**Proiect SGBD – Firma IT**

Realizat

De

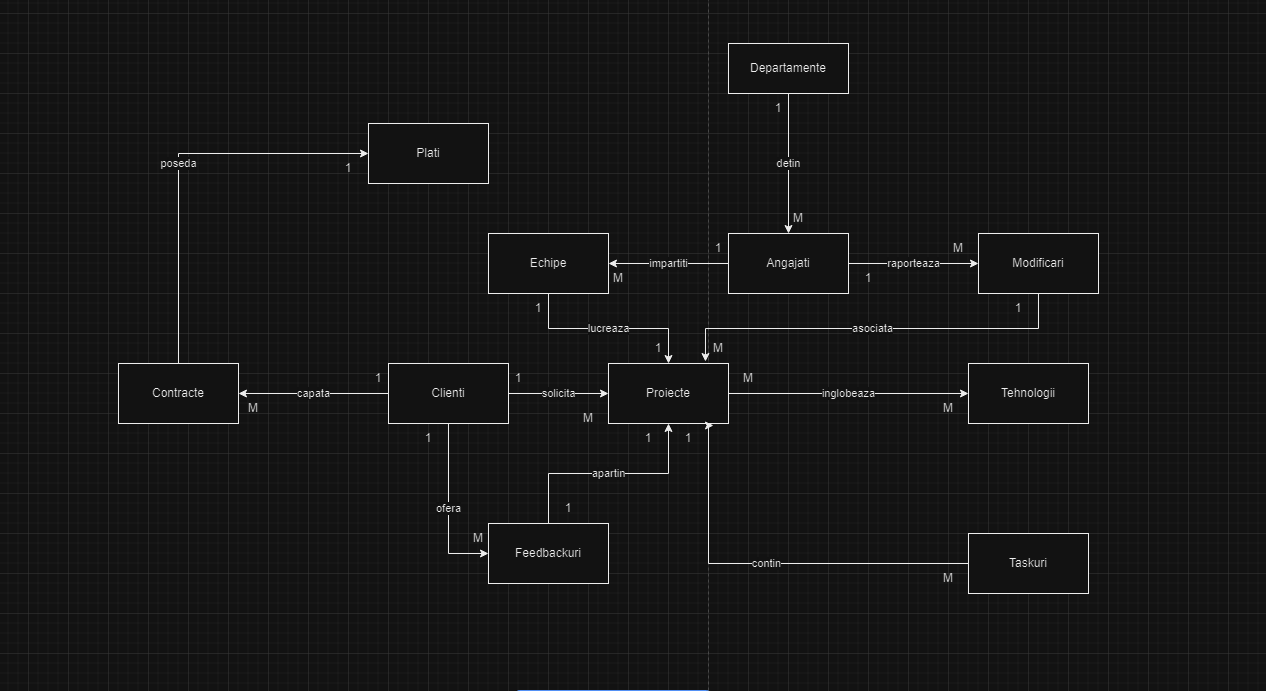
Tudur Rares

Grupa 242

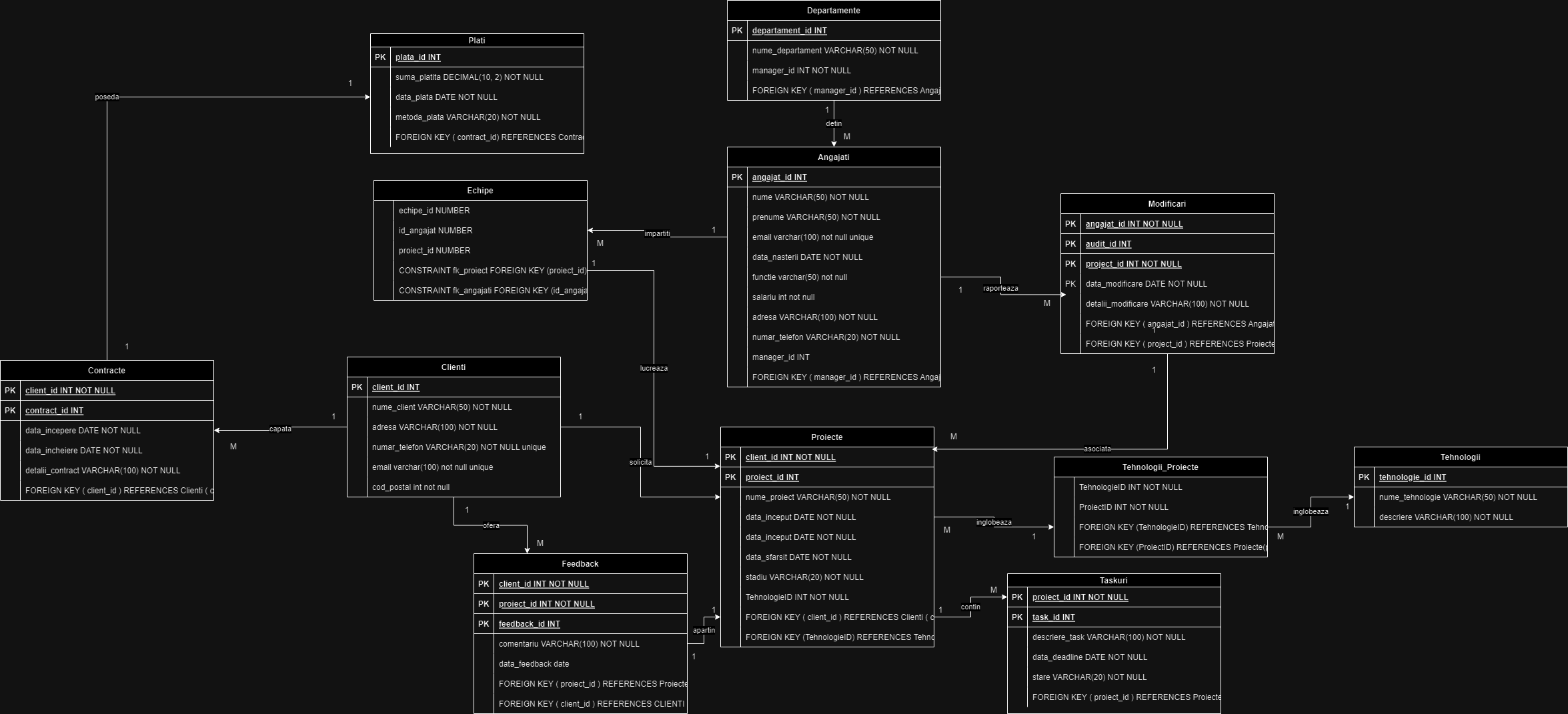
1. Prezentați pe scurt baza de date (utilitatea ei).

In cadrul acestui proiect voi crea baza de date a unei firme de IT. Firma va contine angajati, departamente, clienti, proiecte, echipe, tehnologii, plati, contracte, feedback, si taskuri. Voi folosi o tabela asociativa pentru a stabili tehnologiile folosite la fiecare proiect.

2.Realizați diagrama entitate-relație (ERD): entitățile, relațiile și atributele trebuie definite în limba română (vezi curs SGBD / model de diagrama ERD; nu se va accepta alt format).

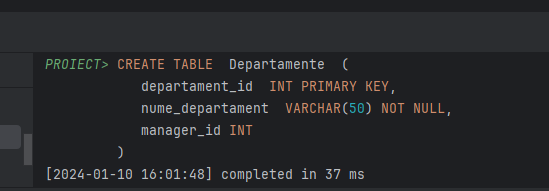


3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare: entitățile, relațiile și atributele trebuie definite în limba română.

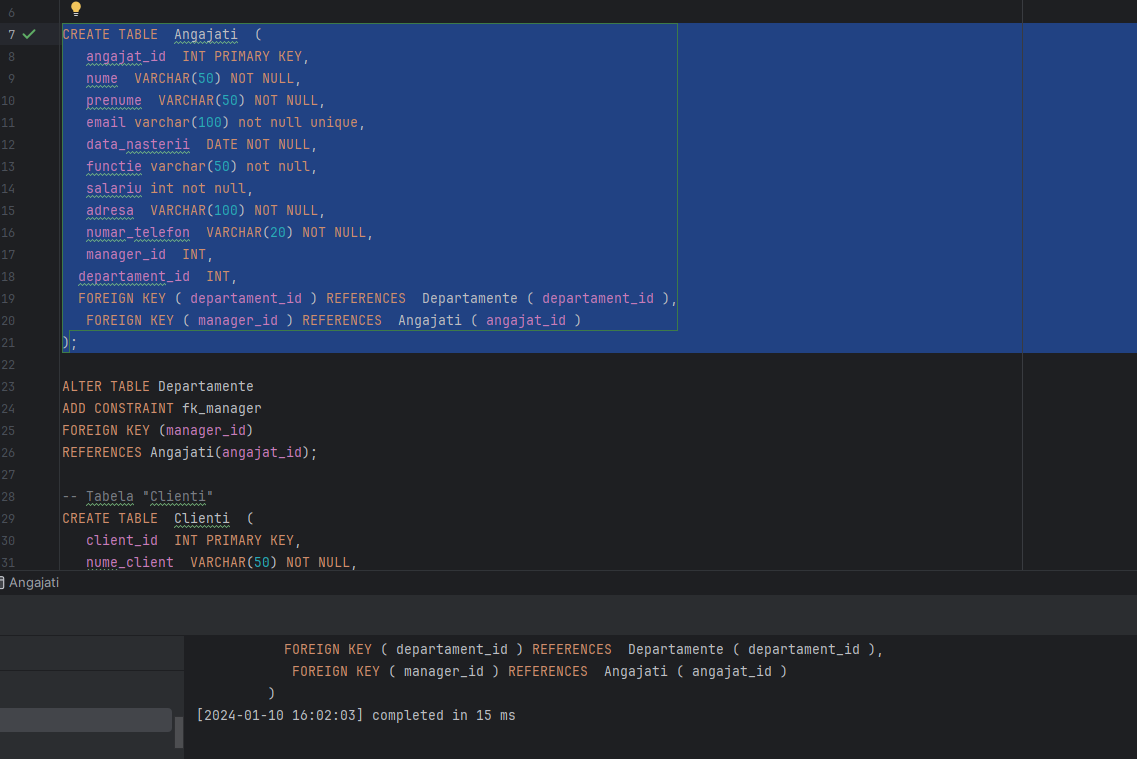


4. Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, definind toate constrângerile de integritate necesare (chei primare, cheile externe etc).

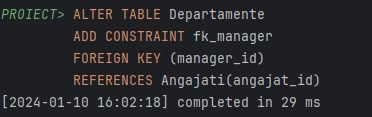
CREATE TABLE Departamente (  
 departament\_id INT PRIMARY KEY,  
 nume\_departament VARCHAR(50) NOT NULL,  
 manager\_id INT  
);



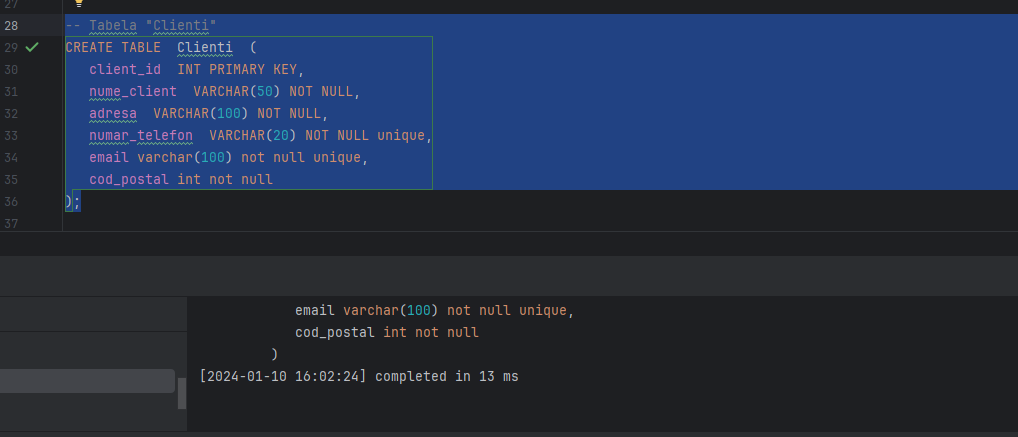
CREATE TABLE Angajati (  
 angajat\_id INT PRIMARY KEY,  
 nume VARCHAR(50) NOT NULL,  
 prenume VARCHAR(50) NOT NULL,  
 email varchar(100) not null unique,  
 data\_nasterii DATE NOT NULL,  
 functie varchar(50) not null,  
 salariu int not null,  
 adresa VARCHAR(100) NOT NULL,  
 numar\_telefon VARCHAR(20) NOT NULL,  
 manager\_id INT,  
 departament\_id INT,  
 FOREIGN KEY ( departament\_id ) REFERENCES Departamente ( departament\_id ),  
 FOREIGN KEY ( manager\_id ) REFERENCES Angajati ( angajat\_id )  
);



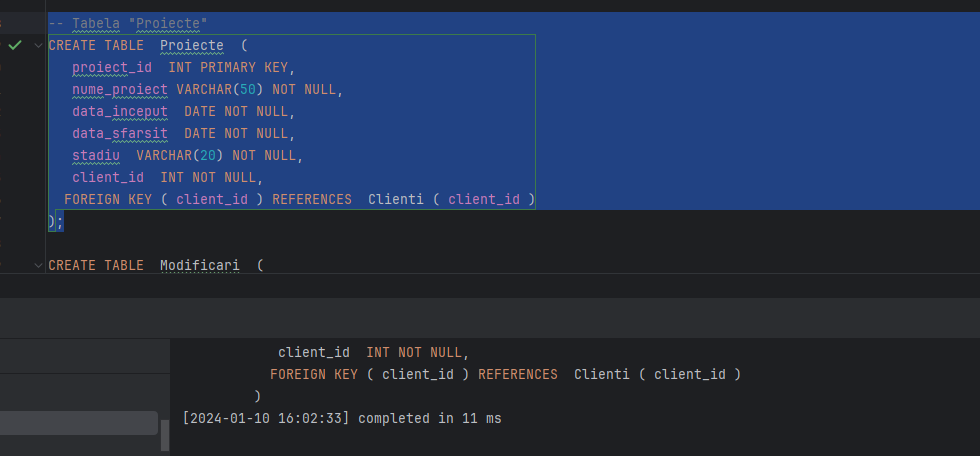
ALTER TABLE Departamente  
ADD CONSTRAINT fk\_manager  
FOREIGN KEY (manager\_id)  
REFERENCES Angajati(angajat\_id);



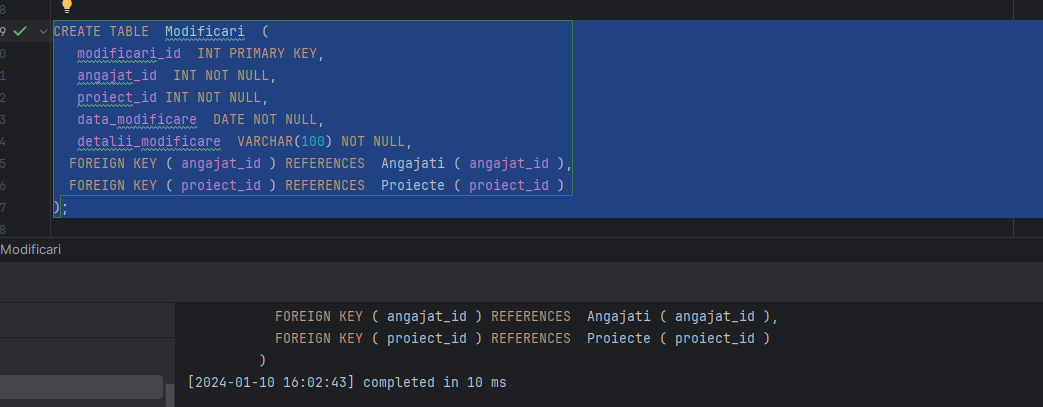
CREATE TABLE Clienti (  
 client\_id INT PRIMARY KEY,  
 nume\_client VARCHAR(50) NOT NULL,  
 adresa VARCHAR(100) NOT NULL,  
 numar\_telefon VARCHAR(20) NOT NULL unique,  
 email varchar(100) not null unique,  
 cod\_postal int not null  
);



CREATE TABLE Proiecte (  
 proiect\_id INT PRIMARY KEY,  
 nume\_proiect VARCHAR(50) NOT NULL,  
 data\_inceput DATE NOT NULL,  
 data\_sfarsit DATE NOT NULL,  
 stadiu VARCHAR(20) NOT NULL,  
 client\_id INT NOT NULL,  
 FOREIGN KEY ( client\_id ) REFERENCES Clienti ( client\_id )  
);



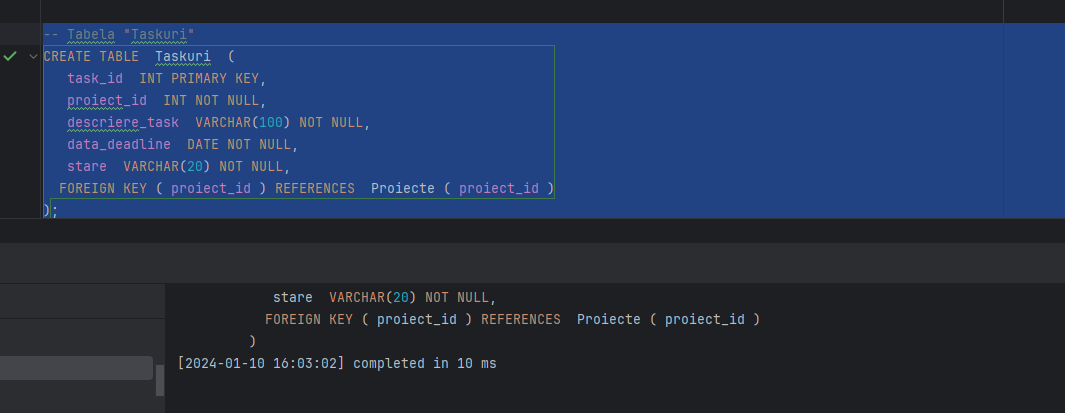
CREATE TABLE Modificari (  
 modificari\_id INT PRIMARY KEY,  
 angajat\_id INT NOT NULL,  
 proiect\_id INT NOT NULL,  
 data\_modificare DATE NOT NULL,  
 detalii\_modificare VARCHAR(100) NOT NULL,  
 FOREIGN KEY ( angajat\_id ) REFERENCES Angajati ( angajat\_id ),  
 FOREIGN KEY ( proiect\_id ) REFERENCES Proiecte ( proiect\_id )  
);



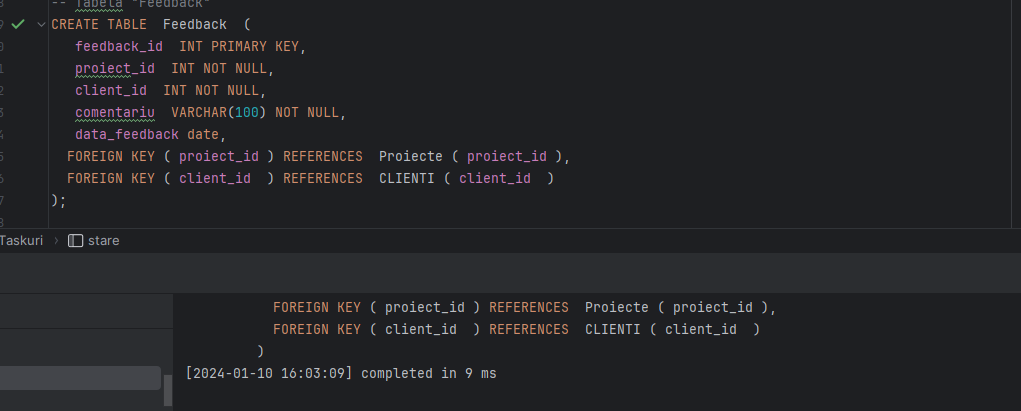
CREATE TABLE Tehnologii (  
 tehnologie\_id INT PRIMARY KEY,  
 nume\_tehnologie VARCHAR(50) NOT NULL,  
 descriere VARCHAR(100) NOT NULL  
 );



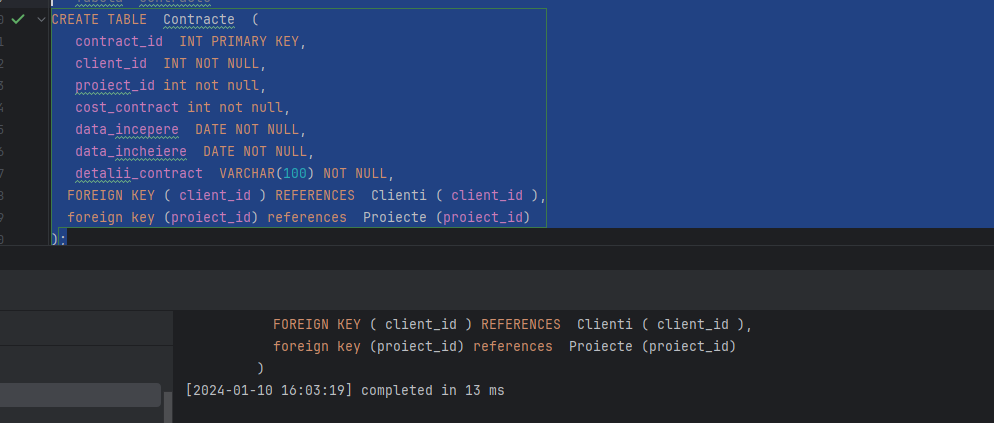
CREATE TABLE Taskuri (  
 task\_id INT PRIMARY KEY,  
 proiect\_id INT NOT NULL,  
 descriere\_task VARCHAR(100) NOT NULL,  
 data\_deadline DATE NOT NULL,  
 stare VARCHAR(20) NOT NULL,  
 FOREIGN KEY ( proiect\_id ) REFERENCES Proiecte ( proiect\_id )  
);



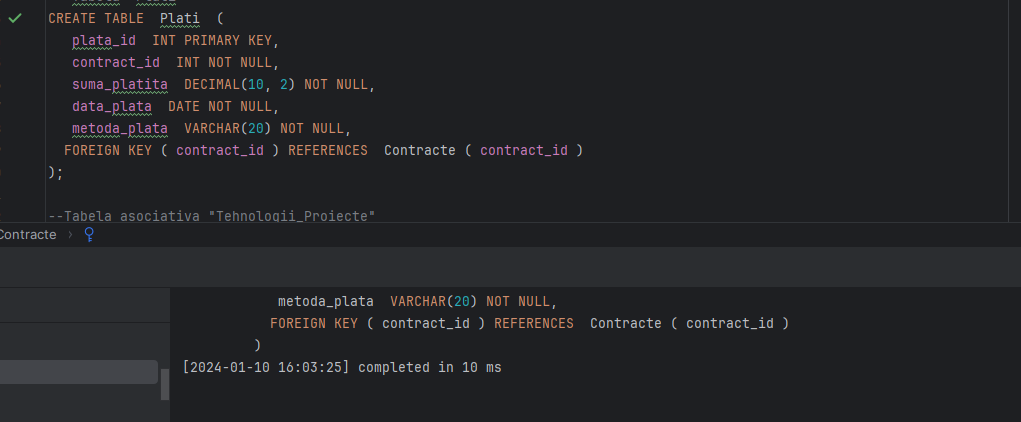
CREATE TABLE Feedback (  
 feedback\_id INT PRIMARY KEY,  
 proiect\_id INT NOT NULL,  
 client\_id INT NOT NULL,  
 comentariu VARCHAR(100) NOT NULL,  
 data\_feedback date,  
 FOREIGN KEY ( proiect\_id ) REFERENCES Proiecte ( proiect\_id ),  
 FOREIGN KEY ( client\_id ) REFERENCES CLIENTI ( client\_id )  
);



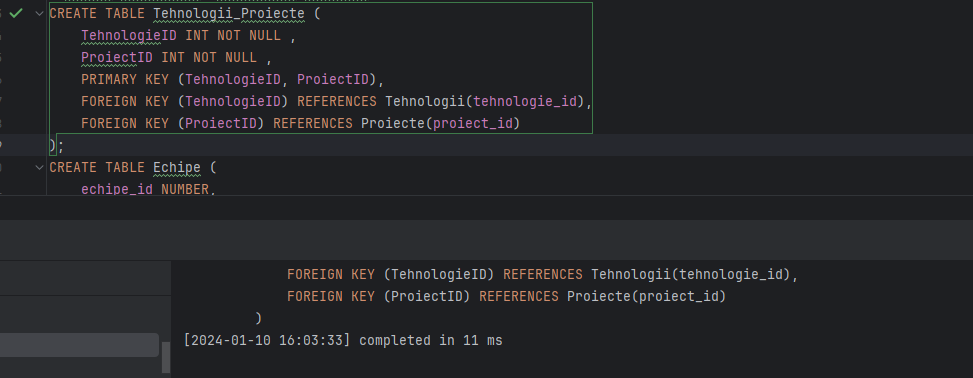
CREATE TABLE Contracte (  
 contract\_id INT PRIMARY KEY,  
 client\_id INT NOT NULL,  
 proiect\_id int not null,  
 cost\_contract int not null,  
 data\_incepere DATE NOT NULL,  
 data\_incheiere DATE NOT NULL,  
 detalii\_contract VARCHAR(100) NOT NULL,  
 FOREIGN KEY ( client\_id ) REFERENCES Clienti ( client\_id ),  
 foreign key (proiect\_id) references Proiecte (proiect\_id)  
);



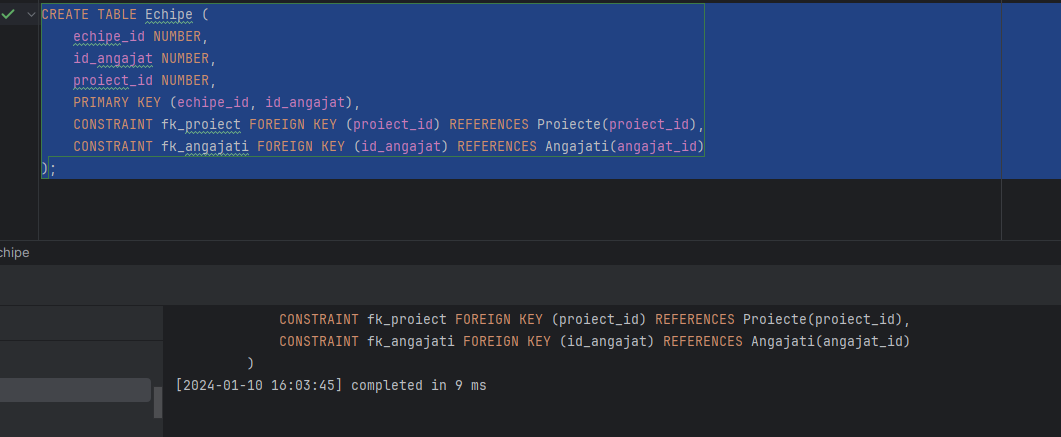
CREATE TABLE Plati (  
 plata\_id INT PRIMARY KEY,  
 contract\_id INT NOT NULL,  
 suma\_platita DECIMAL(10, 2) NOT NULL,  
 data\_plata DATE NOT NULL,  
 metoda\_plata VARCHAR(20) NOT NULL,  
 FOREIGN KEY ( contract\_id ) REFERENCES Contracte ( contract\_id )  
);



CREATE TABLE Tehnologii\_Proiecte (  
 TehnologieID INT NOT NULL ,  
 ProiectID INT NOT NULL ,  
 PRIMARY KEY (TehnologieID, ProiectID),  
 FOREIGN KEY (TehnologieID) REFERENCES Tehnologii(tehnologie\_id),  
 FOREIGN KEY (ProiectID) REFERENCES Proiecte(proiect\_id)  
);

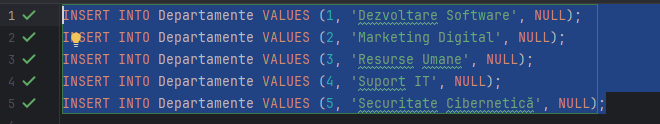


CREATE TABLE Echipe (  
 echipe\_id NUMBER,  
 id\_angajat NUMBER,  
 proiect\_id NUMBER,  
 PRIMARY KEY (echipe\_id, id\_angajat),  
 CONSTRAINT fk\_proiect FOREIGN KEY (proiect\_id) REFERENCES Proiecte(proiect\_id),  
 CONSTRAINT fk\_angajati FOREIGN KEY (id\_angajat) REFERENCES Angajati(angajat\_id)  
);

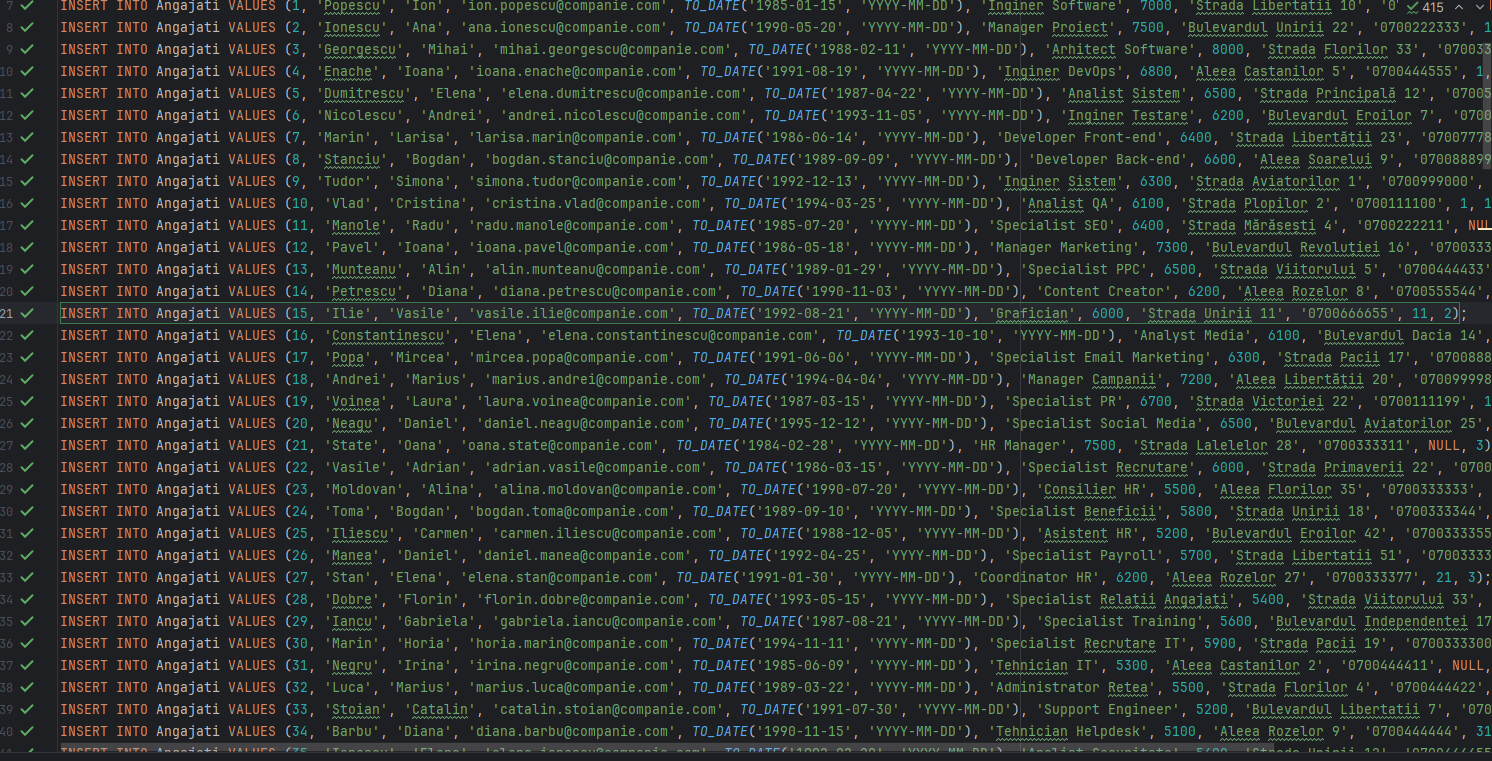


5. Adăugați informații coerente în tabelele create (minim 5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

INSERT INTO Departamente VALUES (1, 'Dezvoltare Software', NULL);  
INSERT INTO Departamente VALUES (2, 'Marketing Digital', NULL);  
INSERT INTO Departamente VALUES (3, 'Resurse Umane', NULL);  
INSERT INTO Departamente VALUES (4, 'Suport IT', NULL);  
INSERT INTO Departamente VALUES (5, 'Securitate Cibernetică', NULL);

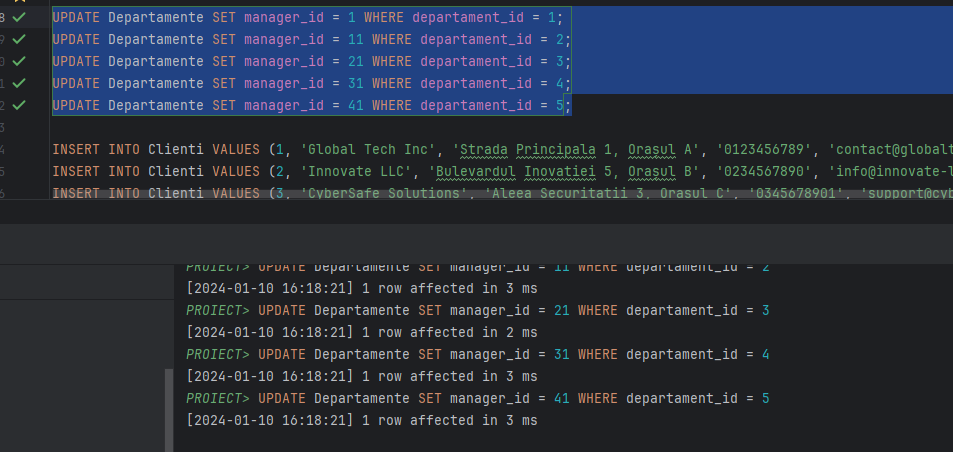


INSERT INTO Angajati VALUES (1, 'Popescu', 'Ion', 'ion.popescu@companie.com', *TO\_DATE*('1985-01-15', 'YYYY-MM-DD'), 'Inginer Software', 7000, 'Strada Libertatii 10', '0700111222',NULL, 1);  
INSERT INTO Angajati VALUES (2, 'Ionescu', 'Ana', 'ana.ionescu@companie.com', *TO\_DATE*('1990-05-20', 'YYYY-MM-DD'), 'Manager Proiect', 7500, 'Bulevardul Unirii 22', '0700222333', 1, 1);  
INSERT INTO Angajati VALUES (3, 'Georgescu', 'Mihai', 'mihai.georgescu@companie.com', *TO\_DATE*('1988-02-11', 'YYYY-MM-DD'), 'Arhitect Software', 8000, 'Strada Florilor 33', '0700333444', 1, 1);  
INSERT INTO Angajati VALUES (4, 'Enache', 'Ioana', 'ioana.enache@companie.com', *TO\_DATE*('1991-08-19', 'YYYY-MM-DD'), 'Inginer DevOps', 6800, 'Aleea Castanilor 5', '0700444555', 1, 1);  
INSERT INTO Angajati VALUES (5, 'Dumitrescu', 'Elena', 'elena.dumitrescu@companie.com', *TO\_DATE*('1987-04-22', 'YYYY-MM-DD'), 'Analist Sistem', 6500, 'Strada Principală 12', '0700555666', 1, 1);  
INSERT INTO Angajati VALUES (6, 'Nicolescu', 'Andrei', 'andrei.nicolescu@companie.com', *TO\_DATE*('1993-11-05', 'YYYY-MM-DD'), 'Inginer Testare', 6200, 'Bulevardul Eroilor 7', '0700666777', 1, 1);  
INSERT INTO Angajati VALUES (7, 'Marin', 'Larisa', 'larisa.marin@companie.com', *TO\_DATE*('1986-06-14', 'YYYY-MM-DD'), 'Developer Front-end', 6400, 'Strada Libertății 23', '0700777888', 1, 1);  
INSERT INTO Angajati VALUES (8, 'Stanciu', 'Bogdan', 'bogdan.stanciu@companie.com', *TO\_DATE*('1989-09-09', 'YYYY-MM-DD'), 'Developer Back-end', 6600, 'Aleea Soarelui 9', '0700888999', 1, 1);  
INSERT INTO Angajati VALUES (9, 'Tudor', 'Simona', 'simona.tudor@companie.com', *TO\_DATE*('1992-12-13', 'YYYY-MM-DD'), 'Inginer Sistem', 6300, 'Strada Aviatorilor 1', '0700999000', 1, 1);  
INSERT INTO Angajati VALUES (10, 'Vlad', 'Cristina', 'cristina.vlad@companie.com', *TO\_DATE*('1994-03-25', 'YYYY-MM-DD'), 'Analist QA', 6100, 'Strada Plopilor 2', '0700111100', 1, 1);  
INSERT INTO Angajati VALUES (11, 'Manole', 'Radu', 'radu.manole@companie.com', *TO\_DATE*('1985-07-20', 'YYYY-MM-DD'), 'Specialist SEO', 6400, 'Strada Mărășești 4', '0700222211', NULL, 2);  
INSERT INTO Angajati VALUES (12, 'Pavel', 'Ioana', 'ioana.pavel@companie.com', *TO\_DATE*('1986-05-18', 'YYYY-MM-DD'), 'Manager Marketing', 7300, 'Bulevardul Revoluției 16', '0700333322', 11, 2);  
INSERT INTO Angajati VALUES (13, 'Munteanu', 'Alin', 'alin.munteanu@companie.com', *TO\_DATE*('1989-01-29', 'YYYY-MM-DD'), 'Specialist PPC', 6500, 'Strada Viitorului 5', '0700444433', 11, 2);  
INSERT INTO Angajati VALUES (14, 'Petrescu', 'Diana', 'diana.petrescu@companie.com', *TO\_DATE*('1990-11-03', 'YYYY-MM-DD'), 'Content Creator', 6200, 'Aleea Rozelor 8', '0700555544', 11, 2);  
INSERT INTO Angajati VALUES (15, 'Ilie', 'Vasile', 'vasile.ilie@companie.com', *TO\_DATE*('1992-08-21', 'YYYY-MM-DD'), 'Grafician', 6000, 'Strada Unirii 11', '0700666655', 11, 2);  
INSERT INTO Angajati VALUES (16, 'Constantinescu', 'Elena', 'elena.constantinescu@companie.com', *TO\_DATE*('1993-10-10', 'YYYY-MM-DD'), 'Analyst Media', 6100, 'Bulevardul Dacia 14', '0700777766', 11, 2);  
INSERT INTO Angajati VALUES (17, 'Popa', 'Mircea', 'mircea.popa@companie.com', *TO\_DATE*('1991-06-06', 'YYYY-MM-DD'), 'Specialist Email Marketing', 6300, 'Strada Pacii 17', '0700888877', 11, 2);  
INSERT INTO Angajati VALUES (18, 'Andrei', 'Marius', 'marius.andrei@companie.com', *TO\_DATE*('1994-04-04', 'YYYY-MM-DD'), 'Manager Campanii', 7200, 'Aleea Libertății 20', '0700999988', 11, 2);  
INSERT INTO Angajati VALUES (19, 'Voinea', 'Laura', 'laura.voinea@companie.com', *TO\_DATE*('1987-03-15', 'YYYY-MM-DD'), 'Specialist PR', 6700, 'Strada Victoriei 22', '0700111199', 11, 2);  
INSERT INTO Angajati VALUES (20, 'Neagu', 'Daniel', 'daniel.neagu@companie.com', *TO\_DATE*('1995-12-12', 'YYYY-MM-DD'), 'Specialist Social Media', 6500, 'Bulevardul Aviatorilor 25', '0700222200', 11, 2);  
INSERT INTO Angajati VALUES (21, 'State', 'Oana', 'oana.state@companie.com', *TO\_DATE*('1984-02-28', 'YYYY-MM-DD'), 'HR Manager', 7500, 'Strada Lalelelor 28', '0700333311', NULL, 3);  
INSERT INTO Angajati VALUES (22, 'Vasile', 'Adrian', 'adrian.vasile@companie.com', *TO\_DATE*('1986-03-15', 'YYYY-MM-DD'), 'Specialist Recrutare', 6000, 'Strada Primaverii 22', '0700333322', 21, 3);  
INSERT INTO Angajati VALUES (23, 'Moldovan', 'Alina', 'alina.moldovan@companie.com', *TO\_DATE*('1990-07-20', 'YYYY-MM-DD'), 'Consilier HR', 5500, 'Aleea Florilor 35', '0700333333', 21, 3);  
INSERT INTO Angajati VALUES (24, 'Toma', 'Bogdan', 'bogdan.toma@companie.com', *TO\_DATE*('1989-09-10', 'YYYY-MM-DD'), 'Specialist Beneficii', 5800, 'Strada Unirii 18', '0700333344', 21, 3);  
INSERT INTO Angajati VALUES (25, 'Iliescu', 'Carmen', 'carmen.iliescu@companie.com', *TO\_DATE*('1988-12-05', 'YYYY-MM-DD'), 'Asistent HR', 5200, 'Bulevardul Eroilor 42', '0700333355', 21, 3);  
INSERT INTO Angajati VALUES (26, 'Manea', 'Daniel', 'daniel.manea@companie.com', *TO\_DATE*('1992-04-25', 'YYYY-MM-DD'), 'Specialist Payroll', 5700, 'Strada Libertatii 51', '0700333366', 21, 3);  
INSERT INTO Angajati VALUES (27, 'Stan', 'Elena', 'elena.stan@companie.com', *TO\_DATE*('1991-01-30', 'YYYY-MM-DD'), 'Coordinator HR', 6200, 'Aleea Rozelor 27', '0700333377', 21, 3);  
INSERT INTO Angajati VALUES (28, 'Dobre', 'Florin', 'florin.dobre@companie.com', *TO\_DATE*('1993-05-15', 'YYYY-MM-DD'), 'Specialist Relații Angajați', 5400, 'Strada Viitorului 33', '0700333388', 21, 3);  
INSERT INTO Angajati VALUES (29, 'Iancu', 'Gabriela', 'gabriela.iancu@companie.com', *TO\_DATE*('1987-08-21', 'YYYY-MM-DD'), 'Specialist Training', 5600, 'Bulevardul Independentei 17', '0700333399', 21, 3);  
INSERT INTO Angajati VALUES (30, 'Marin', 'Horia', 'horia.marin@companie.com', *TO\_DATE*('1994-11-11', 'YYYY-MM-DD'), 'Specialist Recrutare IT', 5900, 'Strada Pacii 19', '0700333300', 21, 3);  
INSERT INTO Angajati VALUES (31, 'Negru', 'Irina', 'irina.negru@companie.com', *TO\_DATE*('1985-06-09', 'YYYY-MM-DD'), 'Tehnician IT', 5300, 'Aleea Castanilor 2', '0700444411', NULL, 4);  
INSERT INTO Angajati VALUES (32, 'Luca', 'Marius', 'marius.luca@companie.com', *TO\_DATE*('1989-03-22', 'YYYY-MM-DD'), 'Administrator Retea', 5500, 'Strada Florilor 4', '0700444422', 31, 4);  
INSERT INTO Angajati VALUES (33, 'Stoian', 'Catalin', 'catalin.stoian@companie.com', *TO\_DATE*('1991-07-30', 'YYYY-MM-DD'), 'Support Engineer', 5200, 'Bulevardul Libertatii 7', '0700444433', 31, 4);  
INSERT INTO Angajati VALUES (34, 'Barbu', 'Diana', 'diana.barbu@companie.com', *TO\_DATE*('1990-11-15', 'YYYY-MM-DD'), 'Tehnician Helpdesk', 5100, 'Aleea Rozelor 9', '0700444444', 31, 4);  
INSERT INTO Angajati VALUES (35, 'Ionescu', 'Elena', 'elena.ionescu@companie.com', *TO\_DATE*('1992-02-20', 'YYYY-MM-DD'), 'Analist Securitate', 5400, 'Strada Unirii 12', '0700444455', 31, 4);  
INSERT INTO Angajati VALUES (36, 'Marinescu', 'Florin', 'florin.marinescu@companie.com', *TO\_DATE*('1987-05-05', 'YYYY-MM-DD'), 'Specialist IT', 5600, 'Bulevardul Revolutiei 3', '0700444466', 31, 4);  
INSERT INTO Angajati VALUES (37, 'Nistor', 'Gabriel', 'gabriel.nistor@companie.com', *TO\_DATE*('1994-10-10', 'YYYY-MM-DD'), 'Administrator Sistem', 5700, 'Strada Pacii 15', '0700444477', 31, 4);  
INSERT INTO Angajati VALUES (38, 'Dumitru', 'Horia', 'horia.dumitru@companie.com', *TO\_DATE*('1988-09-09', 'YYYY-MM-DD'), 'Tehnician Retele', 5300, 'Aleea Crinilor 18', '0700444488', 31, 4);  
INSERT INTO Angajati VALUES (39, 'Popa', 'Ion', 'ion.popa@companie.com', *TO\_DATE*('1993-08-08', 'YYYY-MM-DD'), 'Inginer Suport', 5400, 'Strada Principala 21', '0700444499', 31, 4);  
INSERT INTO Angajati VALUES (40, 'Andrei', 'Laura', 'laura.andrei@companie.com', *TO\_DATE*('1995-06-06', 'YYYY-MM-DD'), 'Asistent Suport', 5100, 'Bulevardul Industrial 24', '0700444400', 31, 4);  
INSERT INTO Angajati VALUES (41, 'Pop', 'Ionut', 'ionut.pop@companie.com', *TO\_DATE*('1986-04-04', 'YYYY-MM-DD'), 'Analist Securitate', 5800, 'Strada Libertatii 27', '0700555511', NULL, 5);  
INSERT INTO Angajati VALUES (42, 'Mihai', 'Bogdan', 'bogdan.mihai@companie.com', *TO\_DATE*('1990-12-12', 'YYYY-MM-DD'), 'Inginer Securitate', 6000, 'Bulevardul Unirii 30', '0700555522', 41, 5);  
INSERT INTO Angajati VALUES (43, 'Nicolae', 'Cristina', 'cristina.nicolae@companie.com', *TO\_DATE*('1987-11-11', 'YYYY-MM-DD'), 'Specialist Securitate IT', 5700, 'Aleea Soarelui 33', '0700555533', 41, 5);  
INSERT INTO Angajati VALUES (44, 'Ilie', 'Doru', 'doru.ilie@companie.com', *TO\_DATE*('1989-10-10', 'YYYY-MM-DD'), 'Consultant Securitate', 5900, 'Strada Victoriei 36', '0700555544', 41, 5);  
INSERT INTO Angajati VALUES (45, 'Constantin', 'Elena', 'elena.constantin@companie.com', *TO\_DATE*('1992-09-09', 'YYYY-MM-DD'), 'Inginer Criptografie', 6100, 'Bulevardul Eroilor 39', '0700555555', 41, 5);  
INSERT INTO Angajati VALUES (46, 'Stanescu', 'Flaviu', 'flaviu.stanescu@companie.com', *TO\_DATE*('1993-08-08', 'YYYY-MM-DD'), 'Analist Malware', 5600, 'Aleea Castanilor 42', '0700555566', 41, 5);  
INSERT INTO Angajati VALUES (47, 'Dinu', 'George', 'george.dinu@companie.com', *TO\_DATE*('1984-07-07', 'YYYY-MM-DD'), 'Specialist Securitate Rețea', 5800, 'Strada Muncii 45', '0700555577', 41, 5);  
INSERT INTO Angajati VALUES (48, 'Ene', 'Horia', 'horia.ene@companie.com', *TO\_DATE*('1985-06-06', 'YYYY-MM-DD'), 'Inginer Securitate Date', 5900, 'Bulevardul Industriei 48', '0700555588', 41, 5);  
INSERT INTO Angajati VALUES (49, 'Fratila', 'Ioana', 'ioana.fratila@companie.com', *TO\_DATE*('1991-05-05', 'YYYY-MM-DD'), 'Manager Securitate', 6200, 'Strada Florilor 51', '0700555599', 41, 5);  
INSERT INTO Angajati VALUES (50, 'Ghita', 'Julian', 'julian.ghita@companie.com', *TO\_DATE*('1994-03-03', 'YYYY-MM-DD'), 'Tehnician Securitate', 5500, 'Aleea Primaverii 54', '0700555500', 41, 5);

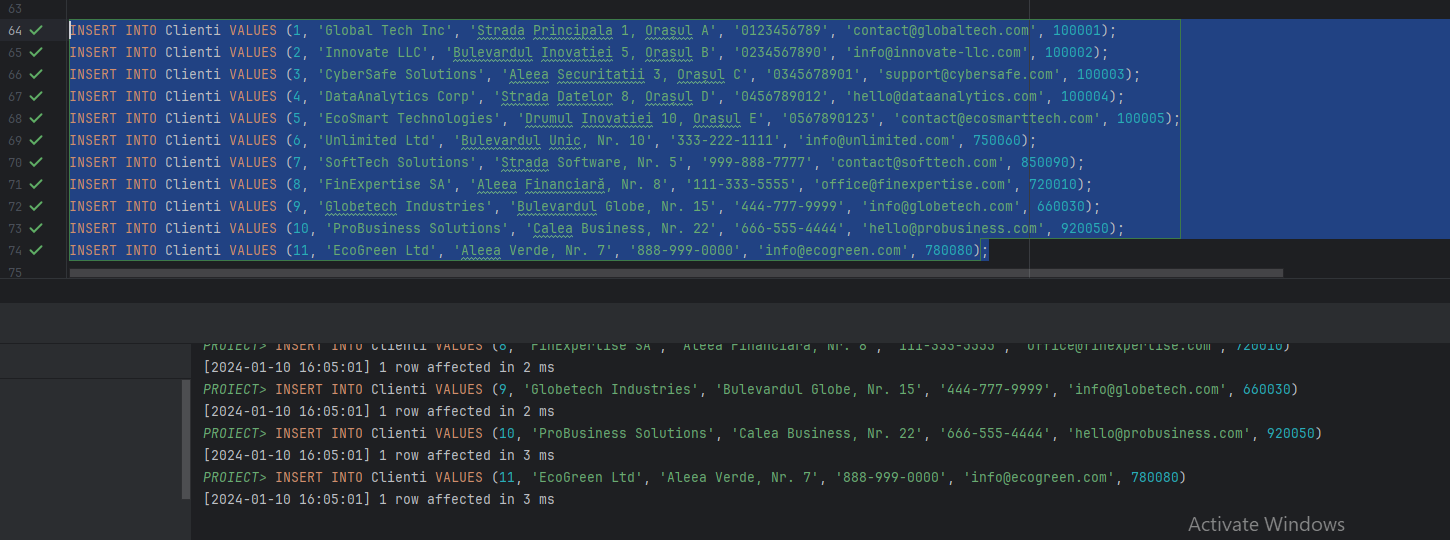




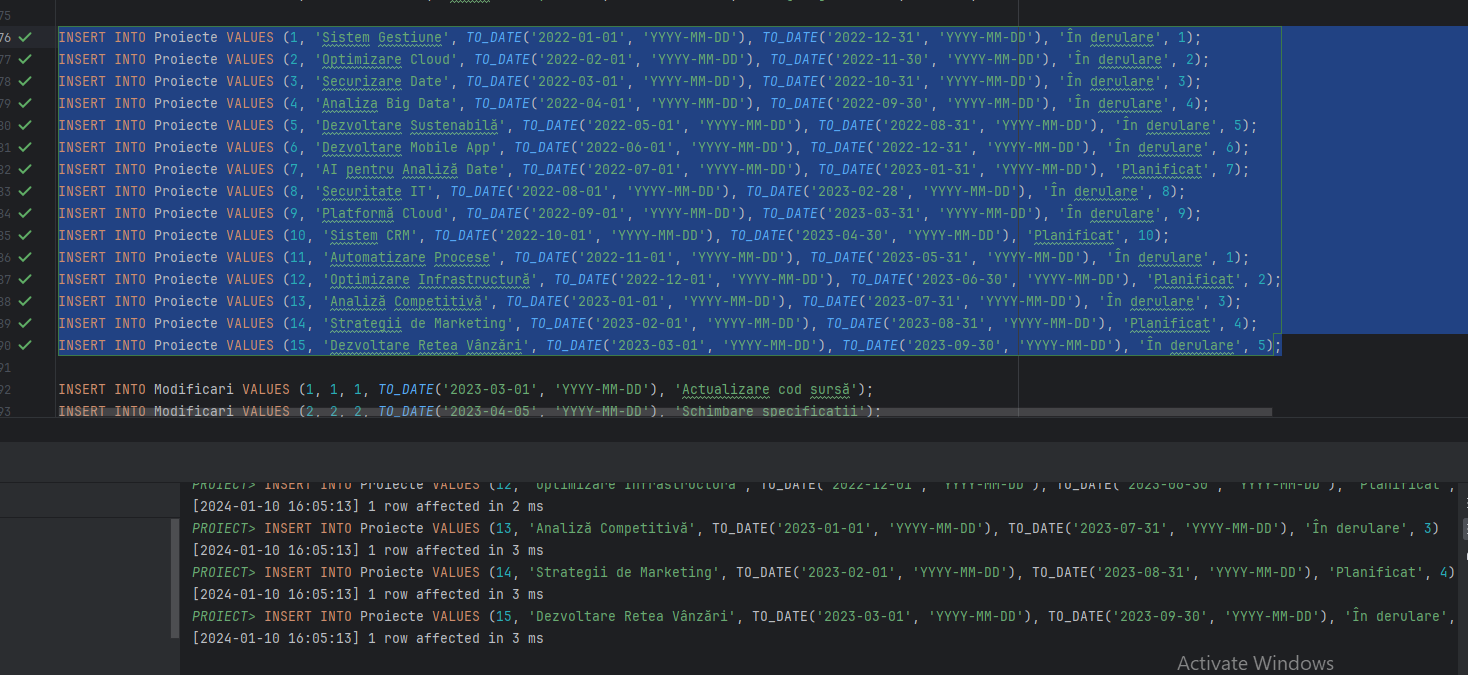
UPDATE Departamente SET manager\_id = 1 WHERE departament\_id = 1;  
UPDATE Departamente SET manager\_id = 11 WHERE departament\_id = 2;  
UPDATE Departamente SET manager\_id = 21 WHERE departament\_id = 3;  
UPDATE Departamente SET manager\_id = 31 WHERE departament\_id = 4;  
UPDATE Departamente SET manager\_id = 41 WHERE departament\_id = 5;



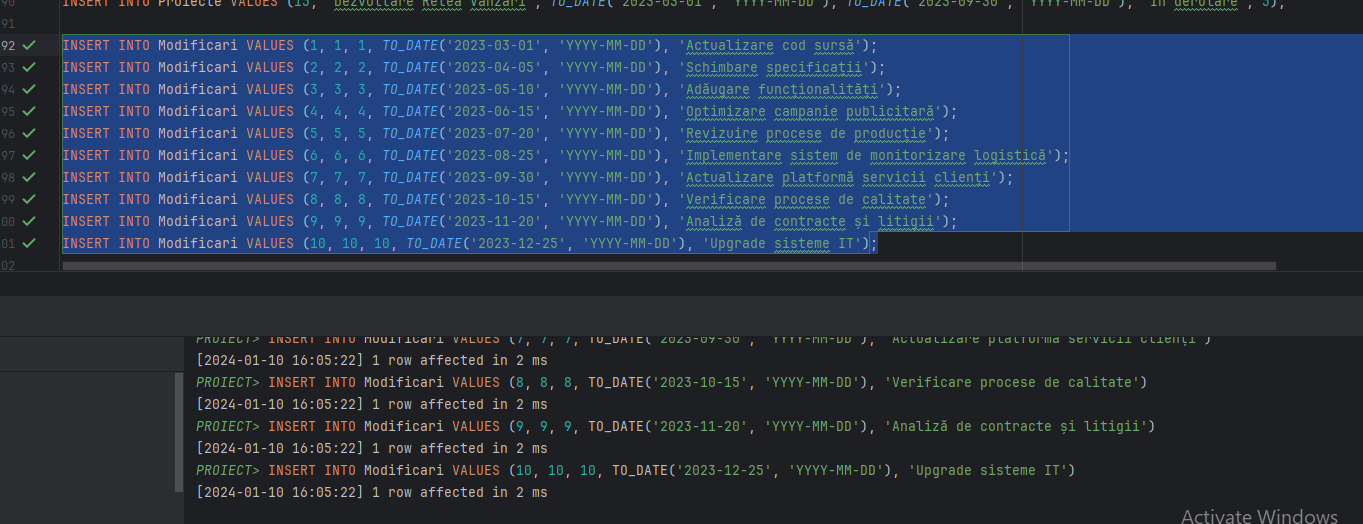
INSERT INTO Clienti VALUES (1, 'Global Tech Inc', 'Strada Principala 1, Orașul A', '0123456789', 'contact@globaltech.com', 100001);  
INSERT INTO Clienti VALUES (2, 'Innovate LLC', 'Bulevardul Inovatiei 5, Orașul B', '0234567890', 'info@innovate-llc.com', 100002);  
INSERT INTO Clienti VALUES (3, 'CyberSafe Solutions', 'Aleea Securitatii 3, Orașul C', '0345678901', 'support@cybersafe.com', 100003);  
INSERT INTO Clienti VALUES (4, 'DataAnalytics Corp', 'Strada Datelor 8, Orașul D', '0456789012', 'hello@dataanalytics.com', 100004);  
INSERT INTO Clienti VALUES (5, 'EcoSmart Technologies', 'Drumul Inovatiei 10, Orașul E', '0567890123', 'contact@ecosmarttech.com', 100005);  
INSERT INTO Clienti VALUES (6, 'Unlimited Ltd', 'Bulevardul Unic, Nr. 10', '333-222-1111', 'info@unlimited.com', 750060);  
INSERT INTO Clienti VALUES (7, 'SoftTech Solutions', 'Strada Software, Nr. 5', '999-888-7777', 'contact@softtech.com', 850090);  
INSERT INTO Clienti VALUES (8, 'FinExpertise SA', 'Aleea Financiară, Nr. 8', '111-333-5555', 'office@finexpertise.com', 720010);  
INSERT INTO Clienti VALUES (9, 'Globetech Industries', 'Bulevardul Globe, Nr. 15', '444-777-9999', 'info@globetech.com', 660030);  
INSERT INTO Clienti VALUES (10, 'ProBusiness Solutions', 'Calea Business, Nr. 22', '666-555-4444', 'hello@probusiness.com', 920050);  
INSERT INTO Clienti VALUES (11, 'EcoGreen Ltd', 'Aleea Verde, Nr. 7', '888-999-0000', 'info@ecogreen.com', 780080);



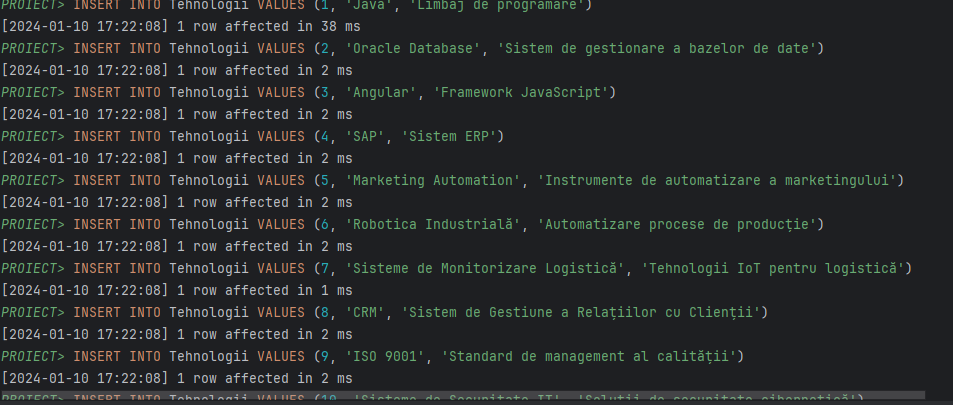
INSERT INTO Proiecte VALUES (1, 'Sistem Gestiune', *TO\_DATE*('2022-01-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-12-31', 'YYYY-MM-DD'), 'În derulare', 1);  
INSERT INTO Proiecte VALUES (2, 'Optimizare Cloud', *TO\_DATE*('2022-02-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-11-30', 'YYYY-MM-DD'), 'În derulare', 2);  
INSERT INTO Proiecte VALUES (3, 'Securizare Date', *TO\_DATE*('2022-03-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-10-31', 'YYYY-MM-DD'), 'În derulare', 3);  
INSERT INTO Proiecte VALUES (4, 'Analiza Big Data', *TO\_DATE*('2022-04-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-09-30', 'YYYY-MM-DD'), 'În derulare', 4);  
INSERT INTO Proiecte VALUES (5, 'Dezvoltare Sustenabilă', *TO\_DATE*('2022-05-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-08-31', 'YYYY-MM-DD'), 'În derulare', 5);  
INSERT INTO Proiecte VALUES (6, 'Dezvoltare Mobile App', *TO\_DATE*('2022-06-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-12-31', 'YYYY-MM-DD'), 'În derulare', 6);  
INSERT INTO Proiecte VALUES (7, 'AI pentru Analiză Date', *TO\_DATE*('2022-07-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-01-31', 'YYYY-MM-DD'), 'Planificat', 7);  
INSERT INTO Proiecte VALUES (8, 'Securitate IT', *TO\_DATE*('2022-08-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-02-28', 'YYYY-MM-DD'), 'În derulare', 8);  
INSERT INTO Proiecte VALUES (9, 'Platformă Cloud', *TO\_DATE*('2022-09-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-03-31', 'YYYY-MM-DD'), 'În derulare', 9);  
INSERT INTO Proiecte VALUES (10, 'Sistem CRM', *TO\_DATE*('2022-10-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-04-30', 'YYYY-MM-DD'), 'Planificat', 10);  
INSERT INTO Proiecte VALUES (11, 'Automatizare Procese', *TO\_DATE*('2022-11-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-05-31', 'YYYY-MM-DD'), 'În derulare', 1);  
INSERT INTO Proiecte VALUES (12, 'Optimizare Infrastructură', *TO\_DATE*('2022-12-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-06-30', 'YYYY-MM-DD'), 'Planificat', 2);  
INSERT INTO Proiecte VALUES (13, 'Analiză Competitivă', *TO\_DATE*('2023-01-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-07-31', 'YYYY-MM-DD'), 'În derulare', 3);  
INSERT INTO Proiecte VALUES (14, 'Strategii de Marketing', *TO\_DATE*('2023-02-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-08-31', 'YYYY-MM-DD'), 'Planificat', 4);  
INSERT INTO Proiecte VALUES (15, 'Dezvoltare Retea Vânzări', *TO\_DATE*('2023-03-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-09-30', 'YYYY-MM-DD'), 'În derulare', 5);



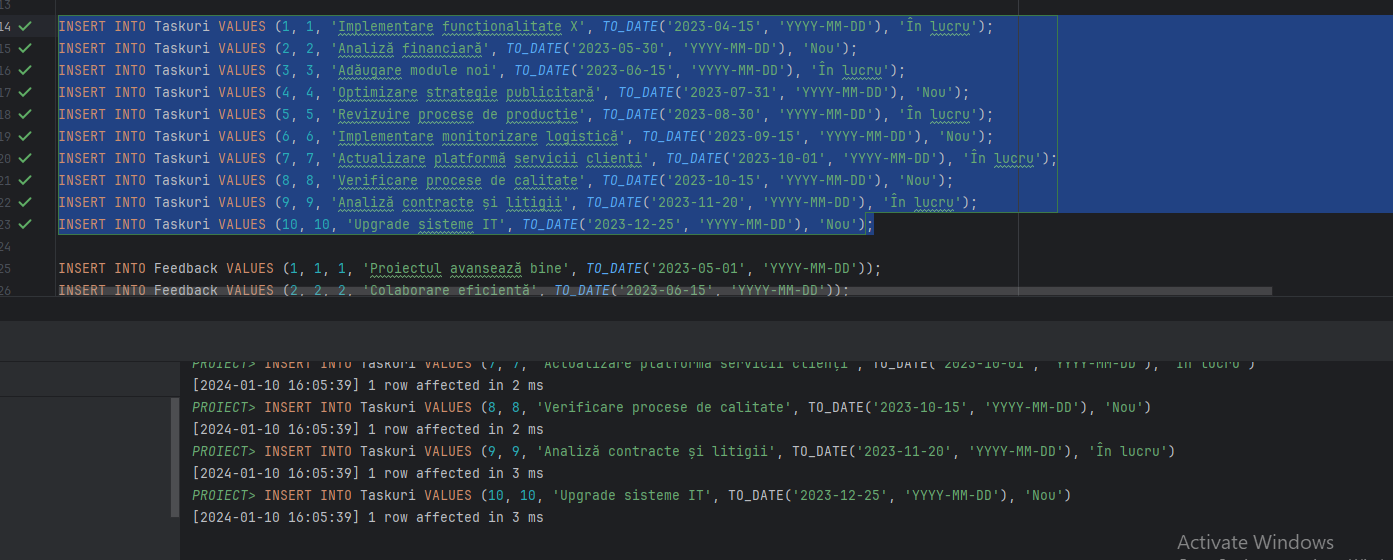
INSERT INTO Modificari VALUES (1, 1, 1, *TO\_DATE*('2023-03-01', 'YYYY-MM-DD'), 'Actualizare cod sursă');  
INSERT INTO Modificari VALUES (2, 2, 2, *TO\_DATE*('2023-04-05', 'YYYY-MM-DD'), 'Schimbare specificații');  
INSERT INTO Modificari VALUES (3, 3, 3, *TO\_DATE*('2023-05-10', 'YYYY-MM-DD'), 'Adăugare funcționalități');  
INSERT INTO Modificari VALUES (4, 4, 4, *TO\_DATE*('2023-06-15', 'YYYY-MM-DD'), 'Optimizare campanie publicitară');  
INSERT INTO Modificari VALUES (5, 5, 5, *TO\_DATE*('2023-07-20', 'YYYY-MM-DD'), 'Revizuire procese de producție');  
INSERT INTO Modificari VALUES (6, 6, 6, *TO\_DATE*('2023-08-25', 'YYYY-MM-DD'), 'Implementare sistem de monitorizare logistică');  
INSERT INTO Modificari VALUES (7, 7, 7, *TO\_DATE*('2023-09-30', 'YYYY-MM-DD'), 'Actualizare platformă servicii clienți');  
INSERT INTO Modificari VALUES (8, 8, 8, *TO\_DATE*('2023-10-15', 'YYYY-MM-DD'), 'Verificare procese de calitate');  
INSERT INTO Modificari VALUES (9, 9, 9, *TO\_DATE*('2023-11-20', 'YYYY-MM-DD'), 'Analiză de contracte și litigii');  
INSERT INTO Modificari VALUES (10, 10, 10, *TO\_DATE*('2023-12-25', 'YYYY-MM-DD'), 'Upgrade sisteme IT');



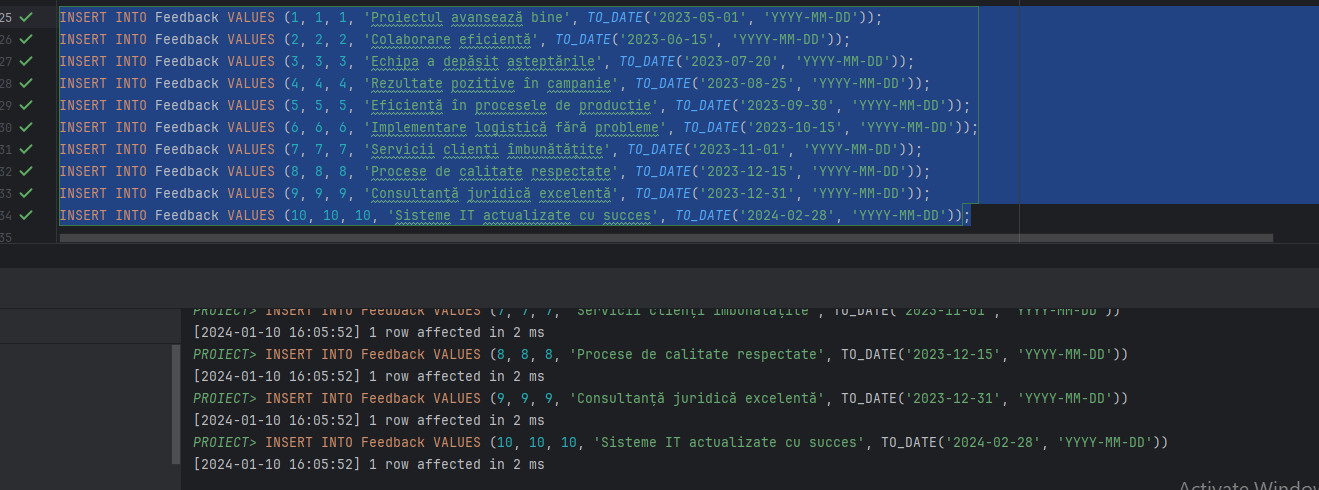
INSERT INTO Tehnologii VALUES (1, 'Java', 'Limbaj de programare');  
INSERT INTO Tehnologii VALUES (2, 'Oracle Database', 'Sistem de gestionare a bazelor de date');  
INSERT INTO Tehnologii VALUES (3, 'Angular', 'Framework JavaScript');  
INSERT INTO Tehnologii VALUES (4, 'SAP', 'Sistem ERP');  
INSERT INTO Tehnologii VALUES (5, 'Marketing Automation', 'Instrumente de automatizare a marketingului');  
INSERT INTO Tehnologii VALUES (6, 'Robotica Industrială', 'Automatizare procese de producție');  
INSERT INTO Tehnologii VALUES (7, 'Sisteme de Monitorizare Logistică', 'Tehnologii IoT pentru logistică');  
INSERT INTO Tehnologii VALUES (8, 'CRM', 'Sistem de Gestiune a Relațiilor cu Clienții');  
INSERT INTO Tehnologii VALUES (9, 'ISO 9001', 'Standard de management al calității');  
INSERT INTO Tehnologii VALUES (10, 'Sisteme de Securitate IT', 'Soluții de securitate cibernetică');



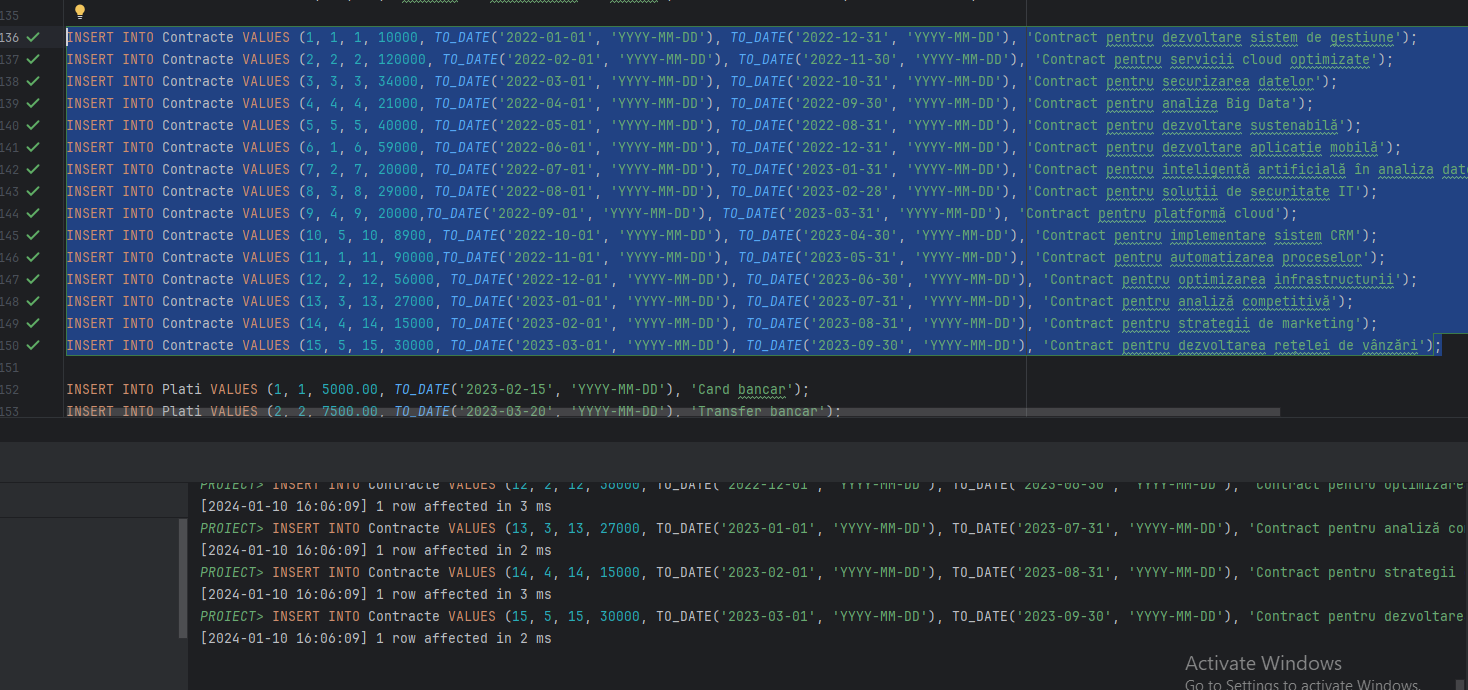
INSERT INTO Taskuri VALUES (1, 1, 'Implementare funcționalitate X', *TO\_DATE*('2023-04-15', 'YYYY-MM-DD'), 'În lucru');  
INSERT INTO Taskuri VALUES (2, 2, 'Analiză financiară', *TO\_DATE*('2023-05-30', 'YYYY-MM-DD'), 'Nou');  
INSERT INTO Taskuri VALUES (3, 3, 'Adăugare module noi', *TO\_DATE*('2023-06-15', 'YYYY-MM-DD'), 'În lucru');  
INSERT INTO Taskuri VALUES (4, 4, 'Optimizare strategie publicitară', *TO\_DATE*('2023-07-31', 'YYYY-MM-DD'), 'Nou');  
INSERT INTO Taskuri VALUES (5, 5, 'Revizuire procese de producție', *TO\_DATE*('2023-08-30', 'YYYY-MM-DD'), 'În lucru');  
INSERT INTO Taskuri VALUES (6, 6, 'Implementare monitorizare logistică', *TO\_DATE*('2023-09-15', 'YYYY-MM-DD'), 'Nou');  
INSERT INTO Taskuri VALUES (7, 7, 'Actualizare platformă servicii clienți', *TO\_DATE*('2023-10-01', 'YYYY-MM-DD'), 'În lucru');  
INSERT INTO Taskuri VALUES (8, 8, 'Verificare procese de calitate', *TO\_DATE*('2023-10-15', 'YYYY-MM-DD'), 'Nou');  
INSERT INTO Taskuri VALUES (9, 9, 'Analiză contracte și litigii', *TO\_DATE*('2023-11-20', 'YYYY-MM-DD'), 'În lucru');  
INSERT INTO Taskuri VALUES (10, 10, 'Upgrade sisteme IT', *TO\_DATE*('2023-12-25', 'YYYY-MM-DD'), 'Nou');



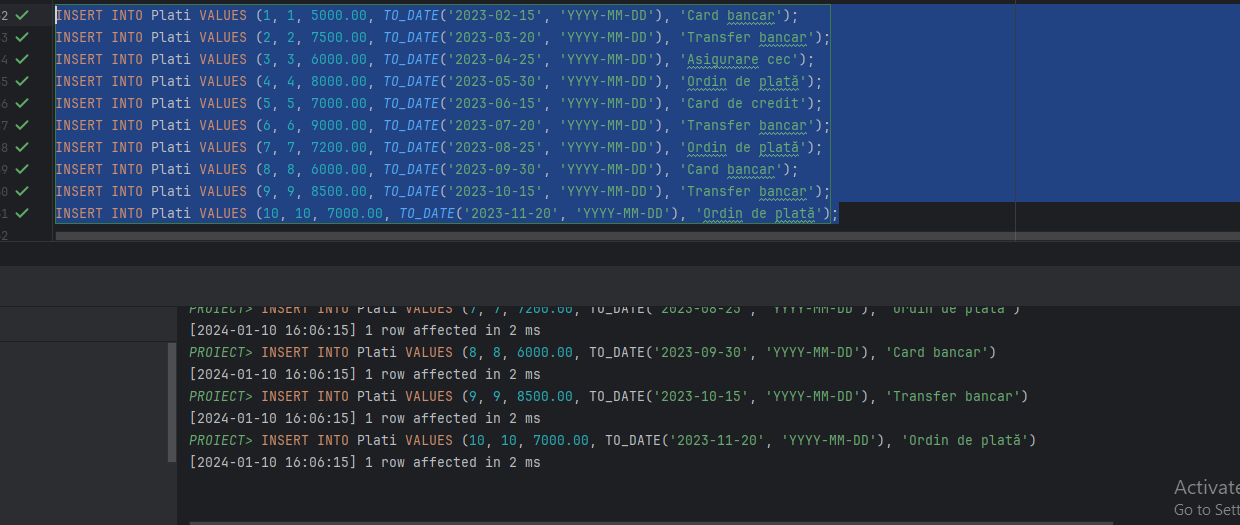
INSERT INTO Feedback VALUES (1, 1, 1, 'Proiectul avansează bine', *TO\_DATE*('2023-05-01', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (2, 2, 2, 'Colaborare eficientă', *TO\_DATE*('2023-06-15', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (3, 3, 3, 'Echipa a depășit așteptările', *TO\_DATE*('2023-07-20', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (4, 4, 4, 'Rezultate pozitive în campanie', *TO\_DATE*('2023-08-25', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (5, 5, 5, 'Eficiență în procesele de producție', *TO\_DATE*('2023-09-30', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (6, 6, 6, 'Implementare logistică fără probleme', *TO\_DATE*('2023-10-15', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (7, 7, 7, 'Servicii clienți îmbunătățite', *TO\_DATE*('2023-11-01', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (8, 8, 8, 'Procese de calitate respectate', *TO\_DATE*('2023-12-15', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (9, 9, 9, 'Consultanță juridică excelentă', *TO\_DATE*('2023-12-31', 'YYYY-MM-DD'));  
INSERT INTO Feedback VALUES (10, 10, 10, 'Sisteme IT actualizate cu succes', *TO\_DATE*('2024-02-28', 'YYYY-MM-DD'));



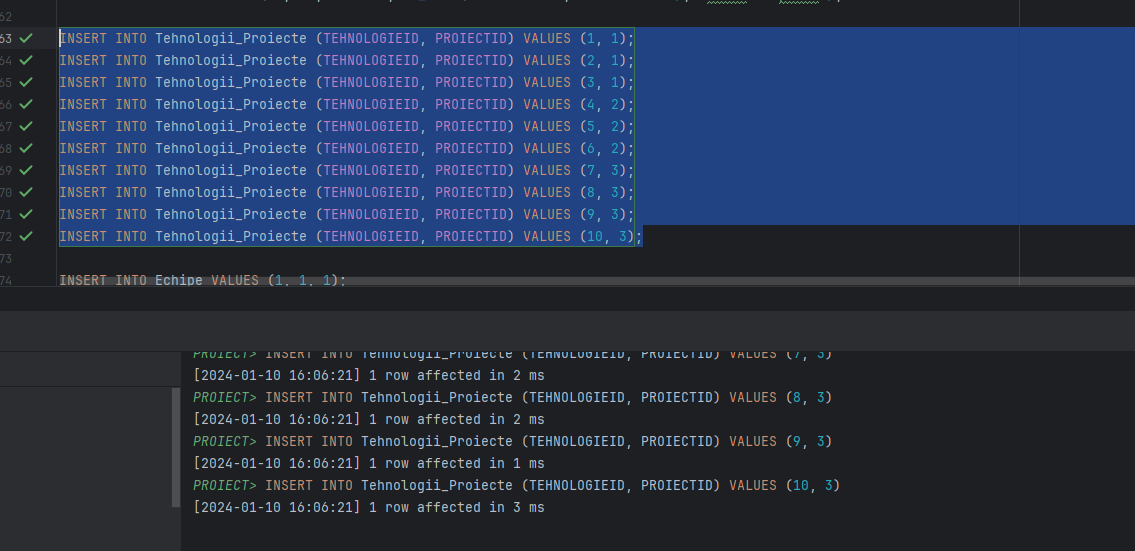
INSERT INTO Contracte VALUES (1, 1, 1, 10000, *TO\_DATE*('2022-01-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-12-31', 'YYYY-MM-DD'), 'Contract pentru dezvoltare sistem de gestiune');  
INSERT INTO Contracte VALUES (2, 2, 2, 120000, *TO\_DATE*('2022-02-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-11-30', 'YYYY-MM-DD'), 'Contract pentru servicii cloud optimizate');  
INSERT INTO Contracte VALUES (3, 3, 3, 34000, *TO\_DATE*('2022-03-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-10-31', 'YYYY-MM-DD'), 'Contract pentru securizarea datelor');  
INSERT INTO Contracte VALUES (4, 4, 4, 21000, *TO\_DATE*('2022-04-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-09-30', 'YYYY-MM-DD'), 'Contract pentru analiza Big Data');  
INSERT INTO Contracte VALUES (5, 5, 5, 40000, *TO\_DATE*('2022-05-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-08-31', 'YYYY-MM-DD'), 'Contract pentru dezvoltare sustenabilă');  
INSERT INTO Contracte VALUES (6, 1, 6, 59000, *TO\_DATE*('2022-06-01', 'YYYY-MM-DD'), *TO\_DATE*('2022-12-31', 'YYYY-MM-DD'), 'Contract pentru dezvoltare aplicație mobilă');  
INSERT INTO Contracte VALUES (7, 2, 7, 20000, *TO\_DATE*('2022-07-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-01-31', 'YYYY-MM-DD'), 'Contract pentru inteligență artificială în analiza datelor');  
INSERT INTO Contracte VALUES (8, 3, 8, 29000, *TO\_DATE*('2022-08-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-02-28', 'YYYY-MM-DD'), 'Contract pentru soluții de securitate IT');  
INSERT INTO Contracte VALUES (9, 4, 9, 20000,*TO\_DATE*('2022-09-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-03-31', 'YYYY-MM-DD'), 'Contract pentru platformă cloud');  
INSERT INTO Contracte VALUES (10, 5, 10, 8900, *TO\_DATE*('2022-10-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-04-30', 'YYYY-MM-DD'), 'Contract pentru implementare sistem CRM');  
INSERT INTO Contracte VALUES (11, 1, 11, 90000,*TO\_DATE*('2022-11-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-05-31', 'YYYY-MM-DD'), 'Contract pentru automatizarea proceselor');  
INSERT INTO Contracte VALUES (12, 2, 12, 56000, *TO\_DATE*('2022-12-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-06-30', 'YYYY-MM-DD'), 'Contract pentru optimizarea infrastructurii');  
INSERT INTO Contracte VALUES (13, 3, 13, 27000, *TO\_DATE*('2023-01-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-07-31', 'YYYY-MM-DD'), 'Contract pentru analiză competitivă');  
INSERT INTO Contracte VALUES (14, 4, 14, 15000, *TO\_DATE*('2023-02-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-08-31', 'YYYY-MM-DD'), 'Contract pentru strategii de marketing');  
INSERT INTO Contracte VALUES (15, 5, 15, 30000, *TO\_DATE*('2023-03-01', 'YYYY-MM-DD'), *TO\_DATE*('2023-09-30', 'YYYY-MM-DD'), 'Contract pentru dezvoltarea rețelei de vânzări');



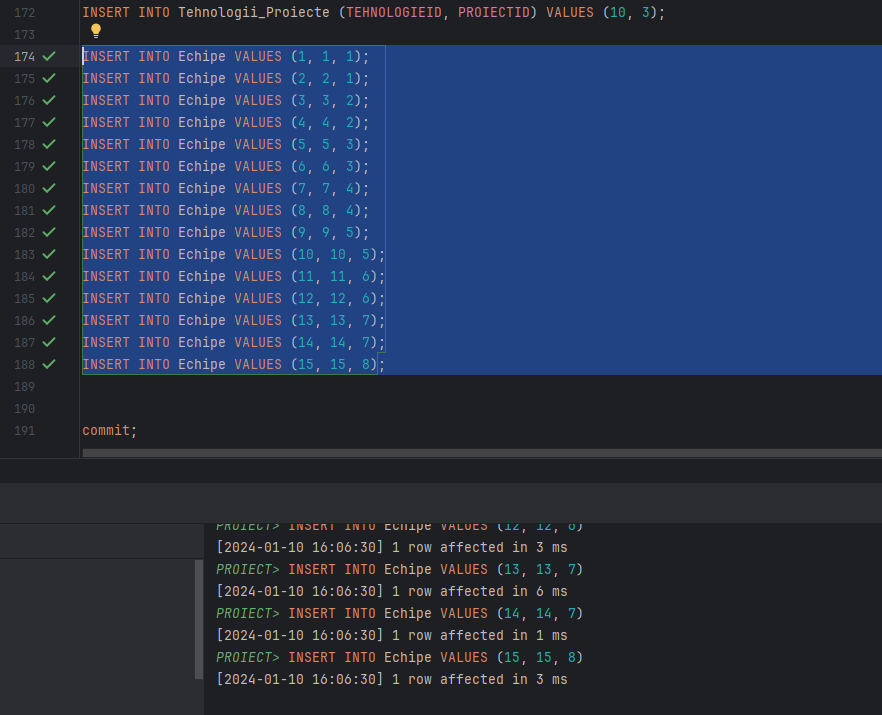
INSERT INTO Plati VALUES (1, 1, 5000.00, *TO\_DATE*('2023-02-15', 'YYYY-MM-DD'), 'Card bancar');  
INSERT INTO Plati VALUES (2, 2, 7500.00, *TO\_DATE*('2023-03-20', 'YYYY-MM-DD'), 'Transfer bancar');  
INSERT INTO Plati VALUES (3, 3, 6000.00, *TO\_DATE*('2023-04-25', 'YYYY-MM-DD'), 'Asigurare cec');  
INSERT INTO Plati VALUES (4, 4, 8000.00, *TO\_DATE*('2023-05-30', 'YYYY-MM-DD'), 'Ordin de plată');  
INSERT INTO Plati VALUES (5, 5, 7000.00, *TO\_DATE*('2023-06-15', 'YYYY-MM-DD'), 'Card de credit');  
INSERT INTO Plati VALUES (6, 6, 9000.00, *TO\_DATE*('2023-07-20', 'YYYY-MM-DD'), 'Transfer bancar');  
INSERT INTO Plati VALUES (7, 7, 7200.00, *TO\_DATE*('2023-08-25', 'YYYY-MM-DD'), 'Ordin de plată');  
INSERT INTO Plati VALUES (8, 8, 6000.00, *TO\_DATE*('2023-09-30', 'YYYY-MM-DD'), 'Card bancar');  
INSERT INTO Plati VALUES (9, 9, 8500.00, *TO\_DATE*('2023-10-15', 'YYYY-MM-DD'), 'Transfer bancar');  
INSERT INTO Plati VALUES (10, 10, 7000.00, *TO\_DATE*('2023-11-20', 'YYYY-MM-DD'), 'Ordin de plată');



INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (1, 1);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (2, 1);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (3, 1);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (4, 2);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (5, 2);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (6, 2);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (7, 3);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (8, 3);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (9, 3);  
INSERT INTO Tehnologii\_Proiecte (TEHNOLOGIEID, PROIECTID) VALUES (10, 3);



INSERT INTO Echipe VALUES (1, 1, 1);  
INSERT INTO Echipe VALUES (2, 2, 1);  
INSERT INTO Echipe VALUES (3, 3, 2);  
INSERT INTO Echipe VALUES (4, 4, 2);  
INSERT INTO Echipe VALUES (5, 5, 3);  
INSERT INTO Echipe VALUES (6, 6, 3);  
INSERT INTO Echipe VALUES (7, 7, 4);  
INSERT INTO Echipe VALUES (8, 8, 4);  
INSERT INTO Echipe VALUES (9, 9, 5);  
INSERT INTO Echipe VALUES (10, 10, 5);  
INSERT INTO Echipe VALUES (11, 11, 6);  
INSERT INTO Echipe VALUES (12, 12, 6);  
INSERT INTO Echipe VALUES (13, 13, 7);  
INSERT INTO Echipe VALUES (14, 14, 7);  
INSERT INTO Echipe VALUES (15, 15, 8);

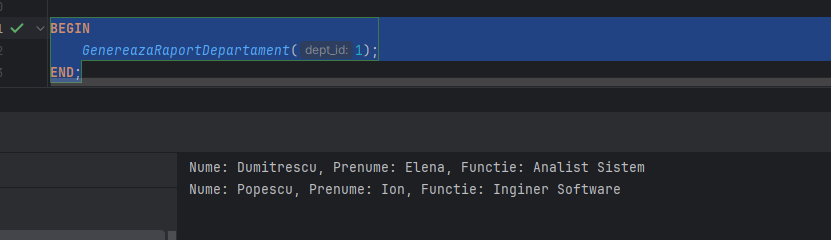
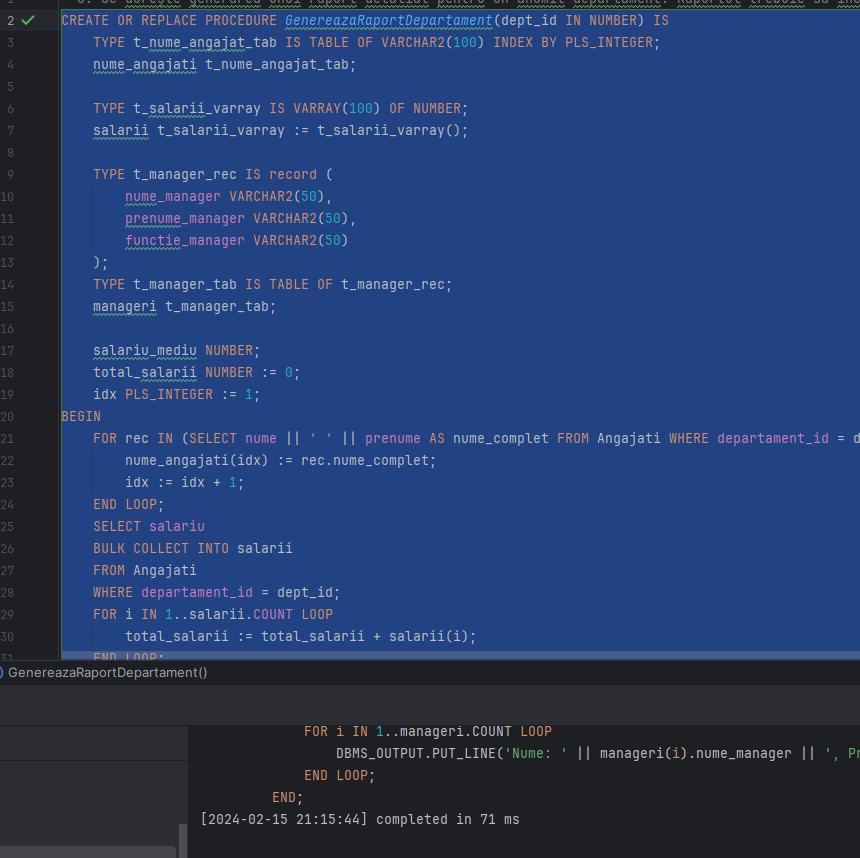


6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze toate cele 3 tipuri de colecții studiate. Apelați subprogramul.

Se dorește generarea unui raport pentru un anumit departament. Raportul trebuie să includă: Lista angajatilor din acel departament, salariul mediu din acel departament si managerii acelui departament.

CREATE OR REPLACE PROCEDURE *GenereazaRaportDepartament*(dept\_id IN NUMBER) IS  
 TYPE t\_nume\_angajat\_tab IS TABLE OF VARCHAR2(100) INDEX BY PLS\_INTEGER;  
 nume\_angajati t\_nume\_angajat\_tab;  
  
 TYPE t\_salarii\_varray IS VARRAY(100) OF NUMBER;  
 salarii t\_salarii\_varray := t\_salarii\_varray();  
  
 TYPE t\_manager\_rec IS record (  
 nume\_manager VARCHAR2(50),  
 prenume\_manager VARCHAR2(50),  
 functie\_manager VARCHAR2(50)  
 );  
 TYPE t\_manager\_tab IS TABLE OF t\_manager\_rec;  
 manageri t\_manager\_tab;  
  
 salariu\_mediu NUMBER;  
 total\_salarii NUMBER := 0;  
 idx PLS\_INTEGER := 1;  
BEGIN  
 FOR rec IN (SELECT nume || ' ' || prenume AS nume\_complet FROM Angajati WHERE departament\_id = dept\_id) LOOP  
 nume\_angajati(idx) := rec.nume\_complet;  
 idx := idx + 1;  
 END LOOP;  
 SELECT salariu  
 BULK COLLECT INTO salarii  
 FROM Angajati  
 WHERE departament\_id = dept\_id;  
 FOR i IN 1..salarii.COUNT LOOP  
 total\_salarii := total\_salarii + salarii(i);  
 END LOOP;  
 salariu\_mediu := CASE WHEN salarii.COUNT > 0 THEN total\_salarii / salarii.COUNT ELSE 0 END;  
 SELECT A.NUME, A.PRENUME, A.FUNCTIE  
 BULK COLLECT INTO manageri  
 FROM Angajati A  
 WHERE angajat\_id IN (SELECT DISTINCT manager\_id FROM Angajati WHERE departament\_id = dept\_id AND manager\_id IS NOT NULL);  
 DBMS\_OUTPUT.*PUT\_LINE*('Raport pentru Departamentul cu ID: ' || dept\_id);  
 DBMS\_OUTPUT.*PUT\_LINE*('Lista Angajatilor:');  
 idx := nume\_angajati.FIRST;  
 WHILE idx IS NOT NULL LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*(nume\_angajati(idx));  
 idx := nume\_angajati.*NEXT*(idx);  
 END LOOP;  
 DBMS\_OUTPUT.*PUT\_LINE*('Salariu Mediu: ' || salariu\_mediu);  
 DBMS\_OUTPUT.*PUT\_LINE*('Lista Managerilor:');  
 FOR i IN 1..manageri.COUNT LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('Nume: ' || manageri(i).nume\_manager || ', Prenume: ' || manageri(i).prenume\_manager || ', Functie: ' || manageri(i).functie\_manager);  
 END LOOP;  
END;

BEGIN  
 GenereazaRaportDepartament(1);  
END;

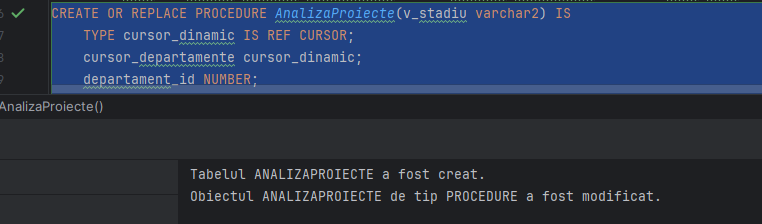


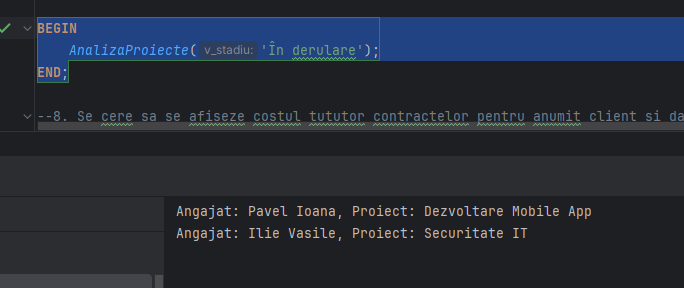
7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat, dependent de celălalt cursor. Apelați subprogramul.

Se cere ca pentru fiecare departament sa se afiseze proiectele care se afla intr-un anumit stadiu, si numele si prenumele fiecarui angajat care lucreaza la acel proiect.

CREATE OR REPLACE PROCEDURE *AnalizaProiecte*(v\_stadiu varchar2) IS  
 TYPE cursor\_dinamic IS REF CURSOR;  
 cursor\_departamente cursor\_dinamic;  
 departament\_id NUMBER;  
 numar\_proiecte NUMBER;  
 CURSOR cursor\_angajati(dept\_id NUMBER) IS  
 SELECT a.nume, a.PRENUME, p.nume\_proiect  
 FROM Angajati a  
 Join ECHIPE E on a.ANGAJAT\_ID = E.ID\_ANGAJAT  
 JOIN Proiecte p ON E.proiect\_id = p.proiect\_id  
 WHERE a.departament\_id = dept\_id AND p.stadiu = v\_stadiu;  
BEGIN  
 OPEN cursor\_departamente FOR  
 SELECT d.departament\_id, *COUNT*(p.proiect\_id) AS numar\_proiecte  
 FROM Departamente d  
 JOIN Angajati a ON d.departament\_id = a.departament\_id  
 Join ECHIPE E on a.ANGAJAT\_ID = E.ID\_ANGAJAT  
 JOIN Proiecte p ON E.proiect\_id = p.proiect\_id  
 WHERE *upper*(p.stadiu) = *upper*(v\_stadiu)  
 GROUP BY d.departament\_id;  
 LOOP  
 FETCH cursor\_departamente INTO departament\_id, numar\_proiecte;  
 EXIT WHEN cursor\_departamente%NOTFOUND;  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Departament ' || departament\_id || ' - Număr Proiecte: ' || numar\_proiecte);  
 FOR REC IN cursor\_angajati(departament\_id) LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('Angajat: ' || REC.NUME ||' '|| REC.PRENUME || ', Proiect: ' || REC.NUME\_PROIECT);  
 end loop;  
 END LOOP;  
 CLOSE cursor\_departamente;  
END;

BEGIN  
 *AnalizaProiecte*('În derulare');  
END;



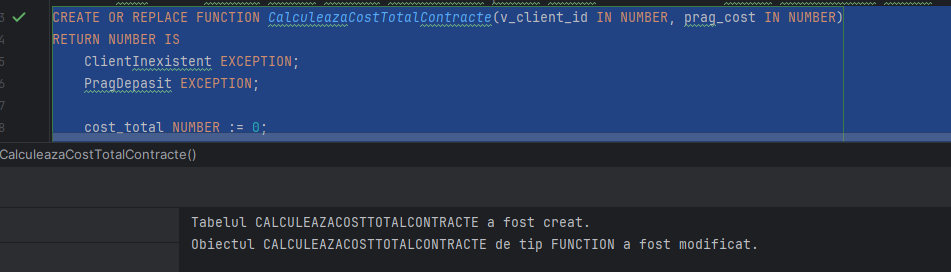


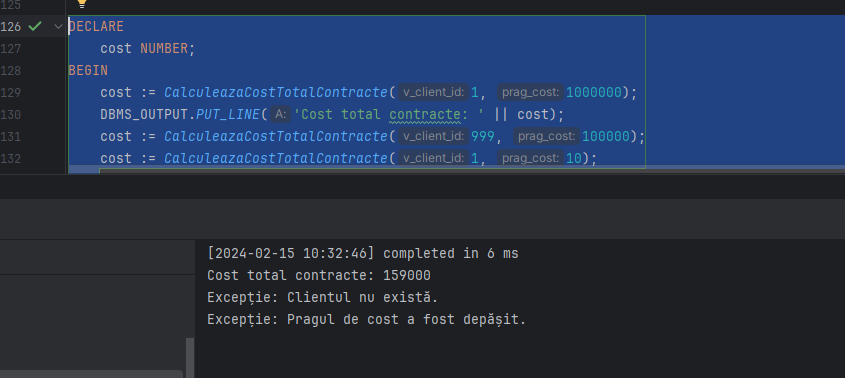
8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Definiți minim 2 excepții proprii. Apelați subprogramul astfel încât să evidențiați toate cazurile definite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

Se cere sa se afiseze costul tututor contractelor pentru anumit client si daca costul contractelor acelui client depaseste un prag.

CREATE OR REPLACE FUNCTION *CalculeazaCostTotalContracte*(v\_client\_id IN NUMBER, prag\_cost IN NUMBER)  
RETURN NUMBER IS  
 ClientInexistent EXCEPTION;  
 PragDepasit EXCEPTION;  
  
 cost\_total NUMBER := 0;  
  
BEGIN  
 SELECT *SUM*(c.cost\_contract)  
 INTO cost\_total  
 FROM Contracte c  
 JOIN Proiecte p ON c.proiect\_id = p.proiect\_id  
 JOIN Clienti cl ON c.client\_id = cl.client\_id  
 WHERE cl.client\_id = v\_client\_id;  
  
 IF cost\_total IS NULL THEN  
 RAISE ClientInexistent;  
 END IF;  
 IF cost\_total > prag\_cost THEN  
 RAISE PragDepasit;  
 END IF;  
  
 RETURN cost\_total;  
  
EXCEPTION  
 WHEN ClientInexistent THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Clientul nu există.');  
 RETURN NULL;  
 WHEN PragDepasit THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Pragul de cost a fost depășit.');  
 RETURN cost\_total;  
END;

DECLARE  
 cost NUMBER;  
BEGIN  
 cost := *CalculeazaCostTotalContracte*(1, 1000000);  
 DBMS\_OUTPUT.*PUT\_LINE*('Cost total contracte: ' || cost);  
 cost := *CalculeazaCostTotalContracte*(999, 100000);  
 cost := *CalculeazaCostTotalContracte*(1, 10);  
END;



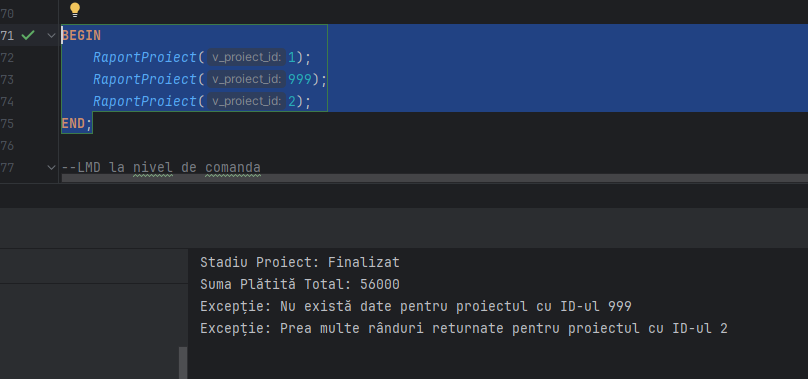
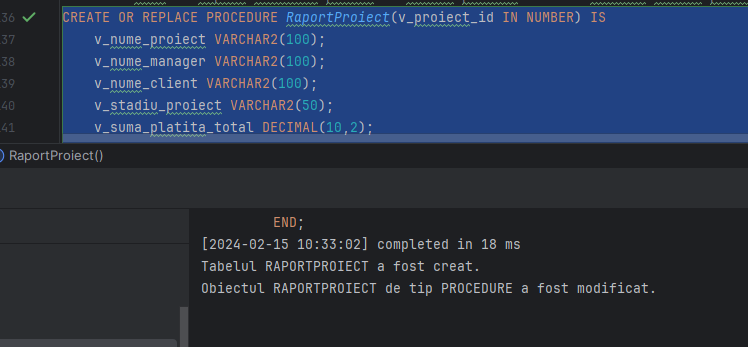


9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO\_DATA\_FOUND și TOO\_MANY\_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

Se cere un raport detaliat pentru un anumit proiect care sa contina numele proiectului, numele managerului, numele clientului, stadiul curent al proiectului, si suma platita pentru acel proiect.

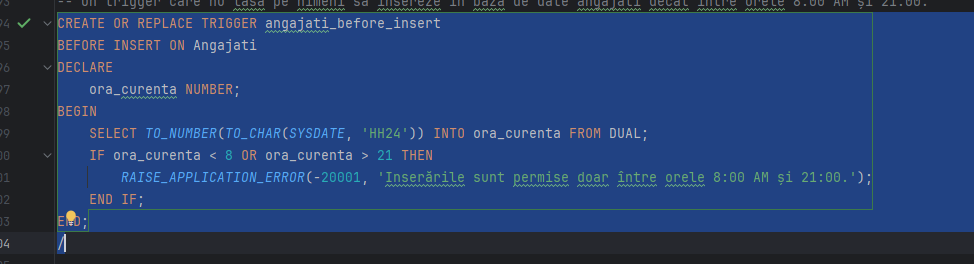
CREATE OR REPLACE PROCEDURE *RaportProiect*(v\_proiect\_id IN NUMBER) IS  
 v\_nume\_proiect VARCHAR2(100);  
 v\_nume\_manager VARCHAR2(100);  
 v\_nume\_client VARCHAR2(100);  
 v\_stadiu\_proiect VARCHAR2(50);  
 v\_suma\_platita\_total DECIMAL(10,2);  
  
BEGIN  
 SELECT p.nume\_proiect, a.nume || ' ' || a.prenume, cl.nume\_client, p.stadiu, *SUM*(pl.suma\_platita)  
 INTO v\_nume\_proiect, v\_nume\_manager, v\_nume\_client, v\_stadiu\_proiect, v\_suma\_platita\_total  
 FROM Proiecte p  
 JOIN Echipe e ON p.proiect\_id = e.proiect\_id  
 JOIN Angajati a ON e.id\_angajat = a.angajat\_id AND a.manager\_id IS NOT NULL  
 JOIN Clienti cl ON p.client\_id = cl.client\_id  
 JOIN Contracte c ON cl.client\_id = c.client\_id  
 JOIN Plati pl ON c.contract\_id = pl.contract\_id  
 WHERE p.proiect\_id = v\_proiect\_id  
 GROUP BY p.nume\_proiect, a.nume, a.prenume, cl.nume\_client, p.stadiu;  
 DBMS\_OUTPUT.*PUT\_LINE*('Nume Proiect: ' || v\_nume\_proiect);  
 DBMS\_OUTPUT.*PUT\_LINE*('Manager Proiect: ' || v\_nume\_manager);  
 DBMS\_OUTPUT.*PUT\_LINE*('Client: ' || v\_nume\_client);  
 DBMS\_OUTPUT.*PUT\_LINE*('Stadiu Proiect: ' || v\_stadiu\_proiect);  
 DBMS\_OUTPUT.*PUT\_LINE*('Suma Plătită Total: ' || v\_suma\_platita\_total);  
  
EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Nu există date pentru proiectul cu ID-ul ' || v\_proiect\_id);  
 WHEN TOO\_MANY\_ROWS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Prea multe rânduri returnate pentru proiectul cu ID-ul ' || v\_proiect\_id);  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție Necunoscută: ' || *SQLERRM*);  
END;

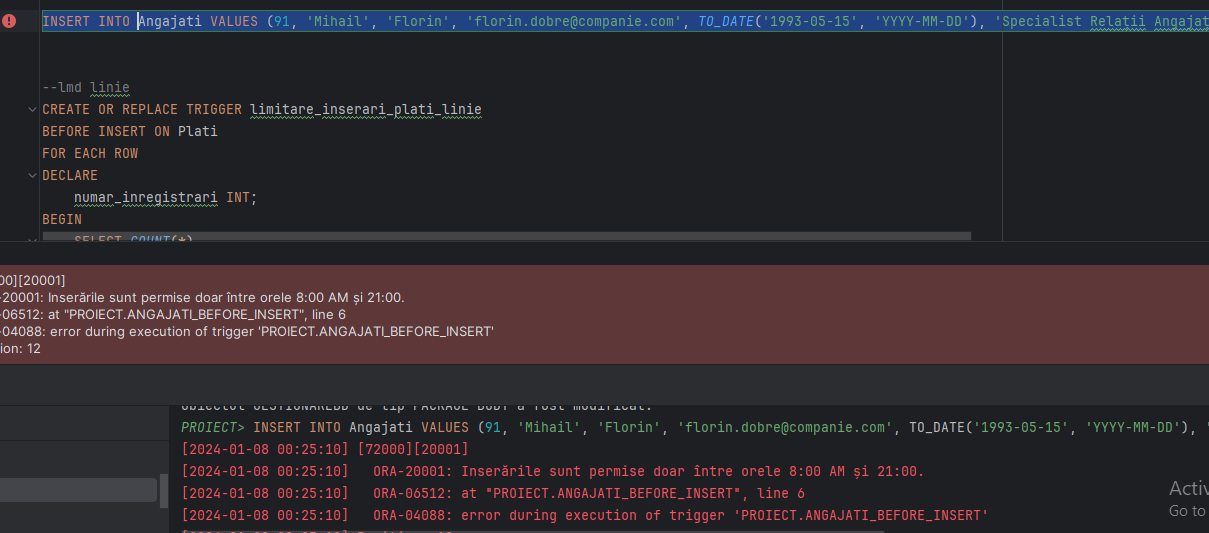
BEGIN  
 *RaportProiect*(1);  
 *RaportProiect*(999);  
 *RaportProiect*(2);  
END;



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

Un trigger care nu lasa pe nimeni sa insereze in baza de date angajati decat intre orele 8:00 AM și 21:00.

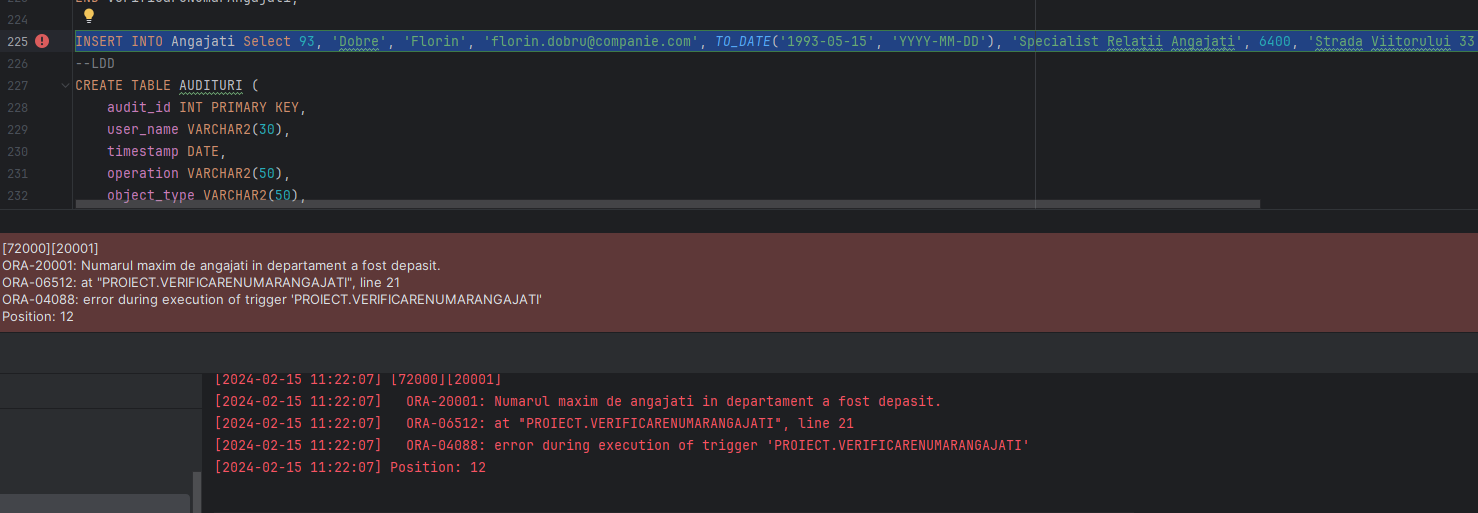
--LMD la nivel de comanda  
CREATE OR REPLACE TRIGGER angajati\_before\_insert  
BEFORE INSERT ON Angajati  
DECLARE  
 ora\_curenta NUMBER;  
BEGIN  
 SELECT *TO\_NUMBER*(*TO\_CHAR*(*SYSDATE*, 'HH24')) INTO ora\_curenta FROM DUAL;  
 IF ora\_curenta < 8 OR ora\_curenta > 21 THEN  
 *RAISE\_APPLICATION\_ERROR*(-20001, 'Inserările sunt permise doar între orele 8:00 AM și 21:00.');  
 END IF;  
END;  
/



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

CREATE OR REPLACE TRIGGER VerificareNumarAngajati  
FOR INSERT OR UPDATE ON Angajati  
COMPOUND TRIGGER  
 TYPE t\_departament\_info IS RECORD (departament\_id Departamente.departament\_id%TYPE, numar\_angajati NUMBER);  
 TYPE t\_tabel\_departamente IS TABLE OF t\_departament\_info INDEX BY PLS\_INTEGER;  
 departamente\_info t\_tabel\_departamente;  
 BEFORE STATEMENT IS  
 BEGIN  
 FOR r\_departament IN (SELECT departament\_id FROM Departamente) LOOP  
 SELECT *COUNT*(\*)  
 INTO departamente\_info(r\_departament.departament\_id).numar\_angajati  
 FROM Angajati  
 WHERE departament\_id = r\_departament.departament\_id;  
  
 departamente\_info(r\_departament.departament\_id).departament\_id := r\_departament.departament\_id;  
 END LOOP;  
 END BEFORE STATEMENT;  
 BEFORE EACH ROW IS  
 BEGIN  
 IF :NEW.departament\_id IS NOT NULL THEN  
 IF departamente\_info.*EXISTS*(:NEW.departament\_id) THEN  
 IF departamente\_info(:NEW.departament\_id).numar\_angajati > 10 THEN  
 *RAISE\_APPLICATION\_ERROR*(-20001, 'Numarul maxim de angajati in departament a fost depasit.');  
 END IF;  
 END IF;  
 END IF;  
 END BEFORE EACH ROW;  
  
END VerificareNumarAngajati;

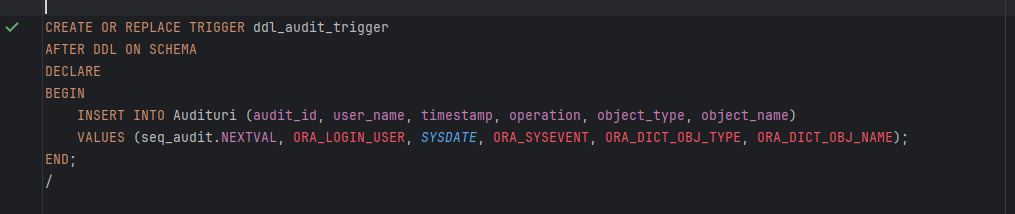
INSERT INTO Angajati Select 93, 'Dobre', 'Florin', 'florin.dobru@companie.com', *TO\_DATE*('1993-05-15', 'YYYY-MM-DD'), 'Specialist Relații Angajați', 6400, 'Strada Viitorului 33', '0700333388', 5, 1 from DUAL;

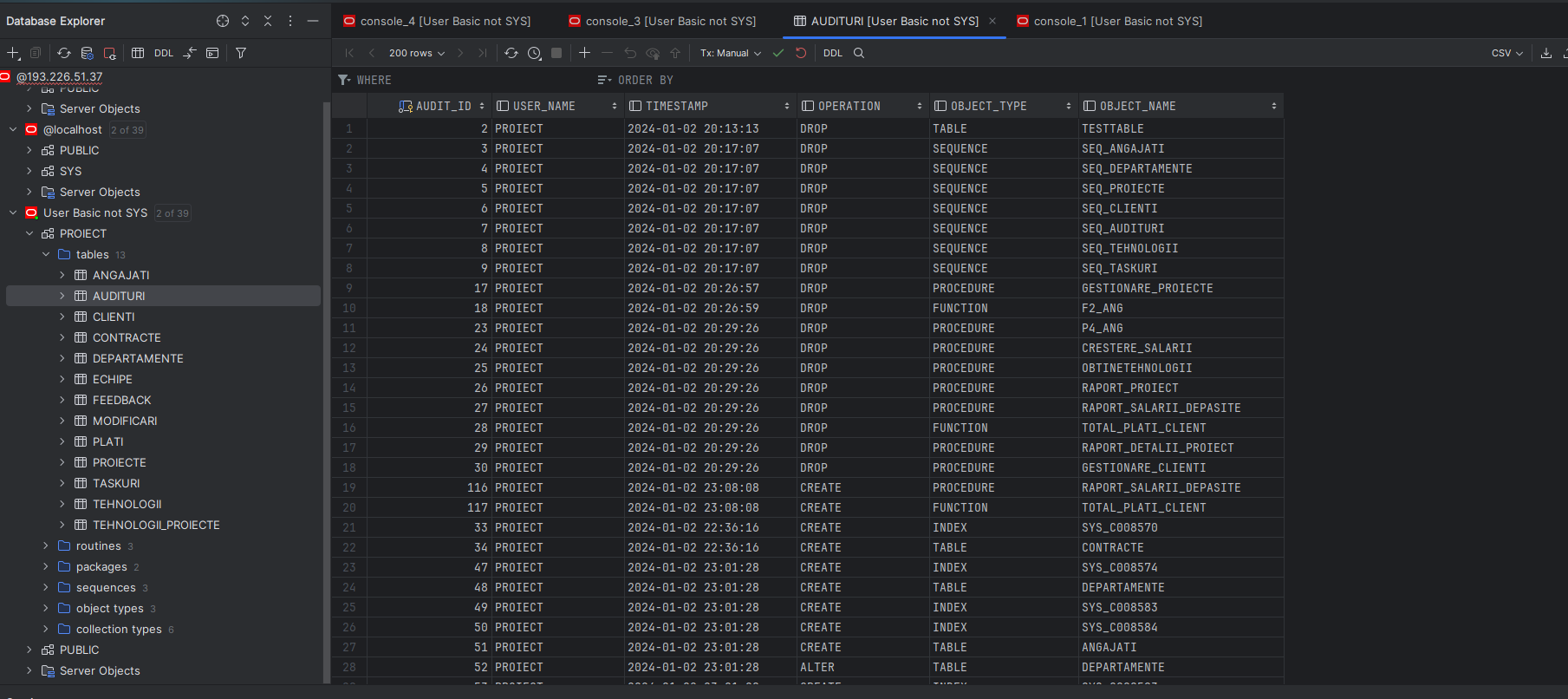


12.Definiți un trigger de tip LDD. Declanșați trigger-ul

Un trigger care inregistreaza in tabela Audituri toate modificarile la nivel de baza de date.

--LDD  
CREATE TABLE AUDITURI (  
 audit\_id INT PRIMARY KEY,  
 user\_name VARCHAR2(30),  
 timestamp DATE,  
 operation VARCHAR2(50),  
 object\_type VARCHAR2(50),  
 object\_name VARCHAR2(50)  
);  
CREATE SEQUENCE seq\_audit  
 START WITH 1  
 INCREMENT BY 1  
 NOCACHE  
 NOCYCLE;  
  
CREATE OR REPLACE TRIGGER ddl\_audit\_trigger  
AFTER DDL ON SCHEMA  
DECLARE  
BEGIN  
 INSERT INTO Audituri (audit\_id, user\_name, timestamp, operation, object\_type, object\_name)  
 VALUES (seq\_audit.NEXTVAL, ORA\_LOGIN\_USER, *SYSDATE*, ORA\_SYSEVENT, ORA\_DICT\_OBJ\_TYPE, ORA\_DICT\_OBJ\_NAME);  
END;  
/

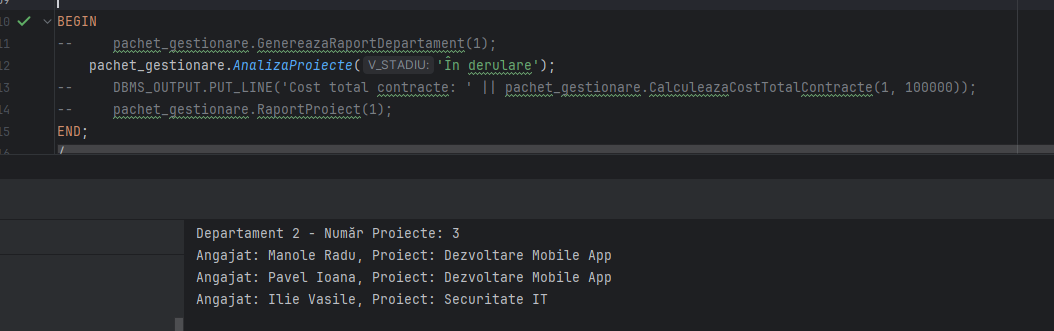
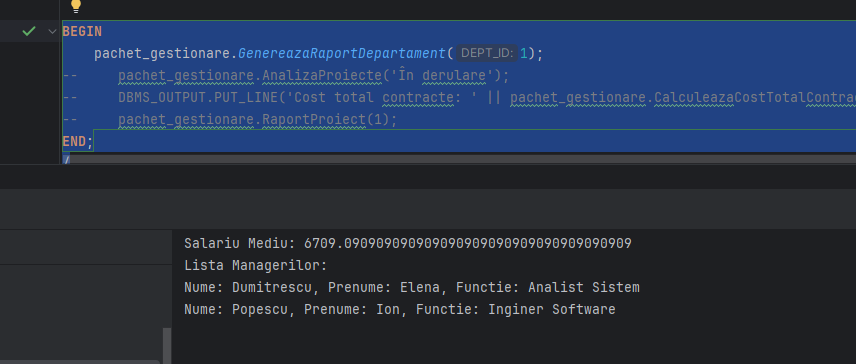
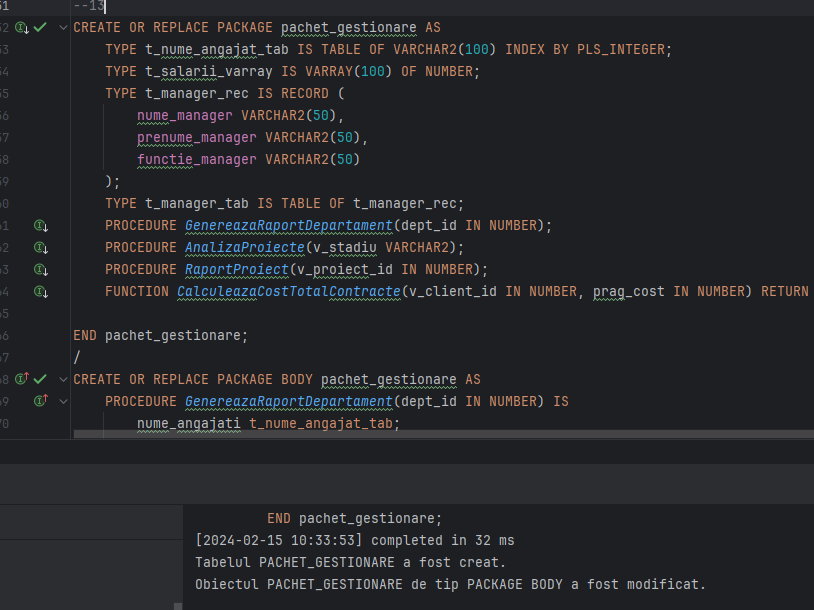




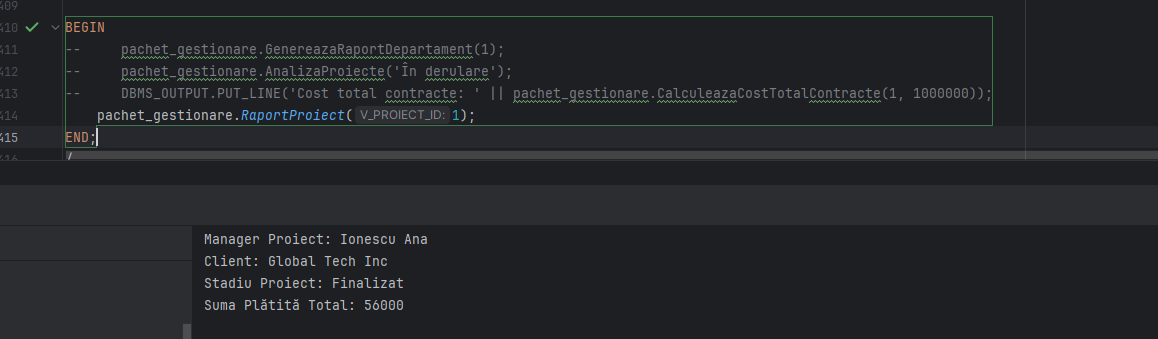
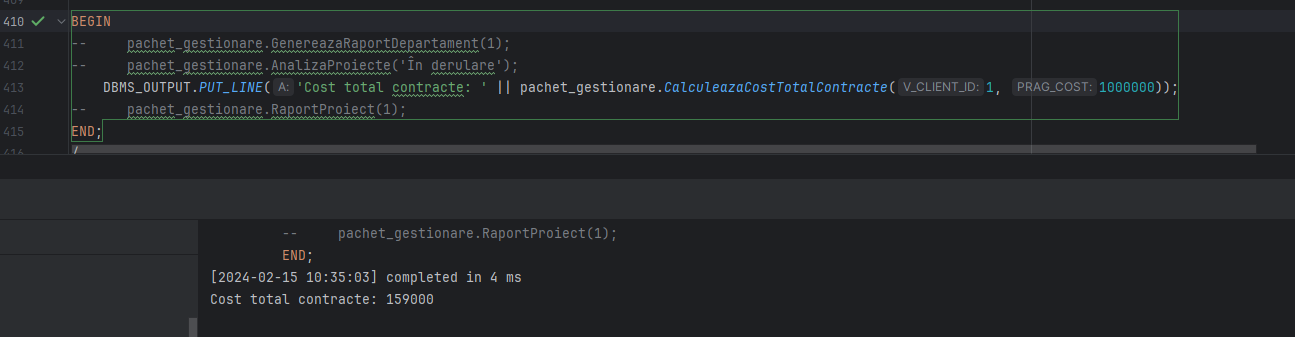
13. Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.

CREATE OR REPLACE PACKAGE pachet\_gestionare AS  
 TYPE t\_nume\_angajat\_tab IS TABLE OF VARCHAR2(100) INDEX BY PLS\_INTEGER;  
 TYPE t\_salarii\_varray IS VARRAY(100) OF NUMBER;  
 TYPE t\_manager\_rec IS RECORD (  
 nume\_manager VARCHAR2(50),  
 prenume\_manager VARCHAR2(50),  
 functie\_manager VARCHAR2(50)  
 );  
 TYPE t\_manager\_tab IS TABLE OF t\_manager\_rec;  
 PROCEDURE *GenereazaRaportDepartament*(dept\_id IN NUMBER);  
 PROCEDURE *AnalizaProiecte*(v\_stadiu VARCHAR2);  
 PROCEDURE *RaportProiect*(v\_proiect\_id IN NUMBER);  
 FUNCTION *CalculeazaCostTotalContracte*(v\_client\_id IN NUMBER, prag\_cost IN NUMBER) RETURN NUMBER;  
  
END pachet\_gestionare;  
/  
CREATE OR REPLACE PACKAGE BODY pachet\_gestionare AS  
 PROCEDURE *GenereazaRaportDepartament*(dept\_id IN NUMBER) IS  
 nume\_angajati t\_nume\_angajat\_tab;  
 salarii t\_salarii\_varray := t\_salarii\_varray();  
 manageri t\_manager\_tab;  
 salariu\_mediu NUMBER;  
 total\_salarii NUMBER := 0;  
 idx PLS\_INTEGER := 1;  
 BEGIN  
 FOR rec IN (SELECT nume || ' ' || prenume AS nume\_complet FROM Angajati WHERE departament\_id = dept\_id) LOOP  
 nume\_angajati(idx) := rec.nume\_complet;  
 idx := idx + 1;  
 END LOOP;  
  
 SELECT salariu  
 BULK COLLECT INTO salarii  
 FROM Angajati  
 WHERE departament\_id = dept\_id;  
  
 FOR i IN 1..salarii.COUNT LOOP  
 total\_salarii := total\_salarii + salarii(i);  
 END LOOP;  
  
 salariu\_mediu := CASE WHEN salarii.COUNT > 0 THEN total\_salarii / salarii.COUNT ELSE 0 END;  
  
 SELECT nume, prenume, functie  
 BULK COLLECT INTO manageri  
 FROM Angajati  
 WHERE angajat\_id IN (SELECT DISTINCT manager\_id FROM Angajati WHERE departament\_id = dept\_id AND manager\_id IS NOT NULL);  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Raport pentru Departamentul cu ID: ' || dept\_id);  
 DBMS\_OUTPUT.*PUT\_LINE*('Lista Angajatilor:');  
 idx := nume\_angajati.FIRST;  
 WHILE idx IS NOT NULL LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*(nume\_angajati(idx));  
 idx := nume\_angajati.*NEXT*(idx);  
 END LOOP;  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Salariu Mediu: ' || salariu\_mediu);  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Lista Managerilor:');  
 FOR i IN 1..manageri.COUNT LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('Nume: ' || manageri(i).nume\_manager || ', Prenume: ' || manageri(i).prenume\_manager || ', Functie: ' || manageri(i).functie\_manager);  
 END LOOP;  
 END *GenereazaRaportDepartament*;  
  
 PROCEDURE *AnalizaProiecte*(v\_stadiu VARCHAR2) IS  
 TYPE cursor\_dinamic IS REF CURSOR;  
 cursor\_departamente cursor\_dinamic;  
 departament\_id NUMBER;  
 numar\_proiecte NUMBER;  
 CURSOR cursor\_angajati(dept\_id NUMBER) IS  
 SELECT a.nume, a.PRENUME, p.nume\_proiect  
 FROM Angajati a  
 Join ECHIPE E on a.ANGAJAT\_ID = E.ID\_ANGAJAT  
 JOIN Proiecte p ON E.proiect\_id = p.proiect\_id  
 WHERE a.departament\_id = dept\_id AND p.stadiu = v\_stadiu;  
 BEGIN  
 OPEN cursor\_departamente FOR  
 SELECT d.departament\_id, *COUNT*(p.proiect\_id) AS numar\_proiecte  
 FROM Departamente d  
 JOIN Angajati a ON d.departament\_id = a.departament\_id  
 Join ECHIPE E on a.ANGAJAT\_ID = E.ID\_ANGAJAT  
 JOIN Proiecte p ON E.proiect\_id = p.proiect\_id  
 WHERE *upper*(p.stadiu) = *upper*(v\_stadiu)  
 GROUP BY d.departament\_id;  
 LOOP  
 FETCH cursor\_departamente INTO departament\_id, numar\_proiecte;  
 EXIT WHEN cursor\_departamente%NOTFOUND;  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Departament ' || departament\_id || ' - Număr Proiecte: ' || numar\_proiecte);  
 FOR REC IN cursor\_angajati(departament\_id) LOOP  
 DBMS\_OUTPUT.*PUT\_LINE*('Angajat: ' || REC.NUME ||' '|| REC.PRENUME || ', Proiect: ' || REC.NUME\_PROIECT);  
 end loop;  
 END LOOP;  
 CLOSE cursor\_departamente;  
 END *AnalizaProiecte*;  
  
 FUNCTION *CalculeazaCostTotalContracte*(v\_client\_id IN NUMBER, prag\_cost IN NUMBER) RETURN NUMBER IS  
 ClientInexistent EXCEPTION;  
 PragDepasit EXCEPTION;  
 cost\_total NUMBER := 0;  
 BEGIN  
 SELECT *SUM*(c.cost\_contract)  
 INTO cost\_total  
 FROM Contracte c  
 JOIN Proiecte p ON c.proiect\_id = p.proiect\_id  
 WHERE c.client\_id = v\_client\_id;  
  
 IF cost\_total IS NULL THEN  
 RAISE ClientInexistent;  
 END IF;  
  
 IF cost\_total > prag\_cost THEN  
 RAISE PragDepasit;  
 END IF;  
  
 RETURN cost\_total;  
 EXCEPTION  
 WHEN ClientInexistent THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Clientul nu există.');  
 RETURN NULL;  
 WHEN PragDepasit THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Pragul de cost a fost depășit.');  
 RETURN cost\_total;  
 END *CalculeazaCostTotalContracte*;  
  
 PROCEDURE *RaportProiect*(v\_proiect\_id IN NUMBER) IS  
 v\_nume\_proiect VARCHAR2(100);  
 v\_nume\_manager VARCHAR2(100);  
 v\_nume\_client VARCHAR2(100);  
 v\_stadiu\_proiect VARCHAR2(50);  
 v\_suma\_platita\_total DECIMAL(10,2);  
 BEGIN  
 SELECT p.nume\_proiect, a.nume || ' ' || a.prenume, cl.nume\_client, p.stadiu, *SUM*(pl.suma\_platita)  
 INTO v\_nume\_proiect, v\_nume\_manager, v\_nume\_client, v\_stadiu\_proiect, v\_suma\_platita\_total  
 FROM Proiecte p  
 JOIN Echipe e ON p.proiect\_id = e.proiect\_id  
 JOIN Angajati a ON e.id\_angajat = a.angajat\_id AND a.manager\_id IS NOT NULL  
 JOIN Clienti cl ON p.client\_id = cl.client\_id  
 JOIN Contracte c ON cl.client\_id = c.client\_id  
 JOIN Plati pl ON c.contract\_id = pl.contract\_id  
 WHERE p.proiect\_id = v\_proiect\_id  
 GROUP BY p.nume\_proiect, a.nume, a.prenume, cl.nume\_client, p.stadiu;  
  
 DBMS\_OUTPUT.*PUT\_LINE*('Nume Proiect: ' || v\_nume\_proiect);  
 DBMS\_OUTPUT.*PUT\_LINE*('Manager Proiect: ' || v\_nume\_manager);  
 DBMS\_OUTPUT.*PUT\_LINE*('Client: ' || v\_nume\_client);  
 DBMS\_OUTPUT.*PUT\_LINE*('Stadiu Proiect: ' || v\_stadiu\_proiect);  
 DBMS\_OUTPUT.*PUT\_LINE*('Suma Plătită Total: ' || v\_suma\_platita\_total);  
 EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Nu există date pentru proiectul cu ID-ul ' || v\_proiect\_id);  
 WHEN TOO\_MANY\_ROWS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție: Prea multe rânduri returnate pentru proiectul cu ID-ul ' || v\_proiect\_id);  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.*PUT\_LINE*('Excepție Necunoscută: ' || *SQLERRM*);  
 END *RaportProiect*;  
  
END pachet\_gestionare;  
/

BEGIN  
 pachet\_gestionare.*GenereazaRaportDepartament*(1);  
 pachet\_gestionare.*AnalizaProiecte*('În derulare');  
 DBMS\_OUTPUT.*PUT\_LINE*('Cost total contracte: ' || pachet\_gestionare.*CalculeazaCostTotalContracte*(1, 1000000));  
 pachet\_gestionare.*RaportProiect*(1);  
END;  
/



14. Definiți un pachet care să includă tipuri de date complexe și obiecte necesare unui flux de acțiuni integrate, specifice bazei de date definite (minim 2 tipuri de date, minim 2 funcții, minim 2 proceduri).



CREATE OR REPLACE PACKAGE GestionareProiecte IS  
  
 TYPE t\_tehnologie\_info IS RECORD (  
 tehnologie\_id INT,  
 nume\_tehnologie VARCHAR2(100),  
 descriere VARCHAR2(200)  
 );  
  
 TYPE t\_tehnologie\_tab IS TABLE OF t\_tehnologie\_info;  
  
 FUNCTION *TehnologieExista*(v\_proiect\_id INT, v\_tehnologie\_id INT) RETURN BOOLEAN;  
 FUNCTION *ObtineTehnologiiProiect*(v\_proiect\_id INT) RETURN t\_tehnologie\_tab;  
 PROCEDURE *EvalueazaStadiuProiect*(v\_proiect\_id INT);  
 PROCEDURE *AdaugaSiEvalueazaProiect*(v\_proiect\_id INT, tehnologii\_noi t\_tehnologie\_tab);  
  
END GestionareProiecte;  
/  
CREATE OR REPLACE PACKAGE BODY GestionareProiecte IS  
  
 FUNCTION *TehnologieExista*(v\_proiect\_id INT, v\_tehnologie\_id INT) RETURN BOOLEAN IS  
 v\_tehnologie INT;  
 BEGIN  
 SELECT *COUNT*(\*)  
 into v\_tehnologie  
 FROM Tehnologii\_Proiecte  
 WHERE PROIECTID = v\_proiect\_id AND TEHNOLOGIEID = v\_tehnologie\_id;  
  
 RETURN (v\_tehnologie > 0);  
 END *TehnologieExista*;  
  
 FUNCTION *ObtineTehnologiiProiect*(v\_proiect\_id INT) RETURN t\_tehnologie\_tab IS  
 t\_tehnologii t\_tehnologie\_tab := t\_tehnologie\_tab();  
 BEGIN  
 SELECT tehnologie\_id, nume\_tehnologie, descriere  
 BULK COLLECT INTO t\_tehnologii  
 FROM Tehnologii  
 WHERE tehnologie\_id IN (SELECT tehnologie\_id FROM Tehnologii\_Proiecte WHERE PROIECTID = v\_proiect\_id);  
  
 RETURN t\_tehnologii;  
 END *ObtineTehnologiiProiect*;  
  
PROCEDURE *EvalueazaStadiuProiect*(v\_proiect\_id INT) IS  
 stadiu\_nou VARCHAR2(50);  
 stadiu\_curent VARCHAR2(50);  
BEGIN  
 select STADIU  
 into stadiu\_curent  
 from PROIECTE  
 where PROIECT\_ID = v\_proiect\_id;  
if *upper*(stadiu\_curent) != 'FINALIZAT' then  
 IF *ObtineTehnologiiProiect*(v\_proiect\_id).COUNT > 3 THEN  
 stadiu\_nou := 'In derulare';  
 ELSE  
 stadiu\_nou := 'Început';  
 END IF;  
  
 UPDATE Proiecte  
 SET stadiu = stadiu\_nou  
 WHERE proiect\_id = v\_proiect\_id;  
end if;  
END *EvalueazaStadiuProiect*;

PROCEDURE *AdaugaSiEvalueazaProiect*(v\_proiect\_id INT, tehnologii\_noi t\_tehnologie\_tab) IS  
 BEGIN  
 FOR i IN 1..tehnologii\_noi.COUNT LOOP  
 IF NOT *TehnologieExista*(v\_proiect\_id, tehnologii\_noi(i).tehnologie\_id) THEN  
 Insert Into TEHNOLOGII(tehnologie\_id, nume\_tehnologie, descriere) VALUES (tehnologii\_noi(i).tehnologie\_id,tehnologii\_noi(i).NUME\_TEHNOLOGIE,tehnologii\_noi(i).DESCRIERE);  
 INSERT INTO Tehnologii\_Proiecte (PROIECTID, TEHNOLOGIEID)  
 VALUES (v\_proiect\_id, tehnologii\_noi(i).tehnologie\_id);  
 END IF;  
 END LOOP;  
 *EvalueazaStadiuProiect*(v\_proiect\_id);  
 END *AdaugaSiEvalueazaProiect*;  
  
END GestionareProiecte;  
/  
  
DECLARE  
 tehnologii\_noi PROIECT.GESTIONAREPROIECTE.T\_TEHNOLOGIE\_TAB;  
BEGIN  
 tehnologii\_noi := PROIECT.GESTIONAREPROIECTE.T\_TEHNOLOGIE\_TAB();  
 tehnologii\_noi.extend;  
 tehnologii\_noi(1).tehnologie\_id := 12;  
 tehnologii\_noi(1).nume\_tehnologie := 'C#';  
 tehnologii\_noi(1).descriere := 'Limbaj de programare';  
 GestionareProiecte.*AdaugaSiEvalueazaProiect*(1, tehnologii\_noi);  
END;  
/

