

Universitatea Tehnica “Gheorghe Asachi”, Iasi

APROVIZIONARE RESTAURANTE

Apostol Bianca-Cristina

Grupa: 1309A

Prof. Coord. : Butincu Cristian-Nicolae,

Avram Sorin

2024-2025

1. Scopul Lucrării

În cadrul acestui proiect, am abordat scopul aprovizionării restaurantelor din perspectiva eficientizării proceselor de achiziție și gestionare a stocurilor de produse alimentare, în mod special carne de pui, porc și vită, care reprezintă baza meniurilor multor restaurante. Aprovizionarea este un aspect esențial pentru buna funcționare a oricărui restaurant, iar scopul meu a fost să creez o soluție informatică care să optimizeze întregul lanț de aprovizionare, reducând risipa, costurile operaționale și asigurându-se că restaurantele dispun de produse proaspete și suficiente pentru a răspunde cerințelor clienților.

În industria restaurantelor, gestionarea aprovizionării poate reprezenta o provocare semnificativă din cauza diversității produselor necesare, fluctuațiilor de preț și cerințelor stricte legate de calitate și termenul de valabilitate. În cadrul acestui proiect, am identificat necesitatea unei platforme integrate care să permită restaurantelor să plaseze comenzi precise către furnizori, să urmărească livrările în timp real și să aibă o imagine clară asupra stocurilor disponibile. Astfel, scopul nostru a fost de a crea un sistem care să automatizeze aceste procese, minimizând intervenția umană și riscurile de erori, care pot duce la pierderi financiare și probleme operaționale.

Un obiectiv major al acestui proiect a fost reducerea erorilor de comandă și de gestionare a stocurilor, care pot apărea atunci când se realizează aprovizionarea manuală. Prin implementarea unui sistem centralizat și automatizat, restaurantele pot comanda produsele necesare într-un mod eficient, bazându-se pe date reale și precise. De exemplu, o comandă de carne de pui se va face în funcție de cerințele restaurantului și de cantitatea disponibilă în stoc, evitându-se astfel achizițiile excesive sau lipsa produselor esențiale. Mai mult, sistemul le permite restaurantelor să actualizeze cantitățile de produse pe măsură ce acestea sunt consumate, ceea ce ajută la menținerea unui echilibru optim între stocuri și necesități.

De asemenea, un alt scop al acestui proiect este asigurarea unui control mai mare asupra fluxului financiar legat de aprovizionare. Prin gestionarea eficientă a comenzilor și stocurilor, restaurantele pot reduce riscurile de cheltuieli suplimentare sau achiziții necontrolate. Fiecare comandă poate fi monitorizată și evaluată, iar restaurantele pot ajusta bugetul și cantitățile de produse în funcție de necesitățile reale ale locației. Aceasta contribuie la optimizarea cheltuielilor și la menținerea unei rentabilități mai mari.

Scopul final al acestui sistem de aprovizionare este de a îmbunătăți eficiența generală a activităților unui restaurant. Printr-o gestionare mai bună a comenzilor și stocurilor, restaurantele pot economisi timp, reduce costurile administrative și asigura un flux continuu de produse de calitate. În plus, procesul de aprovizionare devine mai transparent, iar restaurantele pot urmări istoricul comenzilor și livrărilor, având întotdeauna o imagine clară asupra resurselor disponibile.

2. Structura si inter-relationarea tabelelor

Tabelele folosite:

1. **PRODUSE:**

- Coloană principală: id_produc (cheie primară, identifică unic fiecare produs).
- Alte coloane: denumire, pret, cantitate, id_furnizor.
- Relații: Legătura cu tabelul FURNIZORI prin id_furnizor (fiecare produs este furnizat de un anumit furnizor). Este legat de comenzi-produse prin id_produc (produsele apar în comenzi).

2. **FURNIZORI:**

- Coloană principală: id_furnizor (cheie primară, identifică unic fiecare furnizor).
- Alte coloane: denumire, adresa, telefon.
- Relații: Este legat de tabelul PRODUSE (un furnizor poate avea mai multe produse).

3. **COMENZI:**

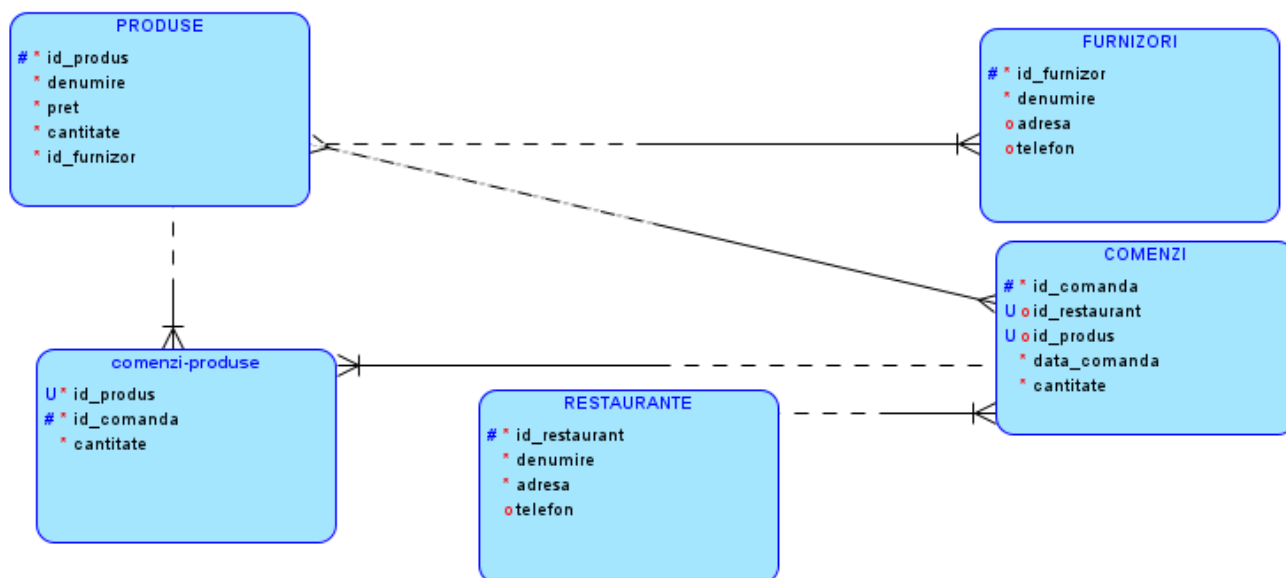
- Coloană principală: id_comanda (cheie primară, identifică unic fiecare comandă).
- Alte coloane: id_restaurant, id_produc, data_comanda, cantitate.
- Relații: Legătura cu RESTAURANTE prin id_restaurant (fiecare comandă este făcută de un restaurant). Legătura cu tabelul intermediar comenzi-produse.

4. **RESTAURANTE:**

- Coloană principală: id_restaurant (cheie primară, identifică unic fiecare restaurant).
- Alte coloane: denumire, adresa, telefon.
- Relații: Este legat de COMENZI prin id_restaurant (un restaurant poate plasa mai multe comenzi).

5. **comenzi-produse:**

- Este un tabel intermediar între PRODUSE și COMENZI.
- Coloane:
 - id_produc (cheie externă, referă un produs din PRODUSE).
 - id_comanda (cheie externă, referă o comandă din COMENZI).
 - cantitate.
- Relații: Conectează PRODUSE și COMENZI, permițând să se înregistreze câte produse sunt incluse într-o comandă.



3. Descrierea constrangerilor folosite

În cadrul acestui proiect, am utilizat mai multe tipuri de constrângeri pentru a asigura integritatea datelor, corectitudinea relațiilor dintre tabele și implementarea unui model de date funcțional pentru gestionarea aprovizionării restaurantelor. Constrângerile sunt reguli definite la nivel de bază de date care impun restricții asupra valorilor inserate într-un tabel sau asupra relațiilor între tabele. Iată o descriere detaliată a principalelor tipuri de constrângeri folosite:

1. Constrângerea PRIMARY KEY:

Constrângerea PRIMARY KEY a fost utilizată pentru a garanta unicitatea fiecărui rând dintr-un tabel. Acesta este un concept esențial pentru menținerea integrității datelor, deoarece fiecare entitate din baza de date (cum ar fi o comandă, un produs sau un restaurant) trebuie să aibă un identificator unic. De exemplu:

În tabelul COMENZI, *id_comanda* și *RESTAURANTE_id_restaurant* formează cheia primară, asigurându-se că fiecare comandă este asociată cu un restaurant specific într-o manieră unică.

În tabelul PRODUSE, *id_produș* este folosit ca cheie primară pentru a asigura că fiecare produs este unic.

În tabelul RESTAURANTE, *id_restaurant* este cheia primară, garantând unicitatea fiecărui restaurant înregistrat.

2. Constrângerea FOREIGN KEY:

Constrângerile de tip FOREIGN KEY sunt folosite pentru a crea relații între tabele, asigurându-se că valorile dintr-o coloană dintr-un tabel corespund valorilor dintr-o coloană a unui alt tabel. Acestea sunt esențiale pentru menținerea consistenței între datele care sunt legate între ele. De exemplu:

În tabelul COMENZI, RESTAURANTE_id_restaurant este o cheie străină care face referire la id_restaurant din tabelul RESTAURANTE. Astfel, fiecare comandă este asociată cu un restaurant valid.

În tabelul comenzi-produse, C_id_comanda și C_R_id_restaurant sunt chei străine care leagă comenzile de produsele comandate, asigurându-se că nu există comenzi care să nu fie asociate cu produse valide sau cu un restaurant existent.

În tabelul FURNIZORI, PRODUSE_id_furnizor este o cheie străină care face legătura între furnizori și produsele pe care le livrează. Acest lucru este necesar pentru a urmări proveniența fiecărui produs.

3. Constrângerea UNIQUE:

Constrângerea UNIQUE asigură că valorile dintr-o coloană sau combinație de coloane sunt unice, evitând astfel duplicarea datelor. Aceasta este importantă în cazul în care există reguli care impun unicitatea unor attribute specifice. De exemplu:

În tabelul COMENZI, id_restaurant și id_produs au constrângeri UNIQUE pentru a preveni ca un restaurant să comande același produs de mai multe ori într-o singură comandă. Acest lucru ajută la gestionarea comenzilor și la evitarea erorilor de procesare.

În tabelul PRODUSE, id_furnizor are o constrângere UNIQUE pentru a asigura că fiecare produs este asociat unui furnizor unic, evitând conflictele și confuzia în ceea ce privește sursa fiecărui produs.

4. Constrângerea NOT NULL:

Constrângerea NOT NULL a fost folosită pentru a asigura că anumite coloane nu pot conține valori NULL. Aceasta este o practică esențială pentru a garanta completitudinea datelor. De exemplu:

În tabelul COMENZI, coloanele id_comanda, data_comanda, cantitate, și RESTAURANTE_id_restaurant sunt definite cu constrângerea NOT NULL pentru a se asigura că toate comenzile sunt corect înregistrate și că nu există comenzi incomplete.

În tabelul PRODUSE, coloanele id_produs, denumire, pret, și cantitate sunt esențiale și nu pot fi lăsate goale, deoarece acest lucru ar crea inconsistență în inventar.

4. Descrierea use-case-urilor

În cadrul acestui proiect, use-case-urile (cazurile de utilizare) se referă la diferitele scenarii în care utilizatorii interacționează cu sistemul pentru gestionarea aprovizionării și comenzilor într-un restaurant. Exemplele principale de use-case includ:

Adăugarea unui restaurant: Un administrator al sistemului poate introduce un nou restaurant, completând informații precum numele, adresa și telefonul acestuia. Acesta va putea apoi să adauge comenzi și să le asocieze cu restaurantul respectiv.

Adăugarea unui furnizor și a produselor sale: Furnizorii sunt adăugați în sistem cu detalii precum denumirea fermei, adresa și produsele pe care le livrează. Acest lucru permite gestionarea aprovizionării cu carne de pui, porc sau vită.

Plasarea unei comenzi: Utilizatorul (restaurant) poate plasa o comandă pentru diferite produse de carne. Această comandă este asociată cu un restaurant și conține informații despre cantitate și produsele solicitate.

Urmărirea stocurilor de produse: Administratorii pot vizualiza și actualiza cantitățile produselor disponibile, pentru a preveni epuizarea stocurilor sau suprastocarea.

Generarea rapoartelor: Sistemul poate genera rapoarte cu privire la comenzile plasate, produsele livrate, stocurile disponibile și furnizorii care au livrat produsele, oferind astfel informații esențiale pentru deciziile de afaceri.

Aceste use-case-uri sunt esențiale pentru fluxul de lucru al unui restaurant și pentru o gestionare eficientă a aprovizionării și comenzilor.

5. Concluzii

Proiectul pentru gestionarea comenzilor într-un sistem de restaurante a fost conceput pentru a asigura o structură eficientă și flexibilă în ceea ce privește gestionarea comenzilor, produselor și furnizorilor. Prin utilizarea a șase tabele principale — COMENZI, RESTAURANTE, PRODUSE, FURNIZORI, comenzi-produse, și prod-com — am reușit să stabilim relații clare și eficiente între diferitele entități, folosind chei primare și externe pentru a garanta integritatea datelor și a facilita operațiunile de interogare și actualizare.

Prin implementarea constrângerilor de unicitate și a relațiilor între tabele, am asigurat că fiecare comandă este corect asociată cu restaurantul și produsele comandate, iar furnizorii sunt legați corespunzător de produsele pe care le furnizează. Structura bazei de date permite gestionarea flexibilă a comenzilor și produselor, cu un accent deosebit pe performanța și integritatea tranzacțiilor.

Această arhitectură este scalabilă, putând fi extinsă ușor pentru a include funcționalități suplimentare, cum ar fi monitorizarea stocurilor sau gestionarea plăților. Astfel, proiectul oferă o soluție completă și robustă pentru un sistem de restaurante, capabil să răspundă eficient cerințelor operaționale și să suporte creșterea în volum de date pe termen lung.