

SCHEMA BAZEI DE DATE VÎNZĂRI

Pentru baza de date VÎNZĂRI, reprezentarea grafică a schemei simplificate este cea din figura A.1.

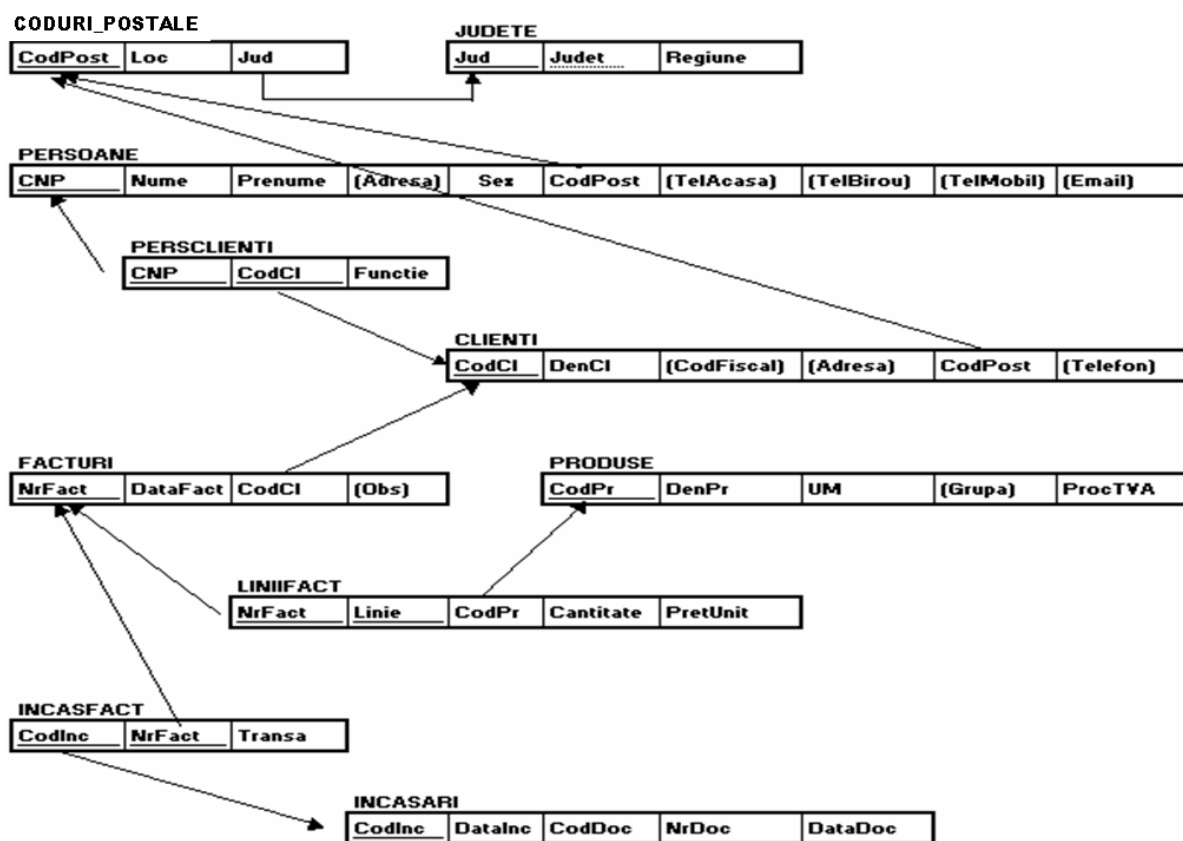


Figura A.1.(și 2.5). Schema simplificată a bazei de date VÎNZĂRI

Tabela **JUDETE** conține informații generale despre județele în care sunt clienți. Fiecare linie a tabelului descrie un județ. Atributele sunt:

Jud – indicativul auto al județului (alcătuit din două litere, majuscule);

Judet – denumirea județului;

Regiune – regiunea istorică (provincia) din care face parte județul.

Cheia primară este atributul Jud. Atributul Judet este cheie alternativă. Pot fi instituite o serie de restricții-utilizator:

Jud este alcătuit numai din majuscule (eventual, un spațiu, pentru București);
Fiecare cuvânt din **Judet** începe cu majusculă; restul literelor sunt mici;
Regiune începe cu majusculă; restul literelor sunt mici;
Regiune poate avea numai una din valorile: Banat, Dobrogea, Muntenia, Oltenia, Transilvania, Moldova.

Tabela **CODURI_POSTALE** conține câte o linie pentru fiecare codpostal dintr-un oraș sau comună.

CodPost;

Loc – denumirea comunei/orașului;

Jud – indicativul auto al județului.

Cheia primară este atributul **CodPost**. Atributul **Jud** este cheie străină, tabela părinte fiind **JUDETE** (prin atributul **Jud**). Pot fi instituite o serie de restricții-utilizator:

Jud este alcătuit numai din majuscule (eventual, un spațiu, pentru București)

Fiecare cuvânt din **Localitate** începe cu majusculă; restul literelor sunt mici

Tabela **CLIENTI** grupează date generale ale clienților firmei pentru care s-a constituit baze de date (o linie – un client). Atribute:

CodCl – codul clientului;

DenCl – denumirea clientului (persoană juridică);

CodFiscal – codul fiscal;

Adresa – adresa sediului firmei client;

CodPost – codul poștal al comunei sau orașului;

Telefon – telefonul (principal) al clientului.

Cheia primară este atributul **CodCl**. Atributul **CodPost** este cheie străină, tabela părinte fiind **LOCALITATI** (prin atributul **CodPost**). **DenCl** și **Adresa** încep cu majuscule. După cum punctam anterior, valorile nule indică o lipsă de informație. Pentru primul client nu se cunoaște telefonul, celui de-al doilea adresa etc.

Tabela **PERSOANE** are ca obiectiv stocarea datelor despre persoanele cheie de la firmele client: directori generali, directori financiari, șefi ai compartimentelor comerciale (aprovizionare și/sau vânzări) etc. Probabil că vi se pare ciudat, dar ceea ce intenționăm cu această tabelă (și următoarea) este să sprijinim *fidelizarea* clientului. Nu costă (mai) nimic dacă de Sfântul Ion se trimite câte o felicitare tuturor Ionilor și Ioanelor din partea firmei noastre. Iar pentru ca lucrurile să fie și mai bine puse la punct, ar fi trebuit să preluăm și informații precum: data nașterii, starea civilă, numele și vârsta copiilor, pasiuni în materie de muzică, arte pastice, literatură, sport etc. Schema luată în considerare s-a oprit la următoarele atribute:

CNP – codul numeric al persoanei;

Nume;

Prenume;

Adresa;

Sex;

CodPost;

TelAcasă – numărul telefonului de acasă;

TelBirou – numărul telefonului (fix) de la birou;

TelMobil – numărul “mobilului”;

Email – adresa e-mail.

Deși este un număr compus din 13 poziții, este recomandabil ca CNP să fie declarat de tip șir de caractere. Aceasta deoarece în firmă există rezidenți din Republica Moldova care au poziții de vârf în firme românești (sau reprezentanțe ale unor firme din țara natală). Or, codul identificator din pașaport este alcătuit dintr-o literă și o serie de cifre.

Cheia primară este atributul CNP. Fiind vorba de persoane exterioare întreprinderii, un atribut precum Marca este mai puțin indicat. CNPul este greoi, relativ lung (de aceea am preferat să scriem, pur și simplu CNP1, CNP2...), dar stabil și unic.

Atributul CodPost este cheie străină, tabela părinte fiind LOCALITATI (prin atributul CodPost).

Celei de-a doua persoane nu i se cunoaște domiciliul (adresa). Cu excepția lui Vasile Ion, nimeni nu are cont deschis pentru poșta electronică.

Alte restricții utilizator:

Fiecare cuvânt din Nume și Prenume începe cu majusculă; restul literelor sunt mici;

Adresa începe cu majusculă;

Atributul Sex poate lua numai două valori: *B* de la *Bărbătesc* și *F* de la *Femeiesc*.

Tabela **PERSCLIENTI** indică funcția deținută de fiecare persoană la unul (sau mai mulți, deși aceste cazuri sunt rare) clienți. O linie din această tabelă reflectă o funcție deținută de o persoană la un client. O persoană poate avea mai multe funcții, chiar la aceeași firmă (cumul de funcții). Cele trei atribute sunt:

CNP – persoana care deține funcția;

CodCl – firma la care persoana deține funcția;

Funcție – funcția deținută.

Cheia primară este compusă din toate cele trei atribute: CNP+CodCl+Funcție, aceasta deoarece am convenit că o aceeași persoană poate cumula două sau mai multe funcții la aceeași firmă. CNP și CodCl sunt chei străine, tabelele părinte fiind PERSOANE, respectiv CLIENTI. Atributul Funcție începe cu majusculă, restul literelor fiind mici.

Tabela **PRODUSE** reprezintă nomenclatorul de produse și servicii pe care le comercializează firma (produsele sunt obținute prin manufacturare sau revânzare). Atribute:

CodPr – codul produsului;

DenPr – denumirea;

UM – unitatea de măsură a produsului;

Grupa – grupa de mărfuri (sortimente) în care se încadrează; acest atribut este important pentru analiza vânzărilor;

ProcTVA – procentul TVA care se aplică la prețul de vânzare (preț care este fără TVA); pare de prisos în condițiile actuale, când toate produsele au un singur procent, 19%, dar nimeni nu poate garanta cum vor gândi promoțiile viitoare de guvernanți “relaxarea fiscală”;

Cheia primară este atributul **CodPr**. Alte restricții utilizator:

DenPr și **Grupa** încep cu majusculă; restul literelor sunt mici;

unitatea de măsură (**UM**) se scrie numai cu litere mici.

Tabela pare una “inofensivă”, dar de modul său de organizare depinde rezolvarea unor probleme deosebit de sensibile. Este foarte important de stabilit regimul în care se va lucra cu produsele care se comercializează sub mai multe forme. După cum se observă, atributul **DenPr** nu a fost declarat cheie alternativă. De ce ? Să luăm un exemplu absolut la întâmplare: o firmă de comerț en-gross vinde (legal), printre altele, produsul *Vodcă Scandic* și la cutii de 1 litru și la cutii de 250 ml.

Deși este același produs (din prea-numeroasele mele încercări, gustul este același (sau, cel puțin, așa îmi amintesc), indiferent de tipul cutiei), pentru o evidență corectă a vânzărilor, este necesar ca în tabela **PRODUSE** să existe două linii afectate vodcii Scandic, fie ca în figura 1.19, fie ca în figura 1.20.

67	Vodca Scandic - cutie 1 l	buc	Spirtoase	0.22
68	Vodca Scandic - cutie 200 ml	buc	Spirtoase	0.22

Figura 1.19. Specificarea, în **DenPr**, a tipului cutiei

67	Vodca Scandic	cutie 1 l	Spirtoase	0.22
68	Vodca Scandic	cutie 200 ml	Spirtoase	0.22

Figura 1.20. Diferențierea (în afară de cod) numai prin **UM**

Este dificil de spus care variantă este mai bună. Eu, unul, aș înclina pentru cea de-a doua, caz în care atributul **DenPr** nu este cheie candidată.

Tabela **FACTURI** conține câte o linie pentru fiecare factură emisă, factură ce reflectă o vânzare (către un client). Atribute:

NrFact – numărul facturii;

DataFact – data întocmirii facturii;

CodCl – codul clientului căruia i s-au vândut produsele/serviciile consemnate în factură;

Obs – observații; e folosit relativ rar, pentru a introduce eventuale detalii sau probleme care au apărut în legătură cu o factură.

Cheia primară este atributul **NrFact**. Explicația este una simplă: gestionarea facturilor emise este, conform legii, strictă. Nu pot exista două facturi (care reflectă două vânzări) cu un același număr. Sau, de fapt, pot exista, dar aceasta înseamnă o ilegalitate destul de gravă (se mai poartă încă prin economia subterană). **CodCl** este cheie străină (tabela părinte este **CLIENTI**). Ca restricție utilizator suplimentară, nu pot fi introduse facturi întocmite înainte de 1 august 2005 (**DataFact** >= DATE'2005-08-01').

Tabela **LIINFAC** detaliază tabela precedentă. Un tuplu se referă la un produs/serviciu vândut și consemnat într-o factură emisă. Pentru fiecare factură vor fi atâtea linii câte produse/servicii au fost consemnate la vânzarea respectivă. Atribute:

NrFact – numărul facturii;

Linie – numărul liniei din factura respectivă;

CodPr – codul produsului/serviciului vândut;

Cantitate – cantitatea vândută;

PretUnit – prețul unitar (fără TVA) la care s-a făcut vânzarea.

Cheia primară este combinația **NrFact+Linie**. **NrFact** și **CodPr** sunt chei străine. Pentru a determina valoarea de încasat a unei facturi (inclusiv TVA), la valoarea fără TVA pentru fiecare linie (obținută prin produsul **Cantitate * PretUnit**) trebuie adăugată TVA colectată, obținută prin aplicarea procentului de TVA al produsului/serviciului (atributul **ProcTVA** din **PRODUSE**) la valoarea fără TVA.

Tabela **INCASARI** reprezintă un nomenclator al încasărilor. Printr-o încasare, un client își stinge una sau mai multe obligații de plată, adică achită una sau mai multe facturi. Documentul primar pe baza căruia se consemnează încasarea poate fi ordinul de plată, cecul, chitanța etc. Atributele tabelului sunt:

CodInc – codul încasării este un număr intern, util pentru a diferenția o încasare de celelalte;

DataInc – data încasării - data la care banii au intrat în contul sau casieria firmei;

CodDoc – codul documentului justificativ al încasării: OP – ordin de plată, CHIT – chitanță, CEC – filă cec;

NrDoc – numărul documentului justificativ

DataDoc – data la care a fost întocmit documentul justificativ; din momentul întocmirii documentului de plată până la data la care banii ajung efectiv în cont/casierie trec câteva zile sau săptămâni (datorită circuitului documentelor între firme și bănci).

Cheia primară este atributul **CodInc**. Ca restricții utilizator pot fi instituite:

data încasării nu o poate precede pe cea a întocmirii documentului (**DataDoc** <= **DataInc**);

data documentului de încasare nu poate fi anterioară datei de început a aplicației: 1 august 2005 (**DataDoc** >= DATE'2005-08-01');

codul documentului se scrie numai cu majuscule.

Tabela **INCASFACT** detaliază tabela precedentă și indică ce facturi (tranșe din facturi) sunt achitate prin fiecare încasare. Un client poate plăti mai multe facturi odată (printr-o singură plată). Pe de altă parte, orice factură poate fi plătită în una sau mai multe tranșe, în funcție de banii de care dispune clientul la momentul întocmirii documentului de plată. Atributele tabelului sunt:

CodInc – codul încasării;

NrFact – factura pentru care se încasează valoarea integrală sau numai o tranșă;

Transa – tranșa din factură (sau întreaga valoare) care se încasează prin documentul primar ce stă la baza încasării.

Cheia primară este combinația **CodInc+NrFact**, deoarece la o încasare se pot achita mai multe facturi, iar, pe de altă parte, o factură poate fi plătită în mai multe tranșe. **CodInc** și **NrFact** sunt chei străine.

Comentarii suplimentare privind restricțiile referențiale

Instituirea unei restricții referențiale într-o bază de date interzice apariția de valori nenule în tabele copil care nu se regăsesc în tabelele părinte (linii orfane). De aceea, la inserarea unei linii într-o tabelă copil sau modificarea unei chei străine, SGBD-ul trebuie verifice dacă noile valori se regăsesc în linii părinte.

Inserarea unei linii într-o tabelă părinte, ca și ștergerea de linii dintr-o tabelă copil nu prezintă nici un pericol “referențial”.

La ștergerea unei linii dintr-o tabelă părinte trebuie precizat cum se va prezerva restricția referențială: fie prin ștergerea în cascadă a tuturor înregistrărilor copil (ON DELETE CASCADE), fie, pur și simplu, prin interzicerea ștergerii (ON DELETE RESTRICT).

În fine, la modificarea unei chei primare/alternative pentru care există, în alte tabele, înregistrări-copil, trebuie precizată acțiunea întreprinsă de SGBD: modificarea în cascadă a tuturor liniilor copil (ON UPDATE CASCADE) sau interzicerea modificării dacă există cel puțin o înregistrare copil (ON UPDATE RESTRICT).

Pentru exemplificare, să luăm cazul tabelii **FACTURI** și trei situații legate de actualizarea acesteia.

Inserarea unei linii. Deoarece **FACTURI** este legată (ca tabelă copil) prin restricție referențială de tabela **CLIENTI** (părinte), trebuie verificat dacă valoarea **CodCl** există în **CLIENTI**. Dacă nu, inserarea trebuie prohibită.

Modificarea unei linii. Aici sunt două situații, corespunzătoare posturilor de părinte și copil a tabelii modificate. Astfel:

dacă se modifică valoarea atributului **CodCl**, trebuie verificat dacă aceasta se regăsește în **CLIENTI** și, dacă nu, operațiunea trebuie anulată;

dacă se modifică valoarea lui **NrFact**, atunci trebuie testat dacă există înregistrări copil în tabelele **LINIIFACT** și **INCASFACT**. Dacă da,

fie se interzice modificarea (ON UPDATE RESTRICT),

fie se modifică în cascadă toate liniile copil - valorile **NrFact** din **LINIIFACT** și **INCASFACT** (ON UPDATE CASCADE).

Firește, nu e obligatoriu ca acțiunea să fie identică pentru ambele tabele copil. Cu alte cuvinte, se poate recurge, pentru **LINIIFACT** la ON UPDATE CASCADE, iar pentru **INCASFACT** la ON UPDATE RESTRICT.

Ștergerea unei linii. atunci trebuie testat dacă există înregistrări copil în tabelele **LINIIFACT** și **INCASFACT**. Dacă da,

fie se interzice ștergerea (ON DELETE RESTRICT),

fie se șterg în cascadă toate liniile copil din **LINIIFACT** și **INCASFACT** (ON DELETE CASCADE).

Există diferențe semnificative între SGBD-uri în privința definirii acțiunilor ce trebuie întreprinse pentru respectarea integrităților referențiale. Unele produse, firește, dintre cele mai bine cotate, permit, la declararea restricțiilor referențiale (este de obicei, simultană cu crearea tabelor), instituirea oricărei reguli din cele patru: ON UPDATE CASCADE, ON UPDATE RESTRICT, ON DELETE CASCADE, ON DELETE RESTRICT.

Altele, precum Oracle, permit declararea acțiunilor numai pentru ștergere, ON DELETE CASCADE și ON DELETE RESTRICT. Pentru modificarea în cascadă sunt necesare declanșatoare speciale construite în limbajul de dezvoltare – PL/SQL.