## 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 Micrososft 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.

Efectuand dublu click pe aceasta optiune apare fereastra de mai jos:

Simple Query Wizard	
	Which fields do you want in your query?           You can choose from more than one table or query.
Tables/Queries Table: materii Available Fields:	Selected Fields:
ID denumire descriere	
	Cancel < Back Next > Finish

Se selectează tabela dorită, în cazul nostru "studenți", din care se includ in query campurile nume si prenume. Aceasta operatie se realizeaza efectuand click pe butonul cu semnul ">" pentru fiecare camp in parte.

Simple Query Wizard	Which	fields do you wao	tip vour querv?	
	You ca	n choose from mo	re than one table	or query.
Tables/Queries Table: studenti	<u> </u>			
<u>A</u> vailable Fields:		Selected Fields:		_
adrese telefon email	>>	prenume		
	<<			
	Cancel	< <u>B</u> ack	<u>N</u> ext >	Einish

După aceea se selctează tabela "note", din care se includ in query câmpurile nota si data.

	Which fields do you want in your query? You can choose from more than one table or query.
Tables/Queries	J
<u>A</u> vailable Fields: id_stud	Selected Fields:
id_prof id_mat	<pre>&gt;&gt; prenume nota data</pre>

Apoi alegem denumirea din tabela "materii".

Simple Query Wizard	
	Which fields do you want in your query? You can choose from more than one table or query.
Tables/Queries Table: materii	]
<u>A</u> vailable Fields: ID descriere	Selected Fields:
Q	ancel < <u>B</u> ack <u>N</u> ext > <u>F</u> inish

In acest moment se apasă butonul Next.

AAA 2000 0000     AAA 2000 0000     GC 2000 0000     BBBB 2000 0000	Detail (shows every field of every record)     Summary	
3 DDD ####### 2 AAA XXX XXX XXX 3 BBB XXX XXX XXX 4 CCC XXX XXX XXX 5 DDD XXX XXX XXX 6 DDD XXX XXX XXX		

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

M	What title do you want for your query?
	<ul> <li>That's all the information the wizard needs to create your query.</li> <li>Do you want to open the query or modify the query's design?</li> <li>Open the query to view information.</li> <li>Modify the query design.</li> </ul>
	Display Help on working with the query?

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

Pentru a restricționa căutarea la codul unui anumit student, va trebui să modificăm interogarea alegând opțiunea *Design View* din meniul de context care se obține efectuând click dreapta pe numele interogării.



In continuare restricționăm query-ul la studenții cu numele de "Tasica". Acest lucru se face scriind "Tasica" in zona de *Criteria* de sub *nume*.

🖷 student	ti Query : Select Qu	uery					
stu	denti	materii	note				
TD num prer adre	e nume isa	ID denumire descriere	nota data id_stud id_prof				
							<u>&gt;</u>
	[	I	1			1	
Field:	hume 💽	prenume	nota	data	denumire		<u>^</u>
Table:	studenti	studenti	note	note	materii		
Sort:							_
Show:		~					
Criteria:	"Tasica"					S	
or:						1	× 1
	<u>s</u>						>
1							

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".

studenti Query : Select Query  studenti  t D nume prenume edesca	SQL View       Image: Datasheet View					
	SQL Specific					>
	Relationships			r		
Field: nume renume	- F P Dropartian	ta	denumire			<u>^</u>
Table: studenti studenti		te	materii		1	
Sort:	and the second	1001	1			
Show: 🔽 🗹		✓				
Criteria: "Tasica"						
or:						~
	104		N6 2		HA.	>

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 si înregistrările selectate de query. Spre exemplu da în loc de "Tasica" scriem "Ion" și pe ecran va apare:

	studenti Que	ry : Select Query	/				
	nume	prenume	nota	data	denumire		
•	lon	Andrei	8	21.03.2004	Bazele Informaticii		
	lon	Andrei	5	22.03.2004	Informatica Aplicata		
*	1.2 × 11.						

Se pot face chiar și modificări mai complexe, cum ar fi să cerem numele de la tastatura. Dacă scriem in loc de "Tasica" un cuvant oarecare, <u>fără să fie încadrat de ghilimele și fără să</u> <u>conțină spații</u>, 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));

In 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:

Ⅲ	clienti : Table			X
	Field Name	Data Type	Description	~
8	client_id	AutoNumber		
	client_nume	Text		_
▶	client_adresa	Text		_
				-
				-
⊢				~
,	1	Field Properties		
0 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	Seneral Lookup 50 Field Size 50 Format 50 Form	s S Control ne		

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

Ⅲ	produse : Table			×
	Field Name	Data Type	Description	~
R	produs_id	AutoNumber		
	produs_nume	Text		
	produs_pret	Currency		
				_
				<u>×</u>
		Field Propertie	s	
	General Lookup Eield Size Lo Field Size Lo New Values In Format Caption Indexed Ye	ng Integer crement s (No Duplicates)		

• 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*:

Ⅲ	contracte : Table			×		
	Field Name	Data Type	Description	~		
8	contract_id	Number		Ħ		
8	produs_id	Number				
	buc	Number				
				<u> </u>		
		Field Properties				
	General Lookup					
F	Field Size	.ong Integer				
F	Format					
0	Decimal Places	Auto				
1	input Mask					
0	Caption					
[	Default Value (	)				
1	/alidation Rule					
	/alidation Text					
F	Required	Jo				
i	indexed	Yes (Dunlicates OK)				
1		ios (bapileacos orty				

ⅲ	contr_info : Table				×
	Field Name		Data Type	Description	~
8	contract_id		Number		T
	client_id		Number		-
	data_liv		Date/Time		
					v
	l Fia	Jd D	Properties	1	
	110	au r	Topercies		
6	Seneral Lookup				
F	ield Size	Lor	ng Integer		
F	Format				
C	Decimal Places	Aut	to		
I	nput Mask				
0	Laption				
C	Default Value	0			
N	alidation Rule				
N	alidation Text				
F	Required	No			
I	ndexed	Yes	; (No Duplicates)		
-					

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 *contract*. 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

Simple Query Wizard	
	Which fields do you want in your query? You can choose from more than one table or query.
Tables/Queries	
Available Fields:	Selected Fields:
client_id	client_nume       client_adresa       <
Ca	ncel < <u>B</u> ack <u>N</u> ext > <u>F</u> inish

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.produs\_nume, produse.produs\_pret, contracte.buc, contracte.contract\_id FROM produse INNER JOIN contracte ON produse.produs\_id = contracte.produs\_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.produs\_nume) FROM produse INNER JOIN contracte ON produse.produs\_id = contracte.produs\_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

🖷 Query2 : Selec	t Query				
clienti * dient_id client_nume client_adresa		contr_info * contract_id client_id data_liv			
Field: client n	ume	client adresa	contra	ict id	data liv 🔼
Table: clienti		clienti	contr	info	contr_info 📃
Sort:					
Criteria:	Criteria:			⊻	∠#04.08.2004#
or:					<
<					>

• Comanda SQL corespunzătoare este:

SELECT clienti.client\_nume, clienti.client\_adresa, contr\_info.contract\_id, contr\_info.data\_liv FROM clienti INNER JOIN contr\_info ON clienti.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:

sum ( produse.produs\_pret \* 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

🖬 Query2 : Select Query 📃 🗖 🔀							
cont clien data	ntr_info ract_id t_id a_liv		contracte     produse       *     rodus_id       produs_id     produs_nume       buc     produs_pret	< <			
Field:	contract id	•	valoare: Sum(produse.produs pret*contracte.bu) data liv				
Table:	contr_info		contr_info				
Total: Sort:	Group By		Expression Where				
Show:	Show:						
Criteria:			>#01.07.2004#	-			
or:							
				~			
	<			>			

• 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.