

**UNIVERSITATEA TEHNICA "GHEORGHE ASACHI" IAȘI**  
**FACULTATEA AUTOMATICĂ ȘI CALCULATOARE**  
**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI**  
**DISCIPLINA BAZE DE DATE**



**TRANSFERUL JUCĂTORILOR DIN NBA**

**Coordonator,**

**Avram Sorin**

**Studenti,**

**Chiteală Tudor Matei**

**Simioniuc Ruxandra**

**Iași, 2020**

## **Titlu Proiect : Transferul Jucatorilor din NBA**

Realizarea unei baze de date și a relațiilor necesare între tabele pentru a modela transferurile și condițiile transferurilor din NBA.

### **Descrierea cerințelor și modul de organizare al proiectului**

Proiectul propune gestionarea jucatorilor din NBA (National Basketball Association), gestionare ce poate fi îngreunată de volumul mare de jucători existent în NBA (National Basketball Association), jucătorii cu istoric diferit și echipe care concurează asupra transferului unui jucător.

Transferurile se realizează prin achitarea prețului de transfer, care este diferit pentru fiecare jucător. Transferul se poate realiza doar dacă jucătorul este disponibil și echipa are un buget suficient pentru a plăti transferul jucătorului. Pentru a putea stabili dacă jucătorul se potrivește cu nevoile echipei, se pot vizualiza, prin accesarea tabelei "Istorie", trăsăturile jucătorului, ce au fost deprinse în cariera sa la diferite echipe.

Informațiile de care avem nevoie sunt legate de:

**Echipa:** este de interes ce echipa vrea să realizeze o tranzacție, pentru a putea evalua dacă are un buget suficient

**Jucători:** datele referitoare la jucătorii sunt împărțite în tabelele "Jucători", "Detalii\_jucatori" și "Istorie". Tabela "Detalii\_jucatori" conține informații generale de identificare (CNP, țara, data nașterii), în timp ce "Jucatori" și "Istorie" conțin detalii mai specifice industriei basketball-ului (salar, poziții, experiența).

**Istoria jucătorului:** aceste informații sunt de interes pentru echipe, pentru a efectua un transfer după nevoile acestora. Echipele pot accesa date despre cariera și pozițiile jucătorului, dobândite în urma meciurilor jucate.

**Pozițiile pe care a jucat:** în urma fiecărui sezon, jucătorul a fost plasat într-o anumită poziție. Acesta poate juca pe mai multe posturi, făcându-l mai dinamic în teren.

### **Descrierea funcțională a aplicației**

Principalele funcții care se pot întâlni în gestionarea echipei sunt:

- Evidența istoricului unui jucător
- Evidența jucătorilor
- Vânzarea jucătorilor

**Tabelele** din aceasta aplicație sunt:

- Detalii\_Jucatori
- Echipe
- Istorie
- Jucatori
- Pozitii

În proiectarea acestei baze de date s-au identificat tipurile de **relații 1:1** și **1:n**.

Între **tabelele jucători** și **detalii\_jucatori** se întâlnește o relație de tip **1:1** deoarece fiecare jucător are doar o tabelă de detalii și detaliile pot aparține doar unui singur jucător.

Între **tabelele jucători** și **echipe** este o **relație 1:n**, pentru că un jucător poate aparține doar unei echipe și o echipă poate avea mai mulți jucători. Invers nu este posibil, întrucât un jucător nu poate fi la mai multe echipe în același timp.

De asemenea între **tabela jucatori** și **istorie** este o **relație 1:n**, **tabela istorie** poate deține informații despre mai mulți jucători.

**Tabela istorie** are tot o **relație 1:n** cu **tabela poziții**, deoarece **tabela istorie** poate avea informații despre diferite poziții pe care jucătorul le-a dobândit pe parcursul carierei.

## Constrângeri

În **tabela detalii\_jucatori** am aplicat constrângeri de tip **check** pe atributele **cnp**, **data nașterii**, **email**:

- CNP-ul trebuie să fie de 13 cifre
- Email-ul conține '@' și caracterul '.'
- Data nașterii și țară sunt atribute diferite de NULL

În **tabela echipe** avem constrângeri pentru:

- Buget, acesta trebuie să fie mai mare sau egal decât 0.

În tabela **istorie** putem identifica următoarele constrângeri:

- Numărul de meciuri a unui jucător într-un sezon (sau mai multe) nu poate fi mai mic ca zero.
- Data\_inceput și data\_final trebuie să fie diferite de NULL
- id\_jucator, id\_echipă, id\_calitate, id\_pozitie și id\_istorie trebuie să fie diferite de 0
- Tabela istorie are ca primary key ( id\_jucator, data\_inceput, data\_final) deoarece un jucător nu poate juca în același timp la 2 echipe.

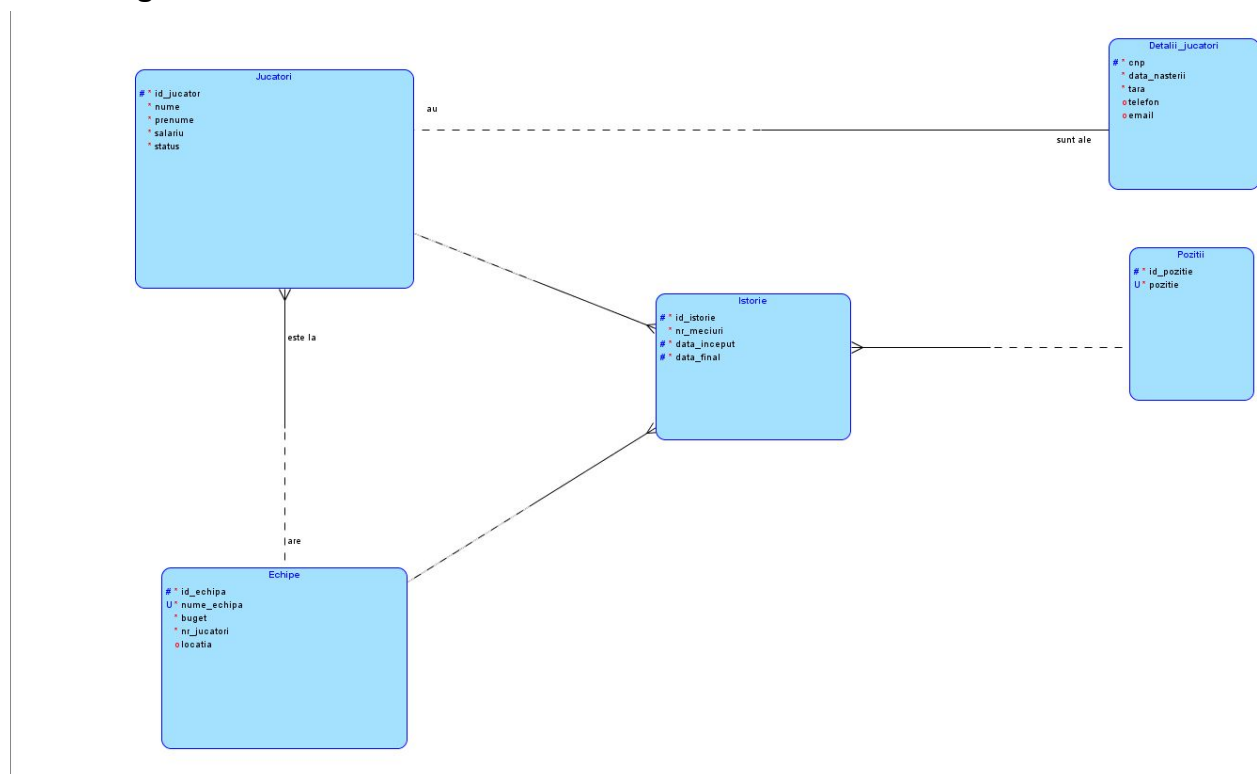
În tabela **jucatori** avem următoarele constrângeri:

- Salariul jucătorului este mai mare ca zero.
- Numele și prenumele jucătorului trebuie să conțină cel puțin un caracter

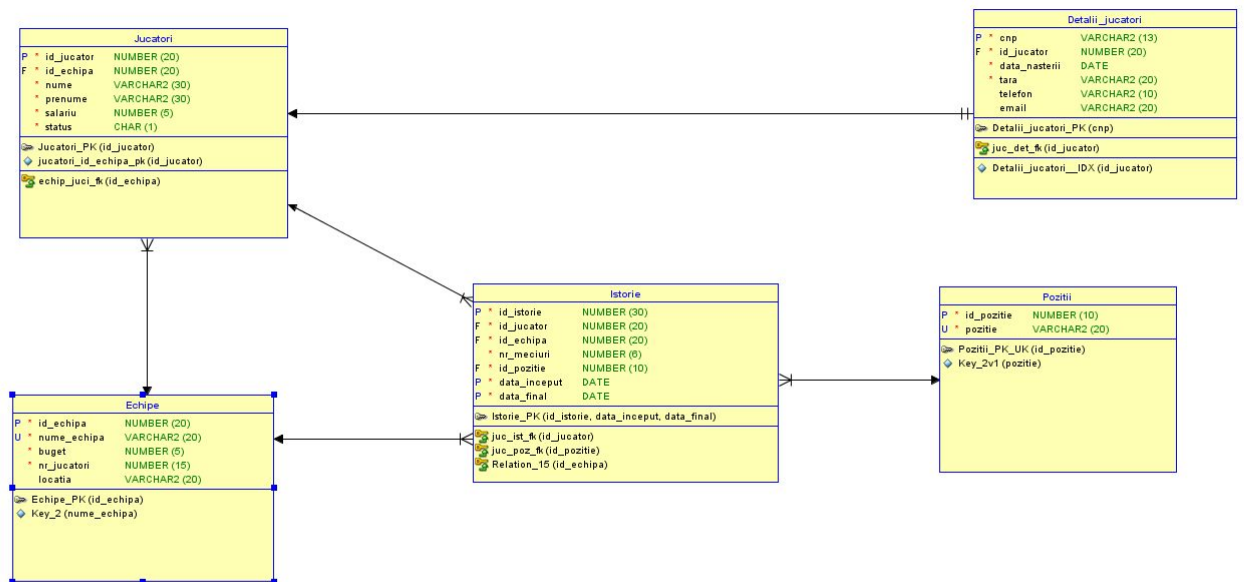
În tabela **pozitii** avem ca și constrângeri :

- Atributul poziție poate lua doar următoarele valori: aruncator, conducator joc, extrema, pivot, pivot mic
- id\_pozitie trebuie sa fie diferit de NULL

## Model logic



## Model relational



## Interfața cu utilizatorul

Pentru a putea ușura manipularea datelor din tabele, a fost construită o interfață grafică. Interfața cu utilizatorul a fost creată în **python 3**, cu ajutorul bibliotecii **tkinter**.

Conexiunea la baza de date și efectuarea schimbărilor din aceasta s-a realizat cu biblioteca **cx\_Oracle**. Tot pentru realizarea conexiunii cu baza de date a fost necesara descarcarea *Oracle Instant Client*.

Interfața este simplă, fiind posibilă interacțiunea cu fiecare tabelă din baza de date. În fereastra principală există câte un *tab* aferent fiecărei tabele din baza de date și un *tab* pentru realizarea tranzacțiilor.

The image shows a window titled 'Jucatori' with a form for adding a player. The form contains the following fields and values:

Nume	Bryant
Prenume	Kobe
Salariu	7
Status	1
ID Echipa	2

At the bottom of the form is a button labeled 'Adauga Jucator'.

## 1. Jucatori

- **Adaugare Jucator:** inserează un jucător nou în tabela
- **Modificare Jucator:** modifica datele unui jucător existent în tabela în funcție de ID-ul jucătorului
- **Stergere Jucator:** sterge un jucător din tabela în funcție de ID-ul jucătorului
- **Afisare Jucatori:** afiseaza continutul tabelii Jucatori

## 2. Echipe

- **Adaugare Echipa:** inserează o nouă echipa, cu bugetul automat 100 și numărul de jucători egal cu 0
- **Modificare Echipa:** pe baza ID-ului echipei se poate modifica numele și locația echipei
- **Stergere Echipa:** șterge o echipa în funcție de ID-ul primit
- **Afisare Echipe:** afișează conținutul tabelii Echipe

## 3. Detalii Jucatori

- **Adauga Detalii Jucatori:** adauga CNP, tara, data nașterii, email, telefon pentru un jucător identificat prin ID
- **Modifica Detalii Jucator:** se pot modifica detaliile, tot pe baza ID-ului
- **Sterge Detalii Jucatori:** se pot șterge toate detaliile unui jucator
- **VizualizeazaDetalii Jucatori:** afișează conținutul tabelii Detalii\_Jucatori

## 4. Istorie

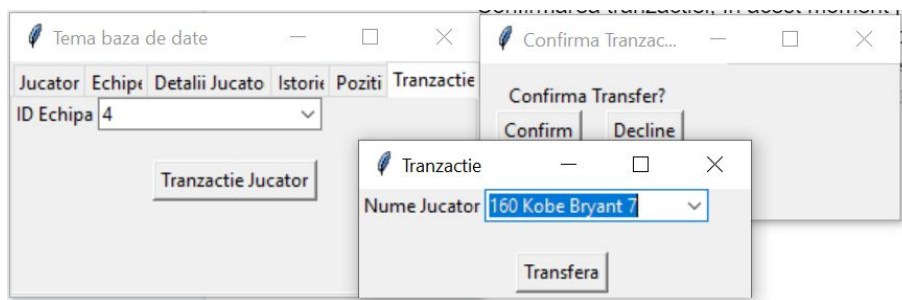
- **Adaugă Istoric:** în funcție de ID-ul jucătorului, se pot adăuga date despre istoria acestuia (unde a jucat, cate meciuri, pe ce poziție, în ce perioada)
- **Sterge Istoric Jucător:** șterge istoricul pe o perioada data de user
- **Modifica Istoric Jucatori:** se poate modifica istoricul unui jucător pe o anumită perioada setata de user
- **Vizualizeaza istoricul jucatorilor:** afiseaza continutul tabele Istorie

## 5. Poziții

- **Vizualizeaza Poziții:** afiseaza continutul tabelei Poziții; este o tabela ce nu poate fi modificata

## 6. Tranzacții

- Acest tab oferă posibilitatea transferării unui jucător dintr-o echipa (sau liber) în alta.
- Tranzacția se realizeaza din perspectiva echipei care vrea sa realizeze transferul
- Transferul nu se poate realiza decat dacă jucătorul are statusul 0 și dacă echipa ce vrea sa realizeze transferul are buget suficient (mai mare decat salariul jucătorului ce se vrea transferat)
- Înainte ca tranzactia sa fie finalizata, apare un pop-up care cere Confirmarea tranzactiei; în acest moment jucătorul este “rezervat”, iar dacă alta echipa încearcă să îl transfere apare o coliziune;
- Dacă se confirma tranzacția, jucătorul este transferat cu succes; daca de anuleaza, alta echipa poate transfera jucătorul;



După fiecare operație (adăugare, modificare, ștergere, tranzacție) apare un mesaj ce confirmă completarea acțiunii. În cazul în care nu sunt respectate unele cerințe (buget prea mic, salariu prea mare pentru un jucător ce se adaugă într-o echipa), apare un mesaj ce specifica problema.

Fiecare operație cu jucătorii (adaugare, transferare) impactează și tabela Echipa, anume coloana nr jucători (valoarea scade/crește în funcție de operație). Dacă o echipa este ștearsă, toți jucătorii săi sunt trecuți automat ca fiind fără echipa și de vânzare (status = 0).

Exemplu de cod: Update al echipei vechi a jucătorului la transferul acestuia; dacă face parte din echipa 0 (fără echipa)

```
# update echipa veche
c.execute("SELECT id echipa FROM Jucatori WHERE id_jucator = " + id_player)
id0 = c.fetchall()
if id0[0][0] != 0:
    sequence = """ UPDATE Echipa
                    SET nr_jucatori = nr_jucatori - 1,
                    buget = buget + (SELECT salariu FROM Jucatori WHERE id_jucator = """ + id_player + ") + """
                    WHERE id echipa = (SELECT id echipa
                    from Jucatori where id_jucator = """ + id_player + ")
    c.execute(sequence)
else:
    sequence = """ UPDATE Echipa
                    SET nr_jucatori = nr_jucatori - 1
                    WHERE id echipa = 0"""
    c.execute(sequence)
# conn.commit()
```

### Contribuțiile aduse la dezvoltarea aplicației:

Tema a fost dezvoltată pe baza proiectului realizat Matei Chiteala la disciplina BD-P.

Matei Chiteala:

- Stabilirea tematicii proiectului
- Stabilirea structurii proiectului, a tabelor necesare și a relațiilor dintre acestea
- Realizarea schemelor logice și relaționale
- Realizarea interfeței cu utilizatorul
- Modificarea și adaptarea proiectului inițial pentru a respecta cerințele temei

Ruxandra Simioniuc:

- Stabilirea tabelor necesare și a relațiilor dintre acestea
- Realizarea documentației
- Dezvoltarea interfeței cu utilizatorul
- Modificarea și adaptarea proiectului inițial pentru a respecta cerințele temei