

SGBD - Proiect part 1

Magazin de haine

Bucătaru Andreea & Bulboacă Maria

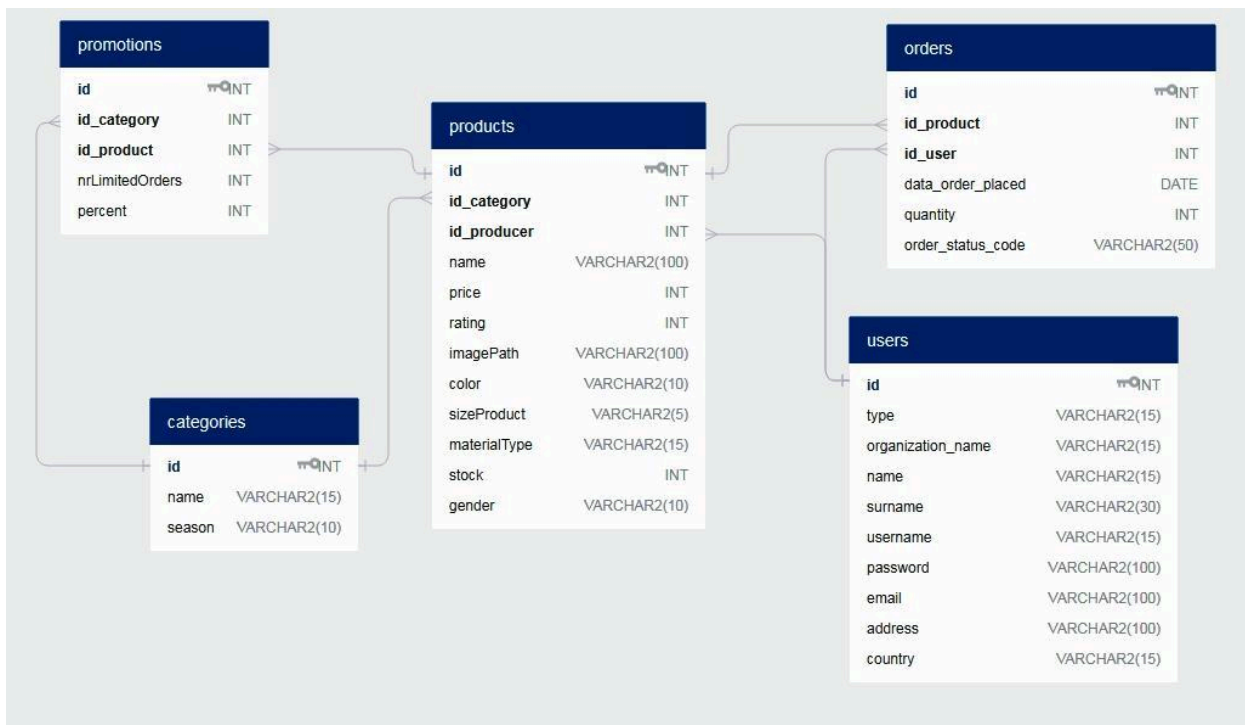
1. Descrierea proiectului

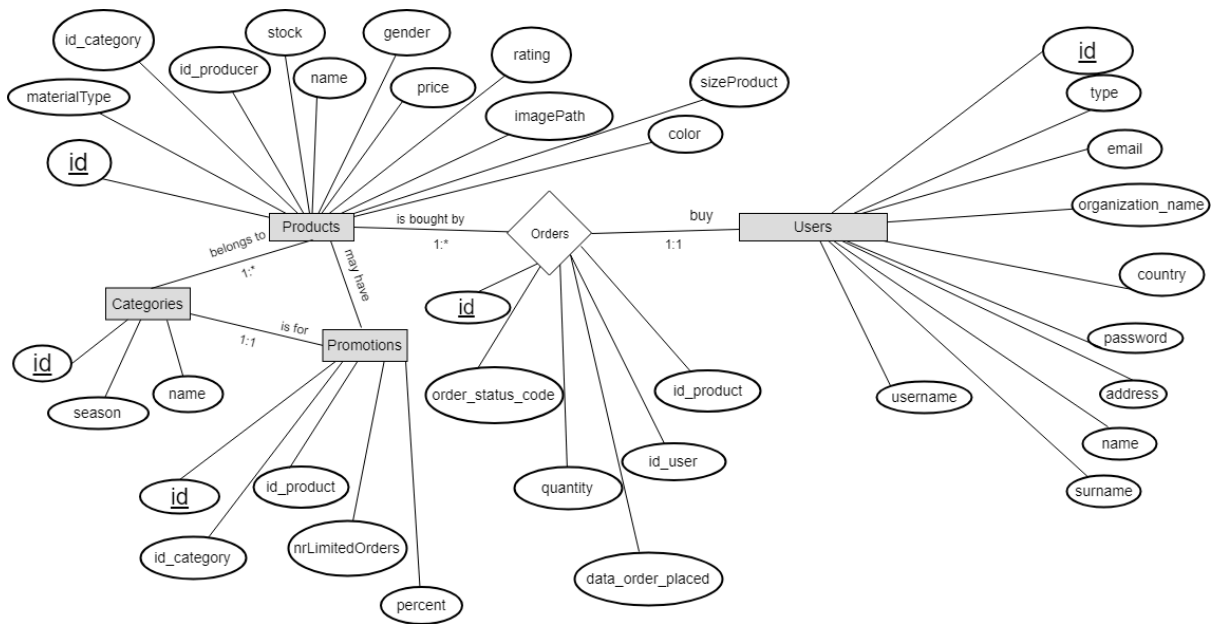
Proiectul constă în crearea unei aplicații care gestionează afacerea unui magazin online de haine.

Aplicația poate fi folosită de către:

- clienți: vor putea vizualiza produsele, dar pentru a plasa o comandă aceștia trebuie să se logheze pe baza unui cont deja existent sau pe care și-l vor putea crea
- producători: aceștia se vor conecta pe baza unui cont care le permite să introducă noi produse sau să le șteargă
- administrator: va gestiona baza de date pe baza unui cont și va putea șterge produse, elimina producători, adăuga promoții (prețurile produselor se vor actualiza automat la începutul și la sfârșitul promoției)

2. Schema structurării bazei de date





3. Demonstrație 4NF

Important de observat în această bază de date este faptul că singurile dependente funcționale existente într-o tabelă oarecare A sunt de forma:

cheie_primara_pt_tabela_A -> atribut_al_tabelei_A

- **1NF** : baza de date este în prima formă normală deoarece fiecare câmp cuprinde numai o valoare atomică, iar fiecare înregistrare este definită astfel încât să fie identificată în mod unic prin intermediul unei chei primare
- **2NF** : toate elementele unei tabele sunt dependente funcțional de totalitatea cheii primare (cheia primară din fiecare tabel este formată dintr-un singur atribut)
- **3NF** : baza de date este în forma 3NF deoarece orice atribut din orice tabel este determinat în mod UNIC de cheia primară a acelui tabel; de exemplu, dacă avem două produse, ambele cu id_category egale (ambele produse sunt bluze, să spunem), atunci nu înseamnă că gender va fi întotdeauna determinat de acea categorie; există bluze și pentru bărbați, și pentru femei
- **BCNF** : Cum nu există nicio dependență funcțională $X \rightarrow Y$ care să nu satisfacă macar una dintre următoarele condiții: Y aparține X (dependența e triviale) sau X e super cheie, atunci baza de date este în forma BCNF. Toate dependentele sunt de forma id_tuplu -> ..., iar id_tuplu e primary key.

5. Utilitatea structurilor de date folosite

Magazinul nostru va avea o pagină prin care un utilizator va putea căuta produse, care sunt stocate în tabela "products". Această tabelă conține

proprietățile produselor, astfel încât un client să le poată filtra în funcție de ce dorește.

Tabela “users” este folosită pentru conturile clienților, dar și a producătorilor, care își pot adăuga produsele singuri pe site.

Clienții pot da comenzi, care vor avea proprietățile stocate în tabela “orders”.

Produsele pot fi de mai multe tipuri, acestea fiind împărțite folosind tabela “categories”.

Fiecare produs sau categorie de produse pot avea anumite promoții sau reduceri, de aceea folosim tabela “promotions”. Câmpul “nrLimitedOrders” este folosit pentru promoțiile de tipul “reducere doar pentru primele x comenzi”.

Indecși:

- tabela “products”: index pentru culoare, mărime, preț, material, rating, producători
- tabela “categories”: index pentru sezon
- tabela “users”: index pentru (email, username, password)

6. Funcții și proceduri folosite

-- despre cont:

signUp(type, name, surname, username, password, email, address, country);

login(email, username, password);

updateUser(name, surname, username, password, email, address, country);

deleteAccount(email, username, password);

-- despre produse:

filterProducts(producer, color, materialType);

addProduct(id_product, id_user);

deleteProduct(id_product, id_user);

updateStock(id_product);

checkIfUserIsProducer(id_user);

isStockAvailableForProduct(id_product);

-- despre promoții

addPromotion(id_category, id_product, nrLimitedOrders, pecentage);

deletePromotion(id_category, id_product, nrLimitedOrders, pecentage);

updatePromotion(id_category, id_product, nrLimitedOrders);

getCurrentPromotions();

isPromotionAvailableForProduct(id_product);

-- despre comenzi:

placeOrder(id_product, quantity);

cancelOrder(id_order);

getOrders(id_user);

getOrderStatus(id_user, id_order);