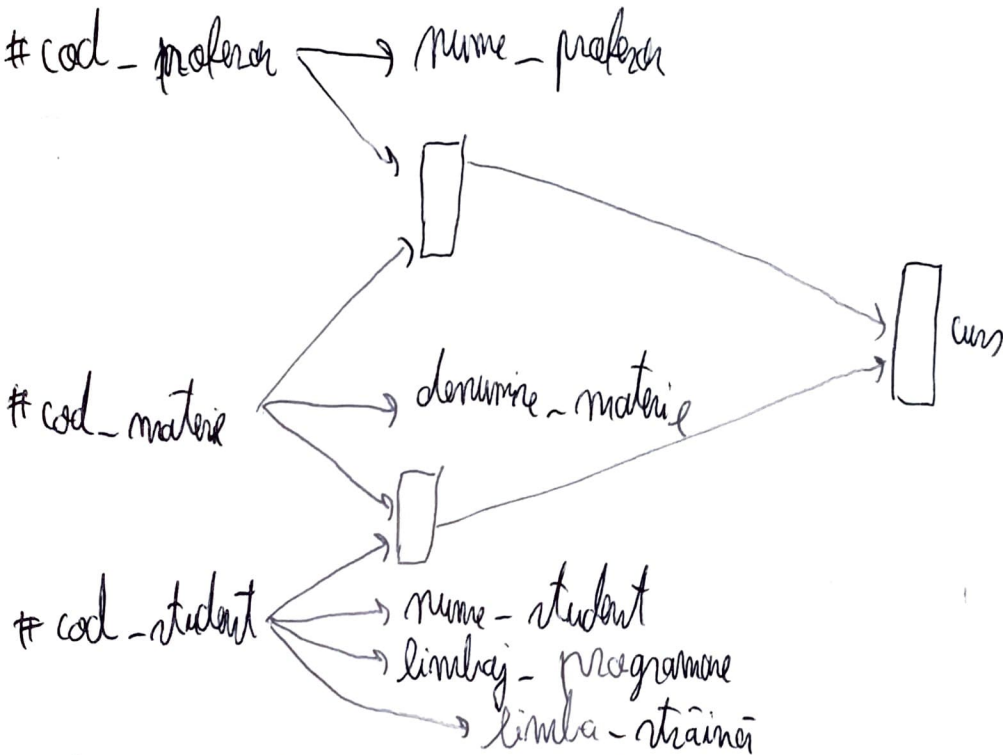


1) Dependente:

~~cod~~ - student \rightarrow nume - student, limbaj - programare, limba - straina

cod - materie \rightarrow denumire - materie

cod - profesor \rightarrow nume - profesor



2) FN1: fiecare atribut trebuie să îi corespundă o valoare atomică. Pt. a transforma în FN1, se desparte relația:

~~Este deja în FN1~~ ~~PT. LIMBAJ - PROGRAMARE = (cod - limbaj #, nume)~~
~~PT. LIMBAJ - STRAINA = (cod - limba #, nume)~~

FN2: - trebuie să fie în FN1 se (este deja)
 fiecare atribut care nu participă la o cheie primară

trebuie să depindă de întreaga cheie primară

~~Este deja în FN2~~. Se repară relația: $S + M + P + L_1 + L_2$

FN3 → trebuie să fie în FN2

→ fiecare atribut care nu participă la o cheie primară
trebuie să depindă direct de cheia primară

Este deja în FN3.

FN4: trebuie să fie în BCNF

Orice dependență multivaloare este o dependență
elementară de forma $X \twoheadrightarrow Y$, Y este o supercheie a lui X .
Ca să reduci rel în FN4, trebuie să elimini dependențele
multivaloare atunci când sunt cel puțin 2 în relație

Ex:

cod - student	nume - student	limbaj - programare	limba - străină
1	Mihai	RUST	Engleză
1	Mihai	Python	Franceză
2	Radu	C	Chineză
2	Radu	Java	Rusă

Este o dependență multivaloare deoarece între limbaj - programare
și limba - străină nu este nicio relație, dar, conform ex, este o valoare între
cele 2. În trebur pt studentul cu cod 1 să mai fie adăugate multe limbi.
Soluția pe care o propun este repararea limbaj - programare și limba - străină în
alte 2 tabele

② a) entități: restaurant, eveniment, organizator, oferta, preparat, ingredient, furnizor, meniu

roluri: ~~persoană~~ persoană fizică, persoană juridică

Diagrama E/R

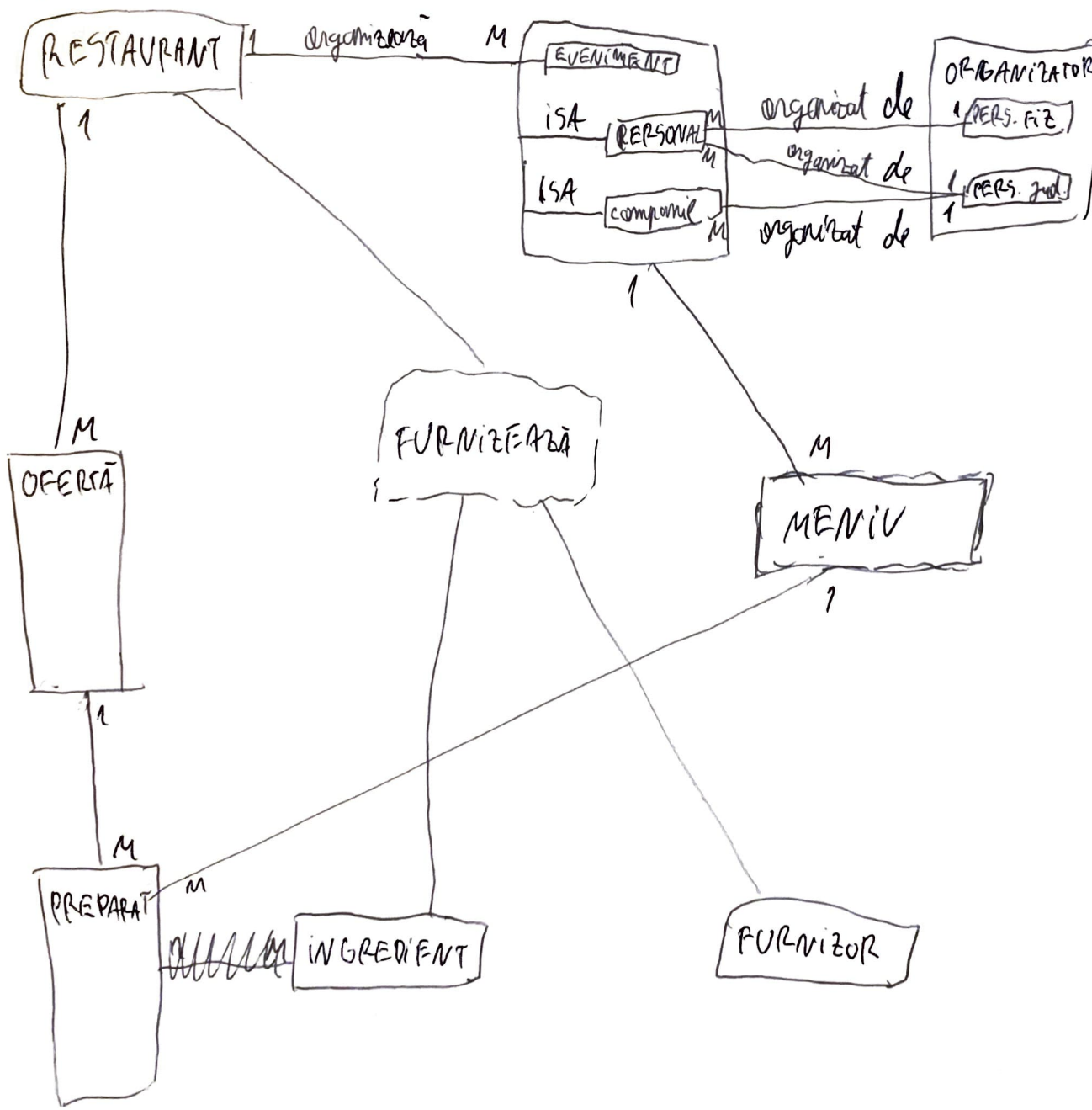
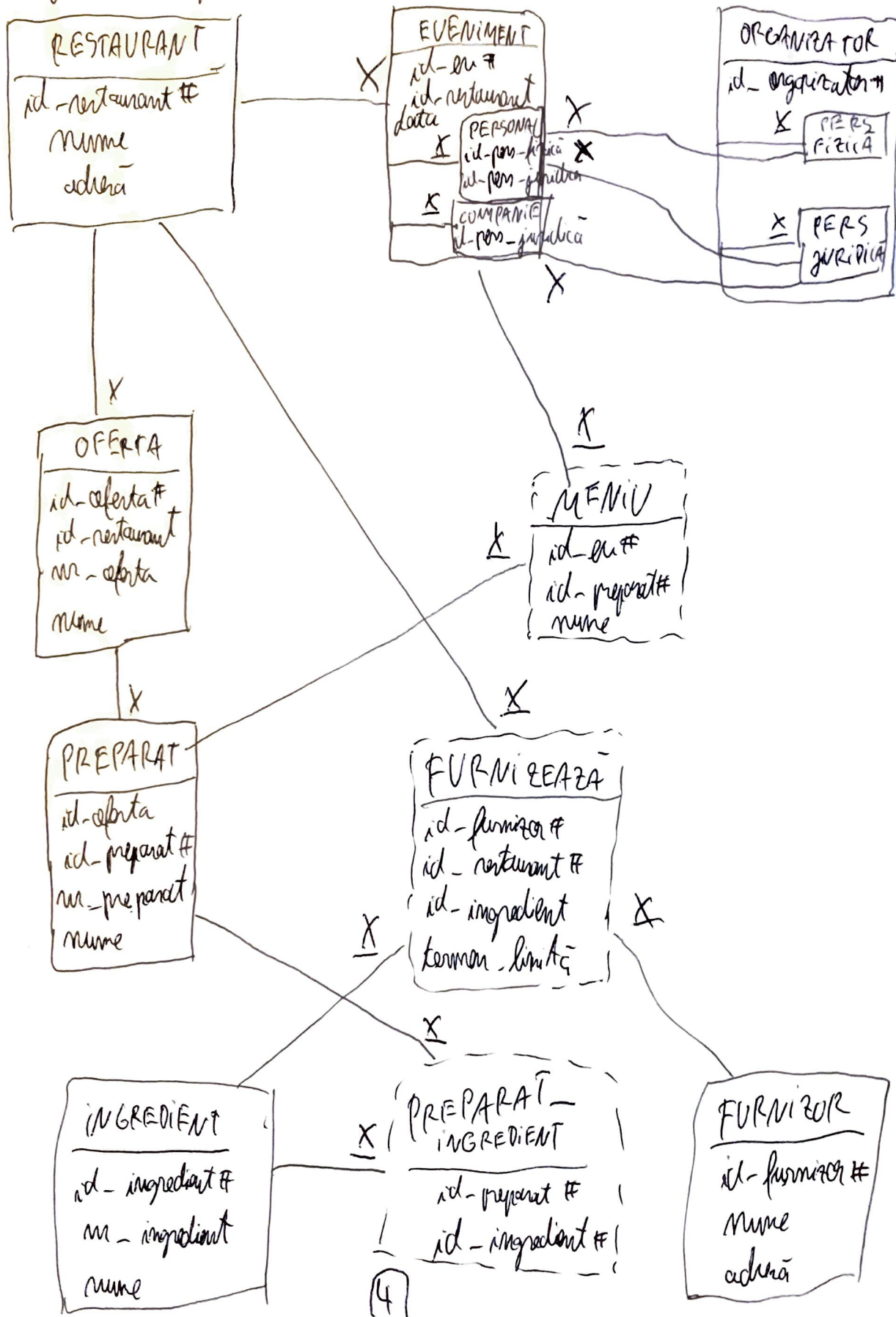


Diagrama conceptuală



c) Chei externe:

RESTAURANT: -

EVENIMENT: id - restaurant

ORGANIZATOR: -

OFERTA: id - restaurant

MENIU: -

PREPARAT: id - oferta

FURNIZEAZA: -

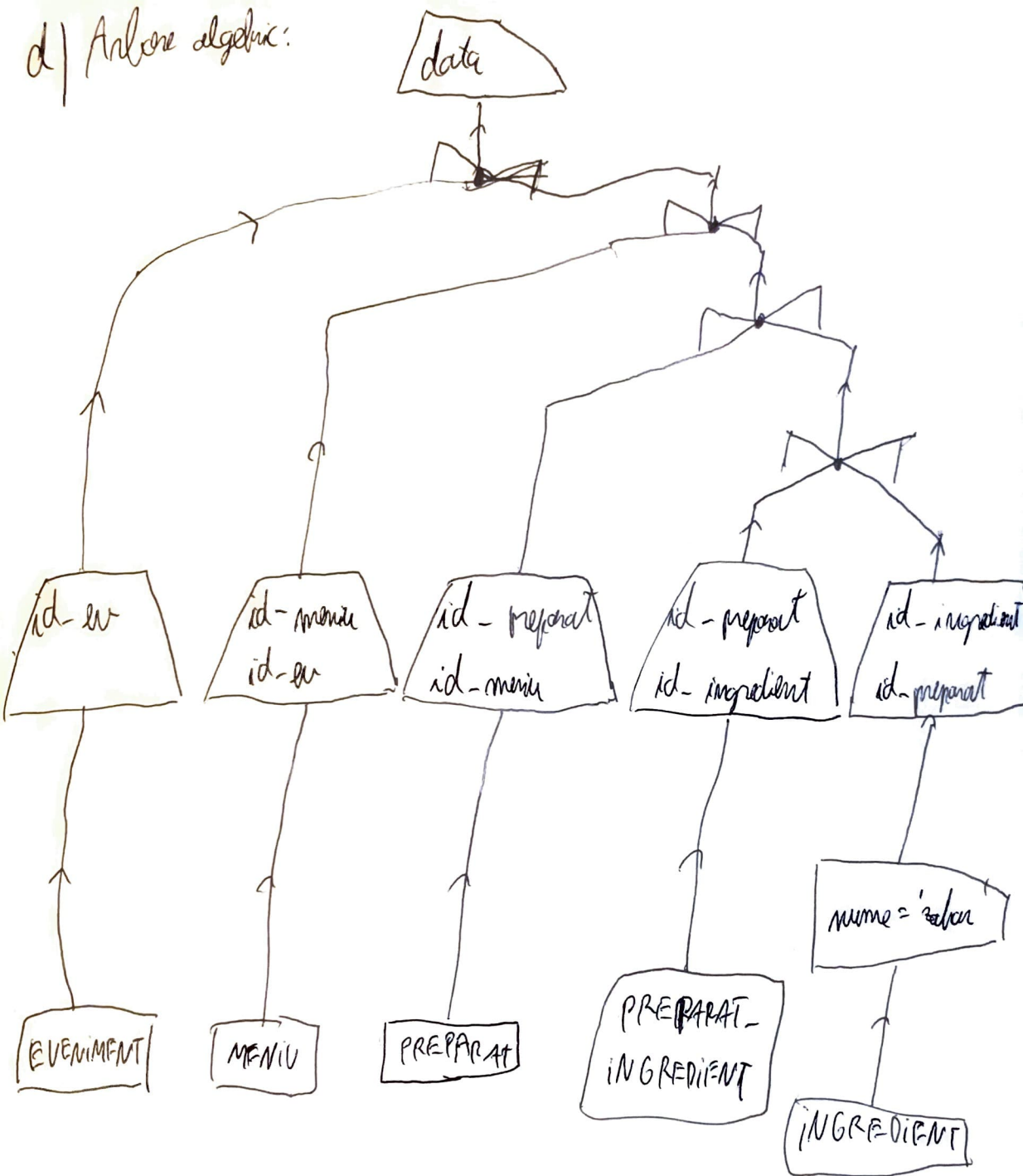
INGREDIENT: -

PREPARAT_INGREDIENT: -

FURNIZOR: -

d) cealaltă pagină

d) Anlönse algebra:



EXPRESIE ALGEBRICĂ: $R_1 = \text{SELECT}(\text{INGREDIENT}, \text{nume} = \text{'salon'})$
 $R_2 = \text{JOIN}(R_1, \text{PREPARAT-INGREDIENT})$
 $R_3 = \text{JOIN}(R_2, \text{PREPARAT})$
 $R_4 = \text{JOIN}(R_3, \text{MENU})$

$R_5 = \text{JOIN}(R_4, \text{EVENTMENT})$

$R_{E2} = \text{PROJECT}(R_5, \text{data})$

→ Rezoluarea este optimă întrucât produsurile carteziene se înlocuiesc cu join-uri. Un produs cartezian între 2 relații este mult mai zămp (ca și cost) decât un join, pt că generarea tuplurilor într-un mod exhaustiv. În modul prezentat mai sus, transformarea s-a realizat folosind legătura dintre produs cartezian, join și selecție.