

**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.

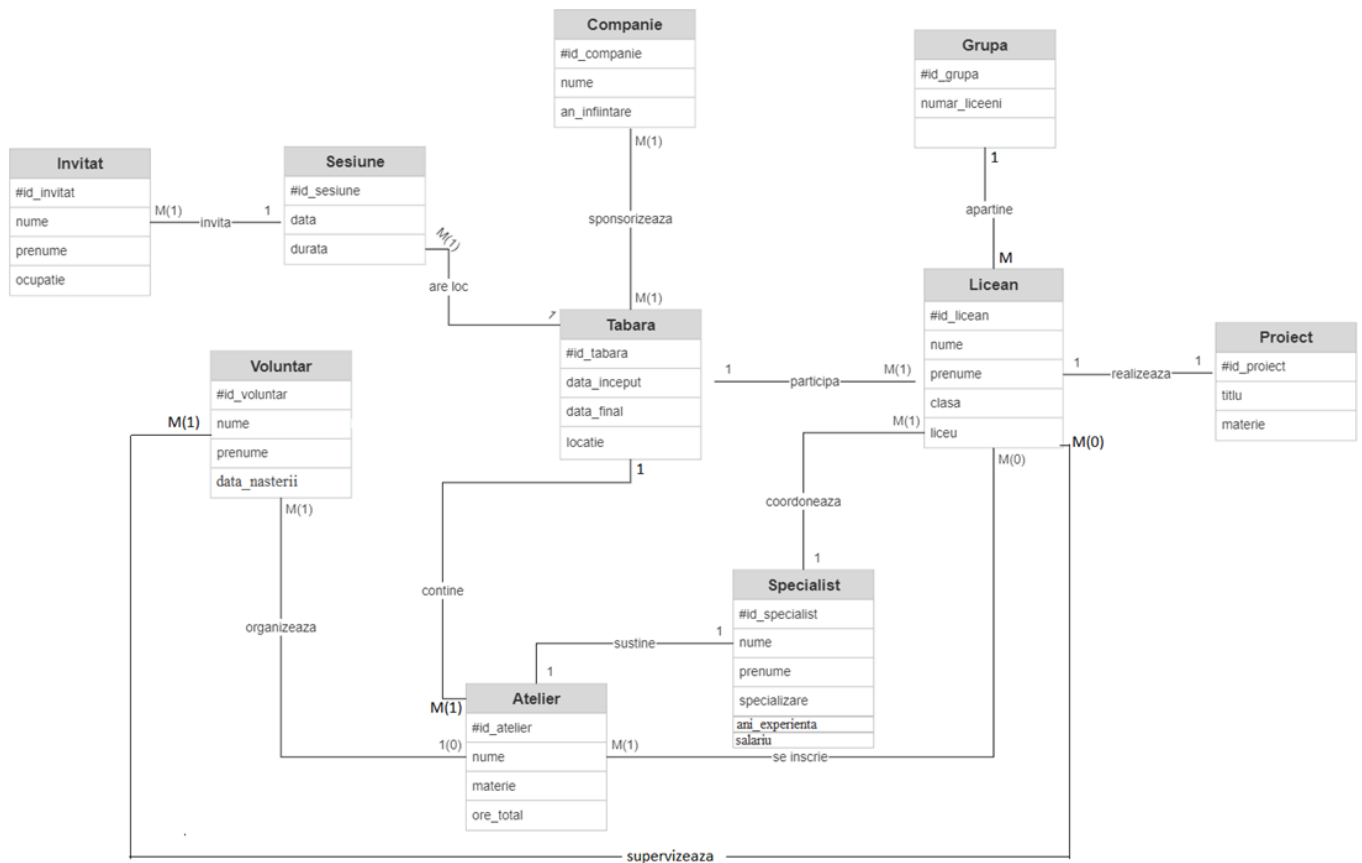
În funcție de numărul de participanți, liceeni sunt împărțiți pe grupe aproximativ egale.

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

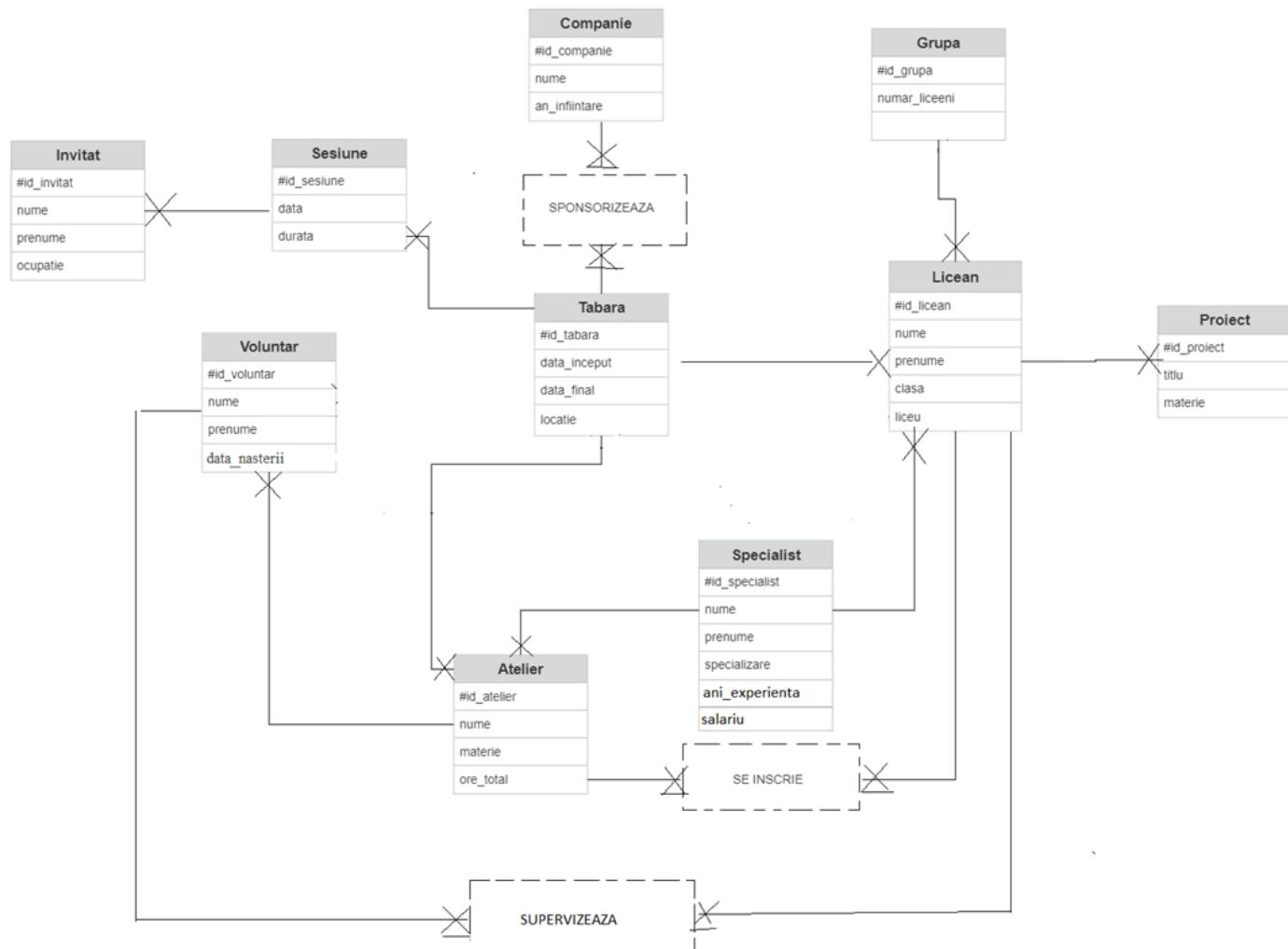
La final, fiecare elev prezintă un proiect cu o tematică specifică unui atelier la care a luat parte. Proiectele se realizează individual.

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

## 2. Diagrama Entitate-Relatie



### 3. Diagrama conceptuală



### 4.

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 SESSIUNE  
VALUES(20, TO_DATE('15-04-2019', 'DD-MM-YYYY'), 2, 30);
```

```
INSERT INTO SESSIUNE  
VALUES(30, TO_DATE('13-08-2019', 'DD-MM-YYYY'), 3, 40);
```

```
INSERT INTO SESSIUNE  
VALUES(40, TO_DATE('13-02-2020', 'DD-MM-YYYY'), 3, 50);
```

```
INSERT INTO SESSIUNE  
VALUES(50, TO_DATE('17-09-2020', 'DD-MM-YYYY'), 3, 60);
```

```
INSERT INTO SESSIUNE  
VALUES(60, TO_DATE('04-03-2018', 'DD-MM-YYYY'), 3, 70);
```

```
INSERT INTO SESSIUNE  
VALUES(70, TO_DATE('07-08-2019', 'DD-MM-YYYY'), 3, 80);
```

```
INSERT INTO SESSIUNE  
VALUES(80, TO_DATE('06-11-2017', 'DD-MM-YYYY'), 3, 90);
```

```
INSERT INTO SESSIUNE  
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);
```

**5.**

```
select * from companie;
```

ID_COMPANIE	NUME	AN_INFIINTARE
10	AQUACARPATICA.SRL	2002
20	SUPERBET.SRL	2004
30	EUROSTAR STUDIOS.SRL	2002
40	KAUFLAND.SRL	1999
50	FLOWERS DESIGN.SRL	2009

Select \* from invitat;

ID_INVITAT	NUME	PRENUME	OCUPATIE	ID_SESIUNE
10	Anghel	Sorana	antreprenor	10
20	Angelescu	Bogdan	atlet	20
30	Milu	Alexandru	atlet	30
40	Barbu	Simona	psiholog	40
50	Antonescu	Alina	politician	50
60	Mircescu	Vlad	antreprenor	60
70	Manea	Bogdan	antreprenor	70
80	Begu	Briana	psiholog	80
100	Deaconu	Razvan	filantrop	70



Select \* from licean;

ID_LICEAN	NUME	PRENUME	CLASA	LICEU	ID_TABARA	ID_GRUPA	ID_SPECIALIST	CHELTUIELI_TRANSPORT
10	Popa	Alexandra	10	Colegiul National Unirea	20	20	10	50
20	Minea	Alin	9	CN Mihai Eminescu	30	30	20	50
30	Pelea	Mihail	12	CN Ion Luca Caragiale	40	20	10	125
40	Manea	Sandra	10	CN A.I. Cuza	50	30	40	97
50	Banciu	Maria	11	CN Tehnologic Buzau	60	40	50	100
60	Vasiliu	Catalin	10	CN Pedagogic Zalau	70	40	50	70
70	Neacsu	Teodora	12	Liceul Teoretic Nr.1 Vaslui	80	20	60	70
80	Velea	Miruna	9	CN Marin Preda	90	30	10	-
100	Nicolae	Daria	9	CN Constantin Noica	20	20	20	120
110	Nania	Alexandra	10	CN Tehnologic Nr.2 Braila	20	40	10	-
120	Popa	Mircea	11	Liceul Teoretic Nr.4 Bucuresti	20	50	10	97
130	Barbu	Laurentiu	12	CN Pedagogic Alexandria	20	50	30	-
140	Chircu	Alex	12	CN Stefan Cel Mare	20	50	30	-
150	Minea	Valeria	12	CN Marin Preda	80	40	30	70
160	Badea	Marina	11	Liceul Teoretic Nr.2 Braila	80	40	20	-
180	Horea	Iulia	10	CN Stefan Cel Mare	90	20	40	-
190	Bordei	Isabela	10	CN Al. I. Cuza	60	50	50	30
200	Patrascu	Andra	12	Liceul Tehnologic Nr.2 Iasi	20	50	20	110

Select \* from atelier;

ID_ATELIER	NUME	ORE_TOTAL	MATERIE	ID_TABARA	ID_SPECIALIST
10	Bazele biologiei	12	biologie	30	30
20	Literatura engleza	12	engleza	20	50
30	Matematica avansata	12	matematica	30	60
40	Algoritmi de baza	12	informatica	40	40
50	Bazele biologiei II	10	biologie	30	10
60	Biologie moleculara	12	biologie moleculara	50	10
70	Structuri de date	14	informatica	60	40
80	Engleza avansata	9	engleza	70	50
90	Fizica aplicata	12	fizica	80	60
100	Fizica si mate	14	fizica	90	20

Select \* from voluntar;

ID_VOLUNTAR	NUME	PRENUME	DATA_NASTERII	ID_ATELIER
10	Popescu	Clara	26-NOV-02	30
20	Marin	Andreea	14-SEP-00	40
30	Mircea	Ana	06-APR-01	50
40	Marinescu	Larisa	24-JAN-02	60
50	Ciurea	Bianca	02-AUG-01	10
60	Bogdan	Ioana	03-SEP-01	20
70	Baciu	Otilia	22-APR-00	30
80	Vladici	Sara	02-APR-00	10
90	Bob	Alice	21-JUN-00	30
100	Bran	Alexandru	12-SEP-01	20
110	Costea	Mihail	20-DEC-00	60
120	Corolescu	Catalin	29-JAN-02	60
130	Mihai	Violeta	28-AUG-00	50
140	Manea	Tudor	17-MAY-00	30
150	Sorescu	Teodor	13-OCT-01	20

Select \* from proiect;

ID_PROIECT	TITLU	MATERIE	ID_LICEAN
10	Fizica de azi	fizica	100
20	Mate aplicata	matematica	30
30	Biologie aprofundata	biologie moleculara	40
40	Probleme informatica	informatica	50
50	Literatura engleza	engleza	60
60	Aplicatii	matematica	10
70	Fizica moderna	fizica	20
80	Sub microscop	biologie moleculara	70
90	Arta cuvintelor	engleza	80
100	Fizica altfel	fizica	110
120	Probleme reale	matematica	120
130	O noua abordare	informatica	130
140	Aplicatii utile	informatica	140
150	Teoria informatiei	informatica	150
160	English Tenses	engleza	160
180	Multimea nr reale	matematica	180
190	Regnul animal	biologie	190
200	Legile lui Newton	fizica	200

Select \* from tabara;

ID_TABARA	DATA_INCEPUT	DATA_FINAL	LOCATIE
10	05-NOV-18	09-NOV-18	ARAD
20	13-APR-19	18-APR-19	BUCURESTI
30	12-AUG-19	18-AUG-19	SIBIU
40	10-FEB-20	14-FEB-20	CLUJ-NAPOCA
50	11-SEP-20	18-SEP-20	BUCURESTI
60	04-MAR-18	09-MAR-18	TIMISOARA
70	05-AUG-19	09-AUG-19	IASI
80	05-NOV-17	09-NOV-17	GALATI
90	02-JUN-17	08-JUN-17	IASI

Select \* from grupa;

ID_GRUPA	NUMAR_LICEENI
10	3
20	5
30	5
40	4
50	5

Select \* from sesiune;

ID_SESIUNE	DATA_SESIUNE	DURATA	ID_TABARA
10	06-NOV-18	3	20
20	15-APR-19	2	30
30	13-AUG-19	3	40
40	13-FEB-20	3	50
50	17-SEP-20	3	60
60	04-MAR-18	3	70
70	07-AUG-19	3	80
80	06-NOV-17	3	90

Select \* from sponsorizeaza;

ID_COMPANIE	ID_TABARA	SUMA
20	20	15000
10	20	12000
20	30	15000
20	70	20000
30	20	5000
40	40	15000
50	50	70000
20	60	15000
10	70	65000
30	80	24500
20	90	15000
10	90	29000
40	80	17000

Select \* from specialist;

ID_SPECIALIST	NUME	PRENUME	SPECIALIZARE	ANI_EXPERIENTA	SALARIU
10	Pop	Alexandra	biologie moleculara	12	1800
20	Pop	Matei	matematica	12	3000
30	Popa	Ion	biologie	8	2500
40	Munteanu	Diana	informatica	15	3300
50	Anton	Maria	engleza	8	2000
60	Bejan	Andrei	matematica	14	3250



Select \* from se\_inscire;

ID_LICEAN	ID_ATELIER	DATA_INSCRIERE
10	10	12-MAR-18
20	10	13-MAR-18
30	20	12-SEP-18
40	30	02-MAR-19
50	40	12-NOV-19
60	50	12-MAR-20
70	60	04-JUN-20
80	10	12-MAR-18
100	40	30-NOV-19
110	50	11-JUN-20
120	60	17-JUN-20
130	20	09-SEP-18
140	20	17-SEP-18
140	10	10-MAR-18
110	40	12-NOV-19
150	50	12-JUN-20
160	10	11-MAR-18
180	20	16-SEP-18
190	60	12-JUN-20
200	60	10-JUN-20
10	20	10-SEP-18
20	20	11-SEP-18

Select \* from supervizeaza;

ID_VOLUNTAR	ID_LICEAN	NR_DE_ORE
10	20	8
20	10	3
30	10	2
40	30	5
50	40	5
60	10	1
70	50	4
70	60	4
40	20	3
30	30	5
20	100	3
70	120	5
10	110	6
20	130	3
40	140	7
50	70	3
20	80	3
70	130	5
20	140	3
20	150	2
40	160	3
50	180	4
10	190	3
20	200	1
20	20	1
40	10	4
20	30	1

6. 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:

```
809         where id_tabara = locatii(i);
810
811         dbms_output.put_line('Suma de ' || sume(i) || ' pentru
812     end loop;
813
814     exception
815     when no_data_found then dbms_output.put_line('Nu exista compania cu codul
816
817 end;
818
819 execute ex_6(670);
820
```

---

```
Statement processed.
Nu exista compania cu codul introdus
```

execute ex\_6(20);

```
813  
814 ✓      exception  
815         when no_data_found then dbms_output.put_line('Nu exista cor  
816  
817 end;  
818  
819 execute ex_6(670);  
820 execute ex_6(20);  
821
```

```
Statement processed.  
Numele companiei este SUPERBET.SRL  
Suma totala acordata in sponsorizari a fost de 80000 dupa cum urmeaza  
Suma de 15000 pentru tabara din BUCURESTI cu id-ul de 20  
Suma de 15000 pentru tabara din SIBIU cu id-ul de 30  
Suma de 20000 pentru tabara din IASI cu id-ul de 70  
Suma de 15000 pentru tabara din TIMISOARA cu id-ul de 60  
Suma de 15000 pentru tabara din IASI cu id-ul de 90
```

7. Pentru liceenii a căror clasă este dată ca parametru, să se afișeze lista cu numele lor, precum și lista cu toate atelierelor 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, după 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 clase = 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);

```
867     end loop;
868
869     close c;
870     if cod is null then dbms_output.put_line('Nu exista clasa data
871     else
872         dbms_output.put_line('NUMELE ELEVULUI INSCRIS LA CELE MAI M
873     end if;
874
875 end;
876
877 execute ex_7(9);
878
879
880
```

Statement processed.  
NUME ELEV: Minea Alin  
Ateliere la care a participat:  
Bazele biologiei coordonat de Popa Ion  
Literatura engleza coordonat de Anton Maria  
NUME ELEV: Velea Miruna  
Ateliere la care a participat:  
Bazele biologiei coordonat de Popa Ion  
NUME ELEV: Nicolae Daria  
Ateliere la care a participat:  
Algoritmi de baza coordonat de Munteanu Diana  
NUMELE ELEVULUI INSCRIS LA CELE MAI MULTE ATELIERE: Minea Alin

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

```
871     else
872         dbms_output.put_line('NUMELE ELEVULUI INSCRIS LA
873     end if;
874
875 end;
876
877 execute ex_7(44);
878
```

Statement processed.  
Nu exista clasa data ca parametru

8. 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;

```

```

945 begin
946     --functia intra pe exceptia cod_notfound
947     dbms_output.put_line('Taberei ii mai trebuie ' || ex_8(900) || ' RON');
948 end;
949

```

```

ORA-20002: Codul nu a fost gasit in tabela! ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_8", line 52
ORA-06512: at line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721

```

```

begin

    --cazul in care functia returneaza o suma

    dbms_output.put_line('Taberei ii mai trebuie ' || ex_8(20) || ' RON');

end;

```

```

942
943 ✓begin
944     --cazul in care functia returneaza o suma
945     dbms_output.put_line('Taberei ii mai trebuie ' || ex_8(20) || ' RON');
946 end;
947

```

---

Statement processed.  
Suma: 15000 de la SUPERBET.SRL  
Suma: 12000 de la AQUACARPATICA.SRL  
Suma: 5000 de la EUROSTAR STUDIOS.SRL  
Taberei ii mai trebuie 18000 RON

begin

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

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

end;

```

942 ✓begin
943     --cazul in care se intra pe exceptia suma_atinsa
944     dbms_output.put_line('Taberei ii mai trebuie ' || ex_8(70) || ' RON');
945 end;
946
947

```

---

Statement processed.  
Suma: 20000 de la SUPERBET.SRL  
Suma: 65000 de la AQUACARPATICA.SRL  
Tabara a atins deja pragul dorit de sponsorizari!  
Taberei ii mai trebuie 0 RON

begin

--exceptia de cod\_invalid(negativ)

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

end;

```

935
936 ✓ begin
937     --exceptia de cod_invalid(negativ)
938     dbms_output.put_line('Taberei ii mai trebuie ' || ex_8(-70) || ' RON');
939 end;
940

```

ORA-20001: Codul dat nu poate fi negativ! ORA-06512: at "SQL\_WJRXKTIUWDPVPITGVGPQBYA.EX\_8", line 48  
 ORA-06512: at line 3  
 ORA-06512: at "SYS.DBMS\_SQL", line 1721

9. 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 lican.id_licean%type)
is

```

```

    nume_lic lican.nume%type;
    prenume_lic lican.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 lican.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 lican 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);

```

```

991 ✓ --intra pe eroare de no_data_found

```

```

992   execute ex_9(430);

```

```

993   --intra pe eroare de too many rows si va afisa toate atelierele coordonate de specialist

```

```

ORA-20000: Nu exista elevul cu codul introdus ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_9", line 35

```

```

ORA-06512: at line 1

```

```

ORA-06512: at "SYS.DBMS_SQL", line 1721

```

```

--intra pe eroare de too_many_rows si va afisa toate atelierele coordonate de specialist
execute ex_9(60);

```

```

993 --intra pe eroare de too_many_rows si va afisa toate atelierele coordonate de specialist
994 execute ex_9(60);
995

```

Statement processed.

Numele liceanului este Vasiliu Catalin, a fost coordonat de specialistul Anton Maria, a participat la tabara organizata in orasul IASI si a realizat proiectul intitulat "Literatura engleza"

Specialistul sustine mai multe ateliere, dupa cum urmeaza:

Literatura engleza

Engleza avansata

```

--nu intra pe erori
execute ex_9(20);

```

```

996 --nu intra pe erori
997 execute ex_9(20);
998

```

Statement processed.

Numele liceanului este Minea Alin, a fost coordonat de specialistul Pop Matei care a organizat atelierul numit Fizica si mate, a participat la tabara organizata in orasul SIBIU si a realizat proiectul intitulat "Fizica moderna"

```

--eroarea de cod negativ
execute ex_9(-20);

```

```

999 --eroarea de cod negativ
1000 execute ex_9(-20);
1001

```

```

ORA-20003: Codul nu poate fi negativ ORA-06512: at "SQL_WJRXTIUWDPVPITGVGMPQBYA.EX_9", line 34
ORA-06512: at line 1
ORA-06512: at "SYS.DBMS SQL", line 1721

```

## 10.

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');

```
1014 --declansare trigger
1015 insert into tabara values(SEQ_TABARA3.NEXTVAL, TO_DATE('12-08-2023', 'DD-MM-YYYY'), TO_DATE('18-08-2023', 'DD-MM-YYYY'), 'BRAILA');
1016
```

```
ORA-20001: Nu se poate adauga o tabara daca inca exista o tabara in curs de desfasurare! ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_10", line 3
ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_10", line 3
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

## 11.

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';
```

```
1035 --declanșare trigger cu update  
1036 update voluntar set id_atelier = 60  
1037 where nume = 'Popescu';  
1038
```

```
ORA-20000: Un atelier nu poate fi organizat de mai mult de 3 voluntari ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_11", line 9  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

```
--declansare trigger cu insert
```

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

```
1041 --declansare trigger cu insert  
1042 insert into voluntar values(160, 'Mongescu', 'Victor', to_date('22-02-2000', 'DD-MM-YYYY'), 60);  
1043
```

```
ORA-20000: Un atelier nu poate fi organizat de mai mult de 3 voluntari ORA-06512: at "SQL_WJRXKTIUWDPVPITGVGMPQBYA.EX_11", line 9  
ORA-06512: at "SYS.DBMS_SQL", line 1721
```

## 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;

Declansare trigger:

Select \* from adofswdf;

update dvfdf set ceva = 'afv';

select \* from eroriBD;

USER_ID	NUME_BD	ERORI
1 GRUPA244 011G		ORA-00942: tabelul sau vizualizarea nu exista
2 GRUPA244 011G		ORA-00942: tabelul sau vizualizarea nu exista
3 GRUPA244 011G		ORA-20005: ORU-10034: context argument must be 1 or 2 or 3 or 4 or 5 or 6 or 7 or 8 or 9ORA-06512: la "SYS.DBMS_UTILITY", linia 114ORA-06512: la

### 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 specialiștii 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.ume%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,
        ume licean.ume%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(
        ume_atelier atelier.ume%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 || ' participant');
    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);

```

982

983 `execute pachet_ex14.voluntarSiLicean(20);`

984

985

```

Statement processed.
Liceanul Popa Alexandra a fost supravegheat de:
Voluntarul: Marin Andreea
Voluntarul: Mircea Ana
Voluntarul: Marinescu Larisa
Voluntarul: Bogdan Ioana
Liceanul Nicolae Daria a fost supravegheat de:
Voluntarul: Marin Andreea
Liceanul Nania Alexandra a fost supravegheat de:
Voluntarul: Popescu Clara
Liceanul Popa Mircea a fost supravegheat de:
Voluntarul: Baci Otilia
Liceanul Barbu Laurentiu a fost supravegheat de:
Voluntarul: Marin Andreea
Voluntarul: Baci Otilia
Liceanul Chircu Alex a fost supravegheat de:
Voluntarul: Marin Andreea
Voluntarul: Marinescu Larisa
Liceanul Patrascu Andra a fost supravegheat de:
Voluntarul: Marin Andreea

```

begin

```
        dbms_output.put_line('Totalul decontat elevilor este de ' ||  
pachet_ex14.calcul_cheltuieli_transport(20));
```

end;

```
978 --functie care calculeaza totalul transportului acoperit pentru elevii participanti la o tabara  
979 begin  
980 dbms_output.put_line('Totalul decontat elevilor este de ' || pachet_ex14.calcul_cheltuieli_transport(20));  
981 end;  
982
```

```
Statement processed.  
Elevul care a primit cea mai mare decontare pe transport (120 RON) este Nicolae Daria  
Totalul decontat elevilor este de 377
```

--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;

```
997 --functie care returneaza atelierul cel mai cautat de elevi si elevii care au participat la acesta  
998 begin  
999     dbms_output.put_line(pachet_ex14.atelier_cautat);  
1000 end;  
1001
```

```
Statement processed.  
6 participanti  
Elevul Popa Alexandra  
Elevul Minea Alin  
Elevul Pelea Mihail  
Elevul Barbu Laurentiu  
Elevul Chircu Alex  
Elevul Horea Iulia  
Literatura engleza
```

```
execute pachet_ex14.sortare_specialisti;
```

```
1000      end;
```

```
1001
```

```
1002     execute pachet_ex14.sortare_specialisti;
```

```
1003
```

Statement processed.

Specialistii sortati crescator in functie de salariu sunt dupa cum urmeaza:

Pop Alexandra cu salariul 1800

Anton Maria cu salariul 2000

Popa Ion cu salariul 2500

Pop Matei cu salariul 3000

Bejan Andrei cu salariul 3250

Munteanu Diana cu salariul 3300

Erorile de NO\_DATA\_FOUND:

```
execute pachet_ex14.voluntarSiLicean(11);
```

```
988
```

```
989     execute pachet_ex14.voluntarSiLicean(11);
```

```
990
```

ORA-20001: Nu exista codul taberei introdus ORA-06512: at "SQL\_LPBTCXGGTMAOTNEVBFBKBJZHL.PACHET\_EX14", line 74

ORA-06512: at line 1

ORA-06512: at "SYS.DBMS\_SQL", line 1721

```
begin
```

```
dbms_output.put_line('Totalul decontat elevilor este de ' ||
```

```
pachet_ex14.calcul_cheltuieli_transport(27));
```

```
end;
```

```
700
```

```
984     --functie care calculeaza totalul transportului acoperit pentru elevii participanti la o tabara
```

```
985     begin
```

```
986     dbms_output.put_line('Totalul decontat elevilor este de ' || pachet_ex14.calcul_cheltuieli_transport(27));
```

```
987     end;
```

ORA-20001: Nu exista codul taberei introdus ORA-06512: at "SQL\_LPBTCXGGTMAOTNEVBFBKBJZHL.PACHET\_EX14", line 111

ORA-06512: at line 2

ORA-06512: at "SYS.DBMS\_SQL", line 1721