BAZE  DE  DATE  sI SISTEME  DE  GESTIUNE  A  BAZELOR  DE  DATE

<http://www.scritub.com/stiinta/informatica/BAZE-DE-DATE-SI-SISTEME-DE-GES14226.php>

***1.1.  Introducere în bazele de date***

Sistemele de baze de date pot fi considerate ca cea mai importanta realizare în domeniul ingineriei programarii. Bazele de date constituie cadrul sistemelor informationale si au modificat fundamental modul de operare al organizatiilor. Predecesorul sistemului de baze de date este sistemul bazat pe fisiere.

***1.1.1. Organizarea datelor în fisiere***

            O posibilitate de organizare a informatiilor este reprezentata de sistemul bazat pe fisiere.

**Fisierul** cuprinde un ansamblu de date memorate pe suport magnetic, fiind elementul fundamental al organizarii informatiei pe disc.

Un **sistem bazat pe fisiere** este o colectie de programe aplicatie care efectueaza servicii pentru utilizatorii finali. Fiecare program defineste si gestioneaza propriile date. O caracteristica esentiala a acestui sistem este abordarea *descentralizata* a datelor operationale ale organizatiei, astfel încât fiecare departament îsi stocheaza si controleaza propriile date.

            Organizarea datelor în fisiere, desi este destul de utilizata, are o serie de neajunsuri care limiteaza eficienta aplicatiilor utilizator.

            · *Redundanta si inconsistenta datelor*

            Fisiere de date independente contin date care se repeta. Acest lucru este o consecinta a descentralizarii datelor. Efectele acestui neajuns sunt *gestionarea complexa* a datelor (deoarece aceleasi date apar în locuri diferite), *actualizarea* *greoaie* a datelor (modificarea unei singure date presupune actualizarea ei în toate fisierele în care apare), consumul inutil al *spatiului de memorie*.

            · *Neintegrarea datelor* (separarea si izolarea datelor)

            Atunci când datele sunt pastrate în fisiere separate, ele sunt mai greu de accesat. Dispersia datelor în fisiere independente complica accesul utilizatorilor la informatii, necesitând crearea de programe particulare pentru extragerea datelor solicitate, deoarece trebuie accesate fisiere diferite.

            · *Dependenta datelor de program*

Structura fizica si organizarea fisierelor de date sunt definite în cadrul aplicatiei ce manevreaza fisierul. Astfel, modificari legate de un fisier,  presupun executarea de modificari în toate programele în care este referit fisierul respectiv.

            · *Dependenta formatului de fisier de limbajul de programare*

            Deoarece structura fisierelor este încorporata în programele aplicatie, ea este dependenta de limbajul în care este scrisa aplicatia. Fisiere create cu limbaje diferite au formate diferite si nu pot fi prelucrate simultan.

            · *Interogarea fixa a programelor de aplicatie*

Sistemele bazate pe fi 242c29c siere fiind în mare masura dependente de programatorul de aplicatii, interogarile sau rapoartele necesare trebuie scrise de catre programator.

· *Dificultatea accesului*

Provine din izolarea datelor în fisiere diferite, necesitatea scrierii unui program pentru accesarea fisierului sau din faptul ca fisierele traditionale nu faciliteaza accesarea datelor dupa mai multe criterii, specifice diferitelor tipuri de utilizatori.

            Într-o întreprindere, o informatie este exploatata de mai multi utilizatori. De exemplu, pentru departamentul care se ocupa cu gestiunea stocurilor, intrarile de materiale trebuie ordonate pe depozite si repere, în timp ce pentru departamentul care se ocupa cu decontarile cu partenerii de afaceri ai întreprinderii, intrarile trebuie ordonate pe furnizori ai materialelor.

            · *Complexitatea deosebita a actualizarilor*

O actualizare presupune adaugarea, modificarea sau stergerea unor informatii din fisiere. Cum prelucrarile se desfasoara în timp real, de la mai multe terminale în mediile multi-utilizator, pot aparea situatii conflictuale atunci când doi utilizatori doresc modificarea simultana a unei aceleiasi date. Rezolvarea acestui gen de conflicte presupune existenta unui program-supervizor al prelucrarilor care este greu de realizat în cazul existentei unei multitudini de fisiere cu formate diferite.

            · *Probleme de securitate*

            Ţin de dificultatea crearii unui mecanism care sa protejeze datele din fisiere de accesul neautorizat.

Toate limitarile tratarii bazate pe fisiere se datoreaza urmatorilor doi factori:

1)      Definitia datelor este încorporata în programele aplicatie în loc de a fi stocata separat si independent.

2)      Nu exista un control al accesului si manipularii datelor dincolo de cel impus de catre programele de aplicatie.

*1.1.2.      Baze de date - definitii si elemente specifice*

*Intuitiv*, o **baza de date** **(BD)** este un set de date legate între ele, iar **sistemul de gestiune a bazelor de date** (**SGBD**) este programul care administreaza si controleaza accesul la baza de date.

*Definitie*. O **baza de date** este un ansamblu integrat de înregistrari sau de fisiere reunite si structurate în mod logic.

*Detalierea definitiei*. Baza de date este un depozit de date unic, larg, care este definit o singura data si este utilizat simultan de catre mai multi utilizatori. Baza de date contine, alaturi de date, si o descriere a acestora. Din acest motiv, ea este definita si ca *o colectie autodescrisa de înregistrari integrate*. Descrierea datelor este cunoscuta sub denumirea de **catalog de sistem** (sau **dictionar de date** sau **meta-date** - "date despre date"). Natura autodescriptiva a bazelor de date este cea care produce *independenta program - date* (se separa definitia datelor de programele aplicatie).

Între datele bazei de date exista **relatii logice**.

O definitie care include caracteristicile bazei de date (acestea vor fi detaliate ulterior) este urmatoarea: **baza de date** este un ansamblu de colectii de date, *organizat* pe niveluri de organizare a datelor în memoria externa, *coerent,* conform unor restrictii de integritate, *structurat*, conform unui model de date, cu o *redundanta*minima si controlata, asigurata printr-o tehnica de proiectare, *accesibil* mai multor utilizatori în timp util.

O **entitate** (**articol, înregistrare logica**) este un obiect distinct (o persoana, loc sau lucru, concept sau eveniment, o operatie economica, un mijloc economic) care trebuie reprezentat în baza de date. Un **atribut** este o caracteristica care descrie un aspect oarecare al obiectului care se înregistreaza în baza de date. **Valoarea** reprezinta marimea ce se atribuie fiecarei caracteristici din cadrul unei entitati. O **relatie**este o asociatie între mai multe entitati. De exemplu, se poate constitui o baza de date care sa contina informatii despre angajatii unei firme. Va exista o *entitate* - Personal - în care se vor preciza, ca *atribute*, numarul de personal, numele, prenumele, adresa si functia fiecarei persoane. Pentru atributul functie pot exista valori ca: manager, adjunct, asistent. Entitatea Personal este legata printr-o *relatie*de entitatea Filiala care contine informatii despre filialele la care lucreaza angajatii firmei. La rândul sau, Filiala va contine *atribute* ca: numar\_filiala, nume\_filiala, adresa, numar\_telefon, numar\_fax, numar\_personal. Aceste atribute reprezinta antetul unui tabel ale carui coloane se completeaza cu *valori* ale atributelor.

Caracteristica principala a aplicatiilor de baze de date consta în faptul ca accentul este pus pe operatiile de memorare si regasire, efectuate asupra unor volume mari de date, si mai putin asupra operatiilor de prelucrare a acestora. Principala operatie care apare în orice aplicatie de baze de date este cea de **regasire** a datelor, în scopul obtinerii de informatii din baza de date. Aceasta este finalitatea si sensul existentei oricarei baze de date. O baza de date este creata pentru a putea fi **interogata**. Interogarea are la baza anumite conditii (criterii) care permit selectarea anumitor date din baza de date. Alaturi de operatii de regasire, se executa operatii de **adaugare**, pentru introducerea de noi date în baza de date, operatii de **stergere** pentru datele devenite inutile sau operatii de **actualizare** a unor date existente.

Organizarea datelor în baze de date constituie o forma de **centralizare** a acestora.

Accesul utilizatorului la informatiile despre **structura unei baze de date** se realizeaza prin intermediul unui software de aplicatii numit **dictionar de date**. Avantajele organizarii informatiilor în baze de date decurg tocmai din existenta acestui fisier de descriere globala a bazei de date. Extragerea si modificarea datelor - altfel spus, lucrul cu fisierele de date - se deruleaza exclusiv prin intermediul dictionarului în care se gasesc informatii privitoare la structura datelor si restrictiile îndeplinite de acestea.

Dictionarul de date contine:

· denumirile, tipurile si dimensiunile articolelor de tip date;

· tabelele bazei de date (denumire, descriere, chei primare si secundare etc.);

· constrângerile de integritate asupra datelor;

· utilizatorii care au drepturi de acces la un anumit tabel;

· optimizarea bazei de date (prin fisiere index, tehnica clustering etc.).

*1.1.3.      Depozite si banci de date*

O baza de date special conceputa prin agregarea tuturor datelor unei întreprinderi în vederea sprijinirii procesului de fundamentare a deciziilor si de analiza a datelor constituie asa numitul **depozit de date**(**data warehouse**).

*Tipuri de date* memorate în depozite de date:

- date curente din sistemele operationale (tranzactionale);

- date de detaliu privind perioadele precedente (date din arhive);

- date agregate cu provenienta interna sau externa necesare procesului decizional (baze de date publice, sondaje, date de prognoza etc.);

- metadate care asigura exploatarea depozitului de date prin reguli de extragere si conversie, reguli de agregare etc.

Spre deosebire de colectiile de date utilizate în *sistemul operational* (bazele de date) - orientate spre optimizareasi siguranta procesarii datelor - datele din depozitul de date sunt organizate într-o maniera care sa permita *analiza* lor. Rezulta ca depozitul de date acopera un *orizont temporal mai mare*, contine atât *date interne*,cât si*externe* si este optimizat pentru a raspunde unor *interogari complexe* ale unei game *diversificate de utilizatori*.

Depozitele de date se pot organiza la diverse niveluri:

· la nivelul entitatii social-economice (data warehouse);

· la nivelul unei structuri functionale (filiala, departament, serviciu) (data marts);

· la nivelul evenimentelor elementare (tranzactionale);

În implementarea unui depozit de date pot fi folosite mai multe variante:

· implementarea unui sistem descentralizat în care datele sunt pastrate în unitati independente (data marts). Aceste unitati contin date relevante pentru un anumit aspect al operatiunilor derulate în entitatea social-economica.

· implementarea unei surse de date unice la care au acces utilizatorii din toate structurile functionale.

· implementarea unei surse de date centralizate la nivelul entitatii social-economice cu existenta unor unitati dependente (dependent data marts) care prezinta subseturi ale datelor (din depozitul central de date) ce au fost selectate si organizate pe domenii de aplicatii.

Utilizarea depozitului de date se face cu aplicatii specializate de analiza a datelor, dar si prin elaborari de rapoarte sau extrageri de date pentru aplicatii de birotica.

Principalul inconvenient al depozitelor de date este dimensiunea enorma, de ordinul giga sau teraoctetilor.

Pentru a putea fi exploatata de catre utilizatori, o baza de date trebuie sa aiba un set de programe care sa permita exploatarea rationala a datelor - un sistem de gestiune a bazei de date. Sistemul de gestiune a bazelor de date interactioneaza cu structura bazei de date, sistemul de calcul, sistemul de operare, programele de aplicatii si utilizatorii constituindu-se **banca de date**. Se pot organiza banci de date în diferite sfere de activitate.

***1.1.4. Sisteme de gestiune a bazelor de date***

**Sistemul de gestiune a bazelor de date** **(SGBD)**este un ansamblu de programe ce permit utilizatorilor sa interactioneze cu o baza de date, în vederea crearii, actualizarii si interogarii acesteia.

***1.1.4.1. Servicii furnizate de un SGBD***

E.F. Codd de la Laboratorul de Cercetare IBM a enuntat 8 servicii pe care trebuie sa le furnizeze un SGBD complet:

**1) Stocarea, regasirea si reactualizarea datelor** este functia fundamentala a unui SGBD. Pentru asigurarea acestei functionalitati, SGBD-ul trebuie sa ascunda fata de utilizator detaliile privind implementarea fizica interna (cum ar fi organizarea fisierelor si a structurilor de stocare).

**2) Un catalog accesibil utilizatorului** care sa contina descrierea articolelor sale. Catalogul de sistem (dictionar de date) trebuie sa fie accesibil atât utilizatorilor, cât si SGBD-ului.

**3) Asigurarea tranzactiilor**. Un SGBD trebuie sa furnizeze un mecanism care sa garanteze ca sunt efectuate toate reactualizarile corespunzatoare unei anumite tranzactii. O **tranzactie** consta dintr-o serie de actiuni realizate de un singur utilizator sau de un program aplicatie prin care se acceseaza sau se schimba continutul bazei de date. De exemplu, adaugarea unui nou membru al personalului în baza de date, reactualizarea salariului unui membru al personalului etc. Daca tranzactia esueaza în timpul executiei - posibil din cauza unei defectiuni a calculatorului - atunci baza de date va fi într-o stare **incoerenta**; unele modificari au fost efectuate, iar altele nu. Pentru a aduce din nou baza de date într-o stare coerenta, modificarile efectuate trebuie anulate.

**4) Servicii de control concurente**. Un SGBD trebuie sa furnizeze un mecanism care sa garanteze ca baza de date este corect reactualizata, atunci când mai multi utilizatori efectueaza simultan operatii. Un obiectiv major în utilizarea unui SGBD este ca acesta sa permita mai multor utilizatori sa acceseze concurent datele partajate.

**5) Servicii de reconstituire**. Un SGBD trebuie sa asigure un mecanism de reconstituire a bazei de date, în cazul în care aceasta este deteriorata într-un fel oarecare.

De exemplu, daca o tranzactie nu a fost efectuata în întregime, atunci baza de date trebuie readusa la o stare coerenta. Esecul poate fi rezultatul unei erori software sau hardware care determina oprirea SGBD-ului sau poate sa fie rezultatul anularii tranzactiei de catre utilizator înainte ca tranzactia sa fie completa. În toate aceste cazuri, SGBD-ul trebuie sa asigure un mecanism de reconstituire a bazei de date într-o stare coerenta.

**6) Servicii de autorizare**. Un SGBD trebuie sa furnizeze un mecanism care va garanta ca numai utilizatorii autorizati pot accesa baza de date.

O serie de date aflate în baza de date trebuie sa fie protejate fata de unii utilizatori. De exemplu, salariile personalului pot fi vazute numai de managerii filialei. Pe lânga aceasta, baza de date trebuie protejata la accesul neautorizat. Termenul de **securitate** se refera la protectia bazei de date împotriva accesului neautorizat, fie intentionat, fie accidental.

**7) Suport pentru comunicarea datelor**. Un SGBD trebuie sa poata fi integrat unui software de comunicatie.

Majoritatea utilizatorilor acceseaza baza de date de la nivelul terminalelor. Uneori, aceste terminale sunt conectate direct la calculatorul ce contine SGBD-ul. Alteori, terminalele sunt situate la distanta si comunica prin retea cu calculatorul ce contine SGBD-ul. În ambele situatii, SGBD-ul primeste cererile ca **mesaje** si raspunde în acelasi mod. Aceste transmisiuni sunt administrate de Gestionarul de Comunicare a Datelor (Data Communication Manager - DCM). Chiar daca DCM nu face parte din SGBD, este necesar ca acesta din urma sa poata fi integrat cu diferite tipuri de DCM, pentru a fi viabil din punct de vedere comercial. SGBD-urile pentru calculatoare personale trebuie sa poata fi rulate într-o retea locala astfel încât sa fie posibila partajarea de catre utilizatori a unei baze de date centralizate. Acest lucru nu înseamna ca baza de date este distribuita în retea în conformitate cu cerintele utilizatorilor, ci înseamna ca utilizatorii trebuie sa poata accesa o baza de date centralizata de la locatii aflate la distanta. Acest tip de topologie se numeste *prelucrare distribuita*.

**8)** **Servicii de integritate**. Un SGBD trebuie sa furnizeze mijloace care sa asigure ca atât datele din baza de date, cât si modificarile acestora respecta anumite reguli.

Integritatea bazei de date se refera la corectitudinea si coerenta datelor stocate. Ea poate fi considerata ca un alt tip de protectie a bazei de date. Fiind legata de securitate, implicatiile ei sunt mai extinse - integritatea are în vedere însasi calitatea datelor. De obicei, integritatea se exprima în termeni de constrângeri, care reprezinta regulile de coerenta pe care baza de date nu are voie sa le încalce. De exemplu, în cadrul unei agentii imobiliare, s-ar putea specifica o constrângere ca nici un membru al personalului sa nu poata gestiona mai mult de 10 proprietati la un moment dat. În acest caz, se doreste ca SGBD-ul sa verifice ca aceasta limita nu este depasita în momentul repartizarii unei proprietati unui membru al personalului si sa împiedice repartizarea daca limita a fost atinsa.

Pe lânga aceste 8 servicii, ar fi rezonabil sa ne asteptam ca si urmatoarele doua sa fie asigurate de un SGBD.

**9) Servicii pentru promovarea independentei de date**. Un SGBD trebuie sa includa facilitati pentru sustinerea independentei programelor fata de structura reala a bazei de date, din punct de vedere *fizic* (modificarea datelor si a tehnicilor fizice de memorare fara rescrierea programelor de aplicatii) si *logic*(adaugarea de noi articole sau extinderea structurii fara a necesita rescrierea programelor);

**10) Servicii utilitare**. Un SGBD trebuie sa ofere un set de servicii utilitare care ajuta la administrarea bazei de date. De exemplu:

·        facilitati de import - export pentru a încarca sau descarca baza de date din sau în fisiere;

·        facilitati de monitorizare, pentru urmarirea utilitatii si operarii bazei de date;

·        programe de analiza statistica, pentru urmarirea utilizarii si operarii bazei de date;

·        gestionarea spatiului fizic al dispozitivelor de stocare (îndepartarea fizica a înregistrarilor, realocarea spatiului);

***1.1.4.2. Obiectivele si functiile unui SGBD***

Din serviciile enumerate, se pot desprinde urmatoarele *obiective* importante asigurate de un SGBD:

· **independenta datelor fata de programe**, atât din punct de vedere fizic, cât si logic;

· **asigurarea integritatii datelor** prin controlul accesului concurent si a posibilitatilor de refacere a bazei de date;

· **asigurarea unei redundante minime si controlate a datelor** prin definirea unei structuri cu o cantitate cât mai mica de date, evitându-se ambiguitatea;

· **facilitati de descriere si utilizare a datelor** prin limbajele de descriere si manipulare a datelor, prin folosirea datelor de catre mai multi utilizatori în diverse aplicatii, prin existenta unor limbaje performante de interogare;

· **securitatea datelor**, prin restrictii de autorizare la accesarea datelor;

· **partajarea datelor între utilizatori** - accesul mai multor utilizatori la aceleasi date pe baza unor criterii de prioritate.

Petru ca un sistem software sa fie SGBD, el trebuie sa asigure obiectivele de mai sus prin patru *functii*proprii: definirea, manipularea, utilizarea si administrarea datelor.

1) Functia de **definire a datelor** permite descrierea structurii bazei de date cu ajutorul unui **limbaj de definire a datelor**(**LDD**), stabilind criterii de validare a datelor, metode de acces la date si de asigurare a confidentialitatii si integritatii lor. Toate aceste elemente se regasesc în *schema bazei de date*. Executia definitiilor limbajului se materializeaza într-un ansamblu de tabele, memorate într-un fisier special denumit *dictionar de date*.

2) Functia de **manipulare a datelor** permite încarcarea/validarea datelor, actualizarea datelor si interogarea (regasirea) lor. Toate aceste operatii se realizeaza cu ajutorul unui **limbaj de manipulare a datelor**(**LMD**). În cadrul acestuia se dezvolta un **limbaj de interogare** a bazei de date. Exista doua tipuri de limbaje LMD - **procedurale** si **neprocedurale**. Primele trateaza baza de date înregistrare cu înregistrare, iar ultimele opereaza asupra unor seturi de înregistrari. Cel mai uzual tip de limbaj neprocedural este **limbajul structurat de interogare** **SQL**, specific SGBD-urilor relationale.

3) Functia de **utilizare a datelor** asigura interfetele necesare comunicarii utilizatorilor cu baza de date.

4) Functia de **administrare a bazei de date** permite gestionarea corecta a bazei de date, atât în procesul de realizare a bazei, cât si apoi în procesul de exploatare a acesteia.

***1.1.4.3. Elementele componente ale unui SGBD***

            Pot fi identificate 5 componente principale ale mediului SGBD:

|  |
| --- |
|  |
|  | http://www.scritub.com/files/informatica/1758_poze/image001.gif |

***Figura 1.1.*** Componentele principale ale mediului SGBD

            1) Pentru a functiona, SGBD-urile si aplicatiile au nevoie de *elemente de hardware*: un singur calculator personal, un singur calculator mainframe sau o retea de calculatoare. Unele SGBD-uri sunt dependente de configuratia hardware si de sistemul de operare, altele functioneaza pe o diversitate de elemente hardware si platforme soft.

            2) *Componenta software*cuprinde programele SGBD-ului si programele aplicatie, împreuna cu sistemul de operare si software-ul de retea, daca SGBD-ul este utilizat într-o retea de calculatoare. De obicei, programele aplicatie sunt scrise într-un limbaj de programare din generatia a treia (Pascal, C, Fortran, Cobol) sau se utilizeaza un limbaj din generatia a patra, cum ar fi SQL, încorporat într-un limbaj din generatia a treia.

3)*Datele*. Baza de date contine atât date operationale, cât si metadate ("date despre date"). Structura bazei de date este denumita **schema**.

            4) *Procedurile* se refera la instructiunile si regulile care guverneaza proiectarea si utilizarea bazei de date. De exemplu:

            · deschiderea unei sesiuni de lucru în SGBD;

            · utilizarea unei anumite facilitati SGBD sau a unui program aplicatie;

            · pornirea si oprirea SGBD-ului;

            · efectuarea de copii de siguranta ale bazei de date;

            · identificarea si tratarea defectiunilor hardware si software;

            · modificarea structurii bazei de date.

            5)*Persoanele*implicate în mediul SGBD pot fi: *administratorii de date si baze de date*, *proiectantii de baze de date*, *programatorii de aplicatii* si *utilizatorii finali*.

*Administratorul de date* este responsabil de gestionarea resurselor de date. *Administratorul de baze de date* este responsabil de realizarea fizica a bazei de date, care include proiectarea si implementarea acesteia, securitatea si controlul integritatii, întretinerea sistemului operational si asigurarea performantelor satisfacatoare pentru aplicatii si utilizatori. *Proiectantul de baze de date* se ocupa de identificarea datelor, de relatiile dintre acestea si de constrângerile asupra celor care vor fi stocate în baza de date. *Programatorii de aplicatii* implementeaza programele aplicatie ce confera functionalitatea ceruta de utilizatorii finali. *Utilizatorii finali* reprezinta pentru baza de date "clientii" - baza de date a fost proiecta, implementata si este întretinuta pentru a le satisface acestora necesitatile informationale.

***1.1.5. Avantajele si dezavantajele organizarii datelor***

***în baze de date***

O baza de date trebuie sa satisfaca cinci conditii esentiale:

**1) O buna reprezentare a realitatii înconjuratoare** prin informatii fiabile si actualizate.

**2) Controlul redundantei datelor**. Informatia trebuie sa fie unica din punct de vedere semantic si fizic. Totusi, în tratarea prin baze de date nu se elimina în întregime redundanta, ci se controleaza volumul acesteia în baza de date.

**3) Independenta datelor fata de prelucrari**. Programele de aplicatii sunt concepute în raport cu structura datelor.

**4) Securitatea si confidentialitatea datelor**. Securitatea se asigura prin procedee fizice, iar confidentialitatea prin procedee care sa împiedice accesul utilizatorilor neautorizati. Se atribuie astfel nume de utilizatori si parole care permit identificarea persoanelor autorizate sa utilizeze baza de date. Accesul la date poate fi, în plus, limitat de tipul operatiei efectuate (extragere, inserare, reactualizare, stergere).

**5) Performante în exploatare**. Orice cerere de prelucrare trebuie sa fie satisfacuta într-un timp convenabil utilizatorului.

            Se mai pot adauga numeroase alte avantaje ale utilizarii unei baze de date:

**6) Coerenta datelor** (decurge din non-redundanta). Daca un articol este stocat o singura data în baza de date, orice reactualizare a valorii sale trebuie efectuata o singura data.

**7) Mai multe informatii de la aceeasi cantitate de date**. Odata cu integrarea datelor operationale, se creeaza posibilitatea extragerii unor informatii suplimentare din aceleasi date.

**8) Costuri scazute**. Combinarea tuturor datelor operationale într-o singura baza de date si crearea unui set de aplicatii care sa functioneze pentru aceasta unica sursa de date pot avea ca rezultat micsorarea costurilor.

**9) Partajarea datelor**. Baza de date ce apartine întregii organizatii poate fi partajata de catre toti utilizatorii autorizati.

**10) Îmbunatatirea serviciilor de salvare, de siguranta si refacere**

**Dezavantaje** ale tratarii prin baze de date:

**1) Complexitatea**. Realizarea functionalitatii unui SGBD de calitate face ca acesta sa devina un element software extrem de complex.

**2) Dimensiunea**. Complexitatea si extinderea functionalitatii face ca  SGBD-urile sa constituie elemente software ce ocupa un spatiu considerabil pe disc si necesita o cantitate substantiala de memorie pentru a functiona eficient.

**3) Costul SGBD-urilor** - variaza semnificativ, functie de mediu si de functionalitatea oferita. Pretul unui SGBD cu un singur utilizator, pentru un calculator personal, difera de cel al unui SGBD mainframe, multiutilizator.

**4) Impactul crescut al unei defectiuni**. Centralizarea resurselor mareste vulnerabilitatea sistemului.

***1.2. Modele de date***

Utilitatea oricarei colectii de date în obtinerea de informatii depinde în mare masura de modul de organizare al acestor date. Posibilitatile de organizare a datelor sunt foarte variate. Prin modelarea datelor se urmareste organizarea acestora în asa fel încât sa fie îndeplinite urmatoarele doua cerinte:

· sa fie reprezentata cât mai fidel situatia reala;

· datele sa fie adaptate reprezentarii si prelucrarii pe calculator.

Modelarea datelor în vederea organizarii lor pentru a fi utilizate în aplicatii presupune identificarea celor mai importante caracteristici ale datelor, referitoare la semnificatia acestora. Generalizarea si formalizarea acestor caracteristici a condus la definirea conceptului de **model de date**. Un model de date este un instrument teoretic care permite obtinerea unei interpretari adecvate a datelor.

Se poate defini un **model de date** ca fiind o colectie integrata de concepte necesare descrierii datelor, relatiilor dintre ele si constrângerilor asupra datelor.

Baza de date este reprezentata de un ansamblu de date organizate într-o structura descrisa printr-un model conceptual (schema). Regulile de structurare a datelor (de construire a schemei) si operatiile permise asupra datelor sunt definite în cadrul modelului de date.

Orice model de date constituie un model al lumii înconjuratoare si încearca sa cuprinda proprietatile acesteia. Aceste proprietati se împart în doua clase: statice si dinamice. Proprietatile statice sunt cele care nu se modifica, deci sunt invariante în timp. Proprietatile dinamice încearca sa surprinda natura evolutiva a lumii înconjuratoare.

Un model de date este compus din doua componente principale:

· o *parte structurala*, constând dintr-un set de reguli conform carora sunt construite bazele de date - *reguli generatoare*. Aceste reguli exprima proprietatile statice ale modelului de date si sunt materializate în cadrul unui SGBD prin *limbajul de descriere a datelor* (LDD).

· o *parte de manipulare*, definind tipurile de operatii care sunt permise asupra datelor (operatii pentru reactualizarea sau regasirea datelor din baza de date si pentru modificarea structurii bazei de date). Aceste operatii exprima proprietatile dinamice ale modelului de date si sunt materializate în cadrul unui SGBD prin *limbajul de manipulare a datelor*(LMD).

Limbajul de descriere a datelor se foloseste pentru definirea structurilor de date permise în cadrul unui model de date. Structurile permise pot fi specificate în doua moduri complementare:

· prin specificarea obiectelor permise si a relatiilor permise dintre ele folosind reguli generice de definire;

· prin specificarea obiectelor si relatiilor nepermise care sunt excluse prin definirea unor restrictii numite *constrângeri*.

Unele modele partitioneaza regulile generatoare în doua parti: *reguli de specificare a structurii* si *reguli de integritate*, prin care se impun constrângeri, fiind astfel garantata corectitudinea datelor.

Din acest punct de vedere, un model ar putea fi compus din trei componente: *partea structurala*, *partea de manipulare* si *partea referitoare la regulile de integritate*.

Limbajul de manipulare a datelor cuprinde operatii care produc o schimbare în starea bazei de date, exprimata prin totalitatea valorilor datelor din baza la un moment dat. Starea unei baze de date reflecta aspectul dinamic al unui model de date prin faptul ca se schimba ca urmare a unei operatii asupra bazei de date. De remarcat faptul ca operatiile din cadrul LMD nu pot afecta structura bazei de date, deci aceste operatii conserva modelul conceptual al bazei de date asupra caruia se actioneaza.

LDD si LMD constituie principalele facilitati ale unui SGBD. Orice SGBD are la baza un anumit model de date, care constituie fundamentul teoretic al construirii acestuia, iar LDD si LMD sunt materializari ale celor doua parti componente ale modelului, reguli generatoare si operatii permise asupra datelor. Deci, în principiu, orice SGBD va putea fi caracterizat pe baza modelului de date asociat.

***1.2.1. Entitati, legaturi între entitati***

Un model conceptual cuprinde descrierea tuturor entitatilor unei baze de date împreuna cu toate legaturile existente între ele. O **entitate** este un continut de sine statator, o relatie obiectiva care exista prin ea însasi. Orice entitate este caracterizata prin proprietatile sale. În cadrul modelelor de date, aceste proprietati sunt reprezentate prin **atribute**. Entitatile, la rândul lor, sunt reprezentate prin **tipuri de entitati**. Un tip de entitate este o reprezentare în cadrul unui model de date care corespunde unei categorii de obiecte din lumea reala si constituie **intensitatea** acestei categorii. Entitatile cu proprietati comune se pot grupa (clasifica) într-o multime de entitati.

Între oricare doua multimi de entitati, M1 si M2, pot exista trei tipuri de legaturi (relatii): 1:1 (una-la-una), 1:n (una-la-mai-multe) si n:m (mai-multe-la-mai-multe).

|  |
| --- |
|  |
|  | http://www.scritub.com/files/informatica/1758_poze/image002.gif |

Într-o relatie 1:1 (una-la-una), unei entitati din multimea M1 îi corespunde o singura entitate din multimea M2 si reciproc (figura 1.5).

***Figura 1.5.*** Relatie de tip 1:1

|  |
| --- |
|  |
|  | http://www.scritub.com/files/informatica/1758_poze/image003.gif |

Într-o relatie 1:n (una-la-mai-multe), unei entitati din M1 îi corespund 0, una sau mai multe entitati din M2, dar fiecarei entitati din M2 îi corespunde o singura entitate din M1 (figura 1.6).

***Figura 1.6.***Relatie de tip 1:n

|  |
| --- |
|  |
|  | http://www.scritub.com/files/informatica/1758_poze/image004.gif |

Într-o relatie n:m (mai-multe-la-mai-multe), unei entitati din M1 îi corespund 0, una sau mai multe entitati din M2, si reciproc (figura 1.7).

***Figura 1.7.***Relatie de tip n:m

***1.2.2. Tipuri de modele***

Scopul unui model de date este sa reprezinte datele si sa le faca întelese. Daca acest lucru s-a realizat, atunci modelul poate fi utilizat cu usurinta pentru a proiecta o baza de date. Pentru a reflecta arhitectura cu trei niveluri, se pot identifica trei tipuri de modele de date, legate între ele:

1) un model de date **extern**, pentru a reprezenta vederea fiecarui utilizator;

2) un model de date **conceptual**, pentru a reprezenta vederea logica (sau generala), care este independenta de SGBD;

3) un model de date **intern**, pentru a reprezenta schema conceptuala în asa fel încât sa poata fi înteleasa de catre SGBD.

Modelele de date pot fi clasificate în trei categorii principale:

·        modele de date **bazate pe înregistrari**;

·        modele de date **bazate pe obiecte**;

·        modele de date **fizice**.

Primele doua sunt utilizate pentru a descrie datele la nivel conceptual, respectiv, extern, iar ultimul pentru descrierea datelor la nivel intern.

Într-un model bazat pe înregistrari, baza de date consta într-un numar de înregistrari cu format fix, de tipuri diferite. Fiecare tip de înregistrare defineste un numar fix de câmpuri, fiecare având, de obicei, o lungime fixa. Exista trei tipuri principale de modele de date logice bazate pe înregistrari:

· ierarhic

· retea

·relational

În modelele de date bazate pe obiecte se utilizeaza concepte cum ar fi: entitate, atribut, relatie. Cele mai utilizate tipuri de modele de date bazate pe obiecte sunt:

·        entitate-relatie (entitate-asociatie)

·        semantic

·        functional

·        orientat spre obiecte.

*1.2.3. Modelul de date relational*

**Modelul relational** a fost propus în 1970 de catre E. F. Codd. Principiile care stau la baza modelului de date relational pornesc de la teoria matematica a relatiilor, extinsa în mod logic pentru a satisface cerintele activitatii de gestionare a datelor.

Puterea si larga utilizare a modelului relational rezida din structura sa logica deosebit de simpla. El poate constitui atât un valoros instrument de studiu în teoria bazelor de date, cât si punctul de pornire pentru realizarea unor SGBD-uri competitive din punctul de vedere al performantelor.

**Sistemele de gestiune a bazelor de date relationale** (SGBDR) au devenit, în zilele noastre, elemente software dominante în prelucrarea datelor. Acestea dispun de limbaje de nivel înalt pentru manipularea datelor, simple si puternice. Caracteristica principala a acestor **limbaje relationale** este capacitatea lor de a permite definirea de relatii noi pe baza unor relatii existente.

***1.2.3.1. Terminologie***

Modelul relational se bazeaza pe conceptul matematic de **relatie**, care este reprezentat fizic sub forma de **tabel**.

            O **relatie** este un tabel format din coloane si rânduri.

            O **baza de date relationala** poate fi definita ca un ansamblu de tabele aflate în legatura.

            Utilizatorul percepe un SGBDR sub forma de tabele, dar numai la nivelul structurii logice a bazei de date, adica la nivelurile extern si conceptual.

            Un **atribut** este o coloana a unei relatii, cu o anumita denumire.

            O relatie este reprezentata printr-un tabel bidimensional în care rândurile corespund înregistrarilor individuale, iar coloanele corespund atributelor.

            De exemplu, informatiile despre filialele unei societati sunt reprezentate de catre o relatie numita Filiala, cu coloane pentru atribute, cu denumirile: NrFil (numarul filialei), Strada, Zona, Orasul, CodP(cod postal), Nr\_Tel si Nr\_Fax. Similar, informatiile despre personal sunt reprezentate prin relatia Personal, cu coloane pentru atributele NrPer (numar personal), Prenume, Nume, Adresa, Nr\_Tel, Functie, DDN (data nasterii), Salariu si NrFil (numarul filialei la care lucreaza angajatul). Figura 1.8 prezinta instantele relatiilor Filiala si Personal.

**Domeniul**reprezinta multimea de valori permise pentru unul sau mai multe atribute.

            Domeniile reprezinta o caracteristica extrem de puternica a modelului relational. Fiecare atribut dintr-o baza de date relationala este definit pe un domeniu. Domeniile pot fi diferite pentru fiecare atribut, sau doua sau mai multe atribute pot fi definite pe acelasi domeniu. De exemplu, Nr\_Tel si Nr\_Fax sunt atribute cu valori în acelasi domeniu, iar atributele Nume si Salariu au domenii diferite de valori. Tabelul 1.1 prezinta domeniile câtorva atribute.

            Conceptul de domeniu este important deoarece permite utilizatorului definirea sensului si sursei de valori pe care le poate lua atributul. Domeniul  determina operatiile ce se pot aplica unui atribut. De asemenea, în cadrul modelului relational, conceptul de domeniu este important în realizarea legaturilor semantice dintre relatii.

|  |
| --- |
|  |
|  | Text Box: CardinalitateText Box: Relatiehttp://www.scritub.com/files/informatica/1758_poze/image007.gif |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NrFil | **Strada** | **Zona** | **Orasul** | **CodP** | **Nr\_Tel** | **Nr\_Fax** |
| B5 | unu | A | Iasi | 6600 | 234890 | 234890 |
| B7 | doi | A | Iasi | 6600 | 167988 | 167988 |
| B3 | trei | B | Bacau | 6000 | 678090 | 678090 |
| B4 | patru | C | Roman | 5500 | 543210 | 543210 |
| B2 | cinci | B | Bacau | 6000 | 432543 | 432543 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NrPer** | **Prenu**  **me** | **Nume** | **Adresa** | **Nr\_**  **Tel** | **Func**  **tie** | **DDN** | **Sala**  **riu** | **NrFil** |
| SL21 | Ana | Albu | unu, 23 | 234567 | manager | 1.10.  1945 | 3000 | B5 |
| SG37 | Ion | Popa | doi, 40 | 123678 | asistent | 2.11.  1960 | 1200 | B3 |
| SG14 | Maria | Scurtu | doi, 34 | 345002 | adjunct | 4.03.  1958 | 1800 | B3 |
| SA9 | Mihai | Popescu | opt, 12 | 126700 | asistent | 9.02.  1970 | 900 | B7 |
| SG5 | Ana | Ionescu | zece, 6 | 270970 | manager | 2.06.  1940 | 2500 | B3 |
| SL41 | Dan | Popa | unu, 57 | 246001 | asistent | 3.06.  1965 | 900 | B5 |

***Figura 1.8.***Instante ale relatiilor Filiala si Personal

            Doua domenii sunt **compatibile** daca ele sunt comparabile din punct de vedere semantic. Conform acestei definitii, doua domenii identice sau aflate în relatie de incluziune sunt compatibile.

            Orice operatie de cuplare a doua relatii sau, mai general, de comparare a unor valori din doua domenii are sens numai daca cele doua domenii sunt compatibile. De exemplu, nu are nici un sens sa se compare numele unei persoane cu un nume de oras.

            Unele SGBD-uri dispun de facilitati ale LDD prin care proiectantul poate specifica în mod explicit daca are sau nu semnificatie comparatia valorilor din doua domenii date. Prin declararea explicita a compatibilitatii dintre domenii, utilizatorul va fi împiedicat sa efectueze anumite operatii considerate ca fiind fara sens în raport cu semnificatia atribuita datelor.

***Tabelul 1.1.*** Domeniile unor atribute ale relatiilor Filiala si

                     Personal

|  |  |  |  |
| --- | --- | --- | --- |
| Atribut | **Denumire domeniu** | **Sensul** | **Domeniul de definitie** |
| NrFil | Numere\_Filiale | Multimea tuturor numerelor de filiala posibile | caracter: dimensiune 3, cuprindere B1-B99. |
| DDN | Date\_nastere | Valorile posibile ale datelor de nastere ale personalului | data, cuprindere de la 1.01.40 la 31.12.80, format zz.ll.aa |
| Salariu | Salarii | Valorile posibile ale salariilor personalului | monetar: 7 caractere, cuprindere 500 - 10000 |
| Nr\_Tel | Numere\_Telefax | Multimea numerelor de telefon si fax | caracter: dimensiune 13 |
| Nr\_Fax | Numere\_Telefax | Multimea numerelor de telefon si fax | caracter: dimensiune 13 |

**Tuplul** este un rând dintr-o relatie. Un tuplu în relatia Personal contine informatii despre un angajat.

Structura unei relatii, împreuna cu specificarea domeniilor si restrictiilor asupra valorilor posibile se numeste **intensitatea** relatiei, care este, de obicei, fixata. Tuplurile sunt denumite **extensia**(sau **starea**) unei relatii, care se modifica în timp.

**Gradul** unei relatii reprezinta numarul de atribute pe care le contine aceasta.

De exemplu, relatia Filiala are 7 atribute sau gradul 7. Aceasta înseamna ca fiecare rând din tabel este un hepta-tuplu, continând 7 valori. O relatie cu un singur atribut are gradul întâi si este denumita relatie **unara** sau un-tuplu. O relatie cu doua atribute este denumita **binara**, cu trei atribute - **ternara**, dupa care se utilizeaza, de regula, termenul de **n-ara**. Gradul unei relatii este o parte din intensitatea acesteia.

**Cardinalitatea** unei relatii reprezinta numarul de tupluri continute de aceasta.

Cardinalitatea relatiei se modifica pe masura ce sunt adaugate sau sterse tupluri. Este deci o proprietate a extensiei unei relatii si este determinata de instanta particulara a relatiei respective.

O alternativa la terminologia anterioara se bazeaza pe faptul ca, din punct de vedere fizic, SGBDR-ul poate stoca fiecare relatie într-un fisier. Astfel, o relatie poate fi denumita **fisier**, tuplurile sunt **înregistrari**, iar atributele **câmpuri**. Tabelul 1.2 prezinta alternative la terminologia de baza.

***Tabelul 1.2.*** Alternative la terminologia de baza

|  |  |  |
| --- | --- | --- |
| Termeni formali | **Alternativa 1** | **Alternativa 2** |
| relatie | tabel | fisier |
| tuplu | rând | înregistrare |
| atribut | coloana | câmp |

***1.2.3.2. Proprietatile relatiilor***

Orice relatie este o **multime** ale carei elemente sunt n-tupluri. Aceasta înseamna ca o relatie are o serie de proprietati care deriva din caracterul de multime al acesteia. Doua proprietati sunt deosebit de importante:

· O prima proprietate este aceea ca multimile sunt colectii de elemente fara repetitie. În contextul relatiilor, aceasta înseamna ca oricare doua tupluri difera prin cel putin o valoare a elementelor lor. Aceasta proprietate sta la baza definirii conceptului de cheie a unei relatii.

· A doua proprietate se refera la faptul ca ordinea elementelor într-o multime este nerelevanta. În consecinta, ordinea de însiruire a tuplurilor într-o relatie este arbitrara, iar prin schimbarea ordinii tuplurilor relatia ramâne neschimbata. Deci oricând este posibila reordonarea convenabila a tuplurilor din cadrul unei relatii.

Adaugând si câteva alte proprietati, se pot enumera urmatoarele caracteristici ale unei relatii:

·                    are o denumire distincta;

·                    fiecare celula a relatiei contine o valoare atomica (singulara);

·                    fiecare atribut are o valoare distincta;

·                    toate valorile unui atribut apartin aceluiasi domeniu;

·                    ordinea atributelor nu are importanta;

·                    fiecare tuplu este distinct (nu exista dubluri);

·                    teoretic, ordinea tuplurilor nu are importanta (totusi, în practica, ordinea poate afecta eficienta accesarii tuplurilor).

Despre o relatie care nu contine grupuri repetitive, ci o singura valoare într-o celula,  se spune ca este **normalizata** sau este în **prima forma normala**.

*1.2.3.3. Chei relationale*

S-a mentionat faptul ca tuplurile unei relatii sunt unice, ceea ce înseamna ca fiecare tuplu poate fi identificat în mod unic prin valorile atributelor sale. De obicei, pentru identificarea unica a unui tuplu nu sunt necesare valorile tuturor componentelor sale, ci sunt suficiente doar valorile unui subset al atributelor relatiei.

 O **cheie candidat** este un atribut sau un set de atribute care identifica în mod unic un tuplu din interiorul unei relatii.

Pentru o relatie pot exista mai multe chei candidat. O astfel de cheie poate fi **simpla** daca este constituita dintr-un singur atribut sau **compusa** când contine mai multe atribute. În relatia Filiala, atributul Orasul nu poate fi selectat drept cheie candidat deoarece, pentru o valoare a acestui atribut, se pot gasi mai multe filiale. Atributul NrFil însa, poate determina cel mult un tuplu (fiecare filiala are un numar propriu), astfel încât atributul NrFil constituie o cheie candidat. În mod similar, atributele Nr\_Tel si Nr\_Fax constituie, de asemenea, chei candidat pentru aceasta relatie.

În relatia Personal, un exemplu de cheie candidat compusa poate fi reprezentata de atributele Nume, Prenume si Adresa.

În orice relatie exista cu certitudine o cheie, în cel mai defavorabil caz cheia fiind întregul tuplu. În consecinta, problema gasirii unei chei se reduce la determinarea setului minimal de atribute care satisface proprietatea de identificare unica.

Orice atribut al unei relatii care face parte din cel putin o cheie se numeste **atribut prim**. Toate celelalte atribute ale relatiei sunt **neprime**.

Într-o relatie pot exista mai multe chei candidat. Pentru fiecare relatie din cadrul unei baze de date se desemneaza, dintre acestea, dupa anumite criterii, o cheie privilegiata numita cheie primara.

O **cheie primara** este o cheie candidat care este selectata pentru a identifica în mod unic tuplurile din cadrul unei relatii.

Atunci când o cheie candidat nu este aleasa drept cheie primara este considerata **cheie alternativa**(**secundara**).

Pentru relatia Filiala, daca se alege NrFil drept cheie primara, Nr\_Tel si Nr\_Fax constituie chei alternative (figura 1.8).

Din moment ce o relatie nu are tupluri duble, este întotdeauna posibila identificarea unica a fiecarui rând. Aceasta înseamna ca o relatie are întotdeauna o cheie primara.

Distinctia dintre o cheie primara si celelalte chei candidate este importanta numai din punct de vedere operational, cheia primara având un rol important în implementarea strategiilor de cautare si regasire a datelor. Cheia primara este aceea dintre cheile unei relatii care este folosita de SGBD în identificarea unica a tuplurilor. Statutul de cheie primara al unei chei candidat este stabilit de utilizator si comunicat SGBD-ului prin intermediul LDD.

Asupra cheii primare, SGBD-ul impune o serie de restrictii:

· nu sunt admise valori nedefinite pentru atributele unei chei primare. Orice alta cheie a unei relatii poate avea valori nedefinite pentru unele dintre atributele sale.

· nici o valoare a unui atribut dintr-o cheie primara nu poate fi modificata în cadrul operatiilor de actualizare.

La selectarea unei chei primare din multimea cheilor candidate se va tine seama de necesitatea ca numarul atributelor cheii primare sa fie cât mai mic posibil, deci va fi desemnata drept cheie primara cheia candidat care are definite toate valorile atributelor sale si care are cel mai mic numar de atribute.

O **cheie straina** este un atribut sau un grup de atribute din cadrul unei relatii care se potrivesc cu cele din cheia candidat a unei alte relatii.

Prezenta unui atribut în mai multe relatii are drept scop constituirea unei legaturi între tuplurile celor doua relatii. De exemplu, includerea atributului NrFil în relatiile Filiala si Personal este destinata legarii filialei de detaliile privind personalul din fiecare filiala. În relatia Personal, atributul NrFil constituie o cheie straina. Se spune ca atributul cheie straina NrFil din relatia Personal *tinteste* atributul cheie primara NrFildin **relatia de baza** Filiala (figura 1.8).

Legatura între tuplurile din relatii diferite se realizeaza prin atribute sau combinatii de atribute numite **chei straine (externe)**. Cheile straine (coloanele de referinta) sunt, asadar, atribute sau combinatii de atribute care pun în legatura linii (tupluri) din relatii diferite.

*1.2.3.4. Reprezentarea schemelor din bazele de date*

Modelul relational foloseste o singura forma de structurare a datelor si anume relatiile. Într-o baza de date relationala toate datele sunt reprezentate prin relatii.

**Intensitatea** unei baze de date relationale este specificata printr-o **schema relationala** care consta din una sau mai multe **scheme de relatie**.

**Schema de relatie** cuprinde numele relatiei si atributele acesteia .

Conventia uzuala de reprezentare a schemei de relatie este de a da numele relatiei, urmat de denumirile atributelor între paranteze. Cheia primara este subliniata.

Filiala (NrFil, Strada, Zona, Orasul, CodP, Nr\_Tel, Nr\_Fax)

Personal (NrPer, Prenume, Nume, Adresa, Nr\_Tel, Functie, DDN, Salariu, NrFil)

Relatia Filala reprezinta orice multime de hepta-tupluri. Un exemplu de hepta-tuplu este urmatorul:

.

Din exemplele prezentate, se poate observa ca o schema de relatie poate fi folosita în mod direct pentru reprezentarea unui tip de entitate (Filiala, Personal).

*Modelul conceptual* sau *schema conceptuala* reprezinta multimea tuturor schemelor de acest tip pentru baza de date. Deci schema relationala este modelul conceptual al bazei de date relationale. Cu alte cuvinte, definirea modelului conceptual pentru o baza de date relationala presupune specificarea fiecarei scheme de relatie din cadrul schemei relationale. Acest lucru se realizeaza prin folosirea facilitatilor limbajului de definire a datelor (LDD). În majoritatea SGBD-urilor relationale, LDD este integrat în modulul software care implementeaza LMD. Prin integrarea LMD si LDD va exista un singur modul pentru implementarea acestora, cu o interfata utilizator unitara pentru ambele tipuri de functiuni.

***1.2.3.5. Reprezentarea legaturilor în modelele relationale***

Pentru reprezentarea legaturilor dintre tipurile de entitati în modelele relationale, se utilizeaza doua tehnici:

· **propagarea cheilor** dintr-o schema de relatie într-alta.

Aceasta tehnica se poate folosi pentru reprezentarea legaturilor de tip 1:1 sau 1:n (legaturi de tip functional). De exemplu, legatura functionala dintre tipurile de entitati Filiala si Personal se poate realiza si reprezenta prin adaugarea atributului NrFil în schema de relatie Personal. Procedeul poarta numele de propagare a cheii, deoarece cheia NrFil din schema de relatie Filiala a fost plasata în schema de relatie Personal. Atributul NrFil este cheie straina în schema de relatie Personal si cheie primara în schema de relatie Filiala.

· **crearea unei scheme de relatie separate** prin care se reprezinta legatura dintre doua tipuri de entitati.

Aceasta tehnica se poate utiliza pentru a reprezenta orice tip de legatura, inclusiv legaturi m:n.

Pentru a exemplifica aceasta metoda, se considera relatiile:

Facultate (Codf, Nume, Adresa)

Personal (Codp, Nume, Functie, Salar, Codf)

Profesor (Codf, Nume, Functie, Disciplina)

Student (Nume, Incadrare, Sit\_scolara, An, Sex)

Note (Nume\_profesor, Nume\_student, Nota)

Pentru a reprezenta legatura de tip n:m dintre tipurile de entitati Profesor si Student, se foloseste o schema de relatie separata, Note, care are ca atribute cheile celor doua tipuri de entitati, respectiv atributele Nume redenumite Nume\_profesor si Nume\_student. Pe lânga cele doua chei mai pot sa apara si alte atribute ce caracterizeaza în mod natural legatura între cele doua tipuri de entitati considerate. De exemplu, Nota.

*1.2.3.6. Integritate relationala*

Pâna acum a fost analizata *partea structurala* a modelului de date relational. Un model de date mai are înca doua parti: o *parte de manipulare* care defineste tipurile de operatii permise asupra datelor si un *set de reguli* *de integritate* care asigura corectitudinea datelor.

Din moment ce fiecare atribut are un domeniu asociat, exista anumite constrângeri (denumite **constrângeri de domeniu**) sub forma de restrictii impuse multimii de valori permise pentru atributele relatiilor. Pe lânga acestea, exista doua **reguli de integritate** importante, care reprezinta constrângeri sau restrictii ce se aplica tuturor instantelor unei baze de date. Pentru modelul relational, cele doua reguli principale sunt cunoscute sub denumirile de **integritate a entitatilor** si **integritate referentiala**.

Un **null**reprezinta valoarea unui atribut care este în mod curent necunoscuta sau nu este aplicabila tuplului respectiv.

Un null poate fi luat cu sensul de valoare logica "necunoscuta". Poate însemna ca o valoare nu este aplicabila unui anumit tuplu sau poate însemna doar ca, deocamdata, nu s-a atribuit nici o valoare. Null-urile reprezinta o modalitate de tratare a datelor incomplete sau deosebite. Totusi, un null nu este acelasi lucru cu o valoare numerica egala cu zero sau cu un sir de text completat cu spatii; zerourile si spatiile sunt valori, pe când un null semnifica absenta unei valori. De exemplu, se înregistreaza în baza de date achizitia unui produs, dar nu se trece, deocamdata, numarul facturii, deoarece aceasta nu a sosit (marfa a fost însotita de aviz de expeditie). Valoarea considerata pentru un atribut Nr\_factura va fi null.

Prima regula de integritate se aplica cheilor primare ale relatiilor de baza. O relatie de baza corespunde unei entitati în schema conceptuala.

**Integritatea entitatilor** consta în faptul ca într-o relatie de baza, nici un atribut al unei chei primare nu poate fi null. Deci valorile cheilor primare trebuie sa fie *unice* si *nenule*.

Prin definitie, o cheie primara este un identificator minim, utilizat pentru identificarea unica a tuplurilor. Admitând un null pentru orice parte a unei chei primare, aceasta implica faptul ca nu toate atributele sunt necesare pentru a deosebi tuplurile, ceea ce contrazice definitia cheii primare.

A doua regula de integritate se aplica cheilor straine.

**Integritatea referentiala** prevede faptul ca valoarea unei chei straine trebuie ori sa coincida cu valoarea unei chei candidat a unui tuplu în relatia sa de baza, ori sa fie în întregime null.

De exemplu, atributul NrFil din relatia Personal este o cheie straina care tinteste atributul NrFil din relatia de baza Filiala. Nu trebuie sa fie posibila crearea unei înregistrari de personal cu numarul de filiala B25, de exemplu, decât daca exista deja o înregistrare corespunzatoare numarului B25 în relatia Filiala(daca exista aceasta filiala). Totusi, trebuie sa existe posibilitatea de a crea o noua înregistrare de personal cu un numar de filiala null. Aceasta corespunde situatiei în care un nou membru al personalului a fost admis în companie, fara a fi repartizat, deocamdata, la o anumita filiala.

În concluzie, orice SGBD relational trebuie sa realizeze urmatoarele trei principii:

· **principiul integritatii domeniului**

· **principiul integritatii entitatii**(sau**al relatiei**)

· **principiul integritatii referintei**.

**Constrângerile de întreprindere** sunt reguli aditionale specificate de catre utilizatorii sau administratorii unei baze de date. De exemplu, se stabileste un numar maxim de 20 de membri ai personalului pentru o filiala.

Un SGBD este considerat **total relational** daca:

· asigura cele trei principii de integritate

· furnizeaza utilizatorului un LMD cel putin echivalent ca putere de expresie cu algebra relationala.

***1.2.3.7. Integritate referentiala***

În cadrul modelului de date relational, legaturile dintre relatiile ce reprezinta tipuri de entitati se realizeaza prin mecanismul de propagare a cheilor, ceea ce conduce la conceptul de **cheie straina**. O cheie straina poate fi rezultatul propagarii oricarei chei, cheie primara sau cheie candidat.

Valoarea unei chei straine reprezinta o **referinta** la tuplul a carui cheie are o valoare identica cu cea a cheii straine. Relatia care contine cheia straina poarta numele de **relatie de referinta**, iar relatia care contine cheia din care s-a propagat cheia straina se numeste **relatie referita**. Conditia de **integritate referentiala**cere ca toate valorile unei chei straine din relatia de referinta sa se regaseasca printre valorile cheii corespunzatoare (care s-a propagat) din relatia referita. Sensul propagarii este de la relatia referita la relatia de referinta; cheia straina prezenta în relatia de referinta provine din relatia referita (**relatia de baza**). Conditia de integritate referentiala introduce în baza de date niste constrângeri între relatii numite **constrângeri referentiale**. Figura 1.9 ilustreaza grafic principiul integritatii referentiale.

|  |
| --- |
|  |
|  | http://www.scritub.com/files/informatica/1758_poze/image008.gif |

***Figura 1.9.*** Principiul integritatii referentiale

Fie o baza de date referitoare la activitatea comerciala a unor firme, reprezentata prin urmatoarea schema relationala:

Furnizor (CodF, Nume, Oras)

Beneficiar (CodB, Nume, Oras)

Produs (CodP, Nume, UM)

Oferte (CodF, CodP, Pret, Cantitate)

Cereri (CodB, CodP, Pret, Cantitate)

Tranzactii (CodT, CodF, CodB, CodP, Pret, Cantitate)

Atributele CodF, CodB si CodP sunt chei primare ale relatiilor Furnizor, Beneficiar si, respectiv, Produs. Atributele CodF si CodP din relatia Oferte sunt chei straine si reprezinta referinte catre atributele corespunzatoare din relatiile Furnizor si Produs. Conditia de integritate referentiala impune ca valorile atributelor CodF si CodP din Oferte (relatia de referinta) sa se regaseasca printre valorile acestor atribute din relatiile referite (Furnizor si Produs). Cu alte cuvinte, conditia de integritate referentiala exprima faptul ca nu este posibila existenta unei oferte daca nu exista un furnizor care sa faca aceasta oferta si un produs care sa fie oferit de acesta.

Atributele CodB si CodP sunt chei straine în relatia Cereri. În acest caz, integritatea referentiala arata ca nu poate exista o cerere fara beneficiarul ei si fara produsul solicitat.

În relatia Tranzactii, exista trei chei straine - CodF, CodB, CodP - care sunt referinte la relatiile Furnizor, Beneficiar si Produs, ceea ce exprima faptul ca orice tranzactie implica un furnizor, un beneficiar si un produs care sa constituie obiectul tranzactiei.

***1.2.3.8. Operatii de adaugare, stergere si actualizare***

Operatiile de adaugare, stergere si actualizare pot afecta integritatea referentiala a unei baze de date. Evident, este necesar ca aceasta sa fie satisfacuta permanent si sa fie mentinuta. Necesitatea satisfacerii si mentinerii conditiilor de integritate referentiala are o serie de consecinte asupra modului în care trebuie realizate operatiile de adaugare, stergere si actualizare. Astfel de consecinte se manifesta atât în cazul relatiilor de referinta, cât si în cazul relatiilor referite.

· **Relatia de referinta**(relatia care contine cheia straina)

**Operatia de adaugare** a unui tuplu în relatia de referinta presupune faptul ca valorile atributelor care fac parte din cheia straina se regasesc printre valorile atributelor corespunzatoare din relatia referita (relatia de la care s-a realizat propagarea cheii straine). Daca aceasta conditie - care sta la baza definitiei cheii straine - nu este satisfacuta, atunci operatia de adaugare nu poate fi executata.

De exemplu, la relatia Cereri nu se poate adauga un tuplu ce reprezinta o noua cerere daca în relatiile Beneficiar si Produs nu exista beneficiarul care solicita cererea, respectiv produsul solicitat.

**Operatia de stergere** a unui tuplu în relatia de referinta se poate face fara nici un fel de restrictii din punctul de vedere al conditiilor de integritate referentiala.

**Operatia de actualizare** poate fi considerata ca fiind formata din doua componente, realizate succesiv: stergere si adaugare. Regulile de mentinere a integritatii referentiale în acest caz sunt rezultatul combinarii regulilor prezentate mai sus.

· **Relatia referita** (relatia de baza de la care s-a propagat cheia straina)

**Operatia de adaugare** a unui tuplu în relatia referita se face fara nici un fel de restrictii din punctul de vedere al constrângerilor de integritate referentiala.

**Operatia de stergere** a unui tuplu din relatia referita trebuie realizata cu precautie, deoarece este posibil ca în relatia de referinta sa existe tupluri care fac referire la tuplul care se sterge. În acest caz, pentru a mentine integritatea referentiala a bazei de date, este posibila una din urmatoarele optiuni: **stergerea cu restrictii**sau **stergerea în cascada**.

**stergerea cu restrictii** presupune faptul ca nu se accepta stergerea unui tuplu din relatia referita daca acesta este referit de cel putin un tuplu din relatia de referinta.

De exemplu, nu se poate sterge un beneficiar din relatia Beneficiar atâta timp cât el are cel putin o cerere în relatia Cereri. Tot astfel, nu se poate sterge un furnizor din relatia Furnizor atâta timp cât el are cel putin o oferta în relatia Oferte.

**stergerea în cascada** a unui tuplu dintr-o relatie referita va fi urmata de stergerea tuturor tuplurilor din relatiile de referinta care fac referire la tuplul sters.

De exemplu, daca se sterge un furnizor din relatia Furnizor, atunci se vor sterge toate ofertele acestuia din relatia Oferte, precum si tranzactiile sale din relatia Tranzactii.

**Operatia de actualizare** a unei chei dintr-o relatie referita se poate face în doua moduri:

· **actualizare cu restrictii** - nu se accepta modificarea valorii unui atribut din relatia referita daca exista cel putin un tuplu în relatiile de referinta care face referire la aceasta valoare.

De exemplu, nu poate fi modificat codul unui furnizor din relatia Furnizor deoarece acest cod apare si în relatiile Oferte si Tranzactii.

· **actualizare în cascada** - modificarea valorii unui atribut al unei chei dintr-o relatie referita va fi urmata de modificarea corespunzatoare în cheia straina a relatiei de referinta.

Modificarea valorii unui cod al unui furnizor în relatia Furnizor va determina aceeasi modificare a codului ce apare în relatiile Oferte si Tranzactii.