

Baze de Date - Proiect

Gestionare magazin online

Luchian Alexei - grupa 1311A

Cadru didactic coordonator: Ș.I.dr.ing Mironeanu Cătălin

Scopul aplicației

Scopul aplicației este gestionarea unei aplicații în care utilizatorii pot vinde/cumpăra lucruri diverse precum haine, electrocasnice, manuale și alte chestii de care nu mai au nevoie, sau pot cumpăra servicii oferite de alți utilizatori. Aplicația funcționează ca și alte platforme marketplace cunoscute cum ar fi Vinted sau OLX, însă aceasta implementează și funcționalitatea Swipe care îl va ajuta pe utilizator să gliseze la stânga dacă acesta nu este interesat de produs sau să gliseze la dreapta în cazul în care acesta este interesat, astfel creându-se un coș de cumpărături doar cu produsele pentru care utilizatorul prezintă interes.

Aplicația este concepută pentru studenți și anume pentru cei de la TUIASI, prin urmare pentru simplitate tranzacțiile se vor efectua fizic (face-to-face) în campusul universitar, la universitate sau în orice altă locație din Iași unde le este convenabil ambilor utilizatori. Datorită acestui fapt nu avem nevoie și de tabele pentru informațiile ce țin de plăți online sau livrări cum ar fi carduri bancare, numere de telefoane ale curierilor, transporturi etc.

Fiecare utilizator va avea și un profil unde ceilalți pot vedea informații de contact, istoric de tranzacții, produse disponibile sau chiar și recenziile pe care utilizatorul le-a primit în urma vânzării altor produse, lucru care îi ajută pe ceilalți utilizatori să decidă dacă au de-a face cu o persoană de încredere sau nu.

Descrierea non-tehnică:

Platforma ajuta utilizatorii sa vanda/cumpere produsele de care nu mai au nevoie. Produsele de pe platforma vor avea preturi pana la 10.000 RON.

Fiecare utilizator va avea un profil unde ceilalti ii pot vedea istoricul de tranzactii si produsele pe care acesta le are de vanzare la momentul dat.

Pentru simplitate presupunem ca platile se vor efectua doar fizic (face-to-face) si o tranzactie va fi mereu compusa dintr-un singur produs.

Platforma va avea si functionalitatea Swipe, aplicatia va avea o sectiune unde utilizatorilor li se vor arata produsele disponibile, iar acestia vor trebui sa gliseze la dreapta daca sunt interesati de produs, si la stanga daca nu sunt interesati, astfel creandu-se un cos de cumparaturi doar cu produsele pentru care utilizatorul prezinta interes.

Structura și inter-relaționarea tabelelor

Model Logic:

Baza de date a aplicației este structurată în jurul a 7 entități, care modelează fluxul unui marketplace online: utilizatori, produse, tranzacții, recenzii, profiluri, detalii produse și coș cumpărături.

Entități:

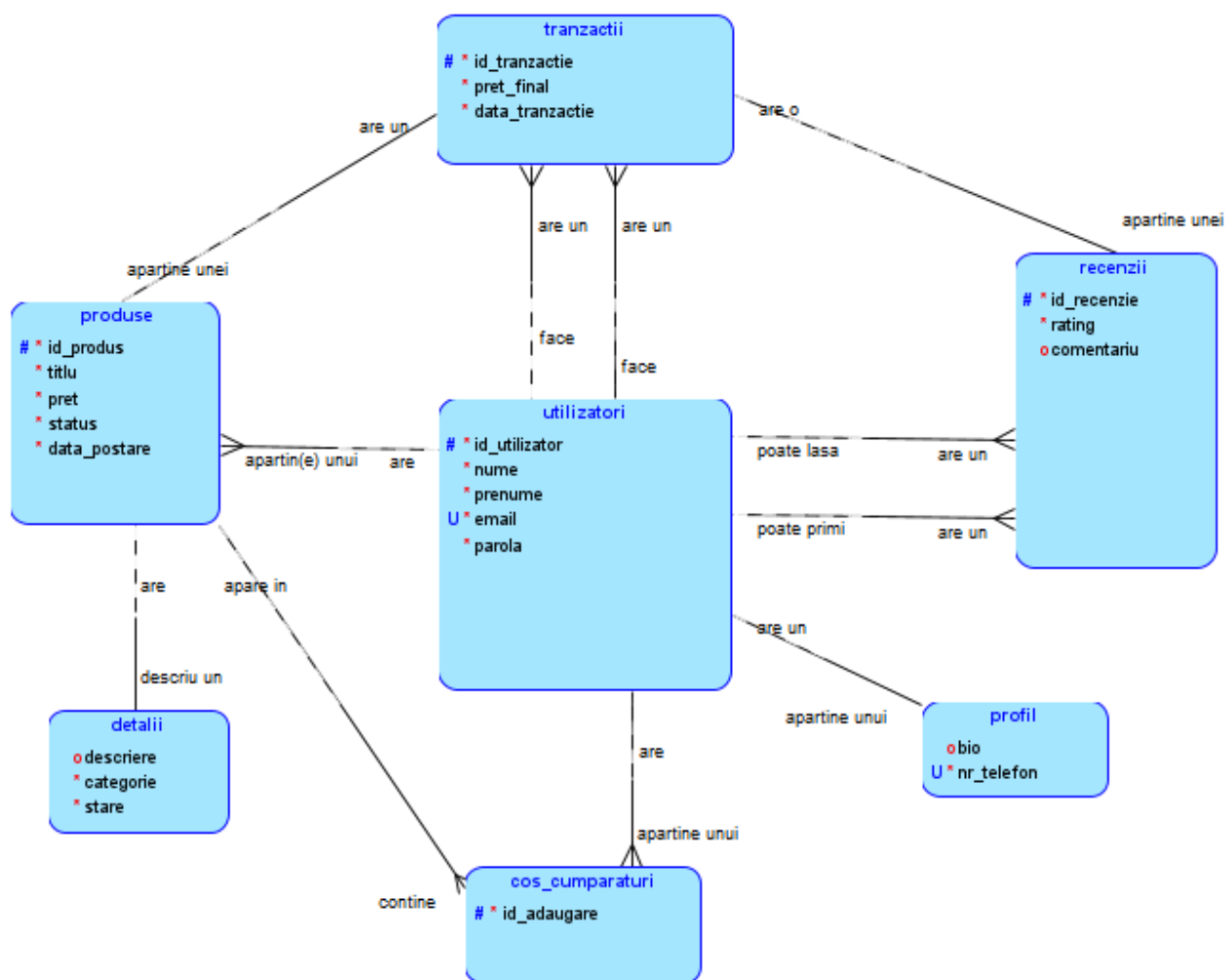
1. utilizatori - stochează informațiile de autentificare și identificare ale utilizatorilor platformei (nume, prenume, email, parolă)
2. produse - conține produsele listate de utilizatori pentru vânzare (titlu, preț, status, dată postare)
3. tranzactii - înregistrează toate tranzacțiile finalizare între utilizatori (preț final, dată tranzacție)
4. recenzii - permite utilizatorilor să lase feedback după finalizarea unei tranzacții (rating, comentariu)
5. profiluri - informații suplimentare despre utilizatori, accesibile public (bio, număr telefon)
6. detalii - categorii și stări ale produselor pentru filtrare și organizare
7. cos_cumparaturi - produsele adăugate de utilizatori manual sau prin funcționalitatea Swipe (glisare la dreapta)

Relațiile dintre entități:

1. utilizatori - produse (1:N) : un utilizator poate lista mai multe produse pentru vânzare, dar un produs aparține unui singur utilizator.
2. utilizatori - tranzacții (1:N) : avem această relație de două ori, deoarece o tranzacție poate avea doar un vânzător și un cumpărător în timp ce un utilizator (fie acesta cumpărător sau vânzător) poate participa în mai multe tranzacții.
3. produse - tranzactii (1:1) : un produs poate fi vândut o singură dată, iar o tranzacție se referă la un singur produs (conform cerinței). Odată ce tranzacția este finalizată, produsul devine indisponibil.
4. tranzactii - recenzii (1:1) : fiecare tranzacție finalizată poate primi exact o recenzie din partea cumpărătorului. Această relație asigură că feedbackul este legat direct de o tranzacție specifică.
5. utilizatori - recenzii (1:N) : această relație, la fel, apare de două ori, deoarece o recenzie are nevoie de doi utilizatori (un autor și un destinatar), iar un utilizator (fie acesta autor sau destinatar) poate lăsa/primi mai multe recenzii.
6. utilizatori - profil (1:1) : fiecare utilizator are exact un profil cu informații suplimentare. Separarea acestor date optimizează interogările de autentificare (care nu necesită bio sau număr telefon).

7. utilizatori - cos_cumparaturi (1:N) : un utilizator poate adăuga mai multe produse în coșul lui de cumpărături, iar un coș de cumpărături îi aparține unui singur utilizator.
8. produse - cos_cumparaturi (1:N) : un produs poate fi adăugat în mai multe coșuri de cumpărături (până a fi vândut) și fiecare rând din cos_cumparaturi reprezintă un singur produs.
9. produse - detalii (1:1) : fiecare set de detalii îi aparține unui singur produs și vice-versa.

Diagrama ER:



Model Relațional:

Transformarea din modelul logic în modelul relațional a respectat următoarele reguli:

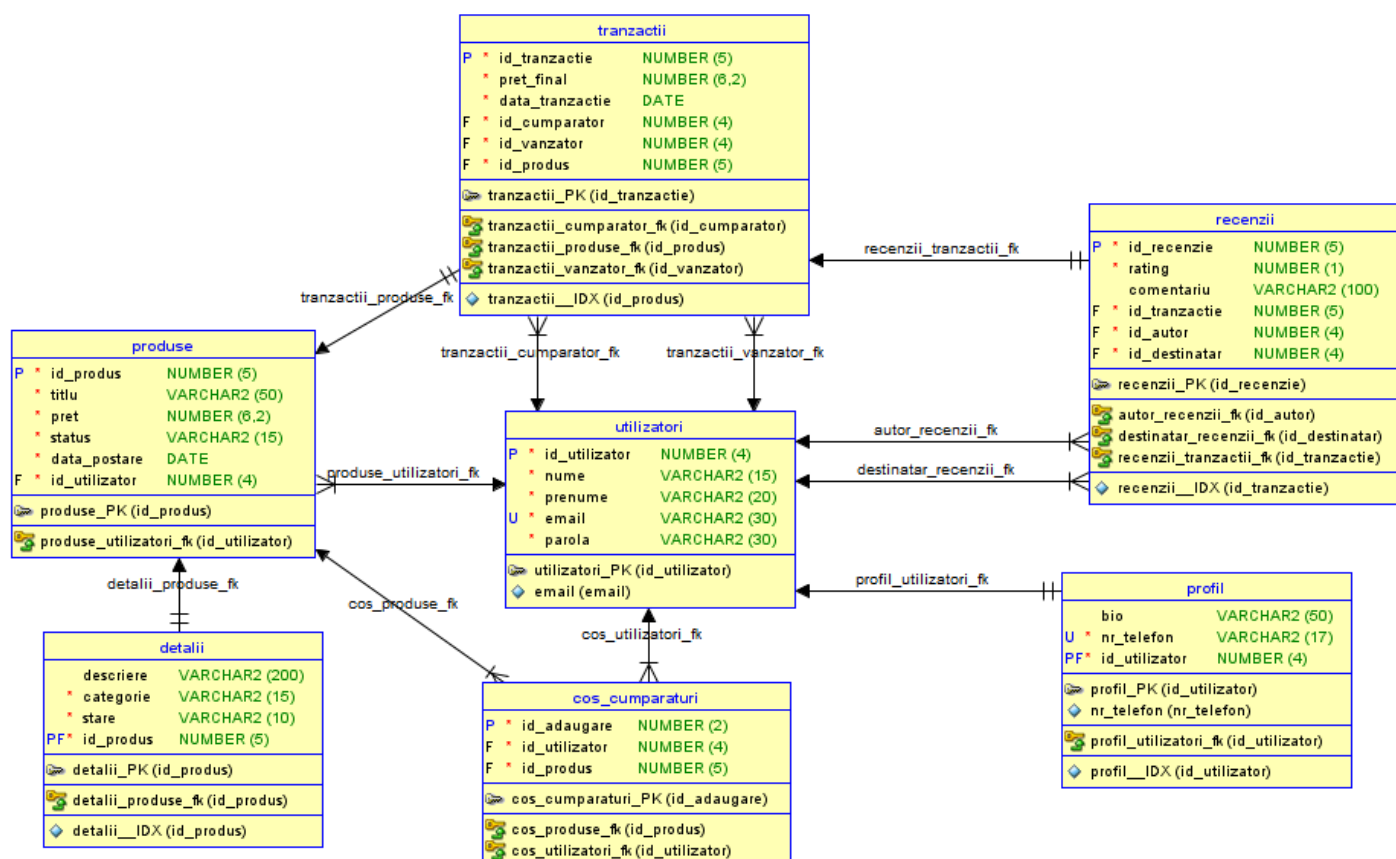
Transformarea entităților:

1. Fiecare entitate a devenit tabel.
2. Atributele au devenit coloane cu tipuri de date specifice SQL.
3. Cheile primare sunt generate automat prin mecanisme de tip autoincrement.

Transformarea relatiilor:

- Relațiile 1:N au fost implementate prin adăugarea cheilor străine în tabelul din partea N:
 - produse.id_utilizator -> utilizatori.id_utilizator
 - tranzactii.id_cumparator -> utilizatori.id_utilizator
 - tranzactii.id_vanzator -> utilizatori.id_utilizator
 - recenzii.id_autor -> utilizatori.id_utilizator
 - recenzii.id_destinatar -> utilizatori.id_utilizator
 - cos_cumparaturi.id_utilizator -> utilizatori.id_utilizator
 - cos_cumparaturi.id_produs -> produse.id_produs
- Relațiile 1:1 au fost implementate astfel:
 - tranzactii.id_produs -> produse.id_produs
 - recenzii.id_tranzactie -> tranzactii.id_tranzactie
 - detalii.id_produs -> produse.id_produs
 - profil.id_utilizator -> utilizatori.id_utilizator

Diagrama:



Descrierea coloanelor din tabele

utilizatori:

- id_utilizator este de tip number(4), deoarece aplicația este concepută pentru un număr restrâns de utilizatori, adică pentru studenții TUIASI cum s-a menționat și mai sus, prin urmare un număr de 9999 utilizatori este suficient să cuprindă toate persoanele interesate în folosirea aplicației pentru etapa inițială.
- prenume este de tip varchar2(20), deoarece pe lângă posibilitatea existenței unei persoane cu un nume lung, mai există posibilitatea ca utilizatorii cu două prenume să-și dorească să le folosească pe amândouă.

produse:

- id_produs este de tip number(5), deoarece 99999 mi s-a părut un număr destul de mare, plecând de la premisa că mulți utilizatori vor intra să cumpere produse, nu să le vândă, prin urmare un număr de 10 produse per persoană pare suficient.
- titlu este de tip varchar2(50), deoarece titlul poate să conțină mai multe cuvinte cum ar fi "Lenovo Ideapad Gaming 3", iar 50 de caractere permite stocarea unui titlu destul de explicit.
- pret este de tip number(6,2), deoarece aplicația permite stocarea produselor cu prețuri până la 10000 RON, 6 fiind numărul total de cifre permis, 2 reprezentând partea zecimală, iar restul (adică 4) partea întreagă, prin urmare prețul maxim poate fi 9999,99RON.

tranzactii:

- id_tranzactie este de tip number(5), deoarece nu au cum să fie mai multe tranzacții decât numărul de produse.
- pret_final este de tip number(6,2), deoarece aplicația permite stocarea produselor cu prețuri până la 10000 RON, 6 fiind numărul total de cifre permis, 2 reprezentând partea zecimală, iar restul (adică 4) partea întreagă, prin urmare prețul maxim poate fi 9999,99RON.

recenzii:

- id_recenzie este de tip number(5), deoarece fiecare tranzacție permite o singură recenzie, prin urmare numărul posibil de recenzii e același cu numărul de tranzacții.
- rating este de tip number(1), deoarece evaluarea se face pe o scară de la 1 la 5 (stele)

profil:

- nr_telefon este de tip varchar(17), deoarece utilizatorii se pot loga atât cu numere din România cât și din străinătate, iar cel mai lung număr posibil e din 17 cifre, inclusiv prefixul.

Descrierea constrângerilor

PRIMARY KEY:

Fiecare tabel are o cheie primară care identifică unic fiecare înregistrare. Toate cheile primare sunt generate automat prin autoincrement, ceea ce elimină riscul de duplicate și simplifică inserările.

Exemple:

- ❖ utilizatori_PK(id_utilizator) - identifică unic fiecare utilizator
- ❖ produse_PK(id_produs) - identifică unic fiecare produs

FOREIGN KEY:

Cheile străine asigură integritatea referențială între tabele, garantând că relațiile sunt întreținute corect și că nu există referințe către înregistrări inexistente.

Exemple:

- ❖ produse_utilizatori_fk(id_utilizator) -> utilizatori(id_utilizator) - asigură că fiecare produs este listat de un utilizator existent în sistem.
- ❖ tranzactii_utilizatori_fk(id_cumparator) -> utilizatori(id_utilizator) - garantează că cumpărătorul într-o tranzacție este un utilizator valid.

UNIQUE:

Constrângerile UNIQUE garantează că anumite valori nu se repetă în baza de date.

Exemple:

- ❖ utilizatori: email(email) - asigură că fiecare utilizator are o adresă de email unică.
- ❖ recenzii: recenzii_IDX(id_tranzactie) - implementează relația 1:1 între tranzacții și recenzii, permițând o singură recenzie per tranzacție.

NOT NULL:

Constrângerile NOT NULL asigură că coloanele esențiale conțin întotdeauna valori, prevenind inserarea de date incomplete.

Exemple:

- ❖ utilizatori: nume, prenume, email, parola - toate sunt NOT NULL, deoarece sunt esențiale pentru identificarea și autentificarea utilizatorului.
- ❖ produse: titlu, pret, status - toate sunt NOT NULL pentru că definesc un produs valid listat pe platformă.

CHECK:

Constrângerile CHECK limitează valorile acceptate la cele logic corecte, prevenind introducerea de valori absurde.

Exemple:

- ❖ produse: CHECK (pret > 0) - asigură că prețul este pozitiv
- ❖ produse: CHECK (status IN ('disponibil', 'vandut', 'rezervat')) - limitează statusul la valorile valide
- ❖ recenzii: CHECK (rating >= 1 AND rating <= 5) - garantează că rating-ul este între 1 și 5 stele, conform sistemului de evaluare.
- ❖ validare email cu regexp: ne asigură că email-ul conține @ și . în poziții valide, prevenind introducerea de adrese invalide.
- ❖ validare nume și prenume cu regexp: acceptă doar litere, spații și cratime, prevenind introducerea de cifre sau caractere speciale.
- ❖ validare număr telefon: acceptă numere cu 10-17 cifre, cu sau fără prefix internațional, suportând atât numere românești cât și internaționale.