

SUBIECT 1

① Ce probleme ridică programarea în contextul bazelor de date (dacă)?

1) controlul concurenței

- multe activități (tranzacții) cu baza de date se desfășoară la același moment de timp și acționează asupra aceluiași bucată de informație.

2) optimizarea interogărilor

- volum mare de date

② Explicați funcționalitatea de transformare a datelor pentru un Sistem de Gestionare de Baze de Date.

Transformarea de date este una dintre funcțiile SGBD-ului. Datele introduse de utilizator nu au întotdeauna structura identică cu cea definită în baza de date.

③ Nivelul conceptual de abstractizare al unui Sistem de Gestionare de Baze de Date.

Există 3 nivele de abstractizare :

- 1) nivel extern
- 2) nivel conceptual (schema logică)
- 3) nivel intern (fizică)

Definiți ce este un model de date.
... este: o reprezentare matematică

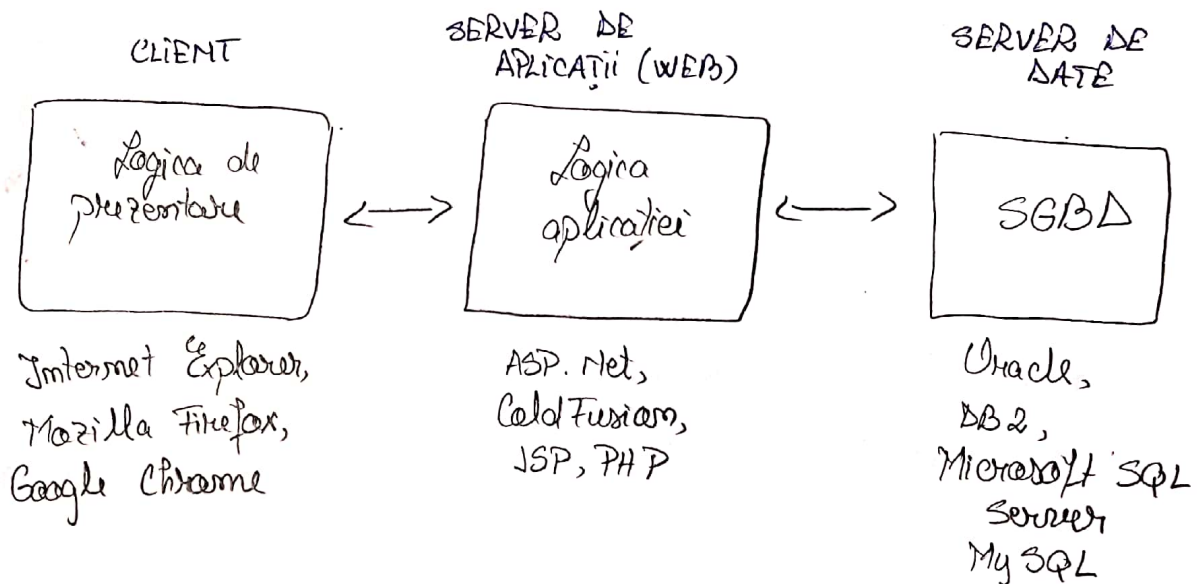
La nivel conceptual se află schema BD logică care descrie datele stocate în BD în termeni ai modelului de date al SGBD.

Pentru un SGBD relațional, schema ~~relațională~~ logică descrie toate tabelele (relațiile) stocate în BD.

Limbaajul folosit se numește Limbaaj de Descriere a Datelor (LDD).

④ Descrieți pe scurt modelul de aplicație trei-straturi.

Model de aplicație 3 straturi



⑤ Care este diferența între "row-level-triggers" și "statement-level-triggers"?

Triggerurile sunt "row-level" sau "statement-level":
FOR EACH ROW indică "row-level"; dacă lipsește indică "statement-level"

Row level-triggers: se executa o singura data pt. fiecare tupla modificata

Statement-level triggers: se executa o singura data pt. o instructiune SQL, indiferent de numarul tuplelor modificate.

⑥ Explicati CLI.

CLI, impreuna cu JDBC, PHP/DB sunt metode de conectare pentru a permite unui limbaj conventional sa acceseze o baza de date.

În loc sa se foloseasca un preprocesor (embedded SQL), se poate folosi o biblioteca cu functii.

Biblioteca pentru C este numita SQL/CLI =
= "Call-Level Interface"

Preprocesorul pentru embedded SQL translateaza instructiunile EXEC SQL... in CLI sau apeluri similare.

⑦ Explicati R scriu-join S folosind un exemplu.

```
SELECT D.dept, D.mame  
SELECT dept.deptno, Dept.dname  
FROM dept D, emp E  
WHERE emp.deptno = dept.deptno  
ORDER BY dept.deptno;
```

Afișează o
listă de
departamente
cu cel puțin
un angajat.

Definiți ce este un număr de date

pag. 90

⑧ Explicați tranzațiile Repeatable-Read.

Dacă Joe execută "serializable", dar Sally nu, atunci Sally se poate să nu vadă prețurile pentru "Joe's Bar".

Dacă datele sunt citite din nou, atunci tot ceea ce a fost văzut prima dată este văzut a doua oară.

Presupunem că Sally execută cu REPEATABLE READ, în ordinea de execuție este (max)(del)(ins)(min).

(max) vede prețurile 2.50 și 3.00.

(min) poate vedea 3.50, dar de asemenea vede 2.50 și 3.00, deoarece aceste prețuri au fost văzute la prima citire de (max).

⑨ Ce indică simbolurile *, +, ? la descrierea elementelor unui document XML cu DTD?

Un tag poate fi urmat de un simbol ce indică multiplicitatea:

☐ * = zero sau mai multe

☐ + = una sau mai multe

☐ ? = zero sau una

SUBIECTUL 2

10) Care sunt caracteristicile pentru alegerea modelului NOSQL?

1) Analizate datele care vor fi stocate
VOLUM MARE, VALOARE SCĂZUTE

2) Analizarea schemei aplicației
DINAMIC

Subject 1

(12)

- (4) Adresa (^Acada, ^Blocalitate, ^Cstrada, ^Dnumar)
- (1) Student (nr-matr, nume, prenume, sex, data-masterii, grupa, cada)
- (5) Profesor (id-prof, nume; prenume, functie, cada)
- (3) Materie (id-mat, denumire, credite)
- (5) Examem (nr-matr, id-prof, id-mat, data, nota)

a) Să se găsească localitate, strada și număr pt. adresele din localitatea 'Dej' și strada 'Narcisilor'.

SQL:

```
SELECT localitate, strada, numar
FROM Adresa
WHERE localitate = 'Dej' AND strada = 'Narcisilor'
```

ARG:

$\Pi_{localitate, strada, numar} \left(\bigvee_{localitate='Dej' \text{ and } strada='Narcisilor'} (Adresa) \right)$

CRT:

$\{ A.localitate, A.strada, A.numar \mid A \in Adresa \wedge A.localitate = 'Dej' \wedge A.strada = 'Narcisilor' \}$

CRIS:

$\{ \langle B, C, D \rangle \mid \langle A, B, C, D \rangle \in Adresa \wedge B = 'Dej' \wedge C = 'Narcisilor' \}$

Answer:

$\Pi_{localitate, strada, numar}$
 \downarrow
 $\bigvee_{localitate='Dej' \text{ and } strada='Narcisilor'}$
 \downarrow
 Adresa

b) Să se găsească nr-math, nume și prenume pentru studenții ce au obținut cel puțin o notă 10.

SQL:

```
SELECT s.nr-math, s.nume, s.prenume
FROM Student s, Examens e
WHERE s.nr-math = e.nr-math and e.mata = 10.
```

ARO:

$a := \text{Student} \bowtie \text{Examens}$

$b := \sigma_{\text{mata} = 10}(a)$

$\text{rezultat} := \pi_{\text{nr-math}, \text{nume}, \text{prenume}}(b)$

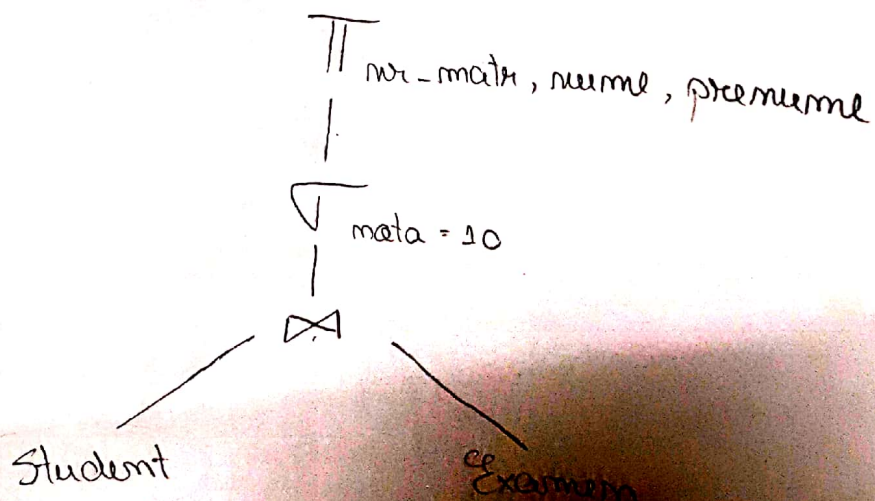
CRT:

$\{ \langle X, Y, Z \rangle \mid \exists S \in \text{Student} \wedge \exists E \in \text{Examens} (S.\text{nr-math} = E.\text{nr-math} \wedge E.\text{mata} = 10 \wedge X.\text{nr-math} = S.\text{nr-math} \wedge X.\text{nume} = S.\text{nume} \wedge X.\text{prenume} = S.\text{prenume}) \}$

CRD:

$\{ \langle X, Y, Z \rangle \mid \exists \langle A, B, C, D, E, F, G \rangle \in \text{Student} \wedge \exists \langle H, I, J, K, L \rangle \in \text{Examens} (A = H \wedge X = A \wedge Y = B \wedge Z = C \wedge L = 10) \}$

Antecede:



c) Să se găsească nume, prenume, și funcție pentru profesorii care au acordat note la cel puțin două materii.

SQL:

```
SELECT p.nume, p.prenume, p.funcție
FROM Profesor p, Examen e1, Examen e2
WHERE p.id-prof = e1.id-prof and
       p.id-prof = e2.id-prof and
       e1.id-mat != e2.id-mat
```

ARO:

$a := \rho_{E(nume, prenume, id-prof, id-mat1, data1, nota1)} (Examens)$

$b := Examens \bowtie Profesor$

$c := a \bowtie b$

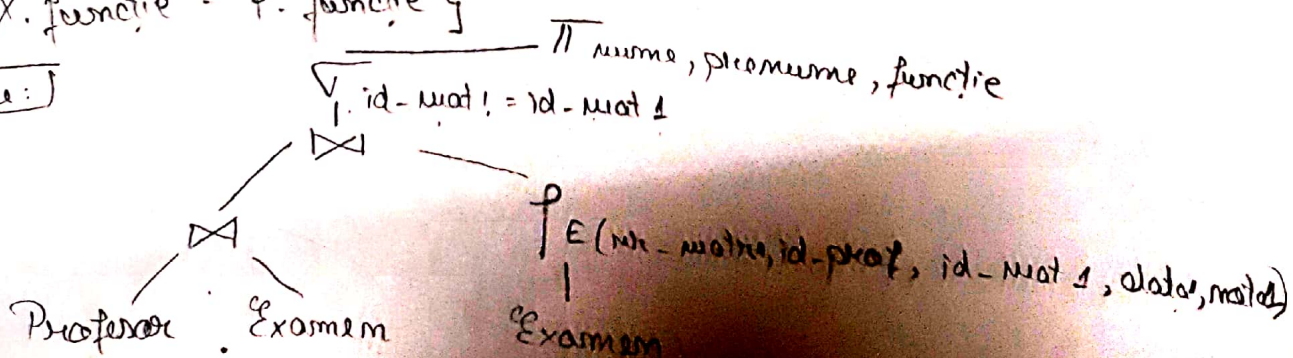
$c := \sigma_{id-mat1 \neq id-mat2}(c)$

$\pi_{nume, prenume, funcție}(c)$

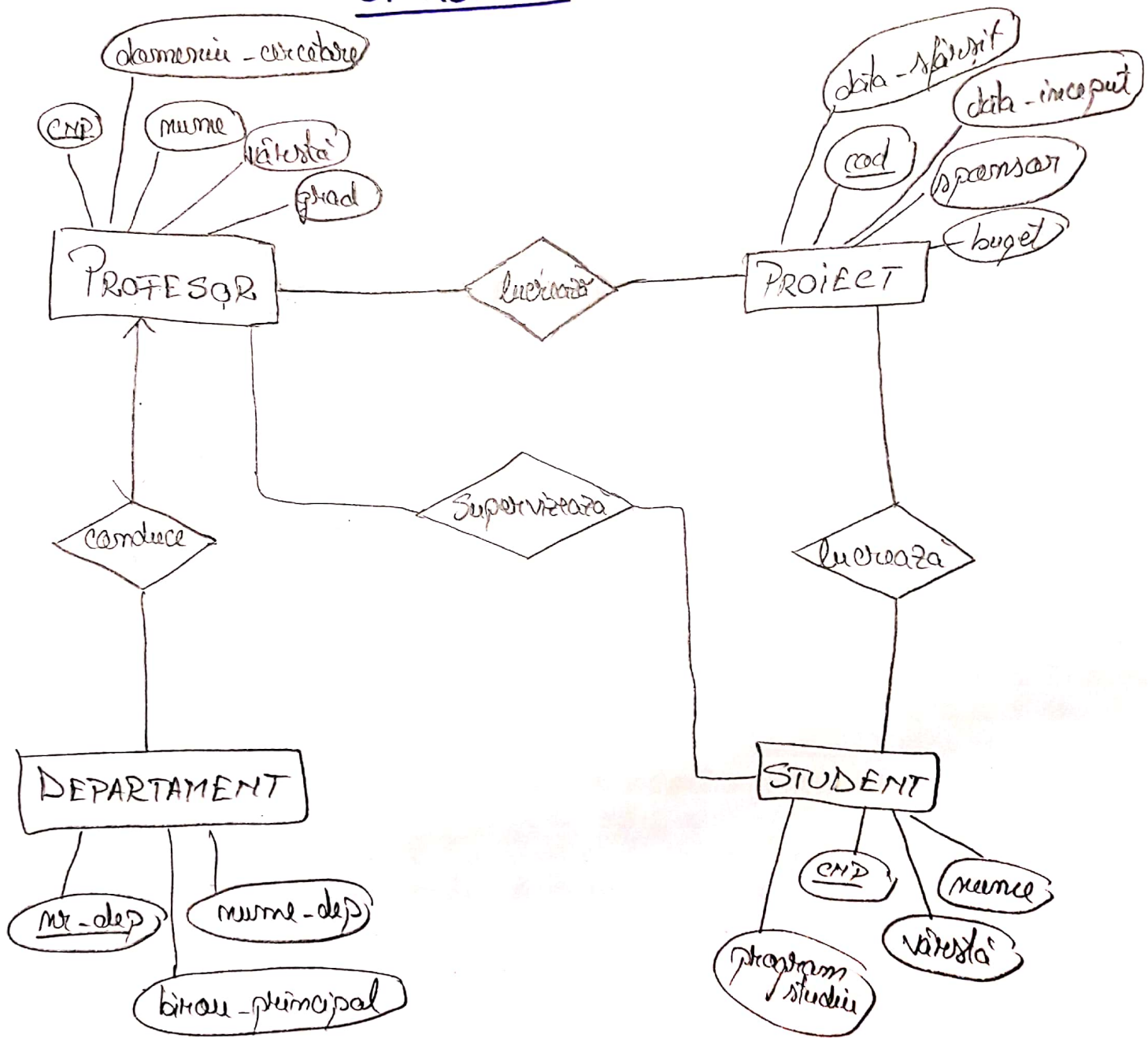
CRD: $\{ \langle x, y, z \rangle \mid \exists \langle A, B, C, D, E \rangle \in Profesor \wedge \exists \langle F, G, I, J, K \rangle \in Examen \wedge \exists \langle L, M, N, O, P \rangle \in Examen (A=B \wedge A=M \wedge I \neq N \wedge x=F \wedge y=G \wedge z=O) \}$

CRT: $\{ x.nume, x.prenume, x.funcție \mid \exists P \in Profesor \wedge \exists E_1 \in Examen \wedge \exists E_2 \in Examen (P.id-prof = E_1.id-prof \wedge P.id-prof = E_2.id-prof \wedge E_1.id-mat \neq E_2.id-mat \wedge x.nume = P.nume \wedge x.prenume = P.prenume \wedge x.funcție = P.funcție) \}$

Alteza:



Subiect 1



Subject 1

XML

<?xml version = "1.0" standalone = "yes"? >

< Universitate >

< Proiect >

< cod > 01 </cod >

< data - inciput > 01-02-2016 </data - inciput >

< data - sfarsit > 01-02-2018 </data - sfarsit >

< buget > 250.000.000 </buget >

< sponsor > Porsche </sponsor >

< Profesor >

< CNP > 27802030015 </CNP >

< nume > IONEL GIGAN </nume >

< varsta > 30 </varsta >

< grad > 1 </grad >

< domeniul > CNC </domeniul >

< Departament >

< nr - dep > 299 </nr - dep >

< nume - dep > ABC </nume - dep >

< birou > 1C </birou >

</Departament >

< Departament >

< nr - dep > 298 </nr - dep >

< nume - dep > ABC </nume - dep >

< birou > 2C </birou >

</Departament >

< Student >

< CNP > 2992620013 </CNP >

< nume > Pop Vasile </nume >

< varsta > 19 </varsta >

< program > 12 </program >

</Student>

<Student>

<CNP> 29825030011 </CNP>

<nume> Sic Ana </nume>

<varsta> 20 </varsta>

<program> 10 </program>

</Student>

</Proiect>

</Universitate>

Subiect 1

c) Să se dea expresia XQUERY pt. a afișa pt. studentul "Ionel Pop" proiecte la care lucrează ca supervisor "Matei Mîrcea".

```
let $p := document("Universitate.xml")/project
for $profesor in $p/profesor
for $departament in $p/departament
for $student in $p/student
where $student/@nume = "Ionel Pop" and
      $profesor/@nume = "Matei Mîrcea"
```

```
return <ret> { $p/@cod } -
</ret>
```