

**UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI**  
**FACULTATEA AUTOMATICĂ ȘI CALCULATOARE**  
**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI**  
**DISCIPLINA BAZE DE DATE: TEMA DE CASA**

**Magazin online de jocuri**

**Coordonator,**  
Ing. Sorin Avram

**Student,**  
Amarandei Narcis-Constantin

Iași, 2021

## Titlu proiect: Magazin de jocuri online

Analiza, proiectarea și implementarea unei baze de date care să modeleze activitatea unui magazin online de jocuri în ceea ce privește evidența angajaților, clienților, dar și a jocurilor.

## Descrierea cerintelor si modul de organizare al proiectului

Coordonarea unui magazin online constituie o adevărată provocare deoarece necesită o organizare mai promptă decât cea a unui magazin fizic, unde toate informațiile pot fi găsite în diferite registre și documente.

Interacțiunea dintre angajați și clienți se realizează în mediul online, astfel este necesară crearea unui cont unic pe baza adresei de email și a unor informații cu caracter personal, precum numele complet, regiunea, etc. Prin intermediul site-ului clienții au posibilitatea de a achiziționa o serie de jocuri sau de a descărca jocurile free puse la dispoziție de către admini. Plata pentru jocuri se realizează cu cardul.

Angajații au rolul de a monitoriza tranzacțiile, de a asigura funcționalitatea site-ului și de a răspunde la eventualele comentarii sau nemulțumiri în ceea ce privește jocurile ce le sunt asignate.

Jocurile au o serie de specificații, unde pot primi un rating, se pot adăuga comentarii și de asemenea se pot vedea numărul de descărcări a jocului pentru a facilita alegerea unui joc pentru noi clienți.

Astfel, se disting următoarele aspecte:

**-Detalii\_clienti:** conține cnp-ul, numele, prenumele, emailul, numărul de telefon, adresa, țara și cardul clienților pentru a ne asigura că nu se introduce o identitate falsă și pentru a evita fraudă.

**-Clienți:** conține câte un id\_client unic pentru fiecare client, un username care, de asemenea, este unic și regiunea din care face parte pentru a fluidiza datele și pentru a ne asigura că jucătorul instalează o versiune compatibilă cu mediul din care provine.

**-Jocuri:** conține un id unic, nume, pret, data cand a aparut jocul, ce fel de joc este(genul) si daca este single sau multiplayer, toate aceste informatii sunt necesare utilizatorilor care doresc sa cumpere jocuri.

**-Specificatii:** contine informatii despre jocuri precum numele jocului, un rating dat de utilizatori bazat pe un nr de stele (maxim 5 stele, incepand de la 1 care semnifica ca acel joc nu este tocmai apreciat in randul altor jucatori, pana la 5 stele care semnifica ca jocul este foarte bun), comentarii adaugate tot de utilizatori pentru a relata experienta lor pentrecuta de-a lungul acelui joc, pentru a semnala eventualele probleme sau a adresa diverse intrebari.

**-Angajați:** conține o serie de informații personale în ceea ce privește angajații precum un id unic, username care este necesar pentru accesarea magazinului online, nume, prenume, adresa, email, telefon si cnp.

### Descrierea funcțională a aplicației

Principalele funcții pe care le putem utiliza sunt:

- Achizitionare de jocuri.
- Evidența Clientilor.
- Evidența Angajaților.
- Evidența Jocurilor și distribuirea catre clienti.

### Descrierea detaliata a entitatilor si a relatiilor dintre tabele

Tabelele din aceasta aplicatie sunt:

- Clienti
- Detalii\_clienti
- Jocuri
- Specificatii
- Angajati
- Relational\_1

### Descrierea a relatiilor dintre tabele

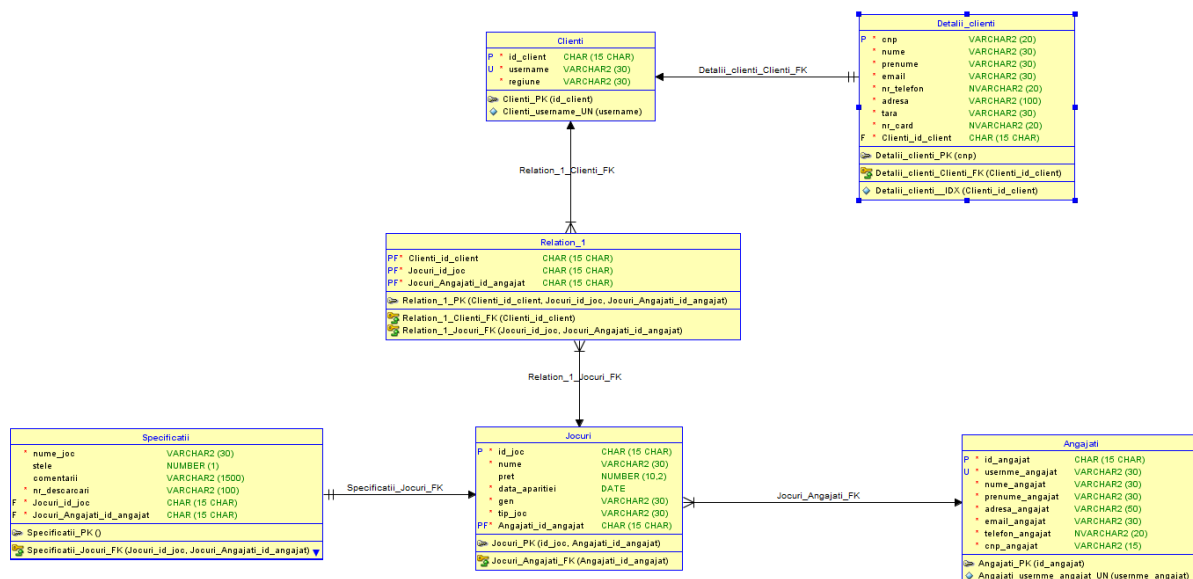
In proiectarea acestei baze de date s-au identificat tipurile de relatii n:1, 1:1 și m:n

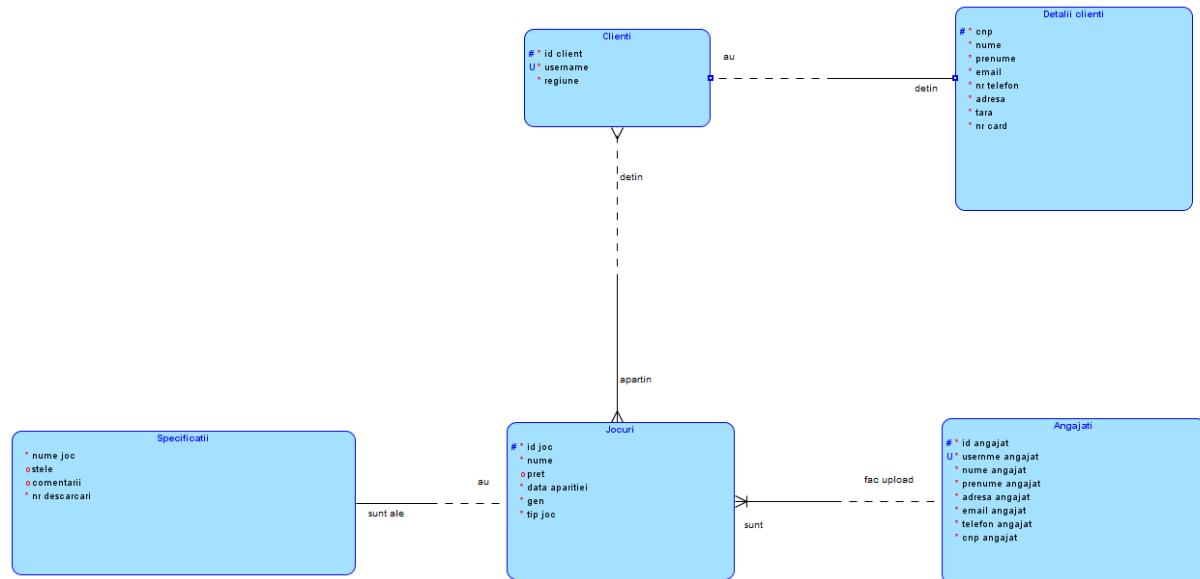
Între tabela Clienti și tabela Detealii\_clienti avem o relație de tip **1:1** deoarece fiecare client are un singur set de date cu informatii personale si fiecare set de date corespunde unui singur client.

Între tabela Clienti și tabela Jocuri avem o relație de tip **m:n** deoarece mai multe Jocuri pot fi asignate aceluiasi client, iar un joc poate fi detinut de mai multi clienti. Această legătură se realizează prin intermediul unei alte tabele aditionale, numită Relation\_1 ce transformă legătura many-to-many în două legături de tip n:1. Relation\_1 conține id\_client din tabela Clienti id\_joc din tabela Jocuri și id\_angajat din tabela Jocuri.

Între Jocuri și Angajati avem o relație de tip **n:1** deoarece un joc dupa ce a fost adaugat in magazin de catre un angajat nu mai poate fi adaugat din nou, dar un angajat poate adauga mai multe jocuri in magazin.

Între tabela Jocuri și Specificatii avem o relație de tip **1:1** deoarece fiecare joc are un set de specificatii proprii la fel cum si specificatiile sunt atribuite unui singur joc.





## Tehnologii folosite

Din punct de vedere a tehnologiilor folosite am lucrat în:

-DataModeler pentru a obține diagrama logică și cea funcțională.

-SQL Developer pentru a mă putea conecta la baza de date a facultății, am implementat un script de creare a bazei de date mentionata anterior si un script de inserare, ce are scopul de a popula aceste tabele.

-În ceea ce privește interfața am utilizat programul Eclipse pentru a o realiza în Java, ojdbc driver ce facilitea conectarea la baza de date, iar din punct de vedere al funcționalității am implementat strict posibilitatea de a vedea din intermediul interfeței tabelele și conținutul lor.

## Constrângeri utilizate:

### 1. DE TIP NOT NULL

Toate atributele sunt de tip NOT NULL cu exceptia la pret,stele si comentarii.

### 2. DE TIP UNIQUE

Este necesar ca fiecare utilizator cat si angajat sa detina un username unic.

Toate primay-key-urile sunt unice predefinite.

### 3. DE TIP CHECK

Deoarece Angajati sunt din Romania este necesar ca CNP-ul să aibă 14 cifre

Regiunea trebuie sa fie neaparat din aceasta lista: America de N, Europa de N si E, Europa de V, Korea, Rusia.

La Stele se pot adauga intre 1 si 5 stele

.Numărul de telefon al Angajaților să înceapă cu 07 și să continue cu alte 8 cifre (trebuie să fie neapărat un nr. de mobil și să aibă formatul corespunzător țării noastre)

Emailul să aibă caracterul @.

Conectarea la baza de date s-a realizat folosind driverul jdbc descarcat de pe site-ul Oracle.

```
public JavaOracleTest() {  
    String dbURL="jdbc:oracle:thin:@ bd-dc.cs.tuiasi.ro:1539:orcl";  
    String username="bd110";  
    String password="bd110";  
    try {  
        connection = DriverManager.getConnection(dbURL,username, password);  
        System.out.println("Conectarea s-a realizat cu succes");  
    }  
    catch (SQLException e)  
    {  
        System.out.println("Eroare, nu s-a reusit conectarea");  
        e.printStackTrace();  
    }  
}
```

Magazin online de jocuri

Clienti	Detalii clienti	Relation_1	Jocuri	Angajati	Specificatii
---------	-----------------	------------	--------	----------	--------------

ID_CLIENT	USERNAME	REGIUNE
1	Sabuc	Korea
2	DarkLorde	America de N
3	KingOfGames	Europa de N și E
4	Faker	Korea
5	Kalashnikov	Rusia
6	Hitler	Europa de V
7	DonDinero	Rusia
8	Costel96	Europa de N și E

Interfat a fost realizata astfel:

```
frmMagazinOnlineDe = new JFrame();
frmMagazinOnlineDe.getContentPane().setEnabled(false);
frmMagazinOnlineDe.getContentPane().setFocusable(false);
frmMagazinOnlineDe.getContentPane().setFocusTraversalKeysEnabled(false);
frmMagazinOnlineDe.setTitle("Magazin online de jocuri");
frmMagazinOnlineDe.setBounds(100, 100, 968, 695);
frmMagazinOnlineDe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frmMagazinOnlineDe.getContentPane().setLayout(null);
JButton btnNewButton_4 = new JButton("Clienti");
btnNewButton_4.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Clienti";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
}
```

```

e1.printStackTrace();
}

}
});

btnNewButton_4.setBackground(Color.LIGHT_GRAY);
btnNewButton_4.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_4.setBounds(29, 36, 110, 35);
btnNewButton_4.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
}
});

table = new JTable();
table.setBounds(170, 389, 0, 0);
frmMagazinOnlineDe.getContentPane().add(table);
frmMagazinOnlineDe.getContentPane().add(btnNewButton_4);
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(29, 90, 860, 540);
frmMagazinOnlineDe.getContentPane().add(scrollPane);
table_1 = new JTable();
scrollPane.setViewportView(table_1);
JButton btnNewButton = new JButton("Detalii clienti");
btnNewButton.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Detalii_clienti";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
e1.printStackTrace();
}
}
});
btnNewButton.setBackground(Color.LIGHT_GRAY);
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
}
});
btnNewButton.setBounds(175, 36, 110, 35);
frmMagazinOnlineDe.getContentPane().add(btnNewButton);

```



```

JButton btnNewButton_1 = new JButton("Relation_1");
btnNewButton_1.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Relation_1";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
e1.printStackTrace();
}
}
});
btnNewButton_1.setBackground(Color.LIGHT_GRAY);
btnNewButton_1.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_1.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
}
});
btnNewButton_1.setBounds(329, 36, 110, 35);
frmMagazinOnlineDe.getContentPane().add(btnNewButton_1);
JButton btnNewButton_2 = new JButton("Jocuri");
btnNewButton_2.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Jocuri";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
e1.printStackTrace();
}
}
});
btnNewButton_2.setBackground(Color.LIGHT_GRAY);
btnNewButton_2.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_2.setBounds(478, 36, 110, 35);
frmMagazinOnlineDe.getContentPane().add(btnNewButton_2);
JButton btnNewButton_3 = new JButton("Angajati");

```

```

btnNewButton_3.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Angajati";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
e1.printStackTrace();
}
}
});
btnNewButton_3.setBackground(Color.LIGHT_GRAY);
btnNewButton_3.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_3.setBounds(633, 36, 110, 35);
frmMagazinOnlineDe.getContentPane().add(btnNewButton_3);
JButton btnNewButton_5 = new JButton("Specificatii");
btnNewButton_5.addMouseListener(new MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
try {
String query = " select * from Specificatii";
PreparedStatement pst = connection.prepareStatement(query);
ResultSet rs = pst.executeQuery();
table_1.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e1)
{
System.out.println("Eroare, nu s-a reusit conectarea");
e1.printStackTrace();
}
}
});
btnNewButton_5.setBackground(Color.LIGHT_GRAY);
btnNewButton_5.setFont(new Font("Times New Roman", Font.BOLD, 11));
btnNewButton_5.setBounds(779, 36, 110, 35);
frmMagazinOnlineDe.getContentPane().add(btnNewButton_5);
frmMagazinOnlineDe.setVisible(true);

}

```