

SISTEM DE REZERVARE BILETE LA FILM

Acest sistem de rezervare este dedicat clientilor. Acestia pot, prin intermediul lui, sa isi rezerve bilete la orice film, in orice cinematograful doresc ei. Fiecare client detine un card prin intermediul caruia poate rezerva, la mai multe cinematografe, bilete la diferite filme care ruleaza in diferite sali, la diferite ore. Pe card se afla toate detalii despre: valabilitatea cardului, cate puncte a acumulat clientul. Rezervarile cuprind numarul de bilete si pretul acestora. Fiecare client poate sa faca una sau mai multe rezervari. O rezervare a unui client contine informatii despre data pentru care s-a facut rezervarea, cate bilete si pretul fiecarui bilet. In plus, o rezervare contine detalii si despre filmele la care s-a facut rezervare. Un film depinde de program(cand ruleaza) si sala (unde ruleaza).

Tabela Users:

Fiecare client se va autentifica la baza de date. Daca nu au cont atunci trebuie sa isi creeze unul. Un cont cuprinde un user NOT NULL si unic (Primary key) (username nu poate fi identic cu un alt username deja existent in baza de date) si un password (care poate sa fie acelasi cu al altui user). Pentru a avea acces la contul sau trebuie ca user + password sa se gaseasca in baza de date in aceasta combinatie.

Tabela Clienti :

Fiecare client cand se inregistreaza pentru prima data in baza de date primeste un ID_client unic dupa care baza de date va face diferenta intre utilizatori (Primary key). Clientii sunt obligati sa completeze la prima inregistrare campurile cu datele personale: campuri obligatorii (NOT NULL) Nume, Prenume si campuri optionale, care pot fi null, email, telefon, data_nasterii.

Tabela Carduri :

Intre Clienti si Carduri exista o relatie 1:1 (fiecare client poate detine un singur card la un moment dat). Cardurile sunt unice si sunt identificate printr-un ID_card (Primary key).

Cardul este caracterizat de: campuri obligatorii(NOT NULL) : Tip_card, Nr_puncte care se afla pe card si valabilitate (cat timp este valabil cardul). ID_client este FK(legatura intre client si carduri – se afla in carduri deoarece trebuie mai intai sa existe un client pentru a adauga un card).

Tabela Reduceri:

Fiecare reducere este identificata printr-un ID_reducere (Primary key) pentru a face diferenta intre reduceri. Reducerile sunt caracterizate de tip_reducere, perioada de valabilitate (cate zile), procent_reducere si valabila (CHECK – o reducere este valabila sau nu, in mod implicit este valabila). Toate campurile sunt NOT NULL deoarece acestea, impreuna, descriu daca o reducere se poate aplica sau a expirat.

Tabela Carduri_Reduceri:

Aceasta face legatura intre tabelele Carduri si Reduceri. Intre aceste tabele exista o relatie n:n care a fost separata in doua relatii 1:n. Pe un card pot exista una sau mai multe reduceri, iar o reducere poate exista pe unul sau mai multe carduri. Tabela Carduri_Reduceri are o Primary key compusa.

Tabela Program:

Tabela Program contine toate informatiile legate de inceperea(ora_inceput,minut_inceput), terminarea(ora_final,minut_final), data si durata unui film. Toate aceste detalii trebuie stiute astfel ca toate aceste campuri trebuie declarate NOT NULL pentru a se sti cand se poate rula un film si cand poate rula altul. Fiecare program este identificat prin intermediul ID_program(Primary key).

Tabela Filme:

Fiecare film are propriul sau ID_film, un identificator unic (Primary key). Filmul este caracterizat de Nume_film, Data_lansare, Cod_film si Tip_film (CHECK – exista trei tipuri – Normal, Premiera si Avandpremiera). In mod implicit Tip_film este Normal deoarece sunt putine filme care se incadreaza in celelalte doua categorii.

Intre Filme si Program exista o relatie 1:n. FK(ID_program) se afla in Filme deoarece intr-un anumit interval orar pot rula, simultan, mai multe filme.

Tabela Sala:

Fiecare sala are propriul ei ID_sala unic(Primary key). Sala este caracterizata de numar_sala, Loc_rand, Loc_scaun si cinematograf.

Tabela Filme_Sala:

Aceasta face legatura intre tabelele Filme si Sala. Intre aceste tabele exista o relatie n:n care a fost separata in doua 1:n. Un film poate rula in una sau mai multe sali, iar intr-o sala pot rula unu sau mai multe filme. Tabela Filme_Sala are o Primary key compusa.

Tabela Rezervari:

Fiecare client poate face una sau mai multe rezervari, dar o rezervare este asociata cu un singur client, de aceea in tabela rezervari am FK(ID_client) – relatie 1:n. Fiecare rezervare este identificata printr-un ID_rezervare (Primary key). In cadrul unei rezervari apare si ziua(data NOT NULL) cand sunt valabile biletele.

Tabela Filme_Rezervari:

Aceasta face legatura intre tabelele Filme si Rezervari. Intre aceste tabele exista o relatie n:n care a fost separata in doua 1:n. La un film se poate face una sau mai multe rezervari, iar pe o rezervare se pot face rezervari la unul sau mai multe filme. Tabela Filme_Reduceri are o Primary key compusa.

Tabela Preturi:

Fiecare pret este identificat prin Tip_pret (Primary key). Campul pret contine valoarea unui bilet.

Tabela Bilete:

Fiecare bilet este identificat prin ID_bilet(Primary key). Fiecare bilet este caracterizat de Tip_bilet(CHECK – exista trei tipuri – Adult , Student si Pensionar). Intre Bilete si Preturi exista o relatie 1:n prin intermediul campului Tip_pret(FK) (un bilet are un singur pret, dar acelasi pret poate fi asociat cu unul sau mai multe bilete). Intre Bilete si Rezervari exista o relatie 1:n prin intermediul campului ID_rezervare(FK) (un anume bilet poate sa apara doar pe o rezervare, dar o rezervare poate sa contina unul sau mai multe bilete).

Constrangeri:

Clienti:

ID_client NOT NULL PRIMARY KEY
Nume NOT NULL
Prenume NOT NULL

Carduri:

ID_card NOT NULL PRIMARY KEY
Tip_card NOT NULL CHECK(TIP_CARD = 'VIP' OR TIP_CARD = 'Student' OR TIP_CARD = 'Basic' OR TIP_CARD = 'Promotional' OR TIP_CARD = 'Pensionar') DEFAULT 'Basic'
Nr_puncte NOT NULL
Valabilitate NOT NULL
ID_client UNIQUE FK

Reduceri:

ID_reducere NOT NULL PRIMARY KEY
Tip_reducere NOT NULL
Perioada NOT NULL
Procent_reducere NOT NULL
Valabila NOT NULL CHECK(VALABILA = 'N' OR VALABILA = 'Y') DEFAULT 'Y'

Carduri_Reduceri:

ID_card FK
ID_reducere FK
PRIMARY KEY:ID_card, ID_reducere

Program:

ID_program NOT NULL PRIMARY KEY
Durata NOT NULL
ora_inceput NOT NULL
minut_inceput NOT NULL
ora_final NOT NULL
minut_final NOT NULL
Data_calendaristica NOT NULL

Filme:

ID_film NOT NULL PRIMARY KEY
Nume_film NOT NULL
Tip_film NOT NULL CHECK(TIP_FILM = 'Normal' OR TIP_FILM = 'Premiera' OR TIP_FILM = 'Avanpremiera')

DEFAULT 'Normal'

Data_lansare NOT NULL
Cod_film NOT NULL UNIQUE
ID_program FK

Sala:

ID_sala NOT NULL PRIMARY KEY
Numar NOT NULL
Randuri NOT NULL
Locuri NOT NULL
Cinematograf NOT NULL

ID_film FK

Filme_Sala:

ID_film FK

ID_sala FK

PRIMARY KEY:ID_film, ID_sala

Rezervari:

ID_rezervare NOT NULL PRIMARY KEY

ID_client FK

Data NOT NULL

Filme_Rezervari

ID_film FK

ID_rezervare FK

PRIMARY KEY:ID_film, ID_rezervare

Preturi:

Tip_pret NOT NULL PRIMARY KEY

CHECK([Tip_pret]='Vineri' OR [Tip_pret]='ZileLucratoare' OR [Tip_pret]='Weekend' OR [Tip_pret]='Matineu') DEFAULT 'Weekend'

Pret NOT NULL

Bilete:

ID_bilet NOT NULL PRIMARY KEY

Tip_pret FK

ID_rezervare FK

Tip_bilet NOT NULL CHECK(TIP_BILET ='Adult' OR TIP_BILET ='Student' OR TIP_BILET ='Pensionar')

DEFAULT 'Adult'

Cod_bilet NOT NULL

Relatii:

Clienti – Carduri : 1:1

Carduri-Carduri_Reduceri: 1:n

Carduri_Reduceri-Reducer: 1:n

Clienti –Rezervari : 1:n

Rezervari – Bilete : 1:n

Bilete – Preturi : n:1

Rezervari-Filme_Rezervari: 1:n

Filme_Rezervari-Filme: n:1

Filme-Filme_Sala: 1:n

Filme_Sala-Sala: n:1

Filme –Program : n:1

Tiputi de date:

Clienti:

ID_client INT
Nume NVARCHAR(50)
Prenume NVARCHAR(50)
email NVARCHAR(50)
telefon CHAR(20)
data_nasterii DATETIME

Carduri:

ID_card INT
Tip_card NVARCHAR(50)
Nr_puncte INT
Valabilitate DATETIME
ID_client INT

Reduceri:

ID_reducere INT
Tip_reducere NVARCHAR(50)
Perioada INT
Procent_reducere INT
Valabila CHAR(1)

Carduri_Reduceri:

ID_card INT
ID_reducere INT

Program:

ID_program INT
Durata INT
ora_inceput INT
minut_inceput INT
ora_final INT
minut_final INT
Data DATETIME

Filme:

ID_film INT
Nume_film NVARCHAR(50)
Tip_film NVARCHAR(50)
Data_lansare DATETIME
Cod_film INT

Sala:

ID_sala INT
Numar INT
Randuri INT
Locuri INT
Cinematograf NVARCHAR(50)
ID_film INT

Filme_Sala:

```

        ID_film INT
        ID_sala INT
Rezervari:
        ID_rezervare INT
        ID_client INT
        Data DATETIME
Filme_Rezervari:
        ID_film INT
        ID_rezervare INT
Preturi:
        Tip_pret CHAR(30)
        Pret INT
Bilete:
        ID_bilet INT
        Tip_pret CHAR(30) R
        ID_rezervare INT
        Tip_bilet CHAR(20)
        Cod_bilet INT

```

APLICATIA

Fiecare client trebuie sa se inregistreze cu propriul sau cont. Pentru clientii care nu detin un cont acestia iai pot crea un cont nou. Aplicatia implementeaza trei operatii : delete,update si insert. Toate aceste operatii se pot face in cadrul a trei tabele : Users (pot sa se sterga numele si parola pe care o au, sa isi schimbe numele si parola in unele noi), tabelul Reduceri si Clienti(sa insereze datele proprii, sa modifice datele personale sau sa stearga inregistrarea).Clientii isi pot crea un cont nou, sa se logheze si ca operatii premise insert si update la tabelul client.Restul functionalitatilor sunt efectuate doar de angajati.

INTEROGARI SIMPLE

1.Numarul de carduri pe care il detine clientul cu ID-ul introdus(Nume + Prenume + numarul de carduri detinute)

```

SELECT C.Nume,C.Prenume,count(C.ID_client) as Numar_carduri_detinute
FROM Clienti C inner join Carduri Ca on (C.ID_client = Ca.ID_client)
WHERE C.Nume = @Nume and C.Prenume = @Prenume
GROUP BY C.ID_client,C.Nume,C.Prenume;

```

2.Ce reduceri de afla pe cardul cu seria introdusa

```

select R.Tip_reducere
from Carduri_Reduceri Cr inner join Reduceri R on (Cr.ID_reducere = R.ID_reducere )
WHERE Cr.ID_card = (ce introduce in textbox)
group by Cr.ID_card,R.Tip_reducere

```

3.Programul filmului introdus(dupa numele filmului)

```

select F.Nume_film , P.Durata , P.Ora_inceput ,P.Minut_inceput, P.Ora_final,P.Minut_final
WHERE F.Nume_film = (ce introduc in textbox)
from Filme F inner join Program P on ( F.ID_program = P.ID_program )

```

4.Ce filme ruleaza in fiecare sala

```
select S.Numar ,S.Cinematograf ,F.Nume_film
from Filme_Rezervari Fr inner join Filme F on (Fr.ID_film = F.ID_film )
inner join Sala S on (Fr.ID_sala = S.ID_sala)
```

5. Pentru clientul cu ID-ul introdus sa se afiseze rezervariile si pentru fiecare rezervare numarul de bilete

```
select C.Nume ,C.Prenume ,R.ID_rezervare,count(B.ID_rezervare ) as numar_bilete
from Clienti C inner join Rezervari R on (C.ID_client = R.ID_client )
inner join Bilete B on (R.ID_rezervare = B.ID_rezervare )
where C.Nume = 'Ionescu' and C.Prenume = 'Maria'
group by B.ID_rezervare ,C.Nume ,C.Prenume ,R.ID_rezervare
```

6. Suma ce trebuie platita de fiecare client, pentru fiecare rezervare

```
select C.Nume ,C.Prenume ,R.ID_rezervare ,sum(P.pret)
from Clienti C inner join Rezervari R on (C.ID_client = R.ID_client )
inner join Bilete B on (B.ID_rezervare = R.ID_rezervare )
inner join Preturi P on (P.Tip_pret = B.Tip_pret)
where C.Nume = @Nume and C.Prenume = @Prenume
group by C.ID_client ,C.Nume ,C.Prenume ,R.ID_rezervare
```

INTEROGARI COMPLEXE

1. Clientii care au carduri

```
select C.Nume, C.Prenume
from Clienti C
where C.ID_client IN (
    select Ca.ID_client
    from Carduri Ca
);
```

2. Clientul cu cea mai mare suma de plata per rezervare

```
select C.Nume ,C.Prenume ,R.ID_rezervare ,sum(P.pret) as sum
from Clienti C inner join Rezervari R on (C.ID_client = R.ID_client )
inner join Bilete B on (B.ID_rezervare = R.ID_rezervare )
inner join Preturi P on (P.Tip_pret = B.Tip_pret)
group by C.ID_client ,C.Nume ,C.Prenume ,R.ID_rezervare
having sum(P.pret) > ALL (
    select sum(P1.pret)
    from Clienti C1 inner join Rezervari R1 on (C1.ID_client = R1.ID_client )
    inner join Bilete B1 on (B1.ID_rezervare =
R1.ID_rezervare )
    inner join Preturi P1 on (P1.Tip_pret = B1.Tip_pret)
    where C1.ID_client != C.ID_client
    group by R1.ID_rezervare
);
```

3. Clientii care si-au facut rezervari

```
select C.Nume , C.Prenume
from Clienti C
where C.ID_client in (
    select R.ID_client
    from Rezervari R
);
```

4. Filmul la care s-au facut cele mai multe rezervari

```
select F.Nume_film,count(F.ID_film) as Nr_rezervari
from Filme F inner join Filme_Rezervari Fr on (F.ID_film = Fr.ID_film )
group by F.ID_film,F.Nume_film
having count(F.ID_film) >= ALL (
```



```
select count(Fr1.ID_film)
from Filme_Rezervari Fr1
group by Fr1.ID_film
);
```