

## 4. Interogări

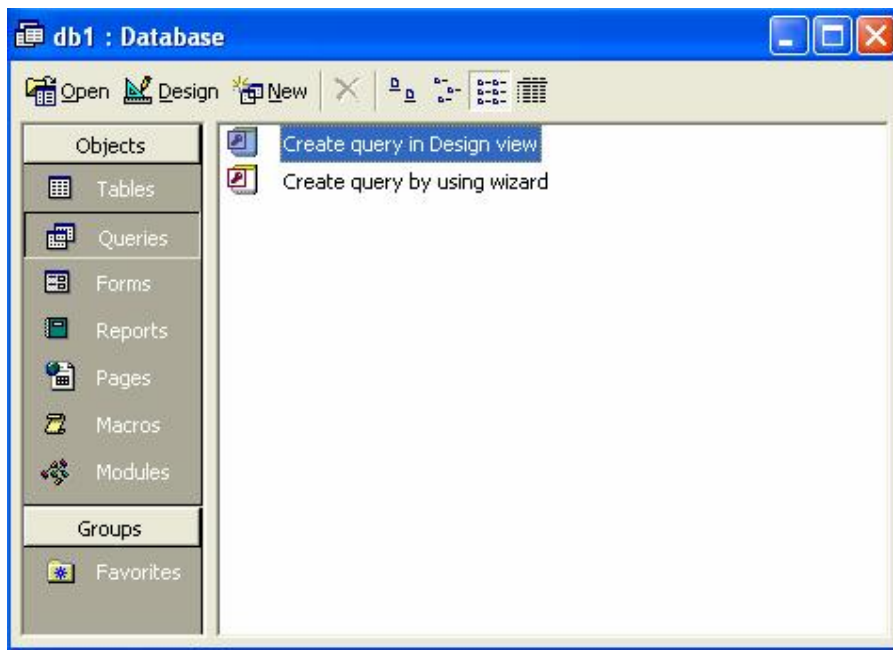
### 4.1 Probleme propuse

1. Folosind baza de date de la capitolul 1.1, sa se realizeze o interogare care să afișeze toate notele studentului al cărui nume este "Tasica".

#### Rezolvare:

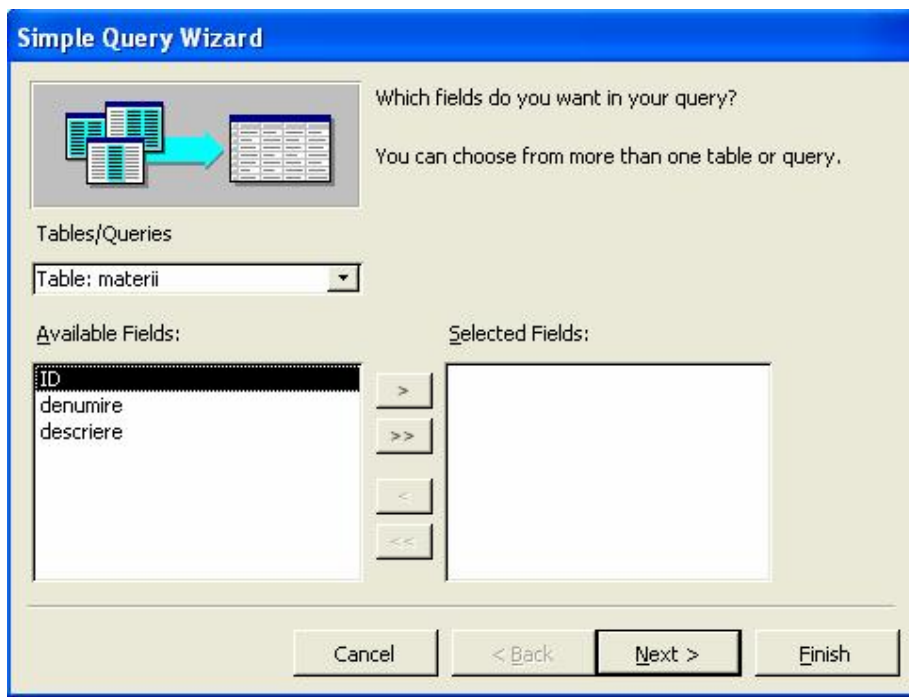
Această problemă se poate rezolva ușor folosind o facilitate a produsului Microsoft Access numită *query* sau *interogare*. În Access, ca in orice sistem de gestiune a bazelor de date, interogările se realizează cu ajutorul unui limbajului SQL. Fiind un produs pentru nespecialiști, Microsoft Access pune la dispoziția utilizatorului o interfață care ascunde detaliile tehnice legate de limbajul SQL utilizatorul putând realiza interogări fără ca sa cunoască acest limbaj.

Pentru a realiza un query este nevoie sa se intre in zona de lucru a query-urilor efectuând click pe butonul *Queries* aflat in panoul dreapta a ferestrei *Database*.

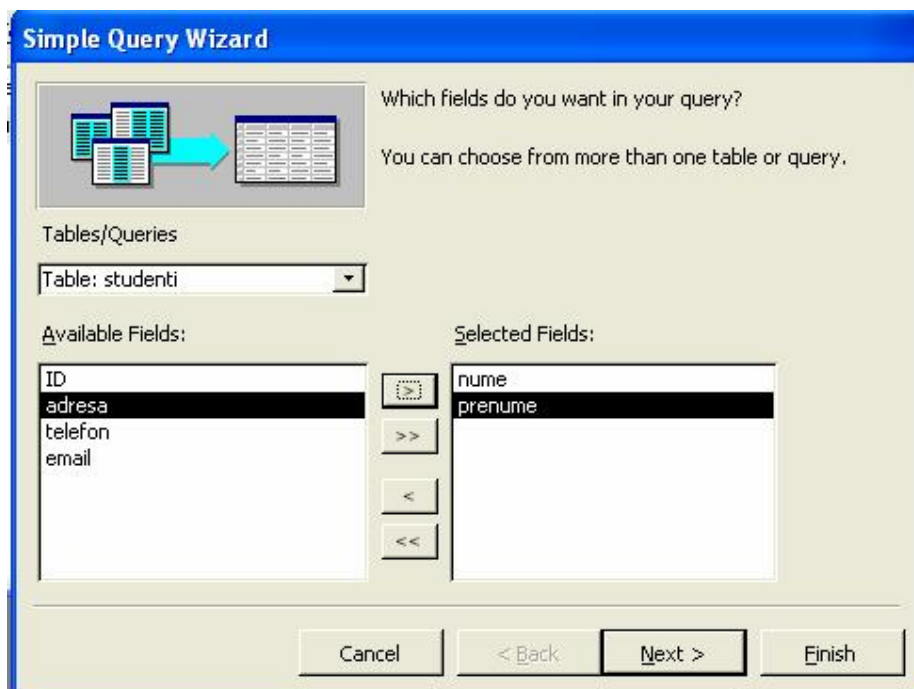


Pentru a crea un query, recomandăm folosirea opțiunii *Create query by using wizard* care este cel mai ușor de folosit.

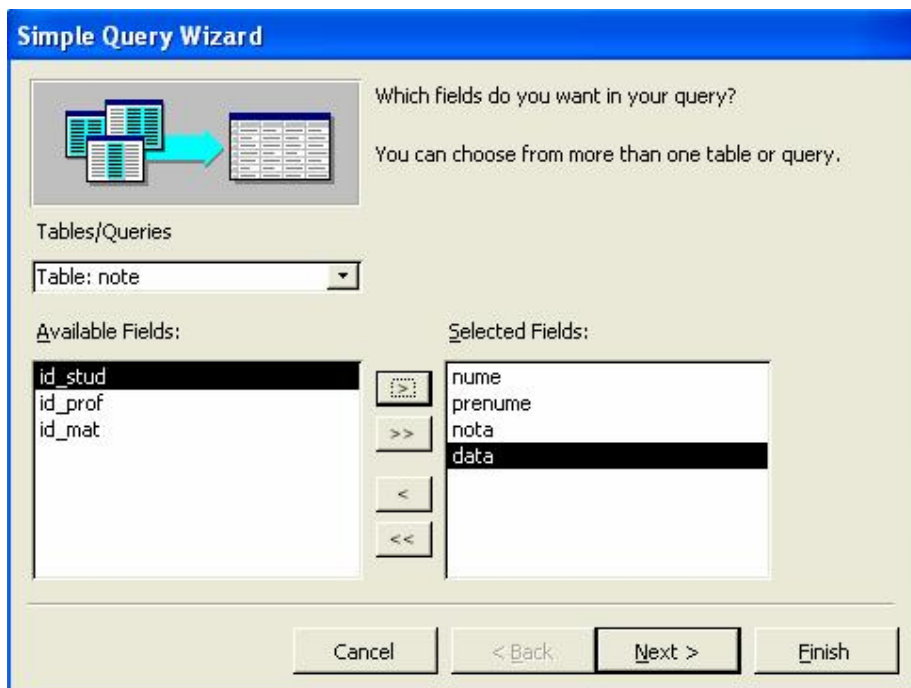
Efectuând dublu click pe aceasta opțiune apare fereastra de mai jos:



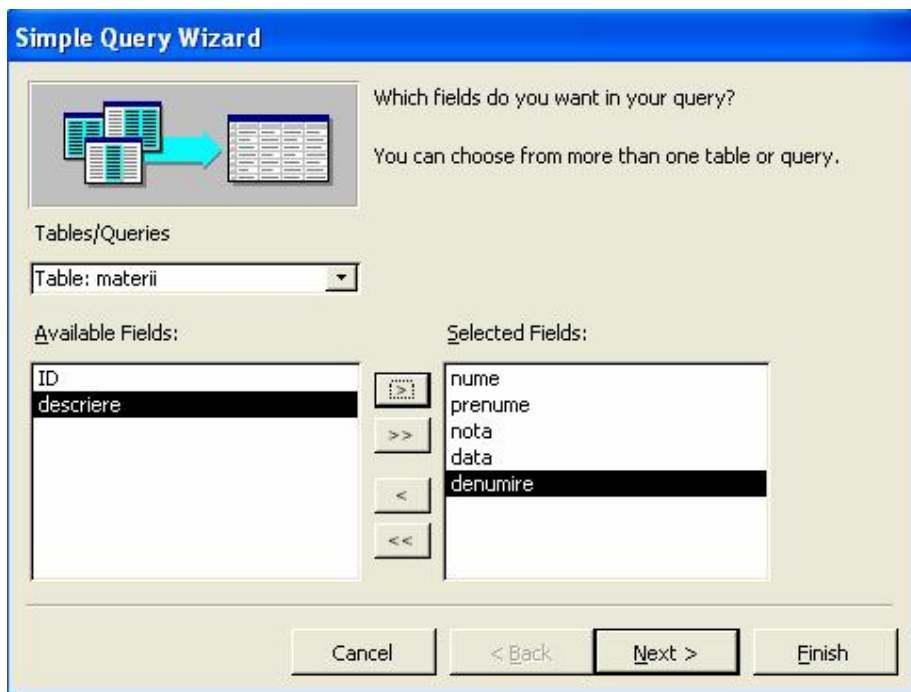
Se selectează tabela dorită, în cazul nostru „studenți”, din care se includ în query câmpurile nume și prenume. Această operație se realizează efectuând click pe butonul cu semnul “>” pentru fiecare câmp în parte.



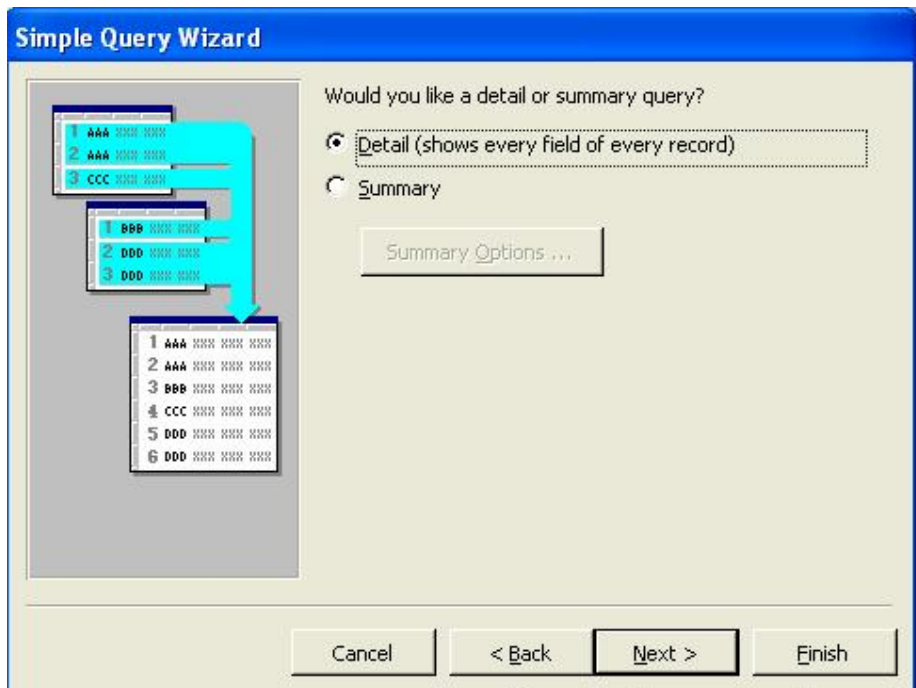
După aceea se selectează tabela „note”, din care se includ în query câmpurile nota și data.



Apoi alegem denumirea din tabela „materii”.



In acest moment se apasă butonul *Next*.



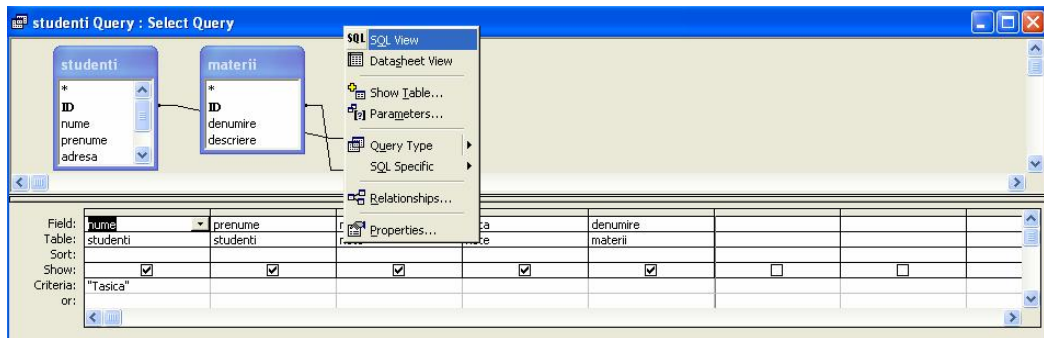
Fereastra de mai sus întrebă utilizatorul dacă dorește o interogare detaliată sau una de tip rezumat. Recomandăm să folosim opțiunile implicite și să se apăsăm butonul *Next* ca să ajungem la pasul *Finish*.



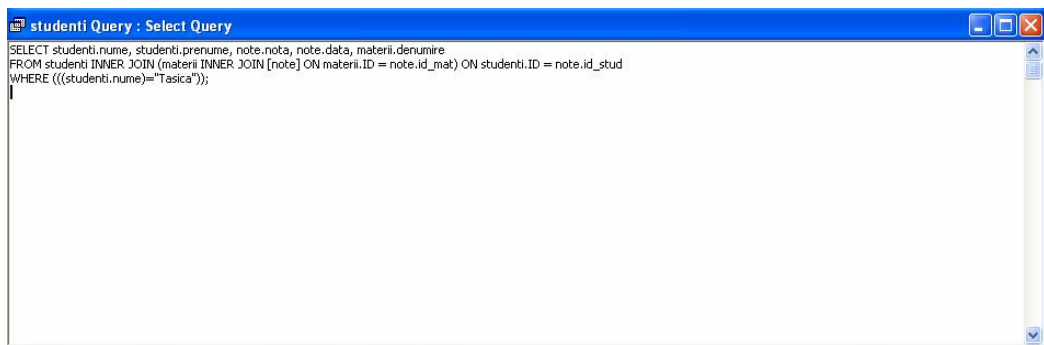
Apăsând *Finish*, pe ecran o să apară rezultatele execuției cererii. În cazul nostru, vor fi afișate notele pentru toți studenții.



Putem sa vizualizăm codul SQL generat in mod automat de interfața grafica efectuând click dreapta in partea de sus a ferestrei si alegând opțiunea “SQL View”.



Pe ecran va apărea codul SQL corespunzător.



Adică:

```
SELECT studenți.nume, studenti.prenume, note.nota, note.data, materii.denumire
FROM studenti INNER JOIN (materii INNER JOIN [note] ON materii.ID = note.id_mat) ON
studenti.ID = note.id_stud
WHERE (((studenti.nume)="Tasica"));
```

Nu vom insista asupra semnificației termenilor de mai sus deoarece depășesc cadrul acestei lucrări. Trecând peste majoritatea elementelor, se observa ca exista o clauza “WHERE” care pune condiția de selectare a înregistrărilor din tabele. Schimbând condiția de aici se modifica și înregistrările selectate de query. Spre exemplu da în loc de “Tasica” scriem “Ion” și pe ecran va apare:



Se pot face chiar și modificări mai complexe, cum ar fi să cerem numele de la tastatura. Dacă scriem în loc de “Tasica” un cuvânt oarecare, fără să fie încadrat de ghilimele și fără să conțină spații, atunci sistemul va solicita de la tastatura o valoare. Sub formă de parametru Spre exemplu putem modifica interogarea astfel:

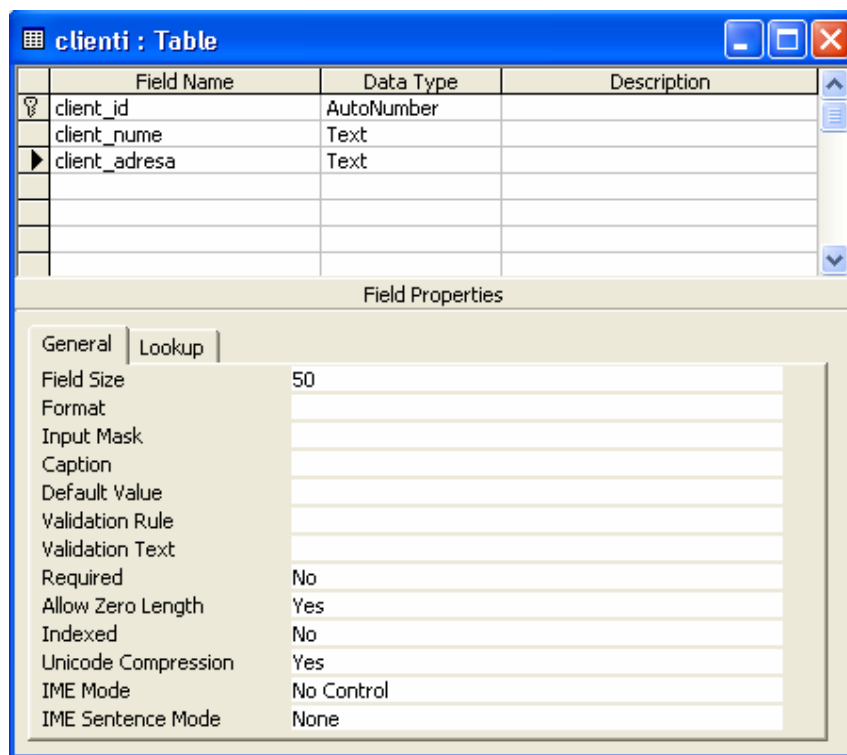
```
SELECT [studenti].[nume], [studenti].[prenume], [note].[nota], [note].[data],  
[materii].[denumire]  
FROM studenti INNER JOIN (materii INNER JOIN [note] ON  
[materii].[ID]=[note].[id_mat]) ON [studenti].[ID]=[note].[id_stud]  
WHERE ((([studenti].[nume])=Dati_numele));
```

În acest caz ni se va solicita să scriem numele, de la tastatură, după care interogarea se va executa pentru valoarea respectivă.

**2. Se dă o firmă care vinde diferite produse. Produsele sunt livrate clienților conform contractelor. Orice contract poate să cuprindă mai multe produse, dar un contract poate fi încheiat cu un singur client.**

După definirea datelor și normalizarea lor până în FN3 (forma normală 3), se poate construi o bază de date cu patru tabele:

- Tabelul *clienti*, cu informații despre clienții firmei, cu care sau negociat diferite contracte:



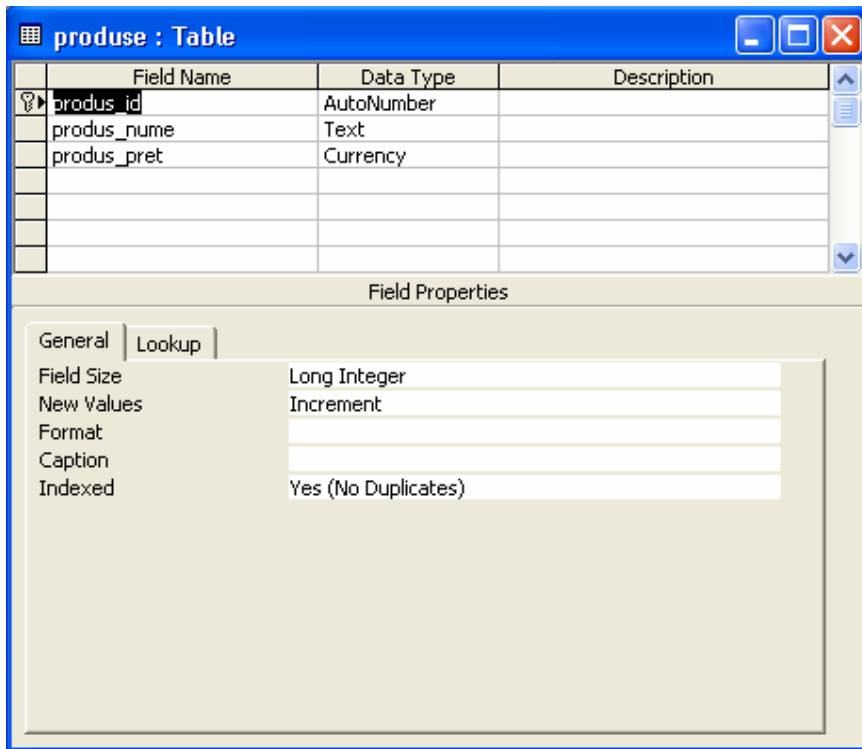
The screenshot displays the Microsoft Access interface for a table named 'clienti'. The top part shows the table structure with columns for Field Name, Data Type, and Description. The bottom part shows the 'Field Properties' window for the 'client\_adresa' field.

Field Name	Data Type	Description
client_id	AutoNumber	
client_nume	Text	
client_adresa	Text	

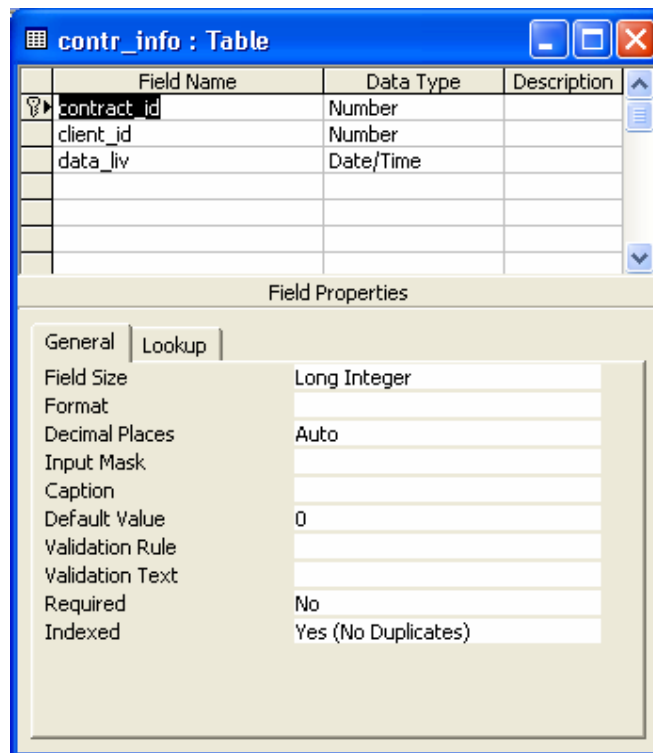
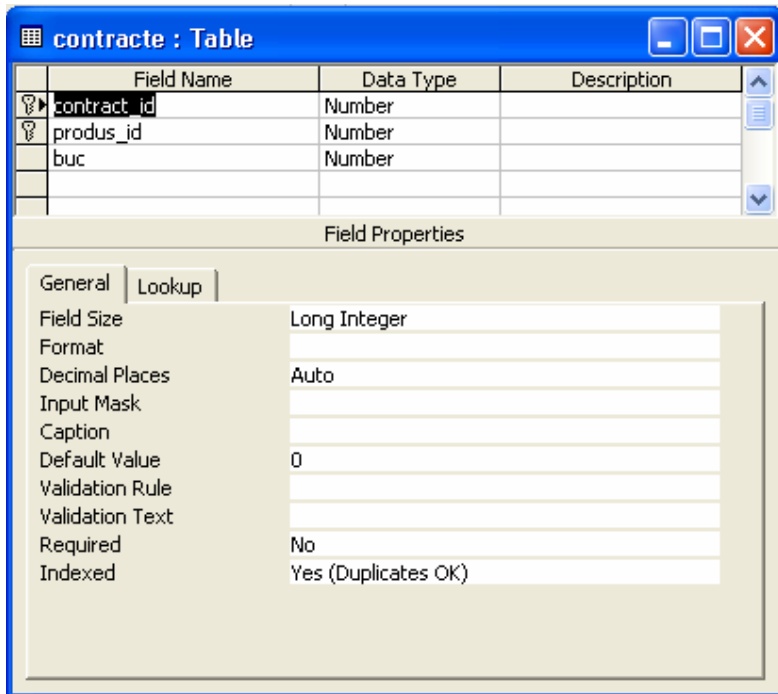
Field Properties	
Property	Value
Field Size	50
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None

- Tabelul *produse*, care cuprinde date referitoare la produsele comercializate de firmă:



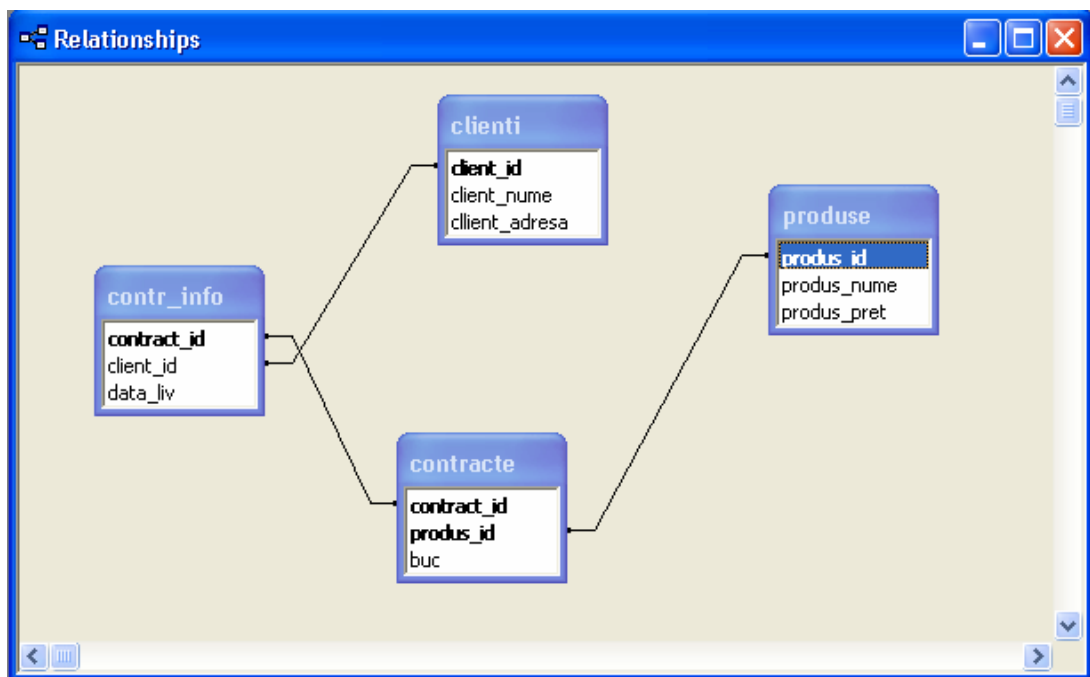
- Datele referitoare la contracte trebuie împărțite în două tabele, ca cerință a normalizării (necesară pentru a aduce datele în forma normală doi) : există o dependență parțială între *contract\_id* și *client\_id*, astfel ultimul nu poate face parte din tabelul contracte. (la fel și cu data livrării – *data\_liv*). Aceste două câmpuri sunt dependente doar de *contract\_id*, nu și de cheia compusă din tabelul *contracte*. Se crează astfel un tabel adițional: *contract\_info*:





Următorul pas este crearea legăturilor dintre tabele:

- Cu un client se pot încheia mai multe contracte, astfel relația dintre tabelele *clienți* și *contr\_info* este de unu - la mai - mulți.
- Un contract poate cuprinde mai multe produse, pentru fiecare combinație contract – produs se crează o linie nouă în tabelul *contracte*. Fiecare produs figurează însă o singură dată în tabelul *produse*. Astfel relația dintre tabelele *produse* și *contracte* este de unu - la mai - mulți.
- La fel este și cea dintre *contr\_info* și *contracte*. Un contract este prezent o singură dată în tabelul *contr\_info*, și de mai multe ori în tabelul *contracte*, datorită faptului, că un contract poate conține mai multe produse de livrat.



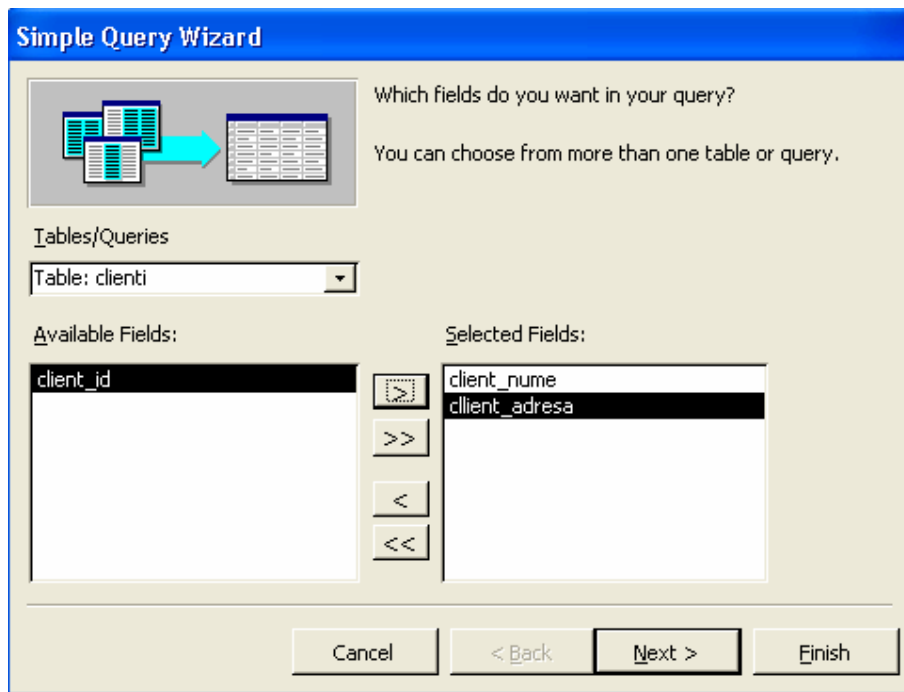
*Access* oferă diferite posibilități de realizare a interogărilor (queries)

- Crearea pas cu pas prin Design View
- Crearea cu vrăjitorul (Wizard)
- Introducerea directă a comenzilor SQL în SQL View

Să realizăm în continuare o suită de interogări, de exemplu:

1. Să se afișeze toți clienții firmei

- se lansează vrăjitorul
- din lista cu tabele se selectează tabelul clienți
- din lista *available fields* se aleg câmpurile *client\_nume* și *client\_adresa*
- se apasă pe butonul *finish*
- din lista interogărilor existente se alege interogarea creată și se deschide prin dublu clic

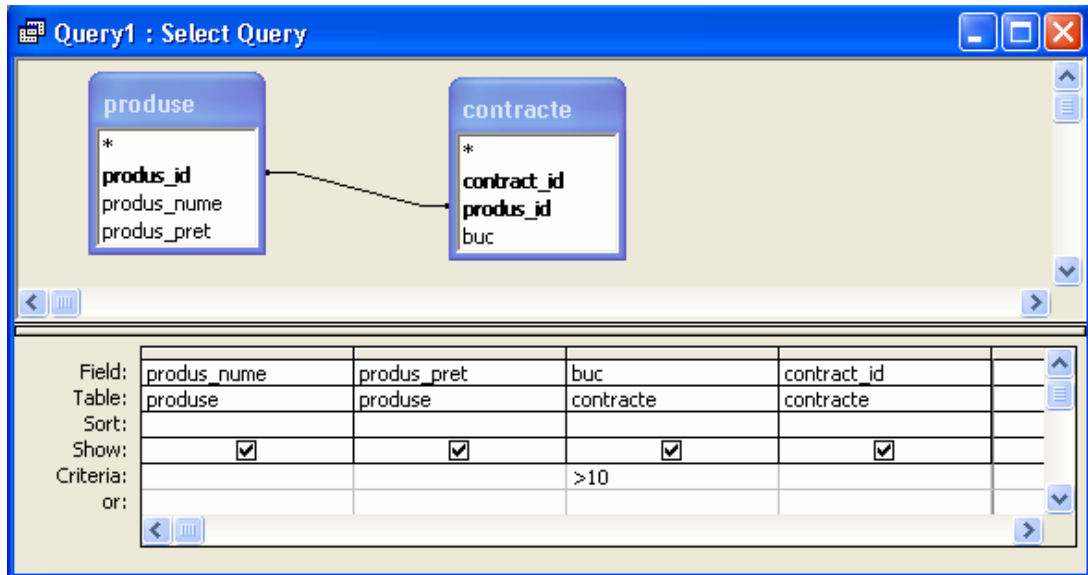


2. Să se determine acele produse, care apar cu mai mult de 10 bucăți în vreun contract. Să se afișeze și identificatorul contractului respectiv.

- se lansează utilitarul de construcție a interogărilor (*design view*)
- se adaugă tabelele *produse* și *contracte*
- se specifică coloanele dorite din cele două tabele
- se fixează condiția, ca pentru fiecare produs din contract numărul de bucăți livrate să fie mai mare decât zece (  $buc > 10$  )

- comanda SQL aferantă interogării este (afișabilă în SQL View):

```
SELECT produse.produc_nume, produse.produc_pret, contracte.buc, contracte.contract_id
FROM produse INNER JOIN contracte ON produse.produc_id = contracte.produc_id
WHERE (((contracte.buc)>10));
```



- dacă se dorește doar numele produselor, care apar în vreun contract cu caracteristica  $buc > 10$ , fără a cere și contractul în care figurează, se poate afișa doar numele produselor, o singură dată. În cazul precedent un produs poate să apară de mai multe ori, în funcție de numărul de contracte în care apare

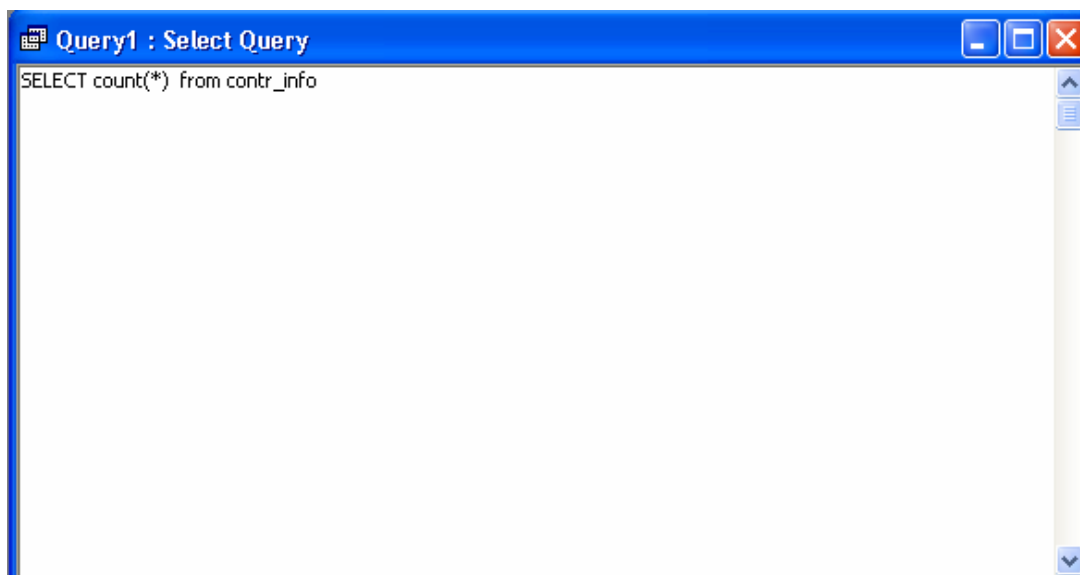
```
SELECT distinct(produse.produc_nume) FROM produse INNER JOIN contracte ON
produse.produc_id = contracte.produc_id WHERE (((contracte.buc)>10));
```

- se execută interogarea prin alegerea opțiunii *Run* din meniul *Query*

### 3. Să se determine numărul total al contractelor încheiate de firmă

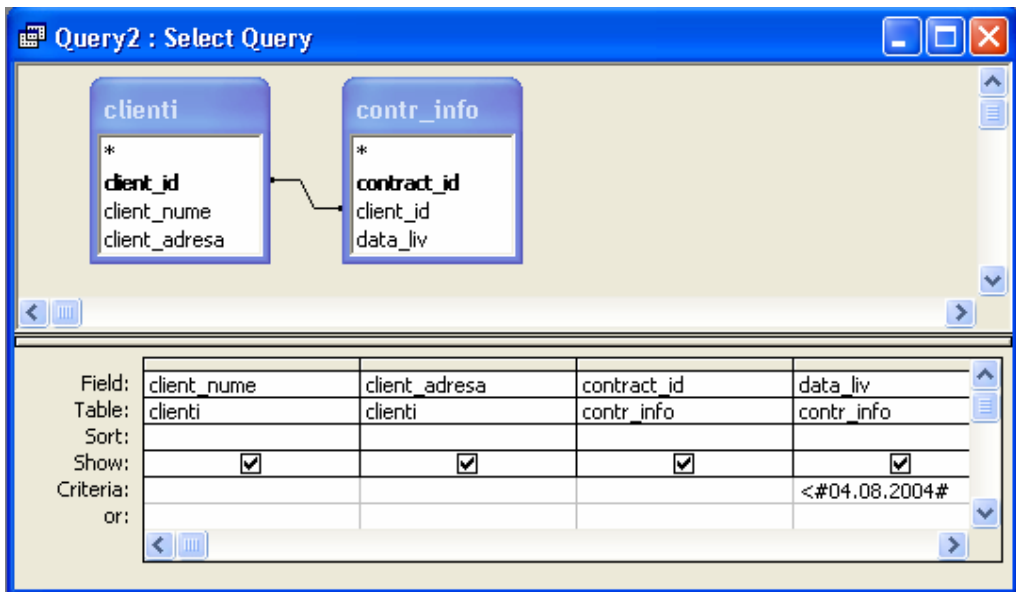
- se calculează numărul liniilor din tabelul *contr\_info*. Aici apare fiecare contract, însă doar într-o singură linie
- se lansează utilitarul de construcție a interogărilor (*design view*)
- se execută clic – dreapta în cadrul ferestrei apărute, și se alege *SQL View*

- se introduce comanda SQL, care returnează numărul contractelor înregistrate în tabelul *contr\_info*
- se alege din meniu comanda *Run*, pentru a executa interogarea
- în fereastra *SQL View*, interogarea arată astfel:



4. Să se afișeze toate informațiile disponibile despre clienții acelor contracte, care se livrează înainte de 10 august 2004

- pentru această interogare avem nevoie de două tabel: *contr\_info*, pentru a afla data livrării, și de *clienți*, pentru a prelua datele necesare despre clienți
- se adaugă tabelele *produse* și *contracte* în *Design View*
- se selectează câmpurile *client\_nume*, *client\_adresa*, *contract\_id*, *data\_liv*
- se fixează condiția pentru *data\_liv*: < 4.08.2004



- Comanda SQL corespunzătoare este:

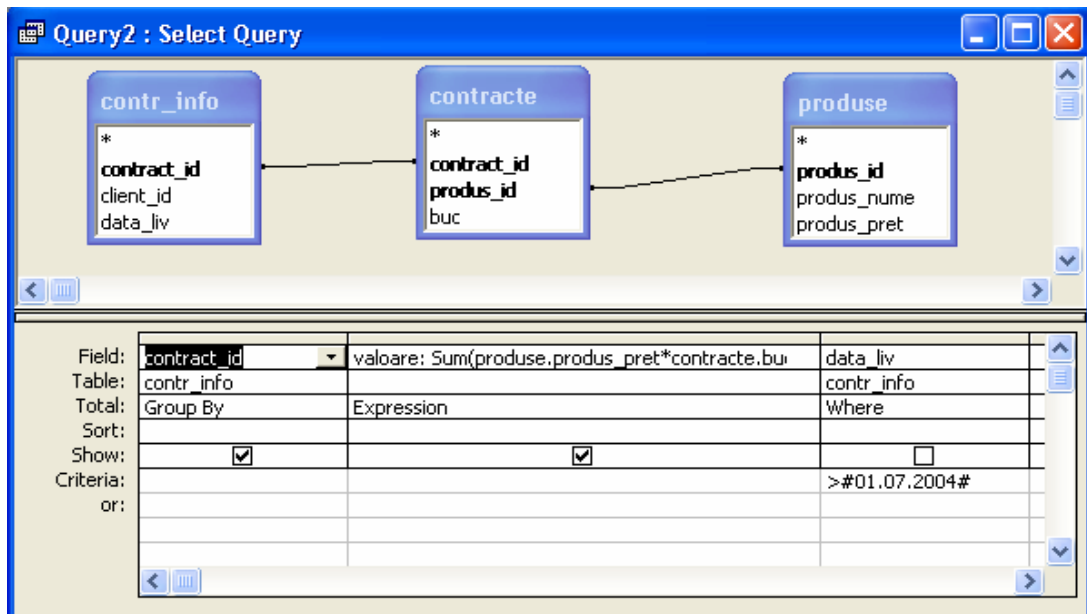
```
SELECT clicenti.client_nume, clicenti.client_adresa, contr_info.contract_id, contr_info.data_liv
FROM clicenti INNER JOIN contr_info ON clicenti.client_id = contr_info.client_id WHERE
((contr_info.data_liv)<#8/4/2004#);
```

- se alege din meniu comanda *Run*, pentru a executa interogarea

5. Să se afișeze valoarea totală a fiecărui contract care se livrează după 1 iulie 2004

- valoarea totală a unui contract se calculează ca suma înmulțirii dintre produse și numărul de bucăți în care acestea vor fi livrate: câmpurile *produs\_pret* din tabelul *produse*, respectiv *buc* din tabelul *contracte*
- se apelează la un treilea tabel: *contracte*, pentru a face legătura necesară
- se adaugă cele trei tabele în *Design View*
- se selectează câmpurile *contract\_id*, *data\_liv*, iar pentru valoarea totală a unui contract avem nevoie de o expresie compusă:  

$$\text{sum} (\text{produse.produs\_pret} * \text{contracte.buc})$$
- pentru a calcula suma, se grupează toate expresiile în funcție de *contract\_id*, prin elementul *group by*
- se fixează condiția pentru *data\_liv*: *< 1.07.2004*. Acest câmp nu trebuie afișat, așa că nu selectăm opțiune *show* pentru el
- se alege din meniu comanda *Run*, pentru a executa interogarea



- Comanda SQL corespunzătoare este:

*SELECT contr\_info.contract\_id, sum(produse.produs\_pret\*contracte.buc) AS valoare  
FROM produse INNER JOIN (contr\_info INNER JOIN contracte ON contr\_info.contract\_id  
= contracte.contract\_id) ON produse.produs\_id = contracte.produs\_id WHERE  
(((contr\_info.data\_liv)>=#7/1/2004#)) GROUP BY contr\_info.contract\_id;*

### **Probleme propuse:**

1. Folosind baza de date de la problema din capitolul 1.1, să se afișeze toți studenții al căror prenume este "Andrei".
2. Realizând o interogare asupra bazei de date de la aceeași problemă, să se afișeze toate notele date de profesorul cu numele "Man".
3. Plecând de la datele din tabela profesori, să se afișeze toate informațiile existente despre profesorul cu numele "Dinică".
4. Sa se afișeze toate notele mai mari decât 5 existente în tabela note.
5. Să se afișeze numele tuturor studenților care au restante..
- 6.Să se afișeze toate materiile la care exista restante.
7. Să se afișeze toți profesorii care au dat note sub cinci.
8. Să se afișeze toți studenții al căror cod este egal cu o valoare citită de la tastatura.
9. Să se determine care sunt materiile la care s-au dat note de 10.
10. Să se afișeze studenții care au obținut note egale cu cinci.