

Examen

2. Tabelul este în 1FN, deoarece intersecția dintre orice coloană și o înregistrare are o singură valoare.
cici data - livrare, nici auto - curier - nr nu trebuie despărțite.

2FN:

Identificăm dependențele:

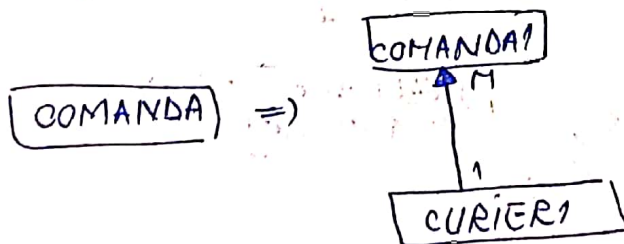
$\{ \text{curier-id} \} \rightarrow \{ \text{nume-curier}, \text{auto-curier-nr}, \text{auto-curier} \}$
 $\{ \text{curier-id}, \text{comanda-id} \} \rightarrow \{ \text{adresa-livrare}, \text{data-livrare} \}$

Cum cheia primară e compusă și formată din $\{ \text{curier-id}, \text{comanda-id} \}$, prima dependență e parțială față de cheia primară.

Deci, formăm noile tabele:

COMANDA1 (curier-id#, comanda-id#, adresa-livrare, data-livrare)
FK pentru curieri

CURIER1 (curier-id#, nume-curier, auto-curier-nr, auto-curier)



3FN:

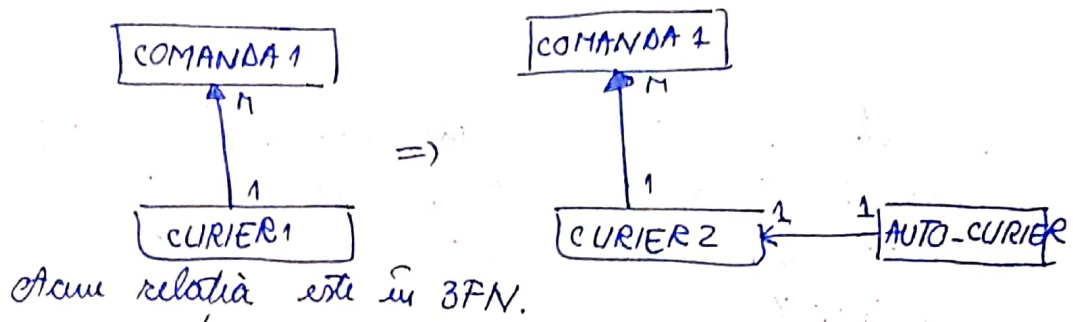
COMANDA1 nu e în 3FN, deoarece avem dependențele:

$\{ \text{auto-curier-nr} \} \rightarrow \{ \text{auto-curier} \}$

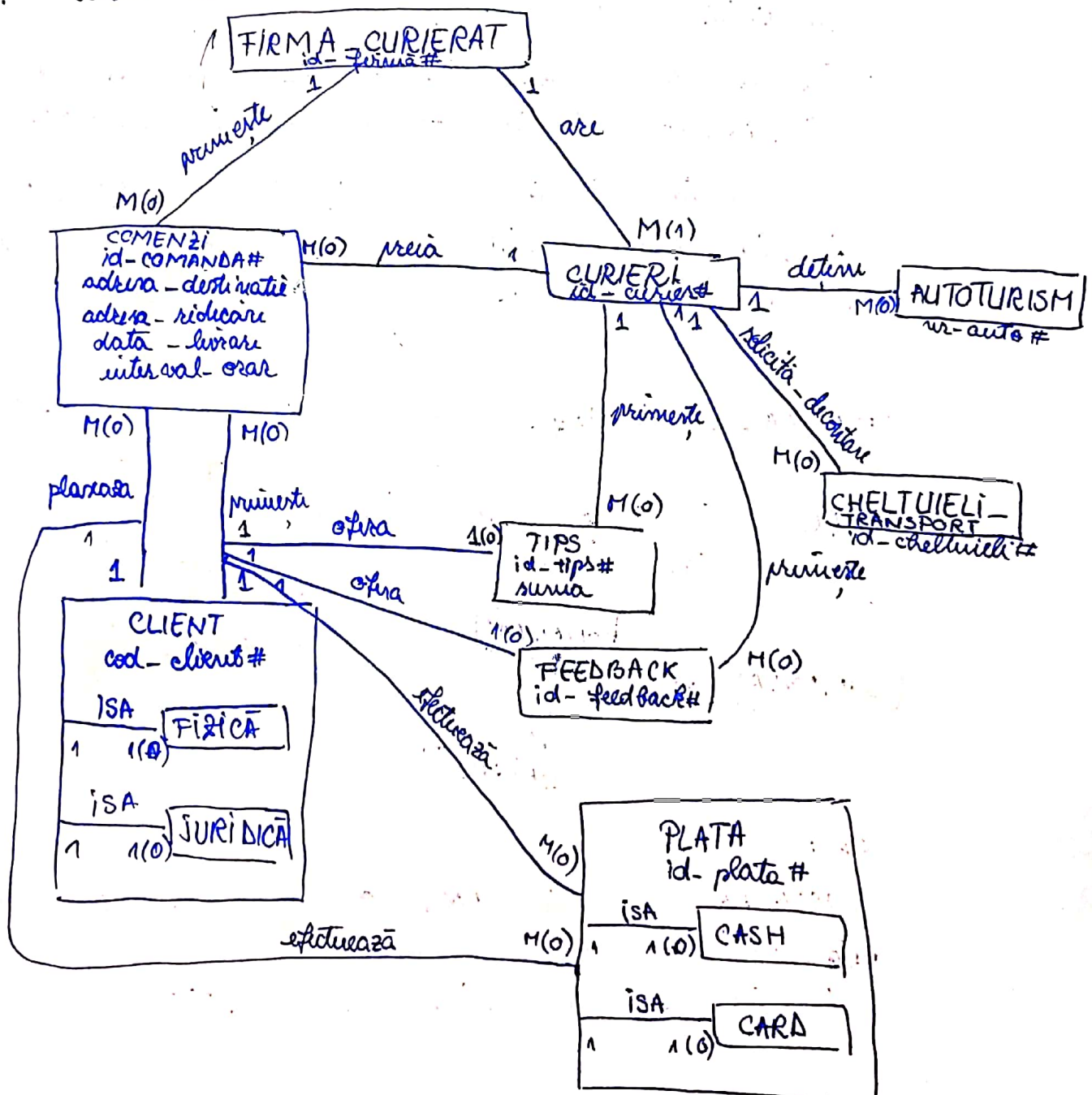
$\{ \text{curier-id} \} \rightarrow \{ \text{auto-curier} \}$ dep. tranzitivă

CURIER2 (curier-id#, curier-nume, auto-curier-nr)
cheie străină

AUTO-CURIER (auto-curier-nr#, auto-curier)



1. Firma de curierat



3. Tipurile de isolation level pentru tranzacții sunt:

- **Serializable**: asigură execuția în serial
- **Read repeated**: se permite numai asupra datelor cărora s-a dat commit să fie citite
 - aceeași valoare trebuie să fie returnată în cazul în care s-au realizat citiri repetate ale aceleiași înregistrări
 - între două citiri ale aceleiași înregistrări, nu este permisă nicio tranzacție să modifice datele
- **Read committed**: pot fi citite numai înregistrări asupra cărora s-a dat commit
 - se poate întâmpla și ca între două read-uri ale unui item, o altă tranzacție să modifice item-ul, dar
 - citiri succesive pot întoarce valori diferite, dar toate committed
- **Read uncommitted**: permite datelor la care nu s-a dat commit să fie citite; este cel mai slab nivel de izolare din SQL.

Exemple pornind de la schema de surierat:

Avem o tranzacție T_1 care dorește să afișeze toate tranzacțiile comenzile care au X drept adresă destinație și o tranzacție T_2 care adaugă o nouă comandă cu adresa X. În cazul execuției concurente, dacă T_2 dă commit, va fi afișată și noua înregistrare.

Dacă avem o tranzacție T_3 în care se dau update la data unei comenzi la '26/06/2020', după care T_4 intervine și face update la data la '27/06/2020', ultima valoare va fi executată va fi cea afișată, înainte de ultimul commit iar înainte de valorarea anterioară trebuie să se dea commit.