

PROIECT BAZE DE DATE

CERINȚE PROIECT

Proiectul constă în dezvoltarea unui sistem de management al dotărilor pentru profesori, oferindu-le acestora posibilitatea de a adăuga, vizualiza și gestiona dotările alocate în cadrul instituției. Sistemul include și funcționalități pentru evidențierea dotărilor recent achiziționate de către profesorii conectați.

FUNCȚIONALITĂȚI

Sistemul propus oferă profesorilor o serie de funcționalități esențiale pentru gestionarea eficientă a dotărilor alocate în cadrul instituției. Autentificarea și înregistrarea sunt punctul de plecare, asigurând securitate și acces personalizat pentru fiecare utilizator. Odată conectați, profesorii pot beneficia de următoarele funcționalități:

1. **Adăugarea Dotărilor:** Profesorii au capacitatea de a adăuga noi dotări în sistem. Completarea informațiilor precum categorie, nume, cantitate, stare și locație este esențială pentru o evidență precisă și detaliată.
2. **Vizualizarea Dotărilor Alocate:** Interfața intuitivă permite profesorilor să vizualizeze toate dotările alocate lor, furnizând informații relevante despre fiecare dotare, cum ar fi categoria, numele, cantitatea, starea și locația.
3. **Dotările Recent Achiziționate:** Un modul special permite profesorilor să identifice rapid și eficient dotările achiziționate în ultimele 3 luni. Acest aspect este crucial pentru a ține evidența și a facilita gestionarea dotărilor recent aduse în instituție.
4. **Interogări și Rapoarte Personalizate:** Sistemul oferă profesorilor posibilitatea de a efectua diferite interogări asupra bazelor de date, inclusiv interogări complexe care furnizează informații consolidate despre dotări, precum cantitatea totală de dotări sau dotările cu cea mai mică cantitate.

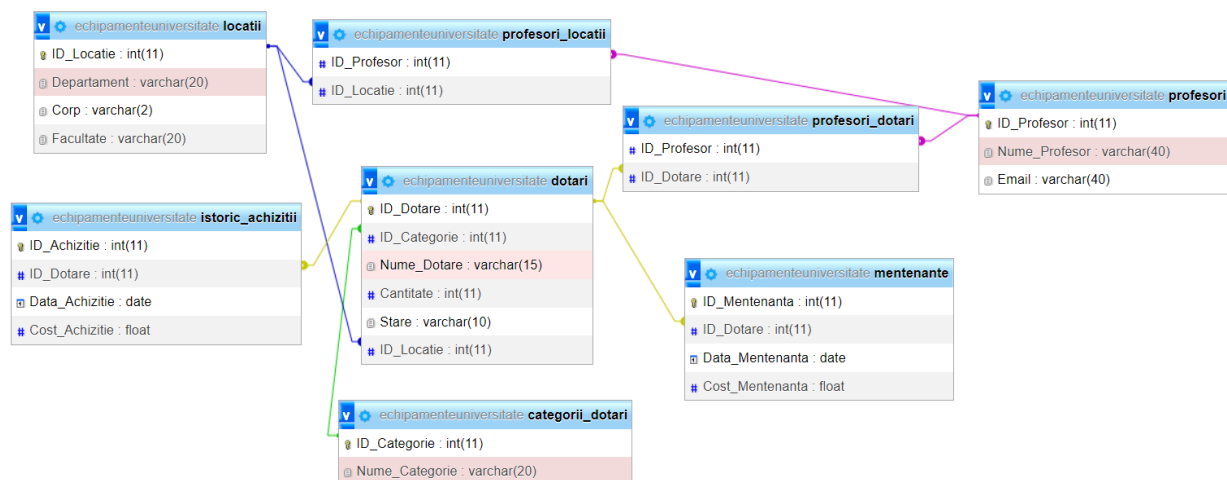
5. **Informații Despre Colegi:** Funcționalitatea integrată permite profesorilor să identifice colegii care activează în aceeași locație, facilitând colaborarea și schimbul de resurse între profesori care lucrează în aceeași locație.
6. **Evidența Achizițiilor Anterioare:** Prin integrarea datelor din istoricul achizițiilor, profesorii pot vizualiza detaliile complete ale achizițiilor anterioare, inclusiv data achiziției. Această funcționalitate consolidează istoricul fiecărei dotări și oferă context important.

ETAPA DE PROIECTARE

Tabelele, Relațiile Dintre Tabele și constrângerile de integritate:

- **Profesori:**
 - ID_Profesor (Primary Key) – int(11)
 - Nume_Profesor – varchar(40)
 - Email – varchar(40)
- **Dotari:**
 - ID_Dotare (Primary Key) – int(11)
 - ID_Categorie (Foreign Key)
 - Nume_Dotare – varchar(15)
 - Cantitate – int(11)
 - Stare – varchar(10)
 - ID_Locatie (Foreign Key) – int(11)
- **Categorii:**
 - ID_Categorie (Primary Key) – int(11)
 - Nume_Categorie – varchar(20)
- **Locatii:**
 - ID_Locatie (Primary Key) – int(11)

- Departament – varchar(20)
- Corp – varchar(2)
- Facultate – varchar(20)
- **Profesori_Locatii:**
 - ID_Profesor (Foreign Key)
 - ID_Locatie (Foreign Key)
- **Profesori_Dotari:**
 - ID_Profesor (Foreign Key)
 - ID_Dotare (Foreign Key)
- **Istoric_Achizitii:**
 - ID_Achizitie (Primary Key) – int(11)
 - ID_Dotare (Foreign Key)
 - Data_Achizitie – date
 - Cost_Achizitie – float
- **Mentenance**
 - ID_Mentenance (Primary Key) – int(11)
 - ID_Dotare (Foreign Key)
 - Data_Mentenance – date
 - Cost_Mentenance – float



INTEROGĂRI

INTEROGĂRI COMPLEXE

```
$querySelectDotari = "SELECT * FROM dotari WHERE ID_Dotare IN
(SELECT ID_Dotare FROM profesori_dotari WHERE ID_Profesor = '$ID_Profesor')";
```

1. Subinterogarea **SELECT ID_Dotare FROM profesori_dotari WHERE ID_Profesor = '\$ID_Profesor'** returnează toate **ID_Dotare**-urile din tabela **profesori_dotari** unde **ID_Profesor** corespunde valorii din variabila **\$ID_Profesor**. Cu alte cuvinte, se obțin toate dotările asociate profesorului specificat.
2. Interogarea principală **SELECT * FROM dotari WHERE ID_Dotare IN (...)** folosește clauza **IN** pentru a selecta toate înregistrările din tabela **dotari** pentru care **ID_Dotare** este prezent în rezultatul subinterogării. Prin urmare, se obțin toate dotările asociate profesorului specificat.

Astfel, în **dotari.php**, această interogare extrage toate dotările care au o asociere cu profesorul identificat de **\$ID_Profesor** din tabela **profesori_dotari**.

```
// alegem dotarea care apartine profesorului si are cea mai mica cantitate
$query = "SELECT *
FROM dotari
WHERE ID_Dotare IN (SELECT ID_Dotare FROM profesori_dotari WHERE ID_Profesor = '$ID_Profesor')
AND Cantitate = (SELECT MIN(Cantitate) FROM dotari);"
```

1. **ID_Dotare în profesori_dotari:** Se asigură că dotarea este asociată profesorului curent, verificând dacă ID-ul dotării există în tabela **profesori_dotari** pentru profesorul cu ID-ul specificat în variabila **\$ID_Profesor**.
2. **Cantitate minimă:** Se asigură că cantitatea dotării este cea mai mică dintre toate dotările. Aceasta este realizată prin intermediul unui subselect (**SELECT MIN(Cantitate) FROM dotari**), care selectează valoarea minimă a coloanei **Cantitate** din tabela **dotari**.

Astfel, interogarea returnează toate dotările care aparțin profesorului curent și au cantitatea minimă între toate dotările disponibile.

```
// alegem dotarea care apartine profesorului si are cea mai mare cantitate
$query = "SELECT *
FROM dotari
WHERE ID_Dotare IN (SELECT ID_Dotare FROM profesori_dotari WHERE ID_Profesor = '$ID_Profesor')
AND Cantitate = (SELECT MAX(Cantitate) FROM dotari);"
```

1. **ID_Dotare în profesori_dotari:** Se asigură că dotarea este asociată profesorului curent, verificând dacă ID-ul dotării există în tabela **profesori_dotari** pentru profesorul cu ID-ul specificat în variabila **\$ID_Profesor**.
2. **Cantitate maximă:** Se asigură că cantitatea dotării este cea mai mare dintre toate dotările. Aceasta este realizată prin intermediul unui subselect (**SELECT MAX(Cantitate) FROM dotari**), care selectează valoarea maximă a coloanei **Cantitate** din tabela **dotari**.

Astfel, interogarea returnează toate dotările care aparțin profesorului curent și au cantitatea maximă între toate dotările disponibile.

```
// selecteaza dotarile de la cea mai recenta pana la cea mai indepartata ca data de achizitie
$query = "SELECT dotari.*,
(SELECT Data_Achizitie
FROM istoric_achizitii
WHERE dotari.ID_Dotare = istoric_achizitii.ID_Dotare) AS Data_Achizitie
FROM dotari
INNER JOIN profesori_dotari ON dotari.ID_Dotare = profesori_dotari.ID_Dotare
WHERE profesori_dotari.ID_Profesor = '$ID_Profesor' ORDER BY Data_Achizitie DESC";
```

1. **Selectare dotări și asocierea datelor de achiziție:** Se utilizează clauza **SELECT dotari.*, (SELECT Data_Achizitie FROM istoric_achizitii WHERE dotari.ID_Dotare = istoric_achizitii.ID_Dotare) AS Data_Achizitie** pentru a selecta toate coloanele din tabela **dotari** și, în plus, se adaugă coloana **Data_Achizitie** care este obținută prin intermediul subselectului (**SELECT Data_Achizitie FROM istoric_achizitii WHERE dotari.ID_Dotare = istoric_achizitii.ID_Dotare**). Acest subselect asigură asocierea datelor de achiziție în funcție de ID-ul dotării.
2. **Condiția de filtrare pe profesor:** **INNER JOIN profesori_dotari ON dotari.ID_Dotare = profesori_dotari.ID_Dotare** asigură că sunt selectate doar acele dotări care sunt asociate profesorului curent, iar **WHERE profesori_dotari.ID_Profesor = '\$ID_Profesor'** filtrează rezultatele pentru profesorul specificat.
3. **Sortare după Data_Achizitie:** **ORDER BY Data_Achizitie DESC** sortează rezultatele în ordine descrescătoare a datelor de achiziție, astfel încât cele mai recente să fie afișate primele.

Această interogare selectează informații despre dotări, împreună cu data de achiziție asociată, pentru dotările care aparțin profesorului curent (**\$ID_Profesor**).

INTEROGĂRI SIMPLE

```
$queryTotalCantitate = "SELECT SUM(Cantitate) AS TotalCantitate
FROM dotari
INNER JOIN profesori_dotari ON dotari.ID_Dotare = profesori_dotari.ID_Dotare
WHERE profesori_dotari.ID_Profesor = '$ID_Profesor'";
```

1. **SUM(Cantitate) AS TotalCantitate:** Această parte a interogării folosește funcția **SUM** pentru a aduna valorile din coloana **Cantitate**. Rezultatul este returnat ca o singură valoare sub numele de alias **TotalCantitate**.
2. **FROM dotari:** Se specifică tabela **dotari** ca sursă pentru date.
3. **INNER JOIN profesori_dotari ON dotari.ID_Dotare = profesori_dotari.ID_Dotare:** Se realizează o joncțiune internă între tabela **dotari** și **profesori_dotari** pe baza condiției că **ID_Dotare** este egal în ambele tabele. Această joncțiune asigură că se iau în considerare doar înregistrările care au o corespondență în ambele tabele.

4. **WHERE profesori_dotari.ID_Profesor = '\$ID_Profesor'**: Se aplică o condiție pentru a filtra rezultatele astfel încât să includă doar înregistrările pentru care **ID_Profesor** din tabela **profesori_dotari** este egal cu valoarea specificată în variabila **\$ID_Profesor**. Aceasta limitează calculul sumei la dotările asociate profesorului specificat.

Prin urmare, această interogare returnează o singură valoare, care reprezintă suma cantităților de dotări asociate profesorului identificat de **\$ID_Profesor**.

```
$queryColegi = "SELECT profesori.ID_Profesor, profesori.Nume_Profesor, profesori.Email
FROM profesori
INNER JOIN profesori_locatii ON profesori.ID_Profesor = profesori_locatii.ID_Profesor
WHERE profesori_locatii.ID_Locatie = '$idLocatieCurenta'
AND profesori.ID_Profesor != '$ID_Profesor'";
```

1. **SELECT profesori.ID_Profesor, profesori.Nume_Profesor, profesori.Email**: Această parte a interogării specifică coloanele pe care dorim să le selectăm din tabela **profesori**. Se selectează **ID_Profesor**, **Nume_Profesor** și **Email** pentru a obține informații despre colegi.
2. **FROM profesori**: Se specifică sursa de date, în acest caz, tabela **profesori**.
3. **INNER JOIN profesori_locatii ON profesori.ID_Profesor = profesori_locatii.ID_Profesor**: Se realizează o joncțiune internă între tabela **profesori** și **profesori_locatii** pe baza condiției că **ID_Profesor** este egal în ambele tabele. Acest lucru asigură că se iau în considerare doar înregistrările care au o corespondență în ambele tabele.
4. **WHERE profesori_locatii.ID_Locatie = '\$idLocatieCurenta'**: Se aplică o condiție pentru a filtra rezultatele astfel încât să includă doar colegii profesorului cu locația specificată de **\$idLocatieCurenta**. Aceasta limitează rezultatele la profesorii care împart aceeași locație cu profesorul curent.
5. **AND profesori.ID_Profesor != '\$ID_Profesor'**: Se adaugă o condiție suplimentară pentru a exclude profesorul curent din rezultate. Astfel, se evită ca profesorul să apară în lista colegilor săi.