**1**

**CAPITOLUL I (INTRODUCERE ÎN BD)**

1. **ASPECTUL INFORMAŢIONAL AL CURSULUI**

***determinat de***

**Pricini**

* **Depozitare - Calitatea deciziilor (soluţiilor)**
* **Acces - Dirijare**
* **Procesare**

**?**

**Obţinute la timpul potrivit**

**VERIDICITATEA**

**?**

**DATE (FAPTE) INFORMAŢIE**

1. **D I materie primă pentru Informație:**

**D (*fixare, acumulare*)**

1. **PRELUCRARE I (*micşorarea volumului Inf.*)**
2. **I** **în planificarea şi adoptarea DECIZIILOR**

**Resursă / Sursă**

**Resurse Tehnologice**

1. **Sisteme informaţionale : =**  **Aplicaţii de progame**



**Omul**

**+ Acumularea şi prelucrarea Deciziilor**

**+ Obţinerea Informaţiei**

**+ Adoptarea deciziilor**  **I**

1. **I**-> **Dirijare** -> **Mişcarea Info** :=



**D** Calea **I** (pînă la transferul în **I** sau**D**)

**se dirijează**

sens

->**FLUX Iconţinut**

frecvență

**are valoare/sens**

**I**

**Mişcarea**

**orizontal**

**Intern**

**Extern**

**F. I.**

**vertical**

**Orizontal S S**

**Vertical** **I I**

* **Aspectul aplicativ al Informaţiei (sursă)**

**2**

* **SAPD – SISTEM AUTOMATIZAT DE PRELUCRAREA DATELOR.**

***EXEMPLU:*****FLUX**

**Nivel superior**

**IEŞIRE**

**INTRARE**

**de dirijare**

**NIVEL STRATEGIC**

**I** **Decizii**

**IEŞIRE**

**INTRARE**

**Mediu MEDIU**

**Decizii D.I.**

**IEŞIRE**

**INTRARE**

**Inferior** **TACTIC**

**Decizii/activităţi**

1. **NOŢIUNI DE BAZĂ A SAPD (*SISTEM AUTOMATIZAT DE PRELUCRARE A DATELOR*)**

* **DOMENIU DE STUDIU ( := U subD) – orice tip de organizarea datelor.**
* **OBIECT := element DS**
* ***A*TRIBUT := descrie Obiectul (din DS) – elementul prin intermediul cărora este descris un obiect din domeniul de studiu.**

***Element de Date: = descrie (A) (factor logic) CÎMP /descrie Tipul(ED)/***



* **VALORI DE DATE** (**ED**) – **reprezintă datele cu care lucrează programatorul pentru a obţine o anumită informaţie din sistemul creat.**

***Exemplu:*DS:ACTIVITĂŢI BANCARE**

|  |  |  |
| --- | --- | --- |
| **OBIECT** | **ATRIBUT** | **VALORI DE DATE** |
| ***Cont*** | **Tip (cîmp)**  **număr**  **sold** | **D/C**  **3345 / 21**  **320.53** |
| ***Client*** | **Nr. cont**  **NP / Denumire**  **Adresă** | **3244**  **Ionică / „Zorile”**  **Chişinău** |

**Noţiunile descriu 3 domenii care stau la baza procesului cunoaşterii care parcurge calea!**

**LUMEA INFORMAŢIA DATE**

**REALĂ**

**Domeniul Domeniul Domeniul**

**Atribut**

**(valorile**

**atributelor)**

ED

(valori

de date)

**Obiect**

**de**

**Studiu**

**Conceptual ideilor de date**

**3**

**PROIECTAREA UNUI SAPD începe cu determinarea/identificarea**

**DOMENIULUI CONCEPTUAL:=*Suma cerinţelor*** *faţă de* ***Obiectele de Studiu****ale unui* ***DS****, determinate de* ***Atributele*** *şi* ***Datele****ce îl descriu luând în considerare* ***legăturile dintre obiecte.***

Este evident că pentru a efectua prelucrarea datelor şi obţinerea informaţiei avem nevoie ca să asigurăm între elementele sistemului proiectat o anumită interconexiune (**legătură**). Această legătură se efectuează cu ajutorul atributelor cheie.

**!!!**

**ATRIBUTUL – CHEIE: =**reprezintă un element al sistemului informaţional, care permite obţinerea informaţiei din unul sau mai multe ***obiecte legate între ele*** prin intermediul acestui atribut.

* Elementul cheie se determină de element de date al Atributului.**/PRIMARĂ, UNICĂ, CANDIDAT, REGULAR, EXTERNĂ/**

**ÎNSCRIEREA DE DATE: =**este totalitatea datelor legate între ele ce descriu ***valorile de date*** ale unui ***obiect*** din ***domeniul de studiu***.

* **Înscriere (înregistrare):fizice / logice** (*cu înscrierile logice lucrează de regulă programatorul, iar cele fizice sunt utilizate pentru operaţiile intrare / ieşire*).

**FIŞIER DE DATE: =**mulţimea de date interdependente ce descriu un **DS**

**BAZĂ DE DATE:=mulţimea de fişiere de date care caracterizează DS**

**IMPORTANT!!!!!** /mulţime de date care caracterizează DS sînt interdependente./

**BAZĂ DE DATE : =**mulţime de **Fişiere de date** care caracterizează**D.S***.*

**SGBD : =**aplicaţii de programă pentru dirijarea cu **Bazele de Date.**

**SISTEMĂ DE BD :=BD, SGBD, componenta HARD, factorul uman**.

**MODUL DE ORGANIZARE A DATELOR**

1. **Modelul Conceptual al Domeniului de Studiu (DS). Există 3 tipuri de organizare a datelor:**
2. **Externă**
3. **Global-logică**
4. **Fizică**

**Extern Logic Fizic**

**Obiect**

**„Ce se doreşte?”**

**ED**

**„În baza la ce?”**

**Atribut**

**„Ce se poate?”**

**Schema de mai sus are caracteristicile:**

1. **EXTERN** - ***Domeniul de existenţă*** al **O**biectului cu proprietăţile ce-l descriu pentru a modela un **DS**
2. **ATRIBUT** - ***Domeniul ideilor şi informaţiei*** reflectă modelul conceptual al **O**biectului şi implicit al **DS**.
3. **FIZIC** - ***Domeniul datelor***.

**CAPITOLUL II. ETAPE ÎN DEZVOLTARE. MODELE.**

**4**

1. **Etapele de dezvoltare a SAPD şi SGBD**

În procesul de dezvoltare a **SAPD şi SGBD** pot fi indentificate câteva etape:

1. ***Etapa* problemelor simple**
2. ***etapa*** care presupune abordarea **metodei universale** ce reprezintă de fapt reuniunea maii multor probleme simple şi dirijarea proceselor de soluţionare a lor cu o unitate de program de monitorizare.

**1…** **2…**

**Rezultate**

**Pr.1**

**Rezultate**

**Pr.Univ.**.

**Problemă**

**Prog.**

………

**Pr.N**

1. **Probleme:**
2. Controlul surplusului de date (*Lucrul cu fişier din diferite criterii*)
3. Controlul Veridicităţii datelor / informaţiei
4. Delimitarea / separarea datelor (*în diferite fişiere*)
5. Restricţii la acces de date (*timpul minim*)
6. Interacţiunea dintre Date şi programele de Aplicaţii (*deseori schimbarea în date – provoacă schimbarea în Aplicaţii*)

**NOTĂ!** este necesară o abordare care:

* ar asigura principiul includerii de noi probleme (**fără a modifica structura datelor şi a programelor**)
* conectarea lor la cele existente
* ar prezenta un NOU mod de organizare a datelor **lipsit de problemele 1-5!**

**ACEASTĂ ABORDARE ESTE POSIBILĂ DACĂ VOR FI ÎNDEPLINITE 2 PRINCIPII:**

* ***datele urmează să fie integrate*** (datele stocate centralizat şi interdependente).
* ***independenţa*** maximală a aplicaţiilor de program în raport cu datele ce descriu **DS**.

În baza acestor 2 principii putem schimba logica creării unui sistem informaţional**. Această nouă logică se bazează pe Modul de organizare a datelor în SAPD ce se determină de triada:**

* **Modelul Conceptual /**cum vede utilizatorul realizarea DS!!!/
* **Modelul Logic /**cum poate fi realizat aceea
* **Modelul Fizic.** Cerinţe de bază.

1. ***Datele să fie integrate*** (*stocate centralizat şi interdependente*)
2. ***Independenţă maximală a Aplicaţiei de Program în raport cu datele*** ce descriu DS.



**Astfel: BD** ();



**SGBD** (Aplicaţii de program pentru preluarea şi dirijarea BD)

**SBD** ();



**Există Baze de Date:*Factologică; Documentală; Mixtă.***

**5**



**SCHIMBUL LOGICEI CREĂRII SAPD**

*Cum vede utilizatorul Modelul Conceptual*

**Date Integrate**

**Prog. Aplicaţie Pr. Nr.1**

**Pr. 1**

**Prog. Aplicaţie Pr. Nr.2**

**D**

**B**

**Pr. 2**

**SGBD**

**Acces**

. .

. .

**Prog. Aplicaţie Pr. Nr.M**

. .

**Pr. M**

*Separă modelul logic Modelul de*

*de cel fizic şi asigură păstrare fizică*

*independenţa datelor a datelor*

*de programă*

**MODELE DE DATE**

**MODELE DE DATE: =**reprezintă un ***INSTRUMENT*** de lucru care **permite** a lua în considerare **toate *cerinţele*** faţă de **O**biectele unui **DS**, care se precaută pentru a fi creat **SAPD *şi modul de prezentare*** a legăturilor dintre tipurile elementelor de date.**S A U**

1. **TOOLS** care permite a lua în considerare ***toate cerinţele*** faţă de **O**biectele unui **DS** care se precaută.
2. **MODUL** de prezentare a legăturilor dintre tipurile elementelor de date.

**Astfel, ORICE MD SE DETERMINĂ DE:**

1. ***Structura de date*** (*atribut, TIP, elemente de date*)
2. ***Mulţimea de operaţii*** posibile asupra **O**biectelor şi asupra **E**lementelor de **D**ate.

**MODEL CONCEPTUAL** („***CE DORIM?*”): =**reprezintă obiectele **DS** şi legăturile dintre ele fără a indica modul lor fizic de realizare.

**MODEULU LOGIC** („***CE SE POATE****?*”)**: =**varianta modelului conceptual, care poate fi realizată într-un **SAPD** cu un **SGBD** concret.

**MODELUL EXTERN** („***CÎT SE POATE****?*”)**: =**submulţimea din modelul logic care permite utilizatorului să obţină o anumită informaţie despre modelul logic.

**MODEL INTERN** („***ÎN BAZA LA CE?*”): =**componenta care asigură obţinerea informaţiei despre modelul logic prin intermediul SGBD. (*prezintă componenta fizică formată*

*în care datele sânt păstrate pentru a fi preluate*).

**DEF:Modalitatea de prezentare a structurii logice a datelor**

**independent de modul fizic de realizare a lor, *se numeşte independenţa datelor.***

Abordarea nouă a creării ***SAPD*** asigură următoarele ***două tipuri de independenţă***:

**6**

1. Dacă **MODELUL CONCEPTUAL** este supus unor modificări, atunci ele nu trebuie să influențeze **modelul extern** elaborat.
2. **MODELUL EXTERN** – nu este supus modificărilor în modelul intern.

**MODELE EXTERNE**

**ME 1**

**ME 2**

**ME M**

**Problema 1**

**Modelul intern**

***Model***

***logic***

***Model***

***conceptual***

**Problema 2**

.

.

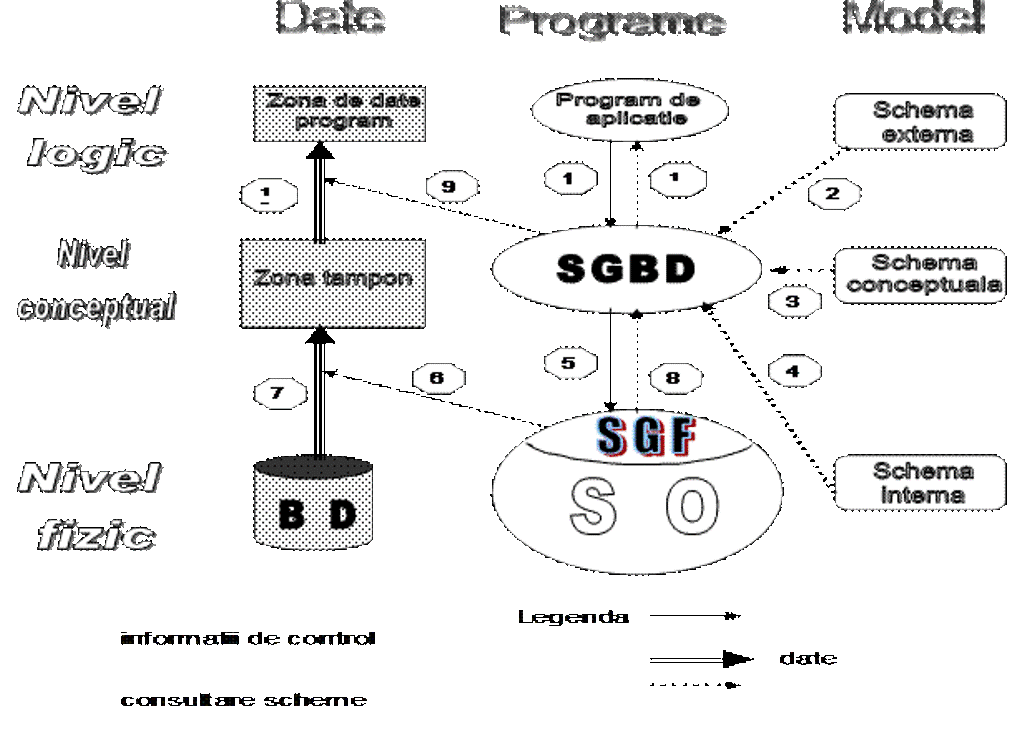
**Problema *N***

.

***Nivelul I de Nivelul II de***

***Independenţa datelor independenţa datelor***

***Cerinţele conceptuale ale utilizatorului unite într-o singură prezentare generalizată!***



**7**

**LEGĂTURI ÎN MODELE DE DATE**

Pentru a modela un **DS** complet în crearea unui **SAPD** utilizînd **BD** vom utiliza **SCHEME** şi **SUBSCHEME**.

**DEF:SCHEMA : =**este modul de prezentare a legăturilor dintre **obiecte,** **atribute** şi **elemente de date**.

De rînd cu noţiunea de schemă utilizăm noţiunea de

**DEF:SUBSCHEMĂ: =*reprezintă o parte a schemei*** cu care lucrează programatorul pentru a pregăti Programul de Aplicaţie.

Aceste noţiuni sînt utilizate în procesul de modelare de rînd cu ***principiile de compoziţie şi decompoziţie***. Există câteva moduri de reprezentare:

Aceste intrări şi ieşiri depind de tipul legăturilor care sunt la rândul lor:

* **1 : 1** 1 : la mai multe **(1 : N)**
* mai multe : mai multe **(N : N)**

***Exemple:***

**OS1** – ***pacientul***

**Atribute:** număr, nume, adresă, nr. paşaport

**OS2 –*chirurgul***

**Atribute:** număr patent, nume

**OS3** – ***patul***

**Atribute:** număr palată, număr pat

**Legăturile Obiectelor:**

**Legăturile Obiectelor:** OS1 OS3; OS1 Nr. de palată (OS3); OS1 OS2



**Legături atribute:** Nume pacient Nr. paşaport;



Nume Număr (pat, secţie, palată);



Nume pacient Nume chirurg



**NOTĂ!**Pentru a prezenta schemele şi legăturile dintre atribute, la etapa de proiectare sînt utilizate ***atributele-cheie***. Ele pot fi ***simple*** sau ***compuse***, dar important este că pe scheme aceste elemente sînt evedenţiate pentru a reflecta legăturile, atît între atribute, cît şi între atributele pe care ele le descriu.

**NOTĂ!***Sunt necesare restricţii, atât asupra DS, cît şi asupra atributelor!*

Pentru ca schemele să fie scrise optimal şi să reflecte o BD integrată vom precăuta următoarele **7*recomandări (reguli):***

* ***Orice schemă are*** o structură simplă (se determină de un element cheie simplu sau compus).

**8**

* ***Elementele schemei sînt*** notate cu nume, de regulă atribut sau obiect.
* ***Nu există într-o schemă*** (subschemă) două elemente înscrise sub unul şi acelaşi nume.
* ***Legăturile se prezintă*** prin una din cele 3 legături (1:1, 1:N, N:N)
* ***Elementele cheie*** trebuie să fie marcate
* ***În scheme sînt prezentate toate legăturile*** cheilor primare şi secundare
* ***Schema este utilizată*** pentru a prezenta proprietăţile interne ale datelor ***ş***i prin urmare ***trebuie să fie una stabilă.***

**ELEMENT-CHEIE. ÎNSCRIERI. MODURI DE REPREZENTARE A LEGĂTURILOR DINTRE ELEMENTELE UNEI SCHEME. INTEGRAME BACHMANN**

* **SUBschemă -> Schemă-> Modelul Conceptual (*Nivel logic [Modelare]*)**
* **Elemente de Date -> Atribut-> înscriere -> BD (*Nivel fizic [Programe]*)**

**Înscrieri Simple (Univoc de Element-Cheie) Elemente cheie Simple**

**Compuse (NU!)**  **Compuse**

**CONCLUZIA!!**

Din punct de vedere a modului de reprezentare a legăturilor dintre elementele unei scheme ***există 3 tipuri de modele***:

1. **IERARHIC**
2. **REŢEA**
3. **RELAŢIONAL**

**Caracteristicile modelelor /specific tuturor/:**

1. **Structura**
2. **Mulţimea de operaţii**
3. **Restricţii la controlul integrităţii**

***Deosebirea:*** modelul de prezentare a legăturilor

**MODELUL *IRERARHIC*-** specific pentru el este **structura datelor** în formă de **graf**. În **nodul-rădăcină** de regulă este plasat un obiect al DS. În nodurile ce urmează pe nivele sînt plasate atributele ce descriu obiectele.

**9**

1. ***Structura ierarhică satisface condiţiile:***
2. Ierarhia mereu începe într-un nod.
3. Orice nod are unul sau mai multe atribute (noduri).
4. Pe cel mai de jos nivel ierarhic ar putea exista noduri dependente.
5. Orice nod de pe un nivel are pe nivelul precedent numai o

singură legătură.

1. Nivelul de bază poate poseda un singur nod sau mai multe noduri.
2. Accesul la orice nod, excepţia nodului rădăcina are loc prin nodul de bază, deci este liniar.

*În modelul ierarhic domeniul de studiu se prezintă ca o reuniune de fişiere, iar orice fişier îşi are propriul lui model ierarhic.*

1. ***Operaţii:*** sortare, memorizare, UPDATE, DELETE, Extragere, INSERT (succesivă – salt nu există!)
2. ***Constrângeri de integritate.***

Se menține doar integritatea relației dintre ***proprietari*** și ***membrii unei relații de grup*** (nu poate exista nici un descendent fără un strămoș). După cum sa menționat deja, nu există o întreținere automată a corespondenței între înregistrările asociate incluse în diferite ierarhii.

Rădăcina

Părinte

Părinte

Copil

Copil

Copil

***Exemplu /îscrieri/:***

**Film**

**Denumire**

**Regizor**

**Regizor**

**Film**

**Denumire**

**Număr de locuri**

**Denumire cinematograf**

**Cinematograf**

**10**

**aterizare**

**Tip comandă**

**Ruta**

**plecare**

**Nr.ruta,data**

**Loc. libere**

***Exemplu:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Numele  Prenumele | data naşterii | | | salariul | adresa |
| ziua | anul | luna |  |  |

*(specialitatea) (Copilul) (animalul domestic)*

Specia

Nume

Vîrsta

Nume

Funcţia

Codul specialităţii

Funcţia

**Prelucrarea în baza cuvintelor-cheie**

***Exemplu:***

Pacient

Nume

Adresă

**I**

OS1 Pacientul

Chirurg (OS2)

Data

Tip oper.

Nr. pat

Nume

Ob. Chirurg Ob. Operaţie

**II**

Nr. pat

Nume

Nr. paşap

Tip oper.

Data

Adresa

Nume

Pacient Operaţia

**III**

Nr. paş

Nume

Data operaţiei

Adresă

Tip operaţie

Nr.patent

Nume

Data operatiei

**Există SGBD:**

* Bune
* Simple ***pozitive***(TEAM-Up, PC-FOCUS, DATA-Eidg)
* Asigurate parţial
* Grele

**11**

* Opreţaii INSERT, DELETE ***negative***
* Delete atent (Free! Norton)

***Avantajul* modelului ierarhic este acela că metodele folosite la regăsirea înregistrărilor asociate din baza de date sunt mai simple decât cele folosite în modelul reţea.**

***Un avantaj* al utilizării bazelor de date ierarhice este acela că utilizatorul poate extrage datele foarte rapid datorită legăturilor explicite definite în structura tabelelor. Un alt avantaj este acela că integritatea referenţială se obţine prin crearea structurii şi nu poate fi încălcată, ceea ce face ca o înregistrare din tabelul copil să fie obligatoriu asociată unei înregistrări existente în tabelul părinte, iar o înregistrare ştearsă din tabelul părinte să impună eliminarea tuturor înregistrărilor asociate din tabelul copil.**

*Probleme deosebite* vor apare în momentul în care utilizatorul doreşte să introducă o înregistrare în tabelul copil care nu are asocieri cu nici o înregistrare din tabelul părinte. Acest tip de bază de date nu poate suporta asocierile complexe şi, de aceea, deseori sunt probleme referitoare la redundanţa datelor, deoarece este posibil să-i fie permisă introducerea de date inconsistente. Această problemă poate fi însă rezolvată prin crearea a două baze de date pentru a înlocui tipurile de relaţii mulţi-la-mulţi, aşa cum se prezintă în figura de mai jos:

Deşi bazele de date ierarhice ofereau un acces direct şi rapid la date, dovedindu-şi superioritatea într-o multitudine de situaţii specifice, devenise evident că era necesară introducerea unui nou model de date pentru a remedia problemele tot mai presante referitoare la redundanţa datelor şi la rezolvarea asocierilor complexe dintre înregistrări.

**MODELUL DE TIP REŢEA** este determinat de faptul că între **date, obiecte** şi **atribute** există alt mod de legătură decît în modelul ierarhic şi anume: ***nodurile de pe un anumit nivel pot fi generate nu doar de un singur nod de pe nivelul precedent.***

1. ***Structura ierarhică satisface condiţiile:***

**Elementul:** colecţie de date care constă din exemplare de date

Posesorul colecţiei, membrul colecţiei

**12**

***Exemplu:***

**Pacient**

INSCRIEREA ***POSESOR***

***INSCRIERE*** Tipul colecției se determină de tipul legăturii

**Operaţia**

INSCRIEREA ***MEMBRU***

**AVANTAJELE** unui astfel de model sînt evidente: posibilitatea de realizare a legăturilor 1:N, N:N precum şi libera executare a operaţiilor de înserare şi distrugere.

Accesul distrugere sau prelucrarea datelor poate fi efectuată începînd cu orice înscriere, asigurîndu-se astfel o universalitate în procesul de căutare a datelor ,

**ÎNSĂ DEZAVANTAJELE SUNT:**

1. ***Toată structura*** datelor este în posesia doar a unui număr limitat de executori.
2. Datorită gradului de libertate în procesul de modelare şi executare a sistemului informaţional, el necesită o ***reorganizare periodică a sistemului***.
3. Sistemele de Gestiune a BD ***complicate***.

**Există SGBD:** Programatorul trebuie să ţină cont

DIAMS toată Structura Datelor

DB-Vista Reorganizarea Datelor periodic

PARADOX SGBD complicate

1. ***Operaţii:*** sortare, memorizare, UPDATE, DELETE, Extragere, INSERT  *Prelucrarea - cu orice înscriere.*

M. ierarhic

M. Reţea

Cerinţele faţă de Domeniul de Studiu

Modul de prezentare a legăturilor dintre date

M.D.

M. Relaţional

Compuse

Simple

Elem. cheie

Subschema

Schema

2-lea

nivel

1-ul

nivel

M.L.

M.I..

M.E.

M.con.

Structura

Restricţii

Operaţii

**13**

**MODELUL *RELAŢIONAL***

1. ***Structura ierarhică satisface condiţiile:***

MR. este indentificat de proprietăţile:

1. El se descrie printr-o relaţie: TABEL.
2. Orice element al relaţiei este un element de date, care descrie un atribut.
3. În relaţii nu există grupe de date, care să se repete pentru unul şi acelaşi set de atribute.
4. Toate coloanele sînt omogene (au acelaşi tip - atribute).
5. Fiecare coloană are un nume unical.
6. În relaţii nu există două înscrieri identice.
7. Coloanele şi rîndurile într-un tabel au aceleaşi proprietăţi în operaţii asupra datelor din ele.
8. Normalizarea presupune pentru o relaţie că toate înscrierile în ea au aceiaşi lungime – acelaşi număr de cîmpuri.

* Evitarea neajunsurilor MI şi MR.
* Idea lui COD. (*1970 - normalizarea*)
* Descrierea Modelului de Date în formă de tabele dreptunghiulare – RELAŢII
* BD – BD relaţionale

**Atribute**

vîrsta

gen

nume

*Exemplu:*

M

F

F

20

15

30

Ion

Vlanetina

Elena

**Obiect:** Student. (Relaţie)

**Atribut:nume, vîrstă, gen**

**înscrieri**

mulţimea de date

**Forma Relaţiei – normală (orice înscriere – aceiaşi lungime).**

**CHEIE** Atributul care identifică înscrierile.



**14**

**Conţinutul Ideii lui COD:**

* **selectarea informaţiei**
* **navigarea în Baza de Date**
* **adresarea nu după adresă, ci după conţinut**
* **fişier – mulţime. Teoria Mulţimilor**
* **operaţiile asupra BD Relaţionale: x, \, Proiecţia, Selectarea şi altele.**



* **BD în PC**: fişier: înscrieri – cîmp

tabel: rînd, coloană

relaţie: cartej, atribut

**ESENŢA:** Exemplar de date, atribut.

**Modelul relaţional a fost elaborat de E. F. Codd în 1970.** Este un model simplu şi intuitiv bazat pe o viziune tabelară asupra datelor. Întregul model relaţional are la bază teoria matematică a relaţiilor şi algebra relaţională.

**Definitie:O relaţie (tabel)este un subansamblu al produsului cartezian de n domenii** (un domeniu este un ansamblu de valori) conform următoarei formule:

**R ⊂ (D1 x D2 x D3 x … x Dn)**

**Exemplu:** Entităţii din lumea reală numită **PRODUSE** îi va corespunde relaţia (tabelul) PRODUSE definită prin **produsul cartezian al domeniilor**, **CODPRODUS, DENUMIREPRODUS, UM, PRET,** adică:

**R=PRODUSE ⊂ (CODPRODUS x DENUMIREPRODUS x UM x PRET)**

**Relaţia este reprezentată** sub forma unui ***tablou dezvoltat pe linii şi coloane*** (de unde şi denumirea de tabel).

**Relaţia este formată** dintr-un ansamblu **de n-tupluri**.

**Fiecare tuplu** reprezintă ***un rând în cadrul tabelului***.

**Fiecare coloană** corespunde ***unui atribut***, descriind o caracteristică a entităţii din lumea reală, aşa cum rezultă din figura ce urmeaza:

**14**



**Numărul tuplurilor dintr-o relaţie reprezintă cardinalul relaţiei.**

**Numărul de atribute ale unei relaţii R reprezintă gradul relaţiei.**

***Exemplu*:** Se consideră o bază de date formată din tabelele **FURNIZORI** şi **FACTURI**, a căror structură este reprezentată în modelul de mai jos:

**Furnizori** (**CodFz**, NumeFz, Localitate, Telefon, ContBancar)

***cheia primară cheia candidat***

**Facturi** (**NrFact**, DataFact, DataScad, **CodFz**)

**cheia primară, cheie externă**

**TIPURI DE LEGĂTURI ÎNTRE RELAȚII**

**Chei**

**O cheie** este determinată de înțelesul atributelor și această *proprietate este invariantă în timp*, deci continuă să existe și după ce se inserează noi tupluri în relație.

* **O relație poate avea** mai multe chei, numite **chei candidat**. De obicei, dintre aceste chei candidat se desemnează una singură, numită ***cheie primară***.
* *Cheile candidat rămase* după alegerea cheii primare se numesc chei alternative.

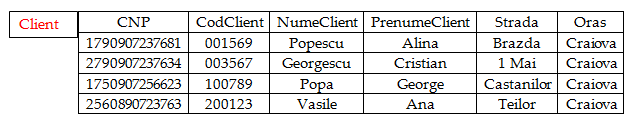
**15**

* ***Când schema relației are mai multe chei candidat***, alegerea dintre ele a cheii primare se face arbitrar. *De obicei, este indicată alegerea cheii primare cu un singur atribut sau cu numărul minim de atribute*.

***Exemplu:***

Se considerăm **schema** relației*:*

**Client\_schema(CNP, CodClient, NumeClient, PrenumeClient,Strada, Oras) si relația (tabelul!!) Client:**



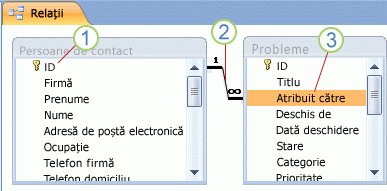
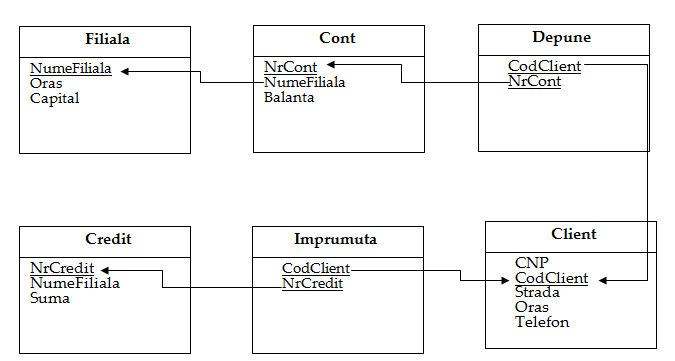
**NOTă:** Atributul ***CNP*** identifică unic toate tuplurile. De asemenea atributul *CodClient* identifică unic toate tuplurile. Unul din cele 2 atribute poate fi ales cheie primara.

* *Spre deosebire* de alte modele implementabile ale bazelor de date, la care legăturile între obiectele bazei de date se realizează la nivel fizic prin lanțuri de pointeri, în cazul modelului relational, legăturile se realizează la nivel logic prin mecanismul **cheilor externe**.
* Se numeşte **cheie externă (secundară)**un atribut (simplu sau compus) utilizat numai în scopul căutării unor tupluri.
* Fie R1 şi R2 două relaţii cu schemele **R1**(A,…) respectiv **R2**(B,…) şi A este cheie primara in R1, iar **dom(A)=dom(B)**. Se spune că atributul B reprezintă o cheie externă definită peste cheia primară a relaţiei R1.

**Tipuri de legături între relații**

1. Legături unu la unu (notație-1:1)
2. -Legături unu-la-mulți (notație-1:m, 1:∞)
3. Legături mulți-la-mulți (notație-m:n, ∞:∞)

**16**



**Legenda!**

**1. Cheia primară**

**2. Această linie reprezintă relația**

**3. Cheia externă**

**ACUM STRUCTURA MODELUILUI RELATIONAL POATE FI PREZENTATĂ DUPĂ CUM URMEAZĂ**

* ***Definitie*:Schema relației**, notată **R(D1, D2…, Dn)** este compusă din numele schemei **R** si o listă de atribute **D1, D2…, Dn**. Fiecare atribut **Di** reprezintă numele unui domeniu **D** din schema relației. **D** este domeniul lui **Di** și mai este notat **dom(Di)**.
* **Schema relației este folosită pentru a descrie o relație.**
* ***Exemplu:***

Schema relaţiei ***Cont\_schema*** poate fi scrisă sub forma:

**Cont\_schema(NrCont, Filială, Balanță).**

* ***Definitie*:**

**17**

Fiind dată o ***colecţie de mulţimi*** cu valori atomice D1, D2,... Dn (nu neapărat distincte) ***numite domenii***, se numeşte relaţie pe mulţimile D1, D2, ... Dn orice subset r(R) al produsului cartezian D1xD2x ... xDn .

* ***Exemplu:*** Fie D1 mulțimea care conține numerele conturilor { MD11ING012392330, MD21BCR014569879}, notat cu atrubutul **NrCont**, D2 mulțimea care conține denumirile filialelor {ING1, BCR2}, **Filiala**, D3 mulțimea care conține balanțele {500, 1000}, **Balanta**.
* **Atunci** **cont**(Cont\_schema)={(MD11ING012392330, ING1, 500), (MD21BCR014569879, BCR2, 1000)} reprezintă o relație .

|  |  |  |
| --- | --- | --- |
| **NrCont (D1)** | **Filiala (D2)** | **Balanta (D3)** |
| MD11ING012392330 | ING1 | 500 |
| MD21BCR014569879 | BCR2 | 1000 |

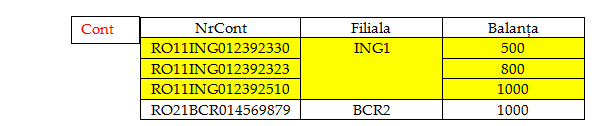
**PROPRIETĂŢILE RELAŢIILOR**

(*Obiecte, Relaţii independente*)

1. *Orice relaţie* are un nume unic *într-o bază de date*.
2. Orice atribut are un nume unic într-o relaţie.
3. Ordinea atributelor într-o relaţie nu are importanţă.
4. Ordinea tuplurilor într-o relaţie nu are importanţă.
5. Schema este un invariant al unei relaţii date.
6. O relaţie este variantă/se schimbă în timp.
7. În afara valorilor specifice fiecărui domeniu, în bazele de date relaţionale se acceptă şi o valoare specială, numită valoarea nulă şi notată **NULL**. Prin **NULL** se desemnează o valoare momentan necunoscută sau o valoare care nu are semnificaţie într-un anumit context.

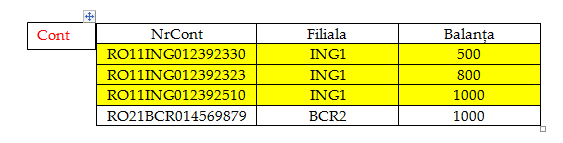
* În BD nu pot exista **obiecte** duble
* În BD nu pot exista **înscrieri** duble
* Orice relaţie depinde de cheile relaţiilor ce descriu un obiect

**Observatie!**  **Nu orice tabel reprezintă o relaţie**!



Prin transformarea tabelului de mai sus se obține următoarea relație

**18**



**REGULI DE INTEGRITATE A DATELOR**

1. ***Constrângeri de integritate.***

* Se numeşte **regulă de integritate** a datelor o **restricţie impusă valorilor datelor**.
* ***Regula de integritate a domeniului*** 
  + Toate ***valorile unui atribut*** trebuie sa fie atomice și de acelaşi tip.
* ***Regula de integritate a entităţii (a cheii primare)*** 
  + **Cheia primară** a unei relaţii **nu poate avea** valori nule.
* ***Regula de integritate a referirii*** 
  + **Valorile unei chei externe** trebuie să fie nule sau să se

regăsească printre valorile înregistrate ale cheii primare peste care este definită.

* ***Reguli de integritate specifice care pot fi definite***
  + Valorile numerice sunt cuprinse între anumite limite.
  + Valorile atributului se găsesc printre valorile dintr-o listă finită.
  + Nu sunt acceptate valori nule.
  + Nu sunt acceptate valori duplicate.

1. ***Operaţii:* COMPOZITIA RELATIILOR**

Ob1: **„Student”** Ob2: **„Obiect de Studiu”**

Nr. semestrului

Den. cursului

specialitate

Nume

curs

**Relaţie compusă: „Studiază”**

Nume\_S

Den. cursului

Nota

**Cheie externă.** Modelul relaţional **impune restricţii** asupra:

* **Integrităţii de referinţă**
* Un rînd (înscriere în Rel.O)
* Orice relaţie în BD trebuie să fie **normalizată.**

*Exemplu:*

**19**

**Student:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume** | **Curs** | **Specialitatea** | **Sport** | |
| **tip** | **categorie** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nume** | **Curs** | **Specialitatea** | **Tip Sport** | **Categoria sport** |

*(forma*

*atomară)*

**Definiţie:O relaţie cu atribute atomare reprezintă** **Prima Formă Normală (1FN).**

**ALTE OPERAŢII ASUPRA RELAŢIILOR: , x, \**



**Numărul tuplurilor dintr-o relaţie reprezintă cardinalul relaţiei /mosnosti/.**

**Numărul de atribute ale unei relaţii R reprezintă gradul relaţiei /stepeni/.**

Scheme compatibile dacă gradul e acelasi si atributele au acelasi tip

**„”**

22

24

18

14

Ion

Petru

Maria

Nicu

**Vîrsta**

**Nume**

22

14

Ion

Nicu

**Vîrsta**

**Nume**

22

24

18

Ion

Petru

Maria

**Vîrsta**

**Nume**



=



**„”**

22

Ion

**Vîrsta**

**Nume**

22

14

Ion

Nicu

**Vîrsta**

**Nume**

22

24

18

**Vîrsta**

Ion

Petru

Maria

**Nume**



=



**„\”**

**Vîrsta**

**Nume**

**Vîrsta**

**Nume**

22

24

18

Ion

Petru

Maria

**Vîrsta**

**Nume**

24

18

Petru

Maria

22

14

Ion

Nicu

\ =

**NumeStudent**

**Data**

**Disciplina**

**NumeStud**

**Data**

**Discuplina**

**„x”**

Ion

Petru

Maria

20

30

20

30

20

30

SGBD

Matematica

SGBD

Matematica

SGBD

Matematica

Ion

Ion

Petru

Petru

Maria

Maria

20

30

SGBD

Matematica

x =

**OPERAŢIILE: PROIECŢIE, CONCATINARE, SELECŢIE**

**PROIECŢIA (unară)–** are loc pe o singură relaţie, **pe o parte de atribute**.

**20**

***Exemplu:*** (STUDENT) Colaboratori Funcţia (*Înscrieri duble lipsesc*)

c/Doctor

L.superior

c/Doctor

L.superior

**Funcţia**

**C**

**C**

**TI**

**TI**

**Catedra**

**c/Doctor**

L.superior

**c/Doctor**

c/Doctor

**L.superior**

**L.superior**

**Funcţia**

C

C

C

TI

**TI**

**TI**

Ion

Petru

Mihai

Maria

Vova

Nicu

**Catedra**

**Nume**

**CONCATENARE (binară)–** are loc asupra a două relaţii.

***Exemplu:***SPECIALITATESTUDENT + după cîmp Cod. Student

**D.Sp.**

**Curs**

**Nume**

**C.St.**

**Curs**

**Nume**

**C.St.**

**D.Specialitate**

**Cod St.**

TI

Man-t

C

Tamara

Ion

Maria

3

1

2

**12**

**8**

**10**

1

1

2

1

3

Ion

Petru

Maria

Ana

Tamara

**8**

9

**10**

11

**12**

TI

Managment

C

**12**

**8**

**10**

**SELECŢIA (unară) –** are loc pe o singură relaţie (*după un anumit criteriu*).

*Exemplu:*  STUDENT

**Curs**

**Curs**

**Nume**

**Nume**

4

3

5

Petru

Elena

Maxim

1

4

3

5

Ion

Petru

Elena

Maxim

**Curs >= 3**

**21**

**PROIECTAREA BD RELATIONALE**

**REPETARE: POSTERUL 14-21 REPETARE**

**Include:**

**NORMALIZAREA RELAŢIILOR**

**Normalizarea relaţiilor –** process **de decompoziţie** a unor relaţii ÎN RELATII mai simple.

La baza procesului de normalizare se află algoritmul propus de cître matematicianul **Edgar Frank Codd** în 1970, care constă **din 2 paşi:**

1. ***Determinarea unei relaţii în prima formă normală (1FN).***
2. ***Excluderea unor dependenţe funcţionale între diferite atribute***

***prezente în prima formă normală.***

***Pentru procesul de normalizare Cod a propus 3 forme normale: 1FN, 2FN, 3FN, 3FNBC.***

***Exemplu:* 1FN – R cu atribute atomare.**

Nume

Specialitatea

Salariul

Sectorul

Telefon

Nr. paşap.

**sau**

Nr. paşap.

Nume

Nr.Copii

**TIPURI DE DEPENDENŢE FUNCŢIONALE**

Fie dată o relaţie **R** şi **x, y** oricare două atribute,

**dacă** în orice moment de timp fiecărui atribut **X** îi corespunde un singur atribut **Y**, atunci atributul **Y** depinde funcţional, **univoc de** **X**.

***Exemplu:*** **LECTOR**

**X Y**

L Gr. (1 : M)

SGBD

1

2

3

Ion

**Disciplină**

**Grupa**

**Nume**



Nume Disciplina (M : N)



(*Grupa şi Disciplina nu sînt legate*)

**22**

**NOTĂ!** ***Normalizarea*** relaţiilor ***depinde*** de ***tipul dependenţei Funcţionale*** dintre Atribute.

***Exemplu:*****R: 1FN**

Telefon

Sectorul

Salariul

Specialitatea

Nume

Nr. paşap.

**DEF:** *Dacă atributul diferit de atributul-cheie depinde de o parte a atributului-cheie compus atunci între ele există o* ***dependenţă parţială.***

***Exemplu:***

Nr.Copii

Nume

Nr. paşap.

1. (2) (3)

În caz contrar, dacă atributul depinde de cel cheie integral, atunci el depinde **funcţional complet** de atributul-cheie compus.

***Exemplu:***1 2 3

Nume/Pre.

Departamentul

Nr. Tel.

Salariul

Vîrsta

Nume

Nr. paşap.

Vîrsta = f(complet - Cheie)

**DEF:** *Fie dată o relaţie R care are 3 atribute:* ***X,Y,Z****. Dacă* ***Y*** *depinde de* ***X****, iar* ***Z*** *depinde de* ***Y****, atunci* ***Z*** *depinde de* ***X*** *şi această dependenţă se numeşte* ***dependenţă tranzitivă. 3>2 2>1 3>1***

***Exemplu:*****X Y Z**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr. paşarpot** | **Nume** | **Salariu** | **Departament** | **Nr. telefon** |
|  |  |  |  |  |

*Nr. paşaport departament Nr. telefon*



**DEF:** *Fie dată relaţia R şi* ***X*** *un atribut al relaţiei date, dacă există mai multe atribute în relaţia dată care depind de atributul* ***X****, atunci avem o dependenţă* ***multivocă.*** *(****1 : N****)*

**A DOUA FORMĂ NORMALĂ (2FN) –** o relaţie se află în a doua formă normală, dacă:

**DEF:;**

1. Se află în prim formă normală (**1FN**)
2. ***Orice atribut defierit de atributul cheie depinde funcţional complet de elementul cheie compus.***

***Exemplu:*** *„***R0”**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nr. paşaport** | **Nume** | **Salariu** | **departament** | **telefon** | **copii** | |
| **nume** | **vîrsta** |
| 11  12 | Ion  Petru | 130  120 | 1  2 | 3-33  5-55 | **Ion**  **Vova**  Elena  Grigore | **20**  **11**  6  18 |

**R 1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nr. paşaport | Nume copil | Vîrsta | F/Nume | Salariu | Telefon | Departament |
| 11  11  11  12 | Ion  Vova  Elena  Grigore | 20  11  6  18 | Ion  Ion  Ion  Petru | 130  130  130  120 | 3-33  3-33  3-33  5-55 | 1  1  1  2 |

***Algoritmul:***

1. **R0** se aduce la prima formă normală (**1FN**).
2. În cazul în care există dependenţă funcţională completă ele se exclud printr-o **operaţie de proiecţie**. Această proiecţie se efectuează pe o parte a elementului cheie compus care depind informaţional de ea.

***Exemplu:* R2** *complet*

|  |  |  |
| --- | --- | --- |
| Nr. paşaport | Nume copil | Vîrsta |
| 11  11  11  12 | Ion  Vova  Elena  Grigore | 20  11  6  18 |

**R3** *parţial*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. paşaport | F/Nume | Salariu | **Departament** | **Telefon** |
| 11  12 | Ion  Petru | 130  120 | **1**  **2** | **3-33**  **5-55** |

**A TREIA FORMĂ NORMALĂ (3FN):**

1. **2FN** care nu are dependenţă funcţională tranzitivă a atributelor de atributele cheie. (**Nu a fost menţionat cheia-compusă!**)

**NOTĂ!** În **R3** există dependenţe tranzitive:

***Nr. paşaport departament Nr. telefon***



**D = f(N) NrT = ϕ(D)**

**23**

**PROBLEME:**

**Algoritmul:**Similar cazului de trecere de la prima formă la a doua cu excluderea dependenţelor funcţionale tranzitive.

**Proiecţia R3 R4 şi R5 /vezi posterul 20/!!!**



|  |  |  |  |
| --- | --- | --- | --- |
| **Nr. paşaport** | **FNP** | **Salariu** | **Departament** |
| 11  12 | Ion  Petru | 130  120 | 1  2 |

|  |  |
| --- | --- |
| **Departament** | **Telefon** |
| 1  2 | 3-33  5-55 |

**R 4: R 5:**

**Putem menţiona că procesul de normalizare conduce la majorarea bruscă a numărului de relaţii**, iar acest proces la rîndul său impune o condiţie dură asupra procesului **reversivibilităţii normalizării**.

**!**

**!**

Acest proces de reversibilitate se păstrează prin proprietatea fundamentală a relaţiilor în procesul de normalizare, care se numeşte ***integritate de referinţă*** a datelor. (*permite ca procesul să fie reversibil şi complet*)

**În formele precăutate mai sus** care în permanenţă exclud dependenţele funcţionale a atributelor de Atributele cheie, **EXISTĂ ÎNSĂ CAZURI CÎND ELEMENTUL CHEIE, SAU O PARTE DIN EL, *DEPINDE DE UN ATRIBUT DIN AFARA LUI.***

**A TREIA FORMĂ NORMALĂ BOIS-COD (3FNBC):** (este a **3FN** şi în ea lipsesc dependenţele funcţionale a atributelor cheie de celelalte atribute).

1. **R 3FN**



1. **În R există dependente partiale a partilor din cheie de atr.Student!**

***Exemplu:* R** - **LUCRARE DE CURS**

**Disciplina**

L1

L2

L3

L1

L1

L2

L3

Lector

**R1**

1

1

1

2

2

3

3

**Student**

**Disciplina**

**Lector**

**Algoritmul!**

1

2

3

4

5

6

7

1

1

1

2

2

3

3

L1

L2

L3

L1

L1

L2

L3

(*să evităm dependenţele*

*Funcţionale parţiale inverse*)

**R2**

1. (2)

1

2

3

4

5

6

7

Student

**Disciplina**

**IMPORTANT:** **U (1) cu (2) în A.Cheie**

1

1

1

2

2

3

3

Disciplina

Student

L

D

**R în 3FN**

**Dar, există dependenţa funcţională parţială a unei**

**Părţi a A.Cheie de un atribut al R** – **PROBLEME**

(***Disciplina*** *depinde de* ***Student*** *şi* ***Lector*** *depinde de* ***Student*)**

R1\_R2 = R (*concatinare*)

**24**

**RELAŢIA SE AFLĂ ÎN A** **PATRA FORMĂ NORMALĂ (4FN)**,

dacă ea este în a **3FNBC** şi în ea **nu există dependenţe funcţionale multiple**.

**RELAŢIA SE AFLĂ ÎN A CINCEA FORMĂ NORMALĂ (5FN),**

dacă ea este în a **4FN** şi **este descompusă în aşa mod încît procesul de reuniune a lor să conducă la relaţia iniţială.**

**Astfel:**

**În concluzie**

***procesul de normalizare a relaţiilor este un proces de decompoziţie care exclude un anumit tip de dependenţe funcţionale, reducind redundanta si surplusul de date:***

1. **P-u 2FN – dependenţe parţiale a atributelor de atribute-cheie in 1FN.**
2. **P-u 3FN – dependenţa tranzitivă a atributelor de atribute-cheie in A 2FN.**
3. **P-u 3FNBC – se exclud dependenţele parţiale a elementelor cheie de alte atribute din 3FN.**
4. **P-u 4FN – dependenţă funcţională multiplă din 3FNBC.**

**Notă! Aceste eliminări de dependenţe funcţionale contribuie la excluderea dublării datelor şi asigurarea liberii executări a operaţiilor de: INSERT, DELETE şi UPDATE.**

**DAR!!!!Normalizarea măreşte numărul de relaţii şi conduce la o anumită măsură la creşterea timpului la etapa de prelucrare a datelor, dar în acelaşi timp conduce şi la gestionarea BD.**

**25**

**PROIECTAREA BAZELOR DE DATE**

1. **Crearea Modelului Conceptual**
2. **Modelul Logic**
3. **Modelul Fizic**

**PBD: DS M.F.; Proces Iterativ**



**+ Aplicativă**

**+ Teoretică**

PR-RE în 2 Etape **Preproiect** ( *1. Justificarea;*

*2. Formularea PR.*)

**Pr. Tehnic** (*1. Pr. BD; 2. Evaluarea EFF.*)

Crearea

Procesului Tehnic

PR-RE într-o etapă

Necesităţile Informaţionale ale Utilizatorilor

DS: Studiul şi analiza, Descrierea DS, Restricţii

Restricţiile SGBD

Proiectarea Modelului Logic

Proiectarea Modelului Fizic a BD

Evaluarea Modelului Fizic al BD

**Realizarea PR. BD**

Proiectarea Modelului Conceptual

Analiza şi Descrierea Aspectului Informaţional al Informaţiei

Analiza datelor, Obiectelor, Legăturilor, PPA, PPA-Potenţiale

Evaluarea caracteristicilor. Pachet de Pr. A. în funcţie de frecvenţa de utilizare şi prioritate

**26**

**CAPITOLUL III**

***Exemplu:***

1. **DS „Angajatul”**
2. **SB**

3. Locurile de muncă

4. Lista copiilor

5. Fondul de încăperi pentru L.M.

6. Informaţie Generală

2. Salariul

1.Lista Generală

**MODELUL CONCEPTUAL**

1. **Ieşiri Informaţionale:**

1 2

Salariul

FNP

N. Î.

FNP

Număr Înregistr.

3 4

Nume Copil

Vîrsta

N. Î.

Telefon

Nr. Biroul

FNP

N. Î.

5 6

N. Î.

Telefon

Nr. Biroul

Salariul

FNP

Vîrsta

N. Copil

Telefon

Nr. Biroul

**3.1** **Cercetarea DS şi evidenţierea Restricţiilor**

**Restricţii!**

**1. NI în mod univoc determinăm „M”**

**2. unul şi acelaşi muncitor nu poate lucra simultan în două pătrăţele.**

**3. În fiecare „B” UN! Telefon**

**4.Orice „M” poate avea 1 copil.**



**5. În „B” .... „M”**

**6. FNP „” FNP „”**



**7. „M” nu are Salariu în cumul**

**3.2 Determinarea Relaţiei „Atribute Ieşiri Informaţionale”**

**27**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ATRIBUTE** | **Ieşiri Informaţionale** | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** |
| **N.Î.** | **+** | **+** | **+** | **+** | **-** | **+** |
| **FNP** | **+** | **+** | **+** | **-** | **-** | **+** |
| **Salariu** | **-** | **+** | **-** | **-** | **-** | **+** |
| **N. Camerei/B** | **-** | **-** | **+** | **-** | **+** | **+** |
| **Telefon** | **-** | **-** | **+** | **-** | **+** | **+** |
| **Nume Copil** | **-** | **-** | **-** | **+** | **-** | **+** |
| **Vîrsta** | **-** | **-** | **-** | **+** | **-** | **+** |

**3.3 Dterminarea Legăturilor dintre Atribute**

(1) II Nr. 1 NI FNP (*Acheia NI*)



(2) Nr. 2 NI FNP; FNP Salariul; NI Salariul (*Atribut Cheie NI*)



**!!! RESTRICŢIA 6 !!!**

(3) Nr. 3NI FNP; NBiroul Ntelefon



*Există legături tranzitive:* **NI FNP NB; NI NB Ntel, FNP NB NT** FNP Nbirou (*Restricţia 3!*)



(4) Nr. 4 NI NumeCopil; NumeCopil Vîrsta (*Acheie NI+NumeCopil*)



(5) Nr.5 Nbirou Ntelefon (*Acheie NBirou*)



(6) Nr. 6

NI FNP; FNP Salariu Nbirou Ntelefon = **\*** Dependenţă de NI



**\***

Vîrsta NumeCopil FNP NI



**LEGĂTURI (Dependenţe)**

* **Univoce - Multivoce**
* **Parţiale - Tranzitive**
* **Complete (*Vîrsta NI*)**



**3.4 În baza 3.3 şi 3.2 descriem schemele relaţiilor care se vor utiliza pentru Ieşiri Informaţionale 1-6**

SAL

FNP

N.Î.

FNP

N.Î.

**1 2**

NTel

NBirou

FNP

N.Î.

**3**

N.Î.

**4 5**

NTel

NBirou

Vîrsta

NumeCopil

NTel

NBirou

SAL

FNP

Vîrsta

NumeCopil

N.Î.

**6**

**28**

**Capitolul IV MODELUL LOGIC**

***(utilizăm Modelul Relaţional)***

1 2

Salariul

FNP

N. Î.

FNP

Număr Înregistr.

3 4

Nume Copil

Vîrsta

N. Î.

Telefon

Nr. Biroul

FNP

N. Î.

5 6

N. Î.

Telefon

Nr. Biroul

Salariul

FNP

Vîrsta

N. Copil

Telefon

Nr. Biroul

**NOTĂ! +R 6:**

1. Operaţia de concatinare 2 şi 3 NÎ (după atribute)
2. Rezolvarea punctului 1) cu relaţia 4

**+R 1:** Proiecţia 3 pe atributeke NÎ & FNP sau Relaţia 2

Relaţiile 1, 6 – se exclud din R – I, rămîn relaţiile 2 -5!

**V. NORMALIZAREA**

*1.* ***R-iile 2, 5 1FN***



*2.* ***R-ia 4 2FN***(Vîrsta dependenţă funcţională completă de atributul-cheie)



*3.* Există dependenţe funcţionale tranzitive **N.Î. NBirou NTelefon**



**Proiecţia 3 pe atributele N.Î. şi FNP**

**FNP**

**FNP**

**N. Î.**

**NBirou**

**Şi NBirou şi Ntelefon 5**

**3.1 3.2**

**Pentru descrierea Domeniului de Studiu obţinem următoarele RELAŢII:**

NBirou

FNP

NTel

NBirou

Vîrsta

NCopil

N. Î.

FNP

N. Î.

N. Î.

SAL

FNP

**1FN 3FN 2FN 3FN 3FN**

**Proiecţia 2: N.Î. şi Salariu Relaţiile ce descriu DS**



**„Salariu”**  **„FNP” „Vîrsta” „Telefon” „Birou”**

Vîrsta

NCopil

N. Î.

FNP

N. Î.

SAL

N. Î.

NBirou

FNP

NTel

NBirou

**29**

**SCHEMA „MODELUL CONCEPTUAL” SCHEMA ER**

***Vîrsta***

Vîrsta

NCopil

N. Î.

***FNP Biroul Telefon***

SAL

N. Î.

NTel

NBirou

NBirou

FNP

FNP

N. Î.

***Salariu***

**VI. ANALIZA MODELULUI FIZIC** *(Caracteristici PC, SPEED, MEMORY)*

**VII. MACHETUL BAZEI DE DATE**

**1. FNP.dbf** Numărul de înregistrare *N.Î, N(6)* **AUTOMAT**

Familia.Numele.Prenumele *FNP, C(30)* **SELECTARE**

**2. SALARIU.dbf** Numărul de înregistrare *N.Î, N(6)* **AUTOMAT**

Salariul *SAL, N (9,2)* **SELECTARE**

**3. BIROU.dbf** Familia.Numele.Prenumele *FNP, C(30)* **AUTOMAT**

Numărul Biroului *NB, N (3)* **SELECTARE**

**4. TELEFON.dbf** Numărul Biroului *NB, N (3)* **AUTOMAT**

Numărul Telefonului *NT, C (8)* **SELECTARE**

**5. VÎRSTA.dbf** Numărul de înregistrare *N.Î, N(6)* **AUTOMAT**

Nume Copil *NUMC, C (10)* **SELECTARE**

Vîrsta *VSTA, N (3)* **SELECTARE**

**VIII. EVALUAREA MEMORIEI BAZEI DE DATE**

***36 oct \* M + 15 oct \*N + 33 oct \* K + 11 oct \*T + 19oct \* Q***

**IX. MENIUL DE BAZĂ!**