Gestiunea unui site destinat vanzarii de tablouri

Aldea Alexia, grupa 244

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

Baza de date contine informatii cu privire la tablourile realizate de diversi artisti contemporani, care sunt incredintate unui admin (al site-ului) si, mai apoi, sunt oferite spre vanzare celor pasionati de arta. Clientii pot lasa un rating (review) fiecarui tablou cumparat. De asemenea, acestia pot lua parte si la evenimentele (targurile) organizate de admin-ul site-ului, prin cumpararea unui bilet. Aceste evenimente sunt sustinute de diferiti sponsori.

Scopul crearii acestei baze de date este de a putea tine evidenta comenzilor plasate pe site si evidenta tablourilor oferite de artisti spre vanzare. Astfel, micii artisti contemporani isi pot arata picturile lumii.

Un tablou este vandut admin-ului de catre artist, care, la randul sau, il pune la vanzare pe site, pentru a-l vedea si altii. Acesta poate fi comandat de catre clienti sau vandut direct la targuri/evenimente organizate de admin. De asemenea, un tablou poate avea unul sau mai multe rating-uri primite de la clienti.

Clientii pot plasa comenzi pe site, pot scrie review-uri tablourilor cumparate (un review per tablou) si isi pot cumpara bilet (un client => un bilet) pentru a lua parte la targ. Targurile sunt organizate periodic de catre admin.

1. Realizați diagrama entitate-relație (ERD).

Diagram

Description automatically generated

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

Diagram

Description automatically generated

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

---------------------------------ADMIN---------------------------

CREATE TABLE ADMIN( id\_admin NUMBER(5) CONSTRAINT pk\_admin PRIMARY KEY,

denumire\_site VARCHAR(100) CONSTRAINT denumire\_site\_admin NOT NULL,

nr\_utilizatori NUMBER(5) CONSTRAINT nr\_utilizatori\_admin NOT NULL,

nr\_tablouri NUMBER(5) CONSTRAINT nr\_tablouri\_admin NOT NULL,

rating NUMBER(5) CONSTRAINT rating\_admin NOT NULL);

Graphical user interface, text, application, email

Description automatically generated

-------------------------------ARTIST------------------------------

CREATE TABLE ARTIST(id\_artist NUMBER(5) CONSTRAINT pk\_artist PRIMARY KEY,

nr\_tablouri NUMBER(5) CONSTRAINT nr\_tablouri\_artist NOT NULL,

varsta NUMBER(5) CONSTRAINT varsta\_artist NOT NULL,

nume VARCHAR(100) CONSTRAINT nume\_artist NOT NULL);

Graphical user interface, text, application, email

Description automatically generated

-----------------------------CONTRACT------------------------

CREATE TABLE CONTRACT(id\_contract NUMBER(5) CONSTRAINT pk\_contract PRIMARY KEY,

data\_semnare DATE CONSTRAINT data\_semnare\_contract NOT NULL,

durata VARCHAR(100) CONSTRAINT durata\_contract NOT NULL,

id\_admin NUMBER(5), CONSTRAINT fk\_contr FOREIGN KEY(id\_admin) REFERENCES ADMIN(id\_admin),

id\_artist NUMBER(5), CONSTRAINT fk\_cont FOREIGN KEY(id\_artist) REFERENCES ARTIST(id\_artist)

);

Graphical user interface, text, application, Word

Description automatically generated

---------------------------------CLIENT---------------------------

CREATE TABLE CLIENT(id\_client NUMBER(5) CONSTRAINT pk\_client primary key,

nume VARCHAR(100) CONSTRAINT nume\_client NOT NULL,

telefon VARCHAR(100) CONSTRAINT telefon\_client NOT NULL,

email VARCHAR(100) CONSTRAINT email\_client NOT NULL);

Graphical user interface, text, application, email

Description automatically generated

----------------------------SPONSOR-----------------------------

CREATE TABLE SPONSOR(id\_sponsor NUMBER(5) CONSTRAINT pk\_sponsor primary key,

nume VARCHAR(100) CONSTRAINT nume\_sponsor not null,

suma\_sponsorizare NUMBER(5) CONSTRAINT suma\_spons\_sponsor not null);

Graphical user interface, text, application, email

Description automatically generated

-------------------------TARG/EXPOZITIE--------------------------

CREATE TABLE TARG(id\_targ NUMBER(5) CONSTRAINT pk\_targ primary key,

data\_eveniment DATE CONSTRAINT data\_ev\_targ not null,

nr\_locuri NUMBER(5) CONSTRAINT nr\_loc\_targ not null,

locatie VARCHAR(100) CONSTRAINT locatie\_targ not null,

id\_admin NUMBER(5), CONSTRAINT fk\_tr FOREIGN KEY (id\_admin) REFERENCES ADMIN(id\_admin),

id\_sponsor NUMBER(5), CONSTRAINT fk\_tg FOREIGN KEY (id\_sponsor) REFERENCES SPONSOR(id\_sponsor) );

Graphical user interface, text, application, email

Description automatically generated

----------------------TABLOU------------------------

CREATE TABLE TABLOU(id\_tablou NUMBER(5) CONSTRAINT pk\_tablou primary key,

denumire VARCHAR(100) CONSTRAINT denumire\_tablou not null,

an\_creare NUMBER(5) CONSTRAINT an\_cr\_tablou not null,

tip\_tablou VARCHAR(100) CONSTRAINT tip\_tabl\_tablou not null,

tip\_materiale VARCHAR(100) CONSTRAINT tip\_mat\_tablou not null,

pret NUMBER(5) CONSTRAINT pret\_tablou not null,

id\_admin NUMBER(5), CONSTRAINT fk\_ta FOREIGN KEY (id\_admin) REFERENCES ADMIN(id\_admin),

id\_artist NUMBER(5), CONSTRAINT fk\_tb FOREIGN KEY (id\_artist) REFERENCES ARTIST(id\_artist),

id\_targ NUMBER(5), CONSTRAINT fk\_tl FOREIGN KEY (id\_targ) REFERENCES TARG(id\_targ) );

Graphical user interface, text, application, Word

Description automatically generated

------------------------------RATING----------------------------

CREATE TABLE RATING(id\_rating NUMBER(5) CONSTRAINT pk\_rating primary key,

data\_scrierii DATE CONSTRAINT data\_scr\_rating not null,

nota\_acordata NUMBER(5) CONSTRAINT nota\_ac\_rating not null,

detalii VARCHAR(100) CONSTRAINT detalii\_rating not null,

id\_client NUMBER(5), CONSTRAINT fk\_ra FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client),

id\_tablou NUMBER(5), CONSTRAINT fk\_rt FOREIGN KEY (id\_tablou) REFERENCES TABLOU(id\_tablou) );

Graphical user interface, text, application, email

Description automatically generated

--------------------------------COMANDA------------------------------------

CREATE TABLE COMANDA(id\_comanda NUMBER(5) CONSTRAINT pk\_comanda primary key,

data\_plasare DATE CONSTRAINT data\_plas\_comanda not null,

data\_sosire DATE CONSTRAINT data\_sos\_comanda not null,

pret NUMBER(5) CONSTRAINT pret\_comanda not null,

greutate NUMBER(5) CONSTRAINT greutate\_comanda not null,

id\_client NUMBER(5), CONSTRAINT fk\_com FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client),

id\_tablou NUMBER(5), CONSTRAINT fk\_co FOREIGN KEY (id\_tablou) REFERENCES TABLOU(id\_tablou) );

Graphical user interface, text, application, Word

Description automatically generated

-----------------------------BILET-----------------------

CREATE TABLE BILET (id\_bilet NUMBER(5) CONSTRAINT pk\_bilet primary key,

pret NUMBER(5) CONSTRAINT pret\_bilet not null,

data\_achizitionare DATE CONSTRAINT data\_achiz\_bilet not null,

id\_client NUMBER(5), CONSTRAINT fk\_bil FOREIGN KEY (id\_client) REFERENCES CLIENT(id\_client),

id\_targ NUMBER(5), CONSTRAINT fk\_blt FOREIGN KEY (id\_targ) REFERENCES TARG(id\_targ) );

Graphical user interface, text, application, Word, email

Description automatically generated

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

insert into admin values(1, 'ProfiArt', 5000, 1243, 8);

insert into admin values(2,'ArtaInPapuci', 234, 200, 6);

insert into admin values(3, 'SHL', 2323, 6463, 10);

insert into admin values(4, 'PicteazaCuNoi', 543, 234,6);

insert into admin values(5, 'FloriVesele', 5426, 23556,8);

A picture containing graphical user interface

Description automatically generated

insert into artist values(6, 300,45,'Popescu Valentin');

insert into artist values(7, 23, 19, 'Mirciulescu Cristina');

insert into artist values(8, 345, 65, 'Floarea Andrei');

insert into artist values(9, 543, 63, 'Dumitrescu Florin');

insert into artist values(10, 36, 25, 'Popa Alexandru');

Graphical user interface, application

Description automatically generated

insert into contract values(11, TO\_DATE('26-01-2020','dd-mm-yyyy'), '3 ani',1,7);

insert into contract values(12, TO\_DATE('14-01-2020','dd-mm-yyyy'), '5 ani', 2, 6);

insert into contract values(13, TO\_DATE('27-09-2019','dd-mm-yyyy'), '6 ani', 3, 10);

insert into contract values(14, TO\_DATE('12-06-2022', 'dd-mm-yyyy'), '6 luni', 4, 8);

insert into contract values(15, TO\_DATE('15-08-2022', 'dd-mm-yyyy'), '1 an', 5, 9);

insert into contract values(16, TO\_DATE('24-06-2022', 'dd-mm-yyyy'), '10 ani', 3, 7);

insert into contract values(17, TO\_DATE('17-07-2021', 'dd-mm-yyyy'), '4 ani', 1, 6);

insert into contract values(18, TO\_DATE('15-02-2021', 'dd-mm-yyyy'), '2 ani', 5, 8);

insert into contract values(19, TO\_DATE('10-12-2020', 'dd-mm-yyyy'), '2 ani', 4,10);

insert into contract values(20, TO\_DATE('11-10-2022', 'dd-mm-yyyy'), '3 luni', 3, 9);

Graphical user interface, text

Description automatically generated with medium confidence

insert into client values(21, 'Dinculescu Cosmina', '0756234543', 'cosmina123@gmail.com');

insert into client values(22, 'Georgescu Diana', '0745387163', 'didi@gmail.com');

insert into client values(23, 'Anghelescu Radu', '0788576209', 'rodede@gmail.com');

insert into client values(24, 'Pentu Maria', '0746012463', 'mery34@gmai.com');

insert into client values(25, 'Florescu Marina', '0789076713', 'flory\_marinush@gmail.com');

Graphical user interface, text, application

Description automatically generated

insert into sponsor values(26, 'TotalSoft', 34000);

insert into sponsor values(27, 'RedBull', 24000);

insert into sponsor values(28, 'HatzSRL', 4500);

insert into sponsor values(29, 'MaryandAlex', 7000);

insert into sponsor values(30, 'Ikea', 20000);

Graphical user interface, text, application

Description automatically generated

insert into targ values(31, TO\_DATE('01-11-2022', 'dd-mm-yyyy'),400, 'Str Kogalniceanu 9', 1, 27);

insert into targ values(32, TO\_DATE('1-12-2022', 'dd-mm-yyyy'), 2000,'Sala Palatului', 4, 30);

insert into targ values(33, TO\_DATE('02-12-2022', 'dd-mm-yyyy'), 30000, 'Palatul Parlamentului', 2, 26);

insert into targ values(34, TO\_DATE('15-01-2023', 'dd-mm-yyyy'), 2400, 'Sala Palatului', 3, 28);

insert into targ values(35, TO\_DATE('01-02-2023', 'dd-mm-yyyy'), 450, 'Str Kogalniceanu 9', 5, 29);

Graphical user interface, text

Description automatically generated with medium confidence

insert into tablou values(36, 'Iarna', 2021, 'pictura', 'acrilice', 340, 1, 8 ,34);

insert into tablou values(37, 'Iarna la schi', 2022, 'desen', 'creion', 200, 4, 9, 31);

insert into tablou values(38, 'Flori de camp', 2020, 'pictura', 'ulei', 500, 2, 6, 35);

insert into tablou values(39, 'Nuante de gri', 2021, 'desen', 'creion', 250, 3,7, 32);

insert into tablou values(40, 'Foc de tabara', 2020, 'pictura', 'acrilice', 700, 5,10, 33);

Graphical user interface, text

Description automatically generated with medium confidence

insert into rating values(41, TO\_DATE('01-02-2022','dd-mm-yyyy'), 7, 'foarte frumos', 25, 36);

insert into rating values(42, TO\_DATE('01-10-2022','dd-mm-yyyy'), 10, 'impresionant', 22, 37);

insert into rating values(43, TO\_DATE('13-04-2021','dd-mm-yyyy'), 8, 'o adevarata capodopera', 24, 38);

insert into rating values(44, TO\_DATE('23-05-2022','dd-mm-yyyy'), 5, 'satisfacator', 21, 39);

insert into rating values(45, TO\_DATE('01-12-2021','dd-mm-yyyy'), 10, 'sunt mare fan',23,40);

Graphical user interface

Description automatically generated with low confidence

insert into comanda values(46, TO\_DATE('24-08-2022','dd-mm-yyyy'), TO\_DATE('01-12-2022', 'dd-mm-yyyy'), 400, 5, 21, 37);

insert into comanda values(47, TO\_DATE('25-10-2022','dd-mm-yyyy'), TO\_DATE('05-12-2022','dd-mm-yyyy'), 625, 3, 24, 36);

insert into comanda values(48, TO\_DATE('13-10-2022','dd-mm-yyyy'), TO\_DATE('23-11-2022', 'dd-mm-yyyy'), 1000, 7, 22, 40);

insert into comanda values(49, TO\_DATE('25-09-2022','dd-mm-yyyy'), TO\_DATE('31-10-2022','dd-mm-yyyy'), 450, 2, 23,38);

insert into comanda values(50, TO\_DATE('17-07-2022','dd-mm-yyyy'), TO\_DATE('31-10-2022','dd-mm-yyyy'), 500, 3, 25, 39);

A picture containing text

Description automatically generated

insert into bilet values(51, 50, TO\_DATE('31-07-2022','dd-mm-yyyy'), 22, 31);

insert into bilet values(52, 100, TO\_DATE('01-10-2022','dd-mm-yyyy'), 21, 35);

insert into bilet values(53, 50, TO\_DATE('23-06-2022', 'dd-mm-yyyy'), 24,33);

insert into bilet values(54, 150, TO\_DATE('12-10-2022','dd-mm-yyyy'), 22, 34);

insert into bilet values(55, 65, TO\_DATE('31-05-2022','dd-mm-yyyy'), 23, 32);

insert into bilet values(56, 50, TO\_DATE('23-06-2022','dd-mm-yyyy'), 25, 34);

insert into bilet values(57, 100, TO\_DATE('14-07-2022','dd-mm-yyyy'), 23, 31);

insert into bilet values(58, 55, TO\_DATE('15-09-2022', 'dd-mm-yyyy'), 21, 35);

insert into bilet values(59, 150, TO\_DATE('25-09-2022','dd-mm-yyyy'), 25, 33);

insert into bilet values(60, 50, TO\_DATE('03-03-2022','dd-mm-yyyy'), 24, 31);

Graphical user interface

Description automatically generated

1. Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.

-- Pentru fiecare tablou sa se afiseze data targului la care va fi expus si pretul mediu al tablourilor pt acel targ

CREATE OR REPLACE PROCEDURE proceduraex6 IS

-- vom crea un tip de date de inregistrare pt a pastra data targului si locatia

TYPE record1 IS RECORD

(data\_ev TARG.data\_eveniment%TYPE,

locatie TARG.locatie%TYPE,

pret\_mediu TABLOU.pret%TYPE);

--toate datele despre targuri vor fi retinute intr-un tablou imbricat

TYPE tabel\_imbricat IS TABLE OF record1;

--vom folosi un vector pentru a retine datele pentru fiecare targ

TYPE vector IS VARRAY(8) OF TARG.data\_eveniment%TYPE;

--cu ajutorul unui cursor vom parcurge toate targurile

CURSOR c IS

SELECT id\_targ, data\_eveniment, locatie

FROM TARG;

t tabel\_imbricat := tabel\_imbricat();

r record1;

v vector := vector();

BEGIN

FOR i in c LOOP

--vom salva datele pentru pretul mediu al tablourilor pt targul respectiv

SELECT AVG(pret)

INTO r.pret\_mediu

FROM TABLOU

WHERE id\_targ = i.id\_targ;

v.EXTEND;

SELECT data\_eveniment

INTO v(v.LAST)

FROM TARG

WHERE id\_targ=i.id\_targ;

r.data\_ev := i.data\_eveniment;

r.locatie := i.locatie;

--adaugsm variabila in tabloul imbricat

t.EXTEND;

t(t.LAST) := r;

DBMS\_OUTPUT.PUT\_LINE('La targul din data ' || t(t.LAST).data\_ev || ' din locatia ' || t(t.LAST).locatie || ' pretul mediu al tablourilor este ' || t(t.LAST).pret\_mediu);

END LOOP;

END proceduraex6;

/

EXECUTE proceduraex6;

/

Graphical user interface, text, application

Description automatically generated

1. 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. Apelați subprogramul.

-- Sa se mareasca pretul tablourilor care au rating >=8 si artistii lor au varsta >=50.

CREATE OR REPLACE PROCEDURE proceduraex7 IS

CURSOR tb IS

SELECT id\_tablou

FROM TABLOU;

CURSOR p (id\_tablou TABLOU.id\_tablou%TYPE) IS

SELECT \*

FROM ARTIST a

RIGHT JOIN TABLOU t ON t.id\_artist= a.id\_artist

LEFT JOIN RATING ra ON t.id\_tablou=ra.id\_tablou

WHERE t.id\_tablou = 36 and ra.nota\_acordata>=4 and a.varsta>=50

FOR UPDATE OF t.pret NOWAIT;

BEGIN

FOR r in tb LOOP

FOR i in p(r.id\_tablou) LOOP

UPDATE TABLOU

SET pret = pret + ((pret \* 25)/100)

WHERE CURRENT OF p;

END LOOP;

END LOOP;

END proceduraex7;

/

EXECUTE proceduraex7;

SELECT \* FROM TABLOU;

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

1. Formulați în limbaj natural o problemă pe care să orezolvați folosind un subprogram stocatindependent de tip funcție care să utilizeze într-o singură comandă SQL3 dintre tabelele definite. Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

-- Se citeste numele unui client de la tastatura. Sa se returneze pretul total al comenzii facute de acesta(pretul comenzii + pretul tabloului comandat). In cazul in care un client

--are mai multe comenzi, se va returna suma tuturor preturilor totale. Se vor arunca erori

--in cazul in care: exista mai multi clienti cu numele dat, nu exista un client cu numele dat --sau nu exista comenzi facute de acest client.

--ne vom folosi de tabelele comanda, tablou si client in aceasta functie

CREATE OR REPLACE FUNCTION functieex8(v\_nume CLIENT.nume%TYPE)

RETURN COMANDA.pret%TYPE IS

pret\_total COMANDA.pret%TYPE;

contor NUMBER;

BEGIN

--tratam intai exceptiile

--numaram cati clienti exista cu numele dat in input

SELECT COUNT (\*) INTO contor

FROM CLIENT WHERE INITCAP(v\_nume) = INITCAP(nume);

--tratam cazul in care nu exista niciun client si cazul in care esita mai multi clienti cu acelasi nume

IF contor = 0

THEN RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista un client cu acest nume');

ELSIF contor>1

THEN RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi clienti cu acest nume');

END IF;

--folosim aceeasi variabila pentru a numara cate comenzi exista pentru clientul dat, iar --daca nu exsita niciuna, vom arunca o eroare

SELECT COUNT(\*) INTO contor

FROM CLIENT RIGHT OUTER JOIN COMANDA USING (id\_client)

WHERE INITCAP(v\_nume) = INITCAP(nume);

IF contor = 0

THEN RAISE\_APPLICATION\_ERROR(-20002, 'Nu exista comenzi pe acest nume');

END IF;

--functia de SUM va trata atat cazul in care exista o singura comanda, cat si cazul in care –exista mai multe

SELECT SUM(c.pret+ t.pret)

INTO pret\_total

FROM COMANDA c JOIN TABLOU t USING(id\_tablou) JOIN CLIENT USING (id\_client)

WHERE INITCAP(v\_nume) = INITCAP(CLIENT.nume);

RETURN pret\_total;

END functieex8;

/

select\* from client;

SELECT functieex8('Dinculescu Cosmina') FROM DUAL;

SELECT functieex8('Popescu Carmen') FROM DUAL;

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. 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 da numele unui artist ca parametru. Sa se afiseze numele tabloului oferit spre cumparare, ratingul acestuia, idul contractului si numele site-ului unde isi vinde tabloul.

CREATE OR REPLACE PROCEDURE proceduraex9 (nume\_artist ARTIST.nume%TYPE) IS

--vom declara variabilele in care vom salva datele care trebuiesc selectate si afisate

nume\_tablou TABLOU.denumire%TYPE;

rating\_tablou RATING.nota\_acordata%TYPE;

id\_contr CONTRACT.id\_contract%TYPE;

nume\_site ADMIN.denumire\_site%TYPE;

BEGIN

--vom folosi join pentru a accesa si selecta ce ne intereseaza

SELECT t.denumire, r.nota\_acordata, c.id\_contract, a.denumire\_site

INTO nume\_tablou, rating\_tablou, id\_contr, nume\_site

FROM ARTIST ar RIGHT JOIN TABLOU t ON (ar.id\_artist = t.id\_artist)

JOIN RATING r ON (t.id\_tablou = r.id\_tablou) JOIN CONTRACT c ON (ar.id\_artist = c.id\_artist)

JOIN ADMIN a ON (c.id\_admin= a.id\_admin)

WHERE INITCAP(nume\_artist) = INITCAP(ar.nume)

GROUP BY t.denumire, r.nota\_acordata, c.id\_contract, a.denumire\_site;

--afisam in consola datele selectate mai sus

DBMS\_OUTPUT.PUT\_LINE('Numele artistului este: ' || INITCAP(nume\_artist));

DBMS\_OUTPUT.PUT\_LINE('Denumirea tabloului oferit spre vanzare este: ' || nume\_tablou);

DBMS\_OUTPUT.PUT\_LINE('Ratingul acestui tablou este: ' || rating\_tablou);

DBMS\_OUTPUT.PUT\_LINE('Numarul contractului este: ' || id\_contr);

DBMS\_OUTPUT.PUT\_LINE('Denumirea site-ului unde este publicat tabloul este: ' || nume\_site);

EXCEPTION

WHEN TOO\_MANY\_ROWS

THEN RAISE\_APPLICATION\_ERROR(-20003, 'Exista mai multi artisti cu acest nume');

WHEN NO\_DATA\_FOUND

THEN RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista artisti cu acest nume');

END proceduraex9;

/

select\* from artist;

EXECUTE proceduraex9('Popescu Valentin');

EXECUTE proceduraex9('Aldea Alexia');

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

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

-- Trigger care interzice stergerea datelor din tabelul targ

/

CREATE OR REPLACE TRIGGER triggerex10

BEFORE DELETE ON TARG

BEGIN

RAISE\_APPLICATION\_ERROR(-20005, 'Nu se pot sterge date din tabelul TARG');

END;

/

DELETE FROM TARG

WHERE id\_targ= 31;

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

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

-- Trigger care interzice cresterea pretului unui bilet cu mai mult de 100

/

CREATE OR REPLACE TRIGGER triggerex11

BEFORE UPDATE ON BILET

FOR EACH ROW

BEGIN

IF(:NEW.pret - :OLD.pret >100)

THEN RAISE\_APPLICATION\_ERROR(-20006, 'Pretul nu poate creste cu mai mult de 100');

END IF;

END;

/

UPDATE BILET

SET pret = pret+ 101

WHERE id\_bilet = 57;

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

1. Definiți un triggerde tip LDD. Declanșați trigger-ul.

--Triger care afiseaza un mesaj de fiecare data cand este rulata o comanda ldd

/

CREATE OR REPLACE TRIGGER triggerex12

AFTER CREATE OR ALTER OR DROP ON SCHEMA

BEGIN

DBMS\_OUTPUT.PUT\_LINE('A fost efectuata o comanda ldd');

END;

/

ALTER TABLE CLIENT

ADD varsta NUMBER(5);

ALTER TABLE CLIENT

DROP COLUMN varsta;

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

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

/

CREATE OR REPLACE PACKAGE pachetex13 AS

FUNCTION functieex8 (v\_nume CLIENT.nume%TYPE) RETURN COMANDA.pret%TYPE;

PROCEDURE proceduraex6;

PROCEDURE proceduraex7;

PROCEDURE proceduraex9(nume\_artist ARTIST.nume%TYPE);

END pachetex13;

/

CREATE OR REPLACE PACKAGE BODY pachetex13 AS

--pretul total pentru comenzile facute de un client

FUNCTION functieex8(v\_nume CLIENT.nume%TYPE)

RETURN COMANDA.pret%TYPE IS

pret\_total COMANDA.pret%TYPE;

contor NUMBER;

BEGIN

SELECT COUNT(\*) INTO contor

FROM CLIENT WHERE INITCAP(v\_nume) = INITCAP(nume);

IF contor = 0

THEN RAISE\_APPLICATION\_ERROR(-20000, 'Nu exista un client cu acest nume');

ELSIF contor > 1

THEN RAISE\_APPLICATION\_ERROR(-20001, 'Exista mai multi clienti cu acest nume');

END IF;

SELECT COUNT(\*) INTO contor

FROM CLIENT RIGHT OUTER JOIN COMANDA USING (id\_client)

WHERE INITCAP(v\_nume) = INITCAP(nume);

IF contor = 0

THEN RAISE\_APPLICATION\_ERROR(-20002, 'Nu exista comenzi pe acest nume');

END IF;

SELECT SUM(c.pret+ t.pret)

INTO pret\_total

FROM COMANDA c JOIN TABLOU t USING(id\_tablou) JOIN CLIENT USING (id\_client)

WHERE INITCAP(v\_nume) = INITCAP(CLIENT.nume);

RETURN pret\_total;

END functieex8;

--pretul mediu al unui tablou

PROCEDURE proceduraex6 IS

TYPE record1 IS RECORD

(data\_ev TARG.data\_eveniment%TYPE,

locatie TARG.locatie%TYPE,

pret\_mediu TABLOU.pret%TYPE);

TYPE tabel\_imbricat IS TABLE OF record1;

TYPE vector IS VARRAY(8) OF TARG.data\_eveniment%TYPE;

CURSOR c IS

SELECT id\_targ, data\_eveniment, locatie

FROM TARG;

t tabel\_imbricat := tabel\_imbricat();

r record1;

v vector := vector();

BEGIN

FOR i in c LOOP

SELECT AVG(pret)

INTO r.pret\_mediu

FROM TABLOU

WHERE id\_targ = i.id\_targ;

v.EXTEND;

SELECT data\_eveniment

INTO v(v.LAST)

FROM TARG

WHERE id\_targ=i.id\_targ;

r.data\_ev := i.data\_eveniment;

r.locatie := i.locatie;

t.EXTEND;

t(t.LAST) := r;

DBMS\_OUTPUT.PUT\_LINE('La targul din data ' || t(t.LAST).data\_ev || ' din locatia ' || t(t.LAST).locatie || ' pretul mediu al tablourilor este ' || t(t.LAST).pret\_mediu);

END LOOP;

END proceduraex6;

--proceduraex7

PROCEDURE proceduraex7 IS

CURSOR tb IS

SELECT id\_tablou

FROM TABLOU;

CURSOR p (id\_tablou TABLOU.id\_tablou%TYPE) IS

SELECT \*

FROM ARTIST a

RIGHT JOIN TABLOU t ON t.id\_artist= a.id\_artist

LEFT JOIN RATING ra ON t.id\_tablou=ra.id\_tablou

WHERE t.id\_tablou = 36 and ra.nota\_acordata>=4 and a.varsta>=50

FOR UPDATE OF t.pret NOWAIT;

BEGIN

FOR r in tb LOOP

FOR i in p(r.id\_tablou) LOOP

UPDATE TABLOU

SET pret = pret + ((pret \* 25)/100)

WHERE CURRENT OF p;

END LOOP;

END LOOP;

END proceduraex7;

--date despre un artist dat ca parametru

PROCEDURE proceduraex9(nume\_artist ARTIST.nume%TYPE) IS

nume\_tablou TABLOU.denumire%TYPE;

rating\_tablou RATING.nota\_acordata%TYPE;

id\_contr CONTRACT.id\_contract%TYPE;

nume\_site ADMIN.denumire\_site%TYPE;

BEGIN

SELECT t.denumire, r.nota\_acordata, c.id\_contract, a.denumire\_site

INTO nume\_tablou, rating\_tablou, id\_contr, nume\_site

FROM ARTIST ar RIGHT JOIN TABLOU t ON (ar.id\_artist = t.id\_artist)

JOIN RATING r ON (t.id\_tablou = r.id\_tablou) JOIN CONTRACT c ON (ar.id\_artist = c.id\_artist)

JOIN ADMIN a ON (c.id\_admin= a.id\_admin)

WHERE INITCAP(nume\_artist) = INITCAP(ar.nume)

GROUP BY t.denumire, r.nota\_acordata, c.id\_contract, a.denumire\_site;

DBMS\_OUTPUT.PUT\_LINE('Numele artistului este: ' || INITCAP(nume\_artist));

DBMS\_OUTPUT.PUT\_LINE('Denumirea tabloului oferit spre vanzare este: ' || nume\_tablou);

DBMS\_OUTPUT.PUT\_LINE('Ratingul acestui tablou este: ' || rating\_tablou);

DBMS\_OUTPUT.PUT\_LINE('Numarul contractului este: ' || id\_contr);

DBMS\_OUTPUT.PUT\_LINE('Denumirea site-ului unde este publicat tabloul este: ' || nume\_site);

EXCEPTION

WHEN TOO\_MANY\_ROWS

THEN RAISE\_APPLICATION\_ERROR(-20003, 'Exista mai multi artisti cu acest nume');

WHEN NO\_DATA\_FOUND

THEN RAISE\_APPLICATION\_ERROR(-20004, 'Nu exista artisti cu acest nume');

END proceduraex9;

END pachetex13;

/

SELECT pachetex13.functieex8('Dinculescu Cosmina') FROM DUAL;

EXECUTE pachetex13.proceduraex6;

EXECUTE pachetex13.proceduraex7;

EXECUTE pachetex13.proceduraex9('Aldea Alexia');

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated