**UNIVERSITATEA TEHNICA**

**“GHEORGHE ASACHI” IASI**

**FACULTATEA DE AUTOMATICA SI CALCULATOARE**

**GESTIONAREA LOCURILOR DE PARCARE CONCESIONATE**

**- BAZE DE DATE-**

**Student,**

**Paraschiv Florin-Vladut, 1306B**

**Iasi, 2020**

1. **Descrierea proiectului**

Analiza, proiectarea si implementarea unei baze de date care se modeleze gestionarea unui sistem cu privire la locurile de parcare concesionate.

Informatiile pe care le vom folosi in baza de date sunt legate de persoana, masina si locul de parcare respectiv. In prima instanta vom avea nevoie de informatii legate de persoana care vrea sa rezerve locul de parcare, respectiv un nume, prenume, CNP, serie, numar, adresa, data nasterii si localitatea, informatii ce se gasesc intr-o carte de identitate pe care orice persoana care ar dori sa rezerve un loc de parcare ar trebui sa detina. Pe langa datele sale personale, mai este obligatoriu un numar de telefon, mobil sau fix, si, optional, o adresa de email pentru ca persoana respectiva sa poate fi contactata pentru orice eventualitate.

In a 2-a parte ne vom ocupa de ocuparea propriu-zisa a locului. Pentru aceasta avem nevoie de informatii despre masina acestuia, locatia locului de parcare si plata. La masina, avem nevoie, in mod special, de numarul de inmatriculare pentru trasarea sau plasarea unei pancarde cu numarul de inmatricularea a masinii care va rezerva. Referitor la plata, va fi nevoie de tipul platii, care se poate realiza doar numerar sau cu cardul, suma platita si data la care s-a realizat plata. Dupa ce cele doua operatiuni de mai sus s-au realizat cu succes, urmeaza ocuparea locului, in care avem nevoie de durata ocuparii, care se reflecta din suma platita, si intervalul de timp in care acesta este ocupat.

1. **Structura si inter-relationarea tabelelor**

In proiectare s-au identificat tipurile de **relatii** one-to-one si one-to-many.

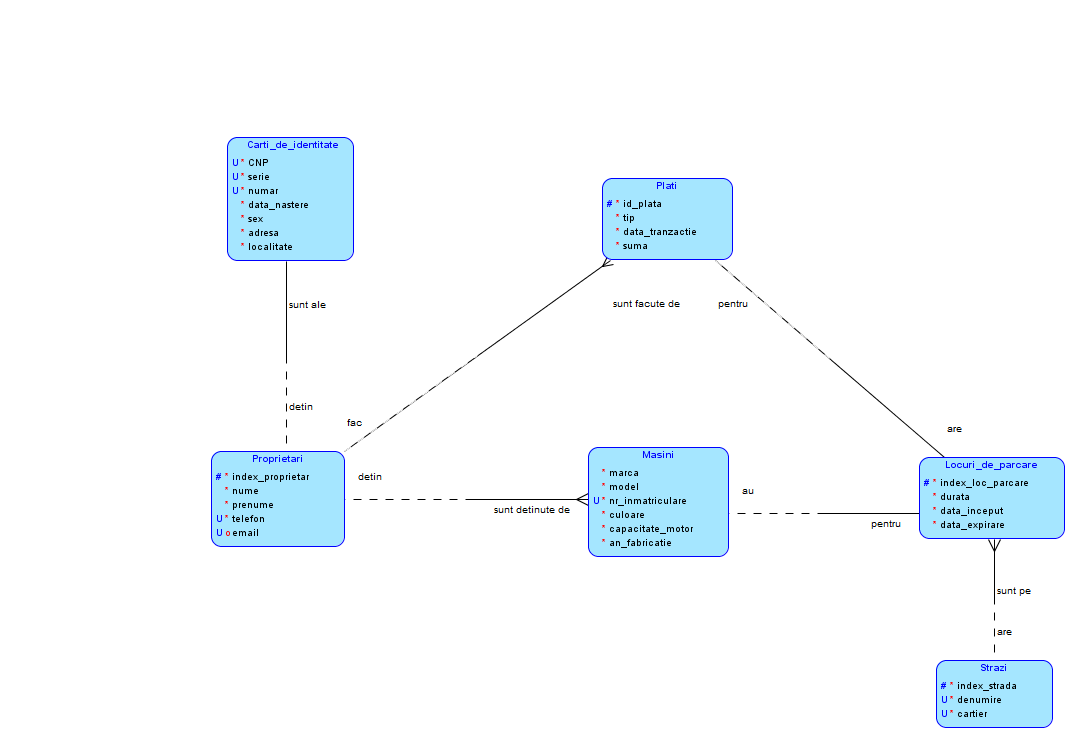
Intre tabelele **carti\_de\_identitate** si **proprietari** exista o relatie de **one-to-one** din cauza ca un proprietar poate sa detina o singura carte de identitate care este unica.

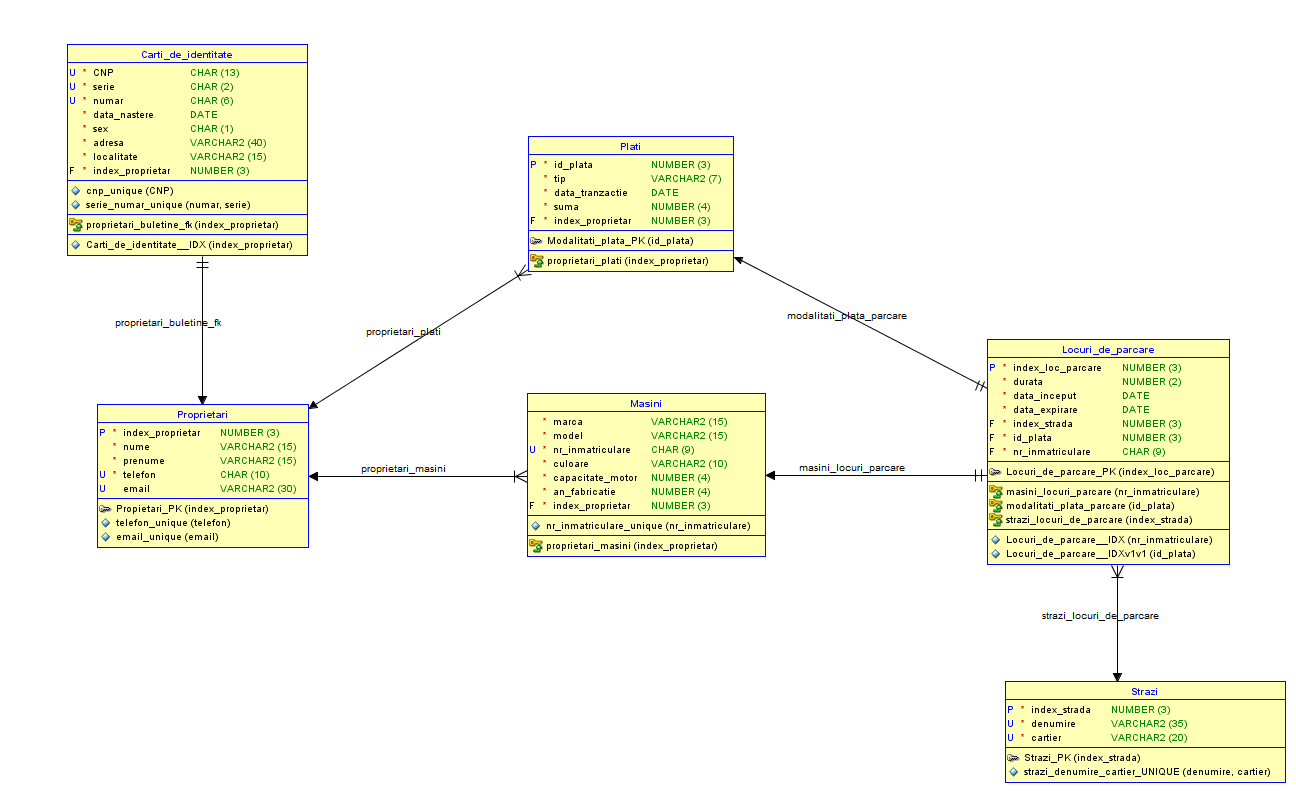
Intre tabelele **proprietari** si **masini** este o relatie de **one-to-many** din cauza faptului ca un proprietar poate sa aibe mai multe masini, respectiv o masina poate sa aiba un singur proprietar.

Intre tabelele **proprietari** si **plati** avem o relatie de **one-to-many** deoarece un proprietar poate sa efectueze mai multe plati din diverse motive, ca de exemplu detine doua masini si vrea sa inchirieze 2 locuri de parcare sau locul de parcare pentru masina a expirat si vrea sa-l reinnoiasca.

Intre tabelele **masini** si **locuri\_de\_parcare**, respectiv **plati** si **locuri\_de\_parcare** avem o relatie de **one-to-many** deoarece tabela locuri\_de\_parcare poate avea asignat o singura plata si o singura masina si viceversa.

Intre tabelele **locuri\_de\_parcare** si **strazi** avem o relatie de **many-to-one** deoarece pe o strada putem avea mai multe locuri de parcare.

****

****

1. **Descrierea constrangerilor folosite si de ce au fost acestea necesare**

Am aplicat o constrangere putea a valida datele de intrare sa fie doar litere pe urmatoarele atribute: serie (**carti\_de\_identitate**), nume, prenume (**proprietari**), marca, culoare (**masini**), denumire, cartier (**strazi**)

In tabela **carti\_de\_identitate** avem urmatoarele constrangeri:

**- CNP-ul:** sa fie unic si de forma:

(1|2|5|6) [0-9][0-9] (0|1)[0-9] (0|1|2|3)[0-9] (0|1|2|3|4|5)[0-9] [0-9][0-9][0-9] [0-9]

S AA LL ZZ JJ NNN C

**- serie:** sa fie de 2 caractere

**- numar:** sa contina doar cifre si sa fie de 6 caractere.

**- sex:** list of values: M/F

**-** Exista o constrangere de tip unic pe atributele **(serie, numar)** ca amandoua simultan sa fie unice (sa nu existe 2 serii cu acelasi numar)

In tabela **proprietar:**

**- telefon:** sa aibe prima cifra 0 si urmatoarea 2 sau 7 (am adaugat 2 pentru telefoanele fixe si 7 pentru telefoanele mobile), o lungime de 10 cifre si unic.

**- email:** sa fie de forma [a@b.c](mailto:a@b.c) si unic.

In tabela **masini:**

**- model:** m-am decis sa nu folosesc vreo constrangere datorita faptului ca pot exista modele care sa contina doar litere (Tiguan), litere si cifre (Golf 7) sau doar cifre (modele de la Porsche: 911, 912 etc)

**- nr\_inmatriculare:** sa fie de forma: LL-DD-LLL, L-DD-LLL sau L-DDD-LLL, unde L-litera si D-cifra. Am ales sa adaug aceasta constrangere datorita numerelor de inmatriculare din Bucuresti. Pe langa forma, sa fie si unic.

**- an\_fabricatie:** list of ranges: 1950 – anul curent. Am ales o plaja mai mare pentru masinile de epoca.

**- capacitate\_motor:** list of ranges: 0-9999. Am ales o plaja mare pentru a putea include masinile electrice (care au capacitatea motor 0), masinile normale ( in jur de 1000-2399) dar si masini cu capacitate mare (peste 6000, in special cele de epoca).

In tabela **plati:**

**- tip:** list of values: ‘numerar’ sau ‘card’.

**- suma:** list of values: 500, 900 sau 1700

In tabela **locuri\_de\_parcare:**

- **durata:** list of values: 6,12 sau 24

In tabela **strazi:**

- O constrangere de tip unic pentru denumire si cartier simultan, in ideea ca aceeasi strada sa poata exista dar sa se afla intr-un cartier diferit, dar niciodata doua strazi cu aceeasi denumire in acelasi cartier.