Analiza vânzării biletelor la un turneu de tenis

**MODUL ANALIZĂ** (click pe cerință pt a accesa implementarea fiecărui exercițiu)

# Descrierea modelului ales și a obiectivelor aplicației

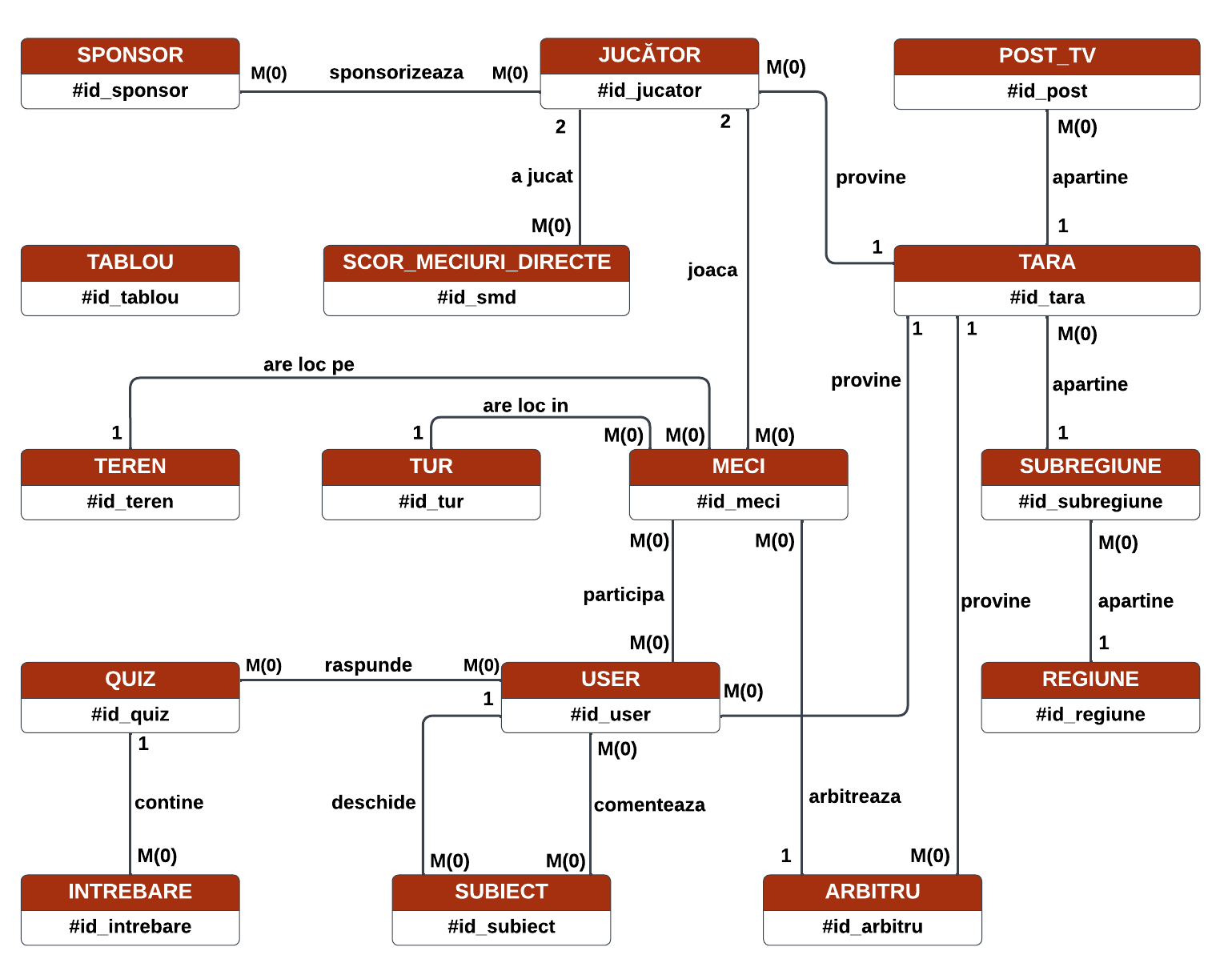
La baza acestui proiect stă un model/aplicație ce expune desfășurarea unui turneu de tenis masculin. În cadrul acestuia au loc meciuri din diferite tururi între exact 2 jucători (ne axăm pe tabloul de simplu al competiției) ce au asignate câte un teren și un arbitru.

Utilizatorii aplicației își pot cumpăra bilete la meciuri (având posibilitatea de a beneficia de o reducere de 20%), pot deschide subiecte de discuție și posta comentarii la acestea și pot răspunde la quiz-urile zilnice.

Obiectivul aplicației de Data Warehouse & Business Intelligence este de a analiza vânzarea de bilete la meciuri, punându-se accent pe perioadele din an și regiunile geografice în care au avut loc achizițiile.

# [Diagramele bazei de date OLTP](#_f5c6yamtvy5g)

## Diagrama entitate - relație a bazei de date OLTP



## [Diagrama conceptuală a bazei de date OLTP](#_ls66rtwhgfqd)

# [Diagrama stea/fulg a bazei de date depozit](#_4589gt2ttwho)



# [Descrierea câmpurilor necesare pentru fiecare tabel din baza de date depozit și modul de populare al acestora cu informații din baza de date OLTP](#_pdzbl34ho764)

* USER

Tabelul dimensiune **USER** conține cheia primară **id\_user** și informațiile **username**, **email**, și **data nașterii** utilizatorului pe care acesta le-a introdus la momentul creării contului. Toate cele 4 coloane vor fi populate cu datele din coloanele cu același nume ale tabelului USER din baza de date OLTP.

* REGIUNE

Tabelul dimensiune **REGIUNE** conține cheia primară **id\_regiune** și informațiile **țară**, **subregiune**, **regiune** din care provine utilizatorul ce achiziționează bilete. La momentul creării contului userii își selectează țara de proveniență din lista cu toate statele lumii. Acestea se găsesc în baza de date OLTP în tabelul TARA ale cărui coloane id\_tara și nume vor popula coloanele id\_regiune și tara ale tabelului REGIUNE din baza de date depozit.

Coloanele subregiune și regiune vor conține datele din coloanele nume a tabelului SUBREGIUNE, respectiv nume a tabelului REGIUNE din OLTP ce se află într-un lanț de relații one-to-many cu tabelul TARA.

* TIMP

Tabelul dimensiune **TIMP** conține informații legate de zilele în care au fost sau nu achiziționate bilete. În el stocăm toate zilele distincte din anii în care a fost pusă la dispoziție opțiunea de cumpărare a biletelor la turneu. Cheia primară **id\_timp** va fi de tip numeric reprezentând numărul de zile scurse de la Julian Day (1 ianuarie 4712 B.C), iar **data** este o coloană de tip dată calendaristică ce reține ziua. Coloanele **sapt\_luna\_an**, **semestru\_an**, **luna\_an** sunt de tip șir de caractere în formatele ‘*yyyy-mm-w*’, ‘*yyyy-sem*’ (sem = semestru 1 sau 2), respectiv ‘*yyyy-mm*’ ce indică perioada din care face parte ziua. Coloanele **zi**, **luna** și **an** sunt de tip numeric și reprezintă ziua din lună, luna din an și anul corespunzătoare zilei. Atributul **nume\_luna** este numele lunii din care face parte ziua în formatul ‘*Month*’. Informațiile despre zilele unui an vor fi generate și inserate în acest tabel la începutul fiecărui an.

* MECI

Tabelul dimensiune **MECI** conține informații legate de meciurile la care au fost cumpărate bilete. Cheia primară **id\_meci**, atributele **data\_ora** și **scor** sunt populate cu datele din coloanele cu același nume ale tabelului MECI din OLTP. Coloanele **nume\_juc1**, **prenume\_juc1**, **clasament\_juc1**, **nume\_juc2**, **prenume\_juc2**, **clasament\_juc2** vor conține informațiile celor 2 adversari aflate în coloanele nume, prenume și clasament din tabelul JUCATOR al bazei de date OLTP.

Atributele **tara\_juc1** și **tara\_juc2** referitoare la țările de proveniență ale jucătorilor sunt populate cu datele din coloana nume a tabelului TARA.

Coloanele **tur** și **pret** semnifică numele rundei partidei și prețul întreg al unui bilet corespunzător acestei etape și vor conține datele din coloanele denumire și pret ale tabelului TUR din OLTP. În cele din urmă, atributele **teren**, **nume\_arbitru** și **prenume\_arbitru** sunt completate cu informațiile coloanelor denumire din tabelul TEREN și nume și prenume din tabelul ARBITRU.

* BILET

Tabelul de fapte **BILET** conține informații legate de biletele achiziționate. Cheia primară **id\_bilet** și coloanele **cantitate**, **nr\_comanda** și **pret\_unitar** vor fi populate cu datele din coloanele id\_bilet, cantitate, nr\_comanda și pret\_platit ale tabelului BILET din baza de date OLTP. Coloana **valoare** va fi calculată la inserare ca produsul dintre cantitate și pret\_unitar. Cheile externe **id\_meci**, **id\_user** și **id\_regiune** referă tabelele dimensiune și vor conține valorile cheilor externe id\_meci, id\_user din tabelul BILET, respectiv id\_tara din tabelul USER. În final, cheia externă **id\_timp** va fi calculată ca numărul de zile dintre Julian Day și data\_achizitie din tabelul BILET al bazei de date OLTP.

# [Identificarea constrângerilor **specifice depozitelor de date** ce trebuie definite, justificând alegerea făcută](#_7s713mlih6ly)

Deoarece tabelele dimensiune nu conțin un volum mare de date, vrem să ne asigurăm că nu există sub nicio formă introducerea unor date eronate în acestea și ne dorim să beneficiem de un index unic ce identifică în mod rapid liniile, vom defini cheile primare cu opțiunile RELY ENABLE VALIDATE.

Tabelul de fapte BILET va conține un număr foarte mare de date. Din această cauză, cheia sa primară va fi definită cu opțiunile RELY DISABLE NOVALIDATE pentru a nu îngreuna procesul de aducere a datelor din baza de date OLTP în cea depozit.

În cadrul procesului ETL vom asigura respectarea constrângerilor referențiale, însă ne dorim să beneficiem de optimizarea adusă de cheile externe fără a încetini încărcarea datelor. Din acest motiv, vom defini cheile externe din tabelul de fapte cu opțiunile RELY DISABLE NOVALIDATE.

# [Identificarea indecșilor **specifici depozitelor de date** ce trebuie definiți asupra modelului; formularea unei cereri în limbaj natural care va determina utilizarea indecșilor specificați și va fi implementată în următoarea etapă](#_u7h3iv9bybz4)

Indecșii specifici depozitelor de date sunt cei de tip **Bitmap**, aceștia fiind, de obicei, utilizați pe coloanele cu cardinalitate distinctă mică. În cazul modelului nostru, coloana tur a tabelului meci este o coloană potrivită pentru un astfel de index, deoarece are doar 5 valori distincte (‘1st Round’, ‘2nd Round’, ‘Quarter-Final’, ‘Semi-Final’ și ‘Final’).

**Obțineți numărul de meciuri de 3 seturi din turul secund și procentajul acestora din numărul total de meciuri.**

# [Identificarea obiectelor de tip dimensiune ce trebuie definite asupra modelului](#_m8o37e2irciv)

Obiectele de tip dimension ne ajută să definim ierarhii între atributele stocate în tabelele dimensiune.

Pentru modelul nostru dorim informații despre vânzările de bilete în funcție de timp. Vom defini următoarea ierarhie pentru care ne creăm obiectul de tip dimensiune:

***an - semestru - luna - saptamana***

Pentru această ierarhie utilizăm coloanele an, semestru\_an, luna\_an și sapt\_luna\_an ale tabelului timp.

# [Identificarea tabelelor care vor fi partiționate și a tipului de partitionare; formularea unei cereri în limbaj natural care va determina utilizarea lor și va fi implementată în următoarea etapă.](#_3a0u8gcojcc6)

Deoarece tabelul de fapte bilet conține date istorice ce rețin ziua achiziției, o alegere potrivită de partiționare este cea prin ordonare pe baza coloanei id\_timp. Submulțimile vor corespunde lunilor din anul 2023.

**Afișați pentru fiecare meci numărul de bilete vândute în luna septembrie 2023 și valoarea totală a acestora.**

# [Formularea în limbaj natural a unei cereri SQL complexe care va fi optimizată în următoarea etapă, folosind tehnici specifice bazelor de date depozit. Precizarea tehnicilor de optimizare ce ar putea fi utilizate pentru această cerere particulară (avantaje / dezavantaje de utilizare pentru o anumită tehnică).](#_efi81afdhjj)

**Afișați pentru fiecare țară lunile în care cantitatea de bilete vândute este mai mare sau egală cu media anuală.**

Deoarece o astfel de cerere necesită grupări și join-uri între tabelul de fapte (ce conține un volum mare de înregistrări) și mai multe tabele dimensiune, o opțiune potrivită pentru optimizarea sa ar fi rescrierea de cereri cu ajutorul vizualizărilor materializate.

Avantajul unei astfel de metode este că datele nu vor mai fi extrase din tabelele originale și join-urile nu vor mai fi făcute la momentul execuției cererii, ci toate aceste informații sunt precalculate în vizualizarea materializată, astfel minimizându-se costul.

Dezavantajul unei astfel de metode este că, în funcție de opțiunile de actualizare a vizualizării materializate, pot fi necesare operații suplimentare sau un timp mai îndelungat de așteptare la permanentizarea tranzacțiilor pentru a ne asigura că datele din vizualizare corespund cu cele din tabelele de bază.

# [Formularea în limbaj natural a cel puțin 5 cereri, specifice DW, cu grad de dificultate diferit, concretizate în rapoarte (grafice) ce vor fi create în următoarele etape.](#_22lzwtqt7ywn)

**1. Afișați pentru fiecare țară top 3 meciuri pentru care utilizatorii au achiziționat bilete în anul 2023.**

**2. Afișați numărul de bilete achiziționate în semestrul 1 al anului 2023:**

**- pentru fiecare regiune, teren și tur**

**- pentru fiecare regiune și teren, indiferent de tur.**

**3. Urmăriți profitul de la lună la lună acumulat pentru fiecare meci în anul 2023.**

**4. Clasați țările ținând cont de procentul de venit adus din totalul veniturilor în anul 2023.**

**5. Afișați media vânzărilor din ultimele 2 luni, luna curentă si luna următoare ale anului 2023 pentru fiecare meci.**

**MODUL BACK-END** (click pe cerință pt a accesa partea de analiză a exercițiilor)

# [Crearea bazei de date OLTP și a utilizatorilor](#_p8qm9hzdzhgl)

CREATE USER dwbi\_oltp IDENTIFIED BY oltp;

GRANT CONNECT TO dwbi\_oltp;

GRANT RESOURCE TO dwbi\_oltp;

GRANT ALL ON DBMS\_CRYPTO TO dwbi\_oltp;

ALTER USER dwbi\_oltp QUOTA UNLIMITED ON USERS;

CREATE TABLE regiune(

id\_regiune NUMBER PRIMARY KEY,

nume VARCHAR2(50) NOT NULL

);

CREATE TABLE subregiune(

id\_subregiune NUMBER PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

id\_regiune NUMBER NOT NULL REFERENCES regiune(id\_regiune)

);

CREATE TABLE tara(

id\_tara NUMBER PRIMARY KEY,

nume VARCHAR2(50) NOT NULL,

id\_subregiune NUMBER NOT NULL REFERENCES subregiune(id\_subregiune)

);

CREATE TABLE post\_tv(

id\_post NUMBER PRIMARY KEY,

denumire VARCHAR2(20) NOT NULL,

id\_tara NUMBER NOT NULL REFERENCES tara(id\_tara),

contract\_transmisie\_completa NUMBER(1) NOT NULL

);

CREATE TABLE jucator(

id\_jucator NUMBER PRIMARY KEY,

nume VARCHAR2(25) NOT NULL,

prenume VARCHAR2(25) NOT NULL,

data\_nastere DATE,

id\_tara NUMBER NOT NULL REFERENCES tara(id\_tara),

clasament NUMBER(4),

inaltime NUMBER(3),

greutate NUMBER(3),

mana VARCHAR2(10),

an\_trecere\_prof NUMBER(4),

poza BLOB,

cap\_serie NUMBER(3),

antrenor VARCHAR2(100)

);

CREATE TABLE sponsor(

id\_sponsor NUMBER PRIMARY KEY,

denumire VARCHAR2(25) NOT NULL,

logo BLOB NOT NULL,

link\_site VARCHAR(1000) NOT NULL

);

CREATE TABLE contract(

id\_contract NUMBER PRIMARY KEY,

id\_sponsor NUMBER NOT NULL REFERENCES sponsor(id\_sponsor) ON DELETE CASCADE,

id\_jucator NUMBER NOT NULL REFERENCES jucator(id\_jucator) ON DELETE CASCADE,

CONSTRAINT uq\_contract UNIQUE (id\_sponsor, id\_jucator)

);

CREATE TABLE scor\_meciuri\_directe(

id\_smd NUMBER PRIMARY KEY,

id\_jucator1 NUMBER NOT NULL REFERENCES jucator(id\_jucator) ON DELETE CASCADE,

id\_jucator2 NUMBER NOT NULL REFERENCES jucator(id\_jucator) ON DELETE CASCADE,

victorii\_jucator1 NUMBER NOT NULL,

victorii\_jucator2 NUMBER NOT NULL,

CONSTRAINT dif\_juc\_smd CHECK(id\_jucator1 <> id\_jucator2)

);

CREATE TABLE tablou(

id\_tablou NUMBER PRIMARY KEY,

tragere\_sorti BLOB NOT NULL,

data\_tablou DATE DEFAULT SYSDATE

);

CREATE TABLE tur(

id\_tur NUMBER PRIMARY KEY,

denumire VARCHAR2(20) NOT NULL,

pret NUMBER NOT NULL

);

CREATE TABLE teren(

id\_teren NUMBER PRIMARY KEY,

denumire VARCHAR2(20) NOT NULL,

capacitate NUMBER(6)

);

CREATE TABLE arbitru(

id\_arbitru NUMBER PRIMARY KEY,

nume VARCHAR2(25) NOT NULL,

prenume VARCHAR2(25) NOT NULL,

id\_tara NUMBER NOT NULL REFERENCES tara(id\_tara)

);

CREATE TABLE meci(

id\_meci NUMBER PRIMARY KEY,

id\_jucator1 NUMBER REFERENCES jucator(id\_jucator) ON DELETE SET NULL,

id\_jucator2 NUMBER REFERENCES jucator(id\_jucator) ON DELETE SET NULL,

id\_teren NUMBER REFERENCES teren(id\_teren) ON DELETE SET NULL,

id\_arbitru NUMBER REFERENCES arbitru(id\_arbitru) ON DELETE SET NULL,

id\_tur NUMBER NOT NULL REFERENCES tur(id\_tur) ON DELETE CASCADE,

data\_ora DATE,

id\_castigator NUMBER,

scor VARCHAR2(20),

durata NUMBER(3),

locuri\_disponibile NUMBER,

CONSTRAINT dif\_juc\_meci CHECK(id\_jucator1 <> id\_jucator2),

CONSTRAINT castigator\_valid CHECK(id\_castigator is null or id\_castigator = id\_jucator1 or id\_castigator = id\_jucator2)

);

CREATE TABLE user\_(

id\_user NUMBER PRIMARY KEY,

user\_name VARCHAR2(255) NOT NULL,

parola VARCHAR2(255) NOT NULL,

email VARCHAR2(255) NOT NULL,

data\_nastere DATE NOT NULL,

puncte NUMBER DEFAULT 0 NOT NULL,

id\_tara NUMBER NOT NULL REFERENCES tara(id\_tara),

CONSTRAINT uq\_user UNIQUE (user\_name),

CONSTRAINT email\_uq UNIQUE (email)

);

CREATE TABLE bilet(

id\_bilet NUMBER PRIMARY KEY,

id\_meci NOT NULL REFERENCES meci(id\_meci),

id\_user NOT NULL REFERENCES user\_(id\_user),

cantitate NUMBER NOT NULL,

nr\_comanda NUMBER NOT NULL,

pret\_platit NUMBER NOT NULL,

data\_achizitie DATE NOT NULL

);

CREATE TABLE subiect(

id\_subiect NUMBER PRIMARY KEY,

titlu VARCHAR2(255) NOT NULL,

descriere VARCHAR2(1000),

id\_user NOT NULL REFERENCES user\_(id\_user) ON DELETE CASCADE,

data\_postare date DEFAULT SYSDATE NOT NULL

);

CREATE TABLE comentariu(

id\_comentariu NUMBER PRIMARY KEY,

continut VARCHAR2(4000) NOT NULL,

id\_subiect NOT NULL REFERENCES subiect(id\_subiect) ON DELETE CASCADE,

id\_user NOT NULL REFERENCES user\_(id\_user) ON DELETE CASCADE,

data\_postare date DEFAULT SYSDATE NOT NULL,

editat NUMBER(1) NOT NULL,

CONSTRAINT bool\_editat CHECK (editat = 0 OR editat = 1)

);

CREATE TABLE quiz(

id\_quiz NUMBER PRIMARY KEY,

titlu VARCHAR2(255) NOT NULL,

data\_postare date DEFAULT SYSDATE UNIQUE NOT NULL

);

CREATE TABLE intrebare(

id\_intrebare NUMBER PRIMARY KEY,

titlu VARCHAR2(3000) NOT NULL,

raspuns1 VARCHAR2(2000) NOT NULL,

raspuns2 VARCHAR2(2000) NOT NULL,

raspuns3 VARCHAR2(2000) NOT NULL,

raspuns4 VARCHAR2(2000) NOT NULL,

raspuns\_corect VARCHAR2(2000) NOT NULL,

id\_quiz NUMBER NOT NULL REFERENCES quiz(id\_quiz) ON DELETE CASCADE,

CONSTRAINT ex\_rc CHECK (raspuns\_corect = raspuns1 OR raspuns\_corect = raspuns2 OR raspuns\_corect = raspuns3 OR raspuns\_corect = raspuns4)

);

CREATE TABLE scor\_quiz(

id\_scor\_quiz NUMBER PRIMARY KEY,

id\_user NUMBER NOT NULL REFERENCES user\_(id\_user) ON DELETE CASCADE,

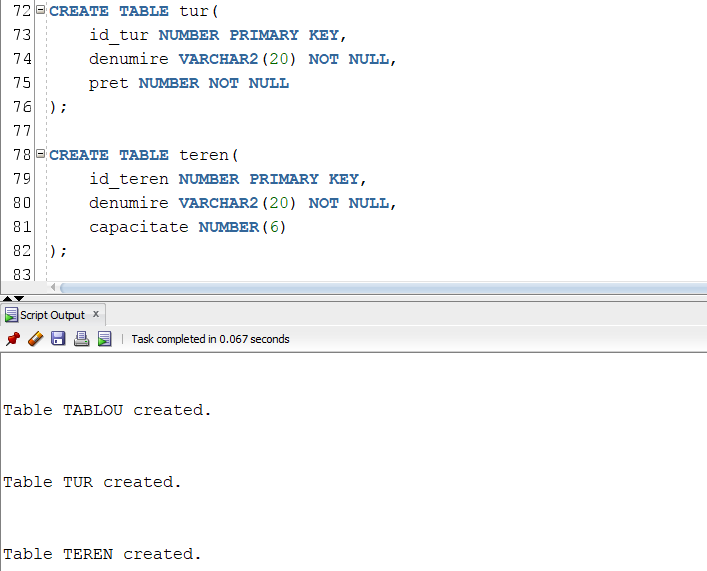
id\_quiz NUMBER NOT NULL REFERENCES quiz(id\_quiz) ON DELETE CASCADE,

scor NUMBER,

timp NUMBER,

CONSTRAINT max\_5\_scor CHECK(scor <= 5),

CONSTRAINT uq\_user\_quiz UNIQUE (id\_user, id\_quiz));



# Generarea datelor și inserarea acestora în tabele

INSERT INTO tur VALUES(1, '1st Round', 10);

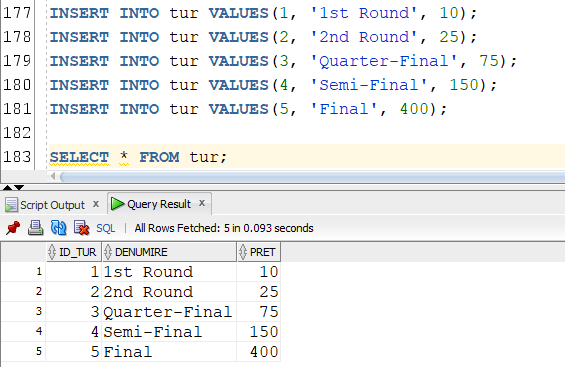
INSERT INTO tur VALUES(2, '2nd Round', 25);

INSERT INTO tur VALUES(3, 'Quarter-Final', 75);

INSERT INTO tur VALUES(4, 'Semi-Final', 150);

INSERT INTO tur VALUES(5, 'Final', 400);

SELECT \* FROM tur;



INSERT INTO teren VALUES(1, 'Court Rainer III', 10200);

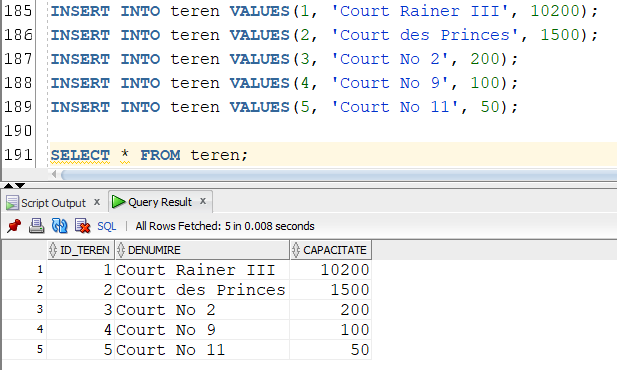
INSERT INTO teren VALUES(2, 'Court des Princes', 1500);

INSERT INTO teren VALUES(3, 'Court No 2', 200);

INSERT INTO teren VALUES(4, 'Court No 9', 100);

INSERT INTO teren VALUES(5, 'Court No 11', 50);

SELECT \* FROM teren;



INSERT INTO regiune VALUES (1, 'Europe');

INSERT INTO regiune VALUES (2, 'Asia');

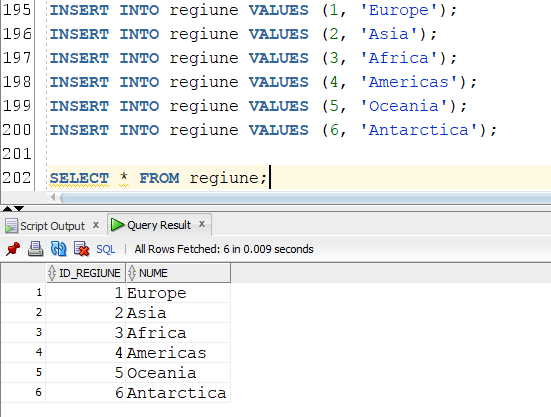
INSERT INTO regiune VALUES (3, 'Africa');

INSERT INTO regiune VALUES (4, 'Americas');

INSERT INTO regiune VALUES (5, 'Oceania');

INSERT INTO regiune VALUES (6, 'Antarctica');

SELECT \* FROM regiune;



INSERT INTO subregiune VALUES (1, 'Eastern Europe', 1);

INSERT INTO subregiune VALUES (2, 'Northern Europe', 1);

INSERT INTO subregiune VALUES (3, 'Southern Europe', 1);

INSERT INTO subregiune VALUES (4, 'Western Europe', 1);

INSERT INTO subregiune VALUES (5, 'Central Asia', 2);

INSERT INTO subregiune VALUES (6, 'Eastern Asia', 2);

INSERT INTO subregiune VALUES (7, 'South-eastern Asia', 2);

INSERT INTO subregiune VALUES (8, 'Southern Asia', 2);

INSERT INTO subregiune VALUES (9, 'Western Asia', 2);

INSERT INTO subregiune VALUES (10, 'Northern Africa', 3);

INSERT INTO subregiune VALUES (11, 'Eastern Africa', 3);

INSERT INTO subregiune VALUES (12, 'Middle Africa', 3);

INSERT INTO subregiune VALUES (13, 'Southern Africa', 3);

INSERT INTO subregiune VALUES (14, 'Western Africa', 3);

INSERT INTO subregiune VALUES (15, 'Caribbean', 4);

INSERT INTO subregiune VALUES (16, 'Central America', 4);

INSERT INTO subregiune VALUES (17, 'South America', 4);

INSERT INTO subregiune VALUES (18, 'Northern America', 4);

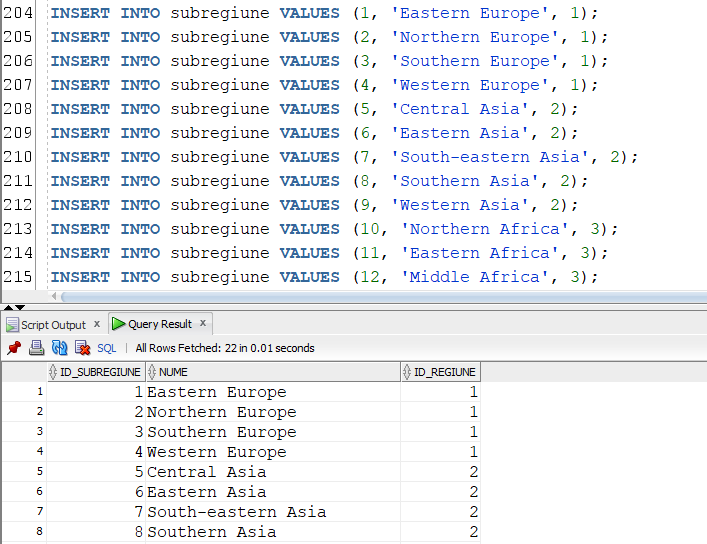
INSERT INTO subregiune VALUES (19, 'Australia and New Zealand', 5);

INSERT INTO subregiune VALUES (20, 'Melanesia', 5);

INSERT INTO subregiune VALUES (21, 'Micronesia', 5);

INSERT INTO subregiune VALUES (22, 'Polynesia', 5);

SELECT \* FROM subregiune;



INSERT INTO tara VALUES (1, 'Afghanistan', 8);

INSERT INTO tara VALUES (2, 'Albania', 3);

INSERT INTO tara VALUES (3, 'Algeria', 10);

INSERT INTO tara VALUES (4, 'Andorra', 3);

INSERT INTO tara VALUES (5, 'Angola', 12);

INSERT INTO tara VALUES (6, 'Antigua and Barbuda', 15);

INSERT INTO tara VALUES (7, 'Argentina', 17);

INSERT INTO tara VALUES (8, 'Armenia', 9);

INSERT INTO tara VALUES (9, 'Australia', 19);

INSERT INTO tara VALUES (10, 'Austria', 4);

INSERT INTO tara VALUES (11, 'Azerbaijan', 9);

INSERT INTO tara VALUES (12, 'Bahamas', 15);

INSERT INTO tara VALUES (13, 'Bahrain', 9);

INSERT INTO tara VALUES (14, 'Bangladesh', 8);

INSERT INTO tara VALUES (15, 'Barbados', 15);

INSERT INTO tara VALUES (16, 'Belarus', 1);

INSERT INTO tara VALUES (17, 'Belgium', 4);

INSERT INTO tara VALUES (18, 'Belize', 16);

INSERT INTO tara VALUES (19, 'Benin', 14);

INSERT INTO tara VALUES (20, 'Bhutan', 8);

INSERT INTO tara VALUES (21, 'Bolivia', 17);

INSERT INTO tara VALUES (22, 'Bosnia and Herzegovina', 3);

INSERT INTO tara VALUES (23, 'Botswana', 13);

INSERT INTO tara VALUES (24, 'Brazil', 17);

INSERT INTO tara VALUES (25, 'Brunei', 7);

INSERT INTO tara VALUES (26, 'Bulgaria', 1);

INSERT INTO tara VALUES (27, 'Burkina Faso', 14);

INSERT INTO tara VALUES (28, 'Burundi', 11);

INSERT INTO tara VALUES (29, 'Cabo Verde', 14);

INSERT INTO tara VALUES (30, 'Cambodia', 7);

INSERT INTO tara VALUES (31, 'Cameroon', 12);

INSERT INTO tara VALUES (32, 'Canada', 18);

INSERT INTO tara VALUES (33, 'Central African Republic', 12);

INSERT INTO tara VALUES (34, 'Chad', 12);

INSERT INTO tara VALUES (35, 'Chile', 17);

INSERT INTO tara VALUES (36, 'China', 6);

INSERT INTO tara VALUES (37, 'Colombia', 17);

INSERT INTO tara VALUES (38, 'Comoros', 11);

INSERT INTO tara VALUES (39, 'Congo', 12);

INSERT INTO tara VALUES (40, 'Costa Rica', 16);

INSERT INTO tara VALUES (41, 'Croatia', 3);

INSERT INTO tara VALUES (42, 'Cuba', 15);

INSERT INTO tara VALUES (43, 'Cyprus', 9);

INSERT INTO tara VALUES (44, 'Czech Republic', 1);

INSERT INTO tara VALUES (45, 'Denmark', 2);

INSERT INTO tara VALUES (46, 'Djibouti', 11);

INSERT INTO tara VALUES (47, 'Dominica', 15);

INSERT INTO tara VALUES (48, 'Dominican Republic', 15);

INSERT INTO tara VALUES (49, 'East Timor', 7);

INSERT INTO tara VALUES (50, 'Ecuador', 17);

INSERT INTO tara VALUES (51, 'Egypt', 10);

INSERT INTO tara VALUES (52, 'El Salvador', 16);

INSERT INTO tara VALUES (53, 'Equatorial Guinea', 12);

INSERT INTO tara VALUES (54, 'Eritrea', 11);

INSERT INTO tara VALUES (55, 'Estonia', 2);

INSERT INTO tara VALUES (56, 'Eswatini', 13);

INSERT INTO tara VALUES (57, 'Ethiopia', 11);

INSERT INTO tara VALUES (58, 'Fiji', 20);

INSERT INTO tara VALUES (59, 'Finland', 2);

INSERT INTO tara VALUES (60, 'France', 4);

INSERT INTO tara VALUES (61, 'Gabon', 12);

INSERT INTO tara VALUES (62, 'Gambia', 14);

INSERT INTO tara VALUES (63, 'Georgia', 9);

INSERT INTO tara VALUES (64, 'Germany', 4);

INSERT INTO tara VALUES (65, 'Ghana', 14);

INSERT INTO tara VALUES (66, 'Greece', 3);

INSERT INTO tara VALUES (67, 'Grenada', 15);

INSERT INTO tara VALUES (68, 'Guatemala', 16);

INSERT INTO tara VALUES (69, 'Guinea', 14);

INSERT INTO tara VALUES (70, 'Guinea-Bissau', 14);

INSERT INTO tara VALUES (71, 'Guyana', 17);

INSERT INTO tara VALUES (72, 'Haiti', 15);

INSERT INTO tara VALUES (73, 'Honduras', 16);

INSERT INTO tara VALUES (74, 'Hungary', 1);

INSERT INTO tara VALUES (75, 'Iceland', 2);

INSERT INTO tara VALUES (76, 'India', 8);

INSERT INTO tara VALUES (77, 'Indonesia', 7);

INSERT INTO tara VALUES (78, 'Iran', 8);

INSERT INTO tara VALUES (79, 'Iraq', 9);

INSERT INTO tara VALUES (80, 'Ireland', 2);

INSERT INTO tara VALUES (81, 'Israel', 9);

INSERT INTO tara VALUES (82, 'Italy', 3);

INSERT INTO tara VALUES (83, 'Ivory Coast', 14);

INSERT INTO tara VALUES (84, 'Jamaica', 15);

INSERT INTO tara VALUES (85, 'Japan', 6);

INSERT INTO tara VALUES (86, 'Jordan', 9);

INSERT INTO tara VALUES (87, 'Kazakhstan', 5);

INSERT INTO tara VALUES (88, 'Kenya', 11);

INSERT INTO tara VALUES (89, 'Kiribati', 21);

INSERT INTO tara VALUES (90, 'North Korea', 6);

INSERT INTO tara VALUES (91, 'South Korea', 6);

INSERT INTO tara VALUES (92, 'Kosovo', 3);

INSERT INTO tara VALUES (93, 'Kuwait', 9);

INSERT INTO tara VALUES (94, 'Kyrgyzstan', 5);

INSERT INTO tara VALUES (95, 'Laos', 7);

INSERT INTO tara VALUES (96, 'Latvia', 2);

INSERT INTO tara VALUES (97, 'Lebanon', 9);

INSERT INTO tara VALUES (98, 'Lesotho', 13);

INSERT INTO tara VALUES (99, 'Liberia', 14);

INSERT INTO tara VALUES (100, 'Libya', 10);

INSERT INTO tara VALUES (101, 'Liechtenstein', 4);

INSERT INTO tara VALUES (102, 'Lithuania', 2);

INSERT INTO tara VALUES (103, 'Luxembourg', 4);

INSERT INTO tara VALUES (104, 'Madagascar', 11);

INSERT INTO tara VALUES (105, 'Malawi', 11);

INSERT INTO tara VALUES (106, 'Malaysia', 7);

INSERT INTO tara VALUES (107, 'Maldives', 8);

INSERT INTO tara VALUES (108, 'Mali', 14);

INSERT INTO tara VALUES (109, 'Malta', 3);

INSERT INTO tara VALUES (110, 'Marshall Islands', 21);

INSERT INTO tara VALUES (111, 'Mauritania', 14);

INSERT INTO tara VALUES (112, 'Mauritius', 11);

INSERT INTO tara VALUES (113, 'Mexico', 16);

INSERT INTO tara VALUES (114, 'Micronesia', 21);

INSERT INTO tara VALUES (115, 'Moldova', 1);

INSERT INTO tara VALUES (116, 'Monaco', 4);

INSERT INTO tara VALUES (117, 'Mongolia', 6);

INSERT INTO tara VALUES (118, 'Montenegro', 3);

INSERT INTO tara VALUES (119, 'Morocco', 10);

INSERT INTO tara VALUES (120, 'Mozambique', 11);

INSERT INTO tara VALUES (121, 'Myanmar', 7);

INSERT INTO tara VALUES (122, 'Namibia', 13);

INSERT INTO tara VALUES (123, 'Nauru', 21);

INSERT INTO tara VALUES (124, 'Nepal', 8);

INSERT INTO tara VALUES (125, 'Netherlands', 4);

INSERT INTO tara VALUES (126, 'New Zealand', 19);

INSERT INTO tara VALUES (127, 'Nicaragua', 16);

INSERT INTO tara VALUES (128, 'Niger', 14);

INSERT INTO tara VALUES (129, 'Nigeria', 14);

INSERT INTO tara VALUES (130, 'North Macedonia', 3);

INSERT INTO tara VALUES (131, 'Norway', 2);

INSERT INTO tara VALUES (132, 'Oman', 9);

INSERT INTO tara VALUES (133, 'Pakistan', 8);

INSERT INTO tara VALUES (134, 'Palau', 21);

INSERT INTO tara VALUES (135, 'Panama', 16);

INSERT INTO tara VALUES (136, 'Papua New Guinea', 20);

INSERT INTO tara VALUES (137, 'Paraguay', 17);

INSERT INTO tara VALUES (138, 'Peru', 17);

INSERT INTO tara VALUES (139, 'Philippines', 7);

INSERT INTO tara VALUES (140, 'Poland', 1);

INSERT INTO tara VALUES (141, 'Portugal', 3);

INSERT INTO tara VALUES (142, 'Qatar', 9);

INSERT INTO tara VALUES (143, 'Romania', 1);

INSERT INTO tara VALUES (144, 'Russia', 1);

INSERT INTO tara VALUES (145, 'Rwanda', 11);

INSERT INTO tara VALUES (146, 'Saint Kitts and Nevis', 15);

INSERT INTO tara VALUES (147, 'Saint Lucia', 15);

INSERT INTO tara VALUES (148, 'Saint Vincent and the Grenadines', 15);

INSERT INTO tara VALUES (149, 'Samoa', 22);

INSERT INTO tara VALUES (150, 'San Marino', 3);

INSERT INTO tara VALUES (151, 'Sao Tome and Principe', 12);

INSERT INTO tara VALUES (152, 'Saudi Arabia', 9);

INSERT INTO tara VALUES (153, 'Senegal', 14);

INSERT INTO tara VALUES (154, 'Serbia', 3);

INSERT INTO tara VALUES (155, 'Seychelles', 11);

INSERT INTO tara VALUES (156, 'Sierra Leone', 14);

INSERT INTO tara VALUES (157, 'Singapore', 7);

INSERT INTO tara VALUES (158, 'Slovakia', 1);

INSERT INTO tara VALUES (159, 'Slovenia', 3);

INSERT INTO tara VALUES (160, 'Solomon Islands', 20);

INSERT INTO tara VALUES (161, 'Somalia', 11);

INSERT INTO tara VALUES (162, 'South Africa', 13);

INSERT INTO tara VALUES (163, 'South Sudan', 11);

INSERT INTO tara VALUES (164, 'Spain', 3);

INSERT INTO tara VALUES (165, 'Sri Lanka', 8);

INSERT INTO tara VALUES (166, 'Sudan', 10);

INSERT INTO tara VALUES (167, 'Suriname', 17);

INSERT INTO tara VALUES (168, 'Sweden', 2);

INSERT INTO tara VALUES (169, 'Switzerland', 4);

INSERT INTO tara VALUES (170, 'Syria', 9);

INSERT INTO tara VALUES (171, 'Taiwan', 6);

INSERT INTO tara VALUES (172, 'Tajikistan', 5);

INSERT INTO tara VALUES (173, 'Tanzania', 11);

INSERT INTO tara VALUES (174, 'Thailand', 7);

INSERT INTO tara VALUES (175, 'Togo', 14);

INSERT INTO tara VALUES (176, 'Tonga', 22);

INSERT INTO tara VALUES (177, 'Trinidad and Tobago', 15);

INSERT INTO tara VALUES (178, 'Tunisia', 10);

INSERT INTO tara VALUES (179, 'Turkey', 9);

INSERT INTO tara VALUES (180, 'Turkmenistan', 5);

INSERT INTO tara VALUES (181, 'Tuvalu', 22);

INSERT INTO tara VALUES (182, 'Uganda', 11);

INSERT INTO tara VALUES (183, 'Ukraine', 1);

INSERT INTO tara VALUES (184, 'United Arab Emirates', 9);

INSERT INTO tara VALUES (185, 'United Kingdom', 2);

INSERT INTO tara VALUES (186, 'United States', 18);

INSERT INTO tara VALUES (187, 'Uruguay', 17);

INSERT INTO tara VALUES (188, 'Uzbekistan', 5);

INSERT INTO tara VALUES (189, 'Vanuatu', 20);

INSERT INTO tara VALUES (190, 'Vatican City', 3);

INSERT INTO tara VALUES (191, 'Venezuela', 17);

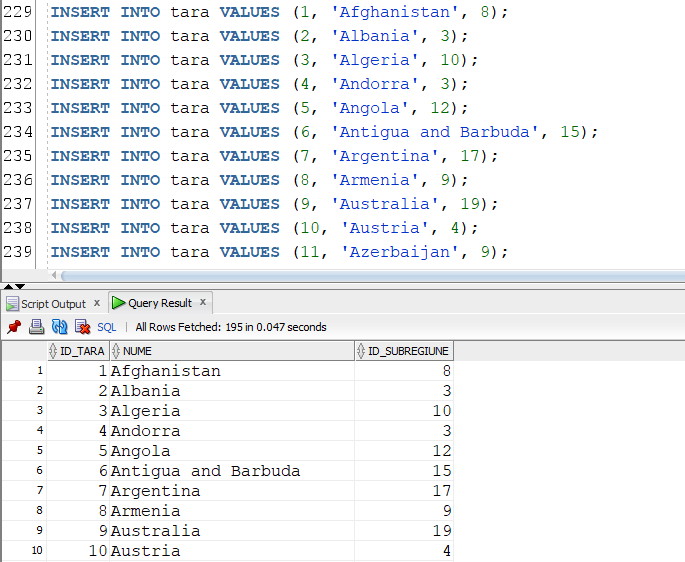
INSERT INTO tara VALUES (192, 'Vietnam', 7);

INSERT INTO tara VALUES (193, 'Yemen', 9);

INSERT INTO tara VALUES (194, 'Zambia', 11);

INSERT INTO tara VALUES (195, 'Zimbabwe', 11);

SELECT \* FROM tara;



INSERT INTO arbitru VALUES(1, 'Lahyani', 'Mohamed', 168);

INSERT INTO arbitru VALUES(2, 'Veljovic', 'Marijana', 154);

INSERT INTO arbitru VALUES(3, 'Tourte', 'Aurelie', 60);

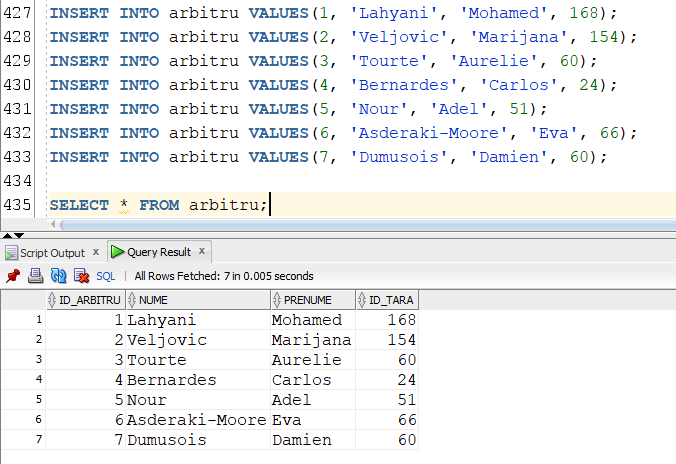
INSERT INTO arbitru VALUES(4, 'Bernardes', 'Carlos', 24);

INSERT INTO arbitru VALUES(5, 'Nour', 'Adel', 51);

INSERT INTO arbitru VALUES(6, 'Asderaki-Moore', 'Eva', 66);

INSERT INTO arbitru VALUES(7, 'Dumusois', 'Damien', 60);

SELECT \* FROM arbitru;



INSERT INTO jucator VALUES(100, 'Karatsev', 'Aslan', to\_date('04-09-1993', 'DD-MM-YYYY'), 144, 20, 185, 85, 'DREAPTA', 2013, null, null, 'Yahor Yatsyk');

INSERT INTO jucator VALUES(101, 'Ruud', 'Casper', to\_date('22-12-1998', 'DD-MM-YYYY'), 131, 8, 183, 77, 'DREAPTA', 2015, null, 7, 'Christian Ruud');

INSERT INTO jucator VALUES(102, 'Harris', 'Lloyd', to\_date('24-02-1997', 'DD-MM-YYYY'), 162, 32, 193, 80, 'DREAPTA', 2015, null, null, 'Anthony Harris');

INSERT INTO jucator VALUES(103, 'Thiem', 'Dominic', to\_date('03-09-1993', 'DD-MM-YYYY'), 10, 15, 185, 79, 'DREAPTA', 2011, null, null, 'Nicolas Massu');

INSERT INTO jucator VALUES(104, 'Garin', 'Cristian', to\_date('30-05-1996', 'DD-MM-YYYY'), 35, 18, 185, 85, 'DREAPTA', 2011, null, null, 'Franco Davin');

INSERT INTO jucator VALUES(105, 'Sinner', 'Jannik', to\_date('16-08-2001', 'DD-MM-YYYY'), 82, 11, 185, 68, 'DREAPTA', 2018, null, null, 'Riccardo Piatti, Andrea Volpini');

INSERT INTO jucator VALUES(106, 'Zverev', 'Alexander', to\_date('20-04-1997', 'DD-MM-YYYY'), 64, 3, 198, 90, 'DREAPTA', 2013, null, 3, 'Alexander Zverev Sr.');

INSERT INTO jucator VALUES(107, 'Berrettini', 'Matteo', to\_date('12-04-1996', 'DD-MM-YYYY'), 82, 7, 196, 90, 'DREAPTA', 2015, null, 6, 'Vincenzo Santopadre, Marco Gulisano, Umberto Rianna');

INSERT INTO jucator VALUES(108, 'Schwartzman', 'Diego', to\_date('16-08-1992', 'DD-MM-YYYY'), 7, 13, 170, 64, 'DREAPTA', 2010, null, null, 'Juan Ignacio Chela, Alejandro Fabbri');

INSERT INTO jucator VALUES(109, 'Monfils', 'Gael', to\_date('01-09-1986', 'DD-MM-YYYY'), 60, 19, 193, 85, 'DREAPTA', 2004, null, null, 'Gunter Bresnik, Richard Ruckelshausen');

INSERT INTO jucator VALUES(110, 'Carreno Busta', 'Pablo', to\_date('12-07-1991', 'DD-MM-YYYY'), 164, 21, 188, 78, 'DREAPTA', 2009, null, null, 'Samuel Lopez, Cesar Fabregas');

INSERT INTO jucator VALUES(111, 'Humbert', 'Ugo', to\_date('26-06-1998', 'DD-MM-YYYY'), 60, 31, 188, 72, 'STANGA', 2016, null, null, 'Nicolas Copin, Thierry Ascione');

INSERT INTO jucator VALUES(112, 'Medvedev', 'Daniil', to\_date('11-02-1996', 'DD-MM-YYYY'), 144, 2, 198, 83, 'DREAPTA', 2014, null, 2, 'Gilles Cervara');

INSERT INTO jucator VALUES(113, 'Auger-Aliassime', 'Felix', to\_date('08-08-2000', 'DD-MM-YYYY'), 32, 9, 193, 88, 'DREAPTA', 2017, null, 8, 'Frederic Fontang, Toni Nadal');

INSERT INTO jucator VALUES(114, 'Basilashvili', 'Nikoloz', to\_date('23-02-1992', 'DD-MM-YYYY'), 63, 23, 185, 79, 'DREAPTA', 2008, null, null, null);

INSERT INTO jucator VALUES(115, 'Evans', 'Daniel', to\_date('23-05-1990', 'DD-MM-YYYY'), 185, 26, 175, 75, 'DREAPTA', 2006, null, null, 'Sebastian Prieto');

INSERT INTO jucator VALUES(116, 'Rublev', 'Andrey', to\_date('20-10-1997', 'DD-MM-YYYY'), 144, 5, 188, 70, 'DREAPTA', 2014, null, 5, 'Fernando Vicente');

INSERT INTO jucator VALUES(117, 'Sonego', 'Lorenzo', to\_date('11-05-1995', 'DD-MM-YYYY'), 82, 27, 191, 76, 'DREAPTA', 2013, null, null, 'Gipo Arbino');

INSERT INTO jucator VALUES(118, 'Norrie', 'Cameron', to\_date('23-08-1995', 'DD-MM-YYYY'), 185, 12, 188, 82, 'STANGA', 2016, null, null, 'Facundo Lugones');

INSERT INTO jucator VALUES(119, 'Shapovalov', 'Denis', to\_date('15-04-1999', 'DD-MM-YYYY'), 32, 14, 183, 75, 'STANGA', 2017, null, null, 'Tessa Shapovalova, Mikhail Youzhny, Jamie Delgado');

INSERT INTO jucator VALUES(120, 'Dimitrov', 'Grigor', to\_date('16-05-1991', 'DD-MM-YYYY'), 26, 28, 191, 81, 'DREAPTA', 2008, null, null, 'Dante Bottini');

INSERT INTO jucator VALUES(121, 'Cilic', 'Marin', to\_date('29-09-1988', 'DD-MM-YYYY'), 41, 29, 198, 89, 'DREAPTA', 2005, null, null, 'Vedran Martic, Vilim Visak');

INSERT INTO jucator VALUES(122, 'Khachanov', 'Karen', to\_date('21-05-1996', 'DD-MM-YYYY'), 144, 30, 198, 87, 'DREAPTA', 2013, null, null, 'Jose Clavet');

INSERT INTO jucator VALUES(123, 'Alcaraz', 'Carlos', to\_date('05-05-2003', 'DD-MM-YYYY'), 164, 33, 185, 72, 'DREAPTA', 2018, null, null, 'Juan Carlos Ferrero');

INSERT INTO jucator VALUES(124, 'Djokovic', 'Novak', to\_date('22-05-1987', 'DD-MM-YYYY'), 154, 1, 188, 77, 'DREAPTA', 2003, null, 1, 'Marian Vajda, Goran Ivanisevic');

INSERT INTO jucator VALUES(125, 'Isner', 'John', to\_date('26-04-1985', 'DD-MM-YYYY'), 186, 24, 208, 108, 'DREAPTA', 2007, null, null, 'David Macpherson');

INSERT INTO jucator VALUES(126, 'Bautista Agut', 'Roberto', to\_date('14-04-1988', 'DD-MM-YYYY'), 164, 17, 183, 75, 'DREAPTA', 2005, null, null, 'Daniel Gimeno-Traver, Tomas Carbonell');

INSERT INTO jucator VALUES(127, 'Murray', 'Andy', to\_date('15-05-1987', 'DD-MM-YYYY'), 185, 87, 191, 82, 'DREAPTA', 2005, null, null, 'Daniel Vallverdu');

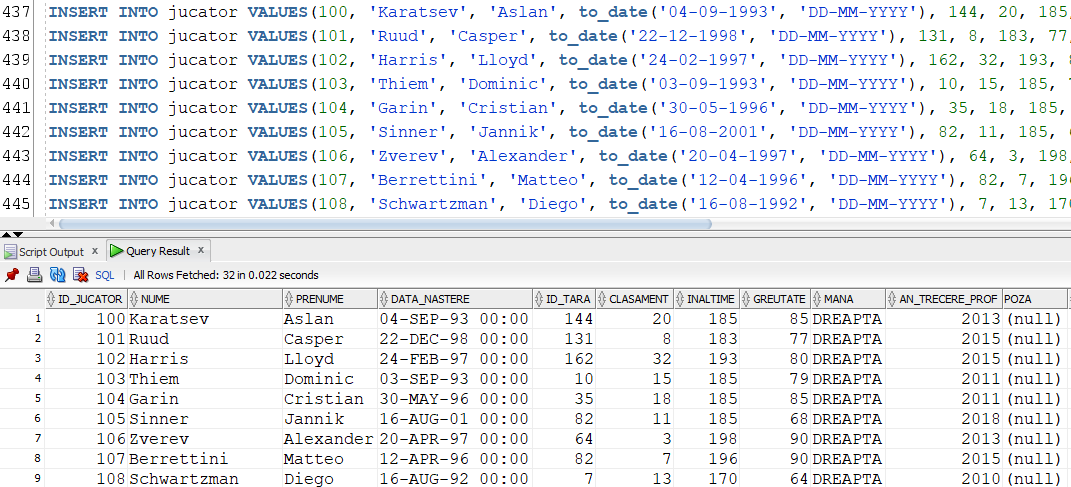
INSERT INTO jucator VALUES(128, 'Tsitsipas', 'Stefanos', to\_date('12-08-1998', 'DD-MM-YYYY'), 66, 4, 193, 85, 'DREAPTA', 2016, null, 4, 'Apostolos Tsitsipas');

INSERT INTO jucator VALUES(129, 'Fritz', 'Taylor', to\_date('28-10-1997', 'DD-MM-YYYY'), 186, 22, 193, 86, 'DREAPTA', 2015, null, null, 'Michael Russell, Paul Annacone, David Nainkin');

INSERT INTO jucator VALUES(130, 'Hurkacz', 'Hubert', to\_date('11-02-1997', 'DD-MM-YYYY'), 140, 10, 196, 81, 'DREAPTA', 2015, null, null, 'Craig Boynton');

INSERT INTO jucator VALUES(131, 'Opelka', 'Reilly', to\_date('28-08-1997', 'DD-MM-YYYY'), 186, 25, 211, 102, 'DREAPTA', 2015, null, null, 'Jay Berger, Jean-Yves Aubone');

SELECT \* FROM jucator;



INSERT INTO meci VALUES(500, 124, 117, 1, 1, 1, to\_date('30-10-2023 18:00', 'DD-MM-YYYY HH24:MI'), 124, '6-3 6-4', 93, 10200);

INSERT INTO meci VALUES(501, 125, 129, 3, 5, 1, to\_date('30-10-2023 14:00', 'DD-MM-YYYY HH24:MI'), 129, '7-5 6-7 7-6', 150, 200);

INSERT INTO meci VALUES(502, 105, 123, 2, 2, 1, to\_date('30-10-2023 18:00', 'DD-MM-YYYY HH24:MI'), 105, '2-6 6-3 6-3', 128, 1500);

INSERT INTO meci VALUES(503, 121, 107, 1, 1, 1, to\_date('30-10-2023 12:00', 'DD-MM-YYYY HH24:MI'), 107, '7-5 7-5', 106, 10200);

INSERT INTO meci VALUES(504, 101, 104, 2, 3, 1, to\_date('30-10-2023 15:00', 'DD-MM-YYYY HH24:MI'), 101, '6-4 6-4', 97, 1500);

INSERT INTO meci VALUES(505, 109, 126, 3, 7, 1, to\_date('30-10-2023 17:00', 'DD-MM-YYYY HH24:MI'), 126, '4-6 6-3 7-5', 133, 200);

INSERT INTO meci VALUES(506, 103, 118, 2, 2, 1, to\_date('30-10-2023 12:00', 'DD-MM-YYYY HH24:MI'), 103, '6-4 2-6 6-3', 114, 1500);

INSERT INTO meci VALUES(507, 122, 106, 1, 4, 1, to\_date('30-10-2023 15:00', 'DD-MM-YYYY HH24:MI'), 106, '6-4 6-3', 99, 10200);

INSERT INTO meci VALUES(508, 116, 130, 1, 6, 1, to\_date('31-10-2023 12:00', 'DD-MM-YYYY HH24:MI'), 116, '7-6 6-4', 107, 10200);

INSERT INTO meci VALUES(509, 110, 112, 1, 4, 1, to\_date('31-10-2023 15:00', 'DD-MM-YYYY HH24:MI'), 112, '7-5 6-3', 111, 10200);

INSERT INTO meci VALUES(510, 128, 127, 1, 7, 1, to\_date('31-10-2023 18:00', 'DD-MM-YYYY HH24:MI'), 128, '6-3 6-3', 87, 10200);

INSERT INTO meci VALUES(511, 120, 115, 2, 7, 1, to\_date('31-10-2023 12:00', 'DD-MM-YYYY HH24:MI'), 120, '5-7 6-3 7-5', 132, 1500);

INSERT INTO meci VALUES(512, 119, 111, 2, 6, 1, to\_date('31-10-2023 15:00', 'DD-MM-YYYY HH24:MI'), 119, '6-2 6-2', 68, 1500);

INSERT INTO meci VALUES(513, 102, 113, 2, 1, 1, to\_date('31-10-2023 18:00', 'DD-MM-YYYY HH24:MI'), 113, '6-3 6-2', 72, 1500);

INSERT INTO meci VALUES(514, 100, 131, 3, 5, 1, to\_date('31-10-2023 14:00', 'DD-MM-YYYY HH24:MI'), 100, '7-6 4-6 7-6', 139, 200);

INSERT INTO meci VALUES(515, 108, 114, 3, 3, 1, to\_date('31-10-2023 17:00', 'DD-MM-YYYY HH24:MI'), 108, '6-4 7-5', 88, 200);

INSERT INTO meci VALUES(516, 124, 129, 1, 4, 2, to\_date('01-11-2023 18:00', 'DD-MM-YYYY HH24:MI'), 124, '6-4 6-4', 92, 10200);

INSERT INTO meci VALUES(517, 105, 107, 1, 1, 2, to\_date('01-11-2023 12:00', 'DD-MM-YYYY HH24:MI'), 105, '6-3 4-6 7-5', 119, 10200);

INSERT INTO meci VALUES(518, 103, 106, 1, 2, 2, to\_date('01-11-2023 15:00', 'DD-MM-YYYY HH24:MI'), 106, '6-4 6-3', 85, 10200);

INSERT INTO meci VALUES(519, 101, 126, 2, 3, 2, to\_date('01-11-2023 16:30', 'DD-MM-YYYY HH24:MI'), 101, '7-5 6-4', 101, 1500);

INSERT INTO meci VALUES(520, 128, 119, 1, 2, 2, to\_date('02-11-2023 15:00', 'DD-MM-YYYY HH24:MI'), 128, '6-3 7-5', 123, 10186);

INSERT INTO meci VALUES(521, 120, 113, 2, 3, 2, to\_date('02-11-2023 16:30', 'DD-MM-YYYY HH24:MI'), 120, '2-6 6-4 6-4', 144, 1153);

INSERT INTO meci VALUES(522, 116, 100, 1, 5, 2, to\_date('02-11-2023 12:00', 'DD-MM-YYYY HH24:MI'), 116, '7-6 6-3', 111, 10162);

INSERT INTO meci VALUES(523, 108, 112, 1, 4, 2, to\_date('02-11-2023 18:00', 'DD-MM-YYYY HH24:MI'), 112, '6-4 6-1', 86, 10187);

INSERT INTO meci VALUES(524, 128, 120, 1, 1, 3, to\_date('03-11-2023 12:00', 'DD-MM-YYYY HH24:MI'), 120, '6-3 7-5', 90, 10189);

INSERT INTO meci VALUES(525, 116, 112, 1, 2, 3, to\_date('03-11-2023 15:00', 'DD-MM-YYYY HH24:MI'), 116, '6-7 7-5 7-5', 170, 10190);

INSERT INTO meci VALUES(526, 124, 105, 1, 3, 3, to\_date('03-11-2023 18:00', 'DD-MM-YYYY HH24:MI'), 124, '6-3 7-6', 101, 10190);

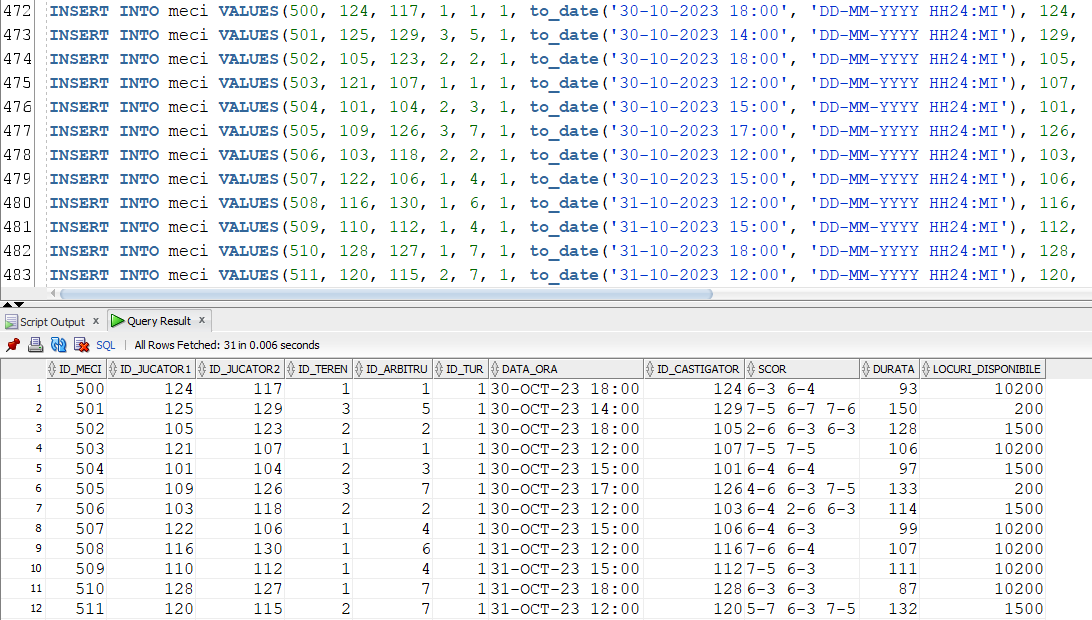
INSERT INTO meci VALUES(527, 101, 106, 2, 4, 3, to\_date('03-11-2023 16:30', 'DD-MM-YYYY HH24:MI'), 101, '6-4 4-6 6-4', 139, 1489);

INSERT INTO meci VALUES(528, 120, 116, 1, 3, 4, to\_date('04-11-2023 17:00', 'DD-MM-YYYY HH24:MI'), 116, '6-4 7-5', 98, 10194);

INSERT INTO meci VALUES(529, 124, 101, 1, 1, 4, to\_date('04-11-2023 14:00', 'DD-MM-YYYY HH24:MI'), 124, '6-1 6-2', 62, 10194);

INSERT INTO meci VALUES(530, 124, 116, 1, 3, 5, to\_date('05-11-2023 16:30', 'DD-MM-YYYY HH24:MI'), 124, '7-5 6-3', 115, 10200);

SELECT \* FROM meci;



CREATE OR REPLACE FUNCTION HASH\_PASSWORD

(user\_name in varchar2,

parola in varchar2)

RETURN varchar2

IS

hash\_parola varchar2(255);

input\_string varchar2(255);

salt varchar2(255) := '2GSK5LS73NC8SVB9DTQ6ZK810PXLGWZUJSA';

BEGIN

input\_string := parola || substr(salt, 10, 20) || user\_name || substr(salt, 25, 30);

hash\_parola := dbms\_crypto.hash(

src => utl\_raw.cast\_to\_raw(input\_string),

typ => dbms\_crypto.hash\_sh256

);

return hash\_parola;

END HASH\_PASSWORD;

/

DECLARE

username user\_.user\_name%TYPE;

tara\_id user\_.id\_tara%TYPE;

parola VARCHAR2(10) := 'Parola1!';

parola\_hash VARCHAR2(255);

data\_nasterii user\_.data\_nastere%TYPE;

email user\_.email%TYPE;

BEGIN

FOR i IN 1..150 LOOP

username := dbms\_random.string('U', dbms\_random.value(7, 15)) || i;

parola\_hash := hash\_password(username, parola);

email := username || '@gmail.com';

SELECT id\_tara INTO tara\_id

FROM (SELECT id\_tara FROM tara

ORDER BY dbms\_random.value)

WHERE rownum = 1;

data\_nasterii := TO\_DATE(TRUNC(dbms\_random.value(to\_char(DATE '1960-01-01', 'J'),

to\_char(DATE '2008-01-01', 'J'))), 'J');

INSERT INTO user\_

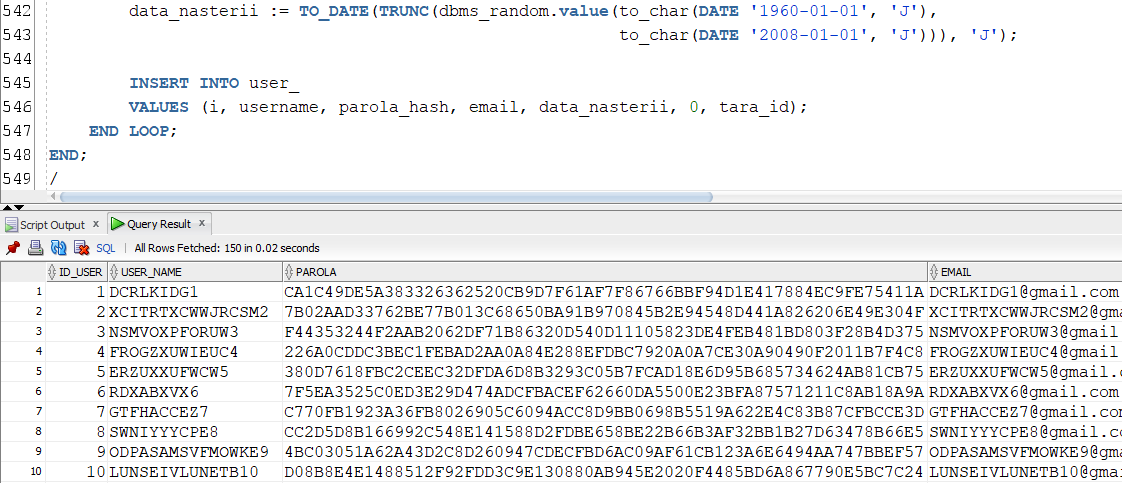
VALUES (i, username, parola\_hash, email, data\_nasterii, 0, tara\_id);

END LOOP;

END;

/

select \* FROM user\_;



DECLARE

meci\_id bilet.id\_meci%TYPE;

data\_ora\_meci meci.data\_ora%TYPE;

pret\_intreg\_meci tur.pret%TYPE;

user\_id bilet.id\_user%TYPE;

cant bilet.cantitate%TYPE;

comanda bilet.nr\_comanda%TYPE;

data\_cumparare bilet.data\_achizitie%TYPE;

pret\_platit bilet.pret\_platit%TYPE;

BEGIN

FOR i IN 1..150000 LOOP

SELECT id\_user INTO user\_id

FROM (SELECT id\_user FROM user\_

ORDER BY dbms\_random.value)

WHERE rownum = 1;

SELECT id\_meci, data\_ora, pret

INTO meci\_id, data\_ora\_meci, pret\_intreg\_meci

FROM (SELECT id\_meci, data\_ora, pret

FROM meci m, tur t

WHERE m.id\_tur = t.id\_tur

ORDER BY dbms\_random.value)

WHERE rownum = 1;

data\_cumparare := TO\_DATE(TRUNC(dbms\_random.value(to\_char(DATE '2023-01-01', 'J'),

to\_char(data\_ora\_meci, 'J'))), 'J');

cant := trunc(dbms\_random.value(1, 6));

comanda := trunc(dbms\_random.value(1000000000, 9999999999));

IF dbms\_random.value(0, 1) < 0.3 THEN

pret\_platit := round(pret\_intreg\_meci \* 0.8);

ELSE

pret\_platit := pret\_intreg\_meci;

END IF;

INSERT INTO bilet

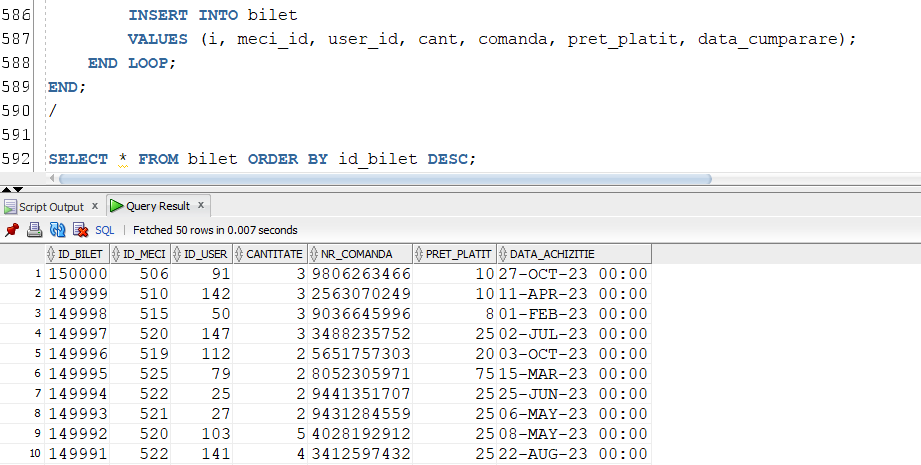
VALUES (i, meci\_id, user\_id, cant, comanda, pret\_platit, data\_cumparare);

END LOOP;

END;

/

SELECT \* FROM bilet ORDER BY id\_bilet DESC;



# [Crearea bazei de date depozit și a utilizatorilor](#_d7kfv9zceigc)

CREATE USER dwbi\_olap IDENTIFIED BY olap;

GRANT CONNECT TO dwbi\_olap;

GRANT RESOURCE TO dwbi\_olap;

ALTER USER dwbi\_olap QUOTA UNLIMITED ON USERS;

GRANT CREATE DIMENSION TO dwbi\_olap;

GRANT CREATE MATERIALIZED VIEW TO dwbi\_olap;

GRANT QUERY REWRITE TO dwbi\_olap;

CREATE TABLE regiune(

id\_regiune NUMBER,

tara VARCHAR2(50) NOT NULL,

subregiune VARCHAR2(50) NOT NULL,

regiune VARCHAR2(50) NOT NULL);

CREATE TABLE user\_(

id\_user NUMBER,

user\_name VARCHAR2(255) NOT NULL,

email VARCHAR2(255) NOT NULL,

data\_nastere DATE NOT NULL);

CREATE TABLE timp(

id\_timp NUMBER,

data DATE NOT NULL,

sapt\_luna\_an VARCHAR2(10) NOT NULL,

semestru\_an VARCHAR2(10) NOT NULL,

luna\_an VARCHAR2(10) NOT NULL,

an NUMBER NOT NULL,

zi NUMBER NOT NULL,

luna NUMBER NOT NULL,

nume\_luna VARCHAR2(20) NOT NULL);

CREATE TABLE meci(

id\_meci NUMBER,

data\_ora DATE NOT NULL,

scor VARCHAR2(20),

nume\_juc1 VARCHAR2(25),

prenume\_juc1 VARCHAR2(25),

nume\_juc2 VARCHAR2(25),

prenume\_juc2 VARCHAR2(25),

clasament\_juc1 NUMBER(4),

clasament\_juc2 NUMBER(4),

tara\_juc1 VARCHAR2(50),

tara\_juc2 VARCHAR2(50),

tur VARCHAR2(20) NOT NULL,

teren VARCHAR2(20) NOT NULL,

nume\_arbitru VARCHAR2(25),

prenume\_arbitru VARCHAR2(25),

pret NUMBER NOT NULL);

CREATE TABLE bilet(

id\_bilet NUMBER,

id\_meci NUMBER NOT NULL,

id\_user NUMBER NOT NULL,

id\_regiune NUMBER NOT NULL,

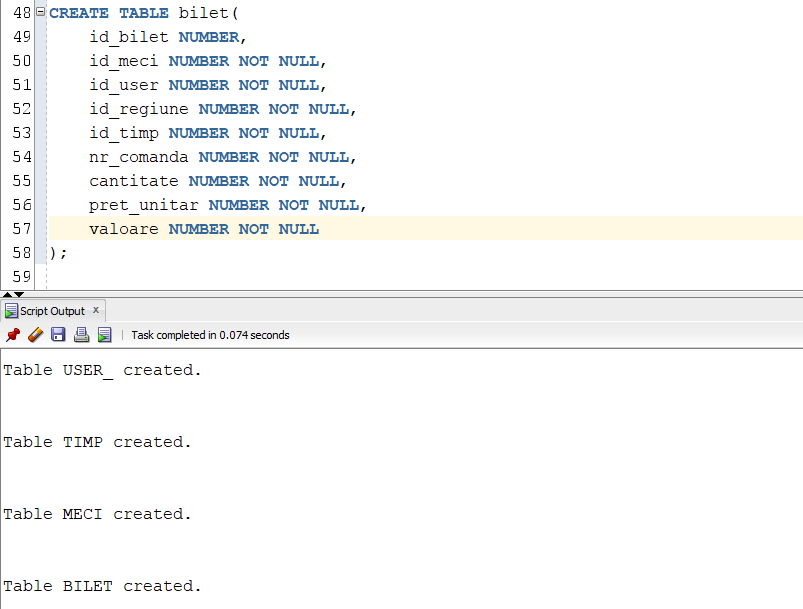
id\_timp NUMBER NOT NULL,

nr\_comanda NUMBER NOT NULL,

cantitate NUMBER NOT NULL,

pret\_unitar NUMBER NOT NULL,

valoare NUMBER NOT NULL);



# [Popularea cu informații a bazei de date depozit folosind ca sursă datele din baza de date OLTP](#_limr38x5fdhs)

-- acordare de privilegii SELECT pentru userul OLAP

BEGIN

FOR i IN (SELECT table\_name

FROM user\_tables

WHERE lower(table\_name) IN ('regiune', 'subregiune',

'tara', 'tur', 'teren',

'arbitru', 'jucator',

'meci', 'bilet', 'user\_')) LOOP

EXECUTE IMMEDIATE 'GRANT SELECT ON ' || i.table\_name || ' TO dwbi\_olap';

END LOOP;

END;

/

INSERT INTO regiune

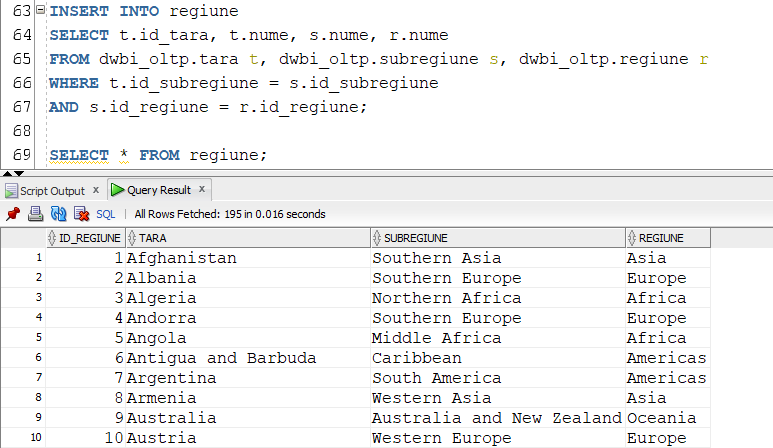
SELECT t.id\_tara, t.nume, s.nume, r.nume

FROM dwbi\_oltp.tara t, dwbi\_oltp.subregiune s, dwbi\_oltp.regiune r

WHERE t.id\_subregiune = s.id\_subregiune

AND s.id\_regiune = r.id\_regiune;

SELECT \* FROM regiune;

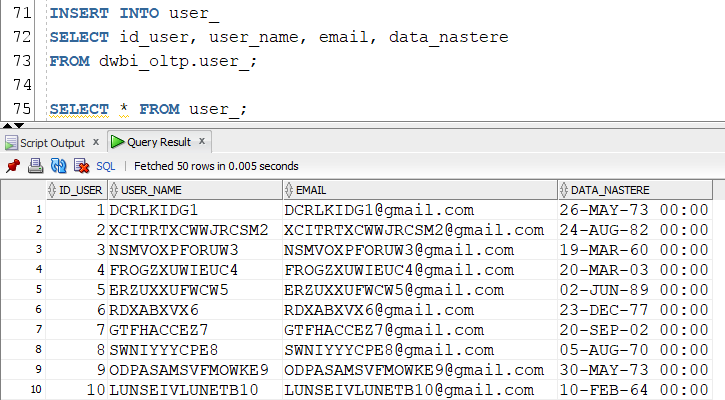


INSERT INTO user\_

SELECT id\_user, user\_name, email, data\_nastere

FROM dwbi\_oltp.user\_;

SELECT \* FROM user\_;



INSERT INTO timp

SELECT to\_char(data, 'j') id\_timp, data data,

to\_char(data, 'yyyy-mm-w') sapt\_luna\_an,

CASE

WHEN EXTRACT(MONTH FROM data)<7 THEN to\_char(data, 'yyyy-') || '1'

ELSE to\_char(data, 'yyyy-') || '2'

END semestru\_an,

to\_char(data, 'yyyy-mm') luna\_an,

EXTRACT(YEAR FROM data) an,

EXTRACT(DAY FROM data) zi,

EXTRACT(MONTH FROM data) luna,

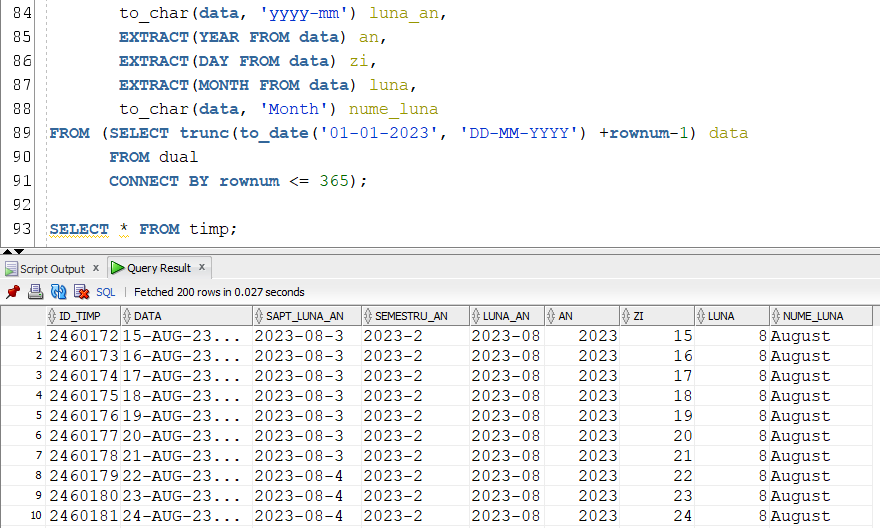
to\_char(data, 'Month') nume\_luna

FROM (SELECT trunc(to\_date('01-01-2023', 'DD-MM-YYYY') +rownum-1) data

FROM dual

CONNECT BY rownum <= 365);

SELECT \* FROM timp;



INSERT INTO meci

SELECT id\_meci, data\_ora, scor, j1.nume, j1.prenume, j2.nume, j2.prenume,

j1.clasament, j2.clasament, t1.nume, t2.nume, tu.denumire,

te.denumire, a.nume, a.prenume, tu.pret

FROM dwbi\_oltp.meci m, dwbi\_oltp.jucator j1, dwbi\_oltp.jucator j2,

dwbi\_oltp.tara t1, dwbi\_oltp.tara t2, dwbi\_oltp.tur tu,

dwbi\_oltp.teren te, dwbi\_oltp.arbitru a

WHERE m.id\_jucator1 = j1.id\_jucator

AND m.id\_jucator2 = j2.id\_jucator

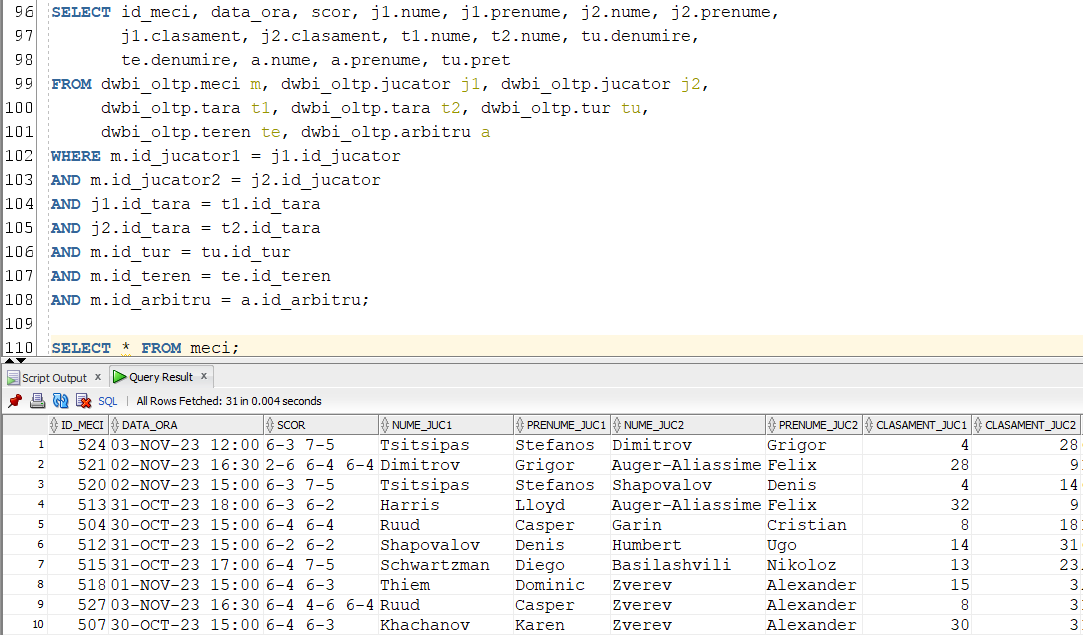
AND j1.id\_tara = t1.id\_tara

AND j2.id\_tara = t2.id\_tara

AND m.id\_tur = tu.id\_tur

AND m.id\_teren = te.id\_teren

AND m.id\_arbitru = a.id\_arbitru;

SELECT \* FROM meci;

INSERT INTO bilet

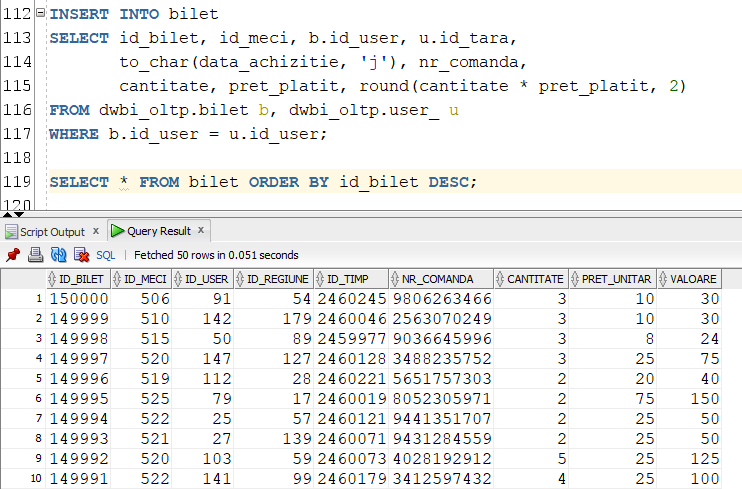
SELECT id\_bilet, id\_meci, b.id\_user, u.id\_tara,

to\_char(data\_achizitie, 'j'), nr\_comanda,

cantitate, pret\_platit, round(cantitate \* pret\_platit, 2)

FROM dwbi\_oltp.bilet b, dwbi\_oltp.user\_ u

WHERE b.id\_user = u.id\_user;

SELECT \* FROM bilet ORDER BY id\_bilet DESC;

# [Definirea constrângerilor](#_4tl22xv4a1am)

-- chei primare definite implicit cu ENABLE VALIDATE si optiunea RELY

ALTER TABLE regiune

ADD CONSTRAINT pk\_regiune

PRIMARY KEY (id\_regiune) RELY;

ALTER TABLE user\_

ADD CONSTRAINT pk\_user

PRIMARY KEY (id\_user) RELY;

ALTER TABLE timp

ADD CONSTRAINT pk\_timp

PRIMARY KEY (id\_timp) RELY;

ALTER TABLE meci

ADD CONSTRAINT pk\_meci

PRIMARY KEY (id\_meci) RELY;

-- cheie primara definita cu RELY DISABLE NOVALIDATE

ALTER TABLE bilet

ADD CONSTRAINT pk\_bilet PRIMARY KEY (id\_bilet)

RELY DISABLE NOVALIDATE;

-- chei externe definite cu RELY DISABLE NOVALIDATE

ALTER TABLE bilet

ADD CONSTRAINT fk\_bilet\_meci FOREIGN KEY (id\_meci)

REFERENCES meci(id\_meci)

RELY DISABLE NOVALIDATE;

ALTER TABLE bilet

ADD CONSTRAINT fk\_bilet\_user FOREIGN KEY (id\_user)

REFERENCES user\_(id\_user)

RELY DISABLE NOVALIDATE;

ALTER TABLE bilet

ADD CONSTRAINT fk\_bilet\_regiune FOREIGN KEY (id\_regiune)

REFERENCES regiune(id\_regiune)

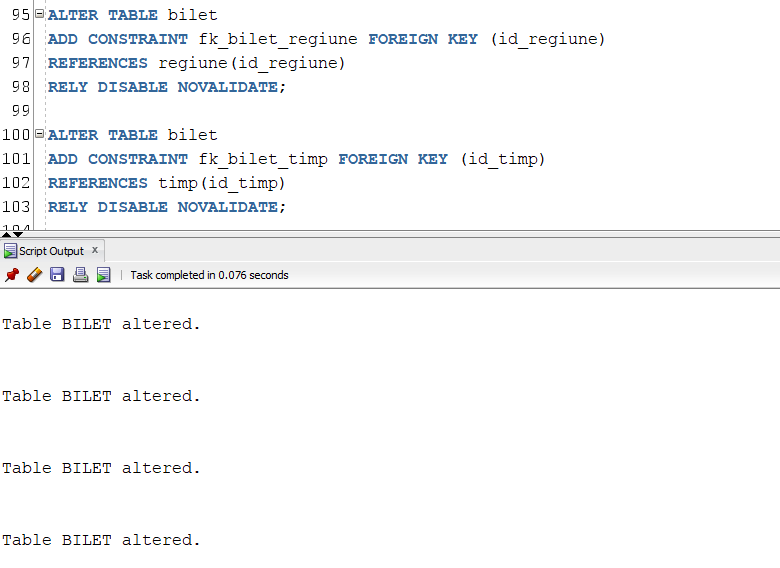
RELY DISABLE NOVALIDATE;

ALTER TABLE bilet

ADD CONSTRAINT fk\_bilet\_timp FOREIGN KEY (id\_timp)

REFERENCES timp(id\_timp)

RELY DISABLE NOVALIDATE;



# [Definirea indecșilor și a cererilor SQL însoțite de planul de execuție al acestora (din care să reiasă că optimizatorul utilizează eficient indecșii definiți)](#_ugnoy9o1ceg3)

CREATE BITMAP INDEX bmp\_meci\_tur

ON meci(tur);

ANALYZE INDEX bmp\_meci\_tur COMPUTE STATISTICS;

SELECT nr\_3\_tur2 nr\_3seturi\_tur2,

ROUND(nr\_3\_tur2 / (SELECT COUNT(\*)

FROM meci) \* 100, 2) || '%' procent

FROM

(SELECT COUNT(\*) nr\_3\_tur2

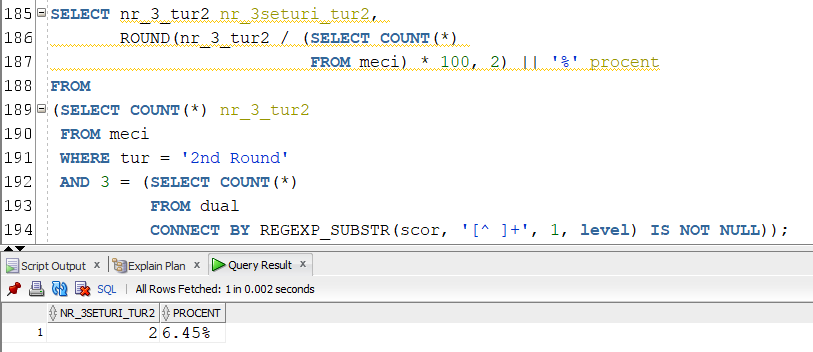
FROM meci

WHERE tur = '2nd Round'

AND 3 = (SELECT COUNT(\*)

FROM dual

CONNECT BY REGEXP\_SUBSTR(scor, '[^ ]+', 1, level) IS NOT NULL));



EXPLAIN PLAN

SET STATEMENT\_ID = 'ex6'

FOR

SELECT nr\_3\_tur2 nr\_3seturi\_tur2,

ROUND(nr\_3\_tur2 / (SELECT COUNT(\*)

FROM meci) \* 100, 2) || '%' procent

FROM

(SELECT COUNT(\*) nr\_3\_tur2

FROM meci

WHERE tur = '2nd Round'

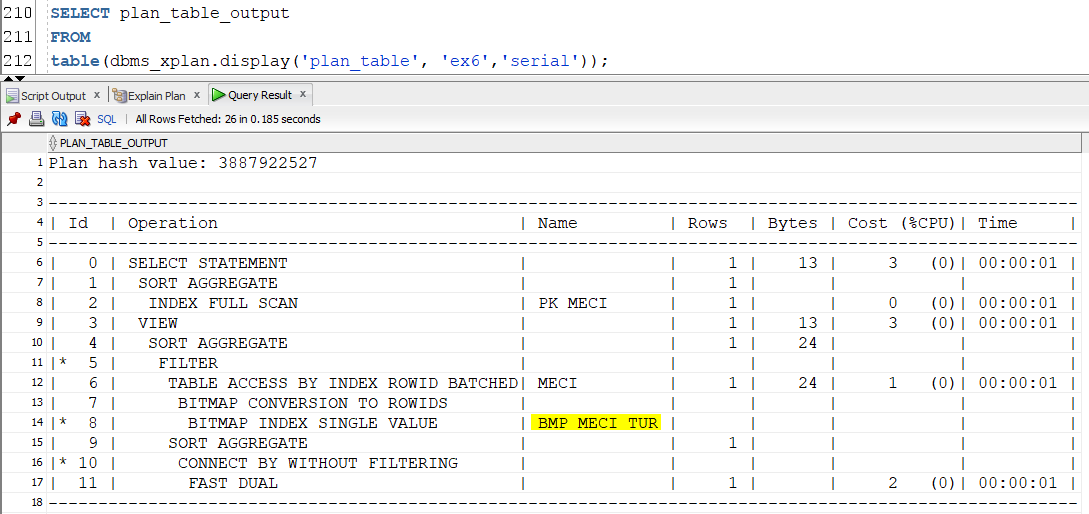
AND 3 = (SELECT COUNT(\*) FROM dual

CONNECT BY REGEXP\_SUBSTR(scor, '[^ ]+', 1, level) IS NOT NULL));

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 'ex6','serial'));



# [Definirea obiectelor de tip dimensiune, validarea acestora (din care să reiasă că datele respectă constrângerile impuse prin aceste tipuri de obiecte](#_vhs81gdiijlt)

CREATE DIMENSION timp\_dim

LEVEL saptamana IS (timp.sapt\_luna\_an)

LEVEL luna IS (timp.luna\_an)

LEVEL semestru IS (timp.semestru\_an)

LEVEL an IS (timp.an)

HIERARCHY h1

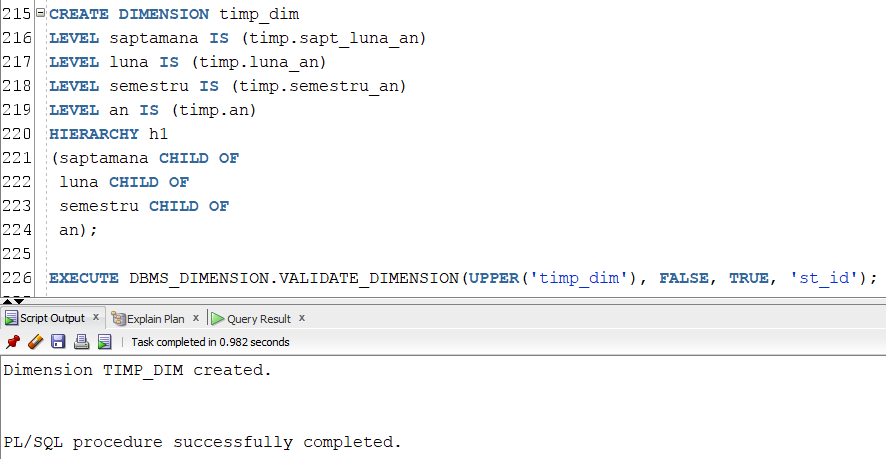
(saptamana CHILD OF

luna CHILD OF

semestru CHILD OF

an);

EXECUTE DBMS\_DIMENSION.VALIDATE\_DIMENSION(UPPER('timp\_dim'), FALSE, TRUE, 'st\_id');



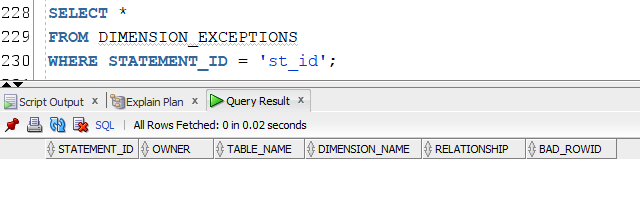
SELECT \*

FROM DIMENSION\_EXCEPTIONS

WHERE STATEMENT\_ID = 'st\_id';

Observăm că interogarea tabelei sistem DIMENSION\_EXCEPTIONS nu ne întoarce nicio linie, semn că datele din coloanele pe baza cărora s-a creat ierarhia obiectului dimension respectă toate constrângerile impuse.

Cu toate că tabelul timp conține doar zilele din anul 2023, deoarece coloanele *sapt\_luna\_an*, *luna\_an* și *semestru\_an* conțin și informația legată de an, adăugarea zilelor din alți ani nu va încălca regula că fiecare copil trebuie să aibă un singur părinte.



# [Definirea partițiilor; definirea cererilor SQL însoțite de planul de execuție al acestora din care să reiasă că optimizorul utilizează eficient partițiile.](#_nl7x8dfqew66)

CREATE TABLE bilet\_ord(

id\_bilet NUMBER,

id\_meci NUMBER NOT NULL,

id\_user NUMBER NOT NULL,

id\_regiune NUMBER NOT NULL,

id\_timp NUMBER NOT NULL,

nr\_comanda NUMBER NOT NULL,

cantitate NUMBER NOT NULL,

pret\_unitar NUMBER NOT NULL,

valoare NUMBER NOT NULL

)

PARTITION BY RANGE(id\_timp)

(

PARTITION bilete\_sub2023

VALUES LESS THAN (2459946),

-- 2459946 = TO\_CHAR(TO\_DATE('01/01/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_jan2023

VALUES LESS THAN (2459977),

-- 2459977 = TO\_CHAR(TO\_DATE('01/02/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_feb2023

VALUES LESS THAN (2460005),

-- 2460005 = TO\_CHAR(TO\_DATE('01/03/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_mar2023

VALUES LESS THAN (2460036),

-- 2460036 = TO\_CHAR(TO\_DATE('01/04/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_apr2023

VALUES LESS THAN (2460066),

-- 2460066 = TO\_CHAR(TO\_DATE('01/05/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_mai2023

VALUES LESS THAN (2460097),

-- 2460097 = TO\_CHAR(TO\_DATE('01/06/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_iun2023

VALUES LESS THAN (2460127),

-- 2460127 = TO\_CHAR(TO\_DATE('01/07/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_iul2023

VALUES LESS THAN (2460158),

-- 2460158 = TO\_CHAR(TO\_DATE('01/08/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_aug2023

VALUES LESS THAN (2460189),

-- 2460189 = TO\_CHAR(TO\_DATE('01/09/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_sept2023

VALUES LESS THAN (2460219),

-- 2460219 = TO\_CHAR(TO\_DATE('01/10/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_oct2023

VALUES LESS THAN (2460250),

-- 2460250 = TO\_CHAR(TO\_DATE('01/11/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_nov2023

VALUES LESS THAN (2460280),

-- 2460280 = TO\_CHAR(TO\_DATE('01/12/2023','DD/MM/YYYY'),'J')

PARTITION bilete\_dec2023

VALUES LESS THAN (2460311),

-- 2460311 = TO\_CHAR(TO\_DATE('01/01/2024','DD/MM/YYYY'),'J')

PARTITION bilete\_rest

VALUES LESS THAN (MAXVALUE)

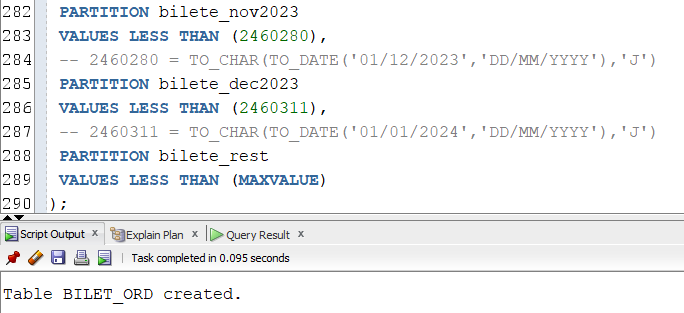
);

INSERT INTO bilet\_ord

SELECT \* FROM bilet;

commit;

ANALYZE TABLE bilet\_ord COMPUTE STATISTICS;



SELECT m.nume\_juc1 || ' vs. ' || m.nume\_juc2 meci, SUM(cantitate), SUM(valoare)

FROM bilet\_ord b, meci m

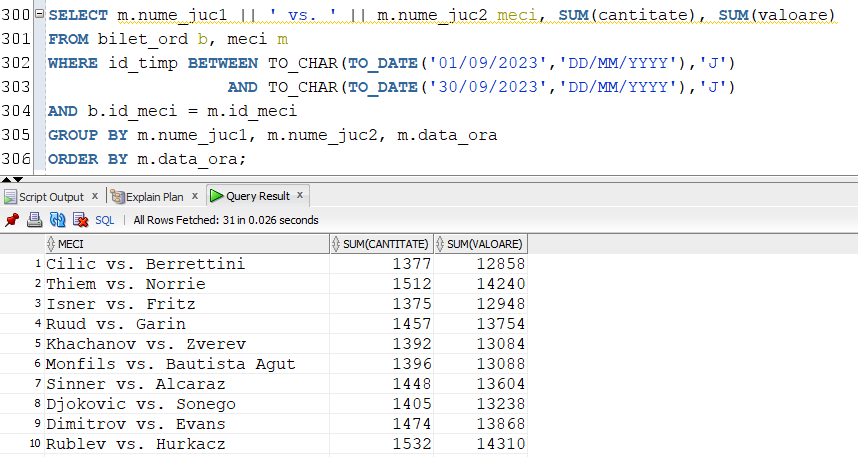
WHERE id\_timp BETWEEN TO\_CHAR(TO\_DATE('01/09/2023','DD/MM/YYYY'),'J')

AND TO\_CHAR(TO\_DATE('30/09/2023','DD/MM/YYYY'),'J')

AND b.id\_meci = m.id\_meci

GROUP BY m.nume\_juc1, m.nume\_juc2, m.data\_ora

ORDER BY m.data\_ora;



EXPLAIN PLAN

SET STATEMENT\_ID = 'ex8'

FOR

SELECT m.nume\_juc1 || ' vs. ' || m.nume\_juc2 meci, SUM(cantitate), SUM(valoare)

FROM bilet\_ord b, meci m

WHERE id\_timp BETWEEN TO\_CHAR(TO\_DATE('01/09/2023','DD/MM/YYYY'),'J')

AND TO\_CHAR(TO\_DATE('30/09/2023','DD/MM/YYYY'),'J')

AND b.id\_meci = m.id\_meci

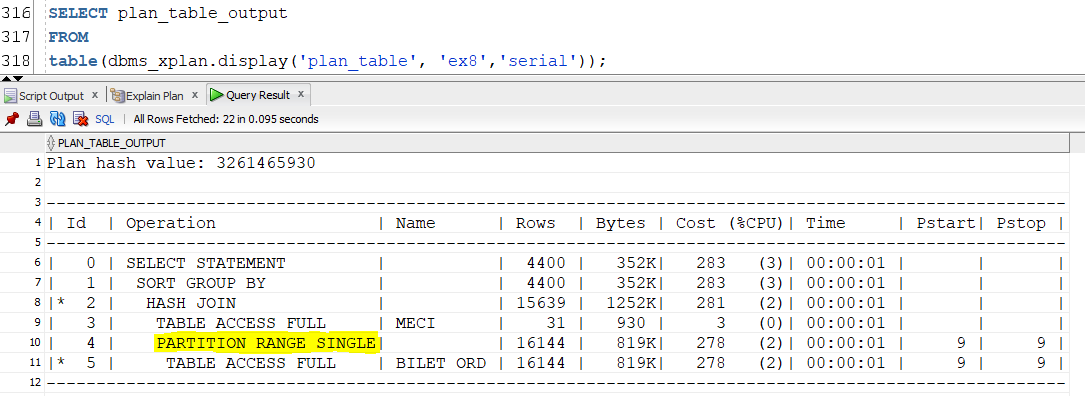
GROUP BY m.nume\_juc1, m.nume\_juc2, m.data\_ora

ORDER BY m.data\_ora;

SELECT plan\_table\_output

FROM

table(dbms\_xplan.display('plan\_table', 'ex8','serial'));



# [Optimizarea cererii SQL propusă în etapa de analiză](#_3uj5n8cuievd)

WITH aux AS

(SELECT tara, luna, nume\_luna, an, sum(cantitate) bilete\_luna

FROM bilet b, regiune r, timp t

WHERE b.id\_regiune = r.id\_regiune

AND b.id\_timp = t.id\_timp

GROUP BY tara, nume\_luna, luna, an)

SELECT tara, nume\_luna, bilete\_luna

FROM aux a

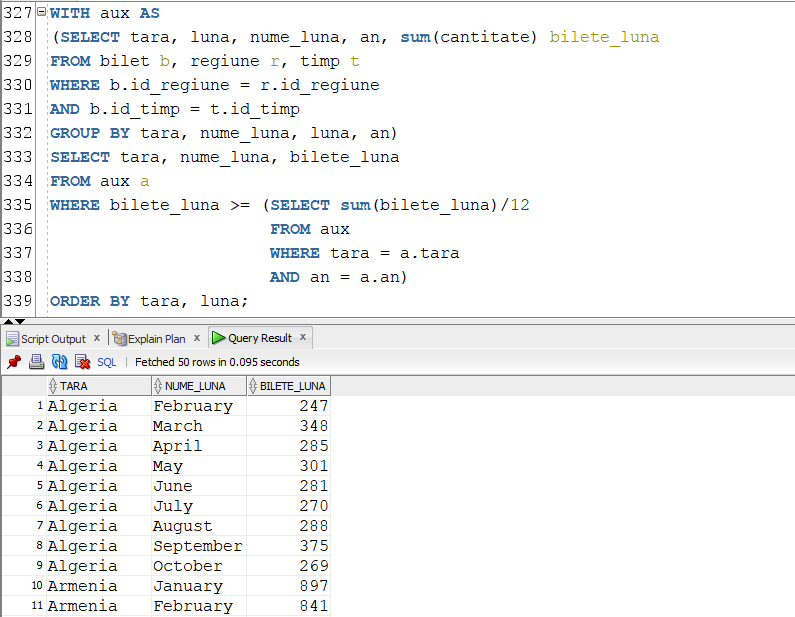
WHERE bilete\_luna >= (SELECT sum(bilete\_luna)/12

FROM aux

WHERE tara = a.tara

AND an = a.an)

ORDER BY tara, luna;



# planul de execuție ales de optimizorul bazat pe cost (explicație etape parcurse)

EXPLAIN PLAN

SET STATEMENT\_ID = 'ex9\_neoptimizat'

FOR

WITH aux AS

(SELECT tara, luna, nume\_luna, an, sum(cantitate) bilete\_luna

FROM bilet b, regiune r, timp t

WHERE b.id\_regiune = r.id\_regiune

AND b.id\_timp = t.id\_timp

GROUP BY tara, nume\_luna, luna, an)

SELECT tara, nume\_luna, bilete\_luna

FROM aux a

WHERE bilete\_luna >= (SELECT sum(bilete\_luna)/12

FROM aux

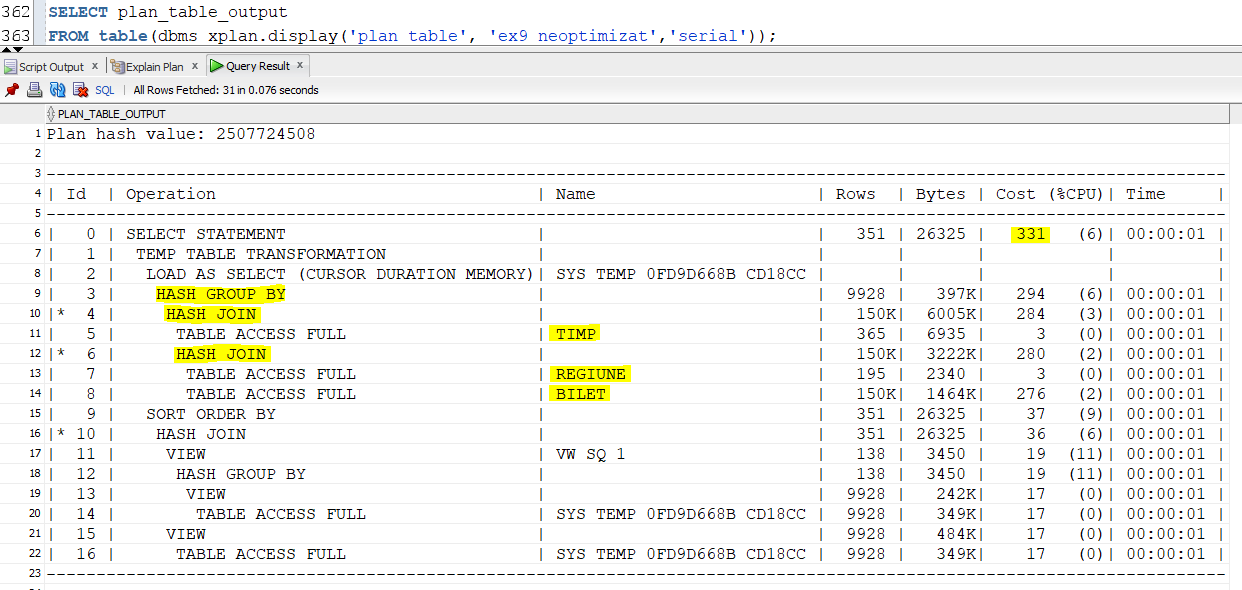
WHERE tara = a.tara

AND an = a.an)

ORDER BY tara, luna;

SELECT plan\_table\_output

FROM table(dbms\_xplan.display('plan\_table', 'ex9\_neoptimizat','serial'));



Observăm din planul de execuție că în blocul WITH este efectuat un HASH JOIN între tabelele REGIUNE și BILET, iar apoi încă un HASH JOIN între rezultat și tabelul TIMP. Ulterior s-a efectuat gruparea după cele 4 coloane, iar vizualizarea temporară creată de blocul WITH a fost accesată complet în cadrul subcererii sincronizate.

**Costul cererii este de 331.**

# sugestii de optimizare a cererii, specificând planul de execuție obținut

Pentru optimizare vom crea o vizualizare materializată cu opțiunea de rescriere de cereri activată ce stochează datele calculate precedent în blocul WITH. Ulterior vom calcula statistici asupra tabelelor și vizualizării.

CREATE MATERIALIZED VIEW vm\_bilete\_timp\_regiuni

BUILD IMMEDIATE

REFRESH COMPLETE

ON COMMIT

ENABLE QUERY REWRITE

AS

SELECT tara, luna, nume\_luna, an, sum(cantitate) bilete\_luna

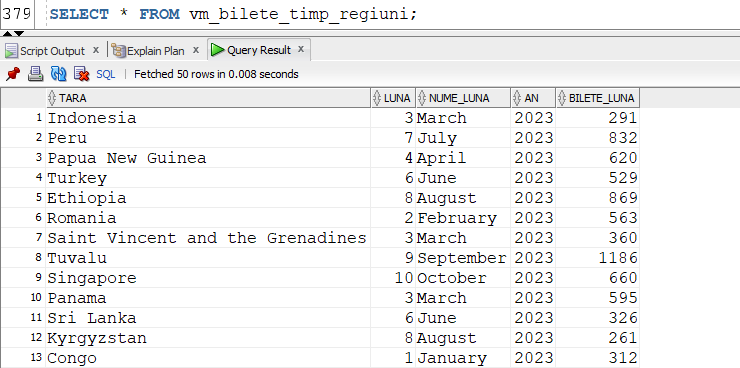
FROM bilet b, regiune r, timp t

WHERE b.id\_regiune = r.id\_regiune

AND b.id\_timp = t.id\_timp

GROUP BY tara, nume\_luna, luna, an;

SELECT \* FROM vm\_bilete\_timp\_regiuni;



ANALYZE TABLE timp COMPUTE STATISTICS;

ANALYZE TABLE regiune COMPUTE STATISTICS;

ANALYZE TABLE bilet COMPUTE STATISTICS;

ANALYZE TABLE vm\_bilete\_timp\_regiuni COMPUTE STATISTICS;

EXPLAIN PLAN

SET STATEMENT\_ID = 'ex9\_optimizat'

FOR

WITH aux AS

(SELECT tara, luna, nume\_luna, an, sum(cantitate) bilete\_luna

FROM bilet b, regiune r, timp t

WHERE b.id\_regiune = r.id\_regiune

AND b.id\_timp = t.id\_timp

GROUP BY tara, nume\_luna, luna, an)

SELECT tara, nume\_luna, bilete\_luna

FROM aux a

WHERE bilete\_luna >= (SELECT sum(bilete\_luna)/12

FROM aux

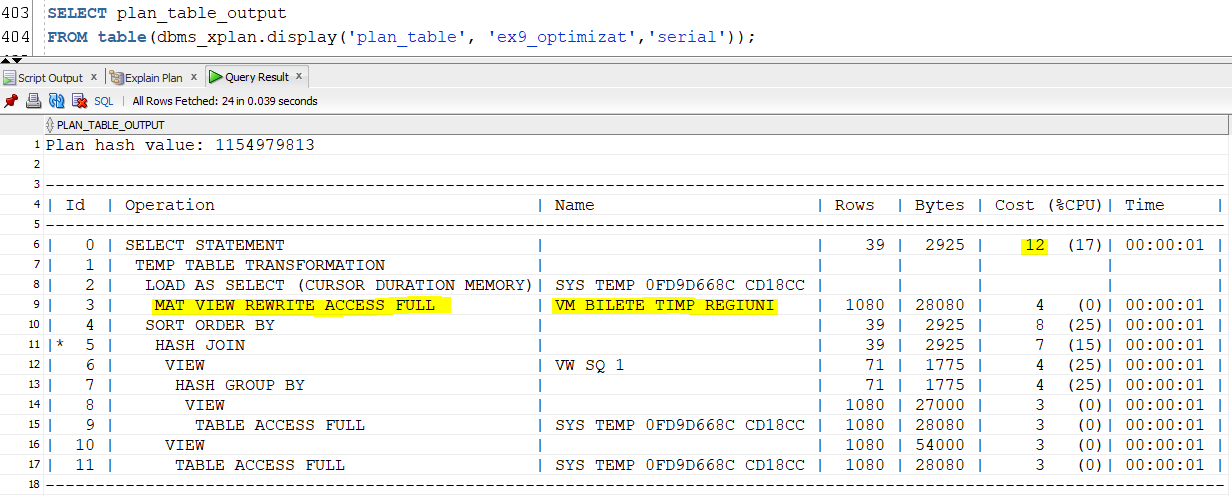
WHERE tara = a.tara

AND an = a.an)

ORDER BY tara, luna;

SELECT plan\_table\_output

FROM table(dbms\_xplan.display('plan\_table', 'ex9\_optimizat','serial'));



Observăm din planul de execuție că optimizatorul rescrie cererea prin accesarea vizualizării materializate în locul tabelelor de bază, celelalte operații fiind executate în același mod.

**Costul cererii a scăzut semnificativ la 12.**

# [Crearea rapoartelor cu complexitate diferită (la acest nivel vor fi scripturi SQL, fără reprezentare grafică)](#_2bmyh9c7pvbt)

-- 1. Afisati pentru fiecare tara top 3 meciuri pentru care utilizatorii au achizitionat bilete in anul 2023.

SELECT \* FROM (

SELECT r.tara, m.nume\_juc1, m.nume\_juc2,

SUM(b.cantitate) cantitate\_totala,

SUM(b.valoare) valoare\_totala,

DENSE\_RANK() OVER (PARTITION BY r.tara

ORDER BY SUM(b.cantitate) DESC) d\_rank\_desc

FROM bilet b

JOIN meci m ON m.id\_meci = b.id\_meci

JOIN timp t ON t.id\_timp = b.id\_timp

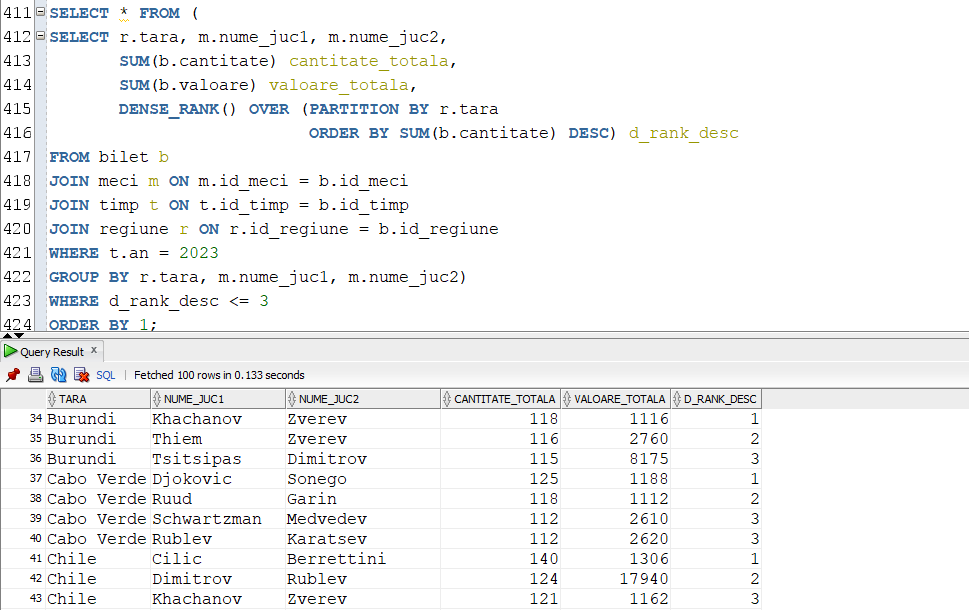
JOIN regiune r ON r.id\_regiune = b.id\_regiune

WHERE t.an = 2023

GROUP BY r.tara, m.nume\_juc1, m.nume\_juc2)

WHERE d\_rank\_desc <= 3

ORDER BY 1;



-- 2. Afisati numarul de bilete achizitionate in semestrul 1 al anului 2023:

-- - pentru fiecare regiune, teren si tur

-- - pentru fiecare regiune si teren, indiferent de tur.

SELECT r.regiune, m.teren, m.tur, SUM(b.cantitate) AS total\_bilete,

GROUPING(r.regiune) AS g\_regiune, GROUPING(m.teren) AS g\_teren,

GROUPING(m.tur) AS g\_tur, GROUPING\_ID(r.regiune, m.teren, m.tur) AS gid

FROM bilet b JOIN meci m ON m.id\_meci = b.id\_meci

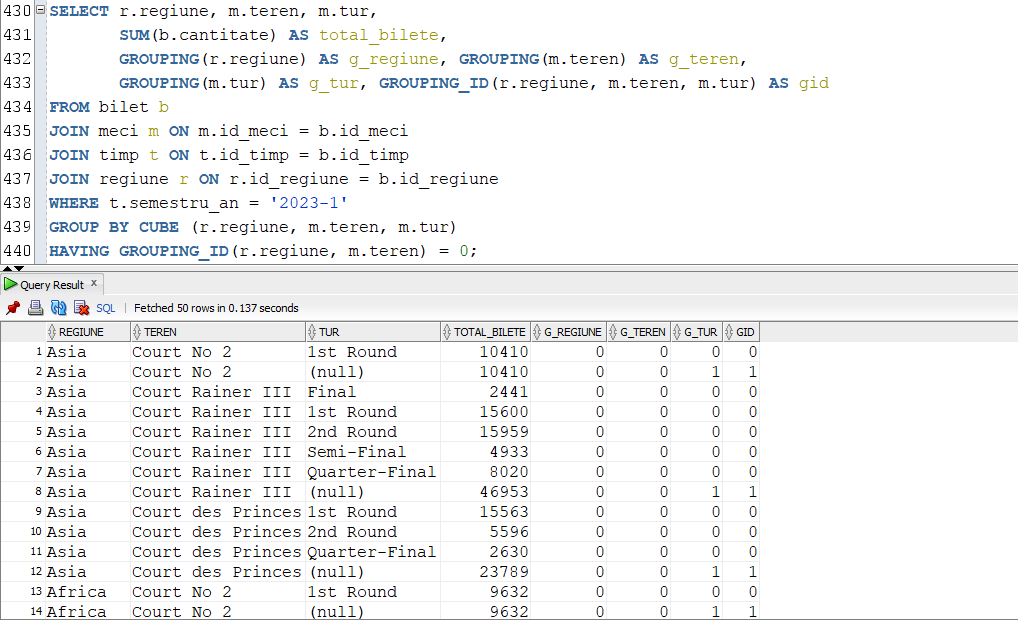
JOIN timp t ON t.id\_timp = b.id\_timp

JOIN regiune r ON r.id\_regiune = b.id\_regiune

WHERE t.semestru\_an = '2023-1'

GROUP BY CUBE (r.regiune, m.teren, m.tur)

HAVING GROUPING\_ID(r.regiune, m.teren) = 0;



-- 3. Urmariti profitul de la luna la luna acumulat pentru fiecare meci in anul 2023.

SELECT id\_meci, luna, total\_valoare\_meci,

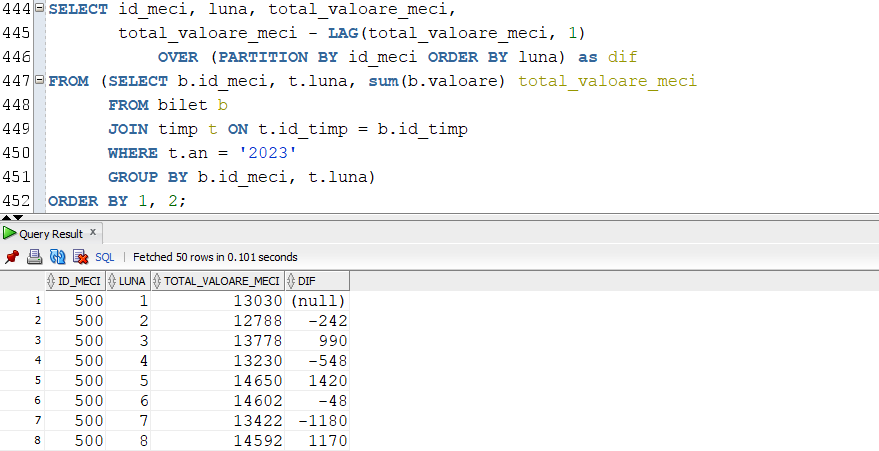
total\_valoare\_meci - LAG(total\_valoare\_meci, 1)

OVER (PARTITION BY id\_meci ORDER BY luna) as dif

FROM (SELECT b.id\_meci, t.luna, sum(b.valoare) total\_valoare\_meci

FROM bilet b JOIN timp t ON t.id\_timp = b.id\_timp

WHERE t.an = '2023' GROUP BY b.id\_meci, t.luna) ORDER BY 1, 2;



-- 4. Clasati tarile tinand cont de procentul de venit adus din totalul veniturilor in anul 2023.

SELECT r.tara,

SUM(b.valoare) AS VANZARI,

SUM(SUM(b.valoare)) OVER () AS TOTAL\_VANZ,

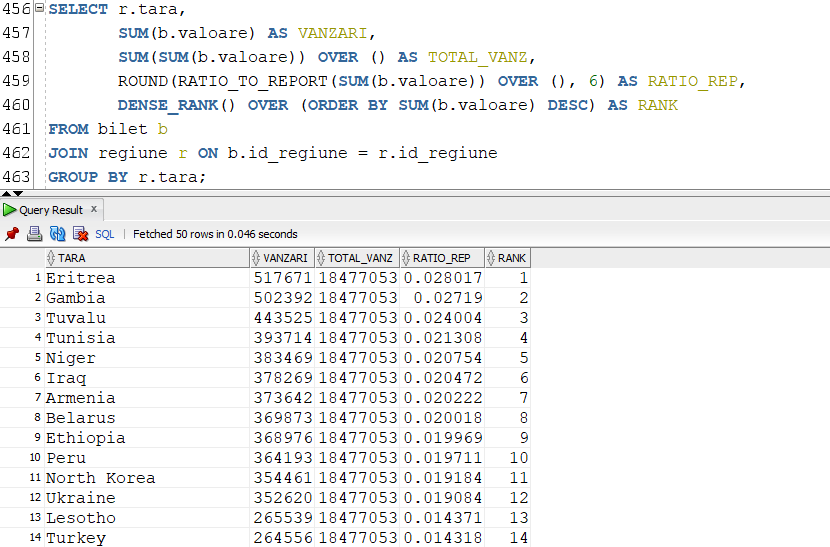
ROUND(RATIO\_TO\_REPORT(SUM(b.valoare)) OVER (), 6) AS RATIO\_REP,

DENSE\_RANK() OVER (ORDER BY SUM(b.valoare) DESC) AS RANK

FROM bilet b

JOIN regiune r ON b.id\_regiune = r.id\_regiune

GROUP BY r.tara;



-- 5. Afisati media vanzarilor in ultimele 2 luni, luna curenta si luna urmatoare ale anului 2023 pentru fiecare meci.

SELECT b.id\_meci, t.luna, SUM(b.valoare) valoare,

ROUND(AVG(SUM(b.valoare))

OVER (PARTITION BY b.id\_meci

ORDER BY b.id\_meci, t.luna

ROWS BETWEEN 2 PRECEDING AND 1 FOLLOWING), 2)

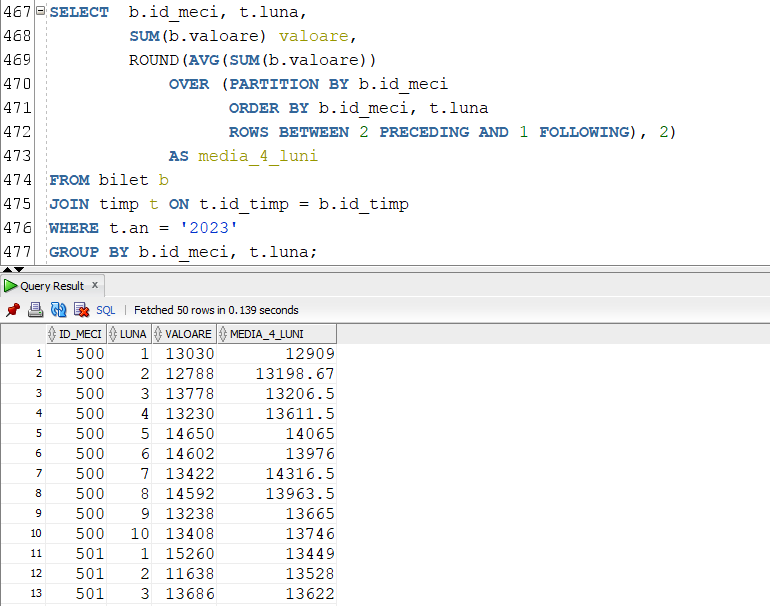
AS media\_4\_luni

FROM bilet b

JOIN timp t ON t.id\_timp = b.id\_timp

WHERE t.an = '2023'

GROUP BY b.id\_meci, t.luna;

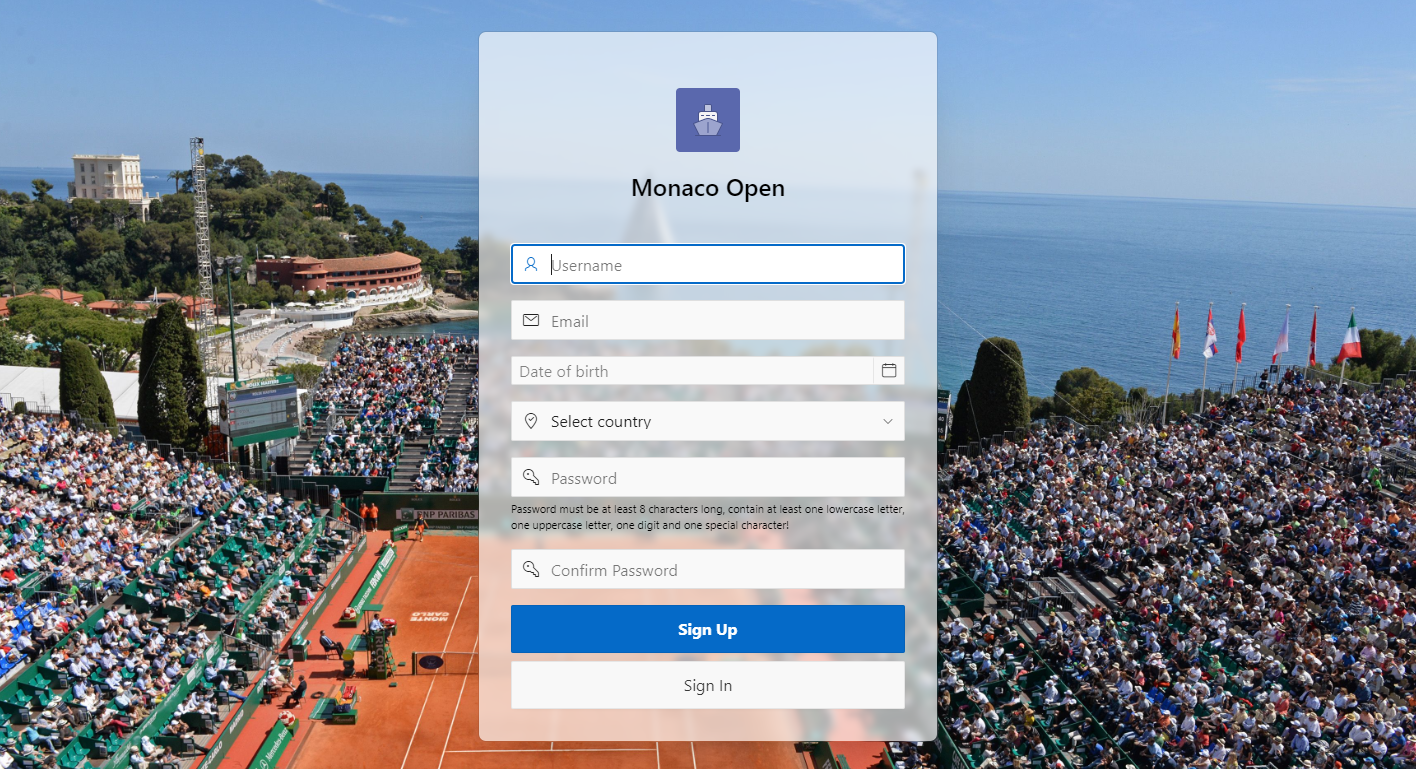


**MODUL FRONT-END**

Pe baza modelului prezentat a fost creată o aplicație web în Oracle Apex prin intermediul căreia utilizatorii pot vedea informații despre turneu, jucători și partide, își pot cumpăra bilete la meciuri, pot posta subiecte și comentarii pe forumul de discuții și pot răspunde la întrebările quiz-urilor zilnice.

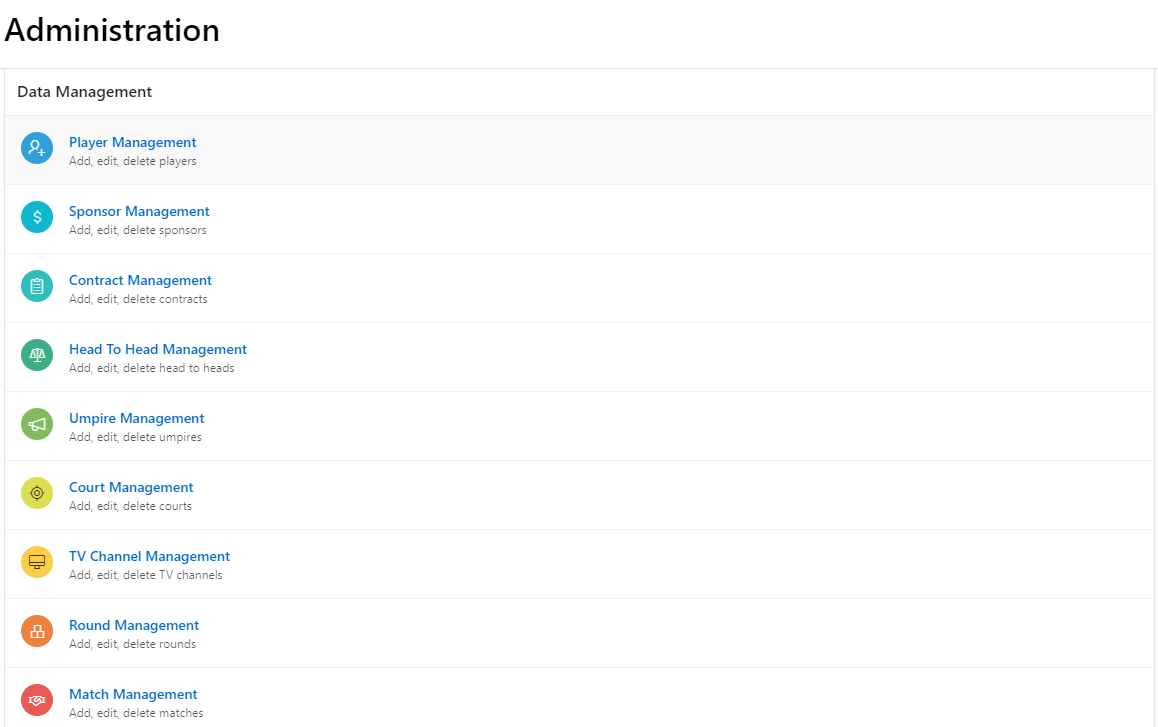
# Modul aplicație prin care se introduc și gestionează informații la nivelul bazei de date OLTP.

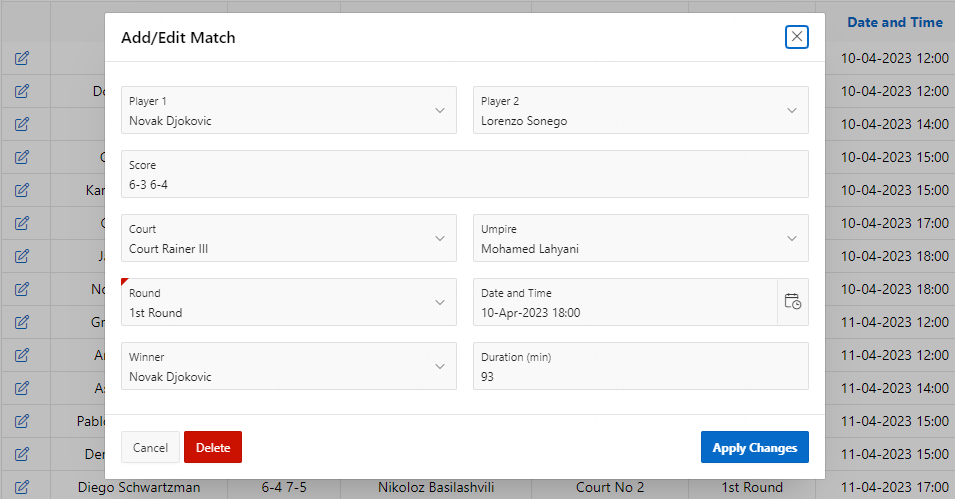
Pentru gestionarea informațiilor din tabelul USER utilizatorii neautentificați completează formularul de Sign Up pentru crearea contului ce le oferă acces la funcționalitățile personalizate precum cumpărarea biletelor.



Tabelele JUCATOR, ARBITRU, TUR, TEREN și MECI sunt gestionate de utilizatorii cu rol de admin ai aplicației. Pentru fiecare dintre acestea sunt puse la dispoziție formulare ce conțin validările corespunzătoare constrângerilor din baza de date OLTP prin intermediul căruia datele pot fi introduse sau actualizate.

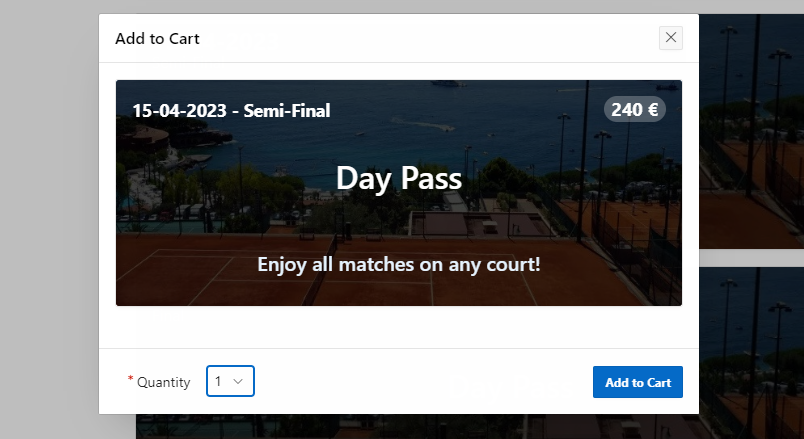
În imaginile de mai jos putem observa panoul de administrare al datelor din tabele și exemplul de formular pentru adăugarea/editarea unui meci, fiind puse la dispoziție pagini similare și pentru restul entităților.

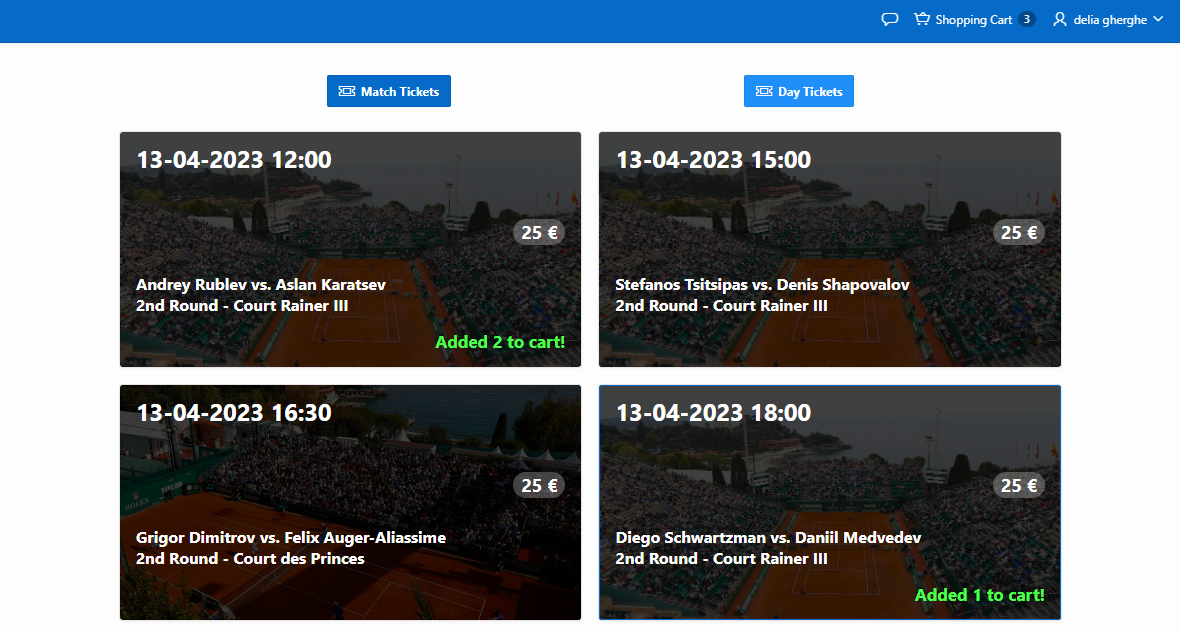




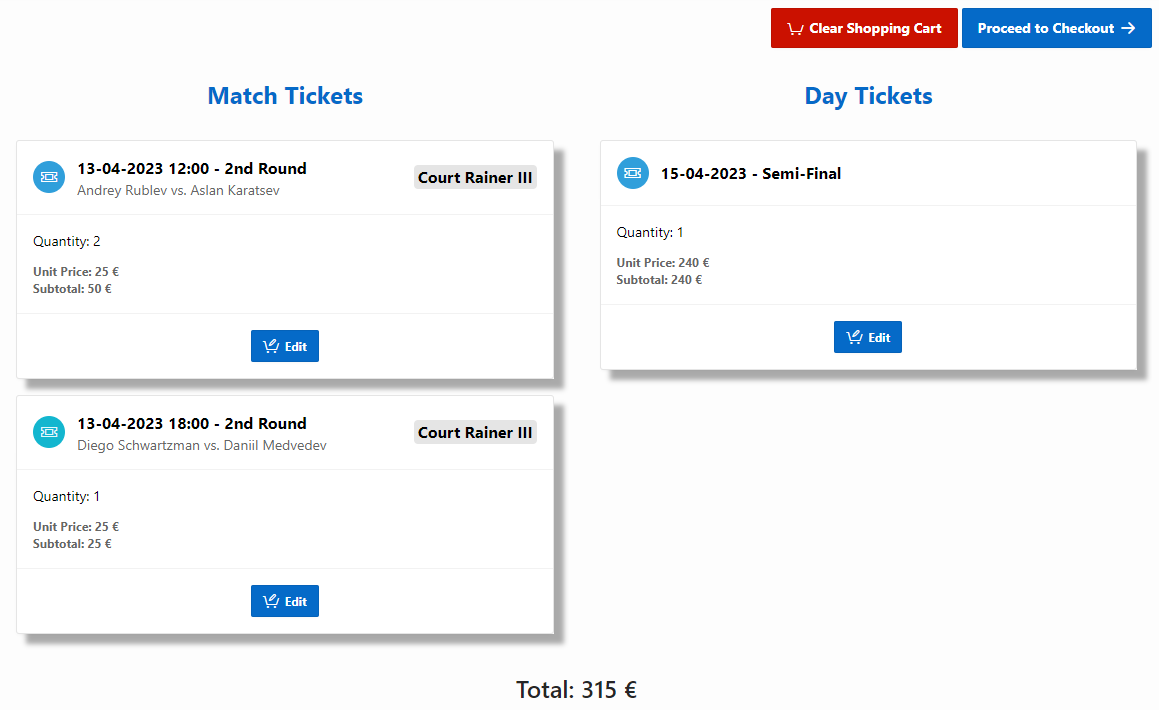
Biletele pot fi cumpărate de un utilizator doar pentru el însuși urmând procesul următor:

* adăugarea în coșul de cumpărături a biletelor și cantităților dorite
* vizualizarea și confirmarea coșului de cumpărături
* aplicarea discountului (dacă este disponibil)
* introducerea datelor de plată
* vizualizarea biletelor împreună cu codurile QR în pagina *My Tickets*

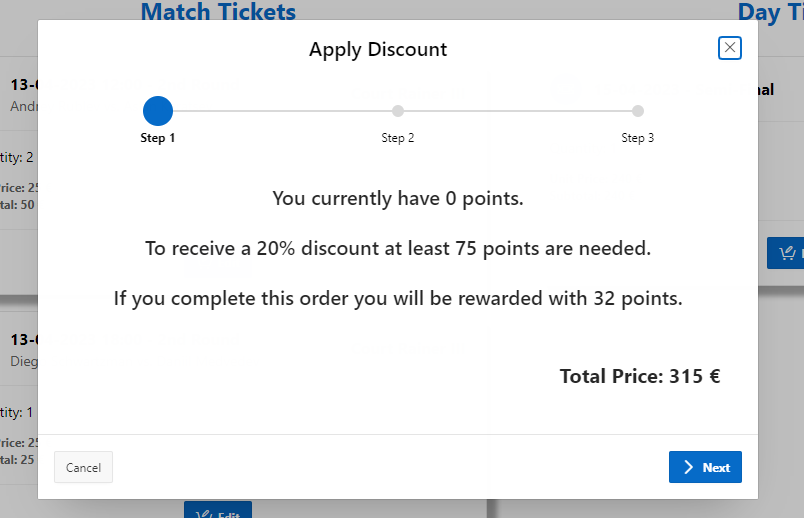




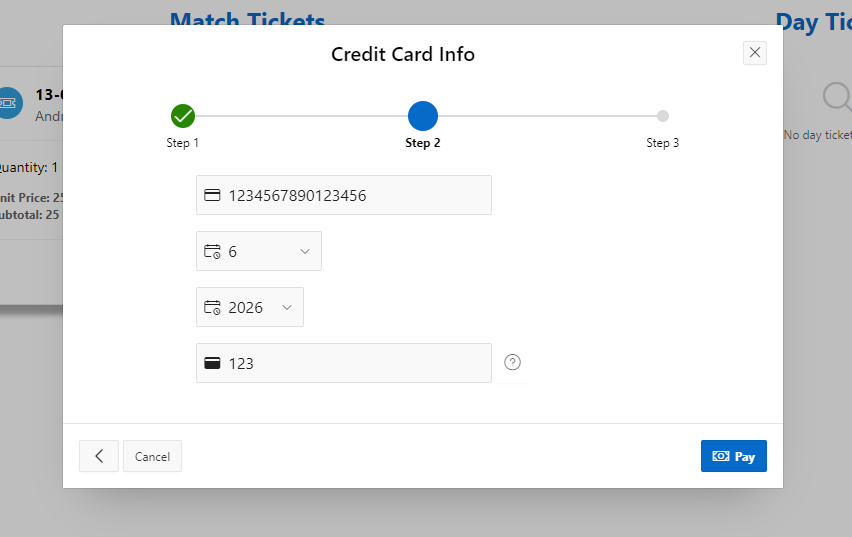
Adăugarea biletelor în coș



Vizualizarea coșului de cumpărături



Aplicarea discountului



Adăugarea datelor cardului de credit

Vizualizarea biletelor achiziționate