

Hutan Mihai-Alexandru
Grupa 243

Gestiunea competitiei Esports de League of Legends

EX1

Prezentati pe scurt baza de date (utilitatea ei).

Baza de date contine informatii cu privire la turneele acestei competitii, grupele si echipele participante, cat si jucatorii, ce meciuri au acestia, prezentatorii, scena si modul de joc al meciurilor. Apar detalii legate de jucatori si echipa. Jucatorii si echipele apartin unei regiuni, iar jucatorii joaca in mod frecvent anumiti campioni salvati in baza de date.

Scopul crearii acestei baze de date este de a facilita generarea de statistici, ce ulterior vor fi analizate in cadrul evenimentelor de tip Esports.

Aceasta competitie are un numar foarte mare de turnee an de an. Fiecare turneu contine informatii relevante precum numele, orasul si data inceperii acestuia si este identificat printr-un id unic.

Un turneu contine mai multe grupe identificate printr-un id unic, ce au si o denumire.

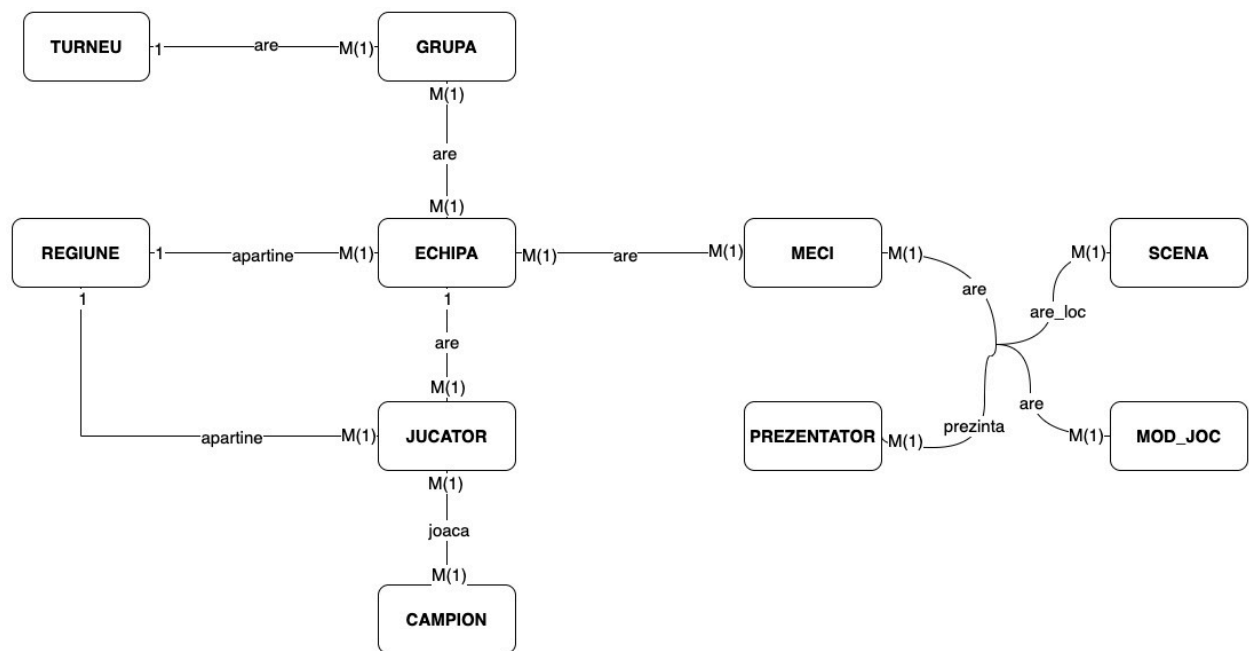
Fiecare grupa contine mai multe echipe despre care cunoastem numele, tag-ul, id-ul si regiunea din care provin.

Regiunea reprezinta un atribut si in cazul jucatorilor despre care mai cunoastem numele, lane-ul si id-ul, cat si campionii pe care ii joaca in mod frecvent, care la randul lor au un id unic, un titlu si resursa folosita.

Fiind un turneu de competitie Esports, vor avea loc meciuri, care vor fi comentate de diferiti prezentatori, vor avea loc intre diferite echipe, moduri de joc multiple. Cunosand toate aceste detalii putem crea un MeciHeader ce va ajuta echipa de marketing pentru a promova meciurile importante.

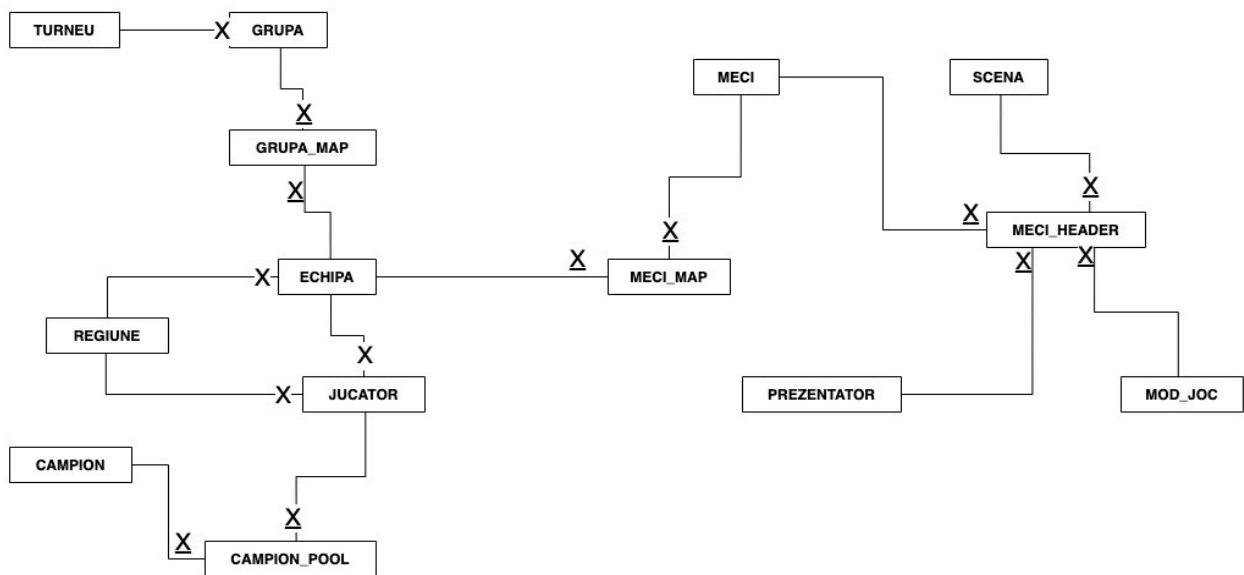
EX2

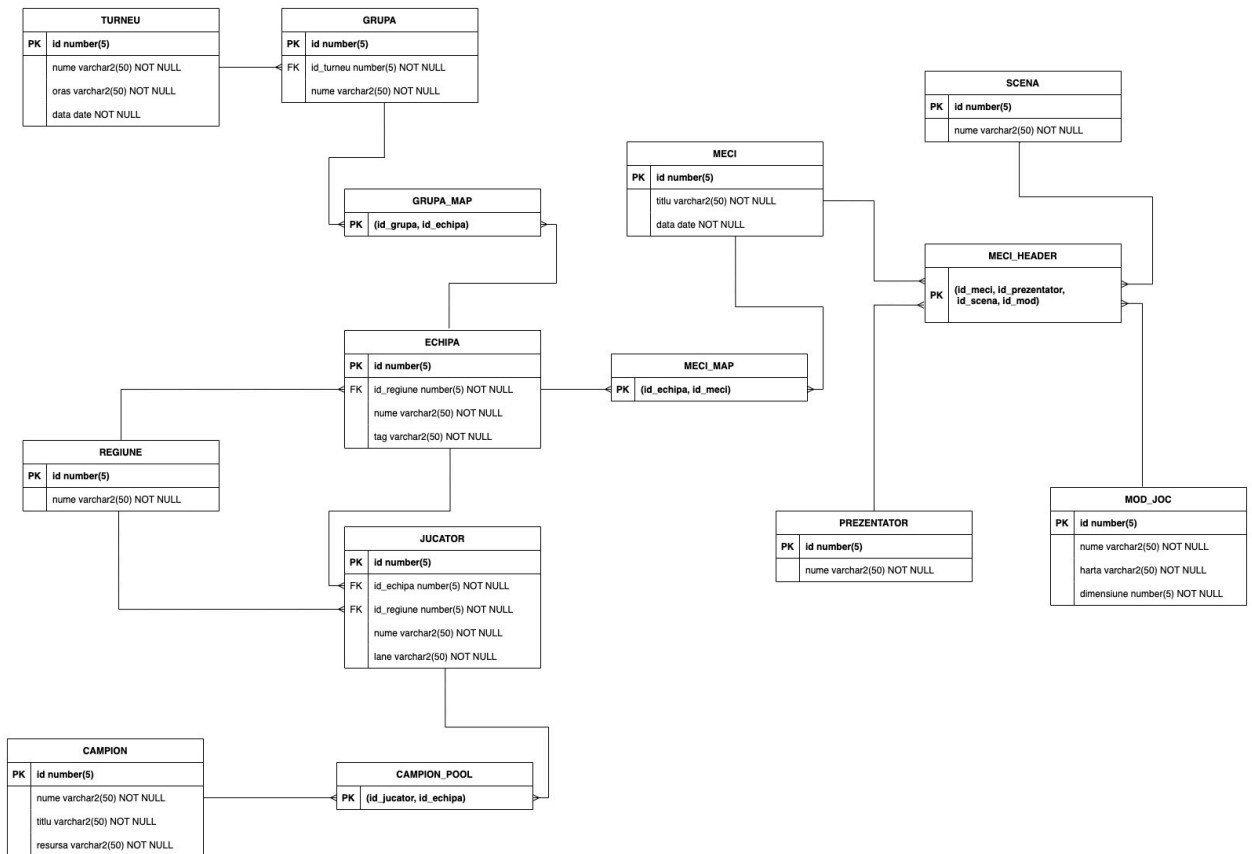
Realizati diagrama entitate-relatie (ERD).



EX3

Pornind de la diagrama entitate-relatie realizați diagrama conceptuala a modelului propus, integrand toate attributele necesare.





EX4 & EX5

Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).

```

CREATE TABLE scena
(
  id NUMBER(5) GENERATED ALWAYS AS IDENTITY
  CONSTRAINT pkey_scena PRIMARY KEY,
  nume VARCHAR2(50)
  CONSTRAINT nume_scena NOT NULL
)
  
```

```
INSERT INTO scena(ume)
VALUES ('Main Stage')
```

```
INSERT INTO scena(ume)
VALUES ('Second Stage')
```

```
INSERT INTO scena(ume)
VALUES ('Third Stage')
```

```
INSERT INTO scena(ume)
VALUES ('Horn Stage')
```

```
INSERT INTO scena(ume)
VALUES ('Panther Stage')
```

```
COMMIT;
```

```
----- PREZENTATOR -----
```

```
CREATE TABLE prezentator
(
  id  NUMBER(5) GENERATED ALWAYS AS IDENTITY
      CONSTRAINT pkey_prezentator PRIMARY KEY,
  ume VARCHAR2(50)
      CONSTRAINT nume_prezentator NOT NULL
)
```

```
INSERT INTO prezentator(ume)
VALUES ('James Patterson')
```

```
INSERT INTO prezentator(ume)
VALUES ('David Turley')
```

```
INSERT INTO prezentator(ume)
VALUES ('Sam Hartman')
```

```
INSERT INTO prezentator(ume)
```

```
VALUES ('Mark Zimmerman')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Emily Rand')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Barento Mohammed')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Julian Carr')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Isaac Bentley')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Max Anderson')
```

```
INSERT INTO presentator(ume)
```

```
VALUES ('Jordan Corby')
```

```
COMMIT;
```

```
----- MOD_JOC -----
```

```
CREATE TABLE mod_joc
```

```
(
```

```
  id      NUMBER(5) GENERATED ALWAYS AS IDENTITY
```

```
  CONSTRAINT pkey_mod_joc PRIMARY KEY,
```

```
  ume     VARCHAR2(50)
```

```
  CONSTRAINT ume_mod_joc NOT NULL,
```

```
  harta   VARCHAR2(50)
```

```
  CONSTRAINT harta_mod_joc NOT NULL,
```

```
  dimensiune NUMBER(5)
```

```
  CONSTRAINT dimensiune_mod_joc NOT NULL
```

```
)
```

```
INSERT INTO mod_joc(ume, harta, dimensiune)
```

```

VALUES ('Normal', 'Summoners Rift', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Ranked', 'Summoners Rift', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Flex', 'Summoners Rift', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Normal', 'Summoners Rift', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Normal', 'Twisted Treelines', '3')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Ranked', 'Twisted Treelines', '3')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Flex', 'Twisted Treelines', '3')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Normal', 'ARAM', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Ranked', 'ARAM', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Flex', 'ARAM', '5')

INSERT INTO mod_joc(ume, harta, dimensiune)
VALUES ('Ranked', 'ARAM', '1')

COMMIT;

```

```

----- MECI -----
CREATE TABLE meci

```

```

(
  id  NUMBER(5) GENERATED ALWAYS AS IDENTITY
      CONSTRAINT pkey_meci PRIMARY KEY,
  titlu VARCHAR2(50)
      CONSTRAINT titlu_meci NOT NULL,
  data  DATE
      CONSTRAINT data_meci NOT NULL
)

INSERT INTO meci(titlu, data)
VALUES ('Revenge', TO_DATE('2022-05-10', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Revenge', TO_DATE('2022-05-11', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Warriors', TO_DATE('2022-05-13', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Destroyers', TO_DATE('2022-05-14', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Debut', TO_DATE('2021-10-01', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Academy', TO_DATE('2021-10-02', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('Academy', TO_DATE('2021-05-03', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('RunnerUps', TO_DATE('2022-01-12', 'YYYY-MM-DD'))

INSERT INTO meci(titlu, data)
VALUES ('RunnerUps', TO_DATE('2022-01-12', 'YYYY-MM-DD'))

```



```
INSERT INTO meci(titlu, data)
VALUES ('Academy', TO_DATE('2021-01-13', 'YYYY-MM-DD'))
```

```
COMMIT;
```

```
----- CAMPION -----
```

```
CREATE TABLE champion
(
  id    NUMBER(5) GENERATED ALWAYS AS IDENTITY
        CONSTRAINT pkey_champion PRIMARY KEY,
  nume  VARCHAR2(50)
        CONSTRAINT name_champion NOT NULL,
  titlu VARCHAR2(50)
        CONSTRAINT titlu_champion NOT NULL,
  resursa VARCHAR2(50)
        CONSTRAINT resursa_champion NOT NULL
)
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Aatrox', 'the Darkin Blade', 'Blood Well')
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Ahri', 'the Nine-Tailed Fox', 'Mana')
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Akali', 'the Rogue Assassin', 'Energy')
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Alistar', 'the Minotaur', 'Mana')
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Amumu', 'the Sad Mummy', 'Mana')
```

```
INSERT INTO champion(nume, titlu, resursa)
VALUES ('Anivia', 'the Cryophoenix', 'Mana')
```

```

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Annie', 'the Dark Child', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Ashe', 'the Frost Archer', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Fiddlesticks', 'the Ancient Fear', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Fizz', 'the Tidal Trickster', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Gangplank', 'the Saltwater Scourge', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Hecarim', 'the Shadow of War', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Jax', 'Grandmaster at Arms', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Jayce', 'the Defender of Tomorrow', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Jhin', 'the Virtuoso', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Jinx', 'the Loose Cannon', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('KhaZix', 'the Voidreaver', 'Mana')

INSERT INTO champion(serie, titlu, resursa)
VALUES ('Lee Sin', 'the Blind Monk', 'Energy')

```

```
INSERT INTO campion(ume, titlu, resursa)
VALUES ('Zed', 'the Master of Shadows', 'Energy')
```

```
INSERT INTO campion(ume, titlu, resursa)
VALUES ('Lux', 'the Lady of Luminosity', 'Mana')
```

```
COMMIT;
```

```
----- TURNEU -----
```

```
CREATE TABLE turneu
(
  id  NUMBER(5) GENERATED ALWAYS AS IDENTITY
      CONSTRAINT pkey_turneu PRIMARY KEY,
  ume VARCHAR2(50)
      CONSTRAINT ume_turneu NOT NULL,
  oras VARCHAR2(50)
      CONSTRAINT oras_turneu NOT NULL,
  data DATE
      CONSTRAINT data_turneu NOT NULL
)
```

```
INSERT INTO turneu(ume, oras, data)
VALUES ('MSI', 'Roma', TO_DATE('2022-05-10', 'YYYY-MM-DD'))
```

```
INSERT INTO turneu(ume, oras, data)
VALUES ('ALLSTARS', 'New York', TO_DATE('2022-01-12', 'YYYY-MM-DD'))
```

```
INSERT INTO turneu(ume, oras, data)
VALUES ('WORLDS', 'Beijing', TO_DATE('2021-10-01', 'YYYY-MM-DD'))
```

```
INSERT INTO turneu(ume, oras, data)
VALUES ('LCK-SUMMER', 'Seul', TO_DATE('2021-06-05', 'YYYY-MM-DD'))
```

```
INSERT INTO turneu(ume, oras, data)
VALUES ('NA-SPRING', 'Boston', TO_DATE('2021-03-13', 'YYYY-MM-DD'))
```

```

INSERT INTO turneu(nome, oras, data)
VALUES ('EUW-SUMMER', 'Paris', TO_DATE('2022-06-05', 'YYYY-MM-DD'))

INSERT INTO turneu(nome, oras, data)
VALUES ('NA-SUMMER', 'Los Angeles', TO_DATE('2022-07-03', 'YYYY-MM-DD'))

INSERT INTO turneu(nome, oras, data)
VALUES ('EUW-SPRING', 'London', TO_DATE('2022-04-01', 'YYYY-MM-DD'))

INSERT INTO turneu(nome, oras, data)
VALUES ('LCK-SPRING', 'Seul', TO_DATE('2021-03-05', 'YYYY-MM-DD'))

INSERT INTO turneu(nome, oras, data)
VALUES ('OCE-SUMMER', 'Sydney', TO_DATE('2022-07-01', 'YYYY-MM-DD'))

COMMIT;

----- REGIUNE -----

CREATE TABLE regiune
(
    id NUMBER(5) GENERATED ALWAYS AS IDENTITY
        CONSTRAINT pkey_regiune PRIMARY KEY,
    nome VARCHAR2(50)
        CONSTRAINT nome_regiune NOT NULL
)

INSERT INTO regiune(nome)
VALUES ('KOR')

INSERT INTO regiune(nome)
VALUES ('EUNE')

INSERT INTO regiune(nome)
VALUES ('EUW')

INSERT INTO regiune(nome)

```

```

VALUES ('NA')

INSERT INTO regiune(ume)
VALUES ('OCE')

INSERT INTO regiune(ume)
VALUES ('SA')

INSERT INTO regiune(ume)
VALUES ('TUR')

INSERT INTO regiune(ume)
VALUES ('RUS')

COMMIT;

----- GRUPA -----

CREATE TABLE grupa
(
    id    NUMBER(5) GENERATED ALWAYS AS IDENTITY
        CONSTRAINT pkey_grupa PRIMARY KEY,
    ume   VARCHAR2(50)
        CONSTRAINT ume_grupa NOT NULL,
    id_turneu NUMBER(5) NOT NULL,
        CONSTRAINT fk_turneu_grupa FOREIGN KEY (id_turneu) REFERENCES turneu (id)
)

INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 1)

INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 1)

INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 1)

```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 1)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 2)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 2)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 2)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 2)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 3)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 3)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 3)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 3)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 4)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 4)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 4)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 4)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 5)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 5)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 5)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 5)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('A', 6)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('B', 6)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('C', 6)
```

```
INSERT INTO grupa(ume, id_turneu)
VALUES ('D', 6)
```

```
COMMIT;
```

```
----- ECHIPA -----
```

```
CREATE TABLE echipa
(
    id      NUMBER(5) GENERATED ALWAYS AS IDENTITY
           CONSTRAINT pkey_echipa PRIMARY KEY,
    ume     VARCHAR2(50)
           CONSTRAINT ume_echipa NOT NULL,
```

```

tag    VARCHAR2(50)
      CONSTRAINT tag_echipa NOT NULL,
id_regiune NUMBER(5) NOT NULL,
      CONSTRAINT fk_regiune_echipa FOREIGN KEY (id_regiune) REFERENCES regiune (id)
)

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (1, 'SKT-T1', 'SKT')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (1, 'Royal Never Give Up', 'RNG')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (3, 'G2 Esports', 'G2')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (4, 'Evil Geniuses ', 'EG')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (6, 'PSG Talon Esports', 'PSG')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (6, 'Canids Kalunga', 'RED')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (1, 'Team Aze', 'AZE')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (5, 'Order', 'ORD')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (7, 'fastpayWildcats', 'IW')

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (8, 'SaigonBuffaloEsports', 'SGB')

```



```

COMMIT;

----- GRUPA_MAP -----

CREATE TABLE grupa_map
(
    id_grupa NUMBER(5)
        CONSTRAINT pk_grupa_map_grupa REFERENCES grupa (id),
    id_echipa NUMBER(5)
        CONSTRAINT pk_grupa_map_echipa REFERENCES echipa (id),
    CONSTRAINT pk_grupa_map PRIMARY KEY (id_grupa, id_echipa)
)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (1, 1)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (1, 2)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (1, 3)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (1, 4)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (2, 5)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (2, 6)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (2, 7)

INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (2, 8)

```

```
INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (3, 9)
```

```
INSERT INTO grupa_map(id_grupa, id_echipa)
VALUES (3, 10)
```

```
COMMIT;
```

```
----- JUCATOR -----
```

```
CREATE TABLE jucator
(
  id      NUMBER(5) GENERATED ALWAYS AS IDENTITY
          CONSTRAINT pkey_jucator PRIMARY KEY,
  nume    VARCHAR2(50)
          CONSTRAINT nume_jucator NOT NULL,
  lane    VARCHAR2(50)
          CONSTRAINT lane_jucator NOT NULL,
  id_echipa NUMBER(5) NOT NULL,
          CONSTRAINT fk_echipa_jucator FOREIGN KEY (id_echipa) REFERENCES echipa (id),
  id_regiune NUMBER(5) NOT NULL,
          CONSTRAINT fk_regiune_jucator FOREIGN KEY (id_regiune) REFERENCES regiune (id)
)
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (1, 1, 'Zeus', 'TOP')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (1, 1, 'Oner', 'JUNGLE')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (1, 1, 'Faker', 'MID')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (1, 1, 'Gumayusi', 'BOT')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
```

```
VALUES (1, 1, 'Keria', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (2, 1, 'Bin', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (2, 1, 'Wei', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (2, 1, 'Xiaohu', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (2, 1, 'Gala', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (2, 1, 'Ming', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (3, 3, 'BrokenBlade', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (3, 2, 'Jankos', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (3, 3, 'Caps', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (3, 3, 'Flakke', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (3, 3, 'Targamas', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (4, 1, 'Impact', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)
```

```
VALUES (4, 2, 'Inspired', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (4, 4, 'Jojopyun', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (4, 4, 'Danny', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (4, 4, 'Vulcan', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (5, 5, 'Hanabi', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (5, 1, 'Juhan', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (5, 1, 'Bay', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (5, 5, 'Unified', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (5, 5, 'Kaiwing', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (6, 6, 'Guigo', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (6, 6, 'Aegis', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (6, 6, 'Avenger', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)
```

```
VALUES (6, 6, 'TitaN', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (6, 6, 'Jojo', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (7, 1, 'Lonely', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (7, 6, 'Dimitry', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (7, 6, 'Aloned', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (7, 1, '5Kid', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (7, 6, 'Straight', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (8, 5, 'BioPanther', 'TOP')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (8, 5, 'Maximize', 'JUNGLE')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (8, 5, 'Kisee', 'MID')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (8, 5, 'Puma', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)  
VALUES (8, 5, 'Corporal', 'BOT')
```

```
INSERT INTO jucator(id echipa, id regiune, nume, lane)
```

```
VALUES (9, 7, 'StarScreen', 'TOP')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (9, 7, 'Ferret', 'JUNGLE')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (9, 7, 'Serin', 'MID')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (9, 7, 'HolyPhoenix', 'BOT')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (9, 7, 'Farfetch', 'BOT')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (10, 8, 'Hasmed', 'TOP')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (10, 8, 'Bean J', 'JUNGLE')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (10, 8, 'Froggy', 'MID')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (10, 8, 'Shogun', 'BOT')
```

```
INSERT INTO jucator(id_echipa, id_regiune, nume, lane)
VALUES (10, 8, 'Taki', 'BOT')
```

```
COMMIT;
```

```
----- CAMPION_POOL -----
```

```
CREATE TABLE campion_pool
```

```
(
```

```
    id_campion NUMBER(5)
```

```
    CONSTRAINT pk_campion_pool_campion REFERENCES campion (id),
```

```

id_jucator NUMBER(5)
    CONSTRAINT pk_campion_pool_jucator REFERENCES jucator (id),
    CONSTRAINT pk_campion_pool PRIMARY KEY (id_campion, id_jucator)
)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (1, 1)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (1, 3)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (1, 5)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (2, 2)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (2, 4)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (2, 6)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (3, 2)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (3, 3)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (3, 6)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (5, 18)

INSERT INTO campion_pool(id_jucator, id_campion)

```

```
VALUES (4, 20)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (4, 11)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (5, 5)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (6, 10)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (7, 11)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (8, 13)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (8, 14)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (8, 15)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (9, 19)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (9, 20)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (10, 18)

INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (13, 9)

INSERT INTO campion_pool(id_jucator, id_campion)
```



```
VALUES (12, 10)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (11, 11)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (12, 12)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (13, 13)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (14, 14)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (15, 15)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (16, 16)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (15, 17)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (16, 18)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (17, 19)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (17, 20)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (18, 2)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
```

```
VALUES (18, 3)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (17, 18)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (18, 4)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (19, 5)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (20, 6)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (19, 7)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (21, 8)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (21, 9)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (20, 10)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (21, 13)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (22, 12)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (22, 11)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
```

```
VALUES (23, 14)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (23, 15)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (23, 16)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (24, 19)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (24, 20)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (25, 18)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (26, 17)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (27, 16)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (28, 15)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (29, 14)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (30, 13)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (31, 12)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
```

```
VALUES (32, 11)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (33, 10)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (34, 9)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (35, 8)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (36, 7)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (37, 6)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (38, 5)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (39, 4)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (40, 3)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (41, 2)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (42, 1)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)  
VALUES (43, 20)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
```

```
VALUES (44, 19)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (45, 18)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (46, 17)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (47, 16)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (48, 15)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (49, 14)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (50, 13)
```

```
INSERT INTO campion_pool(id_jucator, id_campion)
VALUES (50, 12)
```

```
COMMIT;
```

```
----- MECI_MAP -----
```

```
CREATE TABLE meci_map
(
    id_echipa NUMBER(5)
        CONSTRAINT pk_meci_map_echipa REFERENCES echipa (id),
    id_meci NUMBER(5)
        CONSTRAINT pk_meci_map_meci REFERENCES meci (id),
    CONSTRAINT pk_meci_map PRIMARY KEY (id_echipa, id_meci)
)
```

```
INSERT INTO meci_map(id_echipa, id_meci)
```

```
VALUES (1, 1)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (2, 1)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (3, 2)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (4, 2)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (1, 3)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (3, 3)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (1, 4)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (4, 4)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (2, 5)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (3, 5)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (5, 6)
```

```
INSERT INTO meci_map(id echipa, id meci)  
VALUES (6, 6)
```

```
INSERT INTO meci_map(id echipa, id meci)
```

```
VALUES (5, 7)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (7, 7)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (6, 8)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (7, 8)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (8, 9)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (5, 9)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (5, 10)
```

```
INSERT INTO meci_map(id echipa, id_meci)  
VALUES (6, 10)
```

```
COMMIT;
```

```
----- MECI_HEADER -----
```

```
CREATE TABLE meci_header
```

```
(
```

```
    id_meci    NUMBER(5)
```

```
    CONSTRAINT pk_meci_header_meci REFERENCES meci (id),
```

```
    id_prezentator NUMBER(5)
```

```
    CONSTRAINT pk_meci_header_prezentator REFERENCES prezentator (id),
```

```
    id_scena    NUMBER(5)
```

```
    CONSTRAINT pk_meci_header_scena REFERENCES scena (id),
```

```
    id_mod      NUMBER(5)
```

```
    CONSTRAINT pk_meci_header_mod REFERENCES mod_joc (id),
```

```
CONSTRAINT pk_meci_header PRIMARY KEY (id_meci, id_presentator, id_scena, id_mod)
)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (1, 1, 3, 1)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (2, 2, 1, 3)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (3, 4, 2, 7)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (4, 7, 3, 7)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (5, 9, 2, 10)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (6, 10, 3, 5)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (7, 3, 1, 4)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (8, 6, 1, 8)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (9, 8, 3, 9)

INSERT INTO meci_header(id_meci, id_presentator, id_scena, id_mod)
VALUES (10, 10, 4, 10)

COMMIT
```


EX6

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze două tipuri diferite de colecții studiate. Apelați subprogramul.

```
-- Cerința 6:  
-- Pentru un turneu al cărui nume este dat, pentru fiecare regiune, afisati toate echipele si  
jucatorii ce apartin acelei regiuni sau mesajul 'Nu exista!'.
```

```
CREATE OR REPLACE PROCEDURE ex6(nume_turneu turneu.nume%TYPE)  
AS  
    TYPE tabl_idx IS TABLE OF regiune%ROWTYPE INDEX BY PLS_INTEGER;  
    v_regiuni tabl_idx;  
    TYPE tip_lista_nested IS TABLE OF echipa%ROWTYPE;  
    v_echipe tip_lista_nested := tip_lista_nested();  
    TYPE tabl_index IS TABLE OF VARCHAR2(200) INDEX BY PLS_INTEGER;  
    v_nume    tabl_index;  
    v_numar   NUMBER(5);  
BEGIN  
    SELECT * BULK COLLECT INTO v_regiuni FROM regiune;  
  
    SELECT COUNT(*)  
    INTO v_numar  
    FROM echipa e,  
         turneu t,  
         grupa_map gm,  
         grupa g  
    WHERE e.id = gm.id echipa  
          AND t.id = gm.id_grupa  
          AND g.id = gm.id_grupa  
          AND t.id = g.id_turneu  
          AND UPPER(t.nume) LIKE UPPER(nume_turneu);  
  
    v_echipe.extend(v_numar + 1);
```

```

SELECT e.id, e.num, e.tag, e.id_regiune BULK COLLECT
INTO v_echipe
FROM echipa e,
     turneu t,
     grupa_map gm,
     grupa g
WHERE e.id = gm.id echipa
AND t.id = gm.id_grupa
AND g.id = gm.id_grupa
AND t.id = g.id_turneu
AND UPPER(t.num) LIKE UPPER(num_turneu);

FOR i IN v_regiuni.first..v_regiuni.last
LOOP
    dbms_output.put_line('REGIUNEA: ' || v_regiuni(i).num);
    dbms_output.put_line('-----');
    FOR j IN v_echipe.first..v_echipe.last
    LOOP
        dbms_output.put_line('ECHIPA: ' || v_echipe(j).num);
        dbms_output.put_line('-----');
        SELECT j.num BULK COLLECT
        INTO v_num
        FROM jucator j,
             echipa e,
             regiune r
        WHERE j.id echipa = e.id
        AND j.id_regiune = r.id
        AND e.num = v_echipe(j).num
        AND r.num = v_regiuni(i).num;
        IF v_num.count > 0 THEN
            FOR k IN v_num.first..v_num.last
            LOOP
                dbms_output.put_line(v_num(k));
            END LOOP;
        ELSE

```

```

        dbms_output.put_line('Nu exista!');
    END IF;
    dbms_output.put_line('-----');
    dbms_output.put_line('-----');
END LOOP;
END LOOP;
END;
/

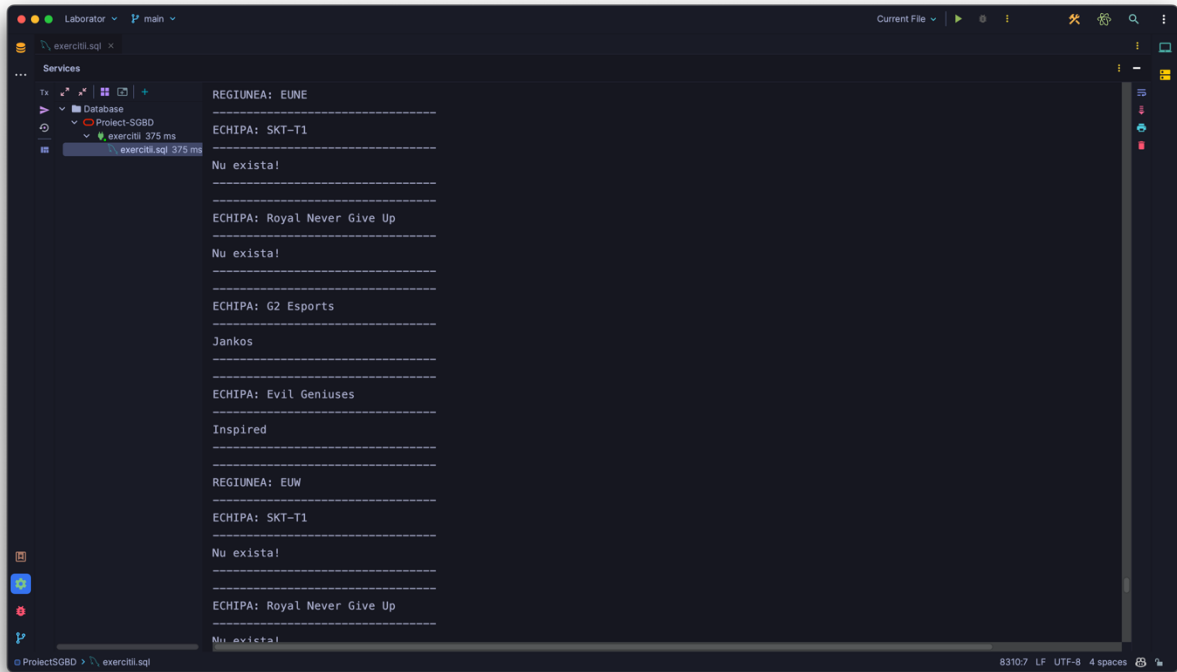
BEGIN
    ex6('MSI');
END;
/

```

The screenshot shows a SQL IDE window with a dark theme. The left sidebar displays a project tree with 'Project-SGBD' and 'exerciti1.sql'. The main editor area shows the execution output of the script. The output is formatted with dashed lines separating different sections. The sections include:

- REGIUNEA: KOR
- ECHIPA: SKT-T1
- Oner
- Faker
- Gumayusi
- Keria
- Zeus
- ECHIPA: Royal Never Give Up
- Bin
- Wei
- Xiaohu
- Gala
- Ming
- ECHIPA: G2 Esports
- Nu exista!
- ECHIPA: Evil Geniuses
- Impact
- REGIUNEA: EUNE

The status bar at the bottom indicates the file is 'exerciti1.sql' and the encoding is 'UTF-8'.



NU ESTE TOT OUTPUTUL

EX7

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent care să utilizeze 2 tipuri diferite de cursoare studiate, unul dintre acestea fiind cursor parametrizat. Apelați subprogramul.

```
-- Cerința 7:
-- Pentru jucatorii care apartin regiunii KOR, afisati campionii pe care ii joaca si folosesc ca
-- resursa mana.

CREATE OR REPLACE PROCEDURE ex7(atr_regiune_nume regiune.nume%TYPE, atr_resursa
campion.resursa%TYPE)
AS
  CURSOR
    regiuni(regiune_nume regiune.nume%TYPE) IS
    SELECT r.id
    FROM regiune r
```

```

WHERE UPPER(r.num) LIKE UPPER(atr_regiune_num);
CURSOR
    jucatori IS
    SELECT j.num, j.id, j.id_regiune
    FROM jucator j;
CURSOR
    campioni(jucator_id jucator.id%TYPE) IS
    SELECT c.num || ' ' || '->' || ' ' || c.resursa AS result
    FROM campion c,
         campion_pool cp
    WHERE c.id = cp.id_campion
        AND cp.id_jucator = jucator_id
        AND UPPER(c.resursa) LIKE UPPER(atr_resursa);
v_num_jucator jucator.num%TYPE;
v_id_jucator   jucator.id%TYPE;
v_id_regiune   regiune.id%TYPE;

BEGIN
FOR regiune IN regiuni(atr_regiune_num)
    LOOP
        dbms_output.put_line('Jucatorii din regiunea ' || atr_regiune_num || ' sunt: ');
        dbms_output.put_line('-----');
        OPEN jucatori;
        LOOP
            FETCH jucatori INTO v_num_jucator, v_id_jucator, v_id_regiune;
            EXIT WHEN jucatori%NOTFOUND;
            IF v_id_regiune = regiune.id THEN
                dbms_output.put_line('Jucatorul: ' || v_num_jucator);
                FOR campion IN campioni(v_id_jucator)
                    LOOP
                        dbms_output.put_line(campion.result);
                    END LOOP;
                dbms_output.put_line('-----');
            END IF;
        END LOOP;
    END LOOP;
END LOOP;

```

```
END;  
/  
  
BEGIN  
    ex7('KOR', 'mana');  
END;  
/
```





EX8

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip funcție care să utilizeze într-o singură comandă SQL 3 dintre tabelele definite.

Definiți minim 2 excepții. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

```
-- Cerința 8:
-- Pentru un jucator dat afisati numarul de campioni pe care acesta ii joaca,
-- daca acesta nu joaca pe lane-ul mid.

CREATE OR REPLACE FUNCTION ex8(atr_jucator_id jucator.id%TYPE)
RETURN number
IS
v_numar_campioni NUMBER(10);
TYPE tip_tabel IS TABLE OF campion_pool%ROWTYPE INDEX BY PLS_INTEGER;
```

```

v_tabel      tip_tabel;
TYPE tip_tabel2 IS TABLE OF jucator%ROWTYPE INDEX BY PLS_INTEGER;
v_tabel2      tip_tabel2;
v_num_jucator  jucator.nume%TYPE;

negative_id EXCEPTION;
no_data_found_1 EXCEPTION;
no_data_found_2 EXCEPTION;
no_player_found EXCEPTION;

BEGIN
  IF atr_jucator_id < 0 THEN
    RAISE negative_id;
  END IF;

  SELECT * BULK COLLECT
  INTO v_tabel2
  FROM jucator
  WHERE id = atr_jucator_id;

  IF SQL%NOTFOUND THEN
    RAISE no_player_found;
  END IF;

  SELECT * BULK COLLECT
  INTO v_tabel
  FROM campion_pool cp
  WHERE cp.id_jucator = atr_jucator_id;

  IF SQL%NOTFOUND THEN
    RAISE no_data_found_1;
  END IF;

  SELECT j.nume INTO v_num_jucator FROM jucator j WHERE j.id = atr_jucator_id;

  SELECT COUNT(c.id)

```



```

    INTO v_numar_campioni
    FROM campion c
        JOIN campion_pool cp ON c.id = cp.id_campion
        JOIN jucator j ON j.id = cp.id_jucator
    WHERE cp.id_jucator = atr_jucator_id
        AND j.lane NOT LIKE 'MID';

    IF v_numar_campioni = 0 THEN
        RAISE no_data_found_2;
    ELSE
        RETURN v_numar_campioni;
    END IF;

EXCEPTION
    WHEN negative_id THEN
        dbms_output.put_line('ID-ul nu poate fi negativ!');
        RETURN -1;
    WHEN no_data_found_1 THEN
        dbms_output.put_line('Nu exista date pentru acest jucator!');
        RETURN -1;
    WHEN no_data_found_2 THEN
        dbms_output.put_line('Jucatorul ' || v_ume_jucator || ' joaca pe lane-ul MID!');
        RETURN -1;
    WHEN no_player_found THEN
        dbms_output.put_line('Nu exista jucator cu acest ID!');
        RETURN -1;
    WHEN OTHERS THEN
        dbms_output.put_line('Cod eroare' || SQLCODE);
        dbms_output.put_line('Mesaj eroare' || SQLERRM);
        RETURN -1;
END;
/

DECLARE
    v_numar    NUMBER;
    v_id_jucator jucator.id%TYPE := &id_jucator;

```

```

BEGIN
    v_numar := ex8(v_id_jucator);
    IF v_numar > -1 THEN
        dbms_output.put_line('Jucatorul cu id-ul ' || v_id_jucator || ' joaca ' || v_numar || '
campioni');
    END IF;
END;
/

-- Apelare cu id-ul 1 -> 3 campioni
-- Apelare cu id-ul 4 -> 2 campioni
-- Apelare cu id-ul -1 -> ID-ul nu poate fi negativ!
-- Apelare cu id-ul 3 -> Jucatorul Faker joaca pe lane-ul MID!
-- Apelare cu id-ul 60 -> Nu exista jucator cu acest ID!

DELETE
FROM campion_pool
WHERE id_jucator = 6;

-- Apelare cu id-ul 6 -> Nu exista date pentru acest jucator!

```

APELARE CU ID 1

```

[2022-12-27 00:04:44] completed in 113 ms
Jucatorul cu id-ul 1 joaca 3 campioni

```

APELARE CU ID 4

```

[2022-12-27 00:07:46] completed in 117 ms
Jucatorul cu id-ul 4 joaca 2 campioni

```

APELARE CU ID -1

```
[2022-12-27 00:08:14] completed in 110 ms  
ID-ul nu poate fi negativ!
```

APELARE CU ID 3

```
[2022-12-27 00:08:39] completed in 112 ms  
Jucatorul Faker joaca pe lane-ul MID!
```

APELARE CU ID 60

```
[2022-12-27 00:09:10] completed in 112 ms  
Nu exista jucator cu acest ID!
```

APELARE CU ID 6

```
DELETE  
FROM campion_pool  
WHERE id_jucator = 6;
```

```
[2022-12-27 00:09:52] completed in 108 ms  
Nu exista date pentru acest jucator!
```

EX9

Formulați în limbaj natural o problemă pe care să o rezolvați folosind un subprogram stocat independent de tip procedură care să utilizeze într-o singură comandă SQL 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea, incluzând excepțiile NO_DATA_FOUND și TOO_MANY_ROWS. Apelați subprogramul astfel încât să evidențiați toate cazurile tratate.

```

-- Cerința 9:
-- Pentru o echipa al carei nume este dat, afisati echipele adverse din meciurile ce au loc pe
-- scena "Horn Stage",
-- prezentate de "Jordan Corby".

CREATE OR REPLACE PROCEDURE ex9(num_echipa echipa.nume%TYPE)
AS
    TYPE 44fffc_index_echipa IS TABLE OF echipa%ROWTYPE INDEX BY PLS_INTEGER;
    v_echipa      44fffc_index_echipa;
    TYPE 44fffc_index_echipa_adversa IS TABLE OF echipa.nume%TYPE INDEX BY PLS_INTEGER;
    v_echipa_adversa 44fffc_index_echipa_adversa;
    v_id_echipa      echipa.id%TYPE;

    no_data_found_1 EXCEPTION;
    no_data_found_2 EXCEPTION;
    too_many_rows EXCEPTION;

BEGIN
    SELECT * BULK COLLECT
    INTO v_echipa
    FROM echipa e
    WHERE e.nume = num_echipa;

    IF SQL%NOTFOUND THEN
        RAISE no_data_found_1;
    END IF;

    IF v_echipa.count > 1 THEN
        RAISE too_many_rows;
    END IF;

    SELECT e.id INTO v_id_echipa FROM echipa e WHERE e.nume = num_echipa;

    SELECT e.nume BULK COLLECT
    INTO v_echipa_adversa
    FROM echipa e

```

```

        JOIN meci_map mm ON e.id = mm.id echipa
        JOIN meci m ON m.id = mm.id meci
        JOIN meci_header mh ON mh.id meci = m.id
        JOIN scena s ON s.id = mh.id scena
        JOIN prezentator p ON p.id = mh.id prezentator
WHERE e.nume != nume echipa
    AND s.nume = 'Horn Stage'
    AND p.nume = 'Jordan Corby'
    AND v_id echipa IN (SELECT mm.id echipa FROM meci_map mm WHERE mm.id meci =
m.id);

IF v echipa adversa.count = 0 THEN
    RAISE no_data_found_2;
END IF;

dbms_output.put_line('Echipele adversare ale celor din ' || nume echipa || ' sunt: ');
FOR I IN 1 .. v echipa adversa.count
    LOOP
        dbms_output.put_line(v echipa adversa(i));
    END LOOP;

EXCEPTION
    WHEN no_data_found_1 THEN
        dbms_output.put_line('Nu exista date pentru aceasta echipa!');
    WHEN no_data_found_2 THEN
        dbms_output.put_line('Nu exista echipe adversare care sa indeplineasca conditiile!');
    WHEN too_many_rows THEN
        dbms_output.put_line('Exista mai multe echipe cu acest nume!');
    WHEN OTHERS THEN
        dbms_output.put_line('Cod eroare' || SQLCODE);
        dbms_output.put_line('Mesaj eroare' || SQLERRM);
END;
/

DECLARE
    nume echipa.nume%TYPE := 'SKT-T1';

```

```

BEGIN
    ex9(ume);
END;
/

-- Pentru ume = 'Canids Kalunga' -> PSG Talon Esports
-- Pentru ume = 'PSG Talon Esports' -> Canids Kalunga
-- Pentru ume = 'Fnatic' -> "Nu exista date pentru aceasta echipa!"
-- Pentru ume = 'Order' -> "Nu exista echipe adversare care sa indeplineasca conditiile!"

INSERT INTO echipa(id_regiune, ume, tag)
VALUES (2, 'SKT-T1', 'SKT_V2')
-- Pentru ume = 'SKT-T1' -> "Exista mai multe echipe cu acest ume!"

```

PENTRU NUME 'Canids Kalunga'

```

[2022-12-27 00:12:24] completed in 109 ms
Echipele adversare ale celor din Canids Kalunga sunt:
PSG Talon Esports

```

PENTRU NUME 'PSG Talon Esports'

```

[2022-12-27 00:12:42] completed in 111 ms
Echipele adversare ale celor din PSG Talon Esports sunt:
Canids Kalunga

```

PENTRU NUME 'Fnatic'

```

[2022-12-27 00:13:02] completed in 108 ms
Nu exista date pentru aceasta echipa!

```

PENTRU NUME 'Order'

```
[2022-12-27 00:13:18] completed in 111 ms  
Nu exista echipe adversare care sa indeplineasca conditiile!
```

PENTRU NUME 'SKT-T1'

```
INSERT INTO echipa(id_regiune, nume, tag)  
VALUES (2, 'SKT-T1', 'SKT_V2')
```

```
[2022-12-27 00:13:37] completed in 109 ms  
Exista mai multe echipe cu acest nume!
```

EX10

Definiți un *trigger* de tip LMD la nivel de comandă. Declanșați *trigger*-ul.

```
-- Cerința 10:  
-- Voi realiza un trigger care se va declansa atunci cand vreau sa modific tabela TURNEU cu  
-- conditiile:  
-- 1. Nu pot modifica tabela TURNEU intre orele 08:00-10:00  
-- 2. Nu pot modifica tabela in ziua de Craciun si Ziua de Anul Nou  
  
CREATE OR  
REPLACE TRIGGER ex10  
BEFORE  
INSERT OR UPDATE OR  
DELETE  
ON turneu  
DECLARE  
  
BEGIN  
IF ((TO_CHAR(SYSDATE, 'HH24') BETWEEN 8 AND 10) OR  
(TO_CHAR(SYSDATE, 'MMDD') = '1225' OR TO_CHAR(SYSDATE, 'MMDD') = '0101')) THEN
```

```

        RAISE_APPLICATION_ERROR(-20001, 'Nu poti modifica tabela in acest interval de timp
sau in aceasta zi!');
    END IF;
END;
/

INSERT INTO turneu (nume, oras, data)
VALUES ('TurneuName', 'Stockholm', TO_DATE('2022-05-10', 'YYYY-MM-DD'));

```

```

[2022-12-27 00:18:56] completed in 131 ms
ADMIN> INSERT INTO turneu (nume, oras, data)
        VALUES ('TurneuName', 'Stockholm', TO_DATE('2022-05-10', 'YYYY-MM-DD'))
[2022-12-27 00:19:01] [72000][20001]
[2022-12-27 00:19:01]   ORA-20001: Nu poti modifica tabela in acest interval de timp sau in aceasta zi!
[2022-12-27 00:19:01]   ORA-06512: at "ADMIN.EX10", line 6
[2022-12-27 00:19:01]   ORA-04088: error during execution of trigger 'ADMIN.EX10'
[2022-12-27 00:19:01] Position: 12

```

Pentru a testa trigger-ul am adaugat ca data in care rulez acest exemplu sa fie de asemenea interzisa.

EX11

Definiți un *trigger* de tip LMD la nivel de linie. Declanșați *trigger*-ul.

```

-- Cerința 11:
-- Voi realiza un trigger care se va declansa atunci cand vreau sa modific tabela meci cu
conditiile:
-- 1. Data meciului nu poate fi mai veche de 1 Aprilie 2011
-- (data in care a avut loc primul joc de League of Legends oficial).
-- 2. Data meciului nu poate fi in ziua de Craciun si Ziua de Anul Nou.

```

```

CREATE
OR
REPLACE TRIGGER ex11
BEFORE
INSERT OR UPDATE OR

```



```

DELETE
ON meci
FOR EACH ROW
DECLARE
BEGIN
    IF (TO_DATE(:new.data, 'YYYY-MM-DD') <= TO_DATE('2011-04-01', 'YYYY-MM-DD') OR
        (TO_CHAR(:new.data, 'MMDD') = '1225' OR TO_CHAR(:new.data, 'MMDD') = '0101'))
    THEN
        RAISE_APPLICATION_ERROR(-20001,
            'Probleme cu data introdusa,' ||
            ' data nu poate fi mai veche de 1 Aprilie 2011' ||
            ' sau in ziua de Craciun sau in Ziua de Anul Nou!');
    END IF;
END;

```

```

INSERT INTO meci (titlu, data)
VALUES ('MeciName', TO_DATE('2022-01-01', 'YYYY-MM-DD'));

```

```

INSERT INTO meci (titlu, data)
VALUES ('MeciName', TO_DATE('2010-10-10', 'YYYY-MM-DD'));

```

```

INSERT INTO meci (titlu, data)
VALUES ('MeciName', TO_DATE('2022-01-01', 'YYYY-MM-DD'));

```

```

INSERT INTO meci (titlu, data)
VALUES ('MeciName', TO_DATE('2010-10-10', 'YYYY-MM-DD'));

```

```

ADMIN> INSERT INTO meci (titlu, data)
        VALUES ('MeciName', TO_DATE('2022-01-01', 'YYYY-MM-DD'))
[2022-12-27 00:21:15] [72000][20001]
[2022-12-27 00:21:15] ORA-20001: Probleme cu data introdusa, data nu poate fi mai veche de 1 Aprilie 2011 sau in ziua de Craciun sau in Ziua de Anul Nou
[2022-12-27 00:21:15] ORA-06512: at "ADMIN.EX11", line 5
[2022-12-27 00:21:15] ORA-04088: error during execution of trigger 'ADMIN.EX11'
[2022-12-27 00:21:15] Position: 12

```

```

ADMIN> INSERT INTO meci (titlu, data)
        VALUES ('MeciName', TO_DATE('2022-01-01', 'YYYY-MM-DD'))
[2022-12-27 00:21:15] [72000][20001]
[2022-12-27 00:21:15] ORA-20001: Probleme cu data introdusa, data nu poate fi mai veche de 1 Aprilie 2011 sau in ziua de Craciun sau in Ziua de Anul Nou
[2022-12-27 00:21:15] ORA-06512: at "ADMIN.EX11", line 5
[2022-12-27 00:21:15] ORA-04088: error during execution of trigger 'ADMIN.EX11'
[2022-12-27 00:21:15] Position: 12

```

EX12

Definiți un *trigger* de tip LDD. Declanșați *trigger*-ul.

```
-- Cerința 12:
-- Voi realiza un trigger care se va declansa atunci cand se executa operatii LDD.
-- Acesta va stoca intr-o noua table numita istoric utilizatorul,
-- operatia executata si baza de date in care are loc operatia.

CREATE TABLE istoric
(
    id          NUMBER(10) GENERATED ALWAYS AS IDENTITY
              CONSTRAINT istoric_pk PRIMARY KEY,
    utilizator  VARCHAR2(50)
              CONSTRAINT utilizator_istoric NOT NULL,
    operatie    VARCHAR2(50)
              CONSTRAINT operatie_istoric NOT NULL,
    baza_de_date VARCHAR2(50)
              CONSTRAINT baza_de_date_istoric NOT NULL
);

SELECT *
FROM istoric;

CREATE
OR
REPLACE TRIGGER ex12
AFTER CREATE OR ALTER OR DROP
ON SCHEMA
BEGIN
    INSERT INTO istoric (utilizator, operatie, baza_de_date)
    VALUES (sys.login_user, sys.sysevent, sys.database_name);
END;
```

```
/

CREATE
  TABLE
    test
  (
    id  NUMBER(10),
    nume VARCHAR2(50)
  );

ALTER
  TABLE
    test
  ADD
    (prenume VARCHAR2(50));

ALTER TABLE
  test
  DROP COLUMN prenume;

DROP
  TABLE
    test;

SELECT *
FROM istoric;
```

	id	utilizator	operatie	baza_de_date
1	1	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
2	2	ADMIN	ALTER	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
3	3	ADMIN	ALTER	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
4	4	ADMIN	DROP	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
5	5	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
6	6	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
7	7	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
8	8	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
9	9	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
10	10	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
11	11	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
12	12	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
13	13	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
14	14	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
15	15	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
16	16	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
17	17	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
18	18	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
19	19	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
20	20	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
21	21	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE
22	22	ADMIN	CREATE	G4A53250AD7EE61_C5U2MT63QFB4Q3QE

EX13

Definiți un pachet care sa contina toate obiectele definite in cadrul proiectului.

```
CREATE OR REPLACE PACKAGE proiect_sgbd_hutan_mihai AS
  PROCEDURE ex6(ume_turneu turneu.ume%TYPE);
  PROCEDURE ex7(atr_regiune_ume regiune.ume%TYPE, atr_resursa
campion.resursa%TYPE);
  FUNCTION ex8(atr_jucator_id jucator.id%TYPE) RETURN NUMBER;
  PROCEDURE ex9(ume_echipa echipa.ume%TYPE);
END proiect_sgbd_hutan_mihai;
/
```

```

CREATE OR REPLACE PACKAGE BODY proiect_sgbd_hutan_mihai
AS
    -- Cerința 6:
    -- Pentru un turneu al carui nume este dat, pentru fiecare regiune, afisati toate echipele si
    jucatorii ce apartin
    -- acelei regiuni sau mesajul 'Nu exista!'.

    PROCEDURE ex6(nume_turneu turneu.nume%TYPE)
    AS
        TYPE tabl_idx IS TABLE OF regiune%ROWTYPE INDEX BY PLS_INTEGER;
        v_regiuni tabl_idx;
        TYPE tip_lista_nested IS TABLE OF echipa%ROWTYPE;
        v_echipe tip_lista_nested := tip_lista_nested();
        TYPE tabl_index IS TABLE OF VARCHAR2(200) INDEX BY PLS_INTEGER;
        v_nume    tabl_index;
        v_numar    NUMBER(5);
    BEGIN
        SELECT * BULK COLLECT INTO v_regiuni FROM regiune;

        SELECT COUNT(*)
        INTO v_numar
        FROM echipa e,
             turneu t,
             grupa_map gm,
             grupa g
        WHERE e.id = gm.id echipa
              AND t.id = gm.id_grupa
              AND g.id = gm.id_grupa
              AND t.id = g.id_turneu
              AND UPPER(t.nume) LIKE UPPER(nume_turneu);

        v_echipe.extend(v_numar + 1);

        SELECT e.id, e.nume, e.tag, e.id_regiune BULK COLLECT
        INTO v_echipe
    
```

```

FROM echipa e,
     turneu t,
     grupa_map gm,
     grupa g
WHERE e.id = gm.id echipa
     AND t.id = gm.id grupa
     AND g.id = gm.id grupa
     AND t.id = g.id turneu
     AND UPPER(t.ume) LIKE UPPER(ume_turneu);

FOR i IN v_regiuni.first..v_regiuni.last
  LOOP
    dbms_output.put_line('REGIUNEA: ' || v_regiuni(i).ume);
    dbms_output.put_line('-----');
    FOR j IN v_echipe.first..v_echipe.last
      LOOP
        dbms_output.put_line('ECHIPA: ' || v_echipe(j).ume);
        dbms_output.put_line('-----');
        SELECT j.ume BULK COLLECT
        INTO v_ume
        FROM jucator j,
             echipa e,
             regiune r
        WHERE j.id echipa = e.id
             AND j.id regiune = r.id
             AND e.ume = v_echipe(j).ume
             AND r.ume = v_regiuni(i).ume;
        IF v_ume.count > 0 THEN
          FOR k IN v_ume.first..v_ume.last
            LOOP
              dbms_output.put_line(v_ume(k));
            END LOOP;
          ELSE
            dbms_output.put_line('Nu exista!');
          END IF;
        dbms_output.put_line('-----');
      END LOOP;
    END LOOP;
  END LOOP;

```

```

        dbms_output.put_line('-----');
    END LOOP;
END LOOP;
END ex6;

-- Cerința 7:
-- Pentru jucatorii care apartin regiunii KOR, afisati campionii pe care ii joaca si folosesc ca
resursa mana.

PROCEDURE ex7(atr_regiune_nume regiune.nume%TYPE, atr_resursa
campion.resursa%TYPE)
AS
    CURSOR
        regiuni(regiune_nume regiune.nume%TYPE) IS
        SELECT r.id
        FROM regiune r
        WHERE UPPER(r.nume) LIKE UPPER(atr_regiune_nume);
    CURSOR
        jucatori IS
        SELECT j.nume, j.id, j.id_regiune
        FROM jucator j;
    CURSOR
        campioni(jucator_id jucator.id%TYPE) IS
        SELECT c.nume || ' ' || '->' || ' ' || c.resursa AS result
        FROM campion c,
             campion_pool cp
        WHERE c.id = cp.id_campion
              AND cp.id_jucator = jucator_id
              AND UPPER(c.resursa) LIKE UPPER(atr_resursa);
    v_nume_jucator jucator.nume%TYPE;
    v_id_jucator   jucator.id%TYPE;
    v_id_regiune   regiune.id%TYPE;

BEGIN
    FOR regiune IN regiuni(atr_regiune_nume)
    LOOP

```

```

dbms_output.put_line('Jucatorii din regiunea ' || atr_regiune_ume || ' sunt: ');
dbms_output.put_line('-----');
OPEN jucatori;
LOOP
    FETCH jucatori INTO v_num_e_jucator, v_id_jucator, v_id_regiune;
    EXIT WHEN jucatori%NOTFOUND;
    IF v_id_regiune = regiune.id THEN
        dbms_output.put_line('Jucatorul: ' || v_num_e_jucator);
        FOR campion IN campioni(v_id_jucator)
            LOOP
                dbms_output.put_line(campion.result);
            END LOOP;
        dbms_output.put_line('-----');
    END IF;
END LOOP;
END LOOP;
END ex7;

```

-- Cerința 8:

*-- Pentru un jucator dat afisati numarul de campioni pe care acesta ii joaca,
-- daca acesta nu joaca pe lane-ul mid.*

```

FUNCTION ex8(atr_jucator_id jucator.id%TYPE)
    RETURN number
IS
    v_numar_campioni NUMBER(10);
    TYPE tip_tabel IS TABLE OF campion_pool%ROWTYPE INDEX BY PLS_INTEGER;
    v_tabel      tip_tabel;
    TYPE tip_tabel2 IS TABLE OF jucator%ROWTYPE INDEX BY PLS_INTEGER;
    v_tabel2     tip_tabel2;
    v_num_e_jucator jucator.num_e%TYPE;

    negative_id EXCEPTION;
    no_data_found_1 EXCEPTION;
    no_data_found_2 EXCEPTION;
    no_player_found EXCEPTION;

```



```

BEGIN
  IF atr_jucator_id < 0 THEN
    RAISE negative_id;
  END IF;

  SELECT * BULK COLLECT
  INTO v_tabel2
  FROM jucator
  WHERE id = atr_jucator_id;

  IF SQL%NOTFOUND THEN
    RAISE no_player_found;
  END IF;

  SELECT * BULK COLLECT
  INTO v_tabel
  FROM campion_pool cp
  WHERE cp.id_jucator = atr_jucator_id;

  IF SQL%NOTFOUND THEN
    RAISE no_data_found_1;
  END IF;

  SELECT j.num INTO v_num_jucator FROM jucator j WHERE j.id = atr_jucator_id;

  SELECT COUNT(c.id)
  INTO v_numar_campioni
  FROM campion c
    JOIN campion_pool cp ON c.id = cp.id_campion
    JOIN jucator j ON j.id = cp.id_jucator
  WHERE cp.id_jucator = atr_jucator_id
    AND j.lane NOT LIKE 'MID';

  IF v_numar_campioni = 0 THEN
    RAISE no_data_found_2;

```

```

ELSE
    RETURN v_numar_campioni;
END IF;

EXCEPTION
    WHEN negative_id THEN
        dbms_output.put_line('ID-ul nu poate fi negativ!');
        RETURN -1;
    WHEN no_data_found_1 THEN
        dbms_output.put_line('Nu exista date pentru acest jucator!');
        RETURN -1;
    WHEN no_data_found_2 THEN
        dbms_output.put_line('Jucatorul ' || v_ume_jucator || ' joaca pe lane-ul MID!');
        RETURN -1;
    WHEN no_player_found THEN
        dbms_output.put_line('Nu exista jucator cu acest ID!');
        RETURN -1;
    WHEN OTHERS THEN
        dbms_output.put_line('Cod eroare' || SQLCODE);
        dbms_output.put_line('Mesaj eroare' || SQLERRM);
        RETURN -1;
END ex8;

-- Cerința 9:
-- Pentru o echipa al carei nume este dat, afisati echipele adverse din meciurile ce au loc pe
scena "Horn Stage",
-- prezentate de "Jordan Corby".

PROCEDURE ex9(nume echipa.echipa.nume%TYPE)
AS
    TYPE tabel_index echipa IS TABLE OF echipa%ROWTYPE INDEX BY PLS_INTEGER;
    v_echipa      tabel_index echipa;
    TYPE tabel_index echipa_adversa IS TABLE OF echipa.nume%TYPE INDEX BY
PLS_INTEGER;
    v_echipa_adversa tabel_index echipa_adversa;
    v_id_echipa      echipa.id%TYPE;

```

```

no_data_found_1 EXCEPTION;
no_data_found_2 EXCEPTION;
too_many_rows EXCEPTION;

BEGIN
  SELECT * BULK COLLECT
  INTO v_echipa
  FROM echipa e
  WHERE e.numa = numa_echipa;

  IF SQL%NOTFOUND THEN
    RAISE no_data_found_1;
  END IF;

  IF v_echipa.count > 1 THEN
    RAISE too_many_rows;
  END IF;

  SELECT e.id INTO v_id_echipa FROM echipa e WHERE e.numa = numa_echipa;

  SELECT e.numa BULK COLLECT
  INTO v_echipa_adversa
  FROM echipa e
    JOIN meci_map mm ON e.id = mm.id_echipa
    JOIN meci m ON m.id = mm.id_meci
    JOIN meci_header mh ON mh.id_meci = m.id
    JOIN scena s ON s.id = mh.id_scena
    JOIN prezentator p ON p.id = mh.id_prezentator
  WHERE e.numa != numa_echipa
    AND s.numa = 'Horn Stage'
    AND p.numa = 'Jordan Corby'
    AND v_id_echipa IN (SELECT mm.id_echipa FROM meci_map mm WHERE mm.id_meci =
m.id);

  IF v_echipa_adversa.count = 0 THEN

```

```

        RAISE no_data_found_2;
    END IF;

    dbms_output.put_line('Echipele adversare ale celor din ' || nume echipa || ' sunt: ');
    FOR i IN 1 .. v echipa_adversa.count
        LOOP
            dbms_output.put_line(v echipa_adversa(i));
        END LOOP;

EXCEPTION
    WHEN no_data_found_1 THEN
        dbms_output.put_line('Nu exista date pentru aceasta echipa!');
    WHEN no_data_found_2 THEN
        dbms_output.put_line('Nu exista echipe adversare care sa indeplineasca conditiile!');
    WHEN too_many_rows THEN
        dbms_output.put_line('Exista mai multe echipe cu acest nume!');
    WHEN OTHERS THEN
        dbms_output.put_line('Cod eroare' || SQLCODE);
        dbms_output.put_line('Mesaj eroare' || SQLERRM);
    END ex9;
END proiect_sgbd_hutan_mihai;
/

-- TESTARE PACKAGE
BEGIN
    proiect_sgbd_hutan_mihai.ex6('MSI');
    proiect_sgbd_hutan_mihai.ex7('KOR', 'mana');

    -- Apelare cu id-ul 1 -> 3 campioni
    dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(1));

    -- Apelare cu id-ul 4 -> 2 campioni
    dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(4));

    -- Apelare cu id-ul -1 -> ID-ul nu poate fi negativ!
    dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(-1));

```

```

-- Apelare cu id-ul 3 -> Jucatorul Faker joaca pe lane-ul MID!
dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(3));

-- Apelare cu id-ul 60 -> Nu exista jucator cu acest ID!
dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(60));
END;
/

DELETE
FROM campion_pool
WHERE id_jucator = 6;

BEGIN
-- Apelare cu id-ul 6 -> Nu exista date pentru acest jucator!
dbms_output.put_line('Rezultat ex8: ' || proiect_sgbd_hutan_mihai.ex8(6));
END;
/

BEGIN
-- Pentru nume = 'Canids Kalunga' -> PSG Talon Esports
proiect_sgbd_hutan_mihai.ex9('Canids Kalunga');
-- Pentru nume = 'PSG Talon Esports' -> Canids Kalunga
proiect_sgbd_hutan_mihai.ex9('PSG Talon Esports');
-- Pentru nume = 'Fnatic' -> "Nu exista date pentru aceasta echipa!"
proiect_sgbd_hutan_mihai.ex9('Fnatic');
-- Pentru nume = 'Order' -> "Nu exista echipe adversare care sa indeplineasca conditiile!"
proiect_sgbd_hutan_mihai.ex9('Order');
END;
/

INSERT INTO echipa(id_regiune, nume, tag)
VALUES (2, 'SKT-T1', 'SKT_V2');
BEGIN
-- Pentru nume = 'SKT-T1' -> "Exista mai multe echipe cu acest nume!"
proiect_sgbd_hutan_mihai.ex9('SKT-T1');

```

END;

/

LA RULAREA TESTELOR PENTRU PACKAGE VOM OBTINE ACELEASI REZULTATE CA SI IN CERINTELE IMPLEMENTATE IN AFARA PACKAGE-ULUI DEOARECE AM TESTAT ACELEASI CAZURI.