**Nume: Aciu Mălina-Andreea**

**Grupa: 244**

**Proiect Sisteme de Gestiune a Bazelor de Date**

***Gestionarea taberelor din România și a elevilor participanți***

1. Liceeni din România pot participa la mai multe tabere sponsorizate de companii. Taberele se desfășoară în intervale diverse, au o dată de început, o dată de sfârșit și o locație de desfășurare. Pe parcursul desfășurării taberei, liceeni sunt supravegheați de voluntari.

De asemenea, elevilor le sunt suportate cheltuielile de masa și cazare, opțional și cheltuielile legate de transport.

În cadrul taberei, participanții se pot înscrie la mai multe ateliere, unde se vor aprofunda/dezvolta anumite cunoștințe pe care elevul dorește să pună accent. Atelierele sunt organizate de către voluntari, iar predarea este asigurată de specialiști.

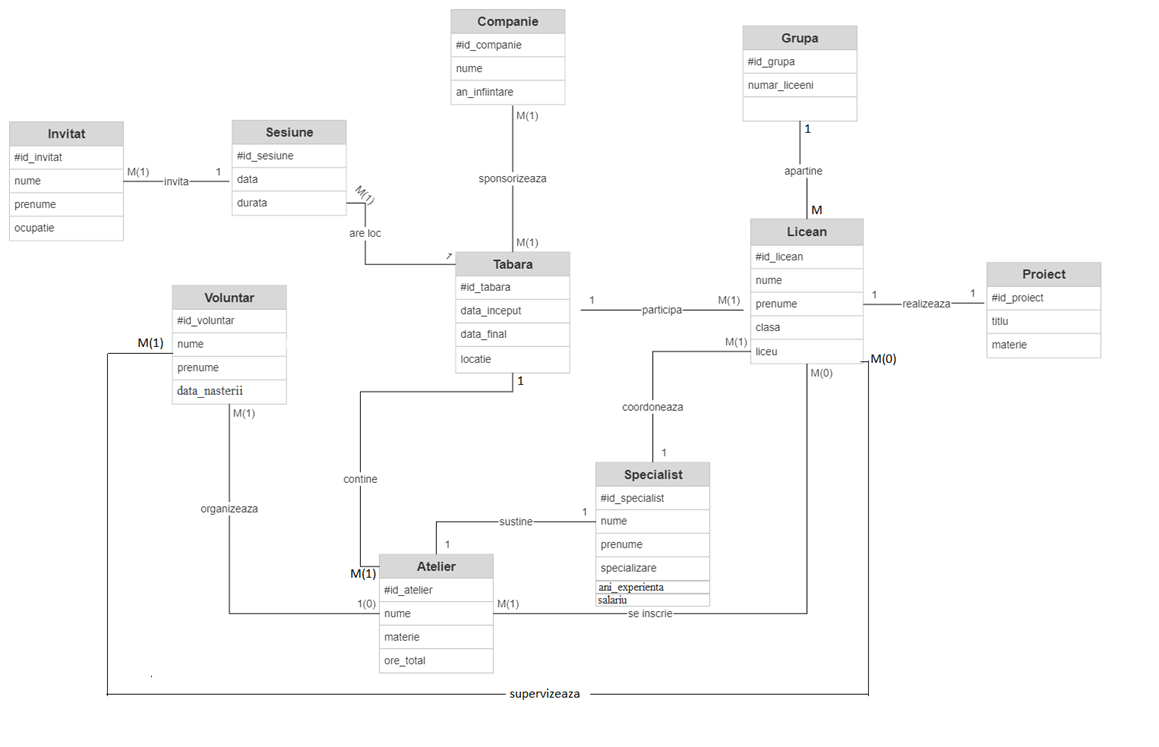
În funcție de numărul de participanți, liceeni sunt împărțiți pe grupe aproximativ egale.

De asemenea, în cadrul taberei au loc sesiuni speciale în care sunt chemați invitați de onoare.

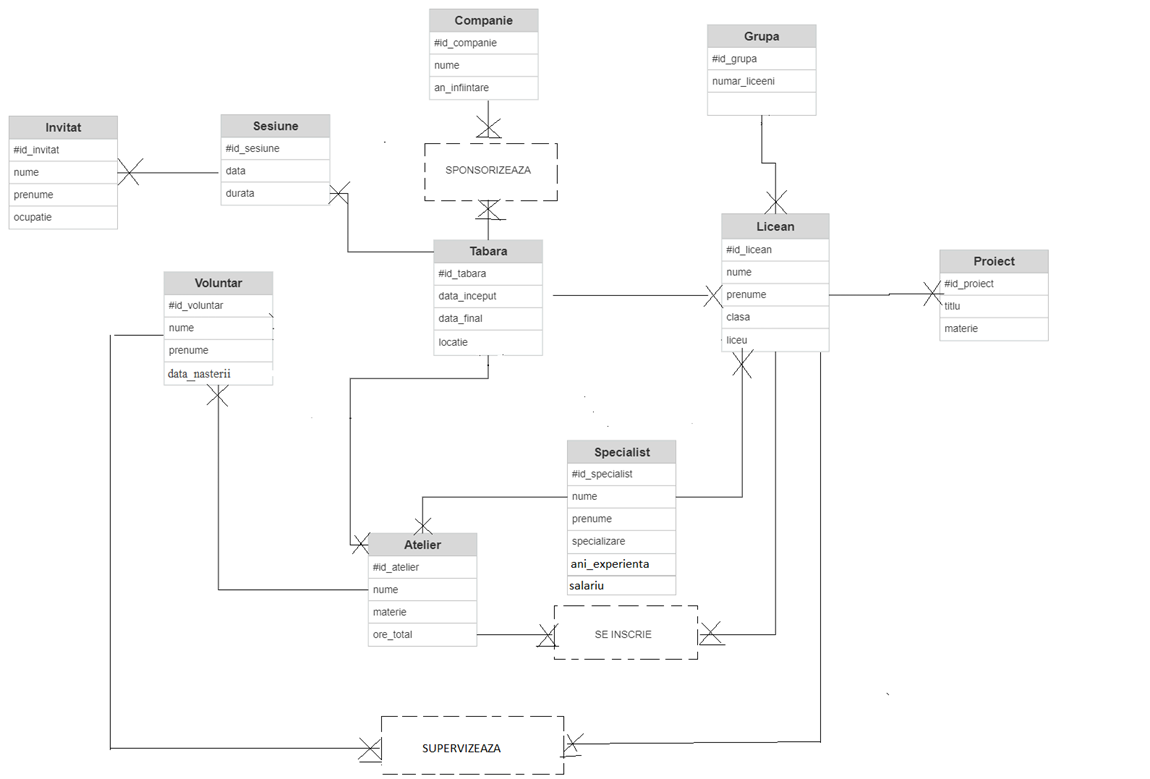
La final, fiecare elev prezintă un proiect cu o tematică specifică unui atelier la care a luat parte. Proiectele se realizează individual.

În vederea realizării proiectului, elevul trebuie să își aleagă un unic coordonator.

1. Diagrama Entitate-Relatie



1. Diagrama conceptuală



CREATE TABLE COMPANIE

(id\_companie NUMBER(5),

nume VARCHAR2(30),

an\_infiintare NUMBER(4),

CONSTRAINT primary\_key\_companie PRIMARY KEY (id\_companie));

ALTER TABLE COMPANIE

MODIFY nume NOT NULL;

CREATE TABLE TABARA

(id\_tabara NUMBER(5),

data\_inceput DATE CONSTRAINT data\_inceput\_nn NOT NULL,

data\_final DATE CONSTRAINT data\_final\_nn NOT NULL,

locatie VARCHAR2(20) CONSTRAINT locatie\_nn NOT NULL,

CONSTRAINT primary\_key\_tabara PRIMARY KEY (id\_tabara));

ALTER TABLE TABARA

ADD CONSTRAINT date\_tabara CHECK (data\_inceput < data\_final);

CREATE TABLE SPECIALIST

(id\_specialist NUMBER(5),

nume VARCHAR2(20) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

specializare VARCHAR2(20) NOT NULL,

ani\_experienta NUMBER(2),

salariu NUMBER (4),

CONSTRAINT primary\_key\_specialist PRIMARY KEY (id\_specialist));

CREATE TABLE SESIUNE

(id\_sesiune NUMBER(5),

data\_sesiune DATE NOT NULL,

durata number(1) NOT NULL,

id\_tabara NUMBER(5),

CONSTRAINT primary\_key\_sesiune PRIMARY KEY (id\_sesiune));

ALTER TABLE SESIUNE

ADD CONSTRAINT id\_tabara\_fk FOREIGN KEY(id\_tabara) REFERENCES TABARA(id\_tabara);

CREATE TABLE INVITAT

(id\_invitat NUMBER(5),

nume VARCHAR2(20) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

ocupatie VARCHAR2(20),

id\_sesiune NUMBER(5),

CONSTRAINT primaty\_key\_invitat PRIMARY KEY (id\_invitat));

ALTER TABLE INVITAT

ADD CONSTRAINT id\_sesiune\_fk FOREIGN KEY(id\_sesiune) REFERENCES SESIUNE(id\_sesiune);

CREATE TABLE GRUPA

(id\_grupa NUMBER(5),

numar\_liceeni NUMBER(1) NOT NULL,

CONSTRAINT primary\_key\_grupa PRIMARY KEY(id\_grupa));

CREATE TABLE ATELIER

(id\_atelier NUMBER(5),

nume VARCHAR2(20) NOT NULL,

ore\_total NUMBER(2),

materie VARCHAR2(20),

id\_tabara NUMBER(5),

id\_specialist NUMBER(5),

CONSTRAINT primaty\_key\_atelier PRIMARY KEY (id\_atelier));

ALTER TABLE ATELIER

ADD CONSTRAINT id\_tabara\_fk\_atelier FOREIGN KEY(id\_tabara) REFERENCES TABARA(id\_tabara);

ALTER TABLE ATELIER

ADD CONSTRAINT id\_specialist\_fk\_atelier FOREIGN KEY(id\_specialist) REFERENCES SPECIALIST(id\_specialist);

CREATE TABLE LICEAN

(id\_licean NUMBER(5),

nume VARCHAR2(20) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

clasa NUMBER(2) CONSTRAINT clasa\_nr CHECK(clasa >= 9 and clasa <= 12),

liceu VARCHAR2(30),

id\_tabara NUMBER(5),

id\_grupa NUMBER(5),

id\_specialist NUMBER(5),

CONSTRAINT primary\_key\_licean PRIMARY KEY (id\_licean));

ALTER TABLE LICEAN

ADD CONSTRAINT id\_tabara\_fk\_licean FOREIGN KEY(id\_tabara) REFERENCES TABARA(id\_tabara);

ALTER TABLE LICEAN

ADD CONSTRAINT id\_grupa\_fk\_licean FOREIGN KEY(id\_grupa) REFERENCES GRUPA(id\_grupa);

ALTER TABLE LICEAN

ADD CONSTRAINT id\_specialist\_fk\_licean FOREIGN KEY(id\_specialist) REFERENCES SPECIALIST(id\_specialist);

CREATE TABLE VOLUNTAR

(id\_voluntar NUMBER(5),

nume VARCHAR2(20) NOT NULL,

prenume VARCHAR2(30) NOT NULL,

data\_nasterii DATE,

id\_atelier NUMBER(5),

CONSTRAINT primary\_key\_voluntar PRIMARY KEY (id\_voluntar));

ALTER TABLE VOLUNTAR

ADD CONSTRAINT id\_atelier\_fk\_voluntar FOREIGN KEY(id\_atelier) REFERENCES ATELIER(id\_atelier);

CREATE TABLE PROIECT

(id\_proiect NUMBER (5),

titlu VARCHAR2(20) NOT NULL,

materie VARCHAR2(20),

id\_licean NUMBER(5),

CONSTRAINT primary\_key\_proiect PRIMARY KEY (id\_proiect));

ALTER TABLE PROIECT

ADD CONSTRAINT id\_licean\_fk\_proiect FOREIGN KEY (id\_licean) REFERENCES LICEAN(id\_licean);

CREATE TABLE SPONSORIZEAZA

(id\_companie NUMBER(5),

id\_tabara NUMBER(5),

suma NUMBER (6),

CONSTRAINT primary\_key\_sponsorizeaza PRIMARY KEY (id\_companie, id\_tabara));

ALTER TABLE SPONSORIZEAZA

ADD CONSTRAINT id\_companie\_fk\_sponsor FOREIGN KEY(id\_companie) REFERENCES COMPANIE(id\_companie);

ALTER TABLE SPONSORIZEAZA

ADD CONSTRAINT id\_tabara\_fk\_sponsor FOREIGN KEY(id\_tabara) REFERENCES TABARA(id\_tabara);

ALTER TABLE SPONSORIZEAZA

MODIFY suma NOT NULL;

CREATE TABLE SUPERVIZEAZA

(id\_voluntar NUMBER(5),

id\_licean NUMBER(5),

nr\_de\_ore NUMBER(1) not null,

CONSTRAINT primary\_key\_supervizeaza PRIMARY KEY (id\_voluntar, id\_licean),

CONSTRAINT id\_voluntar\_fk\_super FOREIGN KEY (id\_voluntar) REFERENCES VOLUNTAR(id\_voluntar),

CONSTRAINT id\_licean\_fk\_super FOREIGN KEY(id\_licean) REFERENCES LICEAN(id\_licean));

CREATE TABLE SE\_INSCRIE

(id\_licean NUMBER(5),

id\_atelier NUMBER(5),

data\_inscriere DATE NOT NULL,

CONSTRAINT primary\_key\_inscrie PRIMARY KEY(id\_licean, id\_atelier),

CONSTRAINT id\_licean\_fk\_inscrie FOREIGN KEY (id\_licean) REFERENCES LICEAN(id\_licean),

CONSTRAINT id\_atelier\_fk\_inscrie FOREIGN Key(id\_atelier) REFERENCES ATELIER(id\_atelier));

CREATE SEQUENCE SEQ\_GRUPA3

INCREMENT BY 10

START WITH 10

MAXVALUE 1000

NOCYCLE;

INSERT INTO GRUPA

VALUES(SEQ\_GRUPA3.NEXTVAL, 3);

INSERT INTO GRUPA

VALUES(SEQ\_GRUPA3.NEXTVAL, 5);

INSERT INTO GRUPA

VALUES(SEQ\_GRUPA3.NEXTVAL, 5);

INSERT INTO GRUPA

VALUES(SEQ\_GRUPA3.NEXTVAL, 4);

INSERT INTO GRUPA

VALUES(SEQ\_GRUPA3.NEXTVAL, 5);

CREATE SEQUENCE SEQ\_TABARA3

INCREMENT BY 10

START WITH 10

MAXVALUE 10000

NOCYCLE;

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('05-11-2018', 'DD-MM-YYYY'), TO\_DATE('09-11-2018', 'DD-MM-YYYY'), 'ARAD');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('13-04-2019', 'DD-MM-YYYY'), TO\_DATE('18-04-2019', 'DD-MM-YYYY'), 'BUCURESTI');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('12-08-2019', 'DD-MM-YYYY'), TO\_DATE('18-08-2019', 'DD-MM-YYYY'), 'SIBIU');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('10-02-2020', 'DD-MM-YYYY'), TO\_DATE('14-02-2020', 'DD-MM-YYYY'), 'CLUJ-NAPOCA');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('11-09-2020', 'DD-MM-YYYY'), TO\_DATE('18-09-2020', 'DD-MM-YYYY'), 'BUCURESTI');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('04-03-2018', 'DD-MM-YYYY'), TO\_DATE('09-03-2018', 'DD-MM-YYYY'), 'TIMISOARA');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('05-08-2019', 'DD-MM-YYYY'), TO\_DATE('09-08-2019', 'DD-MM-YYYY'), 'IASI');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('05-11-2017', 'DD-MM-YYYY'), TO\_DATE('09-11-2017', 'DD-MM-YYYY'), 'GALATI');

INSERT INTO TABARA

VALUES(SEQ\_TABARA3.NEXTVAL, TO\_DATE('02-06-2017', 'DD-MM-YYYY'), TO\_DATE('08-06-2017', 'DD-MM-YYYY'), 'IASI');

SELECT \* FROM TABARA;

--REVENIRE EXERCITIU 10

INSERT INTO COMPANIE

VALUES(10, 'AQUACARPATICA.SRL', 2002);

INSERT INTO COMPANIE

VALUES(20, 'SUPERBET.SRL', 2004);

INSERT INTO COMPANIE

VALUES(30, 'EUROSTAR STUDIOS.SRL', 2002);

INSERT INTO COMPANIE

VALUES(40, 'KAUFLAND.SRL', 1999);

INSERT INTO COMPANIE

VALUES(50, 'FLOWERS DESIGN.SRL', 2009);

INSERT INTO SPECIALIST

VALUES(10, 'Pop', 'Alexandra', 'biologie moleculara', 12, 1800);

INSERT INTO SPECIALIST

VALUES(20, 'Pop', 'Matei', 'matematica', 12, 3000);

INSERT INTO SPECIALIST

VALUES(30, 'Popa', 'Ion', 'biologie', 8, 2500);

INSERT INTO SPECIALIST

VALUES(40, 'Munteanu', 'Diana', 'informatica', 15, 3300);

INSERT INTO SPECIALIST

VALUES(50, 'Anton', 'Maria', 'engleza', 8, 2000);

INSERT INTO SPECIALIST

VALUES(60, 'Bejan', 'Andrei', 'matematica', 14, 3250);

INSERT INTO ATELIER

VALUES (10, 'Bazele biologiei', 12, 'biologie', 30, 30);

INSERT INTO ATELIER

VALUES (20, 'Literatura engleza', 12, 'engleza', 20, 50);

INSERT INTO ATELIER

VALUES (30, 'Matematica avansata', 12, 'matematica', 30, 60);

INSERT INTO ATELIER

VALUES (40, 'Algoritmi de baza', 12, 'informatica', 40, 40);

INSERT INTO ATELIER

VALUES (50, 'Bazele biologiei II', 10, 'biologie', 30, 10);

INSERT INTO ATELIER

VALUES (60, 'Biologie moleculara', 12, 'biologie moleculara', 50, 10);

INSERT INTO ATELIER

VALUES (70, 'Structuri de date', 14, 'informatica', 60, 40);

INSERT INTO ATELIER

VALUES (80, 'Engleza avansata', 9, 'engleza', 70, 50);

INSERT INTO ATELIER

VALUES (90, 'Fizica aplicata', 12, 'fizica', 80, 60);

INSERT INTO ATELIER

VALUES (100, 'Fizica si mate', 14, 'fizica', 90, 20);

INSERT INTO ATELIER

VALUES (110, 'Informatica', 15, 'informatica', 100, 40);

INSERT INTO VOLUNTAR

VALUES(10, 'Popescu', 'Clara', TO\_DATE('26-11-2002', 'dd-mm-yyyy'), 30);

INSERT INTO VOLUNTAR

VALUES(20, 'Marin', 'Andreea', TO\_DATE('14-09-2000', 'dd-mm-yyyy'), 40);

INSERT INTO VOLUNTAR

VALUES(30, 'Mircea', 'Ana', TO\_DATE('06-04-2001', 'dd-mm-yyyy'), 50);

INSERT INTO VOLUNTAR

VALUES(40, 'Marinescu', 'Larisa', TO\_DATE('24-01-2002', 'dd-mm-yyyy'), 60);

INSERT INTO VOLUNTAR

VALUES(50, 'Popescu', 'Clara', TO\_DATE('26-11-2002', 'dd-mm-yyyy'), 10);

INSERT INTO VOLUNTAR

VALUES(60, 'Popescu', 'Clara', TO\_DATE('26-11-2002', 'dd-mm-yyyy'), 20);

UPDATE VOLUNTAR

SET NUME = 'Ciurea' , PRENUME = 'Bianca'

WHERE id\_voluntar = 50;

UPDATE VOLUNTAR

SET NUME = 'Bogdan' , PRENUME = 'Ioana'

WHERE id\_voluntar = 60;

UPDATE VOLUNTAR

SET data\_nasterii = TO\_DATE('02-08-2001', 'dd-mm-yyyy')

WHERE id\_voluntar = 50;

UPDATE VOLUNTAR

SET data\_nasterii = TO\_DATE('03-09-2001', 'dd-mm-yyyy')

WHERE id\_voluntar = 60;

INSERT INTO VOLUNTAR

VALUES(70, 'Baciu', 'Otilia', TO\_DATE('22-04-2000', 'dd-mm-yyyy'),30);

INSERT INTO VOLUNTAR

VALUES(80, 'Vladici', 'Sara', TO\_DATE('02-04-2000', 'dd-mm-yyyy'),10);

INSERT INTO VOLUNTAR

VALUES(90, 'Bob', 'Alice', TO\_DATE('21-06-2000', 'dd-mm-yyyy'), 30);

INSERT INTO VOLUNTAR

VALUES(100, 'Bran', 'Alexandru', TO\_DATE('12-09-2001', 'dd-mm-yyyy'), 20);

INSERT INTO VOLUNTAR

VALUES(110, 'Costea', 'Mihail', TO\_DATE('20-12-2000', 'dd-mm-yyyy'), 60);

INSERT INTO VOLUNTAR

VALUES(120, 'Corolescu', 'Catalin', TO\_DATE('29-01-2002', 'dd-mm-yyyy'), 60);

INSERT INTO VOLUNTAR

VALUES(130, 'Mihai', 'Violeta', TO\_DATE('28-08-2000', 'dd-mm-yyyy'), 50);

INSERT INTO VOLUNTAR

VALUES(140, 'Manea', 'Tudor', TO\_DATE('17-05-2000', 'dd-mm-yyyy'), 30);

INSERT INTO VOLUNTAR

VALUES(150, 'Sorescu', 'Teodor', TO\_DATE('13-10-2001', 'dd-mm-yyyy'), 20);

INSERT INTO SESIUNE

VALUES(10, TO\_DATE('06-11-2018', 'DD-MM-YYYY'), 3, 20);

INSERT INTO SESIUNE

VALUES(20, TO\_DATE('15-04-2019', 'DD-MM-YYYY'), 2, 30);

INSERT INTO SESIUNE

VALUES(30, TO\_DATE('13-08-2019', 'DD-MM-YYYY'), 3, 40);

INSERT INTO SESIUNE

VALUES(40, TO\_DATE('13-02-2020', 'DD-MM-YYYY'), 3, 50);

INSERT INTO SESIUNE

VALUES(50, TO\_DATE('17-09-2020', 'DD-MM-YYYY'), 3, 60);

INSERT INTO SESIUNE

VALUES(60, TO\_DATE('04-03-2018', 'DD-MM-YYYY'), 3, 70);

INSERT INTO SESIUNE

VALUES(70, TO\_DATE('07-08-2019', 'DD-MM-YYYY'), 3, 80);

INSERT INTO SESIUNE

VALUES(80, TO\_DATE('06-11-2017', 'DD-MM-YYYY'), 3, 90);

INSERT INTO SESIUNE

VALUES(90, TO\_DATE('06-06-2017', 'DD-MM-YYYY'), 3, 100);

INSERT INTO INVITAT

VALUES(10, 'Anghel', 'Sorana', 'antreprenor', 10);

INSERT INTO INVITAT

VALUES(20, 'Angelescu', 'Bogdan', 'atlet', 20);

INSERT INTO INVITAT

VALUES(30, 'Milu', 'Alexandru', 'atlet', 30);

INSERT INTO INVITAT

VALUES(40, 'Barbu', 'Simona', 'psiholog', 40);

INSERT INTO INVITAT

VALUES(50, 'Antonescu', 'Alina', 'politician', 50);

INSERT INTO INVITAT

VALUES(60, 'Mircescu', 'Vlad', 'antreprenor', 60);

INSERT INTO INVITAT

VALUES(70, 'Manea', 'Bogdan', 'antreprenor', 70);

INSERT INTO INVITAT

VALUES(80, 'Begu', 'Briana', 'psiholog', 80);

INSERT INTO INVITAT

VALUES(90, 'Carstea', 'Marian', 'antreprenor', 90);

INSERT INTO INVITAT

VALUES(100, 'Deaconu', 'Razvan', 'filantrop', 70);

INSERT INTO LICEAN

VALUES(10, 'Popa', 'Alexandra', 10, 'Colegiul National Unirea', 20, 20, 10);

INSERT INTO LICEAN

VALUES(20, 'Minea', 'Alin', 9, 'CN Mihai Eminescu', 30, 30, 20);

INSERT INTO LICEAN

VALUES(30, 'Pelea', 'Mihail', 12, 'CN Ion Luca Caragiale', 40, 20, 10);

INSERT INTO LICEAN

VALUES(40, 'Manea', 'Sandra', 10, 'CN A.I. Cuza', 50, 30, 40);

INSERT INTO LICEAN

VALUES(50, 'Banciu', 'Maria', 11, 'CN Tehnologic Buzau', 60, 40, 50);

INSERT INTO LICEAN

VALUES(60, 'Vasiliu', 'Catalin', 10, 'CN Pedagogic Zalau', 70, 40, 50);

INSERT INTO LICEAN

VALUES(70, 'Neacsu', 'Teodora', 12, 'Liceul Teoretic Nr.1 Vaslui', 80, 20, 60);

INSERT INTO LICEAN

VALUES(80, 'Velea', 'Miruna', 9, 'CN Marin Preda', 90, 30, 10);

INSERT INTO LICEAN

VALUES(90, 'Popescu', 'Lavinia', 10, 'Colegiul National Unirea', 100, 20, 60);

INSERT INTO LICEAN

VALUES(100, 'Nicolae', 'Daria', 9, 'CN Constantin Noica', 20, 20, 20);

INSERT INTO LICEAN

VALUES(110, 'Nania', 'Alexandra', 10, 'CN Tehnologic Nr.2 Braila', 20, 40, 10);

INSERT INTO LICEAN

VALUES(120, 'Popa', 'Mircea', 11, 'Liceul Teoretic Nr.4 Bucuresti', 20, 50, 10);

INSERT INTO LICEAN

VALUES(130, 'Barbu', 'Laurentiu', 12, 'CN Pedagogic Alexandria', 20, 50, 30);

INSERT INTO LICEAN

VALUES(140, 'Chircu', 'Alex', 12, 'CN Stefan Cel Mare', 20, 50, 30);

INSERT INTO LICEAN

VALUES(150, 'Minea', 'Valeria', 12, 'CN Marin Preda', 80, 40, 30);

INSERT INTO LICEAN

VALUES(160, 'Badea', 'Marina', 11, 'Liceul Teoretic Nr.2 Braila', 80, 40, 20);

INSERT INTO LICEAN

VALUES(170, 'Badescu', 'Gina', 9, 'CN Vasile Alecsandri', 100, 50, 50);

INSERT INTO LICEAN

VALUES(180, 'Horea', 'Iulia', 10, 'CN Stefan Cel Mare', 90, 20, 40);

INSERT INTO LICEAN

VALUES(190, 'Bordei', 'Isabela', 10, 'CN Al. I. Cuza', 60, 50, 50);

INSERT INTO LICEAN

VALUES(200, 'Patrascu', 'Andra', 12, 'Liceul Tehnologic Nr.2 Iasi', 20, 50, 20);

INSERT INTO PROIECT

VALUES(10, 'Fizica de azi', 'fizica', 100);

INSERT INTO PROIECT

VALUES(20, 'Mate aplicata', 'matematica', 30);

INSERT INTO PROIECT

VALUES(30, 'Biologie aprofundata', 'biologie moleculara', 40);

INSERT INTO PROIECT

VALUES(40, 'Probleme informatica', 'informatica', 50);

INSERT INTO PROIECT

VALUES(50, 'Literatura engleza', 'engleza', 60);

INSERT INTO PROIECT

VALUES(60, 'Aplicatii', 'matematica', 10);

INSERT INTO PROIECT

VALUES(70, 'Fizica moderna', 'fizica', 20);

INSERT INTO PROIECT

VALUES(80, 'Sub microscop', 'biologie moleculara', 70);

INSERT INTO PROIECT

VALUES(90, 'Arta cuvintelor', 'engleza', 80);

INSERT INTO PROIECT

VALUES(100, 'Fizica altfel', 'fizica', 110);

INSERT INTO PROIECT

VALUES(110, 'Studiul molecular', 'biologie moleculara', 90);

INSERT INTO PROIECT

VALUES(120,'Probleme reale', 'matematica', 120);

INSERT INTO PROIECT

VALUES(130, 'O noua abordare', 'informatica', 130);

INSERT INTO PROIECT

VALUES(140, 'Aplicatii utile', 'informatica', 140);

INSERT INTO PROIECT

VALUES(150, 'Teoria informatiei', 'informatica', 150);

INSERT INTO PROIECT

VALUES(160, 'English Tenses', 'engleza', 160);

INSERT INTO PROIECT

VALUES(170, 'US History', 'engleza', 170);

INSERT INTO PROIECT

VALUES(180, 'Multimea nr reale', 'matematica', 180);

INSERT INTO PROIECT

VALUES(190, 'Regnul animal', 'biologie', 190);

INSERT INTO PROIECT

VALUES(200, 'Legile lui Newton', 'fizica', 200);

INSERT INTO SPONSORIZEAZA

VALUES(20,20, 15000);

INSERT INTO SPONSORIZEAZA

VALUES(10,20, 12000);

INSERT INTO SPONSORIZEAZA

VALUES(20,30, 15000);

INSERT INTO SPONSORIZEAZA

VALUES(20,70, 20000);

INSERT INTO SPONSORIZEAZA

VALUES(30,20, 5000);

INSERT INTO SPONSORIZEAZA

VALUES(40,40, 15000);

INSERT INTO SPONSORIZEAZA

VALUES(50,50, 70000);

INSERT INTO SPONSORIZEAZA

VALUES(20,60, 15000);

INSERT INTO SPONSORIZEAZA

VALUES(10,70, 65000);

INSERT INTO SPONSORIZEAZA

VALUES(30,80, 24500);

INSERT INTO SPONSORIZEAZA

VALUES(20,90, 15000);

INSERT INTO SPONSORIZEAZA

VALUES(30,100, 100000);

INSERT INTO SPONSORIZEAZA

VALUES(40,100, 1800);

INSERT INTO SPONSORIZEAZA

VALUES(10,90, 29000);

INSERT INTO SPONSORIZEAZA

VALUES(40,80, 17000);

SELECT \* FROM SPONSORIZEAZA;

INSERT INTO SUPERVIZEAZA

VALUES(10, 20, 8);

INSERT INTO SUPERVIZEAZA

VALUES(20,10, 3);

INSERT INTO SUPERVIZEAZA

VALUES(30,10, 2);

INSERT INTO SUPERVIZEAZA

VALUES(40,30, 5);

INSERT INTO SUPERVIZEAZA

VALUES(50,40, 5);

INSERT INTO SUPERVIZEAZA

VALUES(60,10, 1);

INSERT INTO SUPERVIZEAZA

VALUES(70,50, 4);

INSERT INTO SUPERVIZEAZA

VALUES(70,60, 4);

INSERT INTO SUPERVIZEAZA

VALUES(40,20, 3);

INSERT INTO SUPERVIZEAZA

VALUES(30,30, 5);

INSERT INTO SUPERVIZEAZA

VALUES(60,90, 3);

INSERT INTO SUPERVIZEAZA

VALUES(20,100, 3);

INSERT INTO SUPERVIZEAZA

VALUES(70,120, 5);

INSERT INTO SUPERVIZEAZA

VALUES(10,110, 6);

INSERT INTO SUPERVIZEAZA

VALUES(20,130, 3);

INSERT INTO SUPERVIZEAZA

VALUES(40,140, 7);

INSERT INTO SUPERVIZEAZA

VALUES(50,70, 3);

INSERT INTO SUPERVIZEAZA

VALUES(20,80, 3);

INSERT INTO SUPERVIZEAZA

VALUES(70,130, 5);

INSERT INTO SUPERVIZEAZA

VALUES(20,140, 3);

INSERT INTO SUPERVIZEAZA

VALUES(20,150, 2);

INSERT INTO SUPERVIZEAZA

VALUES(40,160, 3);

INSERT INTO SUPERVIZEAZA

VALUES(10,170, 2);

INSERT INTO SUPERVIZEAZA

VALUES(50,180, 4);

INSERT INTO SUPERVIZEAZA

VALUES(10,190, 3);

INSERT INTO SUPERVIZEAZA

VALUES(20,200, 1);

INSERT INTO SUPERVIZEAZA

VALUES(20,20, 1);

INSERT INTO SUPERVIZEAZA

VALUES(40,10, 4);

INSERT INTO SUPERVIZEAZA

VALUES(20,30, 1);

INSERT INTO SE\_INSCRIE

VALUES(10, 10, TO\_DATE('12-03-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(20, 10, TO\_DATE('13-03-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(30, 20, TO\_DATE('12-09-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(40, 30, TO\_DATE('02-03-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(50, 40, TO\_DATE('12-11-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(60, 50, TO\_DATE('12-03-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(70, 60, TO\_DATE('04-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(80, 10, TO\_DATE('12-03-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(90, 30, TO\_DATE('01-04-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(100, 40, TO\_DATE('30-11-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(110, 50, TO\_DATE('11-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(120, 60, TO\_DATE('17-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(130, 20, TO\_DATE('09-09-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(140, 20, TO\_DATE('17-09-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(140, 10, TO\_DATE('10-03-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(110, 40, TO\_DATE('12-11-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(90, 50, TO\_DATE('12-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(150, 50, TO\_DATE('12-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(160, 10, TO\_DATE('11-03-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(170, 40, TO\_DATE('12-11-2019', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(180, 20, TO\_DATE('16-09-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(190, 60, TO\_DATE('12-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(200, 60, TO\_DATE('10-06-2020', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(10, 20, TO\_DATE('10-09-2018', 'DD-MM-YYYY'));

INSERT INTO SE\_INSCRIE

VALUES(20, 20, TO\_DATE('11-09-2018', 'DD-MM-YYYY'));

SELECT \* FROM se\_inscrie;

ALTER TABLE LICEAN

ADD Cheltuieli\_transport NUMBER(3);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 50

WHERE (ID\_LICEAN = 10 OR ID\_LICEAN = 20);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 97

WHERE (ID\_LICEAN = 40 OR ID\_LICEAN = 120);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 70

WHERE (ID\_LICEAN = 60 OR ID\_LICEAN = 70 OR ID\_LICEAN = 150);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 110

WHERE (ID\_LICEAN = 200);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 120

WHERE (ID\_LICEAN = 100);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 100

WHERE (ID\_LICEAN = 50);

UPDATE LICEAN

SET CHELTUIELI\_TRANSPORT = 125

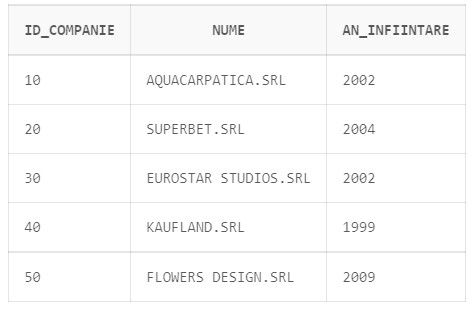
WHERE (ID\_LICEAN = 30);

UPDATE LICEAN

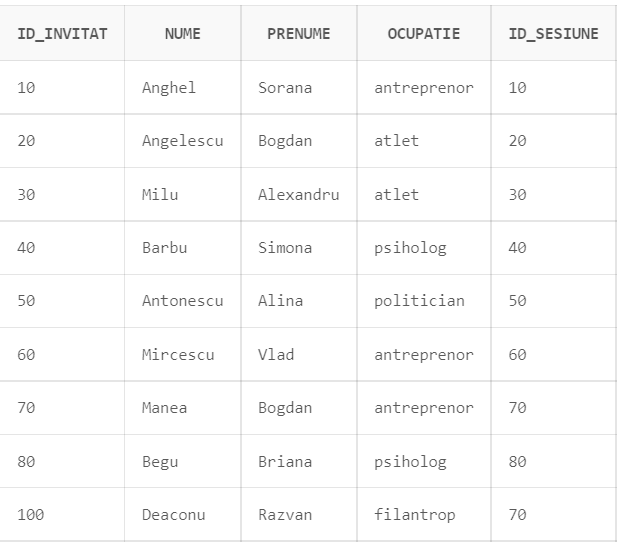
SET CHELTUIELI\_TRANSPORT = 30

WHERE (ID\_LICEAN = 190);

select \* from companie;



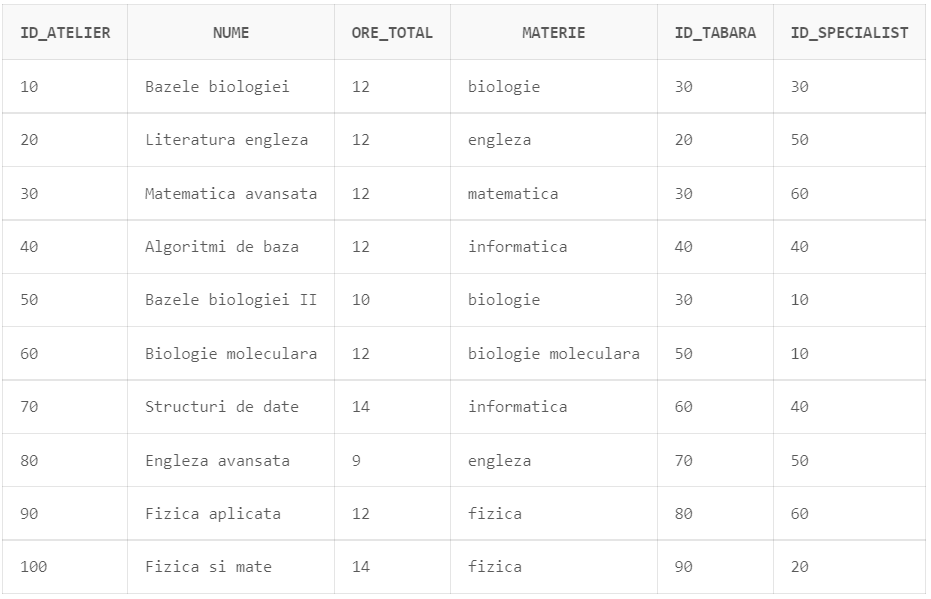
Select \* from invitat;



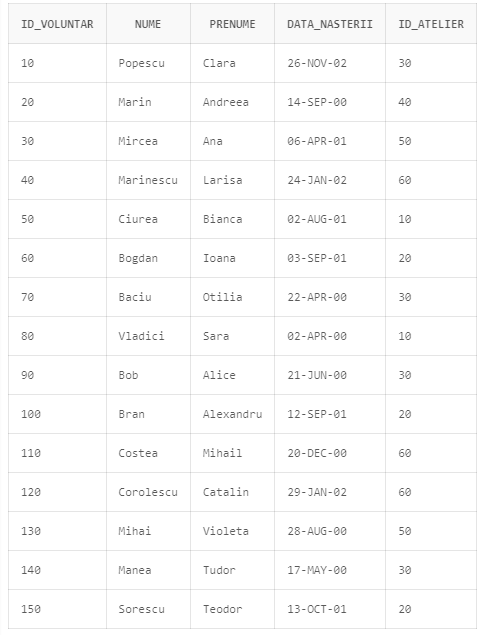
Select \* from licean;



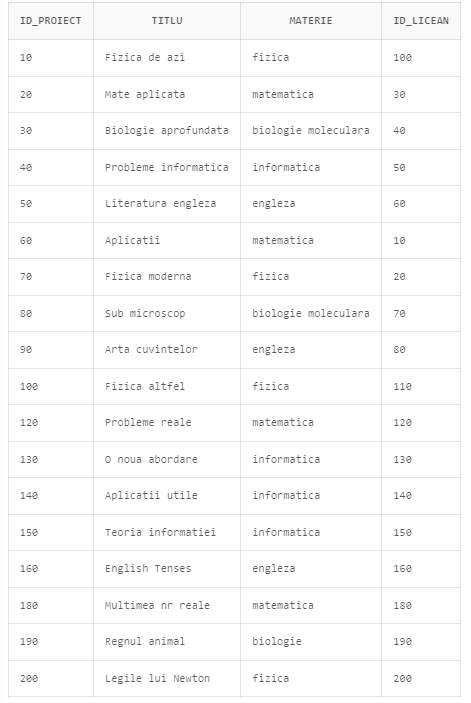
Select \* from atelier;



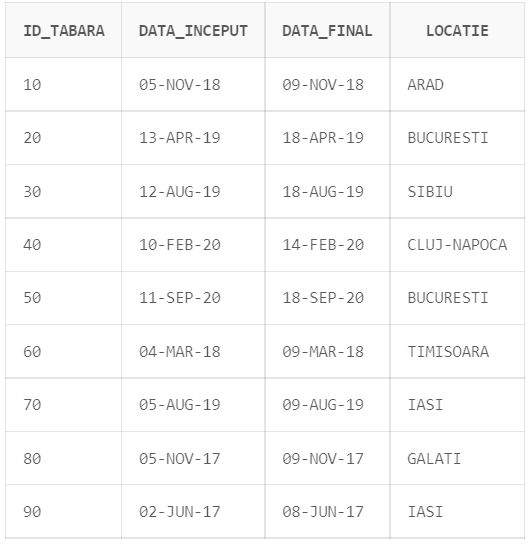
Select \* from voluntar;



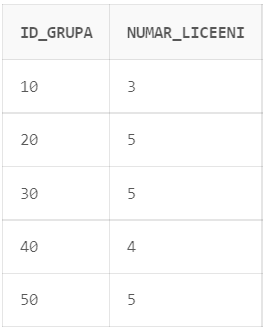
Select \* from proiect;



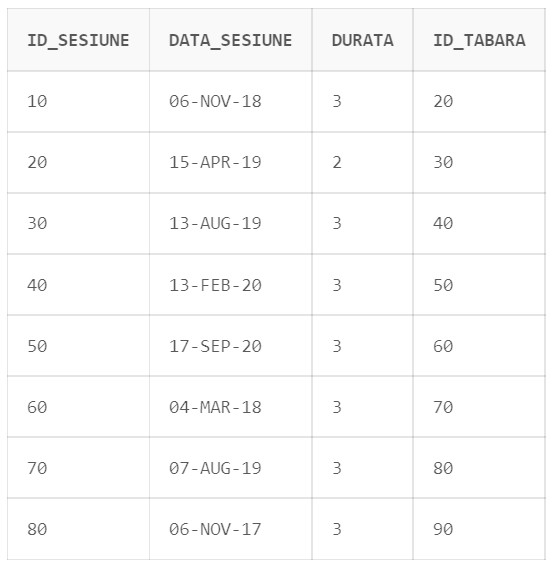
Select \* from tabara;



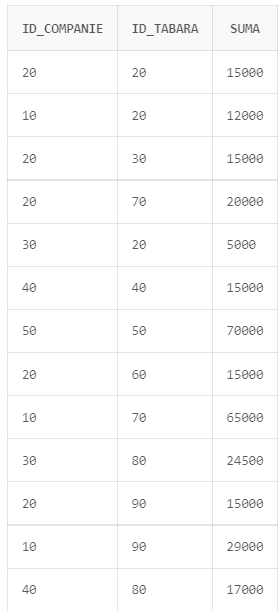
Select \* from grupa;



Select \* from sesiune;



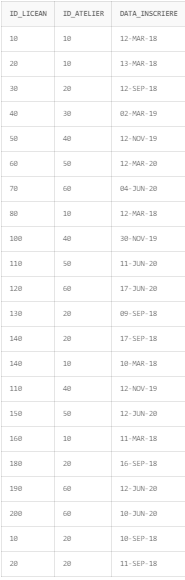
Select \* from sponsorizeaza;



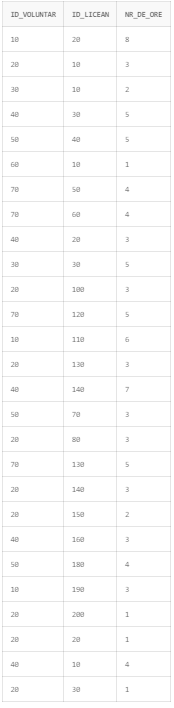
Select \* from specialist;



Select \* from se\_inscrie;



Select \* from supervizeaza;



1. Pentru o companie al cărui cod este dat ca parametru, să se afișeze numele companiei, sumele de bani oferite în sponsorizări, taberele (locatia si id-ul) care au beneficiat de sponsorizare, și totalul oferit. Să se trateze cazul în care codul dat ca parametru nu există.

Am folosit tablou indexat, 2 vectori.

create or replace procedure ex\_6(cod companie.id\_companie%type)

is

type tablou\_indexat is

table of sponsorizeaza%rowtype

index by binary\_integer;

t tablou\_indexat;

type vector\_sume is varray(100) of number;

sume vector\_sume := vector\_sume();

type vector\_locatie is varray(100) of varchar2(300);

locatii vector\_locatie :=vector\_locatie();

sponsor companie.nume%type;

total number(7) := 0;

j number(3) := 1;

nume\_oras tabara.locatie%type;

begin

select nume into sponsor

from companie

where id\_companie = cod;

select \* bulk collect into t

from sponsorizeaza;

for i in t.first..t.last loop

if t(i).id\_companie = cod then

sume.extend();

sume(j) := t(i).suma;

total := total + t(i).suma;

locatii.extend();

locatii(j) := t(i).id\_tabara;

j := j + 1;

end if;

end loop;

dbms\_output.put\_line('Numele companiei este ' || sponsor);

dbms\_output.put\_line('Suma totala acordata in sponsorizari a fost de ' || total || ' dupa cum urmeaza');

for i in locatii.first..locatii.last loop

select locatie into nume\_oras

from tabara

where id\_tabara = locatii(i);

dbms\_output.put\_line('Suma de ' || sume(i) || ' pentru tabara din ' || nume\_oras || ' cu id-ul de ' || locatii(i));

end loop;

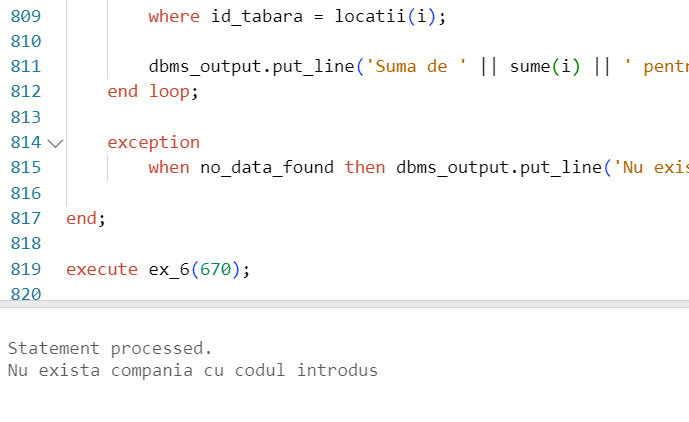
exception

when no\_data\_found then dbms\_output.put\_line('Nu exista compania cu codul introdus');

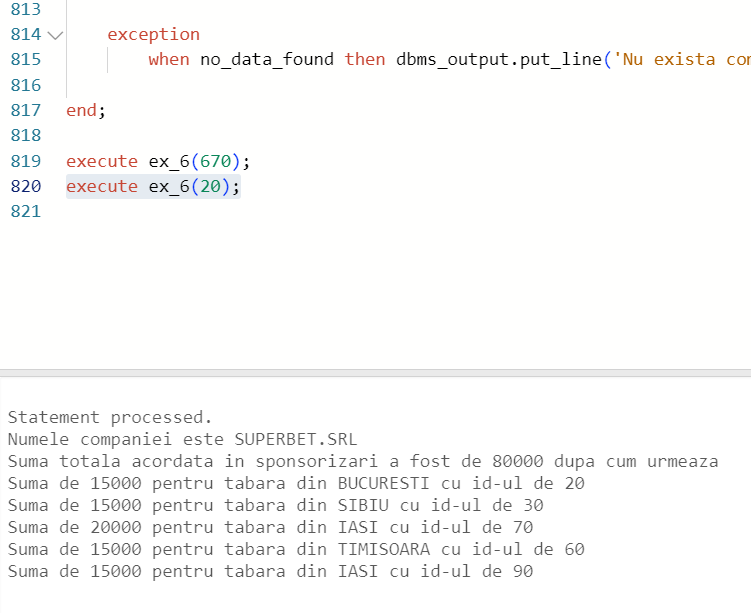
end;

execute ex\_6(670);

În acest caz se va intra pe excepția de NO\_DATA\_FOUND:



execute ex\_6(20);



1. Pentru liceenii a căror clasă este dată ca parametru, să se afișeze lista cu numele lor, precum și lista cu toate atelierele la care s-au înscris, dar și numele specialiștilor care au coordonat atelierul. La final să se afișeze numele și prenumele elevului care a participat la cele mai multe ateliere. Tratați cazul în care clasa dată ca parametru este invalidă.

Am folosit cursor parametrizat pentru obținerea elevilor și ciclu cursor cu subcereri pentru obținerea id-urilor atelierelor la care s-au înscris, dupa care am obținut numele atelierelor și numele specialiștilor coordonatori.

create or replace procedure ex\_7(class licean.clasa%type)

is

cursor c(class licean.clasa%type)

is select id\_licean, nume, prenume from

licean where clasa = class;

cod licean.id\_licean%type;

nume\_lic licean.nume%type;

prenume\_lic licean.prenume%type;

nume\_atelier atelier.nume%type;

spec specialist.id\_specialist%type;

nume\_s licean.nume%type;

prenume\_s licean.prenume%type;

maxim number(2) := 0;

nume\_max licean.nume%type;

prenume\_max licean.prenume%type;

begin

open c(class);

loop

fetch c into cod, nume\_lic, prenume\_lic;

exit when c%notfound;

dbms\_output.put\_line('NUME ELEV: ' || nume\_lic || ' ' || prenume\_lic);

dbms\_output.put\_line('Ateliere la care a participat: ');

for i in (with tabel as (select id\_licean lic, count(\*) numar from se\_inscrie group by id\_licean) select id\_atelier id, numar nr from se\_inscrie, tabel where id\_licean = cod and lic = cod) loop

select nume into nume\_atelier

from atelier

where id\_atelier = i.id;

select id\_specialist into spec

from atelier where id\_atelier = i.id;

select nume, prenume into nume\_s, prenume\_s

from specialist where id\_specialist = spec;

dbms\_output.put\_line(nume\_atelier || ' coordonat de ' || nume\_s || ' ' || prenume\_s);

if i.nr > maxim then maxim := i.nr;

nume\_max := nume\_lic;

prenume\_max := prenume\_lic;

end if;

end loop;

end loop;

close c;

if cod is null then dbms\_output.put\_line('Nu exista clasa data ca parametru');

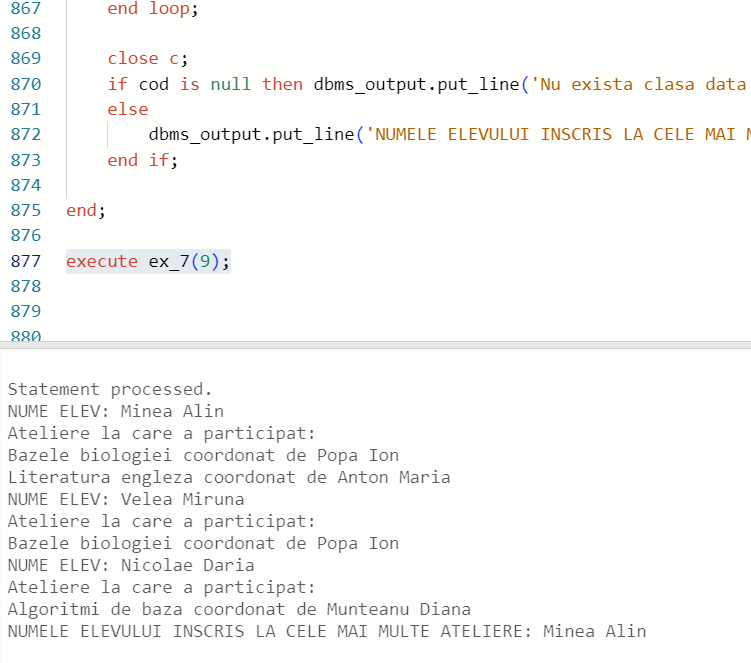
else

dbms\_output.put\_line('NUMELE ELEVULUI INSCRIS LA CELE MAI MULTE ATELIERE: ' || nume\_max || ' ' || prenume\_max);

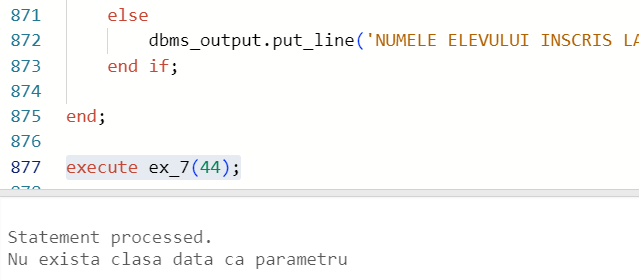
end if;

end;

execute ex\_7(9);



execute ex\_7(44); --clasa nu există, intră pe cazul când codul este null



1. Pentru o tabără al cărui cod este dat ca parametru, să se calculeze suma totală obținută din sponsorizări, lista sponsorilor cu suma oferită de fiecare, precum și suma care mai trebuie obținută pentru a atinge pragul de 50.000 de lei.

Definiți excepții care să trateze situațiile în care codul introdus este incorect , sau dacă tabăra a atins deja pragul dorit de sponsorizări.

create or replace function ex\_8(cod tabara.id\_tabara%type)

return number is

cod\_invalid exception;

cod\_notfound exception;

suma\_atinsa exception;

nume\_comp companie.nume%type;

suma\_max sponsorizeaza.suma%type := 0;

type v\_cod is varray(100) of number;

v v\_cod := v\_cod();

idx number(3) := 1;

gasit number(1) := 0;

begin

if cod <= 0 then raise cod\_invalid;

end if;

for j in (select id\_tabara from tabara) loop

v.extend();

v(idx) := j.id\_tabara;

idx := idx + 1;

end loop;

for i in v.first..v.last loop

if v(i) = cod then gasit := 1;

end if;

end loop;

if gasit = 0 then raise cod\_notfound;

end if;

for i in (select max(s.suma) sum , c.nume name from tabara t join sponsorizeaza s using (id\_tabara)

join companie c using (id\_companie)

where id\_tabara = cod

group by c.nume) loop

suma\_max := suma\_max + i.sum;

dbms\_output.put\_line('Suma: ' || i.sum || ' de la ' || i.name);

end loop;

if suma\_max >= 50000 then raise suma\_atinsa;

end if;

return 50000-suma\_max;

exception

when cod\_invalid then raise\_application\_error(-20001, 'Codul dat nu poate fi negativ!');

when suma\_atinsa then dbms\_output.put\_line('Tabara a atins deja pragul dorit de sponsorizari!');

return 0;

when cod\_notfound then raise\_application\_error(-20002,'Codul nu a fost gasit in tabela!');

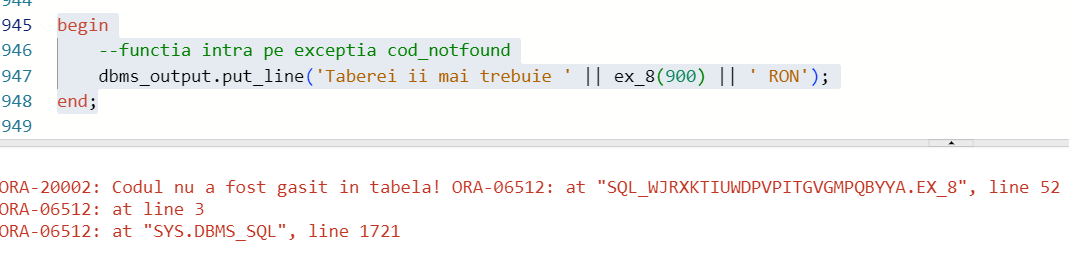
end ex\_8;

begin

--functia intra pe exceptia cod\_notfound

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_8(900) || ' RON');

end;

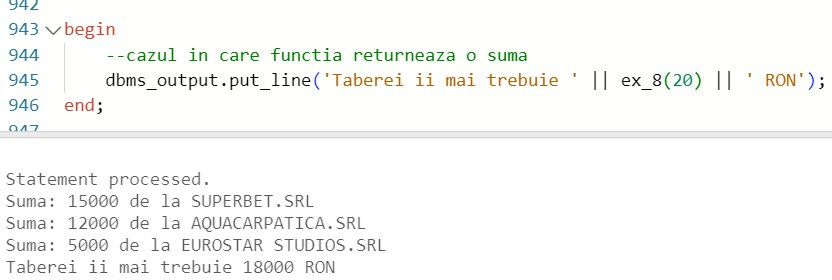


begin

--cazul in care functia returneaza o suma

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_8(20) || ' RON');

end;

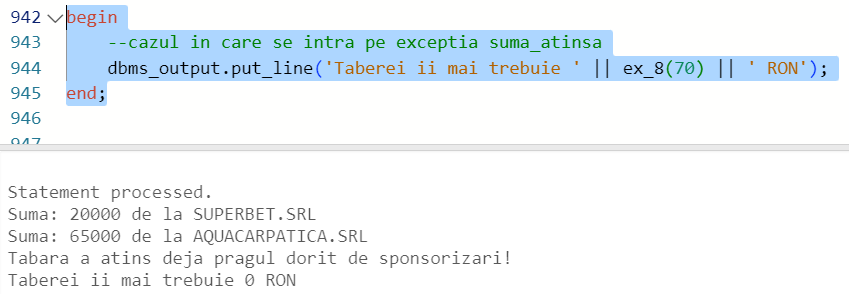


begin

--cazul in care se intra pe exceptia suma\_atinsa

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_8(70) || ' RON');

end;

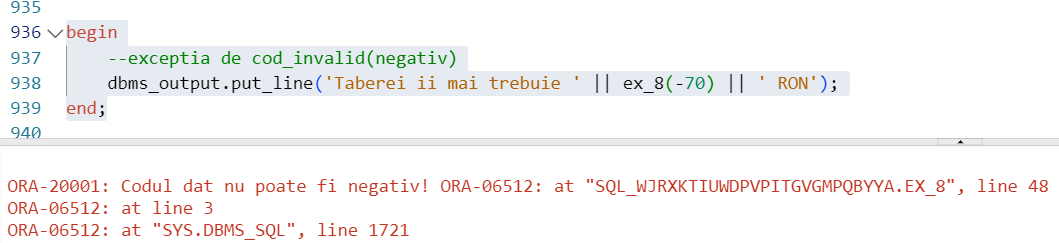


begin

--exceptia de cod\_invalid(negativ)

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_8(-70) || ' RON');

end;



1. Pentru un licean al cărui cod este dat, să se obțină numele și prenumele lui, titlul proiectului realizat, specialistul care l-a coordonat în cadrul taberei, locația în care s-a desfășurat tabăra, precum și atelierul care a fost susținut de specialistul respectiv. Dacă specialistul are mai multe ateliere, să se obțină lista cu toate.

create or replace procedure ex\_9(cod\_lic licean.id\_licean%type)

is

nume\_lic licean.nume%type;

prenume\_lic licean.prenume%type;

nume\_spec specialist.nume%type;

prenume\_spec specialist.prenume%type;

id\_spec specialist.id\_specialist%type;

titlu\_pr proiect.titlu%type;

--cod\_lic licean.id\_licean%type;

nume\_at atelier.nume%type;

oras tabara.locatie%type;

TYPE tabel IS TABLE OF atelier%rowtype INDEX BY PLS\_INTEGER;

tab tabel;

cod\_invalid exception;

begin

if cod\_lic <= 0 then raise cod\_invalid;

end if;

--join pe 5 tabele

select l.nume, l.prenume, s.nume, s.prenume, s.id\_specialist, p.titlu, a.nume, t.locatie

into nume\_lic, prenume\_lic, nume\_spec, prenume\_spec, id\_spec, titlu\_pr, nume\_at, oras

from licean l join tabara t on (l.id\_tabara = t.id\_tabara)

join proiect p on (p.id\_licean = l.id\_licean)

join specialist s on (s.id\_specialist = l.id\_specialist)

join atelier a on (a.id\_specialist = s.id\_specialist)

where l.id\_licean = cod\_lic;

dbms\_output.put\_line('Numele liceanului este ' || nume\_lic || ' ' || prenume\_lic || ', a fost coordonat de specialistul ' || nume\_spec || ' ' ||prenume\_spec ||

' care a organizat atelierul numit ' || nume\_at ||

', a participat la tabara organizata in orasul ' || oras || ' si a realizat proiectul intitulat ' || '"' || titlu\_pr || '"');

exception

when cod\_invalid then raise\_application\_error(-20003, 'Codul nu poate fi negativ');

when no\_data\_found then raise\_application\_error(-20000, 'Nu exista elevul cu codul introdus');

when too\_many\_rows then

dbms\_output.put\_line('Numele liceanului este ' || nume\_lic || ' ' || prenume\_lic || ', a fost coordonat de specialistul ' || nume\_spec || ' ' ||prenume\_spec ||

', a participat la tabara organizata in orasul ' || oras || ' si a realizat proiectul intitulat ' || '"' || titlu\_pr || '"');

dbms\_output.put\_line('Specialistul sustine mai multe ateliere, dupa cum urmeaza: ');

select \* bulk collect into tab

from atelier

where id\_specialist = id\_spec;

for i in tab.first..tab.last loop

dbms\_output.put\_line(tab(i).nume);

end loop;

when others then raise\_application\_error(-20002, 'Au aparut erori neprevazute');

end ex\_9;

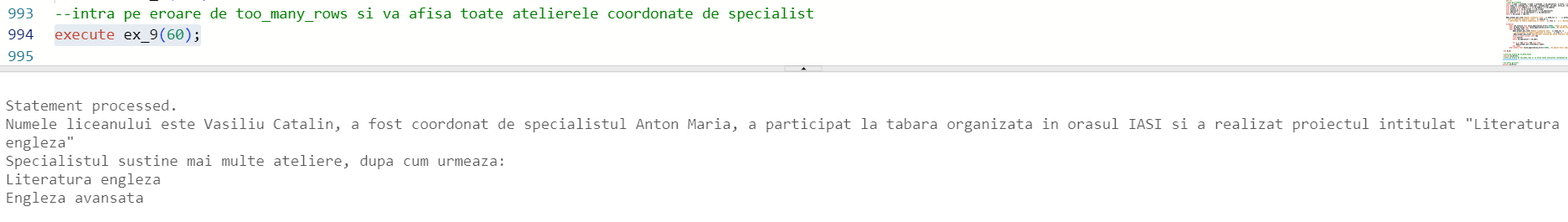
--intra pe eroare de no\_data\_found

execute ex\_9(430);



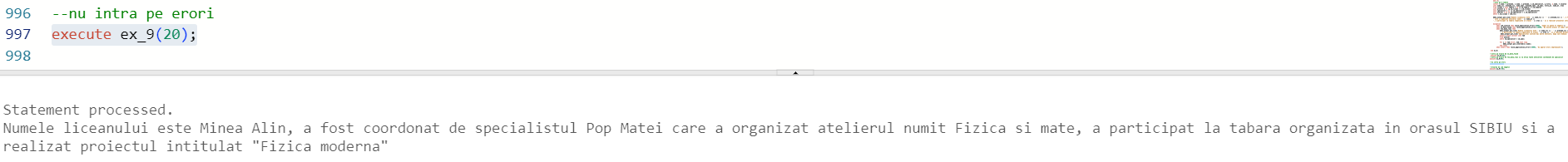
--intra pe eroare de too\_many\_rows si va afisa toate atelierele coordonate de specialist

execute ex\_9(60);



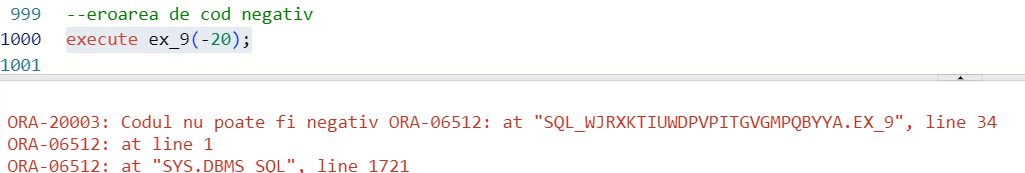
--nu intra pe erori

execute ex\_9(20);



--eroarea de cod negativ

execute ex\_9(-20);



## 



Definiți un trigger care să se declanșeze pentru a preveni adăugarea unei tabere noi în baza de date atât timp cât încă există o tabără în curs de desfășurare.

create or replace trigger ex\_10

before insert on tabara

begin

for i in (select data\_final, data\_inceput from tabara) loop

if i.data\_final > sysdate and i.data\_inceput < sysdate then raise\_application\_error(-20001, 'Nu se poate adauga o tabara daca inca exista o tabara in curs de desfasurare!');

end if;

end loop;

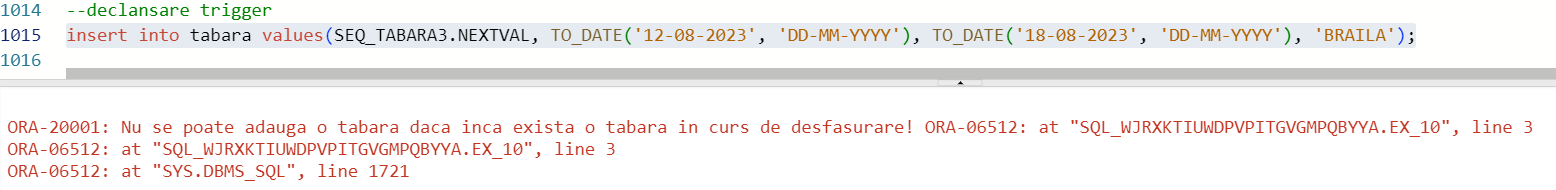
end;

--inserez o tabără care să fie în curs de desfășurare

insert into tabara values(SEQ\_TABARA3.NEXTVAL, TO\_DATE('08-01-2023', 'DD-MM-YYYY'), TO\_DATE('18-01-2023', 'DD-MM-YYYY'), 'SIBIU');

--declanșare trigger

insert into tabara values(SEQ\_TABARA3.NEXTVAL, TO\_DATE('12-08-2023', 'DD-MM-YYYY'), TO\_DATE('18-08-2023', 'DD-MM-YYYY'), 'BRAILA');





Definiți un trigger care să se declanșeze la încercarea de a adăuga sau muta un voluntar cu task-ul de a organiza un atelier care are deja minim 3 voluntari desemnați.

create table voluntar\_aux as select \* from voluntar;

create or replace trigger ex\_11

before insert or update on voluntar

for each row

declare

numar number(3);

begin

select count(\*) into numar

from voluntar\_aux where id\_atelier = :new.id\_atelier;

if numar + 1 > 3

then raise\_application\_error(-20000, 'Un atelier nu poate fi organizat de mai mult de 3 voluntari');

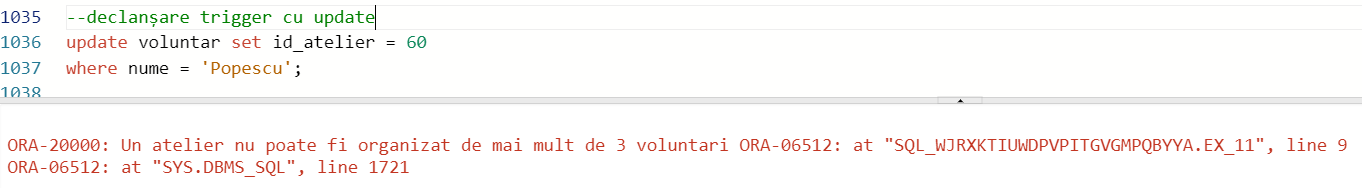
end if;

end;

--declanșare trigger cu update

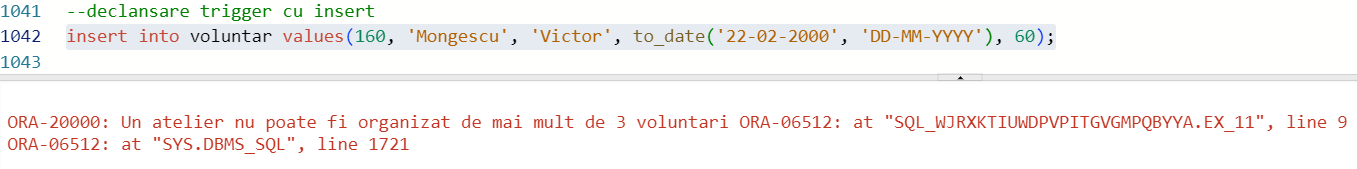
update voluntar set id\_atelier = 60

where nume = 'Popescu';



--declansare trigger cu insert

insert into voluntar values(160, 'Mongescu', 'Victor', to\_date('22-02-2000', 'DD-MM-YYYY'), 60);



**12. Trigger de tip LDD**

Creați un trigger care să populeze un tabel de erori cu date privind user-ul care a generat eroarea, numele bazei de date, eroarea în sine și data generării.

create table eroriBD

(user\_id varchar2(50),

nume\_bd varchar2(50),

erori varchar2(200),

data date);

create or replace trigger ex12

after servererror on schema

begin

insert into eroriBD values (SYS.LOGIN\_USER, SYS.DATABASE\_NAME, DBMS\_UTILITY.FORMAT\_ERROR\_STACK, SYSDATE);

end;

**13.**

create or replace package ex\_13 as

procedure ex\_6(cod companie.id\_companie%type);

procedure ex\_7(class licean.clasa%type);

function ex\_8(cod tabara.id\_tabara%type)

return number;

procedure ex\_9(cod\_lic licean.id\_licean%type);

end ex\_13;

create or replace package body ex\_13 as

procedure ex\_6(cod companie.id\_companie%type)

is

type tablou\_indexat is

table of sponsorizeaza%rowtype

index by binary\_integer;

t tablou\_indexat;

type vector\_sume is varray(100) of number;

sume vector\_sume := vector\_sume();

type vector\_locatie is varray(100) of varchar2(300);

locatii vector\_locatie :=vector\_locatie();

sponsor companie.nume%type;

total number(7) := 0;

j number(3) := 1;

nume\_oras tabara.locatie%type;

begin

select nume into sponsor

from companie

where id\_companie = cod;

select \* bulk collect into t

from sponsorizeaza;

for i in t.first..t.last loop

if t(i).id\_companie = cod then

sume.extend();

sume(j) := t(i).suma;

total := total + t(i).suma;

locatii.extend();

locatii(j) := t(i).id\_tabara;

j := j + 1;

end if;

end loop;

dbms\_output.put\_line('Numele companiei este ' || sponsor);

dbms\_output.put\_line('Suma totala acordata in sponsorizari a fost de ' || total || ' dupa cum urmeaza');

for i in locatii.first..locatii.last loop

select locatie into nume\_oras

from tabara

where id\_tabara = locatii(i);

dbms\_output.put\_line('Suma de ' || sume(i) || ' pentru tabara din ' || nume\_oras || ' cu id-ul de ' || locatii(i));

end loop;

exception

when no\_data\_found then dbms\_output.put\_line('Nu exista compania cu codul introdus');

end ex\_6;

procedure ex\_7(class licean.clasa%type)

is

cursor c(class licean.clasa%type)

is select id\_licean, nume, prenume from

licean where clasa = class;

cod licean.id\_licean%type;

nume\_lic licean.nume%type;

prenume\_lic licean.prenume%type;

nume\_atelier atelier.nume%type;

spec specialist.id\_specialist%type;

nume\_s licean.nume%type;

prenume\_s licean.prenume%type;

maxim number(2) := 0;

nume\_max licean.nume%type;

prenume\_max licean.prenume%type;

begin

open c(class);

loop

fetch c into cod, nume\_lic, prenume\_lic;

exit when c%notfound;

dbms\_output.put\_line('NUME ELEV: ' || nume\_lic || ' ' || prenume\_lic);

dbms\_output.put\_line('Ateliere la care a participat: ');

for i in (with tabel as (select id\_licean lic, count(\*) numar from se\_inscrie group by id\_licean) select id\_atelier id, numar nr from se\_inscrie, tabel where id\_licean = cod and lic = cod) loop

select nume into nume\_atelier

from atelier

where id\_atelier = i.id;

select id\_specialist into spec

from atelier where id\_atelier = i.id;

select nume, prenume into nume\_s, prenume\_s

from specialist where id\_specialist = spec;

dbms\_output.put\_line(nume\_atelier || ' coordonat de ' || nume\_s || ' ' || prenume\_s);

if i.nr > maxim then maxim := i.nr;

nume\_max := nume\_lic;

prenume\_max := prenume\_lic;

end if;

end loop;

end loop;

close c;

if cod is null then dbms\_output.put\_line('Nu exista clasa data ca parametru');

else

dbms\_output.put\_line('NUMELE ELEVULUI INSCRIS LA CELE MAI MULTE ATELIERE: ' || nume\_max || ' ' || prenume\_max);

end if;

end ex\_7;

function ex\_8(cod tabara.id\_tabara%type)

return number is

cod\_invalid exception;

cod\_notfound exception;

suma\_atinsa exception;

nume\_comp companie.nume%type;

suma\_max sponsorizeaza.suma%type := 0;

type v\_cod is varray(100) of number;

v v\_cod := v\_cod();

idx number(3) := 1;

gasit number(1) := 0;

begin

if cod <= 0 then raise cod\_invalid;

end if;

for j in (select id\_tabara from tabara) loop

v.extend();

v(idx) := j.id\_tabara;

idx := idx + 1;

end loop;

for i in v.first..v.last loop

if v(i) = cod then gasit := 1;

end if;

end loop;

if gasit = 0 then raise cod\_notfound;

end if;

for i in (select max(s.suma) sum , c.nume name from tabara t join sponsorizeaza s using (id\_tabara)

join companie c using (id\_companie)

where id\_tabara = cod

group by c.nume) loop

suma\_max := suma\_max + i.sum;

dbms\_output.put\_line('Suma: ' || i.sum || ' de la ' || i.name);

end loop;

if suma\_max >= 50000 then raise suma\_atinsa;

end if;

return 50000-suma\_max;

exception

when cod\_invalid then raise\_application\_error(-20001, 'Codul dat nu poate fi negativ!');

when suma\_atinsa then dbms\_output.put\_line('Tabara a atins deja pragul dorit de sponsorizari!');

return 0;

when cod\_notfound then raise\_application\_error(-20002,'Codul nu a fost gasit in tabela!');

end ex\_8;

procedure ex\_9(cod\_lic licean.id\_licean%type)

is

nume\_lic licean.nume%type;

prenume\_lic licean.prenume%type;

nume\_spec specialist.nume%type;

prenume\_spec specialist.prenume%type;

id\_spec specialist.id\_specialist%type;

titlu\_pr proiect.titlu%type;

--cod\_lic licean.id\_licean%type;

nume\_at atelier.nume%type;

oras tabara.locatie%type;

TYPE tabel IS TABLE OF atelier%rowtype INDEX BY PLS\_INTEGER;

tab tabel;

cod\_invalid exception;

begin

if cod\_lic <= 0 then raise cod\_invalid;

end if;

--join pe 5 tabele

select l.nume, l.prenume, s.nume, s.prenume, s.id\_specialist, p.titlu, a.nume, t.locatie

into nume\_lic, prenume\_lic, nume\_spec, prenume\_spec, id\_spec, titlu\_pr, nume\_at, oras

from licean l join tabara t on (l.id\_tabara = t.id\_tabara)

join proiect p on (p.id\_licean = l.id\_licean)

join specialist s on (s.id\_specialist = l.id\_specialist)

join atelier a on (a.id\_specialist = s.id\_specialist)

where l.id\_licean = cod\_lic;

dbms\_output.put\_line('Numele liceanului este ' || nume\_lic || ' ' || prenume\_lic || ', a fost coordonat de specialistul ' || nume\_spec || ' ' ||prenume\_spec ||

' care a organizat atelierul numit ' || nume\_at ||

', a participat la tabara organizata in orasul ' || oras || ' si a realizat proiectul intitulat ' || '"' || titlu\_pr || '"');

exception

when cod\_invalid then raise\_application\_error(-20003, 'Codul nu poate fi negativ');

when no\_data\_found then raise\_application\_error(-20000, 'Nu exista elevul cu codul introdus');

when too\_many\_rows then

dbms\_output.put\_line('Numele liceanului este ' || nume\_lic || ' ' || prenume\_lic || ', a fost coordonat de specialistul ' || nume\_spec || ' ' ||prenume\_spec ||

', a participat la tabara organizata in orasul ' || oras || ' si a realizat proiectul intitulat ' || '"' || titlu\_pr || '"');

dbms\_output.put\_line('Specialistul sustine mai multe ateliere, dupa cum urmeaza: ');

select \* bulk collect into tab

from atelier

where id\_specialist = id\_spec;

for i in tab.first..tab.last loop

dbms\_output.put\_line(tab(i).nume);

end loop;

when others then raise\_application\_error(-20002, 'Au aparut erori neprevazute');

end ex\_9;

end ex\_13;

**Apelurile pentru exercițiul 6 folosind pachetul:**

execute ex\_13.ex\_6(10);

execute ex\_13.ex\_6(91);

**Apelurile pentru exercițiul 7 folosind pachetul:**

execute ex\_13.ex\_7(10);

execute ex\_13.ex\_7(122);

**Apelurile pentru exercițiul 8 folosind pachetul:**

begin

--cazul in care functia returneaza o suma

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_13.ex\_8(20) || ' RON');

end;

begin

--functia intra pe exceptia cod\_notfound

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_13.ex\_8(900) || ' RON');

end;

begin

--cazul in care se intra pe exceptia suma\_atinsa

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_13.ex\_8(70) || ' RON');

end;

begin

--exceptia de cod\_invalid(negativ)

dbms\_output.put\_line('Taberei ii mai trebuie ' || ex\_13.ex\_8(-70) || ' RON');

end;

**Apelurile pentru exercițiul 9 folosind pachetul:**

--intra pe eroare de no\_data\_found

execute ex\_13.ex\_9(430);

--intra pe eroare de too\_many\_rows si va afisa toate atelierele coordonate de specialist

execute ex\_13.ex\_9(60);

--nu intra pe erori

execute ex\_13.ex\_9(20);

--eroarea de cod negativ

execute ex\_13.ex\_9(-20);

**14.**

Tipurile complexe de date folosite: vector de vector de int-uri (matrice de int-uri), vector de record, și record care să conțină vector.

Pachetul conține 2 funcții și 2 proceduri care se folosesc de tipurile de date declarate anterior.

Procedura voluntarSiLicean se folosește de matricea de int-uri pentru a afișa toți voluntarii care supravegheaza fiecare licean care a participat la o tabără al cărui cod este dat ca parametru.

Funcția calcul\_cheltuieli\_transport returnează suma totală care a fost decontată pe transportul elevilor care au participat la o anumită tabără cu id-ul dat ca parametru, reține date despre elevii care au beneficiat de decont într-un vector de record-uri și la final calculează care este elevul care a primit cea mai mare sumă și afișează numele lui.

Procedura sortare\_specialisti sortează toți specialișii crescător în funcție de salariu.

Funcția atelier\_cautat returnează numele atelierului la care s-au înscris cei mai mulți liceeni reține într-un record numele atelierului și un vector cu id-urile liceenilor pe baza cărora este obținut numele și prenumele elevilor pentru afișare.

create or replace package pachet\_ex14 as

function calcul\_cheltuieli\_transport(cod tabara.id\_tabara%type)

return number;

procedure voluntarSiLicean(cod tabara.id\_tabara%type);

procedure sortare\_specialisti;

function atelier\_cautat

return atelier.nume%type;

end pachet\_ex14;

---------------------------------------------------------

create or replace package body pachet\_ex14 as

type v\_coduri is varray(60) of number;

type matrix\_liceeni is varray(60) of v\_coduri;

vector\_coduri v\_coduri := v\_coduri();

matrice\_liceeni matrix\_liceeni := matrix\_liceeni();

type info\_licean is record

(

id licean.id\_licean%type,

nume licean.nume%type,

prenume licean.prenume%type,

cheltuieli licean.cheltuieli\_transport%type

);

type vect is varray(100) of info\_licean;

vect\_info\_tabara vect := vect();

type v\_vect is varray(100) of number;

coduri v\_vect := v\_vect();

type info\_at is record(

nume\_atelier atelier.nume%type,

liceeni v\_vect

);

procedure voluntarSiLicean(cod tabara.id\_tabara%type) is

type vector\_id is varray(100) of number;

j number(2) := 1;

k number(2) := 1;

vector\_id\_lic vector\_id := vector\_id();

nume\_lic licean.nume%type;

prenume\_lic licean.prenume%type;

nume\_vol licean.nume%type;

prenume\_vol licean.prenume%type;

begin

for i in (select id\_licean from licean where id\_tabara = cod) loop

vector\_id\_lic.extend();

vector\_id\_lic(j) := i.id\_licean;

j := j + 1;

end loop;

j := 1;

for i in vector\_id\_lic.first..vector\_id\_lic.last loop

matrice\_liceeni.extend();

k := 1;

matrice\_liceeni(j) := v\_coduri();

for x in (select id\_voluntar from supervizeaza where id\_licean = vector\_id\_lic(i)) loop

matrice\_liceeni(j).extend();

matrice\_liceeni(j)(k) := x.id\_voluntar;

k := k + 1;

end loop;

j := j + 1;

end loop;

for i in matrice\_liceeni.first..matrice\_liceeni.last loop

select nume, prenume into nume\_lic, prenume\_lic

from licean where id\_licean = vector\_id\_lic(i);

dbms\_output.put\_line('Liceanul ' || nume\_lic || ' ' || prenume\_lic || ' a fost supravegheat de: ');

for j in matrice\_liceeni(i).first..matrice\_liceeni(i).last loop

select nume, prenume into nume\_vol, prenume\_vol

from voluntar where id\_voluntar = matrice\_liceeni(i)(j);

dbms\_output.put\_line('Voluntarul: ' || nume\_vol || ' ' || prenume\_vol);

end loop;

end loop;

exception

when no\_data\_found then raise\_application\_error(-20001, 'Nu exista codul taberei introdus');

end voluntarSiLicean;

function calcul\_cheltuieli\_transport(cod tabara.id\_tabara%type)

return number is

j number(2) := 1;

total number(4) := 0;

maxim number(4) := 0;

nume\_max licean.nume%type;

prenume\_max licean.prenume%type;

begin

for i in (select id\_licean, nume, prenume, cheltuieli\_transport from licean where id\_tabara = cod) loop

vect\_info\_tabara.extend();

vect\_info\_tabara(j).id:= i.id\_licean;

vect\_info\_tabara(j).nume := i.nume;

vect\_info\_tabara(j).prenume := i.prenume;

vect\_info\_tabara(j).cheltuieli := i.cheltuieli\_transport;

if i.cheltuieli\_transport is not null then

total := total + i.cheltuieli\_transport;

end if;

j := j + 1;

end loop;

for i in vect\_info\_tabara.first..vect\_info\_tabara.last loop

if vect\_info\_tabara(i).cheltuieli > maxim then

maxim := vect\_info\_tabara(i).cheltuieli;

nume\_max := vect\_info\_tabara(i).nume;

prenume\_max := vect\_info\_tabara(i).prenume;

end if;

end loop;

dbms\_output.put\_line('Elevul care a primit cea mai mare decontare pe transport (' ||maxim|| ' RON) ' || 'este ' || nume\_max ||' ' || prenume\_max);

return total;

exception

when no\_data\_found then raise\_application\_error(-20001, 'Nu exista codul taberei introdus');

end calcul\_cheltuieli\_transport;

procedure sortare\_specialisti

is

type vector\_sal is varray(100) of specialist.salariu%type;

type vector\_cod is varray(100) of specialist.id\_specialist%type;

v\_sal vector\_sal := vector\_sal();

v\_cod vector\_cod := vector\_cod();

j number(2) := 1;

cursor c

is select id\_specialist, salariu from

specialist;

aux\_sal specialist.salariu%type;

aux\_cod specialist.id\_specialist%type;

nume\_s specialist.nume%type;

prenume\_s specialist.prenume%type;

begin

for i in c loop

v\_sal.extend();

v\_cod.extend();

v\_sal(j) := i.salariu;

v\_cod(j) := i.id\_specialist;

j := j + 1;

end loop;

for i in v\_sal.first..v\_sal.last loop

for j in i+1..v\_sal.last loop

if v\_sal(i) > v\_sal(j) then

aux\_sal := v\_sal(j);

v\_sal(j) := v\_sal(i);

v\_sal(i) := aux\_sal;

aux\_cod := v\_cod(j);

v\_cod(j) := v\_cod(i);

v\_cod(i) := aux\_cod;

end if;

end loop;

end loop;

dbms\_output.put\_line('Specialistii sortati crescator in functie de salariu sunt dupa cum urmeaza: ');

for i in v\_cod.first..v\_cod.last loop

select nume, prenume into nume\_s, prenume\_s

from specialist where id\_specialist = v\_cod(i);

dbms\_output.put\_line(nume\_s || ' ' ||prenume\_s || ' cu salariul ' || v\_sal(i));

end loop;

end sortare\_specialisti;

function atelier\_cautat

return atelier.nume%type is

nume\_at atelier.nume%type;

cod\_at atelier.id\_atelier%type;

numar number(2);

j number(2) := 1;

info\_atelier info\_at;

nume\_lic varchar2(100);

prenume\_lic varchar2(100);

begin

select \* into cod\_at, nume\_at, numar from (select id\_atelier, nume, count(\*) nr

from se\_inscrie join atelier using(id\_atelier)

group by id\_atelier, nume

order by nr desc)

where rownum = 1;

for i in (select id\_licean from se\_inscrie where id\_atelier = cod\_at) loop

coduri.extend();

coduri(j) := i.id\_licean;

j := j +1;

end loop;

info\_atelier.nume\_atelier := nume\_at;

info\_atelier.liceeni := coduri;

dbms\_output.put\_line(numar || ‘ participanti’);

for i in info\_atelier.liceeni.first..info\_atelier.liceeni.last loop

select nume, prenume into nume\_lic, prenume\_lic

from licean where id\_licean = info\_atelier.liceeni(i);

dbms\_output.put\_line('Elevul ' || nume\_lic || ' ' ||prenume\_lic);

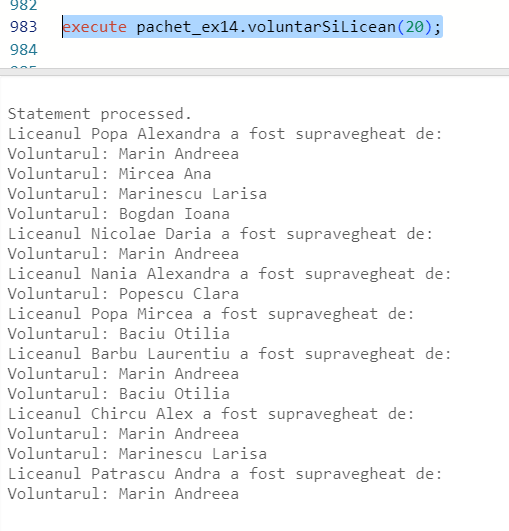
end loop;

return info\_atelier.nume\_atelier;

end atelier\_cautat;

end pachet\_ex14;

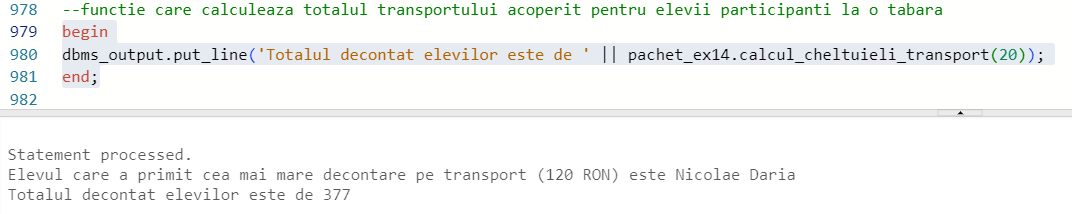
execute pachet\_ex14.voluntarSiLicean(20);



begin

dbms\_output.put\_line('Totalul decontat elevilor este de ' || pachet\_ex14.calcul\_cheltuieli\_transport(20));

end;

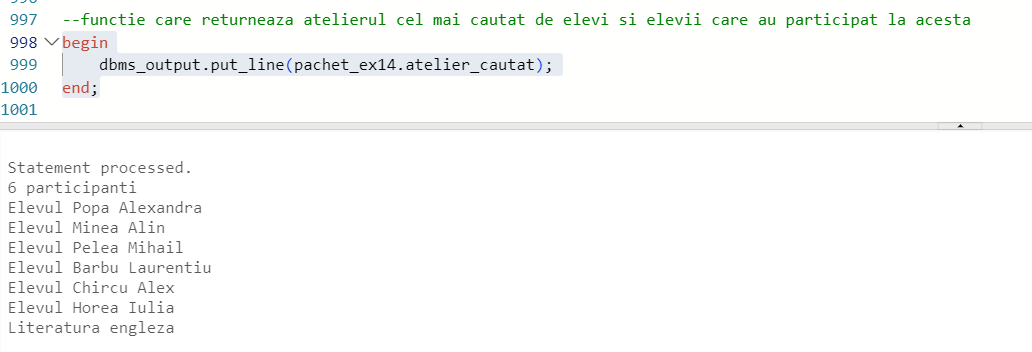


--functie care returneaza atelierul cel mai cautat de elevi si elevii care au participat la acesta

begin

dbms\_output.put\_line(pachet\_ex14.atelier\_cautat);

end;



execute pachet\_ex14.sortare\_specialisti;

