

## Baze de date, IESC– Îndrumar de Laborator

### Capitolul 8 – Probleme Recapitulative

Fie schema logică din Fig. 1, ce reprezintă o variantă simplificată a bazei de date a unui magazin. O posibilă stare a celor trei tabele este prezentată în Fig. 2.

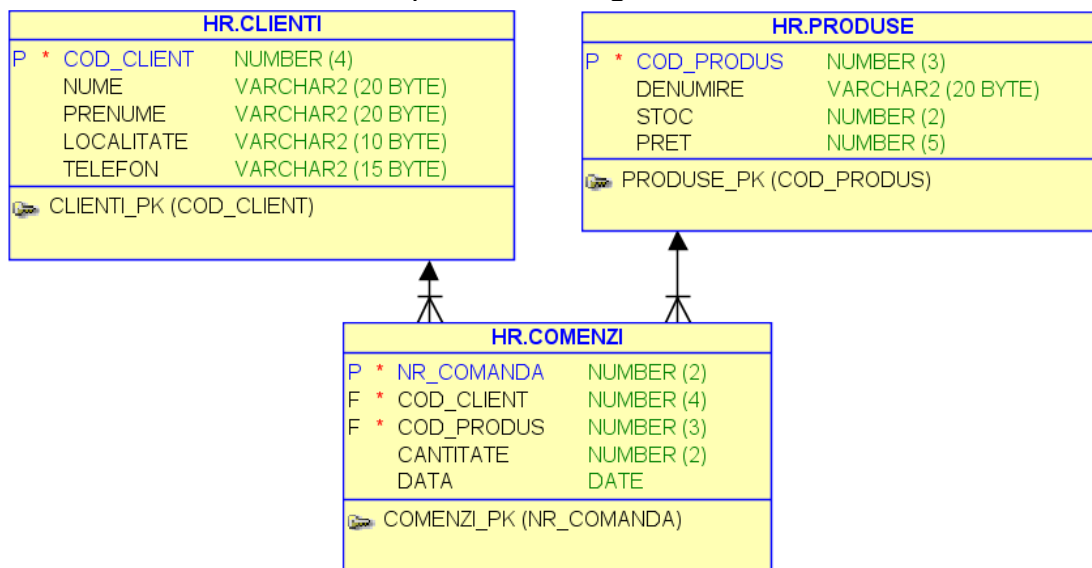


Fig. 1. Tabele asociate gestiunii unui magazin

COD_CLIENT	NUME	PRENUME	LOCALITATE	TELEFON	COD_PRODUS	DENUMIRE	STOC	PRET
1	Popescu	Ion	Iasi	1234	1	Mere Golden	20	5
2	Ionescu	Maria	Bucuresti	0014	2	Mere Ionatan	30	4
3	Petrescu	Veronica	Bucuresti	0234	3	Pere	20	7

NR_COMANDA	COD_CLIENT	COD_PRODUS	CANTITATE	DATA
1	1	2	3	26-DEC-12
2	1	1	2	26-DEC-12
3	3	3	3	27-DEC-12

Fig. 2. Exemplu de stare a celor trei tabele

1. Creați cele trei tabele, inclusiv cheile primare și străine care apar în Fig. 1.
2. Introduceți câteva linii în tabele (puteți personaliza tabelele altfel decât exemplul din Fig. 2).
3. Scrieți comenzile SQL necesare interogării: afișați denumirea și cantitatea tuturor produselor comandate de clientul cu numele Popescu. Puteți folosi un alias de tabelă (de exemplu: A, B etc.) pentru a impune condițiile de joncțiune.  
Pe exemplul din Fig. 2 ar trebui să obțineți următorul rezultat:
- 4) Afișați într-o tabelă numele, prenumele, denumirea și prețul total pentru o comandă, pentru toți clienții. Ordonați rezultatul crescător după nume, apoi după prenume. Exemplu de rezultat:

NUME	DENUMIRE	CANTITATE
Popescu	Mere Ionatan	3
Popescu	Mere Golden	2

NUME	PRENUME	DENUMIRE	PRET_TOTAL
Petrescu	Veronica	Pere	21
Popescu	Ion	Mere Ionatan	12
Popescu	Ion	Mere Golden	10

- 5) Modificați interogarea anterioară pentru a afișa totalul comenzilor plasate după o anumită dată.
- 6) Adăugați o coloană cu numele „Puncte”, de tip număr întreg, în tabela clienți. Realizați un program PL/SQL care să acorde un număr de puncte fiecărui client din tabela Clienți, după cum urmează: câte 10 puncte pentru fiecare 10 lei cheltuiți (de exemplu 30 de puncte dacă clientul a cheltuit între 30 și 39 de lei). Puteți utiliza pentru calculul punctelor instrucțiunea următoare:  $nr\_puncte := 10 * floor(suma/10);$

Pe exemplul de mai sus, starea tabeli Clienți după rularea acestui program PL/SQL ar trebui să fie următoarea:

COD_CLIENT	NUME	PRENUME	LOCALITATE	TELEFON	PUNCTE
1	Popescu	Ion	Iasi	1234	20
2	Ionescu	Maria	Bucuresti	0014	(null)
3	Petrescu	Veronica	Bucuresti	0234	20

7) Modificați programul de la punctul anterior astfel încât să acordați 5 puncte suplimentare fiecărui client care a comandat (o dată sau de mai multe ori) un anumit produs (de exemplu pere). Utilizați un cursor (preferabil implicit) pentru a afla denumirea produselor comandate de fiecare client în parte.

8) Creați un declanșator (trigger) cu numele „comenzi\_insert”, de tip AFTER, pentru fiecare linie, care să aibă următorul efect: la inserarea unei noi linii în tabela Comenzi, să diminueze stocul produsului comandat cu cantitatea comandată.

9) Realizați un trigger cu numele „verifica\_insert”, de tip BEFORE, care înainte de inserarea unei linii în tabela Produse, să verifice dacă denumirea produsului există deja în tabelă, iar dacă există, inserarea să nu fie efectuată. Notă: pentru a împiedica inserarea noii linii se poate utiliza funcția: `raise_application_error(-20000,'Produsul exista deja');`

10) Realizați o funcție stocată cu prototipul *total\_client(cod NUMBER)* care, primind un cod client să returneze costul total al comenzilor plasate de acel client.

La sfârșit eliberați spațiul de lucru rulând următoarele comenzi:

```
DROP TABLE Comenzi;  
DROP TABLE Produse;  
DROP TABLE Clienți;  
DROP TRIGGER comenzi_insert;  
DROP TRIGGER verifica_insert;  
DROP FUNCTION total_client;
```