PROIECT SGBD – Gestiunea unei scoli

Barbu Alexandru-Marian

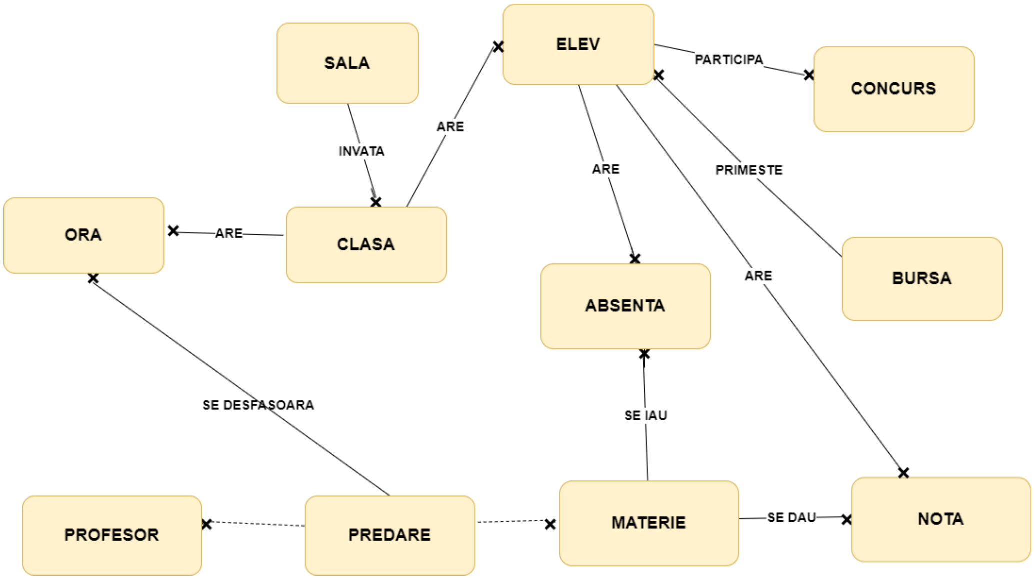
Grupa 344 (laborator sustinut cu 243)

Ex. 1

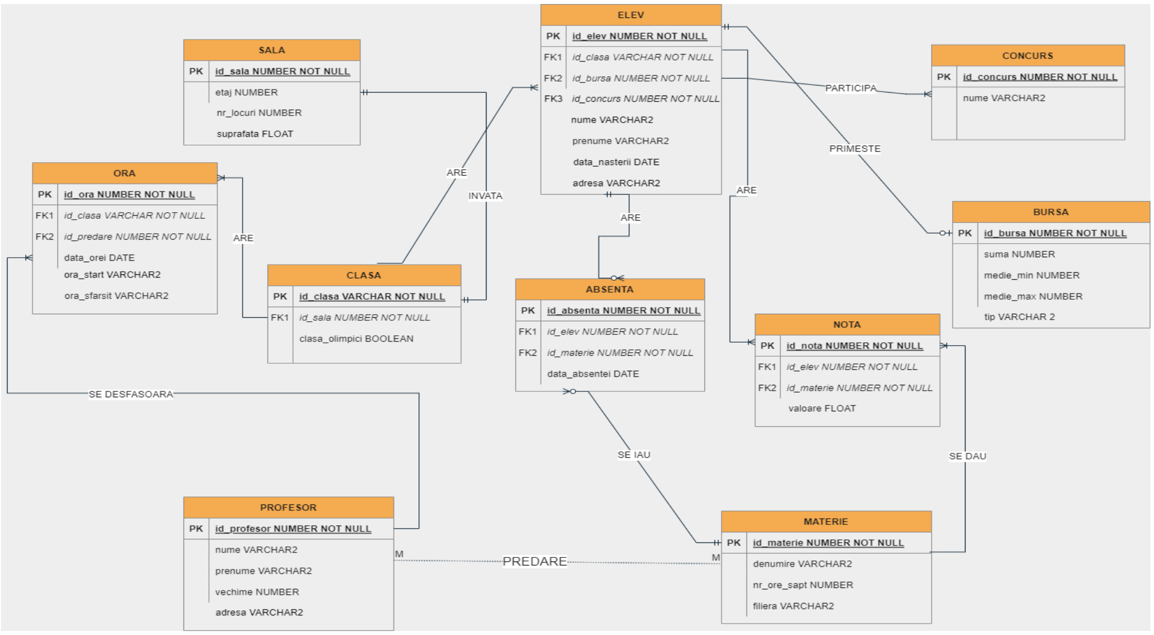
Scopul conceperii acestei baze de date este de a facilita mentinerea unei evidente intr-o unitate de invatamant. Aceasta contine un numar de 11 entitati: **SALA, CLASA, ELEV, NOTA, ABSENTA, MATERIE, PROFESOR, ORE, CONCURS, BURSA, PREDARE** carora le corespund niste atribute sugestive. De asemenea, entitatile sunt legate cu ajutorul unor relatii, cat mai diverse, si anume relatii 1:1, relatii 1-to-many, dar si relatii many-to-many. Datorita celor enuntate mai devreme, avand niste relatii mai complexe, a fost nevoie sa implementez si tabele asociative cum ar fi **PREDARE.**

Aceasta baza de date o consider utila in conceperea orarului, stiind ca acesta este un lucru care pune des probleme conducerii scolilor, liceelor, universitatilor etc. Pe langa asta, am incercat ca implementarea ei sa fie una cat mai simplista, cat mai usor de inteles, cat mai modulara cu putinta. Si nu in ultimul rand, desigur ca si evitarea redundantelor si adaugarea unor constrangeri cat mai folositoare in practica au fost alte lucruri pe care le-am avut in vedere.

Ex. 2



Ex. 3


SALA
PK id_sala NUMBER NOT NULL
etaj NUMBER
nr_locuri NUMBER
suprafata FLOAT
ELEV
PK id_elev NUMBER NOT NULL
FK1 id_clasa VARCHAR NOT NULL
FK2 id_bursa NUMBER NOT NULL
FK3 id_concurs NUMBER NOT NULL
nume VARCHAR2
prenume VARCHAR2
data_nasterii DATE
adresa VARCHAR2
CONCURS
PK id_concurs NUMBER NOT NULL
nume VARCHAR2
PARTICIPA
PRIMESTE
ORA
PK id_ora NUMBER NOT NULL
FK1 id_clasa VARCHAR NOT NULL
FK2 id_predare NUMBER NOT NULL
data_orei DATE
ora_start VARCHAR2
ora_sfarsit VARCHAR2
ARE
INVATA
ARE
CLASA
PK id_clasa VARCHAR NOT NULL
FK1 id_sala NUMBER NOT NULL
clasa_olimpici BOOLEAN
ARE
ARE
ABSENTA
PK id_absenta NUMBER NOT NULL
FK1 id_elev NUMBER NOT NULL
FK2 id_materie NUMBER NOT NULL
data_absentei DATE
BURSA
PK id_bursa NUMBER NOT NULL
suma NUMBER
medie_min NUMBER
medie_max NUMBER
tip VARCHAR 2
NOTA
PK id_nota NUMBER NOT NULL
FK1 id_elev NUMBER NOT NULL
FK2 id_materie NUMBER NOT NULL
valoare FLOAT
SE DESFASOARA
SE IAU
SE DAU
PROFESOR
PK id_profesor NUMBER NOT NULL
nume VARCHAR2
prenume VARCHAR2
vechime NUMBER
adresa VARCHAR2
M
PREDARE
M
MATERIE
PK id_materie NUMBER NOT NULL
denumire VARCHAR2
nr_ore_sapt NUMBER
filiera VARCHAR2

Ex. 4

```
CREATE TABLE SALA (
   ID_SALA NUMBER(3,0) NOT NULL,
   ETAJ VARCHAR2(100),
   NR_LOCURI NUMBER(3,0),
   SUPRAFATA FLOAT,
   CONSTRAINT SALA_PK PRIMARY KEY (ID_SALA)
);


CREATE TABLE CONCURS (
   ID_CONCURS NUMBER(3,0) NOT NULL,
   NUME VARCHAR2(100),
   CONSTRAINT CONCURS_PK PRIMARY KEY (ID_CONCURS)
);
```

```
CREATE TABLE MATERIE (
    ID_MATERIE NUMBER(3,0) NOT NULL,
    DENUMIRE VARCHAR2(100),
    NR_ORE_SAPT NUMBER(3,0),
    FILIERA VARCHAR2(100),
    CONSTRAINT MATERIE_PK PRIMARY KEY (ID_MATERIE)
);

 CREATE TABLE BURSA (
    ID_BURSA NUMBER(3,0) NOT NULL,
    SUMA NUMBER(5,0),
    MEDIE_MIN FLOAT,
    MEDIE_MAX FLOAT,
    TIP VARCHAR2(100),
    CONSTRAINT BURSA_PK PRIMARY KEY (ID_BURSA)
);

CREATE TABLE PROFESOR (
    ID_PROFESOR NUMBER(3,0) NOT NULL,
    NUME VARCHAR2(100),
    PRENUME VARCHAR2(100),
    VECHIME NUMBER(2,0),
    ADRESA VARCHAR2(100),
    CONSTRAINT PROFESOR_PK PRIMARY KEY (ID_PROFESOR)
);

CREATE TABLE PREDARE (
    ID_PREDARE NUMBER(3,0) NOT NULL,
    ID_PROFESOR NUMBER(3,0) NOT NULL,
    ID_MATERIE NUMBER(3,0) NOT NULL,
    CONSTRAINT PREDARE_PK PRIMARY KEY (ID_PREDARE),
    CONSTRAINT FK_PROFESOR FOREIGN KEY (ID_PROFESOR) REFERENCES PROFESOR(ID_PROFESOR),
    CONSTRAINT FK_MATERIE FOREIGN KEY (ID_MATERIE) REFERENCES MATERIE(ID_MATERIE)
);

CREATE TABLE CLASA (
    ID_CLASA VARCHAR2(5) NOT NULL,
    ID_SALA NUMBER(3,0) NOT NULL,
    CLASA_OLIMPICI NUMBER(1),
    CONSTRAINT CLASA_PK PRIMARY KEY (ID_CLASA),
    CONSTRAINT FK_SALA FOREIGN KEY (ID_SALA) REFERENCES SALA(ID_SALA)
);

CREATE TABLE ELEV (
    ID_ELEV NUMBER(3,0) NOT NULL,
    ID_CLASA VARCHAR2(5) NOT NULL,
    ID_BURSA NUMBER(3,0),
    ID_CONCURS NUMBER(5,0),
    NUME VARCHAR2(100),
    PRENUME VARCHAR2(100),
    DATA_NASTERII DATE,
    ADRESA VARCHAR2(100),
    CONSTRAINT ELEV_PK PRIMARY KEY (ID_ELEV),
    CONSTRAINT FK_CLASA FOREIGN KEY (ID_CLASA) REFERENCES CLASA(ID_CLASA),
    CONSTRAINT FK_BURSA FOREIGN KEY (ID_BURSA) REFERENCES BURSA(ID_BURSA),
    CONSTRAINT FK_CONCURS FOREIGN KEY (ID_CONCURS) REFERENCES CONCURS(ID_CONCURS)
);
```

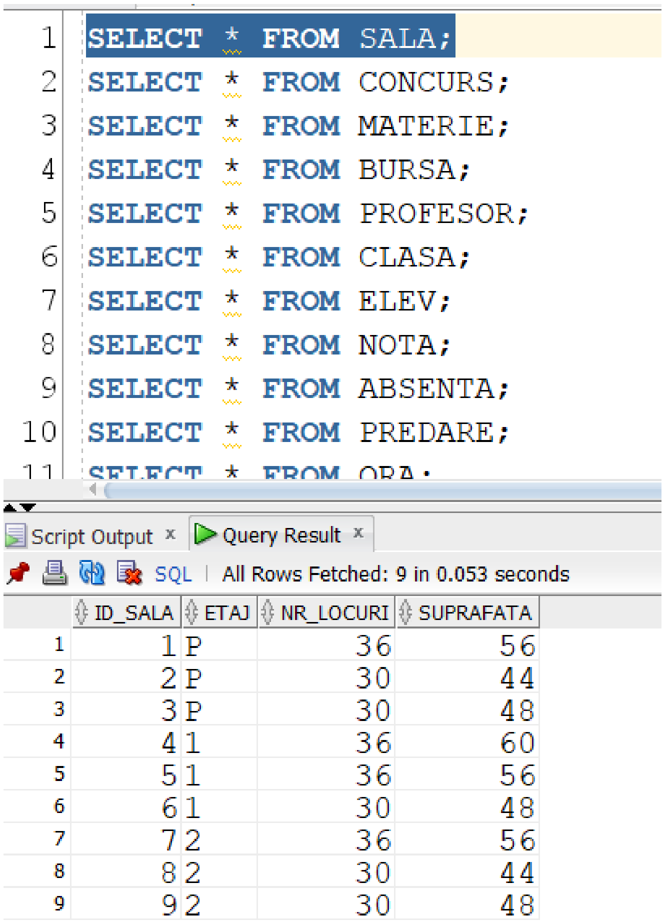
```
CREATE TABLE NOTA (
   ID_NOTA NUMBER(3,0) NOT NULL,
   ID_ELEV NUMBER(3,0) NOT NULL,
   ID_MATERIE NUMBER(3,0) NOT NULL,
   VALOARE FLOAT,
   CONSTRAINT NOTA_PK PRIMARY KEY (ID_NOTA),
   CONSTRAINT FK_ELEV FOREIGN KEY (ID_ELEV) REFERENCES ELEV(ID_ELEV),
   CONSTRAINT FK_MATERIE2 FOREIGN KEY (ID_MATERIE) REFERENCES MATERIE(ID_MATERIE)
);

CREATE TABLE ABSENTA (
   ID_ABSENTA NUMBER(3,0) NOT NULL,
   ID_ELEV NUMBER(3,0) NOT NULL,
   ID_MATERIE NUMBER(3,0) NOT NULL,
   DATA_ABSENTEI DATE,
   CONSTRAINT ABSENTA_PK PRIMARY KEY (ID_ABSENTA),
   CONSTRAINT FK_ELEV2 FOREIGN KEY (ID_ELEV) REFERENCES ELEV(ID_ELEV),
   CONSTRAINT FK_MATERIE3 FOREIGN KEY (ID_MATERIE) REFERENCES MATERIE(ID_MATERIE)
);

CREATE TABLE ORA (
   ID_ORA NUMBER(3,0) NOT NULL,
   ID_CLASA VARCHAR2(5) NOT NULL,
   ID_PREDARE NUMBER(3,0) NOT NULL,
   DATA_OREI DATE,
   ORA_START VARCHAR2(5),
   ORA_SFARSIT VARCHAR2(5),
   CONSTRAINT ORA_PK PRIMARY KEY (ID_ORA),
   CONSTRAINT FK_CLASA2 FOREIGN KEY (ID_CLASA) REFERENCES CLASA(ID_CLASA),
   CONSTRAINT FK_PREDARE FOREIGN KEY (ID_PREDARE) REFERENCES PREDARE(ID_PREDARE)
);
```

Ex. 5

```
INSERT INTO SALA VALUES (1, 'P', 36, 56);
INSERT INTO SALA VALUES (2, 'P', 30, 44);
INSERT INTO SALA VALUES (3, 'P', 30, 48);
INSERT INTO SALA VALUES (4, '1', 36, 60);
INSERT INTO SALA VALUES (5, '1', 36, 56);
INSERT INTO SALA VALUES (6, '1', 30, 48);
INSERT INTO SALA VALUES (7, '2', 36, 56);
INSERT INTO SALA VALUES (8, '2', 30, 44);
INSERT INTO SALA VALUES (9, '2', 30, 48);

INSERT INTO CONCURS VALUES (1, 'Euclid');
INSERT INTO CONCURS VALUES (2, 'Arhimede');
INSERT INTO CONCURS VALUES (3, 'Terra');
INSERT INTO CONCURS VALUES (4, 'Smart');
INSERT INTO CONCURS VALUES (5, 'Winners');
INSERT INTO CONCURS VALUES (6, 'Cangurul');

INSERT INTO MATERIE VALUES (1, 'Matematica', 6, 'Real');
INSERT INTO MATERIE VALUES (2, 'Lb. Romana', 6, 'Uman');
INSERT INTO MATERIE VALUES (3, 'Informatica', 4, 'Real');
INSERT INTO MATERIE VALUES (4, 'Istorie', 1, 'Uman');
INSERT INTO MATERIE VALUES (5, 'Geografie', 1, 'Uman');
INSERT INTO MATERIE VALUES (6, 'Fizica', 3, 'Real');
INSERT INTO MATERIE VALUES (7, 'Chimie', 3, 'Real');
INSERT INTO MATERIE VALUES (8, 'Biologie', 1, 'Real');
INSERT INTO MATERIE VALUES (9, 'Sociologie', 1, 'Uman');
INSERT INTO MATERIE VALUES (10, 'Sport', 2, 'Vocational');

INSERT INTO BURSA VALUES(1, 2500, 10, 10, 'Excelenta');
INSERT INTO BURSA VALUES(2, 2000, 9.75, 10, 'Performanta');
INSERT INTO BURSA VALUES(3, 1750, 9.5, 9.75, 'Merit I');
INSERT INTO BURSA VALUES(4, 1500, 9.25, 9.5, 'Merit II');
INSERT INTO BURSA VALUES(5, 1250, 9, 9.25, 'Merit III');
INSERT INTO BURSA VALUES(6, 1400, NULL, NULL, 'Sociala');


INSERT INTO PROFESOR VALUES(1, 'Stan', 'Bogdan', 10, 'Soseaua Colentina 1');
INSERT INTO PROFESOR VALUES(2, 'Stefanescu', 'Aureliana', 15, 'Sos.Morarilor, Nr.1');
INSERT INTO PROFESOR VALUES(3, 'Filip', 'Pavel', 7, 'Bld. Pache Protopopescu, str. Traian, nr. 169');
INSERT INTO PROFESOR VALUES(4, 'Boboc', 'Andreea', 9, 'Bulevardul Timisoara 33');
INSERT INTO PROFESOR VALUES(5, 'Tataru', 'Antonia', 15, 'Bulevardul Iuliu Maniu 51');
INSERT INTO PROFESOR VALUES(6, 'Neagu', 'Sebastian', 18, 'Bd. Theodor Pallady, Nr. 27');
INSERT INTO PROFESOR VALUES(7, 'Toader', 'Stefania', 25, 'Aleea Alexandru 43');
INSERT INTO PROFESOR VALUES(8, 'Neagu', 'Cristian', 15, 'Bd. Ramnicu Valcea, Nr. 13');
INSERT INTO PROFESOR VALUES(9, 'Puiu', 'Gheorghe', 32, 'Bulevardul Camil Ressu 39');
INSERT INTO PROFESOR VALUES(10, 'Ionescu', 'Maria', 7, 'Sos Oltenitei, Nr 5');
INSERT INTO PROFESOR VALUES(11, 'Burlacu', 'Petru', 12, 'Bulevardul Corneliu Coposu 7');
INSERT INTO PROFESOR VALUES(12, 'Nidelea', 'Eugenia', 8, 'Strada Liviu Rebreanu 45');
INSERT INTO PROFESOR VALUES(13, 'Chiriac', 'Marta', 17, 'Baba Novac, nr 9, Complex Rucar');
INSERT INTO PROFESOR VALUES(14, 'Panagachie', 'Elena', 22, 'Strada Postavarului 12');
INSERT INTO PROFESOR VALUES(15, 'Albu', 'Angela', 10, 'Strada General Stefan Holban 7');
INSERT INTO PROFESOR VALUES(16, 'Pavel', 'Traian', 10, 'Strada Pomarla B21');
```

```sql
INSERT INTO PROFESOR VALUES(17, 'Corvin', 'Serban', 5, 'Strada Sepcari 77');
INSERT INTO PROFESOR VALUES(18, 'Anton', 'Paula', 17, 'Soseaua Garii Catelu 5');
INSERT INTO PROFESOR VALUES(19, 'Dinca', 'Corina', 18, 'Strada Poiana Narciselor 33');
INSERT INTO PROFESOR VALUES(20, 'Cozma', 'Ariana', 5, 'Calea Dorobantilor 4');

INSERT INTO PREDARE VALUES(1, 1, 1);
INSERT INTO PREDARE VALUES(2, 2, 1);
INSERT INTO PREDARE VALUES(3, 3, 1);
INSERT INTO PREDARE VALUES(4, 4, 1);
INSERT INTO PREDARE VALUES(5, 5, 2);
INSERT INTO PREDARE VALUES(6, 6, 2);
INSERT INTO PREDARE VALUES(7, 7, 2);
INSERT INTO PREDARE VALUES(8, 8, 3);
INSERT INTO PREDARE VALUES(9, 8, 3);
INSERT INTO PREDARE VALUES(10, 10, 4);
INSERT INTO PREDARE VALUES(11, 11, 4);
INSERT INTO PREDARE VALUES(12, 12, 5);
INSERT INTO PREDARE VALUES(13, 13, 5);
INSERT INTO PREDARE VALUES(14, 14, 6);
INSERT INTO PREDARE VALUES(15, 15, 6);
INSERT INTO PREDARE VALUES(16, 16, 7);
INSERT INTO PREDARE VALUES(17, 17, 7);
INSERT INTO PREDARE VALUES(18, 18, 8);
INSERT INTO PREDARE VALUES(18, 18, 8);
INSERT INTO PREDARE VALUES(19, 19, 9);
INSERT INTO PREDARE VALUES(20, 20, 10);

INSERT INTO CLASA VALUES('9A', 1, 1);
INSERT INTO CLASA VALUES('9B', 2, 0);
INSERT INTO CLASA VALUES('10A', 3, 0);
INSERT INTO CLASA VALUES('10B', 4, 0);
INSERT INTO CLASA VALUES('11A', 5, 0);
INSERT INTO CLASA VALUES('11B', 6, 0);
INSERT INTO CLASA VALUES('12A', 7, 0);
INSERT INTO CLASA VALUES('12B', 8, 0);


INSERT INTO ELEV VALUES(1, '9A', 1, 1, 'Ionescu', 'Tudor', '02-SEP-07', 'Bulevardul Tineretului 91');
INSERT INTO ELEV VALUES(2, '9A', 2, 3, 'Grigore', 'Marius', '12-MAR-07', 'Strada Barbu Vacarescu 201');
INSERT INTO ELEV VALUES(3, '9A', 1, 4, 'Niculae', 'George', '15-MAY-07', 'Strada I.C. Visarion 9');
INSERT INTO ELEV VALUES(4, '9A', 2, 5, 'Popescu', 'Andreea', '18-AUG-07', 'Bulevardul Energeticienilor 9');
INSERT INTO ELEV VALUES(5, '9A', 2, 1, 'Ion', 'Eugen', '30-APR-07', 'Intrarea Tudor Stefan 44');
INSERT INTO ELEV VALUES(6, '9B', 6, NULL, 'Trandafir', 'Alina', '14-SEP-07', 'Strada Sibiu 5A');
INSERT INTO ELEV VALUES(7, '9B', 5, 2, 'Vintur', 'Silviu', '04-MAY-07', 'Calea Dorobantilor 140');
INSERT INTO ELEV VALUES(8, '9B', 3, 2, 'Tristan', 'Elena', '25-MAY-07', 'Splaiul Unirii 176');
INSERT INTO ELEV VALUES(9, '9B', 4, 5, 'Ivanovici', 'Gigel', '15-APR-07', 'Strada Constantin Disescu 14');
INSERT INTO ELEV VALUES(10, '9B', NULL, NULL, 'Grigoras', 'Cristina', '24-OCT-07', 'Calea Vacaresti 391');
INSERT INTO ELEV VALUES(11, '10A', 4, 5, 'Dijmarescu', 'Antonio', '02-JAN-06', 'Strada Olteni 10');
INSERT INTO ELEV VALUES(12, '10A', 4, NULL, 'Micsunel', 'Teodor', '13-FEB-06', 'Strada Pictor Arthur Verona 13-15');
INSERT INTO ELEV VALUES(13, '10A', 2, NULL, 'Podani', 'Mihai', '14-DEC-06', 'Bulevardul Dimitrie Cantemir 19');
INSERT INTO ELEV VALUES(14, '10A', 1, 6, 'Niculescu', 'Maria', '13-JUL-06', 'Strada Teodor Stefanescu 17');
INSERT INTO ELEV VALUES(15, '10A', 2, 2, 'Matei', 'Georgiana', '8-MAY-06', 'Bulevardul Iuliu Maniu 484');
INSERT INTO ELEV VALUES(16, '10B', 3, 6, 'Grigoroiu', 'Mirela', '7-JUN-06', 'Strada Mircea Vulcanescu 88');
```

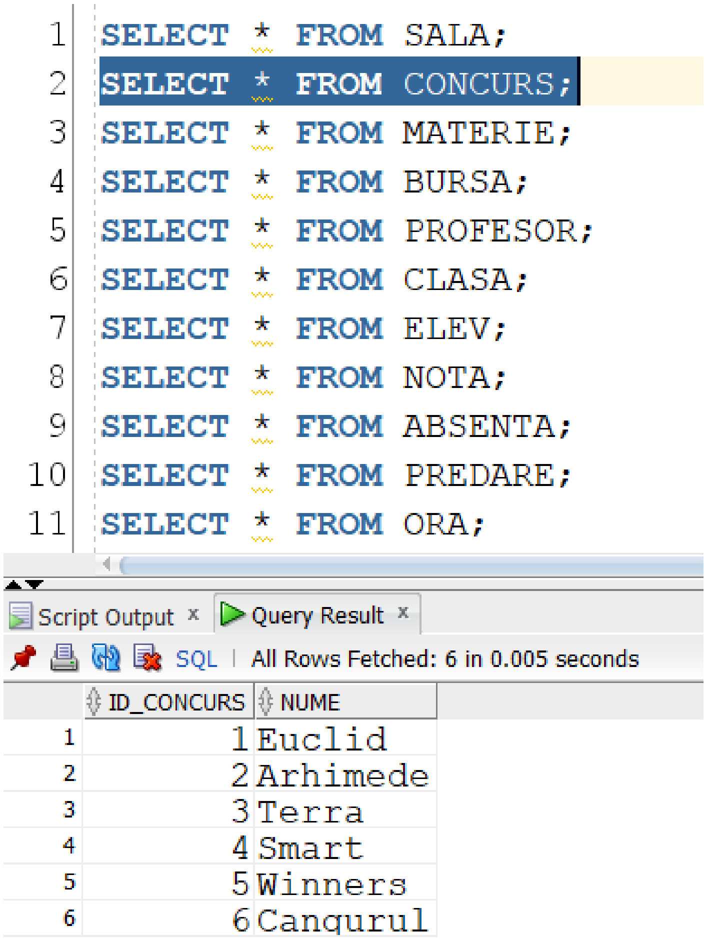
```sql
INSERT INTO ELEV VALUES(17, '10B', 3, 4, 'Barbulescu', 'Gabriela', '27-DEC-06', 'Strada Izvor 2-4');

INSERT INTO ELEV VALUES(18, '10B', 3, 2, 'Barascu', 'Cosmina', '23-JUL-06', 'Splaiul Independentei 210');

INSERT INTO ELEV VALUES(19, '10B', 5, 2, 'Dumitrescu', 'Ilinca', '15-MAR-06', 'Str. Armand Calinescu, nr. 25');

INSERT INTO ELEV VALUES(20, '10B', 4, 5, 'Georgescu', 'Marius', '10-AUG-06', 'Strada Mantuleasa 31');

INSERT INTO ELEV VALUES(21, '11A', 1, 1, 'Nistor', 'Mihaela', '25-JUN-05', 'Intrarea Tarcau 13');

INSERT INTO ELEV VALUES(22, '11A', 2, 3, 'Radu', 'Mihaela', '14-JUN-05', 'Calea Crangasi 29');

INSERT INTO ELEV VALUES(23, '11A', 2, 6, 'Ionel', 'Tudor', '24-JUL-05', 'Calea Giulesti 1-3');

INSERT INTO ELEV VALUES(24, '11A', 4, 5, 'Barbu', 'Daniel', '25-NOV-05', 'Bulevardul Marasesti 4-6');

INSERT INTO ELEV VALUES(25, '11A', 2, 4, 'Alexandrescu', 'Paula', '5-SEP-05', 'Calea Rahovei 266A');

INSERT INTO ELEV VALUES(26, '11B', NULL, 4, 'Dobre', 'Delia', '15-NOV-05', 'Bulevardul Gheorghe Sincai 2');

INSERT INTO ELEV VALUES(27, '11B', 3, NULL, 'Lomota', 'Marian', '26-MAY-05', 'Strada Bihor 70');

INSERT INTO ELEV VALUES(28, '11B', 1, 1, 'Mihalcea', 'Ana', '4-MAY-05', 'Strada Preciziei 24');

INSERT INTO ELEV VALUES(29, '11B', 4, 1, 'Stanescu', 'Maria', '14-APR-05', 'Capitan Juverdeanu 30');

INSERT INTO ELEV VALUES(30, '11B', 5, 3, 'Belu', 'Lucian', '17-MAR-05', 'Bulevardul 1 Decembrie 1918 33A');

INSERT INTO ELEV VALUES(31, '12A', 1, 6, 'Chiriac', 'Alexandru', '24-MAY-04', 'Strada Soimus 33');

INSERT INTO ELEV VALUES(32, '12A', 4, 4, 'Mihailescu', 'Luciana', '20-FEB-04', 'Strada Maica Domnului 61');

INSERT INTO ELEV VALUES(33, '12A', NULL, NULL, 'Chirila', 'Lucica', '10-SEP-04', 'Bulevardul Pache Protopopescu 109');

INSERT INTO ELEV VALUES(34, '12A', 2, NULL, 'Grigorescu', 'Nicusor', '1-OCT-04', 'Str. Apusului, Nr. 27');

INSERT INTO ELEV VALUES(35, '12A', 1, 3, 'Tudor', 'Liviu', '22-JAN-04', 'Strada Bilciuresti 4');

INSERT INTO ELEV VALUES(36, '12B', NULL, NULL, 'Paun', 'Petre', '10-OCT-04', 'Strada Dimitrie Bolintineanu 9');

INSERT INTO ELEV VALUES(37, '12B', NULL, 2, 'Nicolescu', 'Catalina', '24-DEC-04', 'Strada Traian 11');

INSERT INTO ELEV VALUES(38, '12B', NULL, NULL, 'Turcu', 'Roxana', '3-APR-04', 'Soseaua Nordului 7-9');

INSERT INTO ELEV VALUES(39, '12B', NULL, NULL, 'Mihai', 'Iulian', '7-MAY-04', 'Strada Occidentului 25');

INSERT INTO ELEV VALUES(40, '12B', NULL, NULL, 'Naipeanu', 'Costin', '11-JAN-04', 'Strada Grigore Alexandrescu 7');
INSERT INTO NOTA VALUES(1, 1, 2, 10);
INSERT INTO NOTA VALUES(2, 1, 4, 10);
INSERT INTO NOTA VALUES(3, 3, 1, 10);
INSERT INTO NOTA VALUES(4, 3, 6, 10);
INSERT INTO NOTA VALUES(5, 4, 7, 10);
INSERT INTO NOTA VALUES(6, 4, 5, 9);
INSERT INTO NOTA VALUES(7, 5, 3, 8);
INSERT INTO NOTA VALUES(8, 5, 1, 10);
INSERT INTO NOTA VALUES(9, 6, 2, 6);
INSERT INTO NOTA VALUES(10, 7, 6, 10);
INSERT INTO NOTA VALUES(11, 7, 2, 8);
INSERT INTO NOTA VALUES(12, 7, 1, 9);
INSERT INTO NOTA VALUES(13, 8, 5, 7);
INSERT INTO NOTA VALUES(14, 8, 1, 7);
INSERT INTO NOTA VALUES(15, 9, 10, 10);
INSERT INTO NOTA VALUES(16, 9, 9, 9);
INSERT INTO NOTA VALUES(17, 10, 4, 5);
INSERT INTO NOTA VALUES(18, 10, 2, 10);
INSERT INTO NOTA VALUES(19, 11, 2, 8);
INSERT INTO NOTA VALUES(20, 12, 1, 5);
INSERT INTO NOTA VALUES(21, 13, 1, 5);
INSERT INTO NOTA VALUES(22, 14, 1, 10);
INSERT INTO NOTA VALUES(23, 14, 3, 10);
INSERT INTO NOTA VALUES(24, 14, 6, 10);
INSERT INTO NOTA VALUES(25, 15, 1, 8);
INSERT INTO NOTA VALUES(26, 16, 1, 9);
INSERT INTO NOTA VALUES(27, 18, 2, 7);
INSERT INTO NOTA VALUES(28, 20, 3, 4);
INSERT INTO NOTA VALUES(29, 21, 2, 5);
INSERT INTO NOTA VALUES(30, 21, 1, 10);
INSERT INTO NOTA VALUES(31, 22, 6, 9);
INSERT INTO NOTA VALUES(32, 23, 7, 9);
INSERT INTO NOTA VALUES(33, 25, 2, 8);
INSERT INTO NOTA VALUES(34, 26, 1, 2);
INSERT INTO NOTA VALUES(35, 26, 2, 3);
INSERT INTO NOTA VALUES(36, 27, 4, 9);
INSERT INTO NOTA VALUES(37, 27, 1, 6);
INSERT INTO NOTA VALUES(38, 28, 2, 10);
INSERT INTO NOTA VALUES(39, 28, 1, 5);
INSERT INTO NOTA VALUES(40, 30, 9, 10);


INSERT INTO ABSENTA VALUES(1, 1, 1, '5-OCT-22');
INSERT INTO ABSENTA VALUES(2, 7, 2, '7-OCT-22');
INSERT INTO ABSENTA VALUES(3, 12, 6, '8-OCT-22');
INSERT INTO ABSENTA VALUES(4, 16, 7, '9-OCT-22');
INSERT INTO ABSENTA VALUES(5, 21, 8, '10-OCT-22');
INSERT INTO ABSENTA VALUES(6, 22, 9, '11-OCT-22');
INSERT INTO ABSENTA VALUES(7, 25, 9, '14-OCT-22');
INSERT INTO ABSENTA VALUES(8, 27, 9, '16-OCT-22');
INSERT INTO ABSENTA VALUES(9, 36, 5, '17-OCT-22');
INSERT INTO ABSENTA VALUES(10, 37, 3, '20-OCT-22');


INSERT INTO ORA VALUES(1, '9A', 1, '05-OCT-22', '09:00', '10:00');
INSERT INTO ORA VALUES(2, '9A', 2, '06-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(3, '9B', 3, '07-OCT-22', '11:00', '12:00');
INSERT INTO ORA VALUES(4, '9A', 4, '07-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(5, '10A', 5, '08-OCT-22', '09:00', '10:00');
INSERT INTO ORA VALUES(6, '9B', 6, '09-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(7, '10B', 7, '09-OCT-22', '11:00', '12:00');
INSERT INTO ORA VALUES(8, '10B', 8, '10-OCT-22', '09:00', '10:00');
INSERT INTO ORA VALUES(9, '11A', 9, '10-OCT-22', '11:00', '12:00');
INSERT INTO ORA VALUES(10, '11A', 10, '11-OCT-22', '12:00', '13:00');
INSERT INTO ORA VALUES(11, '10A', 11, '12-OCT-22', '09:00', '10:00');
INSERT INTO ORA VALUES(12, '9A', 12, '12-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(13, '9B', 13, '12-OCT-22', '11:00', '12:00');
INSERT INTO ORA VALUES(14, '11A', 14, '13-OCT-22', '08:00', '09:00');
INSERT INTO ORA VALUES(15, '11A', 15, '13-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(16, '11A', 16, '14-OCT-22', '11:00', '12:00');
INSERT INTO ORA VALUES(17, '11B', 17, '14-OCT-22', '12:00', '13:00');
INSERT INTO ORA VALUES(18, '11B', 18, '15-OCT-22', '09:00', '10:00');
INSERT INTO ORA VALUES(19, '12A', 19, '15-OCT-22', '10:00', '11:00');
INSERT INTO ORA VALUES(20, '12B', 20, '16-OCT-22', '09:00', '10:00');
```



```
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 9 in 0.053 seconds

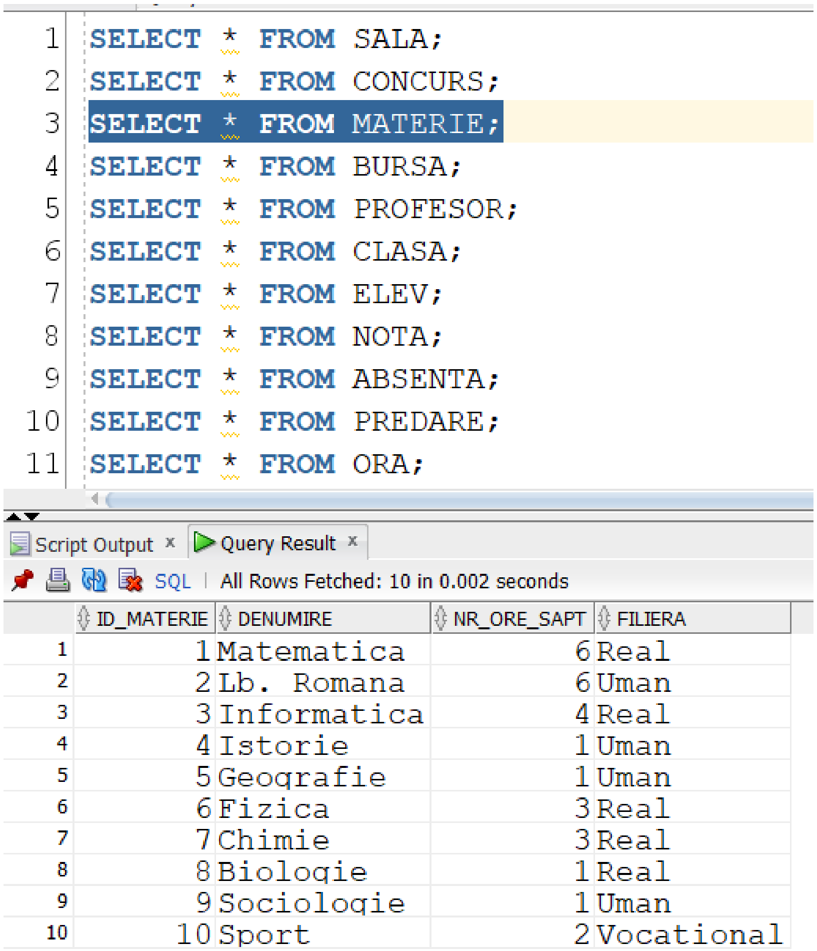
	ID_SALA	ETAJ	NR_LOCURI	SUPRAFATA
1	1	P	36	56
2	2	P	30	44
3	3	P	30	48
4	4	1	36	60
5	5	1	36	56
6	6	1	30	48
7	7	2	36	56
8	8	2	30	44
9	9	2	30	48



```
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 6 in 0.005 seconds

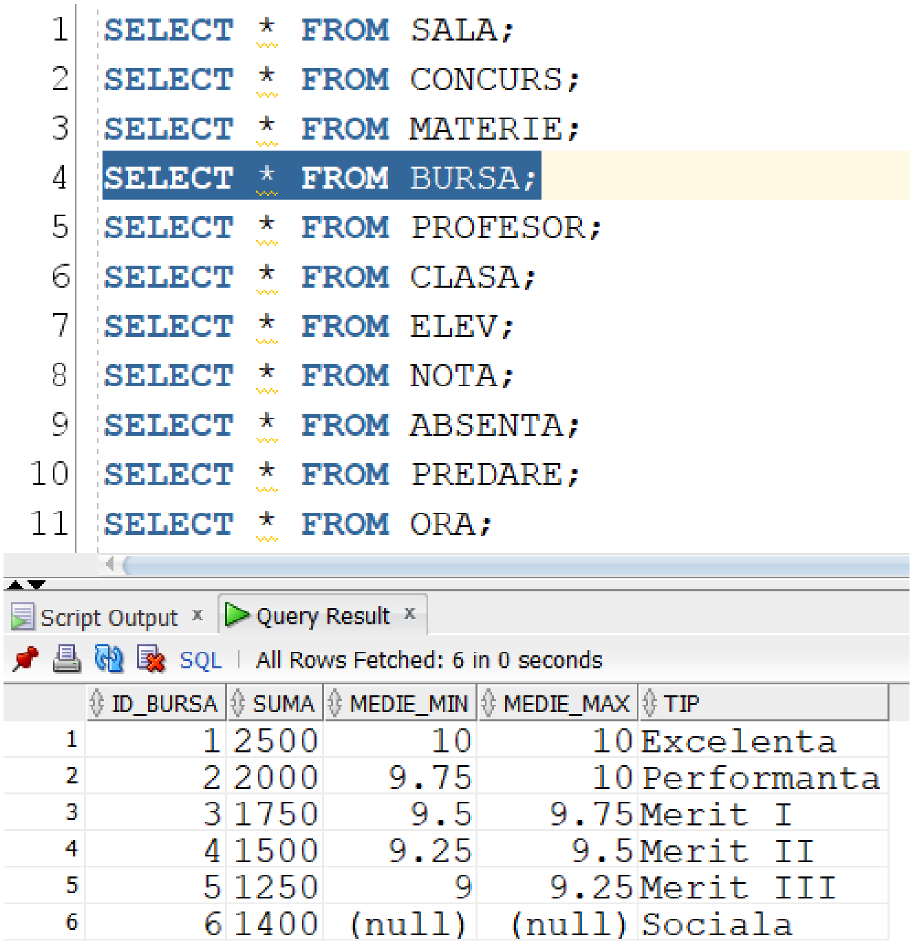
	ID_CONCURS	NUME
1	1	Euclid
2	2	Arhimede
3	3	Terra
4	4	Smart
5	5	Winners
6	6	Cangurul



```
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 10 in 0.002 seconds

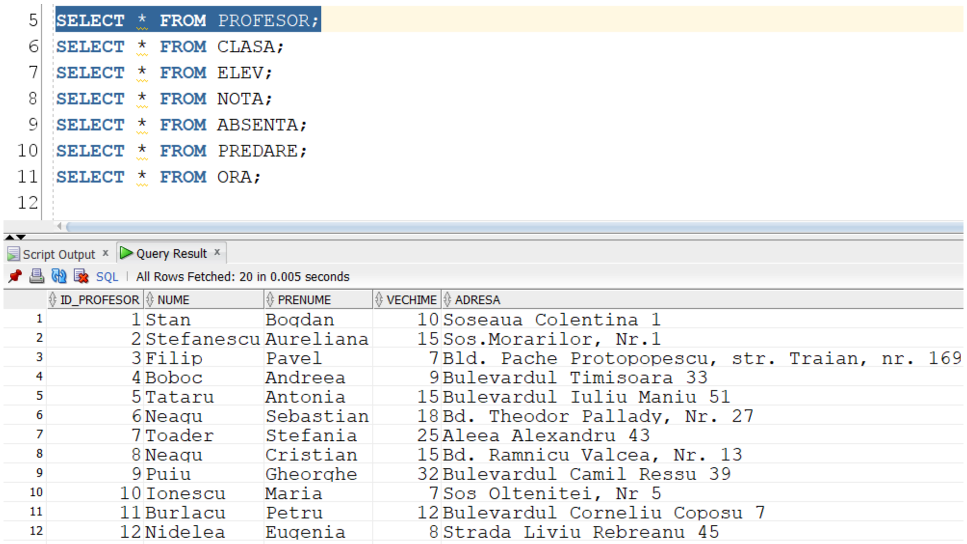
	ID_MATERIE	DENUMIRE	NR_ORE_SAPT	FILIERA
1	1	Matematica	6	Real
2	2	Lb. Romana	6	Uman
3	3	Informatica	4	Real
4	4	Istorie	1	Uman
5	5	Geografie	1	Uman
6	6	Fizica	3	Real
7	7	Chimie	3	Real
8	8	Biologie	1	Real
9	9	Sociologie	1	Uman
10	10	Sport	2	Vocational



```
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 6 in 0 seconds

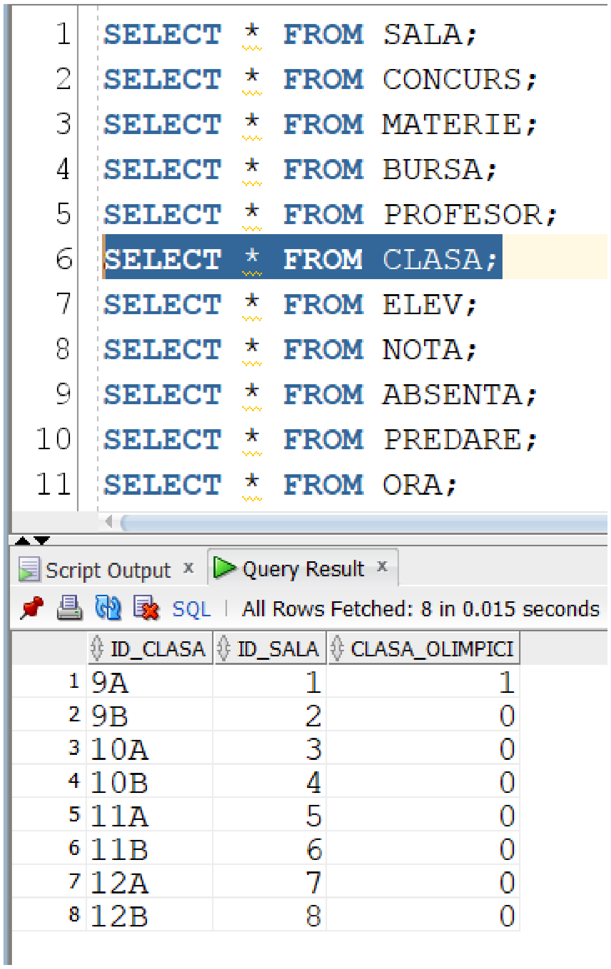
	ID_BURSA	SUMA	MEDIE_MIN	MEDIE_MAX	TIP
1	1	2500	10	10	Excelenta
2	2	2000	9.75	10	Performanta
3	3	1750	9.5	9.75	Merit I
4	4	1500	9.25	9.5	Merit II
5	5	1250	9	9.25	Merit III
6	6	1400	(null)	(null)	Sociala



```
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 20 in 0.005 seconds

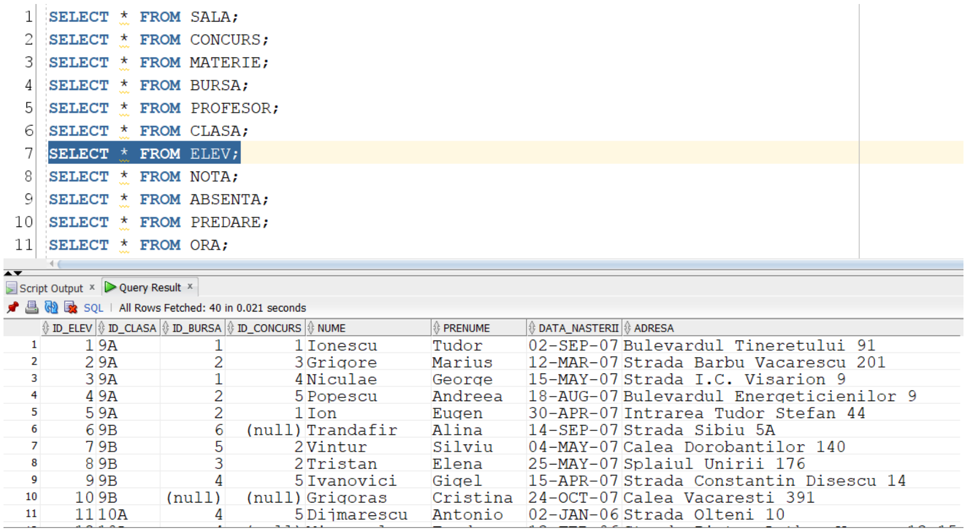
	ID_PROFESOR	NUME	PRENUME	VECHIME	ADRESA
1	1	Stan	Bogdan	10	Soseaua Colentina 1
2	2	Stefanescu	Aureliana	15	Sos.Morarilor, Nr.1
3	3	Filip	Pavel	7	Bld. Pache Protopopescu, str. Traian, nr. 169
4	4	Boboc	Andreea	9	Bulevardul Timisoara 33
5	5	Tataru	Antonia	15	Bulevardul Iuliu Maniu 51
6	6	Neagu	Sebastian	18	Bd. Theodor Pallady, Nr. 27
7	7	Toader	Stefania	25	Aleea Alexandru 43
8	8	Neagu	Cristian	15	Bd. Ramnicu Valcea, Nr. 13
9	9	Puiu	Gheorghe	32	Bulevardul Camil Ressu 39
10	10	Ionescu	Maria	7	Sos Oltenitei, Nr 5
11	11	Burlacu	Petru	12	Bulevardul Corneliu Coposu 7
12	12	Nidelea	Eugenia	8	Strada Liviu Rebreanu 45



```sql
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 8 in 0.015 seconds

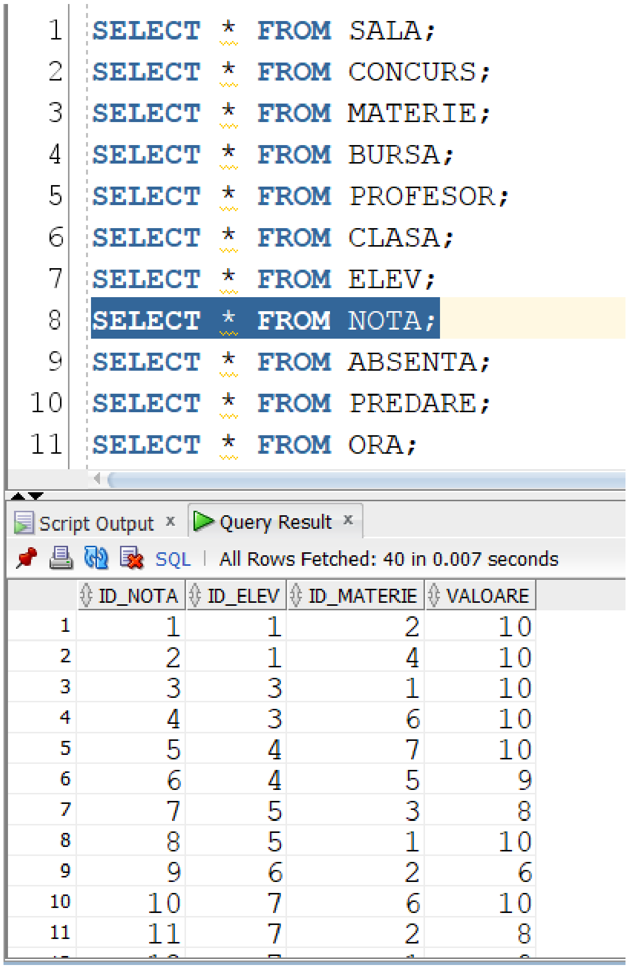
	ID_CLASA	ID_SALA	CLASA_OLIMPICI
1	9A	1	1
2	9B	2	0
3	10A	3	0
4	10B	4	0
5	11A	5	0
6	11B	6	0
7	12A	7	0
8	12B	8	0



```sql
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 40 in 0.021 seconds

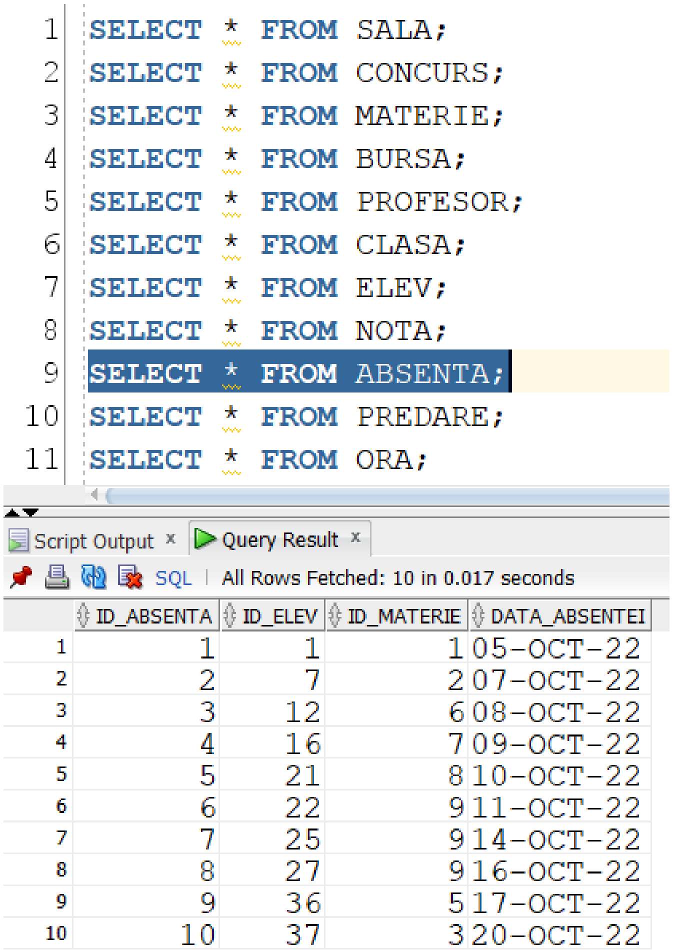
	ID_ELEV	ID_CLASA	ID_BURSA	ID_CONCURS	NUME	PRENUME	DATA_NASTERII	ADRESA
1	1	9A	1	1	Ionescu	Tudor	02-SEP-07	Bulevardul Tineretului 91
2	2	9A	2	3	Grigore	Marius	12-MAR-07	Strada Barbu Vacarescu 201
3	3	9A	1	4	Niculae	George	15-MAY-07	Strada I.C. Visarion 9
4	4	9A	2	5	Popescu	Andreea	18-AUG-07	Bulevardul Energeticienilor 9
5	5	9A	2	1	Ion	Eugen	30-APR-07	Intrarea Tudor Stefan 44
6	6	9B	6	(null)	Trandafir	Alina	14-SEP-07	Strada Sibiu 5A
7	7	9B	5	2	Vintur	Silviu	04-MAY-07	Calea Dorobantilor 140
8	8	9B	3	2	Tristan	Elena	25-MAY-07	Splaiul Unirii 176
9	9	9B	4	5	Ivanovici	Gigel	15-APR-07	Strada Constantin Disescu 14
10	10	9B	(null)	(null)	Grigoras	Cristina	24-OCT-07	Calea Vacaresti 391
11	11	10A	4	5	Dijmarescu	Antonio	02-JAN-06	Strada Olteni 10



```sql
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 40 in 0.007 seconds

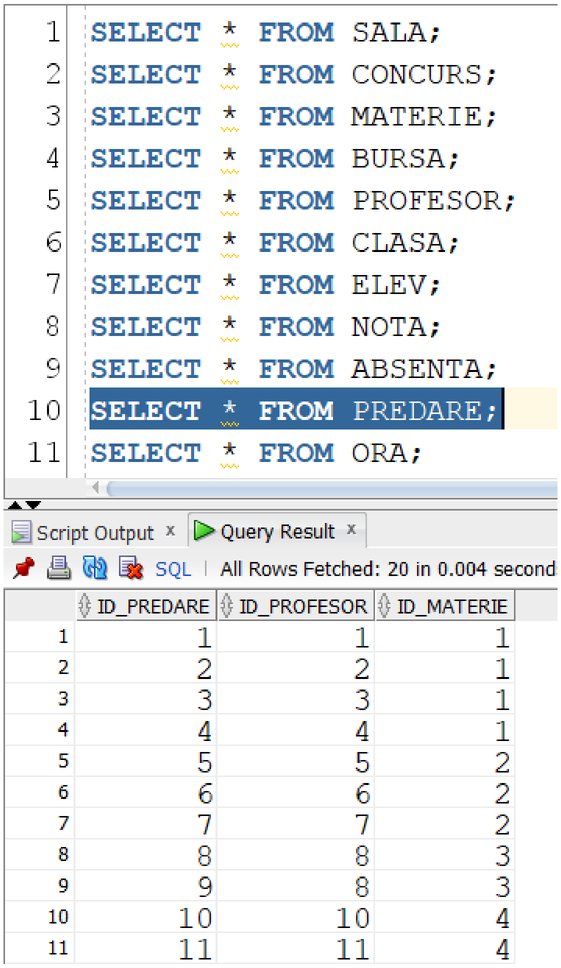
	ID_NOTA	ID_ELEV	ID_MATERIE	VALOARE
1	1	1	2	10
2	2	1	4	10
3	3	3	1	10
4	4	3	6	10
5	5	4	7	10
6	6	4	5	9
7	7	5	3	8
8	8	5	1	10
9	9	6	2	6
10	10	7	6	10
11	11	7	2	8



```
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 10 in 0.017 seconds

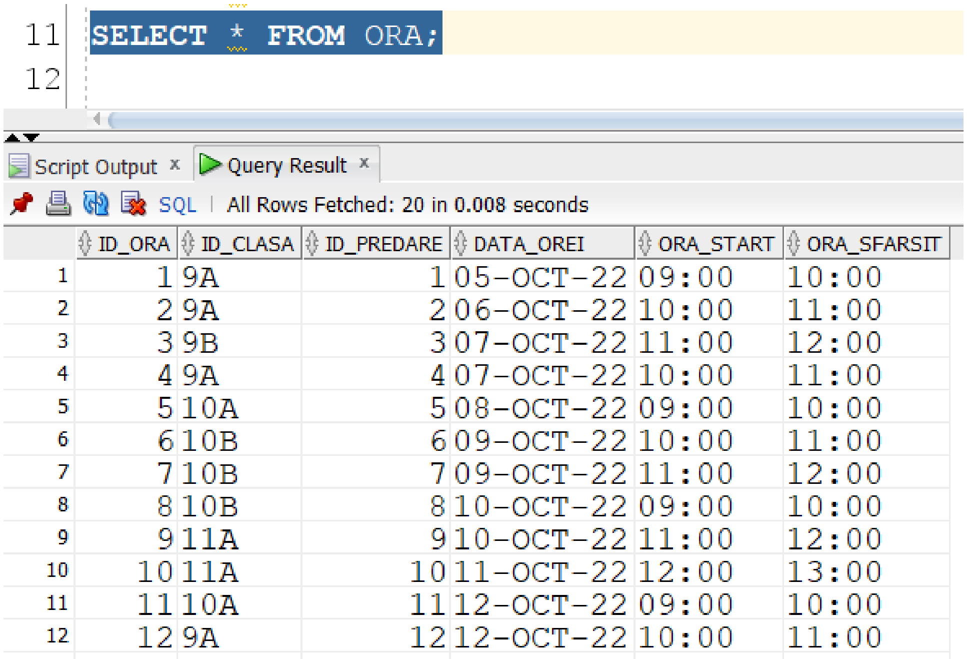
	ID_ABSENTA	ID_ELEV	ID_MATERIE	DATA_ABSENTEI
1	1	1	1	05-OCT-22
2	2	7	2	07-OCT-22
3	3	12	6	08-OCT-22
4	4	16	7	09-OCT-22
5	5	21	8	10-OCT-22
6	6	22	9	11-OCT-22
7	7	25	9	14-OCT-22
8	8	27	9	16-OCT-22
9	9	36	5	17-OCT-22
10	10	37	3	20-OCT-22



```sql
SELECT * FROM SALA;
SELECT * FROM CONCURS;
SELECT * FROM MATERIE;
SELECT * FROM BURSA;
SELECT * FROM PROFESOR;
SELECT * FROM CLASA;
SELECT * FROM ELEV;
SELECT * FROM NOTA;
SELECT * FROM ABSENTA;
SELECT * FROM PREDARE;
SELECT * FROM ORA;
```

All Rows Fetched: 20 in 0.004 second

	ID_PREDARE	ID_PROFESOR	ID_MATERIE
1	1	1	1
2	2	2	1
3	3	3	1
4	4	4	1
5	5	5	2
6	6	6	2
7	7	7	2
8	8	8	3
9	9	8	3
10	10	10	4
11	11	11	4



```sql
SELECT * FROM ORA;

```

All Rows Fetched: 20 in 0.008 seconds

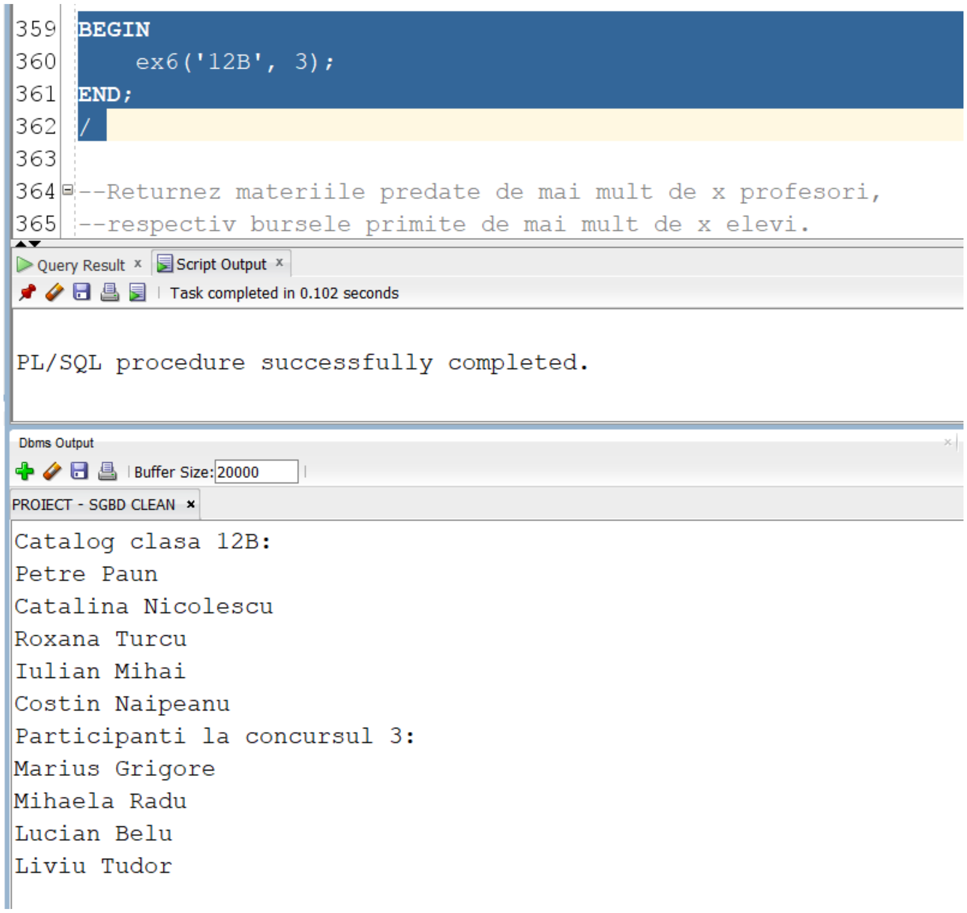
	ID_ORA	ID_CLASA	ID_PREDARE	DATA_OREI	ORA_START	ORA_SFARSIT
1	1	9A	1	05-OCT-22	09:00	10:00
2	2	9A	2	06-OCT-22	10:00	11:00
3	3	9B	3	07-OCT-22	11:00	12:00
4	4	9A	4	07-OCT-22	10:00	11:00
5	5	10A	5	08-OCT-22	09:00	10:00
6	6	10B	6	09-OCT-22	10:00	11:00
7	7	10B	7	09-OCT-22	11:00	12:00
8	8	10B	8	10-OCT-22	09:00	10:00
9	9	11A	9	10-OCT-22	11:00	12:00
10	10	11A	10	11-OCT-22	12:00	13:00
11	11	10A	11	12-OCT-22	09:00	10:00
12	12	9A	12	12-OCT-22	10:00	11:00

Ex. 6

Mentineti intr-o colectie toti elevii dintr-o clasa al carei id este dat ca parametru si in alta colectie toti participantii la un concurs dat.

```
CREATE OR REPLACE PROCEDURE ex6
    (id_cls CLASA.ID_CLASA%TYPE,
    id_conc CONCURS.ID_CONCURS%TYPE)
 IS
    TYPE tablou_elevi IS TABLE OF VARCHAR2(100);
    TYPE tablou_concurs IS VARRAY(50) OF VARCHAR2(100);
    v_nume tablou_elevi;
    v_prenume tablou_elevi;
    p_concurs tablou_concurs;
    n_concurs tablou_concurs;
 BEGIN
    SELECT NUME, PRENUME
    BULK COLLECT INTO v_nume, v_prenume
    FROM ELEV
    WHERE ID_CLASA = id_cls;

    SELECT NUME, PRENUME
    BULK COLLECT INTO n_concurs, p_concurs
    FROM ELEV
    WHERE ID_CONCURS = id_conc;

    DBMS_OUTPUT.PUT_LINE('Catalog clasa ' || id_cls || ': ');
    FOR i IN v_nume.FIRST..v_nume.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(v_prenume(i) || ' ' || v_nume(i));
    END LOOP;
```

```
        DBMS_OUTPUT.PUT_LINE('Participanti la concursul ' || id_conc || ': ' );
        FOR i IN p_concurs.FIRST..p_concurs.LAST LOOP
        DBMS_OUTPUT.PUT_LINE(p_concurs(i) || ' ' || n_concurs(i));
        END LOOP;
    END;
/

BEGIN
    ex6('12B', 3);
END;
/
```



```
BEGIN
    ex6('12B', 3);
END;
/

--Returnez materiile predate de mai mult de x profesori,
--respectiv bursele primite de mai mult de x elevi.
```

Task completed in 0.102 seconds

```
PL/SQL procedure successfully completed.
```

```
Catalog clasa 12B:
Petre Paun
Catalina Nicolescu
Roxana Turcu
Iulian Mihai
Costin Naipeanu
Participanti la concursul 3:
Marius Grigore
Mihaela Radu
Lucian Belu
Liviu Tudor
```

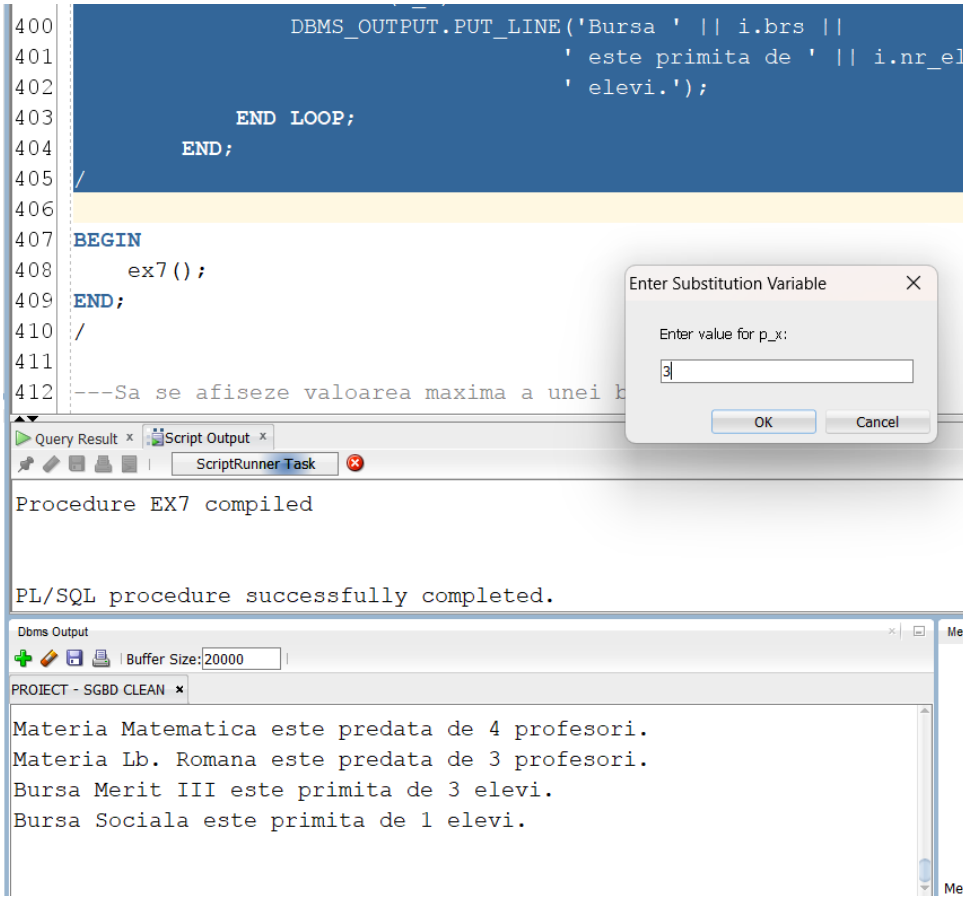
Ex. 7

Returnez materiile predate de mai mult de x profesori, respectiv bursele primite de mai mult de x elevi.

Folosesc cursor explicit parametrizat si ciclu cursor parametrizat.

```
CREATE OR REPLACE PROCEDURE ex7
  AS
    v_x NUMBER(4) := &p_x;
    v_materie MATERIE.DENUMIRE%TYPE;
    v_nr_prof NUMBER(4);

    CURSOR c1 (parametru NUMBER) IS
      SELECT DENUMIRE, COUNT(m.ID_MATERIE)
      FROM MATERIE m
      JOIN PREDARE p
      ON m.ID_MATERIE = p.ID_MATERIE
      GROUP BY DENUMIRE
      HAVING COUNT(m.ID_MATERIE) >= parametru;

    CURSOR c2 (parametru NUMBER) IS
      SELECT b.TIP brs, COUNT(ID_ELEV) nr_elevi
      FROM bursa b
      JOIN elev e
      ON b.ID_BURSA = e.ID_BURSA
      GROUP BY b.TIP
      HAVING COUNT(ID_ELEV) <= parametru;
    BEGIN
      OPEN c1(v_x);
      LOOP
```

```
        FETCH c1 INTO v_materie, v_nr_prof;
        EXIT WHEN c1%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Materia ' || v_materie ||
                  ' este predata de ' || v_nr_prof ||
                  ' profesori.');
      END LOOP;
      CLOSE c1;

      FOR i IN c2(v_x) LOOP
        DBMS_OUTPUT.PUT_LINE('Bursa ' || i.brs ||
                  ' este primita de ' || i.nr_elevi ||
                  ' elevi.');
      END LOOP;
    END;
/

BEGIN
  ex7();
END;
/
```



```
                DBMS_OUTPUT.PUT_LINE('Bursa ' || i.brs ||
                                     ' este primita de ' || i.nr_el
                                     ' elevi.');
            END LOOP;
        END;
/

BEGIN
    ex7();
END;
/

---Sa se afiseze valoarea maxima a unei b
```

Enter value for p_x: 3

```
Procedure EX7 compiled


PL/SQL procedure successfully completed.
```

Buffer Size: 20000

```
Materia Matematica este predata de 4 profesori.
Materia Lb. Romana este predata de 3 profesori.
Bursa Merit III este primita de 3 elevi.
Bursa Sociala este primita de 1 elevi.
```

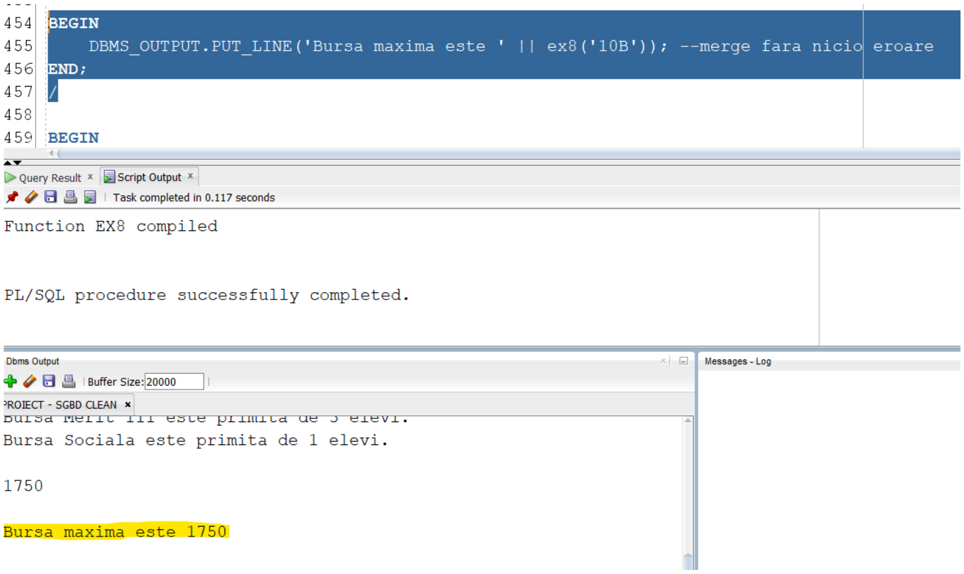
Ex. 8

Sa se afiseze valoarea maxima a unei burse dintr-o clasa al carei id este dat ca parametru.

```
CREATE OR REPLACE FUNCTION ex8
    (id_cls CLASA.ID_CLASA%TYPE)
RETURN NUMBER IS
    bursa_maxima BURSA.SUMA%TYPE;
    flag CLASA.CLASA_OLIMPICI%TYPE;  --definit auxiliar pentru a ma ajuta sa determin daca exista o clasa corespunzatoare id-ului dat
    FARA_BURSA EXCEPTION;
BEGIN
    SELECT CLASA_OLIMPICI
    INTO flag
```

```
        FROM CLASA
        WHERE ID_CLASA = id_cls;

        IF SQL%NOTFOUND THEN
           RAISE NO_DATA_FOUND;
        END IF;

        SELECT MAX(b.SUMA)
        INTO bursa_maxima
        FROM CLASA c
        JOIN ELEV e
        ON c.ID_CLASA = e.ID_CLASA
        JOIN BURSA b
        ON e.ID_BURSA = b.ID_BURSA
        WHERE c.ID_CLASA = id_cls;
        IF BURSA_MAXIMA IS NULL THEN
           RAISE FARA_BURSA;
        ELSE
           RETURN bursa_maxima;
        END IF;
EXCEPTION
        WHEN NO_DATA_FOUND THEN
           DBMS_OUTPUT.PUT_LINE('Nu exista aceasta clasa!');
           RETURN -1;
        WHEN FARA_BURSA THEN
           DBMS_OUTPUT.PUT_LINE('Nu exista bursieri in aceasta clasa!');
           RETURN -1;
END ex8;
/
```

```
BEGIN

   DBMS_OUTPUT.PUT_LINE('Bursa maxima este ' || ex8('10B')); --merge fara nicio eroare

END;
```

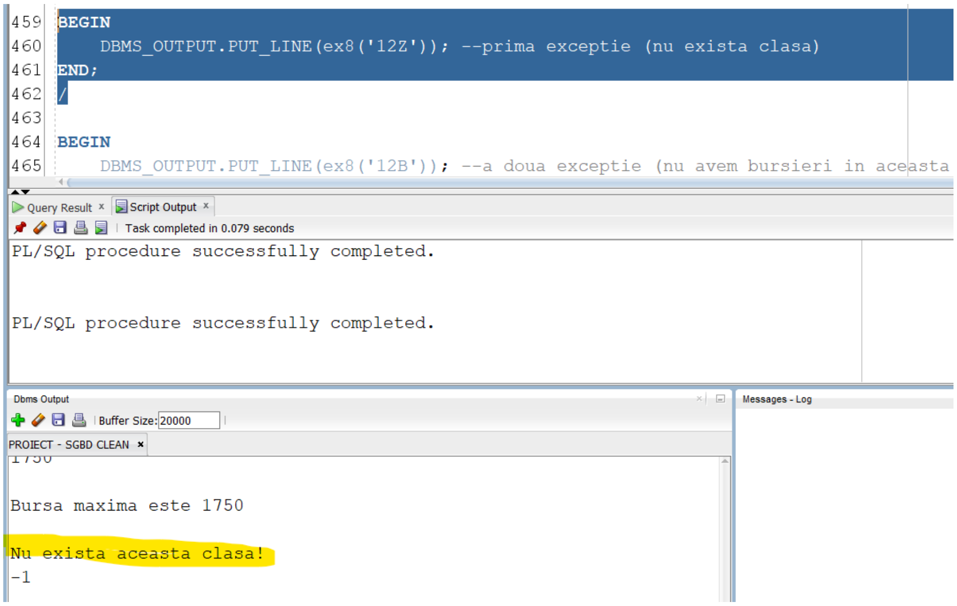


```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Bursa maxima este ' || ex8('10B')); --merge fara nicio eroare
END;
/

BEGIN
```

```
Function EX8 compiled


PL/SQL procedure successfully completed.
```

```
Bursa Merit III este primita de 3 elevi.
Bursa Sociala este primita de 1 elevi.

1750

Bursa maxima este 1750
```

```
/

BEGIN

   DBMS_OUTPUT.PUT_LINE(ex8('12Z')); --prima exceptie (nu exista clasa)

END;

/
```

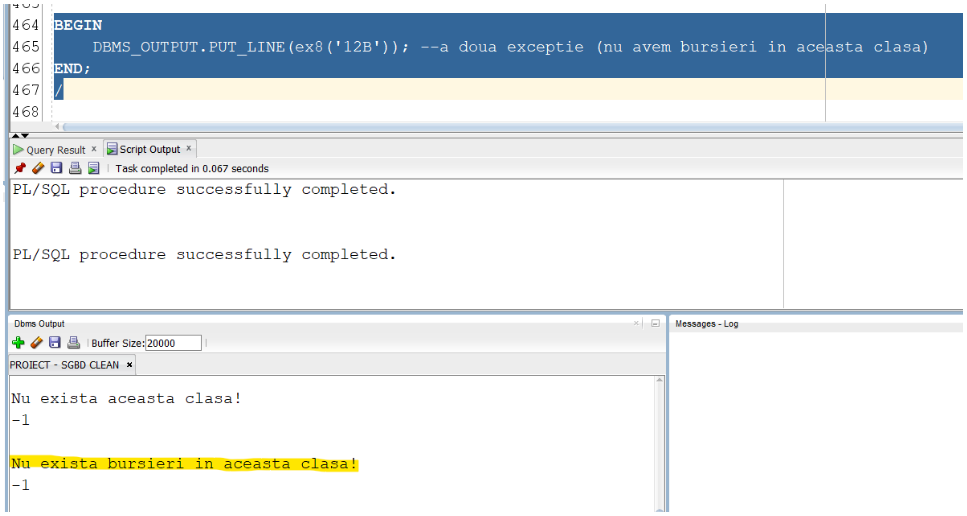


```
BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('12Z')); --prima exceptie (nu exista clasa)
END;
/

BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('12B')); --a doua exceptie (nu avem bursieri in aceasta
```

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

```
1750

Bursa maxima este 1750

Nu exista aceasta clasa!
-1
```

```
BEGIN
   DBMS_OUTPUT.PUT_LINE(ex8('12B')); --a doua exceptie (nu avem bursieri in aceasta clasa)
END;
/
```



```
BEGIN
    DBMS_OUTPUT.PUT_LINE(ex8('12B')); --a doua exceptie (nu avem bursieri in aceasta clasa)
END;
/
```

Task completed in 0.067 seconds

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

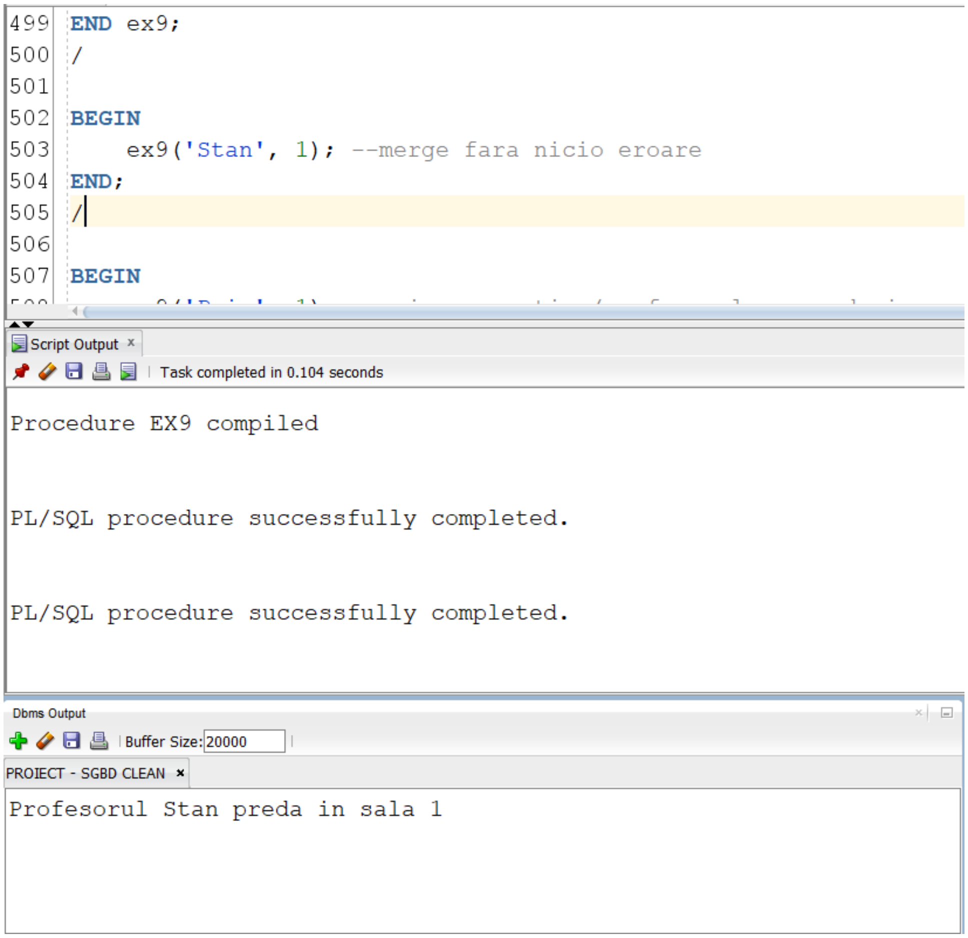
```
Nu exista aceasta clasa!
-1

Nu exista bursieri in aceasta clasa!
-1
```

Ex. 9

Fiind dat numele unui profesor si id-ul unei sali date ca parametri, sa se determine daca acel profesor preda in acea sala

```
CREATE OR REPLACE PROCEDURE ex9(nume_prof PROFESOR.NUME%TYPE, cod_sala
SALA.id_sala%TYPE)

IS

   v_nume PROFESOR.NUME%TYPE;

   FARA_SALA EXCEPTION;

BEGIN

   IF COD_SALA < 1 OR COD_SALA > 9 THEN --nu exista aceasta sala

      RAISE FARA_SALA;

   END IF;
   SELECT prof.NUME
   INTO v_nume
   FROM SALA s
   JOIN CLASA cls
   ON s.ID_SALA = cls.ID_SALA
   JOIN ORA o
   ON cls.ID_CLASA = o.ID_CLASA
   JOIN PREDARE pred
   ON o.ID_PREDARE = pred.ID_PREDARE
   JOIN PROFESOR prof
   ON pred.ID_PROFESOR = prof.ID_PROFESOR
   WHERE s.ID_SALA = cod_sala AND prof.NUME = nume_prof;

   DBMS_OUTPUT.PUT_LINE('Profesorul ' || nume_prof || ' preda in sala ' || cod_sala);

EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20000, 'Profesorul ' || nume_prof || ' NU preda in sala ' || cod_sala);
   WHEN TOO_MANY_ROWS THEN
      RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi profesori cu numele dat');
   WHEN FARA_SALA THEN
      RAISE_APPLICATION_ERROR(-20002, 'Nu exista sala data cu id-ul dat ca parametru');
   WHEN OTHERS THEN
      RAISE_APPLICATION_ERROR(-20003, 'Alta eroare!');
END ex9;
/
```

```
BEGIN

    ex9('Stan', 1); --merge fara nicio eroare

END;

/
```

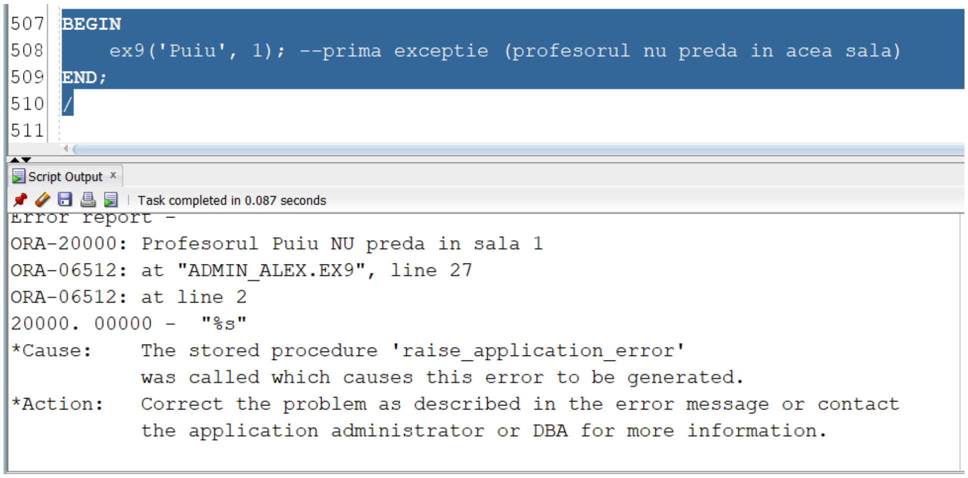


```
END ex9;
/

BEGIN
    ex9('Stan', 1); --merge fara nicio eroare
END;
/

BEGIN
```

```
Procedure EX9 compiled


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

```
Profesorul Stan preda in sala 1
```

```
BEGIN

    ex9('Puiu', 1); --prima exceptie (profesorul nu preda in acea sala)

END;

/
```



```
BEGIN
    ex9('Puiu', 1); --prima exceptie (profesorul nu preda in acea sala)
END;
/
```

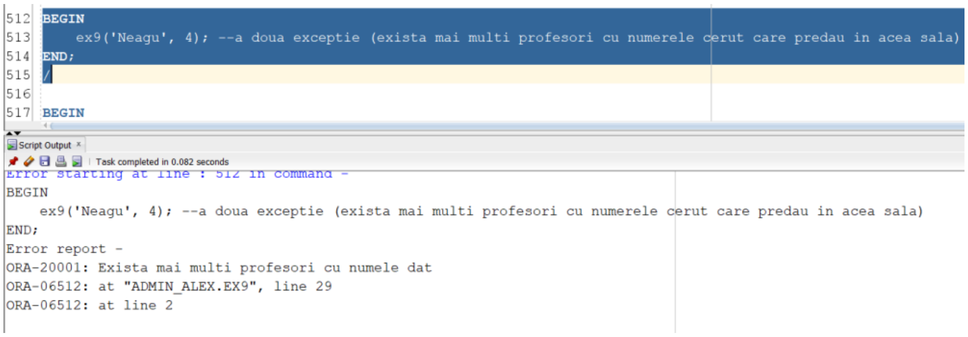
```
Error report -
ORA-20000: Profesorul Puiu NU preda in sala 1
ORA-06512: at "ADMIN_ALEX.EX9", line 27
ORA-06512: at line 2
20000. 00000 -  "%s"
*Cause:    The stored procedure 'raise_application_error'
           was called which causes this error to be generated.
*Action:   Correct the problem as described in the error message or contact
           the application administrator or DBA for more information.
```

BEGIN

    ex9('Neagu', 4); --a doua exceptie (exista mai multi profesori cu numerele cerut care predau in acea sala)

END;

/



```
BEGIN
    ex9('Neagu', 4); --a doua exceptie (exista mai multi profesori cu numerele cerut care predau in acea sala)
END;
/

BEGIN
```

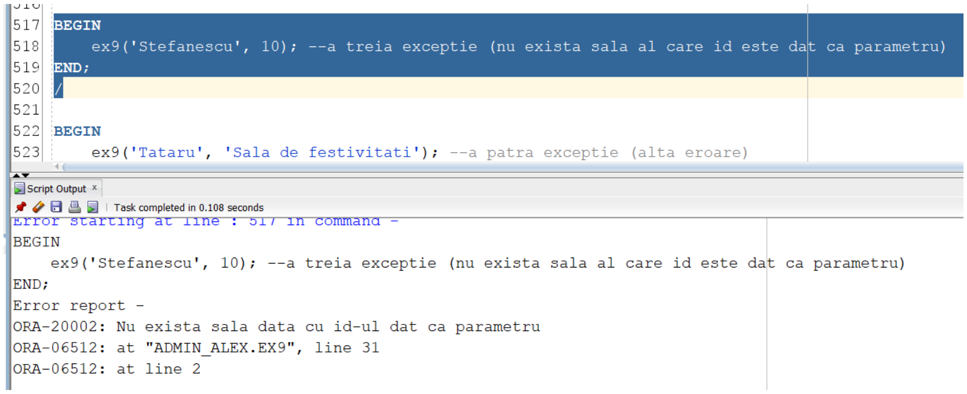
```
Error starting at line : 512 in command -
BEGIN
    ex9('Neagu', 4); --a doua exceptie (exista mai multi profesori cu numerele cerut care predau in acea sala)
END;
Error report -
ORA-20001: Exista mai multi profesori cu numele dat
ORA-06512: at "ADMIN_ALEX.EX9", line 29
ORA-06512: at line 2
```

BEGIN

    ex9('Stefanescu', 10); --a treia exceptie (nu exista sala al care id este dat ca parametru)

END;

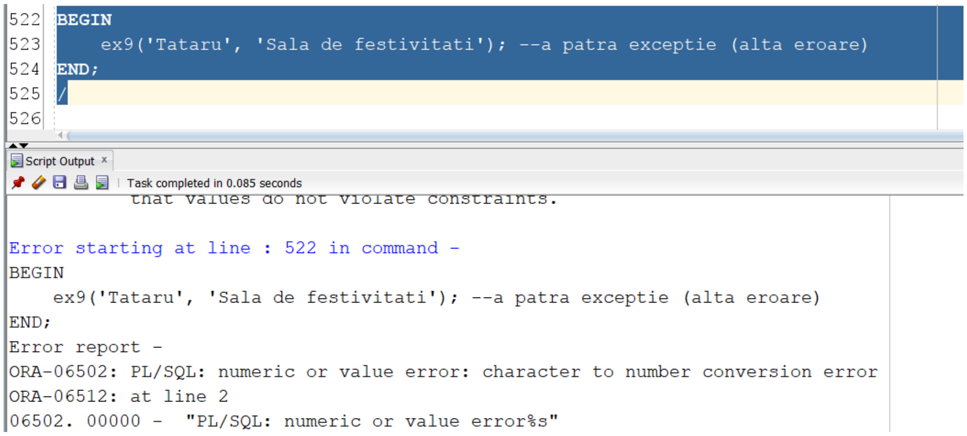
/



```
BEGIN
    ex9('Stefanescu', 10); --a treia exceptie (nu exista sala al care id este dat ca parametru)
END;
/

BEGIN
    ex9('Tataru', 'Sala de festivitati'); --a patra exceptie (alta eroare)
```

```
Error starting at line : 517 in command -
BEGIN
    ex9('Stefanescu', 10); --a treia exceptie (nu exista sala al care id este dat ca parametru)
END;
Error report -
ORA-20002: Nu exista sala data cu id-ul dat ca parametru
ORA-06512: at "ADMIN_ALEX.EX9", line 31
ORA-06512: at line 2
```

```
BEGIN

    ex9('Tataru', 'Sala de festivitati'); --a patra exceptie (alta eroare)

END;

/
```



```
BEGIN
    ex9('Tataru', 'Sala de festivitati'); --a patra exceptie (alta eroare)
END;
/
```

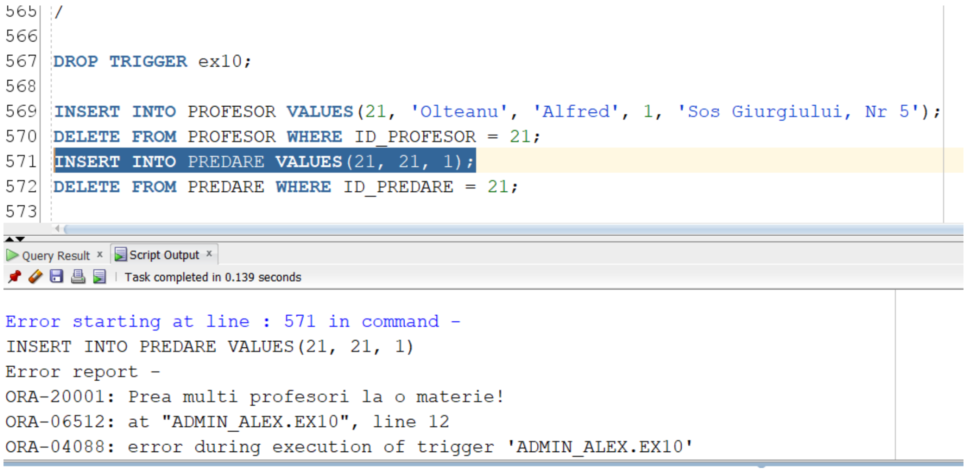
```
           that values do not violate constraints.

Error starting at line : 522 in command -
BEGIN
    ex9('Tataru', 'Sala de festivitati'); --a patra exceptie (alta eroare)
END;
Error report -
ORA-06502: PL/SQL: numeric or value error: character to number conversion error
ORA-06512: at line 2
06502. 00000 -  "PL/SQL: numeric or value error%s"
```

Ex. 10

Trigger care nu permite sa avem mai mult de 4 profesori la o materie.

```
CREATE OR REPLACE TRIGGER ex10

    AFTER INSERT OR UPDATE ON PREDARE
```

```
DECLARE
   nr_profesori NUMBER(3);
BEGIN
   SELECT UNIQUE COUNT(ID_MATERIE)
   INTO NR_PROFESORI
   FROM PREDARE
   GROUP BY ID_MATERIE
   HAVING COUNT(ID_MATERIE) = (SELECT MAX(COUNT(ID_MATERIE))
                                              FROM PREDARE
                                              GROUP BY ID_MATERIE);
   IF nr_profesori > 4 THEN
      RAISE_APPLICATION_ERROR(-20001, 'Prea multi profesori la o materie!');
   END IF;
END;
/
DROP TRIGGER ex10;


INSERT INTO PROFESOR VALUES(21, 'Olteanu', 'Alfred', 1, 'Sos Giurgiului, Nr 5');
DELETE FROM PROFESOR WHERE ID_PROFESOR = 21;
INSERT INTO PREDARE VALUES(21, 21, 1);
DELETE FROM PREDARE WHERE ID_PREDARE = 21;
```



```
565 /
566
567 DROP TRIGGER ex10;
568
569 INSERT INTO PROFESOR VALUES(21, 'Olteanu', 'Alfred', 1, 'Sos Giurgiului, Nr 5');
570 DELETE FROM PROFESOR WHERE ID_PROFESOR = 21;
571 INSERT INTO PREDARE VALUES(21, 21, 1);
572 DELETE FROM PREDARE WHERE ID_PREDARE = 21;
573
```

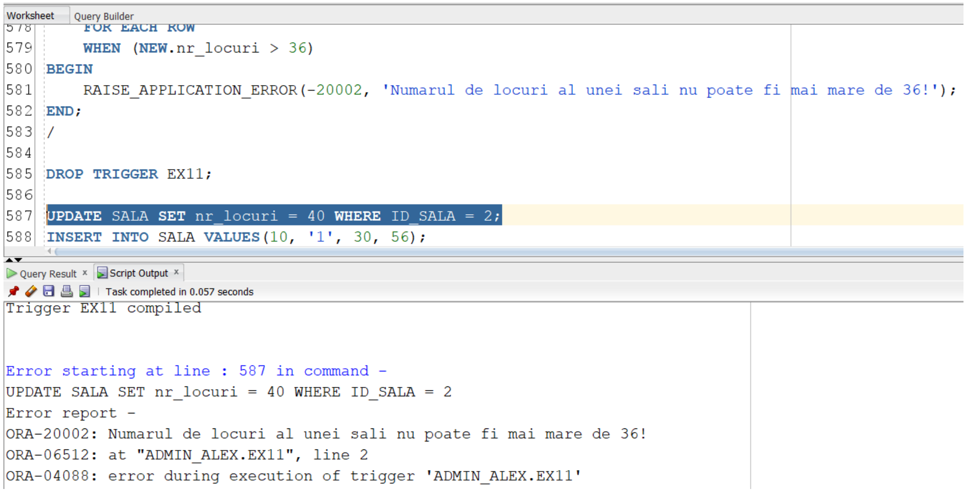
Task completed in 0.139 seconds

```
Error starting at line : 571 in command -
INSERT INTO PREDARE VALUES(21, 21, 1)
Error report -
ORA-20001: Prea multi profesori la o materie!
ORA-06512: at "ADMIN_ALEX.EX10", line 12
ORA-04088: error during execution of trigger 'ADMIN_ALEX.EX10'
```

Ex. 11

Trigger care nu permite ca numarul de locuri dintr-o sala sa fie mai mare decat 36.

```
CREATE OR REPLACE TRIGGER ex11

   BEFORE INSERT OR UPDATE OF nr_locuri ON sala

   FOR EACH ROW

   WHEN (NEW.nr_locuri > 36)

BEGIN

   RAISE_APPLICATION_ERROR(-20002, 'Numarul de locuri al unei sali nu poate fi mai mare de 36!');

END;

/

DROP TRIGGER EX11;


UPDATE SALA SET nr_locuri = 40 WHERE ID_SALA = 2;

INSERT INTO SALA VALUES(10, '1', 30, 56);

DELETE FROM SALA WHERE ID_SALA = 10;
```



```
    FOR EACH ROW
    WHEN (NEW.nr_locuri > 36)
BEGIN
    RAISE_APPLICATION_ERROR(-20002, 'Numarul de locuri al unei sali nu poate fi mai mare de 36!');
END;
/

DROP TRIGGER EX11;

UPDATE SALA SET nr_locuri = 40 WHERE ID_SALA = 2;
INSERT INTO SALA VALUES(10, '1', 30, 56);
```

```
Trigger EX11 compiled


Error starting at line : 587 in command -
UPDATE SALA SET nr_locuri = 40 WHERE ID_SALA = 2
Error report -
ORA-20002: Numarul de locuri al unei sali nu poate fi mai mare de 36!
ORA-06512: at "ADMIN_ALEX.EX11", line 2
ORA-04088: error during execution of trigger 'ADMIN_ALEX.EX11'
```

Ex. 12

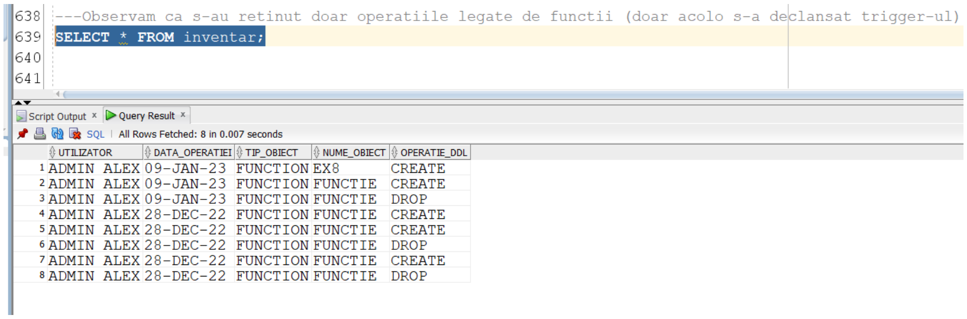
Trigger care retine intr-o tabela informatii cu referire la operatiile de tip LDD efectuate DOAR asupra functiiilor.

```
CREATE TABLE inventar(
   utilizator VARCHAR2(20),
   data_operatiei DATE,
   tip_obiect VARCHAR2(30),
   nume_obiect VARCHAR2(50),
   operatie_ddl VARCHAR2(50)
);
```

```
CREATE OR REPLACE TRIGGER ex12
   AFTER CREATE OR ALTER OR DROP ON SCHEMA
DECLARE
   var_tip inventar.tip_obiect%TYPE;
BEGIN
   var_tip := SYS.DICTIONARY_OBJ_TYPE;
   IF var_tip = 'FUNCTION' THEN
      INSERT INTO inventar VALUES (SYS.LOGIN_USER, SYSDATE, SYS.DICTIONARY_OBJ_TYPE,
SYS.DICTIONARY_OBJ_NAME, SYS.SYSEVENT);
   END IF;
END;
/
DROP TRIGGER ex12;
```

```
---Aici se declanseaza trigger-ul
CREATE OR REPLACE FUNCTION functie
RETURN VARCHAR2 IS
   text VARCHAR2(100);
```

```
BEGIN
   text := 'Hello, world!';
   DBMS_OUTPUT.PUT_LINE(text);
END functie;
/
DROP FUNCTION functie;


---Aici NU se declanseaza trigger-ul
CREATE OR REPLACE PROCEDURE procedura
IS
   text VARCHAR2(100);
BEGIN
   text := 'Hello, world!';
   DBMS_OUTPUT.PUT_LINE(text);
END procedura;
/
DROP PROCEDURE procedura;


---Observam ca s-au retinut doar operatiile legate de functii (doar acolo s-a declansat trigger-ul)
SELECT * FROM inventar;
```



```
---Observam ca s-au retinut doar operatiile legate de functii (doar acolo s-a declansat trigger-ul)
SELECT * FROM inventar;
```

All Rows Fetched: 8 in 0.007 seconds

	UTILIZATOR	DATA_OPERATIEI	TIP_OBIECT	NUME_OBIECT	OPERATIE_DDL
1	ADMIN ALEX	09-JAN-23	FUNCTION	EX8	CREATE
2	ADMIN ALEX	09-JAN-23	FUNCTION	FUNCTIE	CREATE
3	ADMIN ALEX	09-JAN-23	FUNCTION	FUNCTIE	DROP
4	ADMIN ALEX	28-DEC-22	FUNCTION	FUNCTIE	CREATE
5	ADMIN ALEX	28-DEC-22	FUNCTION	FUNCTIE	CREATE
6	ADMIN ALEX	28-DEC-22	FUNCTION	FUNCTIE	DROP
7	ADMIN ALEX	28-DEC-22	FUNCTION	FUNCTIE	CREATE
8	ADMIN ALEX	28-DEC-22	FUNCTION	FUNCTIE	DROP

Ex. 13

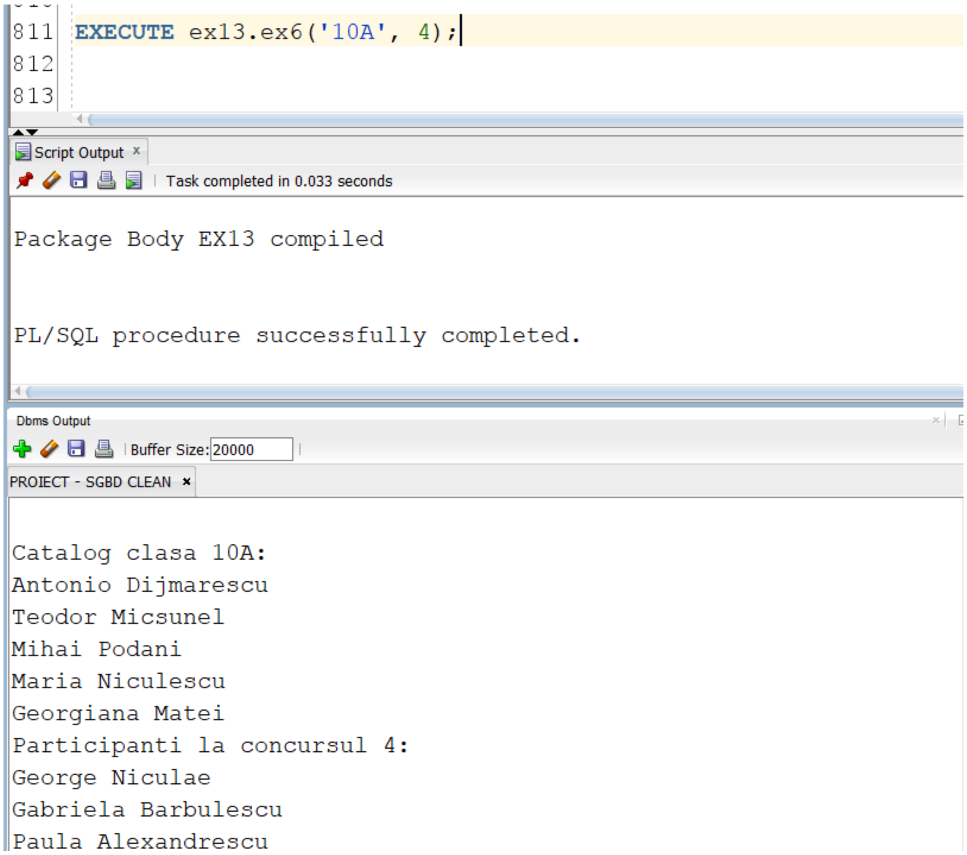
---Definiti un pachet care sa contina toate obiectele definite în cadrul proiectului.

```
CREATE OR REPLACE PACKAGE ex13
AS
  PROCEDURE ex6(id_cls CLASA.ID_CLASA%TYPE,
    id_conc CONCURS.ID_CONCURS%TYPE);
  PROCEDURE ex7;
  FUNCTION ex8 (id_cls CLASA.ID_CLASA%TYPE) RETURN NUMBER;
  PROCEDURE ex9(nume_prof PROFESOR.NUME%TYPE, cod_sala SALA.id_sala%TYPE);
END ex13;
/
CREATE OR REPLACE PACKAGE BODY ex13 AS
  PROCEDURE ex6
    (id_cls CLASA.ID_CLASA%TYPE,
    id_conc CONCURS.ID_CONCURS%TYPE)
  IS
    TYPE tablou_elevi IS TABLE OF VARCHAR2(100);
    TYPE tablou_concurs IS VARRAY(50) OF VARCHAR2(100);
    v_nume tablou_elevi;
    v_prenume tablou_elevi;
    p_concurs tablou_concurs;
    n_concurs tablou_concurs;
  BEGIN
    SELECT NUME, PRENUME
    BULK COLLECT INTO v_nume, v_prenume
    FROM ELEV
    WHERE ID_CLASA = id_cls;

    SELECT NUME, PRENUME
    BULK COLLECT INTO n_concurs, p_concurs
    FROM ELEV
    WHERE ID_CONCURS = id_conc;

    DBMS_OUTPUT.PUT_LINE('Catalog clasa ' || id_cls || ': ');
    FOR i IN v_nume.FIRST..v_nume.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(v_prenume(i) || ' ' || v_nume(i));
    END LOOP;

    DBMS_OUTPUT.PUT_LINE('Participanti la concursul ' || id_conc || ': ' );
    FOR i IN p_concurs.FIRST..p_concurs.LAST LOOP
    DBMS_OUTPUT.PUT_LINE(p_concurs(i) || ' ' || n_concurs(i));
    END LOOP;
  END ex6;

PROCEDURE ex7
  AS
    v_x NUMBER(4) := &p_x;
    v_materie MATERIE.DENUMIRE%TYPE;
    v_nr_prof NUMBER(4);

  CURSOR c1 (parametru NUMBER) IS
    SELECT DENUMIRE, COUNT(m.ID_MATERIE)
    FROM MATERIE m
    JOIN PREDARE p
    ON m.ID_MATERIE = p.ID_MATERIE
    GROUP BY DENUMIRE
    HAVING COUNT(m.ID_MATERIE) >= parametru;

  CURSOR c2 (parametru NUMBER) IS
    SELECT b.TIP brs, COUNT(ID_ELEV) nr_elevi
    FROM bursa b
    JOIN elev e
    ON b.ID_BURSA = e.ID_BURSA
    GROUP BY b.TIP
    HAVING COUNT(ID_ELEV) <= parametru;
  BEGIN
    OPEN c1(v_x);
    LOOP
      FETCH c1 INTO v_materie, v_nr_prof;
      EXIT WHEN c1%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('Materia ' || v_materie ||
                ' este predata de ' || v_nr_prof ||
                ' profesori.');
    END LOOP;
    CLOSE c1;

    FOR i IN c2(v_x) LOOP
      DBMS_OUTPUT.PUT_LINE('Bursa ' || i.brs ||
                ' este primita de ' || i.nr_elevi ||
                ' elevi.');
    END LOOP;
  END ex7;

  FUNCTION ex8
    (id_cls CLASA.ID_CLASA%TYPE)
  RETURN NUMBER IS
    bursa_maxima BURSA.SUMA%TYPE;
    flag CLASA.CLASA_OLIMPICI%TYPE; --definit auxiliar pentru a ma ajuta sa determin daca exista o clasa corespunzatoare id-ului dat
  FARA_BURSA EXCEPTION;
BEGIN
  SELECT CLASA_OLIMPICI
  INTO flag
  FROM CLASA
  WHERE ID_CLASA = id_cls;

  IF SQL%NOTFOUND THEN
    RAISE NO_DATA_FOUND;
  END IF;

  SELECT MAX(b.SUMA)
  INTO bursa_maxima
  FROM CLASA c
  JOIN ELEV e
  ON c.ID_CLASA = e.ID_CLASA
  JOIN BURSA b
  ON e.ID_BURSA = b.ID_BURSA
  WHERE c.ID_CLASA = id_cls;
  IF BURSA_MAXIMA IS NULL THEN
    RAISE FARA_BURSA;
  ELSE
    RETURN bursa_maxima;
  END IF;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Nu exista aceasta clasa!');
    RETURN -1;
  WHEN FARA_BURSA THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nu exista bursieri in aceasta clasa!');
    RETURN -1;
END ex8;

PROCEDURE ex9(nume_prof PROFESOR.NUME%TYPE, cod_sala SALA.id_sala%TYPE)
IS
  v_nume PROFESOR.NUME%TYPE;
  FARA_SALA EXCEPTION;
BEGIN
  IF COD_SALA < 1 OR COD_SALA > 9 THEN --nu exista aceasta sala
    RAISE FARA_SALA;
  END IF;

  SELECT prof.NUME
  INTO v_nume
  FROM SALA s
  JOIN CLASA cls
  ON s.ID_SALA = cls.ID_SALA
  JOIN ORA o
  ON cls.ID_CLASA = o.ID_CLASA
  JOIN PREDARE pred
  ON o.ID_PREDARE = pred.ID_PREDARE
  JOIN PROFESOR prof
  ON pred.ID_PROFESOR = prof.ID_PROFESOR
  WHERE s.ID_SALA = cod_sala AND prof.NUME = nume_prof;

  DBMS_OUTPUT.PUT_LINE('Profesorul ' || nume_prof || ' preda in sala ' || cod_sala);

EXCEPTION
```

```
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Profesorul ' || nume_prof || ' NU preda in sala ' || cod_sala);
        WHEN TOO_MANY_ROWS THEN
            RAISE_APPLICATION_ERROR(-20001, 'Exista mai multi profesori cu numele dat');
        WHEN FARA_SALA THEN
            RAISE_APPLICATION_ERROR(-20002, 'Nu exista sala data cu id-ul dat ca parametru');
        WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20003, 'Alta eroare!');
    END ex9;
END ex13;
/
```

EXECUTE ex13.ex6('10A', 4);



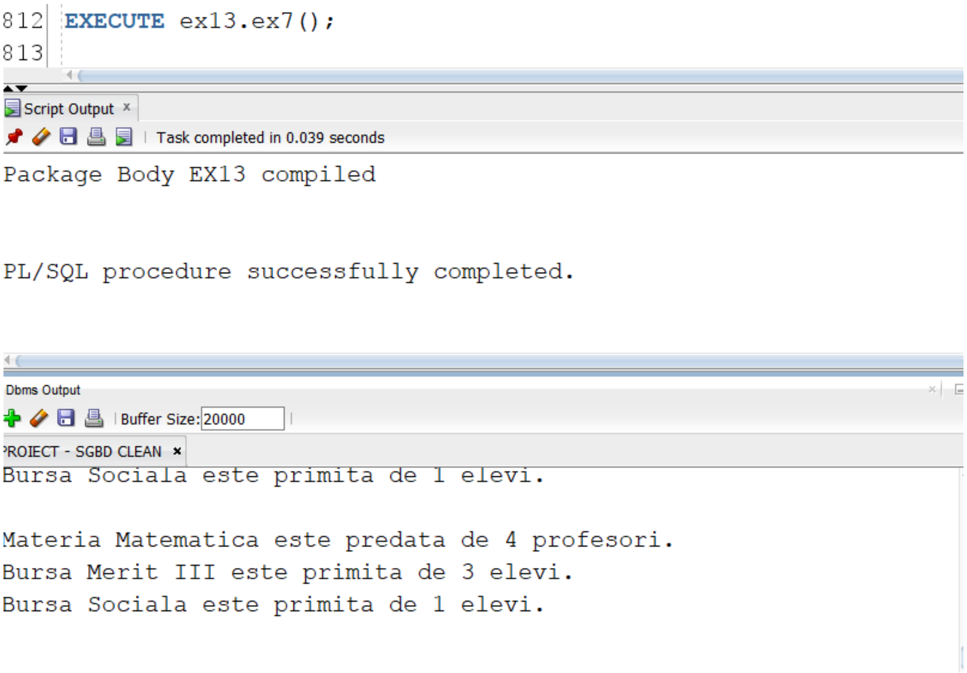
```
EXECUTE ex13.ex6('10A', 4);
```

Task completed in 0.033 seconds

```
Package Body EX13 compiled


PL/SQL procedure successfully completed.
```

```
Catalog clasa 10A:
Antonio Dijmarescu
Teodor Micsunel
Mihai Podani
Maria Niculescu
Georgiana Matei
Participanti la concursul 4:
George Niculae
Gabriela Barbulescu
Paula Alexandrescu
```

EXECUTE ex13.ex7();

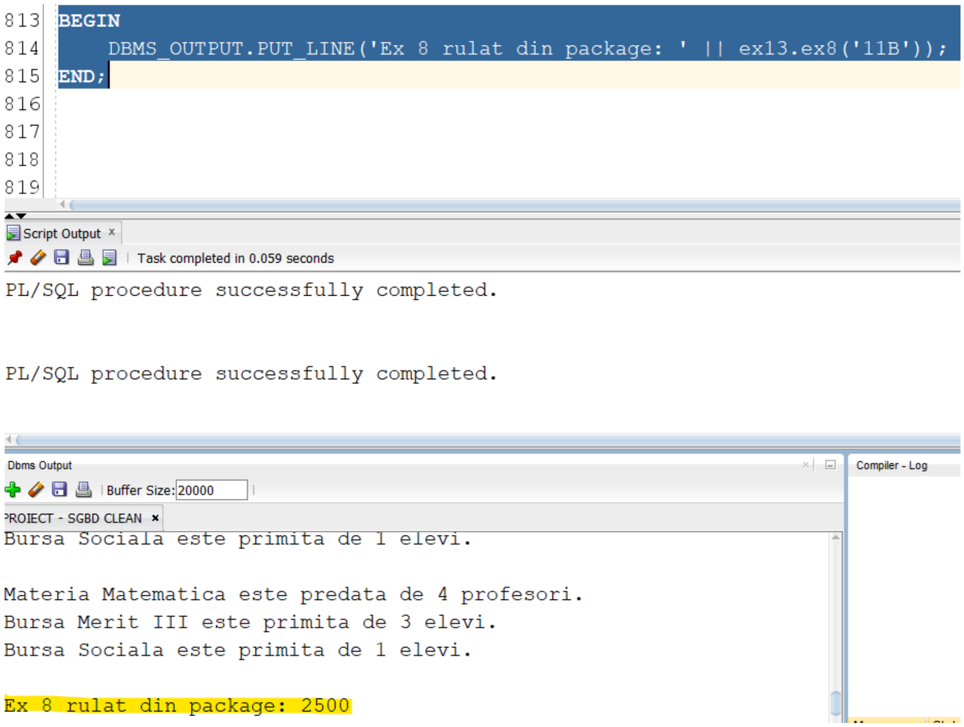


```
EXECUTE ex13.ex7();
```

```
Package Body EX13 compiled


PL/SQL procedure successfully completed.
```

```
Bursa Sociala este primita de 1 elevi.

Materia Matematica este predata de 4 profesori.
Bursa Merit III este primita de 3 elevi.
Bursa Sociala este primita de 1 elevi.
```

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Ex 8 rulat din package: ' || ex13.ex8('11B'));
END;
```

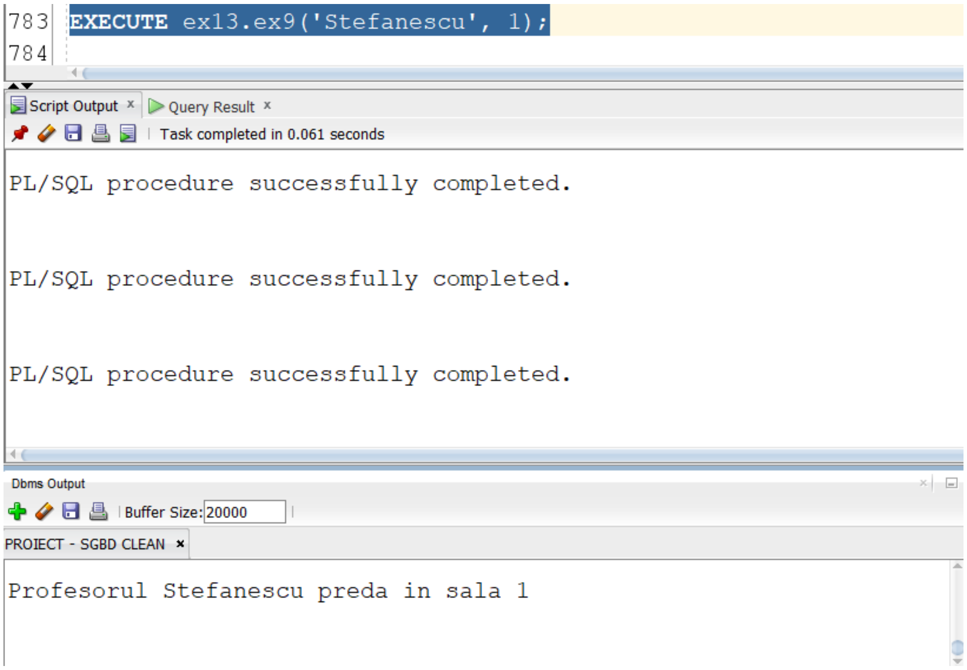


```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Ex 8 rulat din package: ' || ex13.ex8('11B'));
END;
```

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

```
Bursa Sociala este primita de 1 elevi.

Materia Matematica este predata de 4 profesori.
Bursa Merit III este primita de 3 elevi.
Bursa Sociala este primita de 1 elevi.

Ex 8 rulat din package: 2500
```

```
EXECUTE ex13.ex9('Stefanescu', 1);
```



```
EXECUTE ex13.ex9('Stefanescu', 1);
```

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.
```

```
Profesorul Stefanescu preda in sala 1
```