

Proiect la Baze de Date

Baza de Date a aplicației Chess.com

Realizat de Dobrin Ionuț-Alexandru

Grupa 134

Anul universitar 2022-2023

Cuprins

1. Descrierea modelului real
2. Prezentarea constrângerilor
3. Descrierea entităților
4. Descrierea relațiilor
5. Descrierea atributelor
6. Realizarea diagramei entitate-relație
7. Realizarea diagramei conceptuale
8. Enumerarea schemelor relaționale
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).
10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea
12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
 - subcereri sincronizate în care intervin cel puțin 3 tabele
 - subcereri nesincronizate în clauza FROM
 - grupări de date cu subcereri nesincronizate în care intervin cel puțin 3 tabele, funcții grup, filtrare la nivel de grupuri (în cadrul aceleiasi cereri)
 - ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiasi cereri)
 - utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
 - utilizarea a cel puțin 1 bloc de cerere (clauza WITH) Observație: Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.
14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele, o cerere ce utilizează operația division și o cerere care implementează analiza top-n. Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.
16. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile

operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebraică, arbore algebraic și limbaj (SQL), atât anterior cât și ulterior optimizării.

17. a. Realizarea normalizării BCNF, FN4, FN5.

b. Aplicarea denormalizării, justificând necesitatea acesteia.

1. Descrierea Modelului Real

În acest proiect voi prezenta cum arată implementarea “back-endului” aplicației Chess.com (<https://www.chess.com>) , dezvoltând baza de date din spatele aplicației. Până nu mult timp în urmă, bazele de date pentru șah erau reprezentate de cărți. Așa se studia șahul și aşa se “stocau” partidele memorabile jucate de foștii campioni mondiali și Mari Maeștri. În schimb, în secolul 21, odată cu apariția tehnologiei, s-a făcut ușor-ușor tranziția către baze de date online, unde pot fi accesate toate partidele celebre și nu numai. Orice partidă jucată online sau la un turneu oficial, poate fi introdusă într-o bază de date, la care au acces toți jucătorii interesați. Datorită frumuseții jocului, dar și a popularității acestuia, odată cu inițializarea primelor baze de date, au apărut și primele aplicații pe care utilizatorii pot să joace partide, să învețe făcând puzzle-uri, analizând mutările partidelor, să participe la turnee oficiale online, dar și să simtă un sentiment de competitivitate, orice partidă fiind cotată și având în consecință schimbarea rating-ului între 2 utilizatori. De asemenea, aplicația poate fi teoretic categorizată și sub forma unei aplicații sociale, deoarece un utilizator își poate face prieteni, poate să aibă conversații cu aceștia, dar poate să și primească notificări de exemplu pentru turnee sau poate să se uite la streamer-ul lor favorit cum se joacă. Nu în ultimul rând, pentru mai multe beneficii, un utilizator poate să opteze la un abonament plătit pentru a putea accesa tot ce poate să ofere aplicația.

2. Prezentarea Constrângерilor

- Un utilizator poate avea mai multe rating-uri, pentru fiecare timp de joc (de exemplu șah rapid, blitz, zilnic etc.) și nu trebuie să aibă neapărat niciun rating dacă nu a jucat absolut nicio partidă. Un rating poate fi avut de mai mulți utilizatori în același timp, sau poate de niciunul.
- Un utilizator poate avea mai multe conversații, nu doar cu prieteni, ci și cu alții utilizatori. Acestea de asemenea poate să nu aibă nicio conversație. O conversație se face între 2 utilizatori.
- Un utilizator poate să rezolve multe puzzle-uri, sau poate să nu rezolve niciunul. Un puzzle poate să fie rezolvat de mai mulți utilizatori, sau de niciunul.
- Un utilizator poate să aibe mai multe prietenii, dar nu este obligat să aibă nici măcar una. O prietenie este între 2 utilizatori.
- Un utilizator poate să participe la mai multe turnee, dar nu trebuie să participe la niciunul. Un turneu poate avea mulți utilizatori, dar măcar 2.

- Un utilizator poate să aibă mai multe notificări, dar nu trebuie să aibă niciuna. O notificare poate fi direcționată către mulți utilizatori, și chiar unul.
- O notificare are exact un mesaj, și un mesaj este în componență unui singur mesaj.
- Un utilizator poate să facă mai multe plăți, dar pentru că șahul e un joc pentru toți, acesta nu e obligat să facă nicio plată. O plată este făcută de exact un singur utilizator.
- Un utilizator poate să joace multe partide, dar nu e și obligat să joace vreo partidă. O partidă este jucată de exact 2 utilizatori.
- O partidă poate avea analiză dar trebuie să aibă mutări. O analiză se poate întâmpla la fiecare mutare din partidă. O mutare face parte dintr-o anumită partidă și dintr-o anumită analiză.

3. Descrierea Entităților

Pentru modelul de date al aplicației Chess.com, structurile UTILIZATORI, RATING, CONVERSAȚII, PUZZLES, PRIETENIE, TURNEE, NOTIFICĂRI, MESAJ, PLĂȚI, PARTIDE, MUTĂRI, ANALIZĂ reprezintă entități.

UTILIZATORI = Persoane care își fac cont pe aplicație. Cheie primară id_utilizator.

RATING = Se alocă un punctaj (mai exact elo (https://en.wikipedia.org/wiki/Elo_rating_system)) fiecărui jucător pentru orice stil de joc (https://en.wikipedia.org/wiki/Time_control), în funcție de rezultatele pe care le are în partidele pe care le joacă. Cheie primară id_rating.

CONVERSAȚII = O discuție între doi utilizatori. Cheie primară id_conversație.

PUZZLES = Un utilizator se poate antrena, rezolvând unele poziții de natură tactică. Cheie primară id_puzzle.

PRIETENIE = Relația dintre un utilizator și un prieten de-al lui. Cheie primară id_prietenie.

TURNEE = Sunt organizate pe platformă la o anumită dată, ce durează un timp specificat partide între utilizatori, la final fiind un clasament în ordine descrescătoare în funcție de câte puncte de joc a câștigat fiecare utilizator. Cheie primară id_turneu.

NOTIFICĂRI = Un mesaj pe care un utilizator îl primește în inbox. Cheie primară id_notificare.

MESAJ = Textul dintr-o notificare. Cheie primară id_mesaj.

PLĂȚI = O sumă de bani pe care un utilizator o plătește. Cheie primară id_plată.

PARTIDE = Totalitatea jocurilor de șah jucate în istorie. Cheie primară id_partidă.

MUTĂRI = Totalitatea mișcărilor făcute într-o partidă. Cheie primară id_mutare.

ANALIZĂ = Totalitatea rezultatelor partidei posibile în urma unor alte mutări decât cele făcute în partidă. Cheie primară id_analiză.

4. Descrierea relațiilor

UTILIZATORII au RATING = relație dintre UTILIZATOR și RATING reflectând legătura dintre acestea (Ce rating are fiecare utilizator). Ea are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATORII au CONVERSAȚII = relație dintre UTILIZATOR și CONVERSAȚII reflectând legătura dintre acestea (Ce conversații are fiecare utilizator). Ea are cardinalitate minimă 0:2 și cardinalitate maximă m:2.

UTILIZATORII rezolvă PUZZLES= relație dintre UTILIZATOR și PUZZLES reflectând legătura dintre acestea (Ce puzzle a rezolvat fiecare utilizator). Ea are cardinalitate minimă 0:0 și cardinalitate maximă m:n.

UTILIZATORII au PRIETENIE= relație dintre UTILIZATOR și PRIETENIE reflectând legătura dintre acestea (Ce prietenie are fiecare utilizator). Ea are cardinalitate minimă 0:2 și cardinalitate maximă m:2.

UTILIZATORII joacă la TURNEE = relație dintre UTILIZATOR și TURNEE reflectând legătura dintre acestea (Ce turnee joacă fiecare utilizator). Ea are cardinalitate minimă 0:2 și cardinalitate maximă m:n.

UTILIZATORII au NOTIFICĂRI= relație dintre UTILIZATOR și NOTIFICĂRI reflectând legătura dintre acestea (Ce notificări are fiecare utilizator). Ea are cardinalitate minimă 0:1 și cardinalitate maximă m:n.

NOTIFICĂRI au MESAJ= relație dintre NOTIFICĂRI și MESAJ reflectând legătura dintre acestea (Ce mesaj e în fiecare Notificare). Ea are cardinalitatea 1:1.

UTILIZATORII fac PLĂȚI = relație dintre UTILIZATOR și PLATĂ reflectând legătura dintre acestea (Ce utilizator a făcut ce plată). Ea are cardinalitate minimă 0:1 și cardinalitate maximă m:1.

UTILIZATORII joacă PARTIDE = relație dintre UTILIZATOR și PARTIDĂ reflectând legătura dintre acestea (Ce partide a jucată de către fiecare utilizator). Ea are cardinalitate minimă 0:2 și cardinalitate maximă m:2.

PARTIDE au ANALIZĂ și MUTĂRI = relație dintre PARTIDE, ANALIZĂ și MUTĂRI reflectând legătura dintre acestea (Ce mutări și ce analize are fiecare partidă).

5. Descrierea Atributelor

- Entitatea independentă UTILIZATORI are ca atribute:

id_utilizator = variabilă de tip întreg de lungime maximă 9, care reprezintă id-ul unic destinat automat fiecarui utilizator în momentul în care se creează un cont în aplicație.

username = variabilă de tip caracter de lungime maximă 25, care reprezintă nickname-ul fiecarui utilizator.

email = variabilă de tip caracter de lungime maximă 100, care reprezintă email-ul utilizatorului.

parolă = variabilă de tip caracter de lungime maximă 25, care reprezintă parola pe care utilizatorul o alege pentru a se loga în cont.

- Entitatea independentă RATING are ca atribute:

id_rating = variabilă de tip întreg de lungime maximă 4, care reprezintă "elo"-ul fiecarui utilizator.

timp_de_joc = variabilă de tip caracter de lungime maximă 10, care reprezintă stilul de joc la care este destinat rating-ul respectiv ce va putea avea următoarele valori: clasic, rapid, blitz, bullet, daily.

- Entitatea independentă PUZZLES are ca atribute:

id_puzzle = variabilă de tip întreg de lungime maximă 9, care reprezintă id-ul unui puzzle.

soluție = variabilă de tip caracter de lungime maximă 250, care reprezintă mutările corecte pentru un puzzle

poziție = variabilă de tip caracter de lungime maximă 250, care reprezintă pisele și ce câmpuri ocupă pentru a asambla poziția

rating = variabilă de tip întreg de lungime maximă 4, care reprezintă "elo"-ul unui puzzle

- Entitatea independentă PRIETENIE are ca atribute:

id_prietenie = variabilă de tip întreg de lungime maximă 9, care reprezintă id-ul unei prietenii.

id_utilizator1 = variabilă de tip întreg de lungime maximă 9, care reprezintă id-ul unic destinat automat fiecarui utilizator în momentul în care se creează un cont în aplicație, reprezentând un prieten din relația de prietenie.

id_utilizator2 = variabilă de tip întreg de lungime maximă 9, care reprezintă id-ul unic destinat automat fiecarui utilizator în momentul în care se creează un cont în aplicație, reprezentând celălalt prieten din relația de prietenie.

data_început = variabilă de tip dată, care reprezintă data din momentul în care s-a creat o prietenie

- Entitatea independentă TURNEE are ca atribute:

id_turneu = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul unui turneu.

nr_participanți = variabliă de tip întreg de lungime maximă 4, care reprezintă numărul de participanți la turneu

dată_start = variabliă de tip întreg dată, care reprezintă data de început a turneului

durată = variabliă de tip întreg de lungime maximă 2, care reprezintă numărul de zile în care se desfășoară turneul

- Entitatea independentă NOTIFICĂRI are ca atribute:

id_notificare = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul notificării

titlu = variabliă de tip caracter de lungime maximă 20, care reprezintă titlul unei notificări

- Entitatea independentă MESAJ are ca atribute:

id_mesaj = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul mesajului unei notificări

id_notificare = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul notificării la care vrem să atașăm un text

text = variabliă de tip caracter de lungime maximă 10000, care reprezintă text-ul propriu-zis al unei notificări

- Entitatea independentă PLĂȚI are ca atribute:

id_plată = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul unei plăți.

id_utilizator = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul utilizatorului care a efectuat o plată.

sumă = variabliă de tip întreg de lungime maximă 3, care reprezintă suma de bani în dolari pe care a achitat-o.

dată_tranzacție = variabliă de tip dată, care reprezintă data la care s-a chitat plata.

- Entitatea independentă PARTIDE are ca atribute:

id_partidă = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul unei partide.

id_utilizator1 = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul primului jucător care a jucat respectiva partidă.

id_utilizator2 = variabliă de tip întreg de lungime maximă 9, care reprezintă id-ul celui de-al doilea jucător care a jucat respectiva partidă.

`id_analiză` = variabiliă de tip întreg de lungime maximă 9, care reprezintă id-ul analizei din spatele partidei

`timp_de_joc` = variabiliă de tip caracter de lungime maximă 10, care reprezintă stilul de joc la care este destinat rating-ul respectiv ce va putea avea următoarele valori: clasic, rapid, blitz, bullet, daily.

- Entitatea independentă ANALIZĂ are ca atribute:

`id_analiză` = variabiliă de tip întreg de lungime maximă 9, care reprezintă id-ul unei analizei de-al unei partide

`id_partidă` = variabiliă de tip întreg de lungime maximă 9, care reprezintă id-ul partidei asupra căreia se desfășoara analiza

`id_mutare` = variabiliă de tip întreg de lungime maximă 9, care reprezintă mutarea cu care analizăm o partidă

`rezultat_analiză` = variabiliă de tip caracter de lungime maximă 3, care reprezintă rezultatul partidei și va avea următoarele valori: 1-0 când albul câștigă, 0-1 când negrul câștigă și ½ când rezultatul este remiză.

- Entitatea independentă MUTĂRI are ca atribute:

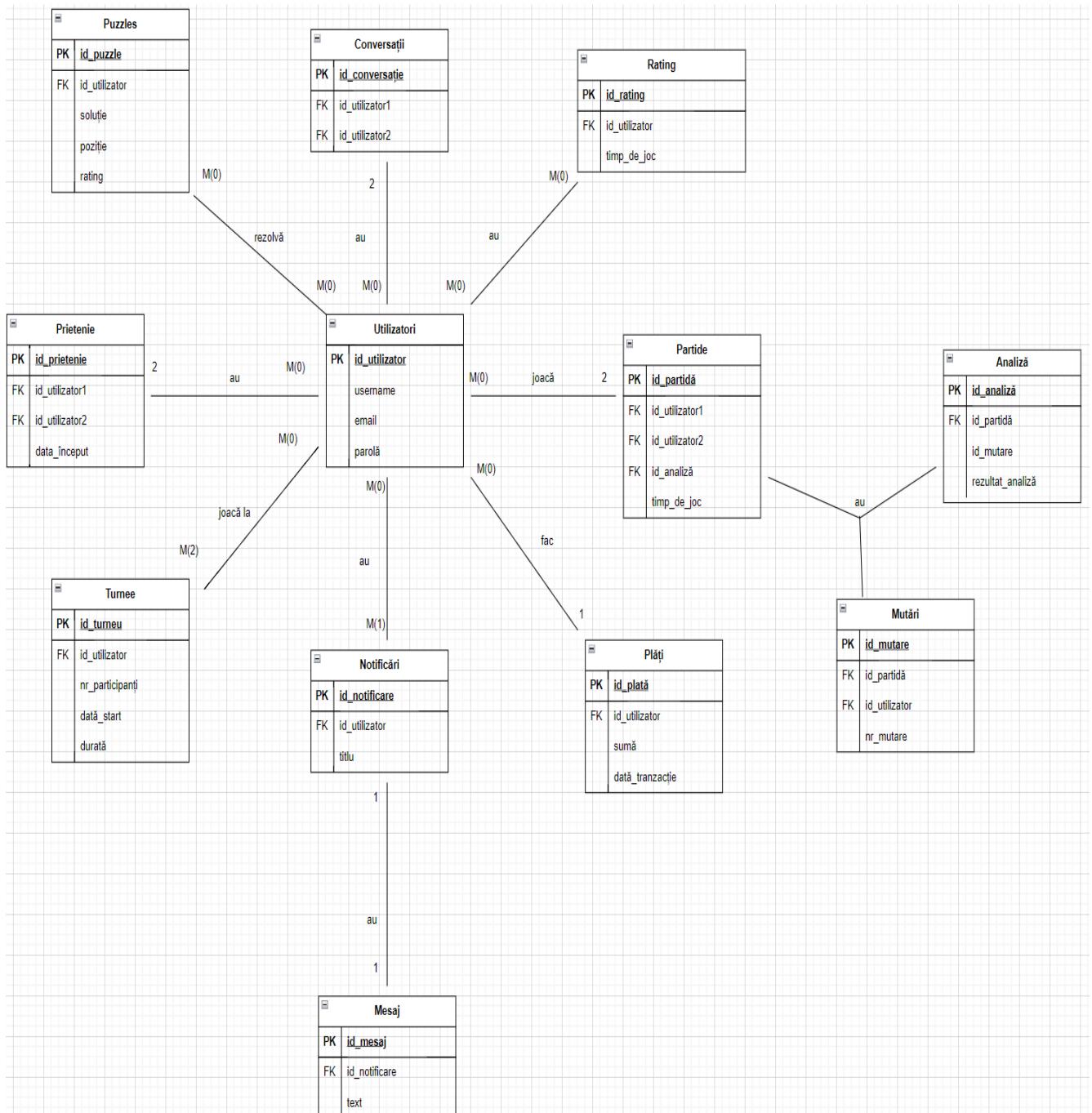
`id_mutare` = variabiliă de tip caracter de lungime maximă 5, care reprezintă mutarea făcută, exemplu e4 (pion se mută pe câmpul e4), Nf6 (din engleză Night = cal se mută pe câmpul f6), 0-0-0 (rocada mare) etc.

`id_utilizator` = variabiliă de tip întreg de lungime maximă 9, care reprezintă id-ul utilizatorului care face o mutare

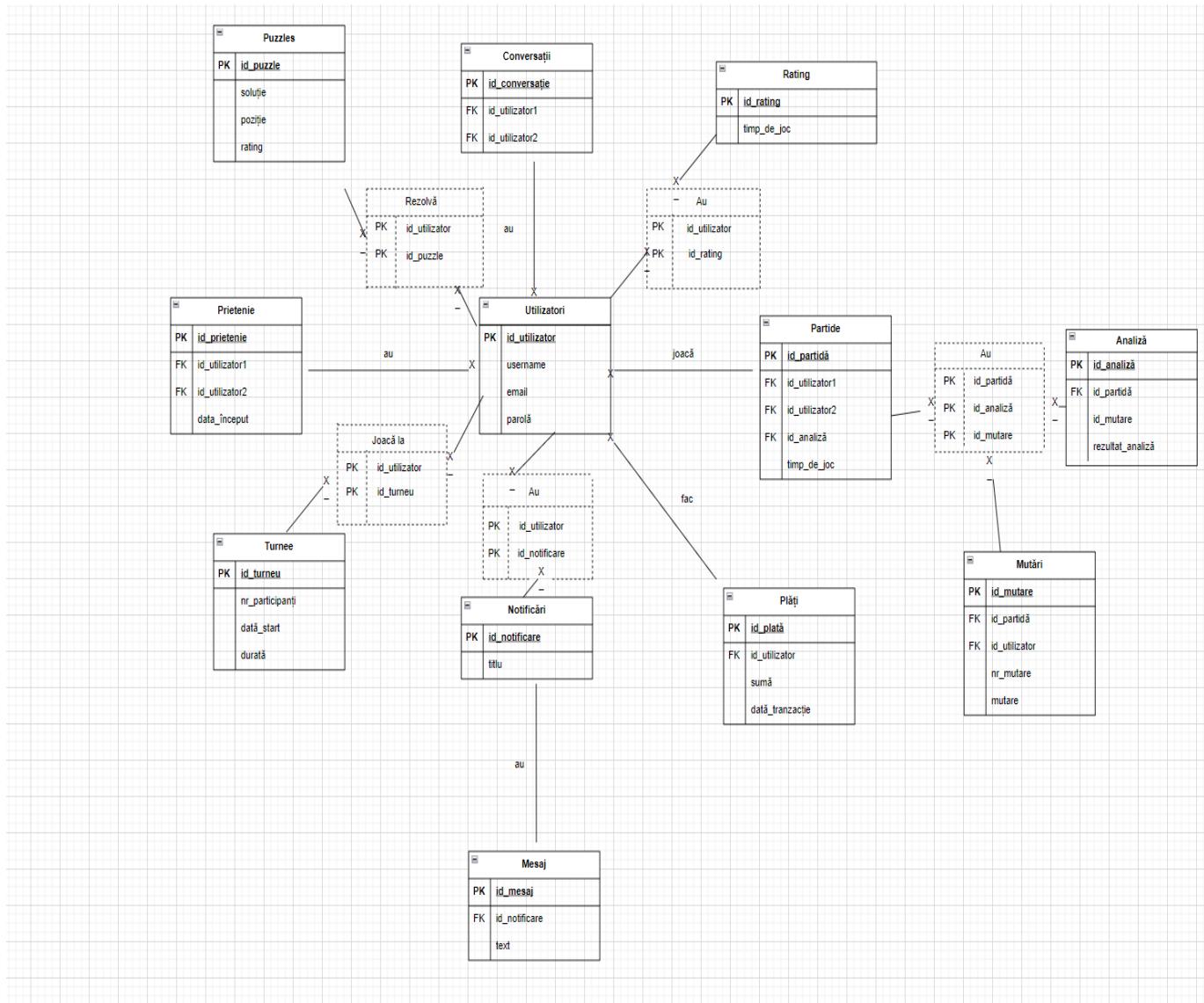
`id_partidă` = variabiliă de tip întreg de lungime maximă 9, care reprezintă id-ul partidei

`nr_mutare` = variabiliă de tip întreg de lungime maximă 4, care reprezintă nr-ul mutării (exemplu la mutarea 1 sau 100 sau 1550)

6. Diagrama Entitate-Relație



7. Diagrama conceptuală



8. Enumerarea Schemelor Relaționale

Schemele relaționale corespunzătoare diagramei conceptuale sunt următoarele:

UTILIZATORI (id_utilizator#, username, email, parolă)

RATING (id_rating#, tim_de_joc)

CONVERSAȚII (id_conversație#, id_utilizator1, id_utilizator2)

PUZZLES (id_puzzle#, soluție, poziție, rating)

PRIETENIE (id_prietenie#, id_utilizator1, id_utilizator2, data_început)

TURNEE (id_turneu#, nr_participanți, dată_start, durată)

NOTIFICĂRI (id_notificare#, titlu)

MESAJ (id_mesaj#, id_notificare, text)

PLĂȚI (id_plată#, id_utilizator, sumă, dată_tranzacție)

PARTIDE (id_partidă#, id_utilizator1, id_utilizator2, id_analiză, timp_de_joc)

MUTĂRI (id_mutare#, id_partidă, id_utilizator, nr_mutare)

ANALIZĂ (id_analiză#, id_partidă, nr_mutare, rezultat_analiză)

AU (id_utilizator#, #id_rating)

REZOLVĂ (id_utilizator#, id_puzzle#)

AU(id_utilizator#, id_notificare#)

JOACĂ_LA(id_utilizator#, id_turneu#)

AU(id_partidă#, id_analiză#, id_mutare#)

9. Realizarea normalizării până la forma normală 3 (FN1-FN3)

Pentru a asigura unicitatea fiecărei linii din fiecare tabel, am folosit cheia primară, care în baza de date este un id de tip UNIQUE. De exemplu, puteam să avem cazul în care în tabela PLĂȚI Ion face o plată de 10\$ de 2 ori în aceeași zi. Fără să fi avut grija să avem primary key id_ul unei plăți individuale, am fi avut următoarea linie în tabel de 2 ori: Ion, 10, 26.05.2023. În schimb, vom avea 2 linii diferite datorită cheii primare: 1, Ion, 10, 26.05.2023; 2, Ion, 10, 26.05.2023.

Pentru că toate tabelele neasociative au doar o singură cheie primară și tabelele asociative au cheie primară compusă, dar nu au și alte atribute în afară de acestea, atunci putem spune că ne aflăm în FN2. Greșit ar fi fost să avem de exemplu în tabela asociativă REZOLVĂ și un atribut cine_e_la_mutare care ar fi spus dacă albul e la mutare într-un puzzle, sau negrul în caz contrar. Ar fi fost greșit deoarece acest atribut depinde doar de id_puzzle, nu și de id_utilizator, care e de asemenea cheie primară pentru această tabelă.

Deoarece toate atributele care nu sunt chei (primary sau foreign) nu depind decât de chei, și nu depinde unele de altele, ne aflăm în FN3. Greșit ar fi fost să fi avut de exemplu în tabela UTILIZATORI un atribut email_fara_@ care ar fi păstrat din fiecare email doar caracterele până la @. Bineînțeles că acest atribut ar fi depinzând de atributul email care nu e cheie, ceea ce nu ar fi fost corect.

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele

```
CREATE SEQUENCE seq
```

```
START WITH 1  
INCREMENT BY 1  
MINVALUE 1  
MAXVALUE 5;
```

11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea

Tabela Utilizatori:

--Inserarea datelor in tabela Utilizatori

```
CREATE TABLE Utilizatori  
(  
    id_utilizator int NOT NULL,  
    username varchar2(25) NOT NULL UNIQUE,  
    email varchar2(100) NOT NULL UNIQUE,  
    parola varchar2(25) NOT NULL,  
    PRIMARY KEY(id_utilizator)  
);
```

```
INSERT INTO Utilizatori  
VALUES (1, 'ionut', 'ionut@gmail.com', 'ionut');  
INSERT INTO Utilizatori  
VALUES (2, 'alex', 'ale@gmail.com', 'pal');  
INSERT INTO Utilizatori
```

```
VALUES (3, 'maremaestru', 'maremaestru@yahoo.com', 'GM');
```

```
INSERT INTO Utilizatori
```

```
VALUES (4, 'queen', 'qel@gmail.com', 'iamqueen');
```

```
INSERT INTO Utilizatori
```

```
VALUES (5, 'd4player', 'plic@gmail.com', 'londonsystem');
```

```
DESCRIBE Utilizatori;
```

```
SELECT *
```

```
FROM Utilizatori;
```

```
1/*CREATE TABLE Utilizatori
2{
3    id_utilizator int NOT NULL,
4    username varchar2(25) NOT NULL UNIQUE,
5    email varchar2(100) NOT NULL UNIQUE,
6    parola varchar2(25) NOT NULL,
7    PRIMARY KEY(id_utilizator)
8);
9
10 INSERT INTO Utilizatori
11 VALUES (1, 'ionut', 'ionut@gmail.com', 'ionut');
12 INSERT INTO Utilizatori
13 VALUES (2, 'alex', 'ale@gmail.com', 'pal');
14 INSERT INTO Utilizatori
15 VALUES (3, 'maremaestru', 'maremaestru@yahoo.com', 'GM');
16 INSERT INTO Utilizatori
17 VALUES (4, 'queen', 'qel@gmail.com', 'iamqueen');
18 INSERT INTO Utilizatori
19 VALUES (5, 'd4player', 'plic@gmail.com', 'londonsystem');
20
21 DESCRIBE Utilizatori;
22 SELECT *
23 FROM Utilizatori;
```

```
Script Output X | Query Result X
Task completed in 0.34 seconds.
. row inserted.
```

```
. row inserted.
```

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER(38)
USERNAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(100)
PAROLA	NOT NULL	VARCHAR2(25)

```
1/*CREATE TABLE Utilizatori
2{
3    id_utilizator int NOT NULL,
4    username varchar2(25) NOT NULL UNIQUE,
5    email varchar2(100) NOT NULL UNIQUE,
6    parola varchar2(25) NOT NULL,
7    PRIMARY KEY(id_utilizator)
8);
9
10 INSERT INTO Utilizatori
11 VALUES (1, 'ionut', 'ionut@gmail.com', 'ionut');
12 INSERT INTO Utilizatori
13 VALUES (2, 'alex', 'ale@gmail.com', 'pal');
14 INSERT INTO Utilizatori
15 VALUES (3, 'maremaestru', 'maremaestru@yahoo.com', 'GM');
16 INSERT INTO Utilizatori
17 VALUES (4, 'queen', 'qel@gmail.com', 'iamqueen');
18 INSERT INTO Utilizatori
19 VALUES (5, 'd4player', 'plic@gmail.com', 'londonsystem');
20
21 DESCRIBE Utilizatori;
22 SELECT *
23 FROM Utilizatori;
```

```
Script Output X | Query Result X
All Rows Fetched: 5 in 0 seconds
ID_UTILIZATOR | USERNAME | EMAIL | PAROLA
1 | ionut | ionut@gmail.com | ionut
2 | alex | ale@gmail.com | pal
3 | 3maremaestru | maremaestru@yahoo.com | GM
4 | 4queen | qel@gmail.com | iamqueen
5 | 5d4player | plic@gmail.com | londonsystem
```

Tabela Rating:

--Inserarea datelor in tabela Rating

```
CREATE TABLE Rating
(
    id_rating int NOT NULL,
    timp_de_joc varchar2(10) NOT NULL,
    PRIMARY KEY(id_rating)
);
```

```
INSERT INTO Rating
VALUES (1500, 'clasic');

INSERT INTO Rating
VALUES (2500, 'rapid');

INSERT INTO Rating
VALUES (250, 'bullet');

INSERT INTO Rating
VALUES (1750, 'blitz');

INSERT INTO Rating
VALUES (1867, 'blitz');
```

INSERT INTO Rating

VALUES (2000, 'blitz');

INSERT INTO Rating

VALUES (3175, 'blitz');

DESCRIBE Rating;

SELECT *

FROM Rating;

```
34 | (
35 |     id_rating int NOT NULL,
36 |     id_utilizator int NOT NULL,
37 |     timp_de_joc varchar2(10) NOT NULL,
38 |     PRIMARY KEY(id_rating),
39 |     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
40 );
41
42 INSERT INTO Rating
43 VALUES (1500, 1, 'clasic');
44 INSERT INTO Rating
45 VALUES (2500, 1, 'rapid');
46 INSERT INTO Rating
47 VALUES (250, 1, 'bullet');
48 INSERT INTO Rating
49 VALUES (1750, 2, 'blitz');
50 INSERT INTO Rating
51 VALUES (1867, 3, 'blitz');
52 INSERT INTO Rating
53 VALUES (2000, 4, 'blitz');
54 INSERT INTO Rating
55 VALUES (3175, 5, 'blitz');
56
57 DESCRIBE Rating;
58 SELECT *
59 FROM Rating;
```

```
SQL Output [ ] | Query Result [ ]
Task completed in 0.072 seconds
TIMP_DE_JOC    NOT NULL VARCHAR2(10)
Name           Null?   Type
----- -----
ID_RATING      NOT NULL NUMBER(38)
ID_UTILIZATOR  NOT NULL NUMBER(38)
TIMP_DE_JOC    NOT NULL VARCHAR2(10)
Name           Null?   Type
----- -----
ID_RATING      NOT NULL NUMBER(38)
ID_UTILIZATOR  NOT NULL NUMBER(38)
TIMP_DE_JOC    NOT NULL VARCHAR2(10)
```

```
34 | (
35 |     id_rating int NOT NULL,
36 |     id_utilizator int NOT NULL,
37 |     timp_de_joc varchar2(10) NOT NULL,
38 |     PRIMARY KEY(id_rating),
39 |     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
40 );
41
42 INSERT INTO Rating
43 VALUES (1500, 1, 'clasic');
44 INSERT INTO Rating
45 VALUES (2500, 1, 'rapid');
46 INSERT INTO Rating
47 VALUES (250, 1, 'bullet');
48 INSERT INTO Rating
49 VALUES (1750, 2, 'blitz');
50 INSERT INTO Rating
51 VALUES (1867, 3, 'blitz');
52 INSERT INTO Rating
53 VALUES (2000, 4, 'blitz');
54 INSERT INTO Rating
55 VALUES (3175, 5, 'blitz');
56
57 DESCRIBE Rating;
58 SELECT *
59 FROM Rating;
```

```
SQL Output [ ] | Query Result [ ]
Task completed in 0.072 seconds
ID_RATING | ID_UTILIZATOR | TIMP_DE_JOC
1 1500    | 1             | clasic
2 2500    | 1             | rapid
3 250    | 1             | bullet
4 1750    | 2             | blitz
5 1867    | 3             | blitz
6 2000    | 4             | blitz
7 3175    | 5             | blitz
```

Tabela Conversații:

--Inserarea datelor in tabela Conversatii

```
CREATE TABLE Conversatii
(
    id_conversatie int NOT NULL,
    id_utilizator1 int NOT NULL,
    id_utilizator2 int NOT NULL,
    PRIMARY KEY(id_conversatie),
    FOREIGN KEY(id_utilizator1) REFERENCES
Utilizatori(id_utilizator),
    FOREIGN KEY(id_utilizator2) REFERENCES
Utilizatori(id_utilizator)
);
```

```
INSERT INTO Conversatii
```

```
VALUES (1, 1, 2);
```

```
INSERT INTO Conversatii
```

```
VALUES (2, 1, 3);
```

```
INSERT INTO Conversatii
```

```
VALUES (3, 1, 4);
```

```
INSERT INTO Conversatii
```

```

VALUES (4, 1, 5);
INSERT INTO Conversatii
VALUES (5, 2, 3);

```

DESCRIBE Conversatii;

```

SELECT *
FROM Conversatii;

```

```

63 |
64=CREATE TABLE Conversatii
65 (
66     id_conversatie int NOT NULL,
67     id_utilizator1 int NOT NULL,
68     id_utilizator2 int NOT NULL,
69     PRIMARY KEY(id_conversatie),
70     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
71     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
72 );
73
74 INSERT INTO Conversatii
75 VALUES (1, 1, 2);
76 INSERT INTO Conversatii
77 VALUES (2, 1, 3);
78 INSERT INTO Conversatii
79 VALUES (3, 1, 4);
80 INSERT INTO Conversatii
81 VALUES (4, 1, 5);
82 INSERT INTO Conversatii
83 VALUES (5, 2, 3);
84
85
86 DESCRIBE Conversatii;
87 SELECT *
88 FROM Conversatii;

```

Script Output | Query Result | Task completed in 0.194 seconds

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_CONVERSATIE	NOT NULL	NUMBER(38)
ID_UTILIZATOR1	NOT NULL	NUMBER(38)
ID_UTILIZATOR2	NOT NULL	NUMBER(38)

```

63|
64=CREATE TABLE Conversatii
65 (
66     id_conversatie int NOT NULL,
67     id_utilizator1 int NOT NULL,
68     id_utilizator2 int NOT NULL,
69     PRIMARY KEY(id_conversatie),
70     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
71     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
72 );
73
74 INSERT INTO Conversatii
75 VALUES (1, 1, 2);
76 INSERT INTO Conversatii
77 VALUES (2, 1, 3);
78 INSERT INTO Conversatii
79 VALUES (3, 1, 4);
80 INSERT INTO Conversatii
81 VALUES (4, 1, 5);
82 INSERT INTO Conversatii
83 VALUES (5, 2, 3);
84
85
86 DESCRIBE Conversatii;
87 SELECT *
88 FROM Conversatii;

```

Script Output | Query Result | All Rows Fetched 5 in 0 seconds

ID_CONVERSATIE	ID_UTILIZATOR1	ID_UTILIZATOR2
1	1	2
2	2	1
3	3	1
4	4	1
5	5	2

Tabela Puzzles:

--Inserarea datelor in tabela Puzzles

CREATE TABLE Puzzles

```
(  
    id_puzzle int NOT NULL,  
    solutie varchar2(250) NOT NULL,  
    pozitie varchar2(250) NOT NULL,  
    rating int NOT NULL,  
    PRIMARY KEY(id_puzzle)  
);
```

INSERT INTO Puzzles

```
VALUES (1, 'Ra8', 'Kd6 Ra1 Kd8', 100);
```

INSERT INTO Puzzles

```
VALUES (2, 'Qd7', 'Kd6 Qc6 Kd8 Rc8 Re8', 150);
```

INSERT INTO Puzzles

```
VALUES (3, 'Qa8 Rc8 Qc8', 'Kh6 Qa2 Rc3 Kh8', 200);
```

INSERT INTO Puzzles

```
VALUES (4, 'Kg7 h4 Kf6 h3 Ke7 h2 c7 Kb7 Kd7 h1=Q c8=Q', 'Kh8 c6 Ka6
h5', 2500);
```

INSERT INTO Puzzles

```
VALUES (5, 'Kc6 Kc8 d6 Kd8 d7 Ke7 Kc7 Ke6 d8=Q', 'Kd6 d5 Kd8', 1000);
```

DESCRIBE Puzzles;

SELECT *

FROM Puzzles;

```
94--CREATE TABLE Puzzles
95(
96    id_puzzle int NOT NULL,
97    id_utilizator int NOT NULL,
98    solutie varchar2(250) NOT NULL,
99    pozitie varchar2(250) NOT NULL,
100   rating int NOT NULL,
101   PRIMARY KEY(id_puzzle),
102   FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
103 );
104
105 INSERT INTO Puzzles
106 VALUES (1, 1, 'Ra8', 'Kd6 Ra1 Kd8', 100);
107 INSERT INTO Puzzles
108 VALUES (2, 1, 'Qd7', 'Kd6 Qc6 Kd8 Rc8 Re8', 150);
109 INSERT INTO Puzzles
110 VALUES (3, 2, 'Qa8 Rc8 Qc8', 'Kh6 Qa2 Rc3 Kh8', 200);
111 INSERT INTO Puzzles
112 VALUES (4, 3, 'Kg7 h4 Kf6 h3 Ke7 h2 c7 Kb7 Kd7 h1=Q c8=Q', 'Kh8 c6 Ka6 h5', 2500);
113 INSERT INTO Puzzles
114 VALUES (5, 4, 'Kc6 Kc8 d6 Kd8 d7 Ke7 Kc7 Ke6 d8=Q', 'Kd6 d5 Kd8', 1000);
115
116
117 DESCRIBE Puzzles;
118 SELECT *
119 FROM Puzzles;
```

Script Output | Query Result | Task completed in 0.192 seconds

1 row inserted.

Name	Null?	Type
ID_PUZZLE	NOT NULL NUMBER(38)	
ID_UTILIZATOR	NOT NULL NUMBER(38)	
SOLUTIE	NOT NULL VARCHAR2(250)	
POZITIE	NOT NULL VARCHAR2(250)	
RATING	NOT NULL NUMBER(38)	

```
94--CREATE TABLE Puzzles
95(
96    id_puzzle int NOT NULL,
97    id_utilizator int NOT NULL,
98    solutie varchar2(250) NOT NULL,
99    pozitie varchar2(250) NOT NULL,
100   rating int NOT NULL,
101   PRIMARY KEY(id_puzzle),
102   FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
103 );
104
105 INSERT INTO Puzzles
106 VALUES (1, 1, 'Ra8', 'Kd6 Ra1 Kd8', 100);
107 INSERT INTO Puzzles
108 VALUES (2, 1, 'Qd7', 'Kd6 Qc6 Kd8 Rc8 Re8', 150);
109 INSERT INTO Puzzles
110 VALUES (3, 2, 'Qa8 Rc8 Qc8', 'Kh6 Qa2 Rc3 Kh8', 200);
111 INSERT INTO Puzzles
112 VALUES (4, 3, 'Kg7 h4 Kf6 h3 Ke7 h2 c7 Kb7 Kd7 h1=Q c8=Q', 'Kh8 c6 Ka6 h5', 2500);
113 INSERT INTO Puzzles
114 VALUES (5, 4, 'Kc6 Kc8 d6 Kd8 d7 Ke7 Kc7 Ke6 d8=Q', 'Kd6 d5 Kd8', 1000);
115
116
117 DESCRIBE Puzzles;
118 SELECT *
119 FROM Puzzles;
```

Script Output | Query Result | All Rows Fetched 5 in 0 seconds

ID_PUZZLE	ID_UTILIZATOR	SOLUTIE	POZITIE	RATING
1	1	1Ra8	Kd6 Ra1 Kd8	100
2	3	2Qa8 Rc8 Qc8	Kh6 Qa2 Rc3 Kh8	200

Tabela Prietenie:

--Inserarea datelor in tabela Prietenie

CREATE TABLE Prietenie

(

 id_prietenie int NOT NULL,

 id_utilizator1 int NOT NULL,

 id_utilizator2 int NOT NULL,

 data_inceput date,

 PRIMARY KEY(id_prietenie),

 FOREIGN KEY(id_utilizator1) REFERENCES

 Utilizatori(id_utilizator),

 FOREIGN KEY(id_utilizator2) REFERENCES

 Utilizatori(id_utilizator)

);

INSERT INTO Prietenie

VALUES (1, 1, 4, to_date('31-03-2023', 'dd-mm-yyyy'));

INSERT INTO Prietenie

VALUES (2, 1, 5, to_date('30-04-2023', 'dd-mm-yyyy'));

```
INSERT INTO Prietenie
```

```
VALUES (3, 2, 3, to_date('19-05-2023', 'dd-mm-yyyy'));
```

```
INSERT INTO Prietenie
```

```
VALUES (4, 2, 4, to_date('30-06-2022', 'dd-mm-yyyy'));
```

```
INSERT INTO Prietenie
```

```
VALUES (5, 3, 5, to_date('18-08-2022', 'dd-mm-yyyy'));
```

```
DESCRIBE Prietenie;
```

```
SELECT *
```

```
FROM Prietenie;
```

```
124#CREATE TABLE Prietenie
125 (
126     id_prietenie int NOT NULL,
127     id_utilizator1 int NOT NULL,
128     id_utilizator2 int NOT NULL,
129     data_inceput date,
130     PRIMARY KEY(id_prietenie),
131     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
132     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
133 );
134
135 INSERT INTO Prietenie
136 VALUES (1, 1, 4, to_date('31-03-2023', 'dd-mm-yyyy'));
137 INSERT INTO Prietenie
138 VALUES (2, 1, 5, to_date('30-04-2023', 'dd-mm-yyyy'));
139 INSERT INTO Prietenie
140 VALUES (3, 2, 3, to_date('19-05-2023', 'dd-mm-yyyy'));
141 INSERT INTO Prietenie
142 VALUES (4, 2, 4, to_date('30-06-2022', 'dd-mm-yyyy'));
143 INSERT INTO Prietenie
144 VALUES (5, 3, 5, to_date('18-08-2022', 'dd-mm-yyyy'));
145
146
147 DESCRIBE Prietenie;
148 SELECT *
149 FROM Prietenie;
```

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_PRIETENIE	NOT NULL	NUMBER(38)
ID_UTILIZATOR1	NOT NULL	NUMBER(38)
ID_UTILIZATOR2	NOT NULL	NUMBER(38)
DATA_INCEPUT		DATE

```
124#CREATE TABLE Prietenie
125 (
126     id_prietenie int NOT NULL,
127     id_utilizator1 int NOT NULL,
128     id_utilizator2 int NOT NULL,
129     data_inceput date,
130     PRIMARY KEY(id_prietenie),
131     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
132     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
133 );
134
135 INSERT INTO Prietenie
136 VALUES (1, 1, 4, to_date('31-03-2023', 'dd-mm-yyyy'));
137 INSERT INTO Prietenie
138 VALUES (2, 1, 5, to_date('30-04-2023', 'dd-mm-yyyy'));
139 INSERT INTO Prietenie
140 VALUES (3, 2, 3, to_date('19-05-2023', 'dd-mm-yyyy'));
141 INSERT INTO Prietenie
142 VALUES (4, 2, 4, to_date('30-06-2022', 'dd-mm-yyyy'));
143 INSERT INTO Prietenie
144 VALUES (5, 3, 5, to_date('18-08-2022', 'dd-mm-yyyy'));
145
146
147 DESCRIBE Prietenie;
148 SELECT *
```

Tabela Turnee:

--Inserarea datelor in tabela Turnee

```
CREATE TABLE Turnee
(
    id_turneu int NOT NULL,
    nr_participanti int NOT NULL,
    data_start date NOT NULL,
    durata int NOT NULL,
    PRIMARY KEY(id_turneu)
);
```

```
INSERT INTO Turnee
```

```
VALUES (1, 5, to_date('31-03-2023', 'dd-mm-yyyy'), 7);
```

```
INSERT INTO Turnee
```

```
VALUES (2, 3, to_date('01-04-2023', 'dd-mm-yyyy'), 3);
```

```
INSERT INTO Turnee
```

```
VALUES (3, 4, to_date('31-01-2023', 'dd-mm-yyyy'), 12);
```

```
INSERT INTO Turnee
```

```
VALUES (4, 2, to_date('28-02-2023', 'dd-mm-yyyy'), 8);
```

```
INSERT INTO Turnee
```

```
VALUES (5, 5, to_date('29-02-2020', 'dd-mm-yyyy'), 15);
```

```
DESCRIBE Turnee;
```

```
SELECT *
```

```
FROM Turnee;
```

```
150
151
152 --Inserarea datelor in tabela Turnee
153
154 CREATE TABLE Turnee
155 (
156     id_turneu int NOT NULL,
157     nr_participanti int NOT NULL,
158     data_start date NOT NULL,
159     durata int NOT NULL,
160     PRIMARY KEY(id_turneu)
161 );
162
163 INSERT INTO Turnee
164 VALUES (1, 5, to_date('31-03-2023', 'dd-mm-yyyy'), 7);
165 INSERT INTO Turnee
166 VALUES (2, 3, to_date('01-04-2023', 'dd-mm-yyyy'), 3);
167 INSERT INTO Turnee
168 VALUES (3, 4, to_date('31-01-2023', 'dd-mm-yyyy'), 12);
169 INSERT INTO Turnee
170 VALUES (4, 2, to_date('28-02-2023', 'dd-mm-yyyy'), 8);
171 INSERT INTO Turnee
172 VALUES (5, 5, to_date('29-02-2020', 'dd-mm-yyyy'), 15);
173
174 DESCRIBE Turnee;
175 SELECT *
```

Script Output | Query Result | Task completed in 0.188 seconds
1 row inserted.

ERROR:
ORA-04043: object Turnee does not exist
Name Null? Type

ID_TURNEU NOT NULL NUMBER(38)
NR_PARTICIPANTI NOT NULL NUMBER(38)
DATA_START DATE
DURATA NOT NULL NUMBER(38)

```
153
154 CREATE TABLE Turnee
155 (
156     id_turneu int NOT NULL,
157     nr_participanti int NOT NULL,
158     data_start date NOT NULL,
159     durata int NOT NULL,
160     PRIMARY KEY(id_turneu)
161 );
162
163 INSERT INTO Turnee
164 VALUES (1, 5, to_date('31-03-2023', 'dd-mm-yyyy'), 7);
165 INSERT INTO Turnee
166 VALUES (2, 3, to_date('01-04-2023', 'dd-mm-yyyy'), 3);
167 INSERT INTO Turnee
168 VALUES (3, 4, to_date('31-01-2023', 'dd-mm-yyyy'), 12);
169 INSERT INTO Turnee
170 VALUES (4, 2, to_date('28-02-2023', 'dd-mm-yyyy'), 8);
171 INSERT INTO Turnee
172 VALUES (5, 5, to_date('29-02-2020', 'dd-mm-yyyy'), 15);
173
174 DESCRIBE Turnee;
175 SELECT *
176 FROM Turnee;
177
178
```

Script Output | Query Result | All Rows Petched 5 in 0 seconds

ID_TURNEU	NR_PARTICIPANTI	DATA_START	DURATA
1	5	31-MAR-23	7
2	3	01-APR-23	3
3	4	31-JAN-23	12
4	2	28-FEB-23	8
5	5	29-FEB-20	15

Tabela Notificari:

--Inserarea datelor in tabela Notificari

```
CREATE TABLE Notificari
(
    id_notificare int NOT NULL,
    titlu varchar2(20) NOT NULL,
    PRIMARY KEY(id_notificare)
);
```

```
INSERT INTO Notificari
VALUES (1, 'Turneu in 2 zile');

INSERT INTO Notificari
VALUES (2, 'Turneu Titled');

INSERT INTO Notificari
VALUES (3, 'Cerere prietenie');

INSERT INTO Notificari
VALUES (4, 'Felicitari!');

INSERT INTO Notificari
VALUES (5, 'Turneu in 5 zile');
```

```

DESCRIBE Notificari;

SELECT *

FROM Notificari;

```

```

178--Inserarea datelor in tabela Notificari
179CREATE TABLE Notificari
180(
181    id_notificare int NOT NULL,
182    titlu varchar2(20) NOT NULL,
183    PRIMARY KEY(id_notificare)
184);
185
186
187
188INSERT INTO Notificari
189VALUES (1, 'Turneu in 2 zile');
190INSERT INTO Notificari
191VALUES (2, 'Turneu Titled');
192INSERT INTO Notificari
193VALUES (3, 'Cerere prietenie');
194INSERT INTO Notificari
195VALUES (4, 'Felicitari!');
196INSERT INTO Notificari
197VALUES (5, 'Turneu in 5 zile');
198
199DESCRIBE Notificari;
200SELECT *
201FROM Notificari;
202
203

```

Script Output Query Result Task completed in 0.233 seconds

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_NOTIFICARE	NOT NULL	NUMBER(38)
TITLU	NOT NULL	VARCHAR2(20)

```

178--Inserarea datelor in tabela Notificari
179CREATE TABLE Notificari
180(
181    id_notificare int NOT NULL,
182    titlu varchar2(20) NOT NULL,
183    PRIMARY KEY(id_notificare)
184);
185
186
187
188INSERT INTO Notificari
189VALUES (1, 'Turneu in 2 zile');
190INSERT INTO Notificari
191VALUES (2, 'Turneu Titled');
192INSERT INTO Notificari
193VALUES (3, 'Cerere prietenie');
194INSERT INTO Notificari
195VALUES (4, 'Felicitari!');
196INSERT INTO Notificari
197VALUES (5, 'Turneu in 5 zile');
198
199DESCRIBE Notificari;
200SELECT *
201FROM Notificari;
202
203

```

Tabela Mesaj:

--Inserarea datelor in tabela Mesaj

```
CREATE TABLE Mesaj
(
    id_mesaj int NOT NULL,
    id_notificare int NOT NULL,
    text varchar2(10000) NOT NULL,
    PRIMARY KEY(id_mesaj),
    FOREIGN KEY(id_notificare) REFERENCES Notificari(id_notificare)
);
```

INSERT INTO Mesaj

VALUES (1, 1, 'Inregistreaza-te acum la turneul "Titled Tuesday" care incepe in 2 zile!');

INSERT INTO Mesaj

VALUES (2, 2, 'Turneu in care participa maestri incep in curand!');

INSERT INTO Mesaj

VALUES (3, 3, 'Ai primit o cerere de prietenie de la d4player!');

```
INSERT INTO Mesaj
```

```
VALUES (4, 4, 'Ai reusit sa obtii un elo record. Felicitari!');
```

```
INSERT INTO Mesaj
```

```
VALUES (5, 5, 'Au mai ramas doar 5 zile in care te poti inscrie la turneul nostru!');
```

```
DESCRIBE Mesaj;
```

```
SELECT *
```

```
FROM Mesaj;
```

```
205
206 CREATE TABLE Mesaj
207 (
208     id_mesaj int NOT NULL,
209     id_notificare int NOT NULL,
210     text varchar2(10000) NOT NULL,
211     PRIMARY KEY(id_mesaj),
212     FOREIGN KEY(id_notificare) REFERENCES Notificari(id_notificare)
213 );
214
215 INSERT INTO Mesaj
216 VALUES (1, 1, 'Inregistreaza-te acum la turneul "Titled Tuesday" care incepe in 2 zile!');
217 INSERT INTO Mesaj
218 VALUES (2, 2, 'Turneu in care participa maestri incep in curand!');
219 INSERT INTO Mesaj
220 VALUES (3, 3, 'Ai primit o cerere de prietenie de la d4player!');
221 INSERT INTO Mesaj
222 VALUES (4, 4, 'Ai reusit sa obtii un elo record. Felicitari!');
223 INSERT INTO Mesaj
224 VALUES (5, 5, 'Au mai ramas doar 5 zile in care te poti inscrie la turneul nostru!');
225
226 DESCRIBE Mesaj;
227 SELECT *
228 FROM Mesaj;
229
230
```

```
Script Output X Query Result X
Task completed in 0.199 seconds
```

```
1 row inserted.
```

```
1 row inserted.
```

Name	Null?	Type
ID_MESAJ	NOT NULL	NUMBER(38)
ID_NOTIFICARE	NOT NULL	NUMBER(38)
TEXT	NOT NULL	VARCHAR2(10000)

```
205
206 CREATE TABLE Mesaj
207 (
208     id_mesaj int NOT NULL,
209     id_notificare int NOT NULL,
210     text varchar2(10000) NOT NULL,
211     PRIMARY KEY(id_mesaj),
212     FOREIGN KEY(id_notificare) REFERENCES Notificari(id_notificare)
213 );
214
215 INSERT INTO Mesaj
216 VALUES (1, 1, 'Inregistreaza-te acum la turneul "Titled Tuesday" care incepe in 2 zile!');
217 INSERT INTO Mesaj
218 VALUES (2, 2, 'Turneu in care participa maestri incep in curand!');
219 INSERT INTO Mesaj
220 VALUES (3, 3, 'Ai primit o cerere de prietenie de la d4player!');
221 INSERT INTO Mesaj
222 VALUES (4, 4, 'Ai reusit sa obtii un elo record. Felicitari!');
223 INSERT INTO Mesaj
224 VALUES (5, 5, 'Au mai ramas doar 5 zile in care te poti inscrie la turneul nostru!');
225
226 DESCRIBE Mesaj;
227 SELECT *
```

Tabela Plăți:

--Inserarea datelor in tabela Plati

```
CREATE TABLE Plati
(
    id_plata int NOT NULL,
    id_utilizator int NOT NULL,
    suma int NOT NULL,
    data_tranzactie date NOT NULL,
    PRIMARY KEY(id_plata),
    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
);
```

INSERT INTO Plati

```
VALUES (1, 1, 15, to_date('01-01-2023', 'dd-mm-yyyy'));
```

```
INSERT INTO Plati
```

```
VALUES (2, 1, 20, to_date('01-02-2023', 'dd-mm-yyyy'));
```

```
INSERT INTO Plati
```

```
VALUES (3, 2, 15, to_date('01-03-2023', 'dd-mm-yyyy'));
```

```
INSERT INTO Plati
```

```
VALUES (4, 4, 15, to_date('31-01-2023', 'dd-mm-yyyy'));
```

```
INSERT INTO Plati
```

```
VALUES (5, 1, 10, to_date('01-05-2023', 'dd-mm-yyyy'));
```

```
DESCRIBE Plati;
```

```
SELECT *
```

```
FROM Plati;
```

```
233
234 CREATE TABLE Plati
235 (
236     id_plata int NOT NULL,
237     id_utilizator int NOT NULL,
238     suma int NOT NULL,
239     data_tranzactie date NOT NULL,
240     PRIMARY KEY(id_plata),
241     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
242 );
243
244 INSERT INTO Plati
245 VALUES (1, 1, 15, to_date('01-01-2023', 'dd-mm-yyyy'));
246 INSERT INTO Plati
247 VALUES (2, 1, 20, to_date('01-02-2023', 'dd-mm-yyyy'));
248 INSERT INTO Plati
249 VALUES (3, 2, 15, to_date('01-03-2023', 'dd-mm-yyyy'));
250 INSERT INTO Plati
251 VALUES (4, 4, 15, to_date('31-01-2023', 'dd-mm-yyyy'));
252 INSERT INTO Plati
253 VALUES (5, 1, 10, to_date('01-05-2023', 'dd-mm-yyyy'));
254
255 DESCRIBE Plati;
256 SELECT *
257 FROM Plati;
258
```

```
Script Output * Query Result *
Task completed in 0.212 seconds
```

```
1 row inserted.
```

```
1 row inserted.
```

Name	Null?	Type
ID_PLATA	NOT NULL	NUMBER(38)
ID_UTILIZATOR	NOT NULL	NUMBER(38)
SUMA	NOT NULL	NUMBER(38)
DATA_TRANZACTIE	NOT NULL	DATE

```
233
234 CREATE TABLE Plati
235 (
236     id_plata int NOT NULL,
237     id_utilizator int NOT NULL,
238     suma int NOT NULL,
239     data_tranzactie date NOT NULL,
240     PRIMARY KEY(id_plata),
241     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
242 );
243
244 INSERT INTO Plati
245 VALUES (1, 1, 15, to_date('01-01-2023', 'dd-mm-yyyy'));
246 INSERT INTO Plati
247 VALUES (2, 1, 20, to_date('01-02-2023', 'dd-mm-yyyy'));
```

Tabela Partide:

--Inserarea datelor in tabela Partide

```
CREATE TABLE Partide
(
    id_partida int NOT NULL,
    id_utilizator1 int NOT NULL,
    id_utilizator2 int NOT NULL,
    id_analiza int NOT NULL,
    timp_de_joc varchar2(10) NOT NULL,
    PRIMARY KEY(id_partida),
    FOREIGN KEY(id_utilizator1) REFERENCES
    Utilizatori(id_utilizator),
    FOREIGN KEY(id_utilizator2) REFERENCES
    Utilizatori(id_utilizator)
);
```

```

INSERT INTO Partide
VALUES (1, 1, 2, 1, 'blitz');

INSERT INTO Partide
VALUES (2, 1, 3, 2, 'bullet');

INSERT INTO Partide
VALUES (3, 1, 4, 3, 'clasic');

INSERT INTO Partide
VALUES (4, 1, 5, 4, 'blitz');

INSERT INTO Partide
VALUES (5, 2, 3, 5, 'blitz');

```

```

DESCRIBE Partide;

SELECT *

FROM Partide;

```

```

262 |
263 CREATE TABLE Partide
264 (
265     id_partida int NOT NULL,
266     id_utilizator1 int NOT NULL,
267     id_utilizator2 int NOT NULL,
268     id_analiza int NOT NULL,
269     timp_de_joc varchar2(10) NOT NULL,
270     PRIMARY KEY(id_partida),
271     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
272     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
273 );
274
275 INSERT INTO Partide
276 VALUES (1, 1, 2, 1, 'blitz');
277 INSERT INTO Partide
278 VALUES (2, 1, 3, 2, 'bullet');
279 INSERT INTO Partide
280 VALUES (3, 1, 4, 3, 'clasic');
281 INSERT INTO Partide
282 VALUES (4, 1, 5, 4, 'blitz');
283 INSERT INTO Partide
284 VALUES (5, 2, 3, 5, 'blitz');
285
286 DESCRIBE Partide;
287 SELECT *

```

Query Result

Task completed in 0.001 seconds

1 row inserted.

Name	Null?	Type
ID_PARTIDA	NOT NULL	NUMBER(38)
ID_UTILIZATOR1	NOT NULL	NUMBER(38)
ID_UTILIZATOR2	NOT NULL	NUMBER(38)
ID_ANALIZA	NOT NULL	NUMBER(38)
TIMP_DE_JOC	NOT NULL	VARCHAR2(10)

```

264 |
265 |     id_partida int NOT NULL,
266 |     id_utilizator1 int NOT NULL,
267 |     id_utilizator2 int NOT NULL,
268 |     id_analiza int NOT NULL,
269 |     timp_de_joc varchar2(10) NOT NULL,
270 |     PRIMARY KEY(id_partida),
271 |     FOREIGN KEY(id_utilizator1) REFERENCES Utilizatori(id_utilizator),
272 |     FOREIGN KEY(id_utilizator2) REFERENCES Utilizatori(id_utilizator)
273 );
274
275 INSERT INTO Partide
276 VALUES (1, 1, 2, 1, 'blitz');
277 INSERT INTO Partide
278 VALUES (2, 1, 3, 2, 'bullet');
279 INSERT INTO Partide
280 VALUES (3, 1, 4, 3, 'clasnic');
281 INSERT INTO Partide
282 VALUES (4, 1, 5, 4, 'blitz');
283 INSERT INTO Partide
284 VALUES (5, 2, 3, 5, 'blitz');
285
286 DESCRIBE Partide;
287 SELECT *
288 FROM Partide;
289

```

ID_PARTIDA	ID_UTILIZATOR1	ID_UTILIZATOR2	ID_ANALIZA	TIMP_DE_JOC
1	1	2	1	blitz
2	2	1	3	bullet
3	3	1	4	3clasnic
4	4	1	5	4blitz
5	5	2	3	5blitz

Tabela Analiză:

-Inserarea datelor in tabela Analiza

CREATE TABLE Analiza

```

(
    id_analiza int NOT NULL,
    id_partida int NOT NULL,
    id_mutare int NOT NULL,
    rezultat_analiza varchar2(3) NOT NULL,
    PRIMARY KEY(id_analiza),
    FOREIGN KEY(id_partida) REFERENCES Partide(id_partida)
);

```

INSERT INTO Analiza

```
VALUES (1, 1, 1, '1-0');
```

```
INSERT INTO Analiza
```

```
VALUES (2, 2, 2, '1/2');
```

```
INSERT INTO Analiza
```

```
VALUES (3, 3, 4, '1-0');
```

```
INSERT INTO Analiza
```

```
VALUES (4, 5, 3, '0-1');
```

```
INSERT INTO Analiza
```

```
VALUES (5, 4, 5, '1/2');
```

```
DESCRIBE Analiza;
```

```
SELECT *
```

```
FROM Analiza;
```

```
293
294 CREATE TABLE Analiza
295 (
296     id_analiza int NOT NULL,
297     id_partida int NOT NULL,
298     id_mutare int NOT NULL,
299     rezultat_analiza varchar2(3) NOT NULL,
300     PRIMARY KEY(id_analiza),
301     FOREIGN KEY(id_partida) REFERENCES Partide(id_partida)
302 );
303
304 INSERT INTO Analiza
305 VALUES (1, 1, 1, '1-0');
306 INSERT INTO Analiza
307 VALUES (2, 2, 2, '1/2');
308 INSERT INTO Analiza
309 VALUES (3, 3, 4, '1-0');
310 INSERT INTO Analiza
311 VALUES (4, 5, 3, '0-1');
312 INSERT INTO Analiza
313 VALUES (5, 4, 5, '1/2');
314
315 DESCRIBE Analiza;
316 SELECT *
317 FROM Analiza;
318
```

```
Script Output: X | Query Result: X
SQL icon Task completed in 0.203 seconds
```

```
1 row inserted.
```

```
1 row inserted.
```

Name	Null?	Type
ID_ANALIZA	NOT NULL	NUMBER(38)
ID_PARTIDA	NOT NULL	NUMBER(38)
ID_MUTARE	NOT NULL	NUMBER(38)
REZULTAT_ANALIZA	NOT NULL	VARCHAR2(3)

```
293
294 CREATE TABLE Analiza
295 (
296     id_analiza int NOT NULL,
297     id_partida int NOT NULL,
298     id_mutare int NOT NULL,
299     rezultat_analiza varchar2(3) NOT NULL,
300     PRIMARY KEY(id_analiza),
301     FOREIGN KEY(id_partida) REFERENCES Partide(id_partida)
302 );
303
304 INSERT INTO Analiza
```

Tabela Mutări:

--Inserarea datelor in tabela Mutari

```
CREATE TABLE Mutari
(
    id_mutare int NOT NULL,
    id_partida int NOT NULL,
    id_utilizator int NOT NULL,
    nr_mutare int NOT NULL,
    mutare varchar2(5) NOT NULL,
    PRIMARY KEY(id_mutare),
    FOREIGN KEY(id_partida) REFERENCES Partide(id_partida),
    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
);
```

```

INSERT INTO Mutari
VALUES (1, 1, 1, 1, 'e4');

INSERT INTO Mutari
VALUES (2, 2, 1, 1, 'e5');

INSERT INTO Mutari
VALUES (3, 3, 4, 2, 'Nf3');

INSERT INTO Mutari
VALUES (4, 4, 5, 3, 'Nf6');

INSERT INTO Mutari
VALUES (5, 5, 2, 10, 'O-O-O');

```

```

DESCRIBE Mutari;

SELECT *
FROM Mutari;

```

```

321    id_mutare int NOT NULL,
322    id_partida int NOT NULL,
323    id_utilizator int NOT NULL,
324    nr_mutare int NOT NULL,
325    mutare varchar2(5) NOT NULL,
326    PRIMARY KEY(id_mutare),
327    FOREIGN KEY(id_partida) REFERENCES Partide(id_partida),
328    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
329 );
330
331 INSERT INTO Mutari
332 VALUES (1, 1, 1, 1, 'e4');
333 INSERT INTO Mutari
334 VALUES (2, 2, 1, 1, 'e5');
335 INSERT INTO Mutari
336 VALUES (3, 3, 4, 2, 'Nf3');
337 INSERT INTO Mutari
338 VALUES (4, 4, 5, 3, 'Nf6');
339 INSERT INTO Mutari
340 VALUES (5, 5, 2, 10, 'O-O-O');
341
342 DESCRIBE Mutari;
343 SELECT *
344 FROM Mutari;
345
346

```

Task completed in 0.175 seconds

1 row inserted.

Name	Null?	Type
ID_MUTARE	NOT NULL	NUMBER(38)
ID_PARTIDA	NOT NULL	NUMBER(38)
ID_UTILIZATOR	NOT NULL	NUMBER(38)
NR_MUTARE	NOT NULL	NUMBER(38)
MUTARE	NOT NULL	VARCHAR2(5)

```

321     id_mutare int NOT NULL,
322     id_partida int NOT NULL,
323     id_utilizator int NOT NULL,
324     nr_mutare int NOT NULL,
325     mutare varchar2(5) NOT NULL,
326     PRIMARY KEY(id_mutare),
327     FOREIGN KEY(id_partida) REFERENCES Partide(id_partida),
328     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator)
329 );
330
331 INSERT INTO Mutari
332 VALUES (1, 1, 1, 1, 'e4');
333 INSERT INTO Mutari
334 VALUES (2, 2, 1, 1, 'e5');
335 INSERT INTO Mutari
336 VALUES (3, 3, 4, 2, 'Nf3');
337 INSERT INTO Mutari
338 VALUES (4, 4, 5, 3, 'Nf6');
339 INSERT INTO Mutari
340 VALUES (5, 5, 2, 10, 'O-O-O');
341
342 DESCRIBE Mutari;
343 SELECT *
344 FROM Mutari;
345
346

```

All Rows Fetched 5 in 0.001 seconds

ID_MUTARE	ID_PARTIDA	ID_UTILIZATOR	NR_MUTARE	MUTARE
1	2	1	1	1e4
2	2	2	1	1e5
3	3	3	4	2Nf3
4	4	4	5	3Nf6
5	5	5	2	10O-O-O

Tabela asociativă Au (Utilizatori<->Rating):

--Inserarea datelor in tabela asociativa UtilizatoriRating

CREATE TABLE UtilizatoriRating

```

(
    id_utilizator int NOT NULL,
    id_rating int NOT NULL,
    PRIMARY KEY(id_utilizator, id_rating),
    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator),
    FOREIGN KEY(id_rating) REFERENCES Rating(id_rating)
);

```

INSERT INTO UtilizatoriRating

VALUES (1, 1500);

```
INSERT INTO UtilizatoriRating
VALUES (1, 2500);

INSERT INTO UtilizatoriRating
VALUES (2, 2500);

INSERT INTO UtilizatoriRating
VALUES (1, 250);

INSERT INTO UtilizatoriRating
VALUES (5, 1500);

INSERT INTO UtilizatoriRating
VALUES (3, 1867);

INSERT INTO UtilizatoriRating
VALUES (3, 3175);

INSERT INTO UtilizatoriRating
VALUES (4, 2000);

INSERT INTO UtilizatoriRating
VALUES (4, 1750);

INSERT INTO UtilizatoriRating
VALUES (5, 3175);

DESCRIBE UtilizatoriRating;
SELECT *
FROM UtilizatoriRating;
```

```
358| INSERT INTO UtilizatoriRating  
359| VALUES (1, 1500);  
360| INSERT INTO UtilizatoriRating  
361| VALUES (1, 2500);  
362| INSERT INTO UtilizatoriRating  
363| VALUES (2, 2500);  
364| INSERT INTO UtilizatoriRating  
365| VALUES (1, 250);  
366| INSERT INTO UtilizatoriRating  
367| VALUES (1, 1500);  
368| INSERT INTO UtilizatoriRating  
369| VALUES (3, 1867);  
370| INSERT INTO UtilizatoriRating  
371| VALUES (3, 3175);  
372| INSERT INTO UtilizatoriRating  
373| VALUES (4, 2000);  
374| INSERT INTO UtilizatoriRating  
375| VALUES (4, 1750);  
376| INSERT INTO UtilizatoriRating  
377| VALUES (5, 3175);  
378|  
379| DESCRIBE UtilizatoriRating;  
380| SELECT *  
381| FROM Mutari;  
382|  
383|
```

Script Output: X | Query Result: X
Task completed in 0.186 seconds

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER(38)
ID_RATING	NOT NULL	NUMBER(38)

```

358 INSERT INTO UtilizatoriRating
359 VALUES (1, 1500);
360 INSERT INTO UtilizatoriRating
361 VALUES (1, 2500);
362 INSERT INTO UtilizatoriRating
363 VALUES (2, 2500);
364 INSERT INTO UtilizatoriRating
365 VALUES (1, 250);
366 INSERT INTO UtilizatoriRating
367 VALUES (5, 1500);
368 INSERT INTO UtilizatoriRating
369 VALUES (3, 1867);
370 INSERT INTO UtilizatoriRating
371 VALUES (3, 3175);
372 INSERT INTO UtilizatoriRating
373 VALUES (4, 2000);
374 INSERT INTO UtilizatoriRating
375 VALUES (4, 1750);
376 INSERT INTO UtilizatoriRating
377 VALUES (5, 3175);
378
379 DESCRIBE UtilizatoriRating;
380 SELECT *
381 FROM UtilizatoriRating;
382
383

```

The screenshot shows the MySQL Workbench interface with two tabs: 'Script Output' and 'Query Result'. The 'Script Output' tab contains the SQL code from line 358 to 383. The 'Query Result' tab shows a table named 'UTILIZATORI' with columns 'ID' and 'RATING'. The data is as follows:

ID	RATING
1	1500
2	2500
3	2500
4	250
5	1867
6	3175
7	3175
8	2000
9	1750
10	3175
5	1500

Tabelă asociativă Rezolvă:

--Inserarea datelor in tabela asociativa UtilizatoriPuzzles

```

CREATE TABLE UtilizatoriPuzzles
(
    id_utilizator int NOT NULL,
    id_puzzle int NOT NULL,
    PRIMARY KEY(id_utilizator, id_puzzle),
    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator),
    FOREIGN KEY(id_puzzle) REFERENCES Puzzles(id_puzzle)
);

```

```

INSERT INTO UtilizatoriPuzzles
VALUES (1, 1);

```

```
INSERT INTO UtilatoriPuzzles
VALUES (1, 2);
INSERT INTO UtilatoriPuzzles
VALUES (2, 1);
INSERT INTO UtilatoriPuzzles
VALUES (1, 3);
INSERT INTO UtilatoriPuzzles
VALUES (5, 4);
INSERT INTO UtilatoriPuzzles
VALUES (3, 4);
INSERT INTO UtilatoriPuzzles
VALUES (3, 5);
INSERT INTO UtilatoriPuzzles
VALUES (4, 5);
INSERT INTO UtilatoriPuzzles
VALUES (4, 1);
INSERT INTO UtilatoriPuzzles
VALUES (5, 2);

DESCRIBE UtilatoriPuzzles;
SELECT *
FROM UtilatoriPuzzles;
```

```
394
395 INSERT INTO UtilizatoriPuzzles
396 VALUES (1, 1);
397 INSERT INTO UtilizatoriPuzzles
398 VALUES (1, 2);
399 INSERT INTO UtilizatoriPuzzles
400 VALUES (2, 1);
401 INSERT INTO UtilizatoriPuzzles
402 VALUES (1, 3);
403 INSERT INTO UtilizatoriPuzzles
404 VALUES (5, 4);
405 INSERT INTO UtilizatoriPuzzles
406 VALUES (3, 4);
407 INSERT INTO UtilizatoriPuzzles
408 VALUES (3, 5);
409 INSERT INTO UtilizatoriPuzzles
410 VALUES (4, 5);
411 INSERT INTO UtilizatoriPuzzles
412 VALUES (4, 1);
413 INSERT INTO UtilizatoriPuzzles
414 VALUES (5, 2);
415
416 DESCRIBE UtilizatoriPuzzles;
417 SELECT *
418 FROM UtilizatoriPuzzles;
419
```

Script Output x | Query Result x
✖️ ↻ ⌂ 🗑 Task completed in 0.185 seconds

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER(38)
ID_PUZZLE	NOT NULL	NUMBER(38)

```

394 |
395 INSERT INTO UtilizatoriPuzzles
396 VALUES (1, 1);
397 INSERT INTO UtilizatoriPuzzles
398 VALUES (1, 2);
399 INSERT INTO UtilizatoriPuzzles
400 VALUES (2, 1);
401 INSERT INTO UtilizatoriPuzzles
402 VALUES (1, 3);
403 INSERT INTO UtilizatoriPuzzles
404 VALUES (5, 4);
405 INSERT INTO UtilizatoriPuzzles
406 VALUES (3, 4);
407 INSERT INTO UtilizatoriPuzzles
408 VALUES (3, 5);
409 INSERT INTO UtilizatoriPuzzles
410 VALUES (4, 5);
411 INSERT INTO UtilizatoriPuzzles
412 VALUES (4, 1);
413 INSERT INTO UtilizatoriPuzzles
414 VALUES (5, 2);
415 |
416 DESCRIBE UtilizatoriPuzzles;
417 SELECT *
418 FROM UtilizatoriPuzzles;
419

```

The screenshot shows the MySQL Workbench interface. At the top, there is a script editor window containing the SQL code from above. Below it is a results window titled "Query Result". The results show a table named "UtilizatoriPuzzles" with columns "ID_UTILIZATOR" and "ID_PUZZLE". The data is as follows:

ID_UTILIZATOR	ID_PUZZLE
1	1
2	1
3	2
4	1
5	3
6	3
7	3
8	4
9	4
10	5

Tabelă asociativă au (Utilizator<->Turnee):

--Inserarea datelor in tabela asociativa UtilizatoriPrietenie

CREATE TABLE UtilizatoriTurnee

```

(
    id_utilizator int NOT NULL,
    id_turneu int NOT NULL,
    PRIMARY KEY(id_utilizator, id_turneu),
    FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator),
    FOREIGN KEY(id_turneu) REFERENCES Turnee(id_turneu)
);
```

INSERT INTO UtilizatoriTurnee

VALUES (1, 1);

```
INSERT INTO UtilizatoriTurnee
VALUES (1, 2);
INSERT INTO UtilizatoriTurnee
VALUES (1, 3);
INSERT INTO UtilizatoriTurnee
VALUES (1, 4);
INSERT INTO UtilizatoriTurnee
VALUES (1, 5);
INSERT INTO UtilizatoriTurnee
VALUES (2, 1);
INSERT INTO UtilizatoriTurnee
VALUES (2, 2);
INSERT INTO UtilizatoriTurnee
VALUES (2, 3);
INSERT INTO UtilizatoriTurnee
VALUES (3, 1);
INSERT INTO UtilizatoriTurnee
VALUES (3, 2);
INSERT INTO UtilizatoriTurnee
VALUES (3, 3);
INSERT INTO UtilizatoriTurnee
VALUES (3, 5);
INSERT INTO UtilizatoriTurnee
VALUES (4, 2);
INSERT INTO UtilizatoriTurnee
VALUES (4, 4);
```

```

INSERT INTO UtilizatoriTurnee
VALUES (5, 1);

INSERT INTO UtilizatoriTurnee
VALUES (5, 2);

INSERT INTO UtilizatoriTurnee
VALUES (5, 3);

INSERT INTO UtilizatoriTurnee
VALUES (5, 4);

INSERT INTO UtilizatoriTurnee
VALUES (5, 5);

```

```

DESCRIBE UtilizatoriTurnee;

SELECT *

FROM UtilizatoriTurnee;

```

```

450 INSERT INTO UtilizatoriTurnee
451 VALUES (3, 2);
452 INSERT INTO UtilizatoriTurnee
453 VALUES (3, 3);
454 INSERT INTO UtilizatoriTurnee
455 VALUES (3, 5);
456 INSERT INTO UtilizatoriTurnee
457 VALUES (4, 2);
458 INSERT INTO UtilizatoriTurnee
459 VALUES (4, 4);
460 INSERT INTO UtilizatoriTurnee
461 VALUES (5, 1);
462 INSERT INTO UtilizatoriTurnee
463 VALUES (5, 2);
464 INSERT INTO UtilizatoriTurnee
465 VALUES (5, 3);
466 INSERT INTO UtilizatoriTurnee
467 VALUES (5, 4);
468 INSERT INTO UtilizatoriTurnee
469 VALUES (5, 5);
470
471 DESCRIBE UtilizatoriTurnee;
472 SELECT *
473 FROM UtilizatoriTurnee;
474
475

```

Script Output Query Result | Task completed in 0.683 seconds

1 row inserted.

1 row inserted.

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER(38)
ID_TURNEU	NOT NULL	NUMBER(38)

```

450| INSERT INTO UtilizatoriTurnee
451| VALUES (3, 2);
452| INSERT INTO UtilizatoriTurnee
453| VALUES (3, 3);
454| INSERT INTO UtilizatoriTurnee
455| VALUES (3, 5);
456| INSERT INTO UtilizatoriTurnee
457| VALUES (4, 2);
458| INSERT INTO UtilizatoriTurnee
459| VALUES (4, 4);
460| INSERT INTO UtilizatoriTurnee
461| VALUES (5, 1);
462| INSERT INTO UtilizatoriTurnee
463| VALUES (5, 2);
464| INSERT INTO UtilizatoriTurnee
465| VALUES (5, 3);
466| INSERT INTO UtilizatoriTurnee
467| VALUES (5, 4);
468| INSERT INTO UtilizatoriTurnee
469| VALUES (5, 5);
470|
471| DESCRIBE UtilizatoriTurnee;
472| SELECT *
473| FROM UtilizatoriTurnee;
474|
475|

```

Screenshot of a SQL query result showing the data inserted into the UtilizatoriTurnee table:

	id_Utilizator	id_Turneu
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	2	2
7	2	3
8	3	1
9	3	2
10	3	3
11	4	2
12	4	3
13	4	4
14	5	1
15	5	2
16	5	3
17	5	4
18	5	5
19	2	1

Tabelă asociativă au (Partide<->Mutări<->Analiză):

--Inserarea datelor in tabela asociativa PartideInfo

CREATE TABLE PartideInfo

```

(
    id_partida int NOT NULL,
    id_analiza int NOT NULL,
    id_mutare int NOT NULL,
    PRIMARY KEY(id_partida, id_analiza, id_mutare),

```

```
    FOREIGN KEY(id_partida) REFERENCES Partide(id_partida),
    FOREIGN KEY(id_analiza) REFERENCES Analiza(id_analiza),
    FOREIGN KEY(id_mutare) REFERENCES Mutari(id_mutare)
);
```

```
INSERT INTO PartideInfo
```

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

```
INSERT INTO PartideInfo
```

```
VALUES (1, 1, 2);
```

```
INSERT INTO PartideInfo
```

```
VALUES (2, 1, 1);
```

```
INSERT INTO PartideInfo
```

```
VALUES (2, 1, 2);
```

```
INSERT INTO PartideInfo
```

```
VALUES (3, 1, 4);
```

```
INSERT INTO PartideInfo
```

```
VALUES (3, 2, 5);
```

```
INSERT INTO PartideInfo
```

```
VALUES (4, 4, 3);
```

```
INSERT INTO PartideInfo
```

```
VALUES (4, 3, 3);
```

```
INSERT INTO PartideInfo
```

```
VALUES (5, 5, 5);
```

```
INSERT INTO PartideInfo
```

```
VALUES (1, 1, 3);
```

```
DESCRIBE PartideInfo;
```

```
SELECT *
```

```
FROM PartideInfo;
```

```
490 INSERT INTO PartideInfo
491 VALUES (1, 1, 1);
492 INSERT INTO PartideInfo
493 VALUES (1, 1, 2);
494 INSERT INTO PartideInfo
495 VALUES (2, 1, 1);
496 INSERT INTO PartideInfo
497 VALUES (2, 1, 2);
498 INSERT INTO PartideInfo
499 VALUES (3, 1, 4);
500 INSERT INTO PartideInfo
501 VALUES (3, 2, 5);
502 INSERT INTO PartideInfo
503 VALUES (4, 4, 3);
504 INSERT INTO PartideInfo
505 VALUES (4, 3, 3);
506 INSERT INTO PartideInfo
507 VALUES (5, 5, 5);
508 INSERT INTO PartideInfo
509 VALUES (1, 1, 3);
510
511
512 DESCRIBE PartideInfo;
513 SELECT *
514 FROM PartideInfo;
515
```

```
Script Output * | Query Result *
✖️ 🚫 🛡️ 🛡️ Task completed in 0.665 seconds
Error report -
ORA-00001: unique constraint (SYS.SYS_C008426) violated
```

```
1 row inserted.
```

Name	Null?	Type
ID_PARTIDA	NOT NULL	NUMBER(38)
ID_ANALIZA	NOT NULL	NUMBER(38)
ID_MUTARE	NOT NULL	NUMBER(38)

```

490 INSERT INTO PartideInfo
491 VALUES (1, 1, 1);
492 INSERT INTO PartideInfo
493 VALUES (1, 1, 2);
494 INSERT INTO PartideInfo
495 VALUES (2, 1, 1);
496 INSERT INTO PartideInfo
497 VALUES (2, 1, 2);
498 INSERT INTO PartideInfo
499 VALUES (3, 1, 4);
500 INSERT INTO PartideInfo
501 VALUES (3, 2, 5);
502 INSERT INTO PartideInfo
503 VALUES (4, 4, 3);
504 INSERT INTO PartideInfo
505 VALUES (4, 3, 3);
506 INSERT INTO PartideInfo
507 VALUES (5, 5, 5);
508 INSERT INTO PartideInfo
509 VALUES (1, 1, 3);
510
511
512 DESCRIBE PartideInfo;
513 SELECT *
514 FROM PartideInfo;
515

```

Script Output | Query Result | All Rows Fetched: 10 in 0.002 seconds

ID_PARTIDA	ID_ANALIZA	ID_MUTARE
1	1	1
2	1	2
3	2	1
4	2	1
5	3	1
6	3	2
7	4	3
8	4	3
9	5	5
10	1	3

Tabela asociativă au (Utilizatori<->Notificări):

----Inserarea datelor in tabela asociativa UtilizatoriNotificari

CREATE TABLE UtilizatoriNotificari

(

```

    id_utilizator int NOT NULL,
    id_notificare int NOT NULL,
    PRIMARY KEY(id_utilizator, id_notificare),
    FOREIGN KEY(id_utilizator) REFERENCES
    Utilizatori(id_utilizator),

```

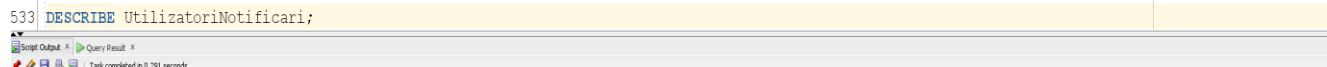
```
    FOREIGN KEY(id_notificare) REFERENCES
Notificari(id_notificare)
);
```

```
DESCRIBE UtilizatoriNotificari;
```

```
INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 2);
INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 1);
INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 3);
INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 5);
INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 4);
```

```
SELECT *
FROM UtilizatoriNotificari;
```

```
517#CREATE SEQUENCE seq
518 START WITH 1
519 INCREMENT BY 1
520 MINVALUE 1
521 MAXVALUE 5;
522
523
524#CREATE TABLE UtilizatoriNotificari
525 (
526     id_utilizator int NOT NULL,
527     id_notificare int NOT NULL,
528     PRIMARY KEY(id_utilizator, id_notificare),
529     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator),
530     FOREIGN KEY(id_notificare) REFERENCES Notificari(id_notificare)
531 );
532
533 DESCRIBE UtilizatoriNotificari;
```



```
ID_PARTIDA NOT NULL NUMBER(38)
ID_ANALIZA NOT NULL NUMBER(38)
ID_MUTARE NOT NULL NUMBER(38)
```

Table UTILIZATORINOTIFICARI created.

Name	Null?	Type
ID_UTILIZATOR	NOT NULL	NUMBER(38)
ID_NOTIFICARE	NOT NULL	NUMBER(38)

```

517=CREATE SEQUENCE seq
518 START WITH 1
519 INCREMENT BY 1
520 MINVALUE 1
521 MAXVALUE 5;
522
523
524=CREATE TABLE UtilizatoriNotificari
525 (
526     id_utilizator int NOT NULL,
527     id_notificare int NOT NULL,
528     PRIMARY KEY(id_utilizator, id_notificare),
529     FOREIGN KEY(id_utilizator) REFERENCES Utilizatori(id_utilizator),
530     FOREIGN KEY(id_notificare) REFERENCES Notificari(id_notificare)
531 );
532
533 DESCRIBE UtilizatoriNotificari;
534
535 INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 2);
536 INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 1);
537 INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 3);
538 INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 5);
539 INSERT INTO UtilizatoriNotificari VALUES (seq.nextval, 4);
540
541 SELECT *
542 FROM UtilizatoriNotificari;

```

Script Output | Query Result |
 SQL | All Rows Fetched: 5 in 0.002 seconds

ID UTILIZATOR	ID NOTIFICARE
1	2
2	1
3	3
4	5
5	4

12. Formulati în limbaj natural și implementați 5 cereri SQL complexe

Cerințele 1 și 2:

--Sa se afiseze toate id-urile si email-urile utilizatorilor care au rating-ul 1500 si au si o conversatie cu utilizatorul 2

--si mai e si prieten cu utilizatorul cu id-ul 5. (Cerințele 1 si 2) (Join pe 4 tabele)

```

SELECT ut.id_utilizator, u.email          --1 rezultat, utilizatorul cu id-ul 1 si
mailu-1 ionut@gmail.com indeplineste toate conditiile

FROM UtilizatoriRating ut, Utilizatori u, Conversatii c

WHERE u.id_utilizator = ut.id_utilizator

AND ut.id_rating = 1500

AND ut.id_utilizator IN

    (SELECT tabela.id_utilizator1

     FROM (SELECT id_utilizator1

            FROM Prietenie

            WHERE id_utilizator2 = 5) tabela

     WHERE tabela.id_utilizator1 = c.id_utilizator1

     AND c.id_utilizator2 = 2);

```

```

1--Sa se afiseze toate idurile si emailurile utilizatorilor care au rating-ul 1500 si au si o conversatie cu utilizatorul 2
2--si mai e si prieten cu utilizatorul cu id-ul 5. (Cerintele 1 si 2) (Join pe 4 tabele)
3
4SELECT ut.id_utilizator, u.email          --1 rezultat, utilizatorul cu id-ul 1 si mailu-1 ionut@gmail.com indeplineste toate conditiile
5FROM UtilizatoriRating ut, Utilizatori u, Conversatii c
6WHERE u.id_utilizator = ut.id_utilizator
7AND ut.id_rating = 1500
8AND ut.id_utilizator IN
9    (SELECT tabela.id_utilizator1
10     FROM (SELECT id_utilizator1
11            FROM Prietenie
12            WHERE id_utilizator2 = 5) tabela
13     WHERE tabela.id_utilizator1 = c.id_utilizator1
14     AND c.id_utilizator2 = 2);
15
16
17--Sa se afiseze cati utilizatori au participat la turneul cu id-ul 2, au incercat sa rezolve puzzle-ul 1
18--Si care au avut macar o plata cu suma 15 (Cerinta 3)
19
20SELECT SUM(COUNT(*)) --3 utilizatori indeplinesc aceste conditii (cei cu idurile 1,2 si 4)
21FROM UtilizatoriTurnee

```

Script Output: All Rows Fetched: 1 in 0.001 seconds

ID UTILIZATOR	EMAIL
1	lionut@gmail.com

Cerință 3:

--Sa se afiseze cati utilizatori au participat la turneul cu id-ul 2, au incercat sa rezolve puzzle-ul 1

--Si care au avut macar o plată cu suma 15 (Cerinta 3)

--3 utilizatori indeplinesc aceste conditii (cei cu id-urile 1,2 si 4)

```
SELECT SUM(COUNT(*))
FROM UtilizatoriTurnee
WHERE id_turneu = 2
AND id_utilizator IN
    (SELECT id_utilizator
     FROM UtilizatoriPuzzles
     WHERE id_puzzle = 1
     AND id_utilizator IN
         (SELECT u.id_utilizator
          FROM Utilizatori u, Plati p
          WHERE u.id_utilizator = p.id_utilizator
          AND p.suma = 15))
GROUP BY id_utilizator;
```

```
16
17--Sa se afiseze cati utilizatori au participat la turneul cu id-ul 2, au incercat sa rezolve puzzle-ul 1
18--Si care au avut macar o plată cu suma 15 (Cerinta 3)
19--3 utilizatori indeplinesc aceste conditii (cei cu id-urile 1,2 si 4)
20
21*SELECT SUM(COUNT(*))
22 FROM UtilizatoriTurnee
23 WHERE id_turneu = 2
24 AND id_utilizator IN
25*   (SELECT id_utilizator
26     FROM UtilizatoriPuzzles
27     WHERE id_puzzle = 1
28     AND id_utilizator IN
29       (SELECT u.id_utilizator
30         FROM Utilizatori u, Plati p
31         WHERE u.id_utilizator = p.id_utilizator
32         AND p.suma = 15))
33 GROUP BY id_utilizator;
34
35
```



Cerință 4:

--Afisati informatiile despre partidele care s-au jucat la stilul de joc blitz in ordine descrescatoare (Cerinta 4)

```
SELECT *
FROM PARTIDE
WHERE DECODE(nvl(timp_de_joc,0), 'blitz' , 1) = 1
ORDER BY id_analiza DESC;
```

```
35 --Afisati informatiile despre partidele care s-au jucat la stilul de joc blitz in ordine descrescatoare (Cerinta 4)
36
37 SELECT *
38 FROM PARTIDE
39 WHERE DECODE(nvl(timp_de_joc,0), 'blitz' , 1) = 1
40 ORDER BY id_analiza DESC;
41
42
```

Script Output X Query Result X
SQL | All Rows Fetched: 3 in 0.002 seconds

ID_PARTIDA	ID_UTILIZATOR1	ID_UTILIZATOR2	ID_ANALIZA	TIMP_DE_JOC
1	5	2	3	5blitz
2	4	1	5	4blitz
3	1	1	2	1blitz

Cerință 5:

--(Cerinta 5) Sa se afiseze mesajul 'exista' daca exista utilizatori care au incercat sa rezolve puzzle-ul care

--in pozitia de start are un turn pe a1 (in componenta atributului pozitie e secventa Ra1)

--si care au lungimea username-ului 4 sau 5

--si care au jucat in macar un turneu care a inceput in 2023

--si care au initiat o prietenie in martie aprilie sau iunie (sunt in atributul id_utilizator1)

--Daca nu exista niciunul, se afiseaza 'nu exista'

```
SELECT CASE
```

```

        WHEN COUNT(DISTINCT id_utilizator) > 0 THEN 'exista' --Avem
utilizatorii 1 si 2, deci exista

        ELSE 'nu exista'

        END

FROM UtilizatoriTurnee

WHERE id_turneu IN

        (SELECT id_turneu --Turneele 1,2,3,4

        FROM Turnee

        WHERE data_start >= '01-JAN-2023')

AND id_utilizator IN

        (SELECT id_utilizator --tot Utilizatorii 1,2 si 4

        FROM Utilizatori

        WHERE LENGTH(username) = 4 OR LENGTH(username) = 5

        AND id_utilizator in

                (SELECT id_utilizator --Utilizatorii 1, 2 si 4

                FROM UtilizatoriPUZZLES

                WHERE id_puzzle =

                        (SELECT id_puzzle --pozitia cu numarul 1

                        FROM Puzzles

                        WHERE Pozitie like '%Ra1%')))

AND id_utilizator IN

        (SELECT DISTINCT(id_utilizator1) --utilizatorii 1 si 2

        FROM Prietenie

        WHERE EXTRACT (MONTH FROM data_inceput) = 3 OR EXTRACT
(MONTH FROM data_inceput) = 4 OR EXTRACT (MONTH FROM data_inceput) =
6);

```

```

41| --Sa se afiseze mesajul 'exista' daca exista utilizatori care au incercat sa rezolve puzzle-ul care
42| --in pozitia de start are un turn pe al (in componenta atributului pozitie e secventa Ra)
43| --si care au lungimea username-ului 4 sau 5
44| --si care au jucat in macar un turneu care a inceput in 2023
45| --si care au initiat o prietenie in martie aprilie sau iunie (sunt in atributul id_utilizator1)
46| --Daca nu exista niciunul, se afiseaza 'nu exista'
47|
48|
49| SELECT CASE
50|     WHEN COUNT(DISTINCT id_utilizator) > 0 THEN 'exista' --Avem utilizatorii 1 si 2, deci exista
51|     ELSE 'nu exista'

```

Cerință 6:

--(Cerinta 6) Sa se afiseze toate informatiile despre turneele
--care au un numar de participanti mai mare sau egal cu numarul mediu de
participantii de la un turneu.
--Sa se afiseze si numarul mediu de participanti la un turneu. Afisarea se va
face printr-o coloana separata denumit 'AverageNumber'.

WITH tabela(AverageNumber) AS
(SELECT avg(nr_participanti)

```

FROM Turnee)

SELECT *

FROM turnee, tabela

WHERE nr_participanti >= tabela.AverageNumber;

```

```

74--(Cerinta 6) Sa se afiseze toate informatiile despre turneele
75--care au un numar de participanti mai mare sau egal cu numarul mediu de participantii de la un turneu.
76--Sa se afiseze si numarul mediu de participanti la un turneu. Afisarea se va face printr-o coloana separata denumita 'AverageNumber'.
77
78WITH tabela(AverageNumber) AS
79    (SELECT avg(nr_participanti)
80     FROM Turnee)
81    SELECT *
82    FROM turnee, tabela
83    WHERE nr_participanti >= tabela.AverageNumber;

```

Script Output | Query Result 1 | Query Result 2 |

All Rows Fetched: 3 in 0.002 seconds

ID_TURNEU	NR_PARTICIPANTI	DATA_START	DURATA	AVERAGENUMBER
1	1	531-MAR-23	7	3.8
2	3	431-JAN-23	12	3.8
3	5	529-FEB-20	15	3.8

13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri

a)

--cerinta a)

INSERT INTO Mutari VALUES (6,1,1,11,'Bf5'); --Inseram temporar o instructiune care nu are copii in alte tabele pentru a o putea sterge apoi

DELETE FROM Mutari --Acum putem sterge toate informatiile din tabela Mutari in care mutarea efectuata este 11

WHERE nr_mutare = 11;

```

86--Exercitiul 13
87INSERT INTO Mutari VALUES (6,1,1,11,'Bf5'); --Inseram temporar o instructiune care nu are copii in alte tabele pentru a o putea sterge apoi
88DELETE FROM Mutari --Acum putem sterge toate informatiile din tabela Mutari in care mutarea efectuata este 11
89WHERE nr_mutare = 11;
90
91SELECT * FROM Mutari;

```

Script Output | Query Result 1 | Query Result 2 |

Task completed in 0.135 seconds

b)

--cerinta b)

INSERT INTO Mutari VALUES(6,1,1,15,'Bf5'); --Inseram temporar o instructiune care nu are copii in alte tabele pentru a o putea modifica.

--De exemplu, sa spunem ca am introdus din gresala mutarea Bf5 la mutare 15,

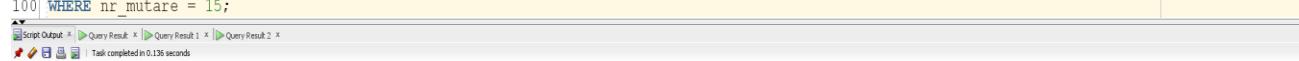
--in loc de mutarea 11 cum ar fi fost corect. Voi face acum modificarea utilizand functia de UPDATE:

UPDATE MUTARI

```
SET id_mutare = 6, id_partida = 1, id_utilizator = 1, nr_mutare = 11, mutare = 'Bf5'
```

```
WHERE nr_mutare = 15;
```

```
94|INSERT INTO Mutari VALUES(6,1,1,15,'Bf5'); --Inseram temporar o instructiune care nu are copii in alte tabele pentru a o putea modifica.  
95|  
96|  
97|  
98|UPDATE MUTARI  
99|SET id_mutare = 6, id_partida = 1, id_utilizator = 1, nr_mutare = 11, mutare = 'Bf5'  
100|WHERE nr_mutare = 15;
```



0 rows updated.

1 row updated.

c)

--cerinta c)

--Acum putem sa stergem linia inserata la punctul b), pentru a lasa tabela asa cum era initial.

--Ca sa facem lucrurile putin mai complicate, sa presupunem ca vrem sa stergem

--din tabela Mutari toate informatille

--Din mutaril care sunt in partidele cu id-urile 1 si 3

DELETE FROM MUTARI

WHERE id_mutare not in (SELECT id_mutare

FROM PartideInfo

WHERE id_partida = 1 OR id_partida = 3);

```
102--cerinta c)
103--Acum putem sa stergem linia inserata la punctul b), pentru a lasa tabela asa cum era initial.
104--Ca sa facem lucrurile putin mai complicate, sa presupunem ca vrem sa stergem
105--din tabela Mutari toate informatille
106--Din mutaril care sunt in partidele cu id-urile 1 si 3
107DELETE FROM MUTARI
108 WHERE id_mutare not in (SELECT id_mutare
109     FROM PartideInfo
110     WHERE id_partida = 1 OR id_partida = 3);
```

1 row inserted.

1 row deleted.

15. Formulați în limbaj natural și implementați în SQL 3 cereri complexe

a)

--cerinta a) (Outer Join pe 7 tabele)

--Selectati username-ul utilizatorilor care au primit o notificare cu un text in care se mentioneaza ca

--participa maestri la un turneu ce va incepe in curand

--si titlul notificarii are cuvantul cheie Titled

--si care a facut o plata de 15\$

--si care a initiat o conversatie cu utilizatorul cu id-ul 2

--si care a initiat o conversatie cu utilizatorul cu numarul 4

SELECT DISTINCT(u.username)

FROM Utilizatori u, UtilizatoriNotificari UN, Plati P

WHERE u.id_utilizator = un.id_utilizator

AND u.id_utilizator IN

(SELECT UN.id_utilizator

FROM UtilizatoriNotificari UN, Notificari n

WHERE un.id_notificare = n.id_notificare

AND titlu LIKE '%Titled%'

AND un.id_utilizator IN(SELECT UN.id_utilizator

FROM UtilizatoriNotificari UN, Mesaj m

```
        WHERE UN.id_notificare = m.id_notificare
        AND m.id_notificare IN (SELECT m.id_notificare
                                FROM Mesaj
                                WHERE m.Text LIKE '%maestri%'))
AND u.id_utilizator IN
        (SELECT id_utilizator
         FROM Plati
         WHERE id_utilizator = u.id_utilizator
         AND suma = 15)
AND u.id_utilizator IN
        (SELECT id_utilizator1
         FROM Conversatii
         WHERE id_utilizator1 = u.id_utilizator
         AND id_utilizator2 = 2)
AND u.id_utilizator IN
        (SELECT id_utilizator1
         FROM Prietenie
         WHERE id_utilizator1 = u.id_utilizator
         AND id_utilizator2 = 4);
```

```

115 --cerinta a) (Outer Join pe 7 tabele)
116 --Selectati username-ul utilizatorilor care au primit o notificare cu un text in care se mentioneaza ca
117 --participa maestri la un turneu ce va incepe in curand
118 --si titlul notificarii are cuvantul cheie Titled
119 --si care a facut o plată de 15$
120 --si care a initiat o conversatie cu utilizatorul cu id-ul 2
121 --si care a initiat o conversatie cu utilizatorul cu numarul 4
122 SELECT DISTINCT(u.username)
123 FROM Utilizatori u, UtilizatoriNotificari UN, Plati P
124 WHERE u.id_utilizator = un.id_utilizator
125 AND u.id_utilizator IN
126=     (SELECT UN.id_utilizator
127     FROM UtilizatoriNotificari UN, Notificari n
128     WHERE un.id_notificare = n.id_notificare
129     AND titlu LIKE '%Titled%'
130     AND un.id_utilizator IN (SELECT UN.id_utilizator
131             FROM UtilizatoriNotificari UN, Mesaj m
132             WHERE UN.id_notificare = m.id_notificare
133             AND m.id_notificare IN (SELECT m.id_notificare
134                     FROM Mesaj
135                     WHERE m.Text LIKE '%maestri%'))
136 AND u.id_utilizator IN
137=     (SELECT id_utilizator
138     FROM Plati
139     WHERE id_utilizator = u.id_utilizator
140     AND suma = 15)
141 AND u.id_utilizator IN |
142=     (SELECT id_utilizator1
143     FROM Conversatii
144     WHERE id_utilizator1 = u.id_utilizator
145     AND id_utilizator2 = 2)
146 AND u.id_utilizator IN
147=     (SELECT id_utilizator1
148     FROM Prietenie
149     WHERE id_utilizator1 = u.id_utilizator
150     AND id_utilizator2 = 4);

```

b)

--cerinta b) (Division)

--Sa se afiseze toate id-urile utilizatorilor cu parola cu cel putin 1 in componenta si care fac parte dintr-un turneu cu id-ul 1 sau 5.

```
SELECT id_utilizator  
FROM Utilizatori u
```

```

WHERE id_utilizator IN
    (SELECT id_utilizator
     FROM Utilizatori
     WHERE parola like '%l%')
GROUP BY id_utilizator
HAVING COUNT(id_utilizator)<=
    (SELECT COUNT(*)
     FROM UtilizatoriTurnee
     WHERE u.id_utilizator = id_utilizator
     AND id_turneu = 1 OR id_turneu = 5);

```

```

152 --cerinta b) (Division)
153 --Sa se afiseze toate id-urile utilizatorilor cu parola cu cel putin 1 in componenta si care fac parte dintr-un turneu cu id-ul 1 sau 5.
154
155=SELECT id_utilizator
156 FROM Utilizatori u
157 WHERE id_utilizator IN
158     (SELECT id_utilizator
159      FROM Utilizatori
160      WHERE parola like '%l%')
161 GROUP BY id_utilizator
162 HAVING COUNT(id_utilizator)<=
163     (SELECT COUNT(*)
164      FROM UtilizatoriTurnee
165      WHERE u.id_utilizator = id_utilizator
166      AND id_turneu = 1 OR id_turneu = 5);

```

Script Output | Query Result | Query Result 1 | Query Result 2

SQL | All Rows Fetched: 2 in 0.004 seconds

ID_UTILIZATOR
1
2
5

c) (pentru n citit de la tastatura care a avut valoarea 3)

--cerinta c) (top-n)

--Sa se afiseze informatiile primilor n utilizatorilor ordonati in mod descrescator dupa numele emailurilor, unde n e un nr nat>0 dat de la tastatura.

```

SELECT *
FROM (SELECT *
      FROM Utilizatori
      ORDER BY email DESC) ordonati_desc
WHERE ROWNUM <= &n;

```

168 --cerinta c) (top-n)
169 --Sa se afiseze informatiile primilor n utilizatorilor ordonati in mod descrescator dupa numele emailurilor, unde n e un nr nat>0 dat de la tastatura
170
171 #SELECT *
172 FROM (SELECT *
173 FROM Utilizatori
174 ORDER BY email DESC) ordonati_desc
175 WHERE ROWNUM <= &n;



The screenshot shows the Oracle SQL Developer interface. At the top, there are tabs for 'Script Output', 'Query Result' (which is active), 'Query Result 1', and 'Query Result 2'. Below the tabs, the SQL command is displayed. In the main area, a table is shown with three rows of data:

ID_UTILIZATOR	USERNAME	EMAIL	PAROLA
1	4queen	gel@gmail.com	iamqueen
2	5d4player	plic@gmail.com	londonsystem
3	3maremaestrumaremaestru	maremaestrumaremaestru@yahoo.com	GM

A status bar at the bottom indicates "All Rows Fetched: 3 in 0.001 seconds".