PROIECT BD: Aplicatie cu baze de date relationale avand interfata vizuala

**Descrierea cerintelor:** Proiectul consta in crearea unui site impreuna cu baza de date aferenta acestuia pentru evidenta dotarilor existente intr-o facultate. Baza de date include informatii despre dotarile existente in diferite sali dintr-o facultate, departamentele acesteia, precum si studentii ce beneficiaza de ele.

* Dotarile existente intr-o facultate vor fi monitorizate printr-o aplicatie.
* Facultatea este organizata in departamente. Fiecare departament este caracterizat de un nume si un cod de identificare. Fiecare departament este condus de un decan.
* Facultatea are un numar de studenti organizati pe departamente. Fiecare student ii corespunde unui singur departament. Astfel Facultatea este caracterizata de urmatorii parametri: nume Facultate, adresa si numar de telefon.
* Pentru fiecare student se retin urmatoarele date : nume, prenume, CNP, email, sex, data nasterii.
* Facultatea are ca scop educarea si formarea studentilor, iar pentru aceasta este nevoie de anumite dotari existente. Dotarile sunt caracterizate de numele acesteia, categoria folosirii dotarii, precum si un stoc disponibil al acestora.
* Astfel, orice dotare poate fi regasite in salile de curs, definite printr-un numar al salii si capacitatea de persoane ce poate incapea in sala.

Din cele de mai sus, identificam urmatoarele 5 entitati cu care vom lucra: facultate, departamente, dotari, student si sali.

**Pentru urmarirea istoricului se disting urmatoare tabele create in baza de date:**

* Tabela **"Facultate"** cu atributele ID\_Facultate ce reprezinta cheia primara, numele universitatii, adresa unde se afla Facultatea, precum si numarul de telefon al acesteia.
* Tabela **"Departamente"** cu atributele ID\_Departament ce reprezinta cheia primara a acestui tabel, numele departamentului, adica al profilului universitatii, un cod al departamentului, numele decanului si ID\_Facultate, care este cheie straina/externa.
* Tabela **“Dotari”** cu cheia primara ID\_Dotar, numele dotarii, descrierea/rolul in care va fi folosita, precum si stocul disponibil al acestei dotari.
* Tabela **"Sali"** cu cheia primara ID\_Sala, numarul salii si capacitatea acesteia.
* Tabela **“Studenti”** cu cheia primara ID\_Student, numele si prenumele acestuia, CNP, sex si data nasterii, dar si si ID\_Departament, care este cheie straina/externa.
* Tabelul de legatura **“Dotari Sali”** are o cheie primara compusa, formata din copii ale campurilor chei primare din cele doua tabele pe care le leaga: ID\_Sala si ID\_Dotare. Totodata, aceste doua campuri sunt si chei externe, care faciliteaza stabilirea unei relatii intre tabelule parinte (Sala, repsectiv Dotare) si tabelul de legatura.

**Relatii intre tabele:**

* Relatie 1:N intre "Facultate" si "Departamente":

Un departament apartine unei universitati, dar o Facultate poate avea mai multe departamente. Pentru reprezentarea relatiei in baza de date, vom lua o copie a cheii primare din tabelul aflat in partea ”unu”, adica Facultate, si o vom introducem in structura tabelului din partea ”mai multi”, adica Departamente, unde aceasta va deveni cheie externa.

1:N

**Facultate:**

* ID\_Facultate(CP)
* Nume
* Adresa
* Numar telefon

**Departamente:**

* ID\_Departament(CP)
* Nume
* Cod department
* Decan
* ID\_Facultate(CE)
* Relatie 1:N intre "Departamente" si "Studenti":

Un student apartine unui departament, dar un departament poate avea mai multi student. Pentru reprezentarea relatiei in baza de date, vom lua o copie a cheii primare din tabelul aflat in partea ”unu”, adica Departamente, si o vom introducem in structura tabelului din partea ”mai multi”, adica Studenti, unde aceasta va deveni cheie externa.

1:N

**Studenti**

* ID\_Student(CP)
* Nume
* Prenume
* CNP
* Email
* Sex
* Data nasterii
* ID\_Departament(CE)

**Departamente:**

* ID\_Departament(CP)
* Nume
* Cod department
* Decan
* ID\_Facultate(CE)
* Relatie N:N intre "Dotari" si "Sali" prin tabela "Dotari\_Sali":

O sala poate avea mai multe dotari, si o dotare poate fi prezenta in mai multe sali. Pentru a realiza aceasta relatie N:N, am creat tabela de legatura "Dotari\_Sali" care contine chei straine "ID\_sala" si "ID\_dotare" pentru a lega tabelele "Dotari" si "Sali”.

**Dotari Sali:**

* ID\_Sala(CP/CE)
* ID\_Dotare(CP/CE)

1:N

**Sali:**

* ID\_Sala(CP)
* Numar
* Capacitate

**Dotari:**

* ID\_Dotare(CP)
* Nume
* Descriere
* Stoc\_disponibil

1:N

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Facultate | Departamente | Dotari | Studenti | Sali |
| Facultate |  | 1:N |  |  |  |
| Departamente |  |  |  | 1:N |  |
| Dotari |  |  |  |  | 1:N |
| Studenti |  |  |  |  |  |
| Sali |  |  | 1:N |  |  |

Dotari-Sali: 1:N+1:N=N:N;

Structura tabelelor:

**Studenti**

* ID\_Student(CP)
* Nume
* Prenume
* CNP
* Email
* Sex
* Data nasterii
* ID\_Departament(CE)

1:N

1:N

1:N

**Dotari:**

* ID\_Dotare(CP)
* Nume
* Descriere
* Stoc\_disponibil

**Sali:**

* ID\_Sala(CP)
* Numar
* Capacitate

**Departamente:**

* ID\_Departament(CP)
* Nume
* Cod department
* Decan
* ID\_Facultate(CE)

1:N

**Dotari Sali:**

* ID\_Sala(CP/CE)
* ID\_Dotare(CP/CE)

Facultate**:**

* ID\_Facultate(CP)
* Nume
* Adresa
* Numar telefon

**Constragerile de integritate impuse:**

Pentru tabela studenti au fost impuse urmatoarele constrangeri:

* Sexul sa fie de tip “F” pentru fete si “M” pentru baieti;
* Emailul sa fie de tip: [numeprenume@yahoo.com](mailto:numeprenume@yahoo.com);
* CNP ul sa inceapa cu 5 pentru baieti si 6 pentru fete;

Totodata, formatul pentru data a fost facut in asa fel incat sa fie toate identici pentru o comparare mai usoara a acestora.

**Functionare aplicatiei:**

* Am afisat 2 tabele la alegerea, cea Departamente si apoi cea Studenti, pentru fiecare din ele realizand 3 butoane functionale: insert, update si delete. In continuare am afisat si celalalte tabele ale aplicatiei, urmand ca apoi sa afisez 10 intrebari sub forma de statistici, iar pentru a vedea raspunsul la intrebare vom apasa pe un buton si vom vedea rezultatul interogarii.(prima interogare complexa avand camp variabil, putand alege daca afisam toate facultățile care au cel puțin un student de sex feminine/masculine).

**Interogarile simple:**

* 1-Afișează numele și capacitatea sălilor împreună cu denumirea dotărilor asociate:

SELECT S.Numar, S.Capacitate, DS.ID\_Dotare, D.Nume AS Nume\_Dotare

FROM Sali S

JOIN Dotari\_Sali DS ON S.ID\_Sala = DS.ID\_Sala

JOIN Dotari D ON DS.ID\_Dotare = D.ID\_Dotare;

* 2-Afisati numele si prenumele studentilor care apartin "Sisteme Electroenergetice" si sunt nascuti dupa anul 2002(inclusiv):

SELECT S.Nume, S.Prenume

FROM Studenti S

JOIN Departamente D ON S.ID\_Departament = D.ID\_Departament

WHERE D.Nume = 'Sisteme Electroenergetice' AND YEAR(S.Data\_nasterii) >= 2002;

* 3-Afișează sălile dotate cu calculatoare:

SELECT DISTINCT S.\*

FROM Sali S

JOIN Dotari\_Sali DS ON S.ID\_Sala = DS.ID\_Sala

JOIN Dotari D ON DS.ID\_Dotare = D.ID\_Dotare

WHERE D.Nume = 'Calculatoare';

* 4-Afișează departamentul și numărul de studenti pentru fiecare departament, inclusiv cele fără studenți:

SELECT D.Nume AS Nume\_Departament, COUNT(S.ID\_Student) AS Numar\_Studenti

FROM Departamente D

LEFT JOIN Studenti S ON D.ID\_Departament = S.ID\_Departament

GROUP BY D.Nume;

* 5-Sa se gaseasca cei mai mari 5 studenti inscrisi la facultate:

SELECT TOP 5 \*

FROM Studenti

WHERE ID\_Departament IN (SELECT ID\_Departament FROM Departamente)

ORDER BY Data\_nasterii ASC;

* 6-Afișează numele dotărilor împreună cu numărul de sali în care sunt disponibile:

SELECT D.Nume AS Nume\_Dotare, COUNT(DS.ID\_Sala) AS Numar\_Sali

FROM Dotari D

LEFT JOIN Dotari\_Sali DS ON D.ID\_Dotare = DS.ID\_Dotare

GROUP BY D.Nume;

* 7-Afișează numele și prenumele studenților împreună cu numele facultății în care sunt înregistrați:

SELECT S.Nume, S.Prenume, F.Nume AS Nume\_Facultate

FROM Studenti S

JOIN Departamente D ON S.ID\_Departament = D.ID\_Departament

JOIN Facultate F ON D.ID\_Facultate = F.ID\_Facultate;

**Interogarile complexe:**

* 1-Afișează toate facultățile care au cel puțin un student de sex feminine/masculine(camp ce variaza):

SELECT DISTINCT F.Nume AS Nume\_Facultate

FROM Facultate F

JOIN Departamente D ON F.ID\_Facultate = D.ID\_Facultate

WHERE EXISTS (

SELECT 1

FROM Studenti S

WHERE S.ID\_Departament = D.ID\_Departament AND S.Sex = 'F'

);

* 2-Afișati salile care au mai mult de doua dotari si care nu au iesit din stocul facultatii:

SELECT S.ID\_Sala, S.Numar, S.Capacitate

FROM Sali S

JOIN Dotari\_Sali DS ON S.ID\_Sala = DS.ID\_Sala

GROUP BY S.ID\_Sala, S.Numar, S.Capacitate

HAVING COUNT(DS.ID\_Dotare) >=2

AND MAX(DS.Data\_iesire) IS NULL;

* 3-Afișează studenții care au cel puțin un coleg cu aceeași dată de naștere:

SELECT S1.ID\_Student, S1.Nume, S1.Prenume, S1.Data\_nasterii

FROM Studenti S1

WHERE EXISTS (

SELECT 1

FROM Studenti S2

WHERE S2.ID\_Student <> S1.ID\_Student AND S2.Data\_nasterii = S1.Data\_nasterii

);

* --4-Afișează departamentele care au cel puțin doi studenți și au pe aceeasi adresa:

SELECT D.Nume AS Nume\_Departament

FROM Departamente D

WHERE (

SELECT COUNT(\*)

FROM Studenti S

WHERE S.ID\_Departament = D.ID\_Departament

) >= 2

AND D.ID\_Facultate IN (

SELECT F.ID\_Facultate

FROM Facultate F

WHERE F.Adresa IN (

SELECT Adresa

FROM Facultate

GROUP BY Adresa

HAVING COUNT(\*) >= 2

)

);