

Gestiunea unei Firme de produse IT

Aplicația realizează gestiunea unei firme de produse IT prin intermediul programului Oracle. Am ales această temă întrucât am dorit să îmi combin pasiunea pentru calculatoare cu abilitatile și cunoștințele informatice.

Ex.1) Modelul de date ales va gestiona informații legate de o firmă ce se ocupa cu asamblarea și vânzarea unor produse IT precum : monitoare, PC-uri și laptop-uri. Pentru a înțelege cu exactitate principiul din spatele acestui proiect, putem compara această firmă cu compania HP(*Hewlett-Packard*), care produce obiectele enumerate mai sus folosindu-se de componente de la un producător extern (eg: Intel Core, nVidia, etc..). Firma creată de noi are scopuri asemănătoare cu exemplul dat, diferența fiind făcută de faptul că noi noi ne propunem să creăm produse la comandă, însemnând că un produs o să fie predestinat unui cumpărător cunoscut.

Astfel, firma noastră se ocupa cu crearea / asamblarea unor produse IT la cererea clienților. Firma nu își produce singura componentele folosite în aceste produse , acestea sunt distribuite de către furnizori de la anumiți producători externi ce se ocupa în special de crearea unui anumit tip de componentă(eg: procesor grafic, microprocesor, placa de bază, display).

Next step, asamblarea sau crearea propriu-zisă a produselor pe care firma urmează să le vândă clienților. Această operațiune este realizată de salariații firmei. Fiecare salariat are aptitudini specifice, care îl fac să fie persoana ideală pentru o anumită echipă. Echipa primește unul sau mai multe task-uri din procesul de creare al unui produs. Task-urile sunt rezolvate pe rând de către întreaga echipă. Pe lângă aceste aptitudini, sunt reținute și informații de bază despre un angajat, cum ar fi adresa, numele complet, CNP-ul și alte date de contact. De asemenea, ca cineva să se poată angaja la firma noastră, trebuie să aibă o dovadă a acestor aptitudini (diploma care atestă completarea unor studii superioare în acest domeniu, scrisoare de recomandare de la un alt loc de muncă similar, proiect individual în domeniu (toate acestea vor fi trecute sub forma unui string în tabelă: descriere_specializare - > dovadă; eg: diploma de absolvire – FMI: Universitatea din București, Recomandare – fost angajat la ..)).

Pretul unui produs este calculat prin dublarea si insumarea preturilor componentelor.

Pentru o buna functionalitate a personalului firmei, angajatii sunt impartiti in echipe ce au ca scop rezolvarea unor taskuri din procesul de creare al unui produs. O echipa poate avea mai multe taskuri, rezolvarea lor facandu-se treptat, cate unul pe rand. Echipele au propriile lor sedii, fiind impartite pe mai multe regiuni sau tari. Cu toate astea , pot exista doua echipe cu sediul in aceeasi tara sau regiune. In functie de unde are sediul echipa, angajatul poate fii platit cu moneda respectiva (eg: o echipa cu sediul in America are angajati platiti in \$). Produsele firmei sunt indexate printr-o valoare numerica unica numita cod_fabricatie astfel incat, indiferent de locatia in care se afla echipele , nu o sa existe doua produse care impart acelasi id (spre exemplu, echipa x va produce produsul cu id-ul y de tip monitor. O alta echipa , z, nu o sa poata sa produca acelasi produs cu acelasi id, dar va putea sa produca un produs cu id diferit de tip monitor) . Un angajat poate sa demisioneze dintr-o anumita echipa si sa intre in alta , daca aptitudinile lui ii permit acest lucru. In situatii ca acestea, datele respectivului angajat trebuie salvate in arhiva, alaturi de data la care a inceput sa lucreze pentru echipa respectiva, cat si data la care a parasit acea echipa.

Angajatii au cate un superior numit lider. Liderul unui angajat trebuie sa fie in aceeasi echipa cu acesta deoarece el este lider de echipa.

Clientii pot achizitiona produsele finale pe care firma le vinde prin plasarea unor comenzi telefonic sau pe website-ul firmei. In cadrul unei comenzi , clientul trebuie sa specifice cantitatea (numarul de produse de tipul x), dar si tipul de livrare :

premium– firma se ocupa personal de livrarea produsului

standard – sunt folosite mijloace normale de livrare a produselor

In functie de tipul livrarii, produsele pot ajunge mai repede.

La plasarea unei comenzi sunt retinute informatii de baza despre un client, cum ar fi numele complet, adresa unde el o sa primeasca comanda si cateva date de contact cum ar fi nr. de telefon si email.

Ex.2) Modelul de date respectă anumite restricții de funcționare:

- Un salariat poate lucra doar intr-o singura echipa, iar o echipa este alcatuita din cel putin un salariat
- O echipa este responsabila pentru asamblarea / crearea oricarui produs din cele 3 tipuri vandute de firma. Echipa poate produce mai multe produse. Un produs cu un id x va fi produs doar de o singura echipa.
- Echipele au un lider. Fiecare salariat din echipa are acelasi lider care provine tot din aceeasi echipa.
- Un task (eg: testare produs) poate sa fie atribuit mai multor echipe. O echipa o sa aibe cel putin un task. Se poate ca o echipa sa aibe mai multe taskuri, dar rezolvarea lor este treptata, cate unul pe rand, la o anumita data.
- Angajatul nu poate asambla de unul singur un anumit produs. Acesta trebuie sa faca parte dintr-o echipa careia ii sunt atribuite task-uri esentiale in realizarea corecta a produsului.
- Un angajat poate sa apara pe mai multe file din arhiva, in functie de cate echipe a schimbat si daca inca este angajat la firma noastra. O fila din istoric ii corespunde doar unui singur angajat. Acelasi lucru se aplica si pentru echipe: o echipa poate sa apara pe mai multe file din arhiva, iar o fila din arhiva ii poate corespunde doar unei echipe.
- Componentele intra in alcatuirea produselor. O componenta(eg: procesor grafic) poate sa fie folosita la mai multe tipuri de produse. De asemenea, crearea unui produs trebuie sa folosim macar o componente(nu putem crea produsul altfel)
- Furnizorii se ocupa cu distribuirea componentelor. Fiecare furnizor are setul sau de componente pe care le distribuie. O componenta nu e neaparat distribuita de un furnizor cunoscut (acestea sunt componentele de pe piata neagra).
- In asamblarea produselor, se pot folosi componente comune (eg: procesorul, e comun pentru toate tipurile de produse), sau ceva specific tipului respectiv(eg: desktop_pc o sa aibe power supply).
- Salariatii pot avea mai multe specializari, iar o specializare se poate regasi la mai multi angajati. Mai mult, fiecare angajat are setul sau de “dovezi” care atesta specializarea sa (prezent in tabela dosar) care il fac sa fie potrivit pentru anumite echipe.(eg: Sa zicem ca angajatul x e specializat in programare -> are un dosar cu numele lui care atesta specializarea respectiva printr-un set de documente (diplome / scrisoare de recomandare); si e specializat si in inginerie - > alt dosar cu numele lui si alte documente.)

- Fiecarei echipe ii corespunde un sediu, iar unui sediu ii corespunde o echipa. Doua echipe nu pot imparti acelasi sediu.

Ex.3) Descrierea entitatilor:

In diagrama E/R avem uramtoarele entitati:

- SALARIAT
- ECHIPA
- PRODUS
- COMPONENTA
- SPECIALIZARE
- TASK
- CLIENT
- FILE_ARHIVA
- SEDIU

Dupa aceea , in diagrama conceptuala mai apare alte 4 noi tabele asociative:

- ATASAT_LA (sparge relatia many to many dintre ECHIPA si TASK)
- DETINE (sparge relatia many to many dinte SALARIAT si SPECIALIZARE)
- COMANDA (sparge relatia many to many dintre CLIENT si PRODUS)
- FOLOSESTE (sparge relatia many to many dintre PRODUS si COMPONENTA)

Astfel, in total vor fii 13 tabele.

Vom prezenta entitățile care apar in diagrama entitate relatie, dând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

SALARIAT = persoana fizica care lucreaza in cadrul firmei. Acesta are un set de aptitudini / specializari care ii permit sa lucreze in cadrul unei echipe din aceasta companie. In functie de activitatea sa in companie, angajatul poate sa aibe si un istoric. Cheia primara a acestei entitati este cnp_salariat (cnp-ul angajatului).

ECHIPA = un grup format din cel putin un salariat al firmei. Acest grup dispune de o locatie proprie in care membrii se pot intalni si pot lucra pentru crearea unui

produs. Echipa primește taskuri care sunt distribuite angajaților competenți. Cheia primară este id echipa.

TASK = o activitate care trebuie întreprinsă de către o întreagă echipă. Aceste activități sunt esențiale în crearea unui produs. Cheia primară este : id task

SEDIUL= Aici sunt reținute informații despre locația în care membrii echipei au facilitățile necesare pentru crearea unui produs. Două echipe nu pot avea același sediu (același id_sediu) dar pot avea sediile apropiate (în același oraș sau regiune, etc..). Cheia primară este: id sediu.

PRODUS = Firma noastră are ca scop asamblarea / crearea unor produse IT (desktop PC, laptop-uri și monitoare) la cererea unor clienți. Aceste produse nu sunt create de la 0 de către firmă, ele sunt asamblate cu ajutorul unor componente de la producători externi. Asamblarea lor este realizată de echipele companiei în sediul fiecăreia. Pot exista mai multe produse de același tip, dar nu pot avea și același cod (codul este o combinație de caractere și cifre, distinctă de la un produs la altul).Prețul produsului este calculat în funcție de prețul componentelor (suma dublului prețurilor componentelor).Cheia primară este cod fabricatie.

COMPONENTA = Componenta reprezintă o bucată mică, dar importantă, care intră în alcatuirea unui produs final al firmei. Aceste componente pot fi procesoare grafice (GPU), microprocesoare, display-uri, SSD-uri, power supply, orice intră în alcatuirea unui laptop, monitor sau desktop PC. Componentele se disting una de alta prin seria lor de fabricație. Cheia primară este serie.

CLIENT = Clientul este o persoană care a făcut o comandă către firma noastră. În funcție de solicitările clientului, firma îi va furniza produsele dorite. Clientul poate să își aleagă modul în care îi sunt furnizate produsele. Cheia primară este id client

FILE_ARHIVA = Se țin minte informații despre un angajat care nu a părăsit compania, dar și-a schimbat echipa în care lucrează , implicit și locația. Aici se salvează date referitoare la vechimea sa . Cheia primară este: id fila .

SPECIALIZARE = Aici este precizată aptitudinea definitorie a unui angajat. Acesta poate să fie un programator desăvârșit, un tester bun, un inginer ,etc..

De asemenea, angajatul trebuie sa aibe o dovada a acestor aptitudini. Cheia primara este: id_specializare .

Ex.4) Relații:

SALARIAT_lucreaza_in_ECHIPA = relație care leagă entitățile SALARIAT și ECHIPA, reflectând legătura dintre acestea (in ce echipa lucreaza un salariat). Ea are cardinalitatea minimă 1:1 (un salariat lucreaza într-o singura echipa, iar o echipa trebuie sa aibe cel putin un salariat) și cardinalitatea maximă n:1 (intr-o echipa pot lucra mai multi salariați, iar un salariat lucreaza doar într-o singura echipa).

SALARIAT_conduce_ECHIPA = aceasta relatie este mai speciala intrucat in entitatea ECHIPA exista atributul “lider” care face referire la o valoare a cheii primare din SALARIAT. O echipa poate sa nuaibe lider sau sa aibe un singur lider, unic, prin urmare fiecare salariat din echipa va fii condus de acel lider (un fel de self join, e explicat in paragraful urmator). Cardinalitatea minima a acestie relatii este 1:0 (un angajat poate sa nu conduca nicio echipa, o echipa e condusa de un angajat(e posibil si ca toti membrii echipei sa colaboreze in mod egal)) si cardinalitate maxima 1:1 (un angajat conduce maxim o echipa si o echipa e condusa maxim de un angajat).

SALARIAT_conduce_SALARIAT = cum am precizat si mai sus, echipa are un lider care o conduce, prin urmare , fiecare memebbru al echipei e condus de lider. Dar membrii echipei sunt salariați, iar liderul este tot un salariat. Astfel a trebuit sa facem aceasta relatie intre aceeasi entitate. Cardinalitatea minima este 0:0 (un angajat nu e neaparat obligat sa conduca un alt angajat, iar un angajat poate sa nu fie condus de nici-un angajat (daca el e deja liderul)) si maxima 1:m(un lider poate conduce mai multi angajat, iar un angajat e condus de maxim un singur lider).

SALARIAT_are_FILE_ARHIVA = relație care leagă entitățile SALARIAT și FILE_ARHIVA, reflectând legătura dintre acestea (Ce fila din arhiva ii corespunde unui salariat). Ea are cardinalitatea minimă 1:0 (un salariat poate sa nu aibe inca o fila in arhiva, iar o fila din arhiva are informatii despre un singur salariat) și

cardinalitatea maximă 1:n (un angajat poate avea mai multe file in arhiva, iar o fila din arhiva are informatii despre un singur salariat).

ECHIPA_apare_in_FILE_ARHIVA = relație care leagă entitățile ECHIPA și FILE_ARHIVA, reflectând legătura dintre acestea (In ce fila din arhiva unui salariat este trecuta si echipa x). Aceasta relatie se realizeaza pe aceleasi principii si cu aceleasi cardinalitati ca in relatia SALARIAT_are_FILE_ARHIVA .

Scurta explicatie: Acum, diagrama noastra ar parea ca formeaza o bucla redundanta intre Salariat, File_Arhiva si Echipa. Cu toate acestea, nu putem elimina legatura dintre Salariat – Echipa deoarece nu putem ajunge de fiecare data la tabela Echipa plecand din Salariat si trecand prin File_Arhiva din cauza cardinalitatii (e posibil ca angajatul sa nu aibe inca o fila in arhiva, iar asta ne-ar impiedica sa ajungem dupa sa ii stim Echipa trecand prin File_Arhive pentru ca nu exista acea fila. De aceea e mai bine sa pastram o legatura directa intre salariat si echipa.)

ECHIPA_are_SEDIU = relație care leagă entitățile ECHIPA si SEDIU, reflectând legătura dintre acestea (unde este sediul unei echipe). Ea are cardinalitatea minimă cat si maxima 1:1 (o echipa are un singur sediu, iar un sediu ii apartine unei singure echipe).

ECHIPA_creeaza_PRODUS = relație care leagă entitățile ECHIPA și PRODUS(alaturi de subentitatile sale), reflectând legătura dintre acestea (De catre care echipa este creat produsul x). Ea are cardinalitatea minimă 1:1 (o echipa creeaza cel putin un produs, iar un produs este creat de o singura echipa) și cardinalitatea maximă 1:n (o echipa poate sa creeze mai multe produse de tipuri diverse, iar orice tip de produs este creat de o singura echipa).

TASK_atasat_la_ECHIPA = relație care leagă entitățile ECHIPA și TASK, reflectând legătura dintre acestea (ce task realizeaza o echipa). Aceasta este o relatie many to many care urmeaza sa fie sparta prin tabelul “ATASAT_LA” in doua relatii one to many. Ea are cardinalitatea minimă 1:1 (o echipa realizeaza cel putin un task(pte. Nu poate sa creeze un produs daca nu face nici-un task necesar), iar un task ii apartine cel putin unei echipe) și cardinalitatea maximă m:n (o echipa poate sa rezolve mai multe task-rui, iar un task se regaseste la mai multe echipe).

SALARIAT_detine_SPECIALIZARE = relație many to many care leagă entitățile SALARIAT și SPECIALIZARE, reflectând legătura dintre acestea (ce specializari

ii corespunde salariatului x). Ea are cardinalitatea minimă 1:1 (un salariat trebuie sa aibe cel putin o specializare, iar o specializare ii apartine cel putin unui salariat) și cardinalitatea maximă n:m. Relatia many to many o sa fie sparta prin tabelul “DETINE” in doua relatii one to many.

PRODUS_foloseste_COMPONENTA = relație many to many care leagă entitățile PRODUS și COMPONENTA, reflectând legătura dintre acestea (din cecomponente este alcatuit un produs). Ea are cardinalitatea minimă 0:1 (un produs contine cel putin o componenta iar o componenta poate sa nu intre neaparat in alcatuirea unui produs (eg: Sa zicem ca o sa inseram componenta 1 tip GPU de la nVidia si componenta 2 tip GPU de la AMD. Este posibil ca toate produsele sa aibe GPU doar de la AMD -> componenta 2, iar componenta 1 sa nu fie furnizata) și cardinalitatea maximă n:m.

Relatia many to many se va sparge prin tabelul “FOLOSESTE” si vom obtine 2 relatii one to many .

CLIENT_comanda_PRODUS = relație many to many care leagă entitățile CLIENT și PRODUS, reflectând legătura dintre acestea (cate si ce produse comanda un client). Ea are cardinalitatea minimă 1:1 (un client poate sa comande cel putin un produs si un tip de produs poate fi comandat de cel putin un client. Un produs nu poate sa nu fie comandat ne nimeni deoarece firma creaza produsele la comanda) și cardinalitatea maximă n:m. Relatia many to many o sa fie sparta prin tabelul asociatvi “COMANDA” in doua relatii one to many.

Ex.5) Atribute:

Entitatea independenta PRODUS are ca atribute:

- cod_fabricatie = variabila de tip intreg de lungime maxim 5, care reprezinta codul unui produs fabricat (un fel de serie de fabricatie).
- denumire_produs = variabila de tip caracter, de lungime maxim 32, care reprezinta numele produsului fabricat.
- pret_final = variabila de tip numeric (real), de lungime maxim 10, care reprezinta pretul final al unui produs(calculat dupa formula: $\text{sum}(p_componente * 2)$).

- `id_echipa` = variabila de tip intreg, de lungime maxim 5, care reprezinta codul echipei responsabile pentru crearea produsului. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul ECHIPA.

Subentitatea MONITOR are ca attribute:

- `cod_fabricatie` = variabila de tip intreg “mostenita” din superentitatea PRODUS, reprezentand codul unui monitor.
- `dimensiune_display` = valoare numerica reala, de lungime maxim 10, care reprezinta dimensiunea diagonalei unui display al unui monitor, masurat in cm.
- `forma` = variabila de tip caracter, de lungime maxim 32, care reprezinta forma monitorului (la adaugarea datelor, forma poate fii doar: normal sau curbat).

Subentitatea LAPTOP are ca attribute:

- `cod_fabricatie` = variabila de tip intreg “mostenita” din superentitatea PRODUS, reprezentand codul unui produs de tip laptop.
- `greutate` = valoare numerica reala, de lungime maxim 10, care reprezinta greutatea unui laptop.
- `tip` = variabila de tip caracter, de lungime maxim 32, care reprezinta forma laptopului(la adaugarea de date, aceasta poate sa fie : notebook, gaming).

Subentitatea DESKTOP_PC are ca attribute:

- `cod_fabricatie` = variabila de tip intreg “mostenita” din superentitatea PRODUS, reprezentand codul unui produs de tip desktop_pc.
- `lungime` = valoare numerica reala, de lungime maxim 10, care reprezinta lungimea unui desktop_pc.
- `latime` = valoare numerica reala, de lungime maxim 10, care reprezinta latimea unui desktop_pc.
- `inaltime` = valoare numerica reala, de lungime maxim 10, care reprezinta inaltimea unui desktop_pc.

Entitatea SALARIAT are ca attribute:

- `cnp_salariat` = variabila tip intreg, de lungime maxim 10, care reprezinta codul unic al unui angajat. (la inserarea datelor s-au folosit cnp-uri fictive).
- `lider` = valoare tip intreg, de lungime maxim 5, care reprezinta codul liderului unui salariat. Este corelat cu lider din cadrul entitatii ECHIPA, si trebuie sa corespunda unei valori a cheii primare tot din SALARIAT. (Un exemplu similar: employees, departments si atributul `manager_id`).

- id echipa = variabila de tip intreg, de lungime maxim 5, care reprezinta codul echipei din care angajatul face parte. Atributul trebuie sa corespunda unei valori a cheii primare din tabelul ECHIPA.
- nume = variabila tip caracter, de lungime maxim 32, care reprezinta numele angajatului.
- prenume = variabila tip caracter, de lungime maxim 32, care reprezinta prenumele angajatului.
- Varsta = variabila tip intreg, de lungime maxim 3, reprezinta varsta angajatului. Poate lua valori doar ≥ 18 .
- salariu = valoare numerica reala, de maxim 10 caractere, reprezentand salariul unui angajat. Nu poate lua valori negative.
- data_angajare = variabilă de tip dată calendaristică, care reprezintă data angajarii salariatului respectiv. La initializare nu poate lua o valoare mai veche de 100 de ani.
- Nr_telefon = variabila tip caracter, de lungime maxim 32, care reprezinta numarul de telefon al angajatului.
- Email = variabila tip caracter, de lungime maxim 32, reprezentand email-ul unui angajat.

Entitatea ECHIPA are ca atribut:

- Id echipa = variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al unei echipe.
- lider = variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al liderului echipei. Acest atribut trebuie sa corespunda unei valori a cheii primare din SALARIAT.
- reputatie = valoare numerica reala, de lungime maxim 5, reprezentand un punctaj ce ilustreaza cat de priceputa este echipa x.
- pseudonim echipa = variabila tip caracter, de lungime maxim 32, reprezentand denumirea echipei aleasa de membrii. Nu este o denumire oficiala.

Entitatea TASK are ca atribut:

- Id_task = variabila tip intreg, lungime maxim 5, reprezentand identificatoru unic al unui task.
- descriere_scurta = variabila de tip caracter, de lungima maxim 128, reprezentand succint activitatea propriu zisa in care consta task-ul. Exemplu: "De programat procesorul pe motherboard" sau "De testat duritate laptop".

Entitatea FILE_ARHIVA are ca atribut:

- Id_fila = variabila tip intreg, de lungime maxim 5, reprezentand numarul unei file din istoric, file pe care se afla date despre un angajat x.
- cnp_salariat = variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al unui angajat care a schimbat echipa pe parcursul activitatii la companie si inca lucreaza in aceasta companie. Acest atribut trebuie sa corespunda unei valori a cheii primare din SALARIAT.
- Id echipa= variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al unei echipe in care un angajat a lucrat o perioada. Acest atribut trebuie sa corespunda unei valori a cheii primare din ECHIPA.
- data_angajare si data_concediere sunt ambele valori de tip data calendaristica reprezentand datele intre care un angajat x a lucrat in echipa y. Cand introducem date, data_concediere trebuie sa fie mai veche decat noua data de angajare a salariatului)

Entitatea SEDIU are ca attribute:

- Id_sediu = variabila tip intreg, de lungime maxim 5, reprezentand codul unui sediu al unei echipe x.
- Id echipa= variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al unei echipe care are acel sediu. Acest atribut trebuie sa corespunda unei valori a cheii primare din ECHIPA.
- Oras = variabila tip caracter, de lungime maxim 32, reprezentand numele unui oras.
- Regiune = variabila tip caracter, de lungime maxim 32, reprezentand numele unei regiuni.

Entitatea COMPONENTA are ca attribute:

- serie = variabila de tip intreg de lungime maxim 5, care reprezinta identificatorul unic al unei componente.
- tip = variabila tip caracter, de lungime maxim 32, reprezentand tipul componentei. La initializare, tipul poate sa fie de forma: microprocesor, procesor grafic / placa video, display, ssd, power supply, motherboard, procesor, RAM, etc..
- producator = variabila de tip caracter de lungime maxim 32, reprezentand numele producatorului tipurilor de componente enumerate mai sus. Pot fii asemanatoare sau de forma : nVidia, IntelCore, AMD, ASRock, etc..
- Pret_componenta = valoare numerica reala, de lungime maxim 10, reprezentand pretul fiecărei componente in parte.

Entitatea CLIENT are ca attribute:

- id_client = variabila tip intreg, de lungime maxim 5, care reprezinta codul unic al unui client ce solicita niste produse. Clientului nu I se solicita CNP-ul , asta fiind o informatie confidentiala.
- nume = variabila tip caracter, de lungime maxim 32, care reprezinta numele clientului.
- prenume = variabila tip caracter, de lungime maxim 32, care reprezinta prenumele clientului.
- adresa = variabila tip caracter, de lungime maxim 128. In acest string se va trece o scurta descriere a adresei, spre exemplu: "Jud. X, orasul Y, strada Z, cladirea nr. 10". Pentru a putea trece in forma normala 1 (FN1) acest atribut va fi inlocuit in diagrama conceptuala de 3 attribute tip caracter, dedicate locatiei exacte.
- Nr_telefon = variabila tip caracter, de lungime maxim 32, care reprezinta numarul de telefon al clientului.
- Email = variabila tip caracter, de lungime maxim 32, reprezentand email-ul clientului.

Entitatea SPECIALIZARE are ca attribute:

- Id_specializare = valoare tip intreg, lungime maxim 5, reprezentand identificatorul unic al unei specializari.
- titlu = variabila tip caracter, lungime maxim 32, reprezentand denumirea specializarii. Exemple : programator, inginer, tester, electrician.

Relatia SALARIAT_**detine**_SPECIALIZARE are attributele: cnp_salariat, id_specializare, document(variabila caracter de maxim 128 caractere ce are un text de forma: diploma de absolvire a facultatii X sau scrisoare de la fosta companie Y.) Atributele id_specializare si cnp_salariat sunt valori numerice intregi de lungime maxim 5 si trebuie sa corespunda unor valori ale cheilor primare din SPECIALIZARE, respectiv SALARIAT.

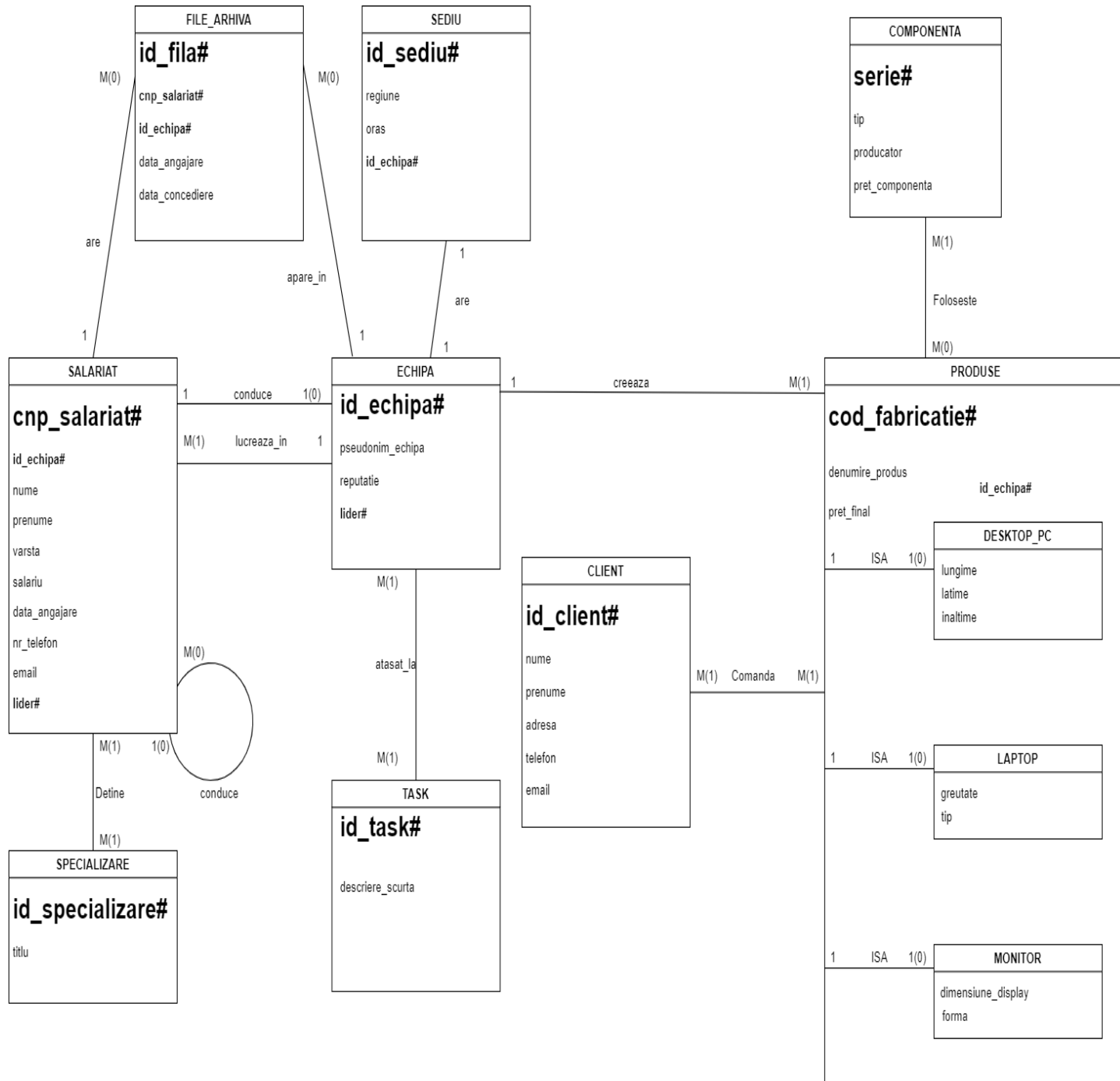
Relatia TASK_**atasat_la**_ECHIPA are ca attribute: id echipa, id_task, data (valoare tip data calendaristica reprezentand data la care trebuie sa se faca task-ul x). Atributele id echipa si id_task sunt valori numerice intregi de lungime maxim 5 si trebuie sa corespunda unor valori ale cheilor primare din ECHIPA, respectiv TASK.

Relatia **CLIENT_comanda_PRODUS** are ca attribute: id_produs, id_client, cantitate (valoare tip intreg, lungime maxim 10, reprezinta cate produse de tip x vrea clientul), tip_livrare(variabila tip caracter, lungime maxim 32, reprezinta tipul de livrare ales de client pentru produs. Valorile pot fii doar : standard sau premium). Atributele id_client si id_produs sunt valori numerice intregi de lungime maxim 5 si trebuie sa corespunda unor valori ale cheilor primare din CLIENT, respectiv PRODUS.

Relatia **PRODUS_foloeste_COMPONENTA** are attributele: cod_fabricatie, serie, denumire_furnizor (variabila tip caracter, lungime maxim 32, reprezentand un nume fictiv, generic de furnizor), id_furnizor (variabila tip intreg, lungima maxim 5, identificator unic pentru furnizor a. in un furnizor sa apartina unui produs si sa ia un tip de componenta. Astfel relatia many to many e sparta in 2 relatii one to many). Atributele cod_fabricatie si serie sunt valori numerice intregi de lungime maxim 5 si trebuie sa corespunda unor valori ale cheilor primare din PRODUS, respectiv COMPONENTA.

Ex.6) Diagrama Entitate Relatie:

Nota: in fiecare entitate, cheia primara este scrisa in parte de sus, cu un font mai mare si ingrosat. De asemenea, apar si cheile straine (unde e cazul), scrise cu font normal printre celelalte attribute.



Ex.9) Inainte de crearea diagramei conceptuale (ex.7), o sa prezentam formele normale corespunzatoare modelului relational, pentru a putea construi diagrama conceptuala direct in forma normala 3.

Pentru inceput, trecerea la forma normala 1.

Observam in entitatea CLIENTI atributul “adresa”. Conform celor scrise mai sus in prezentarea atributelor, adresa stocheaza un string cu mai multe campuri importante precum: judet, oras , cladire. Conform teoriei specifice FN1, fiecarui atribut trebuie sa ii corespunda o valoare indivizibila (atomica).

Astfel, putem solutiona aceasta problema in doua moduri: crearea separata a altei entitati LOCATIE, pe care mai apoi o sa o legam de CLIENT sau adaugarea mai multor attribute in entitatea CLIENT. Noi vom aborda cea de a doua metoda si vom mai adauga in entitatea CLIENT attributele: oras(tip caracter, lungime maxima 32), judet(tip caracter, lungime maxima 32), strada(tip caracter, lungime maxim32) in locul atributului “adresa”.

Acesta a fost un exemplu de normalizare pana la forma normala 1(FN1). In mod asemanator vom proceda si pentru atributul document din tabelul asociativ DETINE care va sparge relatia many to many dintre SALARIAT si SPECIALIZARE(in loc de *document* cum am mentionat anterior o sa avem *diploma* si *scrisoare*).

In ceea ce priveste forma normala 2:

O relatie R este in cea de a doua forma normala (FN2) daca este in FN1 si , in plus, fiecare atribut care nu participa la cheia primara este dependent de intreaga cheie. In cadrul nostru, luam din nou ca exemplu relatia TASK_atasat_la_ECHIPA. Aceasta relatie este deja in forma normala 1. Observam ca aici putem compune cheia primara din id echipa si id_task. Pe langa aceste chei, avem si atributul data. In cazul nostru, data este dependenta atata de id echipa cat si de id_task , reprezentand data la care echipa x face task-ul y. Prin urmare aceasta relatie este in FN2.

Un exemplu de non-FN2 ar fi fost daca in aceasta relatie ar fi aparut si o informatie legata de un anumit salariat care ar fi depins doar de id echipa, nu si de id_task.

Exemplu vizual non-FN2:

#Echipa	Salariatul	#Task	Data
e1	s1	t1	d1
e1	s2	t2	d2
e2	s3	t2	d3

Este evidentiata dependenta Salariatului doar de Echipa nu si de Task.

In ceea ce priveste forma normala 3 (FN3):

Pe langa indeplinirea celorlalte 2 criterii precizate anterior, ca o relatie sa se afla in forma normala 3, trebuie ca oricare coloana noncheie sa depinda direct de cheia primara.

Luam ca exemplu relatia TASK_atasat_la_ECHIPA. Aici singurul atribut care nu este cheie este data si dupa cum am discutat anterior, depinde de intraga cheie primara compusa.

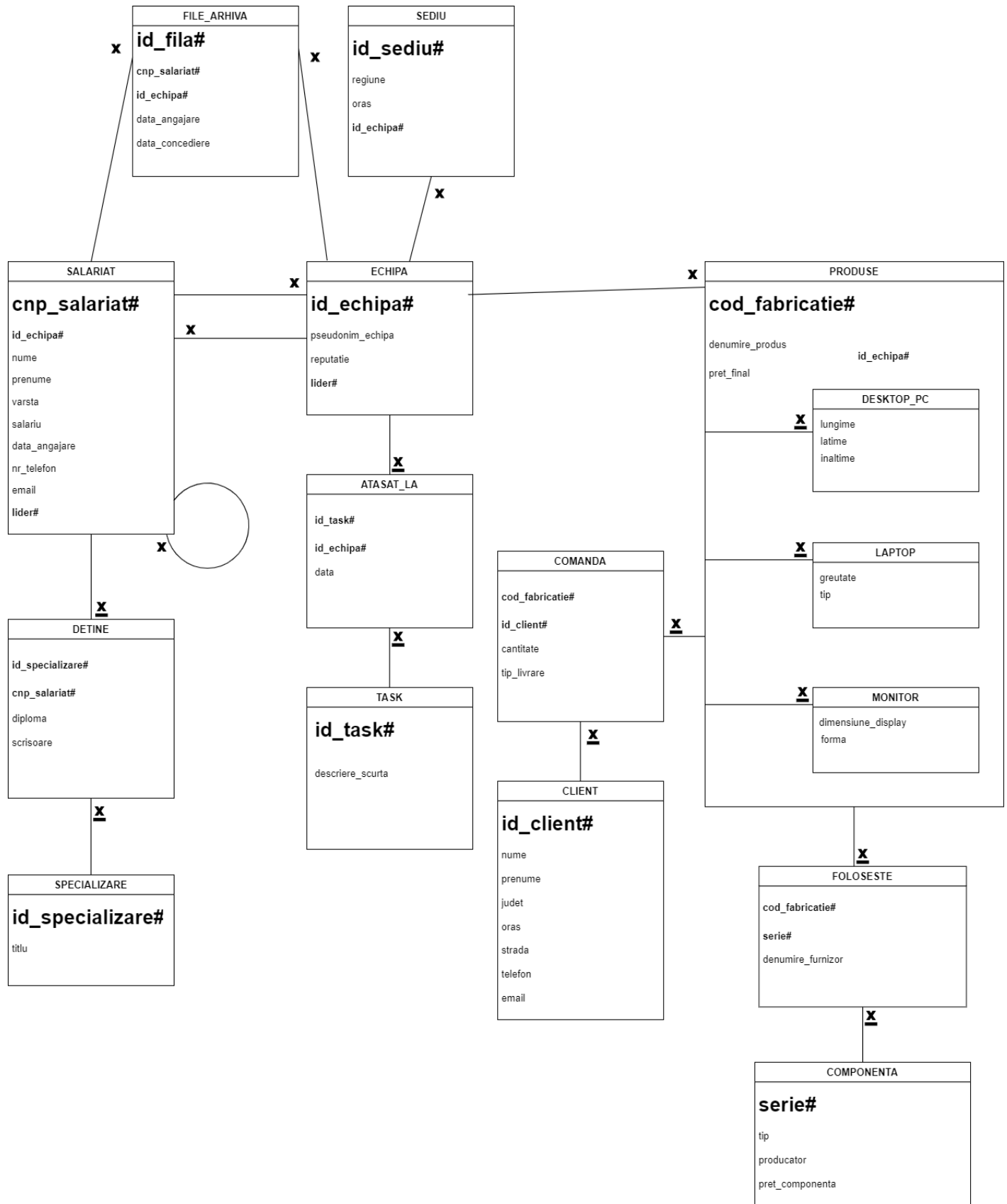
Alt exemplu este relatia CLIENT_comanda_PRODUS. In comanda avem cheia primara compusa din (id_client si id_produș). Astfel, restul coloanelor / atributelor sunt raportate direct la aceasta cheie. Comanda x e facuta de clientul y. De cheia primara vor depinde attributele “cantitate” si “tip_livrare” (Comanda x are inregistrata cantitatea y si livrarea z).

Un exemplu de non-FN3 ar fi fost urmatorul:

Presupunem ca in tabelul asociativ “Comanda” ar exista pret_livrare, prin care livrarile (indiferent daca sunt standard sau premium) ar avea toate preturi variate. Astfel, atributul pret_livrare ar fi dependent direct de tipul de livrare, nu de cheia primara compusa.

Ex.7) Acestea fiind spuse, putem incepe constructia **diagramei conceptuale:**

Nota: Entitatile cu cheie primara independenta o au scrisa la inceput, ingrosat si cu font mai mare decat restul atributelor. Sunt prezente si cheile straine (unde e cazul) printre restul atributelor. In tabelele asociative: “*atasat_la*”, “*detine*”, “*foloseste*” si “*comanda*” cheia primara este compusa din cheile straine prezente in entitate.



Ex.8) Schemele relationale corespunzatoare diagramei conceptuale sunt:

SALARIAT(cnp_salariat#,
id echipa,nume,prenume,varsta,salariu,data_angajare,nr_telefon,email,lider)

ECHIPA(id echipa#, pseudonim echipa, reputatie, lider)

FILE_ARHIVA(id_fila#, cnp_salariat, id echipa, data_angajare, data_concediere)

DETINE(id_specializare#, cnp_salariat#, diploma, scrisoare)

SPECIALIZARE(id_specializare#, titlu)

TASK(id_task#, descriere_scurta)

ATASAT_LA(id_task#,id echipa#, data)

SEDIU(id_sediu#, regiune, oras, id echipa)

PRODUS(cod_fabricatie#, denumire_produs, pret_final, id echipa)

LAPTOP(cod_fabricatie#, greutate, tip)

DESKTOP_PC(cod_fabricatie#, lungime,latime,inaltime)

MONITOR(cod_fabricatie#, dimensiune_display, forma)

COMPONENTA(serie#, tip, producator, pret_componenta)

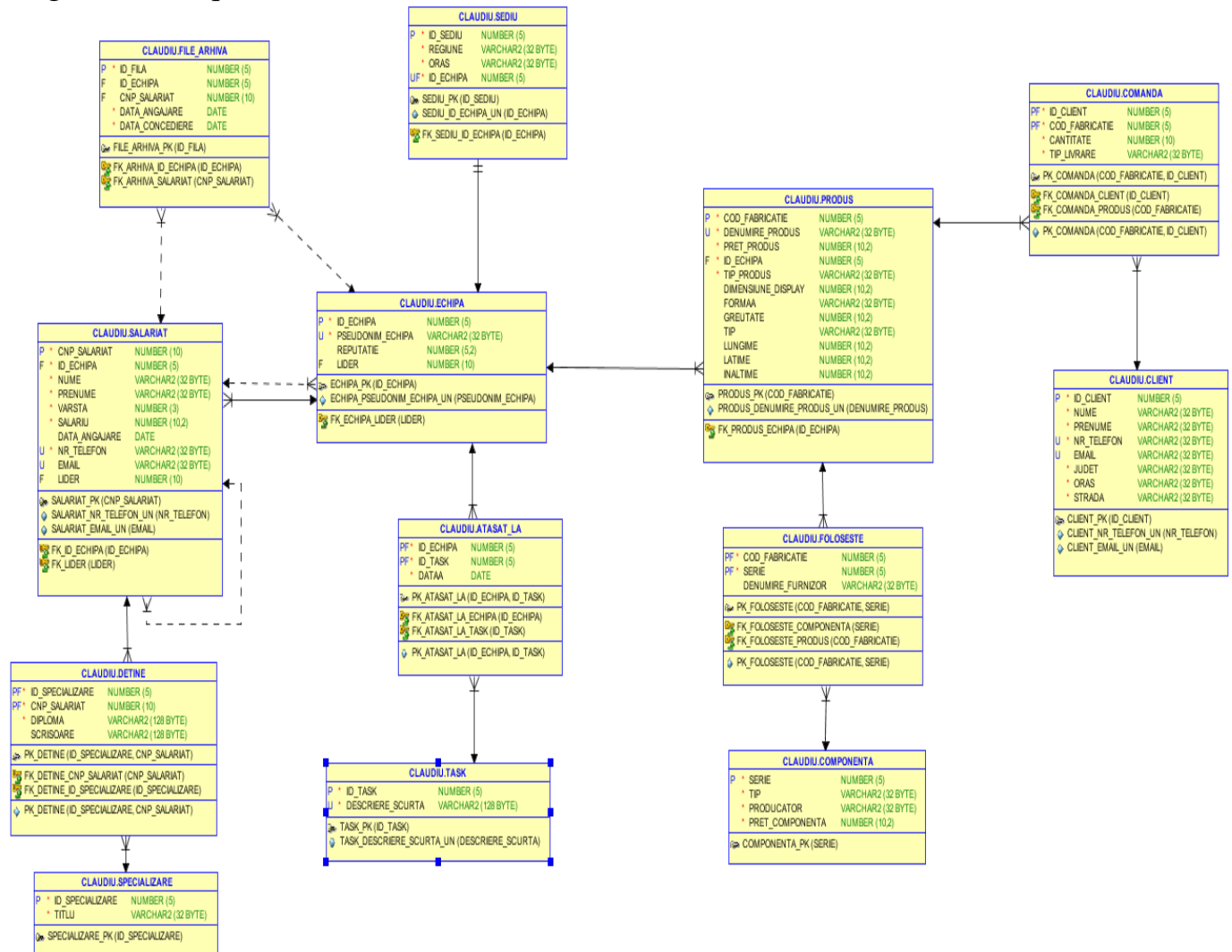
FOLOSESTE(cod_fabricatie#, serie#, denumire_furnizor)

CLIENT(id_client#,nume,prenume,judet,oras,strada,telefon,email)

COMANDA(cod_fabricatie#, id_client#, cantitate, tip_livrare)

Ex.10) Crearea si popularea tabelelor in SQL:

Diagrama din sql arata asa:



Tabelul Salariat:

```
CREATE TABLE SALARIAT
```

```
(
  cnp_salariat number(10,0) NOT NULL PRIMARY KEY,
  id echipa number(5,0) NOT NULL,
  CONSTRAINT fk_id_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA (id_echipa),
  nume varchar2(32) NOT NULL,
  prenume varchar2(32) NOT NULL,
  varsta number(3,0) NOT NULL,
  CONSTRAINT varsta_Ck CHECK (varsta >= 18),
  salariu number(10,2) NOT NULL,
  CONSTRAINT salariu_check CHECK (salariu >0),
  data_angajare date,
  nr_telefon varchar2(32) NOT NULL UNIQUE,
  email varchar2(32) UNIQUE,
  lider number(10,0),
  CONSTRAINT fk_lider FOREIGN KEY (lider) REFERENCES SALARIAT(cnp_salariat)
);
```

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	CNP_SALARIAT		NUMBER(10,0)		No	(null)		1		(null)
2	ID_ECHIPA		NUMBER(5,0)		No	(null)		2		(null)
3	NUME		VARCHAR2(32 BYTE)		No	(null)		3		(null)
4	PRENUME		VARCHAR2(32 BYTE)		No	(null)		4		(null)
5	VARSTA		NUMBER(3,0)		No	(null)		5		(null)
6	SALARIU		NUMBER(10,2)		No	(null)		6		(null)
7	DATA_ANGAJARE		DATE		Yes	(null)		7		(null)
8	NR_TELEFON		VARCHAR2(32 BYTE)		No	(null)		8		(null)
9	EMAIL		VARCHAR2(32 BYTE)		Yes	(null)		9		(null)
10	LIDER		NUMBER(10,0)		Yes	(null)		10		(null)

❖	CNP_SALARIAT	❖	ID_ECHIPA	❖	NUME	❖	PRENUME	❖	VARSTA	❖	SALARIU	❖	DATA_ANGAJARE	❖	NR_TELEFON	❖	EMAIL	❖	LIDER
1	5000000000		10	Jhon		Wick		22	18020.69		28-MAY-21		539-824-3883		johnwick@gmail.com				(null)
2	5000000005		10	Poison		Andra		20	17420.99		28-MAY-21		544-844-3783		andra_p@gmail.com				5000000000
3	5000000010		10	Mike		Wood		32	27820.28		28-MAY-21		529-814-7883		mike_sk@gmail.com				5000000000
4	5000000015		5	Mikasa		Ackerman		27	2000.11		DEC-15		239-824-3783		attack_on_m@gmail.com				(null)
5	5000000020		5	Mihai		Moromete		82	11420.22		JUN-90		639-824-3884		salcamul_m@gmail.com				5000000015
6	5000000025		5	Gerald		Smith		41	5720.10		AUG-00		877-888-3443		smith_g@gmail.com				5000000015
7	5000000030		15	Levi		Ackerman		32	12820.10		AUG-15		549-444-3883		levi_a@gmail.com				(null)
8	5000000035		15	Mike		Moris		21	8820.99		28-MAY-21		229-824-3883		mike_moris@yahoo.com				5000000030
9	5000000040		15	Melissa		Monroe		42	9820.99		10-AUG-00		539-824-3773		monroe_m@gmail.com				5000000030
10	5000000045		20	Ion		Inica		44	9820.99		10-AUG-15		544-8222-3883		ionel_ionelule@gmail.com				(null)
11	5000000050		20	Maria		Marioara		32	1820.99		15-OCT-11		533-887-3003		marie_marioara@gmail.com				5000000045
12	5000000055		25	Dumitru		Ion		35	1820.99		28-MAY-21		566-871-3993		dumitru_i@gmail.com				5000000060
13	5000000060		25	Tony		Stark		37	21820.99		11-NOV-09		577-777-7773		jarvis@gmail.com				(null)
14	5000000065		30	Tom		Ford		41	8920.09		11-NOV-09		522-222-2282		tomford@gmail.com				5000000070
15	5000000070		30	Steve		Jobs		44	29820.99		21-NOV-02		534-444-4444		steve_a@gmail.com				(null)
16	5000000075		30	Rick		Morty		18	5820.99		28-MAY-21		599-999-9999		pickle@gmail.com				5000000070
17	5000000080		35	Codin		Dan		25	8820.99		28-MAY-21		599-866-3999		codin_d@gmail.com				5000000085
18	5000000085		35	Silva		Specter		30	23820.99		12-FEB-10		587-787-7778		silva@gmail.com				(null)

Tabelul **Echipa**:

```

CREATE TABLE ECHIPA
(
    id_echipa number(5,0) NOT NULL PRIMARY KEY,
    pseudonim_echipa varchar2(32) NOT NULL UNIQUE,
    reputatie number(5,2),
    CONSTRAINT reputatie_Ck CHECK (reputatie >= 0),
    lider number(10,0)
);

```

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_ECHIPA	NUMBER(5,0)	No	(null)	1 (null)	
2 PSEUDONIM_ECHIPA	VARCHAR2(32 BYTE)	No	(null)	2 (null)	
3 REPUTATIE	NUMBER(5,2)	Yes	(null)	3 (null)	
4 LIDER	NUMBER(10,0)	Yes	(null)	4 (null)	

❖ ID_ECHIPA	❖ PSEUDONIM_ECHIPA	❖ REPUTATIE	❖ LIDER
1	5 Razboinicii	85.9	50000000015
2	10 Flash	78.8	50000000000
3	15 Echipa Geniilor	98.9	50000000030
4	20 Bit Legion	90.5	50000000045
5	25 Runtime Terror	95.69	50000000060
6	30 Jurassic Code	44.6	50000000070
7	35 Rock N Code	88.8	50000000085

Tabelul **FILE_ARHIVA**:

```

CREATE TABLE FILE_ARHIVA
(
  id_fila number(5,0) NOT NULL PRIMARY KEY,
  id echipa number(5,0),
  cnp_salariat number(10,0),
  data_angajare date NOT NULL,
  data_concediere date NOT NULL,
  CONSTRAINT data_check CHECK ( data_angajare <= data_concediere),
  CONSTRAINT fk_arhiva_id echipa FOREIGN KEY (id echipa) REFERENCES ECHIPA(id echipa),
  CONSTRAINT fk_arhiva_salariat FOREIGN KEY (cnp_salariat) REFERENCES SALARIAT(cnp_salariat)
);

```

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID_FILA	NUMBER(5,0)	No	(null)	1	(null)
2	ID_ECHIPA	NUMBER(5,0)	Yes	(null)	2	(null)
3	CNP_SALARIAT	NUMBER(10,0)	Yes	(null)	3	(null)
4	DATA_ANGAJARE	DATE	No	(null)	4	(null)
5	DATA_CONCEDIERE	DATE	No	(null)	5	(null)

	❖ ID_FILA	❖ ID_ECHIPA	❖ CNP_SALARIAT	❖ DATA_ANGAJARE	❖ DATA_CONCEDIERE
1	1	15	50000000005	20-JAN-21	23-MAY-21
2	2	35	50000000040	10-MAY-00	05-AUG-00
3	3	5	50000000070	14-MAY-01	20-OCT-02
4	4	20	50000000070	10-MAY-00	13-MAY-01
5	5	35	50000000025	03-AUG-00	09-AUG-00
6	6	15	50000000045	05-JAN-12	09-AUG-15
7	7	15	50000000000	20-JAN-21	27-MAY-21
8	8	15	50000000085	14-JAN-10	11-FEB-10
9	9	20	50000000060	20-AUG-09	10-OCT-09
10	10	5	50000000075	10-MAY-21	27-MAY-21

Tabelul Specializari:

```
CREATE TABLE SPECIALIZARE
(
  id_specializare number(5,0) NOT NULL PRIMARY KEY,
  titlu varchar2(32) NOT NULL
);
```

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_SPECIALIZARE	NUMBER(5,0)	No	(null)	1	(null)
2 TITLU	VARCHAR2(32 BYTE)	No	(null)	2	(null)

	❖ ID_SPECIALIZARE	❖ TITLU	
1	1	System Engineer	
2	2	Tester	
3	3	Technician	
4	4	Computer Programmer	
5	5	Software Developer	

Tabelul Detine:

```
CREATE TABLE DETINE
(
  id_specializare number(5,0) NOT NULL,
  cnp_salariat number(10,0) NOT NULL,
  diploma varchar2(128) NOT NULL,
  scrisoare varchar2(128),
  CONSTRAINT fk_detine_id_specializare FOREIGN KEY (id_specializare) REFERENCES SPECIALIZARE(id_specializare),
  CONSTRAINT fk_detine_cnp_salariat FOREIGN KEY (cnp_salariat) REFERENCES SALARIAT(cnp_salariat),
  CONSTRAINT pk_detine PRIMARY KEY (id_specializare, cnp_salariat)
);
```

⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEFAULT	⚡ COLUMN_ID	⚡ COMMENTS
1 ID_SPECIALIZARE	NUMBER(5,0)	No	(null)	1 (null)	
2 CNP_SALARIAT	NUMBER(10,0)	No	(null)	2 (null)	
3 DIPLOMA	VARCHAR2(128 BYTE)	No	(null)	3 (null)	
4 SCRISOARE	VARCHAR2(128 BYTE)	Yes	(null)	4 (null)	

⚡ ID_SPECIALIZARE	⚡ CNP_SALARIAT	⚡ DIPLOMA	⚡ SCRISOARE
1	1 5000000000	Diploma Absolvire U.T.C.N.	(null)
2	2 5000000005	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: STARK IND.
3	3 5000000010	Diploma Absolvire P.B.	(null)
4	4 5000000015	Diploma Absolvire U.T.C.N.	(null)
5	5 5000000020	Diploma Absolvire P.B.	(null)
6	1 5000000025	Diploma Absolvire FMI UNIBUC	(null)
7	2 5000000030	Diploma Absolvire OXFORD	(null)
8	3 5000000035	Diploma Absolvire BABES BOLYAI	(null)
9	4 5000000040	Diploma Absolvire ASE	RECOMANDARE JOB VECHI: SPACEX
10	5 5000000045	Diploma Absolvire U.T.C.N.	(null)
11	1 5000000050	Diploma Absolvire U.T.C.N.	(null)
12	2 5000000055	Diploma Absolvire FMI UNIBUC	(null)
13	4 5000000060	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: STARK IND.
14	3 5000000065	Diploma Absolvire U.T.C.N.	(null)
15	5 5000000070	Diploma Absolvire OXFORD	(null)
16	1 5000000075	Diploma Absolvire P.B.	(null)
17	5 5000000080	Diploma Absolvire P.B.	(null)
18	2 5000000085	Diploma Absolvire U.T.C.N.	(null)
19	4 5000000075	Diploma Absolvire U.T.C.N.	(null)
20	2 5000000020	Diploma Absolvire FMI UNIBUC	RECOMANDARE JOB VECHI: IBM
21	4 5000000045	Diploma Absolvire ASE	RECOMANDARE JOB VECHI: SPACEX
22	3 5000000025	Diploma Absolvire ASE	(null)

Tabelul Sediu:

```
CREATE TABLE SEDIU
```

```
(
  id_sediu number(5,0) NOT NULL PRIMARY KEY,
  regiune varchar2(32) NOT NULL,
  oras varchar2(32) NOT NULL,
  id echipa number(5,0) NOT NULL UNIQUE,
  CONSTRAINT fk_sediu_id_echipa FOREIGN KEY (id_echipa) REFERENCES ECHIPA(id_echipa)
);
```

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 ID_SEDIU	NUMBER (5,0)	No	(null)	1 (null)	
2 REGIUNE	VARCHAR2 (32 BYTE)	No	(null)	2 (null)	
3 ORAS	VARCHAR2 (32 BYTE)	No	(null)	3 (null)	
4 ID_ECHIPA	NUMBER (5,0)	No	(null)	4 (null)	

❖ ID_SEDIU	❖ REGIUNE	❖ ORAS	❖ ID_ECHIPA
1	1 America	New-York	5
2	3 America	Los Angeles	10
3	4 Europa	Paris	15
4	5 Europa	Bucuresti	25
5	6 Europa	Spania	20
6	7 Europa	Bucuresti	30
7	8 America	San Francisco	35

Tabelul Task:

```
CREATE TABLE TASK
```

```
(
  id_task number(5,0) NOT NULL PRIMARY KEY,
  descriere_scurta varchar2(128) NOT NULL UNIQUE
);
```

⚙	COLUMN_NAME	DATA_TYPE	⚙	NULLABLE	DATA_DEFAULT	⚙	COLUMN_ID	⚙	COMMENTS
1	ID_TASK	NUMBER(5,0)	No	(null)		1	(null)		
2	DESCRIERE_SCURTA	VARCHAR2(128 BYTE)	No	(null)		2	(null)		

⚙	ID_TASK	⚙	DESCRIERE_SCURTA
1	4		De asamblat produsul final
2	5		De impachetat produsul
3	1		De programat componenta
4	2		De testat duritate
5	3		De testat rezistenta la caldura

Tabelul Atasat_la:

```

CREATE TABLE ATASAT_LA
(
  id echipa number(5,0) NOT NULL,
  id_task number(5,0) NOT NULL,
  dataa date not NULL,
  CONSTRAINT fk_atasat_la_echipa FOREIGN KEY (id echipa) REFERENCES ECHIPA(id echipa),
  CONSTRAINT fk_atasat_la_task FOREIGN KEY (id_task) REFERENCES TASK(id_task),
  CONSTRAINT pk_atasat_la PRIMARY KEY (id echipa, id_task)
);

```

	⚡ COLUMN_NAME	⚡ DATA_TYPE	⚡ NULLABLE	DATA_DEFAULT	⚡ COLUMN_ID	⚡ COMMENTS
1	ID_ECHIPA	NUMBER (5, 0)	No	(null)	1	(null)
2	ID_TASK	NUMBER (5, 0)	No	(null)	2	(null)
3	DATAA	DATE	No	(null)	3	(null)

	⚡ ID_ECHIPA	⚡ ID_TASK	⚡ DATAA
1	5	1	30-MAY-21
2	5	2	29-MAY-21
3	5	4	30-JUN-21
4	10	1	20-MAR-21
5	10	4	25-MAR-21
6	15	2	12-MAY-21
7	15	3	14-MAR-21
8	15	4	22-APR-21
9	20	1	30-APR-21
10	20	2	30-MAY-21
11	20	4	01-JUN-21
12	30	1	28-MAR-21
13	30	3	27-APR-21
14	30	4	28-APR-21
15	35	1	29-APR-21
16	35	4	30-APR-21
17	35	5	01-MAY-21

Tabelul Produs:

```
CREATE TABLE PRODUS
(
  cod_fabricatie number(5,0) NOT NULL PRIMARY KEY,
  denumire_produs varchar2(32) NOT NULL UNIQUE,
  pret_produs number(10,2) NOT NULL,
  CONSTRAINT check_pret_p CHECK (pret_produs > 0),
  id echipa number(5,0) NOT NULL,
  tip_produs varchar2(32) NOT NULL,
  CONSTRAINT produs_check CHECK (upper(tip_produs)='LAPTOP' or upper(tip_produs)='MONITOR' or upper(tip_produs)='DESKTOP_PC'),
  dimensiune_display number(10,2),
  CONSTRAINT dimensiune_check CHECK (dimensiune_display > 0),
  formaa varchar2(32),
  CONSTRAINT forma_monitor_check CHECK( upper(formaa)='NORMAL' or upper(formaa)='CURBAT'),
  greutate number(10,2),
  CONSTRAINT greutate_check CHECK (greutate > 0),
  tip varchar2(32),
  CONSTRAINT tip_laptop_check CHECK( upper(tip)='NOTEBOOK' or upper(tip)='GAMING'),
  lungime number(10,2),
  CONSTRAINT lungime_check CHECK (lungime > 0),
  latime number(10,2),
  CONSTRAINT latime_check CHECK (latime > 0),
  inaltime number(10,2),
  CONSTRAINT inaltime_check CHECK (inaltime > 0),
  CONSTRAINT fk_produs echipa FOREIGN KEY (id echipa) REFERENCES ECHIPA(id echipa)
);
```

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 COD_FABRICATIE	NUMBER (5, 0)	No	(null)	1	(null)
2 DENUMIRE_PRODUS	VARCHAR2 (32 BYTE)	No	(null)	2	(null)
3 PRET_PRODUS	NUMBER (10, 2)	No	(null)	3	(null)
4 ID_ECHIPA	NUMBER (5, 0)	No	(null)	4	(null)
5 TIP_PRODUS	VARCHAR2 (32 BYTE)	No	(null)	5	(null)
6 DIMENSIUNE_DISPLAY	NUMBER (10, 2)	Yes	(null)	6	(null)
7 FORMAA	VARCHAR2 (32 BYTE)	Yes	(null)	7	(null)
8 GREUTATE	NUMBER (10, 2)	Yes	(null)	8	(null)
9 TIP	VARCHAR2 (32 BYTE)	Yes	(null)	9	(null)
10 LUNGIME	NUMBER (10, 2)	Yes	(null)	10	(null)
11 LATIME	NUMBER (10, 2)	Yes	(null)	11	(null)
12 INALTIME	NUMBER (10, 2)	Yes	(null)	12	(null)

❖	COD_FABRICATIE	❖	DENUMIRE_PRODUS	❖	PRET_PRODUS	❖	ID_ECHIPA	❖	TIP_PRODUS	❖	DIMENSIUNE_DISPLAY	❖	FORMAA	❖	GREUTATE	❖	TIP	❖	LUNGIME	❖	LATIME	❖	INALTIME
1	20	Ultimate View X12	1200	10	monitor	82.5	normal	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
2	30	Slim BOOK	3600	10	laptop	(null)	(null)	892.5	notebook	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
3	50	Gaming Core	2600	5	desktop_pc	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	27.8	10	30			
4	60	Ultimate View X18	1200	20	monitor	120.5	curbat	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
5	70	TUF Storm	5690	25	laptop	(null)	(null)	1892.5	gaming	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
6	80	Ultra Blast	3350	20	desktop_pc	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	27.8	10	30			
7	90	Eagle VIEW Pro	5200	15	monitor	202.5	curbat	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
8	100	Air BOOK	3600	30	laptop	(null)	(null)	562.5	notebook	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
9	110	INSTINCT	3580	30	desktop_pc	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	32.8	12	31			
10	120	Ultimate View Pro	4200	35	monitor	92.5	normal	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
11	130	Slim BOOK 2	5692	25	laptop	(null)	(null)	792.5	notebook	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)
12	140	Gaming Core Lite	2600	35	desktop_pc	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	(null)	24.8	10	20			

Tabelul Componenta:

```
CREATE TABLE COMPONENTA
(
    serie number(5,0) NOT NULL PRIMARY KEY,
    tip varchar2(32) NOT NULL,
    producator varchar2(32) NOT NULL,
    pret_componenta number(10,2) NOT NULL,
    CONSTRAINT check_pret_c CHECK ( pret_componenta > 0) -- nu se cumpara componente pe gratis --
);
```

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	SERIE	NUMBER (5,0)	No	(null)	1	(null)				
2	TIP	VARCHAR2 (32 BYTE)	No	(null)	2	(null)				
3	PRODUCATOR	VARCHAR2 (32 BYTE)	No	(null)	3	(null)				
4	PRET_COMPONENTA	NUMBER (10,2)	No	(null)	4	(null)				

❖	SERIE	❖	TIP	❖	PRODUCATOR	❖	PRET_COMPONENTA
1	112	GPU	nVidia	350.2			
2	121	GPU	AMD	280.5			
3	163	CPU	Intel Core	350.2			
4	182	RAM	KingSton	210.8			
5	176	DISPLAY	LG	382.2			
6	193	CPU	AMD	272.2			
7	243	POWER SUPPLY	HyperX	287.2			

Tabelul Foloseste:

```

CREATE TABLE FOLOSESTE
(
  cod_fabricatie number(5,0) NOT NULL,
  serie number(5,0),
  denumire_furnizor varchar2(32),
  CONSTRAINT fk_foloseste_produs FOREIGN KEY (cod_fabricatie) REFERENCES PRODUS(cod_fabricatie),
  CONSTRAINT fk_foloseste_componenta FOREIGN KEY (serie) REFERENCES COMPONENTA(serie),
  CONSTRAINT pk_foloseste PRIMARY KEY (cod_fabricatie,serie)
);

```

	🔗 COLUMN_NAME	🔗 DATA_TYPE	🔗 NULLABLE	DATA_DEFAULT	🔗 COLUMN_ID	🔗 COMMENTS
1	COD_FABRICATIE	NUMBER (5, 0)	No	(null)	1 (null)	
2	SERIE	NUMBER (5, 0)	No	(null)	2 (null)	
3	DENUMIRE_FURNIZOR	VARCHAR2 (32 BYTE)	Yes	(null)	3 (null)	

	COD_FABRICATIE	SERIE	DENUMIRE_FURNIZOR
1	20	121	(null)
2	30	163	Notorious S.R.L
3	30	176	Fly Tech S.R.L.
4	20	176	Fly Tech S.R.L.
5	30	182	RamTREE
6	50	163	(null)
7	50	121	TorUS S.R.L.
8	50	182	RAMTREE ALPHA
9	50	243	Connect PPL
10	60	176	Notorious S.R.L
11	60	112	Nord TECH S.R.L.
12	90	121	(null)
13	90	176	Lg Official S.R.L.
14	120	176	Ultra View S.R.L.
15	120	112	MinGO P.f.a
16	70	163	Notorious S.R.L
17	70	182	RamTREE
18	70	112	Raptor Cx p.f.a.
19	100	163	Notorious S.R.L
20	100	182	RamTREE
21	100	121	Raptor Fox p.f.a.
22	130	193	Texkit S.R.L
23	130	182	RamTREE
24	130	121	Raptor Cx p.f.a.
25	80	182	RamTREE
26	80	163	Bionic Soft Inc.
27	80	121	Lig X Co.
28	80	243	Connect PPL
29	110	193	Ultra Soft Co.
30	110	121	TorUS S.R.L.
31	110	182	RAMTREE ALPHA
32	110	243	Connect PPL
33	140	163	Bionic Soft Inc.
34	140	121	TorUS S.R.L.
35	140	182	Ram TERRA
36	140	243	Connect PPL

Tabelul Client:

```

CREATE TABLE CLIENT
(
  id_client number(5,0) NOT NULL PRIMARY KEY,
  nume varchar2(32) NOT NULL,
  prenume varchar2(32) NOT NULL,
  nr_telefon varchar2(32) NOT NULL UNIQUE,
  email varchar2(32) UNIQUE,
  judet varchar2(32) NOT NULL,
  oras varchar2(32) NOT NULL,
  strada varchar2(32) NOT NULL
);

```

❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_CLIENT		NUMBER (5, 0)		No	(null)		1		(null)
2	NUME		VARCHAR2 (32 BYTE)		No	(null)		2		(null)
3	PRENUME		VARCHAR2 (32 BYTE)		No	(null)		3		(null)
4	NR_TELEFON		VARCHAR2 (32 BYTE)		No	(null)		4		(null)
5	EMAIL		VARCHAR2 (32 BYTE)		Yes	(null)		5		(null)
6	JUDET		VARCHAR2 (32 BYTE)		No	(null)		6		(null)
7	ORAS		VARCHAR2 (32 BYTE)		No	(null)		7		(null)
8	STRADA		VARCHAR2 (32 BYTE)		No	(null)		8		(null)

❖	ID_CLIENT	❖	NUME	❖	PRENUME	❖	NR_TELEFON	❖	EMAIL	❖	JUDET	❖	ORAS	❖	STRADA
1	1	Ion	Marian	07222222	ion_marian@gmail.com	Arges	Pitesti	Strada Soarelui							
2	2	Ion	Dorian	07223332	ion_dorian@gmail.com	Ilfov	Buftea	Strada Primaverii							
3	3	Maria	Cristea	07245622	cristea_m@gmail.com	Brasov	Bran	Strada Bucuresti							
4	4	Cornel	Marian	07229992	cornel_m@gmail.com	Bucuresti	Bucuresti	Strada Palatului							
5	5	Iulia	Cosa	07446222	(null)	Texas	Houston	Strada PN 123							

Tabelul Comanda:

```

CREATE TABLE COMANDA
(
  id_comanda number(5,0) NOT NULL UNIQUE,
  id_client number(5,0) NOT NULL,
  cod_fabricatie number(5,0) NOT NULL,
  cantitate number(10,0) NOT NULL,
  CONSTRAINT cantitate_check CHECK (cantitate > 0),
  tip_livrare varchar2(32) NOT NULL,
  CONSTRAINT livrare_check CHECK (upper(tip_livrare)='STANDARD' or upper(tip_livrare)='PREMIUM'),
  CONSTRAINT fk_comanda_client FOREIGN KEY (id_client) REFERENCES CLIENT(id_client),
  CONSTRAINT fk_comanda_produs FOREIGN KEY (cod_fabricatie) REFERENCES PRODUS(cod_fabricatie),
  CONSTRAINT pk_comanda PRIMARY KEY (id_comanda,cod_fabricatie,id_client)
);

```


❖	COLUMN_NAME	❖	DATA_TYPE	❖	NULLABLE	DATA_DEFAULT	❖	COLUMN_ID	❖	COMMENTS
1	ID_CLIENT		NUMBER(5,0)		No	(null)		1		(null)
2	COD_FABRICATIE		NUMBER(5,0)		No	(null)		2		(null)
3	CANTITATE		NUMBER(10,0)		No	(null)		3		(null)
4	TIP_LIVRARE		VARCHAR2(32 BYTE)		No	(null)		4		(null)

❖	ID_CLIENT	❖	COD_FABRICATIE	❖	CANTITATE	❖	TIP_LIVRARE	
1	1		30		2		standard	
2	1		50		3		standard	
3	1		70		12		standard	
4	2		80		1		premium	
5	2		60		1		standard	
6	3		70		6		premium	
7	3		80		2		premium	
8	4		90		4		standard	
9	4		70		12		premium	
10	4		100		15		standard	
11	5		110		18		standard	
12	5		120		22		premium	
13	5		20		8		standard	
14	5		130		4		premium	
15	3		20		1		standard	
16	3		140		5		standard	

Intreg codul folosit pentru inserarea datelor este in fisierul separat .

Ex.11), Ex.12) & Ex.13) Mai departe va voi prezenta cateva screenshot-uri cu exemple de cereri, de secventa, de operatii de actualizare.

Daca doriti intreg codul, verificati celelalte fisiere text.

Exemplu inserare date in Salariat:

```
INSERT INTO SALARIAT
VALUES(seq_salariat.nextval,10, 'Jhon', 'Wick',22,18020.69,sysdate,'539-824-3883','johnwick@gmail.com', NULL);
INSERT INTO SALARIAT
VALUES(seq_salariat.nextval,10, 'Poison', 'Andra',20,17420.99,sysdate,'544-844-3783','andra_p@gmail.com', 5000000000);
INSERT INTO SALARIAT
VALUES(seq_salariat.nextval,10, 'Mike', 'Wood',32,27820.00,sysdate,'529-814-7883','mike_sk@gmail.com', 5000000000);
```

Exemplu inserare **lider** in tabelele Echipa si Salariat:

```
UPDATE ECHIPA
SET lider = 5000000045
WHERE id echipa = 20;
UPDATE SALARIAT
SET lider = (select lider from echipa where id echipa = 20)
WHERE salariat.id echipa = 20 and cnp_salariat != (select lider from echipa where id echipa=20);
```

Exemplu secventa folosita la inserare in Salariat:

```
-- Exemplu secventa pentru id salariat --
CREATE SEQUENCE seq_salariat
INCREMENT by 5
START WITH 5000000000
MAXVALUE 59999000000
NOCYCLE;
```

Exemplu de cerere folosita la exercitiul 11:

```
-- Afisati numele produselor,numele liderilor echipelor care au creat produsele si codul clientului care a comandat produse tip laptop: --
select p.denumire_produ, s.num, c.id_client
from salariat s join salariat mg on(s.cnp_salariat = mg.cnp_salariat)
    join echipa e on (mg.cnp_salariat=e.lider)
    join produs p on (p.id echipa = e.id echipa)
    join comanda cm on (cm.cod_fabricatie=p.cod_fabricatie)
    join client c on (cm.id_client=c.id_client)
where upper(p.tip_produ)= 'LAPTOP';
```

Pentru a vedea tot codul folosit si toate subcererile va rog sa verificati celelalte fisiere text.