

**Gestiunea unei competiții Esports de Rocket League (RLCS – „Rocket League Championship Series”)**

Florea Mădălin-Alexandru

Seria 24, Grupa 243

1. **Prezentarea bazei de date**

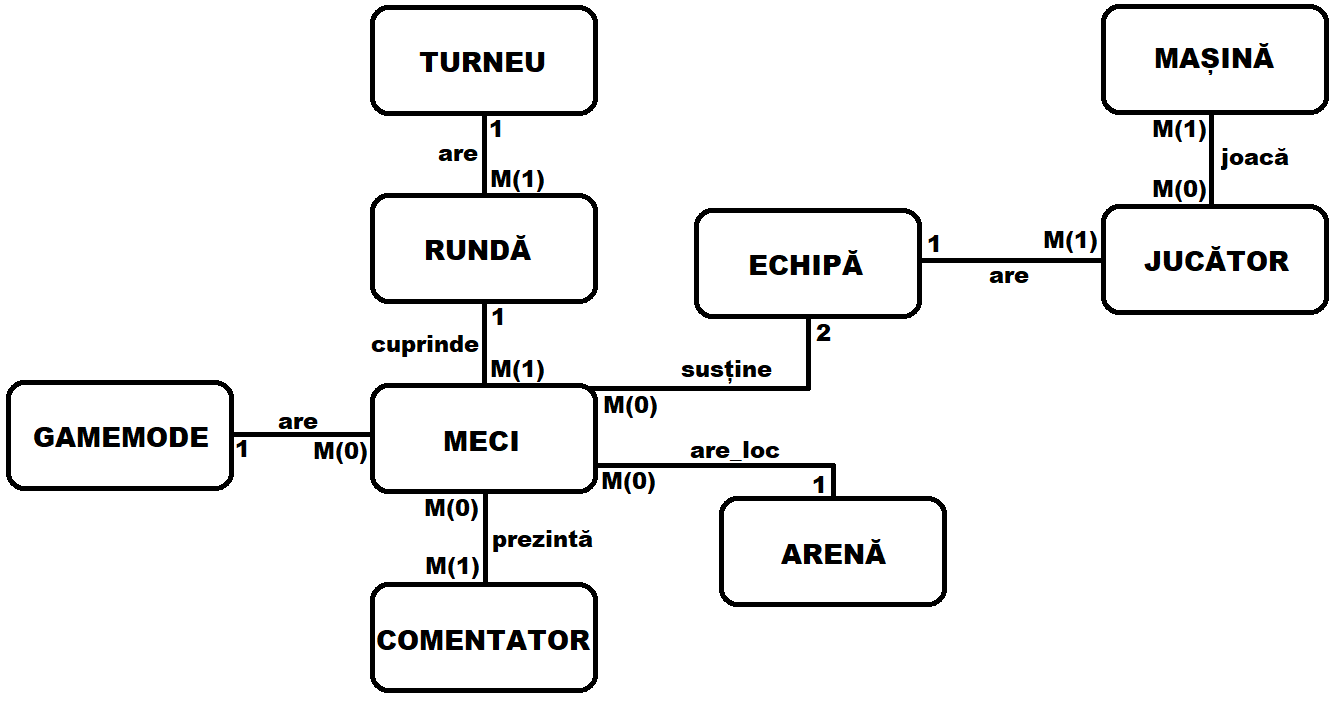
Rocket League Championship Series (RLCS) este o serie de turnee anuale de Esports organizate pentru jocul Rocket League. Ca o scurtă prezentare, în acest joc se desfășoară meciuri de fotbal (Standard, Doubles), baschet (Hoops), hochei (Snow Day) care, spre deosebire de cele din realitate cu care suntem obișnuiți, dispun de o fizică diferită și au mașini în loc de sportivi, sau moduri speciale precum Rumble, Dropshot.

Baza de date va conține informații despre turneele din această serie, echipele participante, jucătorii acestora și mașinile folosite, precum și meciurile jucate cu toate detaliile acestora (comentatori, arenă, gamemode etc.).

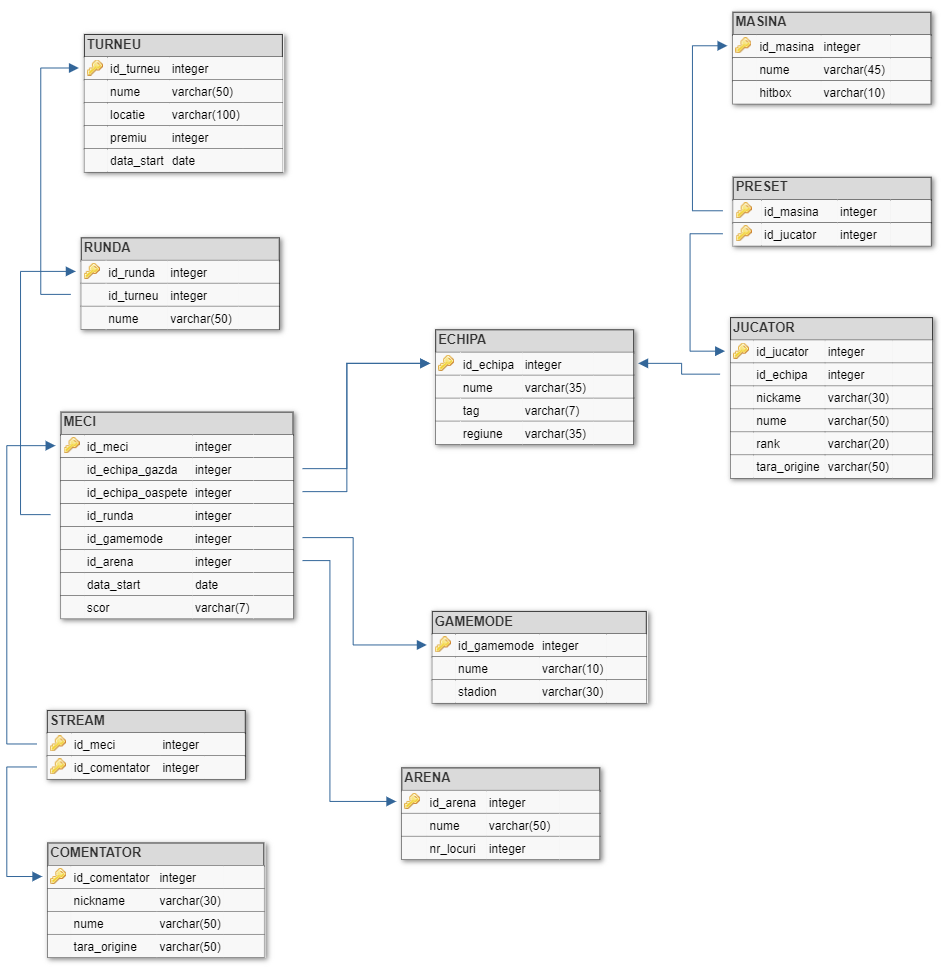
Această bază de date va fi creată cu scopul de a păstra informațiile turneelor de Rocket League, facilitând ulterior generarea și analizarea de statistici din cadrul acestor evenimente.

Anual se desfășoară numeroase turnee, despre care cunoaștem numele, locația și data de începere. Un turneu este format din una sau mai multe runde, fiecare cuprinzând unul sau mai multe meciuri. Acestea se desfășoară între două echipe despre care cunoaștem numele, tag-ul, jucătorii și regiunea din care provin. Despre jucători mai cunoaștem numele, nickname-ul, rank-ul și naționalitatea, iar pentru fiecare mașină vom ști numele și hitbox-ul (tipul de suprafață / formă a mașinii - are importanță în momentul lovirii mingii). În cele din urmă, vom avea informații legate de meciuri – prezentate de unul sau mai mulți comentatori, desfășurate pe o anumită arenă și într-un anumit mod de joc, dar și titlul sau scorul final.

1. **Diagrama entitate-relație (ERD)**

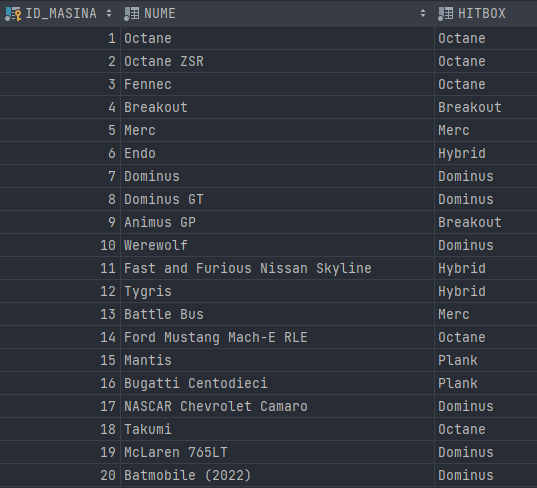


1. **Diagrama conceptuală**

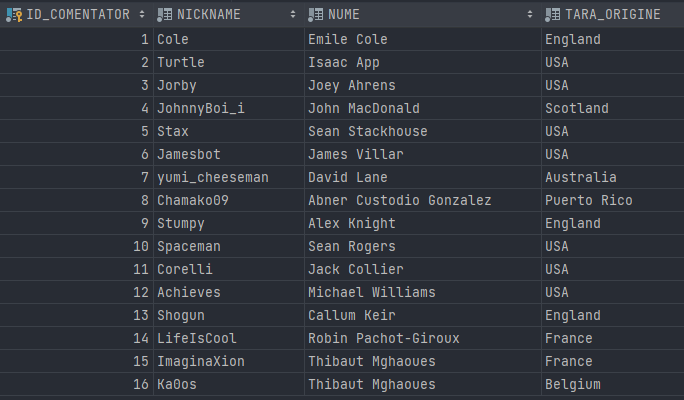


**4 & 5. Crearea tabelelor și inserarea de date coerente**

*-- Crearea tabelului MASINA  
  
CREATE TABLE* MASINA(  
 id\_masina *number*(2) *CONSTRAINT* masina\_pk *PRIMARY KEY*,  
 nume *varchar2*(45) *NOT NULL*,  
 hitbox *varchar2*(10) *NOT NULL*);  
  
*CREATE SEQUENCE* masina\_seq;  
  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Octane', 'Octane');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Octane ZSR', 'Octane');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Fennec', 'Octane');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Breakout', 'Breakout');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Merc', 'Merc');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Endo', 'Hybrid');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Dominus', 'Dominus');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Dominus GT', 'Dominus');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Animus GP', 'Breakout');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Werewolf', 'Dominus');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Fast and Furious Nissan Skyline', 'Hybrid');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Tygris', 'Hybrid');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Battle Bus', 'Merc');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Ford Mustang Mach-E RLE', 'Octane');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Mantis', 'Plank');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Bugatti Centodieci', 'Plank');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'NASCAR Chevrolet Camaro', 'Dominus');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Takumi', 'Octane');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'McLaren 765LT', 'Dominus');  
*INSERT INTO* MASINA *VALUES* (masina\_seq.NEXTVAL, 'Batmobile (2022)', 'Dominus');  
  
*SELECT* \* *FROM* MASINA;



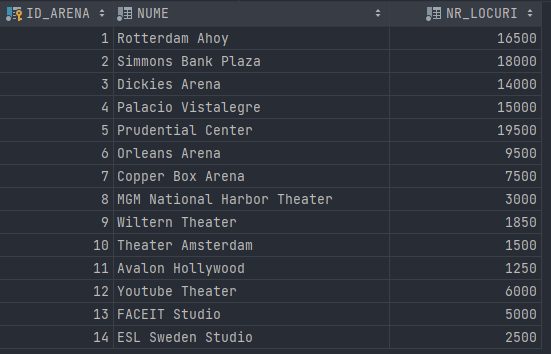
*-- Crearea tabelului COMENTATOR  
  
CREATE TABLE* COMENTATOR(  
 id\_comentator *number*(3) *CONSTRAINT* comentator\_pk *PRIMARY KEY*,  
 nickname *varchar2*(30) *NOT NULL*,  
 nume *varchar2*(50) *NOT NULL*,  
 tara\_origine *varchar2*(50) *NOT NULL*);  
  
*CREATE SEQUENCE* comentator\_seq;  
  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Cole', 'Emile Cole', 'England');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Turtle', 'Isaac App', 'USA');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Jorby', 'Joey Ahrens', 'USA');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'JohnnyBoi\_i', 'John MacDonald', 'Scotland');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Stax', 'Sean Stackhouse', 'USA');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Jamesbot', 'James Villar', 'USA');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'yumi\_cheeseman', 'David Lane', 'Australia');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Chamako09', 'Abner Custodio Gonzalez', 'Puerto Rico');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Stumpy', 'Alex Knight', 'England');   
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Spaceman', 'Sean Rogers', 'USA');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Corelli', 'Jack Collier', 'USA');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Achieves', 'Michael Williams', 'USA');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Shogun', 'Callum Keir', 'England');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'LifeIsCool', 'Robin Pachot-Giroux', 'France');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'ImaginaXion', 'Thibaut Mghaoues', 'France');  
*INSERT INTO* COMENTATOR *VALUES* (comentator\_seq.NEXTVAL, 'Ka0os', 'Thibaut Mghaoues', 'Belgium');  
  
*SELECT* \* *FROM* COMENTATOR;



*-- Crearea tabelului GAMEMODE  
  
CREATE TABLE* GAMEMODE(  
 id\_gamemode *number*(3) *CONSTRAINT* gamemode\_pk *PRIMARY KEY*,  
 nume *varchar2*(10) *NOT NULL*,  
 stadion *varchar2*(30) *NOT NULL*);  
   
*CREATE SEQUENCE* gamemode\_seq;  
  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Dropshot', 'Core 707');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Hoops', 'Dunk House');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Hoops', 'The Block');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Snow Day', 'Utopia Coliseum (Snowy)');   
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Snow Day', 'Beckwith Park (Snowy)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Rumble', 'Forbidden Temple (Fire & Ice)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Rumble', 'Tokyo Underpass');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Rumble', 'Double Goal');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Standard', 'Champions Field');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Standard', 'Neo Tokyo (Comic)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Standard', 'Rivals Arena');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Standard', 'DFH Stadium (Circuit)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Doubles', 'Mannfield (Night)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Doubles', 'Beckwith Park (Midnight)');  
*INSERT INTO* GAMEMODE *VALUES* (gamemode\_seq.NEXTVAL, 'Doubles', 'AquaDome');  
  
*SELECT* \* *FROM* GAMEMODE;



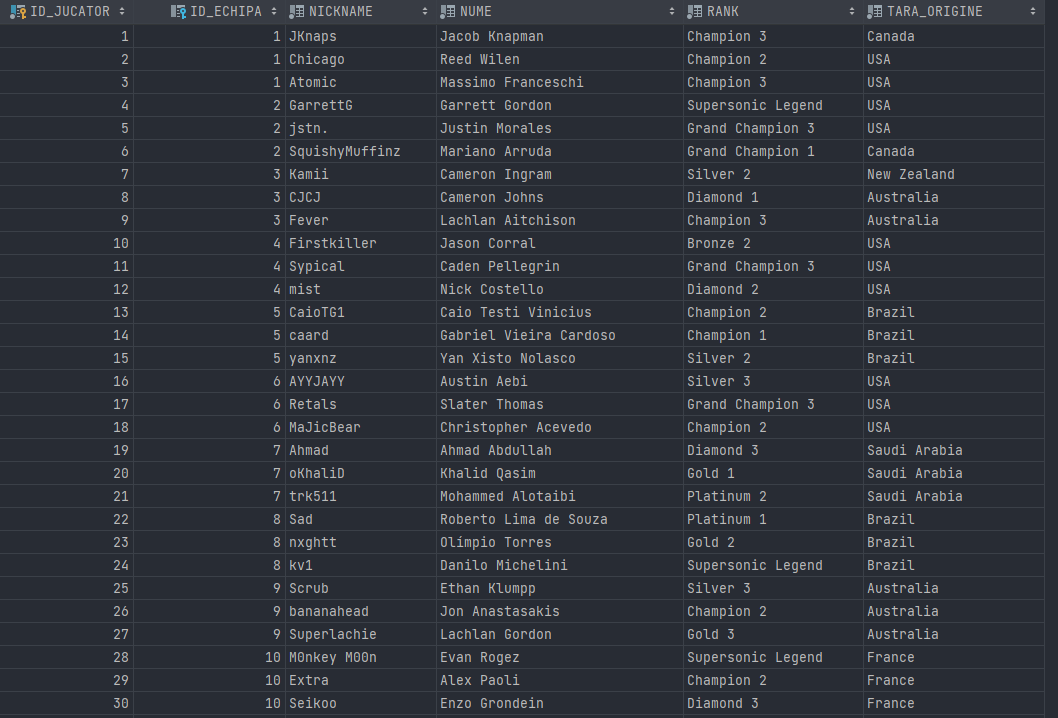
*-- Crearea tabelului ARENA  
   
CREATE TABLE* ARENA(  
 id\_arena *number*(3) *CONSTRAINT* arena\_pk *PRIMARY KEY*,  
 nume *varchar2*(40) *NOT NULL*,  
 nr\_locuri *number*(6) *NOT NULL*);  
   
*CREATE SEQUENCE* arena\_seq;  
  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Rotterdam Ahoy', 16500);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Simmons Bank Plaza', 18000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Dickies Arena', 14000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Palacio Vistalegre', 15000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Prudential Center', 19500);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Orleans Arena', 9500);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Copper Box Arena', 7500);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'MGM National Harbor Theater', 3000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Wiltern Theater', 1850);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Theater Amsterdam', 1500);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Avalon Hollywood', 1250);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'Youtube Theater', 6000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'FACEIT Studio', 5000);  
*INSERT INTO* ARENA *VALUES* (arena\_seq.NEXTVAL, 'ESL Sweden Studio', 2500);  
  
*SELECT* \* *FROM* ARENA;

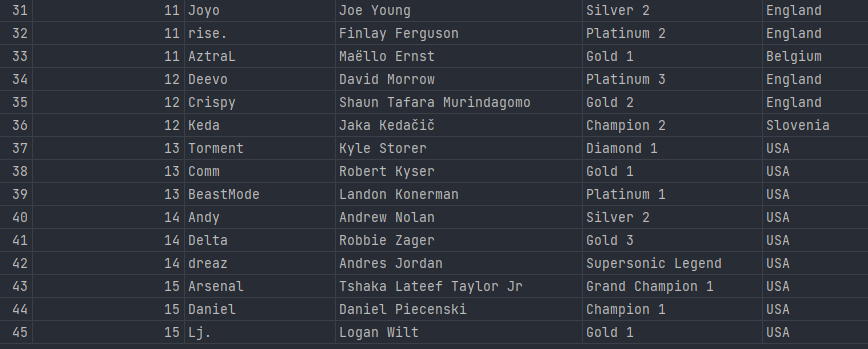


*-- Crearea tabelului ECHIPA  
  
CREATE TABLE* ECHIPA(  
 id\_echipa *number*(3) *CONSTRAINT* echipa\_pk *PRIMARY KEY*,  
 nume *varchar2*(35) *NOT NULL*,  
 *tag varchar2*(7) *NOT NULL*,  
 regiune *varchar2*(35) *NOT NULL*);  
  
*CREATE SEQUENCE* echipa\_seq;  
  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'G2 Esports', '[G2]', 'Europe');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'NRG Esports', '[NRG]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Renegades', '[RNG]', 'Australia');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'FaZe Clan', '[FAZE]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'FURIA Esports', '[FUR]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'OpTic Gaming', '[OG]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Team Falcons', '[TFA]', 'Middle East / North Africa');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Team Secret', '[TS]', 'South America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Kansas City Pioneers', '[KCP]', 'Oceania');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Team BDS', '[BDS]', 'Europe');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Moist Esports', '[MST]', 'Europe');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Endpoint CeX', '[END]', 'Europe');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Version1', '[V1]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Dignitas', '[DIG]', 'North America');  
*INSERT INTO* ECHIPA *VALUES* (echipa\_seq.NEXTVAL, 'Spacestation Gaming', '[SSG]', 'North America');  
  
*SELECT* \* *FROM* ECHIPA;

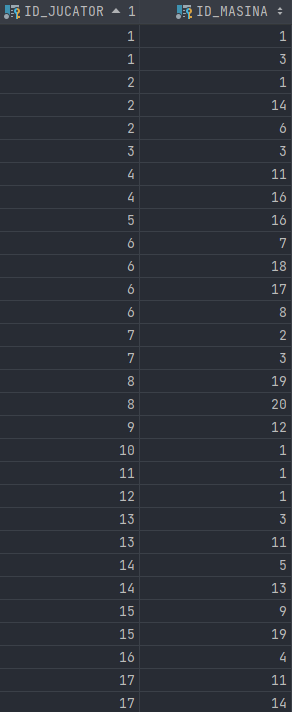
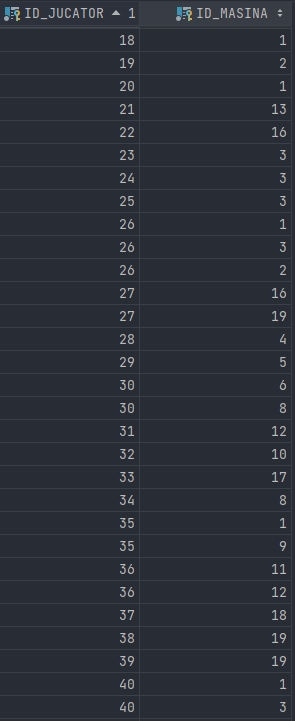


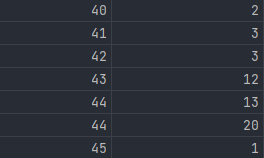
*-- Crearea tabelului JUCATOR  
  
CREATE TABLE* JUCATOR(  
 id\_jucator *number*(3) *CONSTRAINT* jucator\_pk *PRIMARY KEY*,  
 id\_echipa *number*(3), *CONSTRAINT* jucator\_echipa\_fk *FOREIGN KEY*(id\_echipa) *REFERENCES* ECHIPA(id\_echipa),  
 nickname *varchar2*(30) *NOT NULL*,  
 nume *varchar2*(50) *NOT NULL*,  
 rank *varchar2*(20) *NOT NULL*,  
 tara\_origine *varchar2*(50) *NOT NULL*);  
  
*CREATE SEQUENCE* jucator\_seq;  
  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 1, 'JKnaps', 'Jacob Knapman', 'Champion 3', 'Canada');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 1, 'Chicago', 'Reed Wilen', 'Champion 2', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 1, 'Atomic', 'Massimo Franceschi', 'Champion 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 2, 'GarrettG', 'Garrett Gordon', 'Supersonic Legend', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 2, 'jstn.', 'Justin Morales', 'Grand Champion 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 2, 'SquishyMuffinz', 'Mariano Arruda', 'Grand Champion 1', 'Canada');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 3, 'Kamii', 'Cameron Ingram', 'Silver 2', 'New Zealand');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 3, 'CJCJ', 'Cameron Johns', 'Diamond 1', 'Australia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 3, 'Fever', 'Lachlan Aitchison', 'Champion 3', 'Australia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 4, 'Firstkiller', 'Jason Corral', 'Bronze 2', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 4, 'Sypical', 'Caden Pellegrin', 'Grand Champion 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 4, 'mist', 'Nick Costello', 'Diamond 2', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 5, 'CaioTG1', 'Caio Testi Vinicius', 'Champion 2', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 5, 'caard', 'Gabriel Vieira Cardoso', 'Champion 1', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 5, 'yanxnz', 'Yan Xisto Nolasco', 'Silver 2', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 6, 'AYYJAYY', 'Austin Aebi', 'Silver 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 6, 'Retals', 'Slater Thomas', 'Grand Champion 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 6, 'MaJicBear', 'Christopher Acevedo', 'Champion 2', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 7, 'Ahmad', 'Ahmad Abdullah', 'Diamond 3', 'Saudi Arabia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 7, 'oKhaliD', 'Khalid Qasim', 'Gold 1', 'Saudi Arabia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 7, 'trk511', 'Mohammed Alotaibi', 'Platinum 2', 'Saudi Arabia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 8, 'Sad', 'Roberto Lima de Souza', 'Platinum 1', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 8, 'nxghtt', 'Olímpio Torres', 'Gold 2', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 8, 'kv1', 'Danilo Michelini', 'Supersonic Legend', 'Brazil');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 9, 'Scrub', 'Ethan Klumpp', 'Silver 3', 'Australia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 9, 'bananahead', 'Jon Anastasakis', 'Champion 2', 'Australia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 9, 'Superlachie', 'Lachlan Gordon', 'Gold 3', 'Australia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 10, 'M0nkey M00n', 'Evan Rogez', 'Supersonic Legend', 'France');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 10, 'Extra', 'Alex Paoli', 'Champion 2', 'France');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 10, 'Seikoo', 'Enzo Grondein', 'Diamond 3', 'France');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 11, 'Joyo', 'Joe Young', 'Silver 2', 'England');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 11, 'rise.', 'Finlay Ferguson', 'Platinum 2', 'England');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 11, 'AztraL', 'Maëllo Ernst', 'Gold 1', 'Belgium');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 12, 'Deevo', 'David Morrow', 'Platinum 3', 'England');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 12, 'Crispy', 'Shaun Tafara Murindagomo', 'Gold 2', 'England');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 12, 'Keda', 'Jaka Kedačič', 'Champion 2', 'Slovenia');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 13, 'Torment', 'Kyle Storer', 'Diamond 1', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 13, 'Comm', 'Robert Kyser', 'Gold 1', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 13, 'BeastMode', 'Landon Konerman', 'Platinum 1', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 14, 'Andy', 'Andrew Nolan', 'Silver 2', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 14, 'Delta', 'Robbie Zager', 'Gold 3', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 14, 'dreaz', 'Andres Jordan', 'Supersonic Legend', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 15, 'Arsenal', 'Tshaka Lateef Taylor Jr', 'Grand Champion 1', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 15, 'Daniel', 'Daniel Piecenski', 'Champion 1', 'USA');  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 15, 'Lj.', 'Logan Wilt', 'Gold 1', 'USA');  
  
*SELECT* \* *FROM* JUCATOR;



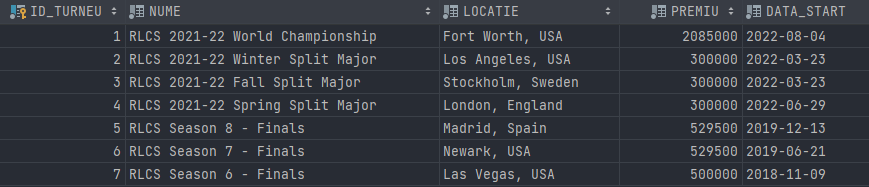


*-- Crearea tabelului PRESET  
  
CREATE TABLE* PRESET(  
 id\_jucator *NUMBER*(3) *NOT NULL*, *CONSTRAINT* id\_jucator\_fK *FOREIGN KEY* (id\_jucator) *REFERENCES* JUCATOR(id\_jucator),  
 id\_masina *NUMBER*(2) *NOT NULL*, *CONSTRAINT* id\_masina\_fK *FOREIGN KEY* (id\_masina) *REFERENCES* MASINA(id\_masina),  
 *CONSTRAINT* preset\_pk *PRIMARY KEY* (id\_masina, id\_jucator)  
);  
  
*INSERT INTO* PRESET *VALUES* (1, 1);  
*INSERT INTO* PRESET *VALUES* (1, 3);  
*INSERT INTO* PRESET *VALUES* (2, 1);  
*INSERT INTO* PRESET *VALUES* (2, 6);  
*INSERT INTO* PRESET *VALUES* (2, 14);  
*INSERT INTO* PRESET *VALUES* (3, 3);  
*INSERT INTO* PRESET *VALUES* (4, 11);  
*INSERT INTO* PRESET *VALUES* (4, 16);  
*INSERT INTO* PRESET *VALUES* (5, 16);  
*INSERT INTO* PRESET *VALUES* (6, 18);  
*INSERT INTO* PRESET *VALUES* (6, 17);  
*INSERT INTO* PRESET *VALUES* (6, 7);  
*INSERT INTO* PRESET *VALUES* (6, 8);  
*INSERT INTO* PRESET *VALUES* (7, 2);  
*INSERT INTO* PRESET *VALUES* (7, 3);  
*INSERT INTO* PRESET *VALUES* (8, 20);  
*INSERT INTO* PRESET *VALUES* (8, 19);  
*INSERT INTO* PRESET *VALUES* (9, 12);  
*INSERT INTO* PRESET *VALUES* (10, 1);  
*INSERT INTO* PRESET *VALUES* (11, 1);  
*INSERT INTO* PRESET *VALUES* (12, 1);  
*INSERT INTO* PRESET *VALUES* (13, 3);  
*INSERT INTO* PRESET *VALUES* (13, 11);  
*INSERT INTO* PRESET *VALUES* (14, 5);  
*INSERT INTO* PRESET *VALUES* (14, 13);  
*INSERT INTO* PRESET *VALUES* (15, 19);  
*INSERT INTO* PRESET *VALUES* (15, 9);  
*INSERT INTO* PRESET *VALUES* (16, 4);  
*INSERT INTO* PRESET *VALUES* (17, 14);  
*INSERT INTO* PRESET *VALUES* (17, 11);  
*INSERT INTO* PRESET *VALUES* (18, 1);  
*INSERT INTO* PRESET *VALUES* (19, 2);  
*INSERT INTO* PRESET *VALUES* (20, 1);  
*INSERT INTO* PRESET *VALUES* (21, 13);  
*INSERT INTO* PRESET *VALUES* (22, 16);  
*INSERT INTO* PRESET *VALUES* (23, 3);  
*INSERT INTO* PRESET *VALUES* (24, 3);  
*INSERT INTO* PRESET *VALUES* (25, 3);  
*INSERT INTO* PRESET *VALUES* (26, 1);  
*INSERT INTO* PRESET *VALUES* (26, 2);  
*INSERT INTO* PRESET *VALUES* (26, 3);  
*INSERT INTO* PRESET *VALUES* (27, 16);  
*INSERT INTO* PRESET *VALUES* (27, 19);  
*INSERT INTO* PRESET *VALUES* (28, 4);  
*INSERT INTO* PRESET *VALUES* (29, 5);  
*INSERT INTO* PRESET *VALUES* (30, 6);  
*INSERT INTO* PRESET *VALUES* (30, 8);  
*INSERT INTO* PRESET *VALUES* (31, 12);  
*INSERT INTO* PRESET *VALUES* (32, 10);  
*INSERT INTO* PRESET *VALUES* (33, 17);  
*INSERT INTO* PRESET *VALUES* (34, 8);  
*INSERT INTO* PRESET *VALUES* (35, 1);  
*INSERT INTO* PRESET *VALUES* (35, 9);  
*INSERT INTO* PRESET *VALUES* (36, 11);  
*INSERT INTO* PRESET *VALUES* (36, 12);  
*INSERT INTO* PRESET *VALUES* (37, 18);  
*INSERT INTO* PRESET *VALUES* (38, 19);  
*INSERT INTO* PRESET *VALUES* (39, 19);  
*INSERT INTO* PRESET *VALUES* (40, 1);  
*INSERT INTO* PRESET *VALUES* (40, 2);  
*INSERT INTO* PRESET *VALUES* (40, 3);  
*INSERT INTO* PRESET *VALUES* (41, 3);  
*INSERT INTO* PRESET *VALUES* (42, 3);  
*INSERT INTO* PRESET *VALUES* (43, 12);  
*INSERT INTO* PRESET *VALUES* (44, 20);  
*INSERT INTO* PRESET *VALUES* (44, 13);  
*INSERT INTO* PRESET *VALUES* (45, 1);  
  
*SELECT* \* *FROM* PRESET;

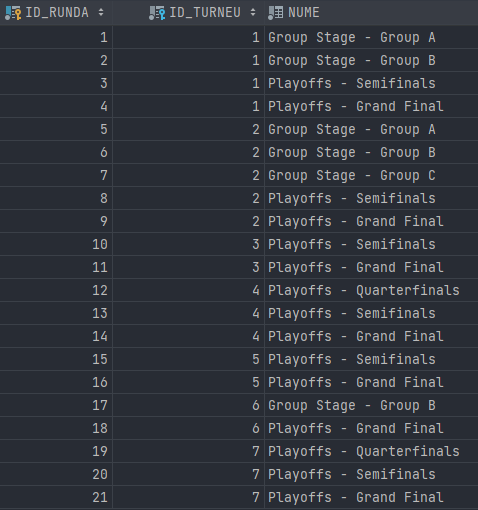
 



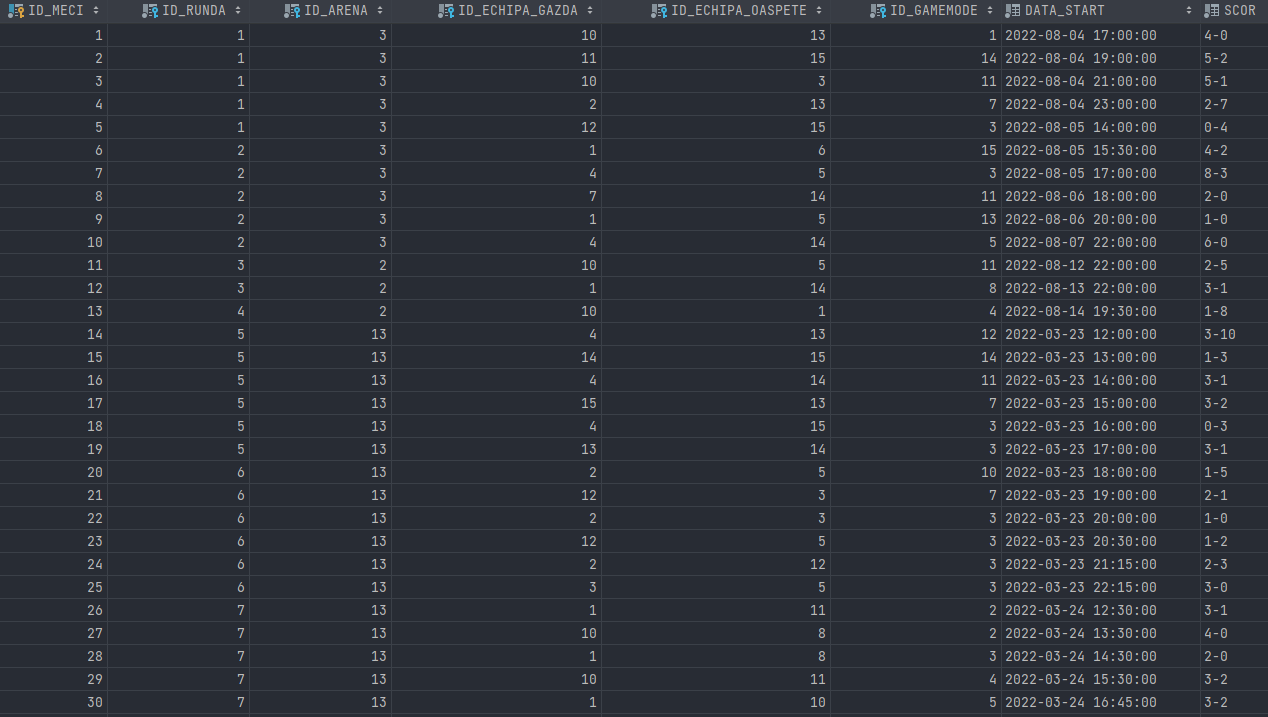
*-- Crearea tabelului TURNEU  
  
CREATE TABLE* TURNEU(  
 id\_turneu *number*(3) *CONSTRAINT* turneu\_pk *PRIMARY KEY*,  
 nume *varchar2*(50) *NOT NULL*,  
 locatie *varchar2*(100) *NOT NULL*,  
 premiu *number*(10) *NOT NULL*,  
 data\_start *date NOT NULL*);  
  
*CREATE SEQUENCE* turneu\_seq;  
  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS 2021-22 World Championship', 'Fort Worth, USA', 2085000, TO\_DATE('04-08-2022', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS 2021-22 Winter Split Major', 'Los Angeles, USA', 300000,TO\_DATE('23-03-2022', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS 2021-22 Fall Split Major', 'Stockholm, Sweden', 300000, TO\_DATE('23-03-2022', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS 2021-22 Spring Split Major', 'London, England', 300000, TO\_DATE('29-06-2022', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS Season 8 - Finals', 'Madrid, Spain', 529500, TO\_DATE('13-12-2019', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS Season 7 - Finals', 'Newark, USA', 529500, TO\_DATE('21-06-2019', 'DD-MM-YYYY'));  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS Season 6 - Finals', 'Las Vegas, USA', 500000, TO\_DATE('09-11-2018', 'DD-MM-YYYY'));  
  
*SELECT* \* *FROM* TURNEU;

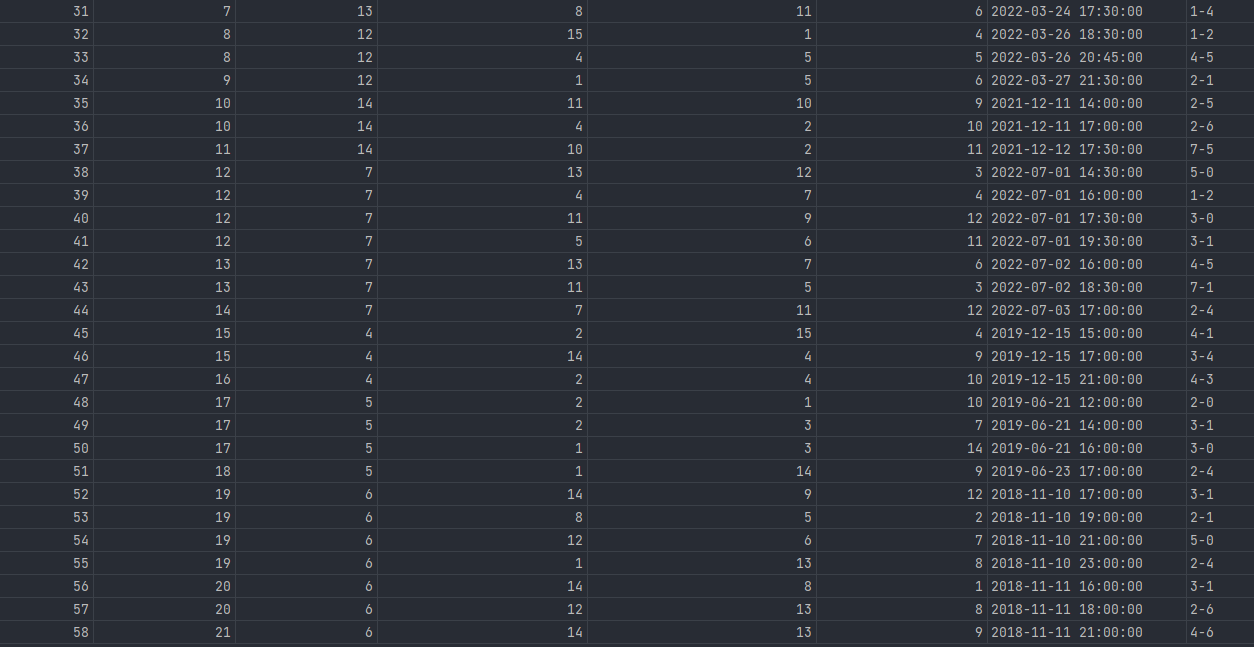


*-- Crearea tabelului RUNDA  
  
CREATE TABLE* RUNDA(  
 id\_runda *number*(5) *CONSTRAINT* runda\_pk *PRIMARY KEY*,  
 id\_turneu *number*(3) *NOT NULL*, *CONSTRAINT* runda\_turneu\_fk *FOREIGN KEY*(id\_turneu) *REFERENCES* TURNEU(id\_turneu),  
 nume *varchar2*(50) *NOT NULL*);  
  
*CREATE SEQUENCE* runda\_seq;  
  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 1, 'Group Stage - Group A');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 1, 'Group Stage - Group B');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 1, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 1, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 2, 'Group Stage - Group A');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 2, 'Group Stage - Group B');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 2, 'Group Stage - Group C');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 2, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 2, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 3, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 3, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 4, 'Playoffs - Quarterfinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 4, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 4, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 5, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 5, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 6, 'Group Stage - Group B');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 6, 'Playoffs - Grand Final');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 7, 'Playoffs - Quarterfinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 7, 'Playoffs - Semifinals');  
*INSERT INTO* RUNDA *VALUES* (runda\_seq.NEXTVAL, 7, 'Playoffs - Grand Final');  
  
*SELECT* \* *FROM* RUNDA;

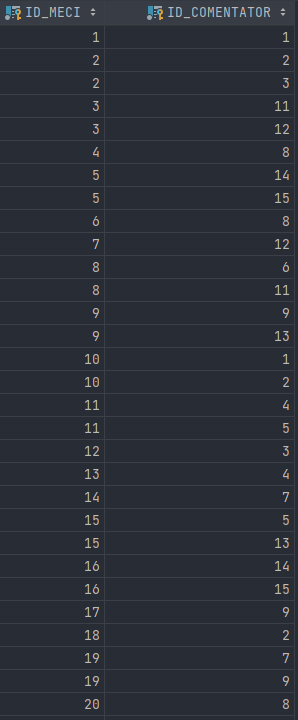
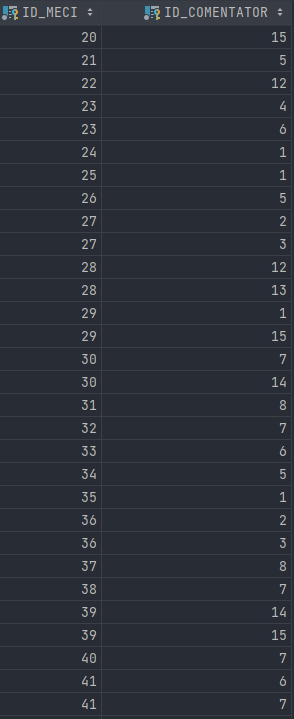


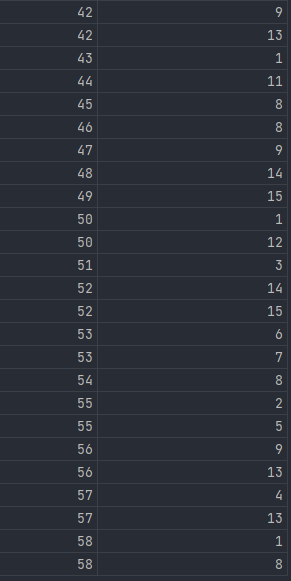
*-- Crearea tabelului MECI  
  
CREATE TABLE* MECI(  
 id\_meci *number*(5) *CONSTRAINT* meci\_pk *PRIMARY KEY*,  
 id\_runda *number*(5) *NOT NULL*, *CONSTRAINT* meci\_runda\_fk *FOREIGN KEY*(id\_runda) *REFERENCES* RUNDA(id\_runda),  
 id\_arena *number*(3) *NOT NULL*, *CONSTRAINT* meci\_arena\_fk *FOREIGN KEY*(id\_arena) *REFERENCES* ARENA(id\_arena),  
 id\_echipa\_gazda *number*(3) *NOT NULL*, *CONSTRAINT* meci\_echipa\_gazda\_fk *FOREIGN KEY*(id\_echipa\_gazda) *REFERENCES* ECHIPA(id\_echipa),  
 id\_echipa\_oaspete *number*(3) *NOT NULL*, *CONSTRAINT* meci\_echipa\_oaspete\_fk *FOREIGN KEY*(id\_echipa\_oaspete) *REFERENCES* ECHIPA(id\_echipa),  
 id\_gamemode *number*(3) *NOT NULL*, *CONSTRAINT* meci\_gamemode\_fk *FOREIGN KEY*(id\_gamemode) *REFERENCES* GAMEMODE(id\_gamemode),  
 data\_start *date NOT NULL*,  
 scor *varchar2*(7) *NOT NULL*);  
  
*CREATE SEQUENCE* meci\_seq;  
  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 10, 13, 1, TO\_DATE('04-08-2022 17:00', 'DD-MM-YYYY HH24:MI'), '4-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 11, 15, 14, TO\_DATE('04-08-2022 19:00', 'DD-MM-YYYY HH24:MI'), '5-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 10, 3, 11, TO\_DATE('04-08-2022 21:00', 'DD-MM-YYYY HH24:MI'), '5-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 2, 13, 7, TO\_DATE('04-08-2022 23:00', 'DD-MM-YYYY HH24:MI'), '2-7');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 12, 15, 3, TO\_DATE('05-08-2022 14:00', 'DD-MM-YYYY HH24:MI'), '0-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 2, 3, 1, 6, 15, TO\_DATE('05-08-2022 15:30', 'DD-MM-YYYY HH24:MI'), '4-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 2, 3, 4, 5, 3, TO\_DATE('05-08-2022 17:00', 'DD-MM-YYYY HH24:MI'), '8-3');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 2, 3, 7, 14, 11, TO\_DATE('06-08-2022 18:00', 'DD-MM-YYYY HH24:MI'), '2-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 2, 3, 1, 5, 13, TO\_DATE('06-08-2022 20:00', 'DD-MM-YYYY HH24:MI'), '1-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 2, 3, 4, 14, 5, TO\_DATE('07-08-2022 22:00', 'DD-MM-YYYY HH24:MI'), '6-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 3, 2, 10, 5, 11, TO\_DATE('12-08-2022 22:00', 'DD-MM-YYYY HH24:MI'), '2-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 3, 2, 1, 14, 8, TO\_DATE('13-08-2022 22:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 4, 2, 10, 1, 4, TO\_DATE('14-08-2022 19:30', 'DD-MM-YYYY HH24:MI'), '1-8');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 4, 13, 12, TO\_DATE('23-03-2022 12:00', 'DD-MM-YYYY HH24:MI'), '3-10');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 14, 15, 14, TO\_DATE('23-03-2022 13:00', 'DD-MM-YYYY HH24:MI'), '1-3');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 4, 14, 11, TO\_DATE('23-03-2022 14:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 15, 13, 7, TO\_DATE('23-03-2022 15:00', 'DD-MM-YYYY HH24:MI'), '3-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 4, 15, 3, TO\_DATE('23-03-2022 16:00', 'DD-MM-YYYY HH24:MI'), '0-3');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 5, 13, 13, 14, 3, TO\_DATE('23-03-2022 17:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 2, 5, 10, TO\_DATE('23-03-2022 18:00', 'DD-MM-YYYY HH24:MI'), '1-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 12, 3, 7, TO\_DATE('23-03-2022 19:00', 'DD-MM-YYYY HH24:MI'), '2-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 2, 3, 3, TO\_DATE('23-03-2022 20:00', 'DD-MM-YYYY HH24:MI'), '1-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 12, 5, 3, TO\_DATE('23-03-2022 20:30', 'DD-MM-YYYY HH24:MI'), '1-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 2, 12, 3, TO\_DATE('23-03-2022 21:15', 'DD-MM-YYYY HH24:MI'), '2-3');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 6, 13, 3, 5, 3, TO\_DATE('23-03-2022 22:15', 'DD-MM-YYYY HH24:MI'), '3-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 1, 11, 2, TO\_DATE('24-03-2022 12:30', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 10, 8, 2, TO\_DATE('24-03-2022 13:30', 'DD-MM-YYYY HH24:MI'), '4-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 1, 8, 3, TO\_DATE('24-03-2022 14:30', 'DD-MM-YYYY HH24:MI'), '2-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 10, 11, 4, TO\_DATE('24-03-2022 15:30', 'DD-MM-YYYY HH24:MI'), '3-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 1, 10, 5, TO\_DATE('24-03-2022 16:45', 'DD-MM-YYYY HH24:MI'), '3-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 7, 13, 8, 11, 6, TO\_DATE('24-03-2022 17:30', 'DD-MM-YYYY HH24:MI'), '1-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 8, 12, 15, 1, 4, TO\_DATE('26-03-2022 18:30', 'DD-MM-YYYY HH24:MI'), '1-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 8, 12, 4, 5, 5, TO\_DATE('26-03-2022 20:45', 'DD-MM-YYYY HH24:MI'), '4-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 9, 12, 1, 5, 6, TO\_DATE('27-03-2022 21:30', 'DD-MM-YYYY HH24:MI'), '2-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 10, 14, 11, 10, 9, TO\_DATE('11-12-2021 14:00', 'DD-MM-YYYY HH24:MI'), '2-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 10, 14, 4, 2, 10, TO\_DATE('11-12-2021 17:00', 'DD-MM-YYYY HH24:MI'), '2-6');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 11, 14, 10, 2, 11, TO\_DATE('12-12-2021 17:30', 'DD-MM-YYYY HH24:MI'), '7-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 12, 7, 13, 12, 3, TO\_DATE('01-07-2022 14:30', 'DD-MM-YYYY HH24:MI'), '5-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 12, 7, 4, 7, 4, TO\_DATE('01-07-2022 16:00', 'DD-MM-YYYY HH24:MI'), '1-2');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 12, 7, 11, 9, 12, TO\_DATE('01-07-2022 17:30', 'DD-MM-YYYY HH24:MI'), '3-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 12, 7, 5, 6, 11, TO\_DATE('01-07-2022 19:30', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 13, 7, 13, 7, 6, TO\_DATE('02-07-2022 16:00', 'DD-MM-YYYY HH24:MI'), '4-5');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 13, 7, 11, 5, 3, TO\_DATE('02-07-2022 18:30', 'DD-MM-YYYY HH24:MI'), '7-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 14, 7, 7, 11, 12, TO\_DATE('03-07-2022 17:00', 'DD-MM-YYYY HH24:MI'), '2-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 15, 4, 2, 15, 4, TO\_DATE('15-12-2019 15:00', 'DD-MM-YYYY HH24:MI'), '4-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 15, 4, 14, 4, 9, TO\_DATE('15-12-2019 17:00', 'DD-MM-YYYY HH24:MI'), '3-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 16, 4, 2, 4, 10, TO\_DATE('15-12-2019 21:00', 'DD-MM-YYYY HH24:MI'), '4-3');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 17, 5, 2, 1, 10, TO\_DATE('21-06-2019 12:00', 'DD-MM-YYYY HH24:MI'), '2-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 17, 5, 2, 3, 7, TO\_DATE('21-06-2019 14:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 17, 5, 1, 3, 14, TO\_DATE('21-06-2019 16:00', 'DD-MM-YYYY HH24:MI'), '3-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 18, 5, 1, 14, 9, TO\_DATE('23-06-2019 17:00', 'DD-MM-YYYY HH24:MI'), '2-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 19, 6, 14, 9, 12, TO\_DATE('10-11-2018 17:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 19, 6, 8, 5, 2, TO\_DATE('10-11-2018 19:00', 'DD-MM-YYYY HH24:MI'), '2-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 19, 6, 12, 6, 7, TO\_DATE('10-11-2018 21:00', 'DD-MM-YYYY HH24:MI'), '5-0');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 19, 6, 1, 13, 8, TO\_DATE('10-11-2018 23:00', 'DD-MM-YYYY HH24:MI'), '2-4');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 20, 6, 14, 8, 1, TO\_DATE('11-11-2018 16:00', 'DD-MM-YYYY HH24:MI'), '3-1');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 20, 6, 12, 13, 8, TO\_DATE('11-11-2018 18:00', 'DD-MM-YYYY HH24:MI'), '2-6');  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 21, 6, 14, 13, 9, TO\_DATE('11-11-2018 21:00', 'DD-MM-YYYY HH24:MI'), '4-6');  
  
*SELECT* \* *FROM* MECI;





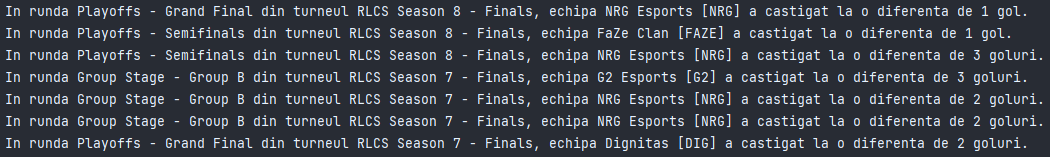
*-- Crearea tabelului STREAM  
  
CREATE TABLE* STREAM(  
 id\_meci *NUMBER*(5) *NOT NULL*, *CONSTRAINT* stream\_meci\_fk *FOREIGN KEY* (id\_meci) *REFERENCES* MECI(id\_meci),  
 id\_comentator *NUMBER*(5) *NOT NULL*, *CONSTRAINT* stream\_comentator\_fk *FOREIGN KEY* (id\_comentator) *REFERENCES* COMENTATOR(id\_comentator),  
 *CONSTRAINT* stream\_pk *PRIMARY KEY* (id\_meci, id\_comentator)  
);  
  
*INSERT INTO* STREAM *VALUES* (1, 1);  
*INSERT INTO* STREAM *VALUES* (2, 3);  
*INSERT INTO* STREAM *VALUES* (2, 2);  
*INSERT INTO* STREAM *VALUES* (3, 11);  
*INSERT INTO* STREAM *VALUES* (3, 12);  
*INSERT INTO* STREAM *VALUES* (4, 8);  
*INSERT INTO* STREAM *VALUES* (5, 14);  
*INSERT INTO* STREAM *VALUES* (5, 15);  
*INSERT INTO* STREAM *VALUES* (6, 8);  
*INSERT INTO* STREAM *VALUES* (7, 12);  
*INSERT INTO* STREAM *VALUES* (8, 6);  
*INSERT INTO* STREAM *VALUES* (8, 11);  
*INSERT INTO* STREAM *VALUES* (9, 13);  
*INSERT INTO* STREAM *VALUES* (9, 9);  
*INSERT INTO* STREAM *VALUES* (10, 1);  
*INSERT INTO* STREAM *VALUES* (10, 2);  
*INSERT INTO* STREAM *VALUES* (11, 4);  
*INSERT INTO* STREAM *VALUES* (11, 5);  
*INSERT INTO* STREAM *VALUES* (12, 3);  
*INSERT INTO* STREAM *VALUES* (13, 4);  
*INSERT INTO* STREAM *VALUES* (14, 7);  
*INSERT INTO* STREAM *VALUES* (15, 5);  
*INSERT INTO* STREAM *VALUES* (15, 13);  
*INSERT INTO* STREAM *VALUES* (16, 14);  
*INSERT INTO* STREAM *VALUES* (16, 15);  
*INSERT INTO* STREAM *VALUES* (17, 9);  
*INSERT INTO* STREAM *VALUES* (18, 2);  
*INSERT INTO* STREAM *VALUES* (19, 7);  
*INSERT INTO* STREAM *VALUES* (19, 9);  
*INSERT INTO* STREAM *VALUES* (20, 15);  
*INSERT INTO* STREAM *VALUES* (20, 8);  
*INSERT INTO* STREAM *VALUES* (21, 5);  
*INSERT INTO* STREAM *VALUES* (22, 12);  
*INSERT INTO* STREAM *VALUES* (23, 4);  
*INSERT INTO* STREAM *VALUES* (23, 6);  
*INSERT INTO* STREAM *VALUES* (24, 1);  
*INSERT INTO* STREAM *VALUES* (25, 1);  
*INSERT INTO* STREAM *VALUES* (26, 5);  
*INSERT INTO* STREAM *VALUES* (27, 2);  
*INSERT INTO* STREAM *VALUES* (27, 3);  
*INSERT INTO* STREAM *VALUES* (28, 12);  
*INSERT INTO* STREAM *VALUES* (28, 13);  
*INSERT INTO* STREAM *VALUES* (29, 15);  
*INSERT INTO* STREAM *VALUES* (29, 1);  
*INSERT INTO* STREAM *VALUES* (30, 14);  
*INSERT INTO* STREAM *VALUES* (30, 7);  
*INSERT INTO* STREAM *VALUES* (31, 8);  
*INSERT INTO* STREAM *VALUES* (32, 7);  
*INSERT INTO* STREAM *VALUES* (33, 6);  
*INSERT INTO* STREAM *VALUES* (34, 5);  
*INSERT INTO* STREAM *VALUES* (35, 1);  
*INSERT INTO* STREAM *VALUES* (36, 2);  
*INSERT INTO* STREAM *VALUES* (36, 3);  
*INSERT INTO* STREAM *VALUES* (37, 8);  
*INSERT INTO* STREAM *VALUES* (38, 7);  
*INSERT INTO* STREAM *VALUES* (39, 14);  
*INSERT INTO* STREAM *VALUES* (39, 15);  
*INSERT INTO* STREAM *VALUES* (40, 7);  
*INSERT INTO* STREAM *VALUES* (41, 6);  
*INSERT INTO* STREAM *VALUES* (41, 7);  
*INSERT INTO* STREAM *VALUES* (42, 9);  
*INSERT INTO* STREAM *VALUES* (42, 13);  
*INSERT INTO* STREAM *VALUES* (43, 1);  
*INSERT INTO* STREAM *VALUES* (44, 11);  
*INSERT INTO* STREAM *VALUES* (45, 8);  
*INSERT INTO* STREAM *VALUES* (46, 8);  
*INSERT INTO* STREAM *VALUES* (47, 9);  
*INSERT INTO* STREAM *VALUES* (48, 14);  
*INSERT INTO* STREAM *VALUES* (49, 15);  
*INSERT INTO* STREAM *VALUES* (50, 1);  
*INSERT INTO* STREAM *VALUES* (50, 12);  
*INSERT INTO* STREAM *VALUES* (51, 3);  
*INSERT INTO* STREAM *VALUES* (52, 14);  
*INSERT INTO* STREAM *VALUES* (52, 15);  
*INSERT INTO* STREAM *VALUES* (53, 6);  
*INSERT INTO* STREAM *VALUES* (53, 7);  
*INSERT INTO* STREAM *VALUES* (54, 8);  
*INSERT INTO* STREAM *VALUES* (55, 5);  
*INSERT INTO* STREAM *VALUES* (55, 2);  
*INSERT INTO* STREAM *VALUES* (56, 13);  
*INSERT INTO* STREAM *VALUES* (56, 9);  
*INSERT INTO* STREAM *VALUES* (57, 4);  
*INSERT INTO* STREAM *VALUES* (57, 13);  
*INSERT INTO* STREAM *VALUES* (58, 1);  
*INSERT INTO* STREAM *VALUES* (58, 8);  
  
*SELECT* \* *FROM* STREAM;



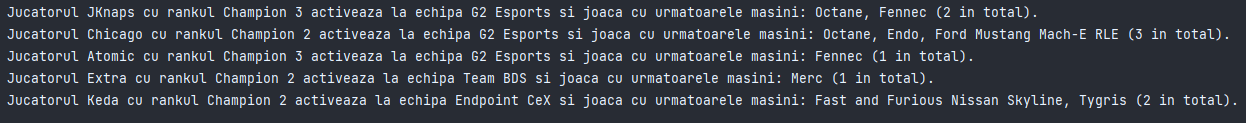
**6. Subprogram stocat independent care să utilizeze 2 tipuri de colecții**

*-- Pentru fiecare meci jucat inainte de anul 2019 inclusiv, in arene cu peste 10000 de locuri,  
-- sa se afiseze numele si tag-ul echipei castigatoare si diferenta de scor la care s-a impus.  
  
CREATE OR REPLACE PROCEDURE* castigatoare2019 *IS  
 TYPE* meciuri\_table *IS TABLE OF* MECI%*ROWTYPE INDEX BY PLS\_INTEGER*;  
 meciuri meciuri\_table;  
  
 *TYPE* castigatoare\_record *IS RECORD* (nume\_turneu TURNEU.nume%*TYPE*, nume\_runda RUNDA.nume%*TYPE*, nume\_echipa ECHIPA.nume%*TYPE*, tag\_echipa ECHIPA.tag%*TYPE*, diferenta\_scor *NUMBER*(2));  
 *c* castigatoare\_record;  
  
 *TYPE* vector *IS VARRAY* (50) *OF NUMBER*(2);  
  
 goluri\_gazda vector := vector();  
 goluri\_oaspete vector := vector();  
 diferenta\_scor vector := vector();  
  
 poz *PLS\_INTEGER*;  
  
*BEGIN  
 SELECT* \* *BULK COLLECT INTO* meciuri  
 *FROM* MECI  
 *WHERE* EXTRACT(*YEAR FROM* data\_start) <= 2019 *AND* id\_arena *IN* (*SELECT* id\_arena *FROM* ARENA *WHERE* nr\_locuri >= 10000);  
  
 *FOR i IN* meciuri.FIRST..meciuri.LAST *LOOP* poz := INSTR(meciuri(*i*).scor, '-');  
  
 goluri\_gazda.EXTEND;  
 goluri\_gazda(*i*) := TO\_NUMBER(SUBSTR(meciuri(*i*).scor, 1, poz - 1));  
  
 goluri\_oaspete.EXTEND;  
 goluri\_oaspete(*i*) := TO\_NUMBER(SUBSTR(meciuri(*i*).scor, poz + 1));  
  
 diferenta\_scor.EXTEND;  
 diferenta\_scor(*i*) := *ABS*(goluri\_gazda(*i*) - goluri\_oaspete(*i*));  
  
 *c*.diferenta\_scor := diferenta\_scor(*i*);  
  
 *SELECT* t.nume *INTO c*.nume\_turneu  
 *FROM* TURNEU t, RUNDA *r  
 WHERE r*.id\_runda = meciuri(*i*).id\_runda *AND* t.id\_turneu = *r*.id\_turneu;  
  
 *SELECT* nume *INTO c*.nume\_runda  
 *FROM* RUNDA  
 *WHERE* id\_runda = meciuri(*i*).id\_runda;  
  
  
 *IF* (goluri\_gazda(*i*) > goluri\_oaspete(*i*))  
 *THEN SELECT* nume, *tag  
 INTO c*.nume\_echipa, *c*.tag\_echipa  
 *FROM* ECHIPA  
 *WHERE* id\_echipa = meciuri(*i*).id\_echipa\_gazda;  
 *ELSE SELECT* nume, *tag  
 INTO c*.nume\_echipa, *c*.tag\_echipa  
 *FROM* ECHIPA  
 *WHERE* id\_echipa = meciuri(*i*).id\_echipa\_oaspete;  
 *END IF*;  
  
 DBMS\_OUTPUT.PUT ('In runda ' || *c*.nume\_runda || ' din turneul ' || *c*.nume\_turneu || ', echipa ' || *c*.nume\_echipa || ' ' || *c*.tag\_echipa || ' a castigat la o diferenta de ' || *c*.diferenta\_scor || ' ');  
 *IF c*.diferenta\_scor = 1 *THEN* DBMS\_OUTPUT.PUT\_LINE ('gol.');  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE ('goluri.');  
 *END IF*;  
  
 *END LOOP*;  
*END* castigatoare2019;  
/  
  
*CALL* castigatoare2019();



**7. Subprogram stocat independent care să utilizeze 2 tipuri de cursoare**

*-- Pentru fiecare jucator cu rank Champion care activeaza la o echipa din Europa,  
-- sa se afizeze masinile jucate si numarul acestora  
  
CREATE OR REPLACE PROCEDURE* masini\_jucate *IS  
 CURSOR* masini (nickname\_jucator JUCATOR.nickname%*TYPE*) *IS  
 SELECT* m.nume  
 *FROM* MASINA m, PRESET p, JUCATOR j  
 *WHERE* m.id\_masina = p.id\_masina *AND* p.id\_jucator = j.id\_jucator *AND* j.nickname = nickname\_jucator;  
  
 masina\_loop MASINA.nume%*TYPE*;  
  
 *CURSOR* jucatori *IS  
 SELECT* j.nickname nickname\_jucator, j.rank rank\_jucator, e.nume nume\_echipa  
 *FROM* JUCATOR j  
 *JOIN* ECHIPA e *ON* j.id\_echipa = e.id\_echipa  
 *WHERE LOWER*(j.rank) *LIKE* '%champion%' *AND  
 UPPER*(e.regiune) = *UPPER*('EUROPE');  
  
 counter *NUMBER*(2);  
  
*BEGIN  
 FOR* jucator\_loop *in* jucatori *LOOP* counter := 0;  
  
 DBMS\_OUTPUT.PUT ('Jucatorul ' || jucator\_loop.nickname\_jucator || ' cu rankul ' || jucator\_loop.rank\_jucator || ' activeaza la echipa ' || jucator\_loop.nume\_echipa || ' si joaca cu urmatoarele masini: ');  
  
 *OPEN* masini(jucator\_loop.nickname\_jucator);  
 *LOOP  
 FETCH* masini *INTO* masina\_loop;  
 *EXIT WHEN* masini%*NOTFOUND*;  
  
 counter := counter + 1;  
  
 *IF* counter > 1 *THEN* DBMS\_OUTPUT.PUT(', ');  
 *END IF*;  
  
 DBMS\_OUTPUT.PUT ( masina\_loop);  
 *END LOOP*;  
  
 DBMS\_OUTPUT.PUT\_LINE(' (' || counter || ' in total).');  
 *CLOSE* masini;  
 *END LOOP*;  
*END* masini\_jucate;  
/  
  
*CALL* masini\_jucate();



**8. Subprogram stocat independent de tip funcție care să utilizeze 3 tabele într-o singură comandă SQL + tratarea posibilelor excepții**

*-- Se da numele unui comentator. Sa se afiseze numarul de goluri marcate in meciurile din a doua jumatate a anului 2022 pe care le-a comentat.  
  
CREATE OR REPLACE FUNCTION* meciuri\_comentator (input\_name COMENTATOR.nume%*TYPE*)  
*RETURN NUMBER IS  
  
 CURSOR* scoruri *IS  
 SELECT* m.scor  
 *FROM* MECI m, STREAM *s*, COMENTATOR *C  
 WHERE UPPER*(input\_name) = *UPPER*(nume) *AND  
 s*.id\_comentator = *c*.id\_comentator *AND  
 s*.id\_meci = m.id\_meci *AND* EXTRACT(*MONTH FROM* m.data\_start) > 6 *AND* EXTRACT(*YEAR FROM* m.data\_start) = 2022;  
  
 scor\_loop MECI.scor%*TYPE*;  
  
 poz *PLS\_INTEGER*;  
 nr\_goluri\_gazda *NUMBER*(3);  
 nr\_goluri\_oaspete *NUMBER*(3);  
 total\_goluri *NUMBER*(3) := 0;  
  
 counter *PLS\_INTEGER* := 0;  
  
 NU\_EXISTA\_NUME *EXCEPTION*;  
 MAI\_MULTI\_COMENTATORI *EXCEPTION*;  
 NU\_EXISTA\_MECIURI *EXCEPTION*;  
*BEGIN  
 SELECT* COUNT(\*) *INTO* counter  
 *FROM* COMENTATOR  
 *WHERE UPPER*(input\_name) = *UPPER*(nume);  
  
 *IF* (counter = 0)  
 *THEN RAISE* NU\_EXISTA\_NUME;  
 *ELSIF* (counter > 1)  
 *THEN RAISE* MAI\_MULTI\_COMENTATORI;  
 *END IF*;  
  
 *SELECT* COUNT(\*) *INTO* counter  
 *FROM* COMENTATOR *c*, STREAM *s  
 WHERE UPPER*(input\_name) = *UPPER*(nume) *AND  
 c*.id\_comentator = *s*.id\_comentator;  
  
 *IF* (counter = 0)  
 *THEN RAISE* NU\_EXISTA\_MECIURI;  
 *END IF*;  
  
  
 *OPEN* scoruri;  
 *LOOP  
 FETCH* scoruri *INTO* scor\_loop;  
 *EXIT WHEN* scoruri%*NOTFOUND*;  
  
 poz := INSTR(scor\_loop, '-');  
 nr\_goluri\_gazda := TO\_NUMBER(SUBSTR(scor\_loop, 1, poz - 1));  
 nr\_goluri\_oaspete := TO\_NUMBER(SUBSTR(scor\_loop, poz + 1));  
  
 total\_goluri := total\_goluri + nr\_goluri\_gazda + nr\_goluri\_oaspete;  
 *END LOOP*;  
 *CLOSE* scoruri;  
  
 *RETURN* total\_goluri;  
  
*EXCEPTION  
 WHEN* NU\_EXISTA\_NUME *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista comentatori cu numele ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN* MAI\_MULTI\_COMENTATORI *THEN* DBMS\_OUTPUT.PUT\_LINE('Exista mai multi comentatori cu numele ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN* NU\_EXISTA\_MECIURI *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun meci comentat de ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN OTHERS THEN* DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);  
 *RETURN* 0;  
*END* meciuri\_comentator;  
/  
  
  
*DECLARE -- decomentate pe rand pentru a verifica functia* nume\_comentator *varchar2*(50) := 'Michael Jones';  
 *-- nume\_comentator varchar2(50) := 'Sean Rogers';  
 -- nume\_comentator varchar2(50) := 'Thibaut Mghaoues';  
 -- nume\_comentator varchar2(50) := 'Alex Knight';  
 -- nume\_comentator varchar2(50) := 'Robin Pachot-Giroux';* nr\_goluri *NUMBER*(3);  
*BEGIN* nr\_goluri := meciuri\_comentator(nume\_comentator);  
 *IF* (nr\_goluri = 1) *THEN* DBMS\_OUTPUT.PUT\_LINE('In a doua jumatate a anului 2022, ' || nume\_comentator || ' a comentat meciuri in care s-a marcat, in total, numai un gol.');  
 *ELSIF* (nr\_goluri > 1) *THEN* DBMS\_OUTPUT.PUT\_LINE('In a doua jumatate a anului 2022, ' || nume\_comentator || ' a comentat meciuri in care s-au marcat, in total, ' || nr\_goluri || ' goluri.');  
 *END IF*;  
*END*;





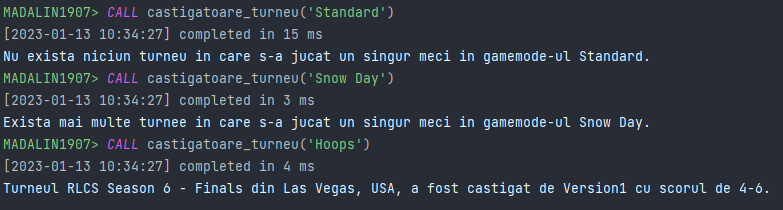






**9. Subprogram stocat independent de tip procedură care să utilizeze 5 tabele într-o singură comandă SQL + tratarea posibilelor excepții**

*-- Sa se afiseze castigatoarea turneului in care s-a jucat un singur meci intr-un gamemode dat.  
  
CREATE OR REPLACE PROCEDURE* castigatoare\_turneu (input\_gamemode GAMEMODE.nume%*TYPE*) *IS  
 TYPE* castigatoare\_record *IS RECORD* (nume\_turneu TURNEU.nume%*TYPE*, locatie\_turneu TURNEU.locatie%*TYPE*, echipa\_gazda ECHIPA.nume%*TYPE*, echipa\_oaspete ECHIPA.nume%*TYPE*, scor MECI.scor%*TYPE*);  
 castigatoare castigatoare\_record;  
  
 poz *PLS\_INTEGER*;  
 nr\_goluri\_gazda *NUMBER*(3);  
 nr\_goluri\_oaspete *NUMBER*(3);  
  
*BEGIN  
 SELECT* t.nume, t.locatie, e1.nume, e2.nume, m.scor  
 *INTO* castigatoare  
 *FROM* TURNEU t  
 *JOIN* RUNDA *r ON r*.id\_turneu = t.id\_turneu  
 *JOIN* MECI m *ON r*.id\_runda = m.id\_runda  
 *JOIN* ECHIPA e1 *ON* m.id\_echipa\_gazda = e1.id\_echipa  
 *JOIN* ECHIPA e2 *ON* m.id\_echipa\_oaspete = e2.id\_echipa  
  
 *JOIN* (*SELECT* t2.id\_turneu, COUNT(\*) *AS* match\_count  
 *FROM* MECI m2, GAMEMODE g2, RUNDA r2, TURNEU t2  
 *WHERE* g2.nume = input\_gamemode *AND* g2.id\_gamemode = m2.id\_gamemode *AND* m2.id\_runda = r2.id\_runda *AND* r2.id\_turneu = t2.id\_turneu  
 *GROUP BY* t2.id\_turneu) g *ON* g.id\_turneu = t.id\_turneu  
 *WHERE r*.nume = 'Playoffs - Grand Final' *AND* g.match\_count = 1;  
  
  
 DBMS\_OUTPUT.PUT ('Turneul ' || castigatoare.nume\_turneu || ' din ' || castigatoare.locatie\_turneu || ', a fost castigat de ');  
  
  
 poz := INSTR(castigatoare.scor, '-');  
 nr\_goluri\_gazda := TO\_NUMBER(SUBSTR(castigatoare.scor, 1, poz - 1));  
 nr\_goluri\_oaspete := TO\_NUMBER(SUBSTR(castigatoare.scor, poz + 1));  
  
 *IF* (nr\_goluri\_gazda > nr\_goluri\_oaspete) *THEN* DBMS\_OUTPUT.PUT\_LINE(castigatoare.echipa\_gazda || ' cu scorul de ' || castigatoare.scor || '.');  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE(castigatoare.echipa\_oaspete || ' cu scorul de ' || castigatoare.scor || '.');  
 *END IF*;  
  
*EXCEPTION  
 WHEN* NO\_DATA\_FOUND *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun turneu in care s-a jucat un singur meci in gamemode-ul ' || input\_gamemode || '.');  
 *WHEN* TOO\_MANY\_ROWS *THEN* DBMS\_OUTPUT.PUT\_LINE('Exista mai multe turnee in care s-a jucat un singur meci in gamemode-ul ' || input\_gamemode || '.');  
 *WHEN OTHERS THEN* DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);  
*END* castigatoare\_turneu;  
/  
  
*CALL* castigatoare\_turneu('Standard');  
*CALL* castigatoare\_turneu('Snow Day');  
*CALL* castigatoare\_turneu('Hoops');



*-- Trigger de tip LMD la nivel de comanda  
-- Se declanseaza daca se incearca modificarea tabelei TURNEU in timpul urmatorului turneu de Rocket League (06-09 aprilie 2023)  
  
CREATE OR REPLACE TRIGGER* urmatorul\_turneu\_rl  
 *BEFORE INSERT OR UPDATE OR DELETE ON* TURNEU  
*BEGIN  
 IF* (*SYSDATE BETWEEN* TO\_DATE('06-04-2023', 'DD-MM-YYYY') *AND* TO\_DATE('09-04-2023', 'DD-MM-YYYY')) *THEN* RAISE\_APPLICATION\_ERROR(-20000, 'Nu se pot modifica datele din tabela TURNEU momentan (un turneu de Rocket League este in desfasurare).');  
 *END IF*;  
*END*;  
  
  
*INSERT INTO* TURNEU *VALUES* (turneu\_seq.NEXTVAL, 'RLCS 2022-23 - Winter Split Major', 'San Diego, USA', 310000, TO\_DATE('06-04-2023', 'DD-MM-YYYY'));  
  
  
*select* \* *from* turneu;

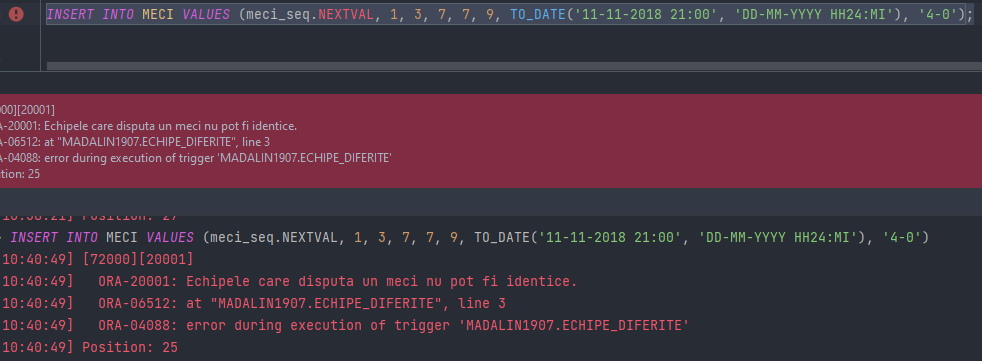
Pentru testarea acestui trigger, am modificat prima dată din IF în 06-01-2023:

**10. Trigger de tip LMD la nivel de comandă**

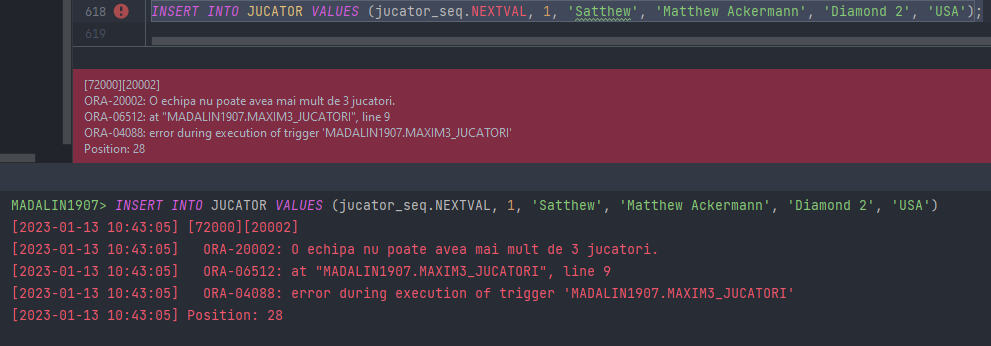


**11. Trigger de tip LMD la nivel de linie**

*-- Trigger de tip LMD la nivel de linie  
-- Se declanseaza cand se incearca introducerea / actualizarea unui meci in care echipa gazda este aceeasi cu cea oaspete  
  
CREATE OR REPLACE TRIGGER* echipe\_diferite  
 *BEFORE INSERT OR UPDATE ON* MECI  
 *FOR EACH ROW  
BEGIN  
 IF* (:*NEW*.id\_echipa\_gazda = :*NEW*.id\_echipa\_oaspete)  
 *THEN* RAISE\_APPLICATION\_ERROR(-20001, 'Echipele care disputa un meci nu pot fi identice.');  
 *END IF*;  
*END*;   
  
  
*INSERT INTO* MECI *VALUES* (meci\_seq.NEXTVAL, 1, 3, 7, 7, 9, TO\_DATE('11-11-2018 21:00', 'DD-MM-YYYY HH24:MI'), '4-0');

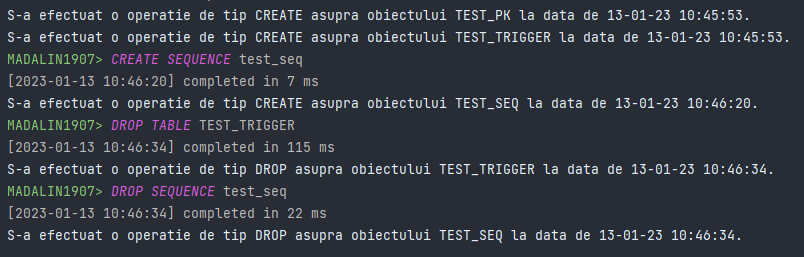


*-- Trigger de tip LMD la nivel de linie  
-- Se declanseaza cand se incearca introducerea / mutarea unui jucator intr-o echipa care are deja 3 jucatori  
  
CREATE OR REPLACE TRIGGER* maxim3\_jucatori  
 *BEFORE INSERT OR UPDATE ON* JUCATOR  
 *FOR EACH ROW  
DECLARE* nr\_jucatori *NUMBER*(1);  
*BEGIN  
 SELECT* COUNT(\*) *INTO* nr\_jucatori  
 *FROM* JUCATOR  
 *WHERE* id\_echipa = :*NEW*.id\_echipa;  
  
 *IF* (nr\_jucatori >= 3)  
 *THEN* RAISE\_APPLICATION\_ERROR(-20002, 'O echipa nu poate avea mai mult de 3 jucatori.');  
 *END IF*;  
*END*;  
  
  
*INSERT INTO* JUCATOR *VALUES* (jucator\_seq.NEXTVAL, 1, 'Satthew', 'Matthew Ackermann', 'Diamond 2', 'USA');



**12. Trigger de tip LDD**

*-- Trigger de tip LDD  
-- Se declanseaza la orice tip de operatie LDD si afiseaza informatii despre operatia care a declansat trigger-ul  
  
CREATE OR REPLACE TRIGGER* operatii\_ldd  
 *AFTER CREATE OR ALTER OR DROP ON SCHEMA  
BEGIN* DBMS\_OUTPUT.PUT\_LINE('S-a efectuat o operatie de tip ' || *SYS*.SYSEVENT || ' asupra obiectului ' || *SYS*.DICTIONARY\_OBJ\_NAME() || ' la data de ' || TO\_CHAR(*SYSDATE*, 'DD-MM-YY HH24:MI:SS') || '.');  
*END*;  
  
  
*CREATE TABLE* TEST\_TRIGGER(  
 id\_test *number*(3) *CONSTRAINT* test\_pk *PRIMARY KEY*);  
*CREATE SEQUENCE* test\_seq;  
*DROP TABLE* TEST\_TRIGGER;  
*DROP SEQUENCE* test\_seq;



**13. Pachet care să conțină toate obiectele definite în proiect**

*-- Pachet care contine toate obiectele definite in proiect  
  
CREATE OR REPLACE PACKAGE* pachet\_cerinta\_13 *AS  
 PROCEDURE* castigatoare2019;  
 *PROCEDURE* masini\_jucate;  
 *FUNCTION* meciuri\_comentator (input\_name COMENTATOR.nume%*TYPE*) *RETURN NUMBER*;  
 *PROCEDURE* castigatoare\_turneu (input\_gamemode GAMEMODE.nume%*TYPE*);  
*END* pachet\_cerinta\_13;  
/  
  
*CREATE OR REPLACE PACKAGE BODY* pachet\_cerinta\_13 *AS  
  
 -- Pentru fiecare meci jucat inainte de anul 2019 inclusiv, in arene cu peste 10000 de locuri,  
 -- sa se afiseze numele si tag-ul echipei castigatoare si diferenta de scor la care s-a impus.  
 PROCEDURE* castigatoare2019 *IS  
  
 TYPE* meciuri\_table *IS TABLE OF* MECI%*ROWTYPE INDEX BY PLS\_INTEGER*;  
 meciuri meciuri\_table;  
  
 *TYPE* castigatoare\_record *IS RECORD* (nume\_turneu TURNEU.nume%*TYPE*, nume\_runda RUNDA.nume%*TYPE*, nume\_echipa ECHIPA.nume%*TYPE*, tag\_echipa ECHIPA.tag%*TYPE*, diferenta\_scor *NUMBER*(2));  
 *c* castigatoare\_record;  
  
 *TYPE* vector *IS VARRAY* (50) *OF NUMBER*(2);  
  
 goluri\_gazda vector := vector();  
 goluri\_oaspete vector := vector();  
 diferenta\_scor vector := vector();  
  
 poz *PLS\_INTEGER*;  
  
 *BEGIN  
 SELECT* \* *BULK COLLECT INTO* meciuri  
 *FROM* MECI  
 *WHERE* EXTRACT(*YEAR FROM* data\_start) <= 2019 *AND* id\_arena *IN* (*SELECT* id\_arena *FROM* ARENA *WHERE* nr\_locuri >= 10000);  
  
 *FOR i IN* meciuri.FIRST..meciuri.LAST *LOOP* poz := INSTR(meciuri(*i*).scor, '-');  
  
 goluri\_gazda.EXTEND;  
 goluri\_gazda(*i*) := TO\_NUMBER(SUBSTR(meciuri(*i*).scor, 1, poz - 1));  
  
 goluri\_oaspete.EXTEND;  
 goluri\_oaspete(*i*) := TO\_NUMBER(SUBSTR(meciuri(*i*).scor, poz + 1));  
  
 diferenta\_scor.EXTEND;  
 diferenta\_scor(*i*) := *ABS*(goluri\_gazda(*i*) - goluri\_oaspete(*i*));  
  
 *c*.diferenta\_scor := diferenta\_scor(*i*);  
  
 *SELECT* t.nume *INTO c*.nume\_turneu  
 *FROM* TURNEU t, RUNDA *r  
 WHERE r*.id\_runda = meciuri(*i*).id\_runda *AND* t.id\_turneu = *r*.id\_turneu;  
  
 *SELECT* nume *INTO c*.nume\_runda  
 *FROM* RUNDA  
 *WHERE* id\_runda = meciuri(*i*).id\_runda;  
  
  
 *IF* (goluri\_gazda(*i*) > goluri\_oaspete(*i*))  
 *THEN SELECT* nume, *tag  
 INTO c*.nume\_echipa, *c*.tag\_echipa  
 *FROM* ECHIPA  
 *WHERE* id\_echipa = meciuri(*i*).id\_echipa\_gazda;  
 *ELSE SELECT* nume, *tag  
 INTO c*.nume\_echipa, *c*.tag\_echipa  
 *FROM* ECHIPA  
 *WHERE* id\_echipa = meciuri(*i*).id\_echipa\_oaspete;  
 *END IF*;  
  
 DBMS\_OUTPUT.PUT ('In runda ' || *c*.nume\_runda || ' din turneul ' || *c*.nume\_turneu || ', echipa ' || *c*.nume\_echipa || ' ' || *c*.tag\_echipa || ' a castigat la o diferenta de ' || *c*.diferenta\_scor || ' ');  
 *IF c*.diferenta\_scor = 1 *THEN* DBMS\_OUTPUT.PUT\_LINE ('gol.');  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE ('goluri.');  
 *END IF*;  
  
 *END LOOP*;  
 *END* castigatoare2019;  
  
  
  
  
  
 *-- Pentru fiecare jucator cu rank Champion care activeaza la o echipa din Europa,  
 -- sa se afizeze masinile jucate si numarul acestora  
  
 PROCEDURE* masini\_jucate *AS  
 CURSOR* masini (nickname\_jucator JUCATOR.nickname%*TYPE*) *IS  
 SELECT* m.nume  
 *FROM* MASINA m, PRESET p, JUCATOR j  
 *WHERE* m.id\_masina = p.id\_masina *AND* p.id\_jucator = j.id\_jucator *AND* j.nickname = nickname\_jucator;  
  
 masina\_loop MASINA.nume%*TYPE*;  
  
 *CURSOR* jucatori *IS  
 SELECT* j.nickname nickname\_jucator, j.rank rank\_jucator, e.nume nume\_echipa  
 *FROM* JUCATOR j  
 *JOIN* ECHIPA e *ON* j.id\_echipa = e.id\_echipa  
 *WHERE LOWER*(j.rank) *LIKE* '%champion%' *AND  
 UPPER*(e.regiune) = *UPPER*('EUROPE');  
  
 counter *NUMBER*(2);  
  
 *BEGIN  
 FOR* jucator\_loop *in* jucatori *LOOP* counter := 0;  
  
 DBMS\_OUTPUT.PUT ('Jucatorul ' || jucator\_loop.nickname\_jucator || ' cu rankul ' || jucator\_loop.rank\_jucator || ' activeaza la echipa ' || jucator\_loop.nume\_echipa || ' si joaca cu urmatoarele masini: ');  
  
 *OPEN* masini(jucator\_loop.nickname\_jucator);  
 *LOOP  
 FETCH* masini *INTO* masina\_loop;  
 *EXIT WHEN* masini%*NOTFOUND*;  
  
 counter := counter + 1;  
  
 *IF* counter > 1 *THEN* DBMS\_OUTPUT.PUT(', ');  
 *END IF*;  
  
 DBMS\_OUTPUT.PUT ( masina\_loop);  
 *END LOOP*;  
  
 DBMS\_OUTPUT.PUT\_LINE(' (' || counter || ' in total).');  
 *CLOSE* masini;  
 *END LOOP*;  
 *END* masini\_jucate;  
  
  
  
  
  
 *-- Se da numele unui comentator. Sa se afiseze numarul de goluri marcate in meciurile din a doua jumatate a anului 2022 pe care le-a comentat.  
  
 FUNCTION* meciuri\_comentator (input\_name COMENTATOR.nume%*TYPE*) *RETURN NUMBER IS  
  
 CURSOR* scoruri *IS  
 SELECT* m.scor  
 *FROM* MECI m, STREAM *s*, COMENTATOR *C  
 WHERE UPPER*(input\_name) = *UPPER*(nume) *AND  
 s*.id\_comentator = *c*.id\_comentator *AND  
 s*.id\_meci = m.id\_meci *AND* EXTRACT(*MONTH FROM* m.data\_start) > 6 *AND* EXTRACT(*YEAR FROM* m.data\_start) = 2022;  
  
 scor\_loop MECI.scor%*TYPE*;  
  
 poz *PLS\_INTEGER*;  
 nr\_goluri\_gazda *NUMBER*(3);  
 nr\_goluri\_oaspete *NUMBER*(3);  
 total\_goluri *NUMBER*(3) := 0;  
  
 counter *PLS\_INTEGER* := 0;  
  
 NU\_EXISTA\_NUME *EXCEPTION*;  
 MAI\_MULTI\_COMENTATORI *EXCEPTION*;  
 NU\_EXISTA\_MECIURI *EXCEPTION*;  
 *BEGIN  
 SELECT* COUNT(\*) *INTO* counter  
 *FROM* COMENTATOR  
 *WHERE UPPER*(input\_name) = *UPPER*(nume);  
  
 *IF* (counter = 0)  
 *THEN RAISE* NU\_EXISTA\_NUME;  
 *ELSIF* (counter > 1)  
 *THEN RAISE* MAI\_MULTI\_COMENTATORI;  
 *END IF*;  
  
 *SELECT* COUNT(\*) *INTO* counter  
 *FROM* COMENTATOR *c*, STREAM *s  
 WHERE UPPER*(input\_name) = *UPPER*(nume) *AND  
 c*.id\_comentator = *s*.id\_comentator;  
  
 *IF* (counter = 0)  
 *THEN RAISE* NU\_EXISTA\_MECIURI;  
 *END IF*;  
  
  
 *OPEN* scoruri;  
 *LOOP  
 FETCH* scoruri *INTO* scor\_loop;  
 *EXIT WHEN* scoruri%*NOTFOUND*;  
  
 poz := INSTR(scor\_loop, '-');  
 nr\_goluri\_gazda := TO\_NUMBER(SUBSTR(scor\_loop, 1, poz - 1));  
 nr\_goluri\_oaspete := TO\_NUMBER(SUBSTR(scor\_loop, poz + 1));  
  
 total\_goluri := total\_goluri + nr\_goluri\_gazda + nr\_goluri\_oaspete;  
 *END LOOP*;  
 *CLOSE* scoruri;  
  
 *RETURN* total\_goluri;  
  
 *EXCEPTION  
 WHEN* NU\_EXISTA\_NUME *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista comentatori cu numele ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN* MAI\_MULTI\_COMENTATORI *THEN* DBMS\_OUTPUT.PUT\_LINE('Exista mai multi comentatori cu numele ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN* NU\_EXISTA\_MECIURI *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun meci comentat de ' || input\_name || ' in baza de date.');  
 *RETURN* 0;  
 *WHEN OTHERS THEN* DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);  
 *RETURN* 0;  
 *END* meciuri\_comentator;  
  
  
  
  
  
 *-- Sa se afiseze castigatoarea turneului in care s-a jucat un singur meci intr-un gamemode dat.  
 PROCEDURE* castigatoare\_turneu (input\_gamemode GAMEMODE.nume%*TYPE*) *IS  
  
 TYPE* castigatoare\_record *IS RECORD* (nume\_turneu TURNEU.nume%*TYPE*, locatie\_turneu TURNEU.locatie%*TYPE*, echipa\_gazda ECHIPA.nume%*TYPE*, echipa\_oaspete ECHIPA.nume%*TYPE*, scor MECI.scor%*TYPE*);  
 castigatoare castigatoare\_record;  
  
 poz *PLS\_INTEGER*;  
 nr\_goluri\_gazda *NUMBER*(3);  
 nr\_goluri\_oaspete *NUMBER*(3);  
  
 *BEGIN  
 SELECT* t.nume, t.locatie, e1.nume, e2.nume, m.scor  
 *INTO* castigatoare  
 *FROM* TURNEU t  
 *JOIN* RUNDA *r ON r*.id\_turneu = t.id\_turneu  
 *JOIN* MECI m *ON r*.id\_runda = m.id\_runda  
 *JOIN* ECHIPA e1 *ON* m.id\_echipa\_gazda = e1.id\_echipa  
 *JOIN* ECHIPA e2 *ON* m.id\_echipa\_oaspete = e2.id\_echipa  
  
 *JOIN* (*SELECT* t2.id\_turneu, COUNT(\*) *AS* match\_count  
 *FROM* MECI m2, GAMEMODE g2, RUNDA r2, TURNEU t2  
 *WHERE* g2.nume = input\_gamemode *AND* g2.id\_gamemode = m2.id\_gamemode *AND* m2.id\_runda = r2.id\_runda *AND* r2.id\_turneu = t2.id\_turneu  
 *GROUP BY* t2.id\_turneu) g *ON* g.id\_turneu = t.id\_turneu  
 *WHERE r*.nume = 'Playoffs - Grand Final' *AND* g.match\_count = 1;  
  
  
 DBMS\_OUTPUT.PUT ('Turneul ' || castigatoare.nume\_turneu || ' din ' || castigatoare.locatie\_turneu || ', a fost castigat de ');  
  
  
 poz := INSTR(castigatoare.scor, '-');  
 nr\_goluri\_gazda := TO\_NUMBER(SUBSTR(castigatoare.scor, 1, poz - 1));  
 nr\_goluri\_oaspete := TO\_NUMBER(SUBSTR(castigatoare.scor, poz + 1));  
  
 *IF* (nr\_goluri\_gazda > nr\_goluri\_oaspete) *THEN* DBMS\_OUTPUT.PUT\_LINE(castigatoare.echipa\_gazda || ' cu scorul de ' || castigatoare.scor || '.');  
 *ELSE* DBMS\_OUTPUT.PUT\_LINE(castigatoare.echipa\_oaspete || ' cu scorul de ' || castigatoare.scor || '.');  
 *END IF*;  
  
 *EXCEPTION  
 WHEN* NO\_DATA\_FOUND *THEN* DBMS\_OUTPUT.PUT\_LINE('Nu exista niciun turneu in care s-a jucat un singur meci in gamemode-ul ' || input\_gamemode || '.');  
 *WHEN* TOO\_MANY\_ROWS *THEN* DBMS\_OUTPUT.PUT\_LINE('Exista mai multe turnee in care s-a jucat un singur meci in gamemode-ul ' || input\_gamemode || '.');  
 *WHEN OTHERS THEN* DBMS\_OUTPUT.PUT\_LINE('Eroare: ' || SQLERRM);  
 *END* castigatoare\_turneu;  
  
*END* pachet\_cerinta\_13;  
  
  
  
*CALL* pachet\_cerinta\_13.castigatoare2019();  
  
  
*CALL* pachet\_cerinta\_13.masini\_jucate();  
  
  
*DECLARE* nume\_comentator *varchar2*(50) := 'Robin Pachot-Giroux';  
 nr\_goluri *NUMBER*(3);  
*BEGIN* nr\_goluri := pachet\_cerinta\_13.meciuri\_comentator(nume\_comentator);  
 *IF* (nr\_goluri = 1) *THEN* DBMS\_OUTPUT.PUT\_LINE('In a doua jumatate a anului 2022, ' || nume\_comentator || ' a comentat meciuri in care s-a marcat, in total, numai un gol.');  
 *ELSIF* (nr\_goluri > 1) *THEN* DBMS\_OUTPUT.PUT\_LINE('In a doua jumatate a anului 2022, ' || nume\_comentator || ' a comentat meciuri in care s-au marcat, in total, ' || nr\_goluri || ' goluri.');  
 *END IF*;  
*END*;  
  
  
*CALL* pachet\_cerinta\_13.castigatoare\_turneu('Hoops');

