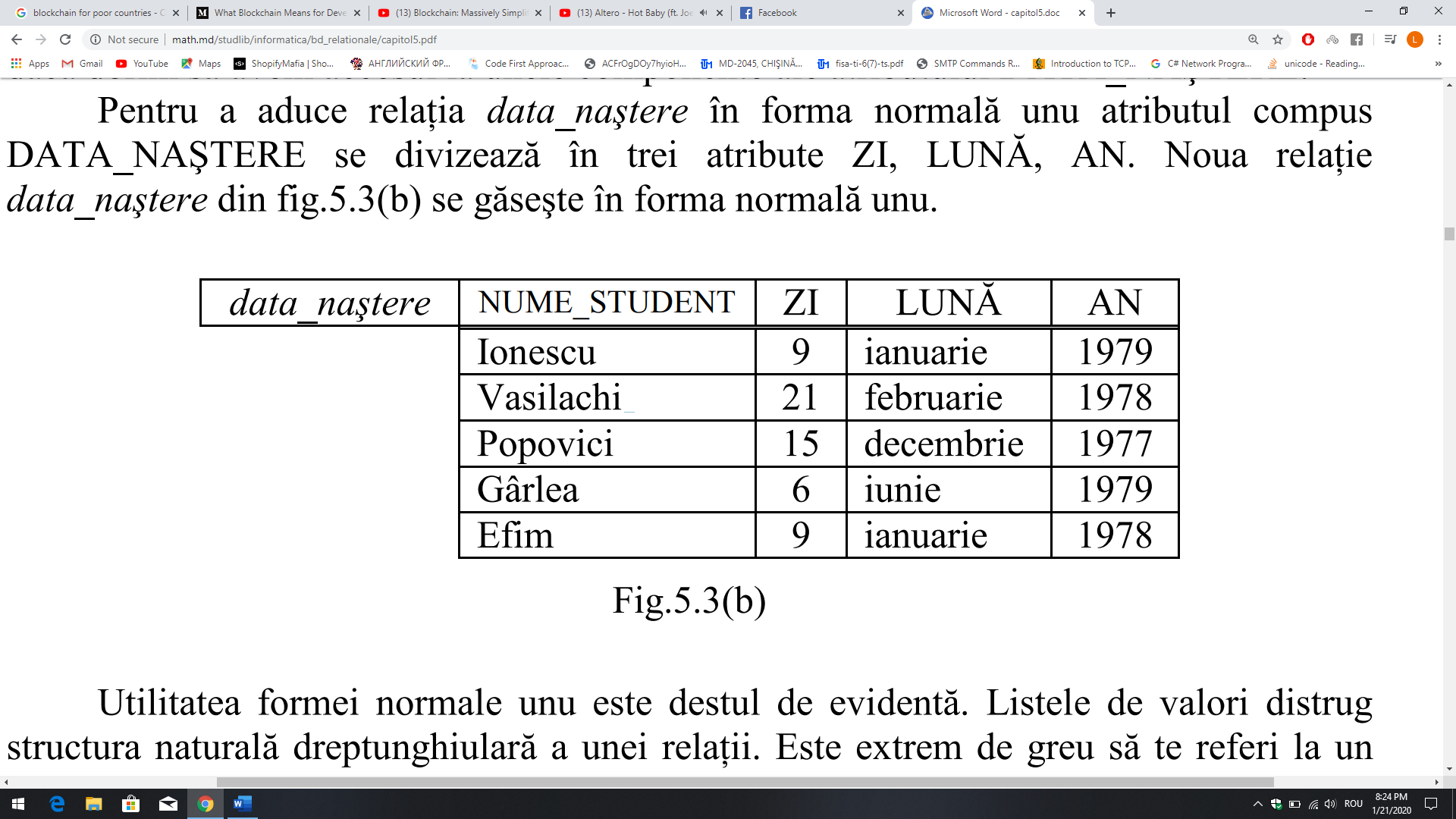
**105. Forma normală unu**

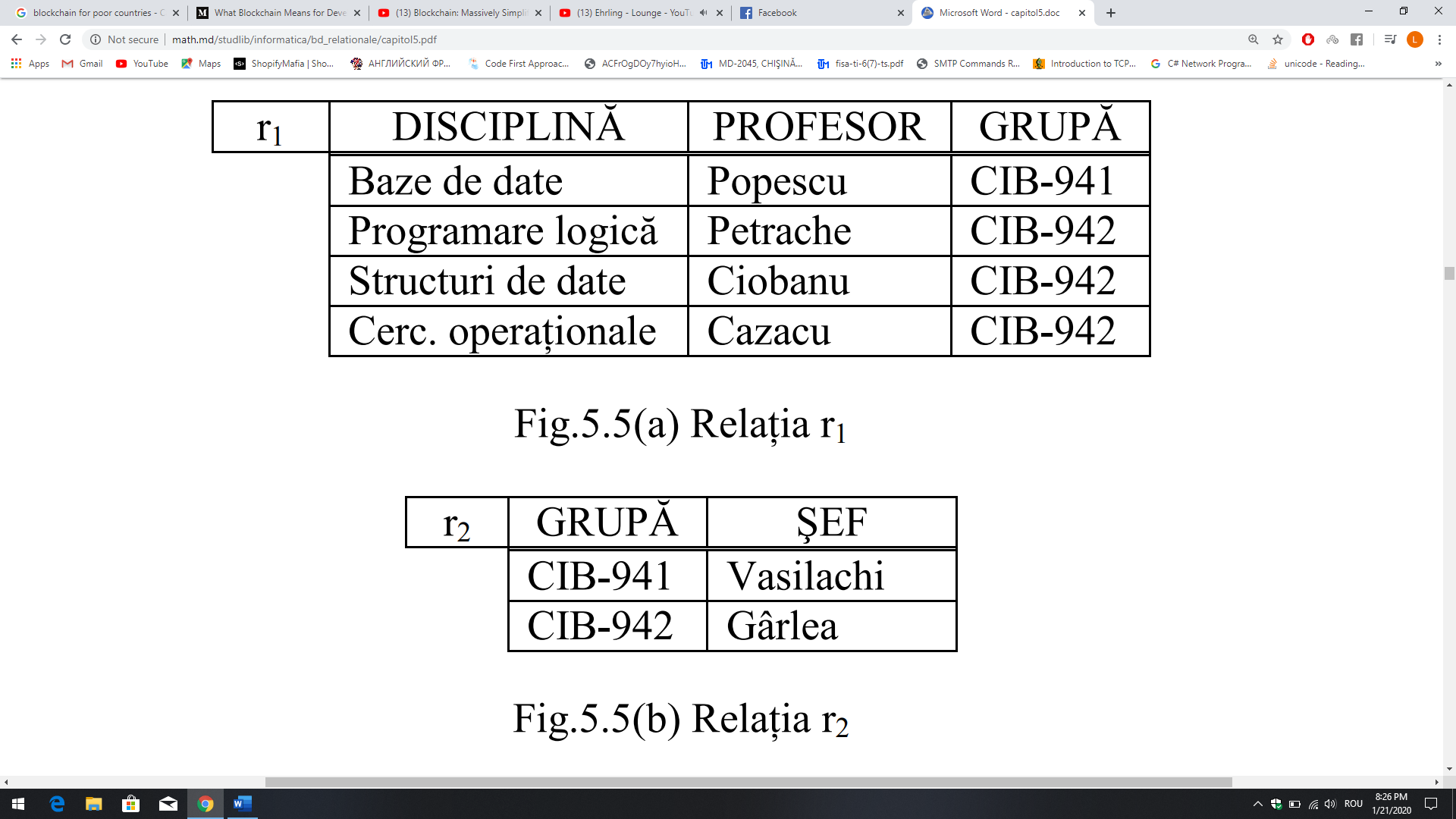
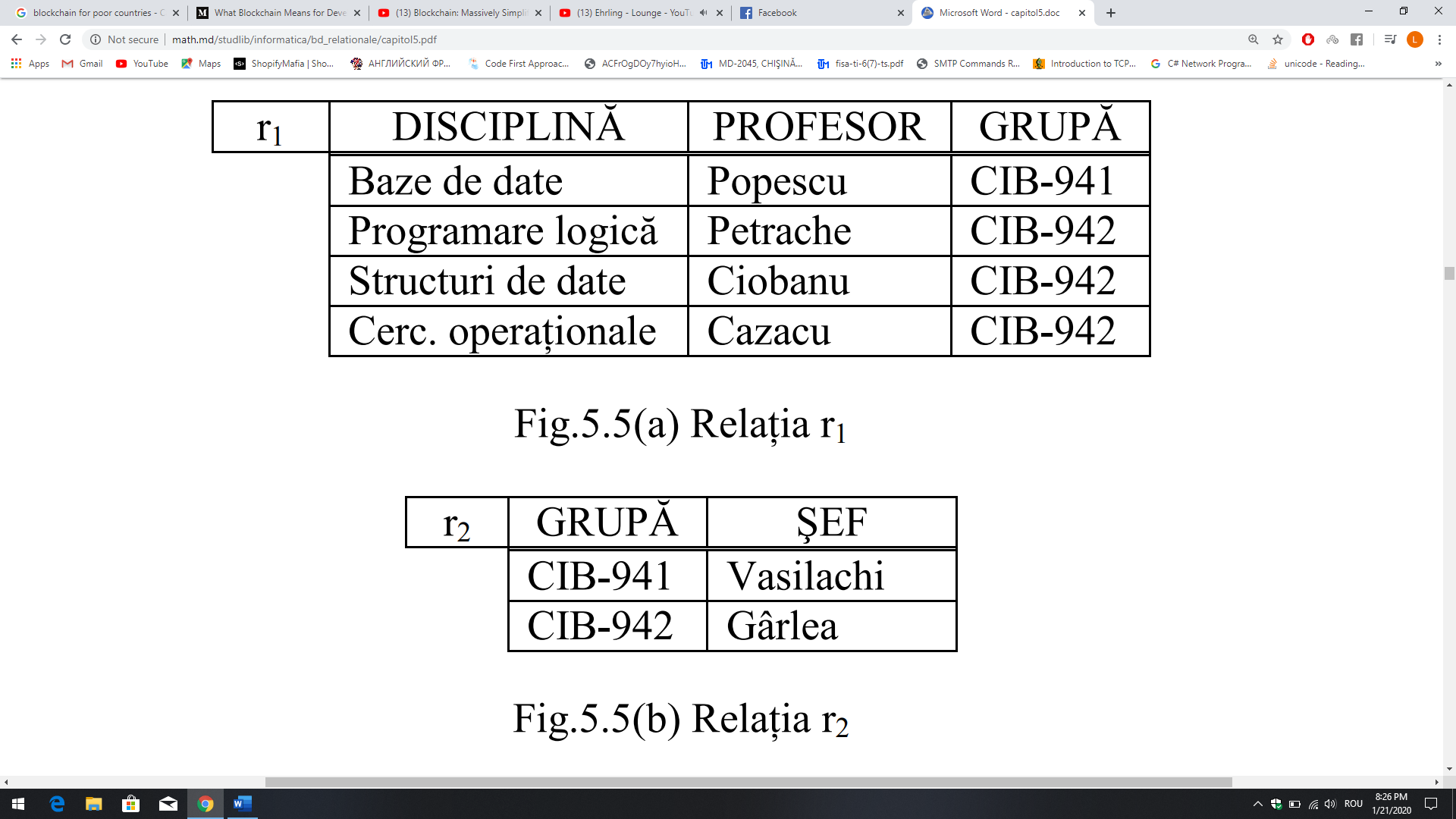
O schema relațională *Sch = (R,F)* se găsește în forma normal unu, dacă orice atribut *A* din *R* este atomic. Schema unei baze de date se găsește în forma normal unu, dacă orice schema relațională constituientă este în forma normală unu. Un atribut atomic este cel care este compus dintr-o valoare unică și care nu poate fi descompusă.



**106. Forma normal doi**

Forma normal doi este motivată de existența reduntanței și a unor anomalii în relațiile care se găsesc în forma normal unu.

O schema *Sch = (R,F)* se găsește în forma normală doi, în raport cu mulțimea de dependențe funcționale *F*, dacă ea se găsește în forma normal unu și orice atribut neprimar din R nu depinde partial de vreo cheie a schemei *Sch*. Schema bazei de date se găsește în forma normal doi, dacă orice schema relațională constituientă se găsește în forma normală doi.



**107. Forma normal trei**

**Raspuns 108.**

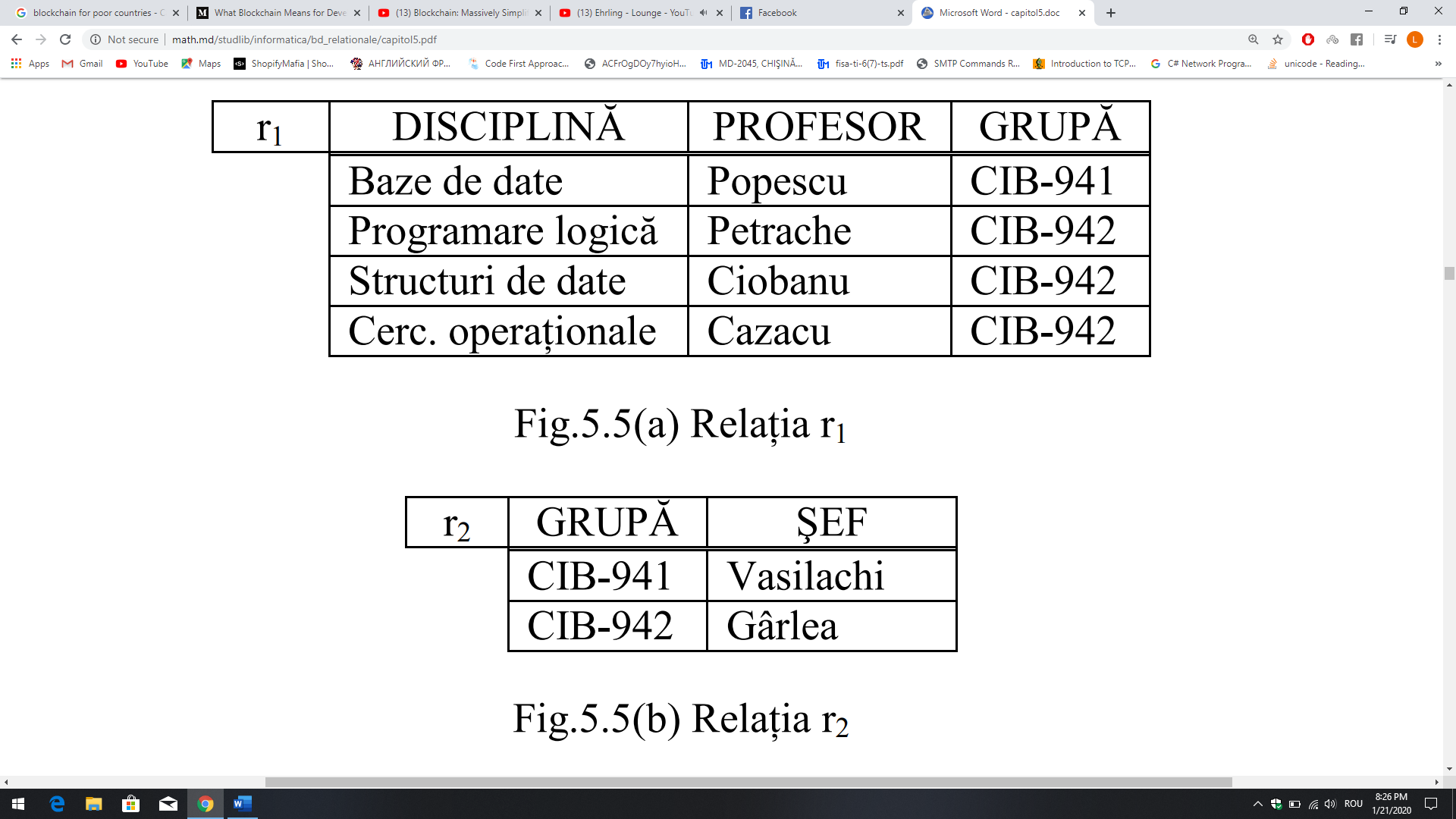
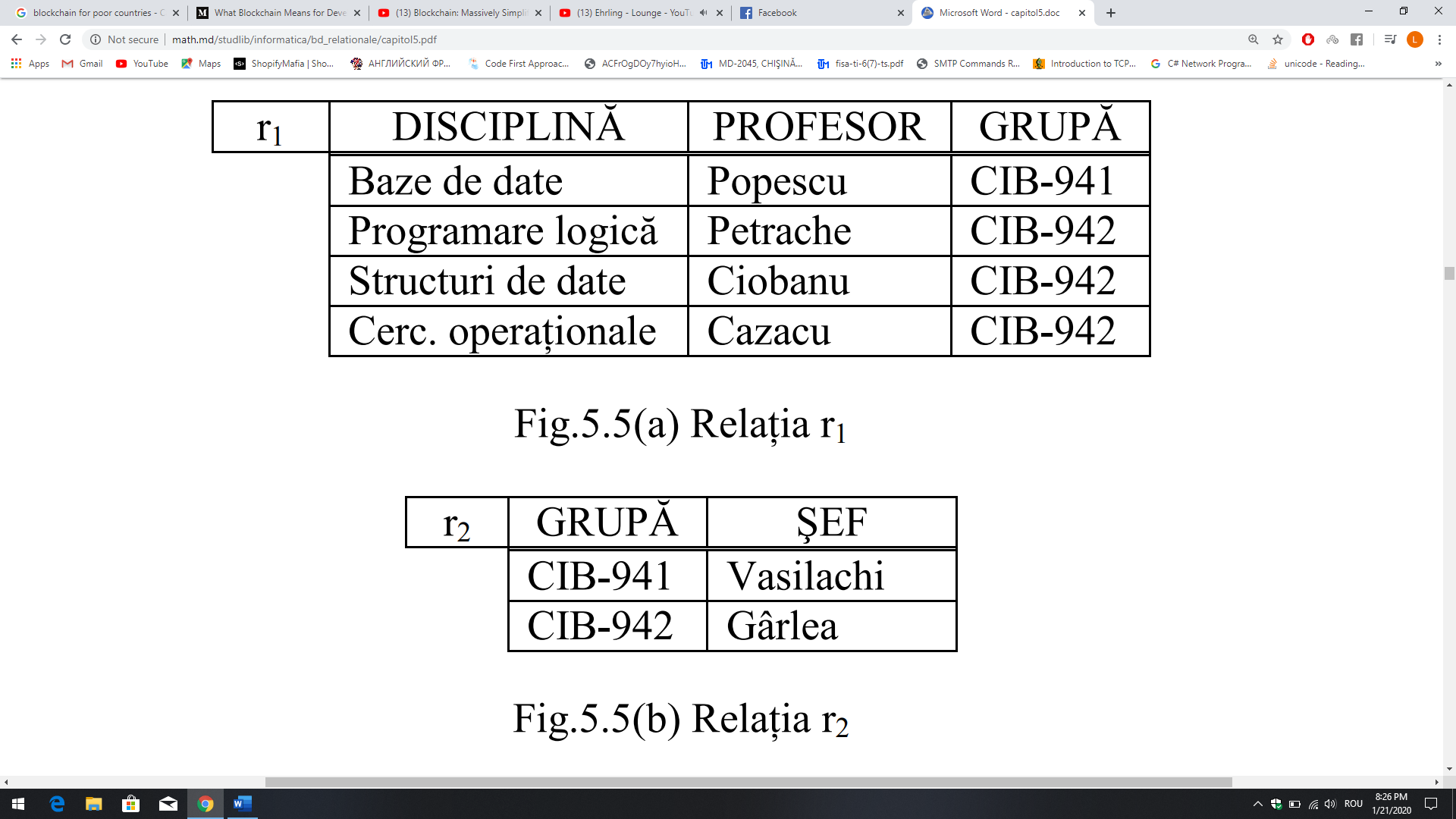
**108 .Corelația dintre forma normal doi si forma normal trei.**

*Schema Sch = (R,F)* se găsește în raport cu mulțimea de dependențe funcționale *F,* daca ea se găsește în forma normal unu și orice atribut neprimar din *R* nu depinde tranzitiv de vreo cheie a schemei *Sch.*  Schema bazei de date se găsește în forma normal trei, dacă orice schema relațională constituent se găsește forma normal trei.

Orice schema ce se găsește în forma normal trei, se găsește și ăn forma normală doi.

Schema *R* se găsește în forma normal trei în raport cu mulțimea de depende funcționale *F,* dacă pentru orice dependență netrivială X -> A aparține F+.

1. X este supercheie pentru R
2. A este atribut primar



**109. Forma normal Boyce-Codd**

**Raspuns 110.**

**110. Corelația dintre forma normal trei și forma normal Boyce-Codd**

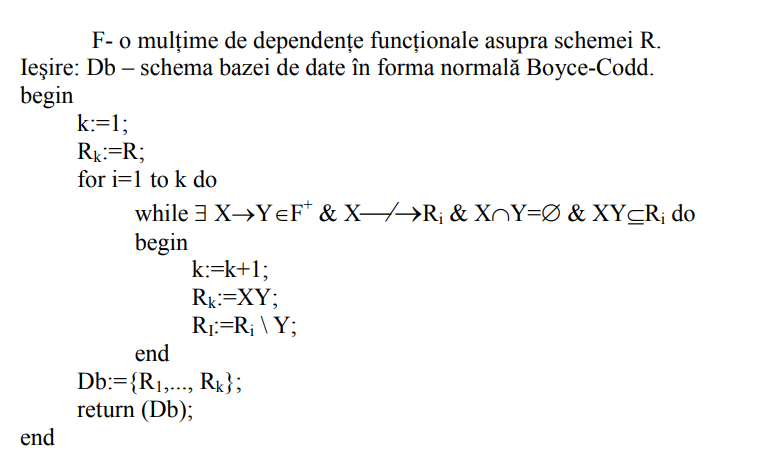
Forma normal trei nu interzice dependența tranzitivă a atributelor primare de cheie. Însă și unele relații în forma normal trei nu sunt lipsite de anomaliile e actualizare a datelor.

Schema *R* se găsește în forma normal Boyce-Codd în raport cu o mulțime de dependețe funcțioanle *F*, dacă pentur orice dependență X ->F aparține F+, determinantul X este o supercheie a schemei R. Schema bazei de date se găsește ăn formal normal Boyce-Codd, dacă orice schema relațională din ea se găsește în forma normal Boyce-Codd.

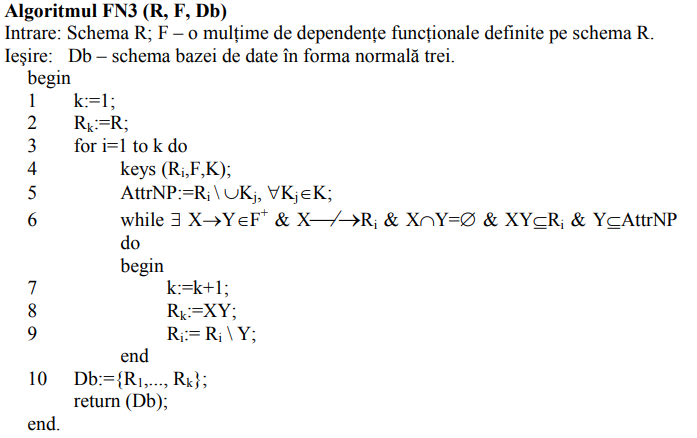
Dacă schema *R* se găsește în forma normal Boyce-Codd, atunci R se găsește în forma normal trei

**111. Aducerea schemei în forma normală Boyce-Codd prin metoda descompunerii fără pierderi. Algoritmul FNBC**

Fie *Sch = (R,F)* o schemă relațională, unde F este o mulțime de dependențe funcționale asupra lui R. Fie {R1,R2,…,Rn}, o descompunere fără pierderi a mulțimii de atribute R, ăn raport cu F și fie {S1,S2} o descompunere fără pierderi a lui R1 în raport cu piR1(F). Atunci descompunerea {S1,S2,S3,S4..} posedă proprietatea joncțiunii fără pierderi, în raport cu mulțimea F de dependențe funcționale.



**112. Aducerea schemei în forma normală trei prin metoda descompunerii. Algoritul FN3**

****

**113. Primul dezavantaj al normalizării prin descompunere.**

Dat fiind faptul că algoritmul FN3 necesită calcularea cheilor schemei și determinarea atributelor nonprimare, iar algoritmii FN3 și FNBC necesită examniarea dependențelor din F+ valide în schema curentă, complexitatea procesului de normalizare nu e polinomială.

**114. Al doilea dezavantaj al normalizării prin descompunere.**

Nu întotdeauna, dintr-o schema data, se obține o schema a bazei de date cu un număr minimal de scheme relaționale normalizate.

**115. Al treilea dezavantaj al normalizării prin descompunere.**

Al treilea dezavantaj apare în cazul aplicării dependețelor parțiale în procesul de descompunere a schemelor. Aceste dependențe genereazăschemeale bazeo de date cu mai multe scheme relaționale decît în cazul cînd toate dependențele applicate sunt complete.  
Acest dezavantaj poate fi evitat, daca depențele utilizate în descompunere sunt reduse(stînga sau dreapta).

**116. Al patrulea dezavantaj al normalizării prin descompunere.**

Al patrulea dezavantaj este că normalizarea prin descompunere nu întotdeauna conserve dependețele funcționale. Acesta se referă la algoritmii FNBC și FN3. Algoritmul FN3\_DP conserve dependențele funcționale, produce scheme ale bazele de date cu un număr sporit de scheme relaționale. Pentru fiecare dependeță ce nu a fost conservată, se construieste o schema relațională care se adaugă la descompunerea finală.

**117. Al cincilea dezavantaj al normalizării prin descompunere.**

Al cincilea dezavantaj este legată de dependențele funcționale latent, care nu întotdeauna pot fi observate în schemel relaționale. Adică, pot fi obținute scheme care nu se găsesc în forma normal preconizată.

**118. Validitatea în normalizarea prin sinteză. Condițiile ce trebuie să le satisfacă un algoritm de sinteză.**

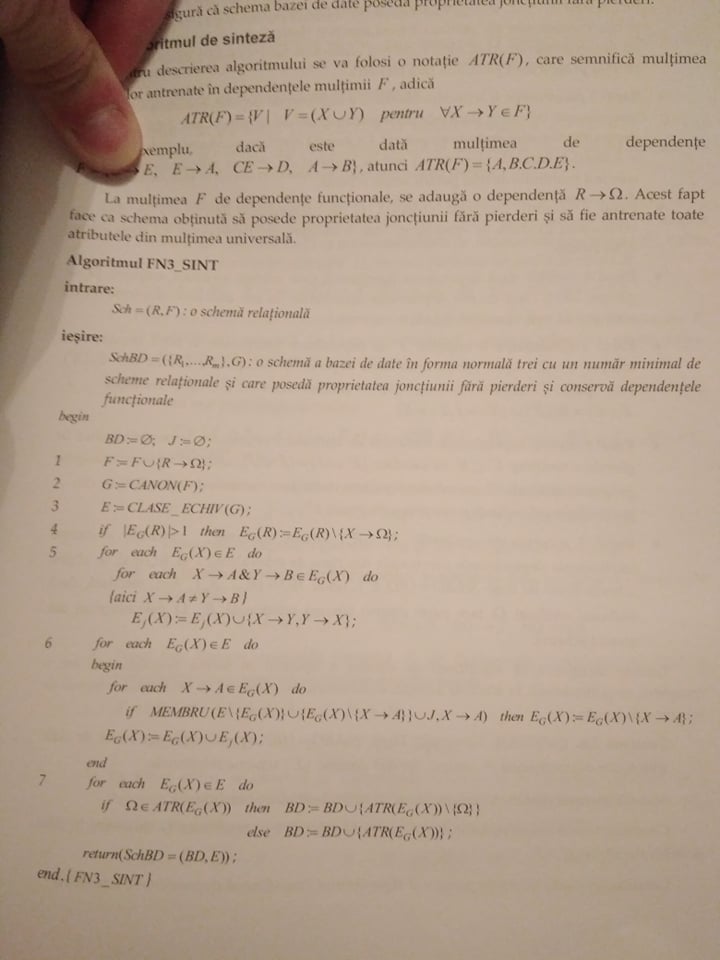
**Condiția 1.** Garantează că proiecțiile relației universal r asupra mulișor BD conserve dependețelle funcționale F. În afară de aceasta, condiția spune că Ri, satisfice numai depenețele ale căror părți stîngi sunt superchei ale schemei Ri.

**Condiția 2.** Este scopul normalizării.

**Condiția 3.** Garantează că orice altă schema construită în baza mulțimii de dependețe F nu conține mai puține scheme relaționale.

**Condiția 4.** Asigură că schema bazei de date posedă proprietatea jocțiunii fără pierderi.

**119. Algoritmul FN3\_SINT**

****

**120. Dezavantajele normalizării prin descompunere. Metoda de ocolire a dezavantajelor.**

Algoritmii de descompunere nu pot asigura, în general, descompuneri FN3 și BCNF care să gie valide și, în același timp, să conserve dependețele funcționale. Se desprinde idea că descompunerea, ca process de ameliorare a structurilor relaționale, este utilă și se justifică efortul de a construe instrumente software corespunzătoate, în măsura în care SGBD-ul nu implementează dependețele funcționale ca mechanism de asigurare a integrității bazei de date.