**Anexa 2**

**Algebra relationala si Bazele de Date Relaționale BDR**



**1. Noțiuni.** O baza de date relationala (BDR) reprezinta un ansamblu de relatii, prin care se reprezinta atât datele cât si legaturile dintre date. In cadrul BDR, datele sunt organizate sub forma unor tablouri bidimensionale (tabele) de date, numite**relatii**.

Tabelele sunt utilkizate in diferite operatii si servesc la obtinerea unor informatii. Operțiile cele mai frecvente în obținerea informațiilor se numesc **asocieri**.

Asocierile dintre relatii se reprezinta explicit prin ***atribute*** de legatura. Aceste atribute figureaza într-una din relatiile implicate in asociere (de regula, în cazul legaturilor de tip "unu la multi") sau sunt plasate într-o relatie distincta, construita special pentru exprimarea legaturilor între relatii (în cazul legaturilor de tip "multi la multi").

**2. Operatorii Modelului Relational**. OMR definesc operatiile care se pot efectua asupra relatiilor, in scopul realizarii functiilor de prelucrare asupra bazei de date, respectiv **consultarea, inserarea, modificarea si stergerea datelor**.

            **3. Restrictiile de Integritate ale Modelului Relational pentru BDR**. RIMRD permit definirea starilor coerente ale bazei de date, adică tabelele din BDR trebuie să indeplinească principiul fundamental al integrității datelor. **Or, acest principiu are următoarea semnificație:** *orice operații asupra tabelelor din BDR trebuie să fie efectuate în așa mod, încît la efectuarea operațiilor inverse să permită obținerea stărilor inițiale ale tabelelor folosite în operațiile menționate*.

            În continuare, vor fi prezentate pe larg aceste componente.

**4. Elementele de baza ale modelului relațional pentru BDR**

            Prezentarea structurii relationale a datelor pentru BDR impune definirea notiunilor de:

* ***domeniu,***
* ***relatie,***
* ***atribut si***
* ***schema a unei relatii***.

 **4.1 Domeniu**

 ***Definitie:*** Domeniu (eng. Domain) = o multime de valori avand asociat un nume.

 Un domeniu se poate defini explicit, prin enumerarea tuturor valorilor apartinând acestuia sau implicit, prin precizarea proprietatilor pe care le au valorile din cadrul domeniului respectiv.

 ***Exemplu 1***: Domenii concrete

D1= **Culori** = {rosu, galben, albastru, violet, verde}

D2= **Nota** = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} sau Nota = {n ∈ N\* | n ≥ 1 si n ≤ 10}

D3= **Sir40** = {Multimea sirurilor de maxim 40 de caractere}

D4={**Numar** = Multimea numerelor intregi pozitive din intervalul [0, 100000]}

           ***Exemplu 2:*** Sa consideram, și un alt exemplu, de descriere și de prezentare a domeniilor. Fie D1, D2, D3, definite după cum urmează:

D1 = {"F","M"}

D2 = {x / x apartine N, x apartine [0,100] }

D3 = {s/s=sir de caractere}

În cazul lui D1 s-a recurs la o definire explicita ca și in Exemplul 1, în timp ce pentru D2 si D3 s-a utilizat definirea implicita.

            Pentru un ansamblu de domenii D1, D2, ..., Dn ***produsul catezian*** al acestora reprezinta ansamblul ***tuplurilor*** (engl. Tuple, tuplu, кортеж) **<v1, v2, ..., vn>**, unde:

* v1 este o valoare apartinând domeniului D1,
* v2 este o valoare din D2 s.a.m.d.

Or, formal acest lucru poate fi scris:

**D1 × D2 × … × Dn = { (v1 , v2 , …, vn ) | vi ∈ Di , i = 1, …, n}**

Astfel, pentru Exemplul 1, am putea precăuta următorul Produs Cartezian:

**PC = Numar × Sir40 × Numar × Numar × Sir40 × Sir40**

 In Exemplul 2, tuplurile: <"Maria", "F", 50>, <"Vasile", "M",15>, <"Vasile","M",20>, <"Vasile", "F",100> apartin produsului cartezian: D3 x D1 x D2,

 **Notă:**Trebuie de mentionat ca in sirul de domenii care participa la un produs cartezian unele se pot regasi in mod repetat.

**4.2. Relatie**

            Sa presupunem ca ***se acorda o anumita semnificatie*** valorilor domeniilor D1, D2, D3, definite anterior.

            Sa consideram, de exemplu ca D1 cuprinde valorile referitoare la sexul unei persoane, D2 contine valori care exprima vârsta unei persoane si D3 cuprinde numele unor persoane.

            ***Numai unele*** dintre tuplurile produsului cartezian: **D3 x D1 x D2** pot avea o semnificatie si anume cele care contin numele, sexul si vârsta aceleiasi persoane. Dintre cele 202 tupluri care au valoarea "Vasile" pe prima pozitie, numai unul poate avea o semnificatie, ***daca  presupunem ca exista o singura persoana cu acest nume***.

            Se desprinde de aici necesitatea definirii unei submultimi de tupluri, din cadrul produsului cartezian al domeniilor, submultime care sa cuprinda ***numai tuplurile cu semnificatie***.

            ***Definitie:*** **Relatia**reprezinta un subansamblu al produsului cartezian al mai multor domenii, subansamblu caracterizat printr-un nume si care contine tupluri cu semnificatie.

 Considerând  de exemplu ca nu se cunosc decât doua persoane definim realtia R prin tuplurile care descriu aceste persoane si anume:

**R = {<"Maria", "F", 30>, <"Vasile", "M", 35>}**

           **ATENTIE!!!** Intr-o relatie, tuplurile trebuie sa fie distincte (nu se admit duplicari ale tuplurilor) .

            ***O reprezentare comoda a relatiei este tabelul bidimensional (tabela de date în care liniile reprezinta tuplurile, iar coloanele corespund domeniilor* (fig.3.1.).**



                **Fig. 3.1.** **Relatie reprezentata sub forma unei tabele de date**

            **Reprezentarea tabelara a unei *Relații* este preferata adesea altor forme de reprezentare a  relatiilor, întrucat este usor de înteles si de utilizat.**

            În prezentarea conceptului de ***Retatie*** se recurge uneori la analogii cu alte concepte, extrem de bine cunoscute în domeniul prelucrarii automate a datelor precum cel de ***fisier***. Relatia poate avea semnificatia unui fisier, ***tuplul*** poate fi considerat drept o înregistrare, iar ***valorile din cadrul tuplului*** pot fi interpretate drept valori ale câmpurilor de înregistrare.

 În cadrul modelului relational nu intereseaza decat ***relatiile finite***, chiar daca în construirea relatiilor se admit domenii infinite.

 ***Definitie:*** Numarul tuplurilor dintr-o relatie reprezinta **cardinalul**relatiei, în timp ce numarul valorilor dintr-un tuplu defineste **gradul** relatiei.

 Un alt exemplu de relatie derivă din produsul cartezian PC de mai sus. Ea poate fi:

**Produse** = {

<101, ‘Imprimanta laser’, 30, 20, ‘Xerox’, ‘Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti’> ,

<105, ‘Calculator PC’, 20, 23, ‘IBM’, ‘Bd. D.Cantemir nr.1, Bucuresti’>,

<124, ‘Copiator’, 10, 20, ‘Xerox’, ‘Str. Daniel Danielopolu 4-6, Sector 1, Bucureşti’> }

 După cum observăm, кelatia de mai sus contine doar 3 dintre elementele produsului cartezian din care provine **(3 tupluri**).

 O reprezentare intuitiva pentru relatia de mai sus este si urmatoarea, in care fiecare element (tuplu) al relatiei devine o linie a unei tabele si fiecare coloana corespunde unui domeniu din produsul cartezian de baza:



**Notă:** *In fapt deci o relatie se reprezinta o tabela care contine date, fiecare coloana avand asociat un anumit tip de date, dat de domeniul din care provine.*

**4.3.** **Atribut**

            In timp ce tuplurile dintr-o relatie trebuie sa fie unice un domeniu poate apare de mai multe ori în produsul cartezian pe baza caruia este definita relatia.

            Sa consideram, de exemplu ca pentru o persoana dispunem de urmatoarele date: nume,sex, vârsta si numele sotului/sotiei.

            O posibilitate de organizare a acestor date o reprezinta relatta din fig.3.2.

R:                     D3D1D2 D3

|  |  |  |  |
| --- | --- | --- | --- |
| “Maria” | “F” | 30 | “Ion” |
| “Vasile” | “M” | 35 | “Maria” |

**Fig.3.2.** **Relatia PERS**

   Relatia PERS reprezinta un subansamblu al produsului cartezian:

**D3 x D1 x D2 x D3.**

            Semnificatia valorilor din cadrul unui tuplu se stabileste în acest caz nu numai pe baza domeniului de care apartin valorile, ci si in functie de pozitia ocupata în cadrul tuplului. Dependenta fata de ordine a datelor inseamna o reducere a flexibiltatii  organizarii datelor.

 Într-o organizare eficienta, flexibila, ordinea liniilor si a coloanelor din cadrul tabelei de date nu trebuie sa prezinte nici o importanta. Pentru a diferentia coloanele care contin valori ale aceluiasi domeniu si a elimina astfel dependenta de pozitie, *în cadrul tabelei se asociaza fiecarei coloane un nume distinct, ceea ce duce la aparitta notiunii de* ***atribut***.

 ***Definitie:*** **Atributul** (eng. attribute) **reprezinta coloana unei relații** (tabelă de date), caracterizata printr-un nume. Numele coloanei (atributului) exprima de obicei semnificatia valorilor din cadrul coloanei respective.

Deoarece o relatie are o reprezentare tabelara putem vorbi de ‘coloana a unei relatii’. In mod obisnuit, într-o tabela coloanele au un nume.

 Pentru relatia de mai sus ***Produse*** putem fixa de exemplu urmatoarele nume de atribute:

1. ***IdP*** – Codul produsului (nu exista doua produse avand acelasi cod)
2. ***NumeP*** – numele produsului
3. ***Qty*** – Cantitate
4. ***IdF*** – Codul furnizorului (nu exista doi furnizori avand acelasi cod)
5. ***NumeF*** – Numele furnizorului
6. ***AdresaF*** – Adresa furnizorului



**4.4. Schema unei relatii**

            Continutul unei relatii (vazuta ca o tabela) poate varia in timp: se pot adauga sau sterge linii sau se pot modifica unele dintre valorile din liniile existente. Ceea ce ramane constanta este structura relatiei: numele relatiei, numarul si tipul atributelor sale.

 **In terminologia relationala structura unei relatiei este denumita si schema relatiei.**

 ***Definitie:*** Prin **schema** (eng. Relation scheme) unei relatii se întelege numele relatiei, urmat de lista atributelor, pentru fiecare atribut precizându-se domeniul asociat.

 Exista mai multe modalitati prin care se poate specifica schema unei relatii.

 Astfel, pentru o relatie R, cu atributele A1, A2, ..., An si domeniile: D1, D2,..., Dm, cu m <= n, schema relatiei R poate fi reprezentata într-una din formele prezentate in fig. 3.4.

**R = (**A1:D1, …, An:Dm)

a)

|  |  |  |
| --- | --- | --- |
| A1:D1 | … | An:Dm |
|  |  |  |

b)

In exemplele urmatoare prezentam cateva dintre acestea cu referire la relatiile din exemplele de mai sus.

**Produse{**IdP, NumeP, Qty, IdF, NumeF, AdresaF}

**Produse{**IdP: Numar, NumeP: Sir40, Qty: Numar, IdF: Numar,NumeF: Sir40, AdresaF: Sir40}

**Produse** = (IdP, NumeP, Qty, IdF, NumeF, AdresaF)

**5.Operatorii modelului relational**

**Modelul relational ofera doua colectii de operatori pe relatii si anume:**

1. Algebra Relationala (**AR**);
2. calculul relational.

**La rândul sau, calculul relational este de doua tipuri:**

1. calcul relational orientat pe tuplu;
2. calcul relational orientat pe domeniu.

 Ne limitam, în cele ce urmeaza, la elemente **de algebra relationala.**

**6. Algebra Relationala si extensiile sale**

            E. F. Codd a introdus ***Algebra Relationala (AR)*** cu operatii pe relatii, fiecare operatie având drept operanzi una sau mai  multe  relatii  si producând ca rezultat o alta relatie. (***Edgar Frank*Codd*(23 august 1923, Insula Portland, Anglia – 18 aprilie 2003, Williams Island, Florida, SUA) a fost un informatician american de origine engleză***)

            Unele operatii ale AR pot fi definite prin intermediul altor operatii. În acest  sens, putem vorbi de:

* 1. operatii de baza, precum: ***reuniunea, diferenta, produsul cartezian etc.***
	2. operatii derivate, ca: ***intersectia, diviziunea etc.***

            **Codd a introdus asa numita AR standard, constituita din 6 operatii de baza:**

***reuniunea, diferenta, produsul cartezian, proiectia, selectia si jonctiunea precum si din doua operatii derivate: intersectia si diviziunea.***

            Ulterior, la operatiile **AR standard**au fost adaugate si alte operatii, asa numitele operatii "aditionale" sau**extensii ale AR  standard**, precum:

***complementara unei relatii, splitarea (spargerea) unei relatii, închiderea tranzitiva etc.***

            Î**n continuare vor fi prezentate principalele operatii ale AR, precum si modul lor de utilizare.**

***1. Reuniunea****.* *Reprezinta o operatie a AR* ***definita pe doua relatii: R1 si R2******ambele cu o aceeasi schema****, operatie care consta din construirea unei noi relatii R3, cu schema identica cu R1 si R2 si având drept extensie tuplurile din R1 si R2 luate impreuna o singura data.*

            Notatia uzuala pentru reuniune este: **R3=R1 U R2**  Reprezentarea grafica a reuniunii este prezentata in fig. 3.3.



**Fig.3.3.** LOCALIT este Reuniunea relatiilor ORASE si MUNICIPII.

**2. *Diferenta****. Reprezinta  operatie din AR* ***definita pe doua relatii: R1 si R2****, ambele cu o aceeasi schemâ, operatia constând din construirea unei noi relatii R3, cu schema identica cu a operanzilor si cu extensia formata din acele tupluri ale relatiei R1 care nu se regasesc si în relatia R2*.

            Notatia uzuala pentru operatia de diferenta a doua relatii este: **R3=R1-R2**

            În fig. 3.4. este prezentat LOCALIT, un exemplu de diferenta a doua relatii ORASE si LOCALITĂȚI.



**Fig.3.4. LOCALIT, Diferenta relatiilor ORAsE si MUNICIPII**

 ***3. Produs cartezian.*** *Reprezinta o operatie a AR* ***definita pe doua relatii: R1 si R2****, operatie care consta din construirea unei noi relatii R3, a carei schema se obtine prin concatenarea schemelor relatiilor R1 si R2 si a carei extensie cuprinde toate combinatiile tuplurilor din R1 cu cele din R2.*

            Notatia uzuala pentru desemnarea operatiei este: **R3=R1xR2**. Fig. 3.5. prezinta un exemplu de produs cartezian a doua relatii.



**Fig.3.5. Produsul cartezian al relatiilor ORAsE si TARIFE**

***4. Proiectia***. *Reprezinta o operatie din AR definita* ***asupra unei relatii R****, operatie care consta din construirea unei noi relatii P, în care se regasesc unele atribute din R, înseamna efectuarea unor taieturi verticale asupra lui R, care pot avea ca efect aparitia unor tupluri duplicate ce se cer a fi eliminate.*

Prin operatia de proiectie se trece de la o relatie de **grad n** la o relatie de **grad p,** mai mic decât cel initial (p < n) ceea ce explica si numele de proiectie atribuit operatiei.

            Notatia uzuala pentru operatia de proiectie este:

**ΠAi,Aj,…,Am(R)**

            În fig.3.6 este exemplificata operatia de proiectie a unei relatii.



**Fig.3.6. Proiectia relatiei ORAsE pe atributul "Judet"**

***5. Selectia***. *Reprezinta o operatie din AR definita* ***asupra unei relatii R****, operatie care consta din construierea unei relatii S, a carei schema este identica cu cea a relatiei R si a carei extensie este constituita din acele tupluri din R care satisfac o conditie mentionata explicit în cadrul operatiei.* Întrucât cel mai adesea, nu toate tuplurile din R satisfac, aceasta conditie, selectia înseamna ***efectuarea unor taieturi orizontale asupra relatiei*** R, adica eliminarea de tupluri.

Conditia precizata în cadrul operatiei de selectie este în general de forma:



 unde: "operator de comparatie" poate fi: **<, <=, >=, > sau <>.**

            Notatia folosita in mod uzual pentru desemnarea operatiei de selectie este urmatoarea:

**Σ(conditie)R**

În fig.3.7. este exemplificata operatia de selectie.



**Fig.3.7. Selectie efectuata asupra relatiei ORAsE**

***6. Jonctiunea (Joinul).*** *Reprezinta o operatie din AR* ***definita pe doua relatii****: R1  si  R2,  operatie  care  consta  din construirea  unei  noi  relatii****R3****,  prin concatenarea unor tupluri din R1 cu tupluri din R2. Se concateneaza acele tupluri din R1 si R2 care satisfac o anumita conditie, specificata explicit  în cadrul operatiei. Extensia relatiei R3 va contine deci combinatiile acelor tupluri care satisfac conditia de concatenare.*

 Notatiiile uzuale pentru desemnarea  operatiei de jonctiune sunt:



  Reprezentarea grafica a aeestei operatii este prezentata în fig. 3.8.



**Fig.3.8. Reprezentarea grafica a operatiei de jonctiune**

            In general, conditia de concatenare mentionata in cadrul operatiei de jonctiune este de forma:



            In functie de operatorul do comparatie din cadrul conditiei de concatenare, joinul poate fi de mai multe tipuri. Cel mai important tip de join, în sensul celei mai frecvente utilizari este **equijoinul**.

***Equijoinul*** reprezinta jonctiunea dirijata de o conditie de forma:



            Operatia de jonctiune se poate exprima cu ajutorul operatiilor de produs cartezian si selectie, rezultatul unui join fiind acelasi cu rezultatul unei **selectii** operate asupra unui **produs cartezian**, adica:

**JOIN (R1,R2, conditie) = Σ(conditie) (PRODUS(R1,R2), conditie)**

            Produsul cartezian reprezinta o operatie laborioasa si foarte costisitoare, ceea ce face ca in locul produsului sa fie utilizat joinul ori de câte ori acest lucru este posibil. Cu toate ca apare drept o operatie derivata, joinul este prezentat de obicei drept o operatie de baza din AR, data fiind importanta sa in cadrul sistemelor relationale, în special la construirea relatiilor virtuale.

In cadrul fig.3.9. este ilustrata  operatia de **equijoin**.



**Fig.3.9. Operatia de equijoin a relatiilor ORAsE si ESTIMARI**

            Au fost definite numeroase variante ale operatiei de jonctiune, variante care difera nu numai în functie de tipul conditiilor de concatenare, ci si dupa modul de definire a schemei si respectiv, extensiei relatiei rezultate prin jonctiune.

 ***Jonctiunea naturală****.* In cazul equijoinului, schema relatiei contine toate atributele celor doi operanzi (fig.3.10.) În toate tuplurile relatiei rezultat vor exista de aceea cel putin doua valori egale. In sensul eliminarii acestei redundante s-a introdus jonctiunea naturala, ca operatie definita pe doua relatii: R1 si R2, prin care este construita o noua relatie R3, a carei schema este obtinuta prin reuniunea atributelor din relatiile R1 si R2 (atributele cu acelasi nume se iau o singura data) si a carei extensie contine tuplurile obtinute prin concatenarea tuplurilor din R1 cu tuplurile din R2 care prezinta aceleasi valori pentru atributele cu acelasi nume.

 Notatia uzuala pentru jonctiunea naturala este:



Reprezentarea grafica a acestei operatii este prezentata în fig. 3.10.



**Fig.3.10. Reprezentarea grafica a operatiei de jonctiune naturala**

În fig.3.11. este exemplificata operatia de jonctiune naturala.



**Fig.3.11. Operatia de jonctiune naturala a relatiilor ORAsE si ESTIMĂRI**

***7. Intersectia.*** *Reprezinta o operatie a AR* ***definita pe doua relatii****: R1 si R2 ambele cu aceeasi schema, operatie care consta din construirea unei noi relatii R3, cu schema identica cu a operanzilor si cu extensia formata din tuplurile comune lui R1 si R2*.

            Notatiile uzuale folosite pentru desemnarea operatiei de intersectie sunt:



Reprezentarea grafica a operatiei de intersectie este prezantata în fig. 3.12., iar

un exemplu de intersectie a doua relatii figureaza în fig. 3.13.



                          **Fig.3.12.** Reprezentarea grafica a operatiei de intersectie



**Fig.3.13. Intersectia relatiilor ORAsE si MUNICIPII**

            Intersectia reprezinta o operatie derivata, care poate fi exprimata prin intermediul unor operatii de baza. De exemplu, operatia de intersectie se poate exprima prin intermediul operatiei de diferenta, cu ajutorul urmatoarelor echivalente:

**R1 R2=R1-(R1-R2)**

**R1 R2=R2-(R2-R1)**

**Sarcini**

**Se dau urmatoarele relatii cu schemele lor:**

1. **Scari** (Nr\_bloc, Scara, Lift)
2. **Apartamente**(Nr\_bloc,Scara,Apartament,Suprafata,Cutii\_postale, Nr\_prize\_tv)
3. **Familii** (Nr\_mat, Nr\_pers, Nr\_pers\_prez, Nr\_chei)

|  |  |
| --- | --- |
| 1. **Locatari (**Nr\_Mat, Nr\_bloc, Scara, Etaj, Apartament,Nume)
 |  |

***Sa se exprime în algebra relationala cererile:***

1. tabel nominal cu locatarii de pe scara = 3 din bloc = 34) = R1
2. tabel nominal cu locatarii de pe scara = 1 din bloc = 34) = R2
3. tabel nominal cu locatarii de pe scara = 2 din bloc = 34) = R3
4. tabel nominal cu locatarii de pe scarile 1,2,3 ale blocului 34
5. lista apartamentelor cu suprafata mai mare decât 50 mp
6. tabel nominal cu persoanele carelocuiesc pe scara 3 bloc 34 si nu locuiesc si pe scara 1 a aceluiasi bloc

**Raspunsuri la exercitii**

1. R1=nume(bloc=34 and scara=3(scari)  (locatari))
2. R2=nume (bloc=34 and scara=1(scari)  (locatari))
3. R3=nume (bloc=34 and scara=2(scari)  (locatari))
4. R = R1  R2  R3
5. R = suprafata > 50(apartamente)
6. R= (R1-R2)

**SARCINI**

Pentru toate exercitiile care urmeaza se va folosi baza de date (schema) INTREPRINDERE. Acesta este numele generic al bazei de date; în realitate fiecare student va folosi propria baza de date care, conform conventiilor stabilite, are acelasi nume cu al contului propriu pentru fiecare din sistemele de gestiune studiate.

În fiecare tabel se vor insera mai întâi un numar oarecare de linii, având în vedere referintele între tabele. Pentru a obtine ca rezultat al interogarilor multimi de linii nevide, este necesar sa fie introduse cel putin liniile care contin valorile atributelor specificate în interogari.

**Pentru relațiile date, calculați si prezentati schematic, in modul definit mai sus, precum și prezentati operațiile pentru a efectua interogarea respectivă /vezi exemplul de mai sus/**

1. ***Reuniunea***

**R3=R1 U R2**

1. ***Diferenta****.*

**R3=R1-R2**

1. ***Produs cartezian***

**R3=R1xR2**

1. ***Proiectia***.

**ΠAi,Aj,…,Am(R)**

1. ***Selectia***.

**Σ(conditie)R**

1. ***Jonctiunea (Joinul).***
2. **Equijoinul**.
3. ***Jonctiunea naturală***
4. ***Intersectia.***

**1**

Selectati si afisati atributele Nume, Prenume ale tuturor tuplurilor din tabelele FURNIZORI, CLIENTI care au valoarea ‘Bucuresti’ a atributului Adresa, folosind operatia de reuniune (UNION, reuniunea)..

**2**

Selectati si afisati produsul cartezian al tabelelor SECTII si ANGAJATI.

**3**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele furnizorilor care au livrat componente în cantitati egale sau mai mari ca 200 ?" Care va fi rezultatul interogarii?

**4**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele angajatilor care s-au ocupat de achizitionarea componentelor cu denumirea ‘Condensator’ ?" Care va fi rezultatul interogarii? Se precizeaza ca atributul cheie straina IdAchizitor din relatia ACHIITITII refera atributul IdAngajat din relatia ANGAJATI.

**5**

Scrieti si executati instructiunea pentru interogarea: "Care sunt numele, prenumele si adresa furnizorilor care au livrat componente cu denumirea 'Rezistenta'?". Care va fi rezultatul interogarii?

**6**

Scrieti si executati instructiunea pentru interogarea: "Care sunt numele, prenumele si adresa furnizorilor care au livrat componenta cu denumirea 'Condensator' în cantitati mai mari sau egale 150?". Care va fi rezultatul interogarii?

**7**

Scrieti si executati comanda pentru interogarea: "Care sunt numele, prenumele si adresa clientilor care au cumparat produsul cu denumirea 'Monitor' în cantitati mai mari sau egale 10?". Care va fi rezultatul interogarii?

**8**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele angajatilor care s-au ocupat de vânzarea produselor cu denumirea ‘Monitor’ ?" Care va fi rezultatul interogarii? Se precizeaza ca atributul cheie straina IdVanzator din relatia VANZARII refera atributul IdAngajat din relatia ANGAJATI.

**9**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele, prenumele si data nasterii angajatilor care participa la proiectul cu denumirea ‘Sistem de achizitie de date’ ?" Care va fi rezultatul interogarii?

**10**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele sectiilor în care lucreaza angajatii care participa la proiectul cu denumirea ‘Sistem de achizitie de date’ ?" Care va fi rezultatul interogarii?

**11**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele, prenumele si data nasterii tuturor inginerilor cu specialitatea ‘Electronica’?" Care va fi rezultatul interogarii?

**12**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele angajatilor din sectia ‘Productie’ care au fii cu prenumele ‘Ion’ ?" Care va fi rezultatul interogarii?

**13**

Scrieti si executati instructiunea SQL pentru interogarea "Care sunt numele si prenumele inginerilor din sectia ‘Productie’?" Care va fi rezultatul interogarii?

**14**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele angajatilor cu care a colaborat furnizorul Popescu Razvan ? Care va fi rezultatul interogarii?

**15**

Scrieti si executati instructiunea pentru interogarea "Care sunt numele si prenumele angajatilor cu care a colaborat clientul Marinescu Ion ? Care va fi rezultatul interogarii?

**16**

Scrieti si executati instructiunea pentru interogarea "Care sunt produsele în care se folosesc componente furnizate du furnizorul Danescu Ovidiu ?" Care va fi rezultatul interogarii?

**17**

Scrieti si executati instructiunea pentru interogarea "Care sunt produsele în care se folosesc componente furnizate du furnizorul Danescu Ovidiu ?" Care va fi rezultatul interogarii?

**18**

Să se creeze o tabelă cu structura **Nume, Data\_angaj, Data\_nast** şi o tabelă cu

structura **Nume, Localitate, Adresa**. Să se introducă in fiecare tabelă cel puţin 8

articole cu date. Nu există nume care să se repete în cadrul niciunei tabele.

Toate numele trecute în prima tabela se vor regăsi în cea de a doua. Cerinţe:

**a.** Să se afişeze pe ecran toţi angajaţii cu o vechime mai mare de 10 ani;

**19**

Să se creeze o tabelă cu structura **Nr\_bilete, Oras, Tara, Pret\_bilet, Cumparator** şi să se introducă cel puţin 8 articole. Nu sunt oraşe care să apară de mai multe ori. Cerinţe:

**a.** Să se biletele înregistrate în tabelă;

**20**

Să se creeze o tabelă cu structura **Nume\_film, Actor\_princ, Tip** şi o tabelă cu structura

**Nume\_actor, Tara.** Să se introducă în tabele minimum 8 articole. Un film apare o

singură dată în prima tabela, iar un nume de actor apare câte o dată în fiecare dintre

cele doua tabele. Numele actorilor din prima tabelă se vor regăsi în cea de a doua.

Cerinţe:

**a.** Să se şteargă din tabelă toate filmele de tip HORROR;

**21**

Să se creeze o tabelă cu urmatoarea structură **Furnizor, Material, Pret, Cantitate.** Să

se introducă în tabelă minimum 8 articole. Un material apare o singură dată în cadrul

unui furnizor. Cerinţe:

**a.** Să se afişeze toate ofertele pentru materialul X - citit de la tastatură, crescător,

după preţul oferit;

**22**

Să se creeze o tabelă cu structura: **Elev, Clasa, Med\_generala** şi să se introducă 8

articole (un articol- un elev; câmpul **Clasa** conţine un cod de la 1 la 9. Sunt mai mulţi

elevi într-o clasa şi toţi au nume diferite). Cerinţe:

**a.** Să se afişeze numărul claselor din tabelă;

**23**

Să se creeze o tabelă cu structura: **Elev, Clasa, Med\_generala** şi o tabelă cu structura

**Clasa, Diriginte**. Să se introducă în tabele minimum 8 articole (un articol-un elev;

câmpul **Clasa** conţine un cod de la 1 la 9. Sunt mai mulţi elevi într-o clasa şi toţi au

nume diferite). Toate codurile de clasa din prima tabelă se vor regasi în a doua tabelă.

Cerinţe:

**a.** Să se afişeze codul clasei din tabelă cu cei mai mulţi elevi;

**24**

Să se creeze o tabelă cu structura: **Produse, Depozite, Cantitate** şi să se introducă

minimum 8 articole (un articol-un produs, câmpul Depozit conţine un cod de la 1 la 9.

Pot fi mai multe produse într-un depozit şi toate au nume diferite). Cerinţe:

**b.** Să se afişeze numărul depozitelorlor care au rămas cu cel mult doua produse în

urma ştergerii articolelor de la cerinţa a;

**25**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese, Salariu** şi o tabelă cu

structura **Atelier, Profil**. Să se introducă în tabele minimum 8 articole (un articol-un

muncitor, codul unui atelier este un număr de la 1 la 9. Câmpul **Profil** este o datâ de tip caracter. Pot fi mai mulţi muncitori într-un atelier şi toţi au nume diferite). Toate codurile de atelier din prima tabela se vor regasi în a doua tabelă. Cerinţe:

**c.** Să se afişeze lista muncitorilor care lucrează în ateliere cu profilul X, profil citit de

la tastatură.

**26**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese, Varsta** şi să se

introducă 8 articole (un articol-un muncitor, câmpul Atelier conţine un cod de la 1 la 9.

Pot fi mai mulţi muncitori într-un atelier şi toţi au nume diferite). Cerinţe:

**b.** Să se afişeze numărul mediu de piese produse de toţi cei selectaţi la prima

cerinţă;

**27**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese, Nr\_copii** şi să se

introducă 8 articole (un articol-un muncitor; câmpul Atelier este cod de la 1 la 9. Pot fi

mai mulţi muncitori într-un atelier şi toţi au nume diferite). Realizaţi un **meniu** care să

permită urmatoarele aplicaţii:

**c.** Să se afişeze lista tuturor muncitorilor care au copii minori şi numărul acestora,

**28**

Să se creeze doua tabele cu structurile:

Tabela 1: **Muncitor, Atelier, Nr\_piese**

Tabela 2: **Atelier Pret\_piesa**

Să se introducă 8 articole în fiecare tabelă. Câmpul **Atelier** conţine un cod de la 1 la 9.

Pot fi mai mulţi muncitori într-un atelier şi toţi au nume diferite. Valorile din câmpul

**Atelier** din prima tabelă se vor regăsi în a doua tabelă. Cerinţe:

**c.** Sa se afiăeze o situaţie centralizatoare sub forma unui raport care sa conţină:

**29**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese** şi să se introducă 8

articole (un articol-un muncitor; câmpul **Atelier** este un cod de la 1 la 9. Pot fi mai mulţi muncitori într-un atelier şi toţi au nume diferite). Realizaţi un **meniu** care să permit urmatoarele aplicaţii:

**c.** Să se afişeze lista tuturor muncitorilor din atelierul **x,**.

**30**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese** şi să se introducă 8

articole (un articol-un muncitor, câmpul atelier conţine un cod de la 1 la 9. Pot fi mai

mulţi muncitori într-un atelier şi toţi au nume diferite). Cerinţe:

**c.** Să se afişeze o situaţie centralizatoare sub forma unui raport care să conţină:

**31**

Să se creeze o tabelă cu structura: **Muncitor, Atelier, Nr\_piese** şi să se introducă 8

articole (un articol-un muncitor, câmpul **Atelier** este un cod de la 1 la 9. Pot fi mai mulţi muncitori într-un atelier şi toţi au nume diferite). Realizaţi un **meniu** care să permit urmatoarele aplicaţii: Muncitorii cu număr maxim de piese produse se vor transfera la alt loc de muncă (se elimină din tabelă). Să se afişeze în ordine alfabetică numele celor transferaţi;

**b.** Să se afişeze numărul de piese produse de către cei rămaşi în vechiile

ateliere;

**32**

Să se creeze o tabelă cu structura: **Produs, Pret\_unit, Cantitate, Necesar** şi să se

introducă 8 articole (un articol-un produs dintr-un depozit; **Cantitate** = numarul de

produse de acel fel din depozit; **Necesar** = cantitatea necesară în depozit, din acel

produs; valoarea unui produs = cantitatea \* pret\_unitar, iar fiecare produs apare exact

cate o dată in tabelă). Cerinţe:

**b.** Să se afişeze produsele ce au cantitate mai mare cu cel putin 20% decât

stoc\_necesar, în ordinea codurilor;

**33**

Să se creeze o tabelă cu structura: **Produs, Pret\_unit, Cantitate, Necesar** şi să se

introducă 8 articole (un articol-un produs dintr-un depozit; **Cantitate** = numarul de

produse de acel fel din depozit; **Necesar** = cantitatea necesară în depozit; valoarea unui

produs = cantitatea \* pret\_unitar, iar fiecare produs apare exact câte o dată în tabelă).

Realizaţi un **meniu** care să permită urmatoarele aplicaţii:

a.Să se afişeze produsele si preturile

**34**

Să se creeze o tabelă cu structura: **Autor, Titlu, Nr\_imprum, Data\_imp** (pentru numele autorului, titlul cărţii, numărul de exemplare împrumutate şi data la care au fost imprumutate) şi să se introducă cel putin 8 articole (un titlu al unui autor apare într-un singur articol). Un titlu de carte va apărea exact o dată în tabelă. Cerinţe:

**c.** Să se afişeze conţinutul tabelei în ordine alfabetică după numele autorului, iar

dacă mai mulţi autori au acelaşi nume, atunci aceştia vor fi ordonaţi după titlu, tot

alfabetic.

**35**

Să se creeze o tabelă cu structura: **Angajat, Salariu, Nr\_copii, Nr\_zile** (pentru numele angajatului, salariul său, numărul de copii minori în întreţinere şi numărul de zile de concediu medical ale angajatului) şi să se introducă cel putin 8 articole. Un articol reprezintă un angajat din întreprindere. Cerinţe:

**a.** Să se afişeze numele angajatului cu cele mai multe zile de concediu medical;

**36**

Să se creeze o tabelă cu structura: **Medic, Pacient, Diagnostic, Data\_cons** (pentru

numele medicului, numele pacientului, diagnosticul acestuia şi data consultaţiei)**,** şi să

se introducă cel putin 8 articole. Un pacient apare exact o dată în tabelă. Cerinţe:

**a.** Să se afişeze numărul pacienţilor consultaţi în luna curentă de către un medic al

cărui nume se introduce de la tastatură;

**37**

Să se creeze o tabelă cu structura: **Elev, Clasa, Promovat** (pentru numele elevului,

clasa din care face parte şi o valoare logică referitoare la calitatea sa de promovat sau

nepromovat) şi să se introducă cel putin 8 articole. Un articol reprezintă un elev dintr-o

clasă a unei şcoli. Cerinţe:

**a.** Să se afişeze clasa cu cel mai mare procent de promovabilitate;

**38**

Să se creeze o tabelă cu următoarea stuctură:**Cod, Denumire, Calorii** (codul,

denumirea unui aliment şi numărul de calorii pentru 100g din acel aliment) în care

introduceţi minimum 8 articole. Cerinţe:

**b.** Să se afişeze o listă având câmpurile: **Denumire, Calorii**, în ordinea

descrescătoare a caloriilor;

**39**

Să se creeze o tabelă cu următoarea stuctură: **Nume, Adresa, Salariu** (pentru numele,

adresa şi salariul unui angajat) în care introduceţi minimum 8 articole. Un nume apare o singură dată în tabelă. Cerinţe:

**c.** Să se afişeze o listă cu 6 rânduri şi cu două coloane*,:*

***Salariu Nr\_angajati***

**40**

Să se creeze o tabelă cu referitoare la Campionatul European de fotbal, tabela în care

introduceţi minimum 8 articole. Sructura tabelei este următoarea: **Nume, Tara, Goluri**

(pentru numele, ţara şi numărul de goluri marcate de către un fotbalist). Cerinţe:

**a.** Să se afişeze numărul ţărilor participante;

**41**

Un depozit aprovizionează cu produse mai multe magazine. Să se creeze o tabelă cu

următoarea structură: **Cod\_prod, Data\_livr, Pret\_un, Cantitate, Cod\_mag** (pentru

codul, data livrării, preţul unitar, cantitatea unui produs şi codul magazinului în care a

fost livrat acel produs) în care introduceţi minimum 8 articole. Fiecare produs livrat unui magazin apare într-un singur articol (un produs nu poate fi livrat aceluiaşi magazin în mai multe articole, dar poate fi livrat mai multor magazine). Cerinţe:

**a.** Pentru o data\_livrare **Y** (citită de la tastatură) să se afişeze toate codurile

magazinelor şi produsele cu care au fost aprovizionate;

**42**

Un magazin se aprovizionează cu produse de la mai multe depozite. Creaţi tabela cu

următoarea structură: **Cod\_prod, Cantitate, Pret\_un, Cod\_dep, Data\_aprov** (pentru

codul, cantitatea şi preţul unitar al unui produs, codul depozitului de unde s-a livrat acel produs şi data livrării sale) în care introduceţi minimum 8 articole. Fiecare produs primit de la un depozit apare într-un singur articol, dar acelaşi produs poate să apară în mai multe articole (de la depozite diferite). Cerinţe:

**b.** Afişaţi valoarea totală a mărfurilor transferate de la un depozit cu codul Z (citit de

la tastatură);

**Anexa 2.1**

**Exemplu**

**Пример 1**. Рассмотрим отношение "Сотрудники" заданное на доменах "Номер\_сотрудника", "Фамилия", "Зарплата", "Номер\_отдела". Т.к. все домены различны, то имена атрибутов отношения удобно назвать так же, как и соответствующие домены. Заголовок отношения имеет вид:

Сотрудники (Номер\_сотрудника, Фамилия, Зарплата, Номер\_отдела)

Пусть в данный момент отношение содержит три кортежа:

(1,Иванов, 1000, 1)

(2, Петров, 2000, 2)

(3, Сидоров, 3000, 1)

такое отношение естественным образом представляется в виде таблицы:

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер\_сотрудника** | **Фамилия** | **Зарплата** | **Номер\_отдела** |
| 1 | Иванов | 1000 | 1 |
| 2 | Петров | 2000 | 2 |
| 3 | Сидоров | 3000 | 1 |

**Таблица 1 Отношение "Сотрудники"**

*Определение 3*. ***Реляционной базой данных*** называется набор отношений.

*Определение 4*. ***Схемой реляционной базы*** данных называется набор заголовков отношений, входящих в базу данных.

Хотя любое отношение можно изобразить в виде таблицы, нужно четко понимать, что *отношения не являются таблицами*. Это близкие, но не совпадающие понятия. Различия между отношениями и таблицами будут рассмотрены ниже.

Термины, которыми оперирует реляционная модель данных, имеют соответствующие "табличные" синонимы:

|  |  |
| --- | --- |
| **Реляционный термин** | **Соответствующий "табличный" термин** |
| База данных | Набор таблиц |
| Схема базы данных | Набор заголовков таблиц |
| Отношение | Таблица |
| Заголовок отношения | Заголовок таблицы |
| Тело отношения | Тело таблицы |
| Атрибут отношения | Наименование столбца таблицы |
| Кортеж отношения | Строка таблицы |
| Степень (-арность) отношения | Количество столбцов таблицы |
| Мощность отношения | Количество строк таблицы |
| Домены и типы данных | Типы данные в ячейках таблицы |

**Свойства отношений**

Свойства отношений непосредственно следуют из приведенного выше определения отношения. В этих свойствах в основном и состоят различия между отношениями и таблицами.

1. *В отношении нет одинаковых кортежей*. Действительно, тело отношения есть *множество*кортежей и, как всякое множество, не может содержать неразличимые элементы (см. понятие множества в гл.1.). Таблицы в отличие от отношений могут содержать одинаковые строки.
2. *Кортежи не упорядочены (сверху вниз)*. Действительно, несмотря на то, что мы изобразили отношение "Сотрудники" в виде таблицы, нельзя сказать, что сотрудник Иванов "предшествует" сотруднику Петрову. Причина та же - тело отношения есть множество, а множество не упорядочено. Это вторая причина, по которой нельзя отождествить отношения и таблицы - строки в таблицах упорядочены. Одно и то же отношение может быть *изображено*разными таблицами, в которых *строки идут в различном порядке*.
3. *Атрибуты не упорядочены (слева направо)*. Т.к. каждый атрибут имеет уникальное имя в пределах отношения, то порядок атрибутов не имеет значения. Это свойство несколько отличает отношение от математического определения отношения (см. гл.1 - компоненты кортежей там *упорядочены*). Это также третья причина, по которой нельзя отождествить отношения и таблицы - столбцы в таблице упорядочены. Одно и то же отношение может быть *изображено*разными таблицами, в которых *столбцы идут в различном порядке*.
4. *Все значения атрибутов атомарны*. Это следует из того, что лежащие в их основе атрибуты имеют атомарные значения. Это четвертое отличие отношений от таблиц - в ячейки таблиц можно поместить что угодно - массивы, структуры, и даже другие таблицы.

Замечание. Из свойств отношения следует, что *не каждая*таблица может задавать отношение. Для того, чтобы некоторая таблица задавала отношение, необходимо, чтобы таблица имела простую структуру (содержала бы только строки и столбцы, причем, в каждой строке было бы одинаковое количество полей), в таблице не должно быть одинаковых строк, любой столбец таблицы должен содержать данные только одного типа, все используемые типы данных должны быть простыми.

Замечание. Каждое отношение можно считать ***классом эквивалентности таблиц***, для которых выполняются следующие условия:

* Таблицы имеют одинаковое количество столбцов.
* Таблицы содержат столбцы с одинаковыми наименованиями.
* Столбцы с одинаковыми наименованиями содержат данные из одних и тех же доменов.
* Таблицы имеют одинаковые строки с учетом того, что порядок столбцов может различаться.

Все такие таблицы есть различные *изображения*одного и того же отношения.

**Первая нормальная форма**

Труднее всего дать определение вещей, которые всем понятны. Если давать не строгое, описательное определение, то всегда остается возможность неправильной его трактовки. Если дать строгое формальное определение, то оно, как правило, или тривиально, или слишком громоздко. Именно такая ситуация с определением отношения в ***Первой Нормальной Форме*** (***1НФ***). Совсем не говорить об этом нельзя, т.к. на основе 1НФ строятся более высокие нормальные формы, которые рассматриваются далее в гл. 6 и 7. Дать определение 1НФ сложно ввиду его тривиальности. Поэтому, дадим просто несколько объяснений.

*Объяснение 1*. Говорят, что отношение  находится в 1НФ, если оно удовлетворяет определению 2.

Это, собственно, тавтология, ведь из определения 2 следует, что других отношений не бывает. Действительно, определение 2 описывает, что является отношением, а что - нет, следовательно, отношений в непервой нормальной форме просто нет.

*Объяснение 2*. Говорят, что отношение  находится в 1НФ, если его атрибуты содержат только скалярные (атомарные) значения.

Опять же, определение 2 опирается на понятие домена, а домены определены на простых типах данных.

Непервую нормальную форму можно получить, если допустить, что атрибуты отношения могут быть определены на сложных типах данных - массивах, структурах, или даже на других отношениях. Легко себе представить таблицу, у которой в некоторых ячейках содержатся массивы, в других ячейках - определенные пользователями сложные структуры, а в третьих ячейках - целые реляционные таблицы, которые в свою очередь могут содержать такие же сложные объекты. Именно такие возможности предоставляются некоторыми современными пост-реляционными и объектными СУБД.

Требование, что отношения должны содержать только данные простых типов, объясняет, почему отношения иногда называют ***плоскими таблицами*** (***plain table***). Действительно, таблицы, задающие отношения двумерны. Одно измерение задается списком столбцов, второе измерение задается списком строк. Пара координат (Номер строки, Номер столбца) однозначно идентифицирует ячейку таблицы и содержащееся в ней значение. Если же допустить, что в ячейке таблицы могут содержаться данные сложных типов (массивы, структуры, другие таблицы), то такая таблица будет уже не плоской. Например, если в ячейке таблицы содержится массив, то для обращения к элементу массива нужно знать *три*параметра (Номер строки, Номер столбца, номер элемента в массиве).

Таким образом появляется третье объяснение Первой Нормальной Формы:

*Объяснение 3*. Отношение  находится в 1НФ, если оно является плоской таблицей.

Мы сознательно ограничиваемся рассмотрением только классической реляционной теории, в которой все отношения имеют только атомарные атрибуты и заведомо находятся в 1НФ.

**Выводы**

Реляционная модель данных состоит из трех частей:

* Структурной части.
* Целостной части.
* Манипуляционной части.

В классической реляционной модели используются только ***простые (атомарные) типы данных***. Простые типы данных не обладают внутренней структурой.

***Домены*** - это типы данных, имеющие некоторый смысл (семантику). Домены ограничивают сравнения - некорректно, хотя и возможно, сравнивать значения из различных доменов.

***Отношение*** состоит из двух частей - ***заголовка отношения*** и ***тела отношения***. Заголовок отношения - это аналог заголовка таблицы. Заголовок отношения состоит из атрибутов. Количество атрибутов называется ***степенью отношения***. Тело отношения - это аналог тела таблицы. Тело отношения состоит из ***кортежей***. Кортеж отношения является аналогом строки таблицы. Количество кортежей отношения называется ***мощностью отношения***.

Отношение обладает следующими свойствами:

* В отношении нет одинаковых кортежей.
* Кортежи не упорядочены (сверху вниз).
* Атрибуты не упорядочены (слева направо).
* Все значения атрибутов атомарны.

***Реляционной базой данных*** называется набор отношений.

***Схемой реляционной базы*** данных называется набор заголовков отношений, входящих в базу данных.

Отношение находится в ***Первой Нормальной Форме*** (***1НФ***), если оно содержит только скалярные (атомарные) значения.