UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: CALCULATOARE
SPECIALIZAREA: TEHNOLOGIA INFORMAȚIEI

Proiect PBD

Coordonator ştiinţific
asoc. ing. Baltariu Ionuţ-Alexandru

Student
Rareș-Andrei Dancău, 1410A

Descrierea proiectului

Proiectul doreşte implementarea unui sistem de gestiune a stocurilor pentru un magazin specializat în componente electronice. Baza de date este concepută pentru a gestiona informaţiile despre produse, stocul disponibil în magazin şi vânzările efectuate.

Tabelul produse conţine informaţii despre produse, inclusiv tipul, producătorul şi starea lor, legate de alte tabele prin chei străine. Tabelul magazie stochează cantitatea disponibilă şi preţul pentru fiecare produs în parte, iar tabelele tip_produs, nume_producator şi stare_produs sunt utilizate pentru a defini şi gestiona tipurile de produse, producătorii şi starea produselor. Triggerele şi constrângerile asigură integritatea datelor, prevenind inserarea sau actualizarea duplicatelor şi verificând disponibilitatea stocului înainte de efectuarea unei vânzări. Astfel, proiectul propus optimizează procesele de gestionare a stocurilor şi vânzărilor în cadrul magazinului nostru de componente electronice.

Testele executate

În cadrul script-ului script_testare.sql sunt efectuate mai multe teste aplicate asupra a două tranzacţii, dar mai este şi tratată partea de SQL Injection. Cele două tranzacţii sunt după cum urmează:

- Prima tranzacţie, definită prin procedura tranzactie primeşte informaţii despre un nou produs şi înserează în tabelele tip_produs, nume_producator, produse şi magazie în această ordine. Dacă apar erori pe parcursul tranzacţiei, se execută rollback. Pentru această tranzacţie se execută următoarele teste:
  - Se inserează un produs GPU produs de AMD ce are caracteristica Placa video, stocul de 300 şi preţul de 1000.
  - Se inserează din nou produsul anterior pentru a testa oprirea în cazul de tip de produs duplicat.
  - Se inserează produsul anterior cu tipul de produs schimbat, dar nume de producător identic pentru a se vedea oprirea la nume de producător duplicat.
  - Se inserează produsul cu numele de producător şi tipul schimbat, ceea ce se face cu succes.
  - Se actualizează elemente existente la noi valori şi se încearcă folosirea tranzacţiei pentru a insera aceste elemente deja existente.
  - Se execută o serie de ştergeri folosind procedurile din pachetele definite pentru a putea insera elementul anterior folosind tranzacţia.
- A doua tranzacţie, definită prin procedura tranzactie2, are ca rol inserarea unui produs în tabelul produse, apoi în tabelul magazie şi la sfârşit în tabelul vanzari care va afecta tabelul magazie. Pentru această tranzacţie se fac următoarele teste:
  - Se foloseşte tranzacţia pentru un produs obişnuit.

    - Se încearcă folosirea din nou a tranzacției pentru produsul anterior pentru a verifica unicitatea produsului.
    - Se face tranzacția, inserând un produs cu o cantitate mai mică în magazie decât se cere la vânzare pentru a testa constrângerea de stoc pozitiv a produsului din magazie.

Pentru SQL Injection, se definește o procedură vulnerabilă selectare_tip_produs_vulnerabil care selectează datele din tabelul tip_produs folosind o concatenare în cadrul instrucțiunii SELECT. Se testează procedura pentru un caz obișnuit, iar apoi este testată folosind un string care face ca rezultatul întors de procedură să ajungă în mod neașteptat pe tabelul stare_produs. Pentru a securiza această procedură, s-au folosit bind variables și a fost creată procedura selectare_tip_produs _securizat care a fost testată pe cazurile anterioare, dar de această dată nu s-a mai afișat conținut din tabelul stare_produs.

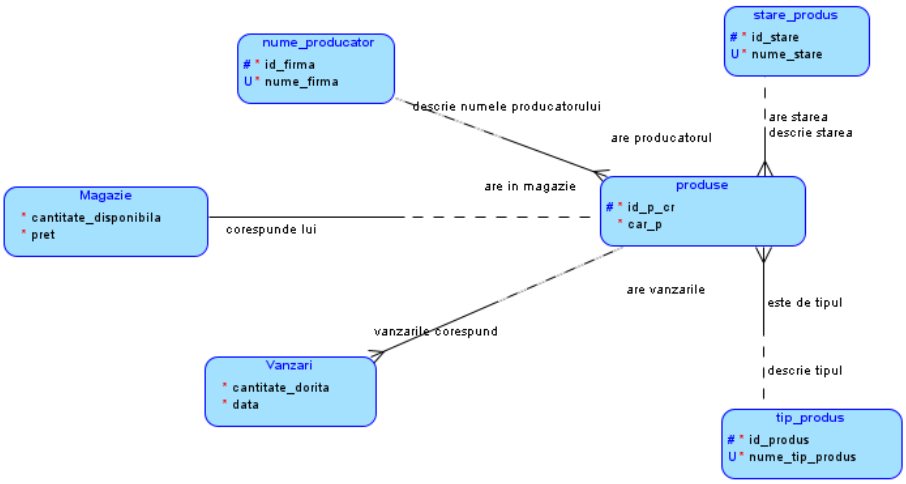
Structura și inter-relaționarea tabelelor

Baza de date este formată din 6 tabele, detaliate mai jos:

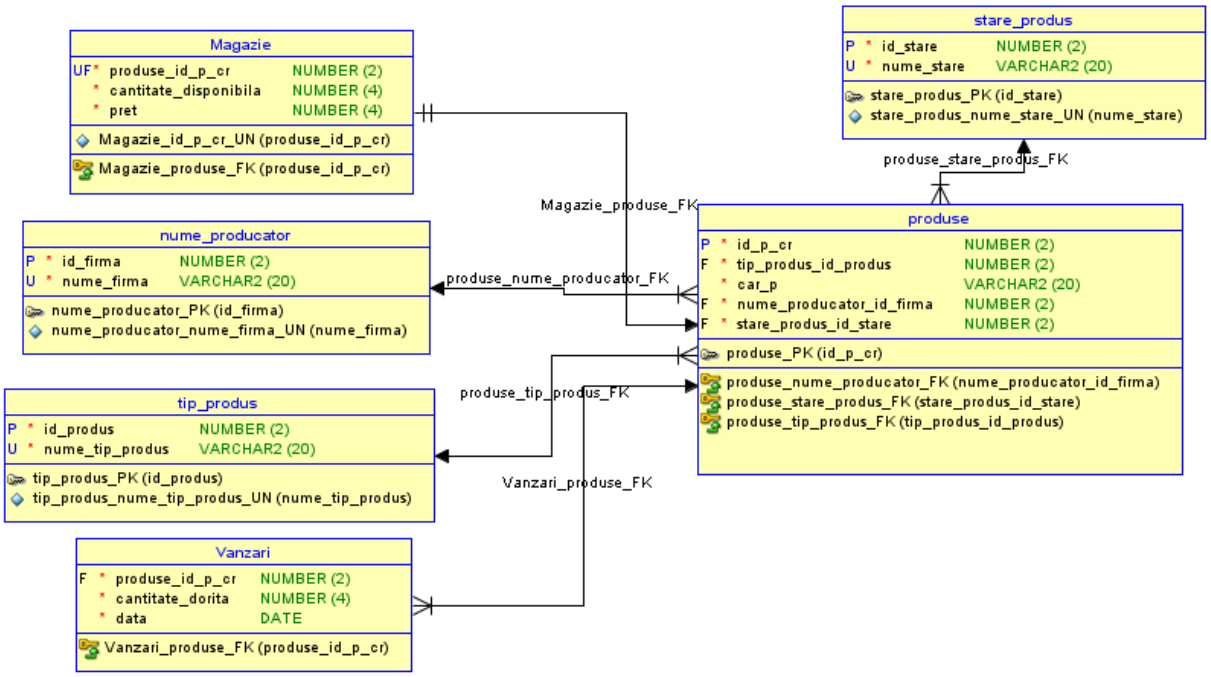
- Tabelul stare_produs conține informații despre starea produsului:
    - id_stare ce stochează id pentru starea curentă.
    - nume_stare care descrie numele stării produsului.
- Tabelul tip_produs conține informații despre tipul produsului:
    - id_produs de tip id pentru tipul curent de produs.
    - nume_tip_produs stochează numele tipului de produs.
- Tabelul nume_producator stochează informații despre producătorii produselor:
    - id_firma de tip id pentru producătorul curent.
    - nume_firma care stochează numele producătorului.
- Tabelul produse conține următoarele informații:
    - id_p_cr ce descrie id-ul produsului curent.
    - car_p ce descrie o caracteristică a produsului.
    - FK către tabelul stare_produs.
    - FK către tabelul tip_produs.
    - FK către tabelul nume_producator.
- Tabelul magazie conține:
    - FK către tabelul produse.
    - cantitate_disponibila ce conține stocul pentru produsul curent.
    - pret care conține prețul produsului.

- Tabelul vanzari conține următoarele informații despre vânzări:
  - FK către tabelul produse.
  - cantitate_dorita conține numărul de produse achiziționate.
  - data conține data la care s-a făcut vânzarea.

În figura următoare poate fi vizualizată diagrama modelului logic:



Pentru modelul relațional, diagrama ER este următoarea:



Descrierea logicii stocate

Logica PL/SQL este formată din mai multe pachete, triggere și proceduri, după cum urmează:

- Pachetul magazie_pkg ce conține funcții și proceduri aferente tabelei magazie.
- Pachetul nume_producator_pkg ce conține funcții și proceduri aferente tabelei nume_producator.
- Pachetul produse_pkg ce conține funcții și proceduri aferente tabelei produse.
- Pachetul stare_produs_pkg ce conține funcții și proceduri aferente tabelei stare_produs.
- Pachetul tip_produs_pkg ce conține funcții și proceduri aferente tabelei tip_produs.
- Pachetul vanzari_pkg ce conține funcții și proceduri aferente tabelei vanzari.
- Trigger-ul trg_produse_duplicate_insert verifică dacă la inserarea unui nou produs deja există acel produs, și dacă da, aruncă o excepție.
- Trigger-ul trg_produse_duplicate_update verifică dacă la actualizare în tabelul produse nu se ajunge la duplicarea unui alt produs deja existent.
- Trigger-ul trg_vanzari_insert verifică dacă mai întâi dacă nu există deja o vânzare cu datele curente și aruncă o excepție dacă există deja, altfel decrementează în tabelul magazie cantitatea dorită din vânzare, aruncând o excepție dacă se ajunge la o cantitate negativă în tabelul magazie.
- Trigger-ul trg_vanzari_update are o funcționalitate similară cu anteriorul, dar decrementează în tabelul magazie doar diferența dintre vechea cantitate dorită și noua cantitate dorită.
- Procedura tranzactie, descrisă anterior, ce execută inserarea de la zero a unui nou produs în magazie cu un nou tip, nou producător și starea Nou. Primește ca parametri de intrare numele noului tip de produs, numele noului producător, caracteristica produsului, cantitatea disponibilă în magazie și prețul produsului.
- Procedura tranzactie2, descrisă anterior inserează în tabelul produse un nou produs cu date deja existente pentru stare, tip și producător, după care inserează produsul în tabelul magazie și apoi execută o vânzare pentru acest produs. Are ca parametrii de intrare numele tipului de produs, numele firmei, numele stării, caracteristica produsului, cantitatea disponibilă în magazie, prețul produsului, cantitatea dorită la vânzare și data vânzării.
- Procedurile selectare_tip_produs_vulnerabil și selectare_tip_produs_securizat, ce au fost descrise anterior, au rolul de a afișa un rând din tabelul tip_produs unde numele tipului corespunde cu parametrul procedurii.

Pachetul magazie_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura insert_magazie de inserare în magazie primește ca parametrii numele tipului de produs, caracteristica produsului, numele producătorului, numele stării produsului, cantitatea disponibilă și prețul produsului. Va afla id-ul produsului din aceste date și va insera în tabelul magazie produsul rezultat.

- Procedura update_magazie va actualiza prețul și cantitatea disponibilă pentru un produs. Primește ca parametrii numele tipului de produs, caracteristica produsului, numele producătorului, numele stării produsului, cantitatea disponibilă și prețul produsului. Va afla id-ul produsului din aceste date și va actualiza cantitatea disponibilă și prețul la valorile date.
- Procedura delete_magazie are rolul de a șterge un produs din magazie. Primește ca parametrii numele tipului de produs, caracteristica produsului, numele producătorului și numele stării produsului. Va afla id-ul produsului din aceste date și va șterge din tabelul magazie produsul cu acest id.
- Funcția get_magazie întoarce un cursor către o înregistrare din tabelul magazie. Primește ca parametrii numele tipului de produs, caracteristica produsului, numele producătorului și numele stării produsului.
- Funcția get_all_magazie întoarce un cursor către tot tabelul magazie.

Pachetul nume_producator_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura inserare_producător care primește ca parametru un nume de producător și-l inserează în tabelul nume_producator.
- Procedura actualizare_producator care primește ca parametru un nume vechi de producător și un nume nou de producător și actualizează în tabelă numele vechi de producător la numele nou de producător.
- Procedura stergere_producator șterge din tabelă acel producător ce are numele dat ca parametru.
- Funcția selectare_producator întoarce un cursor către acea înregistrare care are numele producătorului dat ca parametru.
- Funcția selectare_toti_producatorii întoarce un cursor către tot tabelul nume_producator.

Pachetul produse_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura inserare_produs face inserarea unui produs în tabelul produse. Primește ca parametrii numele tipului de produs, caracteristica produsului, numele producătorului și numele stării.
- Procedura actualizare_produs actualizează datele unui produs. Primește ca parametrii două seturi de proprietăți care descriu produsul și actualizează în tabelul produse produsul indicat de vechile caracteristici.
- Procedura stergere_produs execută ștergerea produsului ale cărui caracteristici au fost date ca parametrii.
- Funcția selectare_produs întoarce un cursor către produsul ale cărui caracteristici sunt parametrii funcției.
- Funcția selectare_toate_produsele întoarce un cursor către toate produsele.

Pachetul stare_produs_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura inserare_stare care primește ca parametru un nume de stare și îl inserează în tabelul stare_produs.
- Procedura actualizare_stare primește ca parametrii un nume vechi de stare și un nume nou de stare și realizează actualizarea în tabelul stare_produs.
- Procedura stergere_stare va șterge starea al cărui nume a fost dat ca parametru.
- Funcția selectare_stare returnează un cursor către înregistrarea care are numele dat ca parametru.
- Funcția selectare_toate_starile returnează un cursor către tabelul stare_produs.

Pachetul tip_produs_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura inserare_tip_produs primește ca parametru numele tipului de produs de inserat și îl inserează în tabelul tip_produs.
- Procedura actualizare_tip_produs primește o pereche de nume de tip de produs, unul vechi și unul nou. Va actualiza în tabelul tip_produs valoarea vechiului nume la noul nume.
- Procedura stergere_tip_produs va șterge tipul de produs al cărui nume este dat ca parametru.
- Funcția selectare_tip_produs va returna un cursor către tipul de produs ce are numele dat ca parametru.
- Funcția selectare_toate_tipurile va returna un cursor către toate înregistrările tabelei tip_produs.

Pachetul vanzari_pkg este format din mai multe proceduri și funcții, după cum urmează:

- Procedura inserare_vanzare primește ca parametrii caracteristicile produsului, cantitatea dorită și data vânzării, după care extrage id-ul produsului și inserează în tabelul vânzări informațiile.
- Procedura actualizare_vanzare primește ca parametrii caracteristicile produsului, vechea cantitate dorită, vechea dată, noua cantitate, noua dată și actualizează în tabelul vânzări cantitatea veche la nouă și data veche la data nouă pentru produsul ale cărui caracteristici au fost date ca parametrii.
- Procedura stergere_vanzare primește ca parametrii caracteristicile produsului, cantitatea dorită și data, după care șterge din tabelul vanzari înregistrarea cu aceste date.
- Funcția selectare_vanzare primește ca parametrii caracteristicile produsului, cantitatea dorită și data, după care returnează un cursor către înregistrarea cu aceste date.
- Funcția selectare_toate_vanzarile returnează un cursor către toate înregistrările din tabelul vanzari.