**PROIECT SGBD**

-Galerie de arta-

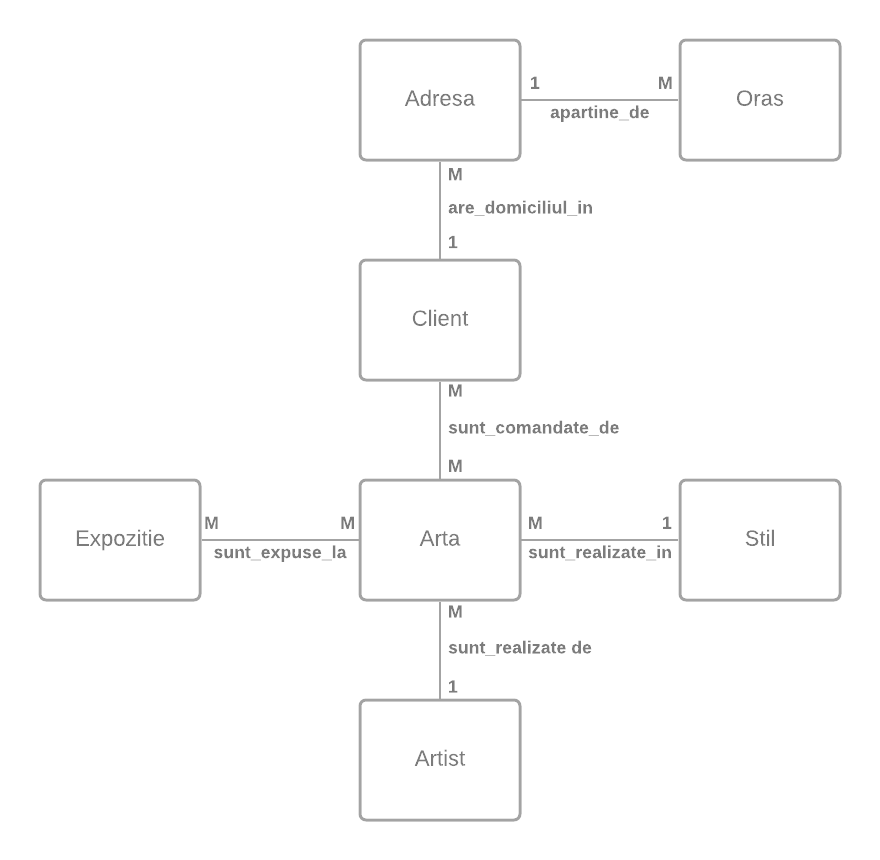
-Trandafir Dragos-

Grupa 243

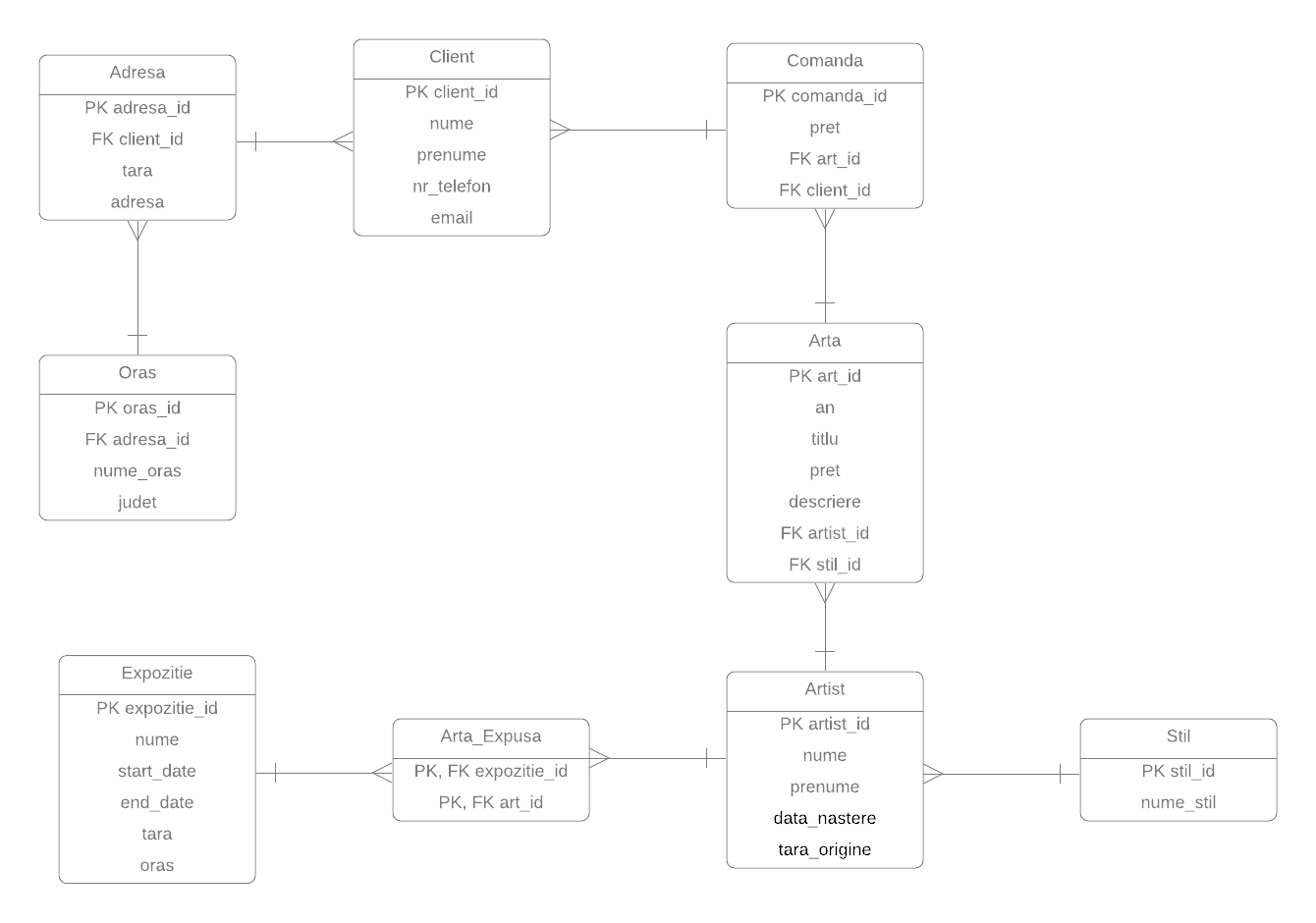
1. Prezentați pe scurt baza de date (utilitatea ei).

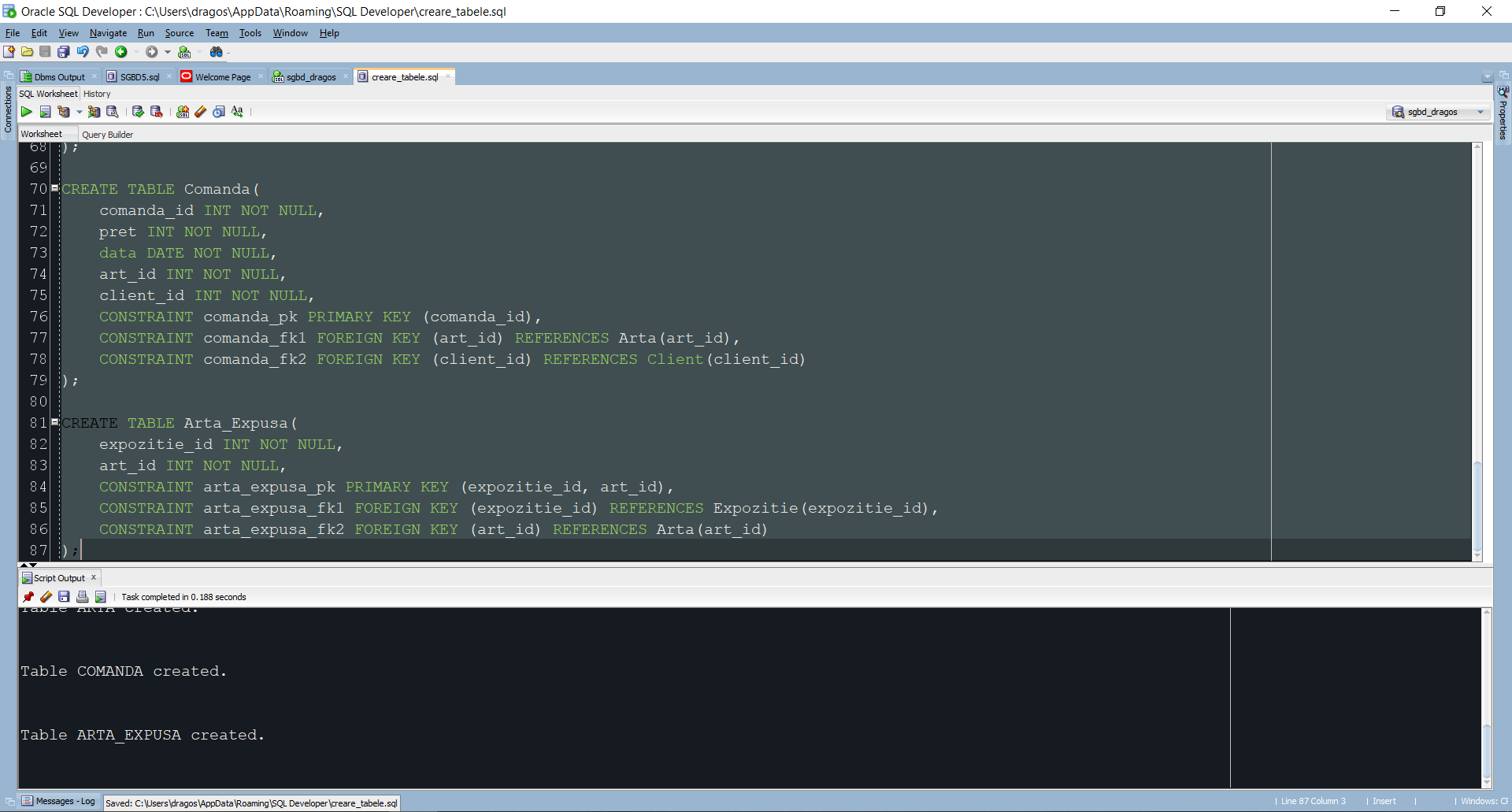
In cadrul acestui proiect prezint baza de date a unei galerii. Aceasta contine artisti, arta, stilul in care e realizata, expozitii, client si adresa acestora. Artistii isi pot expune arta in cadrul unor expozitii si/sau sa o vanda potentialilor client. Folosesc doua tabele asociative pentru a gestiona comenzile si arta care ajunge sa fie expusa.

2. Realizați diagrama entitate-relație (ERD).



1. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



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

DROP TABLE Artist;

DROP TABLE Stil;

DROP TABLE Expozitie;

DROP TABLE Client;

DROP TABLE Adresa;

DROP TABLE Arta;

DROP TABLE Comanda;

DROP TABLE Arta\_Expusa;

CREATE TABLE Artist (

artist\_id INT NOT NULL,

nume VARCHAR2(30) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

data\_nastere DATE,

tara\_origine VARCHAR2(20),

CONSTRAINT artist\_pk PRIMARY KEY (artist\_id)

);

CREATE TABLE Stil (

stil\_id INT NOT NULL,

nume\_stil VARCHAR2(20),

CONSTRAINT stil\_pk PRIMARY KEY (stil\_id)

);

CREATE TABLE Expozitie (

expozitie\_id INT NOT NULL,

nume VARCHAR2(30) NOT NULL,

start\_date DATE,

end\_date DATE,

tara VARCHAR2(20),

oras VARCHAR2(20),

CONSTRAINT expozitie\_pk PRIMARY KEY (expozitie\_id)

);

CREATE TABLE Client (

client\_id INT NOT NULL,

nume VARCHAR2(30) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

nr\_telefon VARCHAR2(10),

email VARCHAR2(40),

CONSTRAINT client\_pk PRIMARY KEY (client\_id)

);

CREATE TABLE Adresa (

adresa\_id INT NOT NULL,

client\_id INT NOT NULL,

tara VARCHAR2(20),

oras VARCHAR2(20),

adresa VARCHAR2(40),

CONSTRAINT adresa\_pk PRIMARY KEY (adresa\_id),

CONSTRAINT adresa\_fk FOREIGN KEY (client\_id) REFERENCES Client(client\_id)

);

CREATE TABLE Arta(

art\_id INT NOT NULL,

an INT,

titlu VARCHAR2(20),

pret INT,

descriere VARCHAR2(100),

artist\_id INT NOT NULL,

stil\_id INT NOT NULL,

CONSTRAINT arta\_pk PRIMARY KEY (art\_id),

CONSTRAINT arta\_fk1 FOREIGN KEY (artist\_id) REFERENCES Artist(artist\_id),

CONSTRAINT arta\_fk2 FOREIGN KEY (stil\_id) REFERENCES Stil(stil\_id)

);

CREATE TABLE Comanda(

comanda\_id INT NOT NULL,

data DATE NOT NULL,

art\_id INT NOT NULL,

client\_id INT NOT NULL,

CONSTRAINT comanda\_pk PRIMARY KEY (comanda\_id),

CONSTRAINT comanda\_fk1 FOREIGN KEY (art\_id) REFERENCES Arta(art\_id),

CONSTRAINT comanda\_fk2 FOREIGN KEY (client\_id) REFERENCES Client(client\_id)

);

CREATE TABLE Arta\_Expusa(

expozitie\_id INT NOT NULL,

art\_id INT NOT NULL,

CONSTRAINT arta\_expusa\_pk PRIMARY KEY (expozitie\_id, art\_id),

CONSTRAINT arta\_expusa\_fk1 FOREIGN KEY (expozitie\_id) REFERENCES Expozitie(expozitie\_id),

CONSTRAINT arta\_expusa\_fk2 FOREIGN KEY (art\_id) REFERENCES Arta(art\_id)

);

--ALTER TABLE Comanda

--DROP COLUMN pret;

--ALTER TABLE Adresa

--DROP COLUMN oras;

CREATE TABLE ORAS (

oras\_id INT NOT NULL,

adresa\_id INT NOT NULL,

nume\_oras VARCHAR2(30),

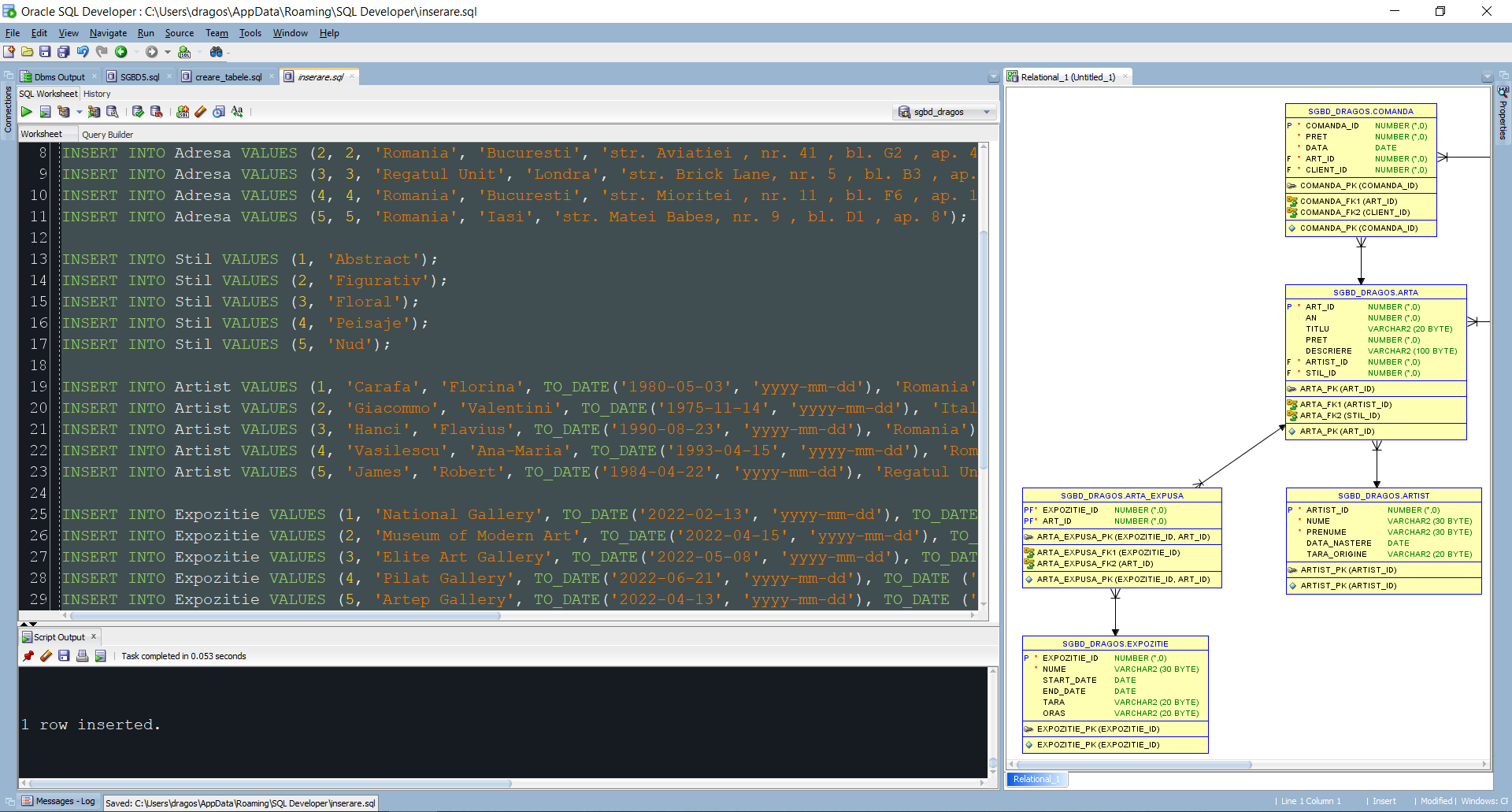
judet VARCHAR2(30),

CONSTRAINT oras\_pk PRIMARY KEY (oras\_id),

CONSTRAINT oras\_fk FOREIGN KEY (adresa\_id) REFERENCES Adresa(adresa\_id)

);

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



INSERT INTO Client VALUES (1,'Popescu','Ion','0744258321','ion.popescu34@gmail.com');

INSERT INTO Client VALUES (2,'Georgescu','Mariana','0751088235','mariangeorgescu12@gmail.com');

INSERT INTO Client VALUES (3,'Axinte','Daniela','0745593834','danielaa24@yahoo.com');

INSERT INTO Client VALUES (4,'Mihail','Alexandru','0761445958','mihailalex6@yahoo.com');

INSERT INTO Client VALUES (5,'Birica','Andrei','0753210045','andrewbir99@gmail.com');

INSERT INTO Adresa VALUES (1, 1, 'Romania', 'str. Iancului, nr. 23, bl. A5 , ap. 22');

INSERT INTO Adresa VALUES (2, 2, 'Romania', 'str. Aviatiei , nr. 41 , bl. G2 , ap. 44');

INSERT INTO Adresa VALUES (3, 3, 'Regatul Unit', 'str. Brick Lane, nr. 5 , bl. B3 , ap.5A');

INSERT INTO Adresa VALUES (4, 4, 'Romania', 'str. Mioritei , nr. 11 , bl. F6 , ap. 10');

INSERT INTO Adresa VALUES (5, 5, 'Romania', 'str. Matei Babes, nr. 9 , bl. D1 , ap. 8');

INSERT INTO Oras VALUES (1, 1, 'Teleorman', 'Alexandria');

INSERT INTO Oras VALUES (2, 2, 'Bucuresti', 'Bucuresti');

INSERT INTO Oras VALUES (3, 3, 'Londra', 'Londra Mare');

INSERT INTO Oras VALUES (4, 4, 'Bucuresti', 'Bucuresti');

INSERT INTO Oras VALUES (5, 5, 'Iasi', 'Iasi');

INSERT INTO Stil VALUES (1, 'Abstract');

INSERT INTO Stil VALUES (2, 'Figurativ');

INSERT INTO Stil VALUES (3, 'Floral');

INSERT INTO Stil VALUES (4, 'Peisaje');

INSERT INTO Stil VALUES (5, 'Nud');

INSERT INTO Artist VALUES (1, 'Carafa', 'Florina', TO\_DATE('1980-05-03', 'yyyy-mm-dd'), 'Romania');

INSERT INTO Artist VALUES (2, 'Giacommo', 'Valentini', TO\_DATE('1975-11-14', 'yyyy-mm-dd'), 'Italia');

INSERT INTO Artist VALUES (3, 'Hanci', 'Flavius', TO\_DATE('1990-08-23', 'yyyy-mm-dd'), 'Romania');

INSERT INTO Artist VALUES (4, 'Vasilescu', 'Ana-Maria', TO\_DATE('1993-04-15', 'yyyy-mm-dd'), 'Romania');

INSERT INTO Artist VALUES (5, 'James', 'Robert', TO\_DATE('1984-04-22', 'yyyy-mm-dd'), 'Regatul Unit');

INSERT INTO Expozitie VALUES (1, 'National Gallery', TO\_DATE('2018-02-13', 'yyyy-mm-dd'), TO\_DATE ('2018-02-16', 'yyyy-mm-dd'), 'Regatul Unit', 'Londra');

INSERT INTO Expozitie VALUES (2, 'Museum of Modern Art', TO\_DATE('2019-04-15', 'yyyy-mm-dd'), TO\_DATE ('2019-04-19', 'yyyy-mm-dd'), 'SUA', 'New York');

INSERT INTO Expozitie VALUES (3, 'Elite Art Gallery', TO\_DATE('2020-05-08', 'yyyy-mm-dd'), TO\_DATE ('2020-05-13', 'yyyy-mm-dd'), 'Romania', 'Bucuresti');

INSERT INTO Expozitie VALUES (4, 'Pilat Gallery', TO\_DATE('2021-06-21', 'yyyy-mm-dd'), TO\_DATE ('2021-06-24', 'yyyy-mm-dd'), 'Romania', 'Bucuresti');

INSERT INTO Expozitie VALUES (5, 'Artep Gallery', TO\_DATE('2021-04-13', 'yyyy-mm-dd'), TO\_DATE ('2021-04-16', 'yyyy-mm-dd'), 'Romania', 'Iasi');

INSERT INTO Arta VALUES (1, 2020, 'Love story', 2000, null, 1, 1);

INSERT INTO Arta VALUES (2, 2016, 'Incatusare', 3500, null, 2, 1);

INSERT INTO Arta VALUES (3, 2018, 'The Rust', 50000, 'The Rust este o lucrare inspirata din natura si imperfectiunea ei.', 3, 2);

INSERT INTO Arta VALUES (4, 2014, 'Fetita cu coronita', 30000, 'Fetita cu coronita , ulei panza', 4, 2);

INSERT INTO Arta VALUES (5, 2017, 'Freedom', 4000, null, 5, 3);

INSERT INTO Arta VALUES (6, 2021, 'Floral Bliss', 2450, 'Lucrarea este pictata pe panza cu sasiu, cu sistem de prindere vertical', 1, 3);

INSERT INTO Arta VALUES (7, 2021, 'Linistea Naturii', 3250, 'Pictura din panza de bumbac pe sasiu de lemn cu sistem de prindere, culori acrilice clasice, vernis.', 2, 4);

INSERT INTO Arta VALUES (8, 2019, 'Amintiri in Verde', 20000, null, 3, 4);

INSERT INTO Arta VALUES (9, 2020, 'Genesis', 30000, 'Adam si Eva.', 4, 5);

INSERT INTO Arta VALUES (10,2020, 'Melting lovers', 25000, null, 5, 5);

INSERT INTO Arta VALUES (11,2021, 'Elena', 4500, null, 1, 5);

INSERT INTO Arta VALUES (12,2021, 'Mintea Libera', 3750, null, 2, 4);

INSERT INTO Arta VALUES (13,2019, 'Parfum', 13500, null, 3, 3);

INSERT INTO Arta VALUES (14,2020, 'Natalie', 10000, null, 4, 2);

INSERT INTO Arta VALUES (15,2019, 'Fascinatie', 9000, 'Pictura vine intinsa pe sasiu profesional din lemn, cu pene de intindere.', 5, 1);

INSERT INTO Comanda VALUES (1, TO\_DATE('2021-06-21', 'yyyy-mm-dd'), 1, 3);

INSERT INTO Comanda VALUES (2, TO\_DATE('2017-03-14', 'yyyy-mm-dd'), 2, 4);

INSERT INTO Comanda VALUES (3, TO\_DATE('2019-06-11', 'yyyy-mm-dd'), 3, 2);

INSERT INTO Comanda VALUES (4, TO\_DATE('2019-11-30', 'yyyy-mm-dd'), 4, 3);

INSERT INTO Comanda VALUES (5, TO\_DATE('2017-11-30', 'yyyy-mm-dd'), 5, 3);

INSERT INTO Comanda VALUES (6, TO\_DATE('2021-08-05', 'yyyy-mm-dd'), 6, 5);

INSERT INTO Comanda VALUES (7, TO\_DATE('2021-09-24', 'yyyy-mm-dd'), 7, 1);

INSERT INTO Comanda VALUES (8, TO\_DATE('2020-01-20', 'yyyy-mm-dd'), 8, 2);

INSERT INTO Comanda VALUES (9, TO\_DATE('2021-01-05', 'yyyy-mm-dd'), 9, 4);

INSERT INTO Comanda VALUES (10, TO\_DATE('2020-05-13', 'yyyy-mm-dd'), 10, 1);

INSERT INTO Arta\_Expusa VALUES (1, 3);

INSERT INTO Arta\_Expusa VALUES (1, 4);

INSERT INTO Arta\_Expusa VALUES (2, 8);

INSERT INTO Arta\_Expusa VALUES (2, 15);

INSERT INTO Arta\_Expusa VALUES (3, 9);

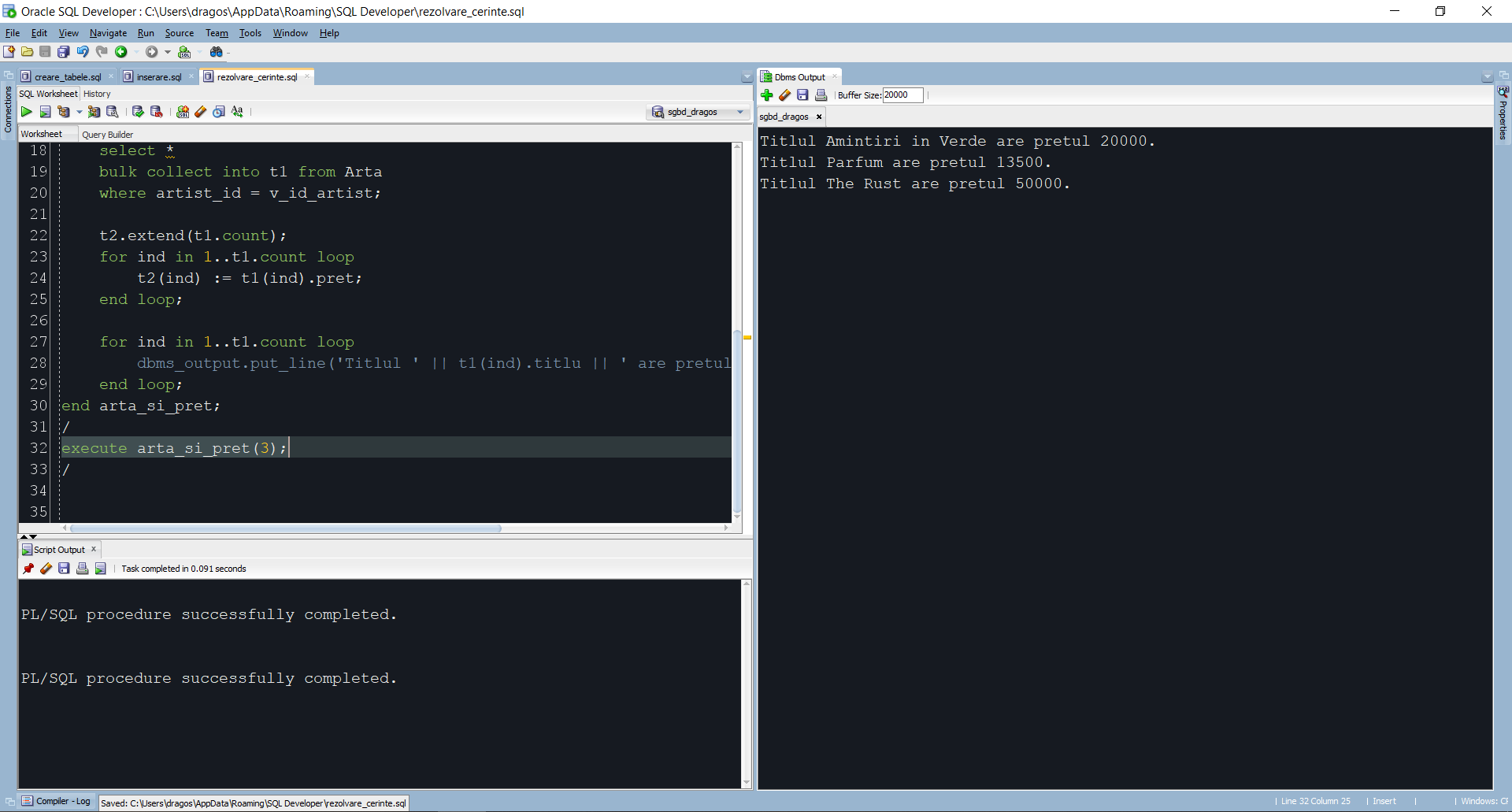
INSERT INTO Arta\_Expusa VALUES (3, 10);

INSERT INTO Arta\_Expusa VALUES (4, 11);

INSERT INTO Arta\_Expusa VALUES (4, 12);

INSERT INTO Arta\_Expusa VALUES (5, 6);

INSERT INTO Arta\_Expusa VALUES (5, 7);

6. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze două tipuri de colecție studiate. Apelați subprogramul.

--Pentru un id de artist introdus la tastatura, sa se afiseze ce piese de arta a creat si pretul acestora.

create or replace procedure arta\_si\_pret

(v\_id\_artist Artist.artist\_id%type)

as

type tablou\_index is table of Arta%rowtype index by pls\_integer;

t1 tablou\_index;

--v\_art\_id Arta.art\_id%type;

type varr is varray(10) of INTEGER;

t2 varr := varr();

--i number;

begin

--i := 0;

select \*

bulk collect into t1 from Arta

where artist\_id = v\_id\_artist;

t2.extend(t1.count);

for ind in 1..t1.count loop

t2(ind) := t1(ind).pret;

end loop;

for ind in 1..t1.count loop

dbms\_output.put\_line('Titlul ' || t1(ind).titlu || ' are pretul ' || t2(ind) ||'.');

end loop;

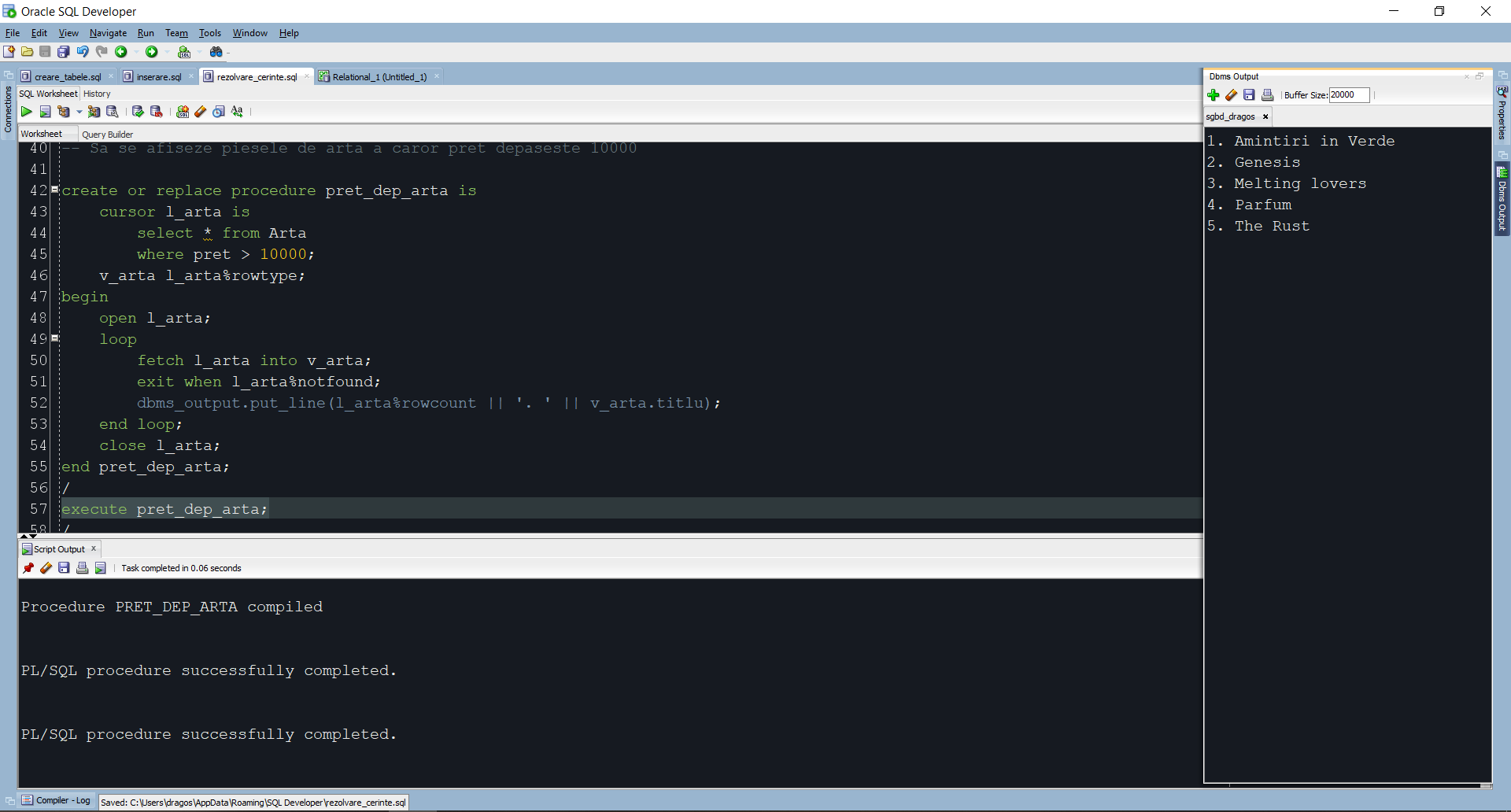
end arta\_si\_pret;

/

execute arta\_si\_pret(3);

/

7. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.



-- Sa se afiseze piesele de arta a caror pret depaseste 10000

create or replace procedure pret\_dep\_arta is

cursor l\_arta is

select \* from Arta

where pret > 10000;

v\_arta l\_arta%rowtype;

begin

open l\_arta;

loop

fetch l\_arta into v\_arta;

exit when l\_arta%notfound;

dbms\_output.put\_line(l\_arta%rowcount || '. ' || v\_arta.titlu);

end loop;

close l\_arta;

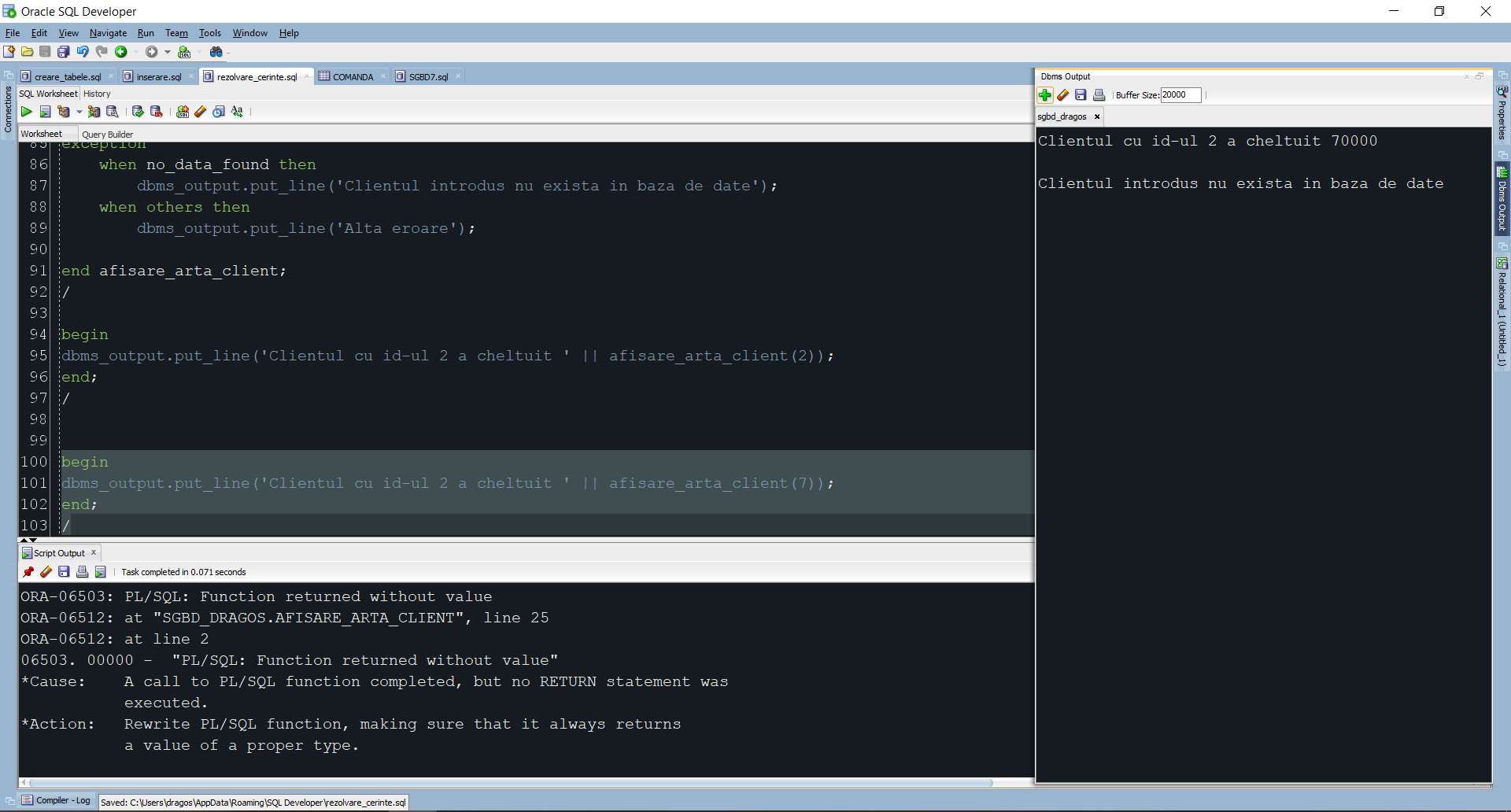
end pret\_dep\_arta;

/

execute pret\_dep\_arta;

/

8. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.



-- Pentru id-ul unui client dat sa se afiseze cat a cheltuit in total pe piesele pe care

-- le-a comandat

create or replace function afisare\_arta\_client

(v\_client\_id Client.client\_id%type)

return number is

rezultat number;

clientel number;

begin

select client\_id into clientel

from Client

where client\_id = v\_client\_id;

select sum (a.pret) into rezultat

from Arta a

join Comanda cc on (a.art\_id = cc.art\_id)

join Client c on (cc.client\_id = c.client\_id)

where c.client\_id = v\_client\_id;

return rezultat;

exception

when no\_data\_found then

dbms\_output.put\_line('Clientul introdus nu exista in baza de date');

when others then

dbms\_output.put\_line('Alta eroare');

end afisare\_arta\_client;

/

begin

dbms\_output.put\_line('Clientul cu id-ul 2 a cheltuit ' || afisare\_arta\_client(2));

end;

/

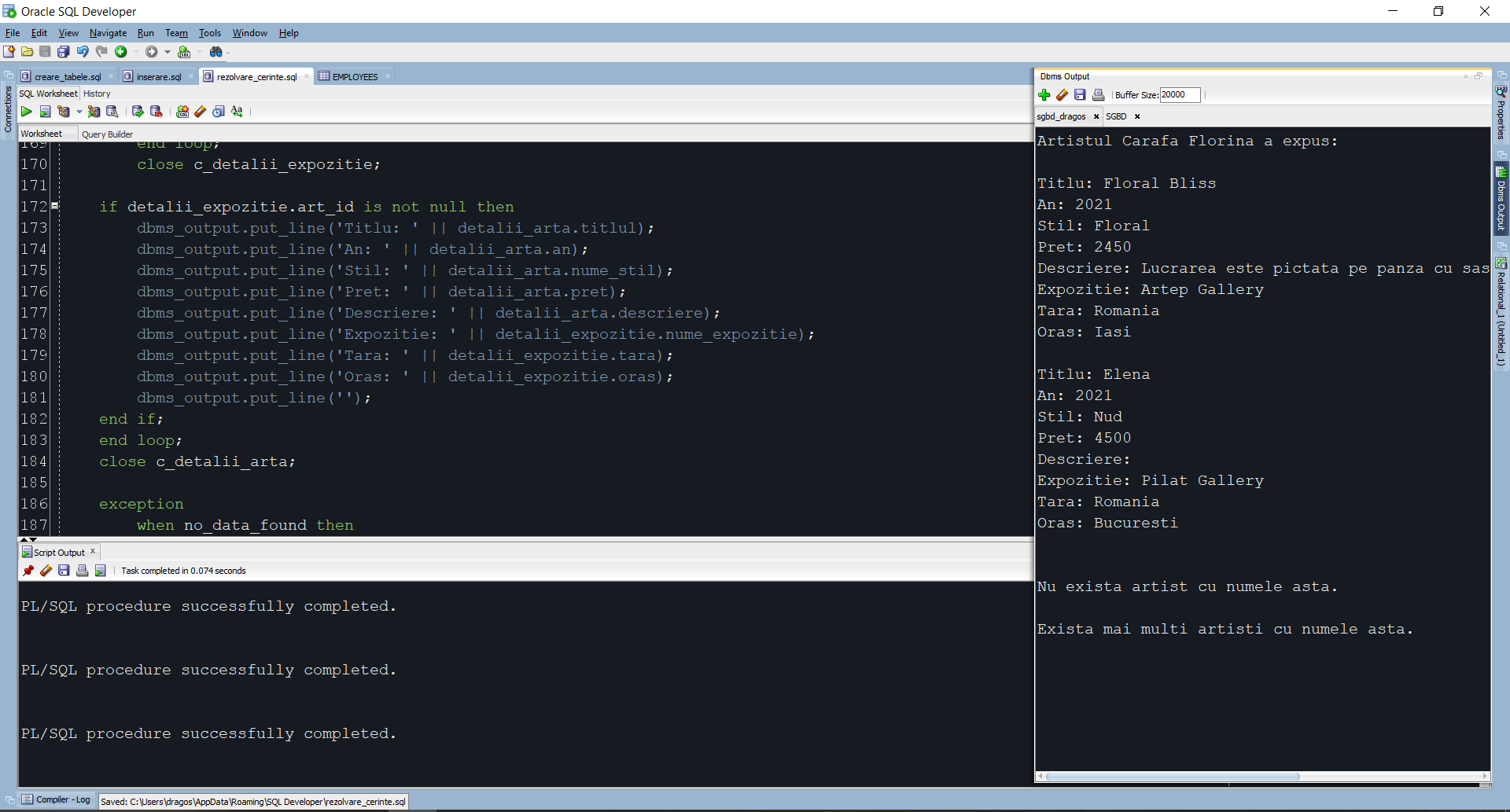
begin

dbms\_output.put\_line('Clientul cu id-ul 7 a cheltuit ' || afisare\_arta\_client(7));

end;

/

9. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat 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.



-- Pentru un artist dat sa se afiseze detaliile pentru picturile expuse si a expozitiilor din care fac parte.

-- (an, titlu, pret, descriere, stil, nume expozitie, tara, oras)

create or replace procedure artstyle

(v\_nume Artist.nume%type, v\_prenume Artist.prenume%type)

is

v\_artist\_id Artist.artist\_id%type;

type t\_detalii is record(

artist\_id Artist.artist\_id%type,

art\_id Arta.art\_id%type,

titlul Arta.titlu%type,

an Arta.an%type,

pret Arta.pret%type,

descriere Arta.descriere%type,

nume\_stil Stil.nume\_stil%type,

nume\_expozitie Expozitie.nume%type,

tara Expozitie.tara%type,

oras Expozitie.oras%type

--art\_id Arta\_expusa.art\_id%type

);

detalii t\_detalii;

cursor c\_detalii return t\_detalii is

select ar.artist\_id, a.art\_id, a.titlu, a.an, a.pret, a.descriere, s.nume\_stil, e.nume, e.tara, e.oras

from Artist ar join Arta a on (ar.artist\_id = a.artist\_id)

join Stil s on (a.stil\_id = s.stil\_id)

join Arta\_expusa ae on (a.art\_id = ae.art\_id)

join Expozitie e on (e.expozitie\_id = ae.expozitie\_id)

where ar.artist\_id = v\_artist\_id;

begin

select artist\_id into v\_artist\_id

from Artist

where lower(nume) like lower(v\_nume)

and lower(prenume) like lower(v\_prenume);

dbms\_output.put\_line('Artistul ' || v\_nume ||' ' || v\_prenume || ' a expus: ');

dbms\_output.put\_line('');

open c\_detalii;

loop

fetch c\_detalii into detalii;

exit when c\_detalii%notfound;

if detalii.art\_id is not null then

dbms\_output.put\_line('Titlu: ' || detalii.titlul);

dbms\_output.put\_line('An: ' || detalii.an);

dbms\_output.put\_line('Stil: ' || detalii.nume\_stil);

dbms\_output.put\_line('Pret: ' || detalii.pret);

dbms\_output.put\_line('Descriere: ' || detalii.descriere);

dbms\_output.put\_line('Expozitie: ' || detalii.nume\_expozitie);

dbms\_output.put\_line('Tara: ' || detalii.tara);

dbms\_output.put\_line('Oras: ' || detalii.oras);

dbms\_output.put\_line('');

end if;

end loop;

close c\_detalii;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista artist cu numele asta.');

when too\_many\_rows then

dbms\_output.put\_line('Exista mai multi artisti cu numele asta.');

when others then

dbms\_output.put\_line('Alte erori');

end artstyle;

/

-- input corect

execute artstyle('Carafa','Florina');

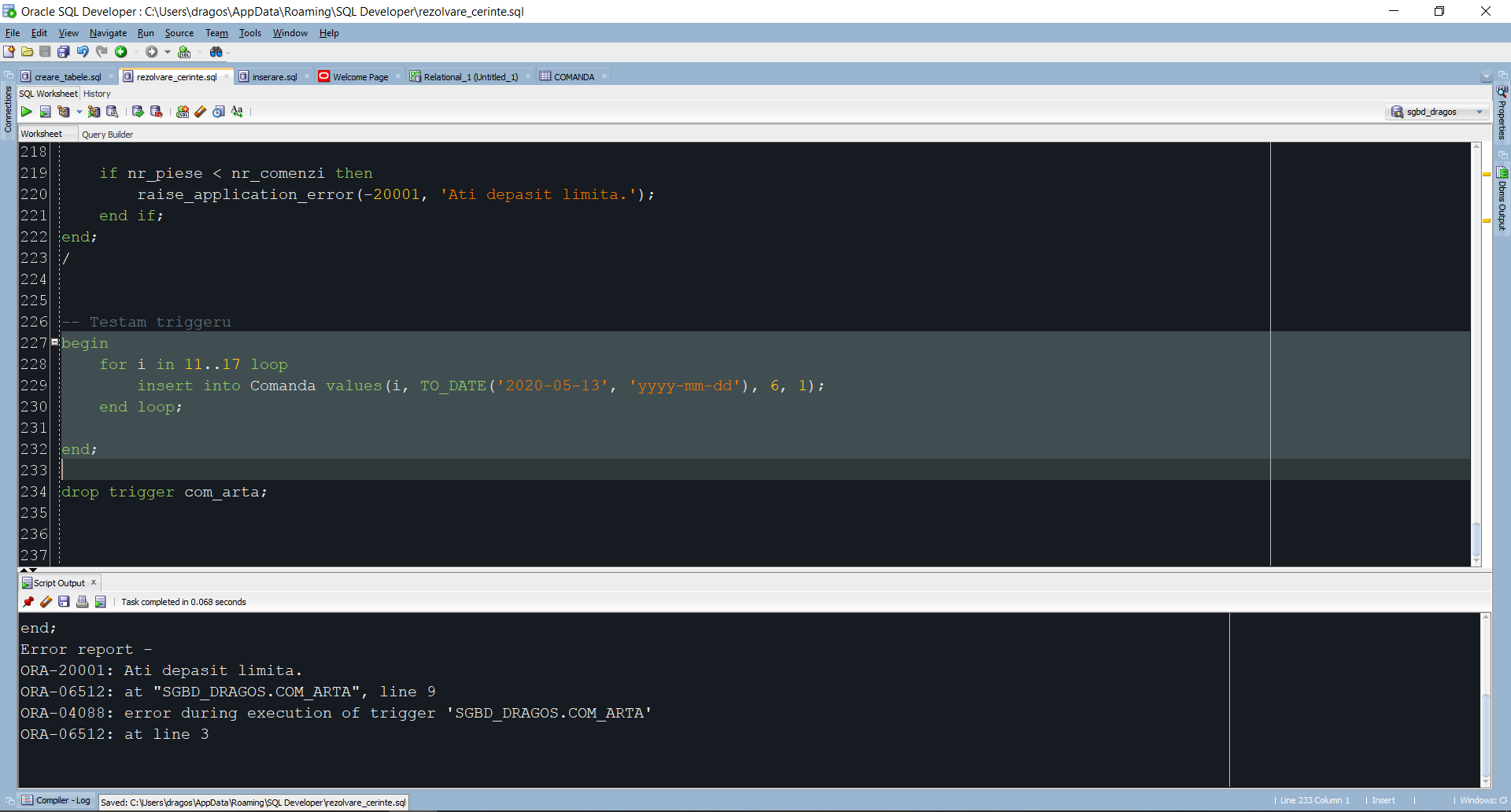
-- nu exista acest artist

execute artstyle('George','Clooney');

-- mai multi cu acest nume

execute artstyle('%', '%');

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



-- Sa nu se poata introduce/edita piese de arta cu pret sub 250

create or replace trigger com\_arta

before insert on Comanda

declare

nr\_comenzi INT;

nr\_piese INT;

begin

select count(comanda\_id) into nr\_comenzi from Comanda;

select count(art\_id) into nr\_piese from Arta;

if nr\_piese < nr\_comenzi then

raise\_application\_error(-20001, 'Ati depasit limita.');

end if;

end;

/

-- Testam triggeru

begin

for i in 11..17 loop

insert into Comanda values(i, TO\_DATE('2020-05-13', 'yyyy-mm-dd'), 6, 1);

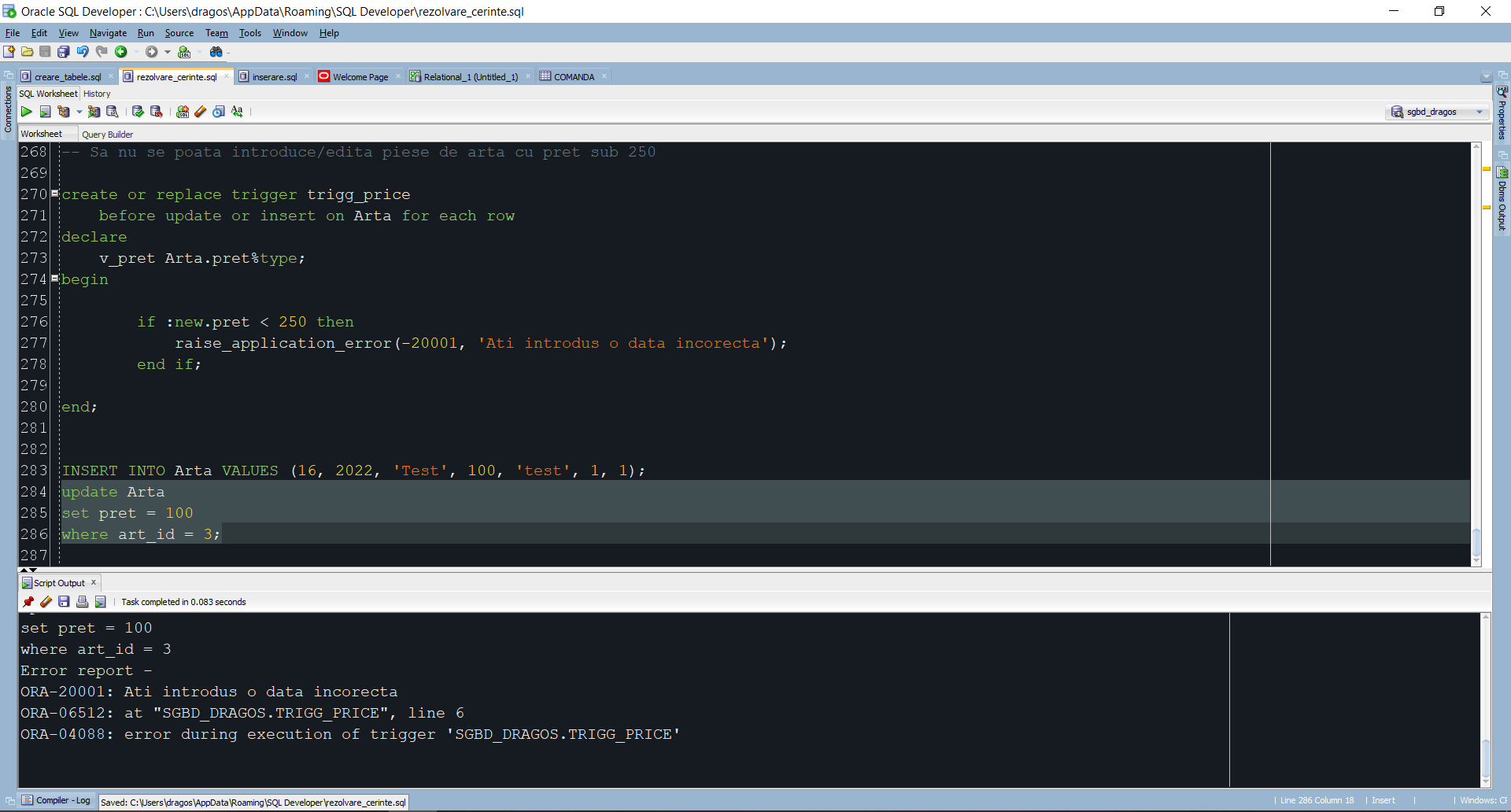
end loop;

end;

delete from Comanda where comanda\_id > 10;

drop trigger com\_arta;

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



-- Sa nu se poata introduce/edita piese de arta cu pret sub 250

create or replace trigger trigg\_price

before update or insert on Arta for each row

declare

v\_pret Arta.pret%type;

begin

if :new.pret < 250 then

raise\_application\_error(-20001, 'Ati introdus o data incorecta');

end if;

end;

/

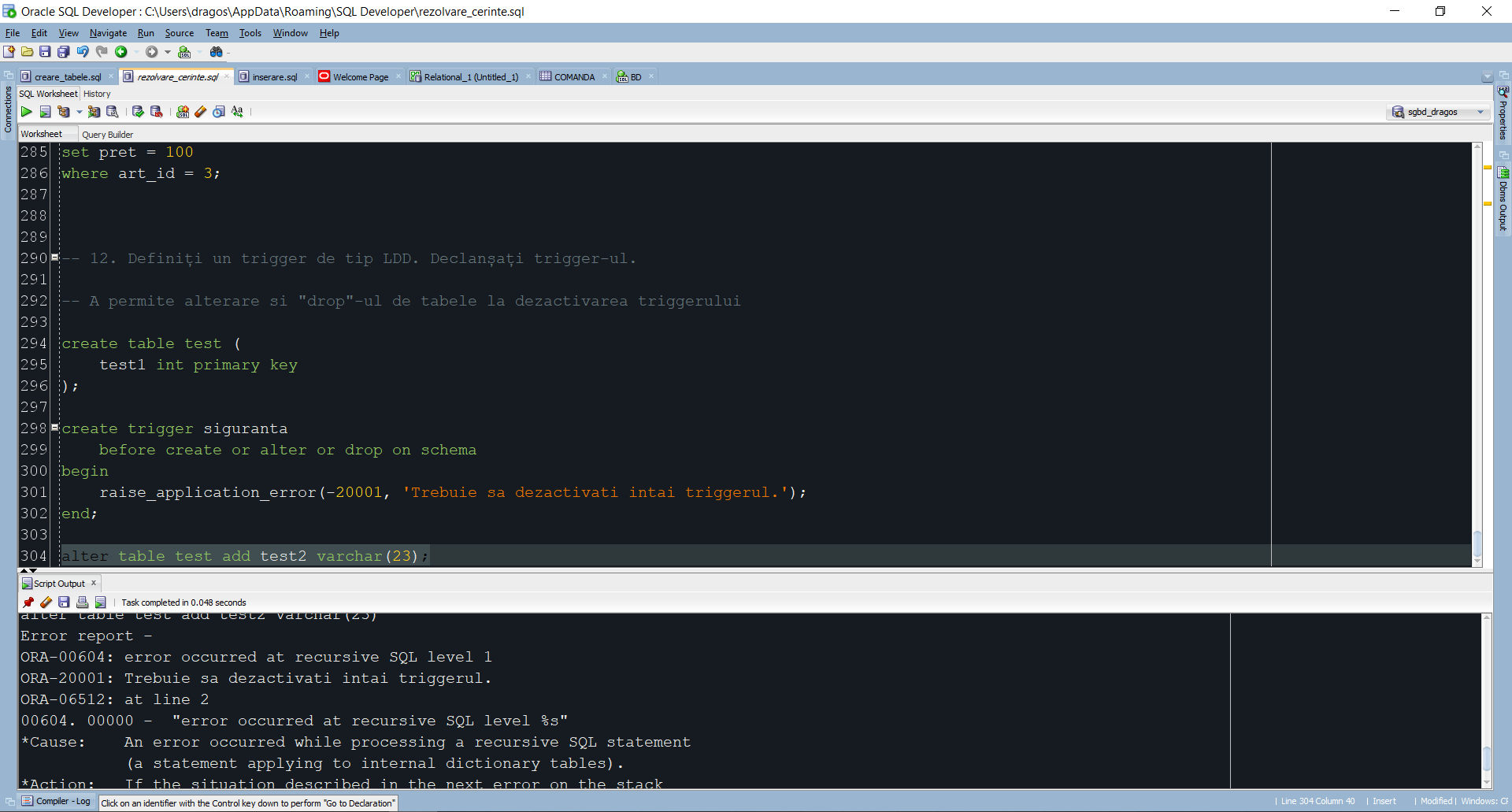
INSERT INTO Arta VALUES (16, 2022, 'Test', 100, 'test', 1, 1);

update Arta

set pret = 100

where art\_id = 3;

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



-- A permite alterare si "drop"-ul de tabele la dezactivarea triggerului

create table test (

test1 int primary key

);

create trigger siguranta

before create or alter or drop on schema

begin

raise\_application\_error(-20001, 'Trebuie sa dezactivati intai triggerul.');

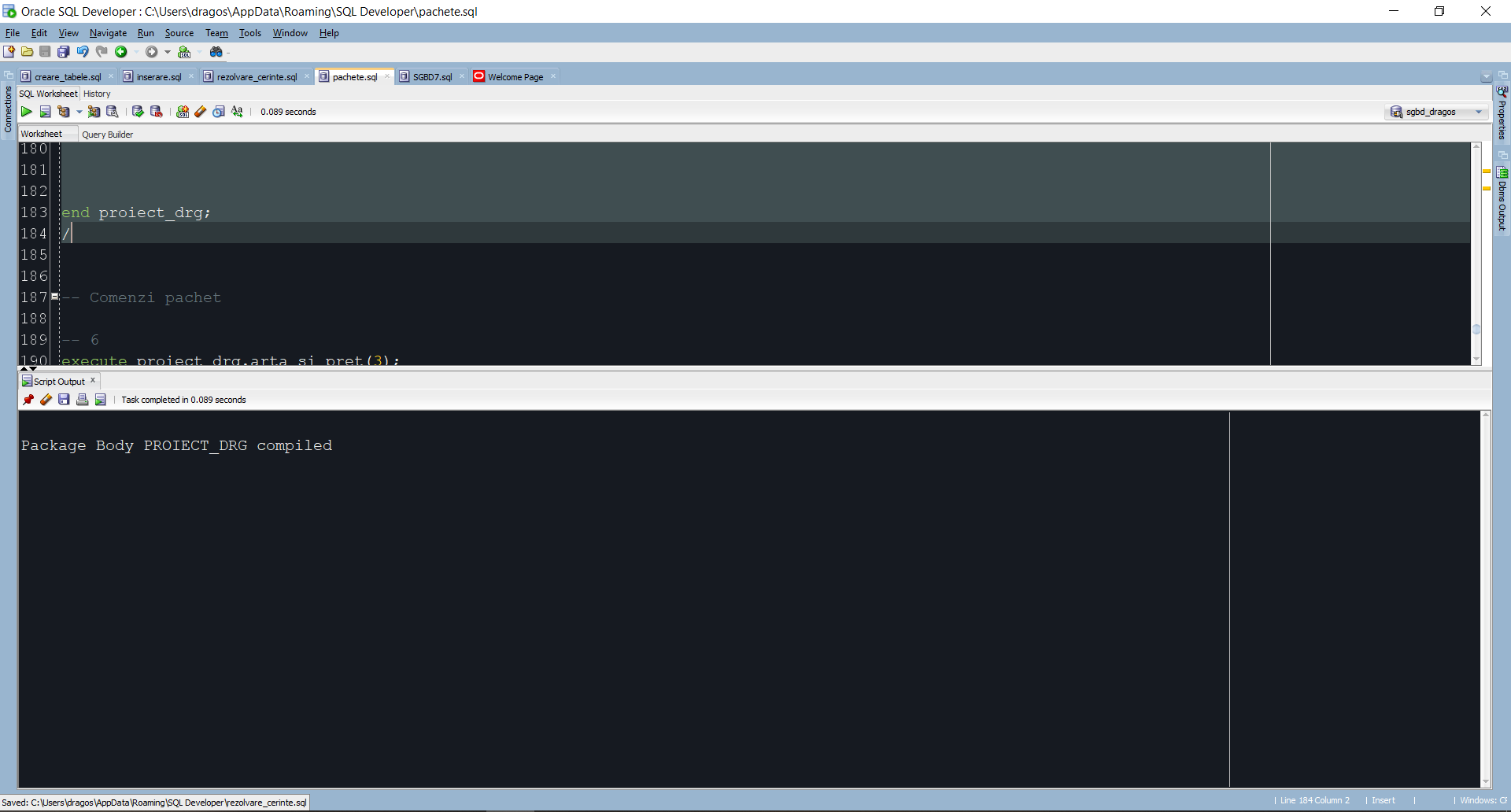
end;

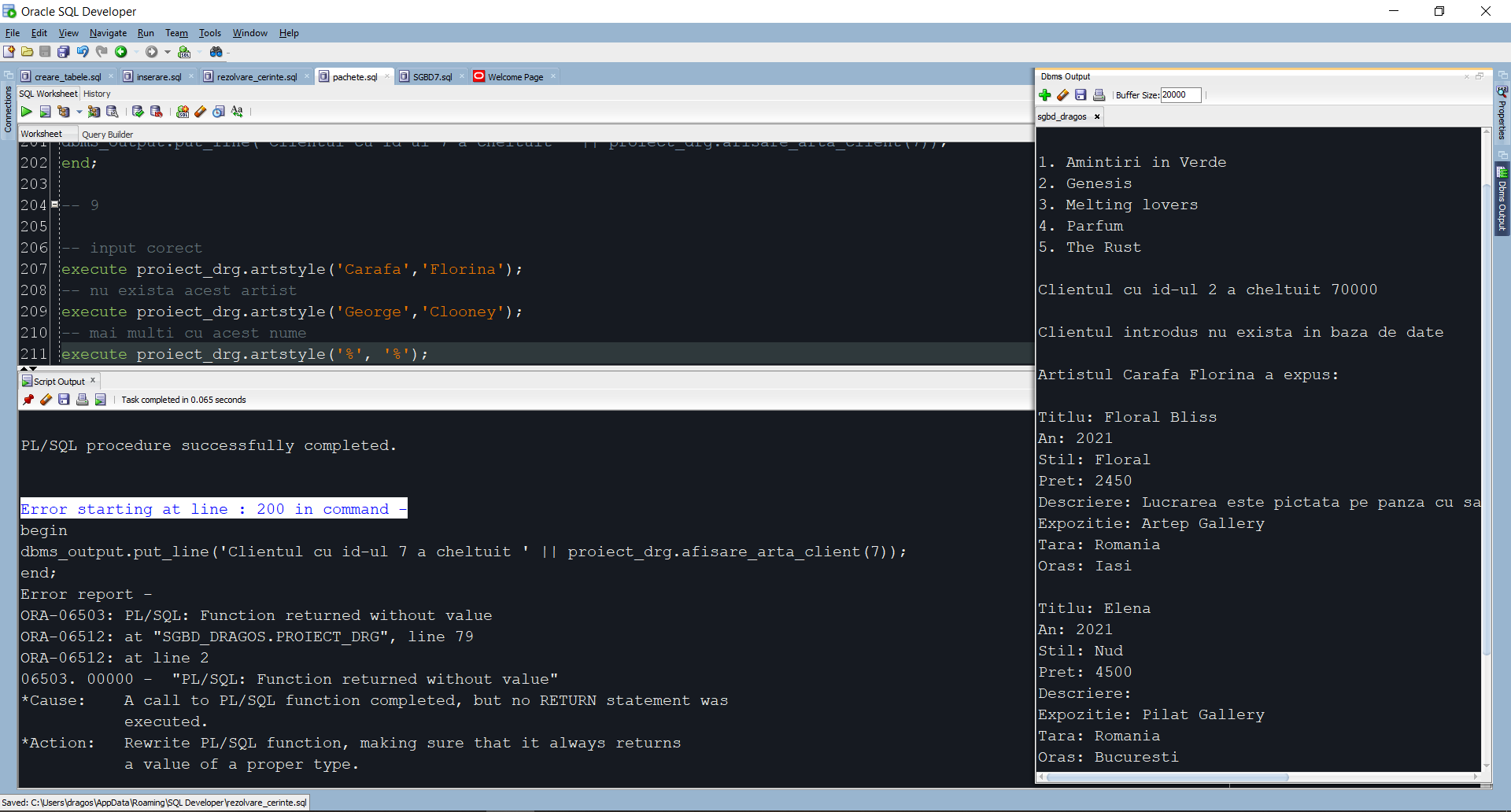
alter table test add test2 varchar(23);

drop trigger siguranta;

drop table test;

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





create or replace package proiect\_drg is

procedure arta\_si\_pret (v\_id\_artist Artist.artist\_id%type);

procedure pret\_dep\_arta;

function afisare\_arta\_client (v\_client\_id Client.client\_id%type) return number;

procedure artstyle (v\_nume Artist.nume%type, v\_prenume Artist.prenume%type);

end proiect\_drg;

/

create or replace package body proiect\_drg is

--6. Formula?i în limbaj natural o problem? pe care s? o rezolva?i folosind un subprogram stocat care s?

-- utilizeze dou? tipuri de colec?ie studiate. Apela?i subprogramul.

-- Pentru un id de artist introdus la tastatura, sa se afiseze ce piese de arta a creat si pretul acestora.

procedure arta\_si\_pret

(v\_id\_artist Artist.artist\_id%type)

as

type tablou\_index is table of Arta%rowtype index by pls\_integer;

t1 tablou\_index;

--v\_art\_id Arta.art\_id%type;

type varr is varray(10) of INTEGER;

t2 varr := varr();

--i number;

begin

--i := 0;

select \*

bulk collect into t1 from Arta

where artist\_id = v\_id\_artist;

t2.extend(t1.count);

for ind in 1..t1.count loop

t2(ind) := t1(ind).pret;

end loop;

for ind in 1..t1.count loop

dbms\_output.put\_line('Titlul ' || t1(ind).titlu || ' are pretul ' || t2(ind) ||'.');

end loop;

end arta\_si\_pret;

-- 7. Formula?i în limbaj natural o problem? pe care s? o rezolva?i folosind un subprogram stocat care s?

--utilizeze un tip de cursor studiat. Apela?i subprogramul.

-- Sa se afiseze piesele de arta a caror pret depaseste 10000

procedure pret\_dep\_arta is

cursor l\_arta is

select \* from Arta

where pret > 10000;

v\_arta l\_arta%rowtype;

begin

open l\_arta;

loop

fetch l\_arta into v\_arta;

exit when l\_arta%notfound;

dbms\_output.put\_line(l\_arta%rowcount || '. ' || v\_arta.titlu);

end loop;

close l\_arta;

end pret\_dep\_arta;

-- 8. Formula?i în limbaj natural o problem? pe care s? o rezolva?i folosind un subprogram stocat de tip

--func?ie care s? utilizeze într-o singur? comand? SQL 3 dintre tabelele definite. Trata?i toate excep?iile

--care pot ap?rea. Apela?i subprogramul astfel încât s? eviden?ia?i toate cazurile tratate.

-- Pentru id-ul unui client dat sa se afiseze cat a cheltuit in total pe piesele pe care

-- le-a comandat

function afisare\_arta\_client

(v\_client\_id Client.client\_id%type)

return number is

rezultat number;

clientel number;

begin

select client\_id into clientel

from Client

where client\_id = v\_client\_id;

select sum (a.pret) into rezultat

from Arta a

join Comanda cc on (a.art\_id = cc.art\_id)

join Client c on (cc.client\_id = c.client\_id)

where c.client\_id = v\_client\_id;

return rezultat;

exception

when no\_data\_found then

dbms\_output.put\_line('Clientul introdus nu exista in baza de date');

when others then

dbms\_output.put\_line('Alta eroare');

end afisare\_arta\_client;

-- 9.Formula?i în limbaj natural o problem? pe care s? o rezolva?i folosind un subprogram stocat de tip

--procedur? care s? utilizeze într-o singur? comand? SQL 5 dintre tabelele definite. Trata?i toate

--excep?iile care pot ap?rea, incluzând excep?iile NO\_DATA\_FOUND ?i TOO\_MANY\_ROWS. Apela?i

--subprogramul astfel încât s? eviden?ia?i toate cazurile tratate.

-- Pentru un artist dat sa se afiseze detaliile pentru picturile expuse si a expozitiilor din care fac parte.

-- (an, titlu, pret, descriere, stil, nume expozitie, tara, oras)

procedure artstyle

(v\_nume Artist.nume%type, v\_prenume Artist.prenume%type)

is

v\_artist\_id Artist.artist\_id%type;

type t\_detalii is record(

artist\_id Artist.artist\_id%type,

art\_id Arta.art\_id%type,

titlul Arta.titlu%type,

an Arta.an%type,

pret Arta.pret%type,

descriere Arta.descriere%type,

nume\_stil Stil.nume\_stil%type,

nume\_expozitie Expozitie.nume%type,

tara Expozitie.tara%type,

oras Expozitie.oras%type

--art\_id Arta\_expusa.art\_id%type

);

detalii t\_detalii;

cursor c\_detalii return t\_detalii is

select ar.artist\_id, a.art\_id, a.titlu, a.an, a.pret, a.descriere, s.nume\_stil, e.nume, e.tara, e.oras

from Artist ar join Arta a on (ar.artist\_id = a.artist\_id)

join Stil s on (a.stil\_id = s.stil\_id)

join Arta\_expusa ae on (a.art\_id = ae.art\_id)

join Expozitie e on (e.expozitie\_id = ae.expozitie\_id)

where ar.artist\_id = v\_artist\_id;

begin

select artist\_id into v\_artist\_id

from Artist

where lower(nume) like lower(v\_nume)

and lower(prenume) like lower(v\_prenume);

dbms\_output.put\_line('Artistul ' || v\_nume ||' ' || v\_prenume || ' a expus: ');

dbms\_output.put\_line('');

open c\_detalii;

loop

fetch c\_detalii into detalii;

exit when c\_detalii%notfound;

if detalii.art\_id is not null then

dbms\_output.put\_line('Titlu: ' || detalii.titlul);

dbms\_output.put\_line('An: ' || detalii.an);

dbms\_output.put\_line('Stil: ' || detalii.nume\_stil);

dbms\_output.put\_line('Pret: ' || detalii.pret);

dbms\_output.put\_line('Descriere: ' || detalii.descriere);

dbms\_output.put\_line('Expozitie: ' || detalii.nume\_expozitie);

dbms\_output.put\_line('Tara: ' || detalii.tara);

dbms\_output.put\_line('Oras: ' || detalii.oras);

dbms\_output.put\_line('');

end if;

end loop;

close c\_detalii;

exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista artist cu numele asta.');

when too\_many\_rows then

dbms\_output.put\_line('Exista mai multi artisti cu numele asta.');

when others then

dbms\_output.put\_line('Alte erori');

end artstyle;

end proiect\_drg;

/

-- Comenzi pachet

-- 6

execute proiect\_drg.arta\_si\_pret(3);

-- 7

execute proiect\_drg.pret\_dep\_arta;

-- 8

begin

dbms\_output.put\_line('Clientul cu id-ul 2 a cheltuit ' || proiect\_drg.afisare\_arta\_client(2));

end;

begin

dbms\_output.put\_line('Clientul cu id-ul 7 a cheltuit ' || proiect\_drg.afisare\_arta\_client(7));

end;

-- 9

-- input corect

execute proiect\_drg.artstyle('Carafa','Florina');

-- nu exista acest artist

execute proiect\_drg.artstyle('George','Clooney');

-- mai multi cu acest nume

execute proiect\_drg.artstyle('%', '%');