

Creare baza de date:

```
CREATE DATABASE [universitate]
GO
USE [universitate]
GO

CREATE TABLE [dbo].[Specializari](
    [id] [int] NOT NULL PRIMARY KEY,
    [nume] [varchar](50) NULL
)

CREATE TABLE [dbo].[Grupe](
    [id] [int] NOT NULL PRIMARY KEY,
    [specializareId] [int] NOT NULL foreign key references Specializari(id)
)

CREATE TABLE [dbo].[Funcții](
    [id] [int] NOT NULL PRIMARY KEY,
    [nume] [varchar](50) NULL,
    [salar] [int] NULL
)

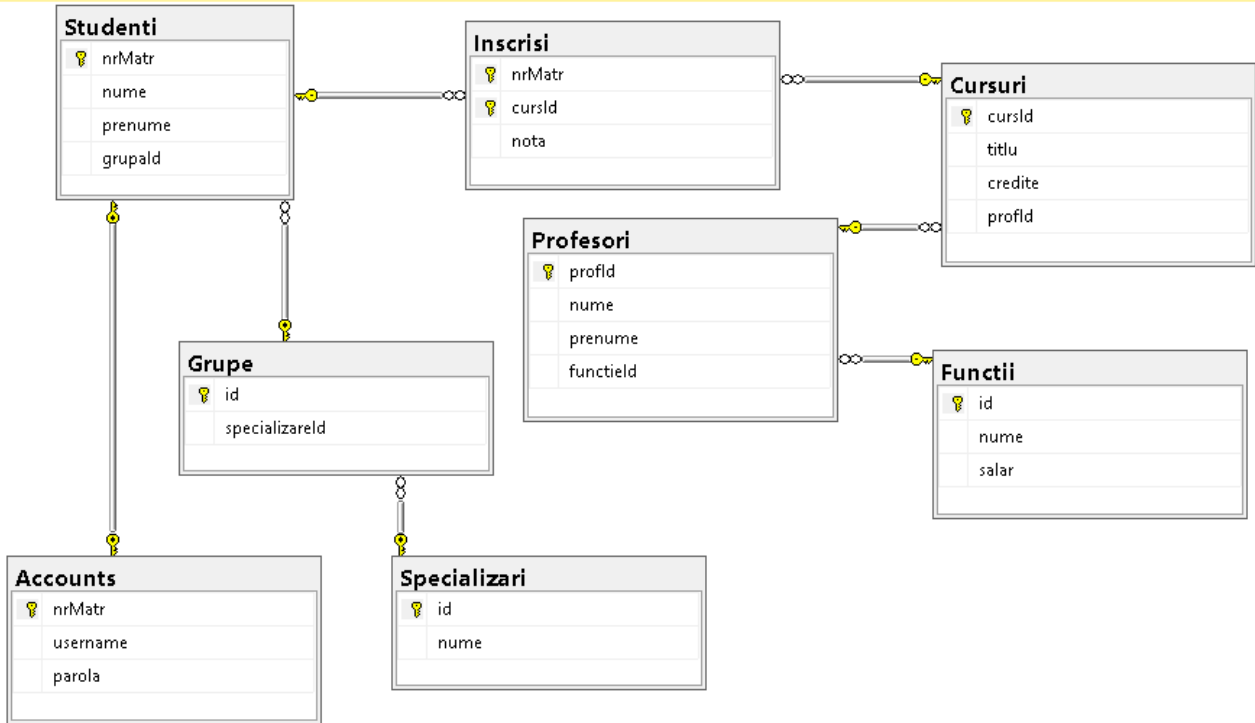
CREATE TABLE [dbo].[Profesori](
    [profId] [int] NOT NULL PRIMARY KEY,
    [nume] [varchar](50) NULL,
    [prenume] [varchar](50) NULL,
    [functieId] [int] NOT NULL foreign key references Funcții(id)
)

CREATE TABLE [dbo].[Cursuri](
    [cursId] [nvarchar](50) NOT NULL PRIMARY KEY,
    [titlu] [nvarchar](50) NULL,
    [credite] [int] NULL,
    [profId] [int] NOT NULL foreign key references Profesori(profId)
)

CREATE TABLE [dbo].[Studenti](
    [nrMatr] [int] NOT NULL PRIMARY KEY,
    [nume] [varchar](50) NULL,
    [prenume] [varchar](50) NULL,
    [grupaId] [int] NOT NULL foreign key references Grupe(id)
)

CREATE TABLE [dbo].[Accounts](
    [nrMatr] [int] foreign key references Studenti(nrMatr),
    [username] [varchar](50) NOT NULL,
    [parola] [varchar](50) NOT NULL,
    Constraint pk_Accounts PRIMARY KEY(nrMatr)
)

CREATE TABLE [dbo].[Inscriși](
    [nrMatr] [int] NOT NULL foreign key references Studenti(nrMatr),
    [cursId] [nvarchar](50) NOT NULL foreign key references Cursuri(cursId),
    [nota] [int] NULL
    constraint pk_Inscriși PRIMARY KEY(nrMatr, cursId)
)
```



Insert into tables: Scrieti comenzi SQL pentru un tabel care sa insereze / modifice / stearga date.

- Conteaza ordinea in care introduceti date in tabele in functie de FK
- Incercati sa introduceti date plauzibile care sa aiba sens
- Introduceti mai multe randuri deodata

USE [universitate]

GO

```

insert into Specializari values (1, 'IR'), (2, 'IE')
insert into Grupe values (711, 1), (712, 1), (911, 2), (912, 2)
insert into Studenti values (1010, 'Popescu', 'Ioana', 711)
insert into Studenti values (1011, 'Popescu', 'Andrei', 711),
(1012, 'Dobrescu', 'Mihai', 712), (1013, 'Candea', 'Ana', 712)
insert into Accounts values (1010, 'tdir1010', 'temp1')
insert into Functii values(1, 'lector universitar', 2000), (2, 'conferentiar
universitar', 3000)
insert into Profesori values(1, 'Bold', 'Ionela', 1)
insert into Cursuri values ('BD', 'Baze de date', 6, 1)
insert into Cursuri values ('SGBD', 'Sisteme de gestiune a bazelor de date', 5, 1), ('SDA',
'Structurii de date', 6, 1)
insert into Inscrisi values (1010, 'BD', 6), (1011, 'BD', 7), (1010, 'SGBD', 10),
(1012, 'SGBD', 9)
  
```

```
select * from Studenti
select * from Inscrisi
```

	nrMatr	nume	prenume	grupald
1	1010	Popescu	Ioana	711
2	1011	Popescu	Andrei	711
3	1012	Dobrescu	Mihai	712
4	1013	Candea	Ana	712

	nrMatr	cursId	nota
1	1010	BD	6
2	1010	SGBD	10
3	1011	BD	7
4	1012	SGBD	9

In comenzile de modificare sau stergere folositi cel puțin o data in clauza WHERE: operatori logici, operatori relationali, IS [NOT] NULL.

```
update Studenti set grupaId=712 where nrMatr=1010
--toti care nu au grupa ii asignam ultimei grupe 912
update Studenti set grupaId=912 where grupaId is NULL
```

```
delete from Accounts where username is NULL
```

a. o interogare cu operatia de reuniune: cu UNION [ALL] sau OR;

```
-- UNION
-- toti studentii de la IR (care sunt in grupa 711 SAU grupa 712)
select * from Studenti where grupaId=711
union select * from Studenti where grupaId = 712
-- equivalent with
select * from Studenti where grupaId=711 OR grupaId=712
```

	nrMatr	nume	prenume	grupald
1	1011	Popescu	Andrei	711
2	1010	Popescu	Ioana	712
3	1012	Dobrescu	Mihai	712
4	1013	Candea	Ana	712

	nrMatr	nume	prenume	grupald
1	1010	Popescu	Ioana	712
2	1011	Popescu	Andrei	711
3	1012	Dobrescu	Mihai	712
4	1013	Candea	Ana	712

b. o interogare cu operatia de intersectie: cu INTERSECT sau IN;

```
--JOIN: cum legam doua tabele?
--vrem sa vedem fiecare student la ce curs s-a inscris
select * from Studenti S
inner join Inscrisi I on S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
```

	nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profilid
1	1010	Popescu	Ioana	712	1010	BD	6	BD	Baze de date	6	1
2	1010	Popescu	Ioana	712	1010	SGBD	10	SGBD	Sisteme de gestiune a bazelor de date	5	1
3	1011	Popescu	Andrei	711	1011	BD	7	BD	Baze de date	6	1
4	1012	Dobrescu	Mihai	712	1012	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1

```

-- INTERSECT
-- cautam studentii inscrisi SI la 'BD' SI la 'SGBD'
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'BD'
intersect
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'SGBD'

-- sau cu IN
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'BD' and S.nrMatr in
(select S.nrMatr from Studenti S inner join Inscrisi I on S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'SGBD')

```

Results		Messages	
nrMatr	numa	prenuma	
1	1010	Popescu	loana

nrMatr	numa	prenuma	
1	1010	Popescu	loana

c. o interogare cu operatia de diferenta: cu EXCEPT sau NOT IN;

```

-- EXCEPT
--cautam studentii inscrisi la 'BD', dar care NU s-au inscris la 'SGBD'
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'BD'
except
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'SGBD'

--sau cu NOT IN
select S.nrMatr, S.numa, S.prenume from Studenti S inner join Inscrisi I on
S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'BD' and S.nrMatr not in
(select S.nrMatr from Studenti S inner join Inscrisi I on S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
where C.cursId = 'SGBD')

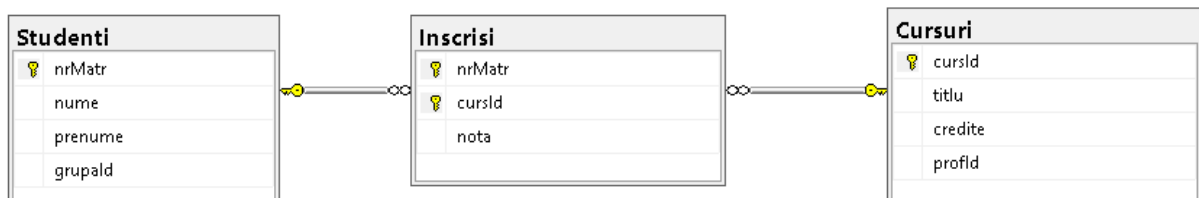
```

nrMatr	nume	prenume
1	1011	Popescu Andrei

nrMatr	nume	prenume
1	1011	Popescu Andrei

d. o interogare cu INNER JOIN si o interogare cu unul dintre operatorii: LEFT JOIN, RIGHT JOIN, FULL JOIN; una dintre interogari va extrage date din trei tabele aflate in relatie many-to-many;

- ce inseamna left, right si full join?
- relația considerată va fi Studenți-Înscriși-Cursuri (m-n)



INNER JOIN

```
-- afisam studentii inscrisi la cursuri -- nu exista null nici la studentii, nici la cursuri
select * from Studenți S inner join Înscriși I on S.nrMatr=I.nrMatr
inner join Cursuri C on I.cursId=C.cursId
```

nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profId	
1	1010	Popescu	loana	712	1010	BD	6	BD	Baze de date	6	1
2	1010	Popescu	loana	712	1010	SGBD	10	SGBD	Sisteme de gestiune a bazelor de date	5	1
3	1011	Popescu	Andrei	711	1011	BD	7	BD	Baze de date	6	1
4	1012	Dobrescu	Mihai	712	1012	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1

LEFT OUTER JOIN

```
-- vrem sa vedem si studentii care nu s-au inscris la nimic -- null la cursuri
select * from Studenți S left outer join Înscriși I on S.nrMatr=I.nrMatr
left outer join Cursuri C on I.cursId=C.cursId
```

nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profId	
1	1010	Popescu	loana	712	1010	BD	6	BD	Baze de date	6	1
2	1010	Popescu	loana	712	1010	SGBD	10	SGBD	Sisteme de gestiune a bazelor de date	5	1
3	1011	Popescu	Andrei	711	1011	BD	7	BD	Baze de date	6	1
4	1012	Dobrescu	Mihai	712	1012	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1
5	1013	Candea	Ana	712	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
--vrem sa vedem doar studentii care nu s-au inscris la nimic
```

```
select * from Studenti S left outer join Inscrisi I on S.nrMatr=I.nrMatr
left outer join Cursuri C on I.cursId=C.cursId
where C.cursId is NULL
```

	nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profId
1	1013	Candea	Ana	712	NULL	NULL	NULL	NULL	NULL	NULL	NULL

RIGHT OUTER JOIN

```
-- right join
-- vrem sa vedem si cursurile la care nu s-au inscris student -- null la studenti
select * from Studenti S right outer join Inscrisi I on S.nrMatr=I.nrMatr
right outer join Cursuri C on I.cursId=C.cursId
```

	nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profId
1	1010	Popescu	Ioana	712	1010	BD	6	BD	Baze de date	6	1
2	1011	Popescu	Andrei	711	1011	BD	7	BD	Baze de date	6	1
3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	SDA	Structuri de date	6	1
4	1010	Popescu	Ioana	712	1010	SGBD	10	SGBD	Sisteme de gestiune a bazelor de date	5	1
5	1012	Dobrescu	Mihai	712	1012	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1

FULL OUTER JOIN

```
-- full join
--vrem sa vedem si studenti care nu s-au inscris la nimic, dar si cursuri la care nu s-a
inscris nici un student
-- null si la studenti si la cursuri
select * from Studenti S full outer join Inscrisi I on S.nrMatr=I.nrMatr
full outer join Cursuri C on I.cursId=C.cursId
```

	nrMatr	nume	prenume	grupald	nrMatr	cursId	nota	cursId	titlu	credite	profId
1	1010	Popescu	Ioana	712	1010	BD	6	BD	Baze de date	6	1
2	1010	Popescu	Ioana	712	1010	SGBD	10	SGBD	Sisteme de gestiune a bazelor de date	5	1
3	1011	Popescu	Andrei	711	1011	BD	7	BD	Baze de date	6	1
4	1012	Dobrescu	Mihai	712	1012	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1
5	1013	Candea	Ana	712	NULL	NULL	NULL	NULL	NULL	NULL	NULL
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	SDA	Structuri de date	6	1

- e. o interogare care contine o subinterogare in clauza WHERE: cu IN sau EXISTS;
- f. o interogare care contine o subinterogare in clauza FROM;

```
-- E
-- F
-- gasiti functiile pentru care avem profesori
select F.id, F.nume, F.salar from Functii F inner join Profesori P on F.id=P.functieId

-- cu EXISTS
select * from Functii F
where exists (select * from Profesori where functieId = F.id)
```

```
-- echivalent cu IN
select * from Functii F
where F.id IN (select functieId from Profesori)

-- echivalent cu FROM
select A.id, A.num, A.salar from (select F.id, F.num, F.salar from Functii F inner join
Profesori P on F.id=P.functieId)A
```

	id	nume	salar
1	1	lector universitar	2000
1	1	lector universitar	2000
1	1	lector universitar	2000
1	1	lector universitar	2000

```
-- media notelor pe curs
select * from ( select I.cursId, avg(nota) as medie from Inscrisi I group by I.cursId )
as Medii inner join Cursuri C on Medii.cursId = C.cursId
```

	cursId	medie	cursId	titlu	credite	profId
1	BD	6	BD	Baze de date	6	1
2	SGBD	9	SGBD	Sisteme de gestiune a bazelor de date	5	1

g. 2 interogari cu clauza GROUP BY; una dintrele va continesiclauza HAVING; se vor folosi cel puțin 2 operatori de agregare dintre: COUNT, SUM, AVG, MIN, MAX;

- ce coloane pot pune in select cand am group by
- ce conditii se pun in WHERE si ce cond in HAVING

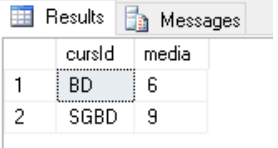
```
--cati studenti am in fiecare grupa
select G.id, count(*) as nrStud
from Grupe G inner join Studenti S on G.id=S.grupaId
group by G.id

--doar pentru grupele cu minim 2 studenti -- having count(*) >=2
select G.id, count(*) as nrStud
from Grupe G inner join Studenti S on G.id=S.grupaId
group by G.id
having count(*) >=2
```

	id	nrStud
1	711	1
2	712	3


	id	nrStud
1	712	3

```
-- care e media notelor la fiecare curs
select C.cursId, AVG(I.nota) as media
from Cursuri C inner join Inscrisi I on C.cursId = I.cursId
group by C.cursId
```



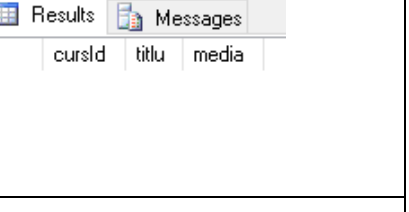
	cursId	media
1	BD	6
2	SGBD	9

```
--cum fac sa afisez si titlul cursului?
select C.cursId, C.titlu, AVG(I.nota) as media
from Cursuri C inner join Inscrisi I on C.cursId = I.cursId
group by C.cursId, C.titlu
```



	cursId	titlu	media
1	BD	Baze de date	6
2	SGBD	Sisteme de gestiune a bazelor de date	9

```
--cum fac sa afisez si titlul cursului doar pentru cursurile cu
media sub 5
select C.cursId, C.titlu, AVG(I.nota) as media
from Cursuri C inner join Inscrisi I on C.cursId = I.cursId
group by C.cursId, C.titlu
--doar pentru cursurile cu media sub 5
having AVG(I.nota) <5
--pot afisa si numele studentilor?? --NU
```

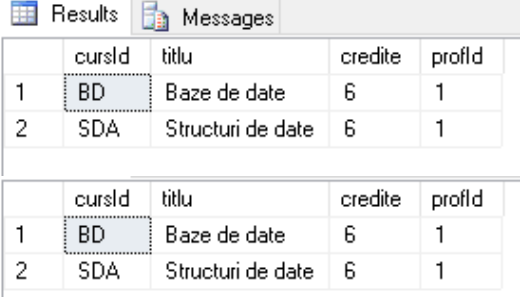


	cursId	titlu	media
--	--------	-------	-------

h. o interogareimbricata cu unuldintreoperatorii ANY sau ALL, undeoperatorul relational este din multimea {=, <, <=, >, >=, <>}.

```
--gasiti cursurile cu nr de credite mai mare decat un alt curs
select * from Cursuri C
where C.credite > ANY (select C2.credite from Cursuri C2)

--gasiti cursurile cu numar maxim de credite
select * from Cursuri C
where C.credite >= ALL (select C2.credite from Cursuri C2)
```



	cursId	titlu	credite	profId
1	BD	Baze de date	6	1
2	SDA	Structuri de date	6	1

```
--ce se intampla la >ALL??? - nu se mai afiseaza nimic - nu mai are sens
select * from Cursuri C
where C.credite > ALL (select C2.credite from Cursuri C2)
```



	cursId	titlu	credite	profId
--	--------	-------	---------	--------

```
--gasiti cursurile cu nr de credite mai mic decat un alt curs
select * from Cursuri C
where C.credite < ANY (select C2.credite from Cursuri C2)

--gasiti cursurile cu numar minim de credite
```



```
select * from Cursuri C
where C.credite <= ALL (select C2.credite from Cursuri C2)
```

cursId	titlu	credite	profId
1	SGBD	5	1

Bonus

```
--** gasiti studentii care au media notelor mai mare decat toate notele studentului 'Popescu Andrei'
select S.nrMatr, AVG(I.nota) as medie
from Studenti S inner join Inscrisi I on S.nrMatr = I.nrMatr
group by S.nrMatr
having AVG(I.nota) > ALL (select I2.nota from Inscrisi I2
inner join Studenti S2 on I2.nrMatr=S2.nrMatr
where S2.numa = 'Popescu' and S2.prenume = 'Andrei')
```

nrMatr	medie	
1	1010	8
2	1012	9

In interogările de mai sus se vorfolosicelputin o data:

- **Expresii aritmetice in clauza SELECT;**

```
--se face o statistica si vrem sa calculam pentru fiecare student suma notelor
inmultite cu nr de credite
select I.nrMatr, I.cursId, I.nota*C.credite as pondere
from Inscrisi I inner join Cursuri C on I.cursId=C.cursId
```

nrMatr	cursId	pondere	
1	1010	BD	36
2	1010	SGBD	50
3	1011	BD	42
4	1012	SGBD	45

```
--avem subselect in FROM!
select T.nrMatr, sum(pondere) as statistica
from (select I.nrMatr, I.cursId, I.nota*C.credite as pondere
from Inscrisi I inner join Cursuri C on I.cursId=C.cursId)as T
group by T.nrMatr
```

	nrMatr	statistica
1	1010	86
2	1011	42
3	1012	45

- **DISTINCT;**

```
--afisati cursurile la care s-au inscris studentii -> CU duplicate (daca exista)!
```

```
select distinct C.titlu
from Cursuri C inner join Inscrisi I on C.cursId= I.cursId
```

```
--afisati cursurile la care s-au inscris studentii -> fara duplicate!
```

```
select distinct C.titlu
from Cursuri C inner join Inscrisi I on C.cursId= I.cursId
```

	titlu
1	Baze de date
2	Sisteme de gestiune a bazelor de date

	titlu
1	Baze de date
2	Sisteme de gestiune a bazelor de date

- **TOP**

```
--primii 2 studenti in ordine alfabetica
```

```
-- primii 2 studenti ordonati dupa nume si in cadrul aceluiasi nume, dupa prenume alfabetic
```

```
select top 2 * from Studenti order by nume, prenume asc
```

	nrMatr	nume	prenume	grupald
1	1013	Candea	Ana	712
2	1012	Dobrescu	Mihai	712

- **ORDER BY**

```
-- ORDER BY
```

```
-- studentii ordonati alfabetic dupa nume
```

```
select * from Studenti order by nume -- asc e implicit
```

```
-- studentii ordonati descrescator dupa grupa
```

```
select * from Studenti order by grupaId DESC
```

```
-- studentii ordonati alfabetic dupa nume si in cadrul aceluiasi nume, dupa grupa crescator
```

```
select * from Studenti order by nume, grupaId
```

Results		Messages		
	nrMatr	nume	prenume	grupald
1	1013	Candea	Ana	712
2	1012	Dobrescu	Mihai	712
3	1010	Popescu	Ioana	712
4	1011	Popescu	Andrei	711

	nrMatr	nume	prenume	grupald
1	1010	Popescu	Ioana	712
2	1012	Dobrescu	Mihai	712
3	1013	Candea	Ana	712
4	1011	Popescu	Andrei	711

	nrMatr	nume	prenume	grupald
1	1013	Candea	Ana	712
2	1012	Dobrescu	Mihai	712
3	1011	Popescu	Andrei	711
4	1010	Popescu	Ioana	712