

Probleme

Curs 6

Examen

- Pentru examen trebuie să intrați pe **Moodle cu contul de scs** și să vă înscrieți la cursul **BD Postuniversitar**
- Vă rog ca cel târziu până joi să vă înscrieți pe Moodle la curs. Astfel dacă apar probleme le putem rezolva cel târziu vineri.
- În timpul examenului vom fi și în call pe **Zoom** unde puteți pune întrebări dacă e ceva neclar, dar va fi un **ALT LINK DE MEETING**
- **Examenul va avea loc sambata, 27 iunie, ora 9 -11**
- Apoi, de la ora 11 vom ține în continuare curs pe Zoom

Quiz Moodle

- La unele întrebări e posibil să aveți nevoie să scrieți soluția pe foaie, să faceți poză și să încărcați poza.
- Pe Moodle limita pentru un fișier e **1MB**
- Pentru fișiere prea mari trebuie să le transformați în fișiere de dimensiune mai mică. Puteți folosi, de ex., <https://compressjpeg.com/>
- Voi face un quiz pe Moodle de test unde puteți încerca să încărcați un fișier.

Modelare - Magazin

- Un magazin vrea să stocheze informații despre următoarele entități:
- Clienți cu informații personale
- Produse care au o denumire și o descriere.
- Produsele sunt împărțite pe categorii, dar un produs poate să facă parte din mai multe categorii.
- Produsele se găsesc în mai multe culori.
- Magazinul colaborează cu mai mulți distribuitori, pentru că ei pot oferi prețuri diferite pentru produse.
- Pentru fiecare vânzare se stochează informațiile facturii.

- Clienti (**idClient**, nume, prenume, cnp, idAdresa)
- Adrese (**idAdresa**, localitate, strada, codPostal, numar, ap, judet)
- Produse (**idProdus**, numeProdus, descriere)
- Culori (**idCuloare**, numeCuloare)
- ProduseCulori (*idCuloare, idProdus*, cantitate)
- Categoriile (**idCategorie**, denumire, descriere)
- ProduseCategoriile (*idCategorie, idProdus*)
- Distribuitori (**idDistribuitor**, nume, idAdresa)
- ProduseDistribuitori (*idProdus, idDistribuitor*, pretCumparare)
- Factura (**idFactura**, nrTranzactie, idClient, data)
- ItemFactura (**idItem**, *idFactura, idProdus*, cantitate, *idCuloare*, pretVanzare)
 - FK (idProdus, idCuloare) -> ProduseCulori

- Inseerați doi clienți în tabelul Clienti (**idClient**, nume, cnp, idAdresa)

insert into Clienti values (1, 'Pop Ana', '123538549', 3) ,

(2, '...', '.....', 5)

- Modificați adresa (a.î. să fie idAdresa = 5) clientului cu idClient = 3

update Clienti

set idAdresa = 5

where idClient = 3

- Afișați toate produsele și culorile lor.

```
select P.numeProdus, C.numeCuloare
```

```
from Produse P inner join ProduseCulori PC on P.idProdus = PC.idProdus
```

```
inner join Culori C on C.idCuloare = PC.idCuloare
```

- Afișați toate informațiile despre clienți și ce au cumpărat aceștia.

```
select *
```

```
from Clienti C inner join Factura F on C.idClient = F.idClient
```

```
inner join ItemFactura I on I.idFactura = F.idFactura
```

```
inner join ProduseCulori PC on PC.idProdus = I.idProdus and  
PC.idCuloare = I.idCuloare
```

- Afișați produsele din categoria *A* sau *B*

```
select P.numeProdus, C.denumire
```

```
from Produse P inner join ProduseCategorii PC on P.idProdus =  
PC.idProdus
```

```
inner join Categori C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'A' OR C.denumire = 'B'
```


- Afișați produsele care fac parte atât din categoria A, cât și din categoria B

```
select P.numProdus, C.denumire
```

```
from Produse P inner join ProduseCategorii PC on P.idProdus =  
PC.idProdus
```

```
inner join Categorii C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'A'
```

```
intersect
```

```
select P.numProdus, C.denumire
```

```
from Produse P inner join ProduseCategorii PC on P.idProdus =  
PC.idProdus
```

```
inner join Categorii C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'B'
```

~~Where C.denumire = 'A' AND C.denumire = 'B'~~

- Afișați produsele care fac parte atât din categoria *A*, cât și din categoria *B*

```
select P.numeProdus, C.denumire
```

```
from Produse P inner join ProduseCategorii PC on P.idProdus =  
                                                PC.idProdus
```

```
inner join Categorii C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'A' and P.idProdus in (
```

```
    select P.idProdus
```

```
    from Produse P inner join ProduseCategorii PC on P.idProdus =  
                                                PC.idProdus
```

```
    inner join Categorii C on C.idCategori = PC.idCategorie
```

```
    where C.denumire = 'B' )
```

- Afișați produsele care se află în categoria A, dar nu se află în B

```
select P.numeProdus, C.denumire
from Produse P inner join ProduseCategorii PC on P.idProdus =
                                                    PC.idProdus
inner join Categori C on C.idCategori = PC.idCategorie
where C.denumire = 'A' and P.idProdus NOT IN (
select P.idProdus
from Produse P inner join ProduseCategorii PC on P.idProdus =
PC.idProdus
inner join Categori C on C.idCategori = PC.idCategorie
where C.denumire = 'B' )
```

- Afișați produsele care se află în categoria A, dar nu se află în B

```
select P.idProodus, P.numProodus -> (1, p1), (2, p1)
from Produse P inner join ProduseCategorii PC on P.idProodus =
                                                PC.idProodus
```

```
inner join Categorii C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'A'
```

```
EXCEPT
```

```
select P.idProodus, P.numProodus -> (2, p1)
```

```
from Produse P inner join ProduseCategorii PC on P.idProodus =
                                                PC.idProodus
```

```
inner join Categorii C on C.idCategori = PC.idCategorie
```

```
where C.denumire = 'B'
```

- Afișați clienții care au cumpărat produse din categoria A

```
select C.num  
from Clienti C inner join Facturi F on C.idClient = F.idClient  
inner join ItemFactura I on I.idFactura = F.idFactura  
inner join Produse P on P.idProdus = I.idProdus  
inner join ProduseCategorii PC on P.idProdus = PC.idProdus  
inner join Categori C on C.idCategorie = PC.idCategorie  
where C.denumire like 'A%'
```

- Afișați numărul de produse din fiecare categorie

```
select C.idCategorie, C.denumire, count(*)  
from ProduseCategorii PC inner join Categori C on  
    C.idCategorie = PC.idCategorie  
group by C.idCategorie, C.denumire
```

- Afișați prețul produselor în euro, știind că în baza de date e stocat în ron

```
select P.numProduce, I.pretVanzare * 5 as pretVanzareEuro  
from ItemFactura I inner join Produse P on P.idProduce = I.idProduce
```


- Afișați numărul de produse distincte cumpărate de fiecare client.

```
select F.idClient, count (distinct I.idProduce) as nrProduse
from ItemFactura I inner join Factura F on F.idFactura = I.idFactura
group by F.idClient
```

- Afișați numărul de produse (bucăți) cumpărate de fiecare client.

```
select F.idClient, sum(I.cantitate) as nrTotalProduse
from ItemFactura I inner join Factura F on F.idFactura = I.idFactura
group by F.idClient
```

```
select F.idClient, I.idProduce, sum(I.cantitate) as nrBucPerProduce
from ItemFactura I inner join Factura F on F.idFactura = I.idFactura
group by F.idClient, I.idProduce
```


- Afișați suma totală a fiecărei facturi

```
select F.idFactura, F.nrTranzactie, sum(I.pretVanzare * I.cantitate) as  
        sumaFactura  
from ItemFactura I inner join Factura F on F.idFactura = I.idFactura  
group by F.idFactura, F.nrTranzactie
```

- Afișați produsele din categoria A în ordine crescătoare a prețului

```
select P.numeProduse, PD.pretCumparare  
from Produse P inner join ProduseCategorii PC  
        on P.idProduce = PC.idProduce  
inner join Categori C on C.idCategorie = PC.idCategorie  
inner join ProduseDistribuitori PD on PD.idProduce=P.idProduce  
where C.denumire = 'A'  
order by PD.pretCumparare asc
```

- Afișați doar produsele cele mai scumpe

```
select top 1 with ties P.numProduse, PD.pretCumparare  
from Produse P inner join ProduseDistribuitori PD on  
PD.idProdus=P.idProdus  
order by PD.pretCumparare desc
```

```
select P.numProdus, PD.pretCumparare  
from Produse P inner join ProduseDistribuitori PD on  
PD.idProdus=P.idProdus  
where PD.pretCumparare = ( select max(PD.pretCumparare)  
                           from ProduseDistribuitori PD)
```

- Idee 3: PD.pretCumparare >= ALL (..toate preturile..)

- Afișați produsele care sunt mai scumpe decât toate produsele din categoria A

```
select P.numeProduce
from Produse P inner join ProduseDistribuitor PD on
    P.idProduce = PD.idProduce
where PD.pretCumparare > ALL (select PD.pretCumparare
    from Produse P inner join ProduseDistribuitor PD on
        P.idProduce = PD.idProduce
    inner join ProduseCategorii PC
    on PC.idProduce = P.idProduce
    inner join Categorii C on C.idCategorie = PC.idCategorie
    where C.denumire = 'A'
)
```

- Afișați categoriile cu cel puțin 10 produse, ordonate descrescător după numărul de produse

```
select C.idCategorie, C.denumire, count(idProdus) as nrProduse
from Categori C inner join ProduseCategorii PC
on C.idCategorie = PC.idCategorie
group by C.idCategorie, C.denumire
having count(*) >= 10
order by count(*) desc
```

- Afișați clienții care au cumpărat de cele mai multe ori din magazin (o cumpărătură are o factură)

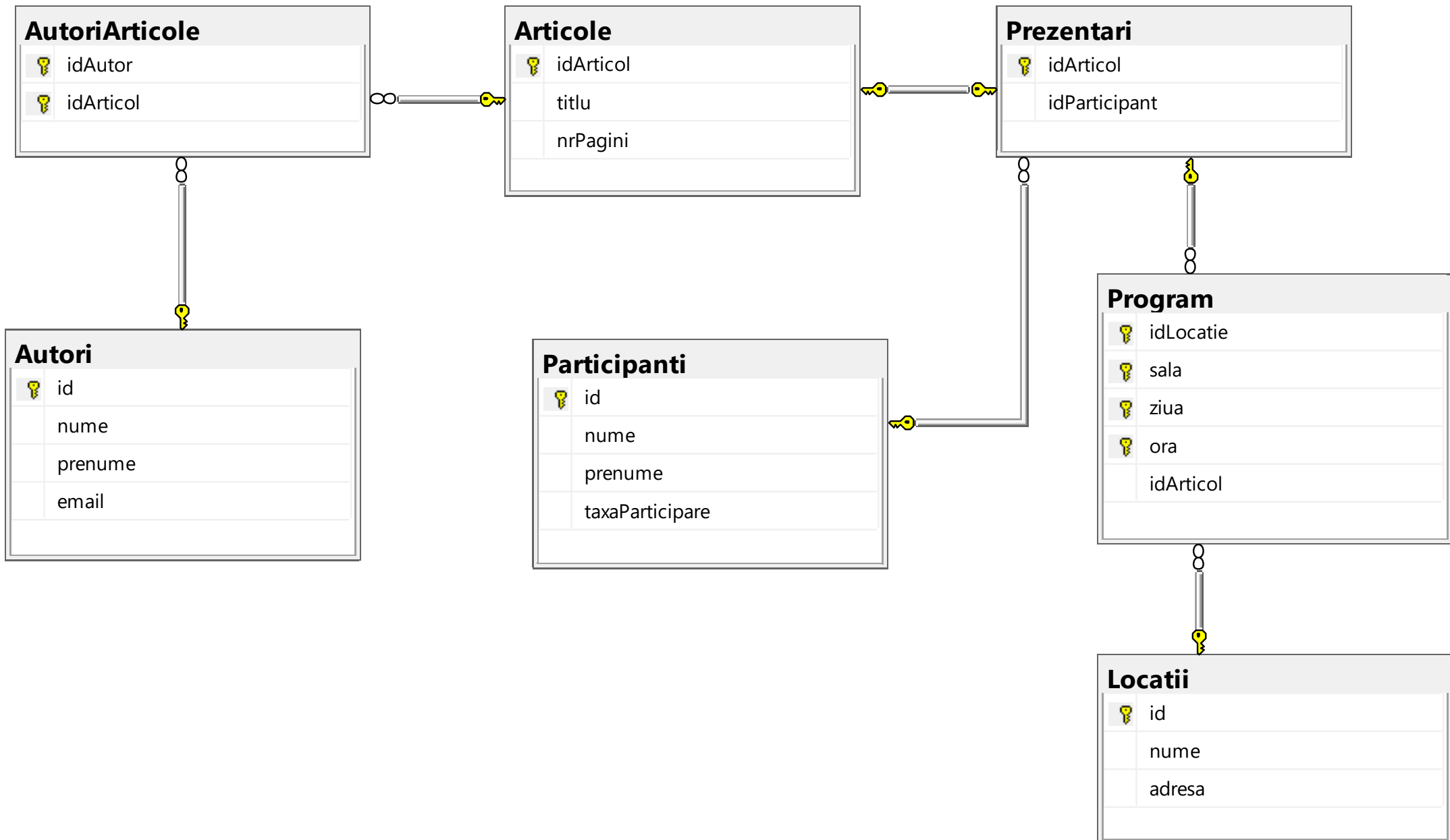
```
select C.idClient, C.num, count(*) as nrFacturi
from Factura F inner join Clienti C on F.idClient = C.idClient
group by C.idClient , C.num
having count(*) >= ALL (
    select count(*)
    from Factura F
    group by F.idClient)
```

- Afișați clienții care au cumpărat de mai multe ori din magazin decât unul din clienții *Popa Anamaria* (daca sunt mai multi)

```
select C.idClient, C.num, count(*) as nrFacturi
from Factura F inner join Clienti C on F.idClient = C.idClient
group by C.idClient , C.num
having count(*) >= ANY (
    select count(*)
    from Factura F inner join Clienti
    on C.idClient = F.idClient
    where C.num = 'Popa'
    and C.prenume = 'Anamaria'
    group by C.idClient
)
```

Modelare - Conferință

- Vrem să ținem evidența unor informații pentru o conferință pentru a gestiona următoarele:
- Autorii articolelor (nume, prenume, email)
- Articolele care urmează să fie prezentate la conferință (titlu, număr pagini, autori)
- Participanții la conferință – nu toți autorii participă la conferință, și nu toți participanții au în mod obligatoriu un articol la conferință
- Fiecare articol trebuie să fie prezentat de către unul din autori (care este și participant la conferință)
- Programul prezentărilor (ziua, ora, sala, locația)



- Afișați autorii (nume și prenume) care nu participă la conferință

- Afișați numele și prenumele participanților împreună cu numărul de articole prezentate de aceștia, pentru participanții care prezintă minim 2 articole

- Afișați autorii de articole (nume și prenume) care participă la conferință.
Obs. Un autor nu participă obligatoriu la conferință, iar un participant la conferință nu e neapărat autor de articol
- Calculați câte prezentări au loc în sala '2/I' de la ora 10 începând în toate zilele conferinței

- Editorii volumului conferinței au decis ca articolele care nu vor fi prezentate la conferință să fie excluse din volumul conferinței. Afișați id-ul și titlul articolelor care nu vor fi prezentate la conferință

- Calculați numărul mediu de pagini pe articol la această conferință

- Calculați câte prezentări au loc în sala '2/I' de la ora 10 începând în toate zilele conferinței

- Afișați titlul articolului cu număr de pagini mai multe ca unul din articolele autorului 'Cionta' (numele)

- Afișați câte prezentări au loc în fiecare sală din fiecare locație - vrem sa vedem numele locației, sala și număr de prezentări
- Afișați articolele cu cele mai multe pagini

- Conferinta are taxe de participare diferite în funcție de statutul participantului: student, profesor, etc. Afișați pentru fiecare tip de taxa câți participanți au plătit taxa respectivă.

- Afișați suma totală încasată de organizatorii conferinței

- Afișați categoria (suma) de taxă de participare pentru fiecare tip de taxă din care s-au strâns cel puțin 800 EUR. Taxele sunt toate în EUR.

- Afișați numele autorilor care au scris mai mult de 1 articol cu minim 10 pagini

- Afișați id-ul participanților și numărul lor de prezentări pentru cei care au mai multe prezentări decât toți participanții care au plătit taxa maximă, adică 100 EUR
- În plus față de cerința anterioară afișați și numele și prenumele participanților

staffNo	dentistName	patientNo	patientName	appointment date time	surgeryNo
S1011	Tony Smith	P100	Gillian White	12-Aug-03 10.00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03 12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sept-03 10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sept-03 10.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03 16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03 18.00	S13

Tabelul dat prezintă anomalii. Dați exemplu de diferite tipuri de anomalii.

?insert – inserăm același staffNo cu alt dentistName

- inserăm un dentist nou?

?update – modificăm dentistName, dar nu în toate aparițiile -> inconsistențe

modificăm patientName...

?delete - ștergem dentistul -> pierdem datele despre pacienți

anulăm o programare -> ștergem datele despre dentist

staffNo	dentistName	patientNo	patientName	appointment date time	surgeryNo
S1011	Tony Smith	P100	Gillian White	12-Aug-03 10.00	S10
S1011	Tony Smith	P105	Jill Bell	13-Aug-03 12.00	S15
S1024	Helen Pearson	P108	Ian MacKay	12-Sept-03 10.00	S10
S1024	Helen Pearson	P108	Ian MacKay	14-Sept-03 10.00	S10
S1032	Robin Plevin	P105	Jill Bell	14-Oct-03 16.30	S15
S1032	Robin Plevin	P110	John Walker	15-Oct-03 18.00	S13

Cum ați normaliza tabelul?

Dentists (**staffNo**, dentistName)

Patients (**patientNo**, patientName)

Appointments (**staffNo**, *patientNo*, **datetime**, *surgeryNo*)

Surgeries (surgeryNo, type)

Forme Normale

Prezentari (idArticol, idParticipant, numeParticipant, prenumeParticipant)

- Cine poate fi cheie primară? (**idArticol, idParticipant**)
 - ?idParticipant : idParticipant -> idArticol, numeParticipant, prenumeParticipant
- Ce dependențe funcționale identificați?
 - idParticipant -> numeParticipant, prenumeParticipant
 - idArticol, idParticipant -> numeParticipant, prenumeParticipant (partiala)
- În ce formă normală e relația următoare? Pot apărea anomalii în relație?
 - 1NF: da
 - 2NF: nu, idArticol, idParticipant -> numeParticipant, prenumeParticipant (partiala)
 - 3NF, BCNF -> nu pt ca nu e in 2NF
- Cum se poate descompune relația?
 - Prezentari(idArticol, idParticipant)
 - Participanti(idParticipant, nume, prenume)

Forme Normale

Articole (idArticol, titlu, nrPagini, autori) , unde autori e lista de autori

- Cine poate fi cheie primară? idArticol
- Ce dependențe funcționale identificați?
 - idArticol -> titlu, nrPagini, autori
- În ce formă normală e relația următoare?
 - 1NF –NU pt ca avem lista de autori
 - 2NF, 3NF, BCNF - NU
- Cum se poate descompune relația?
 - Autori (idAutor, nume)
 - Articole(idArticol, titlu, nrPagini)
 - AturoiArticole (idAutor, idArticol)

Forme Normale

VizualizariFilme (idFilm, titluFilm, regizor, spectator, dataVizualizare)

- Cine poate fi cheie primară? (idFilm, spectator, dataVizualizare)
- Ce dependențe funcționale identificați?
 - idFilm -> titluFilm, regizor
 - idFilm, spectator, dataVizualizare -> idFilm, spectator, dataVizualizare, titluFilm, regizor
- În ce formă normală e relația următoare?
 - 1NF – da
 - 2NF – nu idFilm, ~~spectator, dataVizualizare~~ -> titluFilm, regizor ? Totală?
 - 3NF, BCNF - nu
- Cum se poate descompune relația?
 - VizualizariFilme (idFilm, spectator, dataVizualizare)
 - Filme (idFilm, titluFilm, regizor)

Forme Normale

Medicamente (idMedicament, numeMedicament, pret, producator, producatorAdresa)

- Cine poate fi cheie primară? (idMedicament, producator)
- Ce dependențe funcționale identificați?
 - idMedicament -> numeMedicament
 - idMedicament, producator ->pret
 - Producator -> producatorAdresa
- În ce formă normală e relația următoare?
 - 1NF – da
 - 2NF – nu, pt ca idMedicament, ~~producator~~ -> numeMedicament – parțială
- Cum se poate descompune relația?
 - Medicamente (idMedicament, numeMedicament)
 - Producatori (producator, producatorAdresa)
 - MedicamenteProducatori (idMedicament, producator, pret)

Forme Normale

Vrem ca un medicament sa aibă un singur producator.

Medicamente (idMedicament, numeMedicament, pret, producator, producatorAdresa)

- Cine poate fi cheie primară? idMedicament
- Ce dependențe funcționale identificați?
 - idMedicament -> numeMedicament, pret, producator, producatorAdresa Medicament
 - producator -> producatorAdresa
- În ce formă normală e relația următoare?
 - 1NF – da
 - 2NF – da - pt ca PK are un singur atr
 - 3NF – nu idMedicament -> producator, producator -> producatorAdresa => dep tranz: idMedicament -> producatorAdresa
 - BCNF - nu
- Cum se poate descompune relația?
 - Medicamente (idMedicament, numeMedicament, producator, pret)
 - Producatori (producator, producatorAdresa)

Forme Normale

Concerte (locatie, an, trupa, genMuzical)

- Cine poate fi cheie primară?
- Ce dependențe funcționale identificați?
- În ce formă normală e relația următoare?
- Cum se poate descompune relația?

Forme Normale

Concerte (locatie, an, trupa, numarSpectatori)

- Cine poate fi cheie primară?
- Ce dependențe funcționale identificați?
- În ce formă normală e relația următoare?
- Cum se poate descompune relația?

Forme Normale

Reteta (idMancare, idIngredient, cantitate)

- Cine poate fi cheie primară?
- Ce dependențe funcționale identificați?
- În ce formă normală e relația următoare?
- Cum se poate descompune relația?

Forme Normale

- $R(A, B, C, D)$ cu dependențele funcționale $AC \rightarrow D, A \rightarrow B$
- A, B, C, D – attribute atomare
- Cine poate fi cheie primară? AC
- În ce formă normală e relația următoare?
 - 1NF – da
 - 2NF - nu pt ca $AC \rightarrow B$ parțială
- Cum se poate descompune relația? $A \rightarrow B \Rightarrow A, B \quad R - \{B\}$
 - $R(A, B)$
 - $S(A, C, D)$

Forme Normale

- $R(A, B, C, D)$ cu dependențele funcționale $AC \rightarrow D, D \rightarrow B$
- Cine poate fi cheie primară? AC
- În ce formă normală e relația următoare?
 - 2NF – da $AC \rightarrow D, AC \rightarrow B$ totale
 - 3NF – nu , $AC \rightarrow D, D \rightarrow B \rightarrow$ dep tranz între atr neprim si cheie
- Cum se poate descompune relația?
 - $R(A, C, D)$
 - $S(D, B)$

Forme normale

- Dați exemplu de o relație care este în 1NF, dar nu este în 2NF
 - ProduseFactura(idProdot, denumire, categorie, idFactura)
 - Cheie primară: idProdot, idFactura
 - idProdot, idFactura -> denumire parțială
- Dați exemplu de o relație care este în 2NF, dar nu este în 3NF
 - Factura(idFactura, data, idClient, numeClient)
 - Cheie primara: idFactura
 - 2NF: da , PK – un singur atr
 - 3NF: nu, idFactura -> idClient, idClient->numeClient – dep tranzitivă între numeClient și idFactura

Index

- Se da un index de tip B-arbore pe o cheie compusă (a,b). Pentru care din următoarele condiții poate fi folosit indexul și în ce fel?
 - a) $a > 6$ AND $b = 5$ -> da, pt ambele
 - b) $a < 4$ AND $c < 2$ -> da, dar verific $c < 2$ separat
 - c) $b = 2$ AND $c = 4$ -> nu! Nu am un prefix din cheia de căutare.
 - d) $a < 3$ AND $b > 9$ -> da, pt ambele
 - e) $a = 6$ OR $c = 3$ -> nu

Index

- Se da un index de tip hashtable pe o cheie compusă (a, b). Pentru care din următoarele condiții poate fi folosit indexul și în ce fel?
 - a) $a > 6$ AND $b = 5$ - NU, dacă $a > 6$ nu rezultă că $h_2(a) > h_2(6)$
 - b) $a = 6$ OR $c = 3$ – NU pt ca lipsește valoare la b
 - c) $a = 6$ OR $b = 3$ – NU pt ca avem OR
 - d) $b = 2$ AND $c = 4$ – NU pt ca lipsește valoarea la a
 - e) $a = 3$ AND $b = 9$ – DA
 - f) $a = 5$ – NU pt ca lipsește valoare la b

Index

- Ce indecși pot optimiza fiecare din interogările anterioare?

Feedback

- Socrative – feedback referitor la curs sau sugestii pentru ultimul curs
- Intrați pe site-ul: <https://socrative.com/>
 - Nu aveți nevoie de cont pentru asta
 - Login -> Student Login -> Room Name: BDPOSTUNI
 - Anonim
 - Aveți acolo 5 întrebări scurte (puteți răspunde la care vreți)