



# BAZE DE DATE

CURS 1



# NOTARE

- **Modalitatea de evaluare și cerințele**
  - **Fișiere disponibile pe Moodle după primul curs**

# OPORTUNITĂȚI

ORACLE Academy

- Conturi individuale pe platforma <http://ilearning.oracle.com>
  - ❑ Cursuri de Java și baze de date
  - ❑ Quizz-uri
  - ❑ Certificări junior
- ➔ Cerere prin email: [letitia.marin@unibuc.ro](mailto:letitia.marin@unibuc.ro)



# PLAN CURS

1. Generalități despre baze de date (structură, evoluție, caracteristici, funcționalități, perspective etc.).
2. Proiectarea (*design*-ul) bazelor de date:
  - proiectare diagrame *E/R*
  - modelul relațional, diagrame conceptuale, *UML*
  - prelucrarea și optimizarea cererilor, normalizare, denormalizare, 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



# INTRODUCERE

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



# INTRODUCERE

BD

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



# INTRODUCERE

BD

- **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“





# INTRODUCERE

BD

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



# INTRODUCERE

SGBD

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



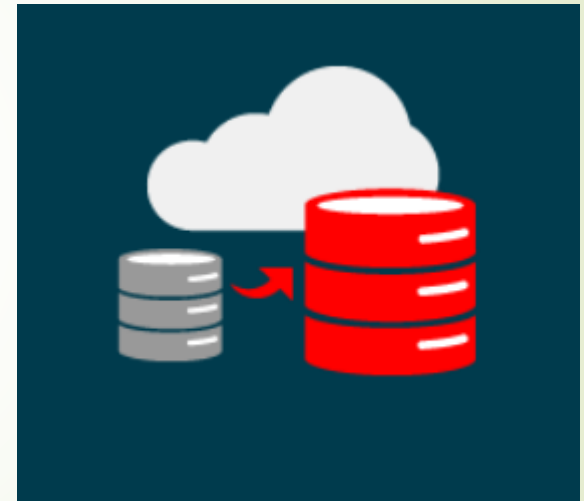
# INTRODUCERE

## 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**.

# INTRODUCERE

BD



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



# INTRODUCERE

## 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?**



# INTRODUCERE

## Dicționarul datelor

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

# INTRODUCERE

## 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.



# INTRODUCERE

## DBA

- Categori de atribuții DBA:
  - atribuții de proiectare,
  - atribuții administrative,
  - atribuții operative și
  - atribuții de coordonare.



# GESTIUNEA BAZELOR DE DATE

## ► Arhitectura unui **sistem de baze de date**:

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**



# GESTIUNEA BAZELOR DE DATE

- **Administratorul de date** (DA) este un manager
  - decide care date trebuie stocate în baza de date;
  - stabilește regulile de întreținere și de tratare a acestor date după ce sunt stocate.



# GESTIUNEA BAZELOR DE DATE

- **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 etc.



# GESTIUNEA BAZELOR DE DATE

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

# GESTIUNEA BAZELOR DE DATE

- **Proiectarea logică** presupune o cunoaștere completă și amănunțită:
  - a **modelului real** de proiectat și
  - a **regulilor de funcționare** a 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 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.

# GESTIUNEA BAZELOR DE DATE

## ► Cerințe minimele BD:

- **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**);



# GESTIUNEA BAZELOR DE DATE

## ➤ **Cerințe minimele 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.



# GESTIUNEA BAZELOR DE DATE

## ➤ Patru 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.

# GESTIUNEA BAZELOR DE DATE

- **Nivelul fizic (intern)** → schema fizică a datelor (bit, octet, adresă);
- **Nivelul conceptual** → schema conceptuală a datelor (articol, înregistrare, zonă) 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.

# GESTIUNEA BAZELOR DE DATE

- **Independența datelor** → două aspecte fundamentale:
  - o modificare a structurii fizice nu va afecta aplicația și
  - reciproc, modificări ale aplicației vor lăsa nealterată structura fizică de date.
- **Independența fizică**
- **Independența logică**

# GESTIUNEA BAZELOR DE DATE

- **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.



# GESTIUNEA BAZELOR DE DATE

## ► 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.

# GESTIUNEA BAZELOR 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*)

# GESTIUNEA BAZELOR DE DATE

- **Limbaje pentru definirea datelor** (LDD – *Data Description Language*).
  - definirea **entităților** și a **atributelor** acestora
  - sunt precizate **relațiile** dintre date și strategiile de acces la ele
  - sunt stabilite criterii diferențiate de **confidențialitate** și de validare automată a datelor utilizate.

# GESTIUNEA BAZELOR DE DATE

- **Limbaje pentru prelucrarea datelor** (LMD – *Data Manipulation Language*).
  - 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 și
    - **neprocedurale**, care descriu doar datele **ce** vor fi obținute și nu modalitatea de obținere a acestora.





# GESTIUNEA BAZELOR DE DATE

- **Limbaje pentru controlul datelor** (LCD – *Data Control Language*).
  - asigurarea **confidențialității** și **integrității** datelor
  - salvarea informației în cazul unor **defecțiuni**
  - rezolvarea unor probleme de **concurență**.

# GESTIUNEA BAZELOR DE DATE

## ► Obiectivele unui SGBD

### ► 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.

# GESTIUNEA BAZELOR DE DATE

## ► **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

# GESTIUNEA BAZELOR DE DATE

- **Obiectivele unui SGBD** (continuare)

- **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.

# GESTIUNEA BAZELOR DE DATE

## ➤ Dezavantajele SGBD-urilor

- **complexitatea și dimensiunea sistemelor** pot să crească considerabil, datorită 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.



# GESTIUNEA BAZELOR DE DATE

## ➤ Structura unui SGBD

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

# GESTIUNEA BAZELOR DE DATE

## ➤ **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* etc.



# Independența logică și fizică

## Nivelul extern

(modul în care utilizatorii percep datele)

Utilizator\_1

Vizualizare\_1

Utilizator 2

Vizualizare\_2

Utilizator\_n

Vizualizare\_n

## 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

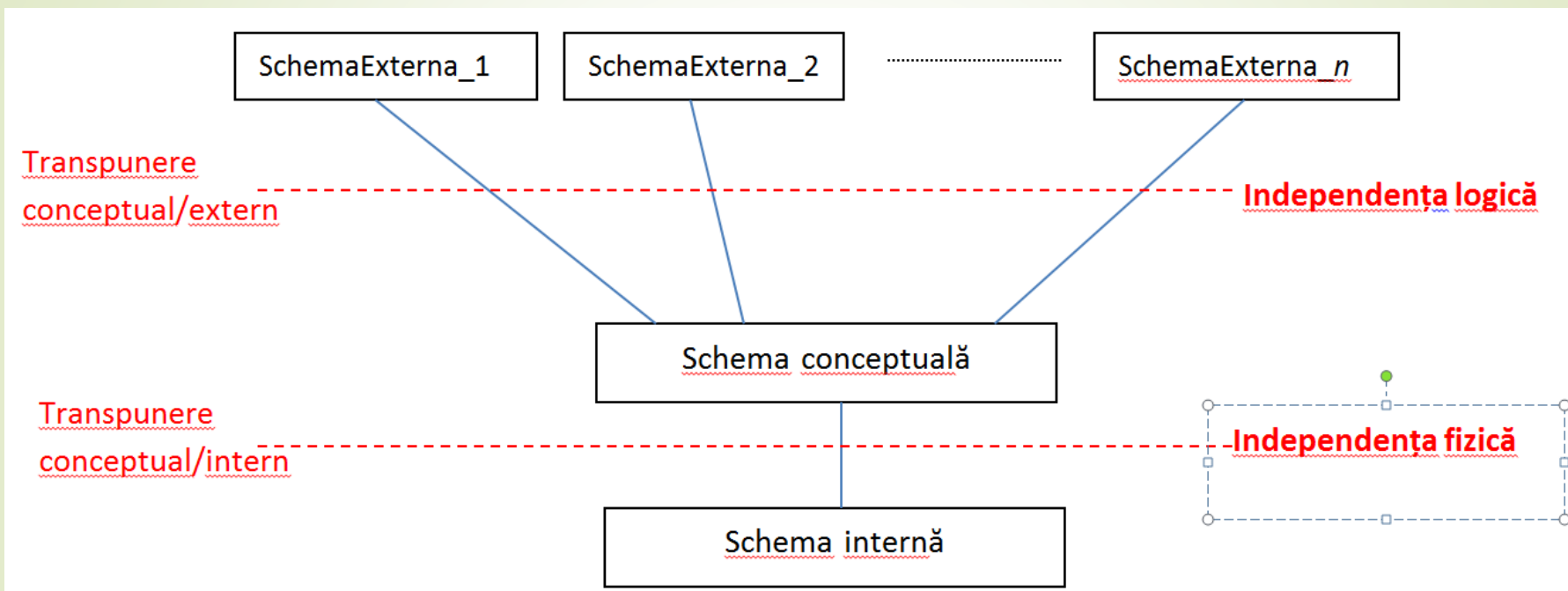
CUM sunt stocate datele

## 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

**Correspondența 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

- **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).

# Arhitectura sistemelor de gestiune a bazelor de date

## ➤ 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 etc.



# Arhitectura sistemelor de gestiune a bazelor de date

- **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.*

# Arhitectura sistemelor de gestiune a bazelor de date

## ➤ 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.*

# Arhitectura sistemelor de gestiune a bazelor de date

## ➤ Corespondențe între niveluri


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

Figura – curs Moodle!



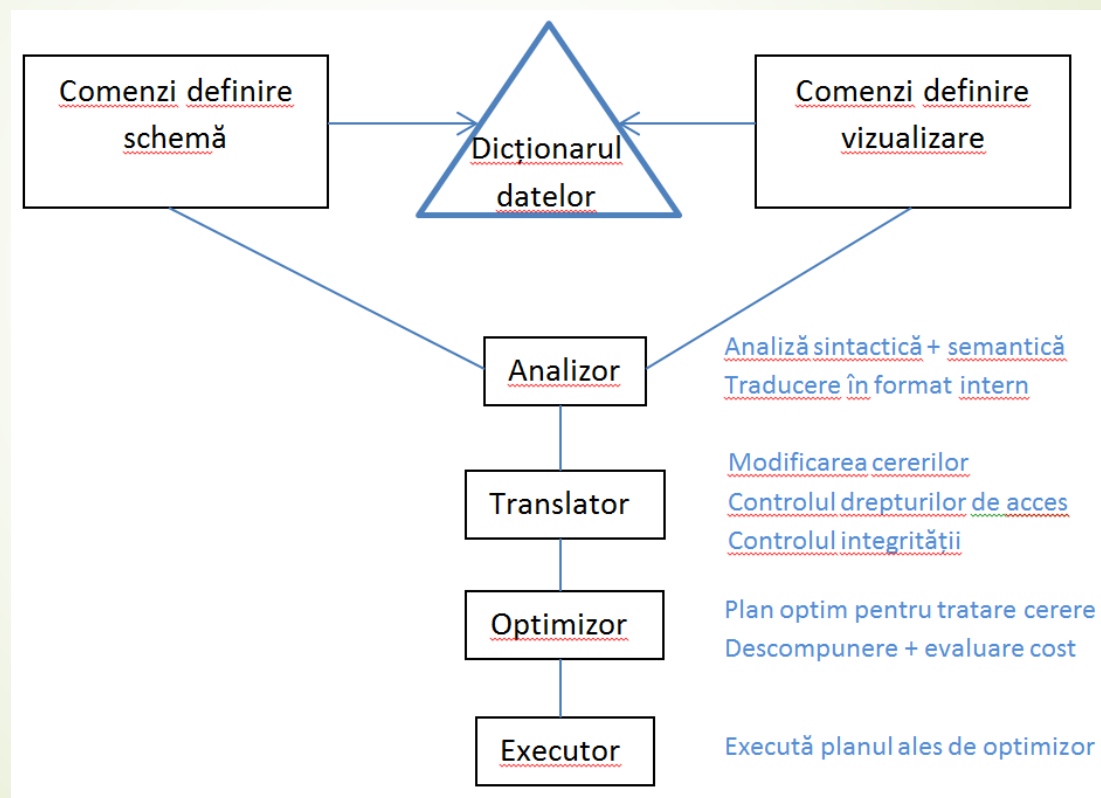


# Arhitectura sistemelor de gestiune a bazelor de date

- **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



# EVOLUȚIA 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.).



# EVOLUȚIA 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.

# EVOLUȚIA 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.



# EVOLUȚIA 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.

# EVOLUȚIA 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.
    - Limbajele modelului relațional permit prelucrarea cu dificultate a obiectelor complexe.
  - 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**.

# EVOLUȚIA 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.



# EVOLUȚIA 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 sau
  - pornind de la un sistem orientat obiect și adăugând caracteristicile relaționale.



# EVOLUȚIA BAZELOR DE DATE

## ➤ Baze de date deductive

- O relație este o mulțime de înregistrări ce reprezintă **fapte**.
- Cunoștințele sunt **aserțiuni** 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.



# EVOLUȚIA BAZELOR DE DATE

- **Baze de date deductive** (continuare)
  - 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.

# EVOLUȚIA 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.

# EVOLUȚIA 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 **exploată 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.

# EVOLUȚIA 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.

# EVOLUȚIA 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.

# EVOLUȚIA 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**).



# EVOLUȚIA 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.



# EVOLUȚIA 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
Deserves un număr mare de utilizatori	Deserves un număr relativ redus de utilizatori din administrație
Orientate spre aplicații	Orientate spre subiect

# EVOLUȚIA 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.

# EVOLUȚIA 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

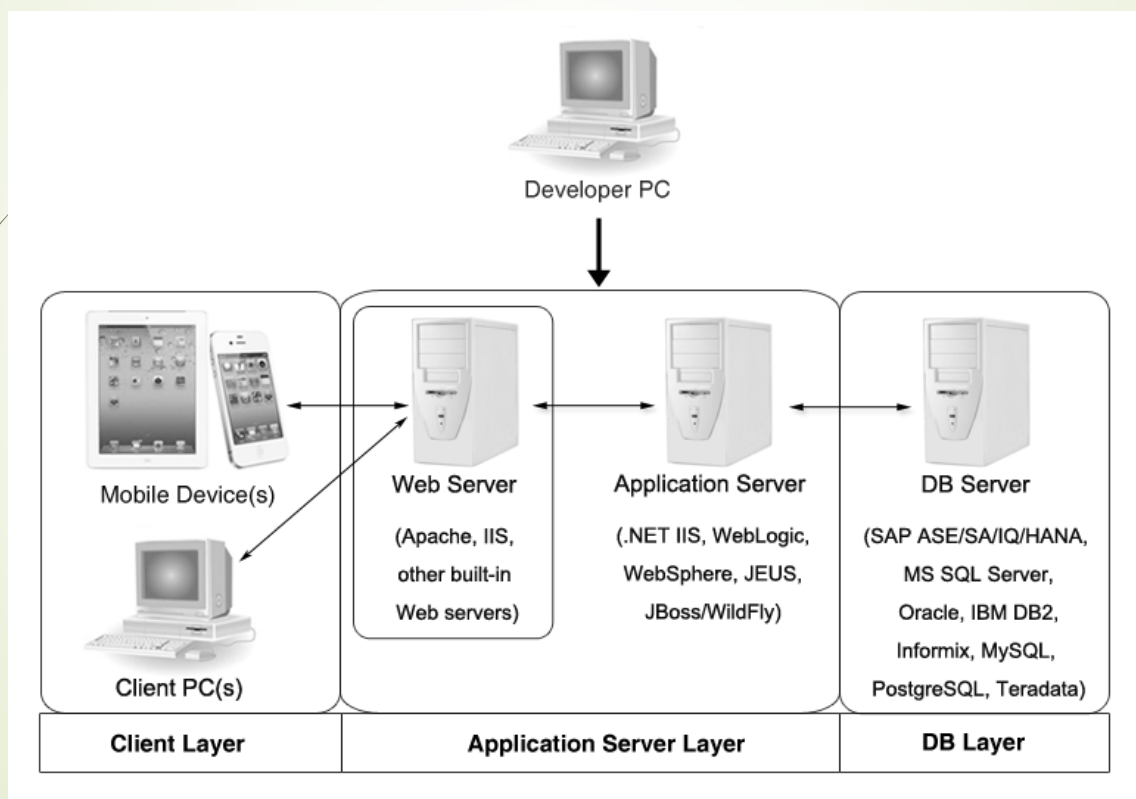
- **Arhitectura cu mai multe niveluri** (*multitier*) conține următoarele elemente:
  - unul sau mai mulți *clienti* care inițiază operații;
  - unul sau mai multe *servere* de aplicații care execută părți ale operațiilor;
  - un *server* de baze de date care stochează datele folosite de operații.
  - 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 *server*-e de aplicații.
- **Server-ul de aplicații** constituie interfața dintre *clienti* 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*

➤ **Arhitectura *n-tier*** :



[https://docs.appeon.com/appeon\\_online\\_help/ps2019/installation\\_guide/ch37s02.html](https://docs.appeon.com/appeon_online_help/ps2019/installation_guide/ch37s02.html)