Universitatea „Ovidius” din Constanţa

Facultatea de Matematică ṣi Informatică

Disciplina: Baze de date

Specializarea: Matematică-Informatică

Anul III

***PROIECT BAZE DE DATE***

Coordonator: Student:

Lect.dr. Băutu Elena

Molomer Sibel Negruț Valentin Mihai

**ECHIPĂ SPORTIVĂ**

Modelul Entităţi – Asociaţii permite reprezentarea informaţiilor despre structura bazelor de date folosind trei elemente de construcţie: entităţi, atribute ale entităţilor ṣi asocieri între entităţi.

Entităţile modelează clase de obiecte concrete sau abstracte despre care se colectează informaţii, au existenţă independentă ṣi pot fi identificate în mod unic. Acestea se reprezintă grafic printr-un dreptunghi în interiorul căruia este scris numele entităţii.

Exemple de entităţi: ANTRENORI, ECHIPE, JUCĂTORI, MECIURI, SĂLI, ADRESE, ARBITRII, COMPLEX\_SPORTIV, ECHIPA\_MEDICALA, MANAGERI.

Atributele modelează proprietăţi distincte ale entităţilor. De exemplu, entitatea ANTRENORI are ca atribute *IDAntrenor* (cheie străină), *nume*, *prenume* ṣi *CNP(*cheie primară*)*. În procesul de modelare vor fi luate în considerare doar acele proprietăţi ale entităţilor care sunt semnificative pentru aplicaţia respectivă. În general, un atribut se reprezintă grafic printr-o elipsă, în interiorul căreia apare numele atributului, fiind legată printr-un segment de entitatea corespunzătoare. În reprezentarea aceasta, atributele sunt trecute sub entităţile corespunzătoare.

Asociaţiile reprezintă o conexiune între entităţi (chiar ṣi între asociaţii, sau entităţi de acelaṣi tip). De exemplu, între entităţile JUCĂTORI ȘI ECHIPE există o asociere many-to-one (n- la- 1), adică mai mulți jucători pot face parte dintr-o echipă, iar o echipă poate avea mai mulți jucăatori. Asociaţiile se reprezintă grafic prin romburi, fiind conectate prin segmente la entităţile pe care le leagă, ṣi având scris în interior numele asocierii.

Pentru entităţile A ṣi B, putem avea una din următoarele tipuri de asocieri R:

* One-to-one (1-la-1): O entitate din A este asociată cel mult unei entităţi din B şi o entitate din B este asociată cel mult unei entităţi din A
* One-to-many (1-la-n): O entitate din A este asociată cu oricâte entităţi din B şi o entitate din B este asociată cel mult unei entităţi din A (de exemplu, între entităţile SĂLI și MECIURI).
* Many-to-one (n-la-1): O entitate din A este asociată cel mult unei entităţi din B şi o entitate din B este asociată oricâtor (0 sau mai multe) entităţi din A (de exemplu, între entităţile JUCĂTORI și ECHIPE ,dar și între entitățile ANTRENORI și ECHIPE).
* Many-to-many (n-la-n): O entitate din A este asociată cu oricâte entităţi din B şi o entitate din B este asociată cu oricâte entităţi din A (de exemplu, între entităţile JUCĂTORI și MECIURI).

Se consideră subuniversul unei echipe sportive de handbal, care este formată din echipe, fiecare având antrenorii săi proprii, jucători, cărora li se vor corela niște adrese, meciuri jurizate de arbitri ce vor avea loc în săli aflate în diferite complexe sportive. De asemenea fiecare echipă de jucători va avea la dispoziție câte o echipă medicală pe parcursul fiecărui meci pentru a nu exista accidente în săli. Pentru a realiza acest subunivers este nevoie de o bază de date în care vom stoca toate informaţiile necesare, pe care am numit-o ECHIPĂ SPORTIVĂ.

Pentru fiecare jucător ne intereseaza să știm numele, prenumele, CNP-ul, vârsta, numărul de pe tricou, adresa și echipa la care acesta joacă.

Despre echipe vrem să știm numele acesteia, numărul echipei, echipa medicală ascociată, managerul ce se ocupă de echipă, orașul și numărul jucătorilor din care aceasta este formată.

Pentru antrenori vom avea ca și atribute numele, prenumele, CNP-ul și echipa la care antrenează.

Pentru meciurile care sunt jucate, vom avea de reținut data la care acesta are loc, ora, arbitrul și sala în care se ține.

Despre sălile unde vor avea loc meciurile ne interesează numărul acesteia, capacitatea pe care o are și complexul sportiv în care se află.

Adresele vor avea ca și atribute orașul, strada și numărul unde se află.

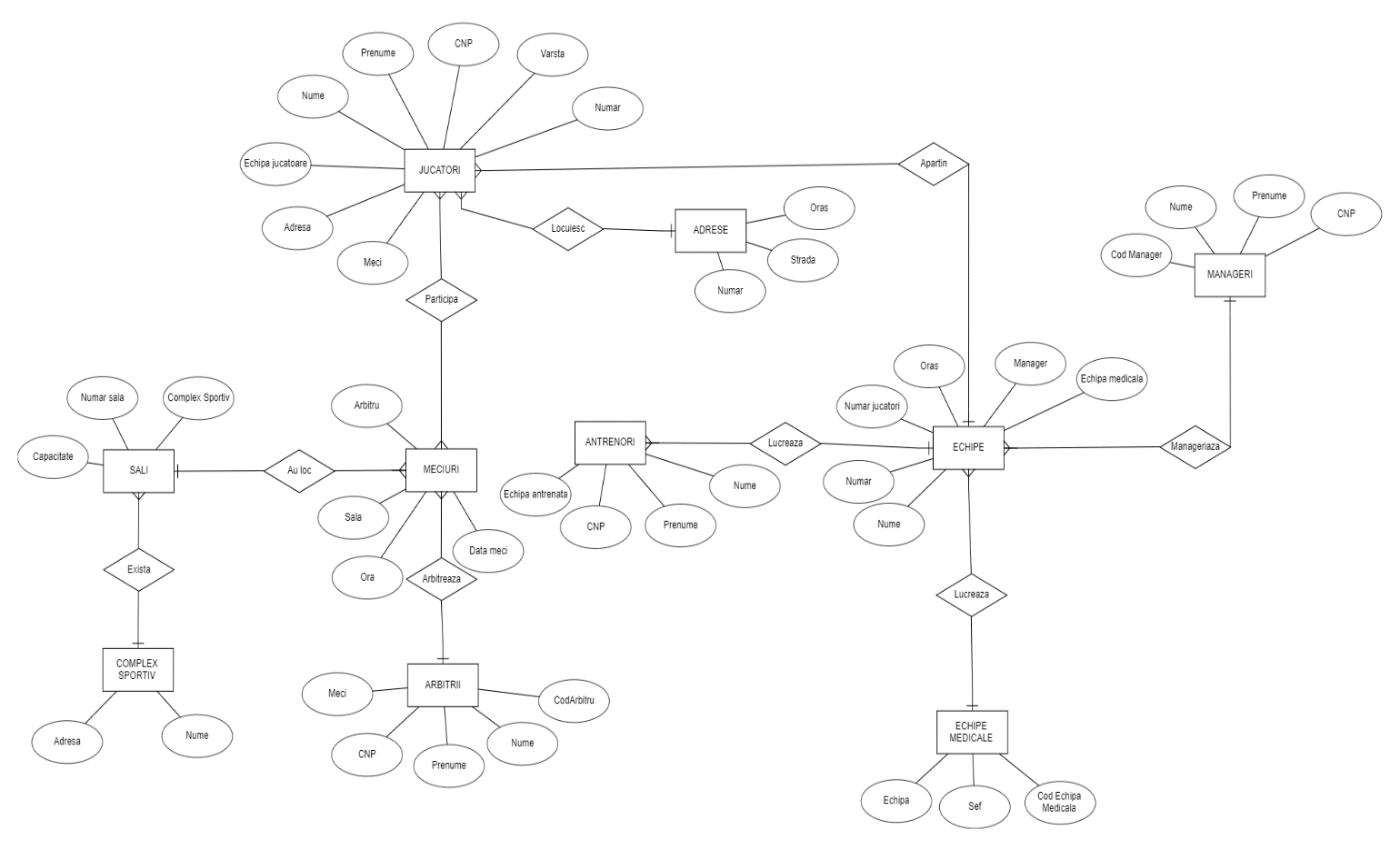
Arbitrii meciurilor vor avea cod arbitru, nume,prenume,CNP, meci ca și atribute în tabel.

Pentru complexul sportiv în care se vor afla sălile pentru meciuri ne interesează să reținem numele și adresa acestuia.

Despre echipele medicale asociate echipelor de jucători vom cunoaște codul echipei medicale, șeful fiecărei echipe medicale și echipa la care este asociată .

Iar pentru managerii fiecărei echipe vom reține codul de manager, numele, prenumele și CNP-ul acestora.

Structura bazei de date poate fi vizualizată prin intermediul următoarei diagrame Entităţi – Asociaţii (DEA):



Pentru proiectarea logică a bazei de date vom folosi Modelul Relaţional al Datelor (MRD). Modelul relațional reprezintă baza de date ca o colecție de relații, fiecăreia fiindu-i asociat un nume unic. O relație poate fi vazută ca un tabel de valori, fiecare linie din tabel reprezentând o colecție de valori interconectate.

Într-o bază de date relaţională, o relaţie este privită ca un tabel bidimensional cu toate valorile atomice. Pentru orice relaţie avem următoarele proprietăţi:

* Numele coloanei unei relaţii se numeşte atribut.
* Numărul de atribute ale unei relaţii reprezintă gradul sau aritatea relaţiei.

linie a tabelului se numeşte tuplu sau înregistrare.

* Numărul tuplurilor dintr-o relaţie reprezintă cardinalul relaţiei.

Pentru a face trecerea de la DEA la MRD trebuie să ţinem cont de următoarele reguli:

* Entitățile (dreptunghi) devin tabele (relații);
* Fiecărei relații i se adaugă cheia surogat de tip autonumber;
* Atributele entităților (conţinutul dreptunghiurilor) devin atribute ale relațiilor respective;
* Funcțiile (segmente/săgeți) devin atribute în relația din care pleacă;
* Pentru fiecare funcție se adaugă o dependență de incluziune (cheie străină) corespunzătoare;
* Asociațiile devin relații, cu câte o cheie străină către relațiile suport.

Pe baza regulilor descrise anterior, entitatea ANTRENORI devine un tabel cu 5 atribute:

* IDAntrenori: cheie primară, de tip AutoNumber, long integer
* Nume: de tip text(20)
* Prenume: de tip text(20)
* CNP: de tip number, integer, unique not null
* Echipa: de tip number

O imagine care conține masă

Descriere generată automat

Entitatea ECHIPE devine de asemenea un tabel cu atributele:

* IDEchipă: cheie primară, de tip AutoNumber, integer
* Nume: de tip text(20)
* NumărEchipă: de tip number, not null
* Oraș: de tip text(20)
* Jucători: de tip number, not null
* Manager: de tip number, not null
* Echipa Medicală: de tip number, not null

O imagine care conține masă

Descriere generată automat

Entitatea JUCĂTORI devine un tabel cu 9 atribute:

* IDJucător: cheie primară, de tip AutoNumber, integer
* Nume: de tip text(20)
* Prenume: de tip text(20)
* CNP: de tip number, integer , unique not null
* Vârstă: de tip number, not null
* Număr: de tip number, unique
* Adresa: de tip number, integer
* Echipă: de tip number, not null
* Meci: de tip number, not null

O imagine care conține masă

Descriere generată automat

Entitatea MECIURI va deveni asemănător un tabel cu 5 atribute:

* IDMeci: cheie primară, de tip AutoNumber, integer
* Data: de tip date, unique
* Ora: de tip time, unique
* Sala: de tip number, integer
* Arbitru: de tip number, integer

O imagine care conține masă

Descriere generată automat

Entitatea SĂLI se transformă într-un tabel cu 4 atribute:

* IDSală: cheie primară, de tip AutoNumber, integer
* NumărSală: de tip number, not null
* Capacitate: de tip number, not null
* Complex Sportiv: de tip number, integer

O imagine care conține masă

Descriere generată automat

Entitatea ADRESE va deveni un tabel cu 4 atribute:

* IDAdresa: cheie primară, de tip AutoNumber, integer
* Oraș: de tip text(20)
* Strada: de tip text(30)
* Număr: de tip number, integer

O imagine care conține masă

Descriere generată automat

Entitatea ARBITRI se va transforma într-un table de forma:

* IDArbitru: cheie primară, de tip AutoNumber, integer
* CodArbitru: de tip number
* Nume: de tip text(20)
* Prenume: de tip text(20)
* CNP: de tip number, integer, unique not null

O imagine care conține masă

Descriere generată automat

Entitatea COMPLEX\_SPORTIV va devein un table cu 3 atribute după cum urmează:

* IDComplexSportiv: cheie primară, de tip AutoNumber, integer
* Nume: de tip text(20)
* Adresa: de tip number, integer

O imagine care conține masă

Descriere generată automat

Entitatea ECHIPĂ\_MEDICALĂ se transformă într-un tabel cu 4 atribute:

* IDEchipăMedicală: cheie primară, de tip AutoNumber, integer
* CodEchipăMedicală: de tip number not null,integer
* Șef: de tip text(20)
* Echipă: de tip number, integer

O imagine care conține masă

Descriere generată automat

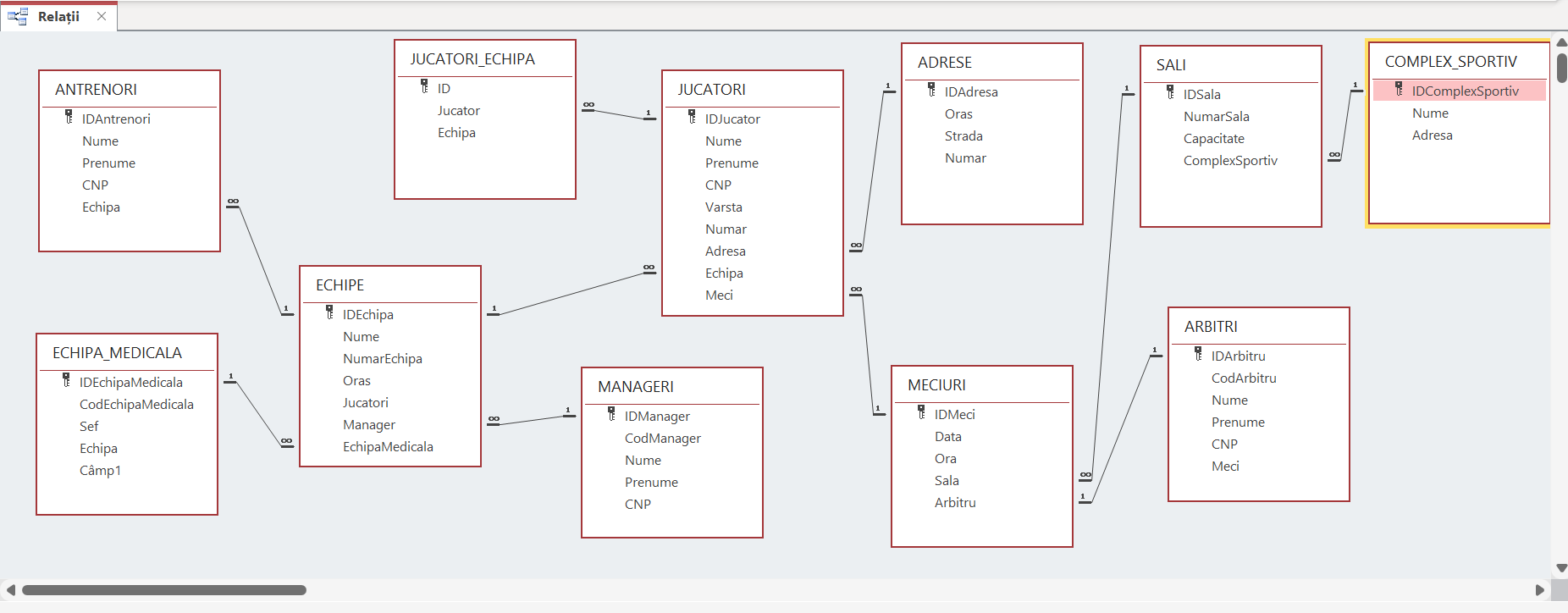
Entitatea MANAGERI devine tabel cu 5 atribute după cum urmează:

* IDManager: cheie primară, de tip AutoNumber, integer
* CodManager: de tip number,integer
* Nume: de tip text(20)
* Prenume: de tip text(20)
* CNP: de tip number, integer, unique not null

O imagine care conține masă

Descriere generată automat

Relaţiile dintre tabele sunt reprezentate după cum se poate observa în imaginea de mai jos:



În continuare, vom prezenta interogările pe care le-am ales să le rezolvăm, pentru a confirma funcţionalitatea bazei de date:

1. EX 1

Să se afișeze mulțimea antrenorilor care antrenează la echipa al cărui număr este mai mare decât 5 și aceștia să fie ordonați după nume:

SELECT a.Nume, a.Prenume

FROM ANTRENORI AS a INNER JOIN ECHIPE AS e ON a.Echipa=e.IDEchipa

WHERE Echipa >5

ORDER BY a.Nume;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 2

Să se afișeze mulțimea echipelor(să fie ordonate după nume) și a jucătorilor care joacă la echipe al cărui număr de jucători este mai mare de 15,numărul afișându-se într-o colonă nouă numită NumărJucători.

SELECT e.Nume, j.Nume, j.Prenume, Jucatori AS NumarJucatori

FROM ECHIPE AS e INNER JOIN JUCATORI AS j ON e.IDEchipa=j.Echipa

WHERE Jucatori >15

ORDER BY e.Nume;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 3

Să se afișeze mulțimea echipelor (să fie ordonate după nume) ai căror jucători au numărul cuprins între 3 si 15.Să se mai afișeze numele și prenumele jucătorilor și numărul acestora ca o coloană nouă denumită NumărTricou.

SELECT e.Nume, j.Nume, j.Prenume, Numar AS NumarTricou

FROM ECHIPE AS e INNER JOIN JUCATORI AS j ON e.IDEchipa=j.Echipa

WHERE Numar BETWEEN 3 AND 15

ORDER BY e.Nume;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 4

Să se afișeze mulțimea managerilor (să fie ordonate după nume) care antrenează la echipele ai căror jucători au vârsta mai mare de 22 de ani. Să se afișeze de asemenea și numele echipelor.

SELECT e.Nume, m.Nume, m.Prenume

FROM MANAGERI AS m INNER JOIN (ECHIPE AS e INNER JOIN JUCATORI AS j ON e.IDEchipa=j.Echipa) ON m.IDManager=e.Manager

WHERE Varsta>22

ORDER BY m.Nume;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 5

Să se afișeze mulțimea arbitrilor care au participat la meciurile ce au avut loc după data de 29.05.2000, cât și meciurile pe care aceștia le-au arbitrat. Rezultatele să fie ordonate descrescător.

SELECT CodArbitru, Nume, Prenume, Meci

FROM ARBITRI AS a INNER JOIN MECIURI AS m ON a.IDArbitru=m.Arbitru

WHERE Data >#29/05/2022#

ORDER BY CodArbitru DESC;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 6

Să se afișeze mulțimea meciurilor care au avut loc în sălile a căror capacitate este cuprinsă între 10000 și 50000. Să se afișeze de asemenea numărul sălii dar și capacitatea acestora, iar rezultatele să fie ordonate descrescător după capacitate.

SELECT m.IDMeci, s.NumarSala, s.Capacitate

FROM MECIURI AS m INNER JOIN SALI AS s ON m.Sala=s.IDSala

WHERE Capacitate BETWEEN 10000 AND 50000

ORDER BY s.Capacitate DESC;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 7

Să se afișeze mulțimea jucătorilor care au meciurile arbitrate de arbitri ai căror nume se termină cu litera "a". Să se afișeze de asemena meciurile la care jucătorii participă ca o coloana denumită MeciJucat și numele + prenumele arbitrilor ca o coloana nouă numită Arbitru. Rezultatele să fie ordonate după numele arbitrului.

SELECT j.Nume, j.Prenume, IDMeci AS MeciJucat, a.Nume + a.Prenume AS Arbitru

FROM JUCATORI AS j INNER JOIN (MECIURI AS m INNER JOIN ARBITRI AS a ON m.Arbitru=a.IDArbitru) ON j.Meci=m.IDMeci

WHERE a.Nume LIKE "\*a"

ORDER BY a.Nume;

Rezultat:

O imagine care conține masă

Descriere generată automat

1. EX 8

Să se afișeze mulțimea meciurilor jucate în sălile din complexele sportive al căror nume se termina cu litera "a". Rezultatele vor fi ordonate după numele complexului sportiv.

SELECT IDMeci AS Meci, NumarSala, ComplexSportiv, c.Nume AS NumeComplexSportiv

FROM MECIURI AS m INNER JOIN (SALI AS s INNER JOIN COMPLEX\_SPORTIV AS c ON s.ComplexSportiv=c.IDComplexSportiv) ON m.Sala=s.IDSala

WHERE c.Nume LIKE "\*a"

ORDER BY c.Nume;

Rezultat:

O imagine care conține text, masă

Descriere generată automat

1. EX 9

Să se afișeze mulțimea șefilor din echipele medicale care lucrează la echipele ai căror jucători au adresa în orașul Cluj-Napoca. Să se afișeze de asemenea Echipa într-o coloana numită EchipăMedicală, numărul echipei la care aceasta lucrează, adresa și orașul jucătorilor.

SELECT em.Sef, em.Echipa AS EchipaMedicala, e.NumarEchipa, a.Numar AS Adresa, a.Oras

FROM ECHIPA\_MEDICALA AS em INNER JOIN ((ECHIPE AS e INNER JOIN JUCATORI AS j ON e.IDEchipa=j.Echipa) INNER JOIN ADRESE AS a ON j.Adresa=a.IDAdresa) ON em.IDEchipaMedicala=e.EchipaMedicala

WHERE Adresa LIKE "2"

ORDER BY em.Sef;

Rezultat:

O imagine care conține text, masă

Descriere generată automat

10.)EX 10

Să se afișeze mulțimea jucătorilor ce au meciurile jucate în sălile din complexele sportive al căror nume se termină cu litera "a" iar adresa acestora este mai mare decât 23. Se va afișa de asemenea meciul, numărul sălii și numele complexului care va apărea într-o coloană nouă numită NumeComplexSportiv, iar rezultatele vor fi ordonate după numele complexului.

SELECT j.Nume, j.Prenume, m.IDMeci AS Meci, s.NumarSala, c.Nume AS NumeComplexSportiv

FROM JUCATORI AS j INNER JOIN ((MECIURI AS m INNER JOIN SALI AS s ON m.Sala=s.IDSala) INNER JOIN COMPLEX\_SPORTIV AS c ON s.ComplexSportiv=c.IDComplexSportiv) ON j.Meci=m.IDMeci

WHERE c.Nume LIKE "\*a" AND c.Adresa >23

ORDER BY c.Nume;

Rezultat:

