Proiect Baze de Date

Profesor coordonator

Silviu Laurențiu Vasile

Student

Ulmeanu Cristian

Grupa 253

Managerierea unei companii auto

Profesor coordonator

Silviu Laurențiu Vasile

Student

Ulmeanu Cristian

Grupa 253

Cuprins

1. Prezentarea modelului și a regulilor acestuia......................................................................4
   1. Descrierea proiectului..............................................................................................4
   2. Impactul unei baze de date într-o companie auto…................................................5
   3. Regulile generale.....................................................................................................6
2. Diagrama entitate-relație......................................................................................................8
   1. Prezentarea schemei.................................................................................................8
   2. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților.................9
3. Diagrama conceptuală........................................................................................................16
   1. Prezentarea schemei...............................................................................................16
   2. Descrierea constrangerilor.....................................................................................17
   3. Crearea de tabele....................................................................................................24
   4. Crearea de constrangeri..........................................................................................26
   5. Inserarea de date....................................................................................................30
4. Prezentarea modelului și a regulilor acestuia
   1. Descriere Proiect

Industria auto își are începutul încă din anii 1860, aceasta arătându-și impactul asupra oamenilor încă din perioada respectiva, și a demonstrat că este un ajutor și avantaj pentru oameni extrem de util. Din aceste considerente, industria a continuat să evolueze cu pași repezi din acele timpuri până în prezent, aceasta fiind în continuă dezvoltare tehnologică. Nu știu daca lumea contemporana și-ar putea imagina traiul fără aceste autovehicule.

Pentru a pune într-o perspectiva popularitatea acestui domeniu putem lua în considerare numerele de producție, estimându-se în anul 2019 realizarea a 92 de milioane de autovehicule un număr care se păstrează aproximativ de 5 ani. Un alt exemplu semnificativ este un studiu realizat în 2007 care arată că pe șosele la timpul respectiv existau în jur de 800 de milioane de autovehicule active.

Una din cele mai vechi companii în domeniul auto este „Ford Motor Company” cu o istorie încă din anul 1903. Datorită modului său de lucru, inovației, managementul și conducerii companiei a dus la supraviețuirea cât și prosperitatea companiei până în zilele noastre, aceasta fiind a 5-a cea mai mare companie din acest domeniu din lume. În acest proiect aș dori să intru puțin în partea manageriala a unei companii auto.

Motivația alegerii acestei teme, este strâns legată de pasiunea mea pentru industria auto, din acest motiv am dorit a crea o legătură între pasiune și realizarea proiectului și am ales crearea unei baze de date specifice domeniului auto, tema fiind managementul unei companii auto.

* 1. Impactul unei baze de date într-o companie auto

Pentru funcționarea corectă a unei companii auto aceasta are nevoie de o bază de date bine definită, ușor de accesat, folosit și întreținut. Din acest motiv am ales crearea unei baze de date care va ține evidența unor factori cheie a unei companii: angajații acesteia, mașinile produse, fabricile unde sunt produse acestea, colaboratorii în materie de piese, showroomurile unde sunt vândute mașinile și clienții companiei.

Consider că baza de date trebuie să aibă legături clare, cu toate componentele necesare pentru crearea tabelelor și lipsa creării unor confuzii rezultate din atributele alocate. Aceasta va avea nevoie să poată înregistra toate datele necesare și schimbarea acestora la nevoie pe întreaga durata de viață a companiei.

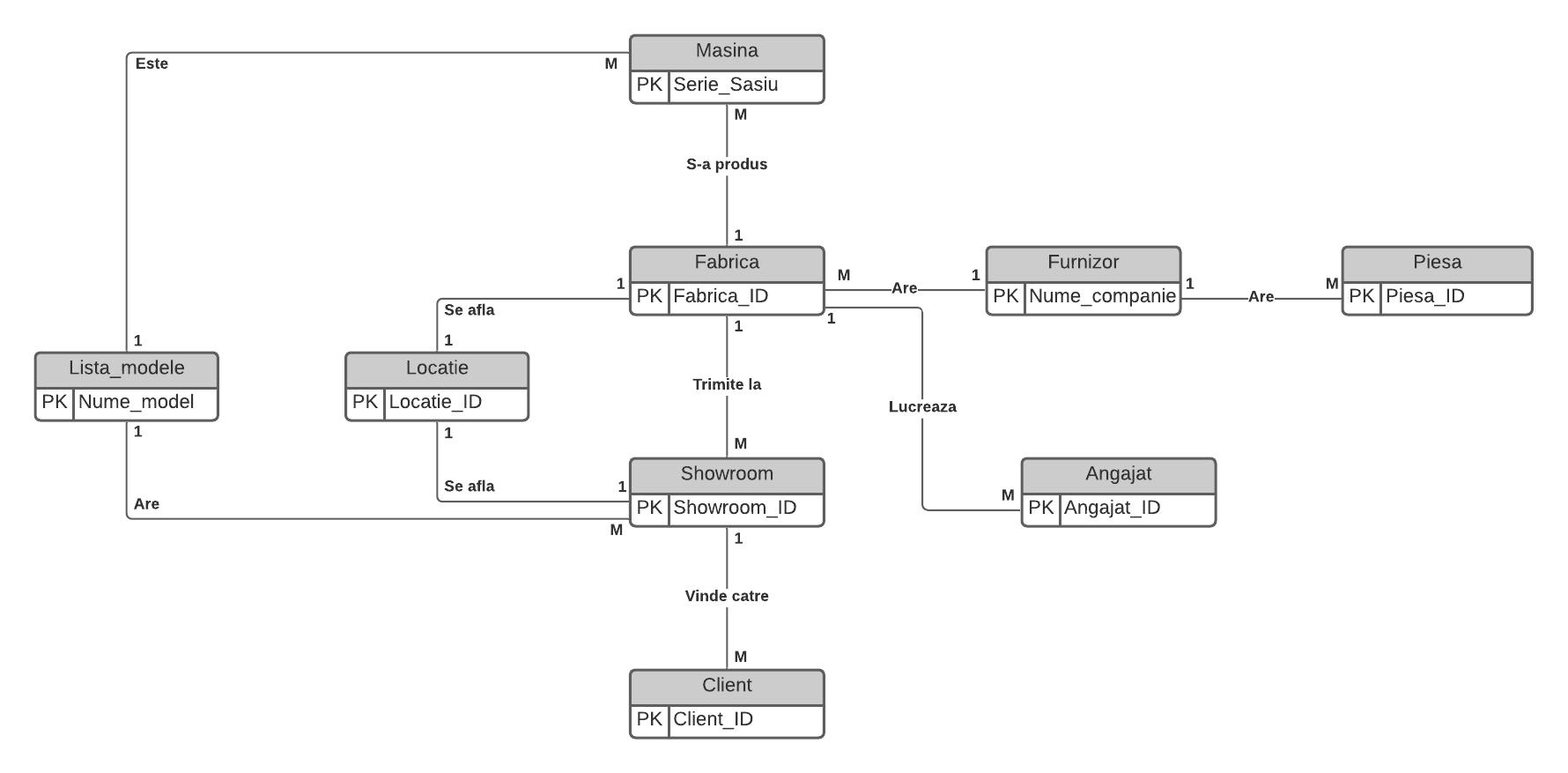
Baza de date va avea un rol crucial în menținerea relațiilor cu clienții săi, putând să rețină datele acestora pentru o colaborare cât mai lipsită de probleme. Totodată, va fi nevoie de gestionarea angajaților într-un mod corespunzător companiei, cunoscându-se detalii importante despre aceștia și alocarea de roluri corespunzătoare fiecărei persoane în companie.

Lucrurile precedente fiind legate de partea umană a companiei, acum pe partea de produse, baza de date trebuie să poată gestiona toate mașinile realizate, reținând date specifice pentru fiecare mașina în parte, ce pot fi utilizate atât pentru partea de construire și proiectare a acestora, cât și pe partea de vânzare. Fiecare mașina este distribuită mai întâi unei fabrici pentru fabricarea acesteia ca mai apoi să fie trimisă mai departe pentru vânzare într-un showroom.

Ultima componentă principală ce va fi definitorie bazei de date este fabricarea mașinilor, unde se va gestiona atât informații despre locul unde sunt realizate aceste mașini, costurile de producție, cât și furnizorii de materiale necesare.

* 1. Reguli generale ale bazei de date
* Entitatea „Angajat” are rolul de a reține toți angajații din companie. Aceștia pot lucra într-o singură fabrică a companiei drept „mecanic”, „vopsitor”, „inginer auto” sau pot exista angajați care nu lucrează în fabrică, ocupând funcții de conducere și de management a firmei, precum „CEO” (Chief Executive Officer), „COO” (Chief Operating Officer), „CFO” (Chief Financial Officer). În cazul în care un angajat lucrează în fabrică FK-ul ce face legătură între „Fabrică” și „Angajat” (Fabrică\_ID), va avea valoarea fabricii unde lucrează. În cazul în care angajatul este de exemplu „președinte” acesta va avea FK ce face legătură cu „Fabrică” „NULL”.
* Entitatea „Locație” este una opțională creată în special pentru o organizare mai bună a tabelului având informația cât mai structurată și specifică pentru fiecare entitate. Deși ea realizează relații de 1 la 1 cu entitățile „Fabrică” respectiv „Showroom” deoarece o fabrică și un showroom nu se pot afla în același loc și o locație este atribuită doar unei singure fabrici sau unui singur showroom, este folosită pentru a nu aglomera cele 2 entități cu atribute mai puțin specifice lor. Din acest motiv un „Locație\_ID” va corespunde unei fabrici prin „Fabrică\_ID” sau unui showroom prin „Showroom\_ID”. Dacă locația corespunde unei fabrici atunci FK-ul ce asigura legătura cu entitatea „Showroom” („Showroom\_ID”) va fi „NULL” și invers dacă locația corespunde unui showroom.
* În cadrul companiei și flotei de mașini, fabricarea unei singure mașini va fi realizată doar de o singură fabrică, acest lucru însemnând că mașina cu „serie\_șasiu” X va fi fabricată doar în fabrică cu „Fabrică\_ID” Y iar procesul fabricării acesteia nu va trece prin mai multe fabrici.
* Toate mașinile salvate în „ Mașină” va reprezenta întreaga flota de mașini pe care firma le-a produs, cu criteriul că acestea vor fi doar cele care sunt produse în serie, nu vor există mașini care sunt încă în stadiul de proiect sau prototip sau sunt ediții limitate create pentru un singur client în mod special și să fie unicate .
* Compania va avea în producție doar mașini care funcționează pe combustie internă, adică nu vor exista mașini care să aibă în coloana „Combustibil” un tip diferit de combustibil în afară de „Diesel” și „Benzină”.
* În tabelul „Piesă” se vor reține tipuri (modele) de piesă și nu se va reține fiecare piesă în parte. Piesele care sunt trecute au rolul de a descrie un furnizor, adică ce tip de piese un furnizor poate furniza. Deci în cazul în care pentru fabricarea mașinilor din fabrică X este nevoie de piesă A și B, această fabrică va avea un furnizor ce are în portofoliu acele tipuri de piese. Aceste piese vor reprezenta catalogul furnizorului. Dacă un furnizor a lucrat cu o anumită piesă, de exemplu: „Motor Diesel” sau „Jante TE37” și fabrică are nevoie de un motor diesel sau un anumit tip de jante pe care fabrică nu le poate confecționa, această poate apela la un anumit furnizor pentru a procura/produce acele componente necesare.
* Dacă o fabrică are un furnizor ce nu asigură toate piesele componente unei mașini, înseamnă că respectivele componente nefurnizate sunt confecționate direct în fabrică. Acest lucru nu va însemna faptul că mașinile dintr-o fabrică vor avea alte specificații sau componente. În cazul în care o fabrică are nevoie de „Motor Diesel” apelează la furnizorul care are în palmares această componentă și se asigură că Furnizorul îi va procura motorul dorit (standard al mașinii) de care este nevoie. Pentru o explicație mai amănunțită, dacă mașina X are nevoie de un motor diesel 1.6 cu anumite specificații, în Fabrică X acest motor poate fi produs în casă, dar în cazul Fabricii Y lucrul acesta nu este posibil, așadar Fabrica Y apelează la Furnizorul X care are în palmares faptul că poate procura motoare diesel, fabrica îi comunică ce fel de motor diesel are nevoie și furnizorul este responsabil de procurarea acestuia exact după standardele impuse de fabrică.
* Un showroom se ocupă cu vânzarea unui singur model de mașină, nu vor exista showroomuri care vând mai multe modele de mașini, totuși vor exista mai multe showroomuri ce vor vinde același model de mașină. Totodată showroomul va primi aceste modele de la fabrică iar pentru a nu exista problema ca un showroom să vândă un tip de mașini pe care fabrica care se ocupă cu livrarea de mașini către acel showroom nu produce acel model, acest lucru va fi verificat separat de fiecare dată când unui showroom i se alocă un model pentru vânzare.
* Modelele de mașini ce vor fi salvate în entitatea „Listă\_modele” vor reprezenta toate modelele pe care compania le-a produs până în prezent.
* Numerele de telefon ce sunt salvate în baza de date nu vor conține prefixele țărilor.
* În cazul în care o persoană are în nume/prenume, simboluri cum ar fi „-” sau „ ’ ” sau „ . ” acestea nu vor fi trecute și vor fi înlocuite cu spațiu.
* Din tabelele „Listă\_modele” și „Mașină” nu se vor șterge date, doar se pot adăuga/modifica, acest lucru fiind determinat de rolul tabelelor, ele având totodată rol de istoric. Adică dacă un model a fost produs în trecut și s-a oprit producerea lui, detalii minime despre acesta vor fi salvate în caz că se dorește reînvierea modelului respectiv, lucru destul de comun în momentul de față. Din acest motiv nu am adăugat constrângere de ștergere (ON DELETE SET NULL/ON DELETE CASCADE) pe „Listă\_modele” din „Mașină” și „Showroom”.

1. Diagrama entitate-relație
2. Prezentare schemă



1. Descrierea entităților, atributelor, cheilor, relațiilor și a cardinalităților

* Mașină

Descrierea entitate:

Entitatea „Mașină” reprezintă mașina pe care compania o are în producție.

Descriere atribute:

* + Dată\_producție: Va reprezenta data în care mașina si-a început procesul de fabricare. Anul poate fi diferit pentru mașini cu același model deoarece un model poate începe producția pe 1 Ianuarie 1991, dar deoarece producția a durat pana în 2020 pot exista mașini cu modelul X fabricată la 2 Februarie 2018 sau 2 Noiembrie 2020. Atât timp cât data este mai mică decât ziua curentă și mai mare decât data apariției modelului.
  + Motorizare: Atributul acesta va reprezenta capacitatea cilindrică a motorului de ex: 1.4, 2.0, 1.9, etc.
  + Combustibil: Mașinile vor putea fi împărțite între Benzina și Diesel.
  + Tip\_caroserie: Va însemna ce tip de mașină este acesta: Sedan, SUV, Cabrio, etc.
  + Tracțiune: Va reprezenta câte roți ale mașinii vor opera: Față, Spate sau Integral.

Chei:

* + Serie\_sașiu(PK): Identificatorul unic pentru fiecare mașină în producție a companiei. Acesta va putea fi folosit pe viitor de clienți pentru a căuta date despre mașina respectivă.
  + Fabrică\_ID(FK): Formează relația cu entitatea „Fabrică”.
  + Nume\_model(FK): Formează relația cu entitatea „Listă\_modele”.

Relații si cardinalități:

* + Cu entitatea „Fabrică”: Cel puțin o mașină a fost sau este alocata unei singure fabrici unde se va realiza fabricarea acesteia.
  + Cu entitatea „Listă\_modele”: Cel puțin o mașină reprezintă un singur model din lista de modele.
* Fabrică

Descrierea entitate:

Entitatea „Fabrică” reprezintă locul unde mașina este fabricată, incluzând asamblarea, vopsirea, tapițarea și controlul calității.

Descriere atribute:

* + Dimensiune: Va reprezenta suprafața în m2 a fabricii.
  + Putere\_asamblare: Va reprezenta o estimare a câte unități se pot produce într-un an.
  + Linii\_asamblare: Va reprezenta numărul de linii de asamblare de care dispune fabrica.
  + Număr\_angajati: Va însemna numărul de angajați de care are nevoie fabrica pentru a funcționa eficient.

Chei:

* + Fabrică\_ID(PK): Identificatorul unic pentru fiecare fabrică.
  + Nume\_Companie(FK): Realizează legătura cu entitatea „Furnizor”.

Relații si cardinalități:

* + Cu entitatea „Mașină”: O fabrică va primi cel puțin o mașină din entitatea „Mașină” pentru a fi produsă. Adică o relație de 1(Fabrică) to many(Mașină), unde mașinile W, X se fabrică în fabrica A, și mașinile Y, Z în fabrica B.
  + Cu entitatea „Locație”: O fabrică se va afla într-o singură locație.
  + Cu entitatea „Furnizor”: Cel puțin o fabrică va avea un singur furnizor pentru a îi furniza piesele necesare producției de mașini.
  + Cu entitatea „Angajat”: Într-o fabrică vor lucra mai mulți angajați.
  + Cu entitatea „Showroom”: O fabrică va trimite mașinile realizate către cel puțin un Showroom pentru vânzare.
* Locație

Descrierea entitate:

Entitatea „Locație” reprezintă locația geografica unde se afla o fabrică sau un showroom.

Descriere atribute:

* + Țara: Va reprezenta țara amplasamentului respectiv..
  + Stat: Va reprezenta statul/județul (în funcție de țară) amplasamentului respectiv.
  + Oraș: Va reprezenta orașul amplasamentului respectiv.
  + Strada: Va reprezenta strada amplasamentului respectiv.
  + Număr: Va reprezenta numărul amplasamentului respectiv.
  + Cod Postal: Va reprezenta codul poștal al amplasamentului respectiv.

Chei:

* + Locație\_ID(PK): Identificatorul unic pentru fiecare locație detaliată.
  + Fabrică\_ID(FK): Formează relația cu entitatea „Fabrică”.
  + Showroom\_ID(FK): Formează relația cu entitatea „Showroom”.

Relații si cardinalități:

* + Cu entitatea „Fabrică”: O singura locație este atribuită unei singure fabrici.
  + Cu entitatea „Showroom”: O singura locație estre atribuită unui singur showroom.
* Furnizor

Descriere entitate:

Entitatea „Furnizor” reprezintă companiile ce se vor ocupa cu asigurarea de materiale necesare (piese) ale unei fabrici.

Descriere atribute:

* + Nume\_contact: Va reprezenta numele persoanei de contact a firmei de furnizare de piese.
  + Prenume\_contact: Va reprezenta prenumele persoanei de contact a firmei de furnizare de piese.
  + Telefon\_contact: Va reprezenta numărul de telefon pe care compania îl are pentru a contacta persoana de contact a furnizorului.
  + Email\_contact: Va însemna emailul pe care compania îl are pentru a contacta persoana reprezentativa a furnizorului.

Chei:

* + Nume\_Companie(PK): Este identificatorul unic pentru fiecare companie ce se ocupă cu furnizarea de piese.

Relații si cardinalități:

* + Cu entitatea „Fabrică”: Un furnizor furnizează piese către cel puțin o fabrică.
  + Cu entitatea „Piesa”: Un furnizor are cel puțin o piesa în catalog.
* Piesă

Descriere entitate:

Entitatea „Piesa” va reprezenta tipul pieselor pe care un furnizor le are în portofoliu cu ce tip de piese se ocupa un furnizor.

Descriere atribute:

* + Tip\_piesă: Va fi tipul piesei, adică numele tipului de piesa, de exemplu: „motor”, „cauciucuri”, „roți”, „arcuri cât și, opțional, modelul piesei respective de exemplu: „motor diesel” sau „evacuare sport”.
  + Producător: Va reprezenta producătorul piesei.
  + Garanție: Va reprezenta perioada de garanție pe care o piesa o are din momentul furnizării.
  + Preț: Va însemna prețul pe care furnizorul îl cere pentru procurarea unei piese.

Chei:

* + Piesă\_ID(PK): Este identificatorul unic pentru toate piesele dintr-un tip de piesă a unui furnizor.
  + Furnizor\_ID(FK): Creează relația cu entitatea „Furnizor”.

Relații si cardinalități:

* + Cu entitatea „Furnizor”: Cel puțin o piesa va aparține unui singur furnizor.
* Angajat

Descriere entitate:

Entitatea „Angajat” va retine toți angajații companiei, indiferent de locul unde lucrează.

Descriere atribute:

* + Nume: Va reține numele angajatului.
  + Prenume: Va reține prenumele angajatului.
  + Poziție: Va reprezenta postul (poziția) pe care o are angajatul.
  + Salariu: Va fi salariul angajatului.
  + Dată\_angajare: Va reprezenta data la care angajatul s-a alăturat firmei pentru prima dată.

Chei:

* + Angajat\_ID(PK): Este identificatorul unic pentru fiecare angajat al companiei în parte.
  + Fabrică\_ID(FK): Face legătura cu entitatea „Fabrică”.

Relații si cardinalități:

* + Cu entitatea „Fabrică”: Va exista cel puțin un angajat care lucrează într-o singură fabrică.
* Showroom

Descriere entitate:

Entitatea „Showroom” va reține datele tuturor showroomurilor pe care compania le deține.

Descriere atribute:

* + Target\_Vânzări: Va însemna numărul de bani pe care trebuie sa îi obțină minim un showroom pentru a fi considerat eficient.
  + Începere\_program: Va reprezenta ora la care se deschide showroomul.
  + Sfârșit\_program: Va reprezenta ora de închidere a showroomului.

Chei:

* + Showroom\_ID(PK): Este identificatorul unic pentru fiecare showroom deținut de companie.
  + Fabrică\_ID(FK): Asigură relația cu entitatea „Fabrică”.
  + Nume\_Model(FK): Asigură legătura cu entitatea „Listă\_modele”.

Relații si cardinalități:

* + Cu entitatea „Locație”: Un showroom se poate afla într-o singură locație.
  + Cu entitatea „Fabrică”: Cel puțin un showroom primește mașini de la o fabrică.
  + Cu entitatea „Client”: Un showroom va avea mai mulți clienți.
  + Cu entitatea „Listă\_modele”: Cel puțin un showroom vinde un tip de model.
* Client

Descriere entitate:

Entitatea „Client” va salva detalii despre toți clienții pe care i-a avut o companie. Dacă același client va cumpăra de doua ori, acesta va primi 2 ID-uri diferite.

Descriere atribute:

* + Nume: Va reține numele clientului.
  + Prenume: Se va salva prenumele clientului.
  + CNP: Va reprezenta codul numeric personal al clientului.
  + Număr\_mașini: Va reține numărul de mașini pe care le-a cumpărat clientul.
  + Număr\_telefon: Va reprezenta un număr de telefon pentru contactarea clientului.
  + Email: Va reprezenta emailul de contact al clientului.

Chei:

* + Client\_ID(PK): Este identificatorul unic pentru fiecare client care cumpăra o mașină. Daca aceeași persoană cumpăra de mai multe ori va fi adăugat din nou cu un ID nou, dar același set de date.
  + Showroom\_ID(FK): Va face legătura cu entitatea „Showroom”.

Relații si cardinalități:

* + Cu entitatea „Showroom”: Mai mulți clienți vor cumpăra mașini dintr-un singur showroom.
* Listă\_modele

Descriere entitate:

Entitatea „Listă\_modele” va salva detalii despre modelele de mașini pe care le produce fabrică.

Descriere atribute:

* + Dată\_apariție: Va reține data în care a apărut modelul de mașină pentru prima oară.
  + Generații: Va reține câte generații de modelul respectiv au existat până în momentul respectiv.

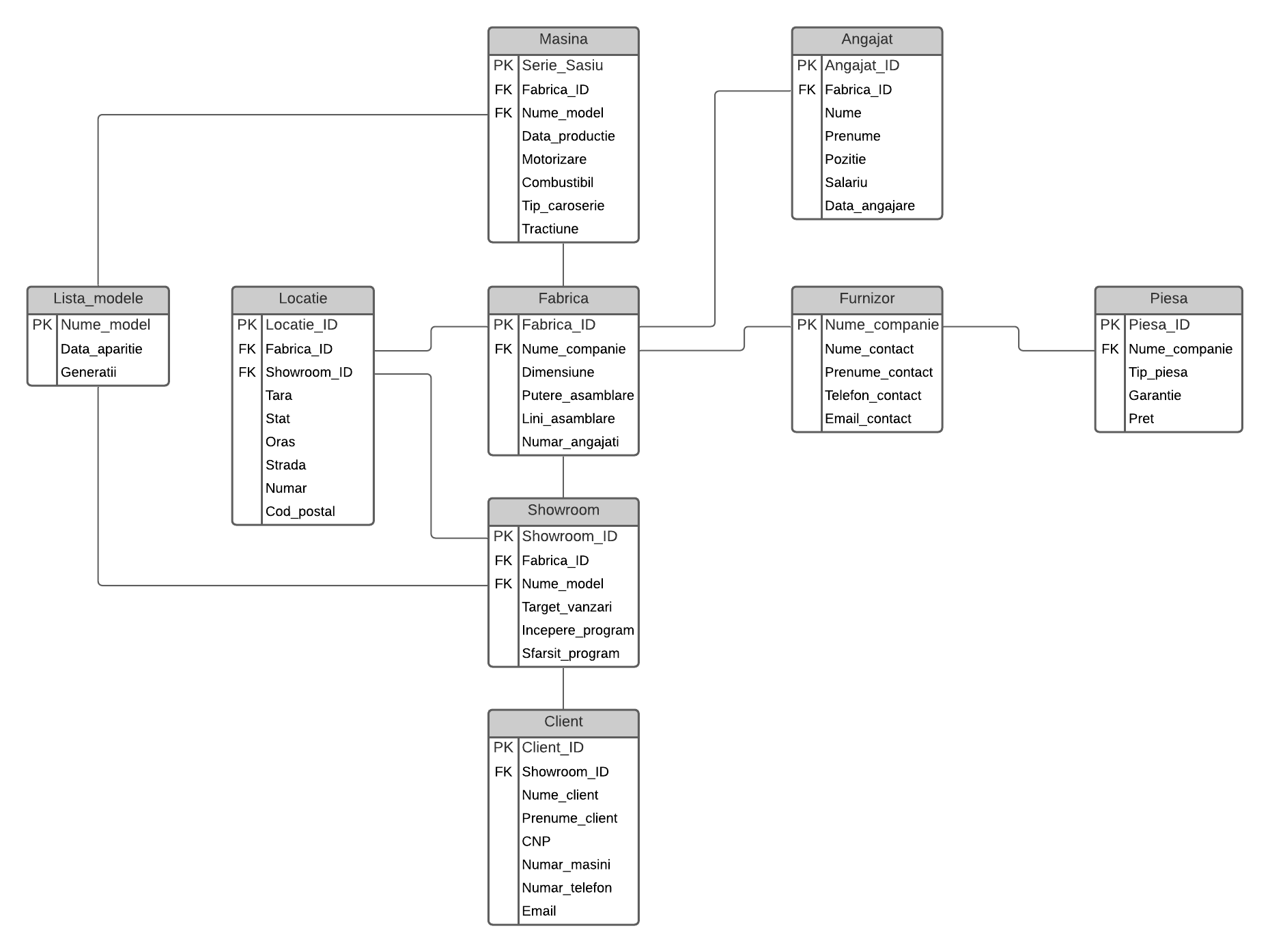
Chei:

* + Nume\_model(PK): Este identificatorul unui singur

Relații si cardinalități:

* + Cu entitatea „Mașină”: Un model aparține mai multor mașini.
  + Cu entitatea „Showroom”: Un model de mașină este vândut în mai multe showroomuri.

1. Diagrama conceptuala
   1. Prezentarea schemei



* 1. Descrierea constrângerilor si a schemelor relaționale
* Mașină

Pentru entitatea mașină am creat următoarele constrângeri:

* Serie\_Șasiu va fi primary key. Acesta va avea o constrângere să fie format din 17 caractere deoarece seria de șasiu are întotdeauna 17 caractere.
* Fabrică\_ID poate fi și NULL în cazul în care o mașină a fost fabricată într-o fabrică la care compania a renunțat. Fabrica\_ID va reprezenta foreign key cu referință pentru Fabrică\_ID din tabelul Fabrică, și va avea on delete set null, în cazul în care se va șterge o fabrică Fabrică\_ID va deveni null(ON DELETE SET NULL).
* Nume\_model va fi NOT NULL deoarece o mașină trebuie să fie obligatoriu un model, daca ar fi null ar însemna ca mașina nu are un model ceea ce e imposibil, orice mașină are un model. Totodată acesta va fi foreign key cu referință pentru Nume\_model din tabelul Listă\_modele.
* Dată\_producție va fi NOT NULL deoarece o mașină când va fi adăugată în tabel va avea obligatoriu data când aceasta a fost produsă, dacă ar fi null ar însemna că mașina nu a fost produsă, deci nu ar avea cum să existe. Printr-un trigger am adăugat constrângere pe data pentru a fi mai mare decât data când a apărut modelul mașinii respective. (Dată\_producție>Dată\_apariție din tabelul Listă\_modele) și să nu fie mai mare decât data curentă(Dată\_producție<sysdate) deoarece o mașină nu poate fi produsă în viitor.
* Motorizare va fi NOT NULL deoarece o mașină trebuie să aibă obligatoriu un motor cu o capacitate cilindrică, totodată printr-o constrângere ce folosește regex voi verifica ca formatul motorizării să fie [1-9].[0-9] sau 0.[1-9] pentru cazul în care capacitatea cilindrica e mai mică de 1.0 sau [1-9] daca capacitatea cilindrica este de tip 1.0/2.0 (se termină cu 0), deoarece daca tipul de dată este Number acesta va reține doar 2 în cazul în care este introdus 2.0.
* Combustibil va fi NOT NULL deoarece mașina funcționează pe un combustibil, iar ca sa restricționez inserturile printr-o constrângere ce verifica dacă insertul este fie „Diesel” fie „Benzina” acestea fiind singurele tipuri de combustibil pe care le pot avea mașina.
* Tip\_caroserie va fi NOT NULL deoarece mașina trebuie sa se încadreze într-un tip de șasiu, dacă ar fi null ar fi o mașină fără șasiu care nu e posibil. Restricționez inserturile printr-o constrângere ce verifică dacă insertul este unul din următoarele: „Berlina”, „Break”, „Coupe”, „Cabriolet”, „Limuzină”, „Roadster”, „Pick-up”, „SUV”, „Sedan”, „Hatchback”. Acestea fiind caroserii comune în momentul de față, pe viitor acest constraint se poate extinde și pe caroserii diferite.
* Tracțiune va fi NOT NULL deoarece mașina trebuie sa aibă o tracțiune daca ar fi null ar însemna că mașina nu va avea tracțiune, acest lucru fiind imposibil. Printr-o constrângere mă voi asigura că datele introduse pot fi doar „Integrală”, „Față” și „Spate”, acestea fiind singurele sisteme de tracțiune utilizate în masă în momentul de față.
* Listă\_modele

Pentru entitatea listă\_modele am creat următoarele constrângeri:

* Nume\_model va fi primary key.
* Dată\_apariție va fi NOT NULL deoarece când un model apare și va fi inserat în baza de date trebuie sa aibă reținut data când acest model a fost arătat publicului daca data ar fi null ar însemna că modelul nu a apărut niciodată. Printr-un trigger am creat constrângere asupra acestuia pentru a verifica că data nu are cum sa fie mai mare decât cea curentă, deoarece listă\_modele va conține modelele pe care le are compania până în prezent.
* Generații va fi NOT NULL deoarece un model va avea cel puțin o generație dacă generația va fi null, nu va exista modelul. În momentul când apare un model pentru prima data acesta va fi prima generație din acel model.
* Fabrică

Pentru entitatea fabrică am creat următoarele constrângeri:

* Fabrică\_ID va fi primary key.
* Nume\_companie va putea fi NULL deoarece poate exista o fabrică care nu are nevoie de un furnizor de piese putând să și le confecționeze pe toate necesare în interiorul fabricii. Totodată va reprezenta foreign key cu referință către Nume\_companie din entitatea Furnizor. În cazul în care se va șterge un furnizor atașat fabricii, numele companiei din fabrică va deveni null(ON DELETE SET NULL) deoarece dacă o companie renunță la a menține legătura cu un furnizor înseamnă că nu mai are nevoie de compania respectivă pentru a furniza materiale necesare, dar nu va însemna ca nu va mai avea fabrică compania.
* Dimensiunea va fi NOT NULL deoarece fabrica va reține obligatoriu dimensiunea fabricii și aceasta nu poate fi null deoarece în cazul acesta nu ar exista. Am mai adăugat și constrângere de dimensiune, aceasta am stabilit că ar trebui sa fie minim 2000m2 fiind o dimensiune minimă pentru o fabrică de a exista.
* Putere\_asamblare va fi NOT NULL deoarece compania va avea nevoie să știe la ce sa se aștepte că fabrica va produce. Insertul nu poate fi null deoarece compania are nevoie să aibă o estimare cât poate produce fabrica pentru a decide dacă o mai păstrează sau nu. Am adăugat o constrângere pe putere aceasta să fie mai mare decât 0 deoarece puterea de asamblare nu poate fi negativă.
* Linii\_asamblare va fi NOT NULL deoarece dacă o fabrică auto pentru a funcționa are nevoie de cel puțin o linie de asamblare. Ca să mă asigur de acest lucru am adăugat o constrângere ce verifică că numărul liniilor să fie mai mare decât 0.
* Număr\_angajați va fi NOT NULL deoarece o fabrică ca să funcționeze are nevoie de un minim de angajați, iar acel minim l-am creat printr-o constrângere care să verifice ca numărul sa fie mai mare de 50. Acest minim va reprezenta numărul cel mai mic de angajați pentru a funcționa, nu neapărat cât să funcționeze eficient.
* Locație

Pentru entitatea locație am creat următoarele constrângeri:

* Locație\_ID va fi primary key.
* Fabrică\_ID va putea fi NULL și totodată UNIQUE deoarece poate exista o locație ce nu aparține unei fabrici ci unui showroom și totodată în cazul în care există o fabrică în locația respectivă aceasta trebuie să fie una singură neputând să existe mai multe fabrici la aceeași adresă. Atributul va fi foreign key cu referință la Fabrică\_ID din Fabrică. În cazul în care se va șterge fabrica care se află într-o locație atunci se va șterge și locația respectivă (ON DELETE CASCADE).
* Showroom\_ID va putea fi NULL și totodată UNIQUE deoarece poate exista o locație ce nu aparține unui showroom ci unei fabrici și totodată în cazul în care există un showroom în locația respectivă acesta trebuie să fie unul singur neputând să existe mai multe showroomuri la aceeași adresă. Atributul va fi foreign key cu referință la Showroom\_ID din entitatea Showroom. În cazul în care se va șterge showroomul care se afla într-o locație atunci se va șterge și locația respectivă (ON DELETE CASCADE).
* Pentru cele două, am mai creat o constrângere care vizează ca cele 2 iduri să nu poată fi NULL sau NOT NULL în același timp pentru a mă asigura că există ceva în locația respectivă.
* Țara va fi NOT NULL deoarece locația trebuie să aparțină unei țări. Pe lângă acest lucru am adăugat o constrângere folosind regex ca țara să fie formată doar din litere.
* Stat va fi NOT NULL deoarece în funcție de țară o locație aparține obligatoriu unui stat/județ/regiune, totodată am adăugat o constrângere similară cu cea a țării aceea de a avea doar litere în nume.
* Oraș va fi NOT NULL deoarece locația trebuie sa se afle obligatoriu într-un oraș și similar cu țara și stat am adăugat constrângere ca orașul să fie format doar din litere.
* Strada va fi NOT NULL deoarece amplasamentul trebuie sa se afle pe o stradă.
* Număr va fi NOT NULL deoarece amplasamentul e obligatoriu să aibă un număr. Totodată, am adăugat o constrângere care verifică formatul numărului folosind regex, acesta să fie format doar din cifre sau să conțină cifre apoi litera în cazul în care numărul este de forma „13A”.
* Cod\_Poștal va fi NOT NULL deoarece amplasamentul trebuie să aibă un cod poștal.
* Am adăugat și o constrângere generală, care verifică că în tabel combinația de Țară, Stat, Oraș și Stradă să fie UNIQUE, astfel nu vor exista 2 fabrici/showroomuri pe aceeași stradă deoarece mi se pare o greșeală din punct de vedere infrastructural.
* Furnizor

Pentru entitatea furnizor am adăugat următoarele constrângeri:

* Nume\_companie va fi primary key.
* Nume\_contact va fi NOT NULL deoarece compania va avea nevoie să retină un contact pentru furnizor pentru a putea face legătura cu acesta. Am adăugat o constrângere ce verifică ca numele să fie format doar din litere.
* Prenume\_contact va fi NOT NULL pentru a avea numele complet al persoanei de contact. Am creat aceeași constrângere ce verifică și prenumele să fie format doar din litere.
* Telefon\_contact va fi NOT NULL pentru a salva una din metodele de contact ale furnizorului și totodată UNIQUE deoarece numerele de telefon inserate sunt românești, deși nu mai adaug prefix acestea for fi unice pentru fiecare persoana în parte. Pe lângă aceasta am creat o constrângere care să verifice ca numărul de telefon să conțină doar cifre si încă una ce verifică că numărul de telefon conține 10 cifre acesta fiind numărul de cifre pe care le au numerele de telefon romanești.
* Email\_contact va fi NOT NULL reprezentând metoda de contact de rezerva a furnizorului si totodată UNIQUE deoarece emailul este diferit de la individ la individ.
* Am mai creat o constrângere ce verifica folosind regex formatul mailului pentru a fi unul valid.
* Piesă

Pentru entitatea piesa am adăugat următoarele constrângeri:

* Piesă\_ID va fi primary key.
* Nume\_companie va fi NOT NULL deoarece piesele aparțin unui catalog de produse ale furnizorului deci trebuie să aparțină sigur unui furnizor. Totodată este foreign key cu referință la Nume\_companie din entitatea Furnizor. În cazul în care se va șterge un furnizor se vor șterge și toate piesele furnizorului (ON DELETE CASCADE).
* Tip\_piesă va fi NOT NULL deoarece atributul are rolul de a reține ce piesă este aceasta, fiind necesar să se cunoască ce reprezintă piesa respectivă.
* Garanție va putea fi NULL deoarece unii furnizori au opțiunea să nu ofere garanție pentru piesele lor. Am mai adăugat o constrângere că garanția dacă există să fie pozitivă.
* Preț va fi NOT NULL deoarece fiecare obiect trebuie să aibă un preț de catalog, acesta va fi întotdeauna pozitiv printr-o constrângere ce verifică că prețul să fie mai mare decât 0.
* Showroom

Pentru entitatea showroom am adăugat următoarele constrângeri:

* Showroom\_ID va fi primary key.
* Fabrică\_ID va fi NOT NULL deoarece un showroom ca să aibă mașini de vânzare trebuie să le primească de la o fabrică. Atributul va fi foreign key cu referință la Fabrică\_ID din Fabrică. În cazul în care se va șterge o fabrică, idul fabricii va fi modificat în null (ON DELETE SET NULL) care din motiv că IDul nu poate fi NULL acest lucru va rezulta într-o eroare. Am dorit să fac acest lucru pentru a forța faptul că înainte de a șterge o fabrică să modifici toate showroomurile care au fabrica respectiva care furnizează mașina. Am dorit asta deoarece mi se pare ciudat ca în momentul în care o companie renunța la o fabrică să renunțe și la toate showroomurile care primesc mașini de la fabrică respectivă, și mi se pare ciudat ca showroomul să modifice fabrica în null deoarece ar însemna că showroomul funcționează și vinde mașini pe care nu le are deoarece nu are aprovizionare de la nicio fabrică rezultând a avea un showroom fără nimic în el, gol.
* Nume\_model va fi NOT NULL deoarece un showroom trebuie să aibă stabilit ce model de mașină va vinde pentru a exista. Atributul va fi foreign key cu referință la Nume\_model din Listă\_modele.
* Printr-un trigger am mai creat o constrângere ce va verifica dacă modelul pe care îl are showroomul (modelul de mașini vândute de către showroom) să fie fabricat de fabrica care se ocupă cu aprovizionarea de mașini pentru respectivul showroom. Acesta va funcționa în următorul mod declar o variabilă counter pentru a reține numărul de apariții a modelului de mașină dorită pentru showroom în fabrică care se va ocupa cu aprovizionarea, dacă numărul respectiv este 0 înseamnă că nu s-a găsit nicio mașină cu modelul respectiv în fabrică care se ocupă cu aprovizionări deci nu acel model nu va avea cum să existe în showroom.
* Target\_vânzări va fi NOT NULL deoarece compania trebuie să impună un target pentru vânzare pentru a nu renunța la el în cazul în care nu ar fi profitabil. Am adăugat constrângere ce va verifica că targetul să fie un număr pozitiv deoarece targetul nu are sens să fie negativ.
* Începere\_program va fi NOT NULL deoarece showroomul trebuie să aibă o ora la care se deschide magazinul. Am creat constrângere ce verifică forma atributului să fie 00:00-24:00.
* Sfârșit\_program va fi NOT NULL pentru că showroomul trebuie să aibă și o data la care se închide, care va avea o constrângere similară să verifice forma atributului sa fie 00:00-24:00.
* Angajat

Pentru entitatea angajat am adăugat următoarele constrângeri:

* Angajat\_ID va fi primary key.
* Fabrică\_ID va putea fi NULL deoarece un angajat poate avea o funcție de conducere ceea ce înseamnă că acesta nu lucrează în nici o fabrică. Atributul va fi totodată și foreign key fiind referință la Fabrică\_ID din entitatea Fabrică, iar în cazul în care se va șterge o fabrică se vor șterge și angajații acelei fabrici (ON DELETE CASCADE).
* Nume va fi NOT NULL deoarece angajatul va trebui să aibă un nume, pe lângă aceasta am adăugat o constrângere ca numele să conțină doar litere.
* Prenume va fi NOT NULL deoarece angajatul trebuie să aibă numele complet deci va trebui să aibă și un prenume care va fi verificat să conțină doar litere.
* Pozate va fi NOT NULL deoarece angajatul trebuie să ocupe o funcție în companie, aceasta printr-o constrângere va conține doar litere.
* Salariu va fi NOT NULL deoarece angajatul trebuie să aibă un salariu. Va exista o constrângere care va verifica că salariul să fie pozitiv.
* Dată\_angajare va NOT NULL deoarece angajatul va avea sigur o dată când acesta s-a alăturat companiei pentru prima oară. Printr-un trigger am adăugat o constrângere ce va verifica că data de angajare să nu fie mai mare decât data curentă.
* Client

Pentru entitatea client am adăugat următoarele constrângeri:

* Client\_ID va fi primary key.
* Showroom\_ID va putea fi NULL deoarece un client ar fi putut cumpăra o mașină și din afara showroomului, acesta ar putea comanda mașină de pe internet. Atributul va fi foreign key, referință la Showroom\_ID din entitatea Showroom. În cazul în care se va șterge un showroom clienții cu showroomul respectiv îl vor modifica în null (ON DELETE SET NULL).
* Nume\_client va fi NOT NULL deoarece clientul va trebui să aibă un nume, pe lângă aceasta am adăugat o constrângere ca numele să conțină doar litere.
* Prenume\_client va fi NOT NULL deoarece clientul trebuie să aibă numele complet deci va trebui să aibă și un prenume care va fi verificat să conțină doar litere.
* CNP va fi NOT NULL deoarece orice persoana are un CNP și totodată UNIQUE deoarece CNP este ce specific fiecărui individ în parte, la care am adăugat constrângere care verifică ca numărul de caractere să fie de 13, acesta fiind numărul de caractere standard pentru CNP.
* Număr\_mașini va fi NOT NULL deoarece dacă este client înseamnă că a cumpărat cel puțin o mașină. Am adăugat o constrângere pentru a fi sigur că numărul de mașini cumpărate este mai mare sau egal cu 1.
* Număr\_telefon va fi NOT NULL deoarece trebuie reținut o metoda de contact a clientului și pe lângă va fi UNIQUE deoarece am ales să populez baza de date doar cu numere de telefon din România, deci acest lucru înseamnă că ele vor fi unice pentru fiecare individ. Am adăugat pentru acesta o constrângere ca numărul de telefon să fie format doar din cifre și numărul de cifre să fie de 10 care este standard pentru numerele de telefon din România.
* Email va fi NOT NULL deoarece trebuie să fie reținut și o metoda de rezervă pentru a păstra legătura cu clientul și va fi UNIQUE deoarece mailul este ceva specific unei singure persoane. Am creat o constrângere pentru a verifica prin regex formatul mailului să fie unul valid.
  1. Crearea de tabele
* Mașină

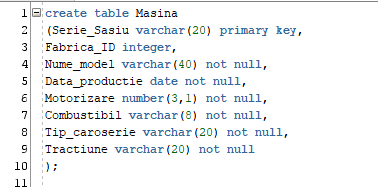


Figura 1. Crearea tabelului „Mașină” si o parte din constrângeri.

* Listă\_modele

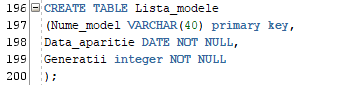


Figura 2. Crearea tabelului „Listă\_modele” si o parte din constrângeri.

* Fabrică

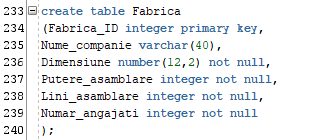


Figura 3. Crearea tabelului „Fabrică” si o parte din constrângeri.

* Locație

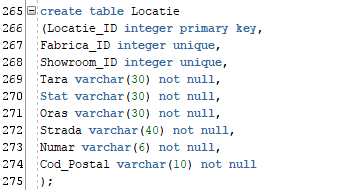


Figura 4. Crearea tabelului „Locație” si o parte din constrângeri.

* Furnizor

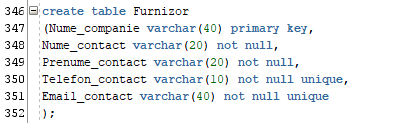


Figura 5. Crearea tabelului „Furnizor” si o parte din constrângeri.

* Piesă

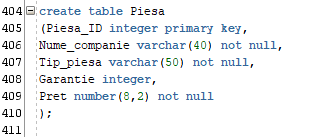


Figura 6. Crearea tabelului „Piesă” si o parte din constrângeri.

* Showroom

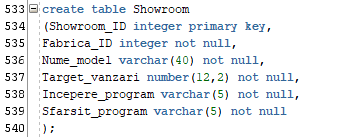


Figura 7. Crearea tabelului „Showroom” si o parte din constrângeri.

* Angajat

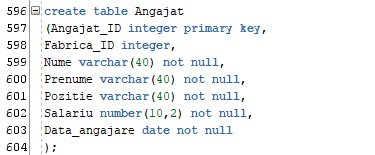


Figura 8. Crearea tabelului „Angajat” si o parte din constrângeri.

* Client

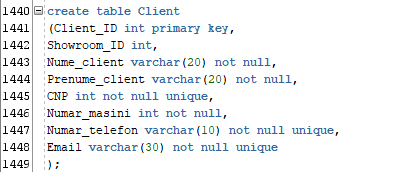


Figura 9. Crearea tabelului „Client” si o parte din constrângeri.

* 1. Implementarea de constrângeri specifice pentru fiecare tabel
* Mașină

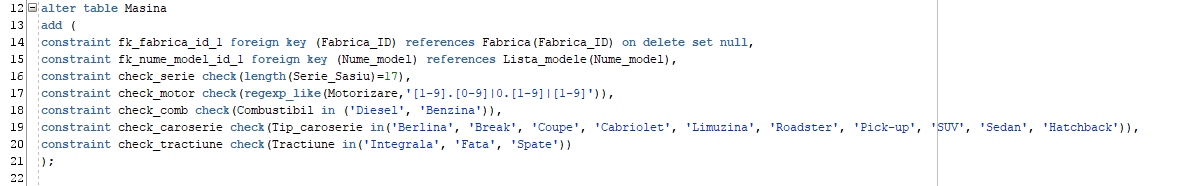


Figura 10. Implementarea constrângerilor specifice pentru tabelul „Mașină”.

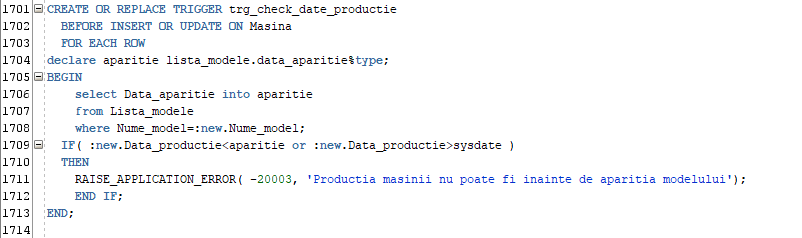


Figura 11. Crearea triggerului pentru a verifica ca data producție sa fie mai mare decât cea a apariției si mai mica decât ziua curenta.

* Listă\_modele

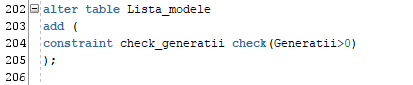


Figura 12. Implementarea constrângerilor specifice pentru tabelul „Listă\_modele”.

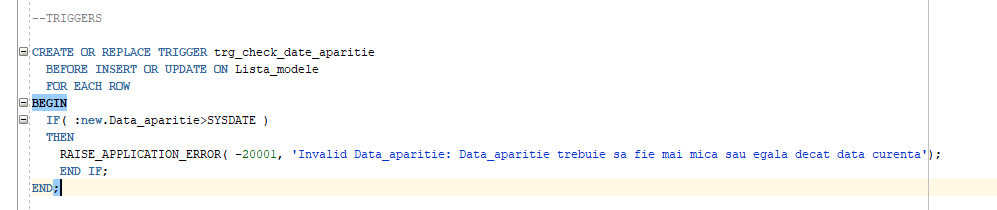


Figura 13. Crearea triggerului ce verifica ca data apariției sa nu fie mai mare decât data curentă.

* Fabrică

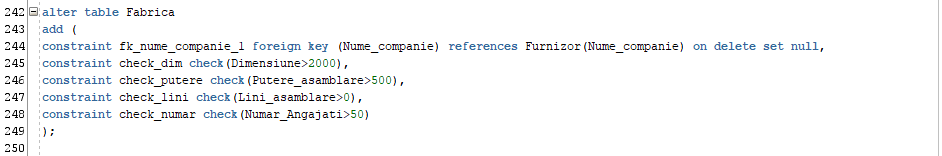


Figura 14. Implementarea constrangerilor specifice pentru tabelul „Fabrică”.

* Locație

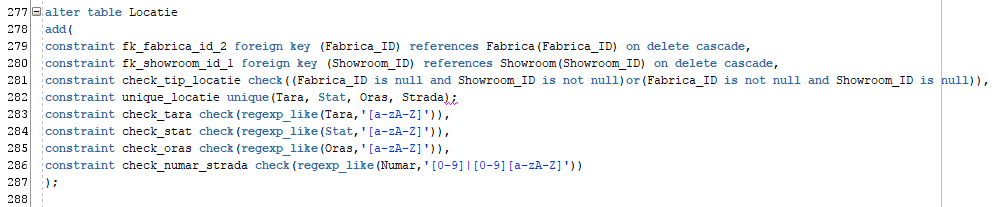


Figura 15. Implementarea constrângerilor specifice pentru tabelul „Locație”.

* Furnizor

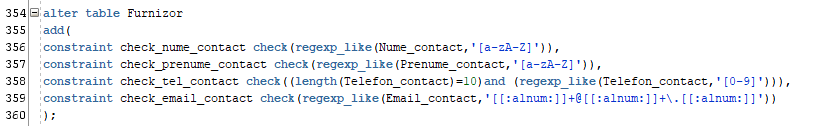


Figura 16. Implementarea constrângerilor specifice pentru tabelul „Furnizor”.

* Piesă

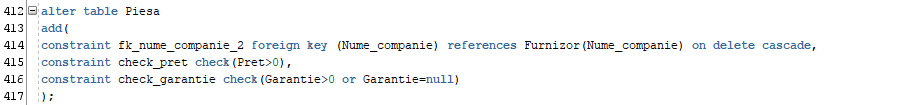


Figura 17. Implementarea constrângerilor specifice pentru tabelul „Piesa”.

* Showroom

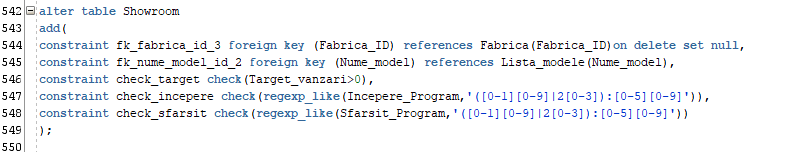


Figura 18. Implementarea constrângerilor specifice pentru tabelul „Showroom”.

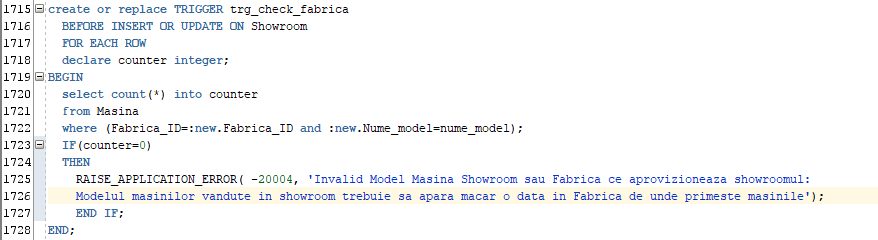


Figura 19. Crearea triggerului ce verifica ca modelul de mașini din showroom să fie produs în fabrica care se ocupa cu aprovizionarea showroomului

* Angajat

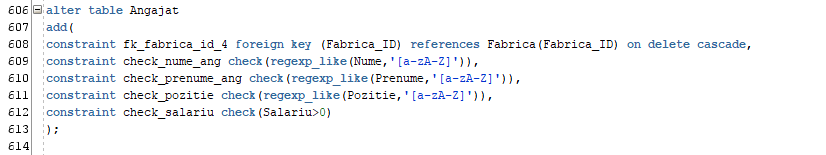


Figura 20. Implementarea constrângerilor specifice pentru tabelul „Angajat”.

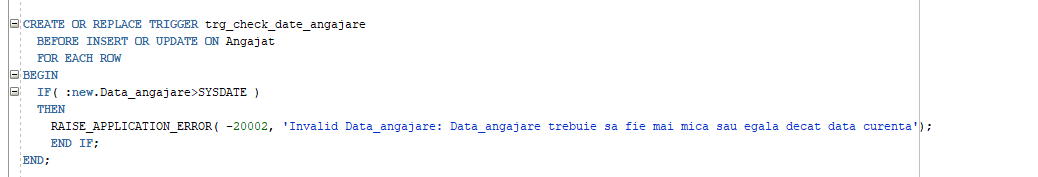


Figura 21. Crearea de trigger ce verifica ca data angajării sa fie mai mica decât data curenta.

* Client

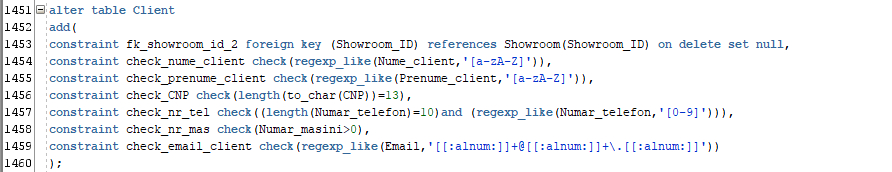


Figura 22. Implementarea constrângerilor specifice pentru tabelul „Client”.

* 1. Inserarea de date in tabel
* Mașină



Figura 23. Inserarea o parte de date in tabelul „Mașină”.

* Listă\_modele



Figura 24. Inserarea de date in tabelul „Listă\_modele”.

* Fabrică

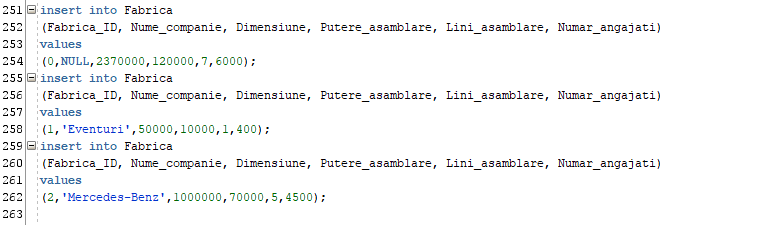


Figura 25. Inserarea de date in tabelul „Fabrică”.

* Locație



Figura 26. Inserare o parte de date in tabelul „Locație”.

* Furnizor



Figura 27. Inserarea de date in tabelul „Furnizor”

* Piesă

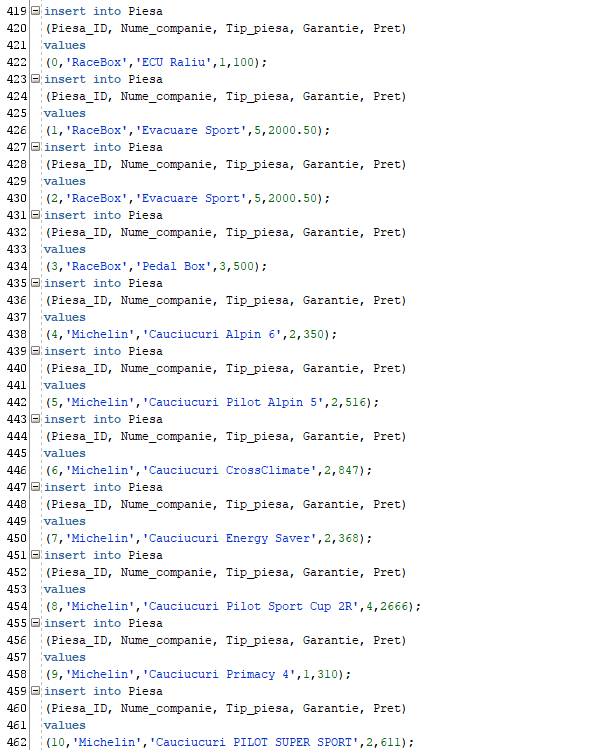


Figura 28. Inserarea de o parte din datele tabelului „Piese”.

* Showroom



Figura 29. Inserarea datelor in tabelul „Showroom”.

* Angajat

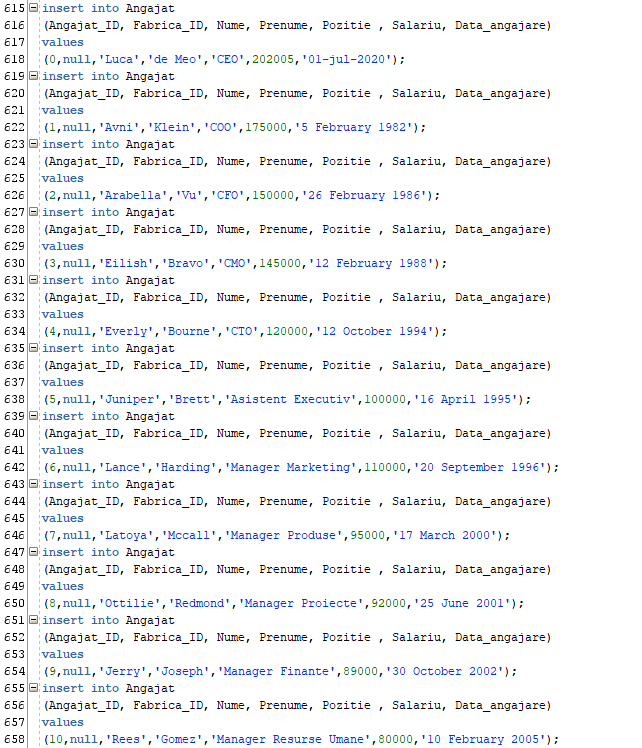


Figura 30. Inserarea o parte din datele tabelului „Angajat”.

* Client



Figura 31. Inserarea o parte de date din „Client”.