

## E-Commerce Online Shop

## Documentație Proiect Bază de Date

- **1. Introducere – Prezentarea modelului din lumea reală**
- Scopul acestui proiect este proiectarea și implementarea unei baze de date relaționale pentru un **Magazin Online E-Commerce**, în conformitate cu cerințele cursului de Baze de Date. Proiectul modelează o platformă reală de comerț electronic, concentrându-se pe entitățile esențiale și regulile care guvernează funcționarea unui astfel de sistem.
- Un sistem e-commerce permite utilizatorilor să își creeze conturi, să navigheze printre produse organizate pe categorii, să plaseze comenzi, să efectueze plăți și să lase recenzii pentru produsele achiziționate. Din perspectiva bazelor de date, este esențial ca toate aceste operații să fie susținute de o schemă bine structurată, care să asigure integritatea datelor, să evite redundanța și să permită interogări eficiente.
- Acest proiect parcurge toate etapele proiectării unei baze de date:
  - Analiza modelului din lumea reală
  - Modelarea Entitate–Relație (ER)
  - Modelarea conceptuală
  - Proiectarea schemelor relaționale
  - Implementarea fizică în PostgreSQL

## **2. Reguli și ipoteze ale modelului din lumea reală**

Următoarele reguli descriu modul de funcționare al sistemului în lumea reală:

1. Un utilizator poate plasa zero sau mai multe comenzi.
2. Fiecare comandă este plasată de exact un utilizator.
3. O comandă poate conține unul sau mai multe produse.
4. Un produs poate apărea în zero sau mai multe comenzi.
5. Produsele sunt organizate în categorii.
6. Fiecare produs aparține exact unei categorii.
7. O comandă poate avea zero sau mai multe plăți.
8. Fiecare plată este asociată exact unei comenzi.
9. Utilizatorii pot scrie recenzii pentru produse.
10. Fiecare recenzie este scrisă de un utilizator și se referă la un produs.

Aceste reguli sunt impuse prin chei primare, chei străine și constrângeri de integritate.

### 3. Diagrama Entitate–Relație (ER)

Diagrama ER reprezintă structura bazei de date la nivel înalt, evidențiind entitățile, atributele și relațiile dintre acestea.

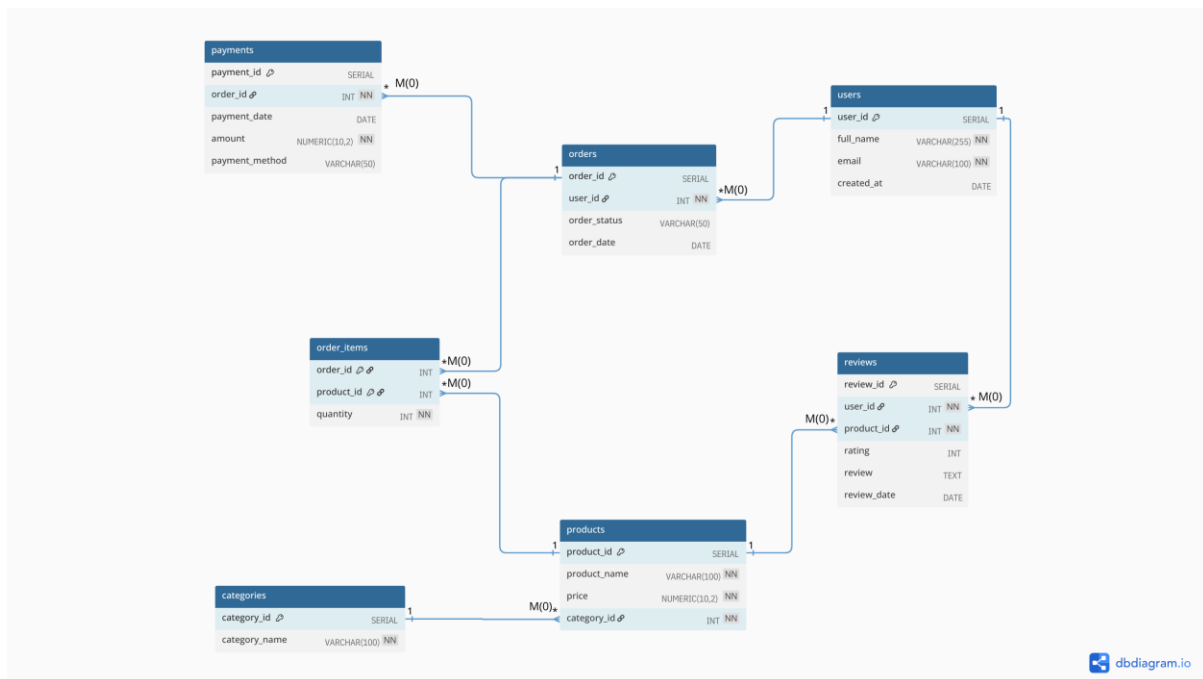
#### Entități

- users (Utilizatori)
- categories (Categorii)
- products (Produse)
- orders (Comenzii)
- order\_Items (Produse\_Comandă)
- payments (Plăți)
- reviews (Recenzii)

#### Relații

- Utilizatorii plasează comenzi (1:N)
- Comenzile conțin produse comandate (1:N)
- Produsele apar în produse comandate (1:N)
- Categoriile grupează produse (1:N)
- Comenzile generează plăți (1:N)
- Utilizatorii scriu recenzii (1:N)
- Produsele primesc recenzii (1:N)

## Ilustrarea diagramei ER :



## 4. Descrierea entităților, atributelor, cheilor și cardinalităților

### 4.1 Utilizatori (Users)

- **Cheie primară:** user\_id
- **Atribute:** full\_name, email, created\_at
- **Descriere:** Stochează datele utilizatorilor înregistrați pe platformă.

### 4.2 Categori (Categories)

- **Cheie primară:** category\_id
- **Atribute:** category\_name
- **Descriere:** Reprezintă clasificarea produselor.

### 4.3 Produse (Products)

- **Cheie primară:** product\_id
- **Cheie străină:** category\_id
- **Atribute:** product\_name, price
- **Descriere:** Stochează informații despre produsele disponibile.

### 4.4 Comenzi (Orders)

- **Cheie primară:** order\_id
- **Cheie străină:** user\_id
- **Atribute:** order\_status, order\_date

### 4.5 Produse\_Comandă (Order\_Items)

- **Cheie primară:** (order\_id, product\_id)
- **Chei străine:** order\_id, product\_id
- **Atribute:** quantity

### 4.6 Plăți (Payments)

- **Cheie primară:** payment\_id

- **Cheie străină:** order\_id
- **Attribute:** payment\_date, amount, payment\_method

#### 4.7 Recenzii (Reviews)

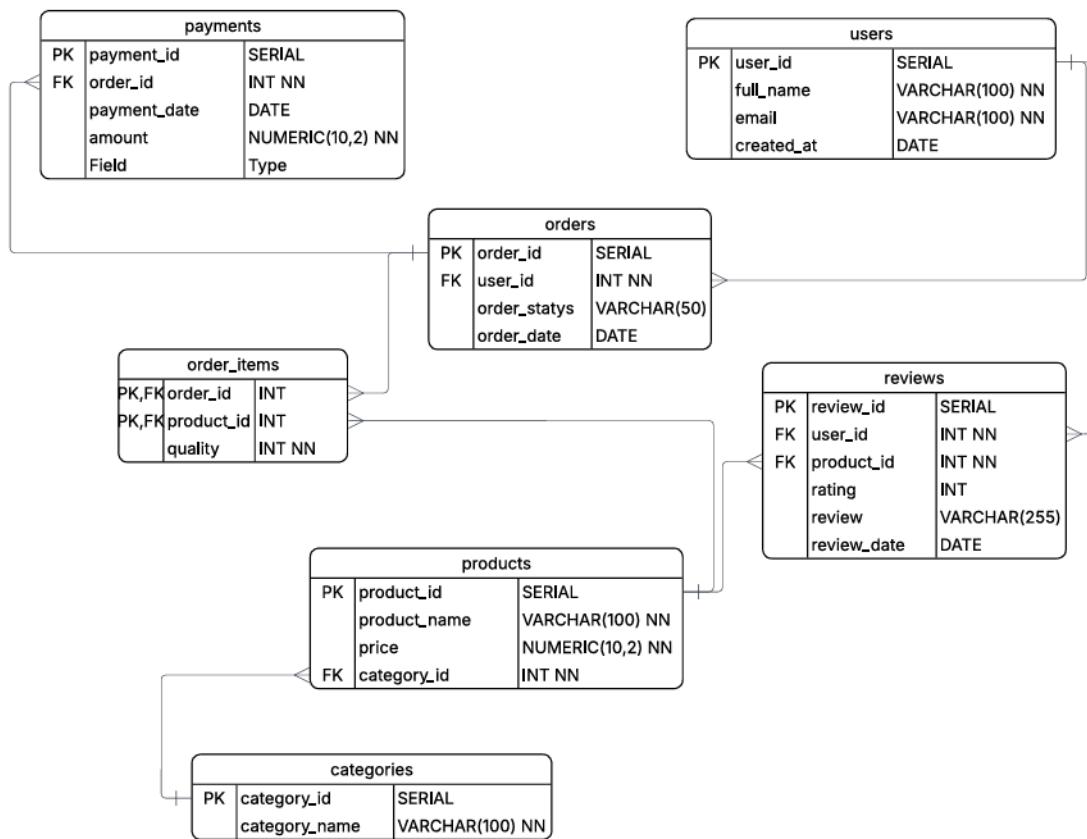
- **Cheie primară:** review\_id
- **Chei străine:** user\_id, product\_id
- **Attribute:** rating, review, review\_date

### 5. Diagrama conceptuală

Diagrama conceptuală rafinează modelul ER prin evidențierea dependențelor, a entităților asociative și a regulilor de integritate.

- Entități independente: users, categories, products
- Entități dependente: orders, payments, reviews
- Entitate asociativă: order\_items

**Ilustrarea diagramei conceptuale:**



## 6. Constrângeri de integritate

Baza de date implementează următoarele tipuri de constrângeri:

### 6.1 Integritatea entității

- Fiecare tabel are cheie primară.
- Cheile primare sunt unice și nu pot avea valori NULL.

### 6.2 Integritatea referențială

- Cheile străine asigură corectitudinea referințelor dintre tabele.
- Este utilizată constrângerea ON DELETE CASCADE acolo unde este necesar.

### 6.3 Integritatea domeniului

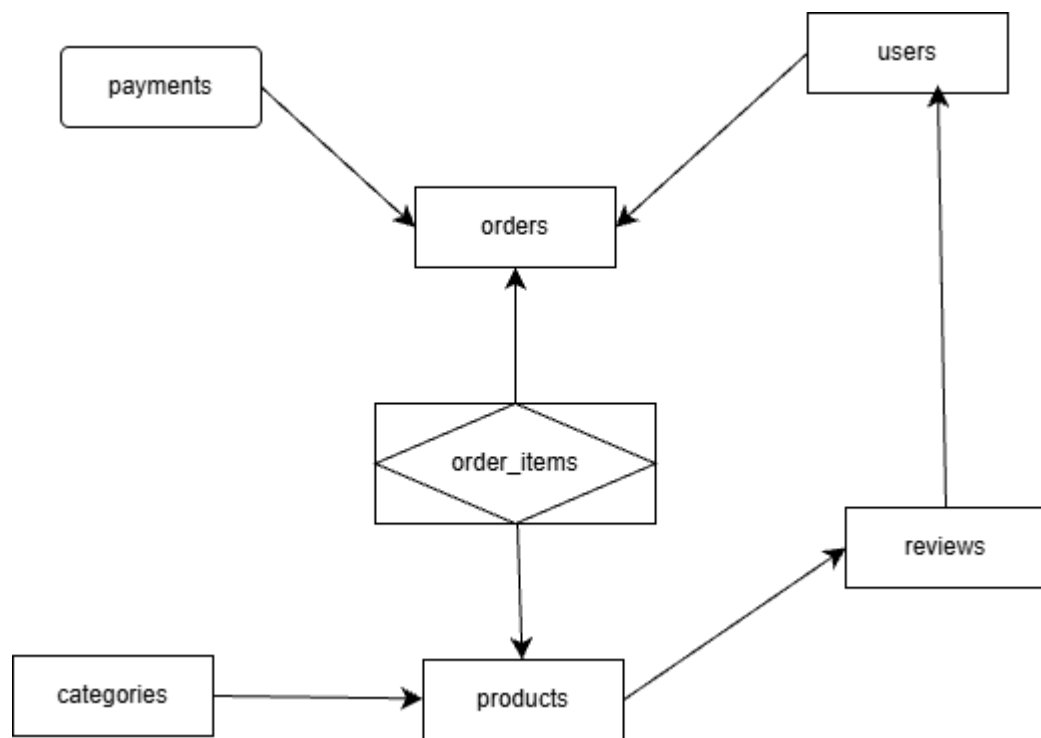
- Constrângeri CHECK pentru preț, cantitate și rating.
- Constrângeri NOT NULL pentru attribute obligatorii.
- Constrângeri UNIQUE pentru email și numele categoriilor.

## 7. Scheme relaționale

- USERS(user\_id PK, full\_name, email, created\_at)
- CATEGORIES(category\_id PK, category\_name)
- PRODUCTS(product\_id PK, product\_name, price, category\_id FK)
- ORDERS(order\_id PK, user\_id FK, order\_status, order\_date)
- ORDER\_ITEMS(order\_id PK/FK, product\_id PK/FK, quantity)
- PAYMENTS(payment\_id PK, order\_id FK, payment\_date, amount, payment\_method)
- REVIEWS(review\_id PK, user\_id FK, product\_id FK, rating, review, review\_date)



## Ilustratie de Schema Relatională:



## 8. Implementarea bazei de date

Baza de date a fost implementată folosind **PostgreSQL**, instalat local. Scripturile SQL au fost scrise în Visual Studio Code și executate folosind interfața PostgreSQL.

### 8.1 Crearea tabelelor

Toate tabelele au fost create utilizând instrucțiuni CREATE TABLE, incluzând chei primare, chei străine și constrângeri.

### 8.2 Introducerea datelor

Fiecare tabel conține cel puțin 10 înregistrări pentru a permite testarea corectă a interogărilor.

## 9. Vizualizări (Views)

Au fost create două vizualizări pentru a respecta cerințele proiectului:

### 9.1 Vizualizare compusă (actualizabilă)

- Afișează utilizatorii și comenzile acestora.
- Permite operații de tip UPDATE asupra tabelelor de bază.

### 9.2 Vizualizare complexă

- Agregă date despre vânzările produselor.
- Utilizează JOIN, GROUP BY și funcții agregate.

## 10. Interogări exemplu și cerințe de interfață

### 10.1 Sortare (ORDER BY)

Produsele sunt afișate în ordine descrescătoare după preț.

### 10.2 Interogare cu mai multe tabele

O interogare care combină utilizatori, comenzi, produse comandate și produse, filtrată după statusul comenzii și preț.

### 10.3 Funcții de grup și HAVING

Categoriile sunt filtrate în funcție de numărul de produse asociate.

## 11. Demonstrarea constrângerii ON DELETE CASCADE

Ștergerea unui utilizator determină ștergerea automată a comenzilor, plăților, produselor comandate și recenziilor asociate, asigurând integritatea referențială.

## 12. Normalizarea bazei de date

Schema bazei de date respectă **Forma Normală a Treia (3NF)**:

- Nu există dependențe parțiale
- Nu există dependențe tranzitive
- Redundanța datelor este minimă

### **13. Instrumente și tehnologii utilizate**

- PostgreSQL
- SQL
- Visual Studio Code
- pgAdmin

### **14. Concluzii**

Acest proiect demonstrează întregul proces de proiectare și implementare a unei baze de date pentru un sistem e-commerce real. Toate cerințele academice au fost respectate, iar baza de date rezultată este coerentă, scalabilă și pregătită pentru integrarea cu o interfață de aplicație.

### **15. Dezvoltări viitoare**

- Implementarea unei interfețe web
- Optimizarea performanței prin indexare
- Extinderea funcționalităților de securitate și autentificare