

SUBIECTUL 2

[1] Definiți ce este un model de date.

Un model de date este: o reprezentare matematică a datelor.

ex: modelul relational = tabele;

modelul semistructurat = arbori/grafuri

- operații cu datele
- constrângeri

[2] Explicați funcționalitatea de import și export a datelor pentru un SGBD.

Importul și exportul datelor sunt niște funcții ale SGBD-ului. Acestea ^{realizează} ~~fac~~ conversia datelor pentru prelucrarea ~~într-un~~ alt SGBD sau cu aplicații terțe (de ex. Excel).

[3] Nivelele interne de abstractizare al unui SGBD.

Există 3 nivele de abstractizare:

- 1) nivel extern
- 2) nivel conceptual
- 3) nivel intern

La nivel intern se află schema fizică care specifică detalii suplimentare legate de stocarea datelor.

Menționează modelul în care tabelele (la modelul relational) descrise prin schema conceptuală sunt stocate pe dispozitive suport secundar, discuri sau benzi magnetice.

Describe tipul fisierelor pentru stocarea pe suport secundar și crearea unor structuri auxiliare de date numite indici în scopul recuperării mai rapide a datelor.

[4] Clauzele GROUP BY, HAVING.

Într-o instrucțiune SELECT-FROM-WHERE se poate folosi clauza GROUP BY cu o listă de atribute, însoțită de la sfârșit.

Tuplele din relația rezultat pentru blocul SELECT-FROM-WHERE sunt grupate conform valorilor atributelor prezente în clauza GROUP BY și se poate aplica orice agregare pe fiecare grup în parte.

ex. cu GROUP BY :

```
SELECT beer, AVG(price)
FROM Sells
GROUP BY beer;
```

CLAUZA HAVING:

- HAVING < condiție > se poate utiliza după clauza GROUP BY.
- dacă apare această clauză, condiția se aplică fiecărui grup și grupurile ce nu respectă condiția sunt eliminate din rezultat.

Ex. cu HAVING:

```
SELECT beer, AVG(price)
FROM Sells
GROUP BY beer
HAVING COUNT(bar) >= 3 OR beer IN (SELECT name
FROM Beers
WHERE name = 'Pils')
```

5) La ce operații sunt testate constrângerile la nivel atribut / Tupla?

Constrângeri la nivel atribut:

- se adaugă CHECK (<condiție>) la declarația pentru un atribut
- condiția poate folosi numele atributului, dar orice altă relație sau atribut ce apar trebuie folosite într-o interogare imbricată.

Ex: CREATE TABLE Sells(
bar CHAR(20),
beer CHAR(20) CHECK (beer IN
(SELECT name FROM Beers)),
price REAL CHECK (price <= 5.00));

- sunt aplicate în momentul când se efectuează o adăugare (insert) sau modificare (update).

Constrangeri la nivel tuple:

- CHECK(<conditie>) poate fi adăugată ca element al schemei de relație.
- condiția poate face referință la orice atribut al relației => alt atribut sau relație necesită o interogare imbricată
- se modifică doar la insert și update

Ex: CREATE TABLE Sells(
bar CHAR(20),
beer CHAR(20),
price REAL,
CHECK (bar = 'Joe's Bar' OR price <= 5.00))

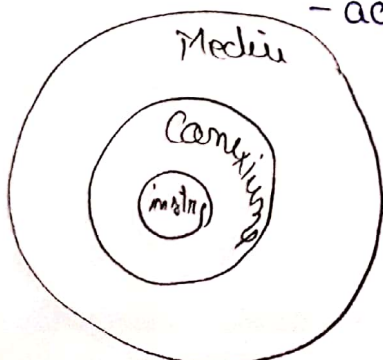
6) Medii, conexiuni, interogări

BD este pentru majoritatea limbajelor de acces la baze de date, un mediu ("environment").

Serverele BD întregim un număr de conexiuni, astfel încât serverele de aplicații pot lansa interogări sau pot efectua actualizări.

Serverul de aplicație emite instrucțiuni:

- interogări
- actualizări



7) Sub ce formă poate fi folosită o interogare SELECT-FROM-WHERE în PSM?

PSM: ("persistent stored modules") : permit stocarea de proceduri ca elemente ale schemei BD.

- un amestec de instrucțiuni convenționale (if, while, etc.) și SQL.

Formatul de bază: CREATE PROCEDURE <nume>(
 <listă de parametri>
 <declarații locale opționale>
 <corp>;

În general interogările SELECT-FROM-WHERE **NU** sunt permise în PSM.

Există 3 moduri pt. a obține efectul unei interogări:

- 1) interogările ce produc o singură valoare pot fi utilizate ca expresie într-o atribuiri
- 2) SELECT ... INTO având rezultat 1 tuplu.
- 3) Căutare.

8) Care sunt operatorii primitivi din algebra relațională?
Argumentați.

Operatorii primitivi:

- Reuniune, \cup
- Diferență
- Produsul cartezian, \Join
- Seleția
- Proiecția
- Redenumirea

usc m

JOIN
c. rel);

9) Explicați ID, IDREF și IDREFS în contextul unui document XML.

ID și IDREF-uri

- Atributele pot fi pautate de la un atribut la altul.
- Permite structurii unui document XML să fie un graf, în loc de arbore.

Crearea ID-urilor:

- presupunem un element E cu un atribut A de tip ID.
- at. când se folosește tag-ul <E> într-un document XML, atributul A primește o valoare unică.

ex: <E A = "xyz" >

Crearea IDREF-urilor:

- pt. a permite elementelor de tip F să facă referințe la un alt element cu un atribut ID, F primește un atribut de tip IDREF.
- Sau, atributul are tipul IDREFS, astfel încât elementul F poate face referințe la orice alte elemente

„RBD”?

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 2

12) a) Să se găsească nr-math, nume și prenume pt. studenții de sex 'F' din grupa 3221.

(SQL):
SELECT nr-math, nume, prenume
FROM Student
WHERE sex = 'F' and grupa = 3221

(ARJ):
 $\Pi_{nr-math, nume, prenume} (\sigma_{sex='F' \text{ and } grupa=3221} (Student))$

(ERT):
 $\{ S.nr-math, S.nume, S.prenume \mid S \in Student \wedge S.sex = 'F' \wedge S.grupa = 3221 \}$

(ERD):
 $\{ \langle A, B, C \rangle \mid \langle A, B, C, D, E, F, G \rangle \in Student \wedge D = 'F' \wedge F = 3221 \}$

(Albore):
 $\Pi_{nr-math, nume, prenume}$
 \mid
 $\sigma_{sex='F' \text{ and } grupa=3221}$
 \mid
Student

b) Sa se gaseasca tripleta nume student, nume profesor, denumire materie cu conditia ca studentul sa primeasca nota de la profesor la materie.

SQL:

SELECT s.nume, p.nume, m.denumire
FROM Student s JOIN Examen e ON (s.id-matn = e.id-matn)
JOIN Profesor p ON (p.id-prof = e.id-prof) JOIN
Materie m ON (m.id-mat = e.id-mat)

ARO:

a := ρ (id-prof, nume, prenume, functie, nota) ^(Profesor)

b := Student \bowtie Examen

c := b \bowtie a

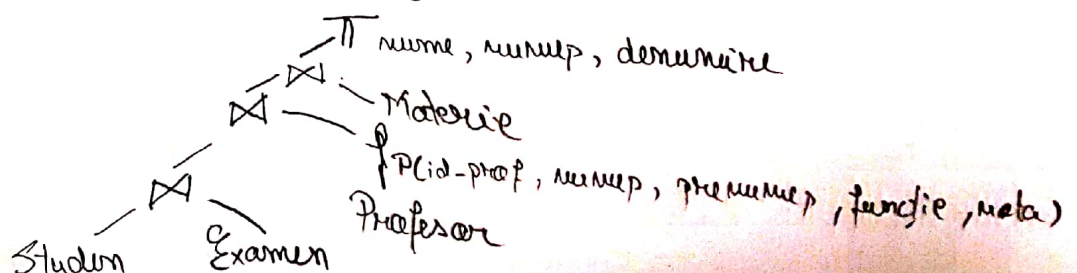
d := c \bowtie Materie

rez := Π nume, nume, denumire (d)

ERT: $\{ X.nume, X.nume, X.denumire / \exists S \in Student \wedge \exists E \in Examen^{\wedge}$
 $\wedge \exists M \in Materie \wedge \exists P \in Profesor (S.id-matn = E.id-matn \wedge$
 $\wedge P.id-prof = E.id-prof \wedge M.id-mat = E.id-mat \wedge$
 $\wedge X.nume = S.nume \wedge X.nume = P.nume \wedge X.denumire =$
 $= M.denumire) \}$

CRD: $\{ \langle x, y, z \rangle / \exists \langle A, B, C, D, E, F, G \rangle \in Student \wedge \exists \langle H, i, j, k, L \rangle \in Examen$
 $\wedge \exists \langle M, N, O \rangle \in Materie \wedge \exists \langle P, Q, R, S, T \rangle \in Profesor (x = B \wedge y = Q \wedge$
 $\wedge z = N \wedge A = H \wedge P = i \wedge j = M) \}$

Arbore:



c) Să se găsească cada pt. adresă la care locuiesc doi studenți diferiți.

SQL:

```
SELECT a.cada
FROM Adresa a
WHERE a.cada IN ( SELECT s1.cada
FROM Student s1, Student s2
WHERE s1.nr-matr != s2.nr-matr
and s1.cada = s2.cada )
```

ARO:

$a := \rho_{s1(nr-matr, nume, prenume, sex, data-nasterii, grupa, cada)}$ (Student)

$b := a \bowtie_{cada=cada1} Student$

$c := \sigma_{nr-matr != nr-matr1}(b)$

$d := Adresa \bowtie_{cada=cada1}(c)$

$rez := \pi_{cada}(d)$

CR1:

$\{ X.cada /$

SQL:

```
SELECT s1.cada
FROM Student s1, Student s2
WHERE s1.nr-matr != s2.nr-matr and s1.cada = s2.cada;
```

ARO:

$a := \rho_{s1(nr-matr, nume, prenume, sex, data-nasterii, grupa, cada)}$ (Student)

$b := a \bowtie_{cada=cada1} Student$

$c := \sigma_{nr-matr != nr-matr1}(b)$

$rez := \pi_{cada}(c)$

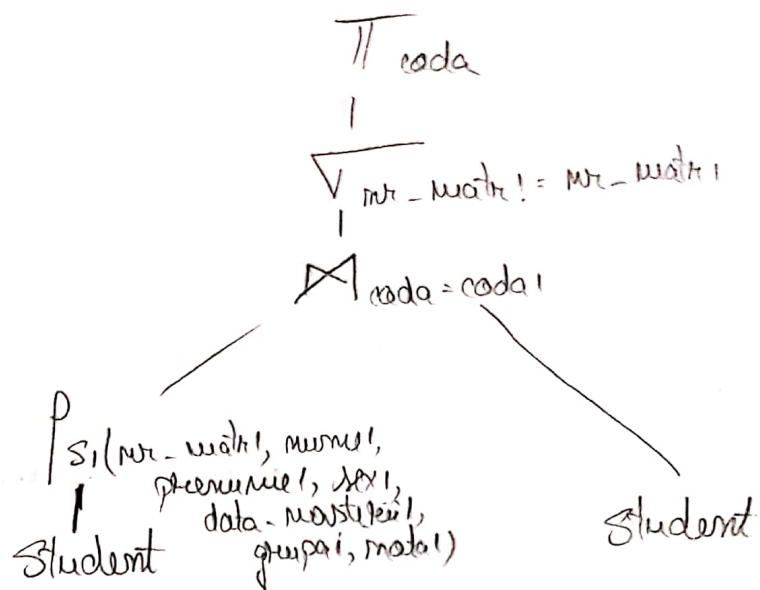
CR1:

$\{ X.cada / \exists s1 \in Student \wedge \exists s2 \in Student (X.cada = s1.cada \wedge s1.nr-matr != s2.nr-matr \wedge s1.cada = s2.cada) \}$

CRD:

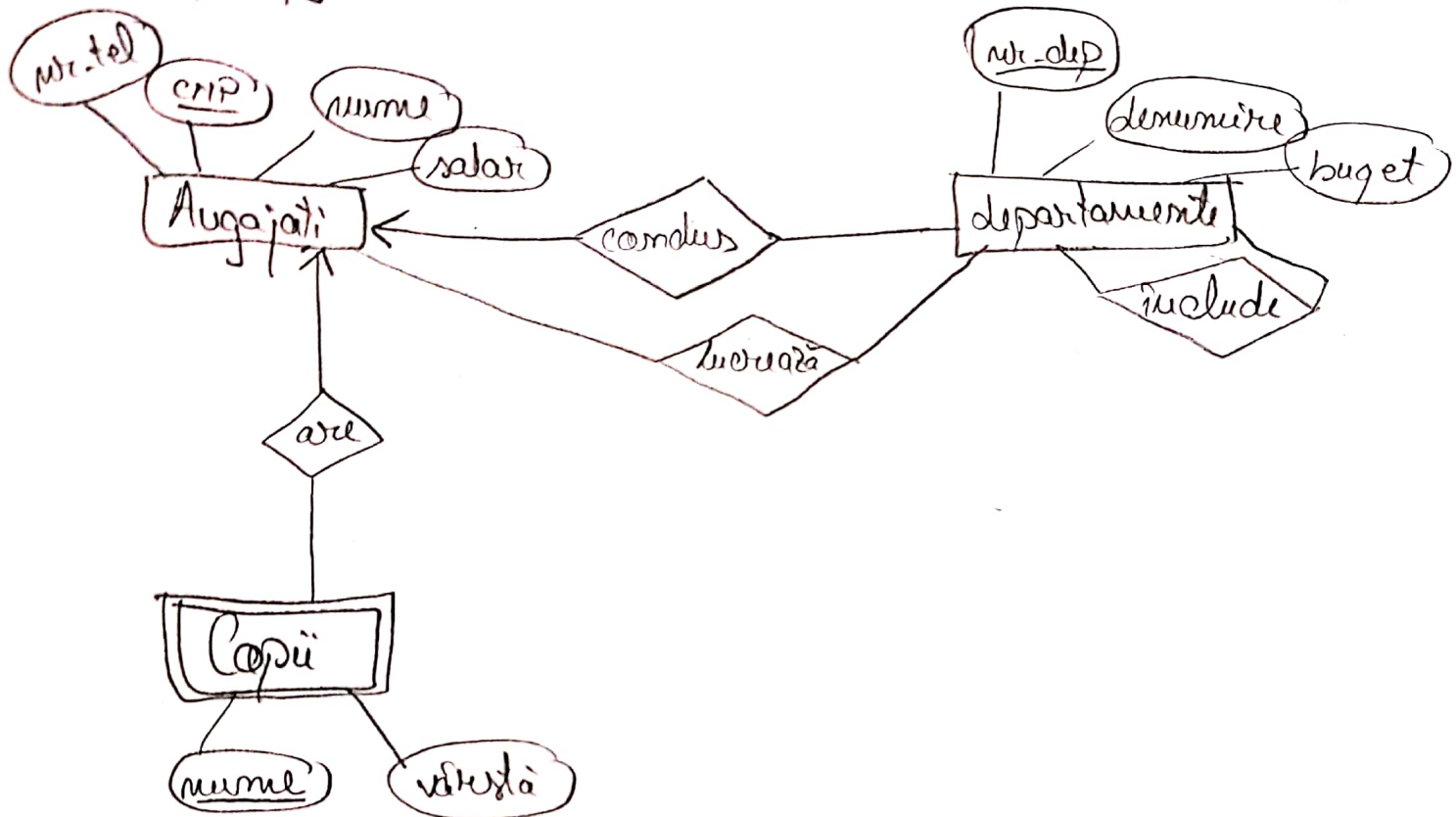
$\{ \langle X \rangle / \exists \langle A, B, C, D, E, F, G \rangle \in Student \wedge \exists \langle H, I, J, K, L, M, N \rangle \in Student (A != H \wedge G = N \wedge X = G) \}$

Answer :



Subiect 2

13) a) ER



XML

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

<Companie>

<Angajat>

<CNP> 26821050011 </CNP>

<nume> Viorela Matei </nume>

<salariu> 10000 </salariu>

<nr-tel> 0758276561 </nr-tel>

<Departament>

<nr-dep> 0015 </nr-dep>

<denumire> CC </denumire>

<buget> 500000 </buget>

<Departament>

<nr-dep> 0001 </nr-dep>

<denumire> C1 </denumire>

<buget> 100000 </buget>

</Departament>

<Departament>

<nr-dep> 0002 </nr-dep>

<denumire> C2 </denumire>

<buget> 50000 </buget>

</Departament>

</Departament>

<Copil>

<nume> Pop Ion </nume>

<varsta> 10 </varsta>

</Copil>

<Copil>

<nume> Pop Ana </nume>

<varsta> 9 </varsta>

</Companie> </Angajat>

</Copil>

Sub2

XQUERY

Sa' M dra experia XQUERY pl. a ofia copii
nuo 18 ani ai angajati "Victoria Motel" 18 ani copii

let \$C = document ("Compasie.xml") / Compasie
for \$angajat in \$C / angajat
for \$copii in \$angajat / copii
where \$angajat / @ nume = "Victoria Motel"
and \$copii / @ varsta < 18

return <ret> { \$copii / @ nume }
< /ret >