

Proiect Baze de Date

Comert electronic/Gestiunea produselor pentru laptopuri

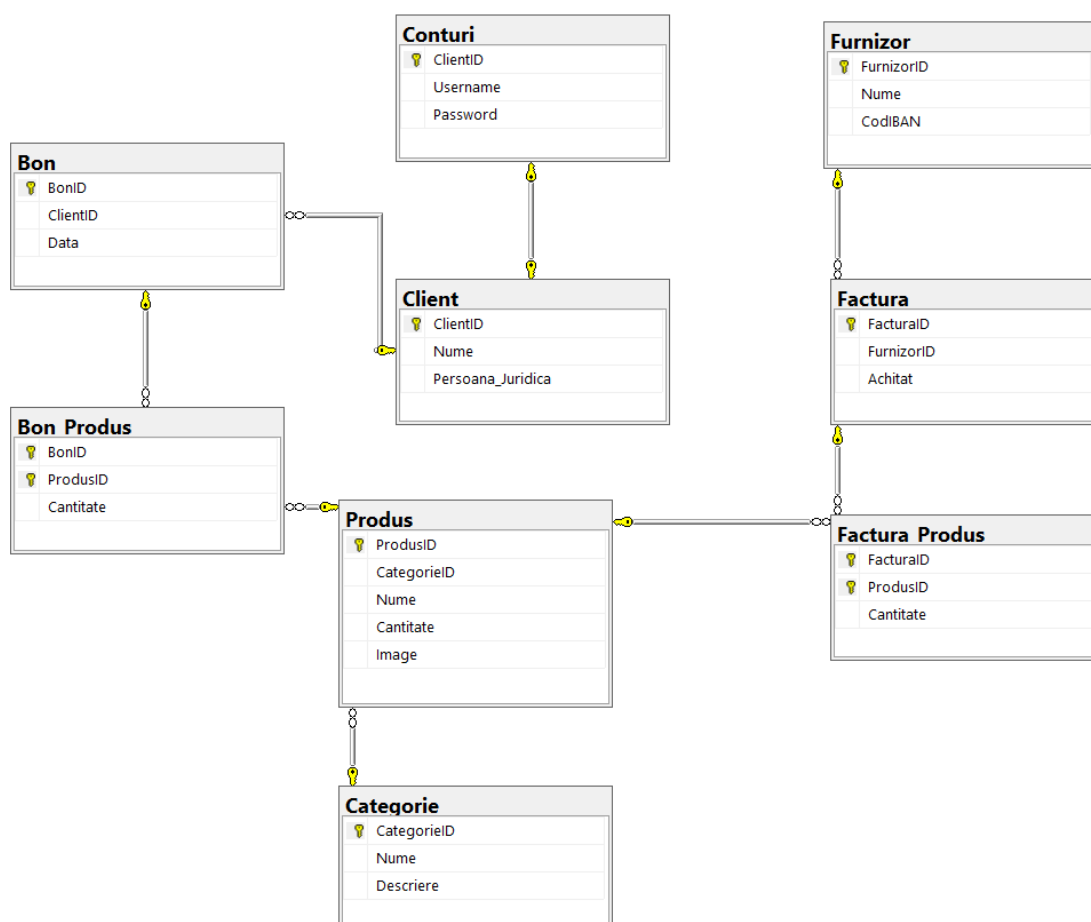
Blejan Alexandru

Descrierea proiectului

Acest proiect leaga functionalitatea unei aplicatii web cu o baza de date, pentru a gestiona clientii, produsele si furnizorii dintr-un magazin de electronice.

Baza de date contine 9 tabele:

- Conturi; pentru retinerea datelor de inregistrare pe aplicatie
- Client; ce contine datele personale ale clientilor
- Bon; reprezinta cosul de cumparaturi al clientului
- Bon_Produs; cosul activ, ce retine produsele aflate in Bon, fiind o relatie N-N
- Produs; reprezinta produsele de pe stock, ce sunt afisate pe aplicatie
- Categorie; pentru sortarea produselor dupa tipul lor
- Factura_Produs; factura activa , ce retine produsele din Facturi, relatie N-N
- Factura; pentru facturarea produselor cumparate de la furnizori
- Furnizor; ce contine datele furnizorilor de electornice



Legaturile dintre tabele:

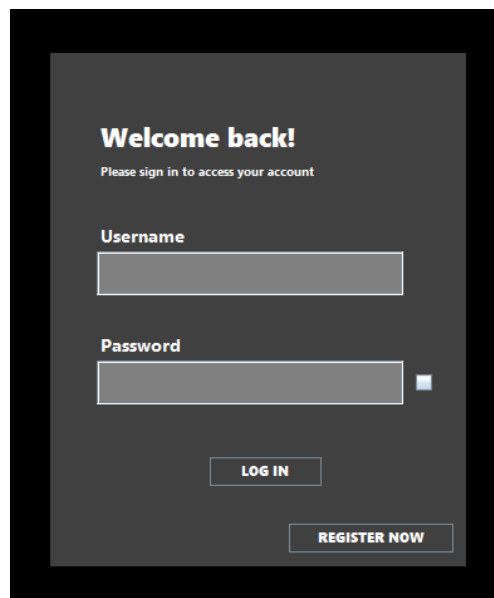
- 1-N : Conturi-Client; Client-Bon; Produs-Categorie; Furnizor-Factura
- N-N : Bon-Produs; Factura-Produs

Constrangerile de integritate impuse:

- Nici un camp nu poate fi null
- In Client, Persoana_Juridica poate fi completat doar cu „Da” sau „Nu”
- In Bon, Data trebuie sa aiba format de data, de exemplu: 02/01/2024
- In Produs, Imaginea trebuie stocata binar
- In Factura, campul Achitat poate fi completat doar cu „Da” sau „Nu”

Functionalitatea aplicatiei

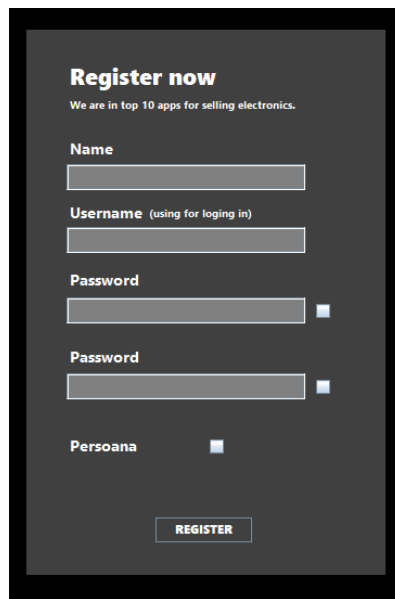
Cand se deschide aplicatia, vom avea o pagina de Login in care vom completa username-ul si parola personala ce este cautata in Tabelul de Conturi.



The image shows a login interface with a dark background. At the top, it says "Welcome back!" followed by "Please sign in to access your account". Below this are two input fields: "Username" and "Password". The "Password" field has a small eye icon to its right. At the bottom, there are two buttons: "LOG IN" and "REGISTER NOW".

```
// Establish the connection
try (Connection connection = DriverManager.getConnection(jdbcUrl)) {
    String sql = "SELECT * FROM Conturi";
    try (PreparedStatement preparedStatement = connection.prepareStatement(sql);
        ResultSet resultSet = preparedStatement.executeQuery()) {
        while (resultSet.next()) {
            String usernameColumn = resultSet.getString("Username");
            String passwordColumn = resultSet.getString("Password");
            String clientIDColumn = resultSet.getString("ClientID");
            if(username.equals(usernameColumn.trim()) && password.equals(passwordColumn.trim())) {
                ok = true;
                CurrentUserID = clientIDColumn.trim();
            }
            System.out.println("" + usernameColumn + passwordColumn);
        }
    }
}
```

In cazul in care, utilizatorul nu are cont, pagina de register va insera datele in tabelul Conturi si Client.



Register now
We are in top 10 apps for selling electronics.

Name

Username (using for logging in)

Password

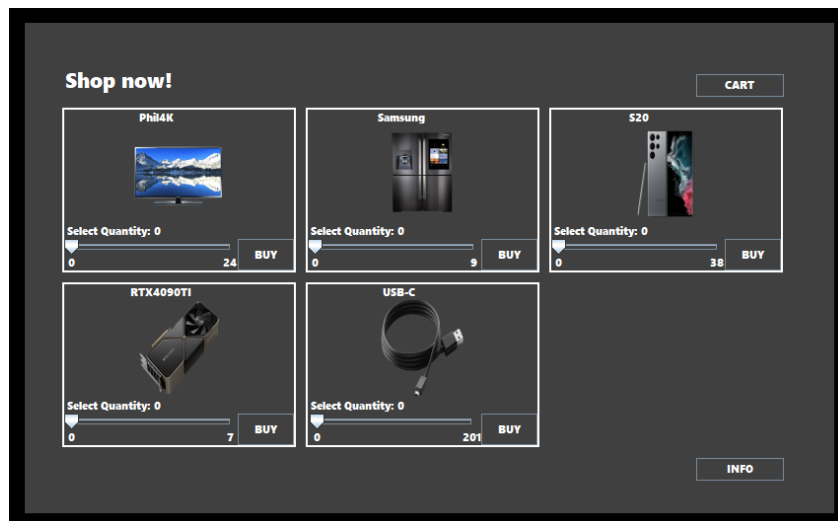
Password

Persoana ☐

REGISTER






```
String AddClient = "INSERT INTO Client(ClientID, Nume, Persoana_Juridica) Values (?, ?, ?)";  
try (PreparedStatement statement = connection.prepareStatement(AddClient)) {  
    // Set parameter values  
    statement.setInt(1, number);  
    statement.setString(2, name);  
    if(juridic)  
        statement.setString(3, "Da");  
    else  
        statement.setString(3, "Nu");  
  
    // Execute the INSERT statement  
    statement.executeUpdate();  
}  
String AddCont = "INSERT INTO Conturi(ClientID, Username, Password) Values (?, ?, ?)";  
try (PreparedStatement statement = connection.prepareStatement(AddCont)) {  
    // Set parameter values  
    statement.setInt(1, number);  
    statement.setString(2, username);  
    statement.setString(3, password);  
  
    // Execute the INSERT statement  
    statement.executeUpdate();  
}
```

Din pagina de cumparaturi se pot selecta produse folosind slider-ul si apasand pe butonul BUY, produsul se va adauga in Bon, daca exista Bon pentru Client, altfel se va crea un Bon nou. Iar daca produsul este deja existent in Bon, doar se va actualiza cantitatea.



Shop now!

CART

Phil4K  Select Quantity: 0 0 24 BUY	Samsung  Select Quantity: 0 0 9 BUY	S20  Select Quantity: 0 0 38 BUY
RTX4090TI  Select Quantity: 0 0 7 BUY	USB-C  Select Quantity: 0 0 201 BUY	

INFO

```

(jdbcUrl);
prepareStatement("UPDATE Bon_Produs SET Cantitate = ? WHERE BonID = ? AND ProdusID = ?"))
execute();

```

```

;
reStatement("INSERT INTO Bon_Produs(BonID, ProdusID, Cantitate) Values (?, ?, ?)"))

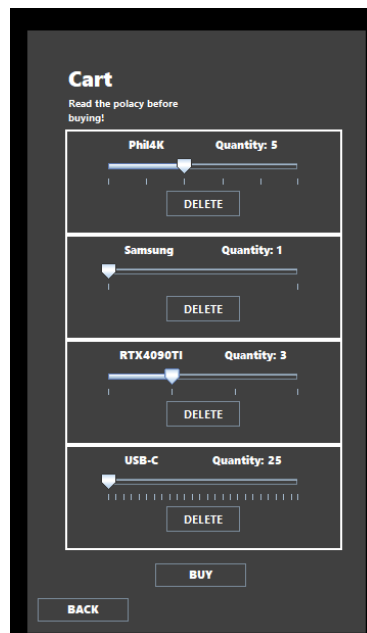
```

```

atement("INSERT INTO Bon(BonID, ClientID, Data) Values (?, ?, ?)"))

```

In cos vor fi afisate toate produsele din Bon. Se poate modifica cantitatea de la fiecare produs. Daca cantitatea stearsa este maxima, se va sterge produsul din Bon, altfel se va actualiza doar cantitatea. Dupa cumpararea produselor, Bon-ul se va sterge, iar produsele de pe stock isi vor actualiza cantitatea sau se vor sterge.



```

);
prepareStatement("Delete From Produs Where ProdusID = ?"))

```

```

Statement("UPDATE Produs SET Cantitate = ? WHERE ProdusID = ?"))

```

```

atement("Delete From Bon_Produs Where BonID = ? Delete From Bon Where BonID = ?"))

```

```

Statement("Delete From Bon_Produs Where BonID = ? AND ProdusID = ?"))

```

```

atement("UPDATE Bon_Produs SET Cantitate = ? WHERE BonID = ? AND ProdusID = ?"))

```

Codul contine:

- 11 select-uri
- 6 insert-uri
- 3 delete-uri
- 3 update-uri

Interogari simple si complexe

Info
You can receive the following informations:

Display the products that are in category:	<input type="text"/>	<input type="button" value="ACTION"/>
The clients that have in cart, all products from the category:	<input type="text"/>	<input type="button" value="ACTION"/>
Number of products bought from category:	<input type="text"/>	<input type="button" value="ACTION"/>
The clients with the most quantity of products from category:	<input type="text"/>	<input type="button" value="ACTION"/>
The most products bought by shop from category:	<input type="text"/>	<input type="button" value="ACTION"/>
The Clients with products from more than 3 categories		<input type="button" value="ACTION"/>
The Clients with products in cart older than 30days		<input type="button" value="ACTION"/>

- Produsele din categoria X

```
jdbcUrl = "jdbc:sqlserver://DESKTOP-LCDDKJ8;Database=Altex;IntegratedSecurity=true;sslProtocol=TLSv1.2";  
sql_1 = "SELECT P.Nume, C.Nume FROM Produs P join Categorie C on P.CategorieID = C.CategorieID WHERE C.CategorieID = " + textField.getText().trim();  
connection connection = DriverManager.getConnection(jdbcUrl);
```

- Clientii care au in cos toate produsele din categoria X

```
String sql_1 = "SELECT\r\n"  
+ "    C.nume AS Nume\r\n"  
+ "FROM\r\n"  
+ "    Client C\r\n"  
+ "WHERE\r\n"  
+ "    NOT EXISTS (\r\n"  
+ "        SELECT 1\r\n"  
+ "        FROM Produs P2\r\n"  
+ "        WHERE P2.CategorieID = " + textField.getText().trim() + "\r\n"  
+ "        AND NOT EXISTS (\r\n"  
+ "            SELECT 1\r\n"  
+ "            FROM Bon B\r\n"  
+ "            JOIN Bon_Produs BP ON B.BonID = BP.BonID\r\n"  
+ "            JOIN Produs P ON BP.ProdusID = P.ProdusID\r\n"  
+ "            WHERE B.ClientID = C.ClientID\r\n"  
+ "            AND P.CategorieID = P2.CategorieID\r\n"  
+ "        )\r\n"  
+ "    );" + textField.getText().trim();
```

- Numarul de produse cumparate din categoria X

```
String sql_1 = "SELECT\r\n"
+ "    C.Nume AS NumeCategorie,\r\n"
+ "    COUNT(DISTINCT FP.ProdusID) AS NumarProduseVandute\r\n"
+ "FROM\r\n"
+ "    Categorie C\r\n"
+ "JOIN\r\n"
+ "    Produs P ON C.CategorieID = P.CategorieID\r\n"
+ "LEFT JOIN\r\n"
+ "    Factura_Produs FP ON P.ProdusID = FP.ProdusID\r\n"
+ "WHERE\r\n"
+ "    C.CategorieID = " + textField.getText().trim() + "\r\n"
+ "GROUP BY\r\n"
+ "    C.CategorieID, C.Nume\r\n"
+ "ORDER BY\r\n"
+ "    NumarProduseVandute DESC;";
```

- Clientii cu cea mai mare cantitate de produse din categoria X in cos

```
String sql_1 = "SELECT\r\n"
+ "    c.nume AS NumeClient,\r\n"
+ "    (\r\n"
+ "        SELECT TOP 1\r\n"
+ "            SUM(fp_inner.cantitate)\r\n"
+ "        FROM\r\n"
+ "            bon b_inner\r\n"
+ "        JOIN\r\n"
+ "            bon_produc bp_inner ON b_inner.BonID = bp_inner.BonID\r\n"
+ "        JOIN\r\n"
+ "            produs p_inner ON bp_inner.ProdusID = p_inner.ProdusID\r\n"
+ "        JOIN\r\n"
+ "            factura_produc fp_inner ON p_inner.ProdusID = fp_inner.ProdusID\r\n"
+ "        WHERE\r\n"
+ "            b_inner.ClientID = c.ClientID\r\n"
+ "        AND p_inner.CategorieID = " + textField.getText().trim() + "\r\n"
+ "        GROUP BY\r\n"
+ "            b_inner.ClientID\r\n"
+ "        ORDER BY\r\n"
+ "            SUM(fp_inner.cantitate) DESC\r\n"
+ "    ) AS TotalCantitatePerCategorie\r\n"
+ "FROM\r\n"
+ "    client c\r\n"
+ "JOIN\r\n"
+ "    bon b ON c.ClientID = b.ClientID\r\n"
+ "JOIN\r\n"
+ "    bon_produc bp ON b.BonID = bp.BonID\r\n"
+ "JOIN\r\n"
+ "    produs p ON bp.ProdusID = p.ProdusID\r\n"
+ "GROUP BY\r\n"
+ "    c.ClientID, c.nume\r\n"
+ "ORDER BY\r\n"
+ "    TotalCantitatePerCategorie DESC, NumeClient;";
```

- Cele mai multe produse cumparate din categoria X

```
String sql_1 = "SELECT\r\n"
+ "    C.Nume AS NumeCategorie,\r\n"
+ "    P.Nume AS NumeProdus,\r\n"
+ "    SUM(FP.Cantitate) AS TotalCantitateVanduta\r\n"
+ "FROM\r\n"
+ "    Categorie C\r\n"
+ "JOIN\r\n"
+ "    Produs P ON C.CategorieID = P.CategorieID\r\n"
+ "JOIN\r\n"
+ "    Factura_Produs FP ON P.ProdusID = FP.ProdusID\r\n"
+ "WHERE\r\n"
+ "    P.CategorieID = 2" + textField.getText().trim() + "\r\n"
+ "GROUP BY\r\n"
+ "    C.CategorieID, C.Nume, P.ProdusID, P.Nume\r\n"
+ "HAVING\r\n"
+ "    SUM(FP.Cantitate) = (\r\n"
+ "        SELECT TOP 1\r\n"
+ "            SUM(FP_inner.Cantitate)\r\n"
+ "        FROM\r\n"
+ "            Factura_Produs FP_inner\r\n"
+ "        WHERE\r\n"
+ "            P.ProdusID = FP_inner.ProdusID\r\n"
+ "        GROUP BY\r\n"
+ "            FP_inner.ProdusID\r\n"
+ "        ORDER BY\r\n"
+ "            SUM(FP_inner.Cantitate) DESC\r\n"
+ "    )\r\n"
+ "ORDER BY\r\n"
+ "    NumeCategorie, TotalCantitateVanduta DESC;";
```

- Clientii cu produse din mai mult de 3 categorii

```
String sql_1 = "SELECT\r\n"
+ "    C.Nume AS NumeClient,\r\n"
+ "    COUNT(DISTINCT P.CategorieID) AS NumarCategoriiAchizitionate\r\n"
+ "FROM\r\n"
+ "    Client C\r\n"
+ "JOIN\r\n"
+ "    Bon B ON C.ClientID = B.ClientID\r\n"
+ "JOIN\r\n"
+ "    Bon_Produs BP ON B.BonID = BP.BonID\r\n"
+ "JOIN\r\n"
+ "    Produs P ON BP.ProdusID = P.ProdusID\r\n"
+ "GROUP BY\r\n"
+ "    C.ClientID, C.Nume\r\n"
+ "HAVING\r\n"
+ "    COUNT(DISTINCT P.CategorieID) >= 3\r\n"
+ "ORDER BY\r\n"
+ "    NumarCategoriiAchizitionate DESC, NumeClient;";
```

- Clientii cu produsele din cos, mai vechi de 30 de zile

```
String sql_1 = "SELECT\r\n"
+ "    C.Nume AS NumeClient,\r\n"
+ "    SUM(FP.Total) AS TotalCheltuit\r\n"
+ "FROM\r\n"
+ "    Client C\r\n"
+ "JOIN\r\n"
+ "    Bon B ON C.ClientID = B.ClientID\r\n"
+ "JOIN\r\n"
+ "    (\r\n"
+ "        SELECT\r\n"
+ "            B1.ClientID,\r\n"
+ "            SUM(BP1.Cantitate) AS Total\r\n"
+ "        FROM\r\n"
+ "            Bon B1\r\n"
+ "        JOIN\r\n"
+ "            Bon_Produs BP1 ON B1.BonID = BP1.BonID\r\n"
+ "        JOIN\r\n"
+ "            Produs P1 ON BP1.ProdusID = P1.ProdusID\r\n"
+ "        WHERE\r\n"
+ "            CONVERT(DATE, B1.Data, 120) >= DATEADD(DAY, -30, CONVERT(DATE, GETDATE(), 120))\r\n"
+ "        GROUP BY\r\n"
+ "            B1.ClientID\r\n"
+ "    ) FP ON C.ClientID = FP.ClientID\r\n"
+ "GROUP BY\r\n"
+ "    C.ClientID, C.Nume\r\n"
+ "ORDER BY\r\n"
+ "    TotalCheltuit DESC, NumeClient;";
```

Baza de date si codul sursa: <https://s.go.ro/g2epdhwb>

Bibliografie

- <https://chat.openai.com/>
- Laboratoare Baze de date