



# **Agentia de pariuri**

## **“Vlad BET”**

Toma Vlad-Florin 1310B  
Coordonator :  
Cătălin Mironeanu

## 1. Tema proiectului

Tema proiectului este crearea unei agentii de pariuri sportive unde pot sosi diferiti oameni pentru a-si folosi cunostintele sportive dar si norocul in scopul de a castiga cat mai multi bani posibil.

Momentan, in acest stadiu al proiectului, o persoana poate paria pe un singur sport : fotbalul. Aceasta are posibilitatea de a plasa unul sau mai multe bilete, fiecare continand unul sau mai multe pariuri. Adica, daca sosete in agentie la momentul X si doreste sa puna un bilet, are posibilitatea de a alege una sau mai multe urmari ale unui sau mai multor evenimente (meciuri) ce au ora de incepere dupa ora plasarii biletului. La momentul plasarii biletului, clientul va alege o suma pe care sa o depuna, iar castigul potential va fi calculat ca fiind acea suma depusa ori cota totala de pe bilet. Cota totala de pe bilet va fi rezultatul inmultirii tuturor cotelor pariurilor alese de pe acel bilet.

## 2. Structura proiectului

### Modelul logic si relational

“Inima” modelului logic este entitatea **Client**. Aceasta va contine detaliile cele mai importante ale persoanelor ce doresc sa se foloseasca de serviciul nostru :

- ID-ul clientului
- Numele si prenumele acestuia
- CNP-ul unic al clientului ce poate fi folosit pentru a identifica persoana atat in baza de date, cat si in viata reala

Entitatea **Detalii client** a aparut din necesitatea unei relatii de 1 la 1 in proiect. Am “imprumutat” ideea prezentata la laborator cum ca in bazele de date ale diferitelor companii, detaliile mai putin semnificative sau optionale ale unui client sunt stocate intr-o entitate diferita. Aceasta contine un numar limitat de detalii, dar se pot adauga usor pe viitor unele noi :

- ID-ul clientului caruia apartin detaliile
- Numarul de telefon al clientului ce poate fi folosit pentru a lua legatura cu el in caz de necesitate
- Adresa de E-mail ce poate fi folosita in acelasi scop de mai sus, dar si pentru a trimite diferite oferte sau reclame care sa ne promoveze compania

Entitatea **Bilet** reprezinta punctul intermediar prin care clientul doreste sa castige bani. Aceasta se afla intr-o relatie de 1 la mai multi din partea entitatii Client. Asadar, un client are posibilitatea de a plasa mai multe bilete, dar un bilet apartine unui singur client. Entitatea contine :

- Un ID ce este folosit pentru identificarea biletului

- Suma jucata ce reprezinta totalul de bani pe care clientul vrea sa depuna
- Cota totala ce este calculata in functie de pariurile de pe acel bilet
- Castigul potential ce reprezinta suma de bani ce ar putea sa fie castigata de client in cazul in care a nimerit toate pariurile de pe bilet
- Data si ora plasarii biletului

Entitatea **Pariu** reprezinta una dintre alegerile clientului de pe bilet. Aceasta va contine :

- ID-ul evenimentului pe care doreste clientul sa parieze
- ID-ul biletului caruia ii apartine pariul
- ID-ul alegerii rezultatului evenimentului ales de client

Entitatea **Tip Pariu** este folosita cu un scop exclusiv informativ pentru a tine evidenta diferitele alegeri posibile ale unui client pentru un eveniment. Structura aceesetia este :

- Un ID ce este folosit pentru a identifica tipul de pariu dorit
- Numele pariului in cauza

Entitatea **Cote Eveniment** reprezinta diferitele cote propuse de agentia in vederea efectuarii unui anumit rezultata pentru un eveniment. Aceasta contine :

- ID-ul evenimentului pentru care este valabila aceasta cota
- ID-ul tipului de pariu pentru care este valabila aceasta cota
- Cota propusa ce este un numar de maxim 2 cifre cu 2 zecimale; calculul numeric este unul usor : daca un client ar paria 100 de lei pe o cota de 1.48, atunci acesta ar primi  $100 * 1.48 = 148$  lei inapoi daca se aadevereste rezultatul

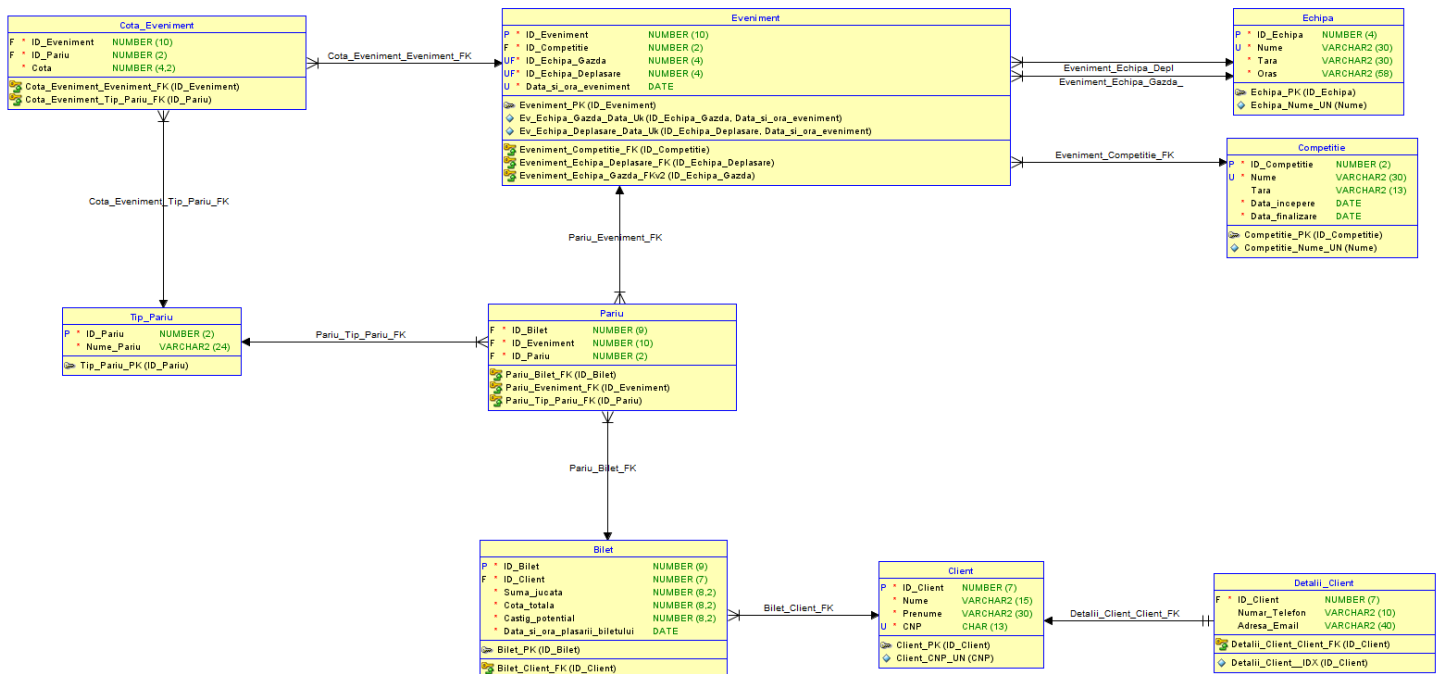
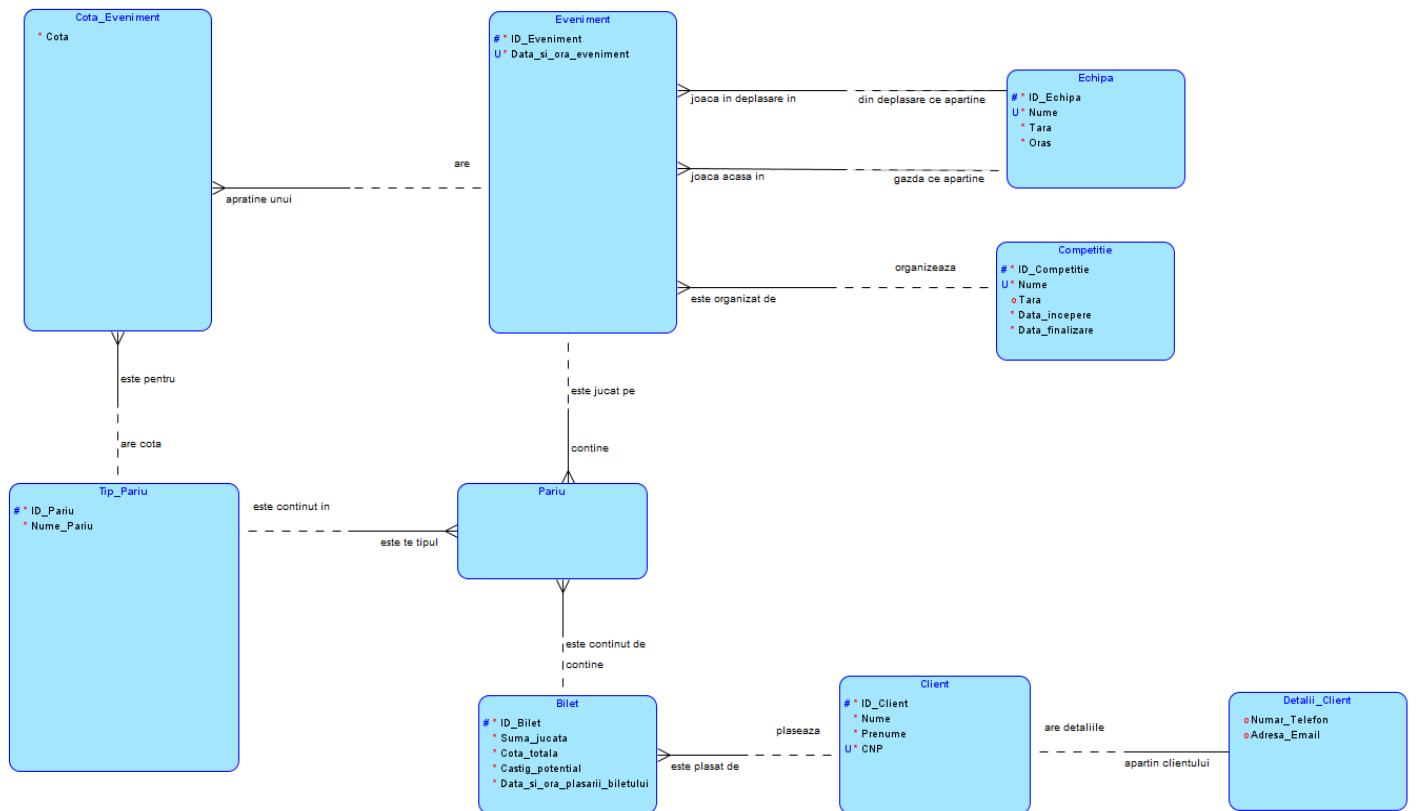
Entitatea **Eveniment** reprezinta detaliile importante ale unui meci de fobal ce urmeaza a fi disputat :

- Un ID ce este folosit pentru a identifica evenimentul dorit
- Echipele gazda si deplasare intre care se va disputa meciul
- Competitia in care se desfasoara acel meci
- Data si ora la care se va desfasura acel eveniment

Entitatile **Echipa** si **Competitie** sunt 2 entitati pur informationale si care ne ajuta sa construim un eveniment. Acestea contin :

- Un ID pentru a ne ajuta sa le identificam in program
- Numele respectivelor echipe/competitii
- Localizarea lor prin intermediul tarii/orasului

## Modelul logic



## Modelul relational

## Constrangeri folosite

### 1. Pentru detalii ale clientului :

#### 1.1. Pentru CNP :

##### 1.1.1. Constrangerea de UNIQUE

Un client are un CNP unic atribuit la nastere, asadar nu ar trebui sa avem mai multi clienti cu acelasi CNP.

##### 1.1.2. Constrangerea de CHECK

Ne ajuta sa validam CNP-ul introdus printr-un set de reguli :

- Prima cifra identifica sexul clientului
- Urmatoarele 6 cifre identifica data de naste sub urmatorul format : 'YYMMDD' (MM va avea valori de la 01 la 12, iar DD de la 01 la 31)
- Urmatoarele 2 cifre identifica judetul in care a fost nascuta persoana (pot lua valori de la 01 la 48)
- Urmatoarele 3 cifre reprezinta un numar ce diferentiaza persoanele ce au toate cifrele precedente comune
- Ultima cifra reprezinta suma de control a primelor 12 cifre din CNP

#### 1.2. Pentru numarul de telefon :

##### 1.2.1. Constrangerea de CHECK

Ne ajuta sa validam ca numarul de telefon primit contine toate cifrele si este valid pentru retelele din Romania.

#### 1.3. Pentru E-mail :

##### 1.3.1. Constrangerea de CHECK

Ne ajuta sa validam ca am introdus un E-mail cu formatul corect : 'orice@orice.orice'

### 2. Pentru tipul de pariu :

#### 2.1. Pentru numele pariului :

##### 2.1.1. Constrangerea de CHECK

Aceasta verifica ca sirul de caractere introdus sa apartina unei anumite liste de valori ce reprezinta toate posibilitatile alegerii unui pariu

### 3. Pentru eveniment :

#### 3.1. Pentru echipa gazda si data evenimentului sau pentru echipa din deplasare si data evenimentului :

##### 3.1.1. Constrangerea de CHECK

Aceasta are rolul de a impiedica o echipa sa apara in baza de date ca joaca mai mult de un meci in acelasi timp

#### 3.2. Pentru verificarea unicitatii echipei gazde si din deplasare

##### 3.2.1. Constrangerea de CHECK

Aceasta are rolul de a verifica ca echipa gazda sa fie diferita fata de echipa din deplasare

### 4. Pentru echipe/competitii :

4.1. Pentru numele acestora :

4.1.1. Constrangerea de UNIQUE

Aceasta verifica unicitatea numelui in cele 2 tabele, deoarece in viata reala nu pot exista doua echipe sau competitii cu aceleasi nume

## **Alte functii folosite**

Pentru fiecare coloana de tip "ID\_x" din tabele care nu este foreign key, a fost introdus un autoincrement ce incepe de la 1 si creste din 1 in 1. In acelasi timp, odata cu introducerea autoincrementului au fost denumite si triggerele corespunzatoare.

## **3. Normalizarea**

Normalizarea a fost folosita la nivelul entitatilor prin folosirea foreign key-urilor pentru a le descompune in entitati mai mici care stocheaza aceleasi date ca și entitatea initiala astfel incat sa fie eliminate redundanta in date. Spre exemplu : Eveniment -> Echipa, Competitie.

Prima Forma Normala este indeplinita deoarece fiecare atribut al unei relatii este atomic (nu apartine unui set sau unei liste de valori). De asemenea, fiecare inregistrare este definita astfel incat sa fie identificata in mod unic prin intermediul unei chei primare.

A doua Forma Normala este indeplinita deoarece toate cheile prezente in baza de date nu sunt compuse.

A treia Forma Normala este indeplinita deoarece toate attributele non-cheie sunt direct dependente de toate cheile candidat.

## **4. Interfata cu utilizatorul**

### **4.1. Detalii interfata**

Aplicația care se ocupă de gestionarea unei baze de date pentru o agentie de pariuri sportive. Ea este destinata exclusiv angajatilor companiei. Functionalitatile de baza ale aplicației sunt: vizualizarea si crearea biletelor pentru clienti noi/existenti si adaugarea/stergerea/modificarea anumitor detalii ale evenimentelor/echipelor/competitiilor de fotbal.

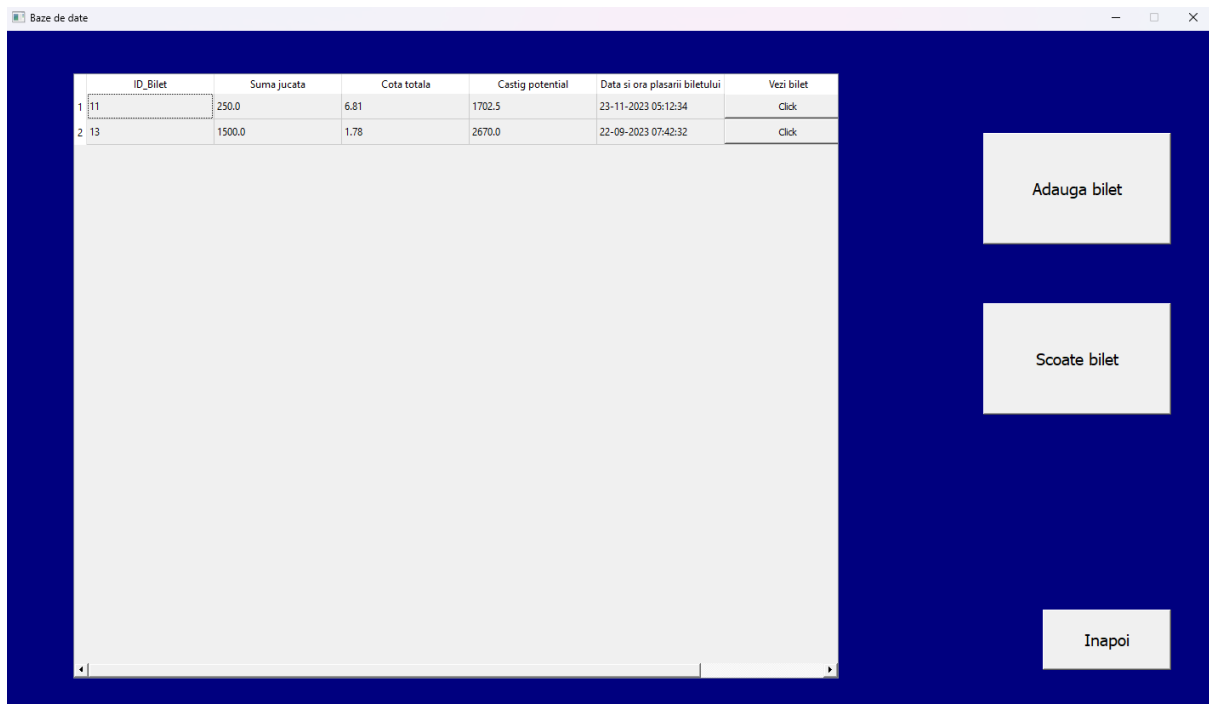
**Conectarea la baza de date** se face prin intermediul modulului cx\_Oracle, folosind instructiunea :

```
cx_Oracle.connect("username", "parola", "nume_domeniu:port/serviciu")
```

### **Tehnologii utilizate:**

- Back-end: SQL și Python
- Front-end: Qt Designer

## 4.2. Exemple de functionalitati ale aplicatiei



	ID_Bilet	Suma jucata	Cota totala	Castig potential	Data si ora plasarii biletului	Vezi bilet
1	11	250.0	6.81	1702.5	23-11-2023 05:12:34	Click
2	13	1500.0	1.78	2670.0	22-09-2023 07:42:32	Click

Adauga bilet

Scoate bilet

Inapoi

Aceasta interfata reprezinta biletele active ale clientului ales de noi fie prin interfata de client nou sau prin cea de client existent.

Functionalitati ale acestei interfete :

- Butonul **"Adauga bilet"** ne va redirectiona catre o noua interfata unde putem crea un nou bilet pentru clientul curent
- Butonul **"Scoate bilet"** ne permite stergerea unui bilet din baza de date; Pentru a folosi butonul trebuie apasat in orice coloana a biletului pe care vrem sa il stergem dupa se va apasa butonul
- Butonul **"Inapoi"** ne permite sa ne intoarcem la interfata principala pentru a selecta un client nou sau a intra in modul de administrator
- Butoanele **"Vezi bilet"** ne vor redirectiona catre o noua interfata unde ne vor fi prezentate pariurile de pe acel bilet

Aceasta interfata ne permite crearea unui nou eveniment in modul administrator.

Functionalitati ale acestei interfete :

- Butoanele **"Refresh"** de sub campurile "Competitie", "Echipa gazda", "Echipa deplasare" ne vor actualiza o lista de valori valide din baza de date din care putem alege in meniul de tip drop-down pentru a selecta informatia dorita
- Butonul **"Creare eveniment"** va insera in baza de date un nou eveniment cu datele completate de angajat pe interfata
- Butonul **"Management echipa"** ne va duce catre o interfata noua unde putem vizualiza detalii/adauga/sterge echipe in baza de date
- Butonul **"Management competitie"** ne va duce catre o interfata noua unde putem vizualiza detalii/adauga/sterge competitii in baza de date
- Butonul **"Inapoi"** ne permite sa ne intoarcem la interfata administratorului

### 4.3. Exemple din cod

```
cur.execute("SELECT c.Nume, eg.Nume, ed.Nume, e.Data_si_ora_eveniment, ce.cota, e.id_eveniment "
"from competitie c, echipa eg, echipa ed, eveniment e, cota_eveniment ce, tip_pariu tp "
"where e.id echipa_gazda = eg.id echipa and e.id echipa_deplasare = ed.id echipa and ce.id eveniment = e.id eveniment "
"and c.id competitie = e.id competitie and ce.id_pariu = tp.id_pariu ORDER BY e.id eveniment, ce.id_pariu")
```

Cu aceasta instructiune ne extragem din baza de date detaliile necesare afisarii evenimentelor pentru interfata de administrator.



```
if(len(evv) > 0):
    for i in range(0, len(evv)):
        cur.execute("DELETE FROM pariu WHERE ID_Eveniment = {}".format(int(evv[i][0])))
        cur.execute("DELETE FROM cota_eveniment WHERE ID_Eveniment = {}".format(int(evv[i][0])))
        cur.execute("DELETE FROM eveniment WHERE ID_Eveniment = {}".format(int(evv[i][0])))
    cur.execute("DELETE FROM Echipa WHERE ID_Echipa = {}".format(id))
```

Aceste comenzi ne permit stergerea unei echipe din baza de date. Pentru a putea sterge o echipa, mai intai trebuie sterse toate evenimentele ce contin acea echipa. Pentru a putea sterge un eveniment, mai intai trebuie sterse toate cotele si pariurile ce contin acel eveniment. Prin folosirea unui bloc try-except, in cazul in care toate instructiunile au fost rulate cu succes se va efectua un commit, altfel utilizatorul va primi un mesaj de eroare si are optiunea de a incerca din nou.