Cristian Bob

333AC

Evidenta inchiriere sali conferinta la un Hotel

Cu istoric

Cuprins

[Descrierea cerintelor 2](#_Toc503140689)

[Proiectarea bazei de date 2](#_Toc503140690)

[Functionarea aplicatiei 6](#_Toc503140691)

[Interogari complexe 10](#_Toc503140692)

[Concluzii 11](#_Toc503140693)

[Anexa (Diagrama baza de date) 12](#_Toc503140694)

# Descrierea cerintelor

Aceasta aplicatie trebuie sa rezolve problemele intalnite la inchirierea unei sali de conferinta a unui hotel. Astfel, utilizand aceasta aplicatie, operatorul poate incheia procesul de inchiriere a unei sali de conferinta in doar cateva minute si, totodata, poate mentine in istoric date depre clienti si date despre conferintele ce au avut loc. Operatorul poate adauga/sterge un client, poate adauga/sterge un angajat, poate adauga/sterge o conferinta sau poate adauga/sterge o facilitate oferita de hotelul respectiv.

# Proiectarea bazei de date

Baza de date creata de mine contine urmatoarele tabele:

* *Conferinte* – acest tabel retine datele despre conferintele ce au avut loc sau despre cele care vor urma;
* *Sala* – acest tabel contine datele despre salile de conferinte oferite de hotel;
* *Client* – acest tabel retine datele clientilor care au rezervat o sala de conferinte a hotelului pentru a tine o conferinta;
* *Angajat* – acest tabel contine date despre angajatii hotelului. Unii angajati se pot ocupa de intretinerea salilor de conferinte;
* *Facilitati* – acest tabel contine date despre facilitatile puse la dispozitie de catre hotel.

Relatiile dintre aceste tabele sunt:

|  |  |
| --- | --- |
| Tabele | Relatie |
| Conferinte – Sala | 1:1 |
| Conferinte – Client | N:1 |
| Sala - Angajat | N:1 |
| Conferinte – Facilitati | N:N |

Deoarece relatia dintre Conferinte si Facilitati este de tip many to many, am introdus un tabel de legatura: *FacilitatiConferinte*.

Dupa ce am terminat de identificat tabelele necesare si relatiile dintre acestea, am trecut la stabilirea campurilor fiecarui tabel.

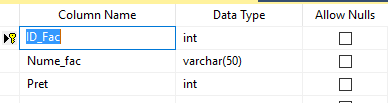


Figura 1: Tabel Facilitati

* ID\_Fac: reprezinta ID-ul facilitatii si este de tip *int* deoarece este coloana identitate. Acest camp va fi folosit in tabelele ulterioare ca si **FOREIGN KEY**;
* Nume\_fac: reprezinta numele facilitatii;
* Pret: reprezinta pretul facilitatii.

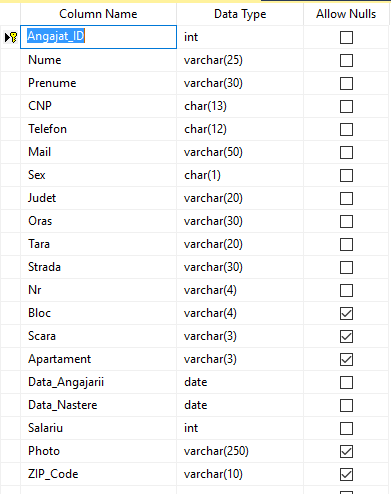


Figura 2: Tabel Angajat

* Angajat\_ID: aceasta coloana reprezinta ID-ul angajatului si este de tip *int* deoarece este o coloana identitate a acestui tabel. De asemenea acest camp este **PRIMARY KEY** pentru acest tabel;
* Nume: aceasta coloana reprezita numele angajatului;
* Prenume: aceasta coloana reprezinta prenumele angajatului;
* CNP: aceasta coloana reprezinta Codul Numeric Personal al angajatului si sufera o constrangere de tip **UNIQUE**;
* Telefon: acest camp contine numerele de telefon ale angajatilor;
* Mail: acest camp contine adresele de E-Mail ale angajatilor si are o constrangere de tip **UNIQUE**;
* Sex: acest camp contine sexul fiecarui angajat si are o constrangere de tip **CHECK**;
* Judet,Oras,Tara,Strada,Nr,Bloc,Scara,Apartament: aceste campuri retin adresa de domiciliu a fiecarui angajat. Campurile Bloc, Scara, Apartament pot avea valori nule deoarece exista oameni care stau la casa.
* ZIP\_Code: acest camp reprezinta codul postal al adresei de domiciliu. Acest camp poate avea valori nule deoarece nu reprezinta o informatie care este trecuta pe CI;
* Photo: acest camp retine locatiile din coputer unde sunt stocate fotografii ale angajatiilor hotelului;
* Data\_Angajarii: acest camp retine data la care angajatul respectiv a fost introdus in baza de date;
* Data\_Nastere: acest camp retine data la care a fost nascut angajatul respectiv;
* Salariu: acest camp retine salariul angajatului respectiv.

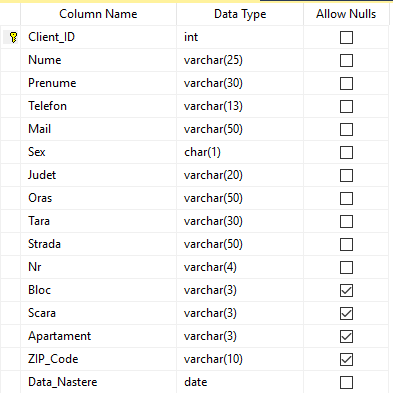


Figura 3: Tabel Client

Acest tabel este asemanator celui cu angajati doar ca lipsesc campurile: CNP, Data\_Angajarii, si Salariu.

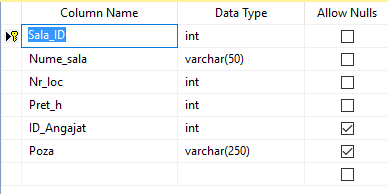


Figura 4: Tabel Sala

* Sala\_ID: reprezinta campul care retine ID-urile salilor de conferinta si are o constrangere de tip **PRIMARY KEY**;
* Nume\_sala: reprezinta campul care retine numele salilor de conferinta si are o constrangere de tip **UNIQUE**;
* Nr\_loc: acest camp contine informatii despre numarul de locuri al fiecarei Sali de conferinta;
* Pret\_h: reprezinta pretul fiecarei sali de conferinta;
* ID\_Angajat: reprezinta ID-ul angajatului care se ocupa de intretinerea salii respective. Acest camp are o constrangere de tip **FOREIGN KEY**. Acest camp poate contine valori nule deoarece daca un angjat este concediat sa nu se stearga si sala de care acesta avea grija din baza de date;
* Poza: acest camp contine calea catre poza care descrie sala de conferinte.

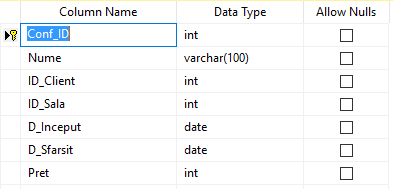


Figura 5: Tabel Conferinte

* Conf\_ID: acest camp reprezinta ID-ul fiecarei conferinte si are o constrangere de tip **PRIMARY KEY**. Aceasta coloana este de tip *int* deoarece este coloana identitate pentru acest tabel;
* Nume: acest camp reprezinta numele fieacarei conferinte si are o constrangere de tip **UNIQUE**;
* ID\_Client: acest camp face referire la ID-ul clientului care a inchiriat sala de conferinte si are o constrangere de tip **FOREIGN KEY**;
* ID\_Sala: acest camp face referire la ID-ul salii in care se va desfasura conferinta si are o constrangere de tip **FOREIGN KEY;**
* D\_Inceput: reprezinta data la care incepe conferinta;
* D\_Sfarsit: reprezinta data la care se termina conferinta;
* Pret: reprezinta pretul cumulat din facilitatile cumparate si inchirierea salii.

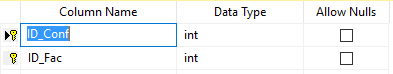


Figura 6: Tabel FacilitatiConferinte

Acest tabel contine doua campuri: ID\_Conf si ID\_Fac; primul camp face referire la ID-ul unei conferinte iar cel de-al doilea face referiere la ID-ul unei facilitati solicitate. Ambele campuri, impreuna, formeaza cheiea primara a acestui tabel, iar, separat, fiecare camp are o constrangere de tip **FOREIGN KEY**.

# Functionarea aplicatiei

Pagina principala a aplicatiei este prezentata in figura 7. La aceasta pagina principala se poate ajunge oricand apasand in coltul din stanga sus pe logo-ul ***HB***. Aceasta pagina ofera utilizatorului posibilitatea de a vizualiza mai multe grafice cum ar fi:

* Cele mai vandute facilitati;
* Cele mai cautate sali;
* Clientii disponibili ofertelor(cei care platesc pe toate conferintele sustinute un pret mai mare decat media unei conferinte);
* Cei mai fideli clienti;
* Conferintele ce se desfasoara in cea mai scumpa sala;
* Raport salariu angajat si pretul salii pe care o intretine;
* Datele despre angajatii care lucreaza in sali cu profit mai mare decat 1000.

De asemenea, utilizatorul poate observa o tranzitie a salilor pe care hotelul le pune la dispozitie in partea dreapta a ecranului. In partea stanga a ecranului se observa mai multe butoane avand nume scrise pe ele. Odata apasate, se afiseaza continutul tabelului respectiv in partea centrala a ferestrei. Sub aceste butoane, in partea de stanga jos, se gasesc butoanele de *ADD* si *DELETE*.

In partea de sus se gaseste bara de cautare, butonul de inchidere si butonul de setari care trimite utilizatorul la configurarea credentialelor serverului SQL.

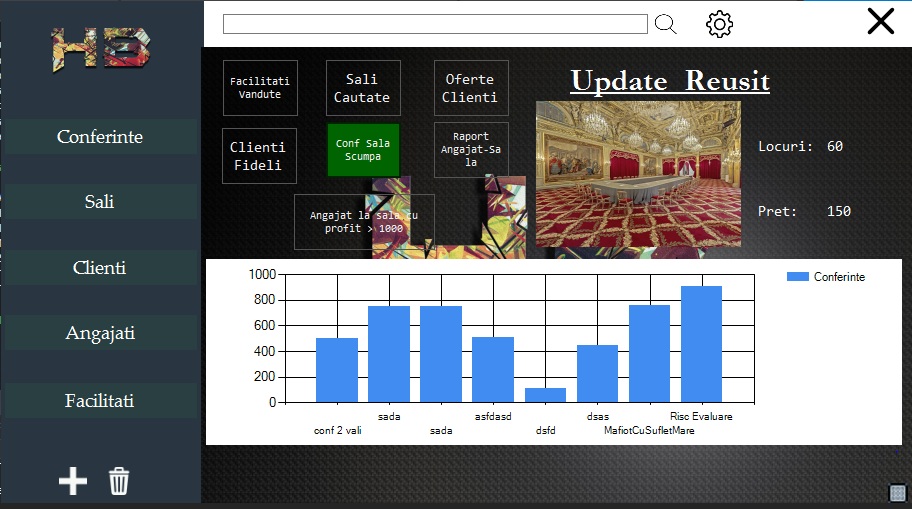


Figura 7: Pagina principala

Voi prezenta in continuare o demonstratie de adaugare si stergere a unei facilitati, restul adaugarilor/stergerilor facandu-se la fel cu cateva exceptii:

* Clientii pot fi adaugati doar daca sustin o conferinta (figura );
* Salile nu pot fi adaugate sau sterse, ele pot fi doar actualizate;
* Orice modificare in tabele poate modifica alte tabele asa ca pentru a nu se sterge lucruri nedorite, trebuie sa se verifice inainte.

Pentru a adauga/sterge o facilitate, din pagina principala apasam butonul pe care scrie *Facilitati* care ne trimite la vizualizarea datelor din tabelul Facilitati ca in figura 8. Dupa acest pas, adaugarea unei facilitati se face prin apasarea butonului *ADD*(plus) din coltul stanga jos. Aceasta operatie ne va trimite in pagina prezentata in figura 9. Completand datele necesare, observam ca dupa ce am parasit campul *Nume* acesta se coloreaza in rosu sau verde in functie de existenta produsului in baza de date sau nu(figura 10). Dupa ce ne-am asigurat ca totul este in regula, apasam butonul *Save* pentru a salva datele in baza de date.

Operatia de stergere se efectueaza selectand o facilitate(click odata pe facilitate; figura 11) si apoi apansand butonul de *DELETE*(cosul de gunoi).

Actualizarea datelor se face apasand de doua ori pe o facilitate, operatie care deschide aceeasi fereastra ca in figura 9 cu datele deja completate.

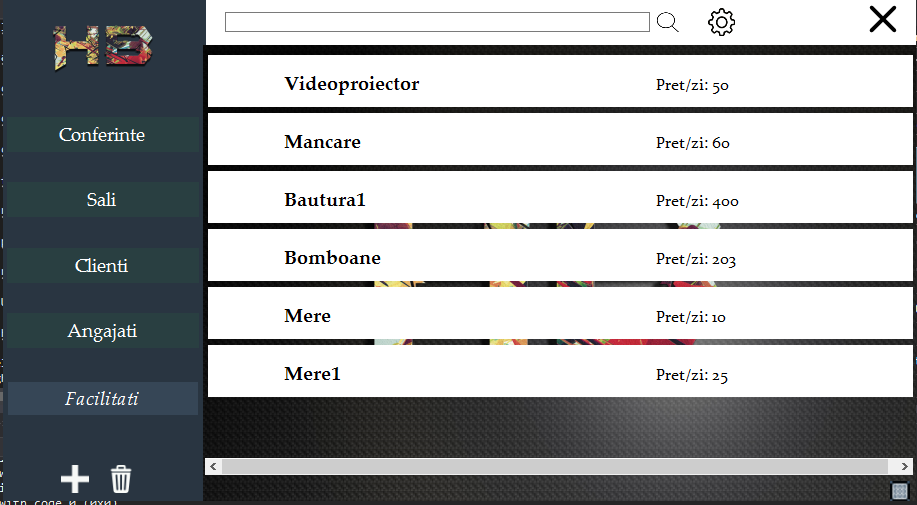


Figura 8: Afisare Facilitati

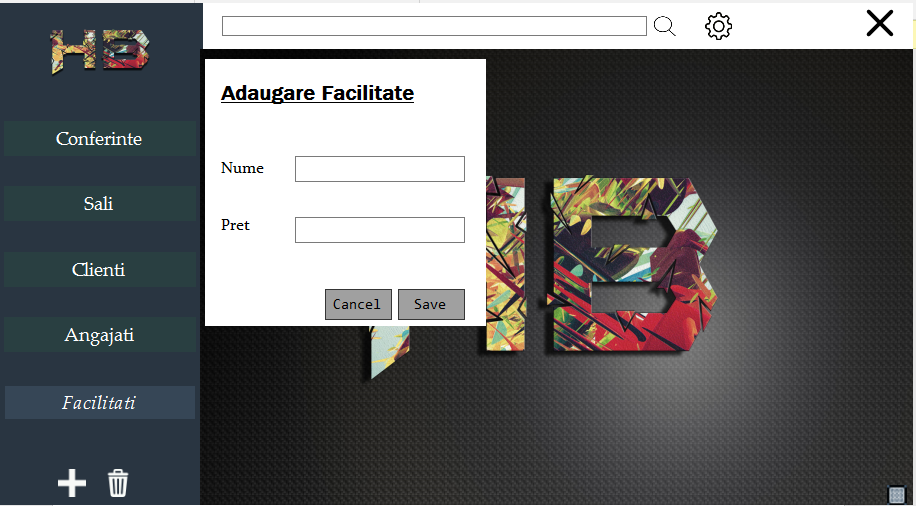


Figura 9: Adaugare Facilitate

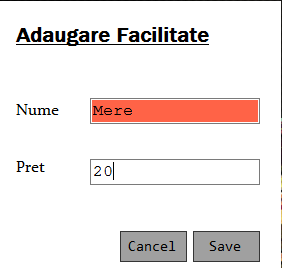
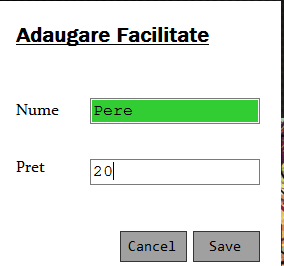


Figura 10: Existenta produs in baza de date

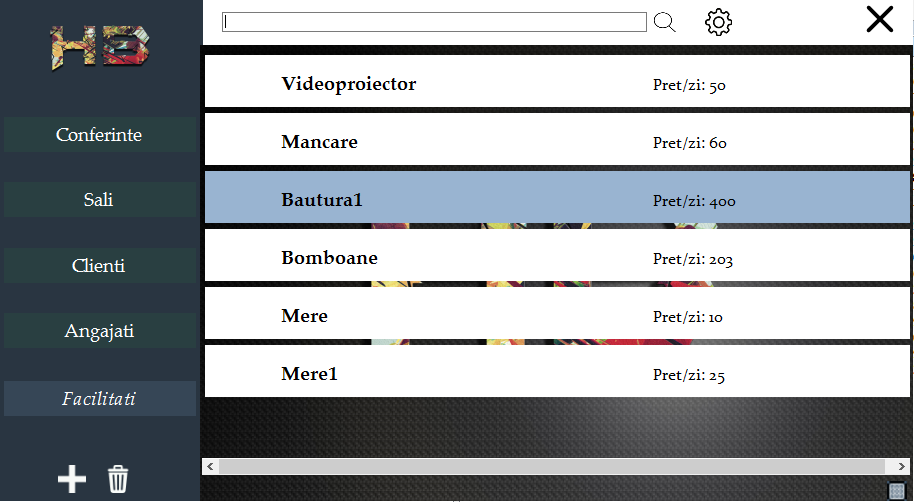


Figura 11: Selectie Facilitate

Restul de adaugari se fac in mod asemanator, doar ca adaugarea unei conferinte poate fi precedata de adaugarea unui client daca acesta nu se gaseste in baza de date. La apasarea butonului *ADD* pentru adaugarea unei conferinte, apare pe ecran verificarea E-Mail-ului clientului pentru a vedea daca acesta se gaseste sau nu in baza de date(figura 12).

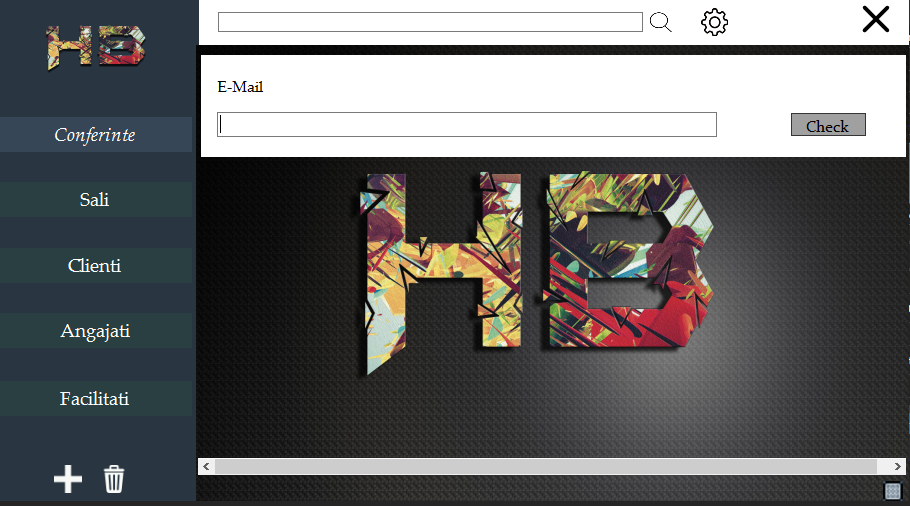


Figura 12: Verificare Client

Daca E-Mail-ul introdus apare in baza de date atunci suntem transferati direct la adaugarea unei conferinte, altfel suntem transferati mai intai la adaugarea clientului cu E-Mail-ul respectiv.

# Interogari complexe

Iterogarile complexe utilizate de mine au fost:

* Select C.Nume,(SELECT CL.Nume + ' ' + CL.Prenume FROM Client CL WHERE CL.Client\_ID = C.ID\_Client) as Nume\_Client, (SELECT S.Nume\_sala FROM Sala S WHERE S.Sala\_ID = C.ID\_Sala),C.D\_Inceput,C.D\_Sfarsit,C.Pret,C.Conf\_ID,(SELECT S.Poza FROM Sala S WHERE S.Sala\_ID = C.ID\_Sala) FROM Conferinte C - Afisare conferinte;
* SELECT S.Nume\_sala, S.Nr\_loc, (SELECT A.Nume+' '+A.Prenume FROM Angajat A WHERE A.Angajat\_ID=S.ID\_Angajat), S.Pret\_h,S.Poza,S.Sala\_ID FROM Sala S – Afisare sala
* SELECT C.Nume,C.Pret FROM Conferinte C WHERE(C.ID\_Sala = (SELECT TOP(1) S.Sala\_ID FROM SALA S ORDER BY S.Pret\_h DESC));
* SELECT A.Nume+' '+A.Prenume, A.Telefon, A.Mail, A.Sex, A.Angajat\_ID FROM Angajat A WHERE(A.Angajat\_ID IN (SELECT S.ID\_Angajat FROM SALA S WHERE (S.Sala\_ID IN (SELECT C.ID\_Sala FROM Conferinte C GROUP BY C.ID\_Sala HAVING SUM(C.Pret)>1000 ))));

# Concluzii

Acest proiect m-a ajutat foarte mult in dezvoltarea aptitudiilor de a lucra cu o baza de date, in planificarea timpului si rezolvarea unor task-uri pana la un deadline impus de mine precum si la managementul unui proiect cu un numar foarte mare de linii de cod.

De asemenea, acest proiect mi-a placut deoarece a fost solicitant din punct de vedere al design-ului dar si din punct de vedere al implementarii.

# Anexa (Diagrama baza de date)

