Розділимо таблиці на проекти-мікросервіси на основі їхньої функціональності:

Проект-мікросервіс No1:

Movies

Actors

Categories

Movies\_Actors

Movies\_Categories

Цей проект-мікросервіс відповідає за управління фільмами, акторами та категоріями. Включає операції додавання, редагування та отримання інформації про фільми, акторів та категорії.

Проект-мікросервіс No2:

Users

Comments

Ratings

Цей проект-мікросервіс відповідає за управління користувачами, коментарями та оцінками. Забезпечує можливість реєстрації та авторизації користувачів, додавання та отримання коментарів, а також ставлення оцінок фільмам.

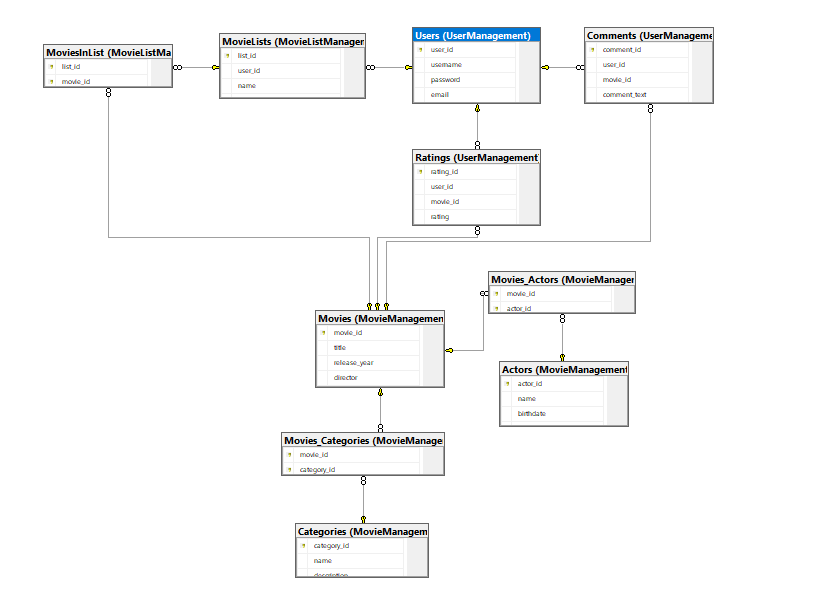
Проект-мікросервіс No3:

MovieLists

MoviesInList

Цей проект-мікросервіс відповідає за управління списками фільмів користувачів. Забезпечує можливість створення списків фільмів, додавання та видалення фільмів до/зі списків.

Кожен проект-мікросервіс може мати свою власну базу даних, яка містить лише необхідні таблиці для його функціональності. Використовуючи мікросервісну архітектуру, кожен проект-мікросервіс може бути незалежним ізолованим компонентом, що спрощує розробку, масштабування та підтримку системи в цілому. UserManagementContext



CREATE DATABASE MovieCatalogDB;

USE MovieCatalogDB;

CREATE SCHEMA MovieManagement;

go

CREATE TABLE MovieManagement.Movies (

movie\_id INT IDENTITY(1,1) PRIMARY KEY ,

title VARCHAR(255) NOT NULL,

release\_year INT NOT NULL,

director VARCHAR(255) NOT NULL,

description TEXT NOT NULL

);

CREATE TABLE MovieManagement.Actors (

actor\_id INT IDENTITY(1,1) PRIMARY KEY,

name VARCHAR(255) NOT NULL,

birthdate DATE NOT NULL,

nationality VARCHAR(255) NOT NULL

);go

CREATE TABLE MovieManagement.Categories (

category\_id INT IDENTITY(1,1) PRIMARY KEY ,

name VARCHAR(255) NOT NULL,

description TEXT NOT NULL

);

CREATE TABLE MovieManagement.Movies\_Actors (

movie\_id INT,

actor\_id INT ,

PRIMARY KEY (movie\_id, actor\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id),

FOREIGN KEY (actor\_id) REFERENCES MovieManagement.Actors(actor\_id)

);

CREATE TABLE MovieManagement.Movies\_Categories (

movie\_id INT,

category\_id INT,

PRIMARY KEY (movie\_id, category\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id),

FOREIGN KEY (category\_id) REFERENCES MovieManagement.Categories(category\_id)

);

go

CREATE SCHEMA UserManagement;

CREATE TABLE UserManagement.Users (

user\_id INT IDENTITY(1,1) PRIMARY KEY ,

username VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL,

registration\_date DATE NOT NULL

);

CREATE TABLE UserManagement.Comments (

comment\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

movie\_id INT NOT NULL,

comment\_text TEXT NOT NULL,

comment\_date DATE NOT NULL,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

CREATE TABLE UserManagement.Ratings (

rating\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

movie\_id INT NOT NULL,

rating DECIMAL(3, 1) NOT NULL,

rating\_date DATE NOT NULL,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

CREATE SCHEMA MovieListManagement;

CREATE TABLE MovieListManagement.MovieLists (

list\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

name VARCHAR(255) NOT NULL,

creation\_date DATE NOT NULL,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id)

);

CREATE TABLE MovieListManagement.MoviesInList (

list\_id INT,

movie\_id INT,

PRIMARY KEY (list\_id, movie\_id),

FOREIGN KEY (list\_id) REFERENCES MovieListManagement.MovieLists(list\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

Створення бд   
CREATE DATABASE MovieCatalogDB;

go

-- Використання бази даних

USE MovieCatalogDB;

go

CREATE SCHEMA MovieManagement;

CREATE TABLE MovieManagement.Movies (

movie\_id INT IDENTITY(1,1) PRIMARY KEY,

title VARCHAR(255),

release\_year INT,

director VARCHAR(255),

description TEXT

);

CREATE TABLE MovieManagement.Actors (

actor\_id INT IDENTITY(1,1) PRIMARY KEY,

name VARCHAR(255),

birthdate DATE,

nationality VARCHAR(255)

);

CREATE TABLE MovieManagement.Categories (

category\_id INT IDENTITY(1,1) PRIMARY KEY,

name VARCHAR(255),

description TEXT

);

CREATE TABLE MovieManagement.Movies\_Actors (

movie\_id INT,

actor\_id INT,

PRIMARY KEY (movie\_id, actor\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id),

FOREIGN KEY (actor\_id) REFERENCES MovieManagement.Actors(actor\_id)

);

CREATE TABLE MovieManagement.Movies\_Categories (

movie\_id INT,

category\_id INT,

PRIMARY KEY (movie\_id, category\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id),

FOREIGN KEY (category\_id) REFERENCES MovieManagement.Categories(category\_id)

);

CREATE SCHEMA UserManagement;

CREATE TABLE UserManagement.Users (

user\_id INT IDENTITY(1,1) PRIMARY KEY,

username VARCHAR(255),

password VARCHAR(255),

email VARCHAR(255),

registration\_date DATE

);

CREATE TABLE UserManagement.Comments (

comment\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT,

movie\_id INT,

comment\_text TEXT,

comment\_date DATE,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

CREATE TABLE UserManagement.Ratings (

rating\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT,

movie\_id INT,

rating DECIMAL(3, 1),

rating\_date DATE,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

CREATE SCHEMA MovieListManagement;

CREATE TABLE MovieListManagement.MovieLists (

list\_id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT,

name VARCHAR(255),

creation\_date DATE,

FOREIGN KEY (user\_id) REFERENCES UserManagement.Users(user\_id)

);

CREATE TABLE MovieListManagement.MoviesInList (

list\_id INT,

movie\_id INT,

PRIMARY KEY (list\_id, movie\_id),

FOREIGN KEY (list\_id) REFERENCES MovieListManagement.MovieLists(list\_id),

FOREIGN KEY (movie\_id) REFERENCES MovieManagement.Movies(movie\_id)

);

Вставка в схему MovieManegement

INSERT INTO MovieManagement.Movies (title, release\_year, director, description)

VALUES

('The Matrix', 1999, 'Lana Wachowski, Lilly Wachowski', 'A computer hacker learns from mysterious rebels about the true nature of his reality and his role in the war against its controllers.'),

('The Shawshank Redemption', 1994, 'Frank Darabont', 'Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.'),

('Pulp Fiction', 1994, 'Quentin Tarantino', 'The lives of two mob hitmen, a boxer, a gangster and his wife, and a pair of diner bandits intertwine in four tales of violence and redemption.'),

('The Dark Knight', 2008, 'Christopher Nolan', 'When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.'),

('Inception', 2010, 'Christopher Nolan', 'A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a C.E.O.'),

('Fight Club', 1999, 'David Fincher', 'An insomniac office worker and a devil-may-care soapmaker form an underground fight club that evolves into something much, much more.'),

('Forrest Gump', 1994, 'Robert Zemeckis', 'The presidencies of Kennedy and Johnson, the events of Vietnam, Watergate, and other historical events unfold through the perspective of an Alabama man with an IQ of 75.'),

('The Lord of the Rings: The Return of the King', 2003, 'Peter Jackson', 'Gandalf and Aragorn lead the World of Men against Sauron''s army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.'),

('Interstellar', 2014, 'Christopher Nolan', 'A team of explorers travel through a wormhole in space in an attempt to ensure humanity''s survival.'),

('The Godfather', 1972, 'Francis Ford Coppola', 'The aging patriarch of an organized crime dynasty transfers control of his clandestine empire to his reluctant son.'),

('The Avengers', 2012, 'Joss Whedon', 'Earth''s mightiest heroes must come together and learn to fight as a team if they are going to stop the mischievous Loki and his alien army from enslaving humanity.'),

('The Social Network', 2010, 'David Fincher', 'Harvard student Mark Zuckerberg creates the social networking site that would become known as Facebook.'),

('Inglourious Basterds', 2009, 'Quentin Tarantino', 'In Nazi-occupied France during World War II, a plan to assassinate Nazi leaders by a group of Jewish U.S. soldiers coincides with a theatre owner''s vengeful plans for the same.'),

('Gladiator', 2000, 'Ridley Scott', 'A former Roman General sets out to exact vengeance against the corrupt emperor who murdered his family and sent him into slavery.'),

('The Departed', 2006, 'Martin Scorsese', 'An undercover cop and a mole in the police attempt to identify each other while infiltrating an Irish gang in South Boston.'),

('The Silence of the Lambs', 1991, 'Jonathan Demme', 'A young F.B.I. cadet must receive the help of an incarcerated and manipulative cannibal killer to help catch another serial killer, a madman who skins his victims.');

select \* from MovieManagement.Actors

INSERT INTO MovieManagement.Actors (name, birthdate, nationality)

VALUES

('Keanu Reeves', '1964-09-02', 'Canadian'),

('Tim Robbins', '1958-10-16', 'American'),

('John Travolta', '1954-02-18', 'American'),

('Christian Bale', '1974-01-30', 'British'),

('Leonardo DiCaprio', '1974-11-11', 'American'),

('Edward Norton', '1969-08-18', 'American'),

('Tom Hanks', '1956-07-09', 'American'),

('Viggo Mortensen', '1958-10-20', 'Danish'),

('Matthew McConaughey', '1969-11-04', 'American'),

('Marlon Brando', '1924-04-03', 'American'),

('Robert Downey Jr.', '1965-04-04', 'American'),

('Jesse Eisenberg', '1983-10-05', 'American'),

('Brad Pitt', '1963-12-18', 'American'),

('Russell Crowe', '1964-04-07', 'New Zealand'),

('Matt Damon', '1970-10-08', 'American'),

('Anthony Hopkins', '1937-12-31', 'Welsh');

select \* from MovieManagement.Movies\_Actors

DELETE FROM MovieManagement.Movies

WHERE movie\_id BETWEEN 17 AND 32;

select \* from MovieManagement.Movies\_Actors

INSERT INTO MovieManagement.Categories (name, description)

VALUES

('Action', 'Movies that involve exciting and fast-paced sequences.'),

('Drama', 'Movies that focus on emotional development and intense character interactions.'),

('Crime', 'Movies that revolve around criminal activities and their consequences.'),

('Sci-Fi', 'Movies that explore scientific and technological concepts.'),

('Thriller', 'Movies that create suspense and excitement.'),

('Adventure', 'Movies that feature exciting and unusual journeys or experiences.'),

('Biography', 'Movies that depict the life stories of real people.'),

('Fantasy', 'Movies that involve magical or supernatural elements.'),

('Comedy', 'Movies that aim to entertain and make the audience laugh.'),

('Mystery', 'Movies that involve solving a puzzle or a crime.'),

('Romance', 'Movies that focus on love and romantic relationships.'),

('Horror', 'Movies that aim to frighten or scare the audience.'),

('Animation', 'Movies that use animation techniques to create moving images.'),

('History', 'Movies that depict historical events and periods.'),

('War', 'Movies that depict warfare and its impact on people and societies.'),

('Music', 'Movies that revolve around music and musicians.');

INSERT INTO MovieManagement.Movies\_Actors (movie\_id, actor\_id)

VALUES

INSERT INTO MovieManagement.Movies\_Actors (movie\_id, actor\_id)

VALUES

(1, 1),

(1, 6),

(2, 2),

(2, 7),

(3, 3),

(3, 11),

(4, 5),

(4, 12),

(5, 6),

(5, 13),

(6, 6),

(6, 14),

(7, 7),

(7, 15),

(8, 8),

(8, 13),

(9, 5),

(9, 16),

(10, 10),

(10, 13),

(11, 11),

(12, 9),

(12, 13),

(13, 4),

(13, 12),

(14, 11),

(14, 10),

(15, 4),

(15, 3);

go

INSERT INTO MovieManagement.Movies\_Categories (movie\_id, category\_id)

VALUES

(1, 1),

(1, 6),

(2, 2),

(2, 7),

(3, 3),

(3, 8),

(4, 1),

(4, 5),

(5, 4),

(5, 8),

(6, 1),

(6, 5),

(7, 2),

(7, 6),

(8, 1),

(8, 9),

(9, 4),

(9, 10),

(10, 1),

(10, 11),

(11, 2),

(11, 12),

(12, 1),

(12, 13),

(13, 3),

(13, 11),

(14, 3),

(14, 14),

(15, 2),

(15, 15);

-- Вставка в таблицю Users

INSERT INTO UserManagement.Users (username, password, email, registration\_date)

VALUES

('user1', 'password1', 'user1@example.com', '2023-05-01'),

('user2', 'password2', 'user2@example.com', '2023-05-02'),

('user3', 'password3', 'user3@example.com', '2023-05-03'),

('user4', 'password4', 'user4@example.com', '2023-05-04'),

('user5', 'password5', 'user5@example.com', '2023-05-05'),

('user6', 'password6', 'user6@example.com', '2023-05-06'),

('user7', 'password7', 'user7@example.com', '2023-05-07'),

('user8', 'password8', 'user8@example.com', '2023-05-08'),

('user9', 'password9', 'user9@example.com', '2023-05-09'),

('user10', 'password10', 'user10@example.com', '2023-05-10')-- Додайте більше записів тут...

go

select \* from UserManagement.Ratings

-- Вставка в таблицю Comments

INSERT INTO UserManagement.Comments (user\_id, movie\_id, comment\_text, comment\_date)

VALUES

(1, 4, 'Interesting plot and good acting.', '2023-05-11'),

(2, 5, 'Awesome visual effects!', '2023-05-12'),

(3, 6, 'Great music in the film.', '2023-05-13'),

(4, 7, 'Emotional drama with strong characters.', '2023-05-14'),

(5, 8, 'Interesting science fiction setting.', '2023-05-15'),

(6, 9, 'An intriguing plot with an unexpected ending.', '2023-05-16'),

(7, 10, 'A well-acted comedy with funny situations.', '2023-05-17'),

(8, 11, 'Interesting characters and unexpected events.', '2023-05-18'),

(9, 12, 'Deep philosophical message.', '2023-05-19'),

(10, 13, 'Energetic action with big battle scenes.', '2023-05-20')

(1, 1, 'This movie is very interesting!', '2023-05-01'),

(2, 2, 'I liked the plot.', '2023-05-02'),

(3, 1, 'One of the best movies I heve seen!', '2023-05-03'),

(4, 3, 'Great acting by the actors.', '2023-05-04'),

(5, 2, 'Impressive special effects!', '2023-05-05'),

(6, 3, 'The plot captivates from the first minutes.', '2023-05-06'),

(7, 4, 'Interesting dramatic story.', '2023-05-07'),

(8, 1, 'Unforgettable emotions from watching.', '2023-05-08'),

(9, 2, 'Excellent artistic direction.', '2023-05-09'),

(10, 3, 'Strong main characters.', '2023-05-10')

;

-- Додайте більше записів тут...

go

-- Вставка в таблицю Ratings

INSERT INTO UserManagement.Ratings (user\_id, movie\_id, rating, rating\_date)

VALUES

(1, 1, 4.5, '2023-05-01'),

(2, 2, 3.8, '2023-05-02'),

(3, 1, 5.0, '2023-05-03'),

(4, 3, 4.2, '2023-05-04'),

(5, 2, 3.5, '2023-05-05'),

(6, 3, 4.7, '2023-05-06'),

(7, 4, 3.9, '2023-05-07'),

(8, 1, 4.8, '2023-05-08'),

(9, 2, 3.6, '2023-05-09'),

(10, 3, 4.4, '2023-05-10'),

(1, 4, 4.5, '2023-05-11'),

(2, 5, 4.8, '2023-05-12'),

(3, 6, 3.9, '2023-05-13'),

(4, 7, 4.2, '2023-05-14'),

(5, 8, 4.6, '2023-05-15'),

(6, 9, 4.3, '2023-05-16'),

(7, 10, 3.8, '2023-05-17'),

(8, 11, 4.1, '2023-05-18'),

(9, 12, 4.7, '2023-05-19'),

(10, 13, 4.4, '2023-05-20');

DELETE FROM UserManagement.Ratings

WHERE rating\_id BETWEEN 11 AND 20;

-- Додайте більше записів тут...

-- Продовжуйте вставляти записи у таблиці Users, Comments та Ratings...

-- Вставка в таблицю MovieLists з повторенням поля name

INSERT INTO MovieListManagement.MovieLists (user\_id, name, creation\_date)

VALUES

(1, 'Favourite Movies', '2023-05-01'),

(2, 'Action Movies', '2023-05-02'),

(3, 'Romantic Movies', '2023-05-03'),

(4, 'Sci-Fi Movies', '2023-05-04'),

(5, 'Comedy Movies', '2023-05-05'),

(6, 'Drama Movies', '2023-05-06'),

(7, 'Thriller Movies', '2023-05-07'),

(8, 'Fantasy Movies', '2023-05-08'),

(9, 'Animated Movies', '2023-05-09'),

(10, 'Classic Movies', '2023-05-10'),

(1, 'Favourite Movies', '2023-05-11'),

(2, 'Action Movies', '2023-05-12'),

(3, 'Romantic Movies', '2023-05-13'),

(4, 'Sci-Fi Movies', '2023-05-14'),

(5, 'Comedy Movies', '2023-05-15'),

(6, 'Drama Movies', '2023-05-16'),

(7, 'Thriller Movies', '2023-05-17'),

(8, 'Fantasy Movies', '2023-05-18'),

(9, 'Animated Movies', '2023-05-19'),

(10, 'Classic Movies', '2023-05-20'),

(1, 'Favourite Movies', '2023-05-21'),

(2, 'Action Movies', '2023-05-22'),

(3, 'Romantic Movies', '2023-05-23'),

(4, 'Sci-Fi Movies', '2023-05-24'),

(5, 'Comedy Movies', '2023-05-25'),

(6, 'Drama Movies', '2023-05-26'),

(7, 'Thriller Movies', '2023-05-27'),

(8, 'Fantasy Movies', '2023-05-28'),

(9, 'Animated Movies', '2023-05-29'),

(10, 'Classic Movies', '2023-05-30');

go

INSERT INTO MovieListManagement.MoviesInList (list\_id, movie\_id)

VALUES

(1, 1),

(1, 2),

(1, 3),

(2, 4),

(2, 5),

(2, 6),

(3, 7),

(3, 8),

(3, 9),

(4, 10),

(4, 11),

(4, 12),

(5, 13),

(5, 14),

(5, 15),

(6, 1),

(6, 2),

(6, 3),

(7, 4),

(7, 5),

(7, 6),

(8, 7),

(8, 8),

(8, 9),

(9, 10),

(9, 11),

(9, 12),

(10, 13),

(10, 14),

(10, 15);

go

-- Вставка в таблицю MovieLists з повторенням поля name

INSERT INTO MovieListManagement.MovieLists (user\_id, name, creation\_date)

VALUES

(1, 'Favourite Movies', '2023-05-01'),

(2, 'Action Movies', '2023-05-02'),

(3, 'Romantic Movies', '2023-05-03'),

(4, 'Sci-Fi Movies', '2023-05-04'),

(5, 'Comedy Movies', '2023-05-05'),

(6, 'Drama Movies', '2023-05-06'),

(7, 'Thriller Movies', '2023-05-07'),

(8, 'Fantasy Movies', '2023-05-08'),

(9, 'Animated Movies', '2023-05-09'),

(10, 'Classic Movies', '2023-05-10'),

(1, 'Favourite Movies', '2023-05-11'),

(2, 'Action Movies', '2023-05-12'),

(3, 'Romantic Movies', '2023-05-13'),

(4, 'Sci-Fi Movies', '2023-05-14'),

(5, 'Comedy Movies', '2023-05-15'),

(6, 'Drama Movies', '2023-05-16'),

(7, 'Thriller Movies', '2023-05-17'),

(8, 'Fantasy Movies', '2023-05-18'),

(9, 'Animated Movies', '2023-05-19'),

(10, 'Classic Movies', '2023-05-20'),

(1, 'Favourite Movies', '2023-05-21'),

(2, 'Action Movies', '2023-05-22'),

(3, 'Romantic Movies', '2023-05-23'),

(4, 'Sci-Fi Movies', '2023-05-24'),

(5, 'Comedy Movies', '2023-05-25'),

(6, 'Drama Movies', '2023-05-26'),

(7, 'Thriller Movies', '2023-05-27'),

(8, 'Fantasy Movies', '2023-05-28'),

(9, 'Animated Movies', '2023-05-29'),

(10, 'Classic Movies', '2023-05-30');

select \* from MovieListManagement.MoviesInList