**CAMPIONAT DE KARTING**

**Prof. Laborator: Anamaria Stefania Zanoaga**

**Prof. Curs: Gabriela Mihai**

**Mircea Andrei Razvan**

**Grupa 232**

Cuprins

[Descriere baza de date 2](#_Toc155891294)

[Diagrama Entitate – Relatie 2](#_Toc155891295)

[Diagrama conceptuala 3](#_Toc155891296)

[Creare Tabele 3](#_Toc155891297)

[Inserturi 16](#_Toc155891298)

[Subprogram cu 3 colectii 27](#_Toc155891299)

[2 Cursoare 29](#_Toc155891300)

[Minim 3 tabele 31](#_Toc155891301)

[Minim 5 tabele 33](#_Toc155891302)

[Minim 5 tabele (v2) 36](#_Toc155891303)

[Trigger LMD la nivel de comanda 39](#_Toc155891304)

[Trigger LMD la nivel de linie 40](#_Toc155891305)

[Trigger LDD 41](#_Toc155891306)

[Pachet cu toate obiectele proiectului 42](#_Toc155891307)

[3 Coletii 51](#_Toc155891308)

[2 Cursoare 52](#_Toc155891309)

[3 Tabele 52](#_Toc155891310)

[5 Tabele 52](#_Toc155891311)

[5 Tabele v2 53](#_Toc155891312)

[Pachet cu tipuri de date complexe 53](#_Toc155891313)

[Lider categorie 58](#_Toc155891314)

[Risc penalizare 59](#_Toc155891315)

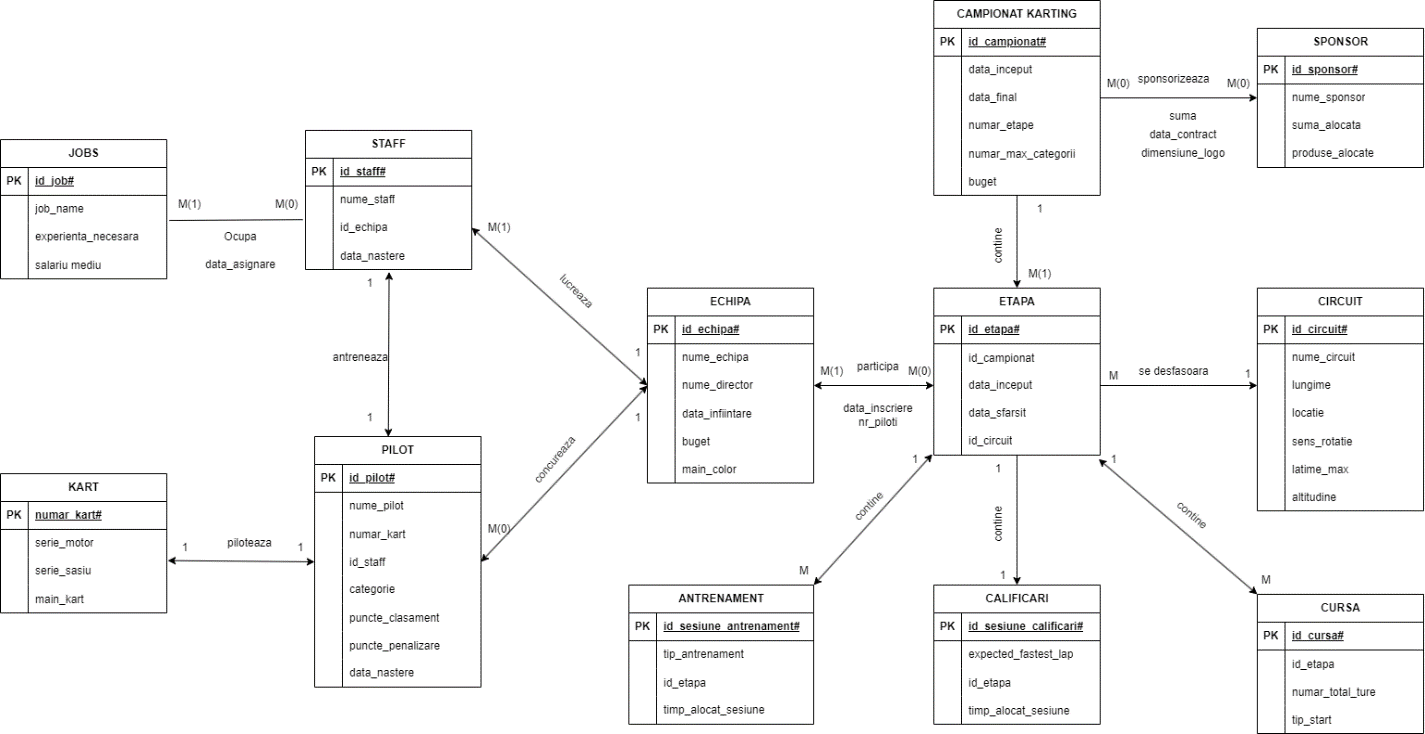
[Salariu staff 60](#_Toc155891316)

[Top 3 piloti 62](#_Toc155891317)

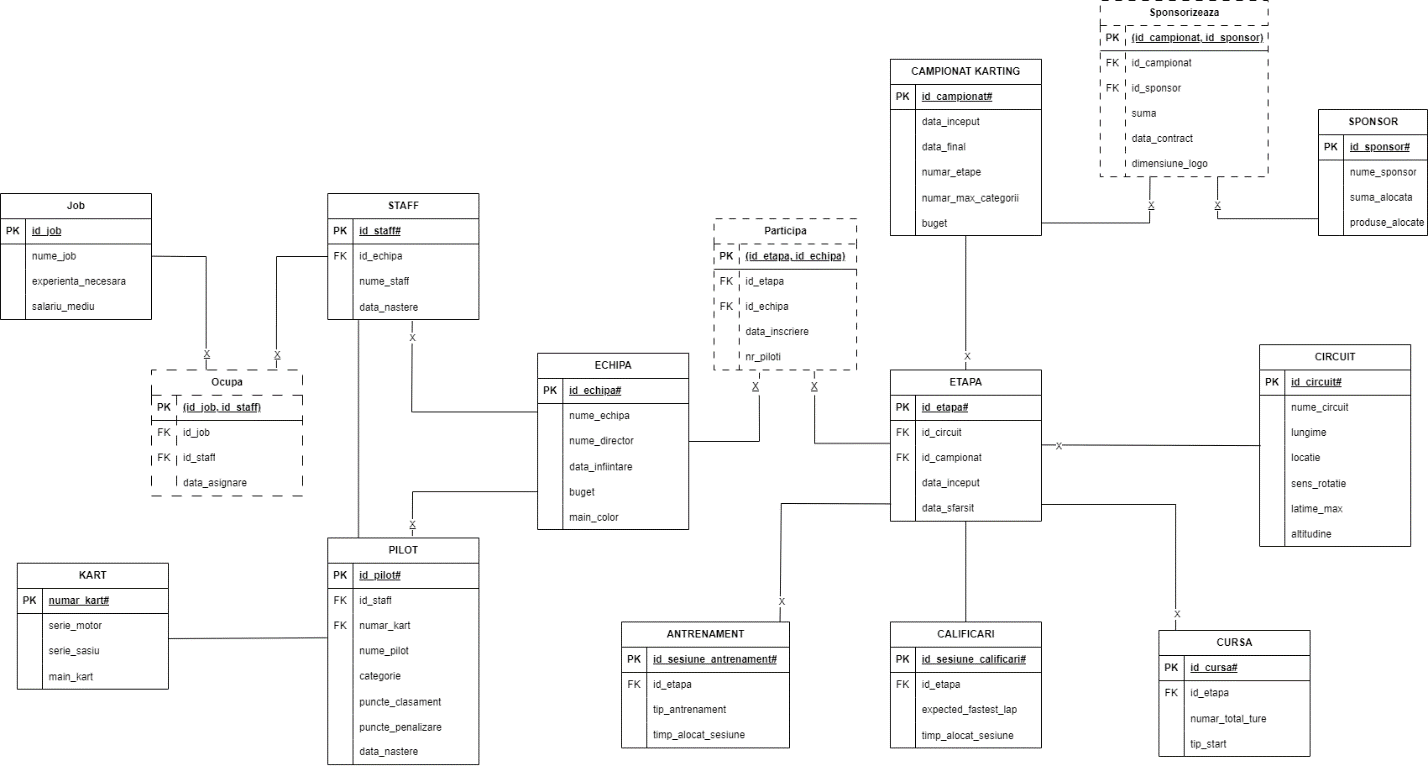
# Descriere baza de date

Diagrama ajuta la procesul de organizare a unui campionat de karting. Campionatul este alcatuit din mai multe etape. La fiecare etapa participa mai multe echipe formate din piloti si mecanici. Pentru fiecare pilot exista macar un mecanic/staff care se ocupa de antrenarea acestuia. Fiecare membru staff detine minim o functie/un job. Fiecare pilot are un kart, ce poate fi principal sau de rezerva. Fiecare etapa este alcatuita din mai multe antrenamente, o sesiune de calificari si mai multe curse. De asemenea, fiecare etapa se desfasoara pe un circuit. Pe tot parcursul campionatului pot exista potentiali sponsori (in general, contractul dintre sponsor si campionat este pe toata durata campionatului, motiv pentru care nu exista o conexiune directa intre o etapa si un sponsor).

# Diagrama Entitate – Relatie



# Diagrama conceptuala



# Creare Tabele

create table Campionat\_Karting

(

id\_campionat number(10) constraint pk\_campionat PRIMARY KEY,

data\_inceput date constraint data\_inceput\_campionat not null,

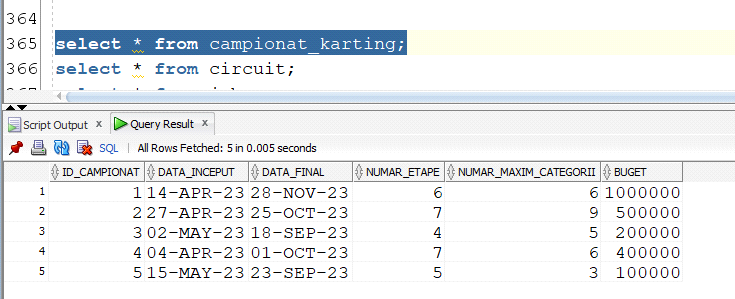
data\_final date constraint data\_final\_campionat not null,

numar\_etape number(10),

numar\_maxim\_categorii number(10),

buget number(10)

);



create table circuit

(

id\_circuit number(10) constraint pk\_circuit PRIMARY KEY,

nume\_circuit varchar2(30),

lungime number(10),

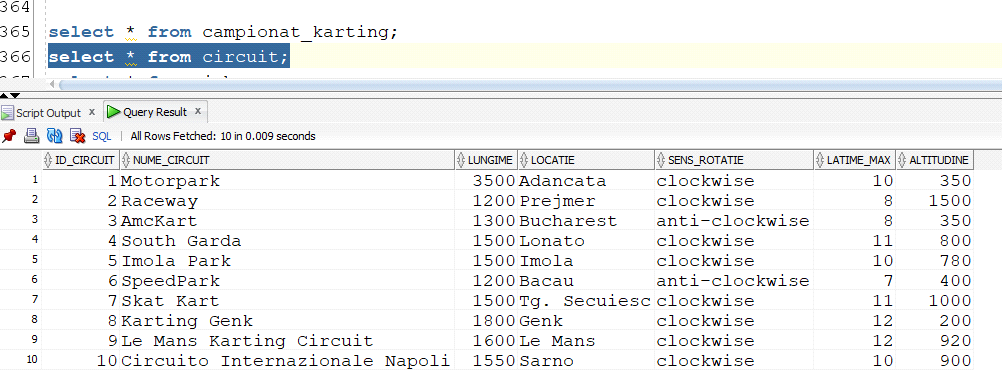
locatie varchar2(30),

sens\_rotatie varchar2(30) check(sens\_rotatie in ('clockwise','anti-clockwise')),

latime\_max number(2),

altitudine number(5)

);



create table sponsor

(

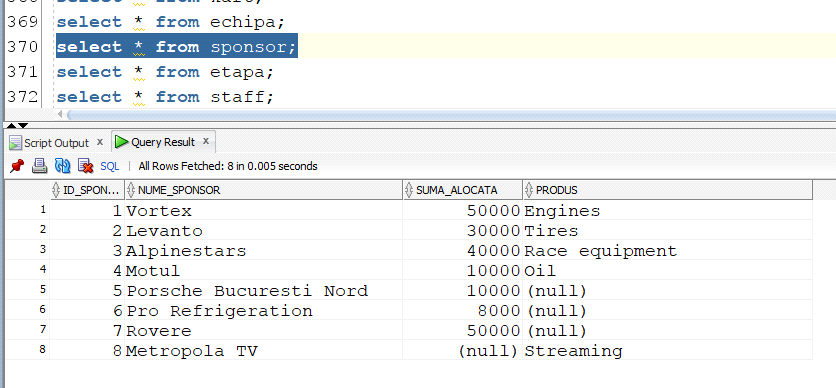
id\_sponsor number(10) constraint pk\_sponsor PRIMARY KEY,

nume\_sponsor varchar2(30),

suma\_alocata number(10),

produs varchar2(30)

);



create table etapa

(

id\_etapa number(10) constraint pk\_etapa PRIMARY KEY,

data\_inceput date constraint data\_inceput\_etapa not null,

data\_final date constraint data\_final\_etapa not null,

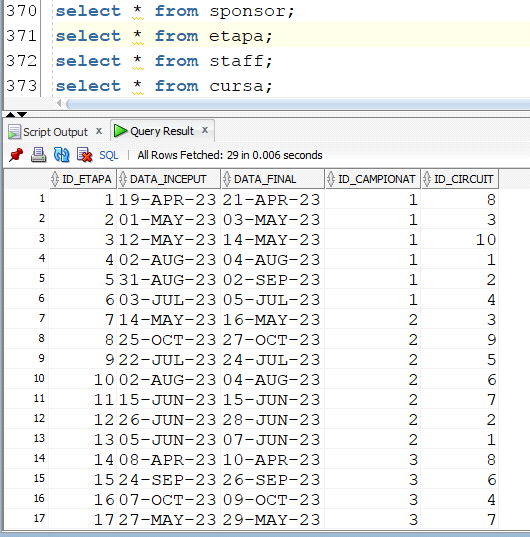
id\_campionat number(3),

id\_circuit number(3),

FOREIGN KEY (id\_campionat) references campionat\_karting(id\_campionat),

FOREIGN KEY (id\_circuit) references circuit(id\_circuit)

);



create table antrenament

(

id\_antrenament number(10) constraint pk\_antrenament PRIMARY KEY,

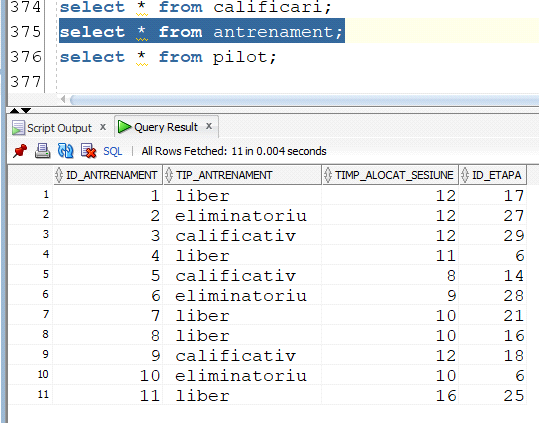
tip\_antrenament varchar2(20),

timp\_alocat\_sesiune number(5),

id\_etapa number(10),

FOREIGN KEY (id\_etapa) references etapa(id\_etapa)

);



create table calificari

(

id\_sesiune\_calificari number(10) constraint pk\_calificari PRIMARY KEY,

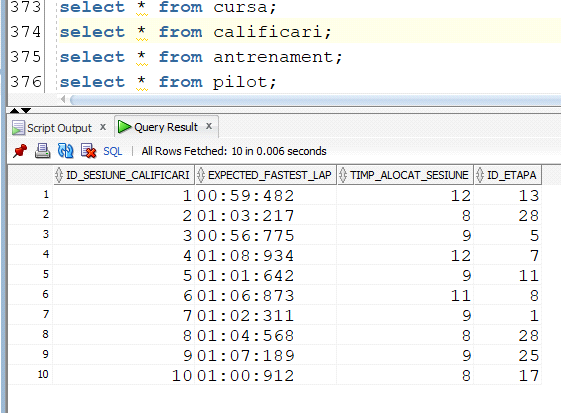
expected\_fastest\_lap varchar2(9),

timp\_alocat\_sesiune number(5),

id\_etapa number(10),

FOREIGN KEY (id\_etapa) references etapa(id\_etapa)

);



create table cursa

(

id\_sesiune\_cursa number(10) constraint pk\_cursa PRIMARY KEY,

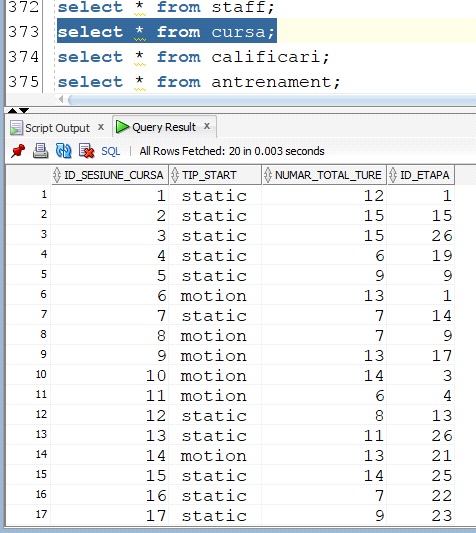
tip\_start varchar2(20),

numar\_total\_ture number(5),

id\_etapa number(10),

FOREIGN KEY (id\_etapa) references etapa(id\_etapa)

);



create table echipa

(

id\_echipa number(10) constraint pk\_echipa PRIMARY KEY,

nume\_echipa varchar2(30),

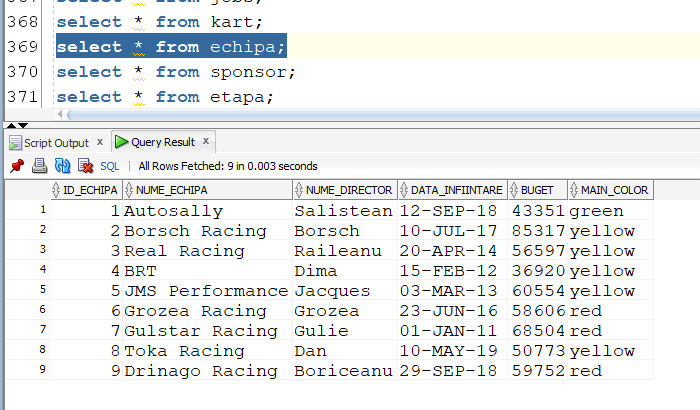
nume\_director varchar2(30),

data\_infiintare date,

buget number(10),

main\_color varchar2(30)

);



create table kart

(

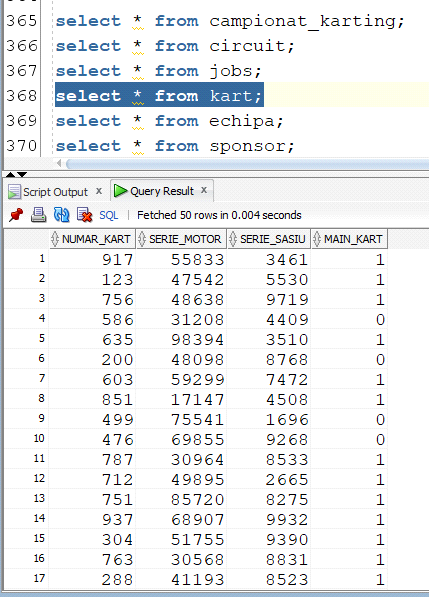
numar\_kart number(3) constraint pk\_kart Primary key,

serie\_motor number(10),

serie\_sasiu number(10),

main\_kart number(1) check (main\_kart in (0,1))

);



create table staff

(

id\_staff number(10) constraint pk\_staff PRIMARY KEY,

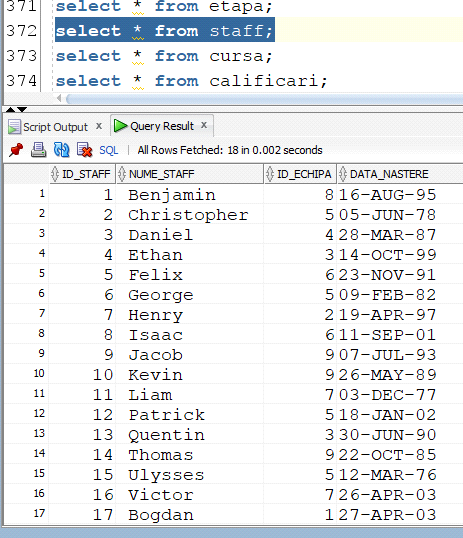
nume\_staff varchar2(30),

id\_echipa number(10),

data\_nastere date,

FOREIGN KEY (id\_echipa) references echipa(id\_echipa)

);



create table pilot

(

id\_pilot number(10) constraint pk\_pilot PRIMARY KEY,

nume\_pilot varchar2(30),

numar\_kart number(3),

id\_staff number(10),

data\_nastere date,

categorie varchar2(10),

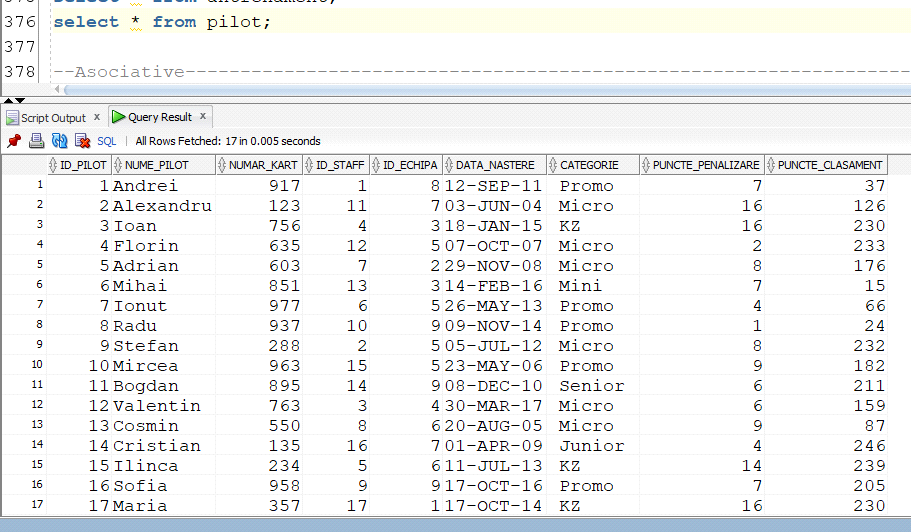
puncte\_penalizare number(2),

puncte\_clasament number(4),

FOREIGN KEY (id\_staff) references staff(id\_staff),

FOREIGN KEY (numar\_kart) references kart(numar\_kart)

);



create table jobs

(

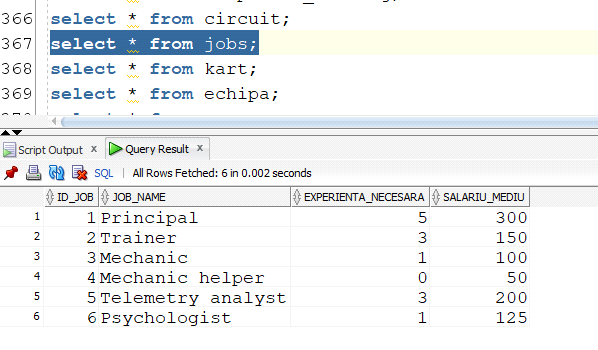
id\_job number(10) constraint pk\_job PRIMARY KEY,

job\_name varchar(30),

experienta\_necesara number(10),

salariu\_mediu number(10)

);



create table ocupa

(

id\_staff number(10),

id\_job number(10),

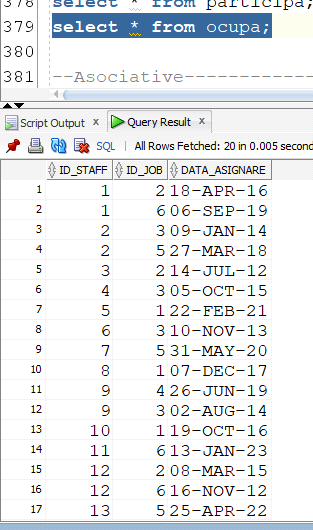
data\_asignare date,

constraint pk\_staff\_job\_ocupa primary key(id\_staff,id\_job),

FOREIGN KEY (id\_staff) references staff(id\_staff),

FOREIGN KEY (id\_job) references jobs(id\_job)

);



create table participa

(

id\_echipa number(10),

id\_etapa number(10),

data\_inscriere date,

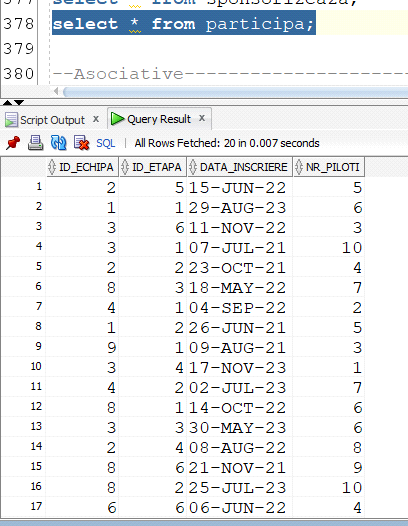
nr\_piloti number(10),

constraint pk\_echipa\_etapa\_participa primary key(id\_echipa,id\_etapa),

FOREIGN KEY (id\_etapa) references etapa(id\_etapa),

FOREIGN KEY (id\_echipa) references echipa(id\_echipa)

);



create table sponsorizeaza

(

id\_campionat number(10),

id\_sponsor number(10),

suma number(10),

data\_contract date,

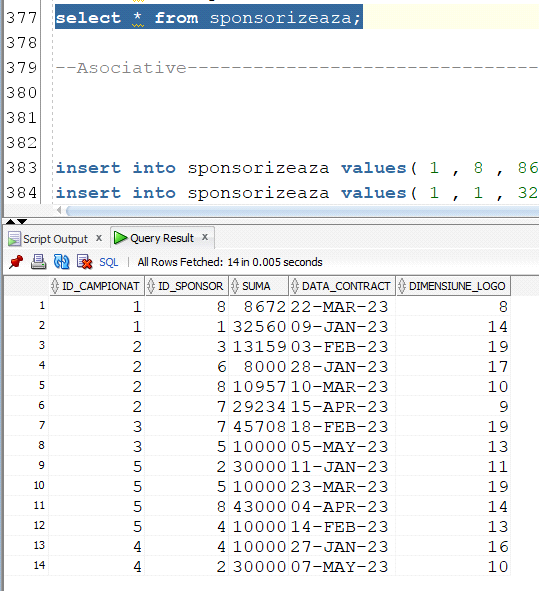
dimensiune\_logo number(10),

constraint pk\_campionat\_sponsor\_sponsorizeaza primary key(id\_campionat, id\_sponsor),

FOREIGN KEY (id\_campionat) references campionat\_karting(id\_campionat),

FOREIGN KEY (id\_sponsor) references sponsor(id\_sponsor)

);



# Inserturi

insert into campionat\_karting values(1,'14-APR-2023','28-NOV-2023',6,6,1000000);

insert into campionat\_karting values(2,'27-APR-2023','25-OCT-2023',7,9,500000);

insert into campionat\_karting values(3,'2-MAY-2023','18-SEP-2023',4,5,200000);

insert into campionat\_karting values(4,'4-APR-2023','1-OCT-2023',7,6,400000);

insert into campionat\_karting values(5,'15-MAY-2023','23-SEP-2023',5,3,100000);

insert into circuit values(3,'AmcKart',1300,'Bucharest','anti-clockwise',8,350);

insert into circuit values(4,'South Garda',1500,'Lonato','clockwise',11,800);

insert into circuit values(5,'Imola Park',1500,'Imola','clockwise',10,780);

insert into circuit values(6,'SpeedPark',1200,'Bacau','anti-clockwise',7,400);

insert into circuit values(7,'Skat Kart',1500,'Tg. Secuiesc','clockwise',11,1000);

insert into circuit values(8,'Karting Genk',1800,'Genk','clockwise',12,200);

insert into circuit values(9,'Le Mans Karting Circuit',1600,'Le Mans','clockwise',12,920);

insert into circuit values(10,'Circuito Internazionale Napoli',1550,'Sarno','clockwise',10,900);

insert into kart values ( 917 , 55833 , 3461 , 1 );

insert into kart values ( 123 , 47542 , 5530 , 1 );

insert into kart values ( 756 , 48638 , 9719 , 1 );

insert into kart values ( 586 , 31208 , 4409 , 0 );

insert into kart values ( 635 , 98394 , 3510 , 1 );

insert into kart values ( 200 , 48098 , 8768 , 0 );

insert into kart values ( 603 , 59299 , 7472 , 1 );

insert into kart values ( 851 , 17147 , 4508 , 1 );

insert into kart values ( 499 , 75541 , 1696 , 0 );

insert into kart values ( 476 , 69855 , 9268 , 0 );

insert into kart values ( 787 , 30964 , 8533 , 1 );

insert into kart values ( 712 , 49895 , 2665 , 0 );

insert into kart values ( 751 , 85720 , 8275 , 1 );

insert into kart values ( 937 , 68907 , 9932 , 1 );

insert into kart values ( 603 , 63818 , 2534 , 0 );

insert into kart values ( 304 , 51755 , 9390 , 1 );

insert into kart values ( 763 , 30568 , 8831 , 0 );

insert into kart values ( 288 , 41193 , 8523 , 1 );

insert into kart values ( 456 , 69051 , 9787 , 0 );

insert into kart values ( 294 , 41213 , 3789 , 0 );

insert into kart values ( 958 , 65983 , 6970 , 1 );

insert into kart values ( 505 , 28137 , 3601 , 0 );

insert into kart values ( 245 , 93059 , 3377 , 0 );

insert into kart values ( 742 , 61745 , 1596 , 1 );

insert into kart values ( 963 , 69807 , 2673 , 1 );

insert into kart values ( 541 , 38017 , 1461 , 1 );

insert into kart values ( 461 , 63418 , 9146 , 0 );

insert into kart values ( 169 , 65918 , 6059 , 0 );

insert into kart values ( 872 , 42742 , 2023 , 0 );

insert into kart values ( 491 , 82365 , 9260 , 0 );

insert into kart values ( 913 , 89496 , 6846 , 1 );

insert into kart values ( 401 , 26990 , 1566 , 0 );

insert into kart values ( 386 , 87162 , 4086 , 0 );

insert into kart values ( 930 , 75083 , 4048 , 0 );

insert into kart values ( 150 , 74597 , 1880 , 0 );

insert into kart values ( 827 , 18928 , 9898 , 0 );

insert into kart values ( 763 , 72327 , 6652 , 1 );

insert into kart values ( 895 , 71097 , 9282 , 1 );

insert into kart values ( 185 , 16346 , 7126 , 1 );

insert into kart values ( 846 , 59401 , 3833 , 0 );

insert into kart values ( 365 , 65945 , 9782 , 1 );

insert into kart values ( 882 , 92634 , 3989 , 1 );

insert into kart values ( 238 , 71967 , 6811 , 0 );

insert into kart values ( 671 , 38430 , 4928 , 1 );

insert into kart values ( 818 , 33547 , 4739 , 1 );

insert into kart values ( 785 , 35960 , 9028 , 1 );

insert into kart values ( 790 , 73556 , 8003 , 0 );

insert into kart values ( 219 , 31246 , 8799 , 1 );

insert into kart values ( 811 , 53143 , 5620 , 0 );

insert into kart values ( 977 , 93806 , 8363 , 1 );

insert into kart values ( 234 , 76771 , 3740 , 1 );

insert into kart values ( 135 , 68879 , 4667 , 1 );

insert into kart values ( 309 , 17247 , 1075 , 1 );

insert into kart values ( 357 , 76292 , 3180 , 0 );

insert into kart values ( 382 , 95438 , 7328 , 0 );

insert into kart values ( 550 , 78240 , 8846 , 1 );

insert into echipa values ( 1 , 'Autosally' ,'Salistean' ,'12-Sep-2018' , 43351 , 'green' );

insert into echipa values ( 2 , 'Borsch Racing' , 'Borsch' ,'10-Jul-2017' , 85317 , 'yellow' );

insert into echipa values ( 3 , 'Real Racing' ,'Raileanu' ,'20-Apr-2014' , 56597 , 'yellow' );

insert into echipa values ( 4 , 'BRT' ,'Dima' ,'15-Feb-2012' , 36920 , 'yellow' );

insert into echipa values ( 5 , 'JMS Performance' ,'Jacques' ,'03-Mar-2013' , 60554 , 'yellow' );

insert into echipa values ( 6 , 'Grozea Racing' ,'Grozea' ,'23-Jun-2016' , 58606 , 'red' );

insert into echipa values ( 7 , 'Gulstar Racing' ,'Gulie' , '01-Jan-2011' , 68504 , 'red' );

insert into echipa values ( 8 , 'Toka Racing' ,'Dan' , '10-May-2019', 50773 , 'yellow');

insert into echipa values ( 9 , 'Drinago Racing' ,'Boriceanu' ,'29-Sep-2018' , 59752 , 'red' );

insert into sponsor values (1,'Vortex',50000,'Engines');

insert into sponsor values (2,'Levanto',30000,'Tires');

insert into sponsor values (3,'Alpinestars',40000,'Race equipment');

insert into sponsor values (4,'Motul',10000,'Oil');

insert into sponsor values (5,'Porsche Bucuresti Nord',10000,NULL);

insert into sponsor values (6,'Pro Refrigeration',8000,NULL);

insert into sponsor values (7,'Rovere',50000,NULL);

insert into sponsor values (8,'Metropola TV',NULL,'Streaming');

insert into etapa values( 1 ,'19-Apr-2023' ,'21-Apr-2023' , 1 , 8 );

insert into etapa values( 2 ,'01-May-2023' ,'03-May-2023', 1 , 3 );

insert into etapa values( 3 ,'12-May-2023' ,'14-May-2023', 1 , 10 );

insert into etapa values( 4 , '02-Aug-2023', '04-Aug-2023', 1 , 1 );

insert into etapa values( 5 ,'31-Aug-2023' ,'2-Sep-2023' , 1 , 2 );

insert into etapa values( 6 ,'03-Jul-2023' ,'05-Jul-2023' , 1 , 4 );

insert into etapa values( 7 ,'14-May-2023' , '16-May-2023', 2 , 3 );

insert into etapa values( 8 ,'25-Oct-2023' , '27-Oct-2023', 2 , 9 );

insert into etapa values( 9 , '22-Jul-2023', '24-Jul-2023', 2 , 5 );

insert into etapa values( 10 , '02-Aug-2023', '04-Aug-2023', 2 , 6 );

insert into etapa values( 11 , '15-Jun-2023', '15-Jun-2023', 2 , 7 );

insert into etapa values( 12 , '26-Jun-2023', '28-Jun-2023', 2 , 2 );

insert into etapa values( 13 ,'05-Jun-2023' , '07-Jun-2023', 2 , 1 );

insert into etapa values( 14 , '08-Apr-2023', '10-Apr-2023', 3 , 8 );

insert into etapa values( 15 , '24-Sep-2023', '26-Sep-2023', 3 , 6 );

insert into etapa values( 16 , '07-Oct-2023', '09-Oct-2023', 3 , 4 );

insert into etapa values( 17 , '27-May-2023', '29-May-2023', 3 , 7 );

insert into etapa values( 18 , '30-Sep-2023', '2-Oct-2023', 4 , 9 );

insert into etapa values( 19 , '16-Oct-2023', '18-Oct-2023', 4 , 2 );

insert into etapa values( 20 , '22-Jul-2023', '24-Jul-2023', 4 , 1 );

insert into etapa values( 21 , '09-Jul-2023', '11-Jul-2023', 4 , 8 );

insert into etapa values( 22 , '16-Oct-2023', '18-Oct-2023', 4 , 3 );

insert into etapa values( 23 , '02-Aug-2023', '04-Aug-2023', 4 , 5 );

insert into etapa values( 24 , '03-Jul-2023', '05-Jul-2023', 4 , 6 );

insert into etapa values( 25 , '10-Aug-2023', '12-Aug-2023', 5 , 1 );

insert into etapa values( 26 , '09-Jul-2023', '11-Jul-2023', 5 , 8 );

insert into etapa values( 27 , '24-Jun-2023', '26-Jun-2023', 5 , 10 );

insert into etapa values( 28 , '17-Sep-2023', '18-Sep-2023', 5 , 4 );

insert into etapa values( 29 , '15-Oct-2023', '17-Oct-2023', 5 , 5 );

insert into staff values( 1 ,' Benjamin ', 8 , '16-Aug-1995');

insert into staff values( 2 ,' Christopher ', 5 , '05-Jun-1978');

insert into staff values( 3 ,' Daniel ', 4 , '28-Mar-1987');

insert into staff values( 4 ,' Ethan ', 3 , '14-Oct-1999');

insert into staff values( 5 ,' Felix ', 6 , '23-Nov-1991');

insert into staff values( 6 ,' George ', 5 , '09-Feb-1982');

insert into staff values( 7 ,' Henry ', 2 , '19-Apr-1997');

insert into staff values( 8 ,' Isaac ', 6 , '11-Sep-2001');

insert into staff values( 9 ,' Jacob ', 9 , '07-Jul-1993');

insert into staff values( 10 ,' Kevin ', 9 , '26-May-1989');

insert into staff values( 11 ,' Liam ', 7 , '03-Dec-1977');

insert into staff values( 12 ,' Patrick ', 5 , '18-Jan-2002');

insert into staff values( 13 ,' Quentin ', 3 , '30-Jun-1990');

insert into staff values( 14 ,' Thomas ', 9 , '22-Oct-1985');

insert into staff values( 15 ,' Ulysses ', 5 , '12-Mar-1976');

insert into staff values( 16 ,' Victor ', 7 , '26-Apr-2003');

insert into staff values( 17 ,' Bogdan ', 1 , '27-Apr-2003');

insert into cursa values( 1 ,' static ', 12 , 1 );

insert into cursa values( 2 ,' static ', 15 , 15 );

insert into cursa values( 3 ,' static ', 15 , 26 );

insert into cursa values( 4 ,' static ', 6 , 19 );

insert into cursa values( 5 ,' static ', 9 , 9 );

insert into cursa values( 6 ,' motion ', 13 , 1 );

insert into cursa values( 7 ,' static ', 7 , 14 );

insert into cursa values( 8 ,' motion ', 7 , 9 );

insert into cursa values( 9 ,' motion ', 13 , 17 );

insert into cursa values( 10 ,' motion ', 14 , 3 );

insert into cursa values( 11 ,' motion ', 6 , 4 );

insert into cursa values( 12 ,' static ', 8 , 13 );

insert into cursa values( 13 ,' static ', 11 , 26 );

insert into cursa values( 14 ,' motion ', 13 , 21 );

insert into cursa values( 15 ,' static ', 14 , 25 );

insert into cursa values( 16 ,' static ', 7 , 22 );

insert into cursa values( 17 ,' static ', 9 , 23 );

insert into cursa values( 18 ,' motion ', 8 , 4 );

insert into cursa values( 19 ,' motion ', 6 , 3 );

insert into cursa values( 20 ,' static ', 13 , 5 );

insert into calificari values( 1 ,'00:59:482', 12 , 13 );

insert into calificari values( 2 ,'01:03:217', 8 , 28 );

insert into calificari values( 3 ,'00:56:775', 9 , 5 );

insert into calificari values( 4 ,'01:08:934', 12 , 7 );

insert into calificari values( 5 ,'01:01:642', 9 , 11 );

insert into calificari values( 6 ,'01:06:873', 11 , 8 );

insert into calificari values( 7 ,'01:02:311', 9 , 1 );

insert into calificari values( 8 ,'01:04:568', 8 , 28 );

insert into calificari values( 9 ,'01:07:189', 9 , 25 );

insert into calificari values( 10 ,'01:00:912', 8 , 17 );

insert into antrenament values( 1 ,' liber ', 12 , 17 );

insert into antrenament values( 2 ,' eliminatoriu ', 12 , 27 );

insert into antrenament values( 3 ,' calificativ ', 12 , 29 );

insert into antrenament values( 4 ,' liber ', 11 , 6 );

insert into antrenament values( 5 ,' calificativ ', 8 , 14 );

insert into antrenament values( 6 ,' eliminatoriu ', 9 , 28 );

insert into antrenament values( 7 ,' liber ', 10 , 21 );

insert into antrenament values( 8 ,' liber ', 10 , 16 );

insert into antrenament values( 9 ,' calificativ ', 12 , 18 );

insert into antrenament values( 10 ,' eliminatoriu ', 10);

insert into pilot values( 1 , 'Andrei' , 917, 1 , '12-Sep-2011' ,' Promo ', 7 ,37);

insert into pilot values( 2 , 'Alexandru' ,123, 11 , '03-Jun-2004' ,' Micro ', 16, 126);

insert into pilot values( 3 , 'Ioan' , 756, 4 , '18-Jan-2015' ,' KZ ', 16 ,230);

insert into pilot values( 4 , 'Florin' , 635, 12 , '07-Oct-2007' ,' Micro ', 2, 233);

insert into pilot values( 5 , 'Adrian' , 603, 7 , '29-Nov-2008' ,' Micro ', 8, 176);

insert into pilot values( 6 , 'Mihai' , 851, 13 , '14-Feb-2016' ,' Mini ', 7, 15);

insert into pilot values( 7 , 'Ionut' , 977, 6 , '26-May-2013' ,' Promo ', 4, 66);

insert into pilot values( 8 , 'Radu' , 937, 10, '09-Nov-2014' ,' Promo ', 1, 24);

insert into pilot values( 9 , 'Stefan' , 288, 2 , '05-Jul-2012' ,' Micro ', 8 ,232);

insert into pilot values( 10 , 'Mircea' , 963, 15 , '23-May-2006' ,' Promo ', 9, 182);

insert into pilot values( 11 , 'Bogdan' , 895, 14 , '08-Dec-2010' ,' Senior ', 6, 211);

insert into pilot values( 12 , 'Valentin' ,763, 3 , '30-Mar-2017' ,' Micro ', 6 ,159);

insert into pilot values( 13 , 'Cosmin' , 550, 8 , '20-Aug-2005' ,' Micro ', 9 ,87);

insert into pilot values( 14 , 'Cristian' ,135, 16 , '01-Apr-2009' ,' Junior ', 4, 246);

insert into pilot values( 15 , 'Ilinca' , 234, 5 , '11-Jul-2013' ,' KZ ', 14 ,239);

insert into pilot values( 16 , 'Sofia' , 958, 9, '17-Oct-2016' ,' Promo ', 7, 205);

insert into pilot values( 17 , 'Maria' , 357, 17,'17-Oct-2014' ,' KZ ', 16, 230);

insert into sponsorizeaza values( 1 , 8 , 8672 ,'22-Mar-2023', 8 );

insert into sponsorizeaza values( 1 , 1 , 32560 ,'9-Jan-2023', 14 );

insert into sponsorizeaza values( 2 , 3 , 13159 ,'3-Feb-2023', 19 );

insert into sponsorizeaza values( 2 , 6 , 20668 ,'28-Jan-2023', 17 );

insert into sponsorizeaza values( 2 , 8 , 10957 ,'10-Mar-2023', 10 );

insert into sponsorizeaza values( 2 , 7 , 29234 ,'15-Apr-2023', 9 );

insert into sponsorizeaza values( 3 , 7 , 45708 ,'18-Feb-2023', 19 );

insert into sponsorizeaza values( 3 , 5 , 38658 ,'5-May-2023', 13 );

insert into sponsorizeaza values( 5 , 2 , 33402 ,'11-Jan-2023', 11 );

insert into sponsorizeaza values( 5 , 5 , 12367 ,'23-Mar-2023', 19 );

insert into sponsorizeaza values( 5 , 8 , 43000 ,'4-Apr-2023', 14 );

insert into sponsorizeaza values( 5 , 4 , 10600 ,'14-Feb-2023', 13 );

insert into sponsorizeaza values( 4 , 4 , 26520 ,'27-Jan-2023', 16 );

insert into sponsorizeaza values( 4 , 2 , 45772 ,'7-May-2023', 10 );

insert into ocupa values( 1 , 2 ,'18-Apr-2016');

insert into ocupa values( 1 , 6 ,'06-Sep-2019');

insert into ocupa values( 2 , 3 ,'09-Jan-2014');

insert into ocupa values( 2 , 5 ,'27-Mar-2018');

insert into ocupa values( 3 , 2 ,'14-Jul-2012');

insert into ocupa values( 4 , 3 ,'05-Oct-2015');

insert into ocupa values( 5 , 1 ,'22-Feb-2021');

insert into ocupa values( 6 , 3 ,'10-Nov-2013');

insert into ocupa values( 7 , 5 ,'31-May-2020');

insert into ocupa values( 8 , 1 ,'07-Dec-2017');

insert into ocupa values( 9 , 4 ,'26-Jun-2019');

insert into ocupa values( 9 , 3 ,'02-Aug-2014');

insert into ocupa values( 10 , 1 ,'19-Oct-2016');

insert into ocupa values( 11 , 6 ,'13-Jan-2023');

insert into ocupa values( 12 , 2 ,'08-Mar-2015');

insert into ocupa values( 12 , 6 ,'16-Nov-2012');

insert into ocupa values( 13 , 5 ,'25-Apr-2022');

insert into ocupa values( 14 , 4 ,'11-Sep-2018');

insert into ocupa values( 15 , 4 ,'30-Jul-2021');

insert into ocupa values( 16 , 2 ,'04-Feb-2017');

insert into participa values( 2 , 5 ,'15-Jun-2022',5);

insert into participa values( 1 , 1 ,'29-Aug-2023',6);

insert into participa values( 3 , 6 ,'11-Nov-2022',3);

insert into participa values( 3 , 1 ,'07-Jul-2021',10);

insert into participa values( 2 , 2 ,'23-Oct-2021',4);

insert into participa values( 8 , 3 ,'18-May-2022',7);

insert into participa values( 4 , 1 ,'04-Sep-2022',2);

insert into participa values( 1 , 2 ,'26-Jun-2021',5);

insert into participa values( 9 , 1 ,'09-Aug-2021',3);

insert into participa values( 3 , 4 ,'17-Nov-2023',1);

insert into participa values( 4 , 2 ,'02-Jul-2023',7);

insert into participa values( 8 , 1 ,'14-Oct-2022',6);

insert into participa values( 3 , 3 ,'30-May-2023',6);

insert into participa values( 2 , 4 ,'08-Aug-2022',8);

insert into participa values( 8 , 6 ,'21-Nov-2021',9);

insert into participa values( 8 , 2 ,'25-Jul-2023',10);

insert into participa values( 6 , 6 ,'06-Jun-2022',4);

insert into participa values( 6 , 2 ,'12-Oct-2021',6);

insert into participa values( 7 , 6 ,'01-Sep-2023',9);

insert into participa values( 5 , 1 ,'19-May-2021',2);

insert into participa values( 5 , 7 ,'18-May-2021',4);

insert into participa values( 6 , 11 ,'13-Oct-2021',5);

insert into participa values( 4 , 19 ,'02-Jul-2023',7);

insert into jobs values(1,'Principal',5,300);

insert into jobs values(2,'Trainer',3,150);

insert into jobs values(3,'Mechanic',1,100);

insert into jobs values(4,'Mechanic helper',0,50);

insert into jobs values(5,'Telemetry analyst',3,200);

insert into jobs values(6,'Psychologist',1,125);

# Subprogram cu 3 colectii

**--**Subprogramul afiseaza etapele cu numar impar si echipele care au participat la acestea, impreuna cu numarul total de ture parcurs pentru fiecare etapa. (tabel indexat -> numere impare, tabel imbricate -> echipe, vector -> cursele)

create or replace procedure ex6 as

type tabel\_index is table of number index by pls\_integer;

type tabel\_echipe is table of varchar2(100);

type array\_ture is varray(100) of number;

etape tabel\_index;

echipe tabel\_echipe := tabel\_echipe();

ture array\_ture := array\_ture();

v\_ture number;

cursor iduri is

SELECT id\_etapa

FROM etapa

WHERE mod(id\_etapa,2) = 1;

begin

for etapa in iduri loop

etape(etape.count + 1) := etapa.id\_etapa;

end loop;

for i in 1..etape.count loop

for ec in

(select nume\_echipa

from echipa e, participa p

where e.id\_echipa = p.id\_echipa

and p.id\_etapa = etape(i)

and p.nr\_piloti >= 0)

loop

echipe.extend;

echipe(echipe.last) := ec.nume\_echipa;

end loop;

for nr\_ture in

(select numar\_total\_ture

from cursa

where id\_etapa = etape(i))

loop

ture.extend;

ture(ture.last) := nr\_ture.numar\_total\_ture;

end loop;

if echipe.count > 0 then

DBMS\_output.put('La etapa ' || etape(i) || ' au participat echipele: ');

for i in 1..echipe.count loop

DBMS\_output.put(echipe(i) || ' - ');

end loop;

v\_ture := 0;

for i in 1..ture.count loop

v\_ture := v\_ture + ture(i);

end loop;

dbms\_output.put\_line('si au parcurs ' || v\_ture || ' ture de circuit');

end if;

ture.delete;

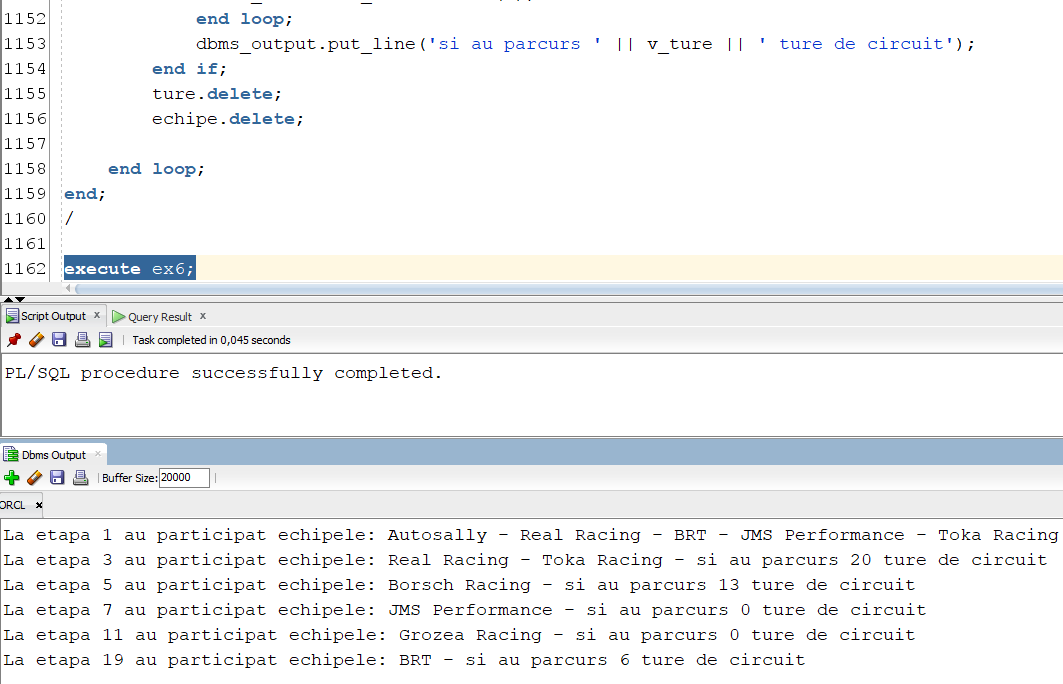
echipe.delete;

end loop;

end;

/

execute ex6;



# 

# 2 Cursoare

CREATE OR REPLACE PROCEDURE ex7 AS

TYPE refcursor is ref cursor;

angajati refcursor;

v\_culoare echipa.main\_color%TYPE := 'yellow';

v\_echipa echipa.nume\_echipa%TYPE;

v\_staff staff.nume\_staff%type;

CURSOR ec(culoare VARCHAR2) IS

SELECT e.nume\_echipa,

CURSOR (

SELECT s.nume\_staff

FROM staff s

WHERE s.id\_echipa = e.id\_echipa

)

FROM echipa e

WHERE e.main\_color = culoare;

BEGIN

OPEN ec(v\_culoare);

LOOP

FETCH ec into v\_echipa,angajati;

EXIT WHEN ec%NOTFOUND;

DBMS\_OUTPUT.PUT('Echipa ' || v\_echipa || ' este formata din: ');

LOOP

fetch angajati into v\_staff;

exit when angajati%notfound;

DBMS\_OUTPUT.PUT(v\_staff || ' ');

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('');

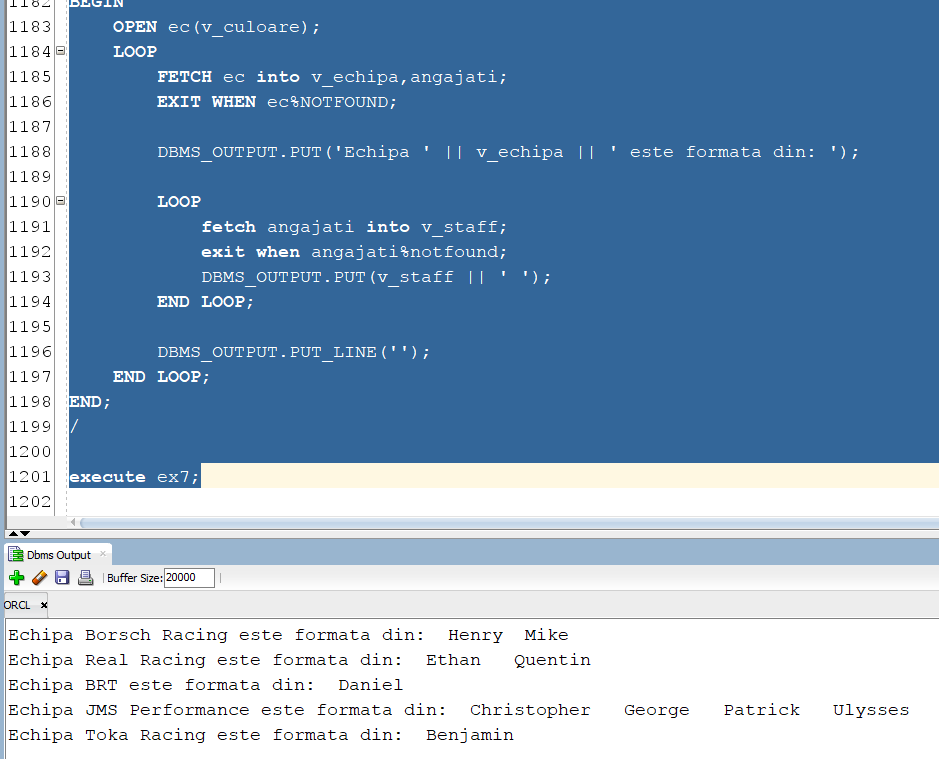
END LOOP;

CLOSE ec;

END;

/

execute ex7;



# Minim 3 tabele

(pilot + staff + participa + etapa)

--problema returneaza ‘da’ sau ‘ nu + “eroarea” ’ daca datele pentru o anumita etapa sunt incorecte (data etapei este gresita sau exista o echipa ce a inscris mai multi piloti decat are). De asemenea un pilot este valid daca acesta are un kart principal atribuit.

create or replace function ex8 return varchar2 is

type tabel\_nr\_piloti is table of varchar2(100);

wrong\_date exception;

bad\_register exception;

nr\_piloti tabel\_nr\_piloti := tabel\_nr\_piloti();

v\_echipa echipa.id\_echipa%type;

v\_etapa etapa.id\_etapa%type;

data\_inc etapa.data\_inceput%type;

data\_fin etapa.data\_final%type;

nr\_p number;

cursor ec is

select id\_echipa,count(p.id\_pilot)

from pilot p, staff s, kart k

where p.id\_staff = s.id\_staff

and k.numar\_kart = p.numar\_kart

and k.main\_kart = 1

group by id\_echipa

order by 1;

cursor et is

select id\_etapa, data\_inceput, data\_final

from etapa;

begin

for i in (select id\_echipa from echipa) loop

nr\_piloti.extend;

end loop;

open ec;

loop

fetch ec into v\_echipa, nr\_p;

exit when ec%notfound;

nr\_piloti(v\_echipa) := nr\_p;

end loop;

close ec;

open et;

loop

fetch et into v\_etapa, data\_inc, data\_fin;

exit when et%notfound;

if data\_inc > data\_fin then

raise wrong\_date;

end if;

end loop;

close et;

for round in (select id\_etapa from etapa) loop

for i in 1..nr\_piloti.count loop

select nr\_piloti

into nr\_p

from participa

where id\_etapa = round.id\_etapa

and id\_echipa = i;

if nr\_p > nr\_piloti(i) then

raise bad\_register;

end if;

end loop;

end loop;

return 'da';

exception

when wrong\_date then

return 'no, invalid date';

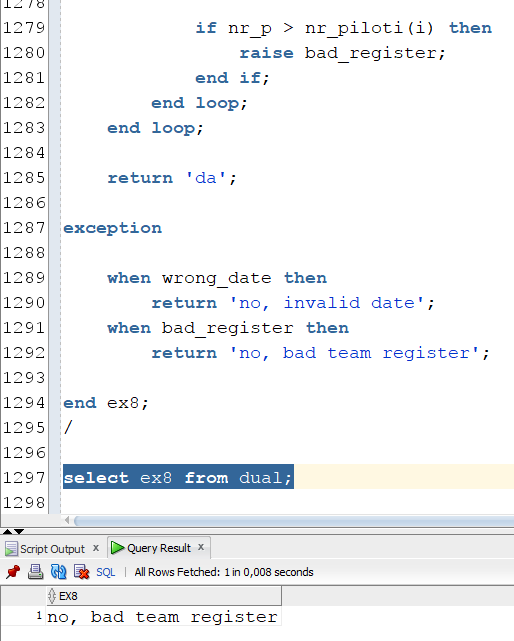
when bad\_register then

return 'no, bad team register';

end ex8;

/

select ex8 from dual;



# Minim 5 tabele

--(sponsorizeaza, campionat\_karting,etapa,circuit,participa)

Cererea calculeaza pentru fiecare etapa suma de bani primita de la sponsori adunata la bugetul campionatului. Pentru fiecare etapa din cadrul unui campionat se calculeaza costul impus de circuit(, 5 \* lungime \* latimea maxima). De asemenea, este calculat si venitul produs din inscrierea pilotilor la acea etapa(400\* numarul total de piloti). La final se calculeaza profitul prin diferenta dintre bugetul total si cheltuielile totale.

create or replace procedure ex9 as

TYPE t\_result IS RECORD (

id\_campionat NUMBER,

tbuget NUMBER,

cost\_total NUMBER,

profit NUMBER

);

TYPE t\_matrix IS TABLE OF t\_result;

v\_results t\_matrix;

nr\_campionate number;

NO\_DATA\_FOUND exception;

TOO\_MANY\_ROWS exception;

even exception;

loss exception;

cursor profitCampionat is

select bu.id\_campionat,tbuget,cost\_total,tbuget-cost\_total as profit

from (select s.id\_campionat,sum(s.suma)+ k.buget as tbuget

from sponsorizeaza s, campionat\_karting k

where k.id\_campionat = s.id\_campionat

group by s.id\_campionat,k.buget) bu,

(select id\_campionat,sum(cost\_etapa) as cost\_total

from (select e.id\_campionat,e.id\_etapa,tcost-tnrp as cost\_etapa

from etapa e,(select id\_circuit, 5 \* lungime \* latime\_max as tcost

from circuit) c,(select id\_etapa, 400\*sum(nr\_piloti) as tnrp

from participa

group by id\_etapa) p

where e.id\_circuit = c.id\_circuit

and p.id\_etapa = e.id\_etapa)

group by id\_campionat) co

where co.id\_campionat = bu.id\_campionat

order by 1;

begin

select count(\*)

into nr\_campionate

from campionat\_karting;

OPEN profitCampionat;

FETCH profitCampionat BULK COLLECT INTO v\_results;

CLOSE profitCampionat;

IF v\_results.COUNT = 0 THEN

raise no\_data\_found;

elsif v\_results.count > nr\_campionate then

raise too\_many\_rows;

else

FOR i IN v\_results.FIRST .. v\_results.LAST LOOP

if v\_results(i).profit = 0 then

dbms\_output.put('ID\_CAMPIONAT: ' || v\_results(i).id\_campionat || ' ');

raise even;

elsif v\_results(i).profit < 0 then

dbms\_output.put('ID\_CAMPIONAT: ' || v\_results(i).id\_campionat || ' ');

raise loss;

end if;

DBMS\_OUTPUT.PUT\_LINE(

'ID\_CAMPIONAT: ' || v\_results(i).id\_campionat ||

' | TOTAL\_BUGET: ' || v\_results(i).tbuget ||

' | COST\_TOTAL: ' || v\_results(i).cost\_total ||

' | PROFIT: ' || v\_results(i).profit

);

END LOOP;

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au găsit date.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe înregistrări găsite.');

when even then

dbms\_output.put\_line('Campionatul nu a produs profit');

when loss then

dbms\_output.put\_line('Campionatul este in pierdere');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare a apărut: ' || SQLERRM);

END ex9;

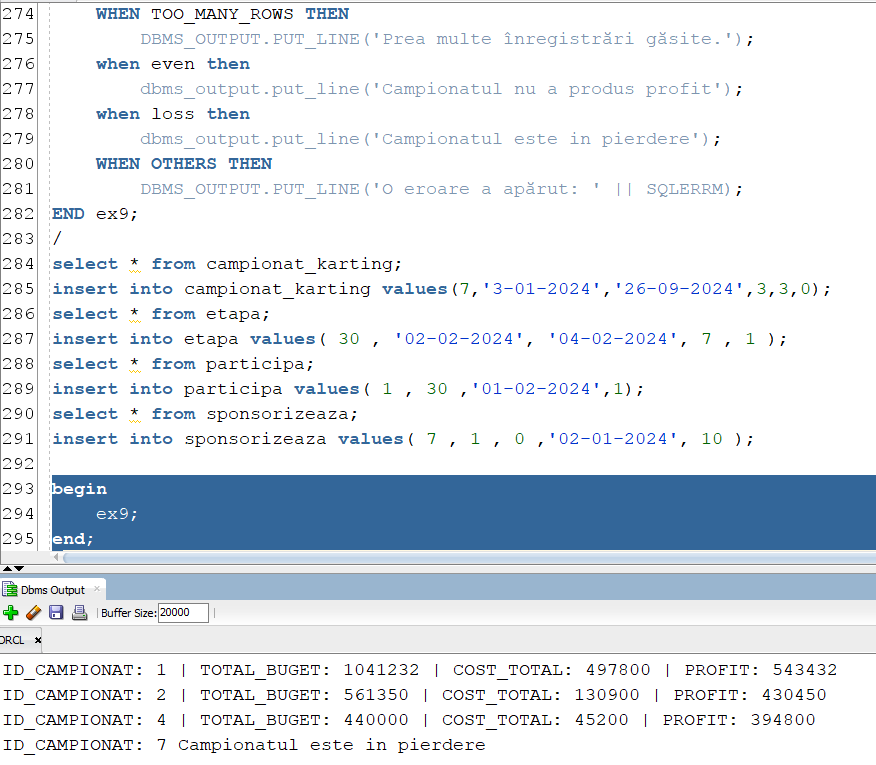
/

insert into campionat\_karting values(7,'3-01-2024','26-09-2024',3,3,0);

insert into etapa values( 30 , '02-02-2024', '04-02-2024', 7 , 1 );

insert into participa values( 1 , 30 ,'01-02-2024',1);

insert into sponsorizeaza values( 7 , 1 , 0 ,'02-01-2024', 10 );



# Minim 5 tabele (v2)

Procedura afiseaza directorii de echipa care sunt inregistrati ca angajati de tipul ‘principal’, impreuna cu pilotul pe care il antreneaza.

create or replace procedure ex9\_v2 as

TYPE t\_result IS RECORD (

nume\_director varchar2(100),

nume\_staff varchar2(100),

nume\_echipa varchar2(100),

nume\_pilot varchar2(100)

);

TYPE t\_matrix IS TABLE OF t\_result;

v\_results t\_matrix;

nr\_staff number;

wrong\_name exception;

cursor director is

select e.nume\_director,s.nume\_staff,e.nume\_echipa , p.nume\_pilot

from echipa e, staff s, ocupa o, jobs j, pilot p

where s.id\_echipa = e.id\_echipa

and s.id\_staff = o.id\_staff

and o.id\_job = j.id\_job

and trim(lower(job\_name)) = 'principal'

and s.id\_staff = p.id\_staff;

begin

select count(\*)

into nr\_staff

from staff;

OPEN director;

FETCH director BULK COLLECT INTO v\_results;

CLOSE director;

IF v\_results.COUNT = 0 THEN

raise no\_data\_found;

elsif v\_results.count > nr\_staff then

raise too\_many\_rows;

else

FOR i IN v\_results.FIRST .. v\_results.LAST LOOP

if v\_results(i).nume\_director <> v\_results(i).nume\_staff then

raise wrong\_name;

end if;

dbms\_output.put\_line('Nume director: '|| v\_results(i).nume\_director ||

', nume echipa: '|| v\_results(i).nume\_echipa ||

', nume pilot: '|| v\_results(i).nume\_pilot);

END LOOP;

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au găsit date.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe înregistrări găsite.');

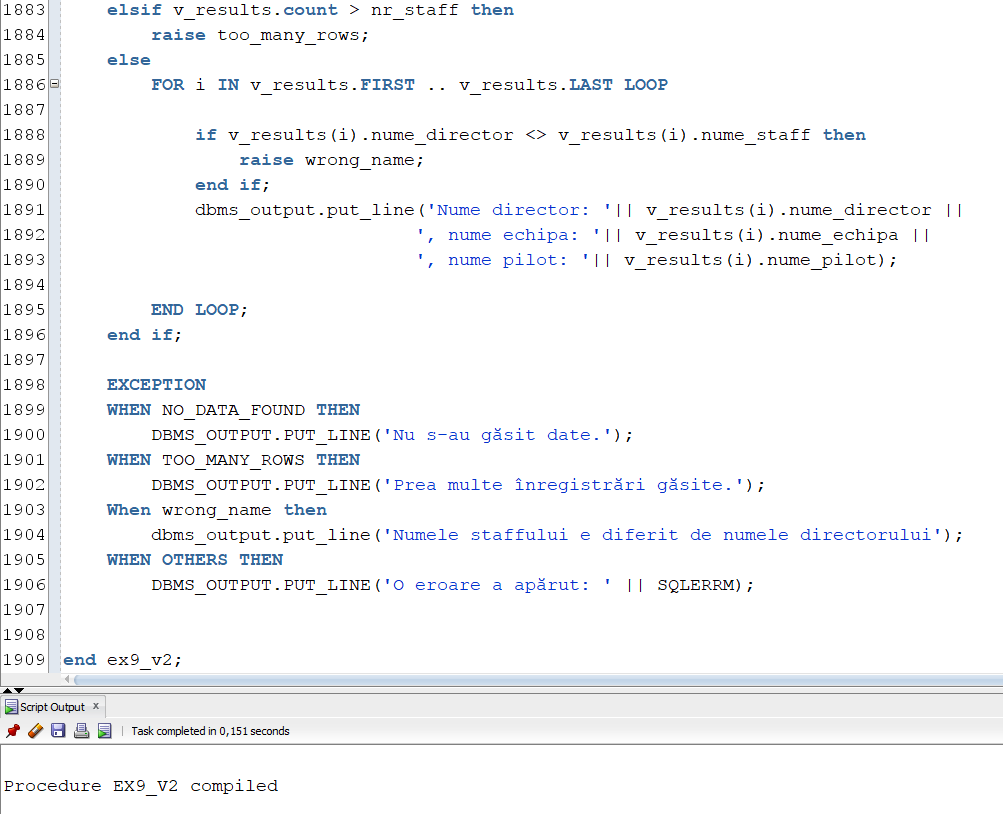
When wrong\_name then

dbms\_output.put\_line('Numele staffului e diferit de numele directorului');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare a apărut: ' || SQLERRM);

end ex9\_v2;

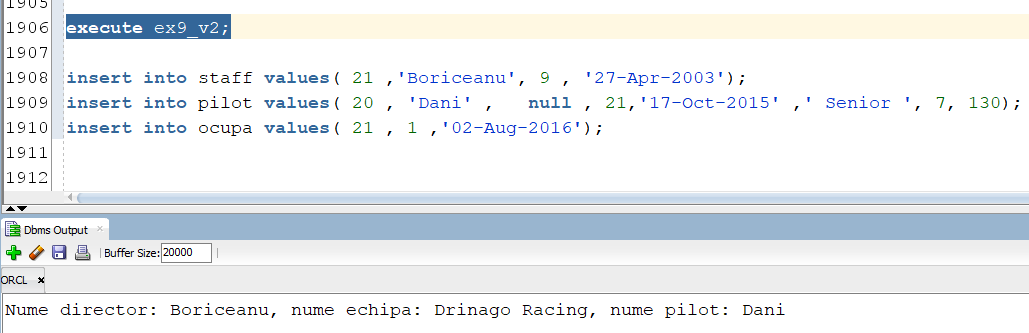
/

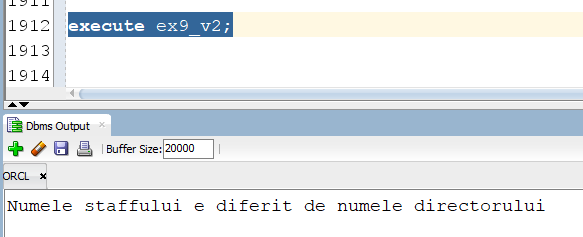
--dupa inserare:

insert into staff values( 21 ,'Boriceanu', 9 , '27-Apr-2003');

insert into pilot values( 20 , 'Dani' , null , 21,'17-Oct-2015' ,' Senior ', 7, 130);

insert into ocupa values( 21 , 1 ,'02-Aug-2016');





# Trigger LMD la nivel de comanda

Triggerul nu permite inscrierea a mai mult de 9 echipe la nivel national.

CREATE OR REPLACE TRIGGER nr\_max\_echipe

BEFORE INSERT ON echipa

DECLARE

v\_nr\_echipe number(2);

BEGIN

select count(\*)

into v\_nr\_echipe

from echipa;

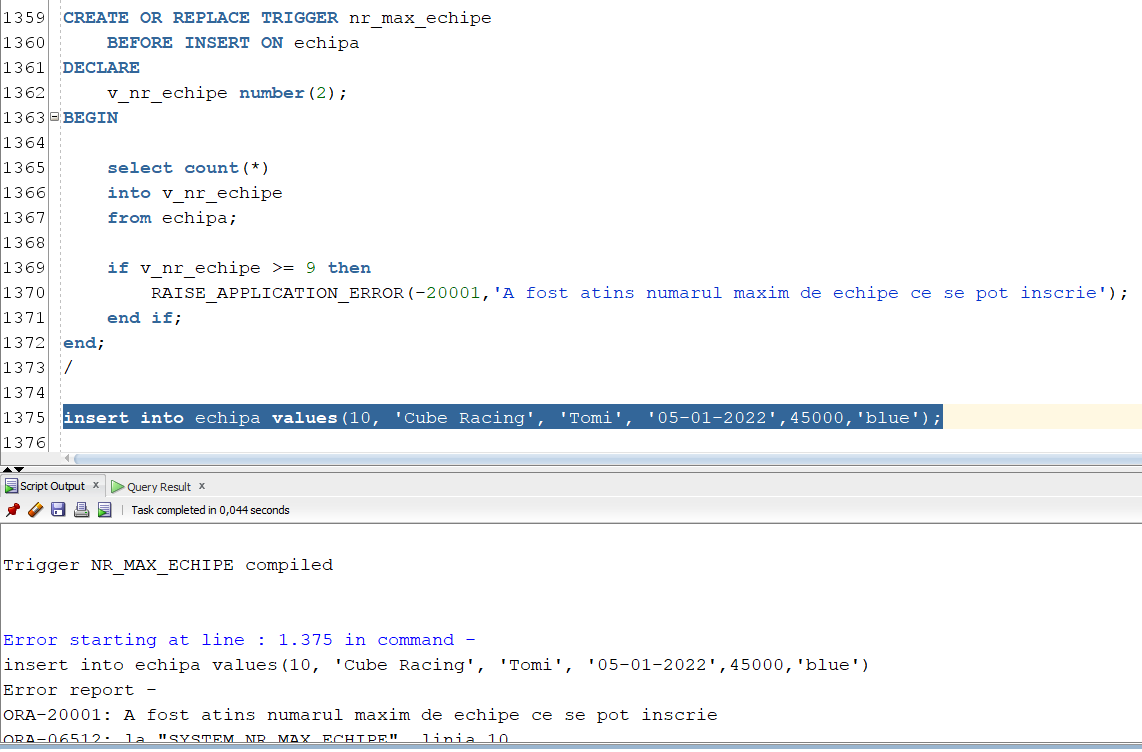
if v\_nr\_echipe >= 9 then

RAISE\_APPLICATION\_ERROR(-20001,'A fost atins numarul maxim de echipe ce se pot inscrie');

end if;

end;

/

insert into echipa values(10, 'Cube Racing', 'Tomi', '05-01-2022',45000,'blue'); 

# Trigger LMD la nivel de linie

Triggerul nu permite inserarea a mai mult de 3 piloti in cadrul aceleiasi categorii.

create or replace trigger limita\_categorie

before insert on pilot

for each row

declare

v\_nr number;

v\_categorie varchar2(100);

begin

v\_categorie := TRIM(LOWER(:new.categorie));

select count(\*)

into v\_nr

from pilot

where trim(lower(categorie)) = v\_categorie;

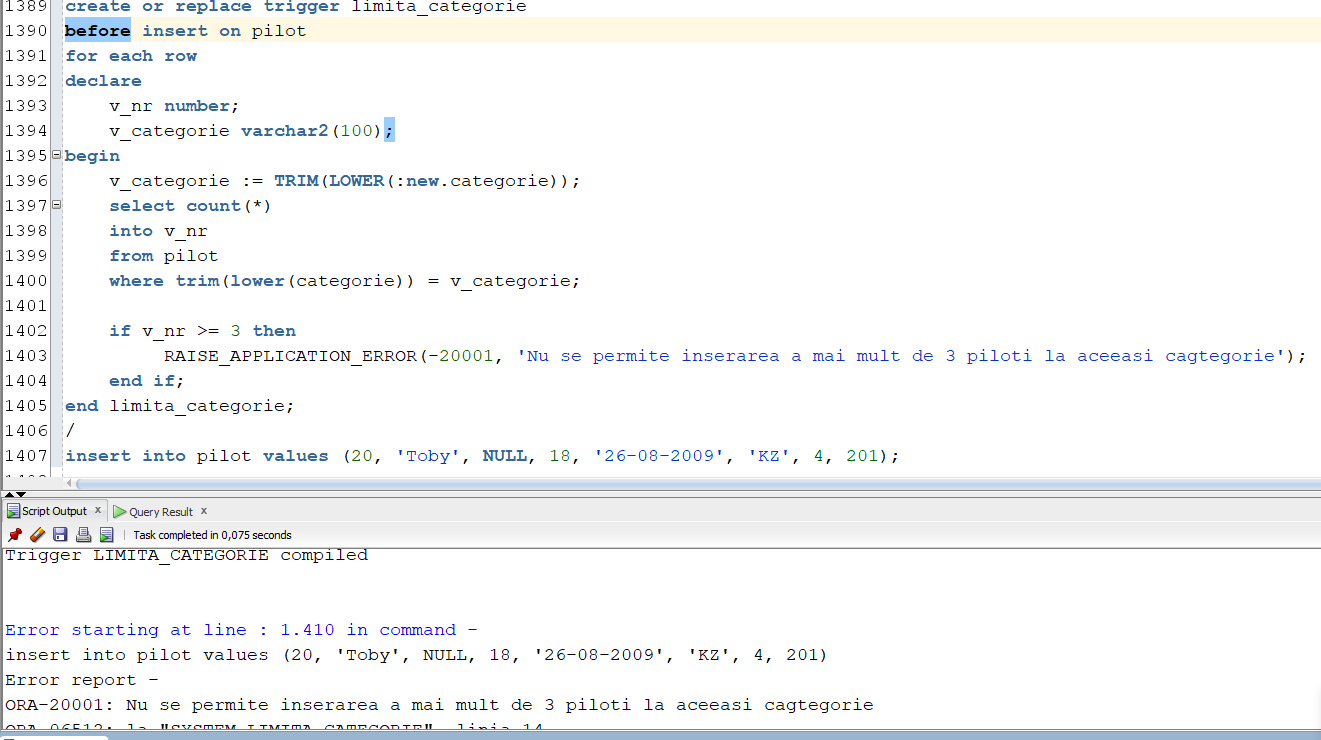
if v\_nr >= 3 then

RAISE\_APPLICATION\_ERROR(-20001, 'Nu se permite inserarea a mai mult de 3 piloti la aceeasi cagtegorie');

end if;

end limita\_categorie;

/

insert into pilot values (20, 'Toby', NULL, 18, '26-08-2009', 'KZ', 4, 201); 

# Trigger LDD

Triggerul nu permite stergerea sau alterarea unei tabele atunci cand este activ.

CREATE OR REPLACE TRIGGER safety

BEFORE DROP OR ALTER ON DATABASE

DECLARE

BEGIN

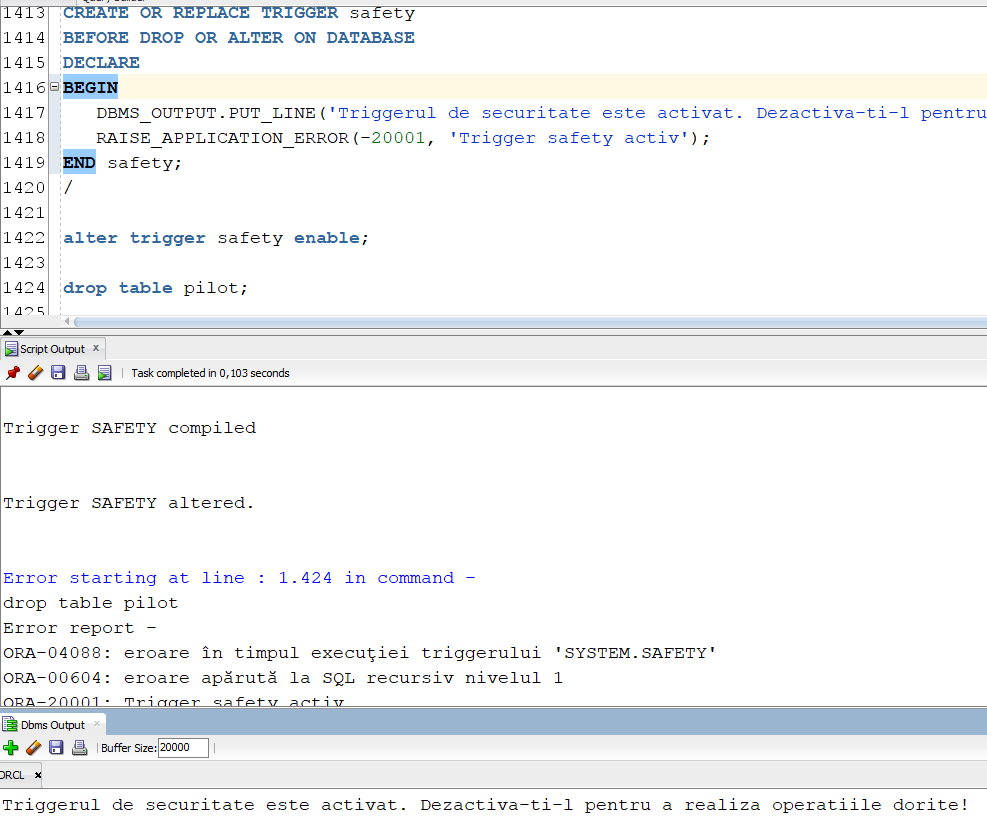
DBMS\_OUTPUT.PUT\_LINE('Triggerul de securitate este activat. Dezactiva-ti-l pentru a realiza operatiile dorite!');

RAISE\_APPLICATION\_ERROR(-20001, 'Trigger safety activ');

END safety;

/

alter trigger safety enable;

drop table pilot; 

# Pachet cu toate obiectele proiectului

create or replace package pachet\_karting as

--ex6

procedure ex6;

--ex7

procedure ex7;

--ex8

function ex8 return varchar2;

--ex9

procedure ex9;

procedure ex9\_v2;

end pachet\_karting;

--#################################################

create or replace package body pachet\_karting as

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

--ex6-----------------------------------------------------------------------------------------------------------

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

procedure ex6 as

type tabel\_index is table of number index by pls\_integer;

type tabel\_echipe is table of varchar2(100);

type array\_ture is varray(100) of number;

etape tabel\_index;

echipe tabel\_echipe := tabel\_echipe();

ture array\_ture := array\_ture();

v\_ture number;

cursor iduri is

SELECT id\_etapa

FROM etapa

WHERE mod(id\_etapa,2) = 1;

begin

for etapa in iduri loop

etape(etape.count + 1) := etapa.id\_etapa;

end loop;

for i in 1..etape.count loop

for ec in

(select nume\_echipa

from echipa e, participa p

where e.id\_echipa = p.id\_echipa

and p.id\_etapa = etape(i)

and p.nr\_piloti >= 0)

loop

echipe.extend;

echipe(echipe.last) := ec.nume\_echipa;

end loop;

for nr\_ture in

(select numar\_total\_ture

from cursa

where id\_etapa = etape(i))

loop

ture.extend;

ture(ture.last) := nr\_ture.numar\_total\_ture;

end loop;

if echipe.count > 0 then

DBMS\_output.put('La etapa ' || etape(i) || ' au participat echipele: ');

for i in 1..echipe.count loop

DBMS\_output.put(echipe(i) || ' - ');

end loop;

v\_ture := 0;

for i in 1..ture.count loop

v\_ture := v\_ture + ture(i);

end loop;

dbms\_output.put\_line('si au parcurs ' || v\_ture || ' ture de circuit');

end if;

ture.delete;

echipe.delete;

end loop;

end ex6;

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

--Ex7-----------------------------------------------------------------------------------------------------------

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

PROCEDURE ex7 AS

TYPE refcursor is ref cursor;

angajati refcursor;

v\_culoare echipa.main\_color%TYPE := 'yellow';

v\_echipa echipa.nume\_echipa%TYPE;

v\_staff staff.nume\_staff%type;

CURSOR ec(culoare VARCHAR2) IS

SELECT e.nume\_echipa,

CURSOR (

SELECT s.nume\_staff

FROM staff s

WHERE s.id\_echipa = e.id\_echipa

)

FROM echipa e

WHERE e.main\_color = culoare;

BEGIN

OPEN ec(v\_culoare);

LOOP

FETCH ec into v\_echipa,angajati;

EXIT WHEN ec%NOTFOUND;

DBMS\_OUTPUT.PUT('Echipa ' || v\_echipa || ' este formata din: ');

LOOP

fetch angajati into v\_staff;

exit when angajati%notfound;

DBMS\_OUTPUT.PUT(v\_staff || ' ');

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('');

END LOOP;

CLOSE ec;

END ex7;

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

--ex8-----------------------------------------------------------------------------------------------------------

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

function ex8 return varchar2 is

type tabel\_nr\_piloti is table of varchar2(100);

wrong\_date exception;

bad\_register exception;

nr\_piloti tabel\_nr\_piloti := tabel\_nr\_piloti();

v\_echipa echipa.id\_echipa%type;

v\_etapa etapa.id\_etapa%type;

data\_inc etapa.data\_inceput%type;

data\_fin etapa.data\_final%type;

nr\_p number;

cursor ec is

select id\_echipa,count(p.id\_pilot)

from pilot p, staff s, kart k

where p.id\_staff = s.id\_staff

and k.numar\_kart = p.numar\_kart

and k.main\_kart = 1

group by id\_echipa

order by 1;

cursor et is

select id\_etapa, data\_inceput, data\_final

from etapa;

begin

for i in (select id\_echipa from echipa) loop

nr\_piloti.extend;

end loop;

open ec;

loop

fetch ec into v\_echipa, nr\_p;

exit when ec%notfound;

nr\_piloti(v\_echipa) := nr\_p;

end loop;

close ec;

open et;

loop

fetch et into v\_etapa, data\_inc, data\_fin;

exit when et%notfound;

if data\_inc > data\_fin then

raise wrong\_date;

end if;

end loop;

close et;

for round in (select id\_etapa from etapa) loop

for i in 1..nr\_piloti.count loop

select nr\_piloti

into nr\_p

from participa

where id\_etapa = round.id\_etapa

and id\_echipa = i;

if nr\_p > nr\_piloti(i) then

raise bad\_register;

end if;

end loop;

end loop;

return 'da';

exception

when wrong\_date then

return 'no, invalid date';

when bad\_register then

return 'no, bad team register';

end ex8;

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

--ex9-----------------------------------------------------------------------------------------------------------

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

procedure ex9 as

TYPE t\_result IS RECORD (

id\_campionat NUMBER,

tbuget NUMBER,

cost\_total NUMBER,

profit NUMBER

);

TYPE t\_matrix IS TABLE OF t\_result;

v\_results t\_matrix;

nr\_campionate number;

NO\_DATA\_FOUND exception;

TOO\_MANY\_ROWS exception;

even exception;

loss exception;

cursor profitCampionat is

select bu.id\_campionat,tbuget,cost\_total,tbuget-cost\_total as profit

from (select s.id\_campionat,sum(s.suma)+ k.buget as tbuget

from sponsorizeaza s, campionat\_karting k

where k.id\_campionat = s.id\_campionat

group by s.id\_campionat,k.buget) bu,

(select id\_campionat,sum(cost\_etapa) as cost\_total

from (select e.id\_campionat,e.id\_etapa,tcost-tnrp as cost\_etapa

from etapa e,(select id\_circuit, 5 \* lungime \* latime\_max as tcost

from circuit) c,(select id\_etapa, 400\*sum(nr\_piloti) as tnrp

from participa

group by id\_etapa) p

where e.id\_circuit = c.id\_circuit

and p.id\_etapa = e.id\_etapa)

group by id\_campionat) co

where co.id\_campionat = bu.id\_campionat

order by 1;

begin

select count(\*)

into nr\_campionate

from campionat\_karting;

OPEN profitCampionat;

FETCH profitCampionat BULK COLLECT INTO v\_results;

CLOSE profitCampionat;

IF v\_results.COUNT = 0 THEN

raise no\_data\_found;

elsif v\_results.count > nr\_campionate then

raise too\_many\_rows;

else

FOR i IN v\_results.FIRST .. v\_results.LAST LOOP

if v\_results(i).profit = 0 then

dbms\_output.put('ID\_CAMPIONAT: ' || v\_results(i).id\_campionat || ' ');

raise even;

elsif v\_results(i).profit < 0 then

dbms\_output.put('ID\_CAMPIONAT: ' || v\_results(i).id\_campionat || ' ');

raise loss;

end if;

DBMS\_OUTPUT.PUT\_LINE(

'ID\_CAMPIONAT: ' || v\_results(i).id\_campionat ||

' | TOTAL\_BUGET: ' || v\_results(i).tbuget ||

' | COST\_TOTAL: ' || v\_results(i).cost\_total ||

' | PROFIT: ' || v\_results(i).profit

);

END LOOP;

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au găsit date.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe înregistrări găsite.');

when even then

dbms\_output.put\_line('Campionatul nu a produs profit');

when loss then

dbms\_output.put\_line('Campionatul este in pierdere');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare a apărut: ' || SQLERRM);

END ex9;

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

--v2------------------------------------------------------------------------------------------------------------

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

procedure ex9\_v2 as

TYPE t\_result IS RECORD (

nume\_director varchar2(100),

nume\_staff varchar2(100),

nume\_echipa varchar2(100),

nume\_pilot varchar2(100)

);

TYPE t\_matrix IS TABLE OF t\_result;

v\_results t\_matrix;

nr\_staff number;

wrong\_name exception;

cursor director is

select e.nume\_director,s.nume\_staff,e.nume\_echipa , p.nume\_pilot

from echipa e, staff s, ocupa o, jobs j, pilot p

where s.id\_echipa = e.id\_echipa

and s.id\_staff = o.id\_staff

and o.id\_job = j.id\_job

and trim(lower(job\_name)) = 'principal'

and s.id\_staff = p.id\_staff;

begin

select count(\*)

into nr\_staff

from staff;

OPEN director;

FETCH director BULK COLLECT INTO v\_results;

CLOSE director;

IF v\_results.COUNT = 0 THEN

raise no\_data\_found;

elsif v\_results.count > nr\_staff then

raise too\_many\_rows;

else

FOR i IN v\_results.FIRST .. v\_results.LAST LOOP

if v\_results(i).nume\_director <> v\_results(i).nume\_staff then

raise wrong\_name;

end if;

dbms\_output.put\_line('Nume director: '|| v\_results(i).nume\_director ||

', nume echipa: '|| v\_results(i).nume\_echipa ||

', nume pilot: '|| v\_results(i).nume\_pilot);

END LOOP;

end if;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nu s-au găsit date.');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Prea multe înregistrări găsite.');

When wrong\_name then

dbms\_output.put\_line('Numele staffului e diferit de numele directorului');

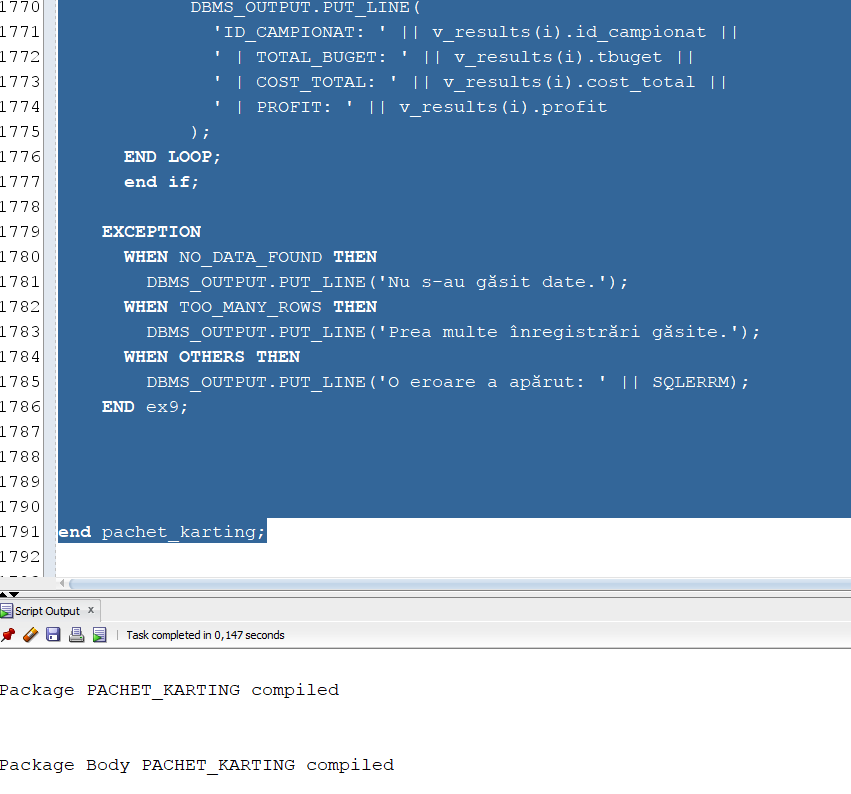
WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('O eroare a apărut: ' || SQLERRM);

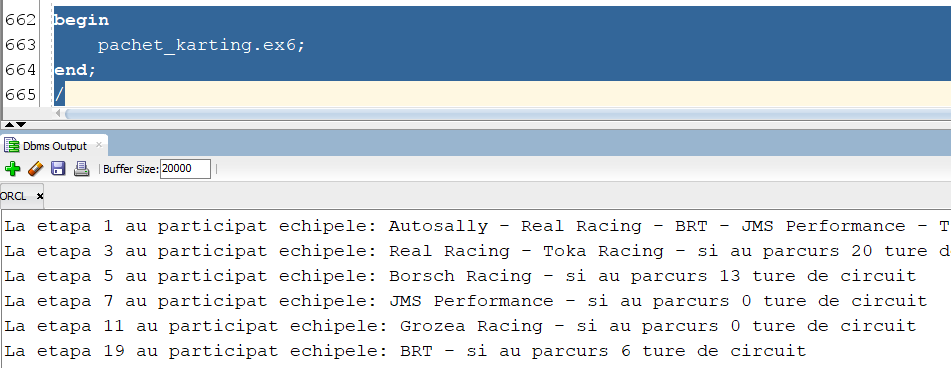
end ex9\_v2;

end pachet\_karting;

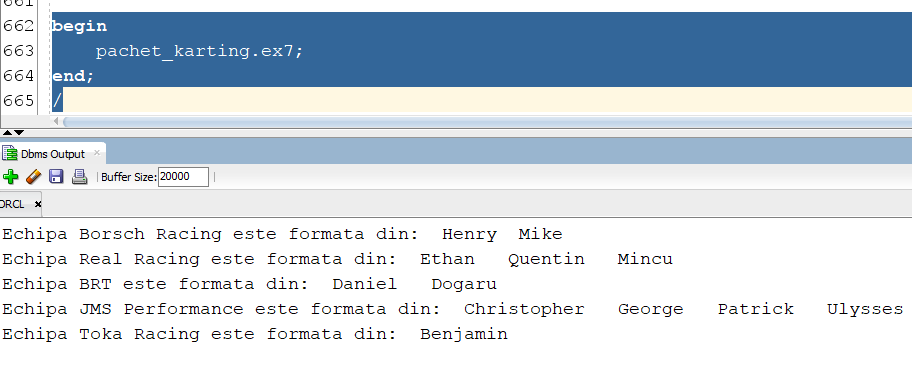
/

****

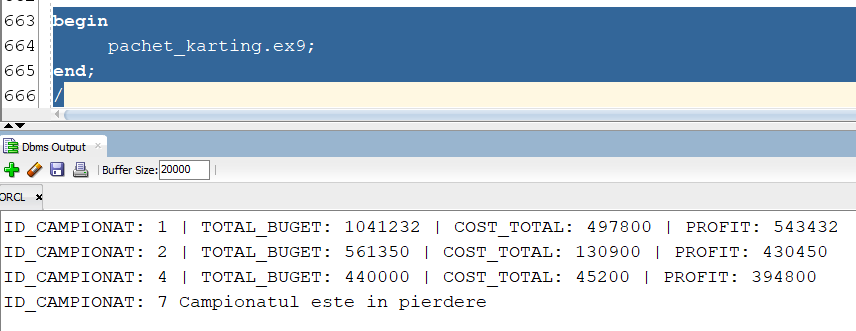
## 3 Coletii



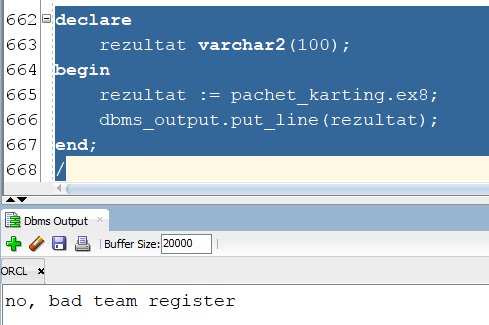
## 2 Cursoare



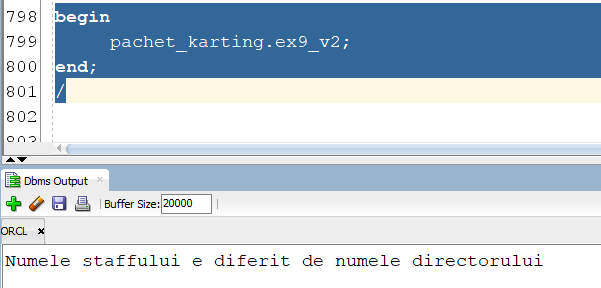
## 3 Tabele



## 5 Tabele



## 5 Tabele v2

****

# Pachet cu tipuri de date complexe

create or replace package pachet\_pilot as

procedure top3\_piloti;

procedure salariu\_staff;

function risc\_penalizare(id pilot.id\_pilot%type) return varchar2;

function lider\_categorie(v\_categorie pilot.categorie%type) return varchar2;

type pilot\_staff is record( nume\_p pilot.nume\_pilot%type, id\_s staff.nume\_staff%type, nume\_s staff.nume\_staff%type);

type info\_top3\_piloti is record ( nume pilot.nume\_pilot%type, id\_mecanic pilot.id\_staff%type, punctaj pilot.puncte\_clasament%type);

end pachet\_pilot;

create or replace package body pachet\_pilot as

procedure top3\_piloti as

TYPE t\_top3 IS TABLE OF info\_top3\_piloti;

v\_results\_top3 t\_top3;

id\_ec echipa.id\_echipa%type;

nume\_ec echipa.nume\_echipa%type;

nume\_st staff.nume\_staff%type;

cursor top3 is

select \*

from(select nume\_pilot, id\_staff, puncte\_clasament

from pilot

order by 3 desc)

where rownum <=3;

cursor all\_staff(ec echipa.id\_echipa%type) is

select nume\_staff

from staff

where id\_echipa = ec;

begin

open top3;

fetch top3 bulk collect into v\_results\_top3;

close top3;

for i in v\_results\_top3.first .. v\_results\_top3.last

loop

select id\_echipa

into id\_ec

from staff

where id\_staff = v\_results\_top3(i).id\_mecanic;

select nume\_echipa

into nume\_ec

from echipa

where id\_echipa = id\_ec;

dbms\_output.put\_line(v\_results\_top3(i).nume || ' a obtinut locul ' || i || ' cu '|| v\_results\_top3(i).punctaj || ' de puncte!');

dbms\_output.put\_line('Multumiri echipei ' || nume\_ec || ' formata din: ');

open all\_staff(id\_ec);

loop

fetch all\_staff into nume\_st;

exit when all\_staff%notfound;

dbms\_output.put(nume\_st || ' ');

end loop;

close all\_staff;

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

end loop;

end top3\_piloti;

--##################################################################################

--##################################################################################

--##################################################################################

procedure salariu\_staff as

TYPE t\_ps IS TABLE OF pilot\_staff;

v\_results\_ps t\_ps;

id\_st pilot.id\_staff%type;

salariu\_s jobs.salariu\_mediu%type;

cursor sal\_st(ids staff.id\_staff%type) is

select sum(j.salariu\_mediu)

from ocupa o, jobs j

where j.id\_job = o.id\_job

and o.id\_staff = ids

group by o.id\_staff;

cursor piloti is

select nume\_pilot, s.id\_staff, nume\_staff

from staff s, pilot p

where s.id\_staff = p.id\_staff;

begin

open piloti;

fetch piloti bulk collect into v\_results\_ps;

close piloti;

for i in v\_results\_ps.first .. v\_results\_ps.last

loop

dbms\_output.put('Membrul staff ' || v\_results\_ps(i).nume\_s || ' il are ca pilot pe ' || v\_results\_ps(i).nume\_p || 'si castiga zilnic ');

open sal\_st(v\_results\_ps(i).id\_s);

loop

fetch sal\_st into salariu\_s;

exit when sal\_st%notfound;

dbms\_output.put\_line(salariu\_s);

end loop;

close sal\_st;

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

end loop;

end salariu\_staff;

--##################################################################################

--##################################################################################

--##################################################################################

function risc\_penalizare(id pilot.id\_pilot%type) return varchar2 as

pct\_penalizare pilot.puncte\_penalizare%type;

punctaj\_invalid exception;

begin

select puncte\_penalizare

into pct\_penalizare

from pilot

where id\_pilot = id;

if pct\_penalizare < 0 then

raise punctaj\_invalid;

elsif pct\_penalizare = 0 then

return 'Pilotul nu are penalizari';

elsif pct\_penalizare < 6 then

return 'Pilotul este in zona sigura';

elsif pct\_penalizare < 11 then

return 'Pilotul este in zona de risc scazut';

elsif pct\_penalizare < 16 then

return 'Pilotul este in zona de risc';

else

return 'Pilotul trebuie penalizat!';

end if;

exception

when punctaj\_invalid then

dbms\_output.put\_line('Punctajul nu poate fi mai mic decat 0');

return null;

when no\_data\_found then

dbms\_output.put\_line('Pilotul nu exista');

return null;

end risc\_penalizare;

--##################################################################################

--##################################################################################

--##################################################################################

function lider\_categorie(v\_categorie pilot.categorie%type) return varchar2 as

nume\_lider pilot.nume\_pilot%type;

begin

select p.nume\_pilot

into nume\_lider

from (select nume\_pilot, puncte\_clasament

from pilot

where trim(lower(categorie)) = trim(lower(v\_categorie))

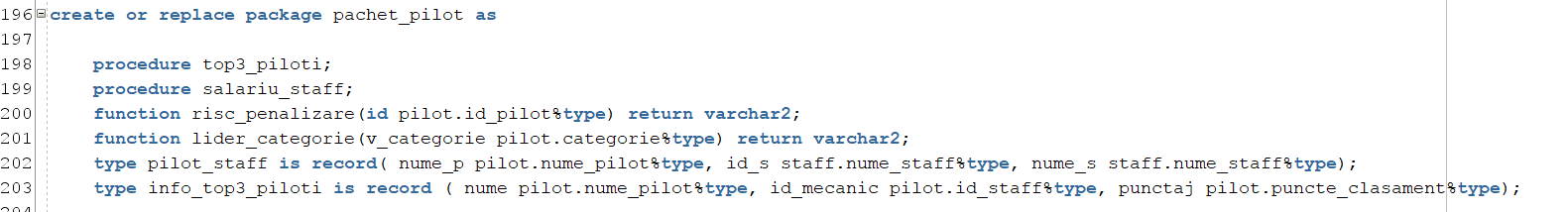
order by puncte\_clasament desc) p

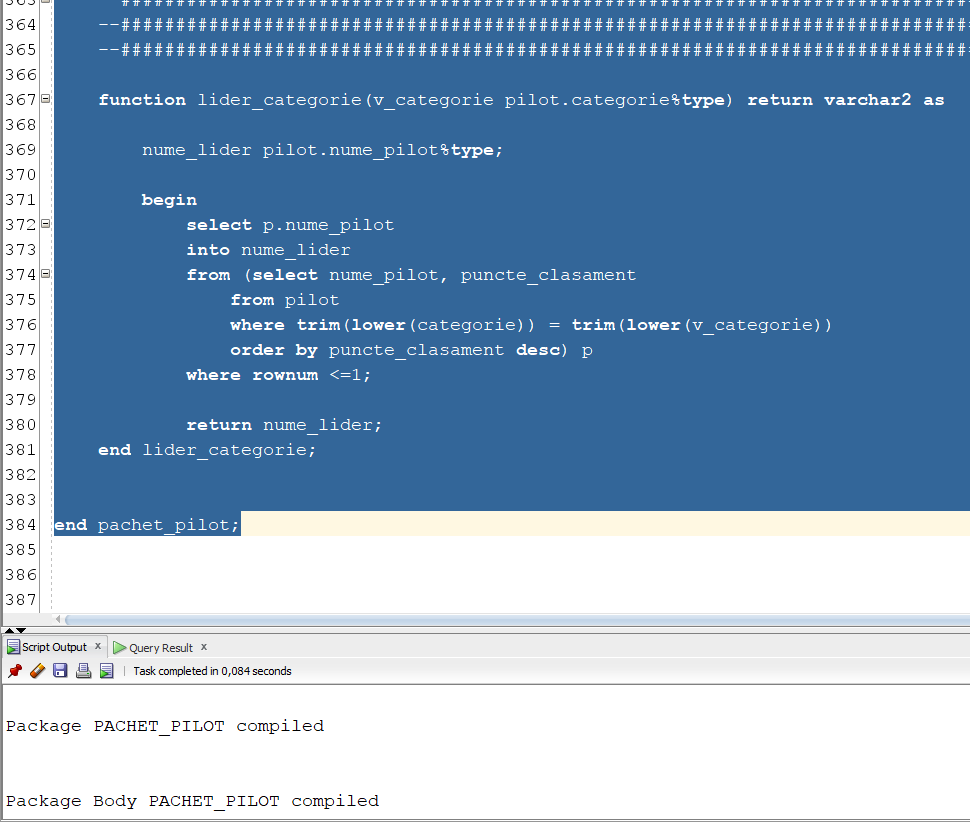
where rownum <=1;

return nume\_lider;

end lider\_categorie;

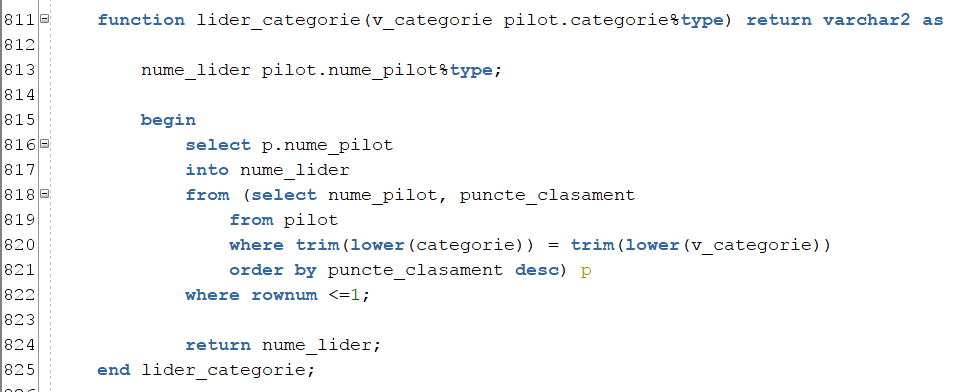
end pachet\_pilot;

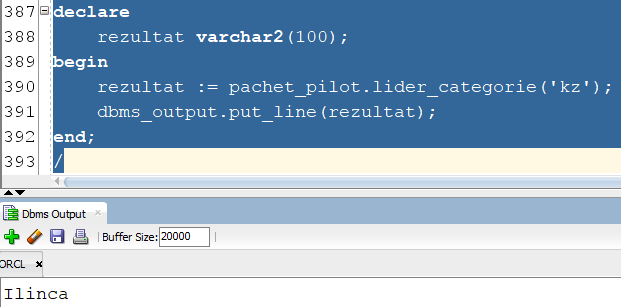




## Lider categorie

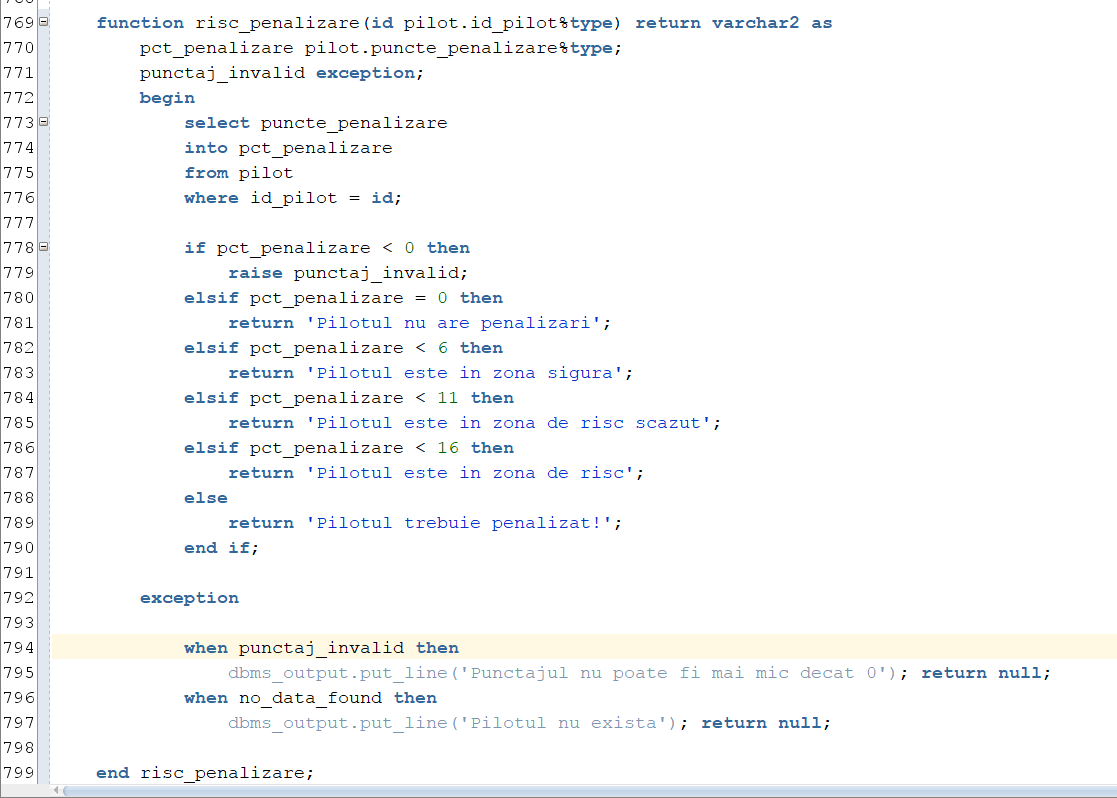
Sa se returneze liderul unei categorii date ca parametru (pilotul cu cele mai multe puncte)

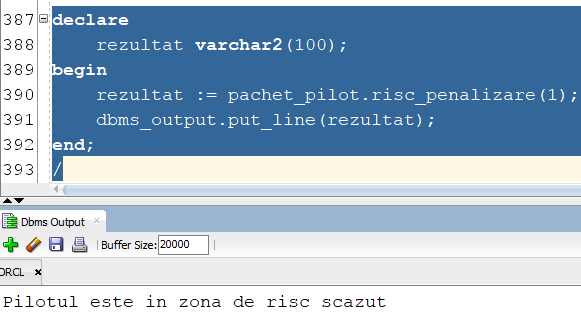




## Risc penalizare

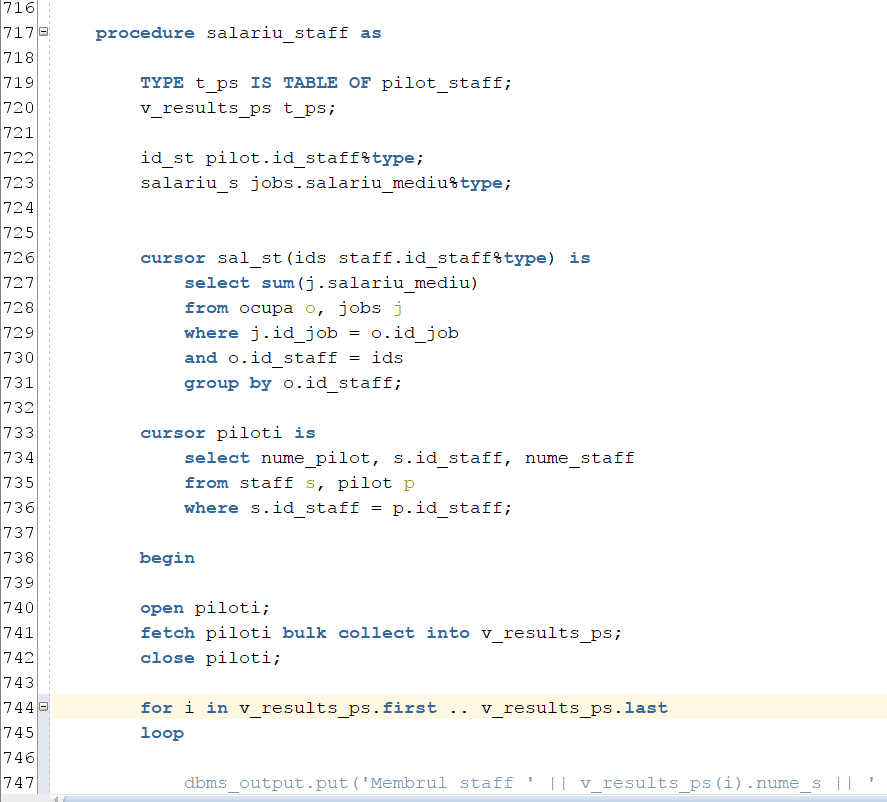
Sa se returneze, in functiile de punctele de penalizare, gradul de risc de penalizare a unui pilot (dat ca parametru). Pana in 6 puncte pilotul nu prezinta risc, intre 6 si 10 are un risc scazut, intre 11 si 15 are un risc ridicat, iar de 16 puncte acesta trebuie penalizat.

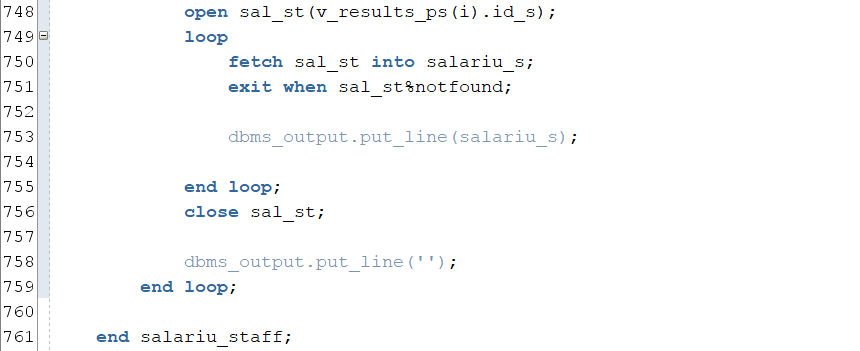


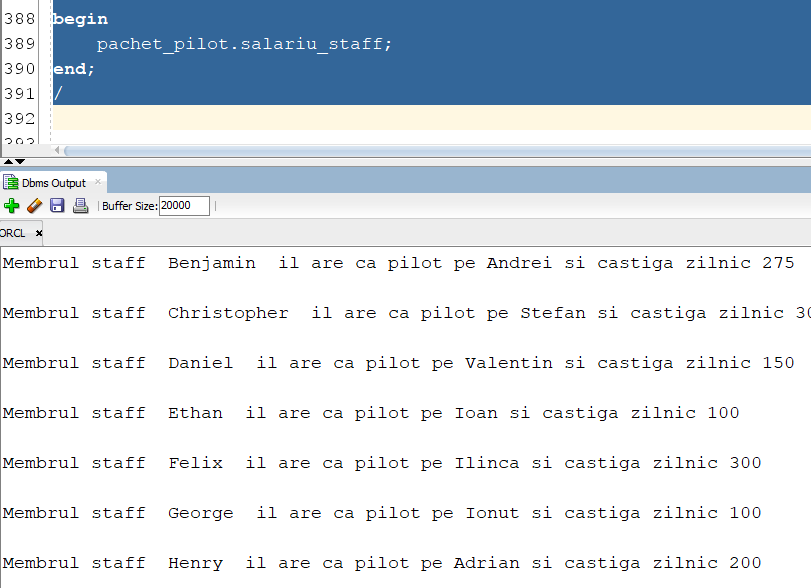


## Salariu staff

Sa se afiseze pentru fiecare pilot salariul zilnic al membrului staff ce se ocupa de el







## Top 3 piloti

Pentru primii 3 piloti din clasament, afisati echipa, precum si toti membrii staff ai acesteia

