

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Бази даних»**

**на тему: «DLL та DML оператори мови SQL»**

Виконав:

Студент гр. ПЗ2011

Кулик С.В.

Прийняв:

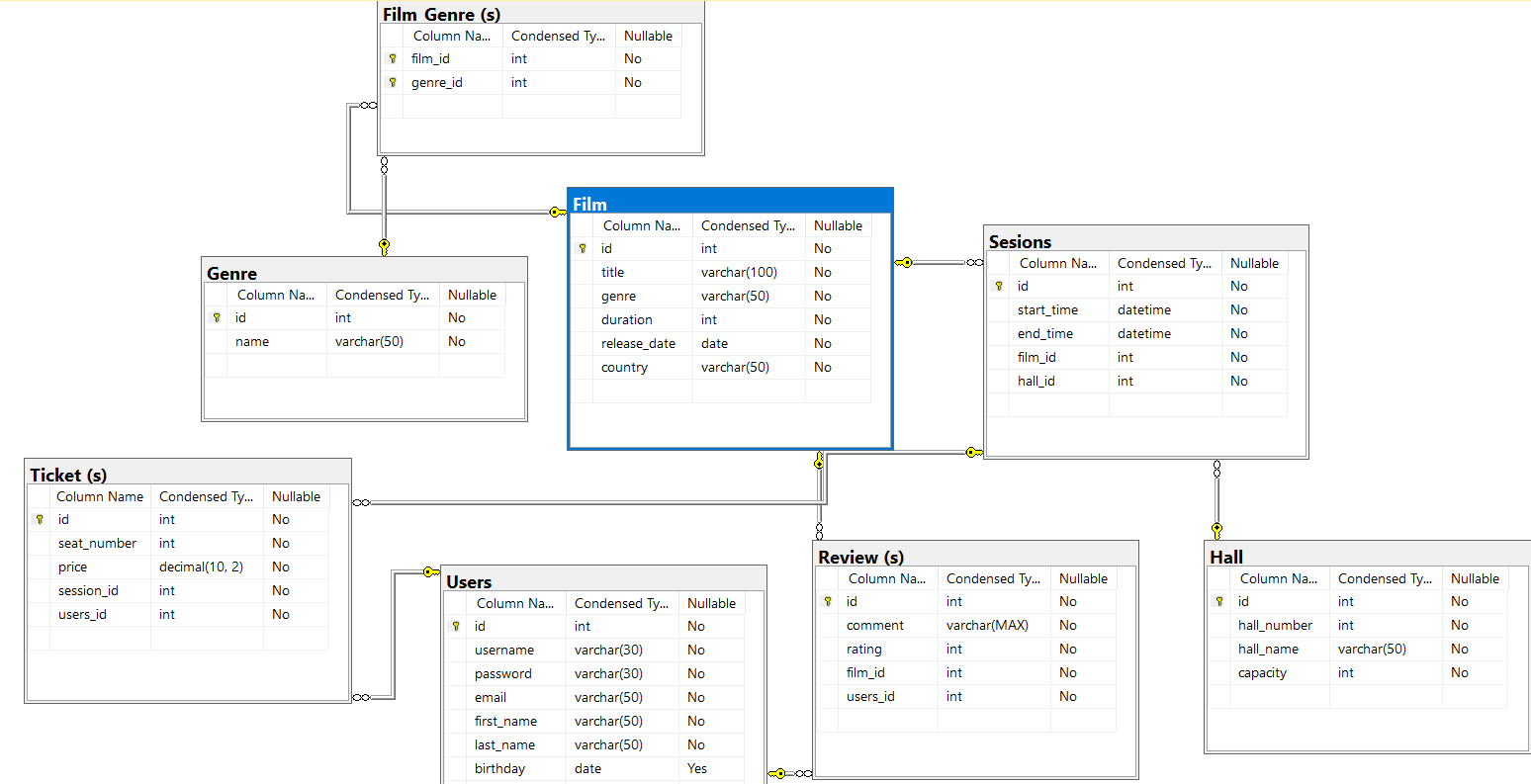
Самойлов С. П.

Дніпро, 2023

# Тема. DLL та DML оператори мови SQL

# Мета. Здобути досвід використання DDL операторів мови SQL для створення реляційних баз даних. Отримати навички використання DML операторів мови SQL для додавання та модифікації записів у реляційних базах даних. Отримати досвід використання операторів мови SQL, що забезпечують цілість динних. Навчитися писати SQL-скрипти розгортання реляційних баз даних.

**Скріншот діаграми бази даних із середовища SQL Server Management Studio**

****

**Опис предметної області та таблиць баз даних**

Таблиця "Фільм" містить інформацію про фільми, які доступні у кінотеатрі. Зв'язок між таблицею "Фільм" та "Жанр" реалізується за допомогою таблиці "Фільм\_Жанр", яка дозволяє встановлювати зв'язок багато-до-багатьох між фільмами та жанрами.

Таблиця "Зал" містить інформацію про кінозали, які є у кінотеатрі. Зв'язок між таблицею "Сеанс" та "Зал" є багато-до-одного, оскільки один сеанс може відбуватися лише в одному залі, але в одному залі може проходити декілька сеансів.

Таблиця "Сеанс" містить інформацію про сеанси, які проводяться у кінотеатрі. Кожен сеанс пов'язаний з певним фільмом з таблиці "Фільм" та залом з таблиці "Зал". Зв'язок між таблицею "Сеанс" та "Квиток" є один-до-багатьох, оскільки на один сеанс може продаватися декілька квитків, але кожен квиток може бути проданий лише на один сеанс.

Таблиця "Користувач" містить інформацію про користувачів, які зареєструвалися на сайті кінотеатру. Зв'язок між таблицею "Квиток" та "Користувач" є багато-до-одного, оскільки кожен квиток може бути проданий лише одному користувачу, але кожен користувач може придбати декілька квитків на різні сеанси.

Таблиця "Квиток" містить інформацію про квитки, які були придбані на сеанси у кінотеатрі. Кожен квиток пов'язаний з певним сеансом з таблиці "Сеанс". Зв'язок між таблицею "Відгук" та "Користувач" та "Фільм" є багато-до-одного, оскільки кожен відгук може бути написаний лише одним користувачем та про один фільм.

Таблиця "Відгук" містить інформацію про відгуки, які були залишені користувачами про певний фільм. Кожен відгук пов'язаний з певним користувачем з таблиці "Користувач" та певним фільмом з таблиці "Фільм".

Загалом, всі таблиці пов'язані між собою, що дозволяє кінотеатру відслідковувати інформацію про фільми, які доступні у кінотеатрі, сеанси, які проводяться, та квитки, які були придбані користувачами. Крім того, користувачі можуть залишати відгуки про фільми, що дозволяє кінотеатру відслідковувати думки своїх клієнтів та покращувати якість своїх послуг.

Начало формы

**Файл розгортки бази даних**

use master

go

drop database CinemaDB

create database CinemaDB

on primary

(name = CinemaDB1\_dat,

filename = 'D:\prog\ms sql\MSSQL16.MSSQLSERVER\MSSQL\DATA\CinemaDB1\_dat.mdf',

size = 10,

maxsize= 1000,

filegrowth = 15% ),

(name = CinemaDB2\_dat,

filename = 'D:\prog\ms sql\MSSQL16.MSSQLSERVER\MSSQL\DATA\CinemaDB2\_dat.mdf',

size = 10,

maxsize= 1000,

filegrowth = 15% ),

filegroup CinemaDBGroup

(name = CinemaDBGroup1\_dat,

filename = 'D:\prog\ms sql\MSSQL16.MSSQLSERVER\MSSQL\DATA\CinemaDBGroup1\_dat.mdf',

size = 10,

maxsize= 1000,

filegrowth = 15% ),

(name = CinemaDBGroup2\_dat,

filename = 'D:\prog\ms sql\MSSQL16.MSSQLSERVER\MSSQL\DATA\CinemaDBGroup2\_dat.mdf',

size = 10,

maxsize= 1000,

filegrowth = 15% )

log on

( NAME = CinemaDB\_log,

FILENAME = 'D:\prog\ms sql\MSSQL16.MSSQLSERVER\MSSQL\DATA\CinemaDB\_log.ldf',

SIZE = 5MB,

MAXSIZE = 30MB,

FILEGROWTH = 5MB ) ;

go

use CinemaDB

go

create schema s

go

-- Створення таблиці "Фільм"

create table dbo.Film(

id int primary key identity,

title varchar(100) not null,

genre varchar(50) not null,

duration int not null,

release\_date date not null,

country varchar(50) not null

)on CinemaDBGroup

-- Створення таблиці "Зал"

create table dbo.Hall(

id int primary key identity,

hall\_number int not null,

hall\_name varchar(50) not null,

capacity int not null

)on CinemaDBGroup

-- Створення таблиці "Сеанс"

create table dbo.Sesions(

id int primary key identity,

start\_time datetime not null,

end\_time datetime not null,

film\_id int not null foreign key references Film(id),

hall\_id int not null foreign key references Hall(id)

)on CinemaDBGroup

-- Створення таблиці "Користувач"

create table dbo.Users(

id int primary key identity,

username varchar(30) not null,

password varchar(30) not null,

email varchar(50) not null,

first\_name varchar(50) not null,

last\_name varchar(50) not null

)

-- Створення таблиці "Квиток"

create table s.Ticket(

id int primary key identity,

seat\_number int not null,

price decimal(10, 2) not null,

session\_id int not null foreign key references Sesions(id),

users\_id int not null foreign key references Users

)

-- Створення таблиці "Відгук"

create table s.Review(

id int primary key identity,

comment varchar(max) not null,

rating int not null,

film\_id int not null foreign key references Film(id),

users\_id int not null foreign key references Users(id)

)

-- Створення таблиці "Жанр"

create table dbo.Genre(

id int primary key identity,

name varchar(50) not null

)

-- Створення таблиці "Фільм\_Жанр" для зв'язку m-n між "Фільм" та "Жанр"

create table s.Film\_Genre(

film\_id int not null foreign key references Film(id),

genre\_id int not null foreign key references Genre(id),

primary key (film\_id, genre\_id)

)

insert into dbo.Film (title, genre, duration, release\_date, country) values

('The Shawshank Redemption', 'Drama', 142, '1994-09-23', 'United States'),

('The Godfather', 'Drama', 175, '1972-03-24', 'United States'),

('The Dark Knight', 'Action', 152, '2008-07-18', 'United States'),

('The Lord of the Rings: The Fellowship of the Ring', 'Adventure', 178, '2001-12-19', 'New Zealand'),

('Forrest Gump', 'Drama', 142, '1994-07-06', 'United States'),

('Inception', 'Action', 148, '2010-07-16', 'United States'),

('The Matrix', 'Action', 136, '1999-03-31', 'United States'),

('Pulp Fiction', 'Crime', 154, '1994-05-21', 'United States'),

('The Silence of the Lambs', 'Thriller', 118, '1991-01-30', 'United States'),

('Fight Club', 'Drama', 139, '1999-10-15', 'United States')

insert into dbo.Hall (hall\_number, hall\_name, capacity) values

(1, 'Main Hall', 200),

(2, 'VIP Hall', 50),

(3, 'Small Hall', 80),

(4, 'Gold Hall', 100),

(5, 'Family Hall', 150),

(6, 'Platinum Hall', 70),

(7, 'Premium Hall', 90),

(8, 'Deluxe Hall', 120),

(9, 'Ultra Hall', 60),

(10, 'Imax Hall', 300)

insert into dbo.Sesions (start\_time, end\_time, film\_id, hall\_id) values

('2023-05-11 14:00:00', '2023-05-11 16:00:00', 1, 1),

('2023-05-11 16:30:00', '2023-05-11 18:30:00', 2, 2),

('2023-05-11 19:00:00', '2023-05-11 21:00:00', 3, 3),

('2023-05-12 13:00:00', '2023-05-12 15:00:00', 4, 1),

('2023-05-12 16:30:00', '2023-05-12 18:30:00', 5, 2),

('2023-05-12 19:00:00', '2023-05-12 21:00:00', 6, 3),

('2023-05-13 14:00:00', '2023-05-13 16:00:00', 7, 1),

('2023-05-13 16:30:00', '2023-05-13 18:30:00', 8, 2),

('2023-05-13 19:00:00', '2023-05-13 21:00:00', 9, 3),

('2023-05-14 13:00:00', '2023-05-14 15:00:00', 10, 1)

insert into dbo.Users (username, password, email, first\_name, last\_name) values

('johndoe', 'password1', 'johndoe@example.com', 'John', 'Doe'),

('janedoe', 'password2', 'janedoe@example.com', 'Jane', 'Doe'),

('bobsmith', 'password3', 'bobsmith@example.com', 'Bob', 'Smith'),

('alicesmith', 'password4', 'alicesmith@example.com', 'Alice', 'Smith'),

('mikewilliams', 'password5', 'mikewilliams@example.com', 'Mike', 'Williams'),

('karenbrown', 'password6', 'karenbrown@example.com', 'Karen', 'Brown'),

('davidlee', 'password7', 'davidlee@example.com', 'David', 'Lee'),

('sarahlee', 'password8', 'sarahlee@example.com', 'Sarah', 'Lee'),

('peterjones', 'password9', 'peterjones@example.com', 'Peter', 'Jones'),

('maryjones', 'password10', 'maryjones@example.com', 'Mary', 'Jones');

insert into s.Ticket (seat\_number, price, session\_id, users\_id) values

(1, 10.50, 1, 2),

(2, 10.50, 1, 3),

(3, 10.50, 1, 4),

(4, 10.50, 2, 2),

(5, 10.50, 2, 3),

(6, 10.50, 2, 4),

(7, 10.50, 3, 2),

(8, 10.50, 3, 3),

(9, 10.50, 3, 4),

(10, 8.50, 4, 2)

insert into s.Review (comment, rating, film\_id, users\_id) values

('Great movie! Highly recommend it', 9, 1, 1),

('Disappointing film, wouldn''t watch again', 4, 2, 2),

('One of the best films I''ve seen this year', 10, 3, 3),

('Decent movie, but not worth the ticket price', 6, 4, 4),

('The acting was superb, but the plot fell flat', 7, 5, 5),

('A heartwarming film that left me feeling uplifted', 8, 6, 6),

('I expected more from this movie, but it was okay', 5, 7, 7),

('One of the most visually stunning films I''ve ever seen', 9, 8, 8),

('Not my cup of tea, but I can see why others might enjoy it', 6, 9, 9),

('An absolute masterpiece. A must-see for all movie lovers', 10, 10, 10)

insert into dbo.Genre (name) values

('Comedy'), ('Action'), ('Drama'), ('Thriller'), ('Horror'),

('Romance'), ('Adventure'), ('Science Fiction'), ('Fantasy'), ('Mystery');

insert into s.Film\_Genre (film\_id, genre\_id) values

(1, 1), (1, 6), (2, 3), (2, 9), (3, 2),

(3, 4), (4, 5), (4, 7), (5, 8), (5, 10);

alter table s.Ticket

add constraint price check(price > 0);

go

alter table dbo.Users

add birthday date;

go

update dbo.Film

set duration = 200

where title = 'The Godfather';

delete from s.Ticket

where price < 10;

go

**Висновок**

Під час виконання лабораторної роботи було отримано досвід використання мови SQL для створення та модифікації реляційних баз даних. Було опрацьовано DDL та DML оператори, а також оператори забезпечення цілісності даних. Також було навченося писати SQL-скрипти розгортання реляційних баз даних. Здобутий досвід дозволить ефективно працювати з базами даних у майбутньому.