



# Proiect – Baze de date

Aplicatie pentru rezervarea biletelor la film

**Student:** Buric Andrei-Alexandru

**Grupa:** 333AA

**An univesitar:** 2021-2022

## 1. Descrierea proiectului

Scopul proiectului este de a proiecta o baza de date pentru organizarea unui cinema, mai precis, a rezervarilor biletelor din cadrul unui cinema. Astfel, aceasta aplicatie ofera clientilor date precise despre fiecare cinematograful, despre fiecare program al cinematografului si odata rezervat un loc la un film, acesta primeste un bilet foarte detaliat astfel incat clientul sa fie multumit de serviciile cinematografului.

## 2. Tehnologii folosite

Microsoft SQL Server 2014

## 3. Arhitectura Bazei de Date

Baza de date are un model relational si contine urmatoarele tabele si campuri :

**Bilet film**(BiletFilmID, CinemaID, FilmID, LocRezervatID, ClientID, OrarCinematografID, PretBilet)

**Client**(ClientID, Nume, Prenume, Varsta, Telefon, AdresaMail, CNP, Strada, Numar, Oras, Judet, Sex, DataNasterii)

**Cinema**(CinemaID, Locatie, Adresa, Telefon, Oras, Judet)

**Film**(FilmID, CategorieID, DenumireFilm, DurataFilm, LimbaDifuzare, DataAparitie, Descriere)

**Categorie Film**(CategorieID, GenulFilm)

**FilmCategorie**(FilmID, CategorieID)

**LocRezervat**(LocRezervatID, SalaID, ClientID, Rand, Loc)

**Sala**(SalaID, Randuri, Locuri, TipSala)

**Orar cinematograf**(OrarCinematografID, SalaID, FilmID, CinemaID, Ora)

	Bilet film	Client	Film	Categorie Film	FilmCategorie	LocRezervat	Sala	Orar Cinematograf	Cinema
Bilet film		1:1	1:1			1:1		1:1	1:1
Client	1:N								
Film	1:N			N:N	1:N			1:N	
Categorie Film			N:N		1:N				
FilmCategorie			1:1	1:1					
LocRezervat	1:1						1:1		
Sala						1:N		1:N	
Orar Cinematograf	1:N		1:1				1:1		1:1
Cinema	1:N							1:N	

În cadrul acestui proiect, am luat în vedere cum ar putea arăta un bilet de film. Astfel, am realizat că pe bilet trebuie să avem Filmul la care clientul a rezervat loc, atunci am realizat legătura între tabela Bilet Film și tabela Film. Între aceste două tabele se stabilește o relație de 1:1(one-to-one) deoarece un Bilet Film poate avea doar un film pe acesta însă un film poate fi pe mai multe bilete, deci între tabela Film și tabela Bilet Film se stabilește relația de 1:N(one-to-many). Rămânând la tabela de filme, un Film poate să aibă mai multe Categorie, iar o Categorie poate să cuprindă mai multe filme. Astfel, am realizat un tabel de legătură, FilmCategorie, unde Film și Categorie Film au relația de 1:N (one-to-many) pentru tabela FilmCategorie, deci putem realiza relația de N:N(many-to-many) între Film și Categorie Film și invers.

Întorcându-ne la Bilet film, pe un Bilet Film există doar un Client, deci între tabela Bilet Film și Client există relația de 1:1(one-to-one). Totuși, Clientul poate să fie pe mai multe Bilete Film, prin urmare între tabela Client și Bilet Film există relația 1:N(one-to-many).

Tot la Bilet Film, pe un Bilet Film există doar un Cinema, deci între Bilet Film și Cinema există relația de 1:1(one-to-one). Cinema-ul poate, însă, să apară pe mai multe Bilete Film, prin urmare între tabela Cinema și Bilet Film se realizează relația 1:N(one-to-many). Desigur, pe Bilet Film trebuie să existe și un loc rezervat, astfel, am creat tabelul LocRezervat în care am trecut Sala, Clientul care a rezervat locul, randul și locul.

Pe un Bilet Film exista doar un loc rezervat, astfel intre tablea Bilet Film si LocRezervat am creat relatia de 1:1(one-to-one). Exact acelasi lucru este si invers pentru ca un loc rezervat are doar un singur bilet, deci si intre tabela LocRezervat si Film exista relatia de 1:1(one-to-one).

Pe tabela LocRezervat regasim SalaID, ceea ce inseamna ca trebuie sa ne creem o tabela SalaID unde putem regasi mai multe detalii despre aceasta sala. Prin urmare am creeat tabela Sala, iar relatia dintre tabela LocRezervat si Sala este de 1:1(one-to-one) deoarece pe un LocRezervat regasim doar o Sala. Totusi, intr-o sala regasim mai multe locuri rezervate, deci intre tabela Sala si LocRezervat regasim relatia de 1:N(one-to-many).

Intorcandu-ne la Bilet Film, regasim variabila OrarCinematografID care reprezinta ocheie externa care ne ofera detalii despre programul filmului care se regaseste pe bilet. Astfel am creat tabela Orar Cinematograf. Relatia care se realizeaza intre Bilet Film si Orar Cinematograf este 1:1(one-to-one) deoarece pe Bilet Film exista doar un orar al cinematografului. Totusi, un Orar Cinematograf poate contine mai multe Bilete Film, deci relatia dintre tabela Orar Cinematograf si Bilete Film este de 1:N(one-to-many).

In tabela Orar Cinematograf regasim si SalaID, FilmID, CinemaID. Aceste lucru demonstreaza ca trebuie sa mai facem niste legaturi. Prin urmare, Orar Cinematograf contine o singura Sala, un singur film si un singur Cinema, de aici rezulta ca relatia dintre tabela Orar Cinematograf si tabela Sala, tabela Cinema si tabela Film este de 1:1(one-to-one). Desigur ca tabela Sala, Film si Cinema au mai multe orare, de aici rezulta ca relatia dintre tablea Sala, Film si cinema este de 1:N(one-to-many) cu tabela Orar Cinematograf.

