\_plasare DATE CONSTRAINT ck\_data\_comanda NOT NULL,  data\_sosire DATE CONSTRAINT ck\_data\_sosire NOT NULL,  adresa\_livrare VARCHAR(100) CONSTRAINT ck\_adresa\_comanda NOT NULL,  id\_magazin NUMBER(5),  CONSTRAINT fk\_comanda\_magazin FOREIGN KEY (id\_magazin) REFERENCES locatie (id\_magazin),  id\_client NUMBER(5),  CONSTRAINT fk\_comanda\_client FOREIGN KEY (id\_client) REFERENCES client (id\_client)  );  INSERT INTO comenzi\_online  VALUES (66, TO\_DATE('16-01-2023', 'DD-MM-YYYY'), TO\_DATE('24-02-2023', 'DD-MM-YYYY'), 'Barbu Vacarescu 70', 31, 64);  INSERT INTO comenzi\_online  VALUES (67, TO\_DATE('31-01-2023', 'DD-MM-YYYY'), TO\_DATE('14-02-2023', 'DD-MM-YYYY'), 'Liviu Rebreanu 13', 33, 61);  INSERT INTO comenzi\_online  VALUES (68, TO\_DATE('14-02-2023', 'DD-MM-YYYY'), TO\_DATE('15-03-2023', 'DD-MM-YYYY'), 'Magheru 13', 32, 65);  INSERT INTO comenzi\_online  VALUES (69, TO\_DATE('22-02-2023', 'DD-MM-YYYY'), TO\_DATE('18-03-2023', 'DD-MM-YYYY'), 'Nicolae Grigorescu 17', 34, 64);  INSERT INTO comenzi\_online  VALUES (70, TO\_DATE('29-03-2023', 'DD-MM-YYYY'), TO\_DATE('14-04-2023', 'DD-MM-YYYY'), 'Vlad Tepes 13', 35, 63);  INSERT INTO comenzi\_online  VALUES (71, TO\_DATE('15-04-2023', 'DD-MM-YYYY'), TO\_DATE('16-05-2023', 'DD-MM-YYYY'), 'Crizantemelor 50', 33, 65);  INSERT INTO comenzi\_online  VALUES (72, TO\_DATE('20-03-2023', 'DD-MM-YYYY'), TO\_DATE('17-04-2023', 'DD-MM-YYYY'), 'Decebal 20', 31, 61);  INSERT INTO comenzi\_online  VALUES (73, TO\_DATE('14-02-2023', 'DD-MM-YYYY'), TO\_DATE('18-04-2023', 'DD-MM-YYYY'), 'Lascar Catargiu 12', 32, 65);  INSERT INTO comenzi\_online  VALUES (74, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), TO\_DATE('13-02-2023', 'DD-MM-YYYY'), 'Regina Elisabeta 3', 34, 65);  INSERT INTO comenzi\_online  VALUES (75, TO\_DATE('22-03-2023', 'DD-MM-YYYY'), TO\_DATE('14-04-2023', 'DD-MM-YYYY'), 'Sfanta Vineri 34', 32, 62);  COMMIT;  SELECT \* FROM comenzi\_online; |
| --- |

****

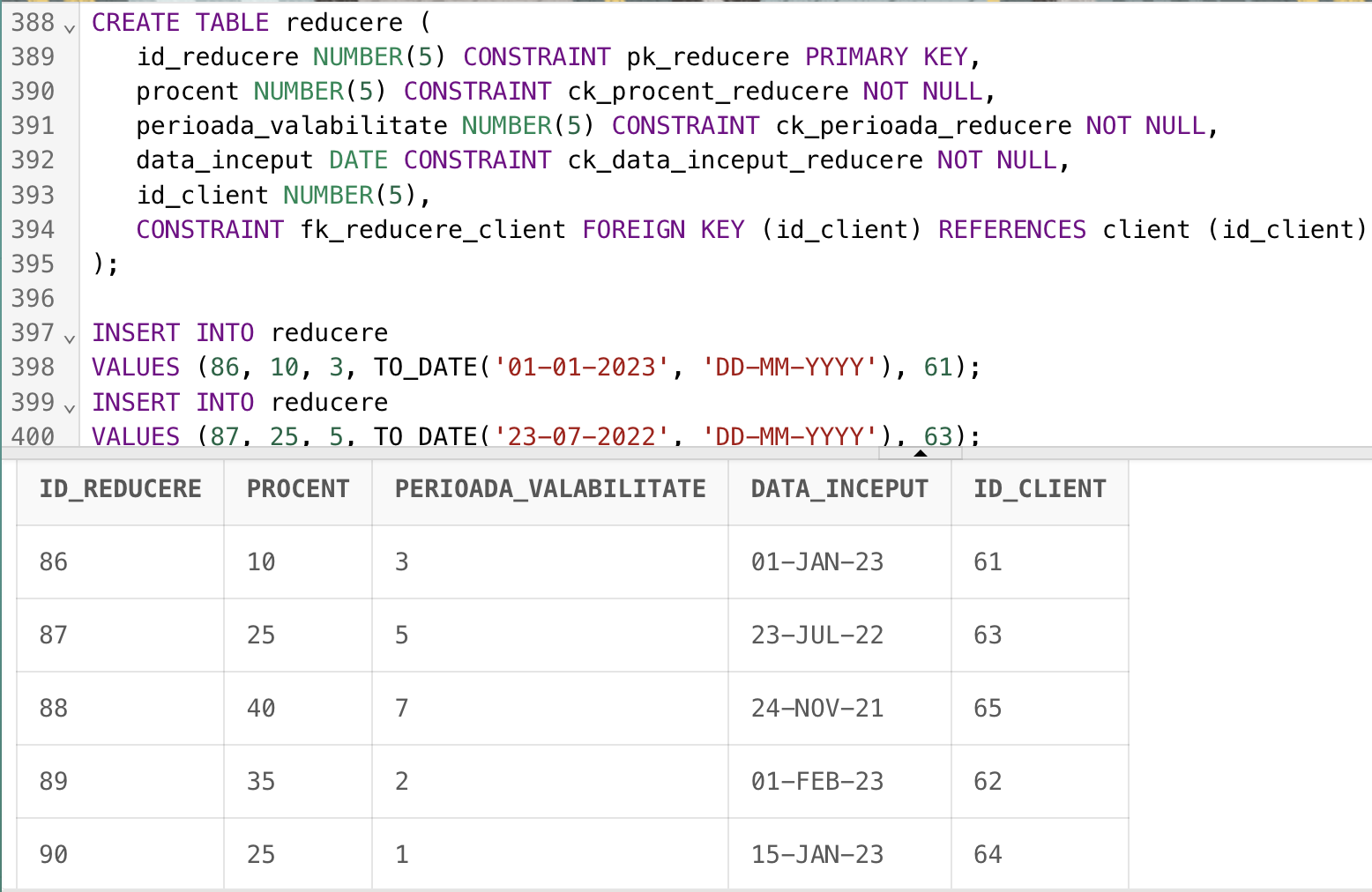
| CREATE TABLE rating (  id\_rating NUMBER(5) CONSTRAINT pk\_rating PRIMARY KEY,  nota\_acordata NUMBER(5) CONSTRAINT ck\_nota\_rating NOT NULL,  id\_angajat NUMBER(5),  CONSTRAINT fk\_rating\_angajat FOREIGN KEY (id\_angajat) REFERENCES angajati (id\_angajat),  id\_client NUMBER(5),  CONSTRAINT fk\_rating\_client FOREIGN KEY (id\_client) REFERENCES client (id\_client)  );  INSERT INTO rating  VALUES (76, 8, 26, 65);  INSERT INTO rating  VALUES (77, 10, 27, 61);  INSERT INTO rating  VALUES (78, 5, 37, 64);  INSERT INTO rating  VALUES (79, 7, 30, 61);  INSERT INTO rating  VALUES (80, 9, 38, 65);  INSERT INTO rating  VALUES (91, 3, 36, 63);  INSERT INTO rating  VALUES (92, 7, 39, 64);  INSERT INTO rating  VALUES (93, 2, 28, 61);  INSERT INTO rating  VALUES (94, 10, 26, 65);  INSERT INTO rating  VALUES (95, 9, 38, 63);  COMMIT;  SELECT \* FROM rating; |
| --- |

****

| CREATE TABLE review (  id\_review NUMBER(5) CONSTRAINT pk\_review PRIMARY KEY,  descriere VARCHAR(100) CONSTRAINT ck\_descriere\_review NOT NULL,  id\_magazin NUMBER(5),  CONSTRAINT fk\_review\_magazin FOREIGN KEY (id\_magazin) REFERENCES locatie (id\_magazin),  id\_client NUMBER(5),  CONSTRAINT fk\_review\_client FOREIGN KEY (id\_client) REFERENCES client (id\_client)  );  INSERT INTO review  VALUES (81, 'angajati foarte prietenosi', 33, 64);  INSERT INTO review  VALUES (82, 'haine calitative', 31, 65);  INSERT INTO review  VALUES (83, 'personal priceput', 32, 63);  INSERT INTO review  VALUES (84, 'ador acest brand', 34, 61);  INSERT INTO review  VALUES (85, 'mai revin aici', 35, 62);  INSERT INTO review  VALUES (96, 'nu mi-a placut', 31, 65);  INSERT INTO review  VALUES (97, 'haine calitative', 33, 63);  INSERT INTO review  VALUES (98, 'personal priceput', 34, 62);  INSERT INTO review  VALUES (99, 'ador acest brand', 31, 61);  INSERT INTO review  VALUES (100, 'mai revin aici', 34, 62);  COMMIT;  SELECT \* FROM review; |
| --- |

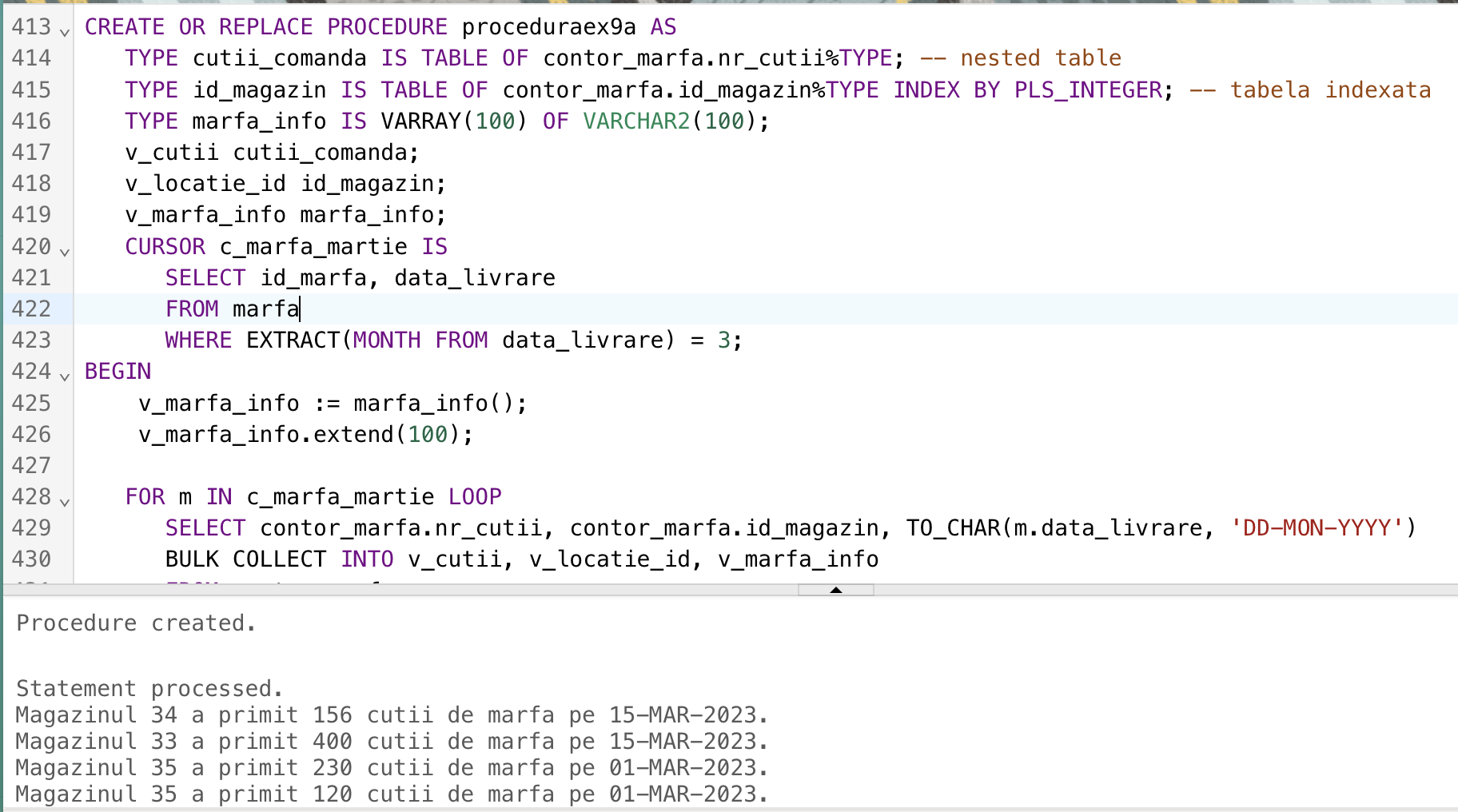
****

| CREATE TABLE reducere (  id\_reducere NUMBER(5) CONSTRAINT pk\_reducere PRIMARY KEY,  procent NUMBER(5) CONSTRAINT ck\_procent\_reducere NOT NULL,  perioada\_valabilitate NUMBER(5) CONSTRAINT ck\_perioada\_reducere NOT NULL,  data\_inceput DATE CONSTRAINT ck\_data\_inceput\_reducere NOT NULL,  id\_client NUMBER(5),  CONSTRAINT fk\_reducere\_client FOREIGN KEY (id\_client) REFERENCES client (id\_client)  );  INSERT INTO reducere  VALUES (86, 10, 3, TO\_DATE('01-01-2023', 'DD-MM-YYYY'), 61);  INSERT INTO reducere  VALUES (87, 25, 5, TO\_DATE('23-07-2022', 'DD-MM-YYYY'), 63);  INSERT INTO reducere  VALUES (88, 40, 7, TO\_DATE('24-11-2021', 'DD-MM-YYYY'), 65);  INSERT INTO reducere  VALUES (89, 35, 2, TO\_DATE('01-02-2023', 'DD-MM-YYYY'), 62);  INSERT INTO reducere  VALUES (90, 25, 1, TO\_DATE('15-01-2023', 'DD-MM-YYYY'), 64);  COMMIT;  SELECT \* FROM reducere; |
| --- |

****

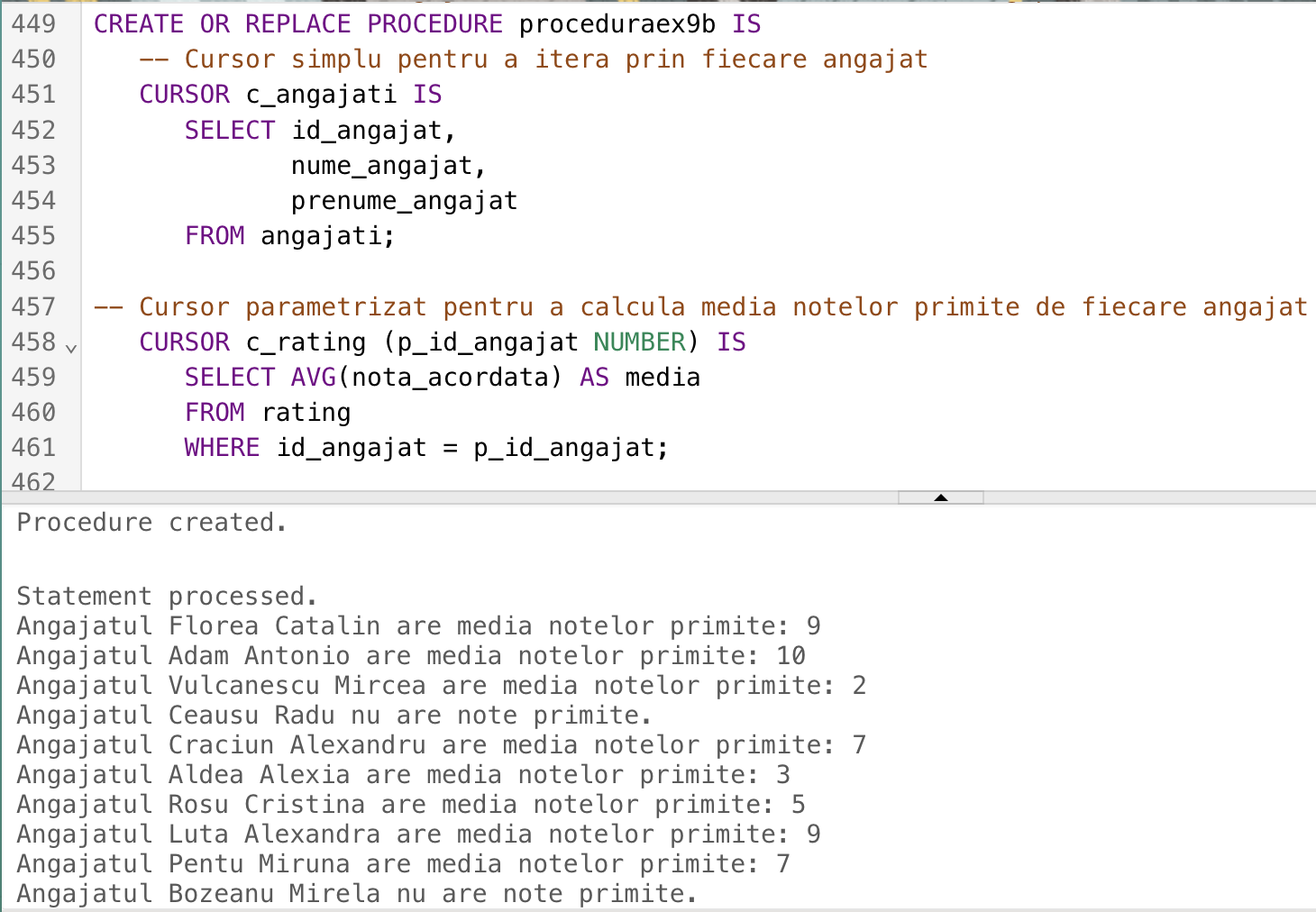
1. **Ilustrarea noțiunilor de PL/SQL învățate**
2. Pentru marfă, o vom lua pe cea care a avut data livrării în luna martie. Apoi, pentru fiecare locație, calculăm cate cutii s-au primit la marfă și de asemenea, salvăm data livrării pentru fiecare cutie de la marfă.

| CREATE OR REPLACE PROCEDURE proceduraex9a AS  TYPE cutii\_comanda IS TABLE OF contor\_marfa.nr\_cutii%TYPE; -- nested table  TYPE id\_magazin IS TABLE OF contor\_marfa.id\_magazin%TYPE INDEX BY PLS\_INTEGER; -- tabela indexata  TYPE marfa\_info IS VARRAY(100) OF VARCHAR2(100);  v\_cutii cutii\_comanda;  v\_locatie\_id id\_magazin;  v\_marfa\_info marfa\_info;  CURSOR c\_marfa\_martie IS  SELECT id\_marfa, data\_livrare  FROM marfa  WHERE EXTRACT(MONTH FROM data\_livrare) = 3;  BEGIN  v\_marfa\_info := marfa\_info();  v\_marfa\_info.extend(100);    FOR m IN c\_marfa\_martie LOOP  SELECT contor\_marfa.nr\_cutii, contor\_marfa.id\_magazin, TO\_CHAR(m.data\_livrare, 'DD-MON-YYYY')  BULK COLLECT INTO v\_cutii, v\_locatie\_id, v\_marfa\_info  FROM contor\_marfa  WHERE contor\_marfa.id\_marfa = m.id\_marfa;    FOR i IN 1..v\_locatie\_id.COUNT LOOP  -- Calculam nr total de cutii pt fiecare locatie  DBMS\_OUTPUT.PUT\_LINE('Magazinul ' || v\_locatie\_id(i) || ' a primit ' || v\_cutii(i) || ' cutii de marfa pe ' || v\_marfa\_info(i) || '.');  END LOOP;  END LOOP;  END proceduraex9a;  /    BEGIN  proceduraex9a;  END;  / |
| --- |



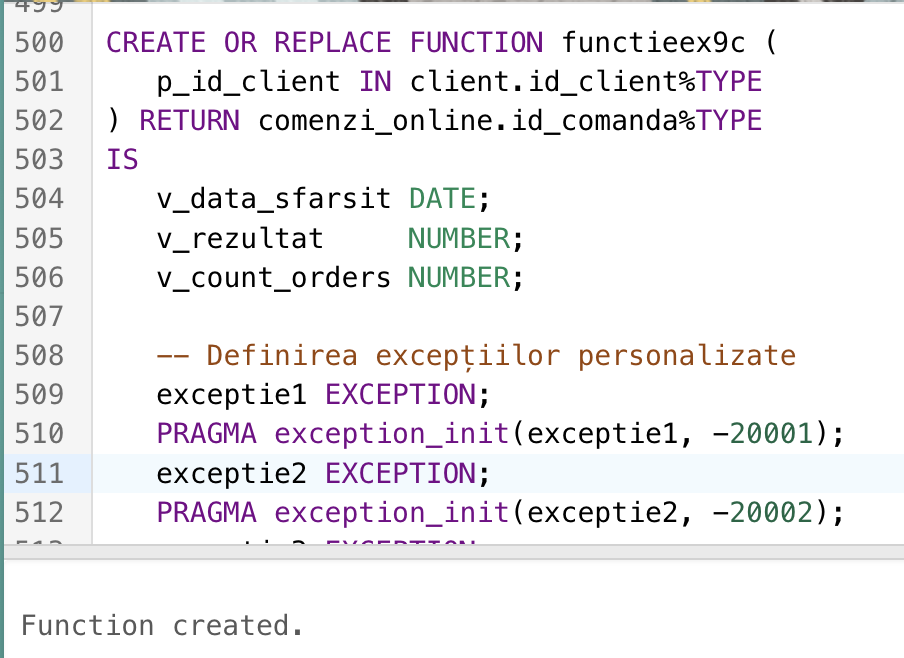
1. Pentru fiecare angajat, afisam media notelor de la rating primite.

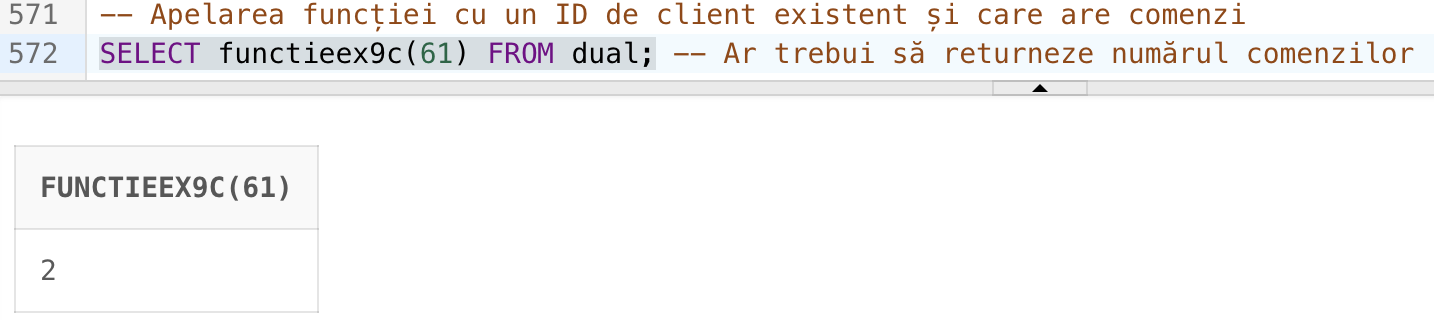
| CREATE OR REPLACE PROCEDURE proceduraex9b IS  -- Cursor simplu pentru a itera prin fiecare angajat  CURSOR c\_angajati IS  SELECT id\_angajat,  nume\_angajat,  prenume\_angajat  FROM angajati;  -- Cursor parametrizat pentru a calcula media notelor primite de fiecare angajat  CURSOR c\_rating (p\_id\_angajat NUMBER) IS  SELECT AVG(nota\_acordata) AS media  FROM rating  WHERE id\_angajat = p\_id\_angajat;  -- Variabile pentru a stoca media notelor primite de fiecare angajat  v\_id\_angajat angajati.id\_angajat%TYPE;  v\_nume angajati.nume\_angajat%TYPE;  v\_prenume angajati.prenume\_angajat%TYPE;  v\_media\_rating rating.nota\_acordata%TYPE;  BEGIN  -- Iteram prin fiecare angajat  FOR r\_angajati IN c\_angajati LOOP  v\_id\_angajat := r\_angajati.id\_angajat;  v\_nume := r\_angajati.nume\_angajat;  v\_prenume := r\_angajati.prenume\_angajat;  -- Calculam media notelor primite de fiecare angajat  OPEN c\_rating(v\_id\_angajat);  FETCH c\_rating INTO v\_media\_rating;  -- Verificăm dacă a fost returnată o medie  IF v\_media\_rating IS NOT NULL THEN  -- Afisam media notelor primite de fiecare angajat  DBMS\_OUTPUT.PUT\_LINE('Angajatul ' || v\_nume || ' ' || v\_prenume || ' are media notelor primite: ' || v\_media\_rating);  ELSE  DBMS\_OUTPUT.PUT\_LINE('Angajatul ' || v\_nume || ' ' || v\_prenume || ' nu are note primite.');  END IF;  CLOSE c\_rating;  END LOOP;  END proceduraex9b;  /  BEGIN  proceduraex9b;  END;  / |
| --- |

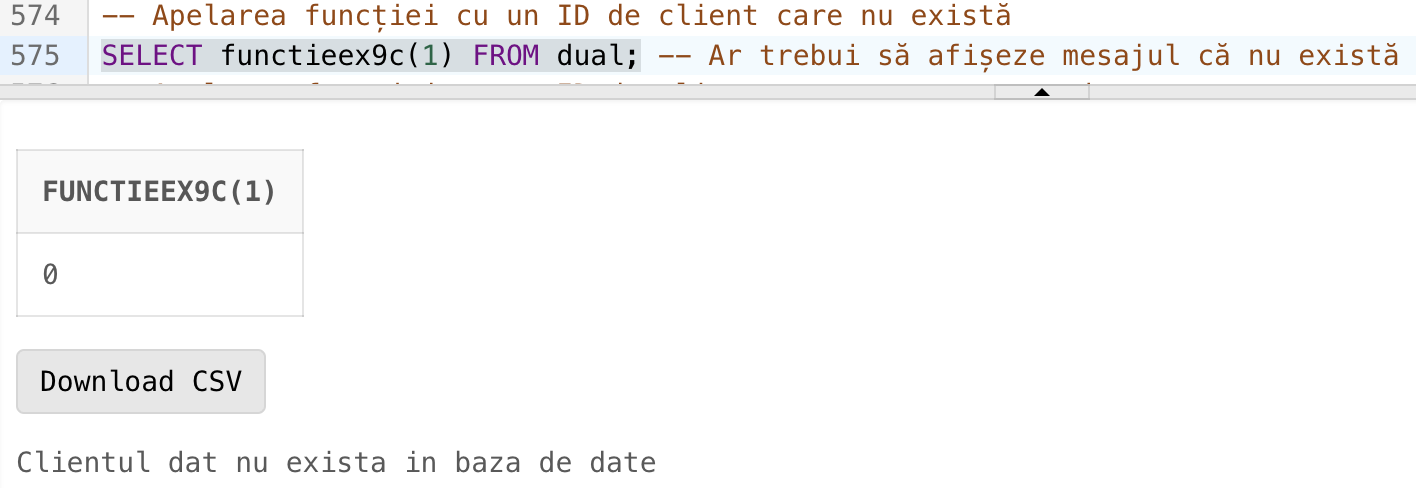


1. Toate comenzile online care se incadreaza in perioada de valabilitate a reducerii (data\_plasare > data\_inceput reducere).

| CREATE OR REPLACE FUNCTION functieex9c (  p\_id\_client IN client.id\_client%TYPE  ) RETURN comenzi\_online.id\_comanda%TYPE  IS  v\_data\_sfarsit DATE;  v\_rezultat NUMBER;  v\_count\_orders NUMBER;  -- Definirea excepțiilor personalizate  exceptie1 EXCEPTION;  PRAGMA exception\_init(exceptie1, -20001);  exceptie2 EXCEPTION;  PRAGMA exception\_init(exceptie2, -20002);  exceptie3 EXCEPTION;  PRAGMA exception\_init(exceptie3, -20003);  BEGIN  -- Verificarea existenței clientului  SELECT COUNT(\*)  INTO v\_rezultat  FROM client  WHERE id\_client = p\_id\_client;  IF v\_rezultat = 0 THEN  RAISE exceptie3; -- Clientul nu există  ELSIF v\_rezultat > 1 THEN  RAISE exceptie2; -- Există mai mulți clienți cu același ID  END IF;  -- Calcularea datei de sfârșit a perioadei de reducere  SELECT ADD\_MONTHS(data\_inceput, perioada\_valabilitate \* 12)  INTO v\_data\_sfarsit  FROM reducere  WHERE id\_client = p\_id\_client;  -- Verificare dacă există o reducere validă  IF v\_data\_sfarsit IS NULL THEN  RAISE exceptie1; -- Nu există reducere  END IF;  -- Extrage comenzile online pentru client în perioada de reducere  SELECT COUNT(co.id\_comanda)  INTO v\_count\_orders  FROM comenzi\_online co  JOIN reducere r ON co.id\_client = r.id\_client  WHERE co.id\_client = p\_id\_client  AND co.data\_plasare BETWEEN r.data\_inceput AND v\_data\_sfarsit;  IF v\_count\_orders = 0 THEN  RAISE exceptie1; -- Nu există comenzi online  END IF;  -- Returnează ID-ul comenzii (poate fi ajustat în funcție de nevoile tale)  RETURN v\_count\_orders;  EXCEPTION  WHEN exceptie1 THEN  DBMS\_OUTPUT.PUT\_LINE('Nu exista nicio comanda plasata pentru clientul dat');  RETURN 0;  WHEN exceptie2 THEN  DBMS\_OUTPUT.PUT\_LINE('Exista mai multi clienti cu acelasi id in baza de date');  RETURN 0;  WHEN exceptie3 THEN  DBMS\_OUTPUT.PUT\_LINE('Clientul dat nu exista in baza de date');  RETURN 0;  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare: ' || SQLERRM);  RETURN 0;  END functieex9c;  /  -- Apelarea funcției cu un ID de client existent și care are comenzi  SELECT functieex9c(61) FROM dual; -- Ar trebui să returneze numărul comenzilor  -- Apelarea funcției cu un ID de client care nu există  SELECT functieex9c(1) FROM dual; -- Ar trebui să afișeze mesajul că nu există  -- Apelarea funcției cu un ID de client care nu are comenzi  SELECT functieex9c(101) FROM dual; -- Ar trebui să afișeze mesajul că nu există comenzi |
| --- |

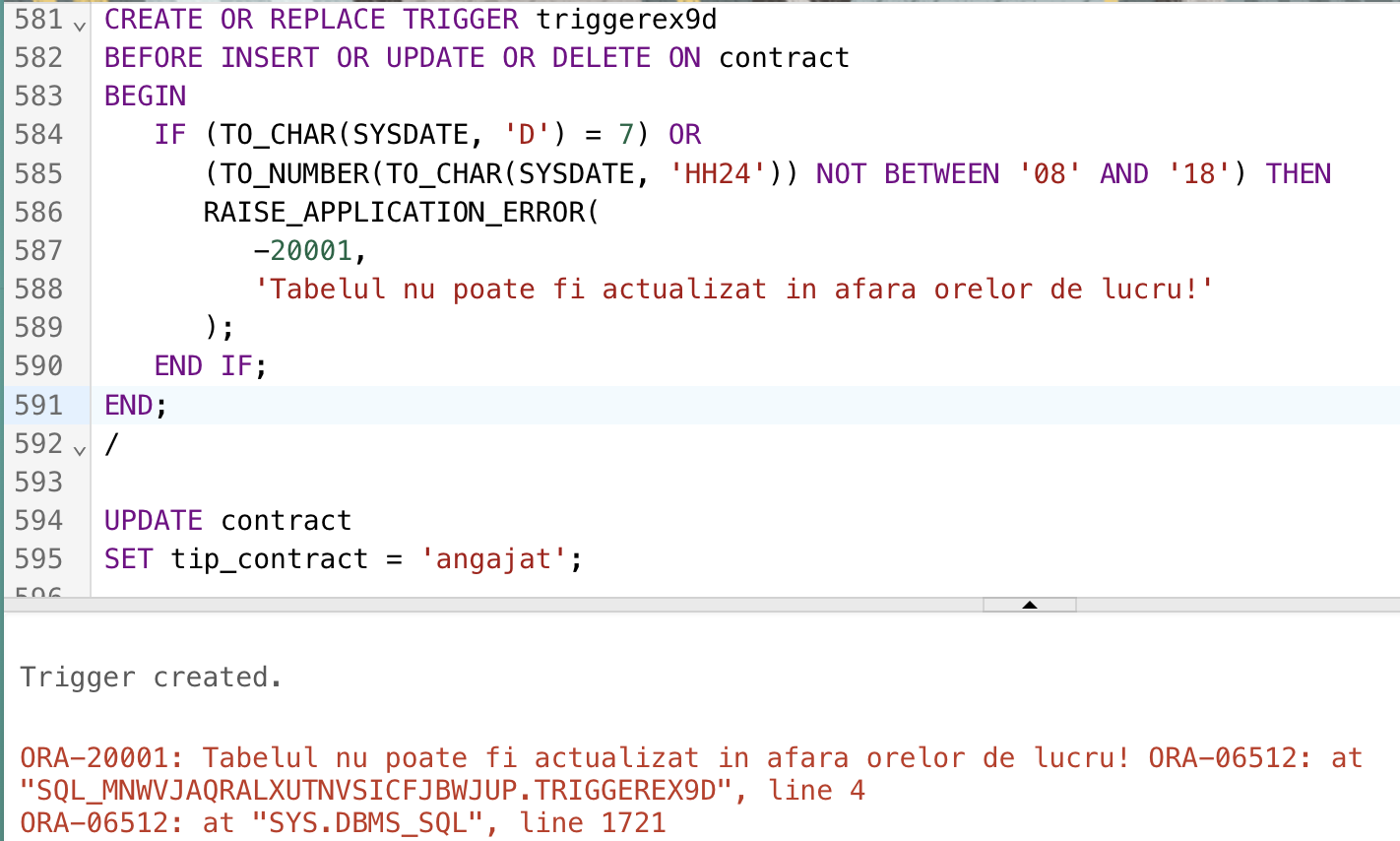






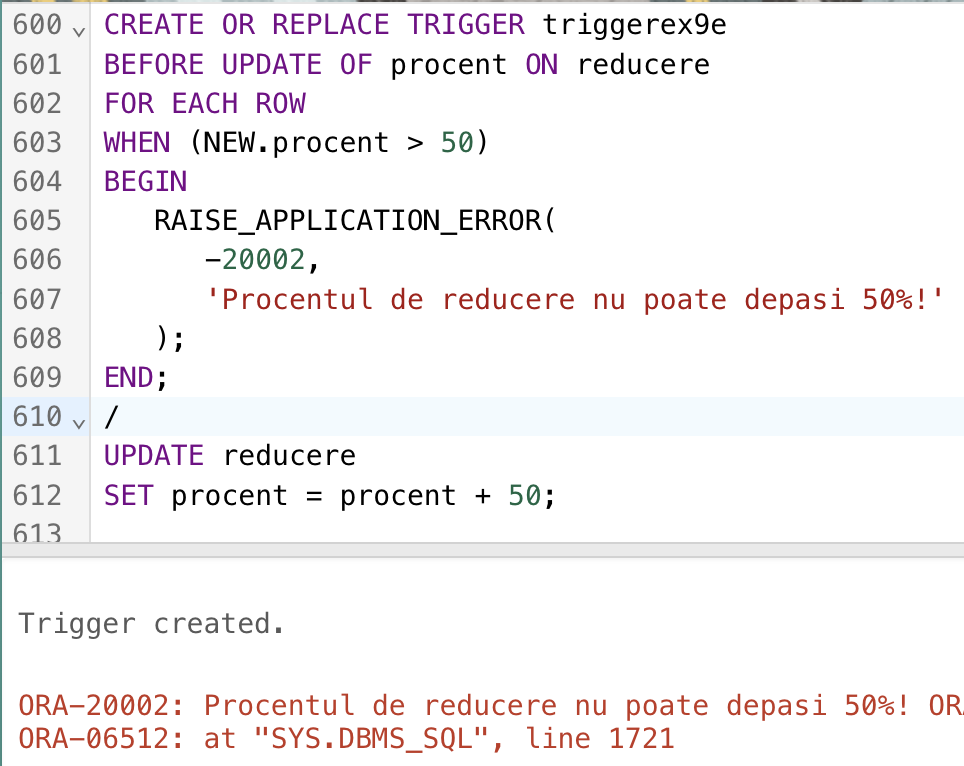
1. Trigger care permite lucrul asupra tabelului CONTRACT doar în intervalul 8-18, de luni pana sambata.

| CREATE OR REPLACE TRIGGER triggerex9d  BEFORE INSERT OR UPDATE OR DELETE ON contract  BEGIN  IF (TO\_CHAR(SYSDATE, 'D') = 7) OR  (TO\_NUMBER(TO\_CHAR(SYSDATE, 'HH24')) NOT BETWEEN '08' AND '18') THEN  RAISE\_APPLICATION\_ERROR(  -20001,  'Tabelul nu poate fi actualizat in afara orelor de lucru!'  );  END IF;  END;  /  UPDATE contract  SET tip\_contract = 'angajat'; |
| --- |



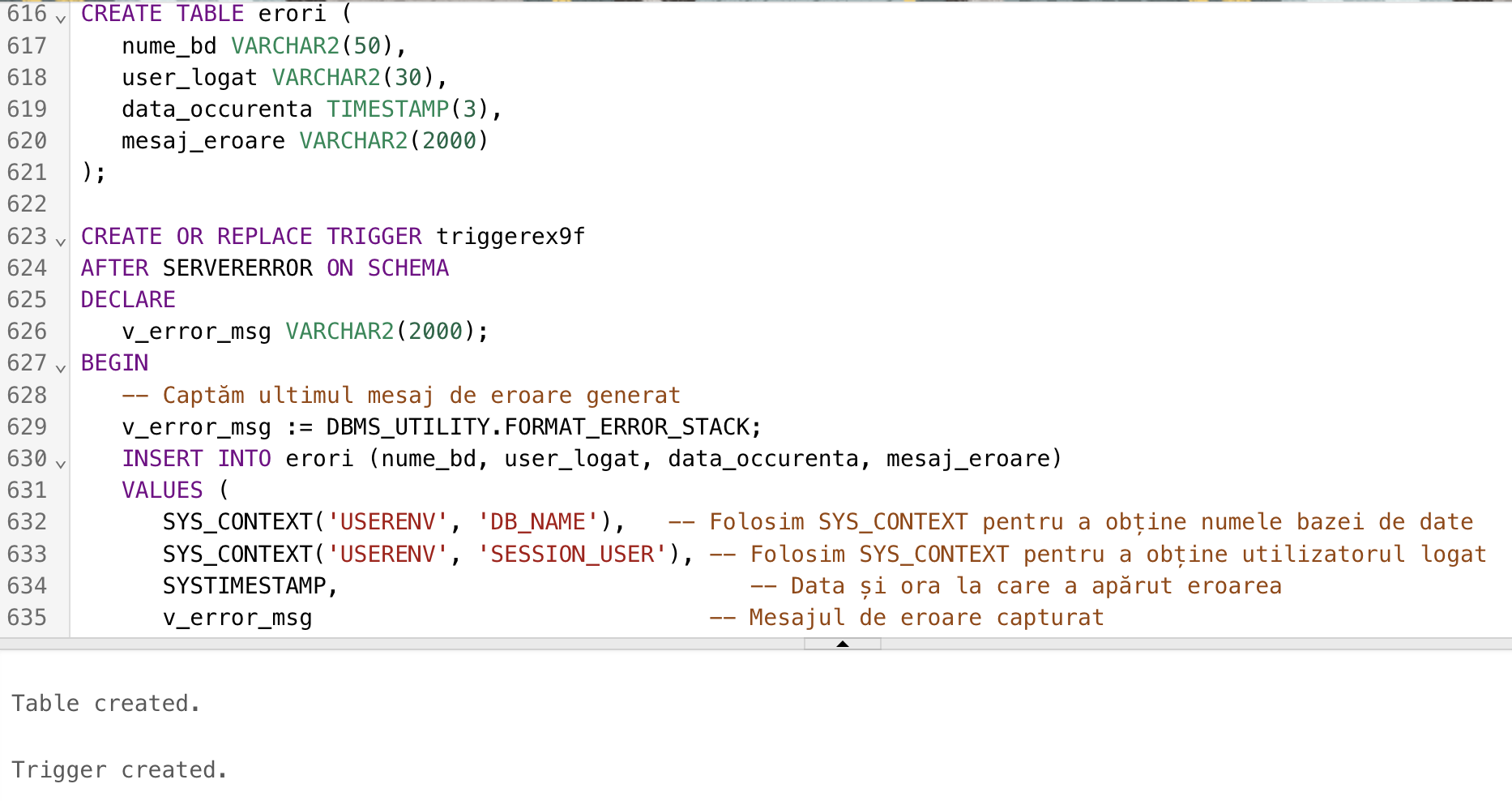
1. Trigger care sa nu permita ca reducerea unui client sa depaseasca 50%.

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



1. Trigger dintr-o comanda LDD

| CREATE TABLE erori (  nume\_bd VARCHAR2(50),  user\_logat VARCHAR2(30),  data\_occurenta TIMESTAMP(3),  mesaj\_eroare VARCHAR2(2000)  );  CREATE OR REPLACE TRIGGER triggerex9f  AFTER SERVERERROR ON SCHEMA  DECLARE  v\_error\_msg VARCHAR2(2000);  BEGIN  -- Captăm ultimul mesaj de eroare generat  v\_error\_msg := DBMS\_UTILITY.FORMAT\_ERROR\_STACK;  INSERT INTO erori (nume\_bd, user\_logat, data\_occurenta, mesaj\_eroare)  VALUES (  SYS\_CONTEXT('USERENV', 'DB\_NAME'), -- Folosim SYS\_CONTEXT pentru a obține numele bazei de date  SYS\_CONTEXT('USERENV', 'SESSION\_USER'), -- Folosim SYS\_CONTEXT pentru a obține utilizatorul logat  SYSTIMESTAMP, -- Data și ora la care a apărut eroarea  v\_error\_msg -- Mesajul de eroare capturat  );  END;  /  BEGIN  EXECUTE IMMEDIATE 'ALTER TABLE a DROP COLUMN nonexistent\_column';  END;  SELECT \* FROM erori; |
| --- |



1. Un pachet care să conțină toate procedurile, funcțiile si trigger-ii creați/create mai sus.

| CREATE OR REPLACE PACKAGE pachet\_marfa AS  PROCEDURE proceduraex9a;  PROCEDURE proceduraex9b;  FUNCTION functieex9c(p\_id\_client IN client.id\_client%TYPE) RETURN comenzi\_online.id\_comanda%TYPE;  END pachet\_marfa;  /  CREATE OR REPLACE PACKAGE BODY pachet\_marfa AS  -- Procedura a)  PROCEDURE proceduraex9a AS  TYPE cutii\_comanda IS TABLE OF contor\_marfa.nr\_cutii%TYPE; -- nested table  TYPE id\_magazin IS TABLE OF contor\_marfa.id\_magazin%TYPE INDEX BY PLS\_INTEGER; -- tabela indexata  TYPE marfa\_info IS VARRAY(100) OF VARCHAR2(100);  v\_cutii cutii\_comanda;  v\_locatie\_id id\_magazin;  v\_marfa\_info marfa\_info;  CURSOR c\_marfa\_martie IS  SELECT id\_marfa, data\_livrare  FROM marfa  WHERE EXTRACT(MONTH FROM data\_livrare) = 3;    BEGIN  v\_marfa\_info := marfa\_info();  v\_marfa\_info.extend(100);    BEGIN  -- Actualizare pentru a declanșa triggerul în afara orelor de lucru  UPDATE contract  SET tip\_contract = 'nou\_tip'  WHERE id\_contract = 1; -- Înlocuiește cu un ID valid existent în tabelul `contract`  EXCEPTION  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM); -- Afișează mesajul de eroare din trigger  END;  FOR m IN c\_marfa\_martie LOOP  SELECT contor\_marfa.nr\_cutii, contor\_marfa.id\_magazin, TO\_CHAR(m.data\_livrare, 'DD-MON-YYYY')  BULK COLLECT INTO v\_cutii, v\_locatie\_id, v\_marfa\_info  FROM contor\_marfa  WHERE contor\_marfa.id\_marfa = m.id\_marfa;    FOR i IN 1..v\_locatie\_id.COUNT LOOP  -- Calculam nr total de cutii pt fiecare locatie  DBMS\_OUTPUT.PUT\_LINE('Magazinul ' || v\_locatie\_id(i) || ' a primit ' || v\_cutii(i) || ' cutii de marfa pe ' || v\_marfa\_info(i) || '.');  END LOOP;  END LOOP;  END proceduraex9a;  -- Procedura b)  PROCEDURE proceduraex9b IS  -- Cursor simplu pentru a itera prin fiecare angajat  CURSOR c\_angajati IS  SELECT id\_angajat,  nume\_angajat,  prenume\_angajat  FROM angajati;    -- Cursor parametrizat pentru a calcula media notelor primite de fiecare angajat  CURSOR c\_rating (p\_id\_angajat NUMBER) IS  SELECT AVG(nota\_acordata) AS media  FROM rating  WHERE id\_angajat = p\_id\_angajat;  -- Variabile pentru a stoca media notelor primite de fiecare angajat  v\_id\_angajat angajati.id\_angajat%TYPE;  v\_nume angajati.nume\_angajat%TYPE;  v\_prenume angajati.prenume\_angajat%TYPE;  v\_media\_rating rating.nota\_acordata%TYPE;  BEGIN  -- Iteram prin fiecare angajat  FOR r\_angajati IN c\_angajati LOOP  v\_id\_angajat := r\_angajati.id\_angajat;  v\_nume := r\_angajati.nume\_angajat;  v\_prenume := r\_angajati.prenume\_angajat;    -- Calculam media notelor primite de fiecare angajat  OPEN c\_rating(v\_id\_angajat);  FETCH c\_rating INTO v\_media\_rating;    -- Verificăm dacă a fost returnată o medie  IF v\_media\_rating IS NOT NULL THEN  -- Afisam media notelor primite de fiecare angajat  DBMS\_OUTPUT.PUT\_LINE('Angajatul ' || v\_nume || ' ' || v\_prenume || ' are media notelor primite: ' || v\_media\_rating);  ELSE  DBMS\_OUTPUT.PUT\_LINE('Angajatul ' || v\_nume || ' ' || v\_prenume || ' nu are note primite.');  END IF;  CLOSE c\_rating;  END LOOP;  END proceduraex9b;  --Funcția c)  FUNCTION functieex9c (  p\_id\_client IN client.id\_client%TYPE  ) RETURN comenzi\_online.id\_comanda%TYPE  IS  v\_data\_sfarsit DATE;  v\_rezultat NUMBER;  v\_count\_orders NUMBER;    -- Definirea excepțiilor personalizate  exceptie1 EXCEPTION;  PRAGMA exception\_init(exceptie1, -20001);  exceptie2 EXCEPTION;  PRAGMA exception\_init(exceptie2, -20002);  exceptie3 EXCEPTION;  PRAGMA exception\_init(exceptie3, -20003);    BEGIN  UPDATE reducere SET procent = 60 WHERE id\_client = 62;  -- Verificarea existenței clientului  SELECT COUNT(\*)  INTO v\_rezultat  FROM client  WHERE id\_client = p\_id\_client;    IF v\_rezultat = 0 THEN  RAISE exceptie3; -- Clientul nu există  ELSIF v\_rezultat > 1 THEN  RAISE exceptie2; -- Există mai mulți clienți cu același ID  END IF;    -- Calcularea datei de sfârșit a perioadei de reducere  SELECT ADD\_MONTHS(data\_inceput, perioada\_valabilitate \* 12)  INTO v\_data\_sfarsit  FROM reducere  WHERE id\_client = p\_id\_client;    -- Verificare dacă există o reducere validă  IF v\_data\_sfarsit IS NULL THEN  RAISE exceptie1; -- Nu există reducere  END IF;    -- Extrage comenzile online pentru client în perioada de reducere  SELECT COUNT(co.id\_comanda)  INTO v\_count\_orders  FROM comenzi\_online co  JOIN reducere r ON co.id\_client = r.id\_client  WHERE co.id\_client = p\_id\_client  AND co.data\_plasare BETWEEN r.data\_inceput AND v\_data\_sfarsit;    IF v\_count\_orders = 0 THEN  RAISE exceptie1; -- Nu există comenzi online  END IF;    -- Returnează ID-ul comenzii (poate fi ajustat în funcție de nevoile tale)  RETURN v\_count\_orders;    EXCEPTION  WHEN exceptie1 THEN  DBMS\_OUTPUT.PUT\_LINE('Nu exista nicio comanda plasata pentru clientul dat');  RETURN 0;  WHEN exceptie2 THEN  DBMS\_OUTPUT.PUT\_LINE('Exista mai multi clienti cu acelasi id in baza de date');  RETURN 0;  WHEN exceptie3 THEN  DBMS\_OUTPUT.PUT\_LINE('Clientul dat nu exista in baza de date');  RETURN 0;  WHEN OTHERS THEN  DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare: ' || SQLERRM);  RETURN 0;  END functieex9c;  END;  /  BEGIN  pachet\_marfa.proceduraex9a;  pachet\_marfa.proceduraex9b;  DBMS\_OUTPUT.PUT\_LINE('Numărul comenzilor online: ' || pachet\_marfa.functieex9c(63));  END;  / |
| --- |

