Aspecte opţionale ale securităţii în bazele de date

Primele versiuni ale sistemelor de gestiune a bazelor de date comerciale au furnizat acest tip de securitate (*discretionary*).

Politicile de securitate constituie un set de reguli care asigură securitatea sistemului. Aceste politici includ 2 categorii :

- politicile obligatorii (*mandatory*) – prezentate anterior ;
- politicile opţionale (*discretionary*).

Reamintim că politicile de tip *mandatory* sunt cele care sunt obligatorii prin natura lor şi independente de aplicaţie.

Politicile de tip *discretionary* sunt cele care sunt specificate de către administrator sau de către o persoană responsabilă asupra mediului în care va opera sistemul respectiv.

Cea mai cunoscută politică de securitate opţională este cea referitoare la controlul accesului. Aceste politici au fost studiate pentru sistemele de operare încă din anii ‘60, iar pentru bazele de date încă din anii ‘70. Cele mai reprezentative sisteme de baze de date, *System R* şi *INGRES* au fost printre primele care au investigat controlul accesului pentru sistemele de baze de date. De atunci, au avut loc modificări determinate de evoluțiile acestor politici.

O altă clasă de politici opţionale de securitate include politicile de administrare.

De asemenea, identificarea şi autentificarea pot fi aduse în discuție în cadrul politicilor opţionale.

Ca şi în prezentarea politicilor obligatorii, introducerea politicilor opţionale se concentrează asupra sistemelor relaţionale, însă majoritatea principiilor sunt aplicabile şi altor sisteme, cum ar fi sistemele de baze de date orientate pe obiecte şi cele distribuite.

1. Politici de control al accesului

Aşa cum am menţionat anterior, aceste politici au fost investigate mai întâi pentru sistemele de operare. Problema esenţială în cadrul sistemelor de operare este dacă unui proces i

se poate acorda accesul la un fişier. Accesul poate fi de citire sau de scriere, iar acesta din urmă poate include accesul pentru modificare, adăugare sau ştergere. Ulterior, aceste principii au fost transferate asupra sistemelor de baze de date, cum ar fi *INGRES* şi *System R*. De atunci, au fost studiate diferite forme ale controlului accesului. Printre acestea, se pot menţiona politicile bazate pe *role*-uri implementate în prezent în mai multe sisteme comerciale.

Politicile de control al accesului includ şi aspecte obligatorii, discutate în cursul anterior.

Diferitele tipuri de control al accesului sunt prezentate în figura 1 şi vor fi descrise în continuare.

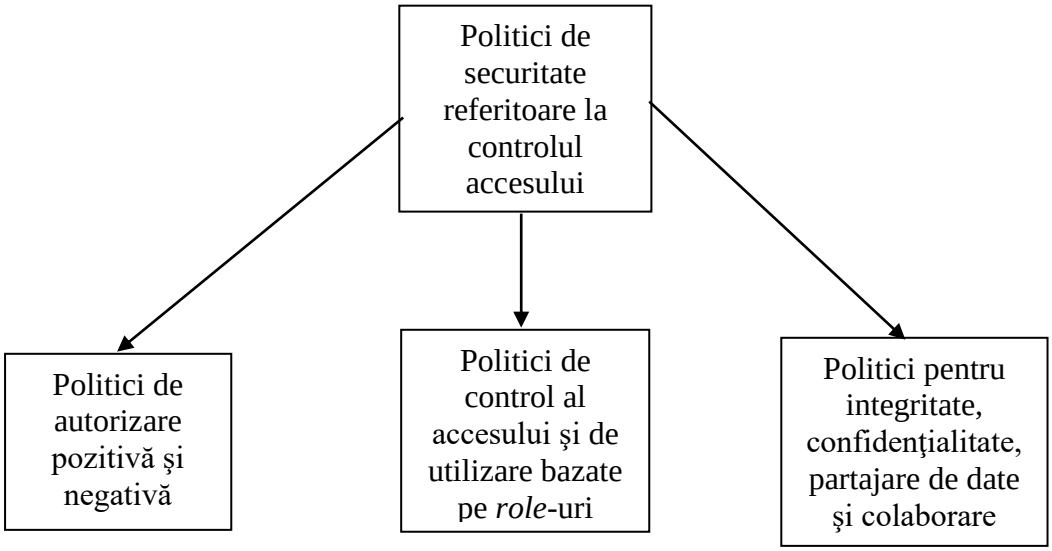

Figura 1. Tipuri de control al accesului

1.1 Politici de autorizare

Majoritatea politicilor de control al accesului se bazează pe politici de autorizare. Acestea presupun faptul că utilizatorilor le este acordat accesul la date pe baza regulilor de autorizare, care pot fi de mai multe tipuri.

- ***Autorizările pozitive.*** Primele sisteme s-au concentrat asupra regulilor care acum sunt denumite reguli de autorizare pozitivă.

  De exemplu, utilizatorului John îi este acordat accesul la relaţia *EMP* sau utilizatorului Jane îi este acordat accesul la relaţia *DEPT*. Acestea sunt reguli de control al accesului asupra relaţiilor. De asemenea, se poate acorda acces asupra atributelor şi tuplurilor. De exemplu, John are acces pentru citirea atributului *salary* şi acces pentru scrierea atributului *name* din relaţia *EMP*.

- ***Autorizările negative.*** Dacă accesul lui John asupra unui obiect nu este specificat, aceasta înseamnă că John nu are acces la acel obiect? În anumite sisteme, nespecificarea unei reguli de autorizare presupune implicit autorizarea negativă, în timp ce în alte sisteme autorizările negative sunt specificate în mod explicit. De exemplu, putem impune reguli astfel încât John să nu aibă acces la relaţia *EMP* sau Jane să nu aibă acces la relaţia *DEPT*.

- ***Rezolvarea conflictelor.*** Atunci când avem reguli în conflict, cum se rezolvă acesta? De exemplu, putem avea o regulă care acordă accesul lui John pentru citire la relaţia *EMP*. De asemenea, putem avea şi o regulă care nu îi acordă lui John accesul la citire asupra atributului *salary* din *EMP*. Acesta este un conflict. De obicei, sistemul aplică regula celui mai mic privilegiu, adică John are acces la EMP mai puţin valorile atributului *salary*.

- ***Autorizare tare şi slabă.*** În cazul autorizării tari regula este valabilă indiferent de conflicte, iar în cazul celei slabe regula nu se aplică dacă există un conflict. De exemplu, dacă John primeşte acces la *EMP* şi aceasta este o regulă tare de autorizare, iar regula prin care John nu primeşte acces la atributul *salary* este o regulă slabă, avem o situaţie conflictuală. Aceasta înseamnă că rămâne valabilă autorizarea tare.

- ***Propagarea regulilor de autorizare.*** De exemplu, dacă John are acces pentru citire asupra relaţiei *EMP*, aceasta înseamnă că John are acces la fiecare element din *EMP*? De obicei, acest lucru are loc cu excepţia cazului în care există reguli care interzic propagarea automată a unei reguli de autorizare. Dacă o astfel de regulă există, trebuie să stabilim explicit reguli de autorizare care specifică obiectele asupra cărora are acces John.

- ***Reguli speciale.*** Constrângerile bazate pe conţinut şi context sunt reguli prin care accesul este acordat în funcţie de conţinutul datelor sau de contextul în care acestea sunt afişate. Aceste reguli constituie extensii ale politicilor obligatorii, însă pot fi aplicate şi în cadrul securităţii opţionale. De exemplu, în cazul constrângerilor bazate pe conţinut, John are acces pentru citire doar asupra tuplurilor din departamentul 100. În cazul constrângerilor bazate pe context sau asociere, John nu are acces pentru citire la nume şi salarii luate

împreună, dar poate avea acces individual la acestea. În cazul constrângerilor bazate pe evenimente, după alegere, John are acces la toate elementele din relaţia *EMP*.

- ***Consistenţa şi completitudinea regulilor.*** Una dintre probleme este asigurarea consistenţei şi completitudinii constrângerilor. Aceasta înseamnă că, dacă regulile sau constrângerile sunt inconsistente, avem reguli de rezolvare a conflictelor care să trateze situaţia? Cum putem asigura că toate entităţile (atribute, relaţii, elemente etc.) sunt cuprinse în cadrul regulilor de control al accesului pentru un utilizator? Cu alte cuvinte, sunt regulile complete? Dacă nu, ce presupuneri trebuie să facem despre entităţile care nu au nici autorizări pozitive, nici negative specificate asupra lor pentru un anumit utilizator sau pentru o clasă de utilizatori.

1.2 Controlul accesului bazat pe *role*-uri

Role-Based Access Control (RBAC) a devenit una dintre cele mai populare metode de control al accesului. Metoda a fost implementată în sistemele comerciale, inclusiv *Oracle*. Ideea este aceea de a acorda acces utilizatorilor pe baza *role*-urilor şi funcţiilor lor.

Utilizatorii au nevoie de acces la date în funcţie de *role*-urile lor. De exemplu, un preşedinte de corporaţie poate avea acces la informaţiile despre vicepreşedinţii săi şi despre membrii consiliului, iar directorul financiar poate avea acces la informaţiile financiare şi la cele referitoare la angajaţii din subordinea sa. Un director de departament poate avea acces la datele despre cei care lucrează în departamentul respectiv iar directorul de resurse umane poate obţine informaţii asupra datelor personale ale angajaţilor din corporaţie.

Controlul accesului pe bază de *role*-uri este un tip de politică de autorizare care depinde de rolul (funcția) utilizatorului şi activităţile corespunzătoare acestuia.

Literatura conţine direcţiile de cercetare asupra ierarhiilor de *role*-uri . Unele dintre problemele care apar în acest domeniu se referă la modul în care este propagat accesul sau la posibilitatea ca un *role* să conţină pe altul.

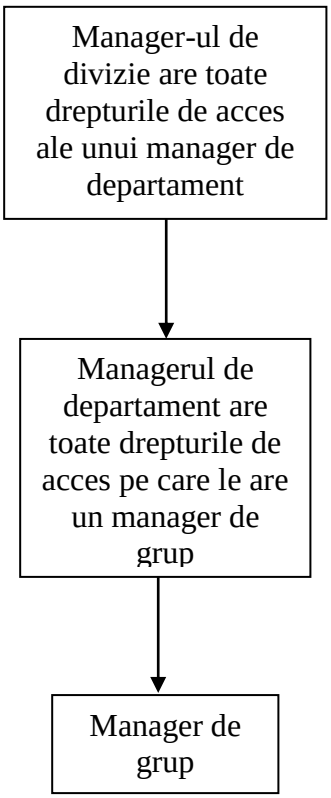

Figura 2. Ierarhie de *role*-uri

Considerăm ierarhia de *role*-uri din figura 2. Dacă acordăm acces unui nod din ierarhie, aceasta înseamnă că accesul este propagat în sus? Dacă un manager de departament are acces la anumite informaţii despre proiecte, accesul este propagat în nodul părinte, care corespunde unui director? Dacă un conducător de secţie are acces la informaţiile angajaţilor din secţia sa, accesul se propagă către managerul de departament (parintele din ierarhia de *role*-uri)? Ce se întâmplă cu nodurile copil? Accesul se propagă în jos? De exemplu, dacă un manager de departament are acces la anumite informaţii, au şi subordonaţii săi acces la informaţiile respective? Există cazuri în care subordonaţii au acces la date la care managerul de departament nu are acces? Ce se întâmplă dacă un angajat trebuie să raporteze la 2 supervizori (managerul de departament şi şeful său de proiect)? Ce se întâmplă când managerul de departament lucrează pe un proiect şi trebuie sa raporteze şefului său de proiect, care este condus de el? Părinţii multipli sunt ilustraţi în figura 3 iar un ciclu este reprezentat în figura 4.

Accesul pe bază de *role*-uri a fost analizat pentru sistemele relaţionale, obiect şi distribuite, dar şi pentru tehnologiile mai recente (*data warehouse*, sistemele de gestiune a cunoştinţelor, *web* semantic, sistemele de comerţ electronic, bibliotecile digitale).

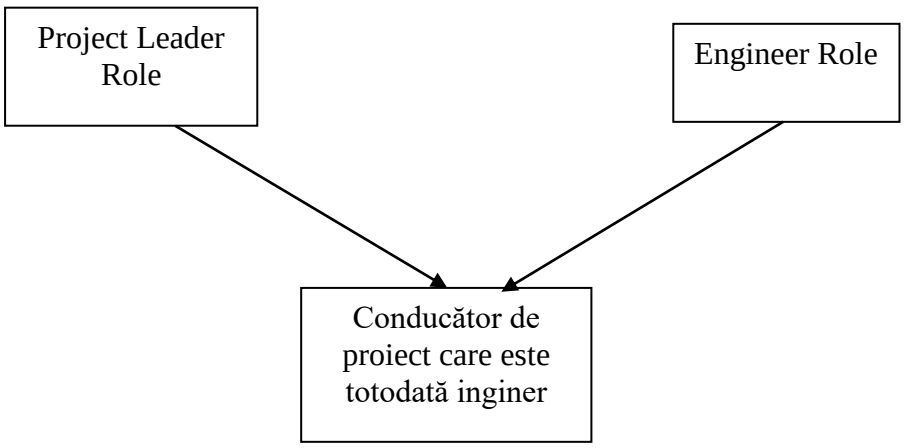

Figura 3. Părinţi multipli

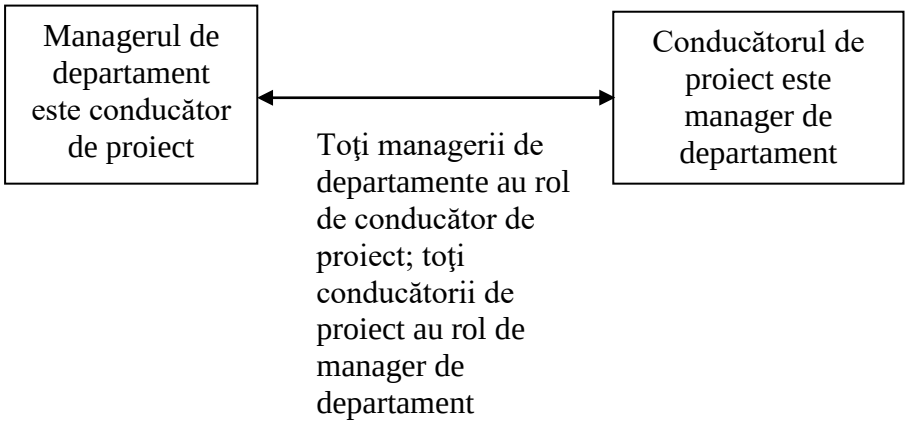

Figura 4. Graf ciclic

1.3 Politici de administrare

Politicile de control al accesului specifică accesul pe care utilizatorii îl au asupra datelor, iar politicile de administrare specifică cine va administra datele. Sarcinile de administrare includ menţinerea datelor într-o stare coerentă, asigurând că metadatele sunt actualizate odată cu modificarea datelor, şi asigurând revenirea în urma căderilor.

Administratorul bazei de date (DBA) este responsabil de actualizarea metadatelor, a indecşilor şi a metodelor de acces şi, de asemenea, de asigurarea faptului că regulile de control al accesului sunt aplicate corespunzător. Un rol important îl poate avea şi responsabilul cu securitatea sistemului (SSO – *System Security Officer*). Problemele legate de securitate pot fi responsabilitatea SSO iar cele referitoare la date pot fi responsabilitatea DBA. Alte politici de administrare se referă la numirea de responsabili asupra datelor. De obicei, posesorii schemelor

au control asupra datelor pe care le creează şi le pot gestiona pe perioada existenţei acestora. Există situaţii în care posesorii nu pot fi disponibili pentru gestiunea datelor, caz în care se recurge la numirea unor respnsabili asupra acestora.

Figura 5 arată principalele politici de administrare.

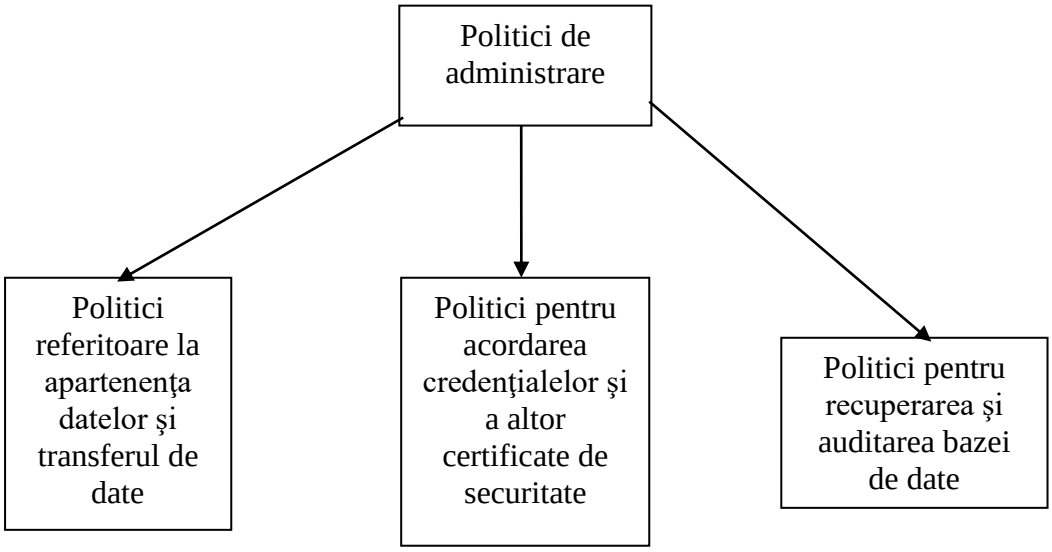

Figura 5. Politici de administrare

1.4 Identificare şi autentificare

Identificarea presupune necesitatea ca utilizatorii să funizeze un *user ID* şi o parolă. Autentificarea reprezintă etapa în care sistemul trebuie să potrivească *user ID*-ul cu parola pentru a asigura că persoana este cea care declară că este. Un utilizator poate avea identităţi multiple, în funcţie de *role*-urile sale.

Au fost raportate numeroase probleme referitoare la schema pe bază de parole. Una dintre acestea este că *hacker*-ii pot pătrunde în sistem, obţine parolele utilizatorilor pentru ca apoi să pretindă că sunt aceştia. Într-un sistem centralizat problemele sunt mai simple decât într-unul distribuit.

Recent, au început să fie utilizate tehnicile biometrice. Acestea includ recunoaşterea feţei şi a vocii pentru autentificarea utilizatorului. Este de aşteptat răspândirea pe scară largă a tehnicilor biometrice pe măsură ce tehnologiile de recunoaşterea feţei evoluează.

1.5 Auditarea unui sistem de baze de date

Auditarea bazelor de date se realizează pentru mai multe scopuri. De exemplu, bazele de date pot fi auditate pentru a se înregistra numărul de cereri lansate, numărul de actualizări, numărul de tranzacţii executate, numărul de accesări ale unităţilor de stocare secundare. Scopul acestora este proiectarea mai eficientă a sistemului. De asemenea, bazele de date pot fi auditate din motive de securitate. De exemplu, în acest mod se poate găsi răspunsul la următoarele întrebări:

- a fost evitată vreo regulă de control al accesului, fiind furnizată informaţie către utilizatori?
- a apărut problema inferenţei?
- a fost încălcată confidenţialitatea?
- au avut loc accesări neautorizate?

Auditările creează un *audit trail,* iar datele de audit pot fi stocate într-o bază de date. Această bază de date poate fi analizată pentru a detecta orice secvenţă sau comportament anormal. Au existat multe preocupări asupra utilizării *data mining* pentru auditare şi detectarea accesărilor neautorizate. Analiza informaţiilor de audit este deosebit de importantă la ora actuală, în contextul tranzacţiilor comerciale pe web. O organizaţie ar trebui să aibă posibilitatea de a demara o analiză şi a determina probleme precum fraudele asupra cardurilor de credit şi furtul identităţilor.

1.6 Vizualizările în contextul securităţii

Ca mecanism de securitate, vizualizările au fost studiate atât pentru securitatea obligatorie cât şi pentru cea opţională. De exemplu, s-ar putea să nu dorim să acordăm accesul asupra unei întregi relaţii mai ales dacă are multe atribute. Prin urmare, DBA-ul ar putea crea vizualizări şi acorda acces la acestea. Similar cazului securităţii obligatorii, vizualizărilor le pot fi asociate niveluri de securitate.

Vizualizările au anumite probleme specifice, inclusiv problema actualizării. Aceasta înseamnă că, dacă vizualizarea este actualizată atunci este nevoie să asigurăm că relaţiile de bază sunt actualizate. Prin urmare, dacă o vizualizare este actualizată de către John, iar acesta nu are acces la relaţia de bază, mai poate fi aceasta actualizată?

2. Aplicarea politicilor de securitate

În anii ’70, *System R* şi *INGRES* au dezvoltat tehnici, cum ar fi mecanismele de modificare a cererilor, pentru aplicarea politicilor de securitate. Limbajul *SQL* a fost extins pentru a permite specificarea politicilor de securitate şi a regulilor de control al accesului. Mai recent, limbaje precum *XML* şi *RDF* au fost extinse pentru a specifica politici de securitate.

2.1 Extensii *SQL* pentru securitate

Extensiile SQL pentru securitate permit specificarea politicilor. *SQL* a fost dezvoltat pentru definirea şi prelucrarea datelor în sistemele relaţionale. Ulterior, au fost dezvoltate diferite versiuni de *SQL* incluzând *SQL* pentru obiecte, multimedia şi *web*. *SQL* a influenţat foarte mult definirea şi prelucrarea datelor în ultimii 20 de ani.

Politicile de securitate pot fi specificate în cadrul definirii datelor. Limbajul *SQL* conţine comenzile *GRANT* şi *REVOKE* pentru acordarea, respectiv revocarea, drepturilor de acces ale utilizatorilor. De exemplu, dacă utilizatorul John primeşte acces pentru citire asupra relaţiei *EMP*, se poate specifica acest lucru prin:

GRANT read ON emp TO John;

Revocarea accesului se poate specifica prin:

REVOKE read ON emp FROM John;

SQL a fost extins cu constrângeri mai complexe, cum ar fi acordarea accesului pentru citirea unui tuplu dintr-o relaţie lui John sau acordarea de acces pentru scriere asupra unui element dintr-o relaţie lui Jane.

2.2 Modificarea cererilor

Modificarea cererilor a fost propusă prima dată în cadrul proiectului *INGRES*. Ideea este aceea de a modifica cererea pe baza constrângerilor de securitate.

Ilustrăm acest algoritm prin câteva exemple. Considerăm o cerere a lui John pentru regăsirea tuturor tuplurilor din *EMP*. Presupunem că John are acces pentru citire doar la tuplurile pentru care salariul este mai mic decât 30000 şi angajatul nu lucrează în departamentul de securitate.

Atunci cererea:

*SELECT * FROM EMP;*

va fi modificată în:

*SELECT * FROM emp*
Where salary < 30000
And dept != 'Security';

Clauza where a cererii conţine toate constrângerile asociate relaţiei. Pot exista constrângeri care implică mai multe relaţii. De exemplu, putem avea 2 relaţii EMP şi DEPT legate prin Dept#.

Apoi, cererea este modificată astfel:

*SELECT * FROM emp, dept*
WHERE emp.salary<30000
And emp.dept#=dept.dept#
And dept.name != 'Security';

Algoritmul de nivel înalt este prezentat în continuare:

Intrare : Cererea *Q*, mulţimea *S* a constrângerilor de securitate
Ieşire : Cererea *Q* modificată

For *c* in *S*
    If *c* relevant_for Q then
        Modify the where clause of *Q* via a negation
    End if;
End for;
Return *Q*;

O perspectivă asupra problemei inferenţei

Problema inferenţei a fost studiată în bazele de date statistice cu câţiva ani înainte de a deveni un subiect foarte important în MLS/DBMS. Inferenţa este procesul de lansare a unor cereri şi deducere a unor informaţii noi din răspunsurile legitime primite. Devine o problemă dacă utilizatorul nu este autorizat să cunoască informaţia dedusă.

Printre primele organizații care au studiat problema în bazele de date statistice s-a aflat biroul de recensământ (SUA). Bazele de date statistice sunt folosite de diferite organizaţii, de la biroul de recensământ până la organizaţii de marketing care doresc să studieze modelele de

comportament ale populaţiei. În esenţă, bazele de date statistice furnizează sume, medii, deviaţii standard etc., valori foarte utile în studiul populaţiei în termeni de numere sau comportament.

Exemple de situaţii în care poate apărea problema inferenţei statistice: furnizarea salariilor medii, dar protejarea valorilor individuale; informaţii referitoare la sănătatea dintr-un anumit judeţ, protejând înregistrările individuale ale persoanelor din acel judeţ. Poate un adversar să obţină media salariilor a 10 persoane, apoi 9, apoi 8 ş.a.m.d ?

O altă practică referitoare la bazele de date statistice este să nu fie folosite toate valorile din baza de date, ci să se lucreze cu valori de selecţie. Adică, mediile sunt calculate pe baza unui eşantion reprezentativ. Din astfel de date, este mai dificil să se obţină informaţii individuale protejate.

Inferenţa statistică a dobândit o importanţă mai mare odată cu dezvoltarea noilor tehnologii, cum ar fi *data warehousing* şi *data mining*. De exemplu, tehnologia *data warehouse* a fost dezvoltată pentru a oferi informaţie specifică pentru luarea deciziilor, incluzând sume şi medii. Datele din *data warehouse* ar putea fi neclasificate, dar ar putea exista informaţii protejate care rezidă în baza de date *back-end*. Pe baza informaţiei pe care o oferă data *warehouse*, se pot determina datele ‚sensibile" din baza de date *back-end*?

Data mining oferă informaţii necunoscute anterior, folosind diferite tehnici de raţionament cum ar fi inferenţa statistică. Astfel, problema inferenţei este privilegiată în *domeniul data mining*. Prin urmare, provocarea constă în descoperirea informaţiei protejate prin *mining*. O direcţie de cercetare recentă se referă la *privacy-preserving data mining*. Ideea este aceea de a asigura confidenţialitatea, dar în acelaşi timp de a oferi informaţii, probabil uşor modificate.

Abordări ale gestiunii inferenţei într-un MLS/DBMS

Abordările în soluţionarea problemei inferenţei pot fi împărţite în 2 grupuri: unul bazat pe constrângerile de securitate şi altul bazat pe structurile conceptuale.

Constrângerile de securitate sunt reguli care atribuie niveluri de securitate datelor. În această abordare, unele constrângeri de securitate sunt tratate în timpul procesării cererilor. Aceasta înseamnă că, în timpul operaţiei corespunzătoare cererii, constrângerile sunt examinate şi cererea este modificată. Mai mult, înainte de prezentarea rezultatelor constrângerile sunt examinate pentru a determina ce informaţie poate fi prezentată. Această abordare permite, de asemenea, ca unele constrângeri să fie procesate în timpul actualizării bazei de date. Astfel, constrângerile sunt examinate în timpul operaţiei de actualizare şi datelor le sunt atribuite niveluri adecvate de securitate. În cele din urmă, anumite constrângeri sunt tratate în timpul proiectării bazei de date, când metadatele sau schemele primesc niveluri de securitate.

În al doilea set de abordări, structurile conceptuale sunt folosite pentru a reprezenta aplicaţia şi raţionamentele asupra ei. Dacă există nerespectări potenţiale ale securităţii prin inferenţă, atunci acestea pot fi detectate la momentul proiectării aplicaţiei. Mai întâi, au fost utilizate grafurile pentru reprezentarea aplicaţiilor (Hinke, 1988). Ulterior, au fost utilizate reţele semantice şi grafuri conceptuale.