

# Managementul competițiilor de înot susținute de Federația Română de Natație și Pentatlon Modern

## Cuprins

### 1. Introducere

- a. Prezentarea succintă a modelului proiectat și a regulilor sale
- b. Diagrama conceptuală
- c. Schemele relaționale
- d. Crearea tabelelor
- e. Prezentarea regulilor de securitate care vor fi aplicate asupra modelului

### 2. Criptarea datelor

### 3. Auditarea activităților asupra bazei de date

– de SYS\_ORCL

- a. Auditare standard
- b. Trigger-i de auditare
- c. Politici de auditare

### 4. Gestirea utilizatorilor unei baze de date și a resurselor computaționale

- a. Proiectarea configurației de management a identităților în baza de date
- b. Implementarea configurației de management a identităților în baza de date

### 5. Privilegii și roluri

- a. Privilegii sistem și obiect
- b. Ierarhii de privilegii
- c. Privilegii asupra obiectelor dependente

### 6. Aplicațiile pe baza de date și securitatea datelor

- a. Contextul aplicației
- b. SQL Injection

### 7. Mascarea datelor

#### I. Inserarea de date

#### II. Crearea de alte view-uri de către admin

III. Rezolvarea proceselor din cadrul aplicației 6-12,17 (celealte fiind deja rezolvate mai sus)

1. Crearea de proceduri de administrator
2. Crearea de blocuri pentru rezolvarea proceselor 6-12 rulate logat ca frnmp\_app\_sportiv1, frnmp\_app\_sportiv2
3. Rezolvarea procesului 12
4. Rezolvarea procesului 17

IV. Un exemplu de update realizat de administrator pentru o competiție

## 1. Introducere

### a. Prezentarea succintă a modelului proiectat și a regulilor sale

Proiectul presupune proiectarea unui model care realizează managementul competițiilor de înot al Federației Române de Natație și Pentatlon Modern. În acest fel, se vor reține date despre cluburile afiliate federației, a antrenorilor și sportivilor care fac parte din acestea, știindu-se că un sportiv și un antrenor pot fi înscrisi doar la un singur club. De asemenea, se ține și evidență în ceea ce privește care este antrenorul sportivului, cât și a unei ierarhii a antrenorilor bazată pe șef-subordonat așa cum reiese din ierarhia pe care o au în federație.

Antrenorii își înscriv sportivii la competițiile alese și la proba dorită, înscrierile apărând în istoricul sportivului, care după evoluția în probele din cadrul concursurilor la care a fost înscris vor fi actualizate locurile în funcție de categoria la care participă (o categorie are mai mulți sportivi) și un altul în funcție de gen, timpul obținut și punctajul.

De asemenea, o altă parte importantă a acestui model, este faptul că un sportiv poate fi sponsorizat de mai mulți sponsori, iar un sponsor poate sponsoriza mai mulți sportivi. Mai mult, sunt mai multe tipuri de competiții, iar un oraș al unei țări poate susține mai multe competiții.

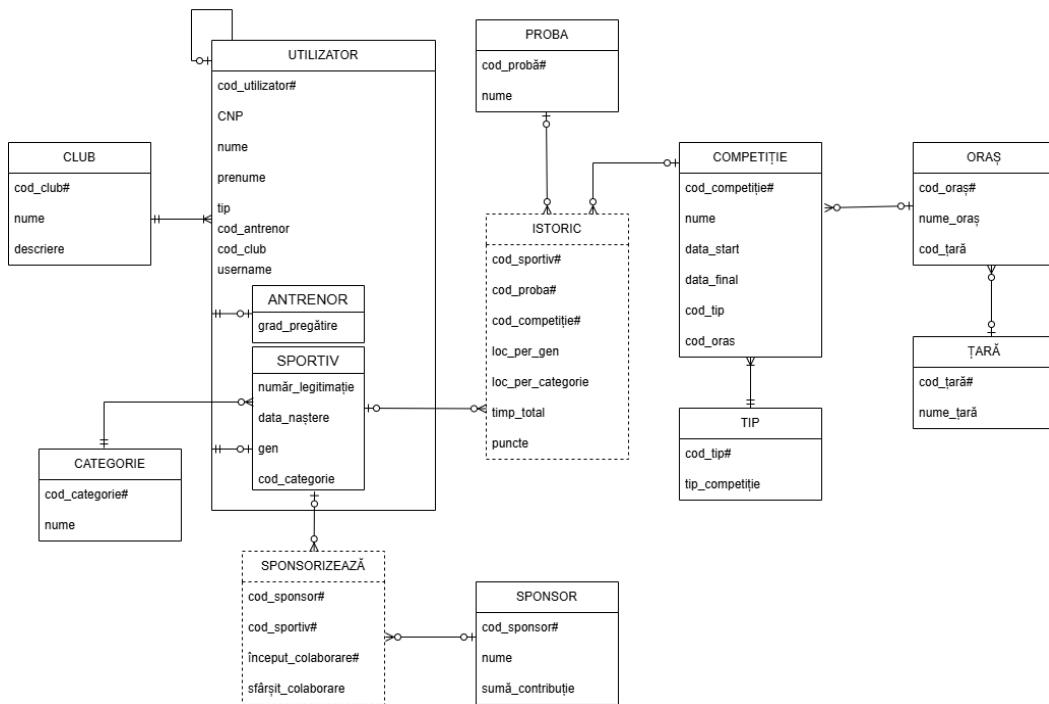
În ceea ce privește coloana gen din tabelul SPORTIV, valorile acesteia sunt ‘M’ sau ‘F’. Privind codurile țărilor, acestea sunt salvate ca primele două litere pentru identificarea țării.

Reguli:

- Un club poate avea mai mulți utilizatori (antrenori, sportivi), dar trebuie să aibă măcar unul, iar unui utilizator nu îi este permis să facă parte din mai multe cluburi, acesta trebuind să facă parte din unul singur.
- Sportivii pot avea sau nu sponsori care să le ofere un sprijin material în cariera sportivă. Un sponsor poate sponsoriza mai mulți sportivi sau nu neapărat.
- De asemenea, un sportiv trebuie să facă parte dintr-o categorie, iar o categorie poate conține mai mulți sportivi, dar nu neapărat.
- Când sunt anunțate competițiile, antrenorii își pot înscrie sportivii la probele dorite, dar pot exista situații în care unii sportivi nu vor fi înscrisi la anumite competiții și nici nu vor fi obligați să participe la toate probele.
- La o probă sportivă un sportiv poate obține mai multe rezultate (la competiții diferite poate obține rezultate diferite pentru aceeași probă la care a mai participat în trecut).

- În istoricul sportivului se vor afișa rezultatele obținute de sportiv la toate probele din cadrul competițiilor la care a participat.
- Competițiile care au loc sunt de diferite tipuri, fiecare competiție trebuind să aibă un singur tip (local, municipal, județean, european, balcanic, mondial), iar un tip poate fi avut de mai multe competiții.
- Aceste competiții au loc în diferite zone: o competiție se poate afla într-un singur oraș, iar într-un oraș se pot afla mai multe competiții. Totodată, un oraș se află într-o țară, iar o țară are mai multe orașe.

b. Diagrama conceptuală



c. Schemele relaționale

CLUB (cod\_club#, nume, descriere)

UTILIZATOR (cod\_utilizator#, CNP, nume, prenume, username, tip, cod\_antrenor, cod\_club)

ANTRENOR (cod\_antrenor#, grad\_pregătire)

SPORTIV (cod\_sportiv#, număr\_legitimatie, data\_naștere, gen, cod\_categorie)

PROBA (cod\_probă#, nume)

ȚARĂ (cod\_țară#, nume\_țară)

ORAȘ (cod\_oraș#, nume\_oraș, cod\_țară)

TIP (cod\_tip#, tip\_competiție)

COMPETIȚIE (cod\_competiție#, nume, data\_start, data\_final, cod\_oras, cod\_tip)

ISTORIC (cod\_sportiv#, cod\_proba#, cod\_competiție#, loc\_per\_gen, loc\_per\_categorie, timp\_total, puncte)

SPONSOR (cod\_sponsor#, nume, sumă\_contribuție)

SPONSORIZEAZĂ (cod\_sponsor#, cod\_sportiv#, început\_colaborare#, sfârșit\_colaborare)

CATEGORIE (cod\_categorie#, nume)

d. Crearea tabelelor

```
CREATE TABLE CLUB
```

```
(cod_club NUMBER(5),
  nume VARCHAR2(50) CONSTRAINT nume_club_nn NOT NULL,
  descriere VARCHAR2(250) CONSTRAINT descriere_nn NOT NULL,
  UNIQUE (nume),
  CONSTRAINT cod_club_pk PRIMARY KEY (cod_club)
);
```

COD_CLUB	NUME	DESCRIERE
1	AQUA TEAM	Facem valuri și cream campioni în fiecare moment
2	DELFINUL BUCURESTI	Aici delfinii însoță cu pasiune și putere
3	BUCHAREST SPORT CLUB ELITE	Excelenta cu fiecare lovitură, succes în fiecare cursă

```
CREATE TABLE TARA
```

```
(cod_tara CHAR(2),
  nume_tara VARCHAR2(25) CONSTRAINT nume_tara_nn NOT NULL,
  UNIQUE(nume_tara),
  CONSTRAINT cod_tara_pk PRIMARY KEY (cod_tara)
);
```

COD_TARA	NUME_TARA
FR	FRANTA
IT	ITALIA
JP	JAPONIA
KO	KOREA
PO	POLONIA
RO	ROMANIA

```
CREATE TABLE ORAS
(cod_oras NUMBER(5),
nume_oras VARCHAR2(25) CONSTRAINT nume_oras_nn NOT NULL,
cod_tara CHAR(2) CONSTRAINT cod_tara_nn NOT NULL,
CONSTRAINT cod_oras_pk PRIMARY KEY (cod_oras),
CONSTRAINT cod_tara_fk FOREIGN KEY (cod_tara) REFERENCES TARA
(cod_tara),
UNIQUE (nume_oras)
);
```

COD_ORAS	NUME_ORAS	COD_TARA
1	BUCURESTI	RO
2	CRAIOVA	RO
3	LUBLIN	PO
4	PARIS	FR
5	KYOTO	JP
6	NAPOLI	IT
7	GWANGJU	KO

```
CREATE TABLE TIP
(cod_tip NUMBER(5),
tip_competitie VARCHAR2(20) CONSTRAINT comp_tip_nn NOT NULL,
UNIQUE(tip_competitie),
CONSTRAINT cod_tip_pk PRIMARY KEY (cod_tip)
);
```

COD_TIP	TIP_COMPETITIE
5	BALCANIC
4	EUROPEAN
3	JUDETEAN
1	LOCAL
6	MONDIAL
2	MUNICIPAL

```
CREATE TABLE COMPETITIE
(cod_competitie NUMBER(5),
nume VARCHAR2(30) CONSTRAINT nume_comp_nn NOT NULL,
data_start DATE CONSTRAINT data_start_nn NOT NULL,
data_final DATE CONSTRAINT data_final_nn NOT NULL,
cod_oras NUMBER(5) NOT NULL,
cod_tip NUMBER(5) NOT NULL,
CONSTRAINT cod_competitie_pk PRIMARY KEY (cod_competitie),
CONSTRAINT cod_tip_fk FOREIGN KEY (cod_tip) REFERENCES TIP (cod_tip),
CONSTRAINT cod_oras_fk FOREIGN KEY (cod_oras) REFERENCES ORAS
(cod_oras),
CONSTRAINT date_concurs CHECK (data_start<data_final),
UNIQUE (nume)
);
```

COD_COMPETITIE	NUME	DATA_START	DATA_FINAL	COD_ORAS	COD_TIP
1	CUPA DE TOAMNA 2024	05-NOV-24	06-NOV-24	1	6
2	CUPA DE IARNA 2024	10-DEC-24	11-DEC-24	1	1
3	CUPA DE ANUL NOU 2025	03-JAN-25	04-JAN-25	4	4

```
CREATE TABLE PROBA
(cod_proba NUMBER(5),
nume VARCHAR2(30) CONSTRAINT nume_proba_nn NOT NULL,
CONSTRAINT cod_proba_pk PRIMARY KEY (cod_proba),
```

UNIQUE (nume)

);

COD_PROBA	NUME
3 100	LIBER
4 200	BRAS
5 200	FLUTURE
6 400	LIBER
1 50	LIBER
2 50	SPATE

CREATE TABLE UTILIZATOR

```
(cod_utilizator NUMBER(5),
CNP RAW(32),
nume VARCHAR2(20),
prenume VARCHAR2(20),
username VARCHAR2(20),
tip VARCHAR2(15) DEFAULT 'SPORTIV',
cod_antrenor NUMBER(5),
cod_club NUMBER(5) NOT NULL,
CONSTRAINT cod_utilizator_pk PRIMARY KEY (cod_utilizator),
CONSTRAINT cod_club_fk FOREIGN KEY (cod_club) REFERENCES CLUB
(cod_club),
CONSTRAINT cod_utilizator_fk FOREIGN KEY (cod_antrenor) REFERENCES
UTILIZATOR (cod_utilizator)
);
```

COD_UTILIZATOR	CNP	NUME	PRENUME	USERNAME	TIP	COD_ANTRENOR	COD_CLUB
1 13A97F0CA9AA926493313139350B66B6	BRAN	IONEL	FRNMP_ANTRENOR1	ANTRENOR	(null)		1
2 3743045C9C4926288E8CCFFB3988E58B	STAN	DENISA	FRNMP_ANTRENOR2	ANTRENOR		1	1
3 46CF93FAFD2F59F27808E70B770C623B	TANCI	MARIA	FRNMP_ANTRENOR3	ANTRENOR		1	2
4 76DC48FE274B8419B5D8AB99ACDF7F4C	GRIGORAS	MARIN	FRNMP_ANTRENOR4	ANTRENOR		2	3
5 226C778E623F76C733821942DCD845E4	DRAGU	STELIAN	FRNMP_APP_SPORTIV1	SPORTIV		1	1
6 E3774303C02D366924BA2AEC7DFCEF2A	MICU	DENISA	FRNMP_APP_SPORTIV2	SPORTIV		2	1
7 DA4C3632AB96EF23822963F73FD1EF08	MARCU	MARIA	FRNMP_APP_SPORTIV3	SPORTIV		3	2
8 93107A27C1617257A1B1F35023675677	STAN	FILIP	FRNMP_APP_SPORTIV4	SPORTIV		4	3
9 232BC8B04AC595BE96C8EE3709E4CC20	MISTI	SAR	FRNMP_APP_SPORTIV5	SPORTIV		1	1

```

CREATE TABLE ANTRENOR
(cod_antrenor NUMBER(5),
grad_pregatire NUMBER(5) NOT NULL,
CONSTRAINT cod_antrenor_pk PRIMARY KEY (cod_antrenor),
CONSTRAINT cod_antrenor_fk FOREIGN KEY (cod_antrenor) REFERENCES
UTILIZATOR(cod_utilizator)
);

```

COD_ANTRENOR	GRAD_PREGATIRE
1	20
2	9
3	5
4	12

```

CREATE TABLE CATEGORIE
(cod_categorie NUMBER(5),
nume VARCHAR2(50) CONSTRAINT nume_categorie_nn NOT NULL,
UNIQUE (nume),
CONSTRAINT cod_categorie_pk PRIMARY KEY (cod_categorie)
);

```

COD_CATEGORIE	NUME
9	CADET F
10	CADET M
13	COPIL 10 F
14	COPIL 10 M
11	COPIL 11 F
12	COPIL 11 M
5	JUNIOR 1 F
6	JUNIOR 1 M
7	JUNIOR 2 F
8	JUNIOR 2 M
1	SENIOR F
2	SENIOR M
3	TINERET F
4	TINERET M

```

CREATE TABLE SPORTIV

(cod_sportiv NUMBER(5),
numar_legitimatie NUMBER(5) CONSTRAINT numar_legitimatie_nn NOT NULL,
data_nastere DATE CONSTRAINT nastere_sp_nn NOT NULL,
gen CHAR(1) CONSTRAINT genul_sp_nn NOT NULL,
CONSTRAINT genul_sp CHECK (gen='M' OR gen='F'),
CONSTRAINT cod_sportiv_pk PRIMARY KEY (cod_sportiv),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
UTILIZATOR(cod_utilizator)

);

ALTER TABLE SPORTIV
ADD cod_categorie NUMBER(5);

ALTER TABLE SPORTIV
ADD CONSTRAINT cod_categorie_fk FOREIGN KEY (cod_categorie)
REFERENCES CATEGORIE(cod_categorie);

```

COD_SPORTIV	NUMAR_LEGITIMATIE	DATA_NASTERE	GEN	COD_CATEGORIE
5	1	21-AUG-05	M	2
6	2	26-APR-05	F	1
7	3	30-DEC-00	F	1
8	4	25-MAR-11	M	8
9	5	15-OCT-11	M	8

```

CREATE TABLE SPONSOR

(cod_sponsor NUMBER(5),
nume VARCHAR2(50) CONSTRAINT nume_sponsor_nn NOT NULL,
suma_contributie RAW(128) CONSTRAINT suma_sponsor_nn NOT NULL,
UNIQUE (nume),

```

```
CONSTRAINT cod_sponsor_pk PRIMARY KEY (cod_sponsor)
);
```

COD_SPONSOR	NUME	SUMA_CONTRIBUTIE
1	RED BULL	5960F26A1CB3C5400FB9AC28FD75C132
2	ISOSTAR	699C764A00205AD420ADD98AEA6D91F4

```
CREATE TABLE ISTORIC
(
cod_sportiv NUMBER(5),
cod_proba NUMBER(5),
cod_competitie NUMBER(5),
loc_per_gen NUMBER(5),
loc_per_categorie NUMBER(5),--pe baza gen si an nastere sportiv
timp_total NUMBER(5), --in secunde
puncte NUMBER(5),
CONSTRAINT pk_sportiv_proba_competitie primary key(cod_sportiv, cod_proba,
cod_competitie),
CONSTRAINT cod_ist_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv),
CONSTRAINT cod_ist_proba_fk FOREIGN KEY (cod_proba) REFERENCES
PROBA (cod_proba),
CONSTRAINT cod_ist_competitie_fk FOREIGN KEY(cod_competitie)
REFERENCES COMPETITIE(cod_competitie)
);
```

COD_SPORTIV	COD_PROBA	COD_COMPETITIE	LOC_PER_GEN	LOC_PER_CATEGORIE	TIMP_TOTAL	PUNCTE
5	3	1	2	3	32	130
9	1	1	20	5	32	110
5	2	1	3	1	27	150
9	2	1	1	1	25	150
5	4	1	1	1	60	150
5	1	2	15	2	24	14
5	1	3	9	3	27	130
5	3	3	1	1	30	150
7	1	1	10	4	33	120
7	3	1	1	1	44	150
7	1	2	1	1	28	15
8	1	1	9	2	20	140
8	2	1	2	2	26	140
5	1	1	15	3	26	130
6	1	1	1	1	27	150
6	3	1	3	2	46	140
6	1	2	2	2	29	14

Dacă tabelele de sus au fost create în schema administratorului, tabelul SPONSORIZEAZA este creat de administrator în schema fiecărui sportiv.

```
CREATE TABLE FRNMP_APP_SPORTIV1.SPONSORIZEAZA
(cod_sponsor NUMBER(5),
cod_sportiv NUMBER(5),
inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
CONSTRAINT datele CHECK (inceput_colaborare < sfarsit_colaborare),
CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
(cod_sponsor,cod_sportiv,inceput_colaborare),
CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES
SPONSOR (cod_sponsor),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv),
);
```

```
CREATE TABLE FRNMP_APP_SPORTIV2.SPONSORIZEAZA
(cod_sponsor NUMBER(5),
cod_sportiv NUMBER(5),
inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
CONSTRAINT datele CHECK (inceput_colaborare<sfarsit_colaborare),
CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
(cod_sponsor,cod_sportiv,inceput_colaborare),
CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES
SPONSOR (cod_sponsor),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv)
);
```

```
CREATE TABLE FRNMP_APP_SPORTIV3.SPONSORIZEAZA
(cod_sponsor NUMBER(5),
cod_sportiv NUMBER(5),
inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
CONSTRAINT datele CHECK (inceput_colaborare<sfarsit_colaborare),
CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
(cod_sponsor,cod_sportiv,inceput_colaborare),
CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES
SPONSOR (cod_sponsor),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv)
);
```

```
CREATE TABLE FRNMP_APP_SPORTIV4.SPONSORIZEAZA
```

```
(cod_sponsor NUMBER(5),
cod_sportiv NUMBER(5),
inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
CONSTRAINT datele CHECK (inceput_colaborare<sfarsit_colaborare),
CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
(cod_sponsor,cod_sportiv,inceput_colaborare),
CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES
SPONSOR (cod_sponsor),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv)
);
```

```
CREATE TABLE FRNMP_APP_SPORTIV5.SPONSORIZEAZA
(cod_sponsor NUMBER(5),
cod_sportiv NUMBER(5),
inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
CONSTRAINT datele CHECK (inceput_colaborare<sfarsit_colaborare),
CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
(cod_sponsor,cod_sportiv,inceput_colaborare),
CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES
SPONSOR (cod_sponsor),
CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES
SPORTIV (cod_sportiv)
);
```

Query Result				
	COD_SPONSOR	INCEPUT_COLABORARE	SFARSIT_COLABORARE	COD_SPORTIV
1	1	07-JAN-25	01-MAR-25	SPORTIV1
2	2	18-JAN-25	23-JAN-25	SPORTIV2

e. Prezentarea regulilor de securitate care vor fi aplicate asupra modelului

Dintre regulile de securitate folosite se pot menționa: criptarea și mascarea datelor, auditarea, gestiunea utilizatorilor și resurselor computaționale, acordarea de privilegii și roluri utilizatorilor astfel încât să fie minimizat accesul la date, crearea unui context al aplicației și protejarea de SQL Injection.

În ceea ce privește criptarea s-a dorit să fie aplicată pentru date precum CNP și suma de contribuție a sponsorilor.

Pentru auditare, s-au folosit trei tipuri, anume: standard, prin triggeri de auditare și politici de auditare pentru ca sys să poată avea o imagine amplă asupra operațiilor efectuate de utilizatori în baza de date și de a identifica posibilele modificări suspecte.

În ceea ce privește gestiunea utilizatorilor bazei de date și resurselor computaționale s-au creat conturi pentru 5 sportivi, 4 antrenori, 2 arbitrii, 1 administrator, 2 reprezentanți ai companiilor, 1 cont pentru public și unul pentru un utilizator extern. Pentru utilizatori s-au creat profile de limitare, printre care a fost inclusă și verificarea parolei care trebuie să respecte anumite condiții impuse printr-o funcție creată. Acordarea privilegiilor și a rolurilor utilizatorilor este în concordanță cu permisiunile din matricea entitate-utilizator. Totodată, acordarea de spațiu de stocare a fost atribuit tot în funcție de matricea amintită mai sus, pentru ca doar utilizatorii care au cu adevărat nevoie să dispună de spațiu de stocare.

Mai mult, s-a creat un context prin care le este impus antrenorilor și arbitrilor o anumită perioadă de timp în care își pot desfășura activitatea.

## 2. Criptarea datelor

Se vor scrie două proceduri pentru criptare respectiv decriptare pentru atributul suma\_contribuție din tabelul SPONSOR, și alte două proceduri pentru atributul CNP al tabelului UTILIZATOR. La decriptare se dorește ca introducându-se numele sponsorului să se afle suma decriptată, iar pentru tabela UTILIZATOR se dorește ca prin introducerea username-ului să se afle CNP-ul decriptat. Mai mult, cheile de criptare sunt specifice per tabel și vor fi salvate într-un tabel unde se va menționa valoarea cheii și numele tabelului unde se aplică. Datele vor fi criptate cu AES-128, PAD\_PKCS5, CHAIN\_CBC. Aceste proceduri de criptare vor fi folosite în inserarea datelor.

În frnmp\_app\_admin

```

CREATE TABLE CHEIE_CRIPT_DECR (
    id_cheie NUMBER(5) PRIMARY KEY,
    cheie_val RAW(128),
    tabel VARCHAR2(50) UNIQUE
);

CREATE OR REPLACE FUNCTION CRIPARE_SUMA_ADM(
    suma_contributie NUMBER
) RETURN RAW IS
    cheie_actuala RAW(128);
    nr_chei NUMBER;
    mod_operare PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES128 +
        DBMS_CRYPTO.PAD_PKCS5 +
        DBMS_CRYPTO.CHAIN_CBC;
BEGIN
    SELECT COUNT(*) INTO nr_chei
    FROM CHEIE_CRIPT_DECR
    WHERE tabel = 'SPONSOR';

    IF nr_chei = 0 THEN

```

```

cheie_actuala := DBMS_CRYPTO.RANDOMBYTES(16);

INSERT INTO CHEIE_CRIPT_DECR (id_cheie, cheie_val, tabel)
VALUES (
    (SELECT NVL(MAX(id_cheie), 0) + 1 FROM CHEIE_CRIPT_DECR),
    cheie_actuala,
    'SPONSOR'
);
ELSE

    SELECT cheie_val INTO cheie_actuala
    FROM CHEIE_CRIPT_DECR
    WHERE tabel = 'SPONSOR';

END IF;

RETURN DBMS_CRYPTO.ENCRYPT(
    UTL_I18N.STRING_TO_RAW(TO_CHAR(suma_contributie)),
    mod_operare,
    cheie_actuala
);
END;
/
CREATE OR REPLACE FUNCTION DECRYPTARE_SUMA ADM(
    nume_sponsor VARCHAR2
) RETURN NUMBER IS
    suma_contrib RAW(128);
    cheie_actuala RAW(128);
    mod_operare PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES128 +

```

```
DBMS_CRYPTO.PAD_PKCS5 +
DBMS_CRYPTO.CHAIN_CBC;

BEGIN

    SELECT suma_contributie INTO suma_contrib
    FROM SPONSOR
    WHERE nume = nume_sponsor;

    SELECT cheie_val INTO cheie_actuala
    FROM CHEIE_CRIPT_DECR
    WHERE tabel = 'SPONSOR';

    RETURN TO_NUMBER(UTL_I18N.RAW_TO_CHAR(
        DBMS_CRYPTO.decrypt(
            suma_contrib,
            mod_operare,
            cheie_actuala
        ),
        'AL32UTF8'));
    END;
    /

INSERT INTO SPONSOR(cod_sponsor,nume,suma_contributie)
VALUES (1,'RED BULL',CRYPTARE_SUMA_ADMIN(500));

INSERT INTO SPONSOR(cod_sponsor,nume,suma_contributie)
VALUES (2,'ISOSTAR',CRYPTARE_SUMA_ADMIN(600.4));
```

```
delete from sponsor;  
select * from sponsor;  
  
SELECT DECRYPTARE_SUMA ADM('RED BULL')  
FROM dual;  
  
SELECT DECRYPTARE_SUMA ADM('ISOSTAR')  
FROM dual;  
  
--pentru tabel UTILIZATOR  
CREATE OR REPLACE FUNCTION CRIPTARE_UTILIZ ADM(  
    cnp_ul VARCHAR2  
) RETURN RAW IS  
    cheie_actuala RAW(128);  
    nr_chei NUMBER;  
    mod_operare PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES128 +  
        DBMS_CRYPTO.PAD_PKCS5 +  
        DBMS_CRYPTO.CHAIN_CBC;  
BEGIN  
    SELECT COUNT(*) INTO nr_chei  
    FROM CHEIE_CRIPT_DECR  
    WHERE tabel = 'UTILIZATOR';  
  
    IF nr_chei = 0 THEN  
        cheie_actuala := DBMS_CRYPTO.RANDOMBYTES(16);  
  
        INSERT INTO CHEIE_CRIPT_DECR (id_cheie, cheie_val, tabel)  
        VALUES (  
    END IF;
```

```
(SELECT NVL(MAX(id_cheie), 0) + 1 FROM CHEIE_CRIPT_DECR),
cheie_actuala,
'UTILIZATOR'
);
ELSE

    SELECT cheie_val INTO cheie_actuala
    FROM CHEIE_CRIPT_DECR
    WHERE tabel = 'UTILIZATOR';

END IF;

RETURN DBMS_CRYPTO.ENCRYPT(
    UTL_I18N.STRING_TO_RAW(cnp_ul),
    mod_operare,
    cheie_actuala
);
END;
/
CREATE OR REPLACE FUNCTION DECRYPTARE_UTILIZ_ADM(
    username_utiliz VARCHAR2
) RETURN NUMBER IS
    cnp_ul RAW(128);
    cheie_actuala RAW(128);
    mod_operare PLS_INTEGER := DBMS_CRYPTO.ENCRYPT_AES128 +
        DBMS_CRYPTO.PAD_PKCS5 +
        DBMS_CRYPTO.CHAIN_CBC;
BEGIN
```

```
SELECT CNP INTO cnp_ul
FROM UTILIZATOR
WHERE username = username_utiliz;

SELECT cheie_val INTO cheie_actuala
FROM CHEIE_CRIPT_DECR
WHERE tabel = 'UTILIZATOR';

RETURN TO_NUMBER(UTL_I18N.RAW_TO_CHAR(
DBMS_CRYPTO.decrypt(
cnp_ul,
mod_operare,
cheie_actuala
),
'AL32UTF8'));

END;
/

INSERT INTO UTILIZATOR
VALUES
(1,CRYPTARE_UTILIZ_ADMIN('1850223145522'), 'BRAN', 'IONEL', 'FRNMP_ANTRENOR1', 'ANTRENOR', NULL, 1);

INSERT INTO UTILIZATOR
VALUES
(2,CRYPTARE_UTILIZ_ADMIN('2960225640002'), 'STAN', 'DENISA', 'FRNMP_ANTRENOR2', 'ANTRENOR', 1, 1);

INSERT INTO UTILIZATOR
VALUES
```

```
(3,CRYPTARE_UTILIZ_ADMIN('2951230111012'),'TANCI','MARIA','FRNMP_ANTRENOR3', 'ANTRENOR',1,2);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(4,CRYPTARE_UTILIZ_ADMIN('1630823578936'),'GRIGORAS','MARIN','FRNMP_ANTRENOR4', 'ANTRENOR',2,3);
```

```
SELECT * FROM UTILIZATOR;
```

```
SELECT DECRYPTARE_UTILIZ_ADMIN('FRNMP_ANTRENOR1') FROM DUAL;
```

```
INSERT INTO ANTRENOR VALUES(1,20);
```

```
INSERT INTO ANTRENOR VALUES(2,9);
```

```
INSERT INTO ANTRENOR VALUES(3,5);
```

```
INSERT INTO ANTRENOR VALUES(4,12);
```

```
SELECT * FROM ANTRENOR;
```

```
select * from UTILIZATOR ;
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(5,CRYPTARE_UTILIZ_ADMIN('7050821555520'),'DRAGU','STELIAN','FRNMP_APP_SPORTIV1', 'SPORTIV',1,1);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(6,CRYPTARE_UTILIZ_ADMIN('6050426785520'),'MICU','DENISA','FRNMP_APP_SPORTIV2', 'SPORTIV',2,1);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(7,CRYPTARE_UTILIZ_ADMIN('6001230121030'),'MARCU','MARIA','FRNMP_APP_SP
```

```
ORTIV3', 'SPORTIV',3,2);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(8,CRIPTARE_UTILIZ ADM('7110325871101'),'STAN','FILIP','FRNMP_APP_SPORT  
IV4', 'SPORTIV',4,3);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
```

```
(9,CRIPTARE_UTILIZ ADM('7111015775232'),'MISTI','SAR','FRNMP_APP_SPORTI  
V5', 'SPORTIV',1,1);
```

```
select * from categorie;
```

```
INSERT INTO SPORTIV
```

```
VALUES (5,1,TO_DATE('21-08-2005','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 2
```

```
WHERE cod_sportiv = 5;
```

```
insert INTO SPORTIV
```

```
VALUES (6,2,TO_DATE('26-04-2005','DD-MM-YYYY'),'F');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 1
```

```
WHERE cod_sportiv = 6;
```

```
INSERT INTO SPORTIV
```

```
VALUES (7,3,TO_DATE('30-12-2000','DD-MM-YYYY'),'F');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 1
```

```
WHERE cod_sportiv = 7;
```

```
INSERT INTO SPORTIV
```

```
VALUES (8,4,TO_DATE('25-03-2011','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 8
```

```
WHERE cod_sportiv = 8;
```

```
INSERT INTO SPORTIV
```

```
VALUES (9,5,TO_DATE('15-10-2011','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 8
```

```
WHERE cod_sportiv = 9;
```

```
commit;
```

```
SELECT * FROM SPORTIV;
```

3. Auditarea activităților asupra bazei de date

– de SYS\_ORCL

a. Auditare standard

Să se monitorizeze toate activitățile de creare a tabelelor pentru administrator. Pentru tabelul UTILIZATOR, CLUB și SPONSOR să se monitorizeze inserările care eșuează,

iar pentru tabelul ISTORIC să se monitorizeze oricând activitățile de interogare, adăugare de date și activitățile de actualizare. Datele monitorizate vor fi stocate în baza de date.

--a. Auditare standard

```
select value from v$parameter where name ='audit_trail'; --DB  
show parameter audit_trail;  
--vrem DB,EXTENDED pentru a stoca si textul comenziilor  
alter system set audit_trail=db,extended  
scope=spfile;  
--diconnect sys_orcl  
--din sqlplus se da shutdown immediate si startup  
--connect sys_orcl  
select value from v$parameter where name ='audit_trail'; --DB,EXTENDED
```

AUDIT CREATE TABLE BY FRNMP\_APP\_ADMIN;

AUDIT INSERT ON FRNMP\_APP\_ADMIN.USUARIU  
WHENEVER NOT SUCCESSFUL;

AUDIT INSERT ON FRNMP\_APP\_ADMIN.SPONSOR  
WHENEVER NOT SUCCESSFUL;

AUDIT INSERT ON FRNMP\_APP\_ADMIN.CLUB  
WHENEVER NOT SUCCESSFUL;

AUDIT SELECT,INSERT,UPDATE ON FRNMP\_APP\_ADMIN.ISTORIC;

--DEZACTIVARE

--NOAUDIT INSERT ON FRNMP\_APP\_ADMIN.SPONSOR WHENEVER NOT

```
SUCCESSFUL;
```

```
--NOAUDIT CREATE TABLE BY FRNMP_APP_ADMIN;
```

```
--NOAUDIT ALL;
```

--Pentru a vedea rezultatele auditului

```
SELECT * FROM DBA_AUDIT_TRAIL;
```

```
SELECT * FROM DBA_AUDIT_TRAIL
```

```
where username ='FRNMP_APP_ADMIN';
```

```
SELECT * FROM DBA_AUDIT_TRAIL
```

```
where username ='FRNMP_ARBITRU1';
```

```
SELECT * FROM DBA_AUDIT_TRAIL
```

```
where username ='FRNMP_ANTRENOR1';
```

--pentru a vedea ca auditarile există

```
select * from DBA_OBJ_AUDIT_OPTS
```

```
where owner ='FRNMP_APP_ADMIN';
```

Când username = 'FRNMP\_APP\_ADMIN'

OS_USERNAME	USERNAME	USERHOST	TERMINAL	TIMESTAMP	OWNER	OBJ_NAME	ACTION	ACTION_NAME
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_ADMIN	ANTRENOR	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_ADMIN	SPORTIV	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_ADMIN	SPONSOR	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_ADMIN	ISTORIC	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_SPORTIV1	SPONSORIZEAZA	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_SPORTIV1	SPONSORIZEAZA	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	02-JAN-25	FRNMP_APP_ADMIN	CHEIE_CRIPT_DECR	1	CREATE TABLE
mirun	FRNMP_APP_ADMIN	Miruna	unknown	03-JAN-25	FRNMP_APP_ADMIN	CLUB	2	INSERT
mirun	FRNMP_APP_ADMIN	Miruna	unknown	03-JAN-25	FRNMP_APP_ADMIN	SPONSOR	2	INSERT
mirun	FRNMP_APP_ADMIN	Miruna	unknown	03-JAN-25	FRNMP_APP_ADMIN	SPONSOR	2	INSERT
mirun	FRNMP_APP_ADMIN	Miruna	unknown	03-JAN-25	FRNMP_APP_ADMIN	SPONSOR	2	INSERT

Când username = 'FRNMP\_ARBITRU1'

OS_USERNAME	USERNAME	USERHOST	TERMINAL	TIMESTAMP	OWNER	OBJ_NAME	ACTION	ACTION_NAME
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	6 UPDATE	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	6 UPDATE	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	6 UPDATE	
mirun	FRNMP_ARBITRU1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	

Când username ='FRNMP\_ANTRENOR1'

OS_USERNAME	USERNAME	USERHOST	TERMINAL	TIMESTAMP	OWNER	OBJ_NAME	ACTION	ACTION_NAME
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	2 INSERT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	2 INSERT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	
mirun	FRNMP_ANTRENOR1	Miruna	unknown	19-JAN-25	FRNMP_APP_ADMIN	ISTORIC	3 SELECT	

#### b. Trigger-i de auditare

Înregistrează în tabelul audit informații despre LMD care stabilesc un număr de puncte mai mare de 15 în istoricul sportivilor cu participări la competiții de tipul LOCAL sau cu număr mai mare de 150 de puncte în istoricul sportivilor cu participări la competiții cu unul din tipurile: MONDIAL, EUROPEAN. Mai mult creează un tabel de audit care să îl extindă pe primul prin adăugarea de informații cu privire la datele inserate sau modificate pentru tabelul istoric care respectă condiționările de mai sus.

```
CREATE TABLE TAB_AUDIT_ISTORIC (
    cod_audit_ist NUMBER(4) PRIMARY KEY,
    utilizator_ VARCHAR2(30) NOT NULL,
    sesiune_ NUMBER(10) NOT NULL,
    host_ VARCHAR2(50) NOT NULL,
    data_ DATE NOT NULL,
    nr_randuri_neregula NUMBER(5),
    operatie VARCHAR2(10) NOT NULL
)
```

```
);
```

```
CREATE TABLE TAB_AUDIT_ISTORIC_DETALIAT (
    cod_audit_ist_det NUMBER(4) PRIMARY KEY,
    nume_competitie VARCHAR2(30) NOT NULL,
    puncte_ist NUMBER(5),
    tip_competitie VARCHAR2(20),
    nume_proba VARCHAR2(30),
    data_operatie DATE NOT NULL,
    CONSTRAINT fk_aud_ist_det FOREIGN KEY (cod_audit_ist_det) REFERENCES
    TAB_AUDIT_ISTORIC(cod_audit_ist)
);
select * from TAB_AUDIT_ISTORIC;
select * from TAB_AUDIT_ISTORIC_DETALIAT;
```

```
CREATE SEQUENCE SEQ_AUDIT_ISTORIC
START WITH 1
INCREMENT BY 1;
```

```
CREATE OR REPLACE TRIGGER trigger_istoric_audit
FOR INSERT OR UPDATE ON FRNMP_APP_ADMIN.ISTORIC
COMPOUND TRIGGER
    v_nr_puncte NUMBER := 0;
    v_tip_op VARCHAR2(10);
    v_audit_id NUMBER;
```

```
BEFORE STATEMENT IS
BEGIN
```

```
IF INSERTING THEN
    v_tip_op := 'INSERT';
ELSE
    v_tip_op := 'UPDATE';
END IF;
SELECT SEQ_AUDIT_ISTORIC.NEXTVAL INTO v_audit_id FROM DUAL;
END BEFORE STATEMENT;
AFTER STATEMENT IS
BEGIN
    SELECT COUNT(*) INTO v_nr_puncte
    FROM FRNMP_APP_ADMIN.ISTORIC ist
        JOIN FRNMP_APP_ADMIN.COMPETITIE compe ON ist.cod_competitie =
compe.cod_competitie
        JOIN FRNMP_APP_ADMIN.TIP tip_ ON compe.cod_tip = tip_.cod_tip
    WHERE ((ist.puncte > 15 AND UPPER(tip_.tip_competitie) = 'LOCAL')
        OR (ist.puncte > 150 AND UPPER(tip_.tip_competitie) IN ('EUROPEAN',
'MONDIAL')));
    INSERT INTO TAB_AUDIT_ISTORIC (
        cod_audit_ist, utilizator_, sesiune_, host_, data_, nr_randuri_neregula, operatie
    ) VALUES (
        v_audit_id,
        SYS_CONTEXT('USERENV', 'SESSION_USER'),
        SYS_CONTEXT('USERENV', 'SESSIONID'),
        SYS_CONTEXT('USERENV', 'HOST'),
        SYSDATE,
        v_nr_puncte,
```

```
v_tip_op
);

INSERT INTO TAB_AUDIT_ISTORIC_DETALIAT (
    cod_audit_ist_det, nume_competitie, puncte_ist, tip_competitie, nume_proba,
    data_operatie
)
SELECT
    v_audit_id,
    compe.nume,
    ist.puncte,
    tip_.tip_competitie,
    prob.nume,
    SYSDATE
FROM FRNMP_APP_ADMIN.ISTORIC ist
JOIN FRNMP_APP_ADMIN.PROBA prob ON ist.cod_proba = prob.cod_proba
JOIN FRNMP_APP_ADMIN.COMPETITIE compe ON ist.cod_competitie =
compe.cod_competitie
JOIN FRNMP_APP_ADMIN.TIP tip_ ON compe.cod_tip = tip_.cod_tip
WHERE (ist.puncte > 15 AND UPPER(tip_.tip_competitie) = 'LOCAL')
    OR (ist.puncte > 150 AND UPPER(tip_.tip_competitie) IN ('EUROPEAN',
'MONDIAL'));

END AFTER STATEMENT;
END;
/
commit;
```

## Afisare TAB\_AUDIT\_ISTORIC

COD_AUDIT_IST	UTILIZATOR_	SESIUNE_	HOST_	DATA_	NR_RANDURI_NEREGULA	OPERATIE
34	56 FRNMP_ARBITRU1	81534	Miruna	19-JAN-25		1 UPDATE
35	57 FRNMP_ARBITRU1	81534	Miruna	19-JAN-25		0 UPDATE
36	58 FRNMP_ARBITRU1	81534	Miruna	19-JAN-25		1 UPDATE

## Afisare TAB\_AUDIT\_ISTORIC\_DETALIAT

COD_AUDIT_IST_DET	NUME_COMPETITIE	PUNCTE IST	TIP_COMPETITIE	NUME_PROBA	DATA_OPERATIE
	10 CUPA DE TOAMNA 2024		170 MONDIAL	50 LIBER	04-JAN-25
	56 CUPA DE ANUL NOU 2025		170 EUROPEAN	50 LIBER	19-JAN-25
	58 CUPA DE ANUL NOU 2025		170 EUROPEAN	50 LIBER	19-JAN-25

## c. Politici de auditare

Creați o politică de auditare pentru a înregistra operațiile de modificare a datei de desfășurare a competițiilor în tabela COMPETITIE și afișați un mesaj corespunzător și o altă politică de auditare pentru a înregistra operațiile de modificare a locurilor și punctelor obținute de sportiv în concurs la proba la care a participat din tabelul ISTORIC și afișați un mesaj corespunzător.

```

SELECT POLICY_TEXT,ENABLED,POLICY_NAME FROM
ALL_AUDIT_POLICIES ;

set serveroutput on;

--procedura rulata logat ca frnmp_app_admin
create or replace procedure proc_audit_alert
( object_schema VARCHAR2, object_name VARCHAR2, policy_name VARCHAR2 )is
begin
dbms_output.put_line('Modificare date competitie');
end;
/
--logat iar la sys_orcl
begin
dbms_fga.add_policy(
object_schema=>'frnmp_app_admin',

```

```
object_name=>'COMPETITIE',
policy_name=>'policy_competitie',
audit_column=>'data_start,data_final',
enable=> TRUE,
statement_types=>'UPDATE',
handler_schema=>'frnmp_app_admin',
handler_module=>'proc_audit_alert'

);

end;
/
--procedura rulata logat ca frnmp_app_admin
create or replace procedure proc_audit_alert_ist
( object_schema VARCHAR2, object_name VARCHAR2, policy_name VARCHAR2 )is
begin
dbms_output.put_line('Modificare loc si puncte istoric');
end;
/
--logat iar la sys_orcl
begin
dbms_fga.add_policy(
    object_schema=>'frnmp_app_admin',
    object_name=>'ISTORIC',
    policy_name=>'policy_istoric',
    audit_column=>'loc_per_gen,loc_per_categorie,puncte',
    enable=> TRUE,
    statement_types=>'UPDATE',
    handler_schema=>'frnmp_app_admin',
    handler_module=>'proc_audit_alert_ist'
```

```
 );
end;
/
SELECT POLICY_TEXT,ENABLED FROM ALL_AUDIT_POLICIES
WHERE OBJECT_NAME='ISTORIC';

--verificare ce au inregistrat politicile de audit
SELECT * FROM DBA_FGA_AUDIT_TRAIL;

--cum se da drop la policy
--begin
--  DBMS_FGA.drop_policy (object_schema    => 'frnmp_app_admin',
--                        object_name      => 'ISTORIC',
--                        policy_name     => 'policy_istoric'
--                      );
--end;
--/
--
--begin
--  DBMS_FGA.drop_policy (object_schema    => 'frnmp_app_admin',
--                        object_name      => 'COMPETITIE',
--                        policy_name     => 'policy_competitie'
--                      );
--end;
--/
```

SELECT \* FROM DBA\_FGA\_AUDIT\_TRAIL;

	SESSION_ID	TIMESTAMP	DB_USER	OS_USER	USERHOST	CLIENT_ID	ECONTEXT_ID	EXT_NAME	OBJECT_SCHEMA	OBJECT_NAME	POLICY_NAME
1	80444 03-JAN-25	FRNMP_APP_ADMIN	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN COMPETITIE	POLICY_COMPETITIE		
2	80444 03-JAN-25	FRNMP_APP_ADMIN	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN COMPETITIE	POLICY_COMPETITIE		
3	80470 04-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
4	80470 04-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
5	80474 04-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
6	80481 04-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
7	80481 04-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
8	81520 19-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
9	81525 19-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
10	81534 19-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP_ADMIN ISTORIC	POLICY_ISTORIC		
11	81534 19-JAN-25	FRNMP_ARBITRU1	mirun	Miruna	(null)	(null)	(null)	FRNMP_APP ADMIN ISTORIC	POLICY ISTORIC		

#### 4. Gestiuinea utilizatorilor unei baze de date și a resurselor computaționale

##### A. Proiectarea configurației de management a identităților în baza de date

- 1) identificarea utilizatorilor principali ai aplicației (indivizi, grupuri, organizații, etc);
- 2) identificarea proceselor din cadrul aplicației, cu descompunerea în funcții din aplicație necesare pentru efectuarea lor (workflow-uri);
- 3) construirea matricii proces – utilizator care să evidențieze necesitățile de acces la resurse ale utilizatorilor;
- 4) construirea matricii entitate – proces;
- 5) construirea matricii entitate – utilizator rezultată din matricile proces-utilizator și entitate-proces.

##### 1. Utilizatorii aplicației:

- Sportivi
- Antrenori
- Arbitrii
- Administratorul aplicației și al bazei de date
- Publicul
- Reprezentanți companii

##### 2. Procesele din cadrul aplicației:

P1: Administrare cluburi

P2: Administrare utilizatori

P3: Înscrierea sportivilor la probe în cadrul competițiilor

P4: Consemnarea rezultatelor obținute de sportivi la proba din competiție după ce o finalizează

P5: Sponsorizarea sportivilor

P6: Vizualizarea rezultatelor per categorie a tuturor sportivilor pentru o competiție și o probă aleasă, incluzându-se numele clubului și al antrenorului

P7: Vizualizarea rezultatelor per gen a tuturor sportivilor pentru competiția și proba și a se afla care sunt cei mai buni 3 sportivi și cele mai bune 3 sportive ale competiției (având în vedere numărul total de puncte obținute în cadrul competiției), cu menționarea clubului de apartenență.

P8: Pentru un sportiv ales să se vizualizeze cel mai bun timp pentru toate probele la care a participat de-a lungul timpului, alături de date obținerei.

P9: Pentru un sportiv ales să se observe evoluția per probe de la toate competițiile la care a participat și să se vadă perioadele evoluției și fluctuația acesteia (cum crește/descresc).

P10: Vizualizarea top 3 cluburi pentru o competiție aleasă în funcție de numărul de medalii obținute la categoria

P11: Vizualizarea top 3 sportivi pentru anul, clubul și proba aleasă

P12: Vizualizarea detaliilor sportivilor care nu sunt sponsorizați în data actuală de un sponsor ales și care au participat la competiții europene sau mondiale care s-au desfășurat în București sau în străinătate (dar nu în Japonia) și care au fost în top 16 sportivi pe clasamentul general indiferent de probă.

P13: Gestionarea competițiilor și a tipurilor acestora

P14: Gestionarea probelor

P15: Gestionarea locațiilor

P16: Crearea de contracte între federația de natație și sponsori

P17: Pentru un antrenor să se afișeze pentru sportivii săi care au participat la concursuri proba la care a avut cea mai bună evoluție.

P18: Vizualizarea pentru un club ales, probă și un filtru pentru timpul total de parcurgere (de exemplu > 25) a sportivilor al căror timpi totali respectă filtrul pentru proba aleasă împreună cu timpul obținut de aceștia.

P19: Gestionarea categoriilor de sportivi

### 3. Matricea proces – utilizator

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19
Sportivi					X	X	X	X	X	X									
Antrenori			X		X	X	X	X	X	X						X			
Arbitrii				X		X	X												
Administrator	X	X											X	X	X	X	X	X	
Public					X	X	X	X	X	X						X			
Reprezentant companie					X	X	X	X	X		X	X				X			

#### 4. Matrice entitate – proces

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19
CLUB	S,U,I					S	S			S	S							S	
UTILIZATOR		S,U,I				S				S	S						S	S	
SPORTIV		S,U,I	S	S		S	S	S	S	S	S					S	S		
ANTRENOR		S,U,I				S										S			
PROBA			S	S		S	S	S	S		S			I,U,S		S	S		
COMPETIȚIE			S	S		S	S	S	S		S	I,U,S				S	S		
ISTORIC			I,S	U,S		S	S	S	S	S	S	S				S	S		
ORAȘ												S		I,U,S					
ȚARĂ												S		I,U,S					
TIP												S	I,U,S						
SPONSORIZEAZĂ					I,U							S							
SPONSOR												S			I,U,S				
CATEGORIE						S	S											S,U,I	

#### 5. Matrice entitate – utilizator

	Sportivi	Antrenori	Arbitrii	Administrator	Public	Reprezentant companie
CLUB	S	S	S	S,U,I	S	S
UTILIZATOR	S	S	S	S,U,I	S	S
SPORTIV	S	S	S	S,U,I	S	S
ANTRENOR	S	S	S	S,U,I	S	S
PROBA	S	S	S	S,U,I	S	S
COMPETIȚIE	S	S	S	S,U,I	S	S
ISTORIC	S	I,S	U,S	S	S	S
ORAȘ				S,U,I		S
ȚARĂ				S,U,I		S
TIP				S,U,I		S
SPONSORIZARE						I,U,S
SPONSOR				S,U,I		S

CATEGORIE	S	S	S	S,U,I	S	S
-----------	---	---	---	-------	---	---

## B. Implementarea configurației de management a identităților în baza de date

### 1. Crearea conturilor utilizatorilor

Toate conturile în afară de cel de guest și cel extern au setarea PASSWORD EXPIRE care obligă utilizatorul să își modifice parola la prima sa logare în cont, ceea ce reduce şansele să se credă că parola contului este cunoscută de sys.

```

CREATE USER frnmp_app_admin IDENTIFIED BY parola PASSWORD EXPIRE; --
parola dupa este Admin2025!

GRANT CREATE SESSION TO frnmp_app_admin;

desc dba_users

select username, account_status, AUTHENTICATION_TYPE
from dba_users;

--conturi sportivi

CREATE USER frnmp_app_sportiv1 IDENTIFIED BY parola PASSWORD EXPIRE; --
parola dupa este Sportiv1!

GRANT CREATE SESSION TO frnmp_app_sportiv1;

CREATE USER frnmp_app_sportiv2 IDENTIFIED BY parola PASSWORD EXPIRE; --
parola dupa este Sportiv2!

GRANT CREATE SESSION TO frnmp_app_sportiv2;

CREATE USER frnmp_app_sportiv3 IDENTIFIED BY parola PASSWORD EXPIRE; --
parola dupa este Sportiv3!

GRANT CREATE SESSION TO frnmp_app_sportiv3;

CREATE USER frnmp_app_sportiv4 IDENTIFIED BY parola PASSWORD EXPIRE; --
parola dupa este Sportiv4!

```

```
GRANT CREATE SESSION TO frnmp_app_sportiv4;
```

```
CREATE USER frnmp_app_sportiv5 IDENTIFIED BY parola PASSWORD EXPIRE; --  
parola dupa este Sportiv5!
```

```
GRANT CREATE SESSION TO frnmp_app_sportiv5;
```

```
--conturi antrenori
```

```
CREATE USER frnmp_antrenor1 IDENTIFIED BY parola PASSWORD EXPIRE; --  
pass: Antrenor1!
```

```
GRANT CREATE SESSION TO frnmp_antrenor1;
```

```
CREATE USER frnmp_antrenor2 IDENTIFIED BY parola PASSWORD EXPIRE;--  
pass: Antrenor2!
```

```
GRANT CREATE SESSION TO frnmp_antrenor2;
```

```
CREATE USER frnmp_antrenor3 IDENTIFIED BY parola PASSWORD EXPIRE;--  
pass: Antrenor3!
```

```
GRANT CREATE SESSION TO frnmp_antrenor3;
```

```
CREATE USER frnmp_antrenor4 IDENTIFIED BY parola PASSWORD EXPIRE;--  
pass: Antrenor4!
```

```
GRANT CREATE SESSION TO frnmp_antrenor4;
```

```
--conturi arbitrii
```

```
CREATE USER frnmp_arbitru1 IDENTIFIED BY parola PASSWORD EXPIRE; --  
pass: Arbitru1!
```

```
GRANT CREATE SESSION TO frnmp_arbitru1;
```

```
CREATE USER frnmp_arbitru2 IDENTIFIED BY parola PASSWORD EXPIRE; --  
pass: Arbitru2!
```

```
GRANT CREATE SESSION TO frnmp_arbitru2;

--conturi reprezentanti companii

CREATE USER frnmp_reprezentant1 IDENTIFIED BY parola PASSWORD EXPIRE; -
- pass: Reprezentant1!

GRANT CREATE SESSION TO frnmp_reprezentant1;

CREATE USER frnmp_reprezentant2 IDENTIFIED BY parola PASSWORD EXPIRE; -
- pass: Reprezentant2!

GRANT CREATE SESSION TO frnmp_reprezentant2;

--cont guest

CREATE USER frnmp_guest IDENTIFIED BY guest; -- pass: guest

GRANT CREATE SESSION TO frnmp_guest;

--cont extern

SELECT value FROM v$parameter

WHERE name = 'os_authent_prefix'; --OPS$

--echo %username%
--mirun
--

--echo %USERDOMAIN%
--MIRUNA

--domeniu\username

CREATE USER "OPS$MIRUNA\mirun" IDENTIFIED EXTERNALLY;
GRANT CREATE SESSION TO "OPS$MIRUNA\mirun";
```

```
select username, account_status, AUTHENTICATION_TYPE
from dba_users
WHERE USERNAME LIKE '%FRNMP_%' OR USERNAME LIKE
'OPS$MIRUNA\mirun' ORDER BY CREATED;
```

USERNAME	ACCOUNT_STATUS	AUTHENTICATION_TYPE
FRNMP_APP_ADMIN	OPEN	PASSWORD
FRNMP_APP_SPORTIV1	OPEN	PASSWORD
FRNMP_APP_SPORTIV2	OPEN	PASSWORD
FRNMP_APP_SPORTIV3	OPEN	PASSWORD
FRNMP_APP_SPORTIV4	OPEN	PASSWORD
FRNMP_ANTRENOR1	OPEN	PASSWORD
FRNMP_ANTRENOR4	OPEN	PASSWORD
FRNMP_ANTRENOR2	OPEN	PASSWORD
FRNMP_ANTRENOR3	OPEN	PASSWORD
FRNMP_ARBITRU1	OPEN	PASSWORD
FRNMP_ARBITRU2	OPEN	PASSWORD
FRNMP_REPREZENTANT2	OPEN	PASSWORD
FRNMP_REPREZENTANT1	OPEN	PASSWORD
FRNMP_GUEST	OPEN	PASSWORD
OPS\$MIRUNA\mirun	OPEN	EXTERNAL
FRNMP_APP_SPORTIV5	OPEN	PASSWORD

## 2. Spații de stocare pentru utilizatorii creați

Administratorul a primit spațiu nelimitat, având în vedere că în schema sa se află toate tabelele, mai puțin tabelul SPONSORIZEAZĂ care a fost creat în schema fiecărui sportiv.

De asemenea, utilizatorii de tip sportiv și antrenor vor avea un spațiu de 4M.

```
--quota doar pentru cei care creeaza obiecte
SELECT tablespace_name, ROUND(SUM(bytes) / 1048576) TotalSpace
```

```
FROM dba_data_files
GROUP BY tablespace_name;

ALTER USER frnmp_app_admin QUOTA unlimited ON users;

ALTER USER frnmp_app_sportiv1 QUOTA 4M ON users;
ALTER USER frnmp_app_sportiv2 QUOTA 4M ON users;
ALTER USER frnmp_app_sportiv3 QUOTA 4M ON users;
ALTER USER frnmp_app_sportiv4 QUOTA 4M ON users;
ALTER USER frnmp_app_sportiv5 QUOTA 4M ON users;

ALTER USER frnmp_antrenor1 QUOTA 4M ON users;
ALTER USER frnmp_antrenor2 QUOTA 4M ON users;
ALTER USER frnmp_antrenor3 QUOTA 4M ON users;
ALTER USER frnmp_antrenor4 QUOTA 4M ON users;

ALTER USER frnmp_arbitru1 QUOTA 0 ON users;
ALTER USER frnmp_arbitru2 QUOTA 0 ON users;

ALTER USER frnmp_reprezentant1 QUOTA 0 ON users;
ALTER USER frnmp_reprezentant2 QUOTA 0 ON users;

ALTER USER frnmp_guest QUOTA 0 ON users;
DESC DBA_TS_QUOTAS
SELECT * FROM DBA_TS_QUOTAS
where username in
('FRNMP_APP_ADMIN','FRNMP_APP_SPORTIV1','FRNMP_APP_SPORTIV2');
```

TABLESPACE_NAME	USERNAME	BYTES	MAX_BYTES	BLOCKS	MAX_BLOCKS	DROPPED
USERS	FRNMP_APP_ADMIN	4194304	-1	512	-1	NO
USERS	FRNMP_APP_SPORTIV1	131072	4194304	16	512	NO
USERS	FRNMP_APP_SPORTIV2	131072	4194304	16	512	NO

### 3. Stabilirea accesului la resursele computaționale

Se creează profiluri pentru rolurile create mai sus. Acestea folosesc o funcție custom de verificare a parolei care impune o lungime mai mare sau egală cu 8, să aibă cel puțin o literă mare, mică, cifră și un caracter special și să nu conțină username-ul. În ceea ce privește limitările, acestea diferă prin timpul conexiunii, al timpului de viață al parolei între diferite tipuri de utilizatori.

```

SELECT *
FROM DBA_users;

CREATE OR REPLACE FUNCTION my_verification_function (
    numeuser    VARCHAR2,
    parola_noua  VARCHAR2,
    parola_veche  VARCHAR2
) RETURN BOOLEAN AS
    parola_curenta VARCHAR2(100);
BEGIN

    IF parola_noua = parola_veche THEN
        RETURN FALSE;
    END IF;

    IF LENGTH(parola_noua) < 8 THEN
        RETURN FALSE;
    END IF;

```

```
IF REGEXP_LIKE(parola_noua, '[A-Z]') = FALSE THEN
    RETURN FALSE;
END IF;

IF REGEXP_LIKE(parola_noua, '[a-z]') = FALSE THEN
    RETURN FALSE;
END IF;

IF REGEXP_LIKE(parola_noua, '[0-9]') = FALSE THEN
    RETURN FALSE;
END IF;

IF REGEXP_LIKE(parola_noua, '[!@#$%^&*()_=+{ };;,.<>?/\|]') = FALSE THEN
    RETURN FALSE;
END IF;

IF INSTR(parola_noua, numeuser) > 0 THEN
    RETURN FALSE;
END IF;

RETURN TRUE;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN FALSE;
    WHEN OTHERS THEN
        RETURN FALSE;
END my_verification_function;
```

```
/  
  
CREATE PROFILE frnmp_sp_antr_profile LIMIT  
SESSIONS_PER_USER 1  
CPU_PER_CALL 6000  
CONNECT_TIME 120  
IDLE_TIME 10  
FAILED_LOGIN_ATTEMPTS 3  
PASSWORD_LIFE_TIME 120  
PASSWORD_GRACE_TIME 15  
PASSWORD_VERIFY_FUNCTION my_verification_function;
```

```
CREATE PROFILE frnmp_arb_profile LIMIT  
SESSIONS_PER_USER 1  
CPU_PER_CALL 6000  
CONNECT_TIME 170  
IDLE_TIME 10  
FAILED_LOGIN_ATTEMPTS 3  
PASSWORD_LIFE_TIME 90  
PASSWORD_GRACE_TIME 15  
PASSWORD_VERIFY_FUNCTION my_verification_function;
```

```
CREATE PROFILE frnmp_repr_profile LIMIT  
SESSIONS_PER_USER 1  
CPU_PER_CALL 6000  
CONNECT_TIME 60  
IDLE_TIME 10  
FAILED_LOGIN_ATTEMPTS 3
```

```
PASSWORD_LIFE_TIME 90
PASSWORD_GRACE_TIME 15
PASSWORD_VERIFY_FUNCTION my_verification_function;

CREATE PROFILE frnmp_guest_profile LIMIT
SESSIONS_PER_USER 1
CPU_PER_CALL 3500
CONNECT_TIME 60
IDLE_TIME 10;

CREATE PROFILE frnmp_admin_profile LIMIT
PASSWORD_VERIFY_FUNCTION my_verification_function;

ALTER USER frnmp_app_sportiv1 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_app_sportiv2 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_app_sportiv3 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_app_sportiv4 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_app_sportiv5 PROFILE frnmp_sp_antr_profile;

ALTER USER frnmp_antrenor1 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_antrenor2 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_antrenor3 PROFILE frnmp_sp_antr_profile;
ALTER USER frnmp_antrenor4 PROFILE frnmp_sp_antr_profile;

ALTER USER frnmp_arbitru1 PROFILE frnmp_arb_profile;
ALTER USER frnmp_arbitru2 PROFILE frnmp_arb_profile;

ALTER USER frnmp_reprezentant1 PROFILE frnmp_repr_profile;
```

```
ALTER USER frnmp_reprezentant2 PROFILE frnmp_repr_profile;

ALTER USER frnmp_guest PROFILE frnmp_guest_profile;

ALTER USER frnmp_app_admin PROFILE frnmp_admin_profile;

---- utilizator pentru a testa creare cont cu conditii parola

--DROP USER frnmp_app_sp;

--CREATE USER frnmp_app_sp IDENTIFIED BY parola PASSWORD EXPIRE;

--GRANT CREATE SESSION TO frnmp_app_sp;

--ALTER USER frnmp_app_sp PROFILE frnmp_sp_antr_profile;

--set serveroutput on;

--

--/

--DECLARE

--  result boolean;

--BEGIN

--  result := my_verification_function('frnmp_app_sp', 'Frnmp01234!', 'Frnmp0123');

--  IF result = true THEN

--    DBMS_OUTPUT.PUT_LINE('Parola poate fi schimbata.');

--  ELSE

--    DBMS_OUTPUT.PUT_LINE('Parola nu indeplineste conditiile.');

--  END IF;

--END;

--/

--ALTER USER frnmp_app_sp IDENTIFIED BY "Frnmp01234!";

--cum se procedeaza pentru schimbarea parolei daca este nevoie

ALTER USER frnmp_app_sportiv1 IDENTIFIED BY "Sportiv1!";
```

```
ALTER USER frnmp_app_sportiv2 IDENTIFIED BY "Sportiv2!";  
ALTER USER frnmp_app_sportiv3 IDENTIFIED BY "Sportiv3!";  
ALTER USER frnmp_app_sportiv4 IDENTIFIED BY "Sportiv4!";  
ALTER USER frnmp_app_sportiv5 IDENTIFIED BY "Sportiv5!";
```

```
ALTER USER frnmp_antrenor1 IDENTIFIED BY "Antrenor1!";  
ALTER USER frnmp_antrenor2 IDENTIFIED BY "Antrenor2!";  
ALTER USER frnmp_antrenor3 IDENTIFIED BY "Antrenor3!";  
ALTER USER frnmp_antrenor4 IDENTIFIED BY "Antrenor4!";
```

```
ALTER USER frnmp_arbitru1 IDENTIFIED BY "Arbitru1!";  
ALTER USER frnmp_arbitru2 IDENTIFIED BY "Arbitru2!";
```

```
ALTER USER frnmp_reprezentant1 IDENTIFIED BY "Reprezentant1!";  
ALTER USER frnmp_reprezentant2 IDENTIFIED BY "Reprezentant2!";
```

```
ALTER USER frnmp_app_admin IDENTIFIED BY "Admin2025!";
```

## 5 . Privilegii și roluri

### a. Privilegii sistem și obiect

Dintre privilegiile sistem oferite se va menționa câte un exemplu:

- Permiterea conectării la baza de date a utilizatorului

```
GRANT CREATE SESSION TO frnmp_app_admin;
```

- Crearea de obiecte în schema proprie

```
GRANT CREATE TABLE TO FRNMP_APP_ADMIN;
```

- Efectuarea unei operații LDD/LMD asupra oricărui obiect din schema oricărui utilizator

Exemplu de privilegii oferite când administratorul trebuie să creeze tabelul SPONSORIZEAZA în schemele utilizatorilor sportivi

```
GRANT CREATE ANY TABLE TO FRNMP_APP_ADMIN;  
GRANT CREATE ANY INDEX TO FRNMP_APP_ADMIN;
```

Dintre privilegiile obiect oferite se va menționa câte un exemplu:

- Privilegii tabel

```
GRANT SELECT ON FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU1;
```

- Privilegii vizualizare

```
GRANT SELECT ON frnmp_app_admin.view_proba TO frnmp_antrenor1;  
GRANT UPDATE ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO  
frnmp_reprezentant1;
```

- Privilegii procedură

```
GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO  
frnmp_antrenor2
```

## b. Ierarhii de privilegii

Pentru ierarhiile de privilegii voi exemplifica folosirea lor prin două exemple.

Exemplu 1

Utilizatorul frnmp\_antrenor1 primește mai întâi privilegiile direct, iar apoi și prin rol. Astfel, dacă sunt șterse din rol, acesta tot va avea privilegiile.

```
GRANT SELECT ON frnmp_app_admin.SPORTIV TO frnmp_antrenor1;  
GRANT SELECT ON frnmp_app_admin.ANTRENOR TO frnmp_antrenor1;  
GRANT SELECT ON frnmp_app_admin.PROBA TO frnmp_antrenor1;  
GRANT SELECT ON frnmp_app_admin.COMPETITIE TO frnmp_antrenor1;  
GRANT SELECT ON frnmp_app_admin.ISTORIC TO frnmp_antrenor1;  
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO frnmp_antrenor1;
```

```
CREATE ROLE vizualizari_best_proba_de_antrenor;
```

```
GRANT SELECT ON frnmp_app_admin.SPORTIV TO
```

```

vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminUTILIZATOR TO
vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminANTRENOR TO
vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminPROBA TO
vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminCOMPETITIE TO
vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminISTORIC TO
vizualizari_best_proba_de_antrenor;

GRANT vizualizari_best_proba_de_antrenor TO
frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;

```

### Exemplul 2

Utilizatorul frnmp\_reprezentant2 primește mai întâi un privilegiile direct de select și update pe o vizualizare, iar apoi prin rol, pe lângă acest privilegiu, primește și alte privilegii. În plus rolul insert\_viz\_spons\_repr are la bază rolul viz\_sponsoriz la care mai adaugă două privilegii.

```

CREATE ROLE viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminCLUB TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminANTRENOR TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminPROBA TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminCATEGORIE TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminSPORTIV TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminUTILIZATOR TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminCOMPETITIE TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminISTORIC TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminORAS TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminTARA TO viz_sponsoriz;

GRANT SELECT ON frnmp_app_adminTIP TO viz_sponsoriz;

```

```
GRANT SELECT ON frnmp_app_admin.SPONSOR TO viz_sponsoriz;
GRANT SELECT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE_2 TO
    viz_sponsoriz;

GRANT SELECT,INSERT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    frnmp_reprezentant2;

create role insert_viz_spons_repr;
grant viz_sponsoriz to insert_viz_spons_repr;
GRANT SELECT,INSERT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    insert_viz_spons_repr;
GRANT UPDATE ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    insert_viz_spons_repr;

GRANT insert_viz_spons_repr TO frnmp_reprezentant2;
```

c. Privilegii asupra obiectelor dependente

Dintre exemple se vor menționa privilegii asupra vizualizărilor și procedurilor în această ordine.

```
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO
frnmp_antrenor4;

GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_categorie TO
frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;
```

Toate privilegiile oferite (mai puțin cele de creare a sesiunii care se regăsesc la 4.b)

```
-- privilegii
SELECT substr(grantee,1,20) grantee, owner,
substr(table_name,1,15) table_name,
```

```
grantor,privilege
FROM DBA_TAB_PRIVS
WHERE grantee like '%FRNMP%';

--PENTRU ADMIN
-- SA CREEZE TABELE IN SCHEMA LUI
GRANT CREATE TABLE TO FRNMP_APP_ADMIN;
GRANT CREATE ANY TABLE TO FRNMP_APP_ADMIN;
GRANT CREATE ANY INDEX TO FRNMP_APP_ADMIN;

--SA CREEZE ROLURI
GRANT CREATE ROLE TO FRNMP_APP_ADMIN;

--PENTRU A FACE TABELA SPONSORIZEAZA IN SCHEMA SPORTIVILOR
GRANT REFERENCES ON FRNMP_APP_ADMIN.SPONSOR TO
    FRNMP_APP_SPORTIV1;
GRANT REFERENCES ON FRNMP_APP_ADMIN.SPORTIV TO
    FRNMP_APP_SPORTIV1;

GRANT REFERENCES ON FRNMP_APP_ADMIN.SPONSOR TO
    FRNMP_APP_SPORTIV2;
GRANT REFERENCES ON FRNMP_APP_ADMIN.SPORTIV TO
    FRNMP_APP_SPORTIV2;

GRANT REFERENCES ON FRNMP_APP_ADMIN.SPONSOR TO
    FRNMP_APP_SPORTIV3;
GRANT REFERENCES ON FRNMP_APP_ADMIN.SPORTIV TO
    FRNMP_APP_SPORTIV3;

GRANT REFERENCES ON FRNMP_APP_ADMIN.SPONSOR TO
    FRNMP_APP_SPORTIV4;
GRANT REFERENCES ON FRNMP_APP_ADMIN.SPORTIV TO
```

```
FRNMP_APP_SPORTIV4;

GRANT REFERENCES ON FRNMP_APP_ADMIN.SPONSOR TO
    FRNMP_APP_SPORTIV5;

GRANT REFERENCES ON FRNMP_APP_ADMIN.SPORTIV TO
    FRNMP_APP_SPORTIV5;

--pentru a putea cripta

GRANT EXECUTE ON DBMS_CRYPTO TO FRNMP_APP_ADMIN;
GRANT CREATE TRIGGER TO FRNMP_APP_ADMIN;
GRANT CREATE PROCEDURE TO FRNMP_APP_ADMIN;

-- auditare

GRANT AUDIT ANY TO FRNMP_APP_ADMIN;
GRANT EXECUTE ON DBMS_FGA TO FRNMP_APP_ADMIN;
-----

-- ANTRENORUL CARE VREA SA INSEREZE IN ISTORIC

GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO frnmp_antrenor1;
GRANT CREATE VIEW TO FRNMP_APP_ADMIN;
GRANT SELECT ON frnmp_app_admin.view_sportivi_antrenor TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_admin.view_competitie TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_admin.view_proba TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO
    frnmp_antrenor1;
---

GRANT SELECT ON frnmp_app_admin.CLUB TO
    frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;
GRANT SELECT ON frnmp_app_admin.CATEGORIE TO
    frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;
```

```
--arbitrul updateaza

GRANT SELECT ON FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU1;
GRANT SELECT ON frnmp_app_admin.view_istoric_arbitru TO FRNMP_ARBITRU1;
GRANT UPDATE (loc_per_gen, loc_per_categorie, timp_total, puncte) ON
    FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU1;

GRANT SELECT ON FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU2;
GRANT SELECT ON frnmp_app_admin.view_istoric_arbitru TO FRNMP_ARBITRU2;
GRANT UPDATE (loc_per_gen, loc_per_categorie, timp_total, puncte) ON
    FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU2;

CREATE ROLE rol_arbitru_extins;

GRANT SELECT ON FRNMP_APP_ADMIN.CLUB TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMINUTILIZATOR TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.SPORTIV TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.ANTRENOR TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.PROBA TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.COMPETITIE TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.ISTORIC TO rol_arbitru_extins;
GRANT SELECT ON FRNMP_APP_ADMIN.CATEGORIE TO rol_arbitru_extins;
GRANT UPDATE (loc_per_gen, loc_per_categorie, timp_total, puncte) ON
    FRNMP_APP_ADMIN.ISTORIC TO rol_arbitru_extins;

GRANT rol_arbitru_extins TO FRNMP_ARBITRU1,FRNMP_ARBITRU2;

--PENTRU A INTRODUCE UN SPONSOR DATE IN TABELUL SPONSORIZEAZA
-- CARE SE AFLA IN SCHEMELE SPORTIVILOR
```

```
--GRANT CREATE VIEW TO FRNMP_APP_ADMIN; -- avuta deja  
GRANT SELECT ON frnmp_app_sportiv1.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT SELECT ON frnmp_app_sportiv2.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT SELECT ON frnmp_app_sportiv3.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT SELECT ON frnmp_app_sportiv4.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT SELECT ON frnmp_app_sportiv5.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
--se da reprezentantilor de companii privilegiul de interogare al vizualizarii  
GRANT SELECT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO  
    frnmp_reprezentant1;  
  
--GRANT CREATE TRIGGER TO FRNMP_APP_ADMIN; --deja il aveam  
--admin trebuie sa poata insera date in tabela sponsorizeaza  
GRANT INSERT ON frnmp_app_sportiv1.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT INSERT ON frnmp_app_sportiv2.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT INSERT ON frnmp_app_sportiv3.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT INSERT ON frnmp_app_sportiv4.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
GRANT INSERT ON frnmp_app_sportiv5.SPONSORIZEAZA TO FRNMP_APP_ADMIN  
    WITH GRANT OPTION;  
  
--SYS ii da privilegiu de insert prin view catre reprezentantii de companii  
GRANT INSERT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO  
    frnmp_reprezentant1;  
  
--TREBUIE SA POATA FACE UPDATE  
GRANT UPDATE ON frnmp_app_sportiv1.SPONSORIZEAZA TO
```

```
FRNMP_APP_ADMIN WITH GRANT OPTION;
GRANT UPDATE ON frnmp_app_sportiv2.SPONSORIZEAZA TO
    FRNMP_APP_ADMIN WITH GRANT OPTION;
GRANT UPDATE ON frnmp_app_sportiv3.SPONSORIZEAZA TO
    FRNMP_APP_ADMIN WITH GRANT OPTION;
GRANT UPDATE ON frnmp_app_sportiv4.SPONSORIZEAZA TO
    FRNMP_APP_ADMIN WITH GRANT OPTION;
GRANT UPDATE ON frnmp_app_sportiv5.SPONSORIZEAZA TO
    FRNMP_APP_ADMIN WITH GRANT OPTION;
GRANT UPDATE ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    frnmp_reprezentant1;
SELECT tablespace_name,
       file_name,
       bytes / 1024 / 1024 AS size_mb,
       maxbytes / 1024 / 1024 AS max_size_mb
FROM dba_data_files
WHERE tablespace_name = 'USERS';

--RULATE CA UTILIZATOR ADMIN
CREATE ROLE inscrieri_sp;
--privilegii date direct utilizatorului
GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO frnmp_antrenor2;
GRANT SELECT ON frnmp_app_admin.view_sportivi_antrenor TO frnmp_antrenor2;
GRANT SELECT ON frnmp_app_admin.view_competitie TO frnmp_antrenor2;
GRANT SELECT ON frnmp_app_admin.view_proba TO frnmp_antrenor2;
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO
    frnmp_antrenor2;
-- privilegii date rolului
GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO inscrieri_sp;
```

```
GRANT SELECT ON frnmp_app_admin.view_sportivi_antrenor TO inscrieri_sp;
GRANT SELECT ON frnmp_app_admin.view_competitie TO inscrieri_sp;
GRANT SELECT ON frnmp_app_admin.view_proba TO inscrieri_sp;
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO inscrieri_sp;
--acord rolul utilizatorului
GRANT inscrieri_sp TO frnmp_antrenor2;
--

GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO frnmp_antrenor3;
GRANT SELECT ON frnmp_app_admin.view_sportivi_antrenor TO frnmp_antrenor3;
GRANT SELECT ON frnmp_app_admin.view_competitie TO frnmp_antrenor3;
GRANT SELECT ON frnmp_app_admin.view_proba TO frnmp_antrenor3;
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO
    frnmp_antrenor3;
--

GRANT EXECUTE ON frnmp_app_admin.antrenor_insert_istoric TO frnmp_antrenor4;
GRANT SELECT ON frnmp_app_admin.view_sportivi_antrenor TO frnmp_antrenor4;
GRANT SELECT ON frnmp_app_admin.view_competitie TO frnmp_antrenor4;
GRANT SELECT ON frnmp_app_admin.view_proba TO frnmp_antrenor4;
GRANT SELECT ON frnmp_app_admin.view_istoric_sportivi_antrenor TO
    frnmp_antrenor4;
GRANT inscrieri_sp TO frnmp_antrenor4;

-- P6
CREATE ROLE vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.CLUB TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.SPORTIV TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO vizualizari_rezultate;
```

```
GRANT SELECT ON frnmp_app_admin.ANTRENOR TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.PROBA TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.COMPETITIE TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.ISTORIC TO vizualizari_rezultate;
GRANT SELECT ON frnmp_app_admin.CATEGORIE TO vizualizari_rezultate;

GRANT vizualizari_rezultate TO frnmp_app_sportiv1;

--REVOKE vizualizari_rezultate from frnmp_app_sportiv1;

GRANT SELECT ON frnmp_app_admin.CLUB TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.SPORTIV TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.ANTRENOR TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.PROBA TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.COMPETITIE TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.ISTORIC TO frnmp_app_sportiv2;
GRANT SELECT ON frnmp_app_admin.CATEGORIE TO frnmp_app_sportiv2;
GRANT vizualizari_rezultate TO frnmp_app_sportiv2;

GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_categorie TO
    frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;
GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_categorie TO
    frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;
GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_categorie TO
    frnmp_arbitru1,frnmp_arbitru2,frnmp_reprezentant1,frnmp_reprezentant2,frnmp_gue
    st;

--P7 CU EXECUTE A TREIA METODA (pentru sportivul 1 si 2 au ramas granturile si
    rolurile date pentru p6)
```

```
GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_gen_si_total_puncte  
TO frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;  
  
GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_gen_si_total_puncte  
TO frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;  
  
GRANT EXECUTE ON frnmp_app_admin.vizualizare_rezultate_per_gen_si_total_puncte  
TO  
frnmp_arbitru1,frnmp_arbitru2,frnmp_reprezentant1,frnmp_reprezentant2,frnmp_gue  
st;
```

--P8 pentru restul de sportivi (3,4,5), antrenori, public si reprezentant companie

```
GRANT EXECUTE ON frnmp_app_admin.get_best_times_for_sportiv TO  
frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;
```

```
GRANT EXECUTE ON frnmp_app_admin.get_best_times_for_sportiv TO  
frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4,frnmp_reprezen  
tant1,frnmp_reprezentant2,frnmp_guest;
```

--p9

```
GRANT EXECUTE ON frnmp_app_admin.evolutie_sportiv TO  
frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;
```

```
GRANT EXECUTE ON frnmp_app_admin.evolutie_sportiv TO  
frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4,frnmp_reprezen  
tant1,frnmp_reprezentant2,frnmp_guest;
```

--p10

```
GRANT EXECUTE ON frnmp_app_admin.top_cluburi TO  
frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;
```

```
GRANT EXECUTE ON frnmp_app_admin.top_cluburi TO  
frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4,frnmp_guest;
```

```
--revoke EXECUTE ON frnmp_app_admin.top_cluburi from  
frnmp_reprezentant1,frnmp_reprezentant2;
```

--p11

```
GRANT EXECUTE ON frnmp_app_admin.top_sportivi_an_cl_proba TO
```

```
frnmp_app_sportiv3,frnmp_app_sportiv4,frnmp_app_sportiv5;  
GRANT EXECUTE ON frnmp_app_admin.top_sportivi_an_cl_proba TO  
    frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4,frnmp_reprezen-  
    tant1,frnmp_reprezentant2,frnmp_guest;  
  
--p12  
GRANT SELECT ON frnmp_app_admin.CLUB TO frnmp_reprezentant1;--de la p6  
GRANT SELECT ON frnmp_app_admin.ANTRENOR TO frnmp_reprezentant1;--de la p6  
GRANT SELECT ON frnmp_app_admin.PROBA TO frnmp_reprezentant1;--de la p6  
GRANT SELECT ON frnmp_app_admin.CATEGORIE TO frnmp_reprezentant1;--de la p6  
  
GRANT SELECT ON frnmp_app_admin.SPORTIV TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.COMPETITIE TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.ISTORIC TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.ORAS TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.TARA TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.TIP TO frnmp_reprezentant1;  
GRANT SELECT ON frnmp_app_admin.SPONSOR TO frnmp_reprezentant1;  
GRANT SELECT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE_2 TO  
    frnmp_reprezentant1;  
  
CREATE ROLE viz_sponsoriz;  
GRANT SELECT ON frnmp_app_admin.CLUB TO viz_sponsoriz;  
GRANT SELECT ON frnmp_app_admin.ANTRENOR TO viz_sponsoriz;  
GRANT SELECT ON frnmp_app_admin.PROBA TO viz_sponsoriz;  
GRANT SELECT ON frnmp_app_admin.CATEGORIE TO viz_sponsoriz;  
  
GRANT SELECT ON frnmp_app_admin.SPORTIV TO viz_sponsoriz;
```

```
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminCOMPETITIE TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminISTORIC TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminORAS TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminTARA TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminTIP TO viz_sponsoriz;
GRANT SELECT ON frnmp_app_adminSPONSOR TO viz_sponsoriz;
GRANT SELECT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE_2 TO viz_sponsoriz;

GRANT viz_sponsoriz TO frnmp_reprezentant1;
GRANT SELECT,INSERT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    frnmp_reprezentant2;

create role insert_viz_spons_repr;
grant viz_sponsoriz to insert_viz_spons_repr;
GRANT SELECT,INSERT ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    insert_viz_spons_repr;
GRANT UPDATE ON FRNMP_APP_ADMIN.VIZ_SPONSORIZARE TO
    insert_viz_spons_repr;

GRANT insert_viz_spons_repr TO frnmp_reprezentant2;

--p17
--antrenor1
GRANT SELECT ON frnmp_app_adminSPORTIV TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_adminANTRENOR TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_adminPROBA TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_adminCOMPETITIE TO frnmp_antrenor1;
GRANT SELECT ON frnmp_app_adminISTORIC TO frnmp_antrenor1;
```

```
GRANT SELECT ON frnmp_app_adminUTILIZATOR TO frnmp_antrenor1;

CREATE ROLE vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminSPORTIV TO
    vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminUTILIZATOR TO
    vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminANTRENOR TO
    vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminPROBA TO vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminCOMPETITIE TO
    vizualizari_best_proba_de_antrenor;

GRANT SELECT ON frnmp_app_adminISTORIC TO vizualizari_best_proba_de_antrenor;

GRANT vizualizari_best_proba_de_antrenor TO
    frnmp_antrenor1,frnmp_antrenor2,frnmp_antrenor3,frnmp_antrenor4;

GRANT vizualizari_best_proba_de_antrenor TO
    frnmp_reprezentant1,frnmp_reprezentant2,frnmp_guest;

--context

create or replace directory DIREXP AS
'D:\An2Master\securitate\Securitateproiect\mascarefolder'; --sys

GRANT READ,WRITE ON DIRECTORY DIREXP TO frnmp_app_admin;
```

În urma rulării

```
SELECT substr(grantee,1,20) grantee, owner,
       substr(table_name,1,15) table_name,
       grantor,privilege
```

```
FROM DBA_TAB_PRIVS
WHERE grantee like '%FRNMP%';
```

GRANTEE	OWNER	TABLE_NAME	GRANTOR	PRIVILEGE
11 FRNMP_APP_ADMIN	FRNMP_APP_SPORTIV4	SPONSORIZEAZA	FRNMP_APP_SPORTIV4	UPDATE
12 FRNMP_APP_ADMIN	FRNMP_APP_SPORTIV5	SPONSORIZEAZA	FRNMP_APP_SPORTIV5	SELECT
13 FRNMP_APP_ADMIN	FRNMP_APP_SPORTIV5	SPONSORIZEAZA	FRNMP_APP_SPORTIV5	INSERT
14 FRNMP_APP_ADMIN	FRNMP_APP_SPORTIV5	SPONSORIZEAZA	FRNMP_APP_SPORTIV5	UPDATE
15 FRNMP_REPREZENTANT1	FRNMP_APP_ADMIN	VIZ_SPONSORIZAR	FRNMP_APP_ADMIN	SELECT
16 FRNMP_REPREZENTANT1	FRNMP_APP_ADMIN	VIZ_SPONSORIZAR	FRNMP_APP_ADMIN	INSERT
17 FRNMP_REPREZENTANT2	FRNMP_APP_ADMIN	VIZ_SPONSORIZAR	FRNMP_APP_ADMIN	INSERT
18 FRNMP_REPREZENTANT2	FRNMP_APP_ADMIN	VIZ_SPONSORIZAR	FRNMP_APP_ADMIN	SELECT
19 FRNMP_REPREZENTANT1	FRNMP_APP_ADMIN	VIZ_SPONSORIZAR	FRNMP_APP_ADMIN	UPDATE
20 FRNMP_APP_SPORTIV2	FRNMP_APP_ADMIN	CLUB1	FRNMP_APP_ADMIN	SELECT
21 FRNMP_APP_SPORTIV2	FRNMP_APP_ADMIN	UTILIZATOR1	FRNMP_APP_ADMIN	SELECT
22 FRNMP_REPREZENTANT1	FRNMP_APP_ADMIN	UTILIZATOR1	FRNMP_APP_ADMIN	SELECT
23 FRNMP_ANTRENOR1	FRNMP_APP_ADMIN	UTILIZATOR1	FRNMP_APP_ADMIN	SELECT
24 FRNMP_APP_SPORTIV1	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	REFERENCES
25 FRNMP_APP_SPORTIV2	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	REFERENCES
26 FRNMP_APP_SPORTIV3	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	REFERENCES
27 FRNMP_APP_SPORTIV4	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	REFERENCES
28 FRNMP_APP_SPORTIV5	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	REFERENCES
29 FRNMP_APP_SPORTIV2	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	SELECT
30 FRNMP_REPREZENTANT1	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	SELECT
31 FRNMP_ANTRENOR1	FRNMP_APP_ADMIN	SPORTIV1	FRNMP_APP_ADMIN	SELECT
32 FRNMP_APP_ADMIN	SYS	DIREXP	SYS	READ
33 FRNMP_APP_ADMIN	SYS	DIREXP	SYS	WRITE
68 FRNMP_ANTRENOR1	FRNMP_APP_ADMIN	ANTRENOR_INSERT	FRNMP_APP_ADMIN	EXECUTE
69 FRNMP_ANTRENOR2	FRNMP_APP_ADMIN	ANTRENOR_INSERT	FRNMP_APP_ADMIN	EXECUTE
70 FRNMP_ANTRENOR3	FRNMP_APP_ADMIN	ANTRENOR_INSERT	FRNMP_APP_ADMIN	EXECUTE
71 FRNMP_ANTRENOR4	FRNMP_APP_ADMIN	ANTRENOR_INSERT	FRNMP_APP_ADMIN	EXECUTE
72 FRNMP_ANTRENOR1	FRNMP_APP_ADMIN	VIEW_COMPETITIE	FRNMP_APP_ADMIN	SELECT
73 FRNMP_ANTRENOR2	FRNMP_APP_ADMIN	VIEW_COMPETITIE	FRNMP_APP_ADMIN	SELECT

## 6. Aplicațiile pe baza de date și securitatea datelor

### A. Contextul aplicației

Antrenorii nu pot introduce sportivii la concursuri în weekend, iar în timpul săptămânii doar în intervalul 9-17. În ceea ce privește arbitrii, aceștia pot face modificări ale datelor din tabela istoric doar în weekend și în ziua de luni când sunt contestațiile.

Cod rulat ca sys

```

set serveroutput on;

SELECT trim(TO_CHAR(SYSDATE, 'DAY',
      'NLS_DATE_LANGUAGE=AMERICAN')) AS ZIUA_CURENTA
FROM DUAL;

CREATE OR REPLACE PROCEDURE proced_aplicatie_ctx IS
v_ora number(2):= to_number(to_char(sysdate,'hh24'));
v_zi_sapt VARCHAR2(10) := TO_CHAR(SYSDATE, 'DAY',
      'NLS_DATE_LANGUAGE=AMERICAN');

BEGIN
  v_zi_sapt := TRIM(v_zi_sapt);
  IF v_zi_sapt IN ('SATURDAY', 'SUNDAY') THEN
    DBMS_OUTPUT.PUT_LINE('Nu puteti inscrie sportivii la competitii in
weekend.');
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX',
      'PROGRAM_POTRIVIT', 'NU');
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX',
      'ARBITRAJ_POTRIVIT', 'DA');
  ELSIF v_ora < 9 OR v_ora > 17 THEN
    DBMS_OUTPUT.PUT_LINE('Nu puteti inscrie sportivii la competitii in afara
programului.');
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX',
      'PROGRAM_POTRIVIT', 'NU');
  ELSE
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX',
      'PROGRAM_POTRIVIT', 'DA');
  END IF;

```

```
IF v_zi_sapt = 'MONDAY' THEN
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX', 'ARBITRAJ_POTRIVIT',
        'DA');
END IF;

IF v_zi_sapt NOT IN ('MONDAY','SATURDAY', 'SUNDAY') THEN
    DBMS_SESSION.SET_CONTEXT('APLICATIE_CTX', 'ARBITRAJ_POTRIVIT',
        'NU');
END IF;
END;
/
CREATE CONTEXT APLICATIE_CTX USING proced_aplicatie_ctx;
EXEC proced_aplicatie_ctx();

create or replace trigger tr_after_logon
after logon
on database
declare
    v_utilizator varchar2(30):=sys_context('userenv','session_user');
begin
    if lower(v_utilizator) like 'frnmp_antrenor%' or lower(v_utilizator) like
        'frnmp_arbitru%' then
        proced_aplicatie_ctx();
    end if;
end;
/

```

Cod rulat ca admin

--pentru antrenori

```
create or replace trigger inscriere_sp_trigger
```

```
before insert on frnmp_app_admin.ISTORIC
```

```
begin
```

```
    if sys_context('APLICATIE_CTX','PROGRAM_POTRIVIT') = 'NU' then
```

```
        raise_application_error(-20001,'Sunteti in afara programului');
```

```
    end if;
```

```
end;
```

```
/
```

--pentru arbitrii

```
create or replace trigger update_inscriere_sp_trigger
```

```
before UPDATE OF loc_per_gen, loc_per_categorie, timp_total, puncte on
frnmp_app_admin.ISTORIC
```

```
begin
```

```
    if sys_context('APLICATIE_CTX','ARBITRAJ_POTRIVIT') = 'NU' then
```

```
        raise_application_error(-20002,'Sunteti in afara programului de arbitrage si
contestatii');
```

```
    end if;
```

```
end;
```

```
/
```

```
GRANT UPDATE (loc_per_gen, loc_per_categorie, timp_total, puncte) ON
FRNMP_APP_ADMIN.ISTORIC TO FRNMP_ARBITRU1;
```

```
SELECT sys_context('APLICATIE_CTX', 'PROGRAM_POTRIVIT') FROM dual;
```

Cum arată când vreau să inserez logată ca frnmp\_antrenor1 un sportiv la o competiție în afara programului. (am rulat într-o zi din săptămână la ora 02:57 AM)

```

37 | --testare in afara programului
38 | EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE ANUL NOU 2025', 0, 0, 0, 0);

```

Script Output X  
Task completed in 0.092 seconds

```

Error starting at line : 38 in command -
BEGIN frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE ANUL NOU 2025', 0, 0, 0, 0); END;
Error report -
ORA-20001: Sunteti in afara programului
ORA-06512: at "FRNMP_APP_ADMIN.INSCHRIERE_SP_TRIGGER", line 3
ORA-04088: error during execution of trigger 'FRNMP_APP_ADMIN.INSCHRIERE_SP_TRIGGER'
ORA-06512: at "FRNMP_APP_ADMIN.ANTRENOR_INSERT_ISTORIC", line 30
ORA-06512: at line 1

```

Cum arată când vreau să fac update logată ca frnmp\_arbitru1 pentru rezultatele obținute de un sportiv la o competiție în afara programului. (am rulat într-o zi de joi)

```

101 | UPDATE frnmp_app_admin.istoric
102 | SET loc_per_gen = 1,loc_per_categorie=1,timp_total=30,puncte=150
103 | WHERE cod_sportiv = 5 and cod_proba = 3 and cod_competitie = 3;
104 | COMMIT;
105 |

```

Script Output X  
Task completed in 0.055 seconds

```

Error starting at line : 101 in command -
UPDATE frnmp_app_admin.istoric
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=30,puncte=150
WHERE cod_sportiv = 5 and cod_proba = 3 and cod_competitie = 3
Error at Command Line : 101 Column : 24
Error report -
SQL Error: ORA-20002: Sunteti in afara programului de arbitrage si contestatii
ORA-06512: at "FRNMP_APP_ADMIN.UPDATE_INSCHRIERE_SP_TRIGGER", line 3
ORA-04088: error during execution of trigger 'FRNMP_APP_ADMIN.UPDATE_INSCHRIERE_SP_TRIGGER'

```

### B. SQL Injection

Pentru un nume de club dat, nume de probă și un filtru pentru timpul total (de exemplu < 26) afișați numele, prenumele, numărul legitimației sportivului, timpul total obținut de acesta și numele competiției la care l-a obținut.

```
CREATE OR REPLACE PROCEDURE sportivi_raport (
```

```
    club_input IN VARCHAR2,
    proba_input IN VARCHAR2,
    filtru_timp_total IN VARCHAR2,
```

```
result OUT SYS_REFCURSOR
) AS
cerere VARCHAR2(1000);
BEGIN
cerere := 'SELECT u.nume, u.prenume, s.numar_legitimatie, i.timp_total,co.nume
FROM
UTILIZATOR u
JOIN CLUB c ON u.cod_club = c.cod_club
JOIN SPORTIV s ON u.cod_utilizator = s.cod_sportiv
JOIN ISTORIC i ON s.cod_sportiv = i.cod_sportiv
JOIN COMPETITIE co ON i.cod_competitie = co.cod_competitie
WHERE c.cod_club IN
(SELECT cod_club FROM club
WHERE nume = "" || club_input || ""
)
AND i.cod_proba IN
(SELECT cod_proba FROM proba
WHERE nume = "" || proba_input || ""
)
AND timp_total '|| filtru_timp_total
;
OPEN result FOR cerere;
END;
/
--SELECT u.nume, u.prenume, s.numar_legitimatie, i.timp_total,co.nume
--    FROM
--    UTILIZATOR u
```

```
-- JOIN CLUB c ON u.cod_club = c.cod_club
-- JOIN SPORTIV s ON u.cod_utilizator = s.cod_sportiv
-- JOIN ISTORIC i ON s.cod_sportiv = i.cod_sportiv
-- JOIN COMPETITIE co ON i.cod_competitie = co.cod_competitie
-- WHERE c.cod_club IN
--   (SELECT cod_club FROM club
--    WHERE nume = 'AQUA TEAM'
--    )
-- AND i.cod_proba IN
--   (SELECT cod_proba FROM proba
--    WHERE nume = '50 LIBER'
--    )
-- AND timp_total < 26
-- ;
SELECT * FROM CLUB;
SELECT * FROM ISTORIC;
SELECT * FROM SPORTIV;
SELECT * FROM UTILIZATOR;
SELECT * FROM PROBA;
```

Folosirea corectă a procedurii

```
set serveroutput on;
```

```
DECLARE
```

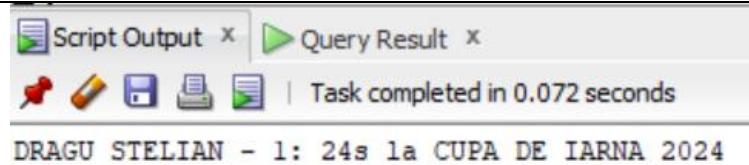
```
  result SYS_REFCURSOR;
  nume VARCHAR2(20);
  prenume VARCHAR2(20);
  numar_legitimatie NUMBER(5);
  timp_total NUMBER(5);
```

```

nume_comp VARCHAR2(30);

BEGIN
    sportivi_raport('AQUA TEAM','50 LIBER','< 26',result);
    LOOP
        FETCH result INTO nume, prenume, numar_legitimatie,timp_total, nume_comp;
        EXIT WHEN result%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE(nume || ' ' || prenume || '-' || numar_legitimatie || ':' || 
            timp_total || 's la ' || nume_comp);
    END LOOP;
    CLOSE result;
END;
/

```



PL/SQL procedure successfully completed.

#### Atacuri

1. Injectare simplă cu OR 1=1

```

DECLARE
    result SYS_REFCURSOR;
    nume VARCHAR2(20);
    prenume VARCHAR2(20);
    numar_legitimatie NUMBER(5);
    timp_total NUMBER(5);
    nume_comp VARCHAR2(30);
BEGIN
    sportivi_raport('AQUA TEAM" OR 1=1 --','50 LIBER" OR 1=1 --','is not null',result); -- acum
    arata tot din istoric

```

```
--sportivi_raport_securizat('AQUA TEAM" OR 1=1 --','50 LIBER" OR 1=1 --,'is not
null',result);

LOOP
  FETCH result INTO nume, prenume, numar_legitimatie,timp_total, nume_comp;
  EXIT WHEN result%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(nume || ' ' || prenume || '-' || numar_legitimatie || ':' ||
timp_total || 's la ' || nume_comp);
END LOOP;
CLOSE result;
END;
/

```

#### Rulare sportivi\_raport – Atacatorul obține toate datele din ISTORIC

Script Output    Query Result    Task completed in 0.077 seconds

DRAGU STELIAN - 1: 27s la CUPA DE ANUL NOU 2025  
DRAGU STELIAN - 1: 30s la CUPA DE ANUL NOU 2025  
DRAGU STELIAN - 1: 24s la CUPA DE IARNA 2024  
MARCU MARIA - 3: 28s la CUPA DE IARNA 2024  
MICU DENISA - 2: 29s la CUPA DE IARNA 2024  
DRAGU STELIAN - 1: 32s la CUPA DE TOAMNA 2024  
MISTI SAR - 5: 32s la CUPA DE TOAMNA 2024  
DRAGU STELIAN - 1: 27s la CUPA DE TOAMNA 2024  
MISTI SAR - 5: 25s la CUPA DE TOAMNA 2024  
DRAGU STELIAN - 1: 60s la CUPA DE TOAMNA 2024  
MARCU MARIA - 3: 33s la CUPA DE TOAMNA 2024  
MARCU MARIA - 3: 44s la CUPA DE TOAMNA 2024  
STAN FILIP - 4: 20s la CUPA DE TOAMNA 2024  
STAN FILIP - 4: 26s la CUPA DE TOAMNA 2024  
DRAGU STELIAN - 1: 26s la CUPA DE TOAMNA 2024  
MICU DENISA - 2: 27s la CUPA DE TOAMNA 2024  
MICU DENISA - 2: 46s la CUPA DE TOAMNA 2024

PL/SQL procedure successfully completed.

Rulare sportivi\_raport\_securizat – nu mai permite injectarea

```
Error report -
ORA-06502: PL/SQL: numeric or value error
ORA-06512: at "SYS.DBMS_ASSERT", line 493
ORA-06512: at "SYS.DBMS_ASSERT", line 583
ORA-06512: at "FRNMP_APP_ADMIN.SPORTIVI_RAPORT_SECURIZAT", line 22
ORA-06512: at line 10
06502. 00000 - "PL/SQL: numeric or value error%$"
*Cause: An arithmetic, numeric, string, conversion, or constraint error
        occurred. For example, this error occurs if an attempt is made to
        assign the value NULL to a variable declared NOT NULL, or if an
        attempt is made to assign an integer larger than 99 to a variable
        declared NUMBER(2).
*Action: Change the data, how it is manipulated, or how it is declared so
        that values do not violate constraints.
```

## 2. SQL Injection clasic bazat pe UNION

Atacatorul vrea să afle numele, prenumele, codurile de utilizatori și username-urile pentru utilizatori.

Coloanele din union trebuie să facă match ca tip cu cele din query din procedura inițială (u.nume, u.prenume, s.numar\_legitimatie, i.timp\_total, co.nume). Acest fapt se face prin încercări: mai întâi trebuie ca atacatorul să găsească numărul corect de atribute care trebuie puse în union, apoi trebuie potrivite tipurile atributelor.

```
DECLARE
    result SYS_REFCURSOR;
    nume VARCHAR2(20);
    prenume VARCHAR2(20);
    numar_legitimatie NUMBER(5);
    timp_total NUMBER(5);
    nume_comp VARCHAR2(30);

BEGIN
    sportivi_raport('AQUA TEAM','50 LIBER','<26 UNION SELECT nume ,prenume,
                     cod_utilizator,4, username FROM utilizator --',result);
    --sportivi_raport('AQUA TEAM','50 LIBER','IS NULL UNION SELECT nume ,prenume,
                     cod_utilizator,4, username FROM utilizator --',result);
    --sportivi_raport_securizat('AQUA TEAM','50 LIBER','< 26 UNION SELECT nume
```

```

,prenume, cod_utilizator,4, username FROM utilizator --',result); -- arata doar ceea ce
trebuia sa se vada

LOOP

  FETCH result INTO nume, prenume, numar_legitimatie,timp_total, nume_comp;

  EXIT WHEN result%NOTFOUND;

  DBMS_OUTPUT.PUT_LINE(nume || ' ' || prenume || ' - ' || numar_legitimatie || ':' ||
    timp_total || ' la ' || nume_comp);

END LOOP;

CLOSE result;

END;
/

```

Rulare prima sportivi\_raport – Atacatorul obține informațiile dorite doar pentru utilizatorii care nu îndeplinesc condiția

---

```

BRAN IONEL - 1: 4s la FRNMP_ANTRENOR1
DRAGU STELIAN - 1: 24s la CUPA DE IARNA 2024
DRAGU STELIAN - 5: 4s la FRNMP_APP_SPORTIV1
GRIGORAS MARIN - 4: 4s la FRNMP_ANTRENOR4
MARCU MARIA - 7: 4s la FRNMP_APP_SPORTIV3
MICU DENISA - 6: 4s la FRNMP_APP_SPORTIV2
MISTI SAR - 9: 4s la FRNMP_APP_SPORTIV5
STAN DENISA - 2: 4s la FRNMP_ANTRENOR2
STAN FILIP - 8: 4s la FRNMP_APP_SPORTIV4
TANCI MARIA - 3: 4s la FRNMP_ANTRENOR3

```

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

Rulare a doua sportivi\_raport – Atacatorul obține informațiile dorite pentru toți utilizatorii

---

```

BRAN IONEL - 1: 4s la FRNMP_ANTRENOR1
DRAGU STELIAN - 5: 4s la FRNMP_APP_SPORTIV1
GRIGORAS MARIN - 4: 4s la FRNMP_ANTRENOR4
MARCU MARIA - 7: 4s la FRNMP_APP_SPORTIV3
MICU DENISA - 6: 4s la FRNMP_APP_SPORTIV2
MISTI SAR - 9: 4s la FRNMP_APP_SPORTIV5
STAN DENISA - 2: 4s la FRNMP_ANTRENOR2
STAN FILIP - 8: 4s la FRNMP_APP_SPORTIV4
TANCI MARIA - 3: 4s la FRNMP_ANTRENOR3

```

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

Rulare sportivi\_raport\_securizat – nu mai este influențat de injectare și afișează doar ceea ce trebuie

```
DRAGU STELIAN - 1: 24s la CUPA DE IARNA 2024
PL/SQL procedure successfully completed.
```

### 3. Blind SQL Injection

Atacatorul vrea să afle structura bazei de date.

```
DECLARE
    result SYS_REFCURSOR;
    nume VARCHAR2(20);
    prenume VARCHAR2(20);
    numar_legitimatie NUMBER(5);
    timp_total NUMBER(5);
    nume_comp VARCHAR2(30);
BEGIN
    sportivi_raport('AQUA TEAM','50 LIBER',' IS NOT NULL AND SUBSTR((SELECT
        table_name FROM all_tables WHERE ROWNUM = 1), 1, 1) = "B" --',result); --
    prima litera a primului tabel nu este B
    --sportivi_raport('AQUA TEAM','50 LIBER',' IS NOT NULL AND SUBSTR((SELECT
        table_name FROM all_tables WHERE ROWNUM = 1), 1, 1) = "D" --',result); --
    prima litera a primului tabel este D
    --sportivi_raport('AQUA TEAM','50 LIBER',' IS NOT NULL AND SUBSTR((SELECT
        table_name FROM all_tables WHERE ROWNUM = 1), 2, 1) = "U" --',result);

    --sportivi_raport_securizat('AQUA TEAM','50 LIBER',' IS NOT NULL AND
        SUBSTR((SELECT table_name FROM all_tables WHERE ROWNUM = 1), 1, 1) =
        "D" --',result); -- prima litera a primului tabel este D
    --sportivi_raport_securizat('AQUA TEAM','50 LIBER',' IS NOT NULL AND
        SUBSTR((SELECT table_name FROM all_tables WHERE ROWNUM = 1), 1, 1) =
```

```

"B" --',result); -- prima litera a primului tabel nu este B

LOOP
    FETCH result INTO nume, prenume, numar_legitimatie,timp_total, nume_comp;
    EXIT WHEN result%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(nume || ' ' || prenume || '-' || numar_legitimatie || ':' || 
        timp_total || ' la ' || nume_comp);
END LOOP;
CLOSE result;
END;
/

```

Pentru prima rulare sportivi\_raport nu afișează nimic, ceea ce înseamnă că prima literă a primului tabel nu este B

The screenshot shows the Oracle SQL Developer interface. The 'Script Output' tab is active, displaying the message "PL/SQL procedure successfully completed." Below the tabs, there are icons for script, edit, save, and run. A status bar at the bottom indicates "Task completed in 0.109 seconds".

A doua și a treia rulare sportivi\_raport afișează, ceea ce înseamnă că prima literă este D și că a doua este U

The screenshot shows the Oracle SQL Developer interface. The 'Script Output' tab is active, displaying a list of names and titles: MISTI SAR - 5: 32 la CUPA DE TOAMNA 2024, DRAGU STELIAN - 1: 24 la CUPA DE IARNA 2024, DRAGU STELIAN - 1: 27 la CUPA DE ANUL NOU 2025, DRAGU STELIAN - 1: 26 la CUPA DE TOAMNA 2024, MICU DENISA - 2: 27 la CUPA DE TOAMNA 2024, and MICU DENISA - 2: 29 la CUPA DE IARNA 2024. Below the list, it says "PL/SQL procedure successfully completed.". The interface includes standard SQLDeveloper toolbars and a task completion status bar.

Prima și a doua rulare a sportivi\_raport\_securizat nu mai permite SQL Injection

```

Error report -
ORA-20015: Error
ORA-06512: at "FRNMP_APP_ADMIN.SPORTIVI_RAPORT_SECURIZAT", line 66
ORA-06512: at line 13

```

Rezolvarea corectă a procedurii pentru a nu mai fi vulnerabilă la SQL Injection

```
CREATE OR REPLACE PROCEDURE sportivi_raport_securizat (
    club_input IN VARCHAR2,
    proba_input IN VARCHAR2,
    filtru_timp_total IN VARCHAR2,
    result OUT SYS_REFCURSOR
) AUTHID CURRENT_USER AS
    sanitized_club_input VARCHAR2(100);
    sanitized_proba_input VARCHAR2(100);
    sanitized_timp_total VARCHAR2(100);

    valid_operator BOOLEAN := FALSE;
    valid_filtru_timp_total VARCHAR2(100);
    used_operator VARCHAR2(4);

BEGIN
    -- Reducerea intrarilor arbitrar
    --validarea parametrului club_input
    IF LENGTH(club_input) > 50 THEN
        RAISE_APPLICATION_ERROR(-20010,'Error');
    END IF;

    --DBMS_ASSERT pentru validare club_input
    sanitized_club_input := DBMS_ASSERT.enquote_literal(club_input);
    IF sanitized_club_input IS NULL THEN
        RAISE_APPLICATION_ERROR(-20011,'Error');
    END IF;
    sanitized_club_input := REPLACE(sanitized_club_input,chr(39),''');
```

```
-- Reducerea intrarilor arbitrare
--validarea parametrului proba_input
IF LENGTH(proba_input) > 30 THEN
    RAISE_APPLICATION_ERROR(-20012,'Error');
END IF;

--DBMS_ASSERT pentru validare club_input
sanitized_proba_input := DBMS_ASSERT.enquote_literal(proba_input);
IF sanitized_proba_input IS NULL THEN
    RAISE_APPLICATION_ERROR(-20013,'Error');
END IF;
sanitized_proba_input := REPLACE(sanitized_proba_input,chr(39),'');

--validare dimensiune filtru
IF filtru_timp_total IS NULL OR LENGTH(filtru_timp_total) < 3 THEN
    RAISE_APPLICATION_ERROR(-20014,'Error');
END IF;

--validare operator de comparatie
IF filtru_timp_total LIKE '<%' THEN
    valid_operator := TRUE;
    used_operator := '<';
ELSIF filtru_timp_total LIKE '>%' THEN
    valid_operator := TRUE;
    used_operator := '>';
ELSIF filtru_timp_total LIKE '=%' THEN
```

```
valid_operator := TRUE;
used_operator := '=';

ELSIF filtru_timp_total LIKE '<=%' THEN
    valid_operator := TRUE;
    used_operator := '<=';

ELSIF filtru_timp_total LIKE '>=%' THEN
    valid_operator := TRUE;
    used_operator := '>=';

END IF;

IF NOT valid_operator THEN
    RAISE_APPLICATION_ERROR(-20015,'Error');
END IF;

sanitized_timp_total := REGEXP_SUBSTR(filtru_timp_total,'\d+(.\d+)?');

IF sanitized_timp_total IS NULL THEN
    RAISE_APPLICATION_ERROR(-20016,'Error');
END IF;

valid_filtru_timp_total := used_operator || ' :timp_total';

OPEN result FOR
'SELECT u.nume, u.prenume, s.numar_legitimatie, i.timp_total,co.nume
FROM
UTILIZATOR u
JOIN CLUB c ON u.cod_club = c.cod_club
JOIN SPORTIV s ON u.cod_utilizator = s.cod_sportiv'
```

```
JOIN ISTORIC i ON s.cod_sportiv = i.cod_sportiv
JOIN COMPETITIE co ON i.cod_competitie = co.cod_competitie
WHERE c.cod_club IN
(SELECT cod_club FROM club
WHERE nume =: club
)
AND i.cod_proba IN
(SELECT cod_proba FROM proba
WHERE nume =: proba
)
AND timp_total '|| valid_filtru_timp_total
USING sanitized_club_input,sanitized_proba_input,sanitized_timp_total
;
END;
```

## 7. Mascarea datelor

Se dorește mascarea datelor care țin de date precum GDPR: nume, prenume, data naștere, gen, dar și a legitimațiilor sportivilor și a clubului din care fac parte utilizatorii

Astfel s-a mascat

CLUB: cod\_club, descriere

UTILIZATOR: nume, prenume, cod\_club

SPORTIV: numar\_legitimatie, data de nastere și genul

Pentru cod\_club se generează un număr nou care să înceapă cu aceeași cifră cu cel original și să aibă aceeași lungime.

Pentru descriere, nume, prenume se vor păstra prima și ultima literă, restul fiind înlocuit de caracterul '#'. În cazul genului acesta va fi înlocuit de '#'.

Pentru data de naștere se va păstra doar anul inițial, ziua și luna fiind generate random.

Pentru numărul de legitimărie se va păstra prima cifră, restul fiind înlocuite cu 0 și să aibă aceeași lungime.

```
create or replace package pachet_mascare is
function f_mascare(sir varchar2) return varchar2;
function f_mascare(nr number) return number;
function f_mascare_legitim(nr number) return number;
function f_mascare_data(data date) return date;
end;
/
create or replace package body pachet_mascare is
type tip_tabind is table of number index by pls_integer;
v_tabind tip_tabind;
function f_mascare(sir varchar2) return varchar2 is
    v_sir varchar2(100);
    v_lung number;
begin
    v_lung := length(sir);
    if v_lung = 1 then
        return '#';
    elsif v_lung = 2 then
        return substr(sir, 1, 1) || '#';
    else
        v_sir := substr(sir, 1, 1) || rpad('#', v_lung - 2, '#') || substr(sir, -1, 1);
        return v_sir;
    end if;
end f_mascare;

function f_mascare(nr number) return number is
    lung number;
```

```

minnou number;
maxnou number;
l_seed VARCHAR2(100);
v_nrnou number;

begin
  if v_tabind.exists(nr) then
    return v_tabind(nr); --va fi folosit cand mascam cheia externa
  else --generam un numar random care sa inceapa cu aceeasi cifra ca nr si sa aiba aceeasi
    lungime
    lung:=length(to_char(nr));
    dbms_output.put_line('lung='||lung);
    minnou:=to_number(rpad(substr(to_char(nr),1,1),lung,'0'));
    maxnou:=to_number(rpad(substr(to_char(nr),1,1),lung,'9'));
    dbms_output.put_line('minnou='||minnou||' maxnou=' ||maxnou);
    l_seed:=TO_CHAR(SYSTIMESTAMP,'YYYYDDMMHH24MISSFFF');
    DBMS_RANDOM.seed (val => l_seed);
    v_nrnou:=round(DBMS_RANDOM.value(low => minnou, high => maxnou),0);
    v_tabind(nr):=v_nrnou;
    return v_nrnou;
  end if;
end f_mascare;

function f_mascare_legitim(nr number) return number is
  v_nrnou number;
  lung number;
begin
  lung:=length(to_char(nr));
  v_nrnou:=to_number(rpad(substr(to_char(nr),1,1),lung,'0'));

```

```
    return v_nrnou;
end f_mascare_legitim;

function f_mascare_data(data date) return date is
    random_day  NUMBER;
    random_month NUMBER;
    masked_date DATE;
begin
    random_day := FLOOR(DBMS_RANDOM.VALUE(1, 29));
    random_month := FLOOR(DBMS_RANDOM.VALUE(1, 13));

    masked_date := TO_DATE(
        LPAD(random_day, 2, '0') || '-' ||
        LPAD(random_month, 2, '0') || '-' ||
        TO_CHAR(data, 'YYYY'),
        'DD-MM-YYYY'
    );
    RETURN masked_date;
end f_mascare_data;

end;
/
```

Se creează un folder în care se va face exportul, respectiv importul de date măcate.

Name	Date modified	Type	Size
export.log	1/22/2025 1:18 AM	Text Document	3 KB
FISEXPORT.DMP	1/22/2025 1:18 AM	DMP File	316 KB
import.log	1/22/2025 1:19 AM	Text Document	2 KB

Pentru a face importul tabelelor li se va schimba numele și se va folosi exclude=CONSTRAINT pentru a exclude constrângerile de pe tabelele de bază (dacă nu pun acest element, o să îmi dea eroare deoarece o să se vrea să se creeze constrângerile cu același nume cu cele deja existente). Astfel, după import le rescriu constrângerile.

--din linie de comanda pas

```
expdp frnmp_app_admin/Admin2025!@localhost:1521/orcl tables=club,utilizator,sportiv
    remap_data=club.descriere:pachet_mascare.f_mascare
    remap_data=club.cod_club:pachet_mascare.f_mascare
    remap_data=utilizator.nume:pachet_mascare.f_mascare
    remap_data=utilizator.prenume:pachet_mascare.f_mascare
    remap_data=utilizator.cod_club:pachet_mascare.f_mascare
    remap_data=sportiv.numar_legitimatie:pachet_mascare.f_mascare_legitim
    remap_data=sportiv.data_nastere:pachet_mascare.f_mascare_data
    remap_data=sportiv.gen:pachet_mascare.f_mascare directory=DIREXP
    dumpfile=FISEXPORT.dmp

--impdp frnmp_app_admin/Admin2025!@localhost:1521/orcl directory=DIREXP
    dumpfile=FISEXPORT.DMP TABLES=club,utilizator,sportiv
    remap_table=club:club1 remap_table=utilizator:utilizator1
    remap_table=sportiv:sportiv1 exclude=CONSTRAINT

impdp frnmp_app_admin/Admin2025!@localhost:1521/orcl directory=DIREXP
    dumpfile=FISEXPORT.DMP tables=club remap_table=club:club1
    exclude=CONSTRAINT

ALTER TABLE CLUB1 ADD CONSTRAINT COD_CLUB1_PK PRIMARY KEY
    (COD_CLUB);

impdp frnmp_app_admin/Admin2025!@localhost:1521/orcl directory=DIREXP
    dumpfile=FISEXPORT.DMP tables=utilizator remap_table=utilizator:utilizator1
    exclude=CONSTRAINT

ALTER TABLE UTILIZATOR1 ADD CONSTRAINT COD_UTILIZATOR1_PK
    PRIMARY KEY (COD_UTILIZATOR);
```

```
ALTER TABLE UTILIZATOR1 ADD CONSTRAINT COD_CLUB1_FK FOREIGN KEY  
(COD_CLUB) REFERENCES CLUB1 (COD_CLUB);
```

```
impdp frnmp_app_admin/Admin2025!@localhost:1521/orcl directory=DIREXP  
dumpfile=FISEXPORT.DMP tables=sportiv remap_table=sportiv:sportiv1  
exclude=CONSTRAINT
```

```
ALTER TABLE SPORTIV1 ADD CONSTRAINT COD_SPORTIV1_PK PRIMARY KEY  
(COD_SPORTIV);
```

```
select * from club1;
```

```
select * from club;
```

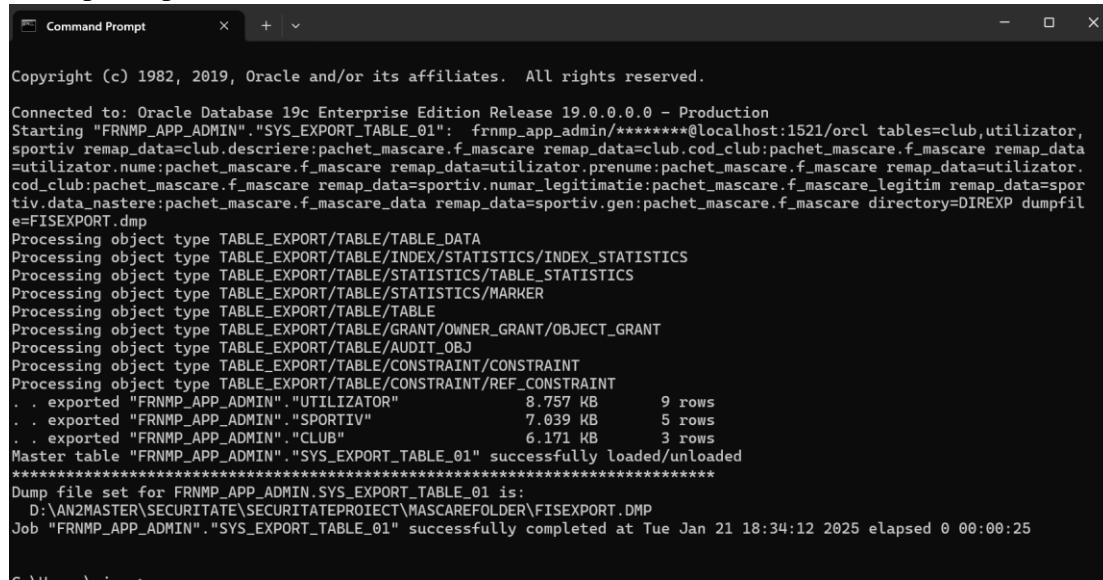
```
select * from utilizator1;
```

```
select * from utilizator;
```

```
select * from sportiv1;
```

```
select * from sportiv;
```

### Exemplu export.



```
Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.  
  
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production  
Starting "FRNMP_APP_ADMIN"."SYS_EXPORT_TABLE_01": frnmp_app_admin/********@localhost:1521/orcl tables=club,utilizator,  
sportiv remap_data=club.descriere:pachet_mascare.f_mascare remap_data=club.cod_club:pachet_mascare.f_mascare remap_data=  
utilizator.nume:pachet_mascare.f_mascare remap_data=utilizator.prenume:pachet_mascare.f_mascare remap_data=utilizator.  
cod_club:pachet_mascare.f_mascare remap_data=sportiv.numar_legitimatie:pachet_mascare.f_mascare_legitim remap_data=spor  
tiv.data_nastere:pachet_mascare.f_mascare_data remap_data=sportiv.gen:pachet_mascare.f_mascare directory=DIREXP dumpfil  
e=FISEXPORT.dmp  
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA  
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS  
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS  
Processing object type TABLE_EXPORT/TABLE/STATISTICS/MARKER  
Processing object type TABLE_EXPORT/TABLE/TABLE  
Processing object type TABLE_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT  
Processing object type TABLE_EXPORT/TABLE/AUDIT_OBJ  
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT  
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT  
. . . exported "FRNMP_APP_ADMIN"."UTILIZATOR" 8.757 KB 9 rows  
. . . exported "FRNMP_APP_ADMIN"."SPORTIV" 7.039 KB 5 rows  
. . . exported "FRNMP_APP_ADMIN"."CLUB" 6.171 KB 3 rows  
Master table "FRNMP_APP_ADMIN"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded  
*****  
Dump file set for FRNMP_APP_ADMIN.SYS_EXPORT_TABLE_01 is:  
D:\AN2MASTER\SECURITATE\SECURITATEPROIECT\MASCAREFOLDER\FISEXPORT.DMP  
Job "FRNMP_APP_ADMIN"."SYS_EXPORT_TABLE_01" successfully completed at Tue Jan 21 18:34:12 2025 elapsed 0 00:00:25
```

```
select * from club1;
```

COD_CLUB	NUME	DESCRIERE
1	AQUA TEAM	F*****t
2	DELFINUL BUCURESTI	A*****e
3	BUCHAREST SPORT CLUB ELITE	E*****a

select \* from utilizator1;

COD_UTILIZATOR	CNP	NUME	PRENUME	USERNAME	TIP	COD_ANTRENOR	COD_CLUB
1	13A97F0CA9AA926493313139350B66B6	B##N	I###L	FRNMP_ANTRENOR1	ANTRENOR	(null)	1
2	3743045C9C4926288E8CCFFB3988E58B	S##N	D####A	FRNMP_ANTRENOR2	ANTRENOR	1	1
3	46CF93FAFD2F59F27808E70B770C623B	T###I	M###A	FRNMP_ANTRENOR3	ANTRENOR	1	2
4	76DC48FE274B8419B5D8AB99ACDF7F4C	G#####S	M###N	FRNMP_ANTRENOR4	ANTRENOR	2	3
5	226C778E623F76C733821942DCD845E4	D###U	S#####N	FRNMP_APP_SPORTIV1	SPORTIV	1	1
6	E3774303C02D366924BA2AEC7DFCEF2A	M##U	D####A	FRNMP_APP_SPORTIV2	SPORTIV	2	1
7	DA4C3632AB96EF23822963F73FD1EF08	M##U	M###A	FRNMP_APP_SPORTIV3	SPORTIV	3	2
8	93107A27C1617257A1B1F35023675677	S##N	F###P	FRNMP_APP_SPORTIV4	SPORTIV	4	3
9	232BC8B04AC595BE96C8EE3709E4CC20	M##I	S#R	FRNMP_APP_SPORTIV5	SPORTIV	1	1

select \* from sportiv1;

COD_SPORTIV	NUMAR_LEGITIMATIE	DATA_NASTERE	GEN	COD_CATEGORIE
5	1	28-JUN-05	#	2
6	2	15-FEB-05	#	1
7	3	18-AUG-00	#	1
8	4	05-JUL-11	#	8
9	5	20-DEC-11	#	8

select \* from sportiv;

COD_SPORTIV	NUMAR_LEGITIMATIE	DATA_NASTERE	GEN	COD_CATEGORIE
5	1	21-AUG-05	M	2
6	2	26-APR-05	F	1
7	3	30-DEC-00	F	1
8	4	25-MAR-11	M	8
9	5	15-OCT-11	M	8

## I. Inserarea de date

1.Date inserate de administrator care nu implica criptarea

-- club

INSERT INTO CLUB

VALUES(1, 'AQUA TEAM','Facem valuri si cream campioni in fiecare moment');

INSERT INTO CLUB

VALUES(2, 'DELFINUL BUCURESTI','Aici delfinii inoata cu pasiune si putere');

INSERT INTO CLUB

VALUES(3, 'BUCHAREST SPORT CLUB ELITE','Excelenta cu fiecare lovitura, succes in fiecare cursa');

select \* from club;

commit;

-- tara

INSERT INTO TARA(cod\_tara,nume\_tara)

VALUES('RO','ROMANIA');

INSERT INTO TARA(cod\_tara,nume\_tara)

VALUES('PO','POLONIA');

INSERT INTO TARA(cod\_tara,nume\_tara)

VALUES('FR','FRANTA');

INSERT INTO TARA(cod\_tara,nume\_tara)

VALUES('JP','JAPONIA');

```
INSERT INTO TARA(cod_tara,nume_tara)
VALUES('IT','ITALIA');
```

```
INSERT INTO TARA(cod_tara,nume_tara)
VALUES('KO','KOREA');
```

```
commit;
```

```
-- oras
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (1,'BUCURESTI','RO');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (2,'CRAIOVA','RO');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (3,'LUBLIN','PO');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (4,'PARIS','FR');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (5,'KYOTO','JP');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (6,'NAPOLI','IT');
```

```
INSERT INTO ORAS(cod_oras,nume_oras,cod_tara)
VALUES (7,'GWANGJU','KO');
```

```
COMMIT;
```

```
--tip
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(1,'LOCAL');
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(2,'MUNICIPAL');
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(3,'JUDETEAN');
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(4,'EUROPEAN');
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(5,'BALCANIC');
```

```
INSERT INTO TIP(cod_tip, tip_competitie)
VALUES(6,'MONDIAL');
```

```
COMMIT;
```

```
--proba
```

```
INSERT INTO PROBA(cod_proba,nume)
```

```
VALUES(1,'50 LIBER');
```

```
INSERT INTO PROBA(cod_proba,nume)  
VALUES(2,'50 SPATE');
```

```
INSERT INTO PROBA(cod_proba,nume)  
VALUES(3,'100 LIBER');
```

```
INSERT INTO PROBA(cod_proba,nume)  
VALUES(4,'200 BRAS');
```

```
INSERT INTO PROBA(cod_proba,nume)  
VALUES(5,'200 FLUTURE');
```

```
INSERT INTO PROBA(cod_proba,nume)  
VALUES(6,'400 LIBER');
```

```
COMMIT;
```

```
--competitie
```

```
INSERT INTO COMPETITIE(cod_competitie,nume,data_start,data_final,cod_oras,cod_tip)  
VALUES(1,'CUPA DE TOAMNA 2024',TO_DATE('05-11-2024','DD-MM-  
YYYY'),TO_DATE('06-11-2024','DD-MM-YYYY'),1,6); --concurs mondial bucuresti
```

```
INSERT INTO COMPETITIE(cod_competitie,nume,data_start,data_final,cod_oras,cod_tip)  
VALUES(2,'CUPA DE IARNA 2024',TO_DATE('10-12-2024','DD-MM-  
YYYY'),TO_DATE('11-12-2024','DD-MM-YYYY'),1,1); --concurs LOCAL bucuresti
```

```
INSERT INTO COMPETITIE(cod_competitie,nume,data_start,data_final,cod_oras,cod_tip)  
VALUES(3,'CUPA DE ANUL NOU 2025',TO_DATE('03-01-2025','DD-MM-
```

```
YYYY'),TO_DATE('04-01-2025','DD-MM-YYYY'),4,4);  
  
commit;  
  
SELECT * FROM COMPETITIE;  
  
--il pun aici ca sa vad ca merge triggerul.  
  
SELECT * FROM TIP;  
  
SELECT * FROM ORAS;  
  
INSERT INTO FRNMP_APP_ADMIN.ISTORIC(cod_sportiv, cod_proba, cod_competitie,  
loc_per_gen,loc_per_categorie, timp_total, puncte)  
VALUES(5,1,1,15,3,26,180);  
  
commit;  
  
--pentru categorie  
  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (1,'SENIOR F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (2,'SENIOR M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (3,'TINERET F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (4,'TINERET M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (5,'JUNIOR 1 F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (6,'JUNIOR 1 M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (7,'JUNIOR 2 F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (8,'JUNIOR 2 M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (9,'CADET F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (10,'CADET M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (11,'COPIL 11 F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (12,'COPIL 11 M');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (13,'COPIL 10 F');  
INSERT INTO CATEGORIE(cod_categorie,nume) VALUES (14,'COPIL 10 M');  
  
SELECT * FROM CATEGORIE;
```

```
commit;
```

2. Pentru inserarea de către administrator a datelor în tabelele cu date criptate (SPONSOR, UTILIZATOR) și a tabelelor SPORTIV și ANTRENOR

```
INSERT INTO SPONSOR(cod_sponsor,nume,suma_contributie)
VALUES (1,'RED BULL',CRYPTARE_SUMA ADM(500));
```

```
INSERT INTO SPONSOR(cod_sponsor,nume,suma_contributie)
VALUES (2,'ISOSTAR',CRYPTARE_SUMA ADM(600.4));
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
(1,CRIPTARE_UTILIZ ADM('1850223145522'),'BRAN','IONEL','FRNMP_ANTREN
OR1','ANTRENOR',NULL,1);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
(2,CRIPTARE_UTILIZ ADM('2960225640002'),'STAN','DENISA','FRNMP_ANTREN
OR2','ANTRENOR',1,1);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
(3,CRIPTARE_UTILIZ ADM('2951230111012'),'TANCI','MARIA','FRNMP_ANTREN
OR3','ANTRENOR',1,2);
```

```
INSERT INTO UTILIZATOR
```

```
VALUES
(4,CRIPTARE_UTILIZ ADM('1630823578936'),'GRIGORAS','MARIN','FRNMP_AN
TRENOR4','ANTRENOR',2,3);
```

```
SELECT * FROM UTILIZATOR;
```

```
SELECT DECRYPTARE_UTILIZ_ADMIN('FRNMP_ANTRENOR1') FROM DUAL;  
INSERT INTO ANTRENOR VALUES(1,20);  
INSERT INTO ANTRENOR VALUES(2,9);  
INSERT INTO ANTRENOR VALUES(3,5);  
INSERT INTO ANTRENOR VALUES(4,12);  
  
SELECT * FROM ANTRENOR;  
select * from UTILIZATOR ;  
INSERT INTO UTILIZATOR  
VALUES  
(5,CRIPTARE_UTILIZ_ADMIN('7050821555520'),'DRAGU','STELIAN','FRNMP_APP_  
SPORTIV1', 'SPORTIV',1,1);  
  
INSERT INTO UTILIZATOR  
VALUES  
(6,CRIPTARE_UTILIZ_ADMIN('6050426785520'),'MICU','DENISA','FRNMP_APP_SPO  
RTIV2', 'SPORTIV',2,1);  
  
INSERT INTO UTILIZATOR  
VALUES  
(7,CRIPTARE_UTILIZ_ADMIN('6001230121030'),'MARCU','MARIA','FRNMP_APP_SP  
ORTIV3', 'SPORTIV',3,2);  
  
INSERT INTO UTILIZATOR  
VALUES  
(8,CRIPTARE_UTILIZ_ADMIN('7110325871101'),'STAN','FILIP','FRNMP_APP_SPORT  
IV4', 'SPORTIV',4,3);  
  
INSERT INTO UTILIZATOR  
VALUES
```

```
(9,CRYPTARE_UTILIZ ADM('7111015775232'),'MISTI','SAR','FRNMP_APP_SPORTIV5', 'SPORTIV',1,1);
```

```
select * from categorie;  
INSERT INTO SPORTIV  
VALUES (5,1,TO_DATE('21-08-2005','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV  
SET cod_categorie = 2  
WHERE cod_sportiv = 5;
```

```
insert INTO SPORTIV  
VALUES (6,2,TO_DATE('26-04-2005','DD-MM-YYYY'),'F');
```

```
UPDATE SPORTIV  
SET cod_categorie = 1  
WHERE cod_sportiv = 6;
```

```
INSERT INTO SPORTIV  
VALUES (7,3,TO_DATE('30-12-2000','DD-MM-YYYY'),'F');
```

```
UPDATE SPORTIV  
SET cod_categorie = 1  
WHERE cod_sportiv = 7;
```

```
INSERT INTO SPORTIV  
VALUES (8,4,TO_DATE('25-03-2011','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 8
```

```
WHERE cod_sportiv = 8;
```

```
INSERT INTO SPORTIV
```

```
VALUES (9,5,TO_DATE('15-10-2011','DD-MM-YYYY'),'M');
```

```
UPDATE SPORTIV
```

```
SET cod_categorie = 8
```

```
WHERE cod_sportiv = 9;
```

```
commit;
```

```
SELECT * FROM SPORTIV;
```

3. Crearea procedurii antrenor\_insert\_istoric de administrator prin care antrenorii să insereze date în ISTORIC.

```
--pentru istoric
```

```
CREATE OR REPLACE PROCEDURE antrenor_insert_istoric (
```

```
    p_legitimatie_sportiv IN NUMBER,
```

```
    p_nume_proba IN VARCHAR2,
```

```
    p_nume_competitie IN VARCHAR2,
```

```
    p_loc_per_gen IN NUMBER,
```

```
    p_loc_per_categorie IN NUMBER,
```

```
    p_timp_total IN NUMBER,
```

```
    p_puncte IN NUMBER
```

```
) IS
```

```
    v_cod_competitie NUMBER(5);
```

```
v_cod_proba NUMBER(5);
v_cod_sportiv NUMBER(5);

BEGIN

    SELECT cod_sportiv
    INTO v_cod_sportiv
    FROM frnmp_app_admin.sportiv
    WHERE numar_legitimatie = p_legitimatie_sportiv;

    SELECT cod_competitie
    INTO v_cod_competitie
    FROM frnmp_app_admin.COMPETITIE
    WHERE nume = p_nume_competitie;

    SELECT cod_proba
    INTO v_cod_proba
    FROM frnmp_app_admin.PROBA
    WHERE nume = p_nume_proba;

    INSERT INTO frnmp_app_admin.ISTORIC (
        cod_sportiv, cod_proba, cod_competitie, loc_per_gen,
        loc_per_categorie, timp_total, puncte
    ) VALUES (
        v_cod_sportiv, v_cod_proba, v_cod_competitie, p_loc_per_gen,
        p_loc_per_categorie, p_timp_total, p_puncte
    );

    COMMIT;
```

```
END;  
/
```

4. Inserări în istoric de către antrenori:

Logat ca frnmp\_antrenor1

```
select * from frnmp_app_admin.view_sportivi_antrenor;  
  
select * from frnmp_app_admin.view_proba;  
  
select * from frnmp_app_admin.view_competitie;  
  
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 LIBER', 'CUPA DE TOAMNA 2024', 0,  
      0, 0, 0);  
  
select * from frnmp_app_admin.view_istoric_sportivi_antrenor;  
  
  
rollback;  
  
SELECT sys_context('APLICATIE_CTX', 'PROGRAM_POTRIVIT') FROM dual;  
  
set SERVEROUTPUT on;  
  
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE TOAMNA 2024', 0,  
      0, 0, 0);  
  
--AM RULAT VINERI LA ORA 18:50 SI IMI ZICE CA SUNT IN AFARA PROGRAMULUI,  
      CEEA CE E BINE  
  
  
--Error starting at line : 7 in command -  
  
--BEGIN frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE TOAMNA 2024',  
      0, 0, 0, 0); END;  
  
--Error report -  
  
--ORA-20001: Sunteti in afara programului  
  
--ORA-06512: at "FRNMP_APP_ADMIN.INSCHRIERE_SP_TRIGGER", line 3  
  
--ORA-04088: error during execution of trigger  
      'FRNMP_APP_ADMIN.INSCHRIERE_SP_TRIGGER'  
  
--ORA-06512: at "FRNMP_APP_ADMIN.ANTRENOR_INSERT_ISTORIC", line 30  
  
--ORA-06512: at line 1
```

--alte inserari

```
EXEC frnmp_app_admin.antrenor_insert_istoric(5, '50 LIBER', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(5, '50 SPATE', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '200 BRAS', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 LIBER', 'CUPA DE IARNA 2024', 0, 0,  
0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 LIBER', 'CUPA DE ANUL NOU 2025',  
0, 0, 0, 0);
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '100 LIBER', 'CUPA DE ANUL NOU 2025',  
0, 0, 0, 0);
```

--testare in afara programului

```
EXEC frnmp_app_admin.antrenor_insert_istoric(1, '50 SPATE', 'CUPA DE ANUL NOU 2025',  
0, 0, 0, 0);
```

Logat ca frnmp\_antrenor2

```
SELECT sys_context('APLICATIE_CTX', 'PROGRAM_POTRIVIT') FROM dual;
```

```
set SERVEROUTPUT on;
```

```
select * from frnmp_app_admin.view_sportivi_antrenor;
```

```
select * from frnmp_app_admin.view_proba;
```

```
select * from frnmp_app_admin.view_competitie;
```

```
select * from frnmp_app_admin.view_istoric_sportivi_antrenor;
```

```
EXEC frnmp_app_admin.antrenor_insert_istoric(2, '50 LIBER', 'CUPA DE TOAMNA 2024', 0,
```

```
0, 0, 0);  
EXEC frnmp_app_admin.antrenor_insert_istoric(2, '100 LIBER', 'CUPA DE TOAMNA 2024',  
0, 0, 0, 0);  
  
EXEC frnmp_app_admin.antrenor_insert_istoric(2, '50 LIBER', 'CUPA DE IARNA 2024', 0, 0,  
0, 0);  
commit;
```

Logat ca frnmp\_antrenor3

```
select * from frnmp_app_admin.view_sportivi_antrenor;  
select * from frnmp_app_admin.view_proba;  
select * from frnmp_app_admin.view_competitie;  
select * from frnmp_app_admin.view_istoric_sportivi_antrenor;  
  
EXEC frnmp_app_admin.antrenor_insert_istoric(3, '50 LIBER', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);  
EXEC frnmp_app_admin.antrenor_insert_istoric(3, '100 LIBER', 'CUPA DE TOAMNA 2024',  
0, 0, 0, 0);  
EXEC frnmp_app_admin.antrenor_insert_istoric(3, '50 LIBER', 'CUPA DE IARNA 2024', 0, 0,  
0, 0);  
commit;
```

Logat ca frnmp\_antrenor4

```
select * from frnmp_app_admin.view_sportivi_antrenor;  
select * from frnmp_app_admin.view_proba;  
select * from frnmp_app_admin.view_competitie;  
select * from frnmp_app_admin.view_istoric_sportivi_antrenor;  
  
EXEC frnmp_app_admin.antrenor_insert_istoric(4, '50 LIBER', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);  
EXEC frnmp_app_admin.antrenor_insert_istoric(4, '50 SPATE', 'CUPA DE TOAMNA 2024', 0,  
0, 0, 0);
```

commit;

5.Antrenorii sunt cei care fac inserările în tabelul ISTORIC, iar arbitrii sunt cei care fac actualizări. Logat ca frnmp\_arbitru1

```
select * from frnmp_app_admin.view_istoric_arbitru;
```

```
set SERVEROUTPUT on;
```

```
UPDATE frnmp_app_admin.istoric
```

```
SET loc_per_gen = 15,loc_per_categorie=3,temp_total=26,puncte=170
```

```
WHERE cod_competitie = 1 and cod_sportiv = 5 and cod_proba = 1; --ESTE COMPETITIE  
MONDIALA SI MAXIMUL ESTE DE 150 DE PUNCTE,
```

```
COMMIT;
```

```
--IAR ACUM ESTE DECLANSAT TRIGGERUL AUDIT SI POLITICA DE AUDIT  
policy_istoric si auditul standard
```

```
--acum se pune numarul corect de puncte
```

```
UPDATE frnmp_app_admin.istoric
```

```
SET loc_per_gen = 15,loc_per_categorie=3,temp_total=26,puncte=130
```

```
WHERE cod_competitie = 1 and cod_sportiv = 5 and cod_proba = 1;
```

```
COMMIT;
```

```
--IAR ACUM ESTE DECLANSATA POLITICA DE AUDIT policy_istoric si auditul standard
```

```
--pentru exercitiul 6 - context aplicatie
```

```
UPDATE frnmp_app_admin.istoric --l-am facut intr-o zi de vineri deci nu trebuie sa ma lase
```

```
SET loc_per_gen = 2,loc_per_categorie=3,temp_total=32,puncte=130
```

```
WHERE cod_competitie = 1 and cod_sportiv = 5 and cod_proba = 3;
```

```
COMMIT;
```

```
--Error report -SQL Error: ORA-20002: Sunteți în afara programului de arbitrage și contestații (a  
apărut eroarea deci e bine)
```

```
--IAR CAND AM RULAT IN WEEKEN S-A INSERAT.
```

--

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=27,puncte=150  
WHERE cod_sportiv = 6 and cod_proba = 1 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 10,loc_per_categorie=4,timp_total=33,puncte=120  
WHERE cod_sportiv = 7 and cod_proba = 1 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 9,loc_per_categorie=2,timp_total=20,puncte=140  
WHERE cod_sportiv = 8 and cod_proba = 1 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 20,loc_per_categorie=5,timp_total=32,puncte=110  
WHERE cod_sportiv = 9 and cod_proba = 1 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 3,loc_per_categorie=2,timp_total=46,puncte=140  
WHERE cod_sportiv = 6 and cod_proba = 3 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric
```

```
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=44,puncte=150  
WHERE cod_sportiv = 7 and cod_proba = 3 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 3,loc_per_categorie=1,timp_total=27,puncte=150  
WHERE cod_sportiv = 5 and cod_proba = 2 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 2,loc_per_categorie=2,timp_total=26,puncte=140  
WHERE cod_sportiv = 8 and cod_proba = 2 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=25,puncte=150  
WHERE cod_sportiv = 9 and cod_proba = 2 and cod_competitie = 1;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=60,puncte=150  
WHERE cod_sportiv = 5 and cod_proba = 4 and cod_competitie = 1;  
COMMIT;
```

```
--comp cu id 2  
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 15,loc_per_categorie=2,timp_total=24,puncte=14  
WHERE cod_sportiv = 5 and cod_proba = 1 and cod_competitie = 2;
```

```
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 2,loc_per_categorie=2,timp_total=29,puncte=14  
WHERE cod_sportiv = 6 and cod_proba = 1 and cod_competitie = 2;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=28,puncte=15  
WHERE cod_sportiv = 7 and cod_proba = 1 and cod_competitie = 2;  
COMMIT;
```

```
--competitie cu id 3
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 9,loc_per_categorie=3,timp_total=27,puncte=130  
WHERE cod_sportiv = 5 and cod_proba = 1 and cod_competitie = 3;  
COMMIT;
```

```
UPDATE frnmp_app_admin.istoric  
SET loc_per_gen = 1,loc_per_categorie=1,timp_total=30,puncte=150  
WHERE cod_sportiv = 5 and cod_proba = 3 and cod_competitie = 3;  
COMMIT;
```

6.Pentru ca reprezentanții companiilor să poată inseră, update date în tabelul SPONSORIZEAZĂ care se află în schema fiecărui sportiv, adminul creează vizualizări și triggeri pe insert și update.

```
--CREATE TABLE FRNMP_APP_SPORTIV1.SPONSORIZEAZA  
--(cod_sponsor NUMBER(5),
```

```
--cod_sportiv NUMBER(5),
--inceput_colaborare DATE CONSTRAINT inc_colab_nn NOT NULL,
--sfarsit_colaborare DATE CONSTRAINT sf_colab_nn NOT NULL,
--CONSTRAINT datele CHECK (inceput_colaborare<sfarsit_colaborare),
--CONSTRAINT pk_cod_sponsor_cod_sportiv_colab PRIMARY KEY
  (cod_sponsor,cod_sportiv,inceput_colaborare),
--CONSTRAINT cod_sponsor_fk FOREIGN KEY (cod_sponsor) REFERENCES SPONSOR
  (cod_sponsor),
--CONSTRAINT cod_sportiv_fk FOREIGN KEY (cod_sportiv) REFERENCES SPORTIV
  (cod_sportiv)
--);
```

SELECT \* FROM SPORTIV; -- COD SPORTIV: 5-9

SELECT \* FROM SPONSOR;

SELECT \* FROM COMPETITIE;

SELECT \* FROM TIP;

select \* from FRNMP\_APP\_SPORTIV1.sponsorizeaza; --se vede doar dupa ce reprezentantul  
companiei da commit la insert

CREATE OR REPLACE VIEW VIZ\_SPONSORIZARE AS

```
  SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
        'SPORTIV1' AS cod_sportiv
    FROM FRNMP_APP_SPORTIV1.SPONSORIZEAZA
  UNION
  SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
        'SPORTIV2' AS cod_sportiv
    FROM FRNMP_APP_SPORTIV2.SPONSORIZEAZA
  UNION
  SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
        'SPORTIV3' AS cod_sportiv
    FROM FRNMP_APP_SPORTIV3.SPONSORIZEAZA
```

```
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
'SPORTIV4' AS cod_sportiv
FROM FRNMP_APP_SPORTIV4.SPONSORIZEAZA
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
'SPORTIV5' AS cod_sportiv
FROM FRNMP_APP_SPORTIV5.SPONSORIZEAZA;
/
--pentru inserarea datelor in tabela sponsorizeaza este nevoie de un trigger instead of
CREATE OR REPLACE TRIGGER TRIGGER_SPONSORIZARE
INSTEAD OF INSERT ON VIZ_SPONSORIZARE
FOR EACH ROW
BEGIN
IF :NEW.COD_SPORTIV='SPORTIV1' THEN
    INSERT INTO FRNMP_APP_SPORTIV1.SPONSORIZEAZA
        VALUES(:NEW.COD_SPONSOR,5,:NEW.INCEPUT_COLABORARE,:NEW.SFARSI
T_COLABORARE);
END IF;
IF :NEW.COD_SPORTIV='SPORTIV2' THEN
    INSERT INTO FRNMP_APP_SPORTIV2.SPONSORIZEAZA
        VALUES(:NEW.COD_SPONSOR,6,:NEW.INCEPUT_COLABORARE,:NEW.SFARSI
T_COLABORARE);
END IF;
IF :NEW.COD_SPORTIV='SPORTIV3' THEN
    INSERT INTO FRNMP_APP_SPORTIV3.SPONSORIZEAZA
        VALUES(:NEW.COD_SPONSOR,7,:NEW.INCEPUT_COLABORARE,:NEW.SFARSI
T_COLABORARE);
```

```
END IF;

IF :NEW.COD_SPORTIV='SPORTIV4' THEN
    INSERT INTO FRNMP_APP_SPORTIV4.SPONSORIZEAZA
        VALUES(:NEW.COD_SPONSOR,8,:NEW.INCEPUT_COLABORARE,:NEW.SFARSI
T_COLABORARE);

END IF;

IF :NEW.COD_SPORTIV='SPORTIV5' THEN
    INSERT INTO FRNMP_APP_SPORTIV5.SPONSORIZEAZA
        VALUES(:NEW.COD_SPONSOR,9,:NEW.INCEPUT_COLABORARE,:NEW.SFARSI
T_COLABORARE);

END IF;

END;
/

CREATE OR REPLACE TRIGGER TRIGGER_SPONSORIZARE_upd
INSTEAD OF update ON VIZ_SPONSORIZARE
FOR EACH ROW
BEGIN
    IF :NEW.COD_SPORTIV='SPORTIV1' THEN
        update FRNMP_APP_SPORTIV1.SPONSORIZEAZA
        set sfarsit_colaborare = :NEW.SFARSIT_COLABORARE
        where cod_sportiv = 5 and cod_sponsor = :NEW.COD_SPONSOR;
    END IF;

    IF :NEW.COD_SPORTIV='SPORTIV2' THEN
        update FRNMP_APP_SPORTIV2.SPONSORIZEAZA
```

```
set sfarsit_colaborare = :NEW.SFARSIT_COLABORARE
where cod_sportiv = 6 and cod_sponsor = :NEW.COD_SPONSOR;
END IF;

IF :NEW.COD_SPORTIV='SPORTIV3' THEN
    update FRNMP_APP_SPORTIV3.SPONSORIZEAZA
    set sfarsit_colaborare = :NEW.SFARSIT_COLABORARE
    where cod_sportiv = 7 and cod_sponsor = :NEW.COD_SPONSOR;
END IF;

IF :NEW.COD_SPORTIV='SPORTIV4' THEN
    update FRNMP_APP_SPORTIV4.SPONSORIZEAZA
    set sfarsit_colaborare = :NEW.SFARSIT_COLABORARE
    where cod_sportiv = 8 and cod_sponsor = :NEW.COD_SPONSOR;
END IF;

IF :NEW.COD_SPORTIV='SPORTIV5' THEN
    update FRNMP_APP_SPORTIV5.SPONSORIZEAZA
    set sfarsit_colaborare = :NEW.SFARSIT_COLABORARE
    where cod_sportiv = 9 and cod_sponsor = :NEW.COD_SPONSOR;
END IF;

END;
/
CREATE OR REPLACE VIEW VIZ_SPONSORIZARE_2 AS
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
cod_sportiv
FROM FRNMP_APP_SPORTIV1.SPONSORIZEAZA
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
```

```

cod_sportiv
FROM FRNMP_APP_SPORTIV2.SPONSORIZEAZA
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
cod_sportiv
FROM FRNMP_APP_SPORTIV3.SPONSORIZEAZA
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
cod_sportiv
FROM FRNMP_APP_SPORTIV4.SPONSORIZEAZA
UNION
SELECT COD_SPONSOR, INCEPUT_COLABORARE, SFARSIT_COLABORARE,
cod_sportiv
FROM FRNMP_APP_SPORTIV5.SPONSORIZEAZA;
/

```

Logat ca frnmp\_reprezentant1

```
INSERT INTO FRNMP_APP_ADMIN.VIZ_SPONSORIZARE VALUES
(1,SYSDATE,TO_DATE('2025-03-01', 'yyyy-mm-dd'),'SPORTIV1');
```

```
SELECT * FROM FRNMP_APP_ADMIN.viz_sponsorizare WHERE cod_sponsor=1;
commit;
```

COD_SPONSOR	INCEPUT_COLABORARE	SFARSIT_COLABORARE	COD_SPORTIV
1	07-JAN-25	01-MAR-25	SPORTIV1

Logat ca frnmp\_reprezentant2

```
INSERT INTO FRNMP_APP_ADMIN.VIZ_SPONSORIZARE VALUES
(2,SYSDATE-3,SYSDATE,'SPORTIV2');
```

```
SELECT * FROM FRNMP_APP_ADMIN.viz_sponsorizare WHERE cod_sponsor=2;
commit;
rollback;
```

```
update FRNMP_APP_ADMIN.viz_sponsorizare
set SFARSIT_COLABORARE = TO_DATE('23-01-2025', 'DD-MM-YYYY')
WHERE COD_SPORTIV = 'SPORTIV2'
```

AND COD\_SPONSOR = 2;

COD_SPONSOR	INCEPUT_COLABORARE	SFARSIT_COLABORARE	COD_SPORTIV
2	18-JAN-25	23-JAN-25	SPORTIV2

## II. Crearea de alte view-uri de către admin

```
SELECT * FROM COMPETITIE;
```

```
CREATE OR REPLACE VIEW view_competitie AS
```

```
SELECT nume
```

```
FROM frnmp_app_admin.competitie;
```

Rulat din contul de frnmp\_antrenor1: select \* from frnmp\_app\_admin.view\_competitie;

NUME
CUPA DE ANUL NOU 2025
CUPA DE IARNA 2024
CUPA DE TOAMNA 2024

```
CREATE OR REPLACE VIEW view_proba AS
```

```
SELECT nume
```

```
FROM frnmp_app_admin.proba;
```

Rulat din contul de frnmp\_antrenor1: select \* from frnmp\_app\_admin.view\_proba;

NUME
1 100 LIBER
2 200 BRAS
3 200 FLUTURE
4 400 LIBER
5 50 LIBER
6 50 SPATE

```
CREATE OR REPLACE VIEW view_sportivi_antrenor AS
```

```
SELECT nume,prenume,numar_legitimatie,data_nastere,gen
```

```
FROM frnmp_app_admin.sportiv sp JOIN frnmp_app_admin.utilizator ut
```

```
ON sp.cod_sportiv = ut.cod_utilizator
```

```
WHERE ut.cod_antrenor = (SELECT cod_utilizator FROM frnmp_app_admin.utilizator
WHERE username = USER);
```

Rulat din contul de frnmp\_antrenor1: select \* from frnmp\_app\_admin.view\_sportivi\_antrenor;

NUME	PRENUME	NUMAR_LEGITIMATIE	DATA_NASTERE	GEN
DRAGU	STELIAN	1	21-AUG-05	M
MISTI	SAR	5	15-OCT-11	M

```
CREATE OR REPLACE VIEW view_istoric_sportivi_antrenor AS
SELECT ut.nume as nume_sp,ut.prenume,sp.numar_legitimatie,sp.data_nastere,sp.gen,
ist.loc_per_gen,ist.loc_per_categorie,ist.timp_total,ist.puncte,pr.nume as
proba,compe.nume as competitie
FROM frnmp_app_admin.istoric ist JOIN frnmp_app_admin.sportiv sp ON ist.cod_sportiv =
sp.cod_sportiv
JOIN frnmp_app_admin.utilizator ut ON sp.cod_sportiv = ut.cod_utilizator
JOIN frnmp_app_admin.PROBA pr ON ist.cod_proba = pr.cod_proba
JOIN frnmp_app_admin.COMPETITIE compe ON ist.cod_competitie = compe.cod_competitie
WHERE ut.cod_antrenor = (SELECT cod_utilizator FROM frnmp_app_admin.utilizator
WHERE username = USER);
```

Rulat din contul de frnmp\_antrenor1: select \* from  
frnmp\_app\_admin.view\_istoric\_sportivi\_antrenor;

NUME_SP	PRENUME	NUMAR_LEGITIMATIE	DATA_NASTERE	GEN	LOC_PER_GEN	LOC_PER_CATEGORIE	TIMP_TOTAL	PUNCTE	PROBA	COMPETITIE
DRAGU	STELIAN	1	21-AUG-05	M	1	1	30	150 100 LIBER	CUPA DE ANUL NOU 2025	
DRAGU	STELIAN	1	21-AUG-05	M	2	3	32	130 100 LIBER	CUPA DE TOAMNA 2024	
DRAGU	STELIAN	1	21-AUG-05	M	1	1	60	150 200 BRAS	CUPA DE TOAMNA 2024	
DRAGU	STELIAN	1	21-AUG-05	M	9	3	27	130 50 LIBER	CUPA DE ANUL NOU 2025	
DRAGU	STELIAN	1	21-AUG-05	M	15	2	24	14 50 LIBER	CUPA DE IARNA 2024	
MISTI	SAR	5	15-OCT-11	M	20	5	32	110 50 LIBER	CUPA DE TOAMNA 2024	
DRAGU	STELIAN	1	21-AUG-05	M	15	3	26	130 50 LIBER	CUPA DE TOAMNA 2024	
DRAGU	STELIAN	1	21-AUG-05	M	3	1	27	150 50 SPATE	CUPA DE TOAMNA 2024	
MISTI	SAR	5	15-OCT-11	M	1	1	25	150 50 SPATE	CUPA DE TOAMNA 2024	

```
CREATE OR REPLACE VIEW view_istoric_arbitru AS
SELECT sp.numar_legitimatie,ut.nume as
nume_sp,ut.prenume,sp.data_nastere,sp.gen,ist.loc_per_gen,ist.loc_per_categorie,ist.timp
_total,ist.puncte,pr.nume as proba,compe.nume as
competitie,sp.cod_sportiv,pr.cod_proba,compe.cod_competitie
FROM frnmp_app_admin.istoric ist JOIN frnmp_app_admin.sportiv sp ON ist.cod_sportiv =
sp.cod_sportiv
JOIN frnmp_app_admin.utilizator ut ON sp.cod_sportiv = ut.cod_utilizator
JOIN frnmp_app_admin.PROBA pr ON ist.cod_proba = pr.cod_proba
```

```

JOIN frnmp_app_admin.COMPETITIE compe ON ist.cod_competitie = compe.cod_competitie;
select * from istoric;
select * from sportiv;
--delete from istoric;

```

Rulat din contul de frnmp\_arbitru1: select \* from frnmp\_app\_admin.view\_istoric\_arbitru;

NUMAR_LEGITIMATIE	NUME_SP	PRENUME	DATA_NASTERE	GEN	LOC_PER_GEN	LOC_PER_CATEGORIE	TIMP_TOTAL	PUNCTE	PROBA	COMPETITIE	COD_SPORTIV	COD_PROBA	COD_COMPETITIE
1 DRAGU	STELIAN	21-AUG-05	M		1	1	30	150 100	LIBER CUPA DE ANUL NOU 2025		5	3	3
1 DRAGU	STELIAN	21-AUG-05	M		2	3	32	130 100	LIBER CUPA DE TOAMNA 2024		5	3	1
3 MARCU	MARIA	30-DEC-00	F		1	1	44	150 100	LIBER CUPA DE TOAMNA 2024		7	3	1
2 MICU	DENISA	26-APR-05	F		3	2	46	140 100	LIBER CUPA DE TOAMNA 2024		6	3	1
1 DRAGU	STELIAN	21-AUG-05	M		1	1	60	150 200	BRAS CUPA DE TOAMNA 2024		5	4	1
1 DRAGU	STELIAN	21-AUG-05	M		9	3	27	130 50	LIBER CUPA DE ANUL NOU 2025		5	1	3
1 DRAGU	STELIAN	21-AUG-05	M		15	2	24	14 50	LIBER CUPA DE IARNA 2024		5	1	2
3 MARCU	MARIA	30-DEC-00	F		1	1	28	15 50	LIBER CUPA DE IARNA 2024		7	1	2
2 MICU	DENISA	26-APR-05	F		2	2	29	14 50	LIBER CUPA DE IARNA 2024		6	1	2
5 MISTI	SAR	15-OCT-11	M		20	5	32	110 50	LIBER CUPA DE TOAMNA 2024		9	1	1
3 MARCU	MARIA	30-DEC-00	F		10	4	33	120 50	LIBER CUPA DE TOAMNA 2024		7	1	1
4 STAN	FILIP	25-MAR-11	M		9	2	20	140 50	LIBER CUPA DE TOAMNA 2024		8	1	1
1 DRAGU	STELIAN	21-AUG-05	M		15	3	26	130 50	LIBER CUPA DE TOAMNA 2024		5	1	1
2 MICU	DENISA	26-APR-05	F		1	1	27	150 50	LIBER CUPA DE TOAMNA 2024		6	1	1
1 DRAGU	STELIAN	21-AUG-05	M		3	1	27	150 50	SPATE CUPA DE TOAMNA 2024		5	2	1
5 MISTI	SAR	15-OCT-11	M		1	1	25	150 50	SPATE CUPA DE TOAMNA 2024		9	2	1
4 STAN	FILIP	25-MAR-11	M		2	2	26	140 50	SPATE CUPA DE TOAMNA 2024		8	2	1

### III.Rezolvarea proceselor din cadrul aplicației 6-12,17 (celelalte fiind deja rezolvate mai sus)

#### 1. Crearea de proceduri de administrator

P6

```
CREATE OR REPLACE PROCEDURE vizualizare_rezultate_per_categorie (
```

```
    p_competitie IN VARCHAR2,
```

```
    p_proba IN VARCHAR2
```

```
)
```

IS

```
    CURSOR categorie_cursor IS
```

```
        SELECT DISTINCT ca.nume AS categorie
```

```
        FROM frnmp_app_admin.istoric i
```

```
        JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
```

```
        JOIN frnmp_app_admin.categorie ca ON s.cod_categoria = ca.cod_categoria
```

```
        WHERE i.cod_competitie = (
```

```
SELECT cod_competitie
FROM frnmp_app_admin.competitie
WHERE nume = p_competitie
)
AND i.cod_proba = (
    SELECT cod_proba
    FROM frnmp_app_admin.proba
    WHERE nume = p_proba
);
CURSOR rezultat_cursor(p_categorie VARCHAR2) IS
SELECT
    u.nume || ' ' || u.prenume AS sportiv,
    c.nume AS club,
    u2.nume || ' ' || u2.prenume AS antrenor,
    i.loc_per_categorie
FROM
    frnmp_app_admin.istoric i
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
JOIN frnmp_app_admin.antrenor a ON u.cod_antrenor = a.cod_antrenor
JOIN frnmp_app_admin.utilizator u2 ON a.cod_antrenor = u2.cod_utilizator
JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
WHERE
    s.cod_categorie = (
        SELECT cod_categorie
        FROM frnmp_app_admin.categorie
        WHERE nume = p_categorie
    );
```

```

)
AND i.cod_competitie = (
    SELECT cod_competitie
    FROM frnmp_app_admin.competitie
    WHERE nume = p_competitie
)
AND i.cod_proba = (
    SELECT cod_proba
    FROM frnmp_app_admin.proba
    WHERE nume = p_proba
)
ORDER BY i.loc_per_categorie;

v_categoria VARCHAR2(50);

BEGIN
    DBMS_OUTPUT.PUT_LINE('Rezultatele pentru competitia: ' || p_competitie || ' la proba: ' ||
        p_proba);
    DBMS_OUTPUT.PUT_LINE(");
    FOR categorie_rec IN categorie_cursor LOOP
        v_categoria := categorie_rec.categorie;
        DBMS_OUTPUT.PUT_LINE('Categoria: ' || v_categoria);
        DBMS_OUTPUT.PUT_LINE('-----');

        FOR rezultat_rec IN rezultat_cursor(v_categoria) LOOP
            DBMS_OUTPUT.PUT_LINE('Loc: ' || rezultat_rec.loc_per_categorie || ', Sportiv: ' ||
                rezultat_rec.sportiv || ', Club: ' || rezultat_rec.club || ', Antrenor: ' || rezultat_rec.antrenor );
        END LOOP;
    END LOOP;

```

```
DBMS_OUTPUT.PUT_LINE(");  
END LOOP;  
END;  
/
```

Logat ca frnmp\_antrenor4  
SET SERVEROUTPUT ON;  
EXEC frnmp\_app\_admin.vizualizare\_rezultate\_per\_categorie('CUPA DE TOAMNA 2024', '50 LIBER');

```
Rezultatele pentru competitia: CUPA DE TOAMNA 2024 la proba: 50 LIBER  
  
Categoria: SENIOR F  
-----  
Loc: 1, Sportiv: MICU DENISA, Club: AQUA TEAM, Antrenor: STAN DENISA  
Loc: 4, Sportiv: MARCU MARIA, Club: DELFINUL BUCURESTI, Antrenor: TANCI MARIA  
  
Categoria: SENIOR M  
-----  
Loc: 3, Sportiv: DRAGU STELIAN, Club: AQUA TEAM, Antrenor: BRAN IONEL  
  
Categoria: JUNIOR 2 M  
-----  
Loc: 2, Sportiv: STAN FILIP, Club: BUCHAREST SPORT CLUB ELITE, Antrenor: GRIGORAS MARIN  
Loc: 5, Sportiv: MISTI SAR, Club: AQUA TEAM, Antrenor: BRAN IONEL  
  
  
PL/SQL procedure successfully completed.
```

P7

--O ALTA METODA PENTRU P7

```
CREATE OR REPLACE PROCEDURE vizualizare_rezultate_per_gen_si_total_puncte (  
    v_competitie IN VARCHAR2,  
    v_proba IN VARCHAR2  
)  
IS  
    CURSOR gen_cursor IS  
        SELECT DISTINCT s.gen AS gen
```

```
FROM frnmp_app_admin.istoric i
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
WHERE i.cod_competitie =
    (
        SELECT cod_competitie
        FROM frnmp_app_admin.competitie
        WHERE nume = v_competitie
    )
AND i.cod_proba =
    (
        SELECT cod_proba
        FROM frnmp_app_admin.proba
        WHERE nume = v_proba
    );
CURSOR rezultat_cursor(p_gen VARCHAR2) IS
SELECT
    u.nume || ' ' || u.prenume AS sportiv,
    c.nume AS club,
    i.loc_per_categorie,
    i.loc_per_gen,
    ca.nume
FROM
    frnmp_app_admin.istoric i
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
JOIN frnmp_app_admin.categorie ca ON s.cod_categorie = ca.cod_categorie
WHERE
    s.gen = p_gen
```

```
AND i.cod_competitie = (
    SELECT cod_competitie
    FROM frnmp_app_admin.competitie
    WHERE nume = v_competitie
)
AND i.cod_proba = (
    SELECT cod_proba
    FROM frnmp_app_admin.proba
    WHERE nume = v_proba
)
ORDER BY i.loc_per_gen;
```

```
CURSOR podium_puncte_cursor(p_gen VARCHAR2) IS
SELECT *
FROM(
    SELECT
        ROW_NUMBER() OVER (ORDER BY SUM(i.puncte) DESC) AS podium_loc,
        u.nume || ' ' || u.prenume AS sportiv,
        c.nume AS club,
        sum(i.PUNCTE) as punctaj
    FROM
        frnmp_app_admin.istoric i
        JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
        JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
        JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
```

```
WHERE
```

```

    s.gen = p_gen
    AND i.cod_competitie =
        SELECT cod_competitie
        FROM frnmp_app_admin.competitie
        WHERE nume = v_competitie
    )
    group by u.nume,u.prenume,c.nume
    ORDER BY sum(i.PUNCTE) desc
)
WHERE podium_loc <=3;

v_gen VARCHAR2(1);

BEGIN
    DBMS_OUTPUT.PUT_LINE('Rezultatele per gen pentru competitia: ' || v_competitie || " la
    proba: ' || v_proba);
    DBMS_OUTPUT.PUT_LINE(");
    FOR gen_rec IN gen_cursor LOOP
        v_gen := gen_rec.gen;
        DBMS_OUTPUT.PUT_LINE('Gen: ' || v_gen);
        DBMS_OUTPUT.PUT_LINE('-----');
        DBMS_OUTPUT.PUT_LINE('Top 3 sportivi per gen in functie de numarul total de puncte
        obtinute in total la aceasta competitie');

        FOR rezultat_rec IN podium_puncte_cursor(v_gen) LOOP
            DBMS_OUTPUT.PUT_LINE(rezultat_rec.podium_loc||": punctaj total: ' ||
            rezultat_rec.punctaj ||", Sportiv: ' || rezultat_rec.sportiv ||', Club: ' || rezultat_rec.club);
        
```

```
END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');
DBMS_OUTPUT.PUT_LINE('Clasamentul pe general pentru proba si competitia aleasa');
FOR rezultat_rec IN rezultat_cursor(v_gen) LOOP
    DBMS_OUTPUT.PUT_LINE('Loc general: ' || rezultat_rec.loc_per_gen ||', Sportiv: ' ||
    rezultat_rec.sportiv ||', Loc categ: ' || rezultat_rec.loc_per_categorie ||', Club: ' ||
    rezultat_rec.club);
END LOOP;

DBMS_OUTPUT.PUT_LINE('');
END LOOP;
END;
/
```

Logat ca frnmp\_antrenor4

EXEC frnmp\_app\_admin.vizualizare\_rezultate\_per\_gen\_si\_total\_puncte('CUPA DE TOAMNA 2024', '50 LIBER');

Rezultatele per gen pentru competitia: CUPA DE TOAMNA 2024 la proba: 50 LIBER

Gen: M

-----  
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie  
1: punctaj total: 560, Sportiv: DRAGU STELIAN, Club: AQUA TEAM  
2: punctaj total: 280, Sportiv: STAN FILIP, Club: BUCHAREST SPORT CLUB ELITE  
3: punctaj total: 260, Sportiv: MISTI SAR, Club: AQUA TEAM

-----  
Clasamentul pe general pentru proba si competitia aleasa

Loc general: 9, Sportiv: STAN FILIP, Loc categ: 2, Club: BUCHAREST SPORT CLUB ELITE  
Loc general: 15, Sportiv: DRAGU STELIAN, Loc categ: 3, Club: AQUA TEAM  
Loc general: 20, Sportiv: MISTI SAR, Loc categ: 5, Club: AQUA TEAM

Gen: F

-----  
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie  
1: punctaj total: 290, Sportiv: MICU DENISA, Club: AQUA TEAM  
2: punctaj total: 270, Sportiv: MARCU MARIA, Club: DELFINUL BUCURESTI

-----  
Clasamentul pe general pentru proba si competitia aleasa

Loc general: 1, Sportiv: MICU DENISA, Loc categ: 1, Club: AQUA TEAM  
Loc general: 10, Sportiv: MARCU MARIA, Loc categ: 4, Club: DELFINUL BUCURESTI

PL/SQL procedure successfully completed.

```
EXEC frnmp_app_admin.vizualizare_rezultate_per_gen_si_total_puncte('CUPA DE TOAMNA 2024', '100 LIBER');
```

```
Rezultatele per gen pentru competitia: CUPA DE TOAMNA 2024 la proba: 100 LIBER

Gen: M
-----
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie
1: punctaj total: 560, Sportiv: DRAGU STELIAN, Club: AQUA TEAM
2: punctaj total: 280, Sportiv: STAN FILIP, Club: BUCHAREST SPORT CLUB ELITE
3: punctaj total: 260, Sportiv: MISTI SAR, Club: AQUA TEAM
-----
Clasamentul pe general pentru proba si competitia aleasa
Loc general: 2, Sportiv: DRAGU STELIAN, Loc categ: 3, Club: AQUA TEAM

Gen: F
-----
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie
1: punctaj total: 290, Sportiv: MICU DENISA, Club: AQUA TEAM
2: punctaj total: 270, Sportiv: MARCU MARIA, Club: DELFINUL BUCURESTI
-----
Clasamentul pe general pentru proba si competitia aleasa
Loc general: 1, Sportiv: MARCU MARIA, Loc categ: 1, Club: DELFINUL BUCURESTI
Loc general: 3, Sportiv: MICU DENISA, Loc categ: 2, Club: AQUA TEAM
```

P8

```
CREATE OR REPLACE PROCEDURE get_best_times_for_sportiv(
    p_nume_sp IN VARCHAR2,
    p_prenume_sp IN VARCHAR2
) AS
    v_cod_sp NUMBER;
    CURSOR best_of(p_cod_sp NUMBER) IS
        SELECT
            p.nume AS proba,
            i.timp_total AS timp,
            TO_CHAR(c.data_start, 'DD-MM-YYYY') AS dataS,
            TO_CHAR(c.data_final, 'DD-MM-YYYY') AS dataF
        FROM
            frnmp_app_admin.istoric i
```

```

join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
join frnmp_app_admin.proba p on i.cod_proba = p.cod_proba

WHERE
    i.cod_sportiv = p.cod_sp
    AND i.timp_total= (
        SELECT MIN(i2.timp_total)
        FROM frnmp_app_admin.istoric i2
        WHERE i2.cod_proba = i.cod_proba
        AND i2.cod_sportiv = i.cod_sportiv
    )
    ORDER BY p.nume;

BEGIN
    IF (REGEXP_LIKE(p_nume_sp, '^[A-Za-z]+$') AND REGEXP_LIKE(p_prenume_sp, '^[A-Za-z]+$')) THEN
        SELECT s.cod_sportiv
        INTO v_cod_sp
        FROM frnmp_app_admin.sportiv s
        JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
        WHERE u.nume = p_nume_sp
        AND u.prenume = p_prenume_sp;

        IF v_cod_sp IS NOT NULL THEN
            DBMS_OUTPUT.PUT_LINE('BEST OF TIME PENTRU sportivul: ' || p_nume_sp || ' ' ||
                p_prenume_sp);
            DBMS_OUTPUT.PUT_LINE('-----');

            FOR rec IN best_of(v_cod_sp) LOOP
                DBMS_OUTPUT.PUT_LINE(rec.proba || ' - ' || rec.timp || ' s, in perioada competitiei: ' ||
                    rec.dataS || ' - ' || rec.dataF);
        END IF;
    END IF;
END;

```

```
END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

ELSE

    RAISE_APPLICATION_ERROR(-20032, 'Error');

END IF;

ELSE

    DBMS_OUTPUT.PUT_LINE('Numele sau prenumele contine caractere invalide!');

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20031, 'ERROR');

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20033, 'Error');

END;
/
```

Logat ca frnmp\_antrenor4

```
EXEC frnmp_app_admin.get_best_times_for_sportiv('DRAGU', 'STELIAN');
```

```
BEST OF TIME PENTRU sportivul: DRAGU STELIAN
-----
100 LIBER - 30 s, in perioada competitiei: 03-01-2025 - 04-01-2025
200 BRAS - 60 s, in perioada competitiei: 05-11-2024 - 06-11-2024
50 LIBER - 24 s, in perioada competitiei: 10-12-2024 - 11-12-2024
50 SPATE - 27 s, in perioada competitiei: 05-11-2024 - 06-11-2024
-----
```

--Numele sau prenumele contine caractere invalide!

```
EXEC frnmp_app_admin.get_best_times_for_sportiv('DRAGU4', 'STELIAN');
```

```
Numele sau prenumele contine caractere invalide!
```

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

P9

--P9

```

CREATE OR REPLACE PROCEDURE evolutie_sportiv(
    p_nume_sp IN VARCHAR2,
    p_prenume_sp IN VARCHAR2
)
IS
    v_cod_sp NUMBER;
    TYPE timp_tab IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
    v_timp i timp_tab;

    CURSOR probe_cursor IS
        SELECT DISTINCT p.nume AS proba
        FROM frnmp_app_admin.istoric i
        JOIN frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
        WHERE i.cod_sportiv = v_cod_sp;

    CURSOR istoric_cursor_sp(p_proba VARCHAR2,p_cod_sp NUMBER) IS
        SELECT
            i.timp_total AS timp,
            TO_CHAR(c.data_start, 'DD-MM-YYYY') AS dataS,
            TO_CHAR(c.data_final, 'DD-MM-YYYY') AS dataF
        FROM
            frnmp_app_admin.istoric i
            join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
            join frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
        WHERE
            i.cod_sportiv = p_cod_sp

```

```

AND UPPER(p.nume) = UPPER(p_proba)
order by c.data_start;

v_proba VARCHAR2(30);
v_index PLS_INTEGER;
BEGIN
IF (REGEXP_LIKE(p_nume_sp, '^[A-Za-z]+$') AND REGEXP_LIKE(p_prenume_sp, '^[A-
Za-z]+$')) THEN
SELECT s.cod_sportiv
INTO v_cod_sp
FROM frnmp_app_admin.sportiv s
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
WHERE upper(u.nume) = upper(p_nume_sp)
AND upper(u.prenume) = upper(p_prenume_sp);

IF v_cod_sp IS NOT NULL THEN
DBMS_OUTPUT.PUT_LINE('Evolutia sportivului: ' || p_nume_sp || ' ' || p_prenume_sp);
DBMS_OUTPUT.PUT_LINE('-----');
FOR proba_rec IN probe_cursor LOOP
    v_proba := proba_rec.proba;
    DBMS_OUTPUT.PUT_LINE('Evolutia in proba: '|| v_proba);
    v_timp.DELETE;
    v_index := 0;
    FOR rec IN istoric_cursor_sp(v_proba,v_cod_sp) LOOP
        DBMS_OUTPUT.PUT_LINE(rec.timp || ' s, in perioada competitiei: ' || rec.dataS ||' -
' ||rec.dataF);
        v_index := v_index + 1;
    END LOOP;
END LOOP;
END IF;
END;

```

```

v_timp(v_index) := rec.timp;
END LOOP;

DBMS_OUTPUT.PUT_LINE(' ');
DBMS_OUTPUT.PUT_LINE('Fluctuatia evolutiei');

IF v_timp.COUNT = 1 THEN
    DBMS_OUTPUT.PUT_LINE(v_timp(1));
END IF;

IF v_timp.COUNT > 1 THEN
    FOR i IN 1..v_timp.COUNT LOOP
        DBMS_OUTPUT.PUT(v_timp(i));
        IF i < v_timp.COUNT THEN
            IF v_timp(i + 1) < v_timp(i) THEN
                DBMS_OUTPUT.PUT(' \'); --TIMPUL ESTE IMBUNATATIT FATA DE
CONCURSUL ANTERIOR
            ELSE
                DBMS_OUTPUT.PUT(' /'); -- A SCOS UN TEMP MAI MARE DECAT IN
CONCURSUL ANTERIOR
            END IF;
        END IF;
    END LOOP;
    END IF;
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('-----');
END LOOP;

ELSE
    RAISE_APPLICATION_ERROR(-20032, 'Error');
END IF;

ELSE

```

```
DBMS_OUTPUT.PUT_LINE('Numele sau prenumele contin caractere invalide!');

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20031, 'ERROR');

END;
/
```

Logat ca frnmp\_antrenor4  
EXEC frnmp\_app\_admin.evolutie\_sportiv('DRAGU', 'STELIAN');

---

Evolutia sportivului: DRAGU STELIAN

---

Evolutia in proba: 100 LIBER  
32 s, in perioada competitiei: 05-11-2024 - 06-11-2024  
30 s, in perioada competitiei: 03-01-2025 - 04-01-2025

Fluctuatie evolutiei  
32 \ 30

---

Evolutia in proba: 50 LIBER  
26 s, in perioada competitiei: 05-11-2024 - 06-11-2024  
24 s, in perioada competitiei: 10-12-2024 - 11-12-2024  
27 s, in perioada competitiei: 03-01-2025 - 04-01-2025

Fluctuatie evolutiei  
26 \ 24 / 27

---

Evolutia in proba: 50 SPATE  
27 s, in perioada competitiei: 05-11-2024 - 06-11-2024

Fluctuatie evolutiei  
27

---

Evolutia in proba: 200 BRAS  
60 s, in perioada competitiei: 05-11-2024 - 06-11-2024

Fluctuatie evolutiei  
60

---

PL/SQL procedure successfully completed.

```
EXEC frnmp_app_admin.get_best_times_for_sportiv('DRAGU4', 'STELIAN');
```

```
Numele sau prenumele contine caractere invalide!
PL/SQL procedure successfully completed.
```

P10

```
--p10
```

```
--Vizualizarea top 3 cluburi pentru o competi?ie aleas? în func?ie de num?rul de medalii ob?inute la categorie
```

```
SET SERVEROUTPUT ON;
```

```
CREATE OR REPLACE PROCEDURE top_cluburi(
```

```
    p_nume_comp IN VARCHAR2)
```

```
IS
```

```
    v_cod_comp NUMBER;
```

```
CURSOR medalii_club_cursor(p_cod_comp NUMBER) IS
```

```
    SELECT *
```

```
    FROM(
```

```
        SELECT
```

```
            ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS podium_loc,
```

```
            c.nume AS club,
```

```
            COUNT(*) AS nr_locuri_podium_categ
```

```
    FROM frnmp_app_admin.istoric i
```

```
    JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
```

```
    JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
```

```
    JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
```

```
    WHERE i.cod_competitie = p_cod_comp
```

```
        AND i.loc_per_categorie IN (1, 2, 3) -- Se iau doar locurile 1, 2 sau 3
```

```
    GROUP BY c.nume
```

```
ORDER BY COUNT(*) DESC)
WHERE podium_loc <=3;
BEGIN
  SELECT cod_competitie
  INTO v_cod_comp
  FROM frnmp_app_admin.competitie
  WHERE UPPER(nume) = UPPER(p_nume_comp);

  IF v_cod_comp IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Top 3 cluburi pentru competitia'|| p_nume_comp);
    DBMS_OUTPUT.PUT_LINE('-----');
    FOR rec IN medalii_club_cursor(v_cod_comp) LOOP
      DBMS_OUTPUT.PUT_LINE('LOC'||rec.podium_loc || ' '|| rec.club || ' cu '
      ||rec.nr_locuri_podium_categ|| ' medalii');
    END LOOP;
    ELSE
      RAISE_APPLICATION_ERROR(-20032, 'Error');
    END IF;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20031, 'ERROR');
  END;
/

```

Logat ca frnmp\_antrenor4  
EXEC frnmp\_app\_admin.top\_cluburi('CUPA DE TOAMNA 2024');

```
Top 3 cluburi pentru competitia CUPA DE TOAMNA 2024
```

```
LOC 1 AQUA TEAM cu 7 medalii
LOC 2 BUCHAREST SPORT CLUB ELITE cu 2 medalii
LOC 3 DELFINUL BUCURESTI cu 1 medalii
```

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

P11

```
--P11
```

```
--Vizualizarea top 3 sportivi pentru anul, clubul ?i proba aleas?
```

```
SET SERVEROUTPUT ON;
```

```
/
```

```
CREATE OR REPLACE PROCEDURE top_sportivi_an_cl_proba(
```

```
    p_nume_club IN VARCHAR2,
```

```
    p_nume_proba IN VARCHAR2,
```

```
    p_an IN NUMBER
```

```
)
```

```
IS
```

```
    v_cod_club NUMBER;
```

```
    v_cod_proba NUMBER;
```

```
    CURSOR top_sportivi_cursor(p_cod_club NUMBER, p_cod_proba NUMBER, p_an  
NUMBER) IS
```

```
        SELECT *
```

```
        FROM(
```

```
            SELECT
```

```
                ROW_NUMBER() OVER (ORDER BY SUM(i.puncte) DESC) AS podium_loc,
```

```
                u.nume ||' '||u.prenume as sportiv,
```

```
                SUM(i.puncte) AS total_puncte
```

```
            FROM frnmp_app_admin.istoric i
```

```
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
JOIN frnmp_app_admin.proba p ON i.cod_proba = p.cod_proba
JOIN frnmp_app_admin.competitie co ON i.cod_competitie = co.cod_competitie
WHERE c.cod_club = p.cod_club
AND p.cod_proba = p.cod_proba
AND EXTRACT(YEAR FROM co.data_start) = p_an
GROUP BY u.nume,u.prenume
ORDER BY SUM(i.puncte) DESC)
WHERE podium_loc <=3;

BEGIN
SELECT cod_club
INTO v_cod_club
FROM frnmp_app_admin.club
WHERE UPPER(nume) = UPPER(p_nume_club);

SELECT cod_proba
INTO v_cod_proba
FROM frnmp_app_admin.proba
WHERE UPPER(nume) = UPPER(p_nume_proba);

IF v_cod_club IS NOT NULL and v_cod_proba IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Top 3 SPORTIVI pentru clubul '|| p_nume_club ||' cu
participari la proba: '|| p_nume_proba ||' in anul '|| to_char(p_an));
    DBMS_OUTPUT.PUT_LINE('-----');
FOR rec IN top_sportivi_cursor(v_cod_club,v_cod_proba,p_an) LOOP
```

```
DBMS_OUTPUT.PUT_LINE('LOC'||rec.podium_loc||' '||rec.sportiv||' cu '
||rec.total_puncte||' puncte');

END LOOP;

ELSE

    RAISE_APPLICATION_ERROR(-20032, 'Error');

END IF;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        RAISE_APPLICATION_ERROR(-20031, 'ERROR');

    WHEN OTHERS THEN

        RAISE_APPLICATION_ERROR(-20033, 'Error');

END;
/
```

Logat ca frnmp\_guest

```
EXEC frnmp_app_admin.top_sportivi_an_cl_proba('aqua team','50 liber',2024);
```

```
Top 3 SPORTIVI pentru clubul aqua team cu participari la proba: 50 liber in anul 2024
-----
LOC 1 MICU DENISA cu 164 puncte
LOC 2 DRAGU STELIAN cu 144 puncte
LOC 3 MISTI SAR cu 110 puncte
```

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

2. Crearea de blocuri pentru rezolvarea proceselor 6-12 rulate logat ca frnmp\_app\_sportiv1, frnmp\_app\_sportiv2

P6

```
select * from istoric;
select * from proba;
SELECT * FROM COMPETITIE;
--implementare P6: Vizualizarea rezultatelor per categorie a tuturor sportivilor pentru o
--competi?ie ?i o prob? aleas? ?i a c?atig?torilor, incluzându-se numele clubului ?i al
--antrenorului
```

```
--CERERA DE PREGATIRE

--select u.numere ||' '| u.prenume sportiv, c.numere club, u2.numere ||' '| u2.prenume antrenor,
      ca.numere categorie, loc_per_categorie

--from frnmp_app_admin.istoric i

--join frnmp_app_admin.sportiv s on i.cod_sportiv = s.cod_sportiv

--join frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator

--join frnmp_app_admin.antrenor a on u.cod_antrenor = a.cod_antrenor

--join frnmp_app_admin.utilizator u2 on a.cod_antrenor = u2.cod_utilizator

--join frnmp_app_admin.club c on u.cod_club = c.cod_club

--join frnmp_app_admin.categorie ca on s.cod_categorie = ca.cod_categorie

--where i.cod_competitie = (select cod_competitie

--          from frnmp_app_admin.competitie

--          where nume = 'CUPA DE TOAMNA 2024')

--  and i.cod_proba = (select cod_proba

--          from frnmp_app_admin.proba

--          where nume = '50 LIBER')

--order by s.cod_categorie, i.loc_per_categorie;

SET SERVEROUTPUT ON;

--in acest mod este folosit rolul vizualizari_rezultate

DECLARE

  v_competitie VARCHAR2(100) := '&v_competitie'; -- CUPA DE TOAMNA 2024
  v_proba VARCHAR2(100) := '&v_proba'; -- 50 LIBER

CURSOR categorie_cursor IS

  SELECT DISTINCT ca.numere AS categorie
    FROM frnmp_app_admin.istoric i
   JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
   JOIN frnmp_app_admin.categorie ca ON s.cod_categorie = ca.cod_categorie
```

```
WHERE i.cod_competitie = (
    SELECT cod_competitie
    FROM frnmp_app_admin.competitie
    WHERE nume = v_competitie
)
AND i.cod_proba = (
    SELECT cod_proba
    FROM frnmp_app_admin.proba
    WHERE nume = v_proba
);

```

```
CURSOR rezultat_cursor(p_categorie VARCHAR2) IS
SELECT
    u.nume || ' ' || u.prenume AS sportiv,
    c.nume AS club,
    u2.nume || ' ' || u2.prenume AS antrenor,
    i.loc_per_categorie
FROM
    frnmp_app_admin.istoric i
    JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
    JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
    JOIN frnmp_app_admin.antrenor a ON u.cod_antrenor = a.cod_antrenor
    JOIN frnmp_app_admin.utilizator u2 ON a.cod_antrenor = u2.cod_utilizator
    JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
WHERE
    s.cod_categorie = (
        SELECT cod_categorie
        FROM frnmp_app_admin.categorie
)
```

```

        WHERE nume = p_categorie
    )
    AND i.cod_competitie =
        SELECT cod_competitie
        FROM frnmp_app_admin.competitie
        WHERE nume = v_competitie
    )
    AND i.cod_proba =
        SELECT cod_proba
        FROM frnmp_app_admin.proba
        WHERE nume = v_proba
    )
ORDER BY i.loc_per_categorie;

v_categorie VARCHAR2(50);

BEGIN
    DBMS_OUTPUT.PUT_LINE('Rezultatele pentru competitia: ' || v_competitie || ' la proba: '
    || v_proba);
    DBMS_OUTPUT.PUT_LINE(");
    FOR categorie_rec IN categorie_cursor LOOP
        v_categorie := categorie_rec.categorie;
        DBMS_OUTPUT.PUT_LINE('Categoria: ' || v_categorie);
        DBMS_OUTPUT.PUT_LINE('-----');
        FOR rezultat_rec IN rezultat_cursor(v_categorie) LOOP
            DBMS_OUTPUT.PUT_LINE('Loc: ' || rezultat_rec.loc_per_categorie || ', Sportiv: ' ||
            rezultat_rec.sportiv || ', Club: ' || rezultat_rec.club || ', Antrenor: ' ||
            rezultat_rec.antrenor );
        END LOOP;
    END LOOP;
END;

```

```
END LOOP;

DBMS_OUTPUT.PUT_LINE(");

END LOOP;

END;

/
```

```
-----  
Rezultatele pentru competitia: CUPA DE TOAMNA 2024 la proba: 50 LIBER  
  
Categoria: SENIOR F  
-----  
Loc: 1, Sportiv: MICU DENISA, Club: AQUA TEAM, Antrenor: STAN DENISA  
Loc: 4, Sportiv: MARCU MARIA, Club: DELFINUL BUCURESTI, Antrenor: TANCI MARIA  
  
Categoria: SENIOR M  
-----  
Loc: 3, Sportiv: DRAGU STELIAN, Club: AQUA TEAM, Antrenor: BRAN IONEL  
  
Categoria: JUNIOR 2 M  
-----  
Loc: 2, Sportiv: STAN FILIP, Club: BUCHAREST SPORT CLUB ELITE, Antrenor: GRIGORAS MARIN  
Loc: 5, Sportiv: MISTI SAR, Club: AQUA TEAM, Antrenor: BRAN IONEL  
  
  
PL/SQL procedure successfully completed.
```

P7

```
SET SERVEROUTPUT ON;
--p7
--Vizualizarea rezultatelor per gen a tuturor sportivilor pentru competi?ia aleas?
--si vizualizarea celor mai buni 3 sportivi ?i celor mai bune 3 sportive ale competi?iei
--(având în vedere num?rul total de puncte ob?inute în cadrul competi?iei), cu men?ionarea
--clubului de apartenen??.  
  
DECLARE
    v_competitie VARCHAR2(100) := '&v_competitie'; -- CUPA DE TOAMNA 2024
    v_proba VARCHAR2(100) := '&v_proba'; -- 50 LIBER
```

```

CURSOR gen_cursor IS
  SELECT DISTINCT s.gen AS gen
  FROM frnmp_app_admin.istoric i
  JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
  WHERE i.cod_competitie =
    SELECT cod_competitie
    FROM frnmp_app_admin.competitie
    WHERE nume = v_competitie
  )
  AND i.cod_proba =
    SELECT cod_proba
    FROM frnmp_app_admin.proba
    WHERE nume = v_proba
);

```

```

CURSOR rezultat_cursor(p_gen VARCHAR2) IS
  SELECT
    u.nume || ' ' || u.prenume AS sportiv,
    c.nume AS club,
    i.loc_per_categorie,
    i.loc_per_gen,
    ca.nume
  FROM
    frnmp_app_admin.istoric i
    JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
    JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
    JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
    JOIN frnmp_app_admin.categorie ca ON s.cod_categorie = ca.cod_categorie

```

```

WHERE
    s.gen = p_gen

    AND i.cod_competitie = (
        SELECT cod_competitie
        FROM frnmp_app_admin.competitie
        WHERE nume = v_competitie
    )

    AND i.cod_proba = (
        SELECT cod_proba
        FROM frnmp_app_admin.proba
        WHERE nume = v_proba
    )

    ORDER BY i.loc_per_gen;

CURSOR podium_puncte_cursor(p_gen VARCHAR2) IS
    SELECT *
    FROM(
        SELECT
            ROW_NUMBER() OVER (ORDER BY SUM(i.puncte) DESC) AS podium_loc,
            u.nume || ' ' || u.prenume AS sportiv,
            c.nume AS club,
            sum(i.PUNCTE) as punctaj
        FROM
            frnmp_app_admin.istoric i
            JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
            JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
            JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
    )

```

```

WHERE
    s.gen = p_gen
    AND i.cod_competitie =
        SELECT cod_competitie
        FROM frnmp_app_admin.competitie
        WHERE nume = v_competitie
    )
    group by u.nume,u.prenume,c.nume
    ORDER BY sum(i.PUNCTE) desc
)
WHERE podium_loc <=3;

v_gen VARCHAR2(1);

BEGIN
    DBMS_OUTPUT.PUT_LINE('Rezultatele per gen pentru competitia: ' || v_competitie ||' la
    proba: ' || v_proba);
    DBMS_OUTPUT.PUT_LINE(");
    FOR gen_rec IN gen_cursor LOOP
        v_gen := gen_rec.gen;
        DBMS_OUTPUT.PUT_LINE('Gen: ' || v_gen);
        DBMS_OUTPUT.PUT_LINE('-----');
        DBMS_OUTPUT.PUT_LINE('Top 3 sportivi per gen in functie de numarul total de
        puncte obtinute in total la aceasta competitie');

        FOR rezultat_rec IN podium_puncte_cursor(v_gen) LOOP
            DBMS_OUTPUT.PUT_LINE(rezultat_rec.podium_loc||': punctaj total: ' ||
            rezultat_rec.punctaj ||', Sportiv: ' || rezultat_rec.sportiv ||', Club: ' || rezultat_rec.club);
        END LOOP;
    END LOOP;
END;

```

```
END LOOP;

DBMS_OUTPUT.PUT_LINE('-----');

DBMS_OUTPUT.PUT_LINE('Clasamentul pe general pentru proba si competitia aleasa');

FOR rezultat_rec IN rezultat_cursor(v_gen) LOOP

    DBMS_OUTPUT.PUT_LINE('Loc general: ' || rezultat_rec.loc_per_gen ||', Sportiv: '
    || rezultat_rec.sportiv || ', Loc categ: ' || rezultat_rec.loc_per_categorie ||', Club: ' ||
    rezultat_rec.club);

    END LOOP;

    DBMS_OUTPUT.PUT_LINE(");

END LOOP;

END;

/
```

```
Gen: M
-----
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie
1: punctaj total: 560, Sportiv: DRAGU STELIAN, Club: AQUA TEAM
2: punctaj total: 280, Sportiv: STAN FILIP, Club: BUCHAREST SPORT CLUB ELITE
3: punctaj total: 260, Sportiv: MISTI SAR, Club: AQUA TEAM
-----
Clasamentul pe general pentru proba si competitia aleasa
Loc general: 9, Sportiv: STAN FILIP, Loc categ: 2, Club: BUCHAREST SPORT CLUB ELITE
Loc general: 15, Sportiv: DRAGU STELIAN, Loc categ: 3, Club: AQUA TEAM
Loc general: 20, Sportiv: MISTI SAR, Loc categ: 5, Club: AQUA TEAM

Gen: F
-----
Top 3 sportivi per gen in functie de numarul total de puncte obtinute in total la aceasta competitie
1: punctaj total: 290, Sportiv: MICU DENISA, Club: AQUA TEAM
2: punctaj total: 270, Sportiv: MARCU MARIA, Club: DELFINUL BUCURESTI
-----
Clasamentul pe general pentru proba si competitia aleasa
Loc general: 1, Sportiv: MICU DENISA, Loc categ: 1, Club: AQUA TEAM
Loc general: 10, Sportiv: MARCU MARIA, Loc categ: 4, Club: DELFINUL BUCURESTI

PL/SQL procedure successfully completed.
```

P8

--p8

SET SERVEROUTPUT ON;

--Pentru un sportiv ales s? se vizualizeze cel mai bun timp

--pentru toate probele la care a participat de-a lungul timpului, al?turi de data ob?inerii.

DECLARE

v\_nume\_sp VARCHAR2(20) := '&v\_nume\_sp'; -- DRAGU

v\_prenume\_sp VARCHAR2(20) := '&v\_prenume\_sp'; -- STELIAN

v\_cod\_sp NUMBER;

CURSOR best\_of(p\_cod\_sp NUMBER) IS

SELECT

p.nume AS proba,

i.timp\_total AS timp,

TO\_CHAR(c.data\_start, 'DD-MM-YYYY') AS dataS,

TO\_CHAR(c.data\_final, 'DD-MM-YYYY') AS dataF

FROM

frnmp\_app\_admin.istoric i

join frnmp\_app\_admin.competitie c on i.cod\_competitie = c.cod\_competitie

join frnmp\_app\_admin.proba p on i.cod\_proba = p.cod\_proba

WHERE

i.cod\_sportiv = p\_cod\_sp

AND i.timp\_total= (

SELECT MIN(i2.timp\_total)

FROM frnmp\_app\_admin.istoric i2

WHERE i2.cod\_proba = i.cod\_proba

AND i2.cod\_sportiv = i.cod\_sportiv

)

```

    ORDER BY p.nume;

BEGIN

IF (REGEXP_LIKE(v_nume_sp, '^[A-Za-z]+$') AND REGEXP_LIKE(v_prenume_sp,
    '^[A-Za-z]+$')) THEN

    SELECT s.cod_sportiv

    INTO v_cod_sp

    FROM frnmp_app_admin.sportiv s

    JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator

    WHERE upper(u.nume) = upper(v_nume_sp)

    AND upper(u.prenume) = upper(v_prenume_sp);

    IF v_cod_sp IS NOT NULL THEN

        DBMS_OUTPUT.PUT_LINE('BEST OF TIME PENTRU sportivul: ' || v_nume_sp
        || ' ' || v_prenume_sp);

        DBMS_OUTPUT.PUT_LINE('-----');

        FOR rec IN best_of(v_cod_sp) LOOP

            DBMS_OUTPUT.PUT_LINE(rec.proba || ' - ' || rec.timp || ' s, in perioada
            competitiei: ' || rec.dataS ||' - ' ||rec.dataF);

            END LOOP;

            DBMS_OUTPUT.PUT_LINE('-----');

        ELSE

            RAISE_APPLICATION_ERROR(-20032, 'Error');

        END IF;

    ELSE

        DBMS_OUTPUT.PUT_LINE('Numele sau prenumele contin caractere invalide!');

    END IF;

```

```
EXCEPTION  
  WHEN NO_DATA_FOUND THEN  
    RAISE_APPLICATION_ERROR(-20031, 'ERROR');  
END;  
/  
-----
```

```
BEST OF TIME PENTRU sportivul: DRAGU STELIAN  
-----  
100 LIBER - 30 s, in perioada competitiei: 03-01-2025 - 04-01-2025  
200 BRAS - 60 s, in perioada competitiei: 05-11-2024 - 06-11-2024  
50 LIBER - 24 s, in perioada competitiei: 10-12-2024 - 11-12-2024  
50 SPATE - 27 s, in perioada competitiei: 05-11-2024 - 06-11-2024  
-----
```

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

P9

```
--p9  
--Pentru un sportiv ales să se observe evolu?ia pe probe  
--de la toate competi?iile la care a participat ?i să se vad? perioadele de evolu?ie  
SET SERVEROUTPUT ON;
```

DECLARE

```
  v_nume_sp VARCHAR2(20) := '&v_nume_sp'; -- DRAGU  
  v_prenume_sp VARCHAR2(20) := '&v_prenume_sp'; -- STELIAN  
  v_cod_sp NUMBER;  
  TYPE timp_tab IS TABLE OF NUMBER INDEX BY PLS_INTEGER;  
  v_timpri timp_tab;
```

```
CURSOR probe_cursor IS  
  SELECT DISTINCT p.nume AS proba  
  FROM frnmp_app_admin.istoric i
```

```

JOIN frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
WHERE i.cod_sportiv = v_cod_sp;

CURSOR istoric_cursor_sp(p_proba VARCHAR2,p_cod_sp NUMBER) IS
SELECT
    i.timp_total AS timp,
    TO_CHAR(c.data_start, 'DD-MM-YYYY') AS dataS,
    TO_CHAR(c.data_final, 'DD-MM-YYYY') AS dataF
FROM
    frnmp_app_admin.istoric i
    join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
    join frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
WHERE
    i.cod_sportiv = p_cod_sp
    AND UPPER(p.nume) = UPPER(p_proba)
    order by c.data_start;

v_proba VARCHAR2(30);
v_index PLS_INTEGER;
BEGIN
IF (REGEXP_LIKE(v_nume_sp, '^[A-Za-z]+$') AND REGEXP_LIKE(v_prenume_sp,
    '^[A-Za-z]+$')) THEN
    SELECT s.cod_sportiv
    INTO v_cod_sp
    FROM frnmp_app_admin.sportiv s
    JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
    WHERE upper(u.nume) = upper(v_nume_sp)
    AND upper(u.prenume) = upper(v_prenume_sp);

```

```

IF v_cod_sp IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('Evolutia sportivului: ' || v_nume_sp || ' ' ||
    v_prenume_sp);
    DBMS_OUTPUT.PUT_LINE('-----');

FOR proba_rec IN probe_cursor LOOP
    v_proba := proba_rec.proba;
    DBMS_OUTPUT.PUT_LINE('Evolutia in proba: '|| v_proba);
    v_timp.DELETE;
    v_index := 0;
    FOR rec IN istoric_cursor_sp(v_proba,v_cod_sp) LOOP
        DBMS_OUTPUT.PUT_LINE(rec.timp || ' s, in perioada competitiei: ' ||
        rec.dataS ||' - '||rec.dataF);
        v_index := v_index + 1;
        v_timp(v_index) := rec.timp;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(' ');
    DBMS_OUTPUT.PUT_LINE('Fluctuatie evolutiei');
    IF v_timp.COUNT = 1 THEN
        DBMS_OUTPUT.PUT_LINE(v_timp(1));
    END IF;
    IF v_timp.COUNT > 1 THEN
        FOR i IN 1..v_timp.COUNT LOOP
            DBMS_OUTPUT.PUT(v_timp(i));
            IF i < v_timp.COUNT THEN
                IF v_timp(i + 1) < v_timp(i) THEN
                    DBMS_OUTPUT.PUT(' \'); --TIMPUL ESTE IMBUNATATIT FATA
                    DE CONCURSUL ANTERIOR
                END IF;
            END IF;
        END LOOP;
    END IF;
END IF;

```

```
        ELSE
            DBMS_OUTPUT.PUT(' '); -- A SCOS UN TIMP MAI MARE DECAT
            IN CONCURSUL ANTERIOR
        END IF;
        END IF;
    END LOOP;
    END IF;
    DBMS_OUTPUT.PUT_LINE('');
    DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
ELSE
    RAISE_APPLICATION_ERROR(-20032, 'Error');
END IF;
ELSE
    DBMS_OUTPUT.PUT_LINE('Numele sau prenumele contin caractere invalide!');
END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20031, 'ERROR');
END;
/

```

```
Evolutia sportivului: DRAGU STELIAN
-----
Evolutia in proba: 100 LIBER
32 s, in perioada competitiei: 05-11-2024 - 06-11-2024
30 s, in perioada competitiei: 03-01-2025 - 04-01-2025

Fluctuatiua evolutiei
32 \ 30
-----
Evolutia in proba: 50 LIBER
26 s, in perioada competitiei: 05-11-2024 - 06-11-2024
24 s, in perioada competitiei: 10-12-2024 - 11-12-2024
27 s, in perioada competitiei: 03-01-2025 - 04-01-2025

Fluctuatiua evolutiei
26 \ 24 / 27
-----
Evolutia in proba: 50 SPATE
27 s, in perioada competitiei: 05-11-2024 - 06-11-2024

Fluctuatiua evolutiei
27
-----
Evolutia in proba: 200 BRAS
60 s, in perioada competitiei: 05-11-2024 - 06-11-2024

Fluctuatiua evolutiei
60
-----
PL/SQL procedure successfully completed.
```

P10

```
--p10
--Vizualizarea top 3 cluburi pentru o competi?ie aleas? în func?ie de num?rul de medalii
ob?inute la categorie

SET SERVEROUTPUT ON;

DECLARE
    v_nume_comp VARCHAR2(30) := '&v_nume_comp'; -- CUPA DE TOAMNA 2024
    v_cod_comp NUMBER;

    CURSOR medalii_club_cursor(p_cod_comp NUMBER) IS
        SELECT *
        FROM(
            SELECT
```

```

ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS podium_loc,
c.nume AS club,
COUNT(*) AS nr_locuri_podium_categ
FROM frnmp_app_admin.istoric i
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
WHERE i.cod_competitie = p_cod_comp
AND i.loc_per_categorie IN (1, 2, 3)
GROUP BY c.nume
ORDER BY COUNT(*) DESC)

WHERE podium_loc <=3;

BEGIN
SELECT cod_competitie
INTO v_cod_comp
FROM frnmp_app_admin.competitie
WHERE UPPER(nume) = UPPER(v_nume_comp);

IF v_cod_comp IS NOT NULL THEN
DBMS_OUTPUT.PUT_LINE('Top 3 cluburi pentru competitia'|| v_nume_comp);
DBMS_OUTPUT.PUT_LINE('-----');

FOR rec IN medalii_club_cursor(v_cod_comp) LOOP
DBMS_OUTPUT.PUT_LINE('LOC'||rec.podium_loc||' '||rec.club||' cu '
||rec.nr_locuri_podium_categ||' medalii');

END LOOP;

ELSE
RAISE_APPLICATION_ERROR(-20032, 'Error');

```

```
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20031, 'ERROR');
END;
/
```

```
Top 3 cluburi pentru competitia CUPA DE TOAMNA 2024
```

```
-----  
LOC 1 AQUA TEAM cu 7 medalii  
LOC 2 BUCHAREST SPORT CLUB ELITE cu 2 medalii  
LOC 3 DELFINUL BUCURESTI cu 1 medalii
```

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

P11

```
--P11
--Vizualizarea top 3 sportivi pentru anul, clubul ?i proba aleas?
SET SERVEROUTPUT ON;
DECLARE
    v_an NUMBER := '&v_an'; --2024
    v_nume_club VARCHAR2(50) := '&v_nume_club'; -- aqua team / delfinul bucuresti
    v_nume_proba VARCHAR2(30) := '&v_nume_proba'; -- 50 liber
    v_cod_club NUMBER;
    v_cod_proba NUMBER;

CURSOR top_sportivi_cursor(p_cod_club NUMBER, p_cod_proba NUMBER) IS
    SELECT *
    FROM(
        SELECT
            ROW_NUMBER() OVER (ORDER BY SUM(i.puncte) DESC) AS podium_loc,
```

```

u.nume ||' '||u.prenume as sportiv,
SUM(i.puncte) AS total_puncte
FROM frnmp_app_admin.istoric i
JOIN frnmp_app_admin.sportiv s ON i.cod_sportiv = s.cod_sportiv
JOIN frnmp_app_admin.utilizator u ON s.cod_sportiv = u.cod_utilizator
JOIN frnmp_app_admin.club c ON u.cod_club = c.cod_club
JOIN frnmp_app_admin.proba p ON i.cod_proba = p.cod_proba
JOIN frnmp_app_admin.competitie co ON i.cod_competitie = co.cod_competitie
WHERE c.cod_club = p.cod_club
AND p.cod_proba = p.cod_proba
AND EXTRACT(YEAR FROM co.data_start) = v_an
GROUP BY u.nume,u.prenume
ORDER BY SUM(i.puncte) DESC)
WHERE podium_loc <=3;
BEGIN
SELECT cod_club
INTO v_cod_club
FROM frnmp_app_admin.club
WHERE UPPER(nume) = UPPER(v_nume_club);

SELECT cod_proba
INTO v_cod_proba
FROM frnmp_app_admin.proba
WHERE UPPER(nume) = UPPER(v_nume_proba);

IF v_cod_club IS NOT NULL and v_cod_proba IS NOT NULL THEN
DBMS_OUTPUT.PUT_LINE('Top 3 SPORTIVI pentru clubul '|| v_nume_club ||' cu
participari la proba: ' || v_nume_proba ||' in anul ' || to_char(v_an));

```

```

DBMS_OUTPUT.PUT_LINE('-----');

FOR rec IN top_sportivi_cursor(v_cod_club,v_cod_proba) LOOP
    DBMS_OUTPUT.PUT_LINE('LOC '||rec.podium_loc || ' ' || rec.sportiv || ' cu '
    ||rec.total_puncte|| ' puncte');
END LOOP;

ELSE
    RAISE_APPLICATION_ERROR(-20032, 'Error');
END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20031, 'ERROR');
END;
/

```

```

-----,
Top 3 SPORTIVI pentru clubul AQUA TEAM cu participari la proba: 50 LIBER in anul 2024
-----
LOC 1 MICU DENISA cu 164 puncte
LOC 2 DRAGU STELIAN cu 144 puncte
LOC 3 MISTI SAR cu 110 puncte

PL/SQL procedure successfully completed.

```

### 3. Rezolvarea procesului 12

Logat ca frnmp\_reprezentant1

```
--p12
select * from frnmp_app_admin.tip;
select * from frnmp_app_admin.istoric;
select * from frnmp_app_admin.COMPETITIE;
select * from frnmp_app_admin.oras;
select * from frnmp_app_admin.TARA;
```

```
select * from frnmp_app_admin.viz_sponsorizare_2;

select DISTINCT(s.cod_sportiv), u.nume||' '||u.prenume as nume, s.numar_legitimatie
from frnmp_app_admin.sportiv s
join frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on i.cod_sportiv = s.cod_sportiv
join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
join frnmp_app_admin.tip t on c.cod_tip = t.cod_tip
join frnmp_app_admin.oras o on c.cod_oras = o.cod_oras
join frnmp_app_admin.tara ta on o.cod_tara = ta.cod_tara
LEFT join frnmp_app_admin.viz_sponsorizare_2 sp on s.cod_sportiv = sp.cod_sportiv
where
UPPER(t.tip_competitie) IN ('EUROPEAN','MONDIAL') AND
(o.nume_oras = 'BUCURESTI' OR ta.nume_tara <> 'JAPONIA') AND
i.loc_per_gen <=16
```

#### MINUS

```
select DISTINCT(s.cod_sportiv), u.nume||' '||u.prenume as nume,s.numar_legitimatie
from frnmp_app_admin.sportiv s
join frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on i.cod_sportiv = s.cod_sportiv
join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
join frnmp_app_admin.tip t on c.cod_tip = t.cod_tip
join frnmp_app_admin.oras o on c.cod_oras = o.cod_oras
join frnmp_app_admin.tara ta on o.cod_tara = ta.cod_tara
RIGHT join frnmp_app_admin.viz_sponsorizare_2 sp on s.cod_sportiv = sp.cod_sportiv
where
sp.cod_sponsor = 1 and
```

UPPER(t.tip\_competitie) IN ('EUROPEAN','MONDIAL') AND  
(o.numere\_oras = 'BUCURESTI' OR ta.numere\_tara <> 'JAPONIA') AND  
i.loc\_per\_gen <=16 AND sp.inceput\_colaborare<=trunc(sysdate) and  
sp\_sfarsit\_colaborare>=trunc(sysdate); --in acest fel sunt eliminati sportivii care au fost macar  
vreodata sponsorizati si care nu sunt sponsorizati in ziua actuala

COD_SPORTIV	NUME	NUMAR_LEGITIMATIE
6	MICU DENISA	2
7	MARCU MARIA	3
8	STAN FILIP	4
9	MISTI SAR	5

Nu apare sportivul cu codul 1 care îndeplinește condițiile de sponsorizare în ceea ce privește rezultatele la competiții, pentru că el este sponsorizat de acest sponsor inclusiv azi 23.01.2025 când a fost rulat codul.

COD_SPONSOR	INCEPUT_COLABORARE	SFARSIT_COLABORARE	COD_SPORTIV
1	07-JAN-25	01-MAR-25	SPORTIV1

Logat ca frnmp\_reprezentant2

```

select * from frnmp_app_admin.tip;
select * from frnmp_app_admin.istoric;
select * from frnmp_app_admin.COMPETITIE;
select * from frnmp_app_admin.oras;
select * from frnmp_app_admin.TARA;
select * from frnmp_app_admin.viz_sponsorizare_2;
SELECT * FROM FRNMP_APP_ADMIN.viz_sponsorizare WHERE cod_sponsor=2;
--p12
select DISTINCT(s.cod_sportiv), u.nume||' '||u.prenume as nume, s.numar_legitimatie
from frnmp_app_admin.sportiv s
join frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on i.cod_sportiv = s.cod_sportiv
join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
join frnmp_app_admin.tip t on c.cod_tip = t.cod_tip

```

```
join frnmp_app_admin.oras o on c.cod_oras = o.cod_oras
join frnmp_app_admin.tara ta on o.cod_tara = ta.cod_tara
LEFT join frnmp_app_admin.viz_sponsorizare_2 sp on s.cod_sportiv = sp.cod_sportiv
where
UPPER(t.tip_competitie) IN ('EUROPEAN','MONDIAL') AND
(o.nume_oras = 'BUCURESTI' OR ta.nume_tara <> 'JAPONIA') AND
i.loc_per_gen <=16
```

## MINUS

```
select DISTINCT(s.cod_sportiv), u.nume||' '||u.prenume as nume,s.numar_legitimatie
from frnmp_app_admin.sportiv s
join frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on i.cod_sportiv = s.cod_sportiv
join frnmp_app_admin.competitie c on i.cod_competitie = c.cod_competitie
join frnmp_app_admin.tip t on c.cod_tip = t.cod_tip
join frnmp_app_admin.oras o on c.cod_oras = o.cod_oras
join frnmp_app_admin.tara ta on o.cod_tara = ta.cod_tara
RIGHT join frnmp_app_admin.viz_sponsorizare_2 sp on s.cod_sportiv = sp.cod_sportiv
where
sp.cod_sponsor = 2 and
UPPER(t.tip_competitie) IN ('EUROPEAN','MONDIAL') AND
(o.nume_oras = 'BUCURESTI' OR ta.nume_tara <> 'JAPONIA') AND
i.loc_per_gen <=16 AND sp.inceput_colaborare<=trunc(sysdate) and
sp.sfarsit_colaborare>=trunc(sysdate); --in acest fel sunt eliminati sportivii care au fost macar
vreodata sponsorizati si care nu sunt sunt sponsorizati in ziua actuala
```

COD_SPORTIV	NUME	NUMAR_LEGITIMATIE
5	DRAGU STELIAN	1
7	MARCU MARIA	3
8	STAN FILIP	4
9	MISTI SAR	5

Nu apare sportivul cu codul 2 care îndeplinește condițiile de sponsorizare în ceea ce privește rezultatele la competiții, pentru că el este sponsorizat de acest sponsor inclusiv azi 23.01.2025 când a fost rulat codul.

COD_SPONSOR	INCEPUT_COLABORARE	SFARSIT_COLABORARE	COD_SPORTIV
1	2 18-JAN-25	23-JAN-25	SPORTIV2

#### 4. Rezolvarea procesului 17

Logat ca frnmp\_antrenor1

--p17

```

select * from frnmp_app_admin.sportiv;
select * from
(
select
s.cod_sportiv, s.numar_legitimatie, p.nume, sum(i.puncte),
row_number() over(
    PARTITION BY s.cod_sportiv
    order by sum(i.puncte) DESC) as rank
from frnmp_app_admin.sportiv s
JOIN frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on s.cod_sportiv = i.cod_sportiv
join frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
where cod_antrenor = 1
group by s.cod_sportiv, numar_legitimatie, p.nume) where rank<2;
```

COD_SPORTIV	NUMAR_LEGITIMATIE	NUME	SUM(I.PUNCTE)	RANK
5	1 100	LIBER	280	1
9	5 50	SPATE	150	1

Logat ca frnmp\_antrenor2

--p17

```

select * from frnmp_app_admin.sportiv;
select * from
(
select
s.cod_sportiv, s.numar_legitimatie, p.nume, sum(i.puncte),
row_number() over(
PARTITION BY s.cod_sportiv
order by sum(i.puncte) DESC) as rank
from frnmp_app_admin.sportiv s
JOIN frnmp_app_admin.utilizator u on s.cod_sportiv = u.cod_utilizator
join frnmp_app_admin.istoric i on s.cod_sportiv = i.cod_sportiv
join frnmp_app_admin.proba p on i.cod_proba = p.cod_proba
where cod_antrenor = 2
group by s.cod_sportiv, numar_legitimatie, p.nume) where rank<2;
```

COD_SPORTIV	NUMAR_LEGITIMATIE	NUME	SUM(I.PUNCTE)	RANK
6	2 50	LIBER	164	1

## IV. Un exemplu de update realizat de administrator pentru o competiție

--PENTRU A TESTA POLITICA AUDIT

set serveroutput on;

```

INSERT INTO COMPETITIE(cod_competitie,nume,data_start,data_final,cod_oras,cod_tip)
VALUES(4,'CUPA DE PRIMAVARA 2025',TO_DATE('03-03-2025','DD-MM-
YYYY'),TO_DATE('04-03-2025','DD-MM-YYYY'),4,4);
```

```
COMMIT;  
UPDATE frnmp_app_admin.competitie  
SET data_final = TO_DATE('05-03-2025','DD-MM-YYYY')  
WHERE cod_competitie = 4;  
COMMIT;  
  
DELETE FROM competitie  
WHERE cod_competitie = 4;  
  
select * from competitie;  
commit;
```

Mesaj care apare din audit pentru că s-a modificat data de desfășurare a competiției.

Modificare date competitie

COD_COMPETITIE	NUME	DATA_START	DATA_FINAL	COD_ORAS	COD_TIP
4	CUPA DE PRIMAVARA 2025	03-MAR-25	05-MAR-25	4	4