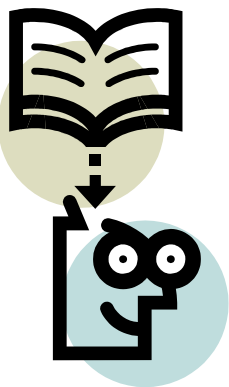


***ANALIZA SI PROIECTAREA  
SISTEMELOR INFORMATICE  
APLICATII IN ACCESS***

**Suport de curs și  
de laborator**

# CAPITOLUL 1. TIPURI DE SISTEME INFORMATICE

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p><i>Prezentarea principalelor elemente ale sistemului informatic</i></p> <p><i>Principalele tipuri și subtipuri de sisteme informatice</i></p>
---	---

## Elementele sistemului informatic:

- ☐ Resursa fizică (hardware)
- ☐ Resursa logică (software)
- ☐ Baza de date
- ☐ Resursa umană și cadrul organizatoric

---

## **1. SISTEME DESTINATE CONDUCERII (*Management Support Systems – MSS*)**

### **1.A. SISTEME SUPT ALE EXECUTIVULUI (*Executive Support Systems – ESS*)**

- sunt sisteme informatice destinate conducerii strategice (top și middle management)
- oferă acces rapid și selectiv la date interne și externe firmei
- oferă informații privind starea curentă și tendințele în evoluția factorilor cheie selectați pentru analiză
- sunt ușor de utilizat (facilități de calcul și reprezentări grafice) și asigură un mod de lucru interactiv

### **1.B. SISTEME SUPT DE DECIZIE (*Decision Support Systems – DSS*)**

- valorifică informațiile interne oferite de TPS și MIS și informațiile provenite din mediul exterior al organizației (curs valutar, rata dobânzii, prețul produselor realizate de firmele concurente, prețul materiilor prime)
- sunt sisteme interactive, care utilizează o bază de modele de decizii (modele matematice)
- oferă managerilor modele de analiză, mijloace de regăsire a datelor; stimulează creativitatea

*Fac parte din DSS :*

- *Sisteme interactive de asistare a deciziei (SIAD)*
  - *SIAD pleacă de la decidenți și de la natura deciziei ce trebuie luată*
  - *decidentul rămâne ultimul mediator care apreciază raportarea la realitate și la experiența sa*
  
- *Sisteme expert (SE)*
  - *se stochează cunoștințele expertului uman referitoare la un anumit domeniu într-o bază de cunoștințe*
  - *motorul inferențial permite deducerea unor concluzii, prin derularea unor raționamente automate*

*Exemple de utilizare SIAD și SE :*

- *decizia de acordare a unor credite*
- *consilierea clienților privind plasamentul de capital*
- *determinarea necesarului de fond de rulment pentru firmele clienți ai băncii*

### **1.C. SISTEME DESTINATE CONDUCERII CURENTE(Management Information Systems – MIS)**

- *sunt destinate managementului operațional*
- *utilizează datele interne ale firmei oferite de TPS*
- *asigură generarea rapoartelor sintetice, de rutină, necesare în procesul conducerii curente; oferă informații*

*managerilor pentru desfășurarea analizelor săptămânale, lunare și anuale*

- *rapoartele au o structură bine stabilită și sunt oferite la cerere și / sau periodic*

## **2. SISTEME OPERAȚIONALE (Operational Support Systems)**

### **2.A. SISTEME PENTRU PROCESAREA TRANZACȚIILOR (Transaction Processing Systems – TPS)**

- *primele utilizate în firme; au rolul de a prelua efortul uman în culegerea și prelucrarea datelor din operațiunile curente, de rutină*
- *asigură actualizarea curentă a bazei de date*
- *prelucrările au caracter repetitiv, complexitate redusă și volum mare*

*TPS cuprind: SI - contabilitatea financiară, SI – gestiunea producției, SI – gestiunea stocurilor*

### **2.B. SISTEME PENTRU ACTIVITATEA DE BIROTICĂ (Office Automation Systems – OAS)**

*- din această categorie face parte:*

*software specializat pentru procesare de texte*

*software de comunicație*

*sisteme pentru lucru colaborativ*

*sisteme pentru procesarea imaginilor*

---

---

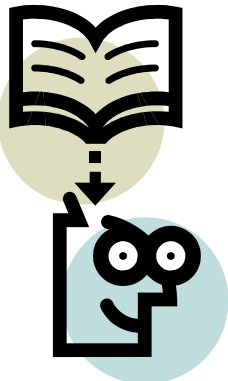
*sisteme multimedia*

*software pentru managementul activității de birou, de exemplu agenda de birou*

## **2.C. SISTEME PENTRU CONTROLUL PROCESELOR (Process Control Systems – PCS)**

*- asigură controlul automat al proceselor industriale*

## CAPITOLUL 2. REALIZAREA SISTEMELOR INFORMATICE

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p><i>Etapele de realizare a pachetelor de programe</i></p> <p><i>Stadiul de concepere</i></p> <p><i>Stadiul de exploatare</i></p> <p><i>Procese tehnologice de realizare a pachetelor de programe</i></p> <p><i>Fazele utilizării pachetelor de programe</i></p> <p><i>Protecția pachetelor de programe</i></p>
--	---

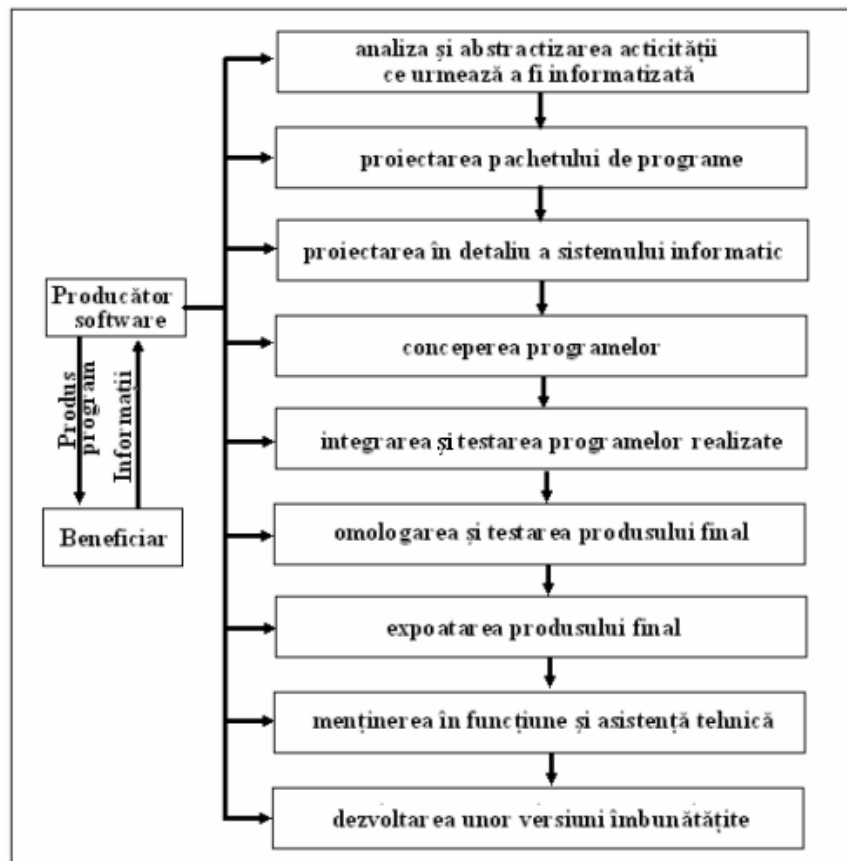
### Etapele de realizare a pachetelor de programe

Realizarea unui pachet de programe este dificilă datorită utilizării unor etape caracterizate de activități specifice, în mod gradat și succesiv [DAV].

De aici deducem că există un proces complex de realizare și întreținere a pachetelor program ce poate fi structurat în următoarele etape tipice:

- analiza și abstractizarea activității ce urmează a fi informatizată;

- proiectarea pachetului de programe;
- proiectarea în detaliu a sistemului informatic;
- conceperea programelor;
- integrarea și testarea programelor realizate;
- omologarea și testarea produsului final;
- exploatarea produsului final;
- menținerea în funcțiune și asistență tehnică;
- dezvoltarea unor versiuni îmbunătățite.



Ciclul de viață al pachetului de programe



Primele șase etape (analiza și abstractizarea activității ce urmează a fi informatizată; proiectarea pachetului de programe; proiectarea în detaliu a sistemului informatic; conceperea programelor; integrarea și testarea programelor realizate; omologarea și testarea produsului final;) reprezintă **stadiul de concepere** al produsului program.

Următoarele trei etape (exploatarea produsului final; menținerea în funcțiune și asistență tehnică; dezvoltarea unor versiuni îmbunătățite) reprezintă **stadiul de exploatare** și menținere în funcțiune al produsului program.

## **1. Stadiul de concepere**

**1. Analiza și abstractizarea activității** ce urmează a fi informatizată - permite identificarea cerințelor globale solicitate de pachetul de programe. În mod concret de urmărește:

- cadrul legislativ sub incidența căruia se declanșează procesele de prelucrare a activității de informatizat;
- analiza operațiilor și regulilor ce descriu maniera de executare a operațiilor;
- identificarea și analiza datelor de intrare și ieșire precum și a structurii ce va fi folosite de pachetul de programe;

- identificarea algoritmilor de calcul sau a modelelor matematice, precum și a metodelor de rezolvare;
- stabilirea cerințelor hardware și software necesare pentru realizarea și funcționare pachetului de programe;
- realizarea unei previziuni de integrare informațională
- reflectarea primară a entităților, proprietăților și relațiilor dintre activitățile de informatizat cu ajutorul modelului conceptual de date specific activității;

**2. Proiectarea pachetului de programe** este etapa în care se desfășoară proiectarea în ansamblu. Această etapă este caracterizată de următoarele acțiuni:

- definirea modului conceptual propriu de date, aferent prelucrărilor produsului program prin specificarea entităților ce sunt necesare li a relațiilor dintre acestea;
- stabilirea arhitecturii produsului program;
- specificarea interfețelor dintre aplicațiile informatice precum și a celor cu utilizatorii;
- proiectarea la nivel logic a componentelor programului;
- stabilirea modelului general de organizare și funcționare a prelucrărilor pachetului de programe

prin reflectarea proceselor, operațiilor, evenimentelor și sincronizărilor necesare.

**3. Proiectarea în detaliu a sistemului informatic** presupune proiectarea tehnică prin structurarea logică a acestuia în unități funcționale și unități de prelucrare, simultan cu definirea elementelor ce urmează:

- definirea modelului logic de date asociat produsului prin prisma conceptelor modelului relațional;
- definirea procedurilor, fazelor și prelucrărilor automate;
- specificarea relațiilor statice și dinamice dintre programe;
- stabilirea modelului de date conform SGBD-ului utilizat;
- gruparea unităților funcționale și a celor de prelucrare conform evenimentelor de declanșare a execuției acestora.

**4. Conceperea programelor** conține totalitatea activităților de realizare a prevederilor automate și constă din:

- structurarea unității de prelucrare în module funcționale și multifuncționale;
- descrierea funcției fiecărui modul prin intrări, ieșiri și parametri;

- descrierea interfețelor dintre module, specificarea condițiilor de lansare în execuție, și terminare a execuției precum și înlănțuirea prelucrărilor;
- codificarea unităților de prelucrare și a modulelor prin transpunerea algoritmilor în modele matematice;
- generarea datelor ce urmează a fi testate, executarea testelor și interpretarea rezultatelor;

**5. Integrarea și testarea programelor realizate** este formată din activități tehnice de realizare a elementelor care asigură interfețele dintre procedurile de execuție pe de o parte, precum și pe cele dintre utilizatori și pachetul de programe. Sunt efectuate activitățile ce urmează:

- analiza statică și dinamică a integrării datelor și procedurilor;
- stabilirea cazurilor de test;
- specificarea procedurilor de test pentru acceptanță;
- construirea bibliotecii din componentele software ca fișiere executabile.

**6. Omologarea și testarea produsului final** constă din verificarea produsului de organisme specializate. Ultimul pas este acordarea dreptului de proprietate de către Institutul Național pentru standarde și Brevete.

---

---

## **2. Stadiul de exploatare**

1. **Exploatarea produsului** se traduce prin activități de utilizare precum și întreținerea operativă a respectivului produs. Aceasta înseamnă:

- asigurarea utilizării produsului program la parametrii proiectați;
- întreținerea produsului prin înlăturarea anomaliilor, modificarea funcțiilor;
- actualizarea bibliotecii de componente software ca fișiere executabile.

2. **Mentținerea în funcțiune și asistență tehnică** se desfășoară de firma producătoare pe baza unui contract de service.

3. **Dezvoltarea unor versiuni îmbunătățite** apare în cazul modificărilor legislative, a utilizării altor sisteme de operare, SGBD-uri, etc.

### **Procese tehnologice de realizare a pachetelor de programe**

Prin proces tehnologic de realizare înțelegem: **specificarea activităților** necesare prin intermediul cărora se realizează obiective concrete, **stabilirea echipei**, programarea sarcinilor și evaluarea efortului de realizare.

În cadrul ciclului de viață al unui produs distingem următoarele tipuri de activități:

- activități de tip managerial: elaborarea și monitorizarea proceselor de proiectare, realizare, omologare și înregistrare a produsului program; programarea activităților de realizare; asigurarea resurselor necesare; organizarea elaborării produsului program; coordonarea și direcționarea tuturor persoanelor sau echipelor.
- activități de tip tehnic de realizare: identificarea și determinarea cerințelor funcționale; proiectarea funcțională și tehnică; proiectarea programelor și modulelor.
- activități de tip auxiliar de realizare: pregătirea datelor de test; efectuarea tuturor testelor pentru atingerea obiectivelor finale;
- activități de utilizare efectivă a pachetului de programe: menținerea produsului program în exploatare efectivă la firma beneficiară; modificarea unor componente software pentru asigurarea acceptării produsului de beneficiar;
- activități de întreținere: oferirea de asistență tehnică operativă la sediul beneficiarului pe parcursul exploatării curente a produsului; îmbunătățirea performanțelor produsului informatic prin modificări software;

- activități de dezvoltare: elaborarea de noi versiuni ale programului;

Principalele teste care se efectuează pentru evaluarea funcționării unei componente software sunt:

- **Testul individual** - acesta verifică buna funcționare a componentelor software din cadrul produsului testat;
- **Testul de integrare al componentelor software** - este un șir de verificări progresive a elementelor software;
- **Testul de integrare în sistemul informatic gazdă** - verifică funcționarea produsului program și a componentelor sale în structura întregului sistem;
- Testul de anduranță - verifică stabilitatea în timp a produsului și a componentelor software;
- Testul de acceptanță a beneficiarului - verifică dacă produsul program corespunde cerințelor prezentate de beneficiar (cerințe de ordin tehnic, economic, uman, etc.).

### **Fazele utilizării pachetelor de programe**

Fazele utilizării pachetelor de programe se desfășoară pe parcursul a cinci etape de prelucrare ( $t_1...t_5$ ) prin care se gestionează datele corespunzătoare perioadei de timp precedente ( $\theta - 1$ ) și perioadei curente ( $\theta$ ).

Pe parcursul acestor etape, pachetul de programe este utilizat de utilizatorul uman prin emiterea și recepționarea de informații privind:

- directoare și unități fizice sursă;
- directoare și unități fizice destinație;
- mesaje de prelucrare;
- parole;
- opțiuni de lucru;
- parametri;
- ferestre de afișare.

În mod obișnuit, pachetul de programe este livrat ca un executabil ce înglobează o bibliotecă sursă direct executabilă (\*.exe sau \*.com). Colecția de date este gestionată prin:

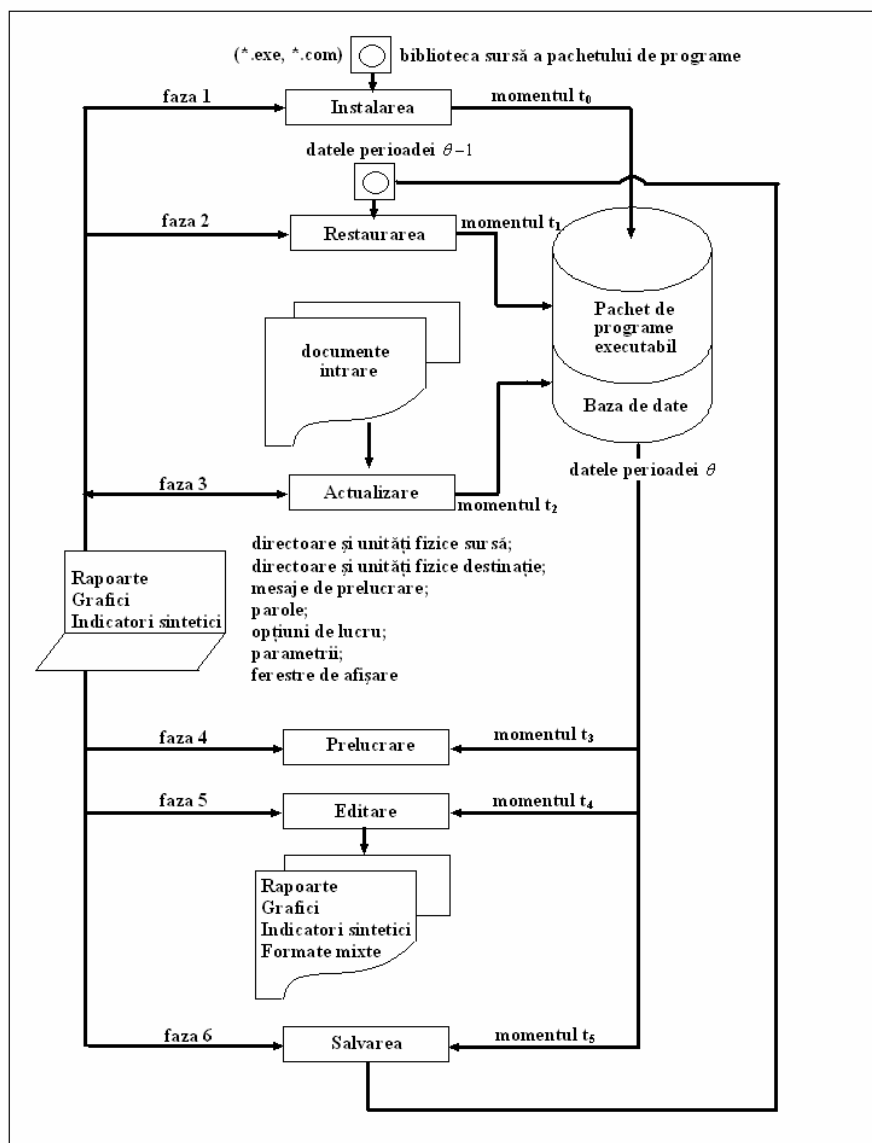
- baze de date, gestionate prin SGBD de tip relațional;
- procesoare de tabele care asigură manipularea structurilor de date.

Principalele faze ale utilizării pachetelor de programe presupun următoarele prelucrări specifice:

1. Instalarea pachetului de programe care este realizată la momentul  $t_0$  prin care este copiată biblioteca sursă a produsului pe sistemul gazdă de calcul.



- 
2. Restaurarea datelor care este asigurată la momentul  $t_1$  și devine utilizabilă numai după prima executare cu date reale a pachetului program.
  3. Actualizarea colecției de date ce este specifică momentului  $t_2$  și constă în operații de adăugare, modificare, inserare și ștergere a datelor din baza de date pe baza celor prelucrate din documentele de intrare specifice programului.
  4. Prelucrarea efectivă a datelor se desfășoară la momentul  $t_3$  și asigură realizarea tuturor calculelor.
  5. Editarea datelor de ieșire se efectuează la momentul  $t_4$  și presupune listarea acestor date pe ecran sau tipărirea la imprimantă sub formă de rapoarte, liste, grafice, indicatori sintetici, formate mixte.
  6. Salvarea datelor prelucrate la momentul  $\theta$  este asigurată în momentul  $t_5$ .



Etapile utilizării pachetelor de programe [DAV]

### Protecția pachetelor de programe

Protecția datelor din cadrul pachetelor de programe este necesară pentru asigurarea securității și confidențialității acestora împotriva accesului neautorizat.

Pentru realizarea protecției pachetelor de programe se recurge la:

- partajare la nivel fizic a componentelor logice pentru controlul accesului la componentele fizice;
- partajare la nivel logic a componentelor în scopul controlului efectelor diferitelor lucrări;
- acordarea drepturilor de acces pentru fiecare utilizator;
- acordarea drepturilor de acces utilizatorilor la nivel de grup de lucru.

Din punct de vedere al realizării, protecția este realizabilă prin trei modalități:

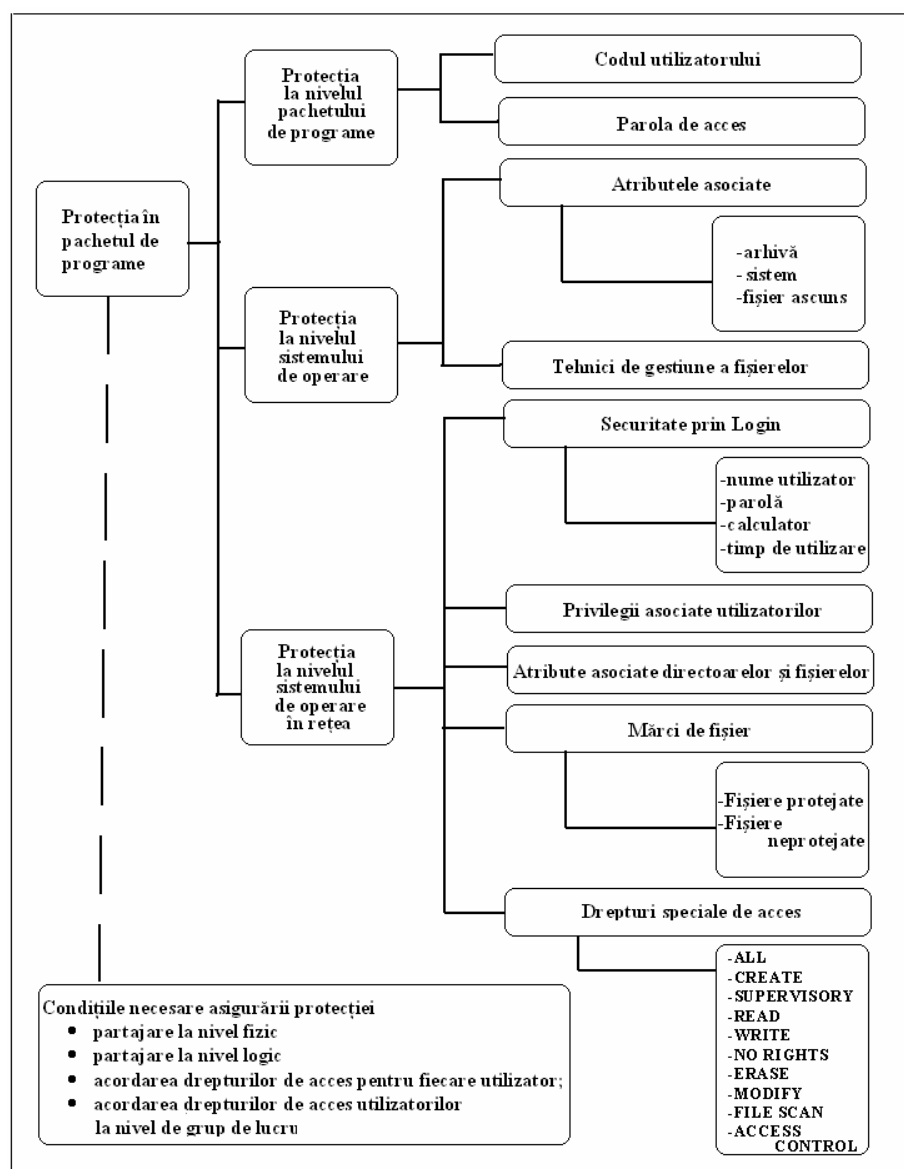
- prin pachetul de programe;
- prin sistemul gazdă;
- prin sistemul de operare în cadrul rețelelor de calculatoare.

**Protecția la nivelul pachetului de programe** poate fi realizată prin mai multe feluri: printr-un cod al utilizatorului care își prezintă identitatea; prin parolă care este un cuvânt ce este cerut de pachetul de programe pentru a se putea lansa procedura de lucru.

**Protecția prin sistemul gazdă** se realizează prin cadrul atributelor fișierelor (fișier de citire, fișier de arhivă, fișier de sistem, fișier ascuns), prin tehnici de gestiune a fișierelor (copiere, ștergere, refacere).

**Protecția prin sistemul de operare** în cadrul rețelelor de calculatoare se realizează prin: securitate de tip LOGIN, privilegii

acordate utilizatorilor, prin atribute asociate directoarelor și fișierelor, prin mărcile de fișier (care clasează fișierele în partajate și nepartajate în funcție de restricțiile programului), prin drepturi de acces pentru fiecare utilizator sau grup de utilizatori (ALL, CREATE, SUPERVISORY, READ, WRITE, NO RIGHTS, ERASE, MODIFY, FILE SCAN, ACCESS CONTROL).



Protecția pachetelor de programe [DAV]



**Intrebări:**

*Care sunt etapele de realizare a pachetelor de programe?*

*În ce constă stadiul de concepere?*

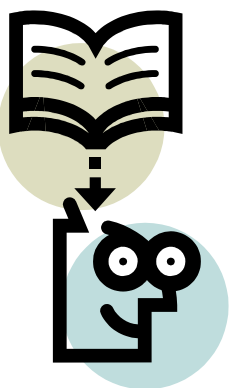
*În ce constă stadiul de exploatare?*

*Care sunt procesele tehnologice de realizare a pachetelor de programme?*

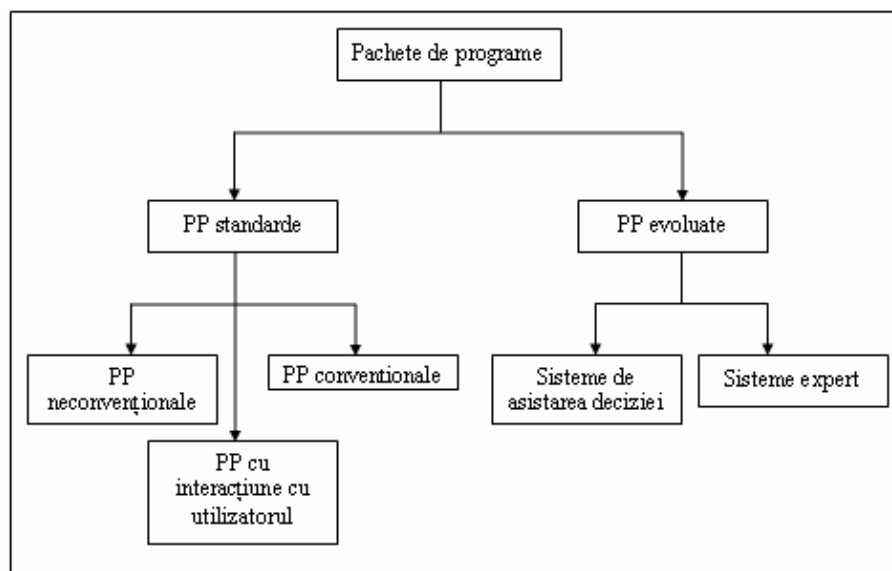
*Care sunt fazele utilizării pachetelor de programe?*

*Ce se înțelege prin protecția pachetelor de programe?*

## CAPITOLUL 3. EVOLUȚIA SISTEMELOR INFORMATICE

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li><i>Stadiile evolutive ale sistemelor informatice</i></li><li><i>Prezentarea sistemelor informatice evolute</i></li><li><i>Prezentarea principalelor caracteristici ale sistemelor informatice de contabilitate și de resurse umane</i></li></ul>
---	---

Din punctul de vedere al evoluției sistemelor informatice se pot observa două mari categorii de clase de produse informatice: clasa pachetelor de programe standard și clasa pachetelor de programe evolute.



Stadiile evolutive ale PP

PP neconvenționale sunt considerate PP închise. PP evaluate/neconvenționale care prelucrează cunoștințe, având la bază tehnicile de inteligență artificială pot fi de două categorii: sisteme de asistarea deciziei și sisteme expert.

**Conceptul de sisteme de asistare a deciziei** poate fi abordat diferit, în funcție de tehnologia utilizată, tipul de "asistare a deciziei" adus, gradul de interactivitate, tipul de utilizare și natura utilizatorilor, conform [URS02]. În funcție de aceste criterii se pot distinge în cadrul sistemelor de asistare a deciziei următoarele cazuri particulare:

- sisteme interactive de asistare a deciziei
- sisteme informatice pentru decidenți

Sistemele interactive de asistare a deciziei se bazează pe un sistem interactiv, bazat pe dialogul om-mașină. La fiecare etapă a procesului calculatorul afișează un rezultat al evaluării procesului de

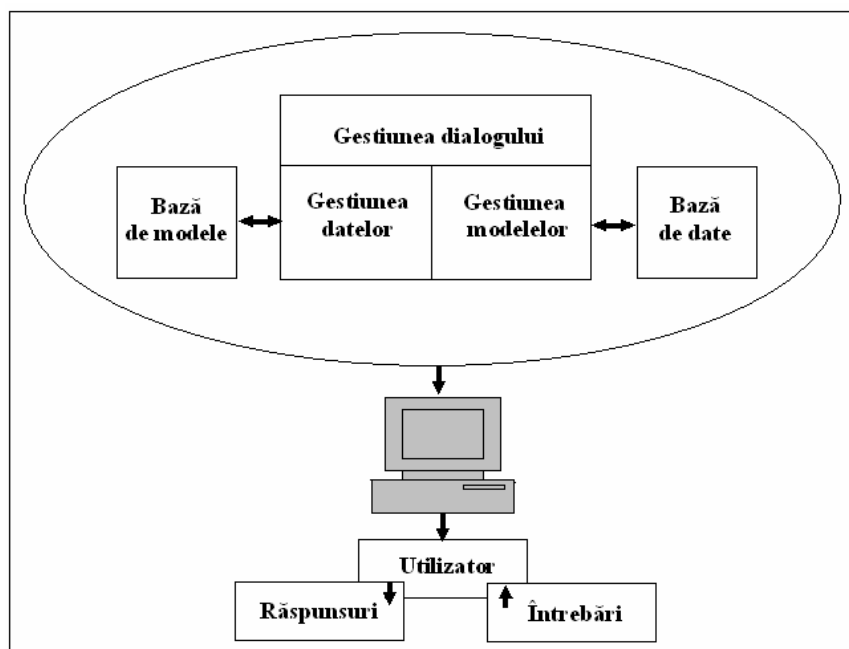
decizie, și plecând de la acesta, decidentul alege sau nu, conținutul etapei următoare și dispune măsurile corespunzătoare.

Un sistem interactiv de asistare a deciziilor se definește prin următoarele elemente[URS02]:

- este un sistem informatic, asistat de calculator;
- furnizează asistență decidenților pentru probleme care nu sunt în totalitate structurate;
- combină judecata umană cu prelucrarea automatizată a informației;
- este un sistem în care controlul derulării procesului de decizie revine decidentului, în cadrul unei cercetări de tip euristic;
- este un sistem care influențează mai mult eficacitatea procesului de decizie (adoptarea deciziei care răspunde obiectivelor stabilite) decât eficiența procesului decizional (adoptarea unor decizii satisfăcătoare).

Un sistem interactiv de asistare a deciziilor este capabil să îndeplinească următoarele operații: - să furnizeze prezentări pertinente ale fenomenului care generează problema de decizie; - să memoreze aceste prezentări , prezentând modelele prin care sunt descrise datele, rezultatele intermediare și finale; - să realizeze prelucrări complexe ale informațiilor pe baza modelelor incluse; - să interpreteze ordinele (comenzile) furnizate de decident în fiecare fază a procesului euristic.





Componentele unui sistem interactiv de asistare a deciziilor

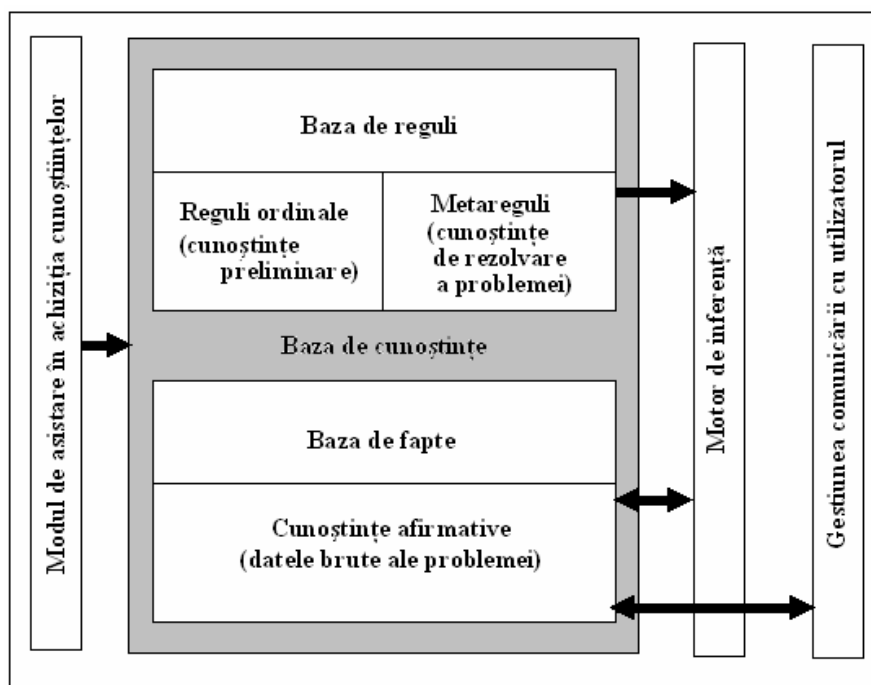
Un sistem interactiv de asistare a deciziilor este alcătuit din trei elemente fundamentale: - o bază de date, cu un sistem de gestiune a bazei de date; - o bază de modele, cu un sistem de gestiune a bazei de modele; - o funcție de gestiune a dialogului om-mașină.

**Sistemul expert** este un program informatic, capabil să stimuleze raționamentul unui expert uman, într-un domeniu specific de cunoaștere.

În funcție de situație, un sistem expert poate fi văzut ca [URS02]:

- un sistem de decizie, în care se rețin alegerile propuse de sistemul expert;
- un sistem de asistare a deciziei, prin care decidentul se bazează pe recomandările formulate de sistem, dar se poate și abate de la acestea;

- un sistem e asistare a studierii, care permite transmiterea cunoștințelor dintr-un domeniu specific, de la un expert uman la altul mai puțin pregătit.



Structura unui sistem expert

Structura unui sistem expert pune în evidență particularitatea acestuia, în raport cu sistemele informatice care prelucrează date, și anume, separarea cunoștințelor referitoare la problemă (descrise într-o bază de cunoștințe), de mecanismul de exploatare al acestor cunoștințe (realizat printr-un motor de inferență).

**Domenii de utilizare** a sistemelor expert [URS02]:

1. Diagnostic, interpretare. Sistemul expert pornește de la un anumit număr de semnale și reușește să caracterizeze

---

---

o stare, o situație, atribuind un sens și o semnificație semnelor primite.

2. Menținere, remedieri. Sistemele expert din acest domeniu au funcții suplimentare celor prezentate anterior, în sensul că, după ce culeg și interpretează cunoștințele, pot formula observații și acțiuni în vederea remedierii disfuncționalităților observate.
3. Previziune, planificare. Plecând de la observarea unei situații date, sistemul propune o imagine a situațiilor viitoare sau poate oferi chiar opțiuni pentru planificare, în funcție de obiectivele fixate.
4. Activități de concepție. sistemul expert combină diferite restricții specifice domeniului de utilizare, precum și consecințele lor asupra funcționalităților care se prevăd pentru a concepe un produs.

---

---

## **Sisteme informatice contabile și financiare**

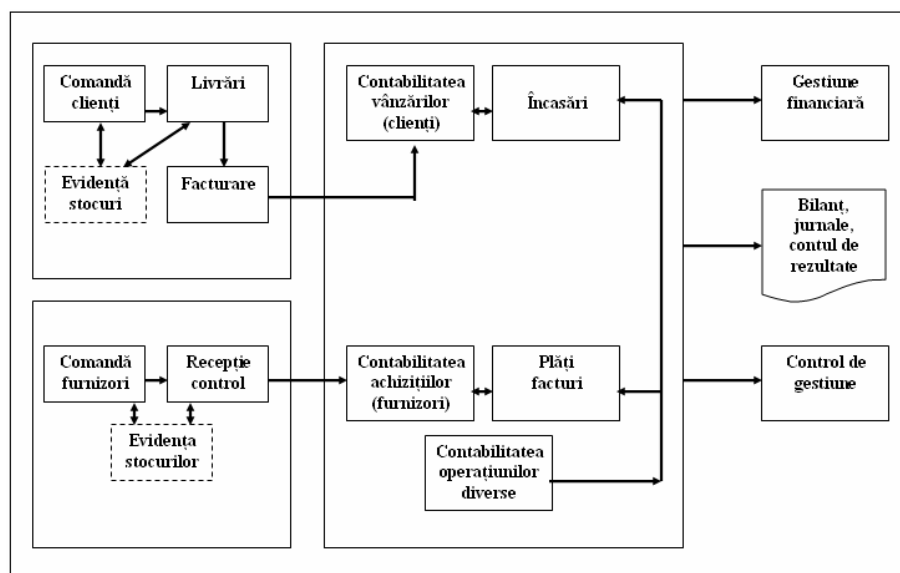
Datorită caracteristicilor contabilității, sistemele informatice contabile și financiare reprezintă exemplul clasic al unui sistem informatic.

Activitatea contabilă este formată din tranzacții numeroase, operațiuni care se repetă, caracter ridicat de veridicitate al rezultatelor, fapt care a condus la apariția pe acest plan pentru prima dată a sistemelor economice informatice.

Sistemele informatice contabile și financiare presupun următoarele componente:

- prelucrarea tranzacțiilor contabile;
- subsistemul de gestiune financiară
- subsistemul de control de gestiune

Prelucrarea tranzacțiilor contabile este asigurată printr-un modul informatic distinct, unde principalele funcții sunt legate de achiziția facturilor de materii prime, materiale, și vânzarea produselor de firmă.

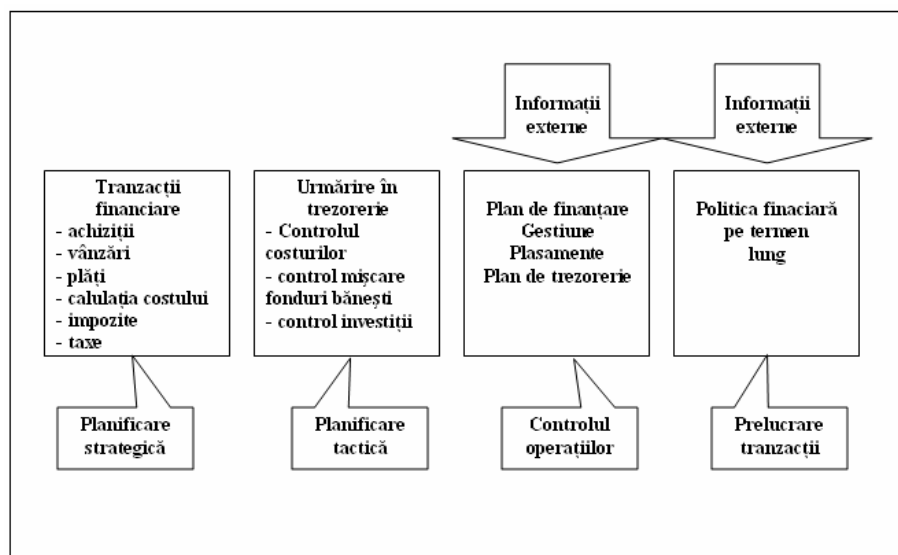


Schema contabilității generale [URS02]

Funcțiile de bază ale programelor de contabilitate generală sunt emiterea facturilor către clienți, pentru produsele expediate, încasarea facturilor de la furnizorii de materii prime, materiale, consumabile, etc.

Deci odată ce sunt stabilite regulile de înregistrare ale tranzacțiilor, pachetul de programe funcționează conform unor modele, ce au la bază de date contabile. Prin această modalitate se poate răspunde unor cerințe punctuale cu caracter nelimitat. Eficiența pachetului de programe crește pe măsură ce crește gradul de exploatare al acestei baze de date, prin intermediul altor module informatice precum cel al gestiunii financiare și al controlului bugetar.

Gestiunea financiară este una din activitățile de bază în cadrul întreprinderii, prin care se încearcă cunoașterea situației actuale, precum și posibilitatea previzionării fluxurilor financiare.



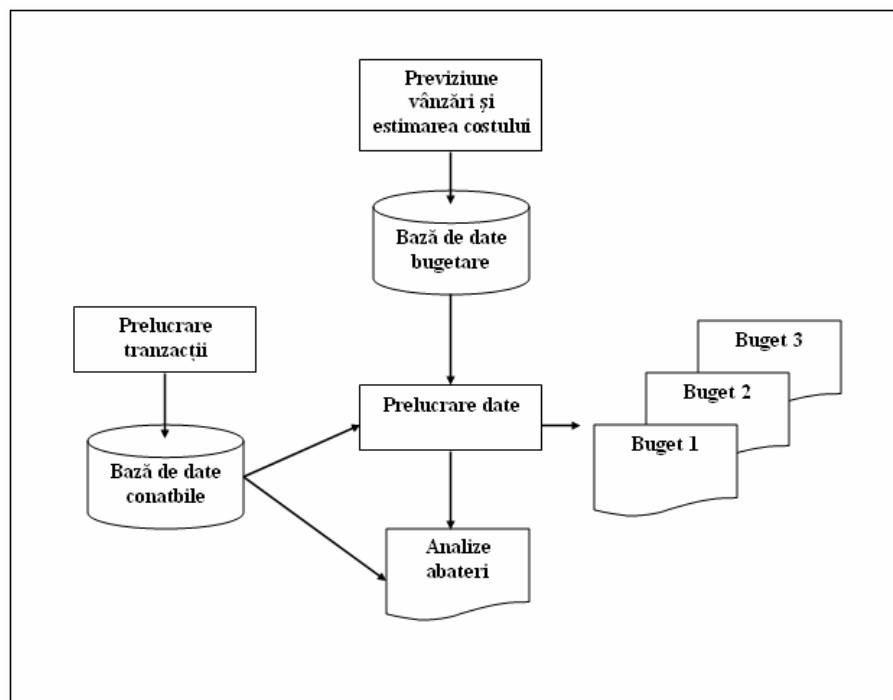
Elementele unui modul informatic pentru gestiune financiară [URS02]

Prelucrarea tranzacțiilor din gestiunea financiară este alimentată de date provenite din contabilitatea generală, referitoare la generarea creanțelor și datoriilor întreprinderii. Prin cunoașterea acestor date se pot afla mișcările de trezorerie, nivelul intrărilor și ieșirilor, expirarea termenelor de plată și de încasare.

În cadrul nivelurilor de prelucrare a tranzacțiilor și de control al operațiilor sursele de date provin cu precădere din contabilitatea generală. Spre deosebire de aceste niveluri, în cazul nivelurilor tactic și strategic, aceste surse de informații sunt completate cu informații din mediul extern firmei, cum ar fi informații de pe piețele financiare.

Prin aceste informații cumulate, sistemul informatic este capabil să răspundă unor întrebări cum ar fi costul finanțărilor, posibilități de plasament financiar, iar din punct de vedere managerial duc la creșterea posibilităților de comunicare.

În cadrul sistemului informatic contabil și financiar se găsește și modulul de contabilitate bugetară care vizează anticiparea evenimentelor financiare viitoare.



Schema modului de control bugetar [URS02]

Modulul de control bugetar se bazează pe date din contabilitatea generală și are scopul de a:

- constata nivelul evoluției indicatorilor economico-financiari;
- previziona evoluția indicatorilor, prin simularea fenomenelor viitoare.

Sistemul informatic financiar contabil asigură integrarea datelor provenite din alte module ale sistemului informatic general, cum ar fi:

gestiunea comercială, gestiunea producției și aprovizionărilor, gestiunea resurselor umane.

Variantele de automatizare a contabilității vizează în special: modul de culegerea a datelor (preluarea datelor, editarea, prelucrarea lor fără intervenția utilizatorului), posibilități de interogare a bazei de date (plecând de la afișarea unor solduri de cont până la funcții de interogare complexe cum ar fi analize pe centre de venituri și de cheltuieli), posibilitatea de editare a unor rapoarte informaționale (care au un format standard, folosite în sistemele de asistare a deciziei).

Factori esențiali care caracterizează un produs informatic de acest gen sunt: **calitatea comunicării** (care trebuie să existe între programele informatice, între modulele acestora, fără de care este posibilă apariția unor efecte negative cum sunt: redondanța, încărcarea canalelor informaționale, distorsiunea informației, etc.), **securitatea informațiilor** (datele introduse în contabilitate au un caracter confidențial și pot fi folosite ca probe în instanță, de asemenea un program informatic de acest gen trebuie să releve în orice moment situația patrimonială a întreprinderii). În ceea ce privește securitatea datelor se pot pune o serie de întrebări cum sunt: Cine este autorizat să introducă date? Cine este autorizat să le modifice? Cine are acces la aceste date? Securitatea datelor se realizează de cele mai multe ori prin închiderea bazelor de date, securitate asigurată de parole de acces, imposibilitatea de a modifica un document validat, etc.

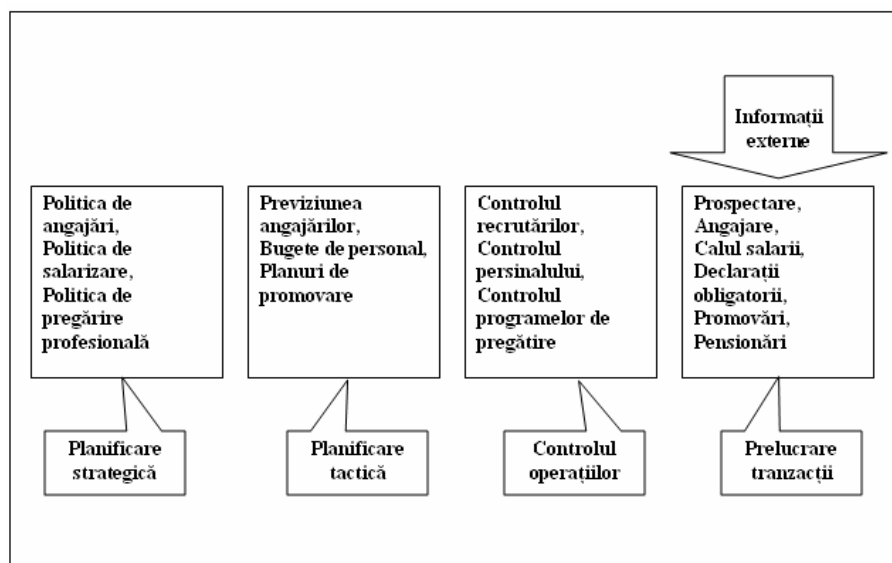


## Sisteme informatice pentru gestiunea resurselor umane

Sisteme informatice pentru gestiunea resurselor umane sunt de o natură mai deosebită deoarece principalul factor asupra căruia de îndreaptă este factorul uman. Individul spre deosebire de alte resurse are o serie de caracteristici particulare, poate cea mai importantă fiind aceea că nu aparține întreprinderii ca element de patrimoniu, ci funcționează pe baza unui contract.

Acest lucru impune anumite restricții asupra sistemelor de gestiune a resurselor umane cum ar fi: necesitatea de a asigura recrutarea, selecția, angajarea, gestionarea angajaților.

Acest flux de resurse umane este inclus în sistemele informatice pentru gestiunea resurselor umane alături de automatizarea operațiunilor de plată către salariați care reprezintă punctul de plecare în gestiunea resurselor umane.

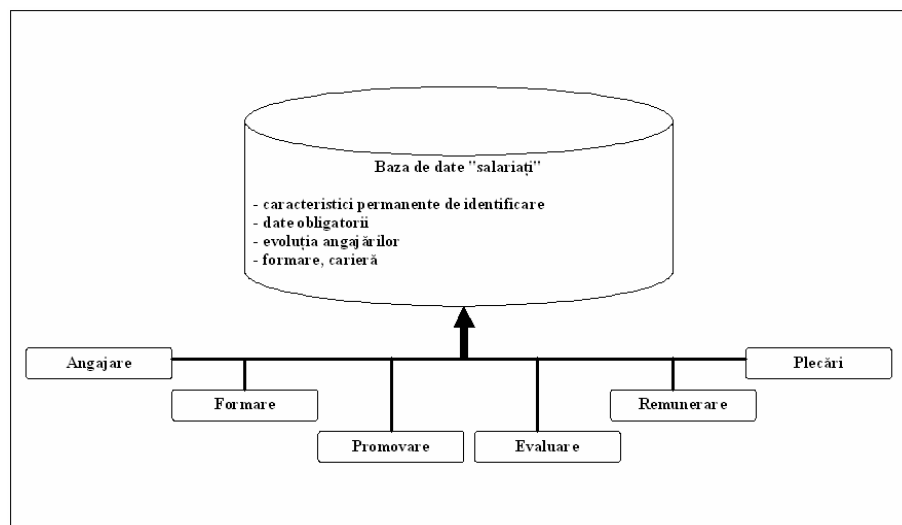


## Elementele unui sistem informatic pentru gestiunea resurselor umane [URS02]

De aici deducem că sistemele informatice pentru gestiunea resurselor umane îndeplinește următoarele funcții principale:

- gestiunea locului de muncă - prin definirea nevoilor specifice, recrutare, angajare;
- gestiunea salarizării - prin politica salarială, bugete de personal, controlul cheltuielilor cu personalul;
- gestiunea valorificării personalului - prin politica de promovare, evaluare și formare.

Asigurarea sistemului informatic pentru gestiunea resurselor umane presupune existența unei baze de date "salariați", care constituie punctul de plecare al sistemului informatic. Astfel, avem:



Prelucrarea datelor în cadrul sistemului informatic de gestiune a salariaților [URS02]

În funcție de gradul de informatizare din firmă, sistemul de gestiune a salariaților poate fi parțial automatizat (cazul cel mai frecvent, când datele obligatorii se păstrează sub formă digitală, iar cele auxiliare se păstrează pe hârtie) și total automatizat (când toate datele și documentele se păstrează în formă digitală, la cerere putând fi afișate, actualizate, modificate).

În cazul firmelor românești, cel mai adesea sistemele informatice de acest gen folosesc la asigurarea obținerii notelor contabile referitoare la activitatea de salarizare, precum și editarea de documente necesare plății diferitelor rețineri.

Sistemele informatice de acest gen pot fi caracterizate de următoarele trăsături: grad ridicat de fidelitate a informațiilor, grad ridicat de confidențialitate, posibilitatea definirii unor indicatori de performanță, plecând de la care se pot construi baze de date informaționale, de pot fi folosite în operațiuni de recrutare, selecție, angajare, promovare, etc.

**Intrebări:**

*Care sunt elementele generale ale pachetelor de programe de gestiune economică?*

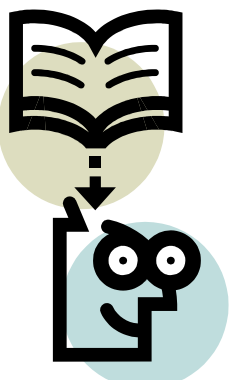
*Care sunt sistemele informatice funcționale?*

*Ce înțelegeți prin sisteme informatice contabile și financiare?*

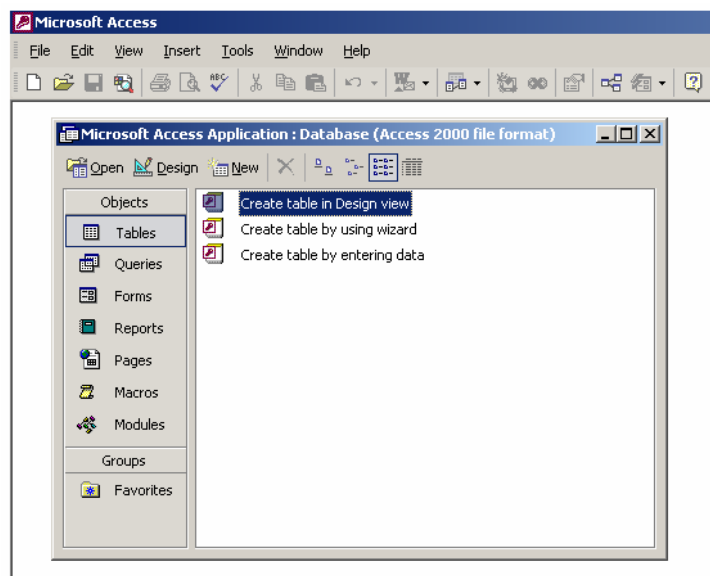
*Ce înțelegeți prin sisteme informatice pentru gestiunea resurselor umane?*

# TEMA 1

## STOCAREA DATELOR ÎN BAZE DE DATE

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li>Crearea unui nou fișier access</li><li>Crearea tabelelor</li><li>Modificarea structurii unei tabele</li><li>Introducerea și vizualizarea datelor</li><li>Realizarea relațiilor între tabele</li><li>Modificarea și ștergerea datelor</li></ul>
--	---

Un fișier Access este format din obiecte cum ar fi: tabele, formulare, queries, rapoarte pagini, macros și module.

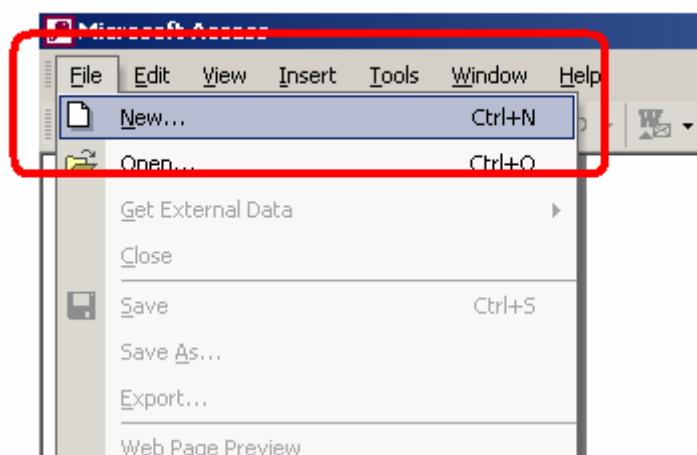


Fereastra de bază - Access

## CREAREA UNUI NOU FIȘIER ACCESS

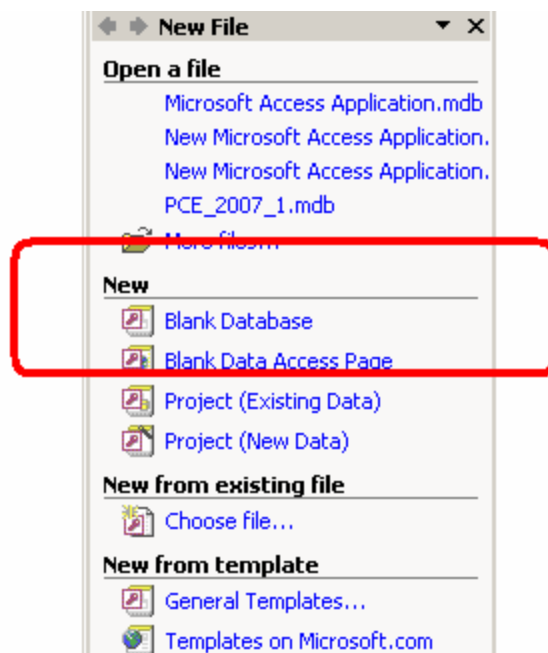
Fișierele Access au extensia .mdb  Microsoft Access Application.mdb .

Pentru crearea unui nou fișier Access se alege în meniul FILE opțiunea NEW.



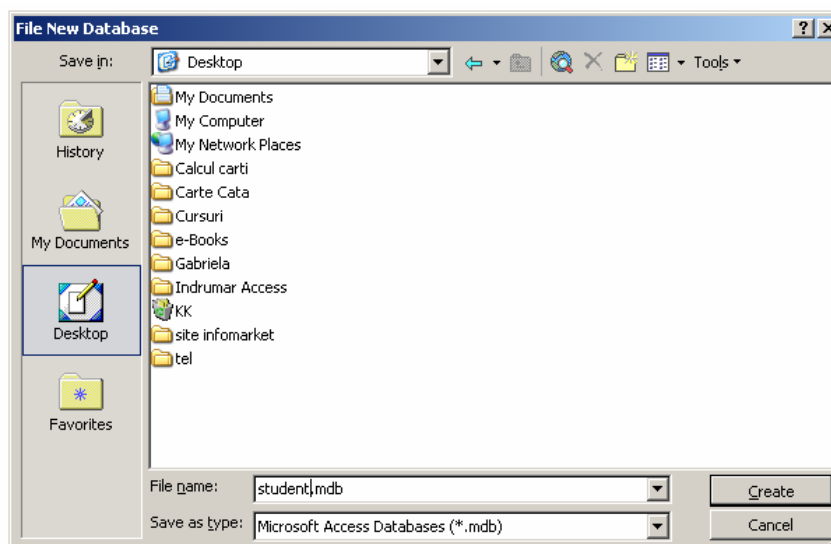
FILE-NEW

Din meniul din partea din dreapta care apare se alege opțiunea **Blank Database**



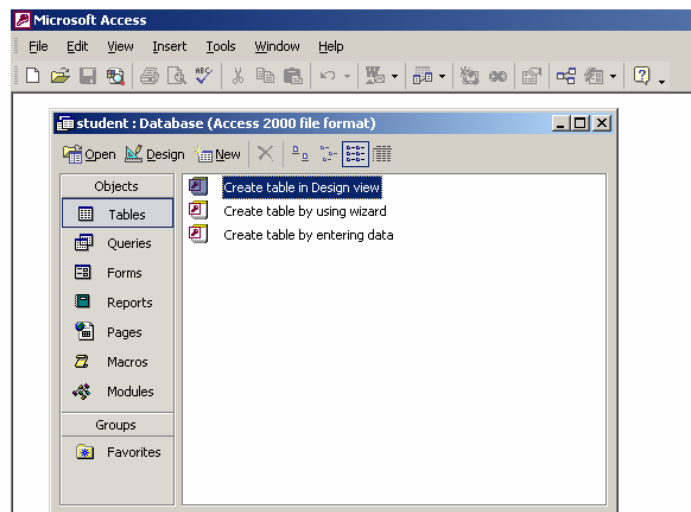
Alegerea opțiunii Blank Database

Următorul pas în crearea bazei de date este alegerea unui nume corespunzător fișierului mdb, precum și alegerea locului de salvare al fișierului pe calculator.



Alegerea numelui și al locului de salvare

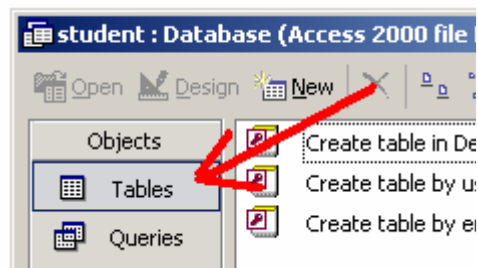
Fișierul Access creat este gata de lucru. În acest moment în fișierul nu conține nici o informație (baze de date, interogări, pagini etc.)



Fișierul student.mdb este creat și gata de lucru

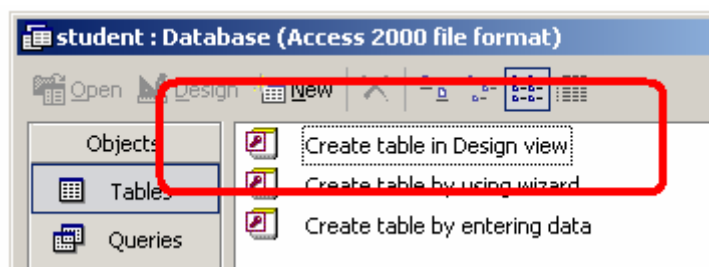
## CREAREA TABELELOR

În primul rând, deoarece dorim să lucrăm la nivel de tabele, vom selecta din panoul de comandă din cadrul meniului de obiecte (OBJECTS) opțiunea TABLES (tabele).



Selectarea opțiunii TABLES din panoul de comandă

Pentru crearea unei tabele **nu** vom utiliza generatoare de tip wizard. Crearea unei tabele nu trebuie să aibă limitări, de aceea din panoul de comandă alegem prima opțiune: CREATE TABLE IN DESIGN VIEW.



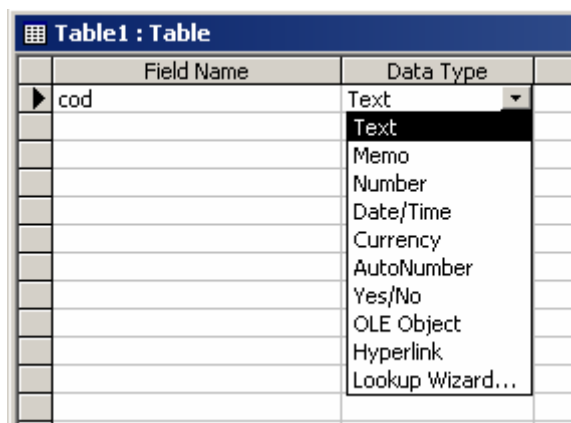
Selectarea opțiunii de creare a unui nou tabel

Pentru a exemplifica crearea de tabele vom crea tabela angajați care are forma:

### Angajați

câmp	tip
cod (CP)	numeric
nume	text
prenume	text
data_nasterii	data
sal_brut	numeric
cod_dep	text

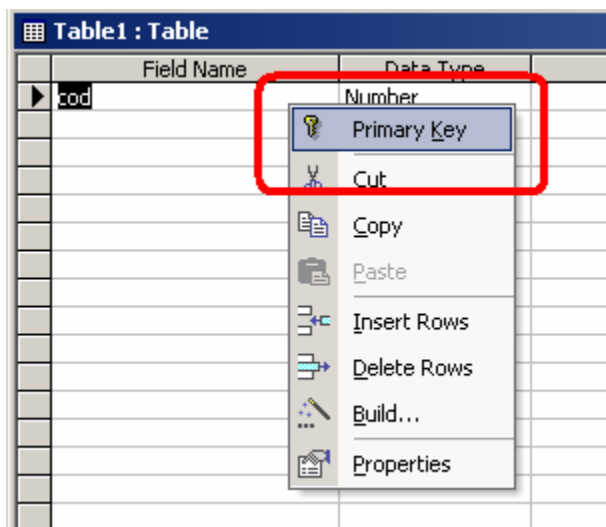
Introducerea datelor se face în fereastra TABLE. La FIELD NAME se introduce numele fiecărui câmp al tabelului după care se alege de la DATA TYPE tipul corespunzător (text, memo, numeric, dată/timp, sumă de bani, număr-automat, da/nu, obiect OLE, hyperlink).



Introducerea câmpurilor și selectarea tipului corespunzător

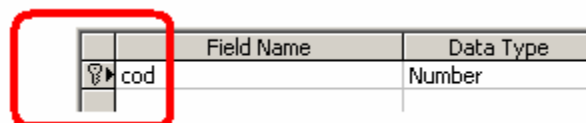


În momentul în care un câmp se dorește a fi setat ca cheie primară, pe linia unde este numele câmpului de da clic dreapta de mouse si se alege opțiunea PRIMARY KEY.



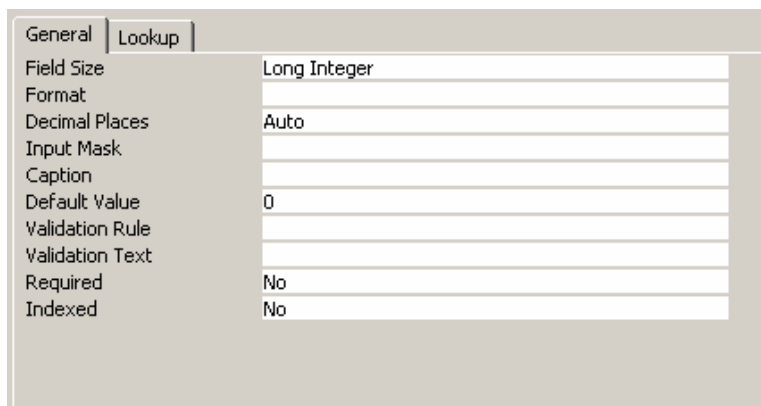
Alegerea opțiunii PRIMARY KEY

La selectarea opțiunii de cheie primară, va apare în dreptul numelui câmpului un simbol de cheie.



Cheie primară

Pentru informații suplimentare privind câmpul respectiv, în partea de jos a ecranului avem următoarele opțiuni:



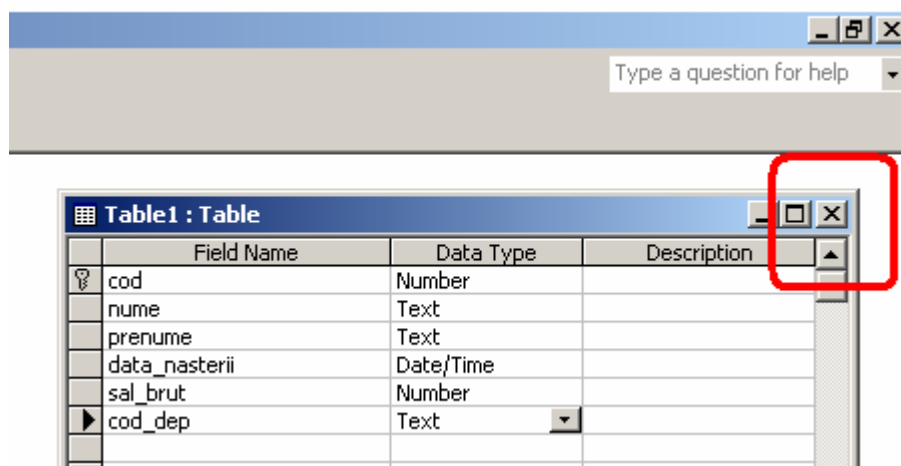
Opțiunile câmpului

În final, tabela angajati va avea următoarea structură:

Table1 : Table		
	Field Name	Data Type
?	cod	Number
	nume	Text
	prenume	Text
	data_nasterii	Date/Time
	sal_brut	Number
▶	cod_dep	Text

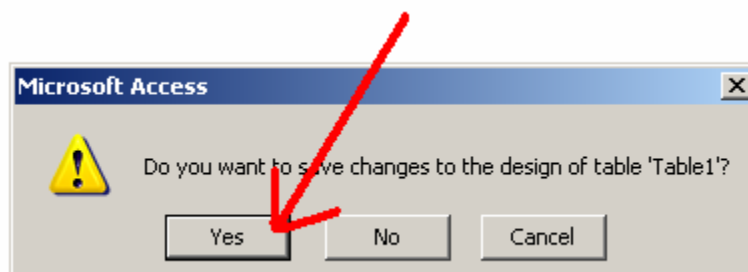
Tabela Angajati

Închiderea configurării tablei se face apăsând simbolul X din dreapta ferestrei TABLE (**ATENȚIE NU ÎNCHIDEREA PROGRAMULUI ACCESS!**).



Închiderea ferestrei de configurare a tablei

Salvarea tablei realizate:

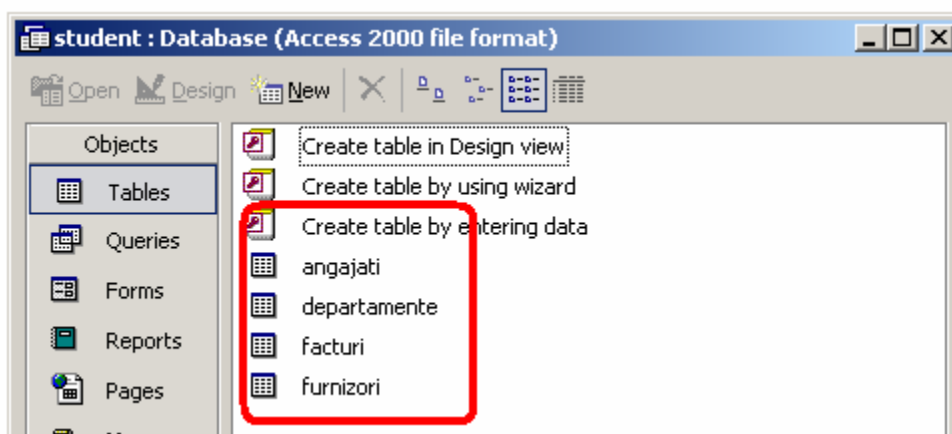


Salvarea tablei

**Numele tabelii**

Denumirea tabelii realizate

În panoul de comandă apar toate tablele realizate, în ordine alfabetică:

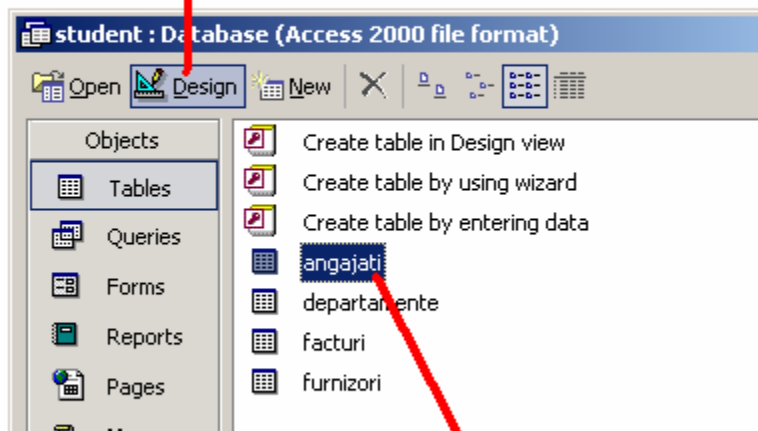


Lista tabelor realizate

**MODIFICAREA STRUCTURII UNEI TABELE**

După realizarea unei tabeli se poate observa că lipsesc unele câmpuri, trebuie modificată denumirile câmpurilor sau tipul lor.

Modificarea structurii se realizează prin selectarea tabelii ce se dorește modificată și apoi alegerea opțiunii DESIGN din panoul de comandă.

**Pasul 2 - se alege opțiunea DESIGN****Pasul 1 - se selectează tabela**

Modificarea structurii unei tabele

Prin acești pași se reintră în fereastra de configurare a tabelului unde pot fi operate toate modificările necesare.

**INTRODUCEREA ȘI VIZUALIZAREA DATELOR**

Pentru a introduce sau vizualiza datele dintr-o tabelă se dă dublu clic pe numele ei din tabloul de comandă.

angajati : Table						
	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
▶					0	

Introducerea / vizualizarea datelor

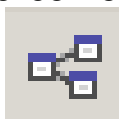
angajati : Table						
	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
	100	Popescu	Ion	20.01.1975	780	conta
	102	Ionescu	Dan	25.06.1980	450	info
	104	Petrescu	Dana	30.09.1970	800	conta
	105	Oprescu	Vasile	12.12.1960	1500	mn
	106	Basescu	Ionela	06.06.1950	1200	mn
					0	

Datele introduse în tabela ANGAJATI

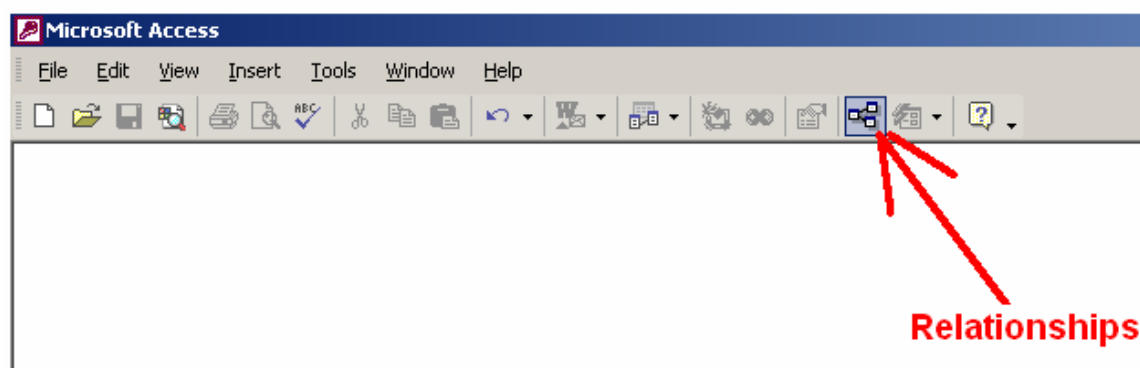
## REALIZAREA RELAȚIILOR ÎNTRE TABELE

Între tabelele existente se pot crea relații pe baza cheilor primare și externe existente.

În bara de comenzi se alege opțiunea RELATIONSHIPS dând

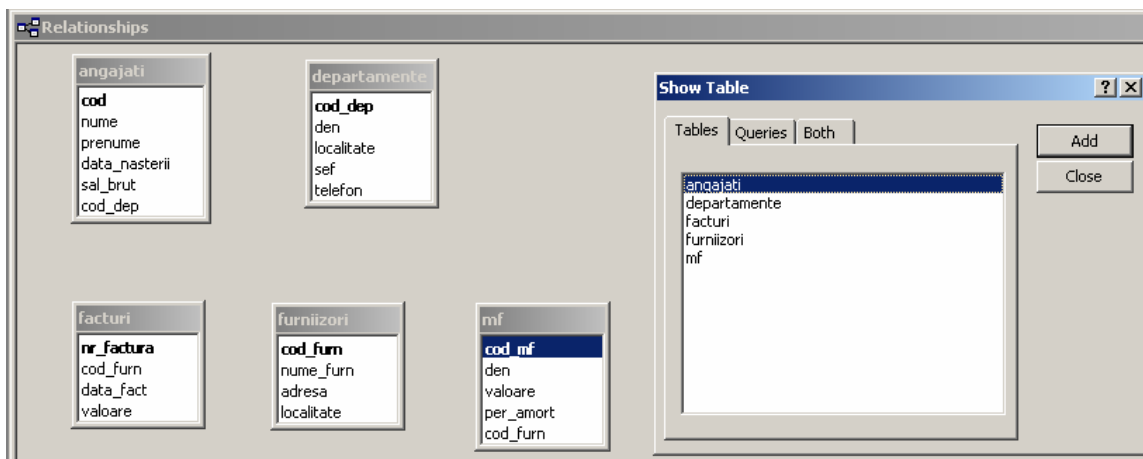


clic pe simbolul



Crearea relațiilor între tabele

Pentru adăugarea tabelelor în fereastra de relații se selectează tabela dorită după care se apasă butonul ADD.



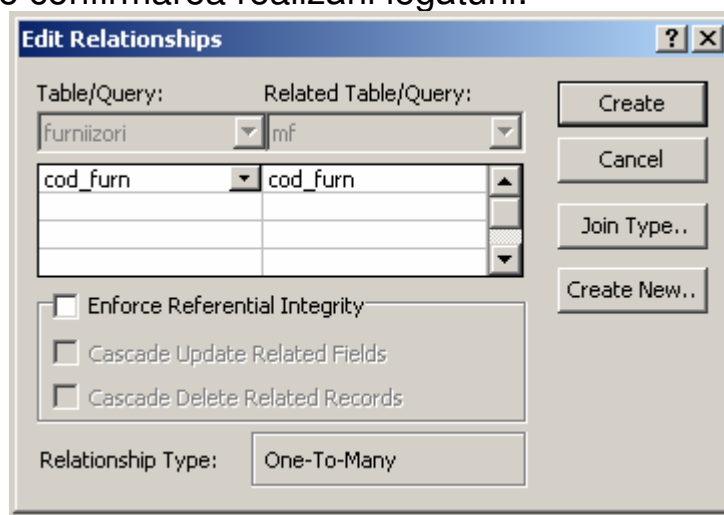
Adăugarea tabelelor în fereastra de relații

Pentru realizarea relațiilor se selectează cheia externă din tabelă și se trage pe cheia primară a tablei cu care se creează relația.



Realizarea unei relații

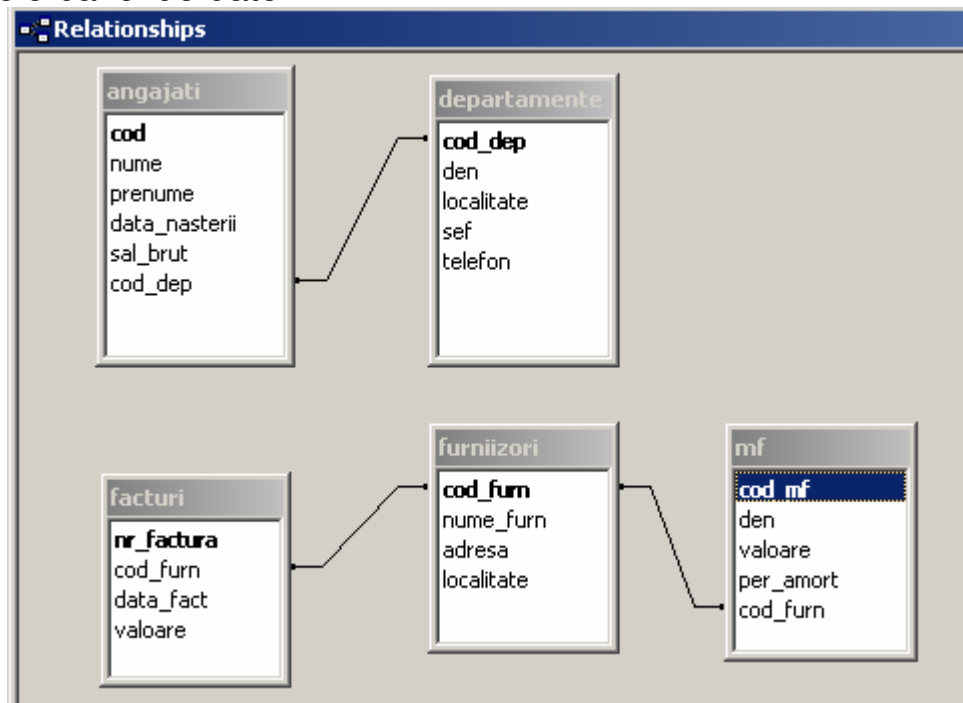
Programul vede legătura între cele două câmpuri din cele două tabele și cere confirmarea realizării legăturii:



Realizarea legăturii

Se procedează la fel cu toate relațiile existente între toate tabelele.

În final se pot vizualiza toate relațiile existente între toate tabelele bazei de date.

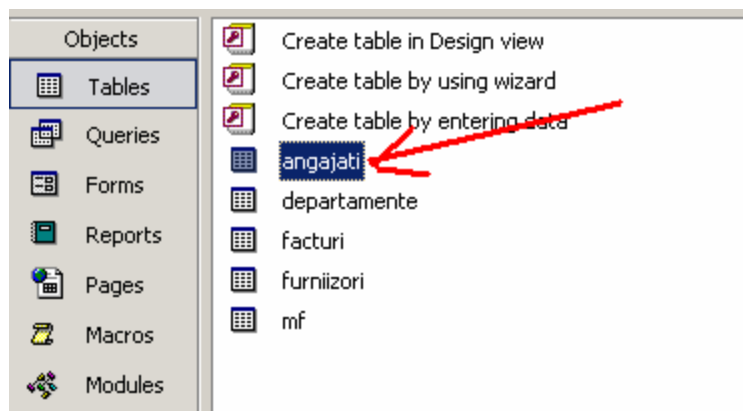


Vizualizarea legăturilor existente în baza de date

## MODIFICAREA ȘI ȘTERGERA DATELOR

Modificarea, inserarea și ștergerea sunt operații care pot fi făcute atât manual (vizualizând datele tabeli) și operând direct cât și pe baza de comenzi scrise sub formă de QUERIES.

Pentru modificarea sau ștergerea datelor dintr-o tabelă se intră în tabela din panoul de comandă, cu dublu click.



Vizualizarea datelor din tabelă

Modificarea datelor este foarte facilă, orice modificare făcută în tabelă memorându-se automat.

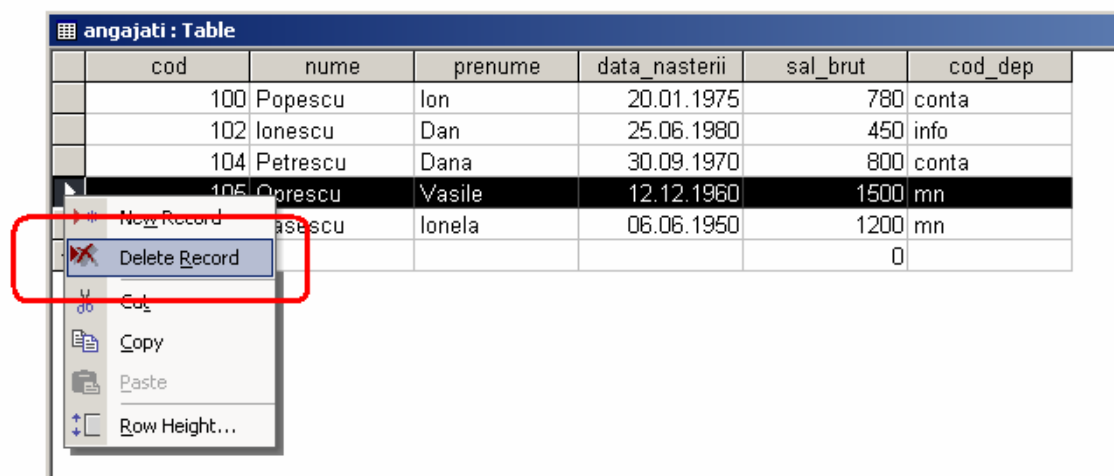
**Vasilica**

angajati : Table						
	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
	100	Popescu	Ion	20.01.1975	780	conta
	102	Ionescu	Dan	25.06.1980	450	info
	104	Petrescu	Dana	30.09.1970	800	conta
▶	105	Oprescu	Vasile	12.12.1960	1500	mn
	106	Basescu	Ionela	06.06.1950	1200	mn
*	0				0	

Modificarea unei valori in tabelă

Pentru ștergerea unei valori din tabelă se dă buton dreapta de mouse pe suprafața liniei ce se dorește a fi ștearsă. Va apare un meniu din care se ale opțiunea DELETE RECORD.

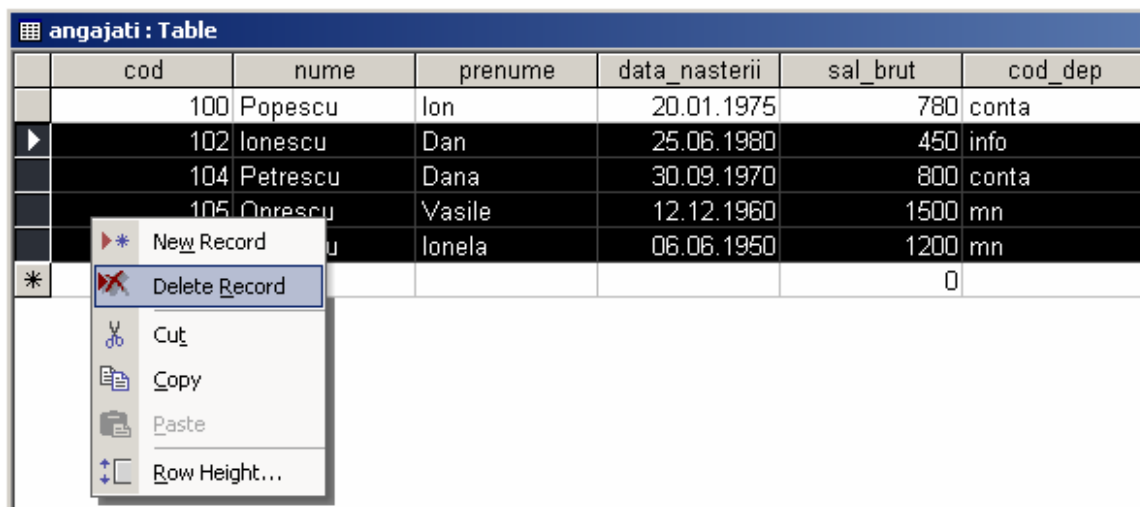




	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
	100	Popescu	Ion	20.01.1975	780	conta
	102	Ionescu	Dan	25.06.1980	450	info
	104	Petrescu	Dana	30.09.1970	800	conta
	105	Onrescu	Vasile	12.12.1960	1500	mn
		asescu	Ionela	06.06.1950	1200	mn
					0	


### Ștergerea unei linii dintr-o tabelă

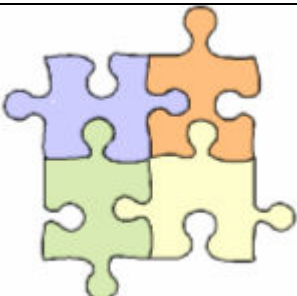
Pentru ștergerea mai multor linii dintr-o tabelă se procedează similar cu mențiunea că înainte trebuie selectate toate liniile ce vor fi șterse.



	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
	100	Popescu	Ion	20.01.1975	780	conta
	102	Ionescu	Dan	25.06.1980	450	info
	104	Petrescu	Dana	30.09.1970	800	conta
	105	Onrescu	Vasile	12.12.1960	1500	mn
		u	Ionela	06.06.1950	1200	mn
*					0	

### Selectarea mai multor linii pentru a fi șterse

	<p style="text-align: center;"><b>Intrebări:</b></p> <p>Care sunt componentele unei tabele?          Ce înseamnă actualizarea conținutului unei tabele? Cum se actualizează conținutul unei tabele?          Prezentați modul de vizualizare a conținutului unei tabele.          Prezentați modul de actualizare a structurii unei tabele.</p>
---	---

	<p style="text-align: center;"><b>Studiu individual</b></p> <p>Aplicația 1: Creare și actualizare tabelă.</p>
---	---

## APLICAȚIA 1

Creați o tabelă următoarea structură:

Nume	C	30
Prenume	C	30
Cod	N	3
Oras	C	20
Medie	N	6 ; 2
Admis	C	

și cu următorul conținut :

Nume	Prenume	Cod	Oras	Medie	Admis
Ionescu	Maria	1	Iasi	9.27	
Vlad	Mirela	8	Deva	9.50	
Georgecu	Ana	5	Cluj	8.85	
Popescu	Denisa	4	Arad	9.33	
Rauta	Andreea	6	Deva	9.50	
Stanciu	Roxana	2	Deva	9.10	

---

---

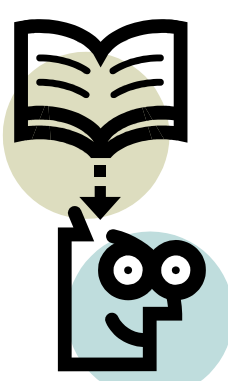
Radulescu	Ruxandra	3	Cluj	9.29	
Matei	Iuliana	7	Ploiesti	9.60	

- a) Să se adauge un nou câmp, denumit Admis.
- b) Se presupune că doar candidatele cu medii peste 9.25 au fost admise. Să se completeze valorile corespunzătoare pentru câmpul Admis.
- c) Candidatele din Deva sunt de fapt din București. Să se modifice valorile corespunzătoare din câmpul oraș.
- d) Să se adauge înregistrarea:

Cretulescu	Alexandra	9	Orastie	8.00	F
------------	-----------	---	---------	------	---

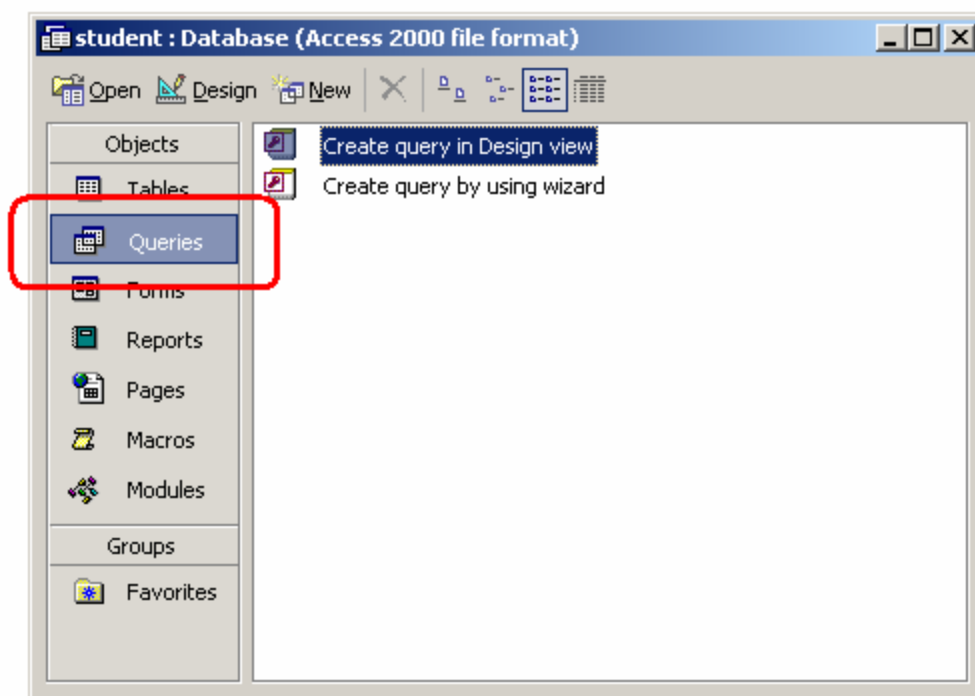
# TEMA 2

## CREAREA INTEROGĂRILOR

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li>Insert - adăugarea de linii în tabelă</li><li>Modificarea unei comenzi</li><li>Delete - ștergerea înregistrărilor</li><li>Update - modificarea datelor dintr-un câmp</li></ul>
--	---

Interogările sau queries pot fi făcute pentru una sau mai multe tabele existente în baza de date.

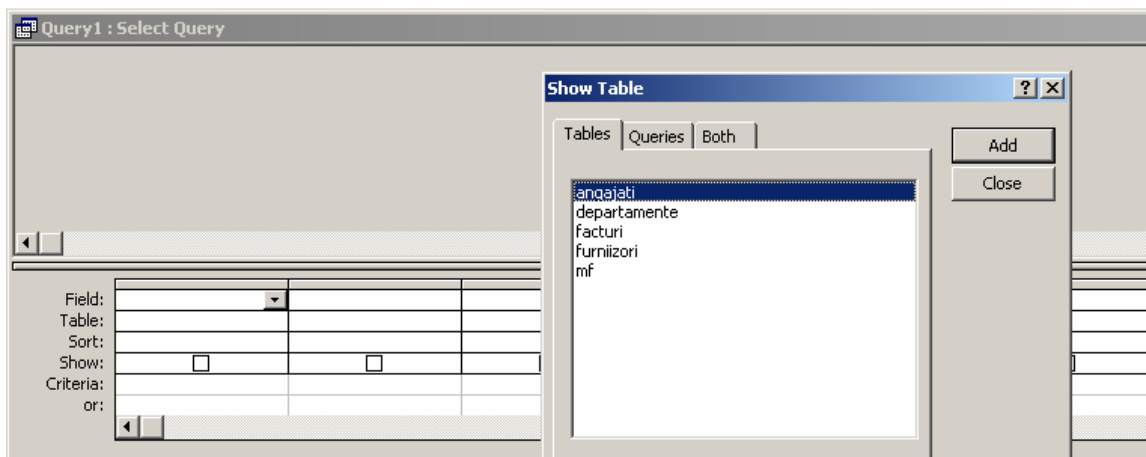
Pentru început trebuie selectată din cadrul meniului OBJECTS opțiunea QUERIES.



Selectarea opțiunii QUERIES

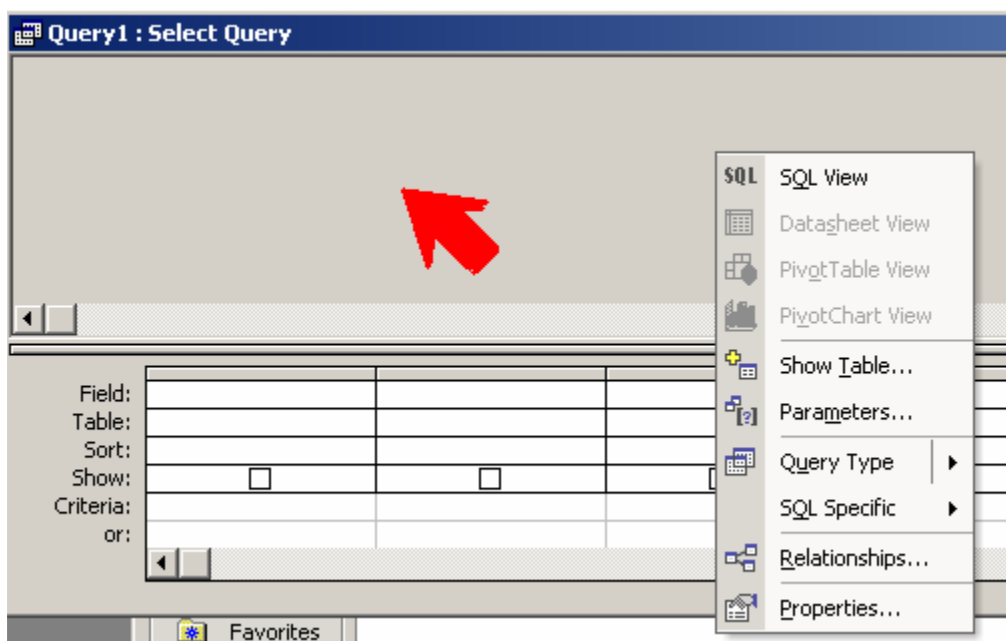
Și pentru crearea comenzilor de tip query avem la dispoziție un generator de comenzi de tip wizard. Din motive de limitări ale acestui generator nu îl vom utiliza. În consecință pentru crearea unei interogări vom alege prima opțiune CREATE QUERY IN DESIGN VIEW.

Fereastra care va apare SHOW TABLE este o fereastră în care se pot selecta tabelele care vom lucra. După selectare se apasă ADD pentru adăugarea lor în sintaxa de interogare. Adăugarea tabelelor din această fereastră nu este obligatorie, deoarece ele pot fi adăugate și ulterior (manual prin scriere) când scriem sintaxa interogării.



Adăugarea tabelelor în interogare

În fereastra QUERY, pe suprafața gri prin click dreapta de mouse se poate accesa următorul meniu:

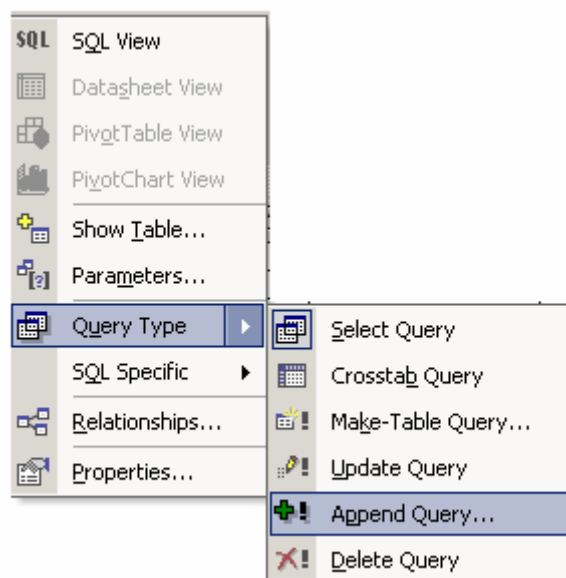


Meniu QUERY

Prima opțiune SQL o vom folosi pentru a scrie interogările.

Pentru ca principalele cuvinte cheie ale interogării să fie scrise putem selecta din cadrul meniului opțiunea QUERY TYPE.

Astfel în acrul acestui submeniu avem următoarele opțiuni:



Opțiunile submeniului QUERY TYPE

## INSERT - ADĂUGAREA DE LINII ÎN TABELĂ

Pentru adăugarea unor linii noi în tabelă se utilizează INSERT.

```
INSERT INTO TABELA [(ATRIBUT1, ATRIBUT2,... .)] VALUES
(VALOARE_ATRIBUT1, VALOARE_ATRIBUT2,... .)
```

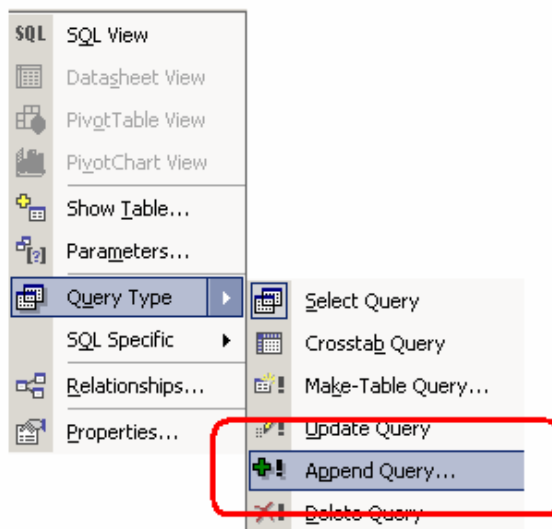
De exemplu dacă pentru tabela DEPARTAMENTE avem următoarele date:

departamente : Table					
	cod_dep	den	localitate	sef	telefon
+	conta	contabilitate	brasov	popescu ion	410410
+	info	informatica	codlea	anton dorin	430430
+	mn	management	brasov	chiriac dan	420420
▶					

Datele tabelii DEPARTAMENTE

și dacă dorim să adăugăm un nou departament care are datele:  
 cod\_dep: fin, den: financiar, localitate: bucuresti, sef: tecar gabriela,  
 telefon: 1234567  
 vom avea:

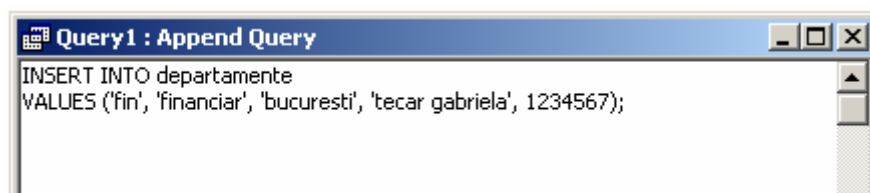
Din meniul QUERY TYPE se alege opțiunea APPEND QUERY



Alegerea opțiunii APPEND QUERY

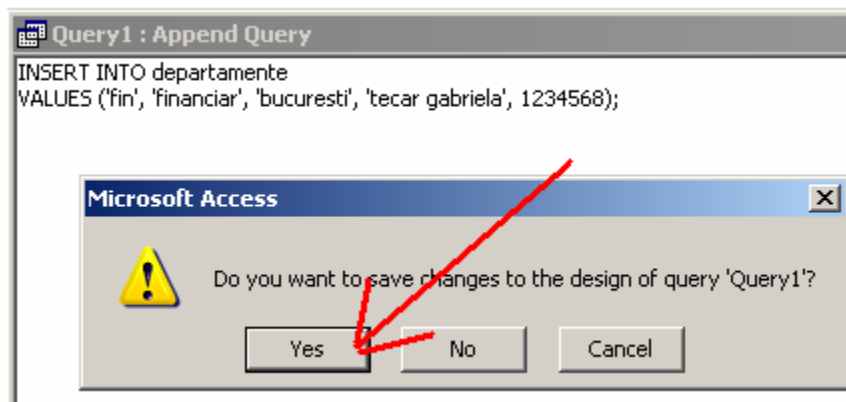
sau se poate direct ale opțiunea SQL VIEW

În fereastra de comenzi SQL vom scrie comanda:



Exemplu de comanda INSERT

După scrierea oricărei comenzi, aceasta trebuie salvată pentru a putea fi executată ulterior.



Salvarea comenzii



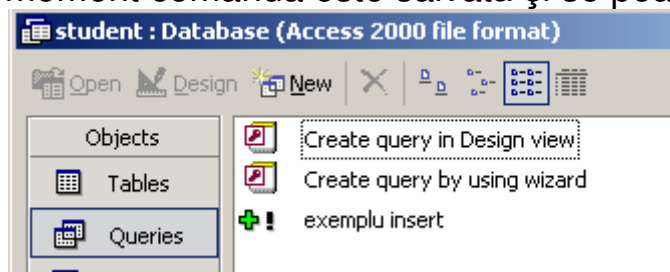
Odată salvată comanda trebuie să aibă un nume pentru a putea fi recunoscută și executată ulterior.

**Atenție! Numele trebuie să fie cât mai sugestiv, executarea unor comenzi eronate poate produce deteriorarea datelor din baza de date.**



Numirea comenzii

În acest moment comanda este salvată și se poate executa.

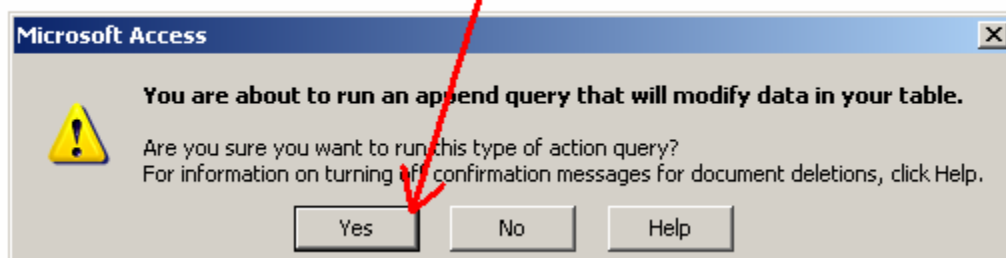


Vizualizarea comenzii nou create

Pentru executarea comenzii vom da dublu click pe numele ei în panoul de comandă.

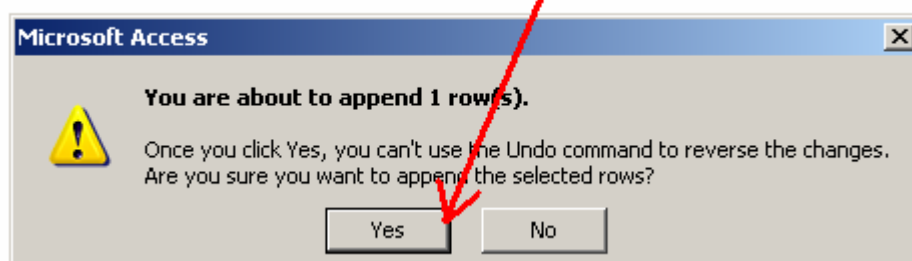
Pentru verificarea execuției comenzii, programul cere 2 confirmări:

1) pentru execuția comenzii



Acceptul de rulare a comenzii

2) pentru a adăuga în tabelă o nouă înregistrare



Acceptul de adăugare în tabelă a unei  
noi înregistrări

Dacă vizualizăm datele tabelii DEPARTAMENTE vom vedea  
înregistrarea făcută prin comanda INSERT

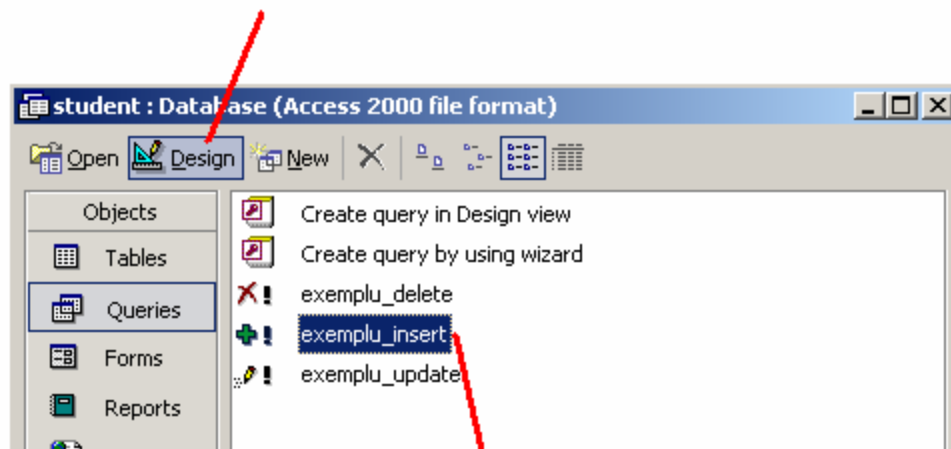
departamente : Table					
	cod_dep	den	localitate	sef	telefon
+	conta	contabilitate	brasov	popescu ion	410410
+	fin	financiar	bucuresti	tecar gabriela	1234568
+	info	informatica	codlea	anton dorin	430430
+	mn	management	brasov	chiriac dan	420420

Vizualizarea tabelii DEPARTAMENTE

## MODIFICAREA UNEI COMENZI

Există posibilitatea să apară necesitatea modificării unei  
comenzi existente. Modificarea se face urmând următorii pași:

## 2) selectarea opțiunii de modificare



## 1) selectarea comezii ce urmează să fie modificată

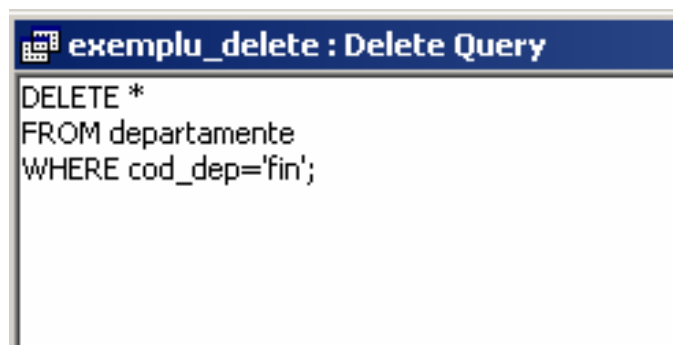
Modificarea unei comenzi

## DELETE - ȘTERGEREA ÎNREGISTRĂRILOR

Comanda SQL pentru ștergerea uneia sau mai multor linii dintr-o tabela este DELETE.

DELETE FROM TABELA WHERE CONDITIE

De exemplu dacă dorim să ștergem înregistrarea ce cuprinde datele departamentului financiar (pentru care cod\_dep este fin) vom avea:



Ștergerea departamentului financiar

## UPDATE - MODIFICAREA DATELOR DINTR-UN CÂMP

Pentru a modifica valoarea unuia sau mai multor atribute sau mai multe linii dintr-o tabela se foloseste comanda UPDATE cu formatul general:

```
UPDATE TABELA SET ATRIBUT1=EXPRESIE
[ATRIBUT2=EXPRESIE2.... ] WHERE PREDICAT
```

De exemplu dacă dorim ca pentru angajații din cadrul departamentului "conta" să creștem salariul brut cu 50 de lei vom avea:

```
exemplu_update : Update Query
UPDATE angajati SET sal_brut = sal_brut+50
WHERE cod_dep="conta";
```

Modificarea valorilor unui câmp

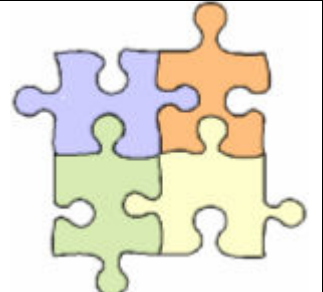
angajati : Table						
	cod	nume	prenume	data_nasterii	sal_brut	cod_dep
	100	Popescu	Ion	20.01.1975	830	conta
	102	Ionescu	Dan	25.06.1980	450	mn
	104	Petrescu	Dana	30.09.1970	850	conta
	105	Oprescu	Vasile	12.12.1960	1500	mn
	106	Basescu	Ionela	06.06.1950	1200	mn
	n					n

Vizualizarea datelor modificate în tabela ANGAJATI



### Intrebări:

1. Cum se pot modifica valorile unui câmp?
2. Cum se pot șterge înregistrările unei tabele după o condiție?

	<p style="text-align: center;"><b>Studiu individual</b></p> <p>Aplicația 2: Introducere de date, selecții simple și utilizând condiții simple și compuse.</p>
---	---

## APLICAȚIA 2

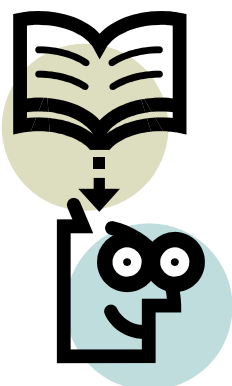
1. Realizați o tabelă FURNIZORI cu structura tabeli FURNIZORI din ANEXA 1.
2. Introduceți următoarele date:

Furnizori				
	Cod_furn	Nume_furn	Adresa	Localitate
	c1	ALFA	ALUNULUI 3	BRASOV
	c2	GAMA	REPUBLICII 34	BRASOV
	c3	PLUTO	LUNGA 20	CLUJ
	c4	MERCUR	DACIA 5	BRASOV
	c5	ORIZONT	ZIZINULUI 3	BRASOV
	c6	ONIRIS	VLAHUTA 332	BRASOV
	c7	JEO	CALEA BUCURESTI 33	BRASOV
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	m5	STAR	BARITIU 234	BRASOV
	c9	PLANET	TOMIS 43	CONSTANTA
	m1	SUN	MAMAIA 78	CONSTANTA
	m3	COSMOS	BALCESCU 34	BRASOV
	n10	GRIG	SLAVICI 43	BUCURESTI
	n9	POMPI	NEGOIU 778	BRASOV
	n7	VEZUVIU	CLOSCA 4	CLUJ

3. Modificați localitatea pentru angajați din BRASOV în RASNOV.
4. Ștergeți toți furnizorii din BRASOV
5. Ștergeți toți furnizorii din CLUJ
6. Înlocuiți numele fiecărei localități cu prima sa literă.

# TEMA 3

## SELECȚIA ȘI PROIECȚIA



### Principalele obiective ale capitolului sunt:

Introducere în conceptul de interogare  
a bazelor de date  
Sintaxa și utilizarea comenzii SELECT

Clauza SELECT corespunde operatorului proiecție din algebra relațională, fiind utilizată pentru desemnarea listei de attribute (coloane) din rezultat. Clauza FROM este cea în care sunt enumerate relațiile din care vor fi extrase informațiile aferente consultării. Clauza WHERE desemnează predicatul selectiv al algebrei relaționale (condiția), relativ la attribute ale relațiilor care apar în clauza FROM.

La modul general, o consultare SQL poate fi prezentată sub forma:

```
select c1,c2,...,cn  
from r1,r2,...,rm  
where p
```

unde:

cj - reprezintă coloanele rezultat;  
rj - reprezintă relațiile ce trebuie parcurse;  
p - reprezintă predicatul, condiția ce trebuie îndeplinită de  
tupluri (linii) pentru a fi incluse în rezultat.

Predicatul poate fi simplu sau compus (din mai multe condiții).

Când clauza WHERE este omisă se consideră implicit că predicatul p are valoare logică "adevărat", astfel încât în rezultat vor fi incluse toate liniile din tabelă sau produsul cartezian al tabelelor, enumerate în clauza FROM.

Dacă în locul coloanelor c1,c2,...,cn apare simbolul \*, rezultatul va fi alcătuit din toate coloanele relațiilor specificate în clauza FROM.

Atributele rezultatului preiau numele din tabela (tabelele) specificate în clauza FROM. Schimbarea numelui se realizează prin clauza AS.

exemplu:

```
select c1,c2 as NUME_NOU from r1
```

unde numele coloanei c2 devine NUME\_NOU;

SQL nu elimină automat liniile identice din rezultat, deci pentru ca fiecare linie să apară o singură dată este necesară utilizarea opțiunii DISTINCT.

exemplu:

```
select distinct c1,c2 from r1
```

În concluzie, o frază SELECT, corespunde:

- unei selecții algebrice (clauza WHERE p)
- unei proiecții (SELECT ci)
- unui produs cartezian (FROM -  $r1 \otimes r2 \otimes \dots \otimes r_m$ )

și conduce la obținerea unui rezultat cu n coloane, fiecare coloană fiind: un atribut din r1,r2,...,rm sau expresie calculată pe baza unor atribute din r1,r2,...,rm.

Execuția unei fraze SELECT realizează un rezultat sub formă tabelară.

Rezultatul poate fi:

1. o listă (text)
2. o tabelă propriu-zisă
3. o tabelă temporară
4. o tabelă derivată (imagine)
5. o variabilă masiv (tablou)

### Exemplu:

Utilizăm tabela ANGAJATI în care introducem următoarele date:

Angajati						
Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep	
1	Popescu	Ana	01/31/80	600	info	
2	Ionescu	Dana	12/17/78	850	cont	
3	Iordache	Ion	02/23/67	1000	info	
4	Mihalcea	Dan	06/24/84	400	mk	
5	Dobrescu	Mircea	04/04/80	740	info	
6	Dancea	Mirela	12/21/78	500	mn	
7	Pavel	Maria	12/18/67	360	cont	
8	Grigorescu	Ana	02/06/78	500	mn	
9	Enescu	Ioan	05/12/80	550	mk	
10	Varga	Ion	05/08/81	390	info	

Datele tabelii ANGAJATI

1. Care sunt datele conținute în tabela angajati?

select cod, nume, prenume, data\_nasterii, sal\_brut, cod\_dep from angajati

sau:

select \* from angajati

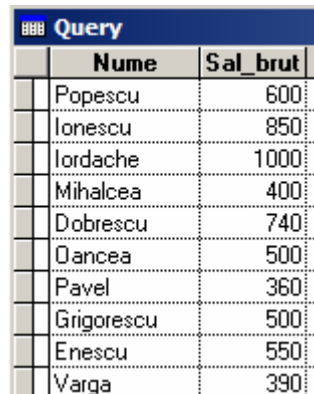
Query						
Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep	
1	Popescu	Ana	01/31/80	600	info	
2	Ionescu	Dana	12/17/78	850	cont	
3	Iordache	Ion	02/23/67	1000	info	
4	Mihalcea	Dan	06/24/84	400	mk	
5	Dobrescu	Mircea	04/04/80	740	info	
6	Dancea	Mirela	12/21/78	500	mn	
7	Pavel	Maria	12/18/67	360	cont	
8	Grigorescu	Ana	02/06/78	500	mn	
9	Enescu	Ioan	05/12/80	550	mk	
10	Varga	Ion	05/08/81	390	info	

Rezultatele exemplului 1



2. Care sunt numele salariaților și salariul brut?

```
select nume, sal_brut from angajati
```



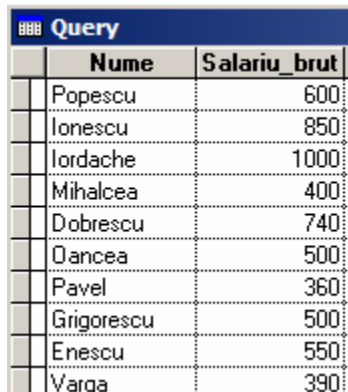
The screenshot shows a query window titled 'Query' with a table of results. The table has two columns: 'Nume' and 'Sal\_brut'. The data rows are as follows:

	Nume	Sal_brut
	Popescu	600
	Ionescu	850
	Iordache	1000
	Mihalcea	400
	Dobrescu	740
	Dancea	500
	Pavel	360
	Grigorescu	500
	Enescu	550
	Varga	390

Rezultatele exemplului 2

3. Care sunt numele salariaților și salariul brut, dar pentru coloana sal\_brut denumim coloana salariu\_brut?

```
select nume, sal_brut as salariu_brut from angajati
```



The screenshot shows a query window titled 'Query' with a table of results. The table has two columns: 'Nume' and 'Salariu\_brut'. The data rows are as follows:

	Nume	Salariu_brut
	Popescu	600
	Ionescu	850
	Iordache	1000
	Mihalcea	400
	Dobrescu	740
	Dancea	500
	Pavel	360
	Grigorescu	500
	Enescu	550
	Varga	390

Rezultatele exemplului 3

4. Afișați numele angajaților și salariul brut, pentru salariații care au salariul brut mai mare de 700 lei

```
select nume from angajati where sal_brut>700
```

Query		
	Nume	Sal_brut
▶	Ionescu	850
	Iordache	1000
	Dobrescu	740

Rezultatele exemplului 4

5. Care sunt datele salariaților care lucrează în departamentul informatic (cod\_dep="info")?

```
select * from angajati WHERE cod_dep="info"
```

Query						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
▶	1	Popescu	Ana	01/31/80	600	info
	3	Iordache	Ion	02/23/67	1000	info
	5	Dobrescu	Mircea	04/04/80	740	info
	10	Varga	Ion	05/08/81	390	info

Rezultatele exemplului 5

6. Care sunt datele salariaților care lucrează în departamentul informatic (cod\_dep="info") și care au salariul brut mai mare de 700 de lei?

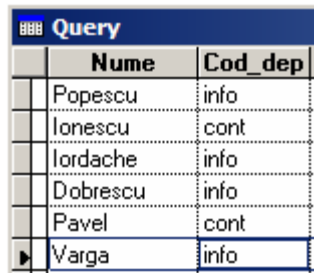
```
select * from angajati where cod_dep="info" and sal_brut>700
```

Query						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
▶	3	Iordache	Ion	02/23/67	1000	info
	5	Dobrescu	Mircea	04/04/80	740	info

Rezultatele exemplului 6

7. Afișați numele salariaților și codul departamentului pentru angajații care sunt fie în departamentul informatic (cod\_dep="info") fie în departamentul contabilitate (cod\_dep="cont")

```
select nume, cod_dep from angajati where cod_dep="info" or cod_dep="cont"
```

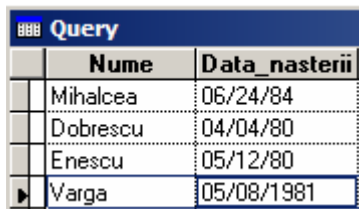


	Nume	Cod_dep
	Popescu	info
	Ionescu	cont
	Iordache	info
	Dobrescu	info
	Pavel	cont
▶	Varga	info

Rezultatele exemplului 7

8. Afișați numele angajaților și data nașterii pentru angajații născuți după 15-03-1980 (formatul pentru data este #mm-dd-yyyy#)

```
select nume, data_nasterii from angajati where data_nasterii > #03-15-1980#
```

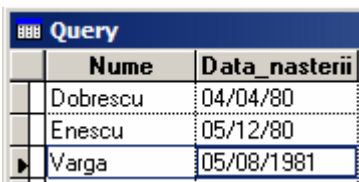


	Nume	Data_nasterii
	Mihalcea	06/24/84
	Dobrescu	04/04/80
	Enescu	05/12/80
▶	Varga	05/08/1981

Rezultatele exemplului 8

9. Afișați numele angajaților și data nașterii pentru angajații născuți între 15-03-1980 și 20-05-1983

```
select nume, data_nasterii from angajati where data_nasterii > #03-15-1980# and data_nasterii < #05-20-1980#
```

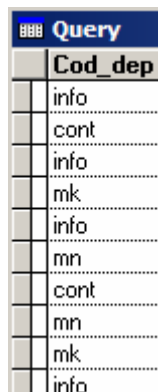


	Nume	Data_nasterii
	Dobrescu	04/04/80
	Enescu	05/12/80
▶	Varga	05/08/1981

Rezultatele exemplului 9

10. Care sunt codurile de departament din tabela angajati?

```
select cod_dep from angajati
```

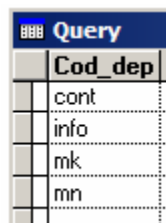


Cod_dep
info
cont
info
mk
info
mn
cont
mn
mk
info

Rezultatele exemplului 10

11. Care sunt codurile de departament din tabela angajati, afișate o singură dată?

```
select distinct cod_dep from angajati
```

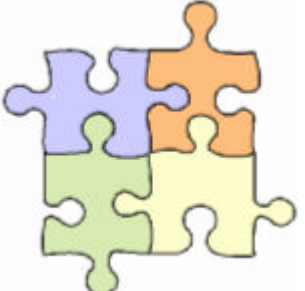


Cod_dep
cont
info
mk
mn

Rezultatele exemplului 11

**Intrebări:**

1. Care sunt clauzele principale ale unei selecții?
2. Care sunt clauzele obligatorii ale unei selecții?
3. Câte tabele putem trece la clauza FROM?

	<p style="text-align: center;"><b>Studiu individual</b></p> <p>Aplicația 3: Introducere de date, selecții simple și utilizând condiții simple și compuse.</p>
---	---

### APLICAȚIA 3

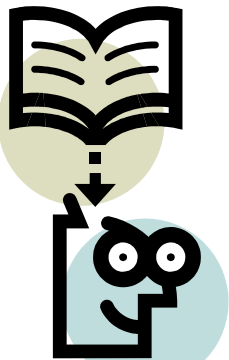
1. Realizați o tabelă FURNIZORI cu structura tabelii FURNIZORI din ANEXA 1.
2. Introduceți următoarele date:

Furnizori				
	Cod_furn	Nume_furn	Adresa	Localitate
	c1	ALFA	ALUNULUI 3	BRASOV
	c2	GAMA	REPUBLICII 34	BRASOV
	c3	PLUTO	LUNGA 20	CLUJ
	c4	MERCUR	DACIA 5	BRASOV
	c5	ORIZONT	ZIZINULUI 3	BRASOV
	c6	ONIRIS	VLAHUTA 332	BRASOV
	c7	JEO	CALEA BUCURESTI 33	BRASOV
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	m5	STAR	BARITIU 234	BRASOV
	c9	PLANET	TOMIS 43	CONSTANTA
	m1	SUN	MAMAIA 78	CONSTANTA
	m3	COSMOS	BALCESCU 34	BRASOV
	n10	GRIG	SLAVICI 43	BUCURESTI
	n9	POMPI	NEGOIU 778	BRASOV
▶	n7	VEZUVIU	CLOSCA 4	CLUJ

3. Care sunt furnizorii din Brașov?
4. Care sunt furnizorii din Brașov sau București?
5. Care sunt furnizorii care nu stau în Constanța?
6. Ștergeți furnizorii care sunt din București.
7. Introduceți trei furnizori noi utilizând INSERT.
8. Modificați localitatea în Cluj pentru furnizorii din Constanta.

# TEMA 4

## Coloane, expresii

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Definirea unor coloane noi Realizarea de calcule pe baza unor date din baza de date.</p>
--	--

O facilitate importantă în multe interogări SQL ține de definirea, pe lângă attributele tabelelor, a unor coloane noi, pe baza unor expresii. Clauza AS ermite denumirea coloanelor calculate sau redenumirea unor coloane ale tabelelor.

De exemplu dacă dorim să aflăm care este TVA-ul aferent unei facturi, vom avea:

In tabelă FACTURI avem introduse următoarele date:

Facturi				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1003	c2	01/14/06	1800
	1004	c1	01/14/06	3000
	1005	c4	01/14/06	2100
	1006	c1	01/16/06	2300
	1007	c5	01/17/06	5600
	1008	c9	01/18/06	8900
	1009	c2	02/01/06	8000

Datele tabelii FACTURI

Pentru a afișa valoarea tva-ului, precum și valoarea cu TVA pentru fiecare factură vom avea selectul:

```
select nr_factura, cod_fur, data_fact, valoare, valoare*19/100 as TVA, valoare*119/100 as val_totala from facturi
```

și vom obține Query-ul:

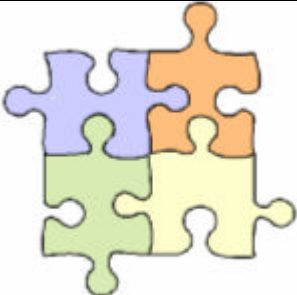
Query						
	Nr_factura	Cod_furn	Data_fact	Valoare	Tva	Val_totala
	1002	c1	01/12/06	600	114.0000	714.0000
	1003	c2	01/14/06	1800	342.0000	2142.0000
	1004	c1	01/14/06	3000	570.0000	3570.0000
	1005	c4	01/14/06	2100	399.0000	2499.0000
	1006	c1	01/16/06	2300	437.0000	2737.0000
	1007	c5	01/17/06	5600	1064.0000	6664.0000
	1008	c9	01/18/06	8900	1691.0000	10591.0000
	1009	c2	02/01/06	8000	1520.0000	9520.0000

Rezultatele selecției



### Intrebări:

1. Cum putem să denumim coloana din rezultat?
2. Cum putem obține o coloană nouă utilizând în selecție o condiție?

	<p style="text-align: center;"><b>Studiu individual</b></p> <p>Aplicația 4: Calcularea amortizării lunare prin metoda liniară.</p>
---	--

## APLICAȚIA 4

1. Realizați tabela MF după structura tabeli MF din ANEXA 1.
2. Introduceți următoarele date:

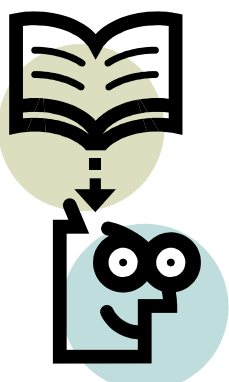
Mf					
	Cod_mf	Den	Valoare	Per_amort	Cod_furn
	1000	masina 1	50000	3	c1
	10001	cladire 1	4000000	20	c3
	10002	masina 2	43000	10	c2
	10003	cladire 2	340000	20	c4
	10004	hala 2	20000	10	c5
	10005	masina 5	23000	3	c1
	10007	cladire 4	67000	20	c1
	1002	hala 1	678000	10	c2
	1004	calculator	2000	3	c4

3. Care sunt mijloacele fixe cu valoare mai mare de 30000?
4. Dacă PER\_AMORT reprezintă perioada de amortizare în ani, calculați care este valoarea amortizării lunare liniare.
5. Denumiți coloana cu amortizarea lunară liniară AM\_LUN.
6. Care sunt mijloacele fixe care au amortizarea lunară mai mare de 1000?
7. Modificați perioada de amortizare în 5 pentru mijloacele fixe cu codul 10004.
8. Calculați care este amortizarea lunară pentru "masina 2"
9. Calculați amortizarea lunară liniară pentru mijloacele fixe cumpărate de la furnizorul c1 sau c2.



# TEMA 5

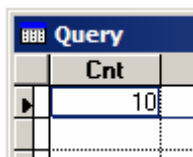
## Funcțiile agregat: COUNT, SUM, AVG, MIN, MAX

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Utilizarea funcțiilor Count, sum, avg. min, max Realizarea unor noi coloane care să conțină date obținute prin utilizarea funcțiilor de mai sus.</p>
--	--

Funcția COUNT contorizează valorile neutre ale unei coloane sau numărul de linii dintr-un rezultat al unei interogări, altfel spus, în rezultatul unei consultări, COUNT numără câte valori diferite de NULL are o coloană specificată sau câte linii sunt:

De exemplu dacă dorim să aflăm câte facturi avem în tabela FACTURI (câte linii sunt) avem:

```
select count (*) from facturi
```

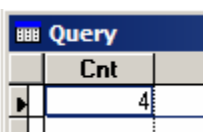


Query
Cnt
10

Numărul de facturi din tabela FACTURI

Sau de exemplu dacă dorim să vedem câte facturi au valoarea peste 3000 vom avea:

```
select count(*) from facturi WHERE valoare>3000
```



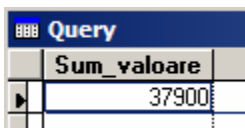
Query
Cnt
4

Numărul de facturi din tabela FACTURI  
cu valoare mai mare de 3000

Funcția SUM este una dintre cele mai utilizate funcții în aplicațiile economice, deoarece datele financiar-contabile și cele ale evidenței tehnico-operative sunt preponderent cantitative.

De exemplu dacă dorim să aflăm care este valoarea totală a facturilor din tabela FACTURI avem:

```
select sum(valoare) from facturi
```

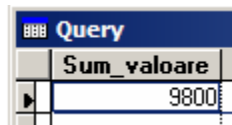


Query
Sum_valoare
37900

Valoarea totală a facturilor din tabela FACTURI

Sau de exemplu dacă dorim să aflăm care este valoarea totală a facturilor din tabela FACTURI a furnizorului cu cod\_furn "c2" avem:

```
select sum(valoare) from facturi where cod_furn="c2"
```



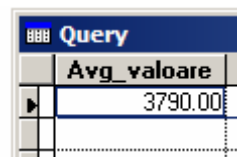
Query	
Sum_valoare	
	9800

Valoarea totală a facturilor din tabela FACTURI  
a furnizorului cu cod\_furn "c2"

Funcția AVG (AVERAGE) calculează media aritmetică a unei coloane prin divizarea sumei valorilor coloanei respective la numărul de valori nenule ale acesteia.

De exemplu dacă dorim să aflăm care este valoarea medie a facturilor din tabela FACTURI avem:

```
select avg(valoare) from facturi
```

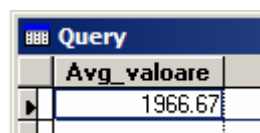


Query	
Avg_valoare	
	3790.00

Valoarea medie a facturilor din tabela FACTURI

Sau de exemplu dacă dorim să aflăm care este valoarea medie a facturilor din tabela FACTURI a furnizorului cu cod\_furn "c1" avem:

```
select avg(valoare) from facturi where cod_furn="c1"
```



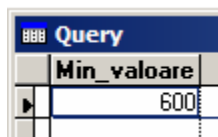
Query	
Avg_valoare	
	1966.67

Valoarea medie a facturilor din tabela FACTURI  
a furnizorului cu cod\_furn "c1"

Deosebit de utile în diverse tipuri de analiză cele două funcții MIN și MAX determină valoarea minimă și maximă pentru o coloană. Se pot folosi și pentru attribute de tip șir de caractere, caz în care elementul de comparație este codul ASCII al caracterelor.

De exemplu dacă dorim care este valoare minimă din tabela facturi vom avea:

```
select min(valoare) from facturi
```

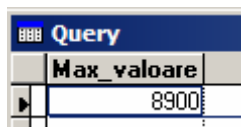


Query	
Min_valoare	
	600

Valoarea minimă din FACTURI

Sau de exemplu dacă dorim care este valoare maximă din tabela facturi vom avea:

```
select max(valoare) from facturi
```

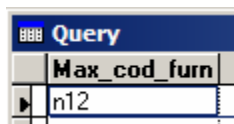


Query	
Max_valoare	
	8900

Valoarea maximă din FACTURI

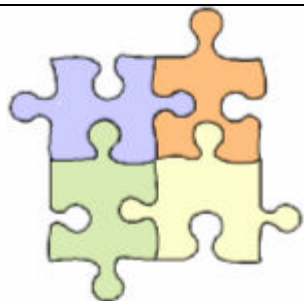
Dacă dorim să aflăm care este din punct de vedere alfabetic cod\_furn cu cea mai mare valoare vom avea:

```
select max(cod_furn) from facturi
```



Query	
Max_cod_furn	
	n12

Valoarea maximă a cod\_furn din FACTURI



### Studiu individual

APLICAȚIA 6: Utilizarea funcțiilor: AVG; MIN; MAX; SUM; COUNT

**APLICAȚIA 5**

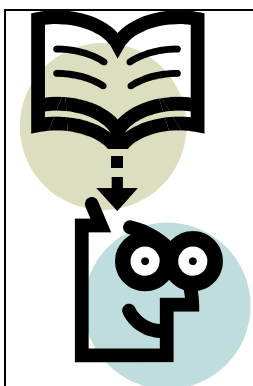
1. Realizați tabela FACTURI cu structura tabeli FACTURI din ANEXA 1.
2. Introduceți în tabela FACTURI următoarele date:

Facturi				
	Nr_factura	Cod_furn	Data_fact	Valoare
	3525	f1	01/12/06	5325
	3522	f3	01/14/06	3232
	3231	f2	01/14/06	9970
	3445	g44	01/14/06	3000
	54	h2	01/16/06	2788
	33322	d3	01/17/06	29000
	4433	c9	01/18/06	6700
	43455	c2	02/01/06	22000
	2221	j3	02/02/06	33000
	4324	i3	02/02/06	11000

3. Care este media valorii facturilor?
4. Care este valoarea cea mai mare a unei facturi?
5. Care este valoarea cea mai mică a unei facturi?
6. Calculați valoarea medie a valorii facturilor dar nu cu AVG.
7. Câte facturi avem în tabelă?
8. Cate facturi avem în tabelă cu o valoare mai mare de 5000?
9. Câte facturi avem în tabelă cu o valoare între 4000 și 30000? (prin două metode)
10. Care este valoare medie a facturilor cu nr. 3525 și 3522?

# TEMA 6

## OPȚIUNEA ORDER BY



**Principalele obiective ale capitolului sunt:**

Ordonarea datelor unei tabele după unul sau mai multe atribute

Una dintre caracteristicile modelului relațional este că nici ordinea atributelor, nici ordinea liniilor în relații nu reprezintă importanță din punctul de vedere al conținutului informațional.

Însă în practică forma de prezentare a rezultatelor interogării este importantă. Spre exemplu o listă a localităților dintr-o tabelă este mai ușor citită în ordine alfabetică.

De exemplu dacă dorim pentru tabela facturi să afișăm toate datele ordonate după câmpul `cod_furn` vom avea

```
select * from facturi order by cod_furn
```

Query				
Nr_factura	Cod_furn	Data_fact	Valoare	
1002	c1	01/12/06	600	
1004	c1	01/14/06	3000	
1006	c1	01/16/06	2300	
1003	c2	01/14/06	1800	
1009	c2	02/01/06	8000	
1005	c4	01/14/06	2100	
1007	c5	01/17/06	5600	
1008	c9	01/18/06	8900	

Datele obținute în urma ordonării după câmpul cod\_furn


Aranjarea se poate face atât crescător (ASC) cât și descrescător (DESC). În plus se pot specifica mai multe coloane care să servească drept criterii suplimentare de ordonare. Astfel, la valori egale ale primului atribut, intră în acțiune criteriul de "balotaj", care este al doilea atribut, etc.

De exemplu dacă dorim să ordonăm datele din facturi întâi după cod\_furn și apoi după valoare (după acest criteriu descrescător) vom avea:

```
select * from facturi order by cod_furn, valoare desc
```

Query				
Nr_factura	Cod_furn	Data_fact	Valoare	
1004	c1	01/14/06	3000	
1006	c1	01/16/06	2300	
1002	c1	01/12/06	600	
1009	c2	02/01/06	8000	
1003	c2	01/14/06	1800	
1005	c4	01/14/06	2100	
1007	c5	01/17/06	5600	
1008	c9	01/18/06	8900	

Obținerea datelor ordonate după cod\_furn și apoi descrescător după valoare



**Intrebări:**

1. După câte câmpuri putem ordona și cum?



**Studiu individual**

APLICAȚIA 6: Ordonarea rezultatelor unei interogări.

## APLICAȚIA 6

1. Realizați tabela MF după structura tabeli MF din ANEXA 1.
2. Introduceți următoarele date:

MF					
	Cod_mf	Den	Valoare	Per_amort	Cod_furn
	1000	masina 1	50000	3	c1
	10001	cladire 1	4000000	20	c3
	10002	masina 2	43000	10	c2
	10003	cladire 2	340000	20	c4
	10004	hala 2	20000	10	c5
	10005	masina 5	23000	3	c1
	10007	cladire 4	67000	20	c1
	1002	hala 1	678000	10	c2
	1004	calculator	2000	3	c4

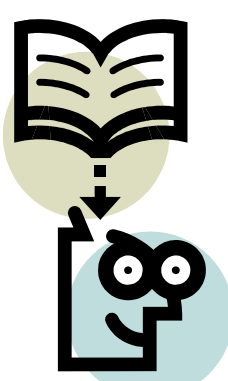
3. Care sunt datele din tabela MF, ordonate alfabetic după denumire?
4. Care sunt datele din tabela MF, ordonate alfabetic după denumire și apoi după valoare?



- 
5. Care este amortizarea lunară liniară pentru fiecare mijloc fix, datele fiind ordonate după denumirea mijlocului fix?
  6. Afișați denumirea mijloacelor fixe, ordonată alfabetic pentru mijloacele fixe cu o valoare mai mare de 32000 inclusiv.
  7. Afișați alfabetic codurile furnizor - o singură dată.
  8. Care este valoarea cea mai mare unui mijloc fix?
  9. Care este valoare medie a amortizării lunare a mijloacelor fixe?
  10. Câte mijloace fixe avem?

# TEMA 7

## OPERATORII BETWEEN, LIKE ȘI IN

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Utilizarea operatorilor BETWEEN LIKE IN</p>
--	---

Pentru formularea predicatului de selecție, SQL permite utilizarea, pe lângă >, <, >=, <=, = și a altor operatori, dintre care ne vom opri la BETWEEN (cuprins între), LIKE (ca și), IN (în), la care se adaugă IS NULL.

Operatorul BETWEEN este util pentru definirea intervalelor de valori.

De exemplu dacă dorim să afișăm doar facturile care au valori între 4000 și 8000 vom avea:

```
select * from facturi where valoare between 4000 and 8000
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1007	c5	01/17/06	5600
	1009	c2	02/01/06	8000

Date afișate utilizând condiția BETWEEN

Același rezultat îl puteam obține și cu o condiție dublă:

```
select * from facturi where valoare >= 4000 and valoare <=8000
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1007	c5	01/17/06	5600
	1009	c2	02/01/06	8000

Date afișate utilizând condiția dublă

Operatorul LIKE permite compararea unui atribut (expresii) cu un literal utilizând o "mască" construită cu ajutorul specificatorilor multipli \* și ?. Simbolurile procent și underscore sunt denumite jokeri. Procentul substituie un șir de lungime variabilă, 0-n caractere, în timp ce underscore substituie un singur caracter.

Pentru exemplificare mai introducem două facturi cu următoarele date:

Facturi				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1003	c2	01/14/06	1800
	1004	c1	01/14/06	3000
	1005	c4	01/14/06	2100
	1006	c1	01/16/06	2300
	1007	c5	01/17/06	5600
	1008	c9	01/18/06	8900
	1009	c2	02/01/06	8000
	1010	m5	02/02/06	5000
	1011	n12	02/02/06	600

Introducerea suplimentară a 2 facturi

De exemplu dacă dorim să afișăm care sunt toți furnizorii a căror cod\_client începe cu litera "c" vom avea:

```
select * from facturi where cod_furn like "c*"
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1003	c2	01/14/06	1800
	1004	c1	01/14/06	3000
	1005	c4	01/14/06	2100
	1006	c1	01/16/06	2300
	1007	c5	01/17/06	5600
	1008	c9	01/18/06	8900
	1009	c2	02/01/06	8000

Furnizorii a căror cod\_furn începe cu litera "c"

De exemplu dacă dorim să afișăm care sunt toți furnizorii a căror cod\_client începe cu litera "m" vom avea:

```
select * from facturi where cod_furn like "m*"
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1010	m5	02/02/06	5000

Furnizorii a căror cod\_furn începe cu litera "m"

De exemplu dacă dorim să afișăm care sunt toți furnizorii a căror cod\_client are două caractere și al doilea caracter este 5 vom avea:

```
select * from facturi where cod_furn like "?5"
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1007	c5	01/17/06	5600
	1010	m5	02/02/06	5000

Furnizorii a căror cod\_furn are două caractere și al doilea caracter este 5

Atunci când se testează dacă valoarea unui atribut este încadrabilă într-o listă de valori dată, în locul folosirii abundente a operatorului OR este mai elegant să se apeleze la operatorul IN.

De exemplu dacă dorim să afișăm facturile furnizorilor care au cod\_furn c1, c3 și c4 avem:

```
select * from facturi where cod_furn in ("c1","c3","c4")
```

Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1004	c1	01/14/06	3000
	1005	c4	01/14/06	2100
	1006	c1	01/16/06	2300

Facturile furnizorilor care au cod\_furn c1, c3 și c4

Sau de exemplu dacă dorim să selectăm facturile care au valoare de 500, 600, 5000 vom avea:

```
select * from facturi where valoare in (500, 600, 5000)
```

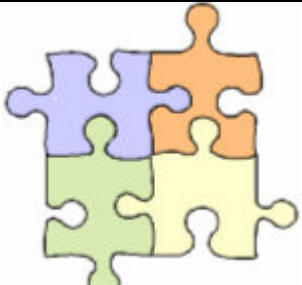
Query				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1010	m5	02/02/06	5000
	1011	n12	02/02/06	600

Facturile care au valoare de 500, 600, 5000



### Intrebări:

1. Care este diferența între operatorii BETWEEN și IN?
2. Cu ce poate fi substituit operatorul BETWEEN?

	<p style="text-align: center;"><b>Studiu individual</b></p> <p>APLICAȚIA 7: Utilizarea operatorilor BETWEEN; LIKE și IN în calcularea unor date pe baza tabelului ANGAJATI.</p>
---	---

## APLICAȚIA 7

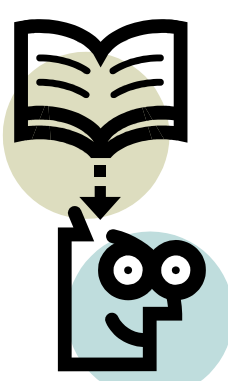
1. Realizați tabela SALARIATI pe baza structurii tabelului ANGAJATI din ANEXA 1.
2. În tabelă introduceți următoarele date:

Angajati						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
	1	Popescu	Ana	01/31/80	600	info
	2	Ionescu	Dana	12/17/78	850	cont
	3	Iordache	Ion	02/23/67	1000	info
	4	Mihalcea	Dan	06/24/84	400	mk
	5	Dobrescu	Mircea	04/04/80	740	info
	6	Dancea	Mirela	12/21/78	500	mn
	7	Pavel	Maria	12/18/67	360	cont
	8	Grigorescu	Ana	02/06/78	500	mn
	9	Enescu	Ioan	05/12/80	550	mk
	10	Varga	Ion	05/08/81	390	info

3. Care sunt salariații care au salariu cuprins între 550 și 1000?
4. Care sunt angajații care au salariu 600, 1000, 750, sau 300?
5. Câți angajați avem?
6. Care este salariul mediu?
7. Care este numele salariatului cu salariul cel mai mare?
8. Care sunt angajații a căror nume începe cu litera M sau I?
9. Câți angajați au prenumele din trei litere?
10. Care sunt salariații care nu au prenumele ION sau IOANA?

# TEMA 8

## REUNIUNE, PRODUS CARTEZIAN

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Prezentarea operatorului UNION Realizarea produsului cartezian</p>
--	--

Primii trei operatori asamblați prezintă operatori SQL dedicați: UNION, INTERSECT, MINUS (EXTRACT), în timp ce produsul cartezian se calculează automat prin simpla enumerare a celor două tabele din clauza FROM.

La reuniunea a două tabele, SQL elimină automat liniile identice din rezultat. Dacă se dorește prelucrarea tuturor liniilor celor două relații și implicit apariția de linii duplicate, se folosește clauza ALL (UNION ALL).

De exemplu, după aceeași structură a tablei FURNIZORI creăm tabela FURNIZORI2.

Tabela FURNIZORI2 conține următoarele date:

Furnizori2				
	Cod_furn	Nume_furn	Adresa	Localitate
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	n10	GRIG	SLAVICI 43	BUCURESTI
	n13	Milenium	OCTAVIAN GOGA 3	VASLUI
	n15	Roza	ION RATIU 445	BACAU
	n17	Corleone	GRIGORE VIANU 45	BRASOV

Datele tabelii FURNIZORI2

Furnizori				
	Cod_furn	Nume_furn	Adresa	Localitate
	c1	ALFA	ALUNULUI 3	BRASOV
	c2	GAMA	REPUBLICII 34	BRASOV
	c3	PLUTO	LUNGA 20	CLUJ
	c4	MERCUR	DACIA 5	BRASOV
	c5	ORIZONT	ZIZINULUI 3	BRASOV
	c6	ONIRIS	VLAHUTA 332	BRASOV
	c7	JEO	CALEA BUCURESTI 33	BRASOV
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	m5	STAR	BARITIU 234	BRASOV
	c9	PLANET	TOMIS 43	CONSTANTA
	m1	SUN	MAMAIA 78	CONSTANTA
	m3	COSMOS	BALCESCU 34	BRASOV
	n10	GRIG	SLAVICI 43	BUCURESTI
	n9	POMPI	NEGOIU 778	BRASOV
	n7	VEZUVIU	CLOSCA 4	CLUJ

Datele tabelii FURNIZORI

Dacă dorim să afișăm toate datele (utilizând reuniunea) din tabelele FURNIZORI și FURNIZORI2 avem:

```
select * from furnizori union select * from furnizori2
```



Query				
	Cod_furn	Nume_furn	Adresa	Localitate
	c1	ALFA	ALUNULUI 3	BRASOV
	c2	GAMA	REPUBLICII 34	BRASOV
	c3	PLUTO	LUNGA 20	CLUJ
	c4	MERCUR	DACIA 5	BRASOV
	c5	ORIZONT	ZIZINULUI 3	BRASOV
	c6	ONIRIS	VLAHUTA 332	BRASOV
	c7	JEO	CALEA BUCURESTI 33	BRASOV
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	c9	PLANET	TOMIS 43	CONSTANTA
	m1	SUN	MAMAIA 78	CONSTANTA
	m3	COSMOS	BALCESCU 34	BRASOV
	m5	STAR	BARITIU 234	BRASOV
	n10	GRIG	SLAVICI 43	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	n13	Milenium	OCTAVIAN GOGA 3	VASLUI
	n15	Roza	ION RATIU 445	BACAU
	n17	Corleone	GRIGORE VIANU 45	BRASOV
	n7	VEZUVIU	CLOSCA 4	CLUJ
	n9	POMPI	NEGOIU 778	BRASOV

Toate datele (utilizând reuniunea) din tabelele FURNIZORI și FURNIZORI2

SQL nu pune la dispoziție vreun operator special dedicat produsului cartezian.

Produsul cartezian se realizează prin simpla enumerare a tabelelor în clauza FROM.

De exemplu dacă dorim să realizăm produsul cartezian între tabelele FACTURI și FURNIZORI avem:

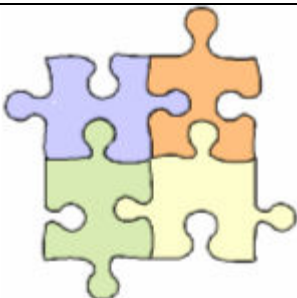
```
select * from facturi, furnizori
```

Query								
Nr_factura	Cod_furn_a	Data_fact	Valoare	Cod_furn_b	Nume_furn	Adresa	Localitate	
1002	c1	01/12/06	600	c2	GAMA	REPUBLICII 34	BRASOV	
1002	c1	01/12/06	600	c3	PLUTO	LUNGA 20	CLUJ	
1002	c1	01/12/06	600	c4	MERCUR	DACIA 5	BRASOV	
1002	c1	01/12/06	600	c5	ORIZONT	ZIZINULUI 3	BRASOV	
1002	c1	01/12/06	600	c6	ONIRIS	VLAHUTA 332	BRASOV	
1002	c1	01/12/06	600	c7	JEO	CALEA BUCURESTI 33	BRASOV	
1002	c1	01/12/06	600	c8	BONNY	CALEA MOSILOR 4	BUCURESTI	
1002	c1	01/12/06	600	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI	
1002	c1	01/12/06	600	m5	STAR	BARITIU 234	BRASOV	
1002	c1	01/12/06	600	c9	PLANET	TOMIS 43	CONSTANTA	
1002	c1	01/12/06	600	m1	SUN	MAMAIA 78	CONSTANTA	
1002	c1	01/12/06	600	m3	COSMOS	BALCESCU 34	BRASOV	
1002	c1	01/12/06	600	n10	GRIG	SLAVICI 43	BUCURESTI	
1002	c1	01/12/06	600	n9	POMPI	NEGOLIU 778	BRASOV	
1002	c1	01/12/06	600	n7	VEZUVIU	CLOSCA 4	CLUJ	
1002	c1	01/12/06	600					
1003	c2	01/14/06	1800	c1	ALFA	ALUNULUI 3	BRASOV	
1003	c2	01/14/06	1800	c2	GAMA	REPUBLICII 34	BRASOV	
1003	c2	01/14/06	1800	c3	PLUTO	LUNGA 20	CLUJ	
1003	c2	01/14/06	1800	c4	MERCUR	DACIA 5	BRASOV	
1003	c2	01/14/06	1800	c5	ORIZONT	ZIZINULUI 3	BRASOV	
1003	c2	01/14/06	1800	c6	ONIRIS	VLAHUTA 332	BRASOV	
1003	c2	01/14/06	1800	c7	JEO	CALEA BUCURESTI 33	BRASOV	
1003	c2	01/14/06	1800	c8	BONNY	CALEA MOSILOR 4	BUCURESTI	
1003	c2	01/14/06	1800	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI	
1003	c2	01/14/06	1800	m5	STAR	BARITIU 234	BRASOV	
1003	c2	01/14/06	1800	c9	PLANET	TOMIS 43	CONSTANTA	

Produsul cartezian al tabelelor FACTURI și FURNIZORI

**Intrebări:**

1. Ce reprezintă operatorul UNION?
2. Cum se realizează produsul cartezian?

**Studiu individual**

APLICAȚIA 8: Evidența mijloacelor fixe ale unei întreprinderi

## APLICAȚIA 8

1. Realizați tabela MF1 și MF2 după structura tabeli MF din ANEXA 1.
2. Introduceți următoarele date:

Mf					
	Cod_mf	Den	Valoare	Per_amort	Cod_furn
	1000	masina 1	50000	3	c1
	10001	cladire 1	4000000	20	c3
	10002	masina 2	43000	10	c2
	10003	cladire 2	340000	20	c4
	10004	hala 2	20000	10	c5
	10005	masina 5	23000	3	c1
	10007	cladire 4	67000	20	c1
	1002	hala 1	678000	10	c2
	1004	calculator	2000	3	c4

Datele tabeli MF1

Mf					
	Cod_mf	Den	Valoare	Per_amort	Cod_furn
	3832	cladire 4	20000	56	c3
	3223	cladire 5	50000	20	c3
	432	masina 10	40000	10	c2
	3212	cladire 8	43000	20	c4
	4321	hala 6	56999	10	c5
	5678	masina 7	422200	3	c1
	4321	masina 6	60000	20	c1
	1002	hala 1	10000	10	c2
	1004	calculator	2000	3	c4

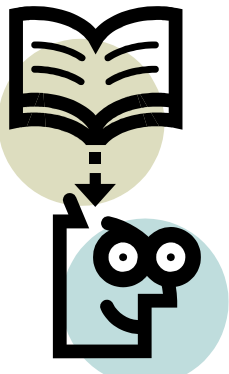
Datele tabeli MF2

3. Care sunt datele din tabela MF1, ordonate alfabetic după denumire?
4. Care sunt datele din tabela MF2, ordonate alfabetic după denumire și apoi după valoare?
5. Afișați datele celor două tabele într-un query

- 
6. Care sunt mijloacele fixe ce se găsesc în MF1 și nu se găsesc în MF2?
  7. Care sunt mijloacele fixe ce se găsesc și în MF1 și în MF2?
  8. Care este valoarea medie a mijloacelor fixe din MF1?
  9. Care este valoarea medie a mijloacelor fixe din MF2 a căror nume începe cu litera C?
  10. Care sunt mijloacele fixe din tabela MF2 furnizate de furnizorii cu codurile: c1, c2, c4, c5,c8?

# TEMA 9

## THETA ȘI ECHI-JONCȚIUNEA



**Principalele obiective ale capitolului sunt:**

Realizarea unei legături între două sau mai multe  
tabele

Utilizarea condițiilor în echi-joncțiune

SQL nu prezintă clauze sau operatori speciali pentru joncțiune, joncțiunea fiind o combinație între produs cartezian și selecție.

Deci pentru theta-joncțiunea dintre două relații R1 și R2 avem:

```
select * from R1, R2 where R1.A>=R2.E
```

unde:

R1			R2		
A	B	C	C	D	E
20	XYZ	30	25	XYZ	30
30	XXZ	20	40	YYX	25
40	YYX	25	30	XXZ	40

iar pentru echi-joncțiune avem:

```
select * from R1, R2 where R1.A=R2.E
```

Varianta cu INNER JOIN (joncțiune internă) este:

```
select * from R1 inner join R2 on R1.A>=R2.E
```

respectiv

```
select * from R1 inner join R2 on R1.A=R2.E
```

Pentru următorul exemplu utilizăm următoarele tabele:  
FURNIZORI, FACTURI cu următoarele date:

Furnizori				
	Cod_furn	Nume_furn	Adresa	Localitate
	c1	ALFA	ALUNULUI 3	BRASOV
	c2	GAMA	REPUBLICII 34	BRASOV
	c3	PLUTO	LUNGA 20	CLUJ
	c4	MERCUR	DACIA 5	BRASOV
	c5	ORIZONT	ZIZINULUI 3	BRASOV
	c6	ONIRIS	VLAHUTA 332	BRASOV
	c7	JEO	CALEA BUCURESTI 33	BRASOV
	c8	BONNY	CALEA MOSILOR 4	BUCURESTI
	n12	KEONE	CADEREA BASTILIEI 2	BUCURESTI
	m5	STAR	BARITIU 234	BRASOV
	c9	PLANET	TOMIS 43	CONSTANTA
	m1	SUN	MAMAIA 78	CONSTANTA
	m3	COSMOS	BALCESCU 34	BRASOV
	n10	GRIG	SLAVICI 43	BUCURESTI
	n9	POMPI	NEGOIU 778	BRASOV
	n7	VEZUVIU	CLOSCA 4	CLUJ

Facturi				
	Nr_factura	Cod_furn	Data_fact	Valoare
	1002	c1	01/12/06	600
	1003	c2	01/14/06	1800
	1004	c1	01/14/06	3000
	1005	c4	01/14/06	2100
	1006	c1	01/16/06	2300
	1007	c5	01/17/06	5600
	1008	c9	01/18/06	8900
	1009	c2	02/01/06	8000
	1010	m5	02/02/06	5000
	1011	n12	02/02/06	600

Se cere să se afișeze pentru fiecare factură numele furnizorului.

```
select facturi.*, furnizori.ume_furn from facturi, furnizori where
facturi.cod_furn=furnizori.cod_furn
```

Query					
	Nr_factura	Cod_furn	Data_fact	Valoare	Nume_furn
	1002	c1	01/12/06	600	ALFA
	1006	c1	01/16/06	2300	ALFA
	1004	c1	01/14/06	3000	ALFA
	1009	c2	02/01/06	8000	GAMA
	1003	c2	01/14/06	1800	GAMA
	1005	c4	01/14/06	2100	MERCUR
	1007	c5	01/17/06	5600	ORIZONT
	1011	n12	02/02/06	600	KEONE
	1010	m5	02/02/06	5000	STAR
	1008	c9	01/18/06	8900	PLANET

Numele furnizorului pentru fiecare factură

Același rezultat se poate obține utilizând INNER JOIN:

```
select facturi.*, furnizori.ume_furn from facturi inner join furnizori ON
facturi.cod_furn=furnizori.cod_furn
```

Query					
	Nr_factura	Cod_furn	Data_fact	Valoare	Nume_furn
	1002	c1	01/12/06	600	ALFA
	1006	c1	01/16/06	2300	ALFA
	1004	c1	01/14/06	3000	ALFA
	1009	c2	02/01/06	8000	GAMA
	1003	c2	01/14/06	1800	GAMA
	1005	c4	01/14/06	2100	MERCUR
	1007	c5	01/17/06	5600	ORIZONT
	1011	n12	02/02/06	600	KEONE
	1010	m5	02/02/06	5000	STAR
	1008	c9	01/18/06	8900	PLANET

Numele furnizorului pentru fiecare factură

Dacă dorim să afișăm numele furnizorului pentru fiecare factură numai pentru furnizorul cu cod\_furn "c1" vom avea:

```
select facturi.*, furnizori.nume_furn from facturi, furnizori where facturi.cod_furn=furnizori.cod_furn and facturi.cod_furn="c1"
```

Query					
	Nr_factura	Cod_furn	Data_fact	Valoare	Nume_furn
	1006	c1	01/16/06	2300	ALFA
	1004	c1	01/14/06	3000	ALFA
	1002	c1	01/12/06	600	ALFA

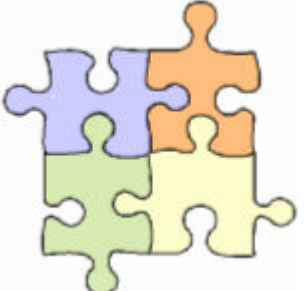
Numele furnizorului pentru fiecare factură numai pentru furnizorul cu cod\_furn "c1"



### Intrebări:

1. De ce două tabele sunt legate logic prin valori?
2. Care este condiția necesară pentru echi-joncțiune?



	<p style="text-align: center;"><b>Studiu individual</b></p> <p>APLICAȚIA 9: Evidența salariaților dintr-o firmă</p>
---	---

## APLICAȚIA 9

1. Să se realizeze tabelele ANGAJATI și DEPARTAMENTE după structura tabelor ANGAJATI și DEPARTAMENTE din ANEXA 1
2. Introduceți în tabele următoarele date:

Departamente					
	Cod_dep	Den	Localitate	Sef	Telefon
	info	informatica	Brasov	vlad	125566
	cont	contabilitate	Bucuresti	Ioana	5636363
	mk	marketing	Brasov	Grigore	313002
	mn	management	Constanta	Ana	566627

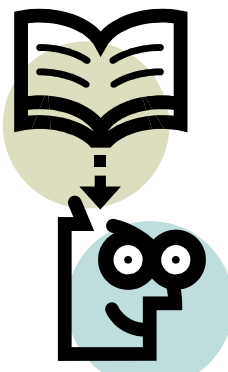
Angajati						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
	1	Popescu	Ana	01/31/80	600	info
	2	Ionescu	Dana	12/17/78	850	cont
	3	Iordache	Ion	02/23/67	1000	info
	4	Mihalcea	Dan	06/24/84	400	mk
	5	Dobrescu	Mircea	04/04/80	740	info
	6	Dancea	Mirela	12/21/78	500	mn
	7	Pavel	Maria	12/18/67	360	cont
	8	Grigorescu	Ana	02/06/78	500	mn
	9	Enescu	Ioan	05/12/80	550	mk
	10	Varga	Ion	05/08/81	390	info

3. Afișați șeful pentru fiecare angajat.
4. Care sunt angajații care lucrează în Brașov?
5. Câți angajați lucrează în București?
6. Care este numele angajatului cu salariul maxim?
7. Care este fondul de salarii pentru fiecare cod\_dep?

- 
8. Care sunt angajații angajați după 01/01/1980?
  9. Care sunt angajații care lucrează la departamentul marketing?
  10. Câți subalterni are Vlad?
  11. Care este salariul mediu pentru departamentul management?
  12. Care sunt angajații a căror șef începe cu litera "A"?
  13. Care sunt angajații care lucrează în Brașov sau București?

# TEMA 10

## GRUPAREA TUPLURILOR: GROUP BY ȘI HAVING

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li>Utilizarea grupurilor</li><li>Utilizarea condițiilor pentru grupuri</li></ul>
--	--

Clauza GROUP BY formează grupe (grupuri) de tupluri ale unei relații, pe baza valorilor comune ale unui atribut. În frazele SELECT formulate până în acest paragraf, prin intermediul WHERE au fost selectate tupluri ale tabelului. Prin asocierea unei clauze HAVING la GROUP BY este posibilă selectarea anumitor grupuri de tupluri ce îndeplinesc un criteriu, valabil numai la nivel de grup (nu și la nivel de linie).

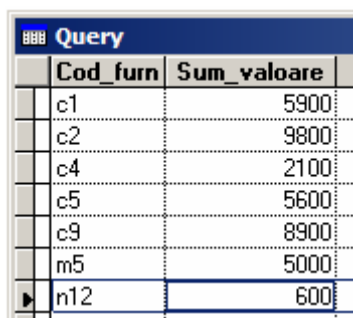
Clauza unei fraze SELECT ce conține această clauză se obține prin regruparea tuturor liniilor din tabelele enumerate în FROM, extrăgându-se câte o apariție pentru fiecare valoare distinctă a coloanei/grupului de coloane.

Formatul general este:

```
select coloană1, coloană2,...,coloană m from tabelă group by  
coloană_de_regrupare
```

De exemplu dacă dorim să vedem care este valoarea totală a facturilor pentru fiecare cod\_furn vom avea:

```
select cod_furn, sum(valoare) from facturi group by cod_furn
```



Cod_furn	Sum_valoare
c1	5900
c2	9800
c4	2100
c5	5600
c9	8900
m5	5000
n12	600

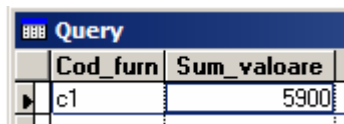
Valoarea totală a facturilor pentru fiecare cod\_furn

Cea mai simplă definiție a clauzei HAVING este: clauza HAVING este WHERE-ul ce operează la nivel de grupuri. Dacă WHERE acționează la nivel de tuplu, selectând acele linii care îndeplinesc o condiție specificată, HAVING permite specificarea unor condiții de selecție care se aplică tuplurilor de linii create prin GROUP BY.

Din rezultat sunt eliminate toate grupurile care nu satisfac condiția specificată.


De exemplu dacă dorim să vedem care este valoarea totală a facturilor pentru cod\_furn egal cu "c1" vom avea:

```
select cod_furn, sum(valoare) from facturi group by cod_furn having  
cod_furn="c1"
```



Cod_furn	Sum_valoare
c1	5900

Valoarea totală a facturilor pentru  
cod\_furn egal cu "c1"

	<p><b>Intrebări:</b></p> <p>Când se utilizează GROUP BY? Care este definiția condiției HAVING?</p>
---	--

	<p><b>Studiu individual</b></p> <p>APLICAȚIA 10: Situații de grup la nivelul salariaților</p>
---	---

## APLICAȚIA 10

1. Să se realizeze tabelele ANGAJATI și DEPARTAMENTE după structura tabelor ANGAJATI și DEPARTAMENTE din ANEXA 1
2. Introduceți în tabele următoarele date:

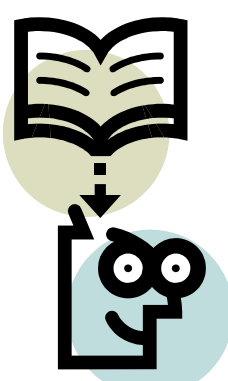
Departamente					
	Cod_dep	Den	Localitate	Sef	Telefon
	info	informatica	Brasov	vlad	125566
	cont	contabilitate	Bucuresti	Ioana	5636363
	mk	marketing	Brasov	Grigore	313002
	mn	management	Constanta	Ana	566627

Angajati						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
	1	Popescu	Ana	01/31/80	600	info
	2	Ionescu	Dana	12/17/78	850	cont
	3	Iordache	Ion	02/23/67	1000	info
	4	Mihalcea	Dan	06/24/84	400	mk
	5	Dobrescu	Mircea	04/04/80	740	info
	6	Dancea	Mirela	12/21/78	500	mn
	7	Pavel	Maria	12/18/67	360	cont
	8	Grigorescu	Ana	02/06/78	500	mn
	9	Enescu	Ioan	05/12/80	550	mk
	10	Varga	Ion	05/08/81	390	info

3. Afișați șeful pentru fiecare angajat.
4. Care este salariul total pentru fiecare departament?
5. Care este salariul total pentru departamentul MANAGEMENT?
6. Care este suma totală a salariilor din departamentele management și contabilitate?
7. Care este numărul de salariați din fiecare departament?
8. Care este departamentul cu angajații cei mai mulți?
9. Care este șeful departamentului cu cei mai mulți angajați?
10. Câți angajați are Vlad?

# TEMA 11

## REALIZAREA FORMULARELOR

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li>Utilizarea formularelor</li><li>Utilizarea generatoarelor wizard pentru crearea formularelor</li><li>Rularea formularelor</li></ul>
--	--

### UTILIZAREA FORMULARELOR

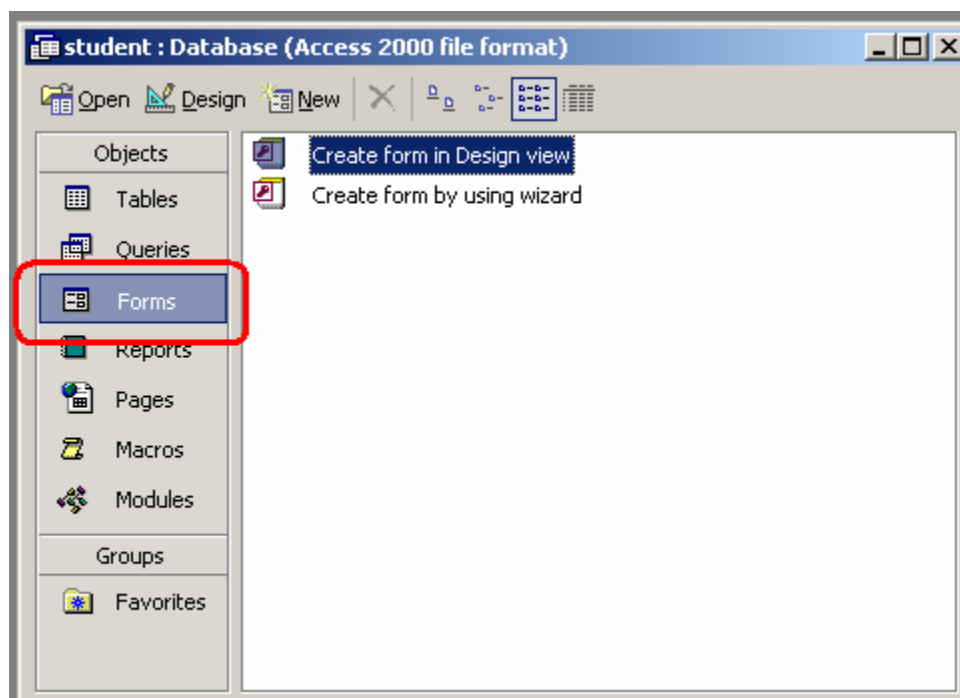
Formularele sunt ecrane cu ajutorul cărora se pot introduce date în tabele sau se pot lansa diferite comenzi și aplicații.

De obicei, formularele conțin casete în care se poate scrie text, butoane prin care se pot lansa comenzi sau aplicații, elemente de tip listă pentru alegerea unor valori sau opțiuni de tip check-box.

Pentru facilitarea lucrului cu formulare, programul dispune de un generator de tip wizard prin care se pot crea formulare de introducere a datelor în tabele.

Formularele pot fi făcute atât pe baza unor tabele existente cât și pe baza unor interogări deja definite (QUERIES).

Pentru realizarea formulelor din panoul de comandă se alege opțiunea FORMS din cadrul meniului OBJECTS.

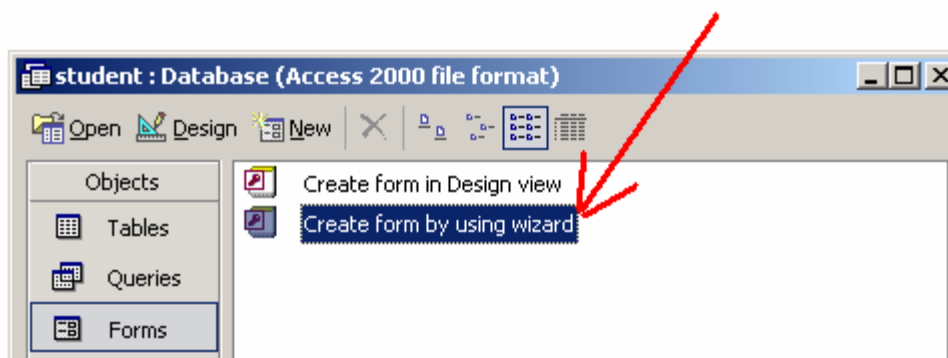


Alegerea opțiunii FORMS

## UTILIZAREA GENERATOARELOR WIZARD PENTRU CREAREA FORMULELOR

Pentru utilizarea generatorului de formulare în panoul de comandă se alege opțiunea CREATE FORM BY USING WIZARD.

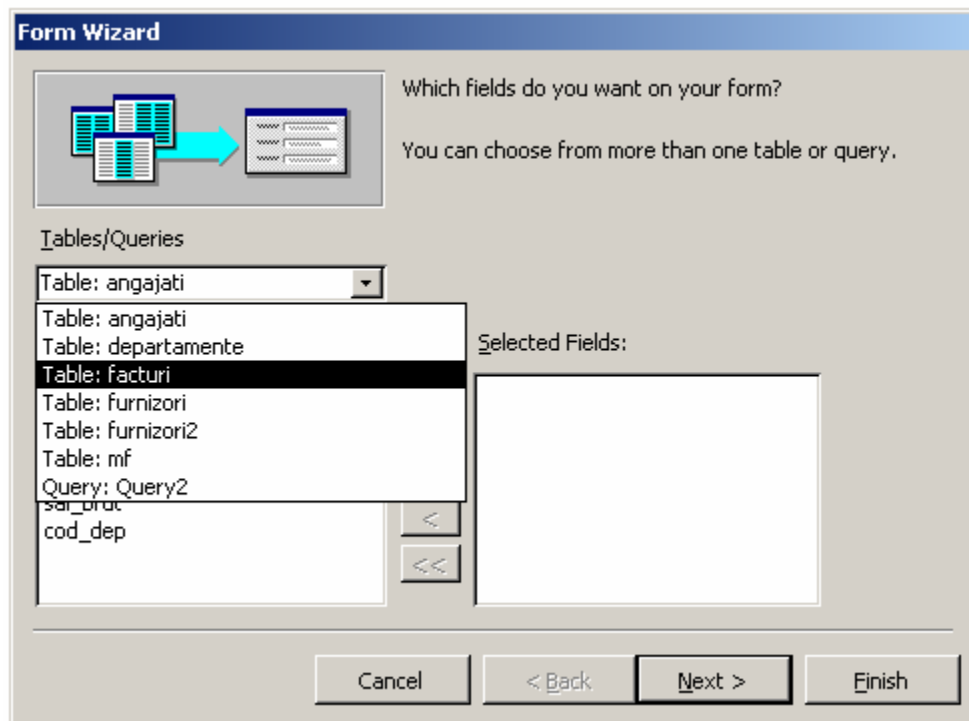
Dacă utilizatorul dorește să creeze formulare manual va alege opțiunea CRAETE FORM IN DESIGN VIEW.



Utilizarea generatorului de formulare

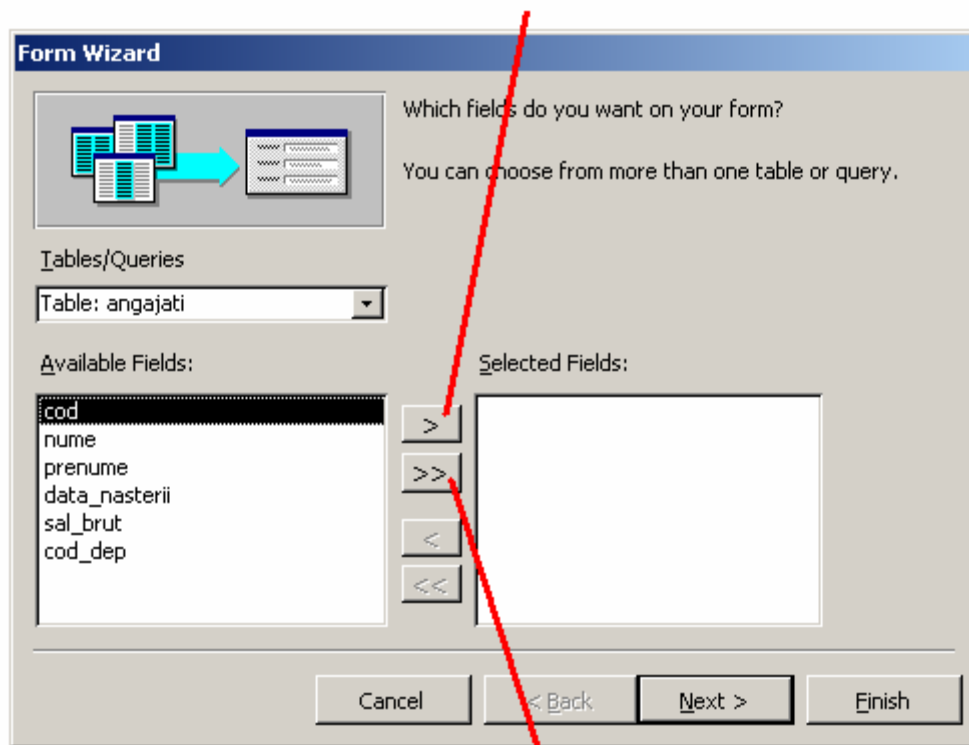


În fereastra FORM WIZARD primul pas în crearea formularului este alegerea tabelului în care vor fi introduse datele. Tabela dorită se selectează prin opțiunea TABLES / QUERIES.



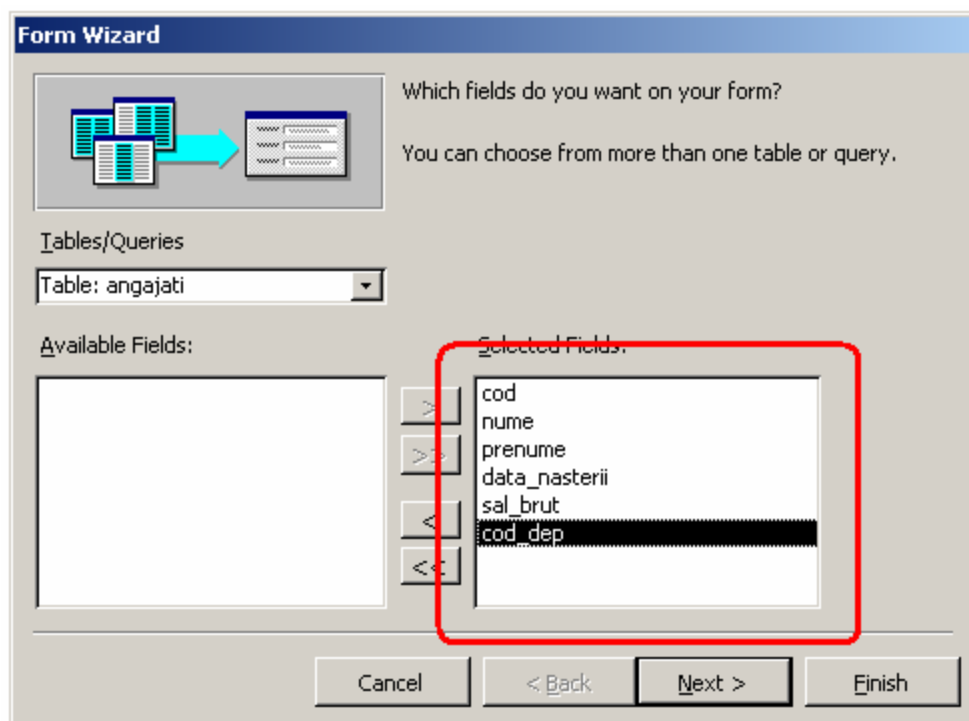
Selectarea tabelii

Cel de-al doilea pas este selectarea câmpurilor în care se vor introduce date. Selectarea câmpurilor poate făcută separat pentru fiecare câmp (oferind posibilitatea ca unul sau mai multe câmpuri să lipsească) sau pentru toate câmpurile.

**Adăugarea în formular doar a câmpului selectat****Adăugarea în formular a tuturor câmpurilor**

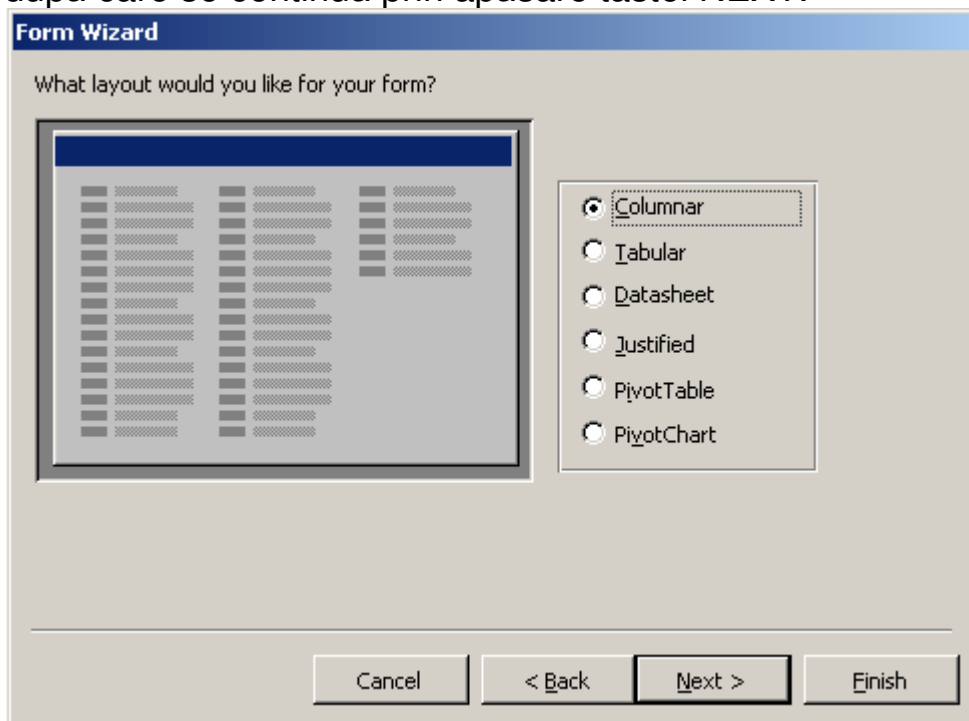
Selectarea câmpurilor dorite în formular

Câmpurile selectate se mută în ecranul din partea dreaptă. În exemplul de mai jos au fost selectate toate câmpurile. Se vor selecta elementele dorite după care se continuă prin apăsare tastei NEXT.



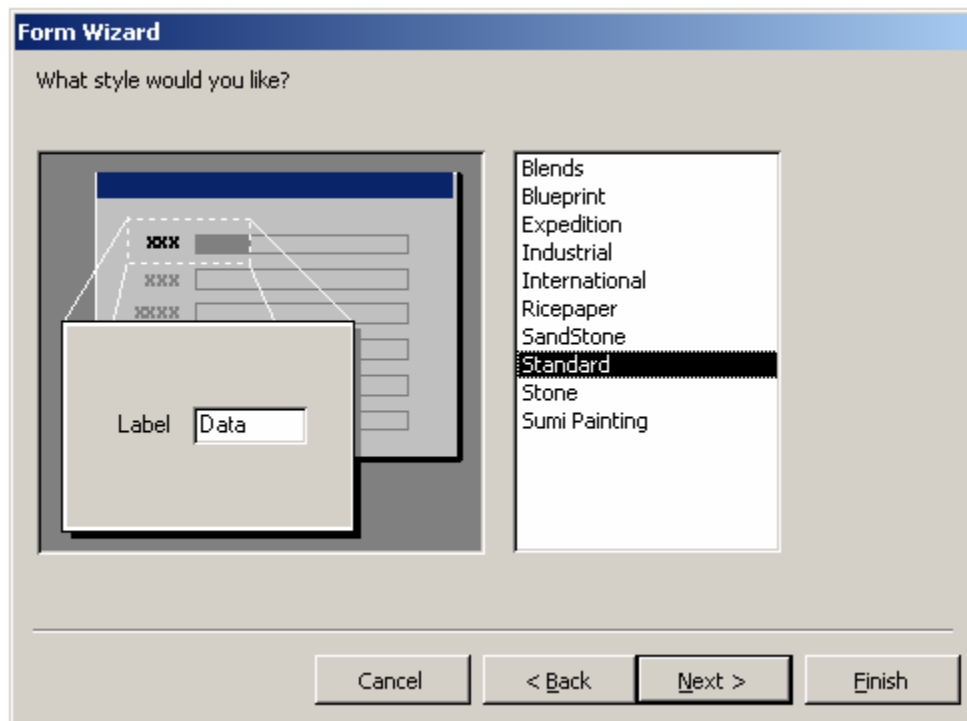
Câmpurile selectate

În continuare se alege modul de afișare a câmpurilor în formular (pe coloane, tabular, foaie de lucru, etc.). Se va selecta tipul dorit după care se continuă prin apăsare tastei NEXT.



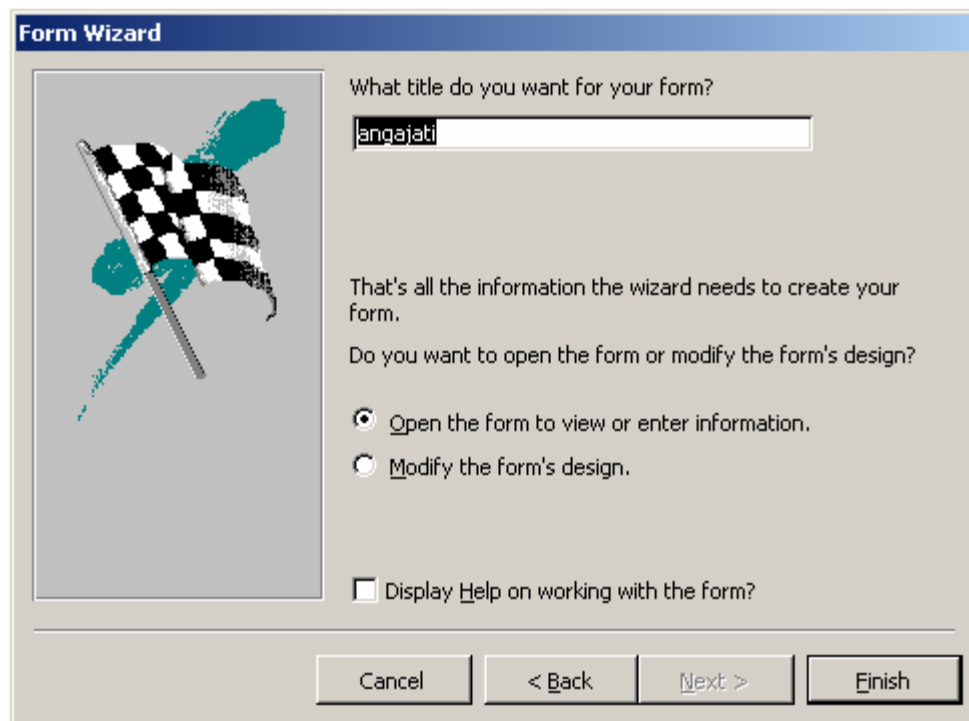
### Selectarea modului de afișare

Din punct de vedere grafic sunt disponibile mai multe tipuri de formulare. Se va selecta tipul dorit după care se continuă prin apăsare tastei NEXT.



Alegerea stilului formularului

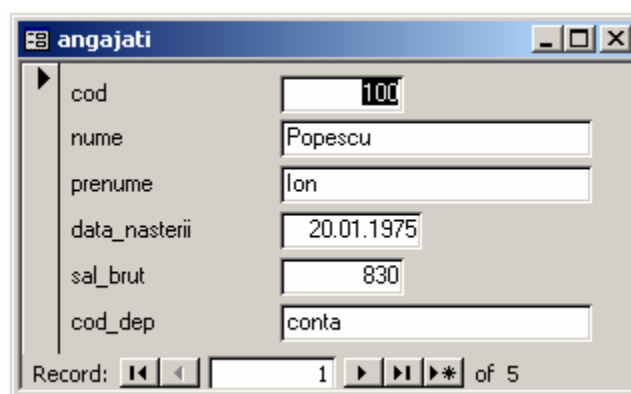
Ultimul pas în realizarea formularelor este denumirea formularului creat. Implicit numele este cel al tabelii în care vor fi scrise datele.



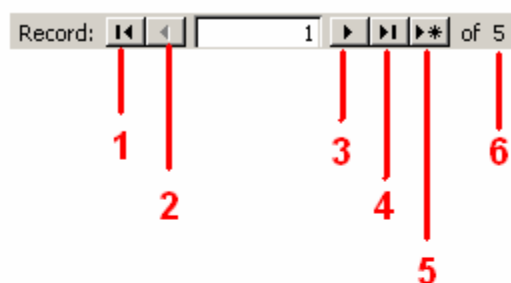
Denumirea formularului

## RULAREA FORMULARULUI

Pentru a rula un formular se da dublu click pe numele sau în fereastra de comandă.



Rularea formularului ANGAJATI

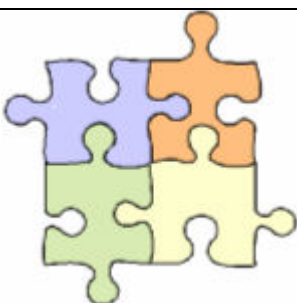


- 1 Prima înregistrare**
- 2 Înregistrarea precedentă**
- 3 Următoarea înregistrare**
- 4 Ultima înregistrare**
- 5 Înregistrare nouă**
- 6 Număr total de înregistrări**



### Intrebări:

Pentru ce se utilizează formularele?  
Care sunt pași necesari în conceperea unui formular?



### Studiu individual

APLICAȚIA 11: Introducerea datelor în tabele cu ajutorul formularelor

**APLICAȚIA 11**

1. Să se realizeze tabelele ANGAJATI și DEPARTAMENTE după structura tabelor ANGAJATI și DEPARTAMENTE din ANEXA 1
2. Realizați formulare pentru introducerea datelor în cele două tabele
3. Introduceți în tabele următoarele date:

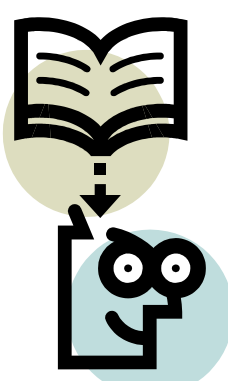
Departamente					
	Cod_dep	Den	Localitate	Sef	Telefon
	info	informatica	Brasov	vlad	125566
	cont	contabilitate	Bucuresti	Ioana	5636363
	mk	marketing	Brasov	Grigore	313002
	mn	management	Constanta	Ana	566627

Angajati						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
	1	Popescu	Ana	01/31/80	600	info
	2	Ionescu	Dana	12/17/78	850	cont
	3	Iordache	Ion	02/23/67	1000	info
	4	Mihalcea	Dan	06/24/84	400	mk
	5	Dobrescu	Mircea	04/04/80	740	info
	6	Dancea	Mirela	12/21/78	500	mn
	7	Pavel	Maria	12/18/67	360	cont
	8	Grigorescu	Ana	02/06/78	500	mn
	9	Enescu	Ioan	05/12/80	550	mk
	10	Varga	Ion	05/08/81	390	info

4. Care este salariul total pentru fiecare departament?
5. Care este salariul total pentru departamentul MANAGEMENT?
6. Care este suma totală a salariilor din departamentele management și contabilitate?
7. Care este numărul de salariați din fiecare departament?
8. Care este departamentul cu angajații cei mai mulți?
9. Care este șeful departamentului cu cei mai mulți angajați?
10. Câți angajați are Vlad?

# TEMA 12

## REALIZAREA RAPOARTELOR

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Rolul rapoartelor Modul de realizarea a rapoartelor</p>
--	---

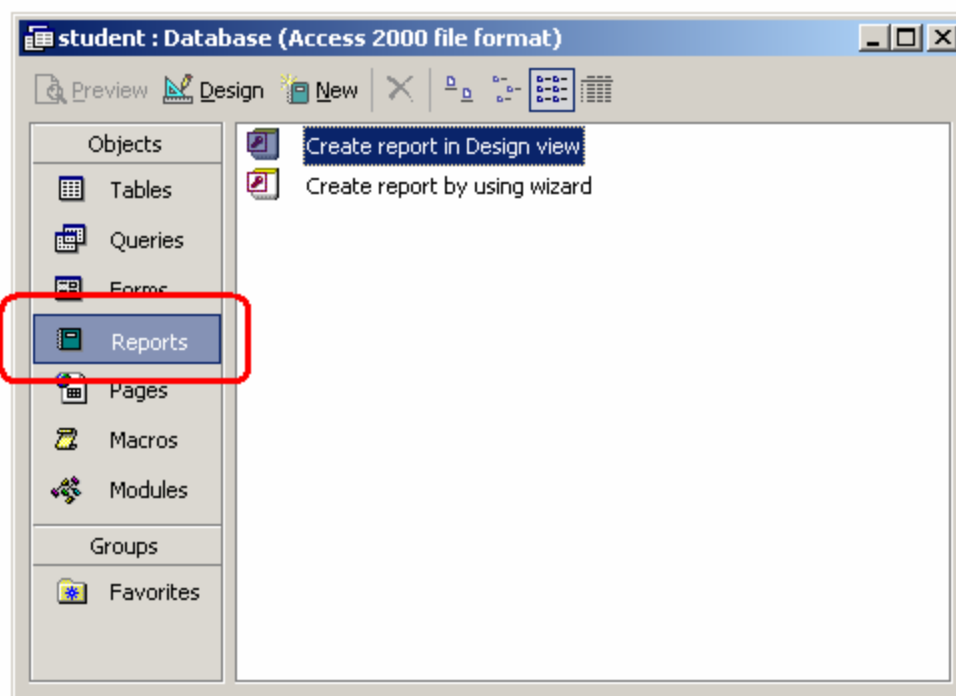
### UTILIZAREA RAPOARTELOR

Rapoartele sunt utilizate pentru afișarea estetică a datelor dintr-o tabelă sau query.

Primul pas în realizarea unui raport îl reprezintă selectarea opțiunii REPORTS din cadrul meniului OBJECTS.

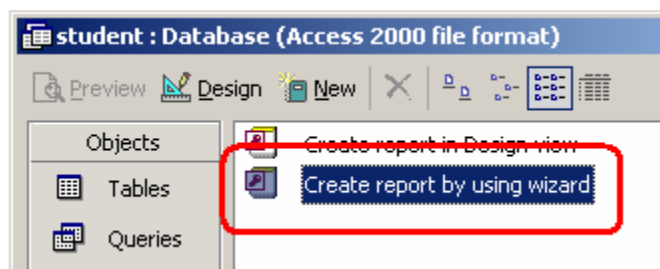
Și pentru realizarea rapoartelor avem la dispoziție posibilitatea realizării manual sau în mod asistat de program cu ajutorul generatorului de tip wizard.





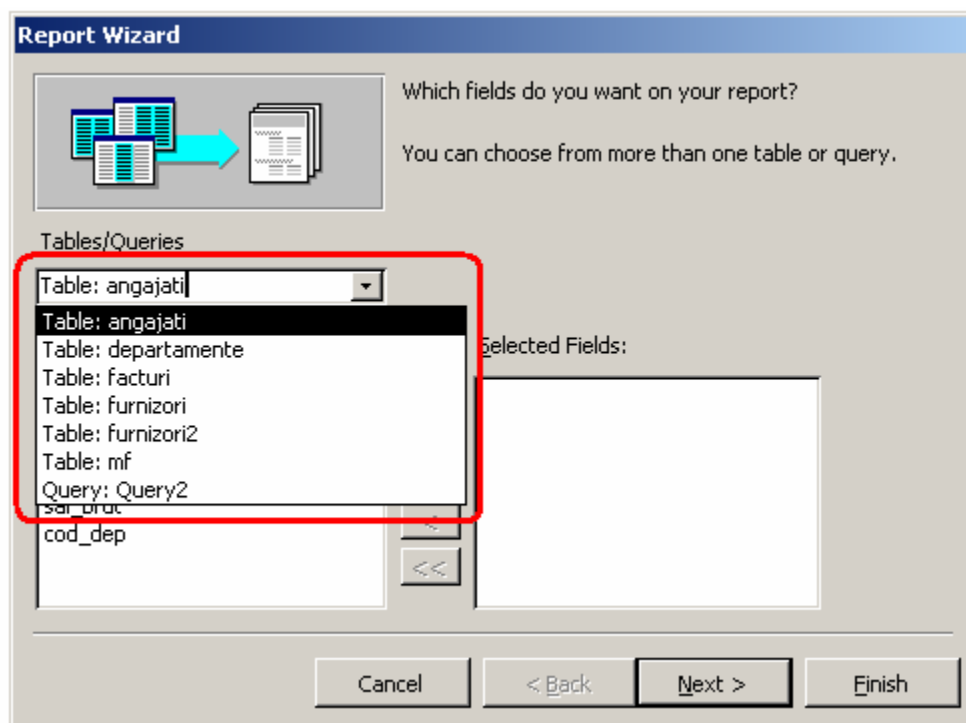
Selectarea opțiunii REPORTS

Pentru facilitarea lucrului vom utiliza generatorul de rapoarte. Acest lucru se face cu ajutorul opțiunii CREATE REPORT BY USING WIZARD.



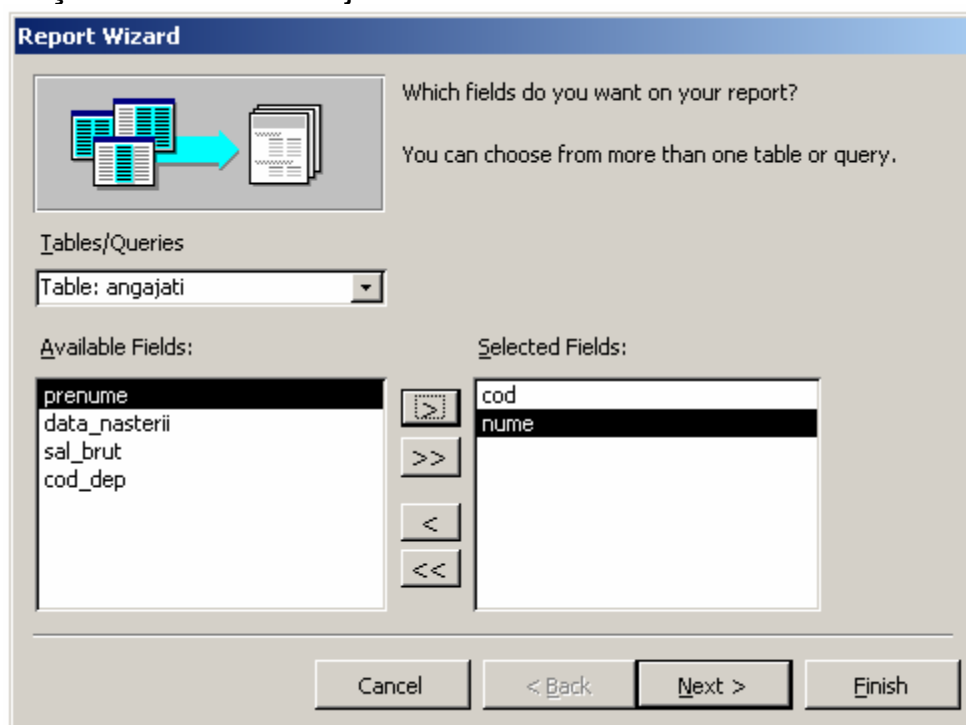
Selectarea generatorului de rapoarte

În realizarea raportului primul, în fereastra REPORT WIZARD trebuie selectată tabela pentru care se vor afișa datele.



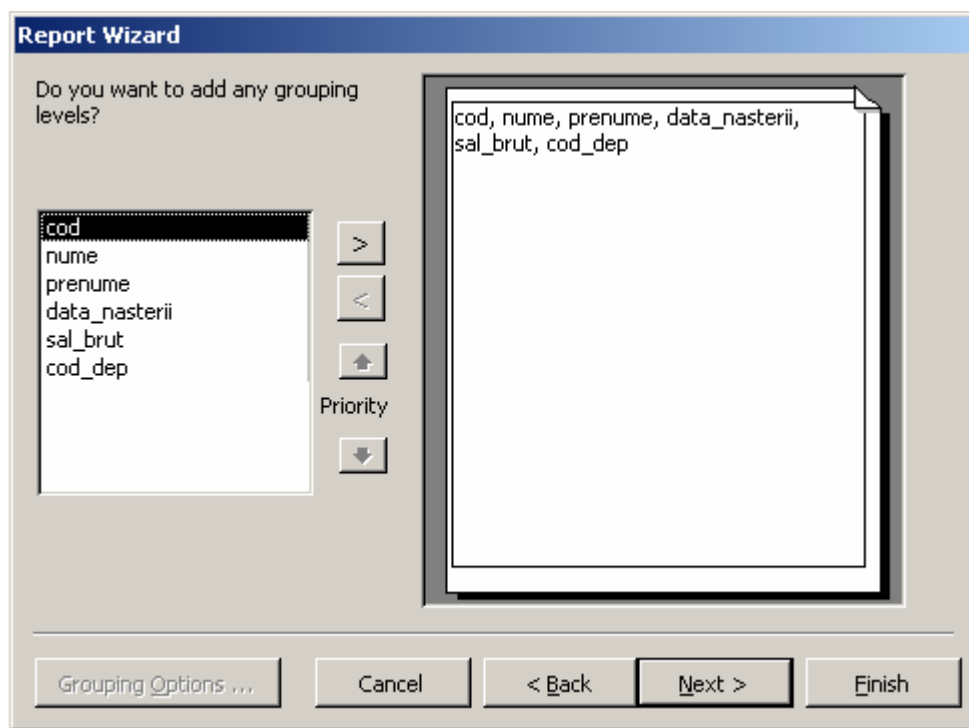
Selectarea tabelii

Similar creării formularelor, trebuie precizate câmpurile care vor fi afișate din tabela deja selectată.



Selectarea câmpurilor din tabelă

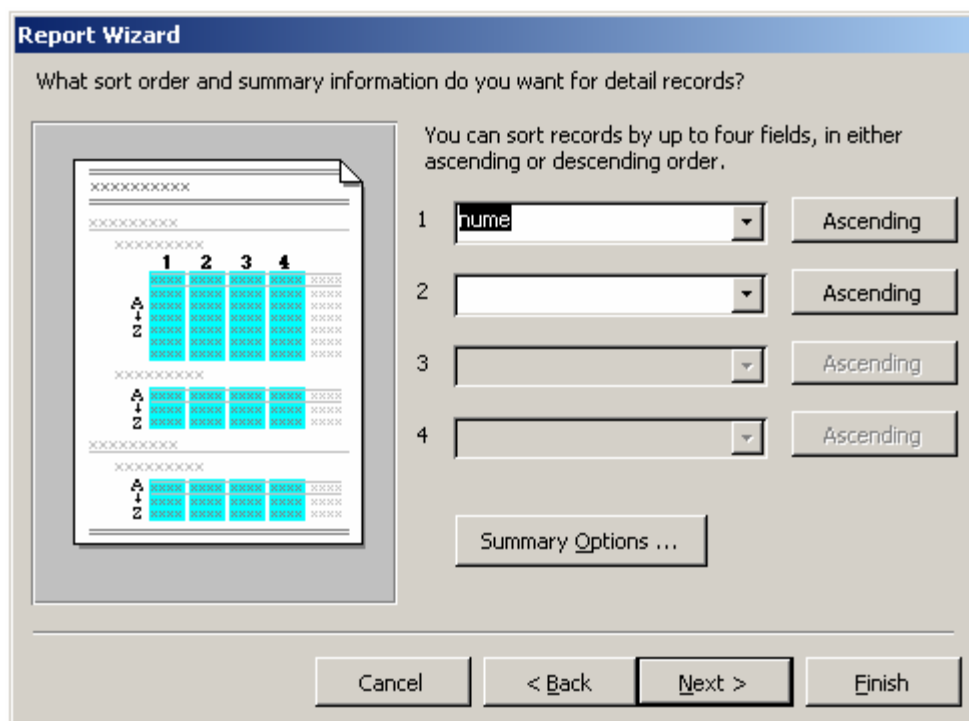
Pentru câmpurile selectate se poate face o grupare, în momentul afișării datelor după un câmp.



Gruparea datelor după un câmp

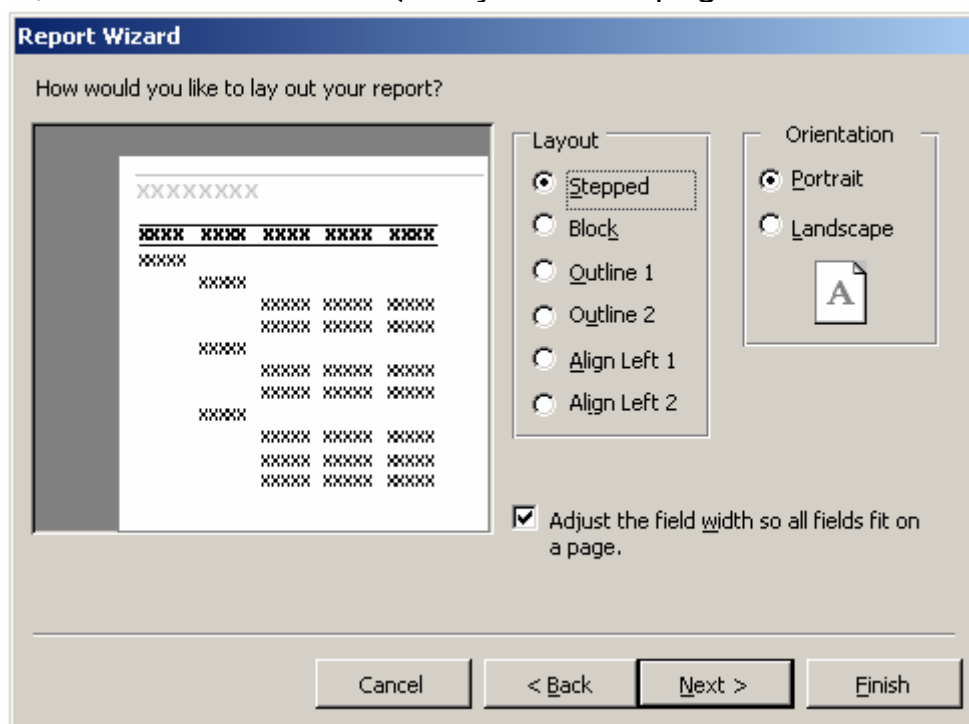
Totodată datele ce vor fi afișate pot fi ordonate după unul sau mai multe criterii.

Aceste criterii sunt alfabetice sau invers-alfabetice, crescătoare sau descrescătoare, în funcție de tipul câmpului text sau numeric.



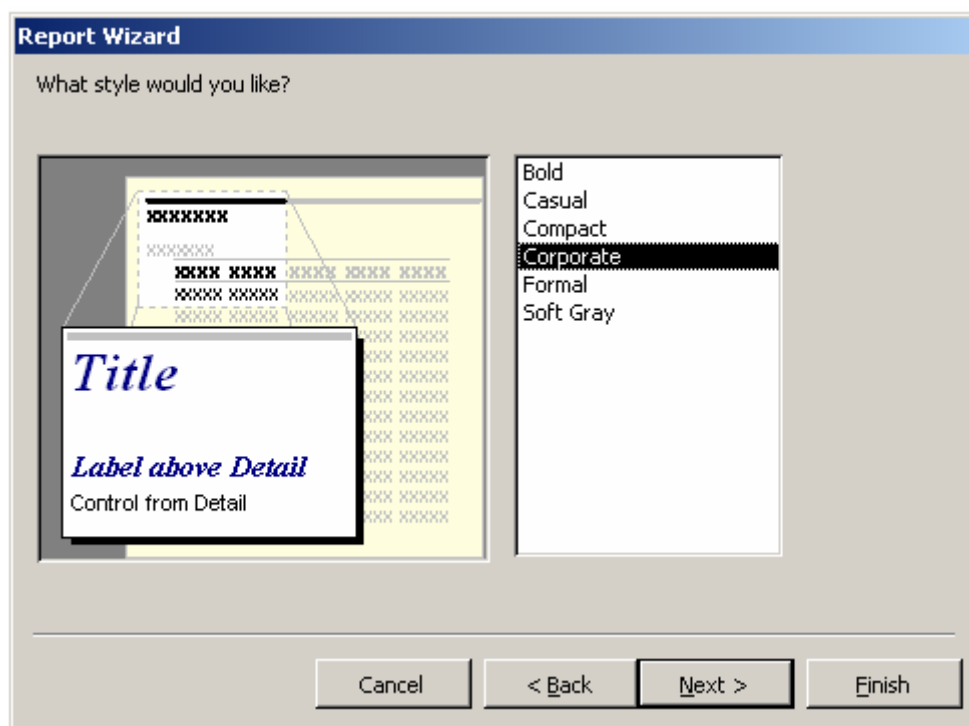
Selectarea criteriilor de ordonare a datelor

Următorul pas este alegerea unei forme de prezentare a datelor, de aliniere a informațiilor și formatul paginii.



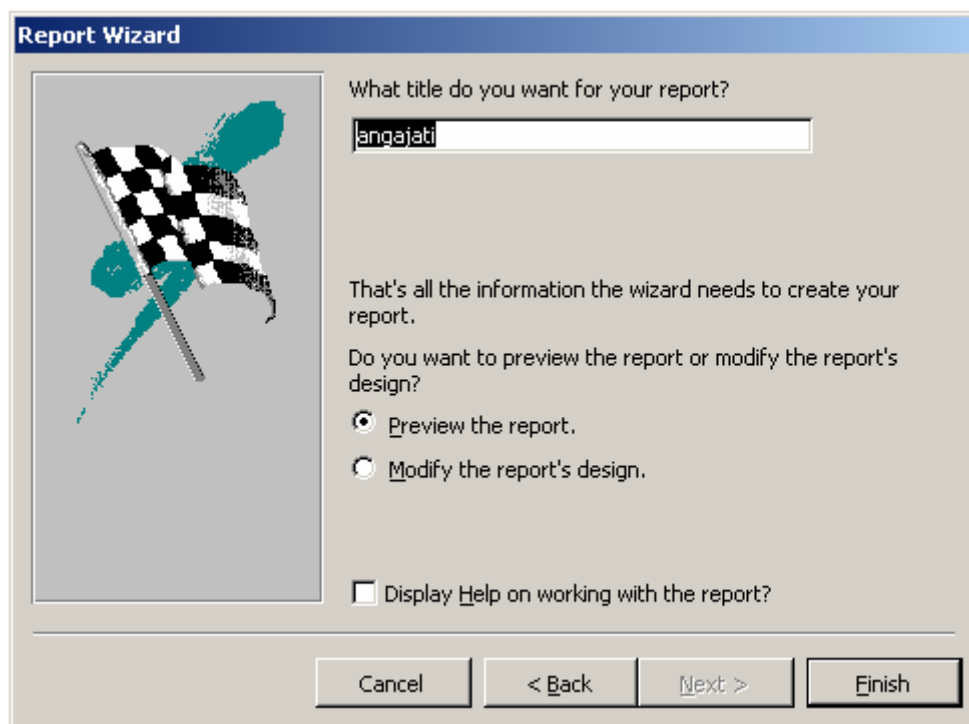
Alegerea formatului de prezentare a datelor

Tot în legătură cu forma de afișare a datelor este și stilul raportului. Pentru alegerea stilului se selectează din meniul din partea din dreapta opțiunea dorită. Această opțiune este doar din punct de vedere estetică.



Selectarea stilului raportului

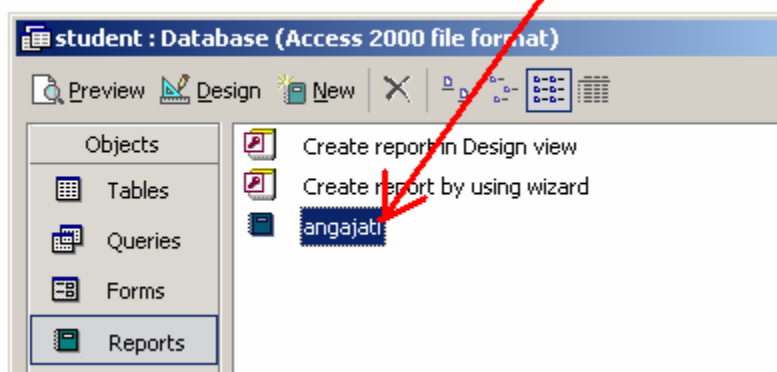
Ultimul pas în realizarea raportului este denumirea raportului realizat. Și în acest caz, denumirea implicită este cea a tabelului din care sunt luate datele. După tastarea noului nume se tastează FINISH. În acest moment raportul este gata și poate fi rulat.



Denumirea raportului

## EXECUȚIA RAPORTULUI

Pentru a rula un raport se da dublu click pe numele sau în fereastra de comandă.



Lansare în execuție a raportului

Raportul odată lansat este afișat pe ecran. Poate fi utilizată funcția zoom pentru mărire sau micșorare. Totodată raportul poate fi și listat.

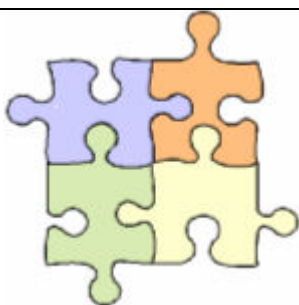
<i>angajati</i>				
<i>cod_dep</i>	<i>nume</i>	<i>cod_prenume</i>	<i>da_nasterii</i>	<i>sal_brut</i>
<i>conta</i>	Petrescu	104 Dana	30.09.1970	850
	Popescu	100 Ion	20.01.1975	830
<i>info</i>	Ionescu	102 Dan	25.06.1980	450
<i>mre</i>	Basescu	106 Ionela	06.06.1950	1200
	Oprea	105 Vasile	12.12.1960	1500

Raportul obținut



### Intrebări:

Ce este un raport?  
Cum se realizează un raport?



### Studiu individual

APLICAȚIA 13: Utilizarea rapoartelor

## APLICAȚIA 13

1. Realizați tabela SALARIATI pe baza structurii tabelului ANGAJATI din ANEXA 1.
2. În tabelă introduceți următoarele date:

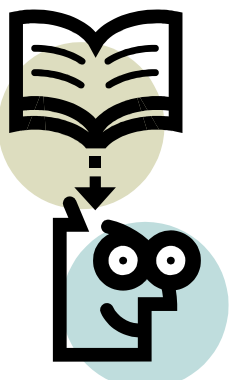
Angajati						
	Nr	Nume	Prenume	Data_nasterii	Sal_brut	Cod_dep
	1	Popescu	Ana	01/31/80	600	info
	2	Ionescu	Dana	12/17/78	850	cont
	3	Iordache	Ion	02/23/67	1000	info
	4	Mihalcea	Dan	06/24/84	400	mk
	5	Dobrescu	Mircea	04/04/80	740	info
	6	Dancea	Mirela	12/21/78	500	mn
	7	Pavel	Maria	12/18/67	360	cont
	8	Grigorescu	Ana	02/06/78	500	mn
	9	Enescu	Ioan	05/12/80	550	mk
	10	Varga	Ion	05/08/81	390	info

3. Realizați un raport pentru afișarea tuturor salariaților
4. Realizați un raport pentru afișarea tuturor salariaților, grupați pe departamente și ordonați alfabetic după nume.
5. Realizați un raport pentru afișarea numelui salariaților și a departamentului unde lucrează.



# TEMA 13

## CONSTRUCȚIA PAGINILOR WEB PENTRU LUCRUL CU BAZE DE DATE

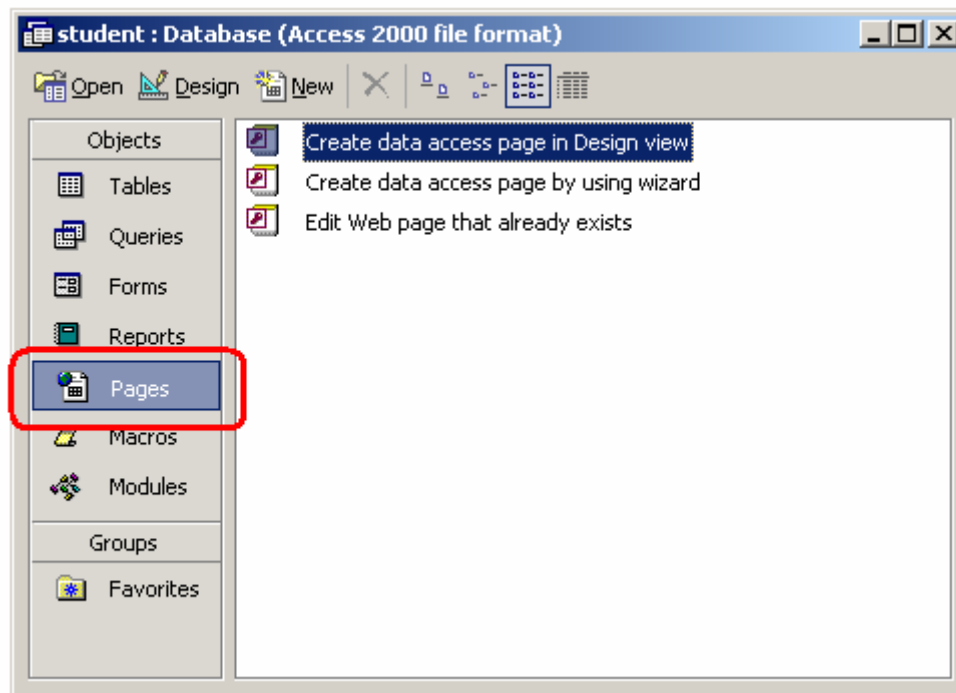
	<p><b>Principalele obiective ale capitolului sunt:</b></p> <p>Rolul formularelor de tip pagina web Realizarea paginilor web cu formulare pentru introducerea datelor</p>
--	--

### ROLUL PAGINILOR WEB CU FORMULAR PENTRU INTRODUCEREA DATELOR

Bazele de date și aplicațiile realizate în ACCESS sunt foarte mult utilizate pentru aplicații web. În consecință, pentru a putea introduce date în tabelele ACCESS este nevoie de pagini web ce pot fi publicate pe un server de web.

## REALIZAREA PAGINILOR WEB

La fel ca și în cadrul celorlalte module primul pas în realizare îl constituie selectarea opțiunii PAGES din cadrul meniului OBJECTS.

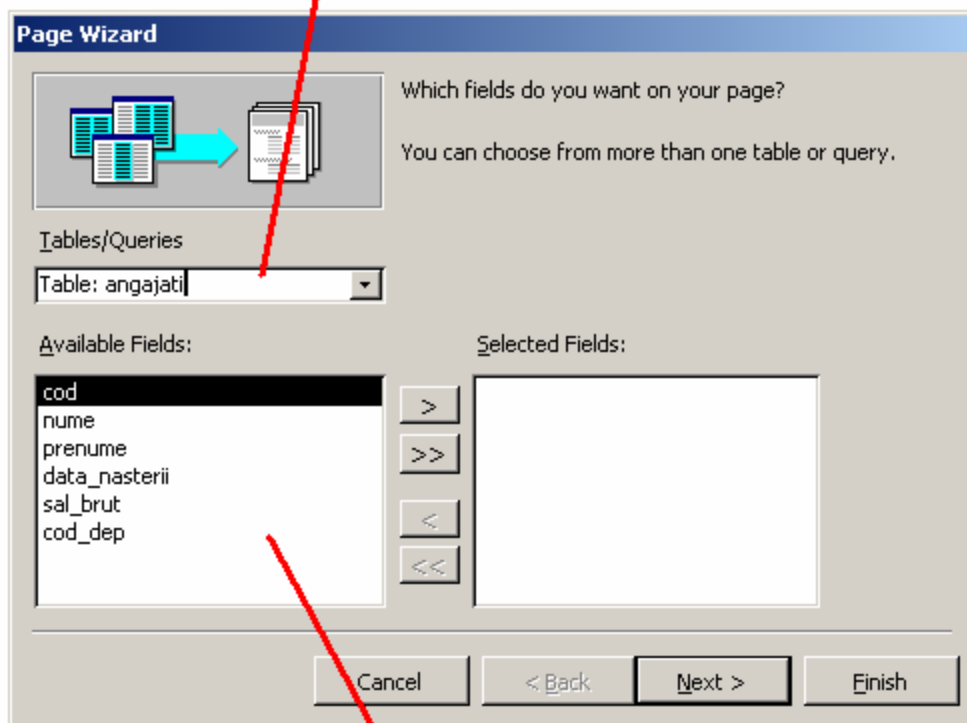


Selectarea opțiunii PAGES

Și pentru crearea formularelor în pagini web avem la dispoziție un generator de tip wizard.

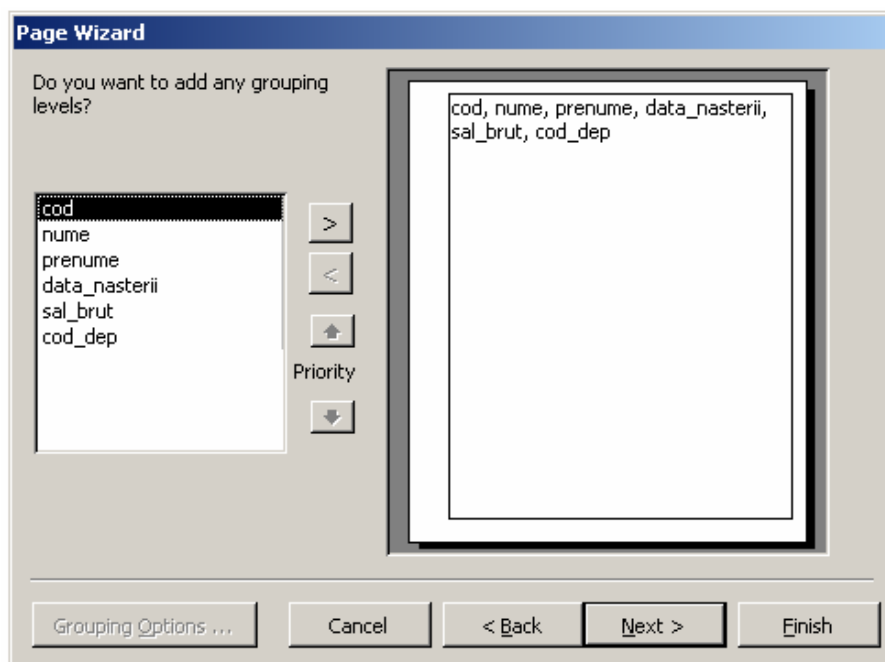
Pornirea generatorului se face prin selectarea opțiunii CREATE DATA ACCESS PAGE BY USING WIZARD.

La fel ca și în cazul formularelor și rapoartelor trebuie selectată tabela și câmpurile care vor apare in pagină.

**Selectarea tabelii****Selectarea câmpurilor dorite**

Selectarea tabelii și a câmpurilor

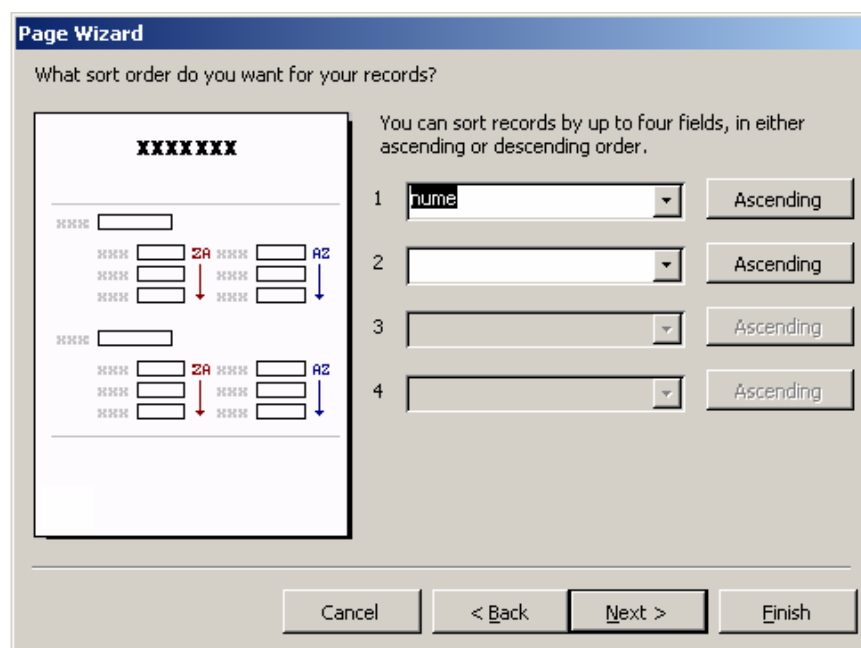
Datele pot fi grupate după unul din câmpurile tabelii care vor apare în pagină. Această grupare nu este însă obligatorie.



Gruparea datelor după un câmp

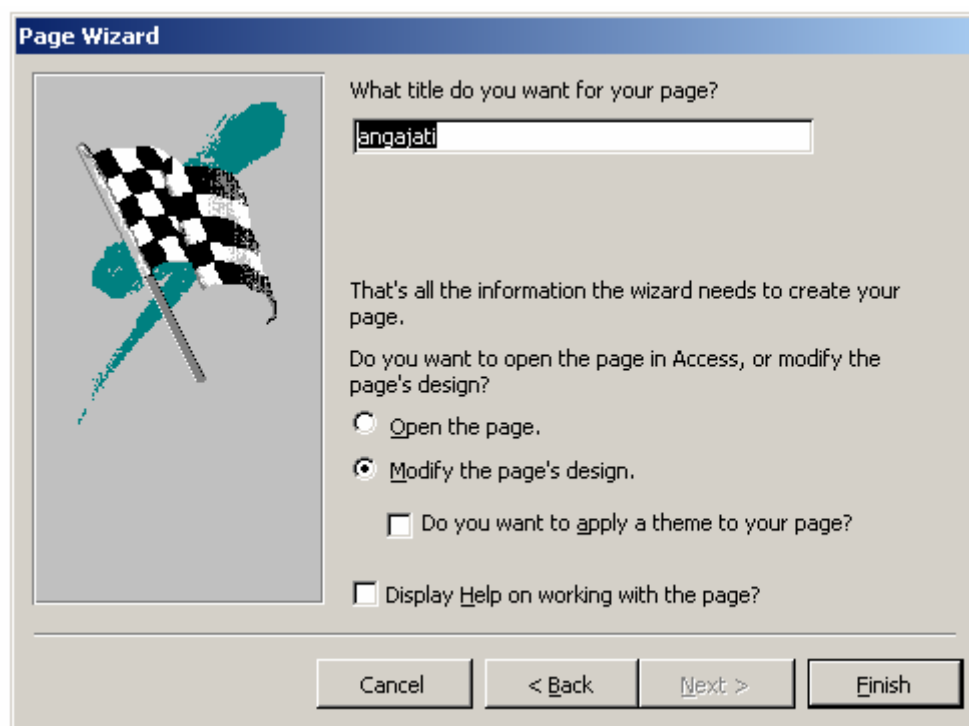
În continuare, tot opțional se pot alege câmpurile după care se va face ordonarea datelor.

Aceste criterii de ordonare sunt alfabetice sau invers-alfabetice, crescătoare sau descrescătoare, în funcție de tipul câmpului text sau numeric.



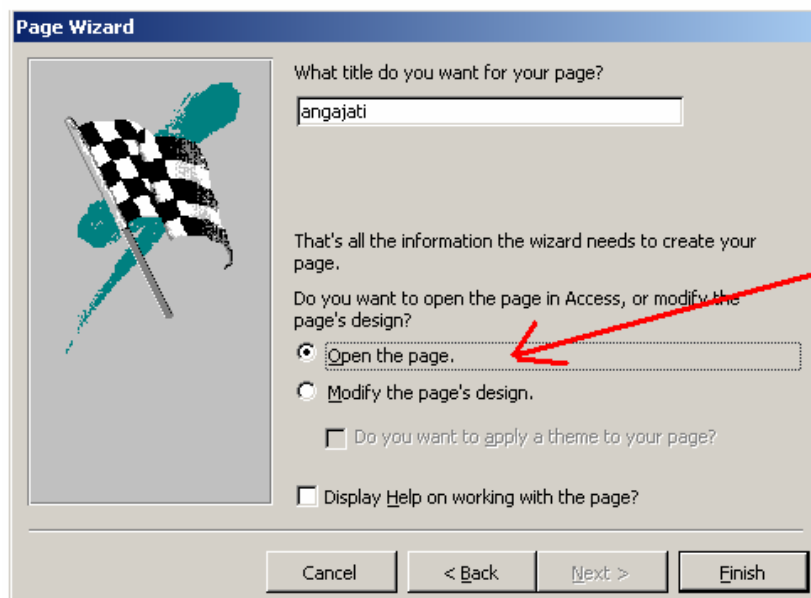
Ordonarea datelor

Ultimul pas în realizarea paginii este denumirea titlului paginii realizate. Și în acest caz, denumirea implicită este cea a tabelului din care sunt luate datele. După tastarea noului nume se tastează FINISH.



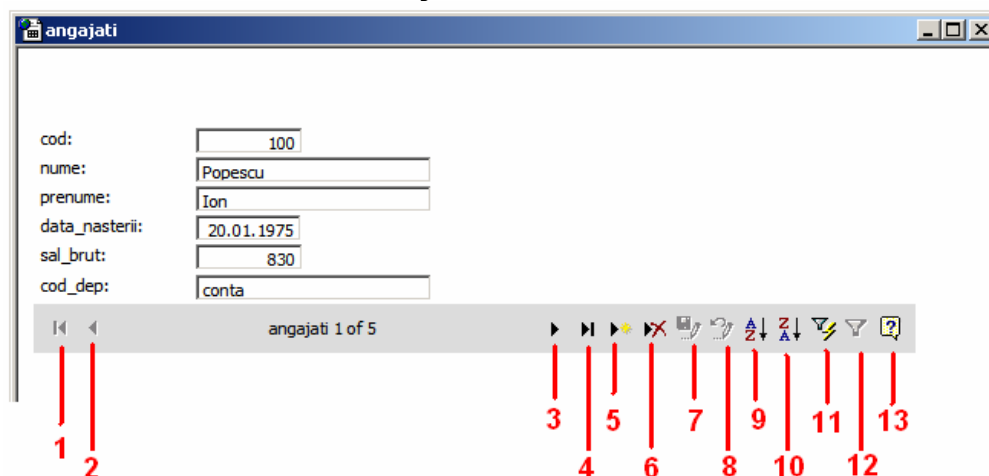
Titlul paginii

Pentru deschiderea paginii se alege opțiunea OPEN THE PAGE.



### Opțiunea OPEN THE PAGE

Vizualizarea paginii permite introducerea datelor asemănător unui formular obișnuit.

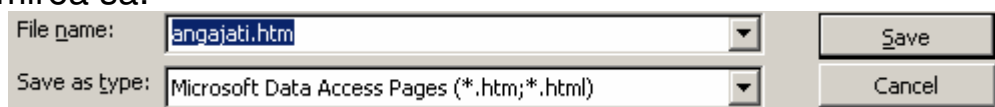


1. Prima înregistrare
2. Înregistrarea precedentă
3. Înregistrarea următoare
4. Ultima înregistrare
5. Înregistrare nouă
6. Șterge
7. Salvează

8. Undo
9. Ordonare alfabetică
10. Ordonare invers-alfabetică
11. Filtrare prin selecție
12. Filtrare
13. Ajutor

### Vizualizarea paginii


Pentru salvarea paginii este necesară închiderea acesteia și denumirea sa.



Salvarea paginii

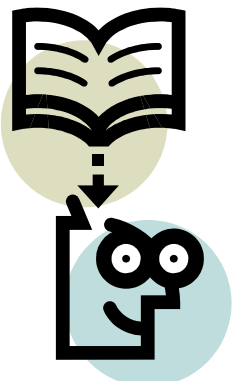
Atenție deoarece pagina creată este o pagină web trebuie să aibă extensia HTM, HTML sau ASP.

Vizualizarea paginii creată ca formular de introducere a datelor în Access poate fi făcută similar cu orice pagina web.

	<p><b>Intrebări:</b></p> <p>La ce folosesc paginile web create în Access? Care sunt pașii de realizarea a unei astfel de pagini?</p>
--	--

# TEMA 14

## REALIZAREA UNOR APLICAȚII ECONOMICE SIMPLIFICATE

	<p><b>Principalele obiective ale capitolului sunt:</b></p> <ul style="list-style-type: none"><li>Realizarea tabelelor</li><li>Realizarea formularelor</li><li>Realizarea rapoartelor</li><li>Realizarea interfeței</li></ul>
--	--

### Evidența facturilor întocmite către clienții unei societăți

Presupunem o firmă care întocmește facturi pentru vânzarea de produse unor clienți.

Produsele sunt caracterizate de nume și preț unitar (nu luăm în calcul situațiile mai complicate în care produsul este caracterizat de greutate, culoare etc.).



## REALIZAREA TABELELOR

Pentru fiecare client dorim să cunoaștem date de identificare cod\_cli (codul unic de înregistrare), numele, adresa, localitate, contul bancar și banca. Câmpul data\_s semnifică data scadenței.

clicli : Table		
	Field Name	Data Type
	cod_cli	Number
	nume	Text
	adr	Text
	cont_banca	Text
	nume_banca	Text
	tel	Number
	loc	Text

Tabela clicli

produse : Table		
	Field Name	Data Type
	cod_prod	Text
	den	Text
	pu	Number

Tabela produse

Pentru fiecare factură trebuie să putem înregistra mai multe produse precum și dacă factura respectivă este plătită sau nu.

Pentru înregistrarea mai multor produse pentru fiecare factură trebuie să avem o tabelă suplimentară (detalii\_fact) în care vom specifica numărul facturii și produsele corespunzătoare vândute.

În momentul încasării facturii se va trece în câmpul platit tipul documentului și numărul.

Astfel, dacă la câmpul platit nu este trecut nimic (are valoare null) înseamnă că factura respectivă nu este încă încasată.

facturi : Table		
	Field Name	Data Type
	nr_fact	Number
	data	Date/Time
	cod_cli	Number
	data_s	Date/Time
	platit	Text

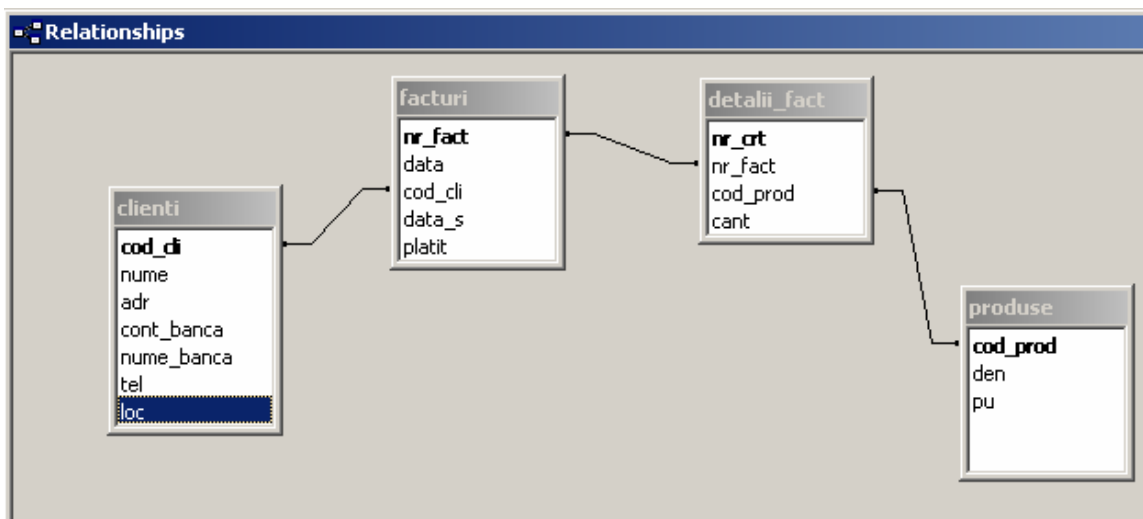
Tabela facturi

detalii_fact : Table		
	Field Name	Data Type
	nr_crt	Number
	nr_fact	Number
	cod_prod	Text
	cant	Number

Tabela detalii\_fact

Pentru relațiile dintre tabele avem:

- cheia primară (nr\_fact) din tabela facturi=cheia externă din tabela detalii\_fact (nr\_fact)
- cheia primară (cod\_cli) din tabela clienti=cheia externă din tabela facturi (cod\_cli)
- cheia primară (cod\_prod) din tabela produse=cheia externă din tabela detalii\_fact (cod\_prod)



Relațiile dintre tabele

## REALIZAREA INTEROGĂRILOR

1. Care sunt clienții care nu au plătit, numărul facturii și data\_s?  
 SELECT nume, nr\_fact, data\_s FROM clienti, facturi WHERE  
 clienti.cod\_cli=facturi.Cod\_cli and platit is null;

2. Care este valoarea totală a fiecărei facturii?  
 SELECT nr\_fact, sum(cant\*pu) AS Valoare, sum(cant\*pu)\*0.19 AS  
 TVA, sum(cant\*pu)\*1.19 AS Valoare\_cu\_tva FROM detalii\_fact,

```
produse WHERE produse.cod_prod=detalii_fact.cod_prod  
GROUP BY nr_fact;
```

3. Care sunt facturile neplătite?

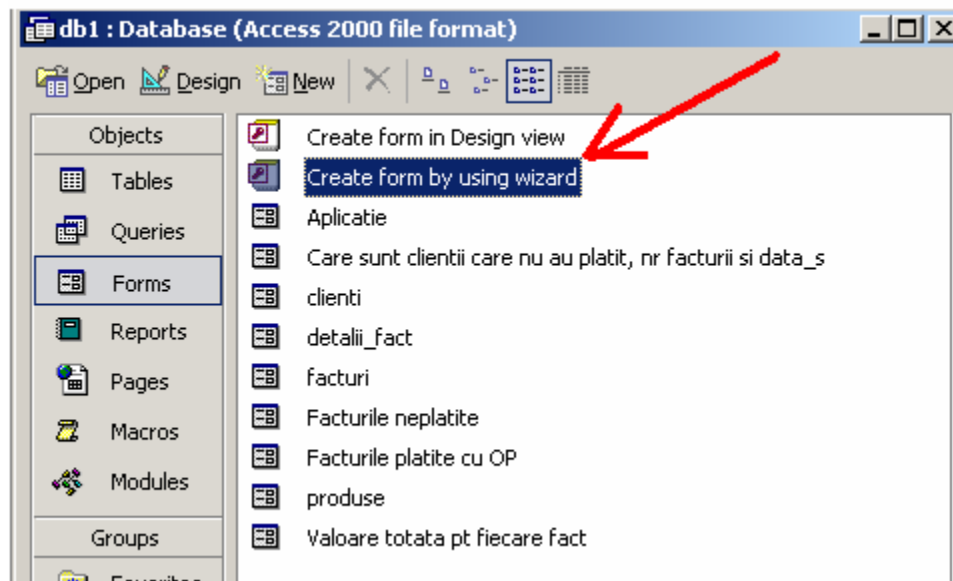
```
SELECT * FROM facturi WHERE platit is null;
```

4. Care sunt facturile plătite prin ordin de plată?

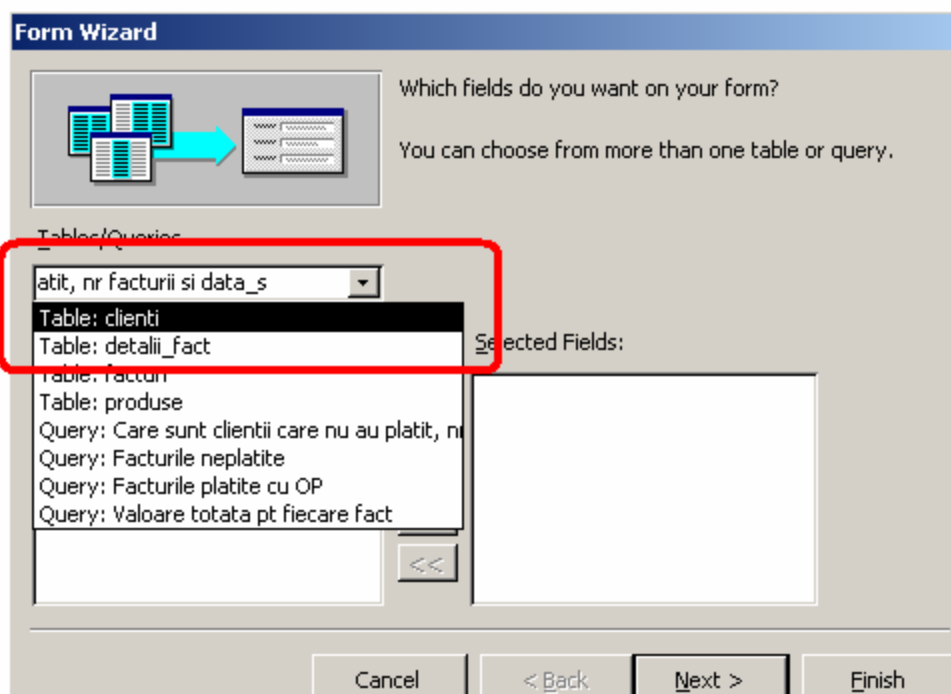
```
SELECT * FROM facturi WHERE platit like "op*";
```

## REALIZAREA FORMULARELOR DE INTRODUCERE A DATELOR

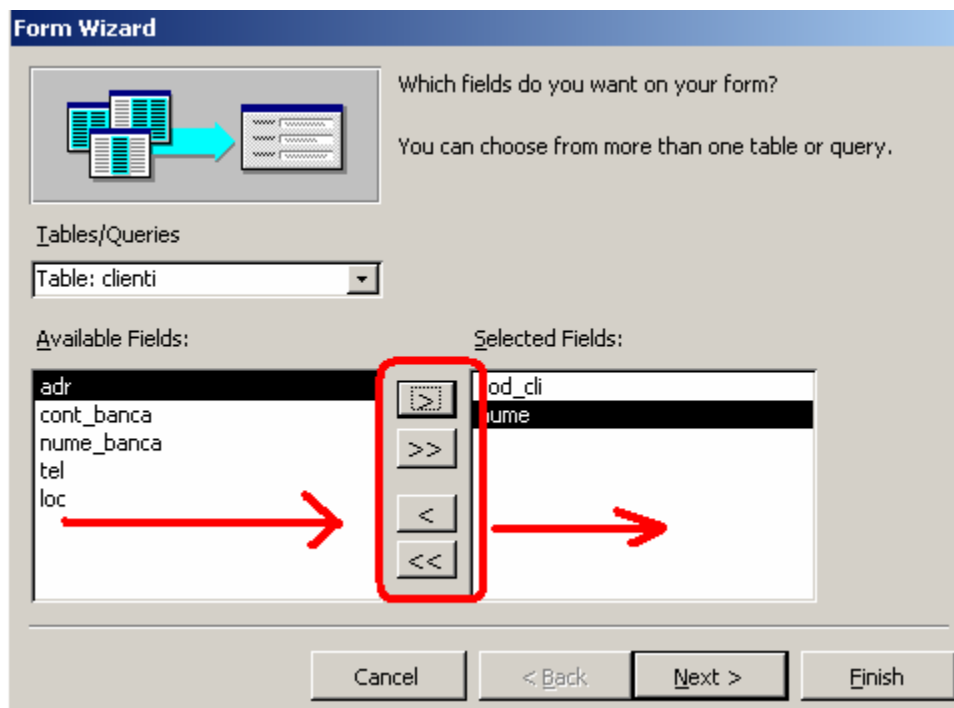
Pentru introducerea datelor vom realiza câte un formular pentru fiecare tabelă. Astfel, din panoul de comandă vom alege opțiunea FORMS și apoi CREATE FORM BY USING WIZARD.



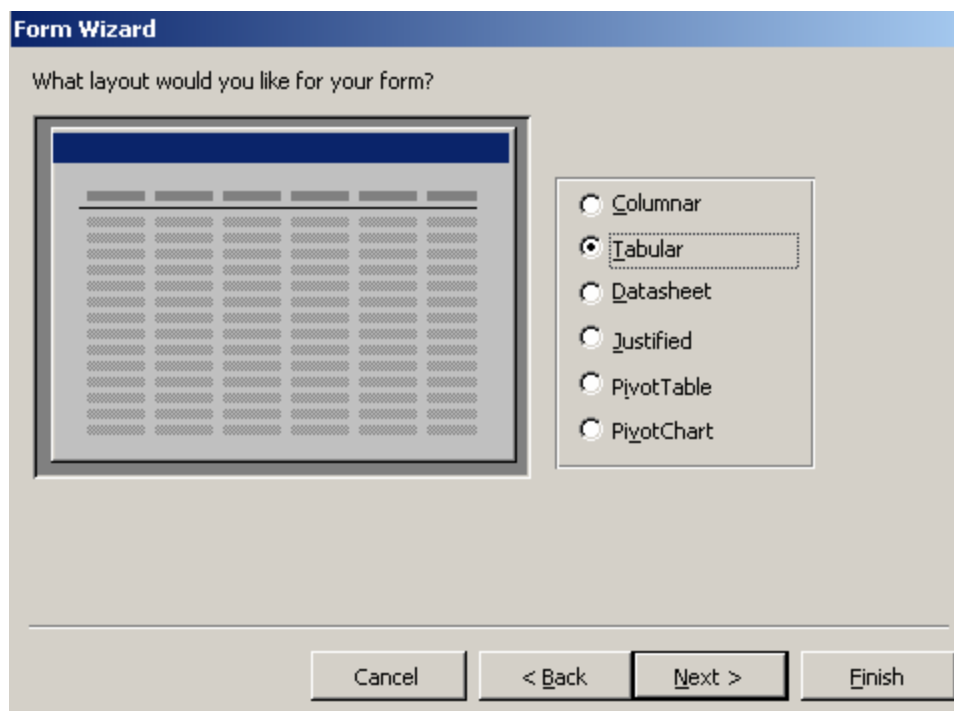
Realizarea formularelor utilizând generatorul de formulare



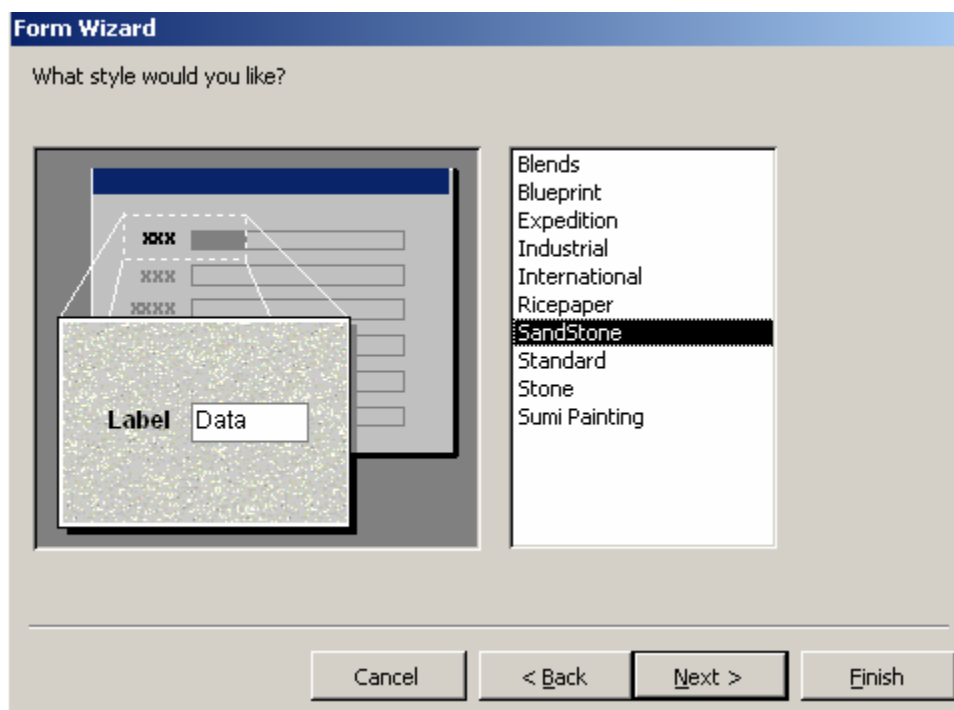
Sectarea tabelului pentru care se dorește crearea formularului



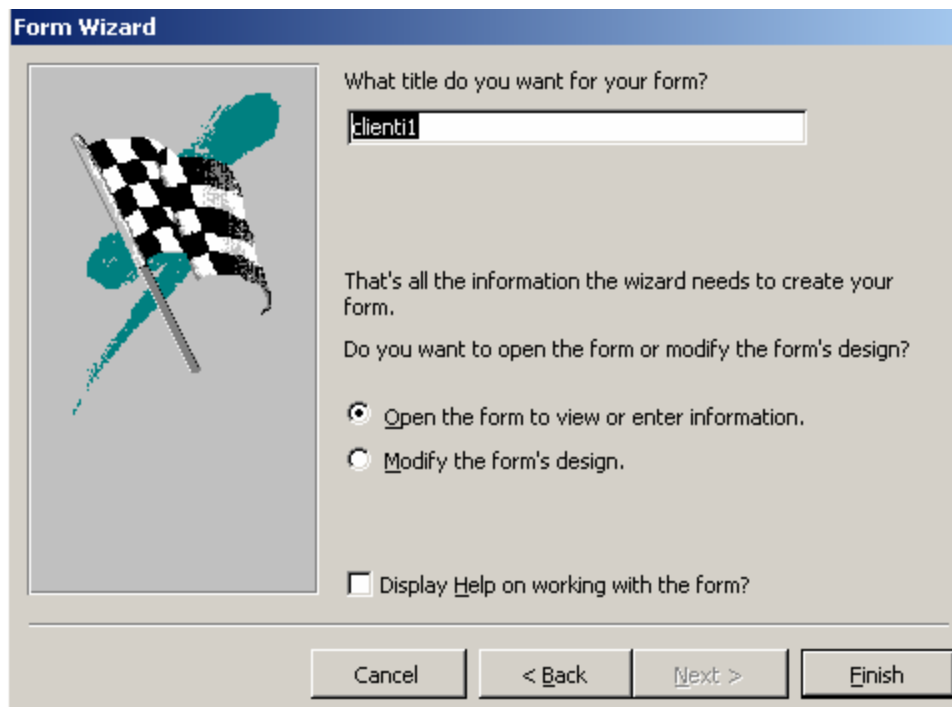
Selectarea câmpurilor din tabelă care vor apare în formular



Selectarea formei de prezentare a formularului

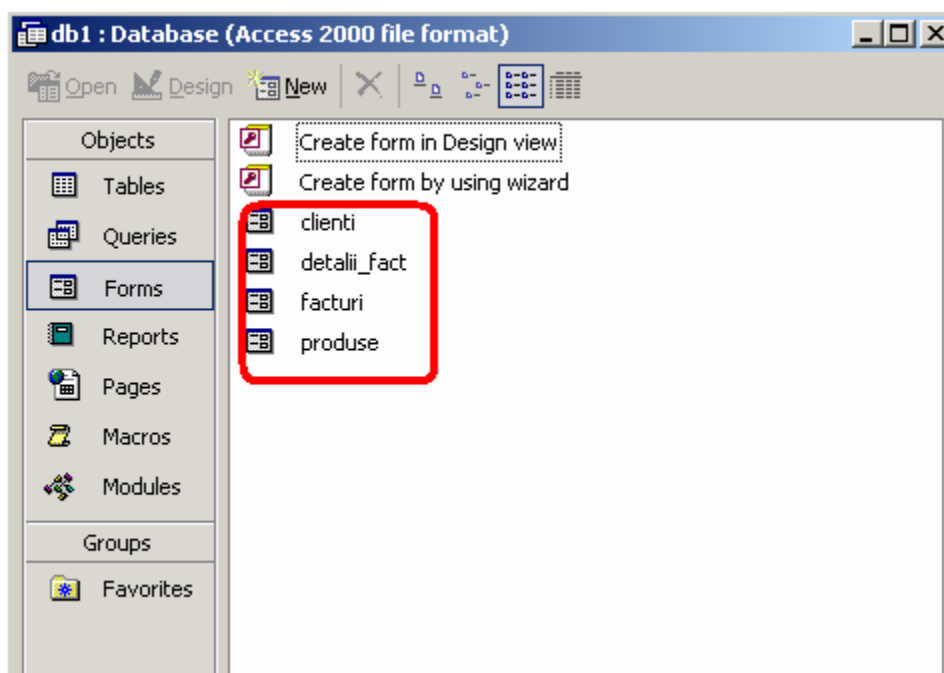


Selectarea stilului formularului



Denumirea formularului

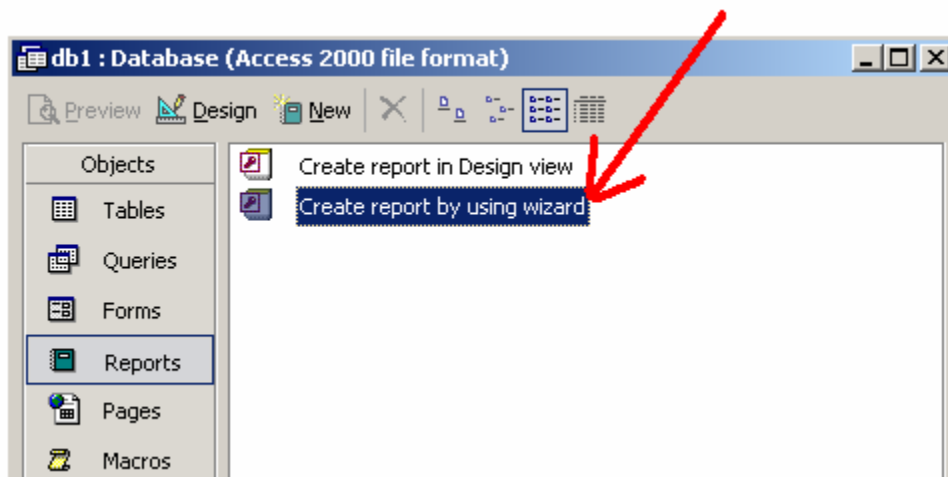
În mod similar se vor face formulare de introducere a datelor pentru toate tabelele și de afișare a datelor din interogările create.



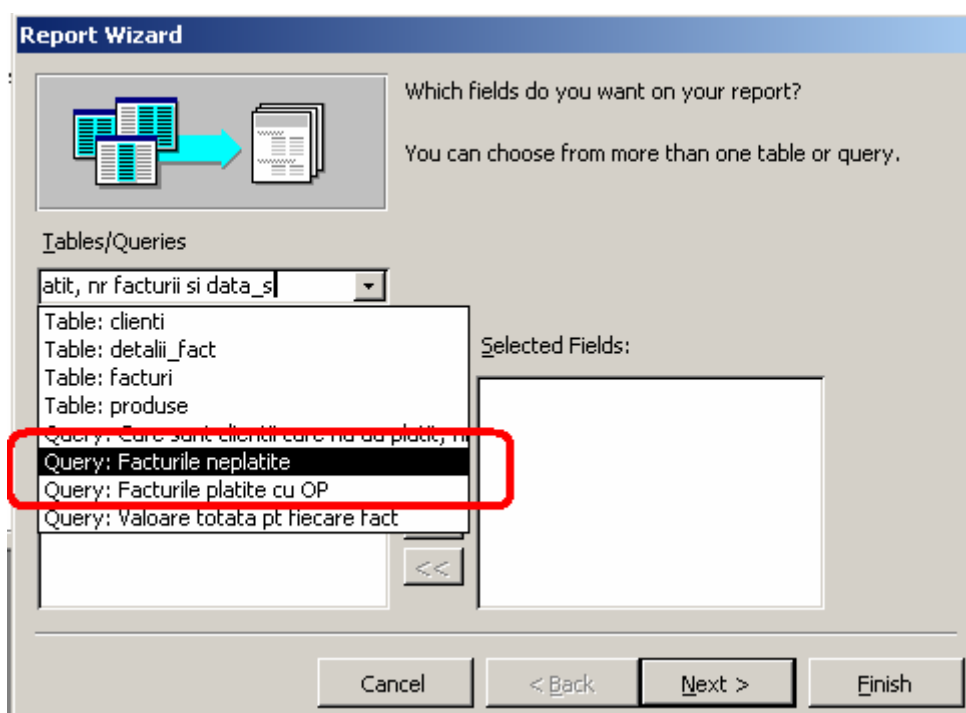
Formularele create pentru toate tabelele și pentru interogări

## REALIZAREA RAPOARTELOR

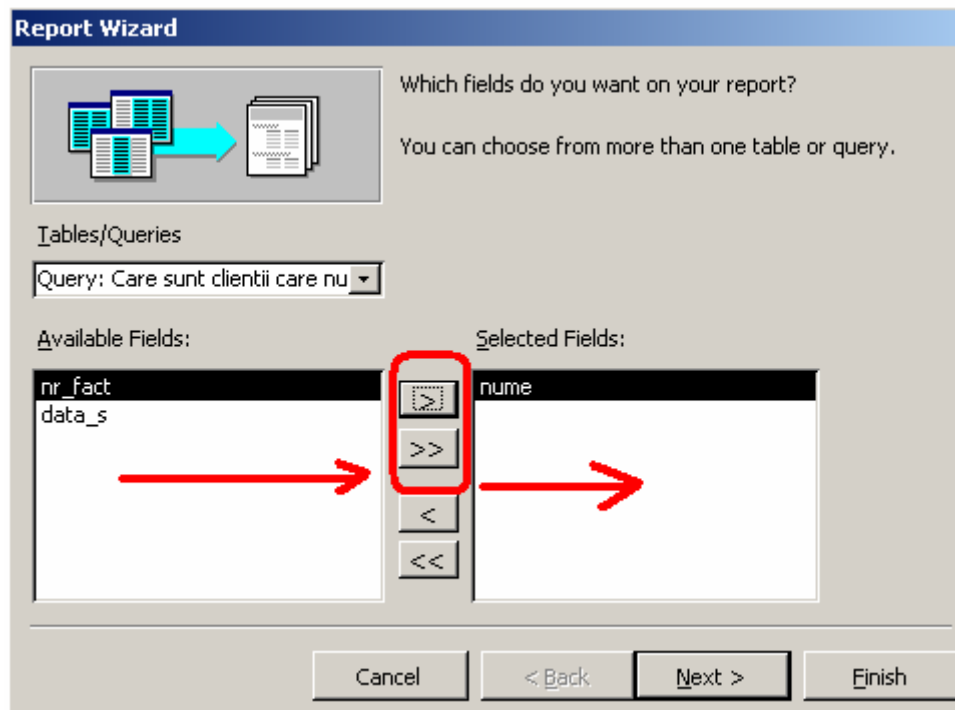
Pentru realizarea rapoartelor prin care se afișează datele din interogări vom alege opțiunea REPORTS - CREATE REPORT BY USING WIZARD.



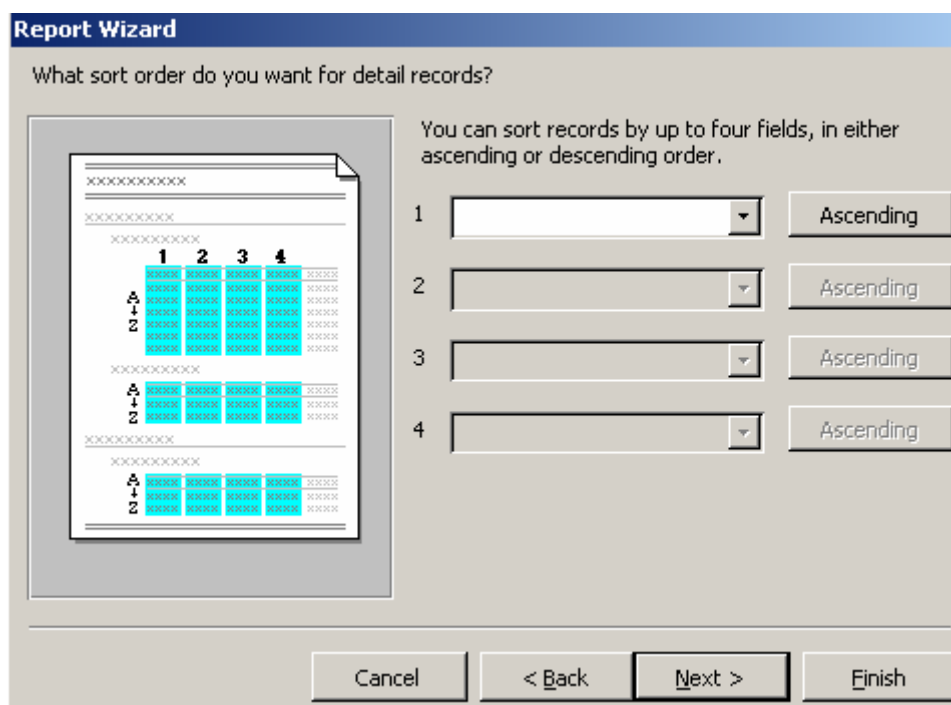
Alegerea opțiunii CREATE REPORT BY USING WIZARD



Alegerea interogării pentru care se face raportul



Selectarea câmpurilor din tabelă care vor fi afișate în raport



Selectarea unui câmp sau mai multe după care se face o ordonare a datelor



---

*clienti*

---

<i>nume</i>	<i>nr_fact</i>	<i>data_s</i>
<i>sc fl</i>		

---

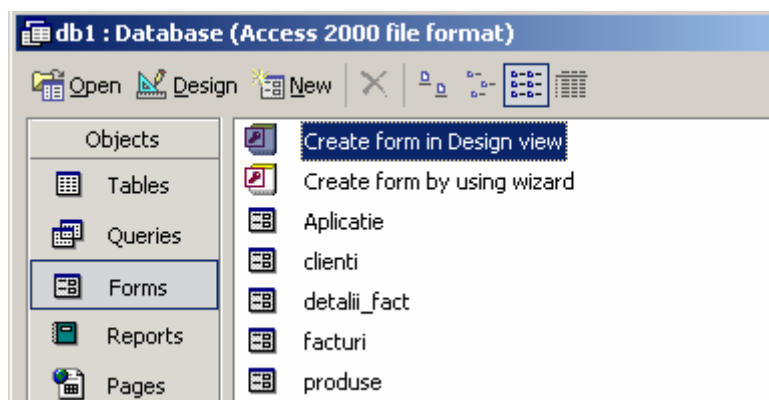
10202 29.03.2008

Afișarea raportului

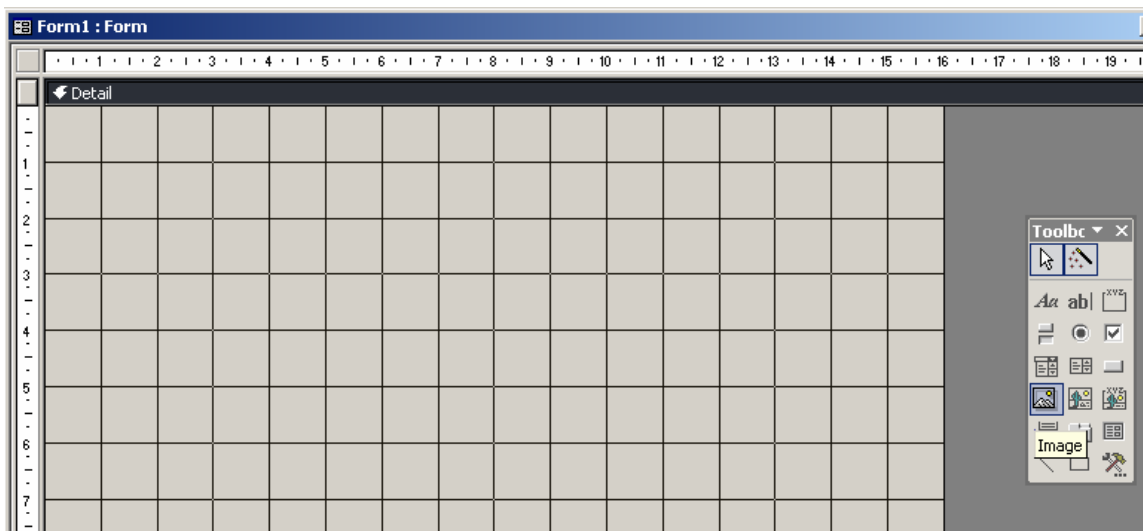
## REALIZAREA UNEI INTERFEȚE PENTRU APLICAȚIE

Pentru facilitatea lucrului vom face un formular, în care vom face butoane de pornire a formularelor și rapoartelor realizate.

Alegem opțiunea FORM-CREATE FORM IN DESIGN VIEW.

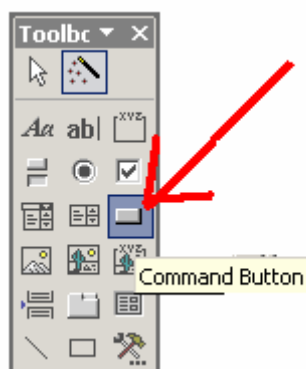


CREATE FORM IN DESIGN VIEW



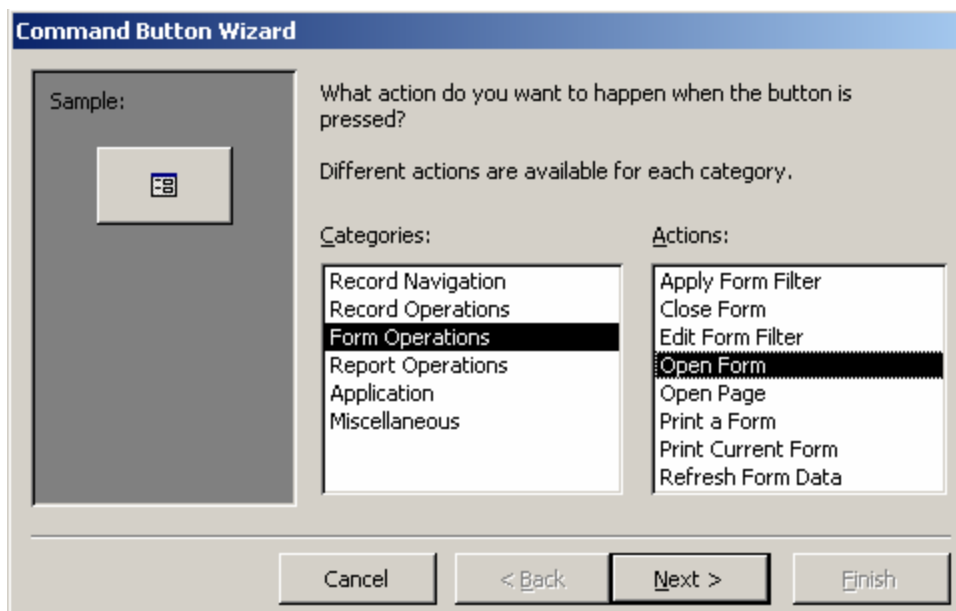
Crearea formularului "INTERFATA APLICATIE"

Pentru a realiza în formular butoanele alegem opțiunea COMMAND BUTTON din TOOLBOX.

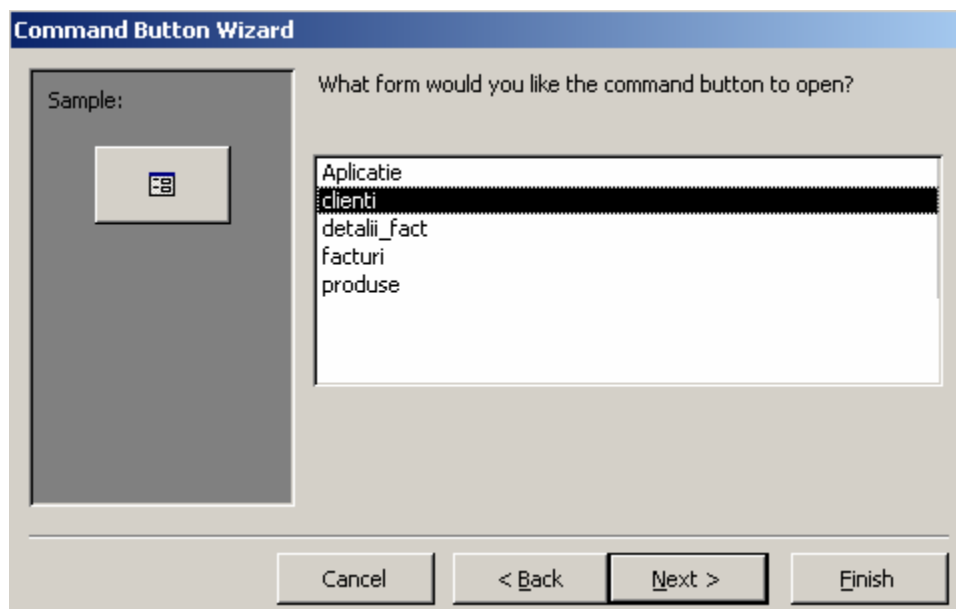


Alegerea opțiunii COMMAND BUTTON

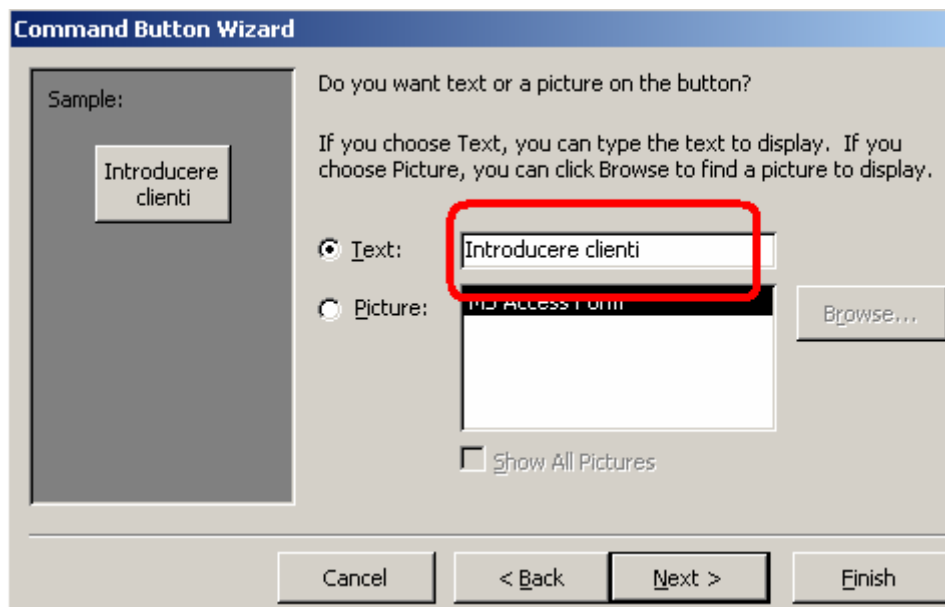
Pentru realizarea butoanelor care vor lansa formularele create vom alege opțiunile FORM OPERATIONS - OPEN FORM.



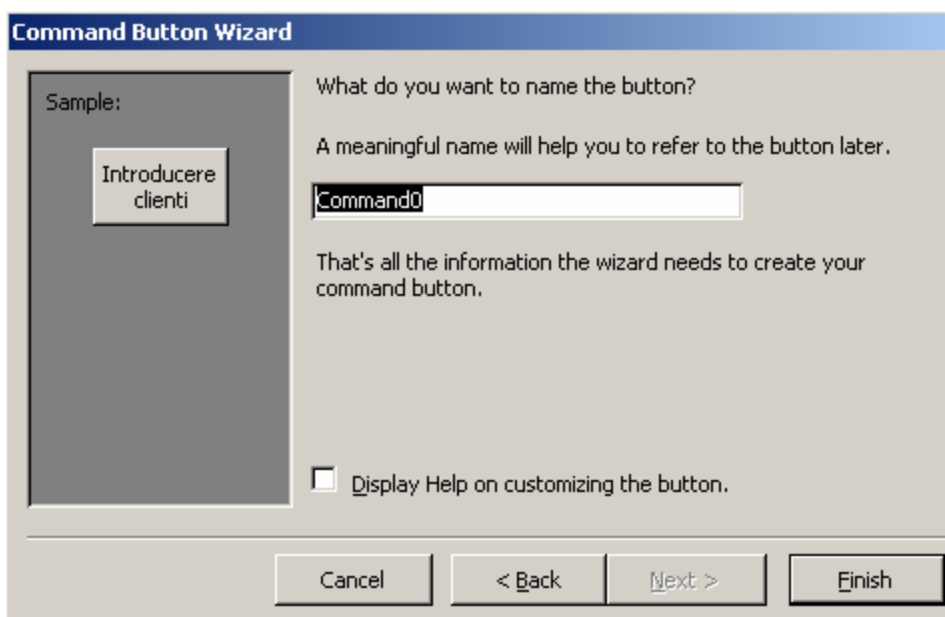
Alegerea opțiunii FORM OPERATIONS - OPEN FORM



Alegerea formularului care se va deschide

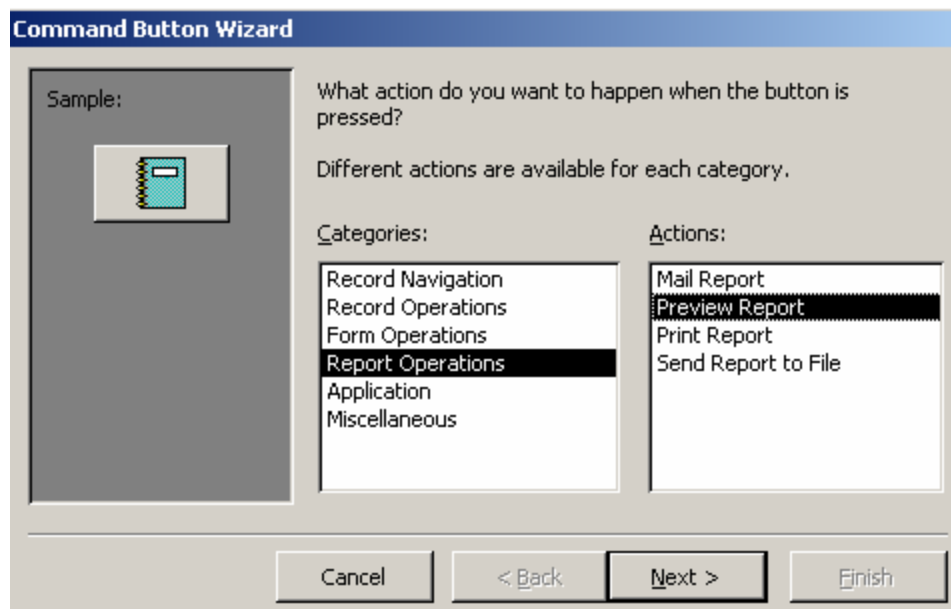


Introducerea textului care va fi afișat pe buton

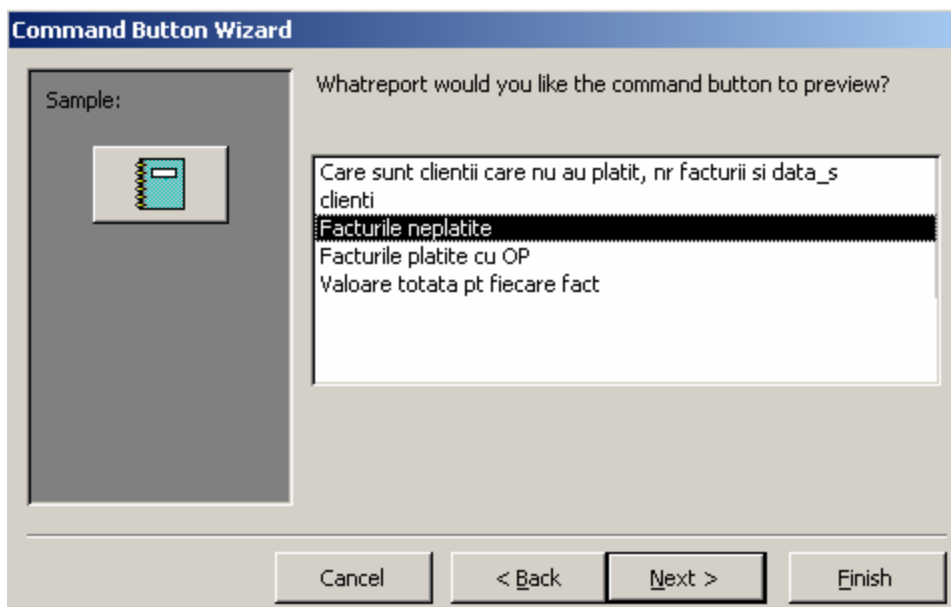


Numele butonului (nu e obligatoriu de dat)

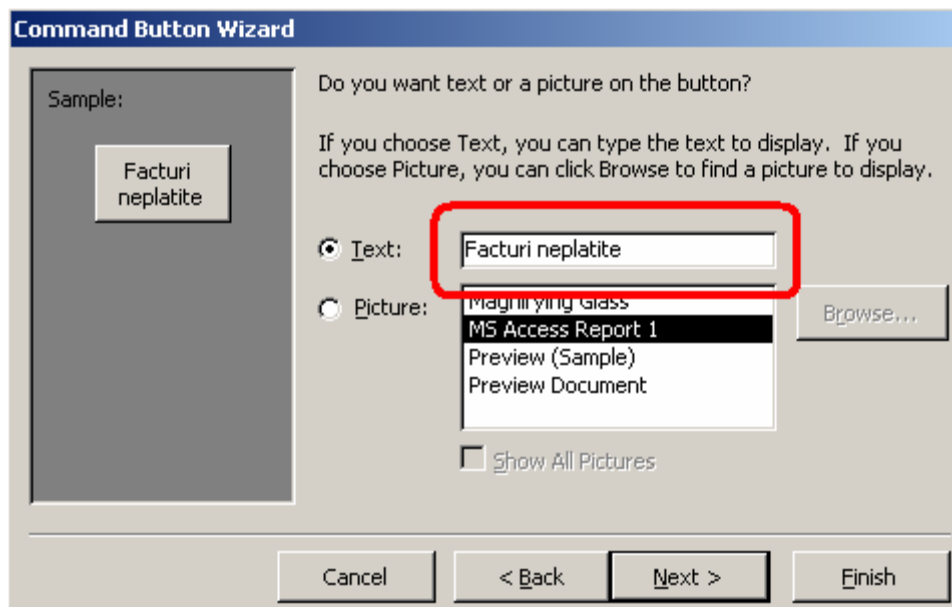
Pentru lansarea în execuție a rapoartelor alegem în momentul realizării butonului opțiunea REPORT OPERATIONS - PREVIEW REPORT.



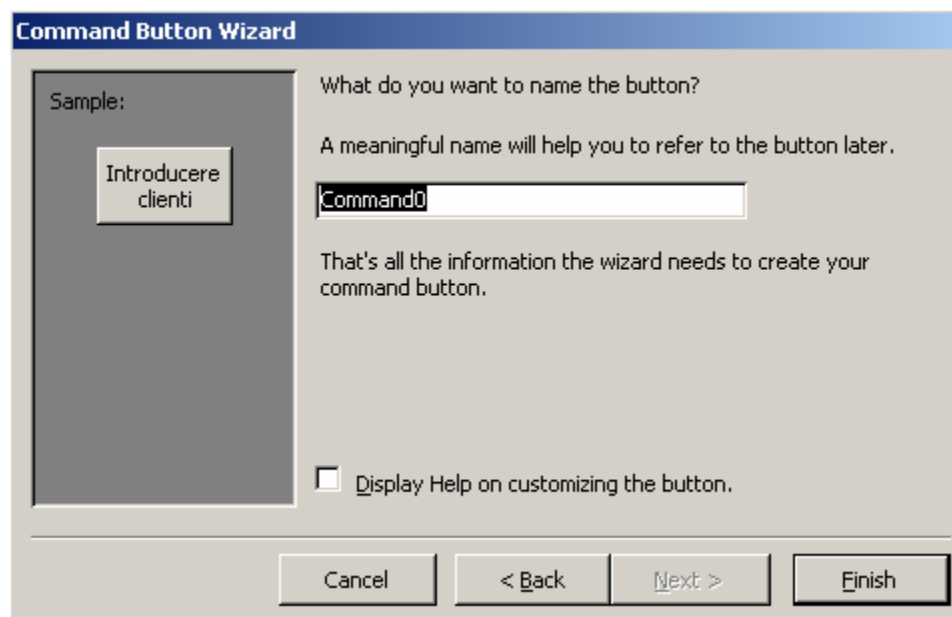
### REPORT OPERATIONS - PREVIEW REPORT



Selectarea raportului pentru care dorim să facem butonul



Introducerea textului afișat pe buton



Numele butonului (nu e obligatoriu de dat)

La sfârșit, interfața va arăta astfel:

Interfața aplicației modul design

Interfața aplicației în momentul rulării

**Temă:**

Realizați o aplicație pentru ținerea evidenței mijloacelor fixe dintr-o întreprindere. Pentru fiecare mijloc fix calculați care este amortizarea lunară prin metoda liniară pentru mijloacele fixe neamortizate. Calculați care este valoarea actuală a fiecărui mijloc fix. Pentru realizarea aplicației utilizați următoarele tabele:

	<p><b>MFIX(nr_inv, nume, cod_clasa, val_ini, data_achizitiei, cod_furn)</b> unde: nr_inv - număr inventar cod_clasa - codul clasei de amortizare din care face parte val_ini - valoare inițială cod_furn - codul furnizorului</p> <p><b>FURNIZORI (cod_f, den, loc, adr, tel, pers_contact)</b> unde: cod_f - cod furnizor den - denumirea furnizorului loc - localitate adr - adresa tel - telefon pers_contact - persoana de contract</p> <p><b>DURATE(cod_clasa, denumire, nr_ani)</b> unde: cod_clasa - codul clasei de amortizare din care face parte nr_ani - numărul de ani în care se amortizează mijlocul fix</p>
--	--



**Anexa 1****Tabele utilizate****Tabela: angajati**

<b>câmp</b>	<b>tip</b>
cod (CP)	numeric
nume	text
prenume	text
data_nasterii	data
sal_brut	numeric
cod_dep	text

**Tabela: departamente**

<b>câmp</b>	<b>tip</b>
cod_dep (CP)	text
den	text
localitate	text
sef	text
telefon	numeric

**Tabela: facturi**

<b>câmp</b>	<b>tip</b>
nr_factura (CP)	numeric
cod_furn	text
data_fact	data
valoare	numeric

**Tabela: furnizori**

câmp	tip
cod_furn (CP)	text
nume_furn	text
adresa	text
localitate	text

**Tabela: mf**

câmp	tip
cod_mf (CP)	numeric
den	text
valoare	numeric
per_amort	numeric
cod_furn	text

## BIBLIOGRAFIE

1.	[DAV]	Davidescu, N., Hotăran, S., Lupu, L., Ivancenco, V., Davidescu, R., Ilie., L., <i>Produse program generalizabile în domeniul financiar-contabil</i> , Editura Didactică și Pedagogică, București
2.	[URS02]	Ursăcescu, M., <i>Sisteme Informatice</i> , Editura Economică, București, 2002
3.	[OME02]	Oprea, D., Meșniță G., <i>Sisteme informaționale pentru manageri</i> , Polirom, București. 2002
4.	[CAT83]	Cătuneanu, V., Mihalache, A., <i>Bazele teoretice ale fiabilității</i> , Editura Academiei Republicii Socialiste România, București, 1983.
5.	[MIH95]	Mihalache, A., <i>Când calculatoarele greșesc...</i> , Editura didactică și pedagogică, București, 1995.
6.	[CAT89]	Cătuneanu, V., Popențiu F., <i>Optimizarea fiabilității sistemelor</i> , Editura Academiei Republicii Socialiste România, București, 1989.
7.	[OPR99]	OPREA D., <i>Analiza și proiectarea sistemelor informaționale economice</i> , Ed. Polirom, 1999
8.	[LUN03]	LUNGU I., ș.a., <i>Sisteme informatice. Analiză, proiectare și implementare</i> , Ed. Economică, 2003.
9.	[STA02]	STANCIU V., ș.a., <i>Proiectarea sistemelor informatice</i> , Ed. Dual Tech, 2002.
10.	[STA02b]	STANCIU V., ș.a., <i>Proiectarea sistemelor informatice</i> , Ed. Dual Tech, 2002.
11.	[HER03]	HERNANDEZ M., <i>Proiectarea bazelor de date</i> , Ed. Teora, 2003.
12.	[ZAH01]	ZAHARIE D., ș.a., <i>Sisteme informatice pentru asistarea deciziei</i> , Ed. Dual Tech, 2001.
13.	[VIR04]	VÎRLAN G., <i>Tehnologii Client / Server în dezvoltarea sistemelor informatice în economie</i> , Ed. Didactică și Pedagogică, 2004.
14.	[CON01]	CONNOLLY Th., ș.a., <i>Baze de date. Proiectare, Implementare, Gestionare</i> , Ed. Teora, 2001.
15.	[MUR06]	MUREȘAN M., <i>Sisteme informatice</i> , Ed. Universul Juridic, 2006.

