Modul Analiza

CUPRINS:

[1. Descrierea modelului ales si obiectivele aplicatie 1](#_Toc134127096)

[2. Diagramele bazei de date OLTP 2](#_Toc134127097)

[2.1 Diagrama Entiate-Relatie 2](#_Toc134127098)

[2.2 Diagrama Conceptuala 3](#_Toc134127099)

[3. Diagrama stea a bazei de date depozit 4](#_Toc134127100)

[4. Descrierea campurilor tabelelor bazei de date depozit 4](#_Toc134127101)

[5. Constrangerile depozitului 6](#_Toc134127102)

[6. Identificarea indecsilor specifici depozitelor de date 10](#_Toc134127103)

[7. Identificarea obiectelor de tip dimensiune 10](#_Toc134127104)

[8. Identificarea tabelelor care vor fi partiționate și a tipului de partiționare 10](#_Toc134127105)

[9. Rapoarte: 11](#_Toc134127106)

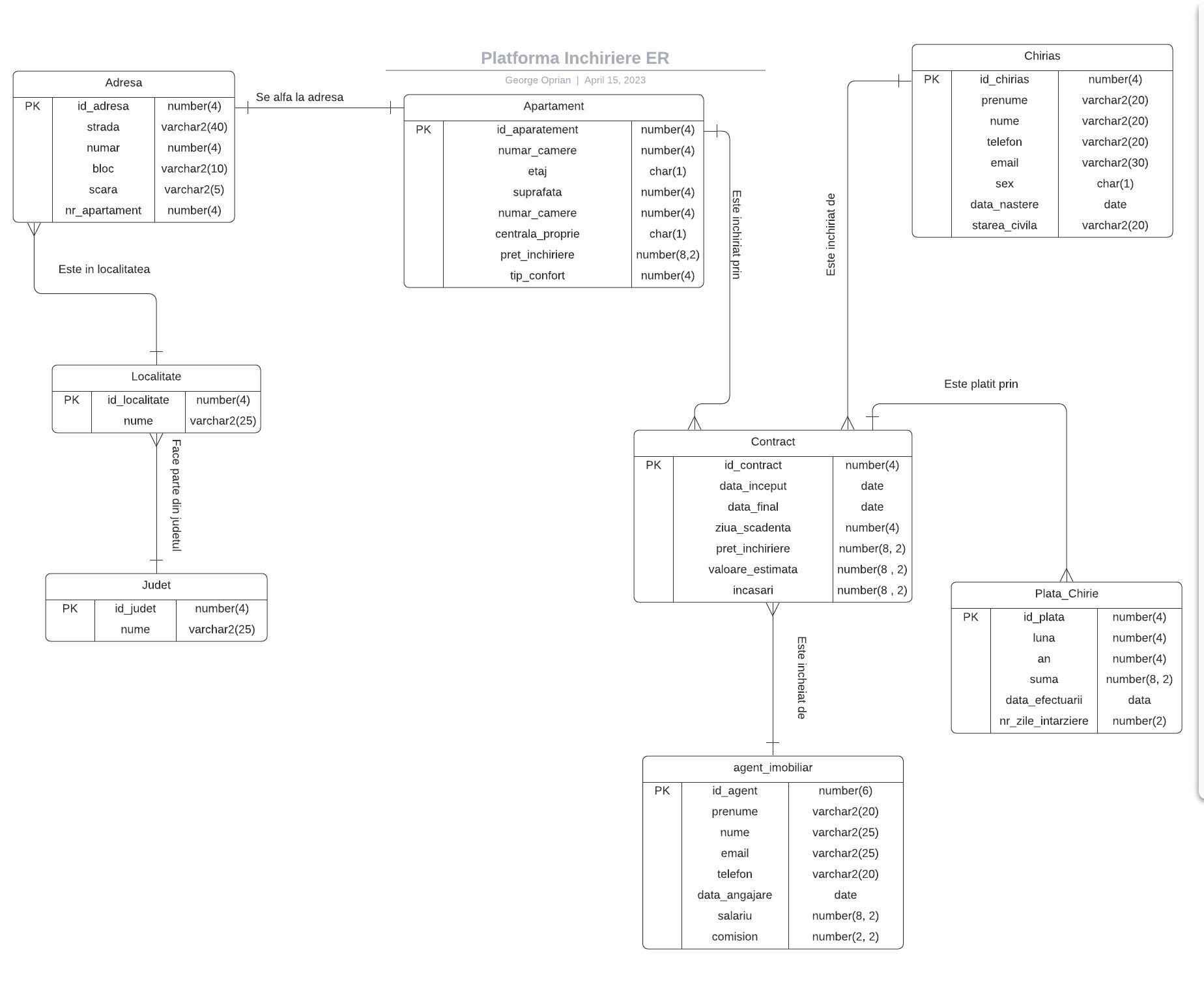
# Descrierea modelului ales si obiectivele aplicatie

Pentru a indeplini cerintele proiectului am ales sa implementam o aplicatie de gestiune a unei agentii imobiliare care inchiriaza locuinte. In acest domeniu de activitate este nevoie de aplicatii de gestiune a proprietatilor doarece in ultimii ani cererea de inchiere de locuinte a crescut, din mai multe motive printe care putem enumera preturile destul de mari ale locuintelor, oamenii nu vor sa mai achizitioneze case doarece stilul lor de viata si locul de munca le permite sa schimbe locul in care locuiesc destul de des, sau poate doar doresc o locuinta temporara pana isi croiesc un rost in viata.

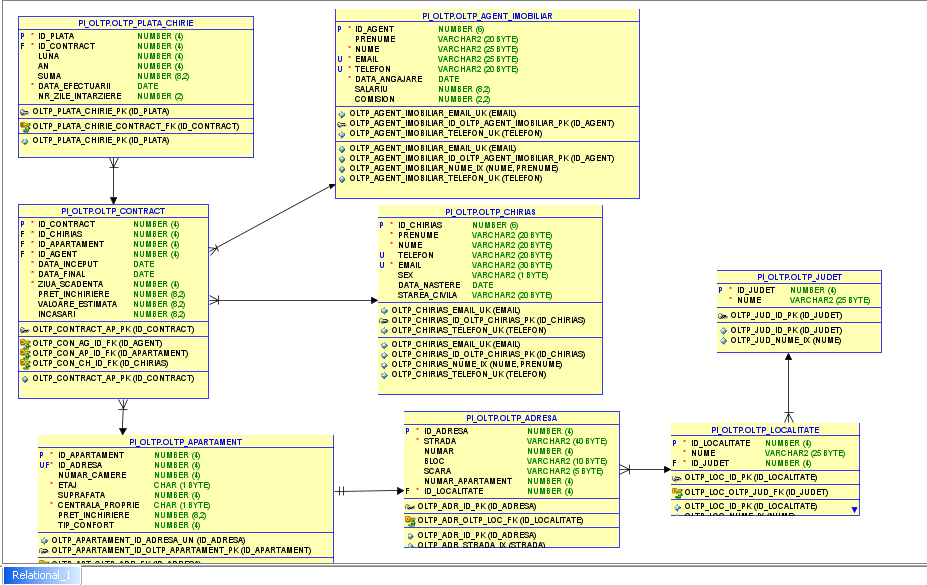
Astfel aplicatia noastra isi propune sa rezolve acesta problema prin gestionarea principalelor entitati prezente intr-o astfel de platforma: apartamente, chiriasi, angajati, contracte si plati.

# Diagramele bazei de date OLTP

## Diagrama Entiate-Relatie



## Diagrama Conceptuala



# Descrierea modului de distribuire

Am decis sa implementam o distribuire a bazei de date centralizate in doua baze de date locale in functie de localizarea apartamentului. Cele doua baze de date sunt definite pe aceeasi masina fizica. Astfel, am impartit baza de date centralizata in doua baze de date locale: Provincie si Bucuresti. Datele din ceele doua baze de date urmand sa fie inglobate intr-o singura schema, nationala.

Cele 3 scheme sunt impartite pe doua baze de date fizice: modbd1 si modbd2. Urmand exemplul din laborator, pe modbd1 au fost definite schema pentru baza de date Bucuresti si National, iar pe modbd2 a fost definite schema pentru baza de date Provincie.

# Argumentarea deciziei de replicare

Pentru a putea defini cele doua baze de date locale, baza de date centralizata a trebuit supusa mai multor metode de fragmentare.

## Fragmentarea orizontala primara

Pentru a putea sparge tabelele curente in cel putin doua fragmente orizontale, am decis sa aplicam o fragmentare orizontala bazata pe localizarea apartamentelor. Astfel, am plecat de la idea ca apartamentele pot fi impartite in doua categorii: apartamente din Bucuresti si apartamente din provincie. Totusi, pentru a putea face aceasta impartire trebuie plecat de la judetul in care se afla apartamentul.

Astfel, fragmentarea orizontala trebuie sa inceapa cu fragmentare tabelei OLTP\_JUDET.

Aplicarea algoritmului COMP\_MIN:

**input**

R = OLTP\_JUDET

Pr = {nume=Bucuresti, nume<>Bucuresti} // Pr = {p1, p2}

**declare**

F: {} //multime de predicate simple completa si minimala

**output**

Pr' <- {nume=Bucuresti} // multime de predicate simple completă și minimala

Pr <- Pr \ {nume=Bucuresti}

F = {f1} // f1 = fragmenul lui p1

` **repeta**

//cautam un predicat pj (unde 1<j<=2) din Pr care partitioneaza fragmente fk din Pr' conform regulii 1

//Consideram j <- 2

k <- 2

Pr' <- Pr' <- {nume=Bucuresti, nume<>Bucuresti}

Pr <- Pr \ {p2} <- {}

F <- {f1, f2}

if exista p2 apartine lui Pr' nerelevant -> FALS

pana cand Pr' completa // dupa primul pas Pr' completa => algoritmul se termina

Deci multimea de predicate simpla si completa este **Pr'** = {nume=Bucuresti, nume<>Bucuresti}, **Pr** = {}, **F** = {f1, f2}, unde **f1 = multimea inregistrarilor cu nume=Bucuresti, f2 = multimea inregistrarilor cu nume<>Bucuresti**.

Prin urmare, vom avea doua fragmente orizontale ale relatiei OLTP\_JUDET: JUDET\_BUCURESTI si JUDET\_PROVINCIE.

## Fragmentarea orizontala derivata

Plecand de la fragmentare tabelului OLTP\_JUDET putem fragmenta si urmatoarele table care tin de localizarea unui apartament.

Astfel fragmentarea orizintala se propaga in urmatoarele tabele:

* OLTP\_LOCALITATE: LOCALITATE\_BUCURESTI si LOCALITATE\_PROVINCIE
* OLTP\_ADRESA: ADRESA\_BUCURESTI si ADRESA \_PROVINCIE
* OLTP\_APARTAMENT: APARTAMENT\_BUCURESTI si APARTAMENT\_PROVINCIE
* OLTP\_CONTRACT: CONTRACT\_BUCURESTI si CONTRACT\_PROVINCIE
* OLTP\_PLATA\_CHIRIE: PLATA\_CHIRIE\_BUCURESTI si PLATA\_CHIRIE\_PROVINCIE

## Fragmentarea verticala

Am ales sa fragmentam vertical tabelul OLTP\_AGENT\_IMOBILIAR deoarece face destul de usor o separare intre coloanele acestui tabel. Astfel am imparit coloanele in doua categorii: coloane ce tin de datele persoanale ale agentului si coloane care tin de salarizare.

De adaugat @Petru

# Verificarea corectitudinii fragmentarilor

todo

# Argumentarea deciziilor de replicare

Am ales sa replicam datele din tabelele CHIRIAS si detaliile personale din AGENT\_IMOBILIAR, mai exact detaliile despre agent din bazele de date locale.

Principalul motiv pentru care am ales sa replica detaliile despre chirias a fost dat de faptul ca un chirias se poate muta dintr-un oras in altul, iar in cazul in care doreste sa inchirieze un alt apartament din alt oras este neplacut ca acesta sa isi introduca datele personale daca acestea nu s-au modificat.

In cazul replicarii tabelului de agenti, am avut un motiv asemanator. Unii angajati pot fi relocati si nu este nevoie ca acestia sa isi introduca detaliile personale de fiecare data cand se relocheaza.