

Subiect 6

[1] Descrieți funcționalitatea de control a accesului concurent la date pentru un SGBD.

Controlul accesului concurent la date este una dintre funcțiile SGBD-ului, unde fiecare utilizator are impresia că lucrează de unul singur (serializarea operațiilor).

[2] Câte nivele de independență a datelor cunoașteți pt. un SGBD?

Aplicațiile sunt izolate față de modificările la nivel conceptual sau la nivel fizic prin cele trei nivele de abstractizare (nivel intern, conceptual și extern).

Independența logică a datelor:

- vederea (view în modelul relational, tabele virtuale, schema externă) asigură posibilitatea modificării structurii datelor (schema conceptuală), acest lucru fiind ascurat aplicațiilor

Independența fizică a datelor:

- schema conceptuală asigură posibilitatea modificării aranjării datelor pe suport secundar sau a indexilor, acest lucru fiind de asemenea ascurat aplicațiilor.

[3] Dați un exemplu de OUTER-JOIN în SQL.

ex. left outer join:

"left outer join" →

```
SELECT tabela1.colonaa, tabela2.colonaa
FROM tabela1, tabela2
WHERE tabela1.colonaa = tabela2.colonaa(+);
```

[4] Dați exemplu de o procedură PSM.

PSM ("persistent stored modules", permite stocarea de proceduri ca elemente ale schemei BD.

PSM = un amestec de instrucțiuni convenționale în SQL.

```
CREATE PROCEDURE JoeMenu(
    IN b CHAR(20),
    IN p REAL)
INSERT INTO Sells VALUES ('Joe's Bar', b, p);
```

[5] Care este mecanismul embedded SQL?

Solua de bază: Un preprocesor translatează instrucțiunile SQL în apeluri de proceduri ce se încadrează în codul limbajului gazdă.

Toate instrucțiunile "embedded SQL" încep cu EXEC SQL, în așa fel încât preprocesorul să le detecteze și să le proceseze.

```
EXEC SQL BEGIN DECLARE SECTION;
      <declarații limbaj gazdă>
EXEC SQL END DECLARE SECTION;
```

6) Explicați diferența între ghilimele și apostrof în PHP.

Ghilimele semnifică "se înlocuiește"
~~Apostroful~~ semnifică "nu se înlocuiește".
Apostrofafele

7) Definiți entitatea "weak" în contextul diagramei ER.

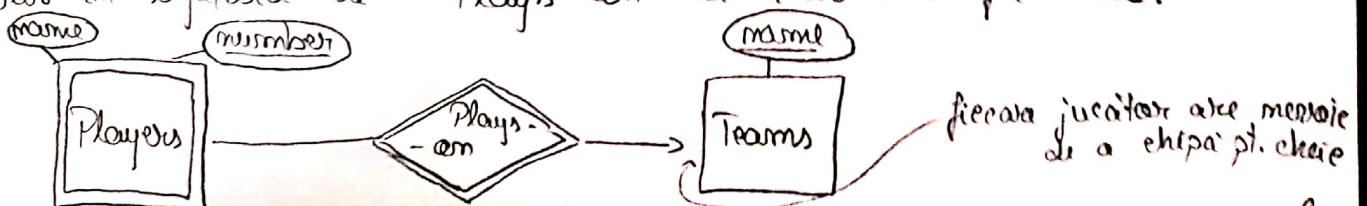
Ocazional, entitățile unei set entitate
necesită "ajutor" pentru a le identifica în
mod unic

Setul de entitate E se spune că este
"weak" dacă pt. a identifica unic entitățile din E,
este necesar de a urmări relații de legătură
"many-one" plecând de la E și să se
include cheia entităților din seturile
entitate conectate.

Ex: name → este aproape o cheie pt. jucători
de fotbal, dar pot exista doi jucători cu același "name".

number → este un lucru ce nu poate fi cheie,
deoarece jucători din 2 echipe pot avea același "number"

Dar numărul de pe tricou, împreună cu numele echipei
pus în legătură de "Plays-on" ar trebui să fie unic.



linie dubla pt. numărul relației de leg. N:1 multinaționale.
 Linie dubla pt. dreptunghiul setului entitate "weak".

8) Ce este o diagramă "GRANT"? Dați un exemplu.

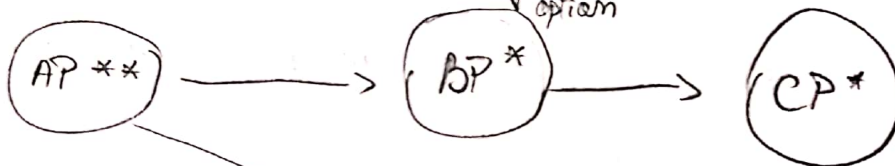
Moduri = utilizator / privilegii / opțiune "grant"?
 este "owner"?

- UPDATE ON R, UPDATE (a) ON R n
 UPDATE (b) ON R aparțin la moduri diferite.
- UPDATE ON R n SELECT ON R WITH
 GRANT OPTION aparțin la moduri diferite.

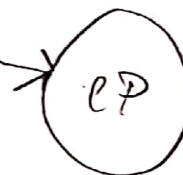
Acela $X \rightarrow Y$ semnifică modul X a fost folosit
 pentru "grant" Y.

A: Grant P to
 B with grant
 option

B: Grant P to C
 with grant option



A este proprietarul
 obiectului pt. care se
 acordă privilegiul P.



A: Grant P
 to C.

9) Descrieți XSLT.

XSLT (Extensible Stylesheet Language - Transformations) este un alt limbaj de procesare a documentelor XML. La origine era intenționat ca un limbaj pt. prezentare: să transforme XML într-o pagină HTML care să fie afișată.

Dar poate de asemenea să transforme XML → XML, astfel servind ca un limbaj de interacțiune.

La n. schema XML, un program XSLT este el însuși un document XML.

XSLT are un namespace special de tag-uri, de obicei indicat prin xsl::

10) Care sunt caracteristicile pentru alegerea modelului NOSQL?

1) Analizarea datelor care vor fi stocate
VOLUM MARE, VALOARE SCĂZUTE

2) Analizarea schemei aplicației
DINAMIC.

Subiect 6

(12) a) Se va găsi nr-matr^{nume} și prenume pt. studenții născuți în data '01.01.1998' sau cu codul 1114.

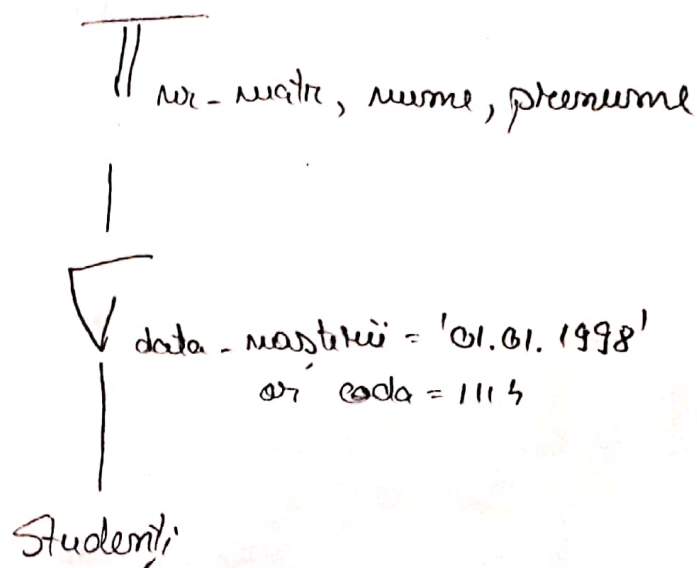
[SQL:] SELECT nr-matr, nume, prenume
FROM Studenți
WHERE data-nasterii = '01.01.1998' or codul = 1114.

[APD] $\Pi_{nr-matr, nume, prenume} \left(\sigma_{data-nasterii='01.01.1998' \vee codul=1114} (Studenți) \right)$

[CRT:] $\{ S.nr-matr, S.nume, S.prenume \mid S \in Studenți \wedge (S.data-nasterii = '01.01.1998' \vee S.codul = 1114) \}$

[RDB:] $\{ \langle A, B, C \rangle \mid \langle A, B, C, D, E, F, G \rangle \in Studenți \wedge (E = '01.01.1998' \vee G = 1114) \}$

[Altecare:]



b) Să se găsească numele, prenumele studenților cu nota mai mare decât nota studentului cu nr-matru 123 la aceeași materie și de la același profesor.

SQL:

```

SELECT nume, prenume
FROM Student s1, Examens e1
WHERE s1.nr-matru = e1.nr-matru and
      e1.nota > (SELECT e2.nota
                  FROM Examens e2
                  WHERE e2.nr-matru = 123 and
                        e2.id-matru = e1.id-matru
                        and e2.id-prof = e1.id-prof)
  
```

ARO:

a := $\rho_{E_1(nr-matru, id-prof, id-matru, data, nota)}$ (Examens)

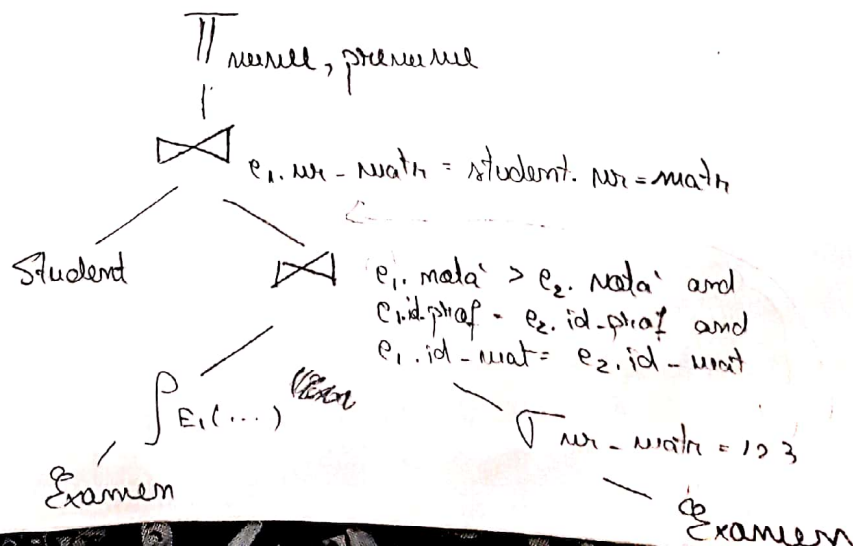
b := $\sigma_{nr-matru = 123}$ (Examens)

c := a \bowtie $\begin{matrix} nota > nota' \\ id-prof = id-prof' \\ id-matru = id-matru' \end{matrix}$ b

d := Student \bowtie $nr-matru = Student.nr-matru$ c

rez := $\pi_{nume, prenume}(d)$

Antecede:



CRT:

$\{ X.name, X.prenume / \exists E_1 \in Examen \wedge \exists E_2 \in Examen$
 $\wedge S \in Student (X.name = S.name \wedge X.prenume = S.prenume \wedge$
 $\wedge S.nr_matr = E_1.nr_matr \wedge E_2.nr_matr = 123 \wedge E_1.noia > E_2.noia$
 $\wedge E_2.id_matr = E_1.id_matr \wedge E_2.id_prof = E_1.id_prof) \}$

CRD:

$\{ \langle Y, Z \rangle / \exists \langle A, B, C, D, E \rangle \in Examen \wedge$
 $\wedge \exists \langle F, G, H, I, J \rangle \in Examen \wedge \exists \langle K, L, M, N, O, P, Q \rangle \in Student$
 $(Y=L \wedge Z=M \wedge K=A \wedge F=123 \wedge C=H \wedge B=G \wedge E>J) \}$

c) Sa se gasca cada pentru adresa la care locuiesc
o singura persoana (student sau profesor).

SQL:

```
SELECT a.coda
FROM Adresa a
WHERE a.coda NOT IN (SELECT s1.coda
                      FROM Student s1, Student s2
                      WHERE s1.nr_matr != s2.nr_matr
                        and s1.coda = s2.coda)
AND a.coda NOT IN (SELECT p1.coda
                   FROM Profesor p1, Profesor p2
                   WHERE p1.id_prof !=
                     p2.id_prof and
                     p1.coda = p2.coda)
AND a.coda NOT IN (SELECT p1.coda
                   FROM Profesor p1, Student s
                   WHERE p1.coda =
                     s.coda)
```


ARC:

$a := \rho_{s1}(\dots), \text{Student}$

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

$c := \sqrt{\text{wr_math}! = \text{wr_math}!}^{(b)}$

$d := \Pi_{\text{coda}}(c)$

$a_1 := \rho_{p1}(\dots), \text{Professor}$

$b_1 := \text{Professor} \bowtie a_1$

$c_1 := \sqrt{\text{id_prof}! = \text{id_prof}!}^{(b_1)}$

$d_1 := \Pi_{\text{coda}}(c_1)$

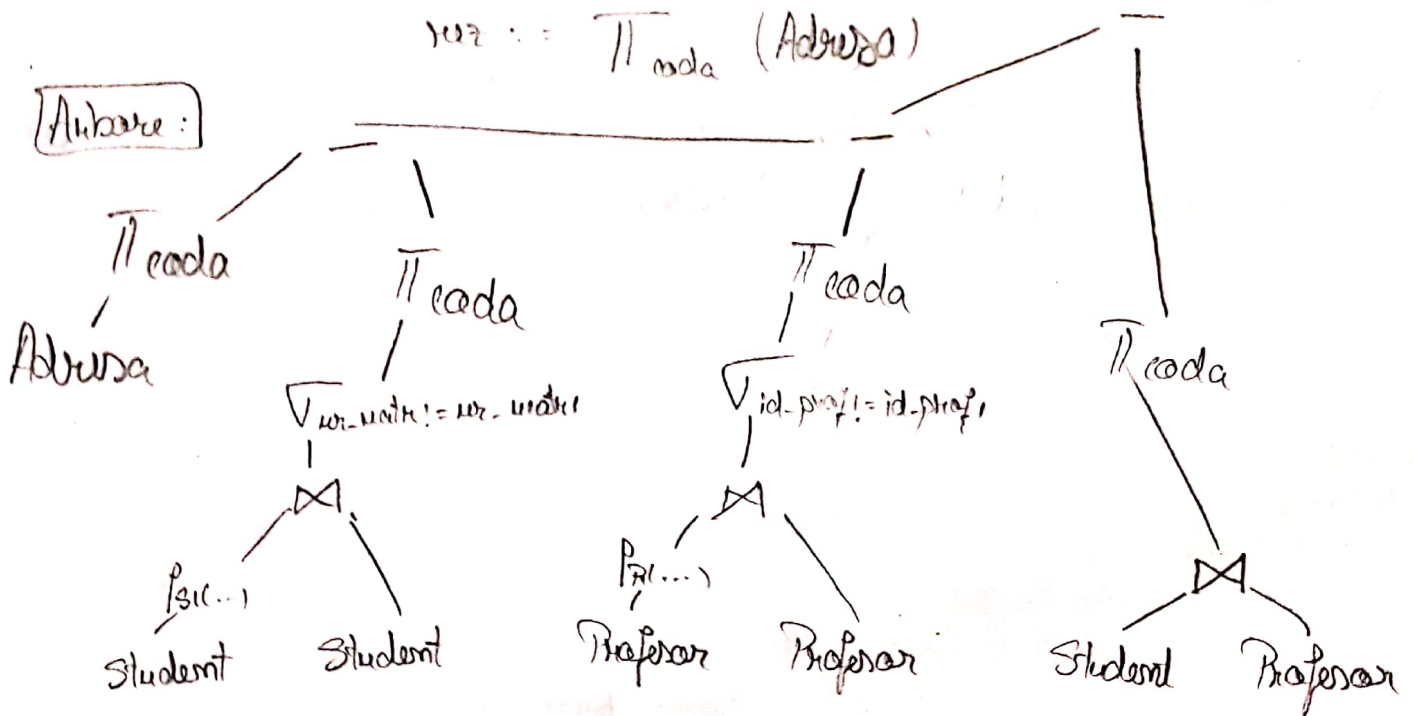
$a_2 := \text{Student} \bowtie \text{Professor}$

$d_2 := \Pi_{\text{coda}}(a_2)$

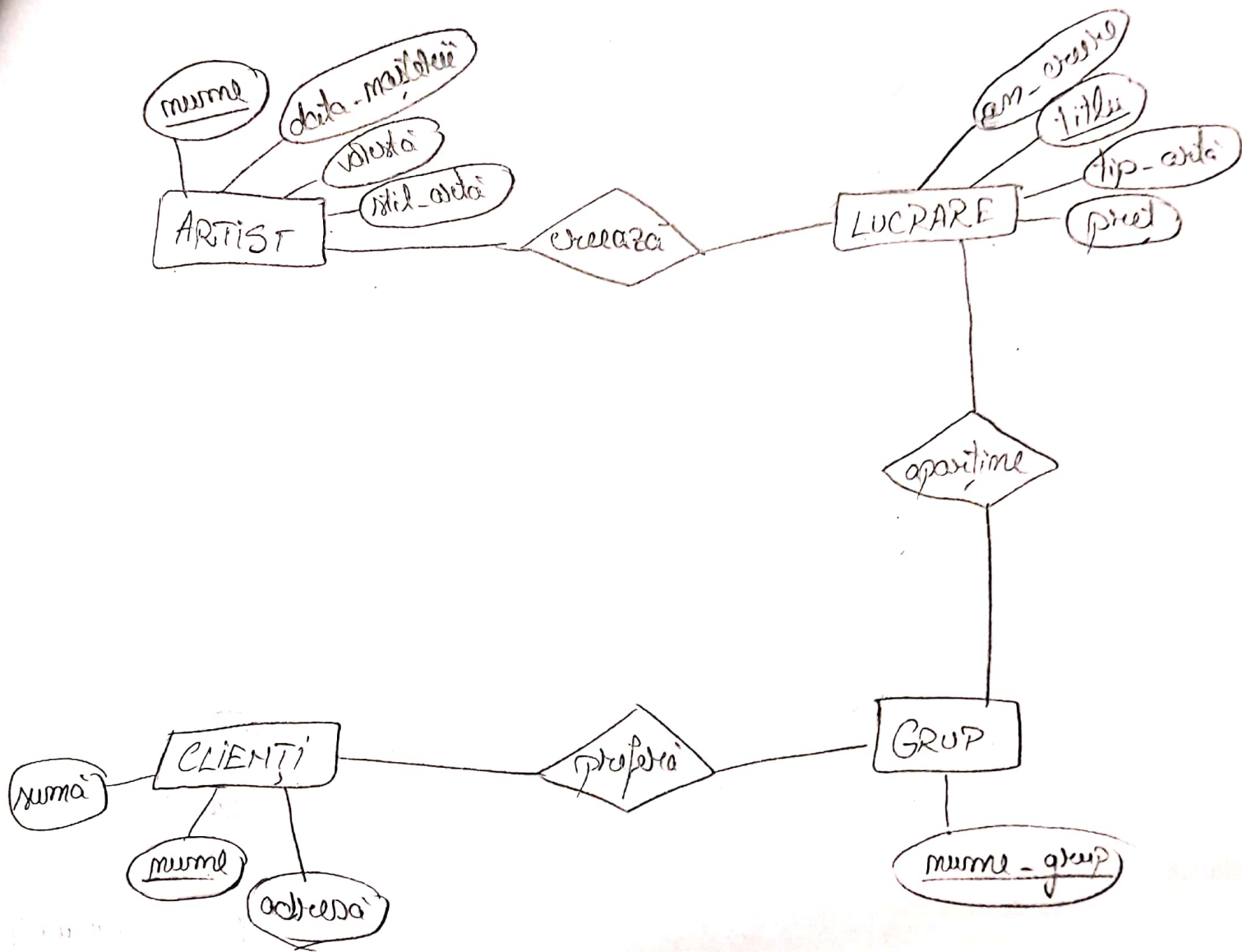
~~res1~~ $\text{res1} := \text{res} - d - d_1 - d_2$

$\text{res} := \Pi_{\text{coda}}(\text{Address})$

Address:



Subiect 6



Subject 6

c) Să se dea expresia XQUERY pt. a găsi lucrările de artă ale artistului "Ion Vlașiu" cumpărate de clientul "Andrei Nicolae".

```
let $d := document("Galerie.xml")/group
for $lucrare in $d/lucrare
for $artist in $d/artist
for $client in $d/client
where $artist/@nume = "Ion Vlașiu" and
      $client/@nume = "Andrei Nicolae"
return <BBP lucrare = { $d/@title }>
</BBP>
```