



BAZE DE DATE

CURS 1

Cuprinsul cursului

1. Generalități despre baze de date (structură, evoluție, caracteristici, funcționalități, perspective etc.)
2. Proiectarea (*design*) bazelor de date (diagrame *E/R*, diagrame conceptuale, *UML*, modelul relațional, proiectare, optimizare, normalizare, denormalizare, prelucrarea și optimizarea cererilor, regulile lui Codd etc.)
3. Introducere în bazele de date nerelaționale
4. Limbaje pentru gestionarea datelor. Neprocedural în baze de date (standardul *SQL*)

Bibliografie

- Connolly, T.M., Begg, C.E., Database Systems: A Practical Approach to Design, Implementation and Management, 6th edition, Pearson Education, 2014
- Popescu I., Alecu A., Velcescu L., Florea (Mihai) G., Programare avansată în Oracle9i, Editura Tehnică, București, 2004.
- Popescu, I., Velcescu, L., ***Proiectarea bazelor de date***, Editura Universității din București, 2008
- Popescu, I., Velcescu, L., ***Neprocedural în Oracle 10g***, Editura Universității din București, 2008

Concepte de baza

- Bază de date
- Sistem de gestiune a bazelor de date
- Dicționarul datelor

Concepte de baza – Baze de Date

- Ce este o bază de date (BD) ?
- Exemple?
- Cui aparțin datele?
- Cine le poate accesa?

Concepte de baza – Baze de Date

- **BAZA DE DATE** = ansamblu **structurat de date coerente**, fără **redundanță inutilă**, astfel încât acestea pot fi prelucrate eficient de mai mulți utilizatori într-un **mod concurent**
- Colecție de date **persistente**, care sunt folosite de către sistemele de aplicații ale unei anumite „întreprinderi“

Concepte de baza – Baze de Date

- „Întreprindere“ (eng. *Enterprise*):
 - **Reguli** proprii de funcționare
 - **Mulțime de date** referitoare la modul său de operare
- Datele din BD:
 - **Integrate**
 - **Partajate**

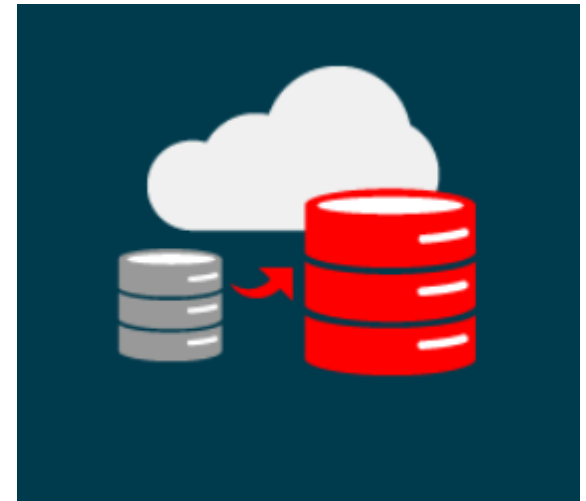
Concepte de baza - SGBD

- Ce este un Sistem de Gestiune a Bazelor de Date?
- Exemple?

Concepte de baza - SGBD

- **Sistem de Gestiune a Bazelor de Date (SGBD = DBMS – *Data Base Management System*)** este **un produs *software*** care asigură interacțiunea cu o bază de date
 - definirea
 - consultarea
 - actualizarea datelor din baza de date.
- Cererile de **acces la baza de date** sunt tratate și controlate de către **SGBD**.

Concepte de baza – BD si SGBD



<https://www.oracle.com/database/technologies/upgrades.html>

Concepte de baza - Dicționarul datelor

- **Dicționarul datelor** (catalog de sistem) = metabază de date
 - „date despre date“
 - structurat și administrat ca o bază de date
- **Ce informații regăsim în DD?**
- **Cum obținem aceste informații?**

Concepte de baza - Dicționarul datelor

- DD conține:
 - **descrierea** tuturor obiectelor unei baze de date
 - **starea** acestor obiecte
 - diversele **constrângeri** de securitate și de integritate
- DD poate fi interogat ca orice altă bază de date

Concepte de baza - DBA

- **Administratorul bazei de date** (DBA – *Data Base Administrator*) = persoană sau un grup de persoane ce răspund de ansamblul activităților legate de baza de date
 - analiză
 - proiectare
 - implementare
 - exploatare
 - întreținere etc.

Concepte de baza - DBA

- Categori de atribuții DBA:
 - atribuții de proiectare
 - atribuții administrative
 - atribuții operative
 - atribuții de coordonare

Gestiunea Bazelor de Date

Arhitectura unui sistem de baze de date are urmatoarele componente:

1. baza de date propriu-zisă în care se memorează datele
2. sistemul de gestiune a bazei de date, care realizează gestionarea și prelucrarea complexă a datelor
3. un dicționar al bazei de date (metabaza de date), ce conține informații despre date, structura acestora, statistici, documentație
4. mijloace *hardware* (comune sau specializate)
5. reglementări administrative destinate bunei funcționări a sistemului
6. personalul implicat (utilizatori finali, administratorul datelor, administratorul bazei de date, proiectanți / programatori de aplicații) → **4 categorii de utilizatori**

Categorii de utilizatori

- **Administratorul de date (DA)** este un manager care:
 - decide ce date trebuie stocate în baza de date
 - stabilește regulile de întreținere și de tratare a acestor date după ce sunt stocate
- **Administratorul bazei de date (DBA)** este un profesionist în domeniul IT, care:
 - creează baza de date reală
 - implementează elementele tehnice de control
 - este responsabil cu asigurarea funcționării sistemului la performanțe adecvate, cu monitorizarea performanțelor
 - furnizează diverse servicii tehnice

Categorii de utilizatori

- **Proiectanții de baze de date** pot acoperi 2 aspecte ale concepției (*design*-ului):
 - Proiectare fizică
 - Proiectare logică
- **Utilizatorii finali**

GESTIUNEA BAZELOR DE DATE – Proiectarea logica

- **Proiectarea logică** presupune o cunoaștere completă și amănunțită:
 - a **modelului real** de proiectat
 - a **regulilor de funcționare** ale acestuia
- Proiectarea **conceptuală** a bazei de date
- Modelul creat este **independent** de programele de aplicații, de limbajele de programare
- Proiectarea logică a bazei de date, îndreptată spre un anumit **model de date** (relațional, orientat obiect, ierarhic etc.)

GESTIUNEA BAZELOR DE DATE – Proiectarea fizica

- **Proiectarea fizică:**
 - preia modelul logic de date
 - stabilește cum va fi realizat fizic
 - presupune cunoașterea funcționalităților SGBD-ului, avantajele și dezavantajele fiecărei alternative
- Transpunerea modelului logic într-un set de **tabele** supuse unor **constrângeri**. Se selectează **structuri de stocare** și **metode de acces** specifice, astfel încât să se asigure **performanțe**. Se iau măsuri privind **securitatea** datelor

Cerinte minimale BD

Cerintele minimale ale unei baze de date sunt:

- **redundanță minimă** în date
- furnizarea în **timp util** a informațiilor solicitate (timpul de răspuns la o interogare)
- asigurarea unor **costuri minime** în prelucrarea și întreținerea informației
- capacitatea de a satisface, cu aceleași date, **necesități informaționale** ale unui număr mare de utilizatori
- posibilitatea de adaptare la cerințe noi, răspunsuri la interogări neprevăzute inițial (**flexibilitate**)
- exploatarea simultană a datelor de către mai mulți utilizatori (**sincronizare**)

Cerinte minimale BD

(continuare)

- asigurarea **securității datelor** prin mecanisme de protecție împotriva accesului neautorizat (confidențialitate)
- înglobarea unor facilități destinate validării datelor și recuperării lor în cazul unor deteriorări accidentale, garantarea (atât cât este posibil) că datele din baza de date sunt corecte (**integritate**)
- posibilitatea de valorificare a eforturilor anterioare și anticiparea nevoilor viitoare (**compatibilitate** și **expandabilitate**)
- permisivitatea, prin ierarhizarea datelor după criteriul frecvenței acceselor, a unor reorganizări (eventual dinamice) care sporesc **performanțele** bazei

Niveluri de abstractizare și de percepție a datelor

Sunt 4 niveluri de abstractizare și de percepție a datelor:

- intern (fizic)
- conceptual
- logic
- extern

Datele există doar la nivel **fizic**, iar celelalte trei niveluri reprezintă **virtualizări** ale acestora.

Niveluri de abstractizare și de percepție a datelor

- **Nivelul fizic (intern)** → schema fizică a datelor (bit, octet, adresă);
- **Nivelul conceptual** → schema conceptuală a datelor și reprezintă **viziunea programatorilor de sistem** asupra datelor;
- **Nivelul logic** → una din schemele logice posibile ale datelor și reprezintă **viziunea programatorului de aplicație** asupra datelor;
- **Nivelul virtual (extern)** → reprezintă **viziunea utilizatorului final** asupra datelor;

Independenta datelor

Independența datelor → două aspecte fundamentale:

- o modificare a structurii fizice nu va afecta aplicația
- reciproc, modificări ale aplicației vor lăsa nealterată structura fizică de date

Exista 2 tipuri principale de independență a datelor:

- Independența fizică
- Independența logică

Acestora li se adaugă independența față de strategiile de acces

Independenta datelor

- **Independența fizică** → posibilitatea **modificării schemei fizice** a datelor fără ca aceasta să implice modificarea schemei conceptuale, a schemei logice și a programelor de aplicație
- **Independența logică** → posibilitatea **modificării schemei conceptuale** a datelor fără ca aceasta să implice modificarea schemei logice și a programelor de aplicație
 - Independența logică \leftrightarrow fiecare utilizator are iluzia că este singurul beneficiar al unor date pe care, în realitate, le folosește în comun cu alți utilizatori
- **Independența față de strategiile de acces** → permite programului să **precizeze data pe care dorește să o acceseze**, dar nu modul cum accesează această dată

SGBD-ul va stabili drumul optim de acces la date

Limbaje pentru baze de date

Limbaje pentru baze de date

- **Limbaje pentru definirea datelor** (LDD – *Data Description Language*)
- **Limbaje pentru prelucrarea datelor** (LMD – *Data Manipulation Language*)
- **Limbaje pentru controlul datelor** (LCD – *Data Control Language*)

Limbaje pentru baze de date

Limbaje pentru definirea datelor (LDD – *Data Description Language*) permit:

- definirea **entităților** și a **atributelor** acestora
- precizarea **relațiilor** dintre date și strategiilor de acces la ele
- stabilirea de criterii diferențiate de **confidențialitate** și de validare automată a datelor utilizate

Limbaje pentru baze de date

Limbaje pentru prelucrarea datelor (LMD – *Data Manipulation Language*) – permit adaugarea, modificarea sau stergerea datelor

- o comandă are următoarea structură: operația, criterii de selecție, mod de acces (secvențial, indexat etc.)

Există limbaje LMD:

- **procedurale** – care specifică **cum** se obține rezultatul unei comenzi LMD
- **neprocedurale** – care descriu doar datele **ce** vor fi obținute și nu modalitatea de obținere a acestora

Limbaje pentru baze de date

Limbaje pentru controlul datelor (LCD – *Data Control Language*) – permit:

- asigurarea **confidențialității** și **integrității** datelor
- salvarea informației în cazul unor **defecțiuni**
- rezolvarea unor probleme de **concurență**

Obiectivele unui SGBD

Un **SGBD** (Sistem de Gestiune a Bazelor de Date) are următoarele **obiective fundamentale**:

- **Independența fizică** – independența structurilor de stocare în raport cu structurile de date din lumea reală
- **Independența logică** – fiecare grup de lucru poate să cunoască doar o parte a semanticii datelor, să vadă doar o submulțime a datelor și numai sub forma în care le dorește
- **Prelucrarea datelor de către neinformaticieni**
- **Administrarea centralizată a datelor**
- **Coerența datelor** – informația trebuie să satisfacă constrângeri statice sau dinamice, locale sau generale

Obiectivele unui SGBD

(continuare)

- **Neredundanța datelor**
 - Administrarea coerentă a datelor trebuie să asigure neduplicarea fizică a datelor
 - **Pentru a realiza performanțe referitoare la timpul de acces la date și răspuns la solicitările utilizatorilor, se acceptă o anumită redundanță a datelor**
- **Partajabilitatea datelor**
 - Aplicațiile pot să partajeze datele din baza de date în timp și simultan
 - O aplicație poate folosi date ca și cum ar fi singura care le utilizează, fără a ști că altă aplicație, concurent, le poate modifica

Obiectivele unui SGBD

Securitatea și confidențialitatea datelor

- Datele trebuie protejate de un acces neautorizat sau rău intenționat
 - există mecanisme care permit identificarea și **autentificarea** utilizatorilor
 - există proceduri de **acces autorizat** care depind de date și de utilizator
- Sistemul de gestiune trebuie să asigure securitatea fizică și logică a informației și să garanteze că **numai utilizatorii autorizați** pot efectua operații corecte asupra bazei de date

Dezavantajele SGBD-urilor

Un sistem de gestiune a bazelor de date are o **serie de limitări / dezavantaje** inerente:

- **complexitatea și dimensiunea sistemelor** pot să crească considerabil, din cauza necesității extinderii funcționalităților sistemului
- **costul**, care variază în funcție de mediu și funcționalitatea oferită, la care se adaugă cheltuieli periodice de întreținere
- **costuri adiționale** pentru elemente de *hardware*
- **costul conversiei** aplicațiilor existente, necesară pentru ca acestea să poată funcționa în noua configurație *hardware* și *software*
- **impactul unei defecțiuni** asupra aplicațiilor, bazei de date sau sistemului de gestiune

Structura unui SGBD

Structura unui Sistem de Gestiune a Bazelor de Date:

- are complexitate variabilă
- nivelul real de funcționalitate diferă de la produs la produs
- conține cel puțin 5 clase de module

Structura unui SGBD

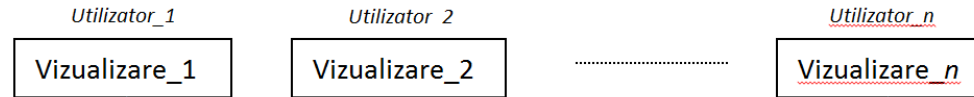
Module SGBD:

- **programe de gestiune a bazei de date (PGBD)** care realizează accesul fizic la date ca urmare a unei comenzi
- **module pentru tratarea LDD** ce permit traducerea unor informații în obiecte ce pot fi apoi exploatate în manieră procedurală sau neprocedurală
- **module pentru tratarea LMD** care permit utilizatorilor inserarea, ștergerea, reactualizarea sau consultarea informației dintr-o bază de date
- **module utilitare** care asigură întreținerea, prelucrarea, exploatarea corectă și ușoară a bazei de date
- **module de control** care permit controlul programelor de aplicație, asigurarea confidențialității și integrității datelor, rezolvarea unor probleme de concurență, recuperarea informației în cazul unor avarii sau defecțiuni *hardware* sau *software*

Independența logică și fizică

Nivelul extern

(modul în care
utilizatorii percep
datele)



Nivelul conceptual

(realizează transpunerea și
independența dintre
nivelul extern și cel intern)

Schema conceptuală

CE date sunt stocate și relațiile
dintre ele

Nivelul intern

(nivelul la care SGBD și SO percep
datele)

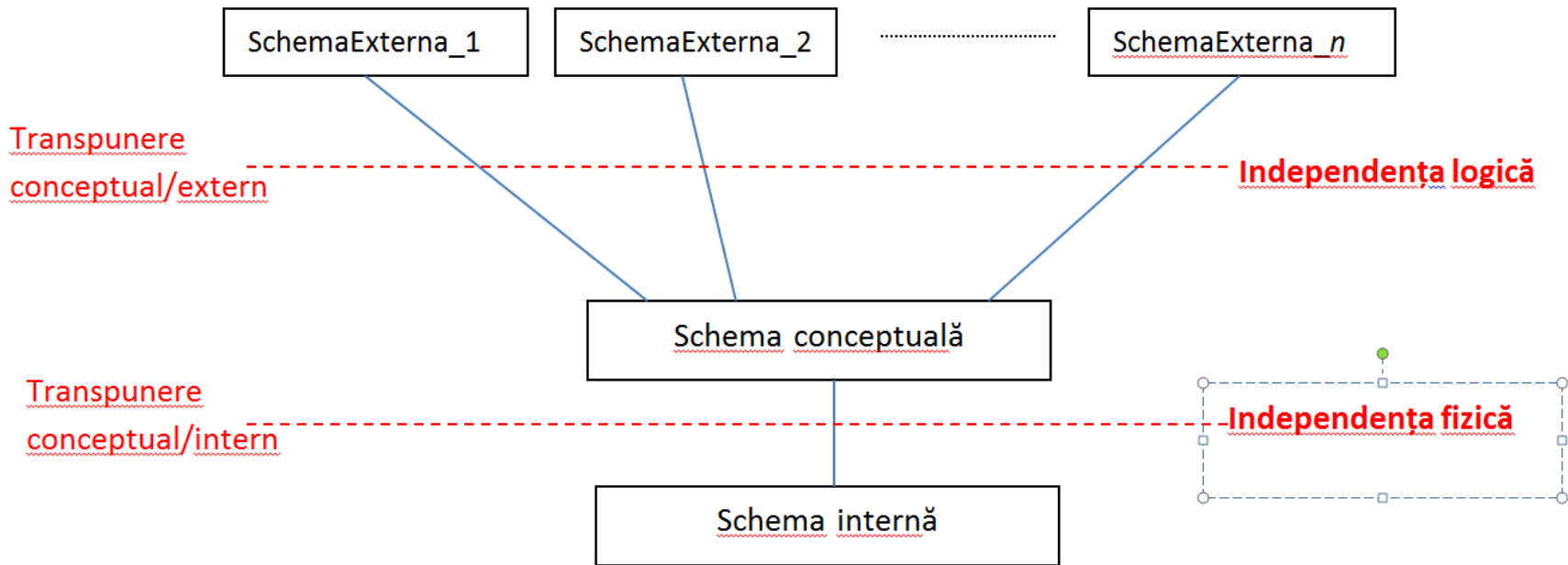
Schema internă

CUM sunt stocate datele

Organizarea fizică a datelor



Independența logică și fizică



Independența logică – imunitatea schemelor externe față de modificările efectuate în schema conceptuală

Independența fizică – imunitatea schemei conceptuale față de modificările efectuate în schema internă

Corespondența extern-extern – permite definirea unei vizualizări în funcție de altele, fără a necesita o definiție explicită a corespondenței cu nivelul conceptual

Arhitectura sistemelor de gestiune a bazelor de date

Niveluri ale arhitecturii sistemelor de gestiune a bazelor de date

Independența fizică și logică a datelor => adoptarea unei arhitecturi de baze de date organizată pe trei niveluri:

- **nivelul intern** (baza de date fizică);
- **nivelul conceptual** (modelul conceptual, schema conceptuală);
- **nivelul extern** (modelul extern, subschema, vizualizarea).

Nivelul conceptual al arhitecturii SGBD

Nivelul central -> nivelul conceptual

- structura canonică a datelor ce caracterizează procesul de modelat (structura semantică a datelor fără implementarea pe calculator)

Schema conceptuală permite:

- definirea tipurilor de date ce caracterizează proprietățile elementare ale entităților
- definirea tipurilor de date compuse care permit regruparea atributelor pentru a descrie entitățile modelului și legăturile între aceste entități
- definirea regulilor pe care trebuie să le respecte datele

Nivelul intern al arhitecturii SGBD

Nivelul intern = structura internă de stocare a datelor

Schema internă permite:

- descrierea datelor unei baze sub forma în care sunt stocate în memoria calculatorului

La **nivel conceptual sau intern**, schemele descriu o bază de date

La **nivel extern** schemele descriu doar o parte din date care prezintă interes pentru un utilizator sau un grup de utilizatori

Nivelul extern al arhitecturii SGBD

Nivel extern

- **Schema externă** reprezintă o descriere a unei părți a bazei de date ce corespunde viziunii unui program sau unui utilizator
- Modelul extern folosit este dependent de limbajul utilizat pentru prelucrarea bazei de date
- Schema externă permite **asigurarea unei securități** a datelor
 - Un grup de lucru va accesa doar datele descrise în schema sa externă, iar restul datelor sunt protejate împotriva accesului neautorizat sau rău intenționat

Pentru o bază de date particulară există o singură schemă internă, o singură schemă conceptuală, dar există mai multe scheme externe

Coreșpondențe între niveluri

- **coreșpondența conceptual-intern** → modul în care înregistrările și câmpurile conceptuale sunt reprezentate la nivel intern
- **coreșpondența extern-conceptual** → cheia independenței logice de date
- **coreșpondența extern-extern** → permite definirea unor vizualizări externe în funcție de altele, fără a necesita o definiție explicită a coreșpondenței cu nivelul conceptual

Arhitectura sistemelor de gestiune a bazelor de date

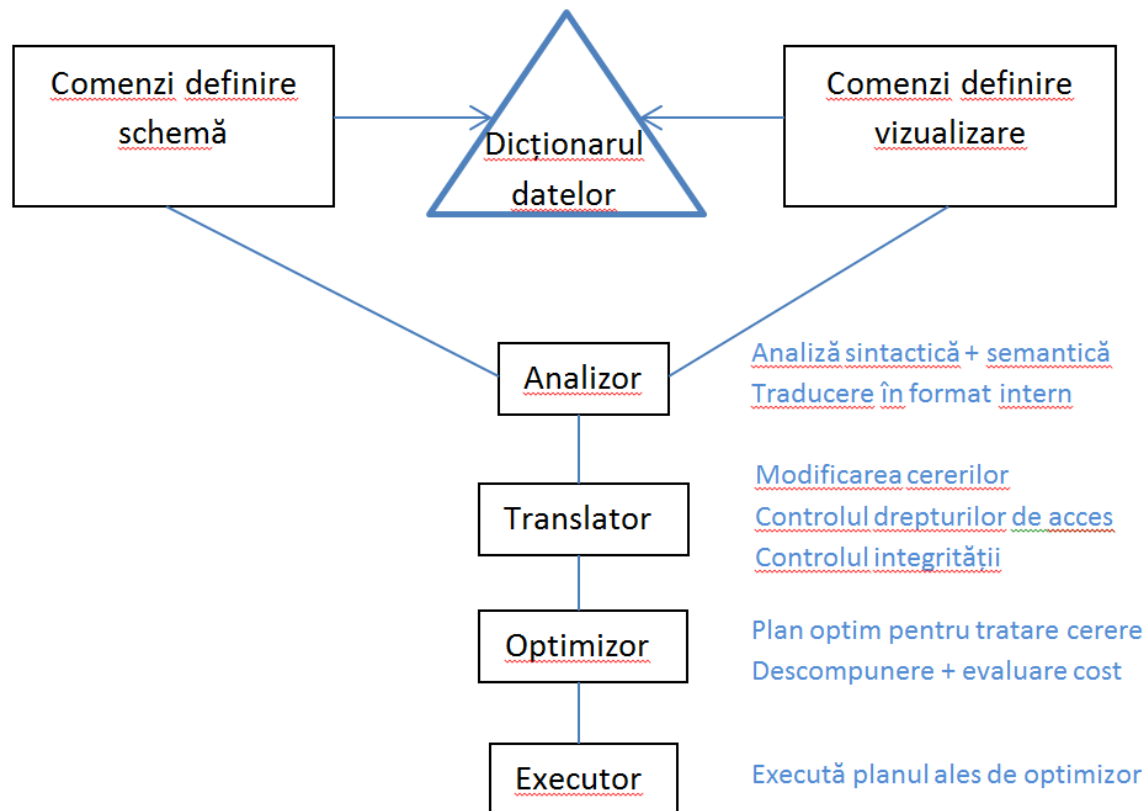
Arhitectura funcțională de referință propusă de grupul de lucru **ANSI/X3/SPARC** este axată pe dicționarul datelor și cuprinde două părți:

- prima, permite **descrierea datelor** (compoziția dicționarului datelor)
- a doua, permite **prelucrarea datelor** (interogarea și reactualizarea bazei de date)

Arhitectura sistemelor de gestiune a bazelor de date

- Gardarin a propus o arhitectură funcțională apropiată de arhitectura sistemelor de gestiune actuale.
- Arhitectura are la bază doar două niveluri:
 - **Schema** – care corespunde integrării schemelor interne și conceptuale
 - **Vizualizarea** – care este o schemă externă

Arhitectura sistemelor de gestiune a bazelor de date



Evolutia bazelor de date

Istoria BD și SGBD poate fi rezumată în **trei generații**:

- sisteme ierarhice și rețea
- sisteme relaționale
- sisteme avansate (orientate obiect, relaționale orientate obiect, deductive, multimedia, multibaze, active, temporale, distribuite, decizionale, magazii de date etc.)

Evolutia bazelor de date

Baze de date ierarhice și rețea

- datele sunt reprezentate la nivel de articol prin legături ierarhice (arbore) sau de tip graf
- slaba independență fizică a datelor complică administrarea și prelucrarea acestora
- limbajul de prelucrare a datelor impune programatorului să specifice drumurile de acces la date

Evolutia bazelor de date

Baze de date relaționale

- Modelul relațional (1970) tratează entitățile ca niște relații. Piața actuală de baze de date este acoperită în majoritate de sisteme **relaționale**.
- Bazele de date relaționale sunt caracterizate de:
 - structuri de date simple, intuitive
 - inexistența pointerilor vizibili pentru utilizator
 - constrângeri de integritate
 - operatori aplicați relațiilor care permit definirea, căutarea și reactualizarea datelor

Evolutia bazelor de date

- Bazele de date relaționale nu folosesc obiecte complexe și dinamice
- Nu realizează gestiunea datelor distribuite
- Nu realizează gestiunea cunoștințelor

A treia generație de SGBD-uri, **sistemele avansate**, încearcă să depășească aceste limite ale sistemului relațional

Evolutia bazelor de date

Baze de date orientate obiect

Ce nu realizează BD relaționale?

- Suportul obiectelor complexe și dinamice, prelucrarea acestora
- Sistemele relaționale nu modelează obiecte complexe ca grafuri, liste etc.
 - Un obiect complex poate să fie descompus în relații, dar apar dificultăți atât la descompunerea, cât și la refacerea acestuia prin compunere
- Un sistem relațional nu suportă obiecte dinamice care încorporează atât partea de date (informații) efective, cât și o parte relativă la tratarea acestora.

Îmbinarea tehnicii limbajelor orientate obiect cu a bazelor de date a permis realizarea bazelor de date orientate obiect

Evolutia bazelor de date

Baze de date orientate obiect (continuare)

Avantaje SGBDOO:

- realizează o modelare superioară a informației
- furnizează posibilități superioare de deducție (ierarhie de clase, moștenire)
- permit luarea în considerare a aspectelor dinamice și integrarea descrierii structurale și comportamentale

Dezavantaje:

- absența unui SGBDOO de referință
- gestiunea obiectelor complexe este mai dificilă
- utilizatorii au investit sume uriașe în sistemele relaționale și nu le pot abandona cu ușurință. Trecerea la tehnologia orientată obiect implică investiții mari și nu păstrează aproape nimic din vechile soluții

Evolutia bazelor de date

Baze de date relaționale orientate obiect

Simplitatea modelului relațional + puterea tehnologiei orientate obiect => baze de date relaționale orientate obiect

Construcția unui sistem de gestiune de baze de date relaționale orientate obiect (SGBDROO) trebuie să pornească de la cele existente. Aceasta se poate realiza în două moduri:

- dezvoltând un sistem relațional prin adăugarea caracteristicilor obiectuale necesare
- pornind de la un sistem orientat obiect și adăugând caracteristicile relaționale

Evolutia bazelor de date

Baze de date deductive

- O relație este o mulțime de înregistrări ce reprezintă **fapte**
- Cunoștințele sunt **asertiuni** generale și abstracte asupra faptelor
 - Cunoștințele permit să raționezi, ceea ce permite deducerea de noi fapte, plecând de la fapte cunoscute
- Un SGBD relațional suportă o formă limitată de cunoștințe, și anume constrângerile de integritate, iar restul trebuie integrate în programele de aplicație
- Prin programarea logică se gestionează cunoștințe relativ la baze de date care, în general, sunt relaționale
- Deducerea de noi informații, plecând de la informațiile stocate în baza de date

Evolutia bazelor de date

Baze de date deductive (continuare)

Un SGBD deductiv posedă:

- un **limbaj de definire a datelor** care permite definirea structurii predicatelor sub formă de relații și constrângeri de integritate asociate
- un **limbaj de prelucrare a datelor** care permite efectuarea reactualizărilor asupra datelor și formularea unor cereri
- un **limbaj de reguli de deducție** care permite ca, plecând cu predicatele definite anterior, să se specifice cum pot fi construite predicate derivate

Evolutia bazelor de date

Baze de date distribuite

- Sistem distribuit = ansamblu de mașini ce sunt interconectate printr-o rețea de comunicație și utilizate într-un scop global
- **Obiectiv BDD: Administrarea și prelucrarea datelor distribuite**, situate pe diferite calculatoare și exploatate de sisteme eterogene
- Bazele de date distribuite sunt sisteme de baze de date cooperante care rezidă pe mașini diferite, în locuri diferite
- Această mulțime de baze de date este **exploatată de utilizator ca și cum ar fi o singură bază de date**
- Programul de aplicație care exploatează o bază de date distribuită poate avea acces la date rezidente pe mai multe mașini, fără ca programatorul să cunoască localizarea datelor

Modelul **relațional** a rămas instrumentul principal prin care se realizează prelucrarea datelor distribuite.

Evolutia bazelor de date

Baze de date cu suport decizional

- Sistemele informatice, în particular bazele de date, au ajuns la maturitate
- Cantitate mare de informații ale companiilor, păstrată în tabele istorice
 - nefolositoare sistemelor operaționale ale companiei, care funcționează cu date curente
- Analiza date istorice → informații despre tendințe și evoluții care ar putea interesa compania
 - Sunt necesare tehnologii și instrumente speciale
 - Sunt analizate colecții de date provenind din sistemele operaționale ale companiei sau din surse externe
- Principalul scop al acestor baze de date a fost de a întâmpina nevoile sistemelor operaționale, a căror natură este inherent tranzacțională

Evolutia bazelor de date

Baze de date cu suport decizional (continuare)

- Sistemele tranzacționale sunt interesate, în primul rând, să controleze la un moment dat o singură tranzacție
- Un sistem operațional tipic operează cu evenimente predefinite și necesită acces rapid la date
- Nevoile sistemelor operaționale nu se schimbă mult în timp

Evolutia bazelor de date

Baze de date cu suport decizional (continuare)

- S-au pus la punct principii și tehnologii noi care să servească procesului de **analiză** și **administrare** a datelor
- O bază de date optimizată în acest scop definește o **Data Warehouse** (magazie de date)
- Principiul pe care îl urmează este cunoscut sub numele de procesare analitică (**OLAP – On Line Analytical Processing**)
- Principiul pe care se bazează sistemele tranzacționale a fost numit procesare tranzacțională (**OLTP – On Line Transactional Processing**)

Evolutia bazelor de date

Baze de date cu suport decizional (continuare)

- Aplicațiile unei *Data Warehouse* trebuie să ofere răspunsuri unor întrebări de tipul:
 - „Care zi din săptămână este cea mai aglomerată?”
 - „Ce clienți fideli nu au beneficiat de reduceri de prețuri?”
- Interogările pe BD analitice sunt *ad-hoc*, nu sunt predefinite → baza de date trebuie **optimizată** astfel încât să fie capabilă **să răspundă la orice fel de întrebare** care poate implica mai multe tabele

Evolutia bazelor de date

OLTP *versus* Data Warehouse

Sistemele OLTP	Data Warehouse
Păstrează date curente	Păstrează date istorice
Stochează date detaliate	Stochează date detaliate, agregate ușor sau puternic
Datele sunt dinamice	Datele sunt în mare măsură statice
Prelucrare repetitivă	Prelucrare ad-hoc, nestructurată și euristică
Nivel înalt de transfer al tranzacțiilor	Nivel mediu sau scăzut de transfer al tranzacțiilor
Tipar de utilizare previzibil	Tipar de utilizare imprevizibil
Conduse prin tranzacții	Conduse prin analiză
Susțin deciziile de zi cu zi	Susțin deciziile strategice
Deservesc un număr mare de utilizatori	Deservesc un număr relativ redus de utilizatori din administrație
Orientate spre aplicații	Orientate spre subiect

Evolutia bazelor de date

Baze de date cu suport decizional (continuare)

Marii producători de sisteme de gestiune a bazelor de date relaționale, precum *Oracle*, au introdus în produsele lor construcții care să faciliteze accesul la datele din sistemele fundamentale pentru luarea de decizii

- modalitate mai inteligentă de a realiza **operația de compunere** între două sau mai multe tabele
- metode de **indexare** noi, potrivite pentru marile cantități de date statice cu care operează sistemele *Data Warehouse*
- capacitatea de a detecta și **optimiza** interogări de un tip special
- posibilitatea de a folosi **mai multe procesoare** pentru a rezolva o interogare

Evolutia bazelor de date

Baze de date cu suport decizional (continuare)

- Un efort ce trebuie făcut pentru construirea unui sistem de suport pentru decizii (**DSS – *Decision Support System***) constă în procesul de descoperire a informațiilor utile din baza de date
- Acest proces, numit ***Data Mining*** sau ***Knowledge Discovery in Databases (KDD)***, procesează mari cantități de date, a căror corelare nu este neapărat evidentă, în vederea descoperirii de tendințe și tipare

Arhitectura *multitier* a sistemului *Oracle*

Arhitectura cu mai multe niveluri (*multitier*) conține următoarele elemente:

- unul sau mai mulți *clienti* care inițiază operații
- unul sau mai multe *servere* de aplicații care execută părți ale operațiilor
- un *server* de baze de date care stochează datele folosite de operații
- privilegiile *server*-ului de aplicații sunt limitate pentru a preveni execuția operațiilor nedorite sau inutile în timpul unei operații *client*

Arhitectura *multitier* a sistemului *Oracle*

- **Clientul** (un browser Web sau un proces user) → cerere pentru baza de date
 - Conectarea la server-ul bazei de date se face printr-unul sau mai multe servere de aplicații
- Server-ul de aplicații constituie interfața dintre clienți și serverul bazei de date, asigurând accesul la informații. De asemenea, el include un nivel adițional pentru securitate
- Server-ul de baze de date pune la dispoziția server-ului de aplicații informațiile necesare pentru soluționarea operațiilor lansate de către client

Arhitectura *multitier* a sistemului *Oracle*

