

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра інформаційні системи та технології

Тема "База Даних «Сценарій до фільму»"

Курсова робота  
З дисципліни «Бази даних»

Керівник

Бойко О.В.

«Допущений до захисту»

\_\_\_\_\_  
(Особистий підпис керівника)  
« » \_\_\_\_\_ 2018р.  
Захищений з оцінкою

\_\_\_\_\_  
(оцінка)

Члени комісії:

\_\_\_\_\_  
(особистий підпис)

\_\_\_\_\_  
(особистий підпис)

Виконавець

ст. Мелешко Ю.С

залікова книжка № \_\_\_\_ – \_\_\_\_

гр. ІА-24

\_\_\_\_\_  
(особистий підпис виконавця)  
« »  2023р.

\_\_\_\_\_  
(розшифровка підпису)

\_\_\_\_\_  
(розшифровка підпису)

Київ -2023

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

(назва навчального закладу)

Кафедра ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Дисципліна «Бази даних»

Курс 2 Група ІА-24 Семестр 3

**ЗАВДАННЯ**  
**на курсову роботу студента**

**Мелешко Юлія Сергіївна**  
(прізвище, ім'я, по батькові)

1. Тема роботи База даних «Сценарій до фільму».
2. Строк здачі студентом закінченої роботи 28.12.2023
3. Вихідні дані до роботи:  
вивести інформацію про акторів, які приймають участь у фільмі,  
вивести інформацію, які ролі грають які актори,  
вивести інформацію які локації задіяні у зйомках,  
вивести інформацію, які зйомки знімали які режисери,  
вивести інформацію, який декор задіяно у яких зйомках,
4. Зміст розрахунково – пояснювальної записки (перелік питань, що підлягають розробці)  
постановка задачі, огляд існуючих рішень, визначення вимог до бази даних, інфологічне проектування, визначення вимог до оперативної обстановки, вибір конкретної СУБД, даталогічне проектування, фізичне проектування.  
**Додатки:**  
Додаток А – створення зв'язку в MySQL Workbench, Додаток Б – розробка скриптів для створення таблиць, обмежень, Додаток В – відпрацювання скриптів у СУБД, Додаток Г – скрипти запитів до БД, Додаток Д – заповнення БД даними.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
ER-діаграма інфологічної моделі, даталогічна модель, фізична модель, знімки екрана SQL скриптів, знімки екрана налаштування середовища, знімки екрана відпрацювання скриптів.
- 6.

Дата видачі завдання 20.09.2023

## КАЛЕНДАРНИЙ ПЛАН

№, п/п	Назва етапів виконання курсової роботи	Строк виконання етапів роботи	Підписи або примітки
1.	Видача теми курсової роботи	22.09.2023	
2.	Вступ, постановка задачі	24.10.2023	
3.	Визначення вимог до проекту бази даних	26.10.2023	
4.	Огляд існуючих рішень	02.11.2023	
5.	Інфологічне проектування	09.11.2023	
6.	Даталогічне проектування	23.11.2023	
7.	Визначення вимог до операційного середовища, вибір СУБД	07.12.2023	
8.	Фізичне проектування	21.12.2023	
9.	Титульний аркуш, завдання, анотація	28.12.2023	
10.	Захист курсової роботи	06.12.2024	
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			

Студент Мелешко Юлія Сергіївна

\_\_\_\_\_

(підпис)

(прізвище, ім'я, по батькові)

Керівник \_\_\_\_\_

\_\_\_\_\_

(підпис)

(прізвище, ім'я, по батькові)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

## АНОТАЦІЯ

Метою курсової роботи є проектування бази даних для сценарію до фільму, який знімають режисери

Під час роботи виконується повний цикл інфологічного та даталогічного проектування з урахуванням вимог до предметної області. На основі цього будується фізична реалізація бази даних за допомогою СУБД, що обирається спираючись на можливості операційної системи та інші фактори, що залежать від обчислювальних можливостей. Результатом роботи є програмний продукт, що готовий до застосування.

## ЗМІСТ

<u>ВСТУП</u> .....	5
1. <u>Постановка задачі</u> .....	6
2. <u>Огляд існуючих рішень</u> .....	7
3. <u>Визначення вимог до бази даних</u> .....	8
4. <u>Інфологічне проектування</u> .....	9
4.1 <u>Аналіз предметної області</u> .....	10
4.2 <u>Побудова моделі предметної області</u> .....	13
4.3 <u>Визначення меж інформаційної підтримки</u> .....	18
4.4 <u>Зовнішні специфікації функції</u> .....	19
4.5 <u>Групи користувачів бази даних</u> .....	20
5. <u>Визначення вимог до оперативної обстановки</u> .....	21
6. <u>Вибір конкретної СУБД</u> .....	22
7. <u>Даталогічне проектування</u> .....	23
7.1 <u>Класифікація зв'язків та формалізація зв'язків</u> .....	23
7.2 <u>Нормалізація бази даних</u> .....	25
7.3 <u>Введення обмежень цілісності та визначення типів даних</u> .....	29
8. <u>Фізичне проектування</u> .....	33
8.1 <u>Розробка скриптів під конкретну СУБД</u> .....	33
8.2 <u>Відпрацювання скриптів (створення фізичної бази даних)</u> .....	35
<u>ВИСНОВКИ</u> .....	37
<u>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</u> .....	38
ДОДАТКИ.....	39
<u>ДОДАТОК А</u> .....	39
<u>ДОДАТОК Б</u> .....	40
<u>ДОДАТОК В</u> .....	47
<u>ДОДАТОК Г</u> .....	55
<u>ДОДАТОК Д</u> .....	64

## ВСТУП

У сучасному світі, на фоні стрімкого розвитку технологій, важливим елементом для багатьох творчих індустрій є розробка сценаріїв до фільмів. Завдяки поєднанню технічних можливостей та творчого підходу, сценаристи отримали можливість створювати захоплюючі та емоційні шедеври. Зростаюча конкуренція в індустрії кіно вимагає від творців не лише таланту, але й раціонального підходу до процесу створення сценарію.

Із впровадженням цифрових технологій у кіноіндустрію, створення та збереження сценаріїв на електронних носіях стає дедалі актуальнішим. Це дозволяє авторам ефективно обробляти та вдосконалювати свої творчі ідеї, зменшуючи час, який колись витрачався на ручну обробку та переробку матеріалів.

Світовий інтернет поєднав талановитих сценаристів з різних куточків планети, створюючи унікальну можливість для співпраці та обміну ідеями. Величезний обсяг інформації, який стає доступним завдяки Інтернету, дозволяє сценарістам здійснювати дослідження та вдосконалювати свої творіння.

У цьому контексті, створення ефективної системи зберігання та обробки сценаріїв стає надзвичайно важливим завданням. В даній роботі будемо розглядати ключові етапи створення сценарію для фільмів, включаючи аналіз вимог до фільму, логічне та сценарне проектування, а також фізичне втілення ідей у конкретний сценарний матеріал.

Важливо враховувати, що правильно створений сценарій є основою успішного фільму, адже він визначає структуру, діалоги та загальний настрій стрічки. Мета даної роботи - створити зручний інструмент для сценаристів, що дозволить їм ефективно реалізовувати свої творчі задуми та вносити свій внесок у світову кінематографію.

## 1. Постановка задачі

Предметна область «Сценарій до фільму».

База даних створюється для режисерів, сценаристів, акторів, дизайнерів, та інших співробітників для зручного керування процесом знімання та монтажу фільму.

База даних повинна містити дані про співробітників знімальної компанії, локації для зйомок, інформацію про сценаристів, режисерів, інформацію про потрібний декор та обладнання для акторів, час зайнятості акторів. Повинна мати можливість надати звіт із зйомки.

База повинна працювати без програмного додатку, самостійно. При цьому не виключається можливість побудови додатку до бази даних.

Метою даної роботи буде побудувати саме такий програмний продукт.

У даній роботі потрібно пройти через наступні етапи побудови бази даних:

аналіз предметної області, визначення вимог до бази даних , інфологічне проектування, визначення вимог до оперативної обстановки, вибір конкретної СУБД, даталогічне проектування, фізичне проектування, реалізація.

## 2. Огляд існуючих рішень

Під час дослідження готових рішень для сценаріїв фільму було знайдено кілька варіантів на ринку. Однією зі знайдених баз даних є та, яку розробила команда Flames. Також програма Celtx. Вони створили базу даних спеціально для сценаріїв фільмів, і, ймовірно, вона вже має певну структуру та набір таблиць, придатних для роботи зі сценаріями та пов'язаними з ними даними.

Крім того, були знайдені готові таблиці MS Excel, які, за вказівкою, не є базами даних і можуть бути менш зручними для зберігання та обробки великих обсягів даних. Вони, ймовірно, призначені для простіших завдань і не надають такого рівня структурованості та зручності, як бази даних.



### 3. Визначення вимог до бази даних

Від самої бази даних вимагається швидкий пошук, простота використання та оновлення даних, зменшення надлишковості даних, захист від несанкціонованого доступу, цілісність, надійність. Запити повинні відпрацьовуватися швидко, здебільшого вони будуть невеликими, що відповідає спрямуванню OLAP (online analytical processing). Тобто необхідно правильно вибрати систему управління базами даних.

База даних повинна зберігати фінансову інформацію сценарію фільму та персональні дані співробітників. Саме тому база даних має бути надійною та захищеною.

Перелік питань, на які має відповідати база даних :

1. Вивести інформацію про співробітників, які приймають участь у створенні фільму
2. Вивести інформацію про заплановані зйомки.
3. Вивести інформацію про час зайнятості акторів.
4. Вивести інформацію локацій, на яких заплановані зйомки.
4. Вивести інформацію обладнання, яке потрібне для зйомок.
5. Вивести кошторис декору, гриму, обладнання акторів.
6. Вивести дати та місця зйомок.
7. Вивести дані про орендований транспорт.
9. Вивести інформацію про продюсерів, які підтримують фільм.
10. Вивести детальну інформацію про декор, на який потрібна передплата.

#### **4. Інфологічне проектування**

Інфологічне проектування є важливим етапом у проектуванні бази даних. Саме тут необхідно вивчити всі вимоги сценарію, дізнатися про всі тонкощі предметної області. Це спілкування з сценаристами, менеджерами. Треба показати клієнту майбутню базу даних саме в такій формі, у якій зрозуміють обидві сторони, а найважливіше – режисер та команда. Цей етап впливає на подальші етапи проектування, бо саме від цього залежить, який напрямок оберуть інженери при розробленні БД.

## 4.1 Аналіз предметної області

База даних створюється для інформаційного обслуговування співробітників команди, працівників, які безпосередньо займаються сценарієм фільму.

База також має підтримуватися під час зйомки, для отримання співробітникам актуальної інформації. База необхідна для планування зйомок, та після них, для створення звіту про витрати та заробітні плати.

База даних повинна містити дані про співробітників, заплановані зйомки, їх результат, локації, декорації, транспорт, що наданий для пересування, і надавати можливість отримувати всілякі звіти.

Відповідно до предметної області система будується з урахуванням наступних особливостей:

- кожен актор може грати лише одну роль
- команда, яка їде на зйомку користується одним транспортом
- зйомкою займається декілька співробітників
- для однієї ролі потрібно декілька обладнань
- для однієї зйомки використовується одна локація
- Для однієї зйомки потрібно декілька обладнань з декору
- кожен співробітник може займатися декількома зйомками
- Однією зйомкою займається багато співробітників
- В одній зйомці може приймати участь безліч акторів
- з однієї зйомки може впливати лише один звіт

Після аналізу предметної області можна виділити ряд сутностей, і приступити до проектування інфологічної моделі.

Виділимо базові сутності цієї предметної області та їх атрибути:

- Актор. Атрибути – ID, Ім'я, Прізвище, Дата народження, Освіта, номер телефону, Стать, Роль яку виконує
- Роль. Атрибути – ID, Назва, зйомки у яких приймає участь
- Зйомка. Атрибути – ID, Назва, Час та дата, декор який використовується
- Сценарист – Атрибути - ID, Ім'я, Прізвище, зйомки які написав
- Співробітник- Атрибути - ID, Ім'я, Прізвище, номер телефону  
Співробітник повинен зберігати зйомки, в яких він приймає участь
- Режисер – Атрибути - ID, Ім'я, Прізвище, освіта, номер телефону

- Продюсер – Атрибути - ID, Ім'я, Прізвище, номер телефону, фінанси які готовий надати
- Звіт – Атрибути - ID, Назва, Дата, ID зйомки, опис, фінансові витрати  
Звіт повинен зберігати зйомку, по якій він зроблений
- Декор –Атрибути - ID, Назва декору, ID зйомки, ціна, ID локації
- Локація – Атрибути - ID, Назва, Опис, Площа, місце проведення, зйомки у яких використовуються
- Транспорт – Атрибути – ID, Назва, Опис, Вид транспорту, Ціна використання

Таблиця 4.1.2 – Сутності та їх атрибути

Сутності	Атрибути
Актор	ID, ПІБ, Дата народження, Освіта, Номер телефону, Стать, Роль
Роль	ID, Назва, Актор
Зйомка	ID, Назва, Час та дата, Опис, ID Режисера
Сценарист	ID, Ім'я, Прізвище
Співробітник	ID, Ім'я, Прізвище, номер телефону, кількість зайнятих годин
Режисер	ID, Ім'я, Прізвище, освіта, номер телефону
Продюсер	ID, Ім'я, Прізвище, номер телефону, фінанси які готовий надати
Звіт	ID, Назва, Дата, опис, фінансові витрати
Локація	ID, Назва, Опис, Площа, місце проведення
Транспорт	ID, Назва, Опис, Вид транспорту, Ціна використання

У наступній таблиці наведемо можливі зв'язки між сутностями, та атрибути, які вони матимуть :

Таблиця 4.1.2 - Зв'язки між сутностями

Зв'язок	Атрибути
Фінансування зйомки	Продюсер, Декор, Транспорт, Ціна
Налаштування локації	Локація, Співробітники
Вибір акторів на роль	Актор, Роль, Режисер
Гримування персонажів	Співробітники, Ролі
Декор	ID, Назва, ID зйомки, ID локації, Опис, Ціна

## 4.2 Побудова моделі предметної області

Класифікуємо сутності предметної області у наступній таблиці:

Таблиця 4.2.1 - Класифікація сутностей

Сутність	Вид сутності
Актор	Стержнева
Роль	Слабка
Зйомка	Стержнева
Співробітник	Стержнева
Режисер	Слабка
Продюсер	Стержнева
Звіт	Стержнева
Локація	Стержнева
Транспорт	Слабка
Декор	Асоціативна
Фінанси	Асоціативна
Налаштування локації	Асоціативна
Гримування актора	Асоціативна
Процес зйомки ролей	Асоціативна
Переміщення	Асоціативна

### Класифікуємо атрибути сутностей предметної області:

Таблиця 4.2.2 - Класифікація атрибутів

Сутності	Атрибути	Вид атрибутів
Переміщення	ID локації	Допоміжний
	ID транспорту	Допоміжний
Процес зйомки ролей	ID зйомки	Допоміжний
	ID ролі	Допоміжний
Актор	ID	Ідентифікатор
	ПІБ	Вказівний
	Дата Народження	Описовий
	Освіта	Описовий
	Номер телефону	Вказівний
	Стать	Описовий
Роль	ID	Ідентифікатор
	Назва	Описовий
	ID Актора	Допоміжний
Зйомка	ID	Ідентифікатор
	Назва	Описовий
	Опис	Описовий
	Час та дата	Вказівний
	Кошторис	Описовий
Співробітник	ID	Ідентифікатор
	ПІБ	Описовий
	Номер телефону	Вказівний
	К-ть зайнятих годин	Описовий
Режисер	ID	Ідентифікатор
	ПІБ	Описовий
	Дата народження	Описовий
	Номер телефону	Вказівний
	ID Зйомки	ID зйомки
Продюсер	ID	Ідентифікатор
	ПІБ	Описовий
	Номер телефону	Вказівний
	Фінанси	Описовий
Звіт	ID	Ідентифікатор
	Назва	Описовий
	Дата	Описовий
	Опис	Описовий
	Фінансові витрати	Описовий
	ID зйомки	Допоміжний
Локація	ID	Ідентифікатор

	Назва	Описовий
	Опис	Описовий
	Площа	Описовий
	Місце	Вказівний
Транспорт	ID	Ідентифікатор
	Назва	Описовий
	Опис	Описовий
	Вид	Описовий
	Ціна	Описовий
Декор	ID	Ідентифікатор
	Назва	Описовий
	ID зйомки	Допоміжний
	ID локації	Допоміжний
	Опис	Описовий
	Ціна	Описовий
Фінанси	ID	Ідентифікатор
	Назва	Описовий
	ID Продюсера	Допоміжний
	ID зйомки	Допоміжний
Налаштування локації	ID Співробітника	Допоміжний
	ID Локації	Допоміжний
Гримування персонажів	ID Співробітника	Допоміжний
	ID Актора	Допоміжний



**ER-діаграма (нотація Чена) інфологічної моделі предметної області представлена на наступному рисунку:**

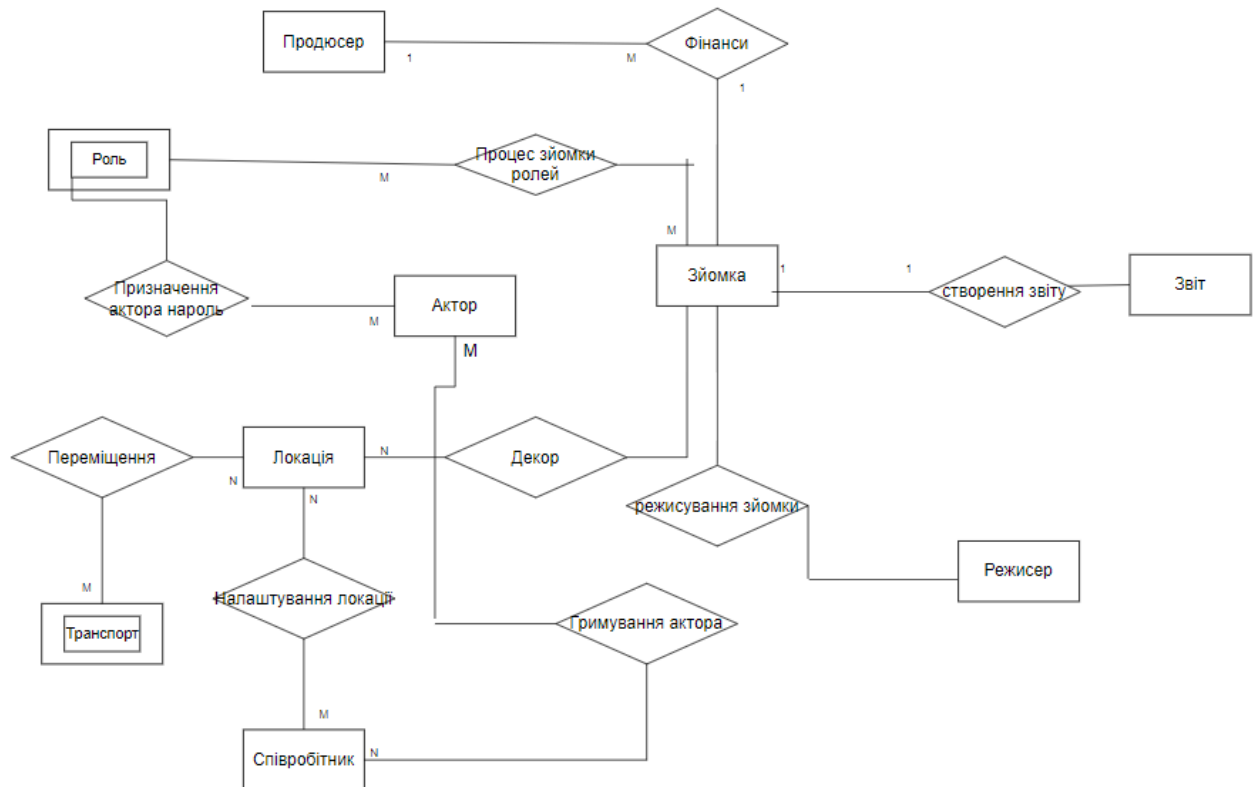
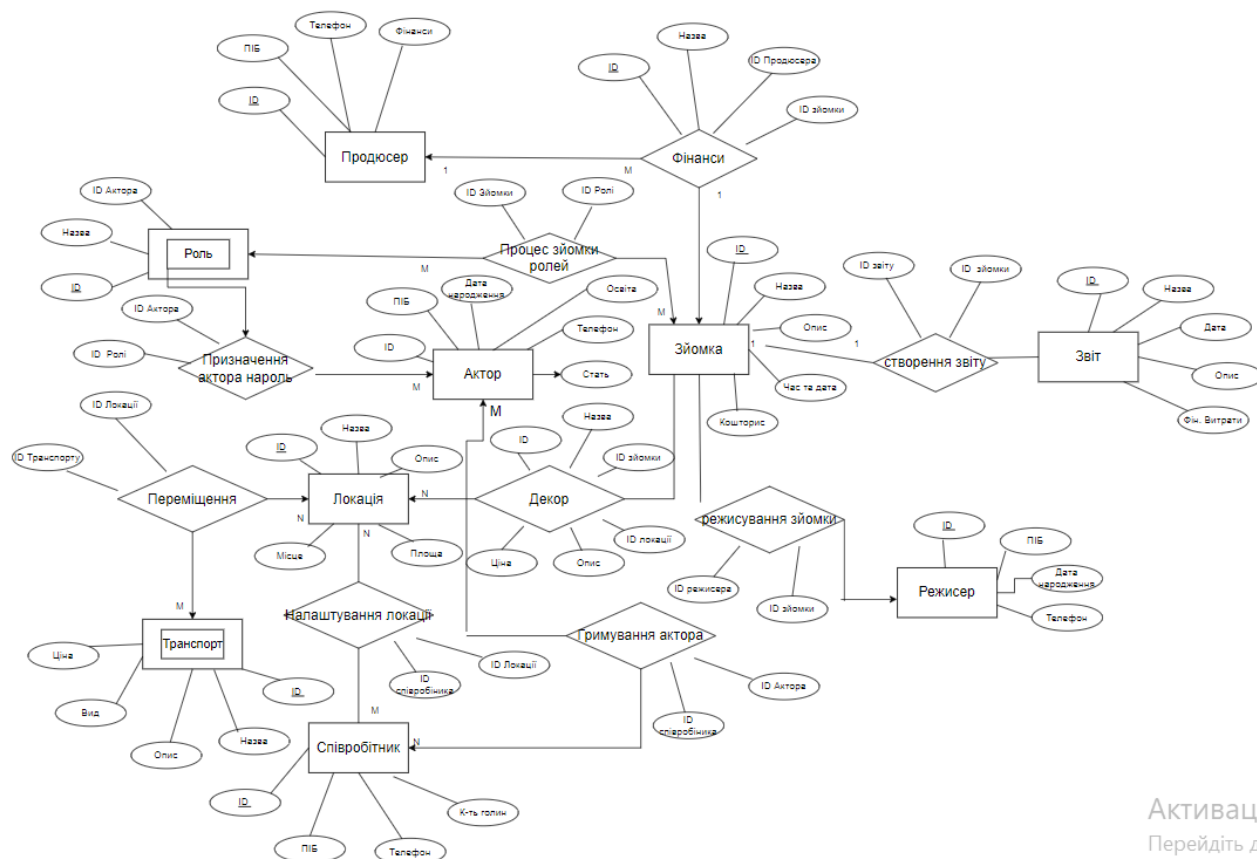


Рисунок 4.2.1 - ER-діаграма сутностей

**ER-діаграма (нотація Чена) сутностей предметної області з атрибутами представлена на наступному рисунку:**



Активация  
Перейдіть до  
Windows

Рисунок 4.2.2 - ER-діаграма сутностей із атрибутами

### 4.3 Визначення меж інформаційної підтримки

База даних даних заповнюється до кінця зйомки фільму, як елемент планування сценарію та витрат на нього. База також має підтримуватися інформацією навіть під час зйомки, для отримання співробітниками актуальної

інформації. Для певної однієї зйомки база необхідна і після закінчення самої зйомки, щоб отримувати бухгалтерські звіти по кошторису фільму, заробітнім платам співробітників.

Система працюватиме лише у межах зйомки фільму. Її метою не є підтримувати фінансову, документарну діяльність фільму. Тобто система не зберігатиме копії документів, витрати на усіх співробітників, їх лікарняні та відпустки. Не є метою системи зберігати дані товарообігу чи дані про перебіг справ. І також в системі не буде інформації про завдання, надані співробітникам, відділи компанії, її структуру тощо.

#### 4.4 Зовнішні специфікації функцій

- Функції для роботи із зйомками;
- Функції для роботи з декором;
- Функції для роботи з співробітниками;
- Функції для роботи з локаціями;
- Функції для роботи з транспортуванням;
- Функції для роботи з акторами;
- Захист даних шляхом обмеження доступу для певних груп користувачів.

Перелік питань, на які має відповідати база даних :

1. Вивести інформацію про співробітників, що приймають участь в зйомці
2. Вивести інформацію про заплановані зйомки.
3. Вивести інформацію про заброньований декор для зйомки.
4. Вивести інформацію ролей, які грають в зйомці.
5. Вивести кошторис зйомки.
6. Вивести дати та місця проведення зйомки.
7. Вивести дані про орендований транспорт.
8. Вивести дані про звіт уже знятих зйомок.
9. Вивести інформацію про всі локації, на яких будуть зйомки.
10. Вивести інформацію, які співробітники допомагають підготуватись яким акторам

## 4.5 Групи користувачів бази даних

Наведемо деяких можливих користувачів та деякі групи користувачів, що можуть бути у предметній області «Сценарій фільму» та активно користуватися базою даних :

- Режисер;
- Менеджер;
- Сценарист;
- Відповідальний за кошторис;
- Співробітник;
- Бухгалтер;

## **5. Визначення вимог до оперативної обстановки**

Від оперативної обстановки вимагається наявність середовища ОС Linux для роботи бази даних та можливість роботи в ньому. Можливість роботи в середовищі ОС Linux здебільшого необхідна для роботи бази даних на сервері. Також вимагається можливість роботи в ОС Windows для тестування скриптів та ПЗ.

Для даної роботи будемо вимагати лише можливість роботи в середовищі ОС Windows 10. Оскільки вона доступна, та показала свою надійність з роками. На машині мається досить дискового простору, здебільшого 50 ГБ буде більше ніж досить невеликої бази. Процесор Intel Core i5 виконає всі поставлені задачі до бази даних та 8 ГБ оперативної пам'яті цілком достатньо для роботи системи управління реляційними базами даних.

## 6. Вибір конкретної СУБД

Для реалізації бази даних оберемо Систему Управління Базами Даних (СУБД) MySQL, через її надійність та доступність. Також одним із критеріїв є вміння працювати з даною СУБД та безкоштовність ліцензійного використання цього продукту.

Система управління базами даних MySQL має наступні вимоги до операційної обстановки:

- Не більше 1ГБ оперативної пам'яті
- Використання одного ядра процесора
- Користувачські дані займають на диску не більше 11ГБ місця
- Може бути встановлена на різних операційних системах, включаючи Windows, Linux та інші

На диску вистачить достатньо місця для даної СУБД. Через те, що MySQL ефективно використовує ресурси та не вимагає значних обсягів оперативної пам'яті, база даних не навантажуватиме систему надто інтенсивно.

Встановимо СУБД MySQL Workbench 8.0 CE(я уже мала встановлений)

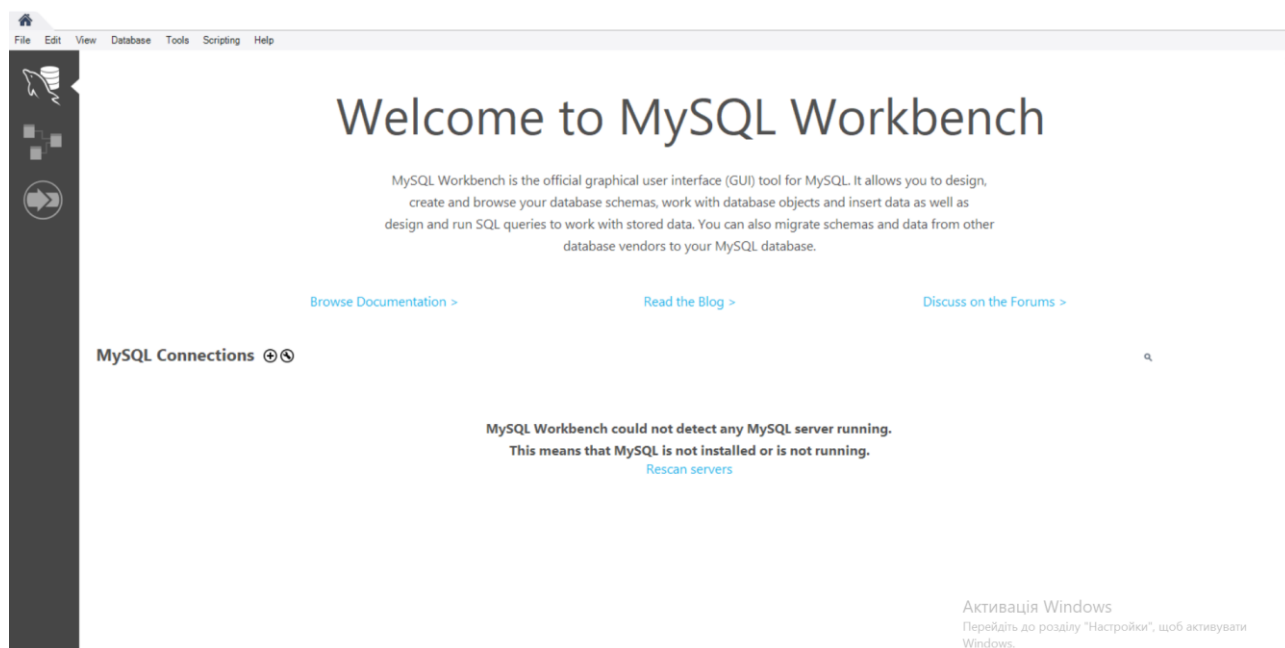


Рисунок 6.1 Workbench

## 7. Даталогічне проектування

На цьому етапі необхідно перетворити результати інфологічного проектування у зручну та оптимальну форму. Потрібно формалізувати зв'язки між сутностями, нормалізувати усі реляційні відношення до третьої нормальної форми, задля виключення дублювання даних. Відсутність цього етапу призводить до протиріччя БД або істотно ускладнює обробку БД. Також введемо обмеження цілісності.

### 7.1 Класифікація зв'язків та формалізація зв'язків

У наступній таблиці наведемо існуючі зв'язки у інфологічній моделі бази даних, опишемо їх, класифікуємо, та оберемо тактику формалізації.

Таблиця 7.1.1 - Класифікація та формалізація зв'язків

Зв'язок	Опис
Локація-Транспорт (переміщення)	1:N, умовний, формалізується за допомогою асоціативної сутності переміщення
Співробітник-Локація (налаштування локації)	M:N, умовний, формалізується за допомогою асоціативної сутності налаштування локації
Співробітник-Актор (гримування актора)	M:N, безумовний, формалізується за допомогою асоціативної сутності «Гримування актора»
Продюсер-зйомка (фінанси)	1:N, безумовний, формалізується за допомогою асоціативної сутності «Фінанси»
Роль-Зйомка (процес зйомки ролей)	M:N, безумовний, формалізується за допомогою асоціативної сутності «процес зйомки»
Зйомка-звіт (Звітування зйомки)	1:N, безумовний, формалізується за допомогою допоміжного атрибута в сутності звіт
Режисер-Зйомка (знімання)	1:N, безумовний, формалізується за допомогою допоміжного атрибута в сутності «Зйомка»
Роль-Актор (призначення актора на роль)	1:1, безумовний, формалізується за допомогою допоміжного атрибута в сутності «Роль»



Локація-Зйомка (Декор)	1:М, умовний, формалізується за допомогою асоціативної сутності «Декор»
---------------------------	---

## 7.2 Нормалізація бази даних

Наведемо етапи нормалізації даталогічної моделі бази даних :

1. Атрибут “ПІБ” є складним і порушує 1НФ. Розібемо його на атрибути: “Ім’я”, “Прізвище”, “По-батькові”.

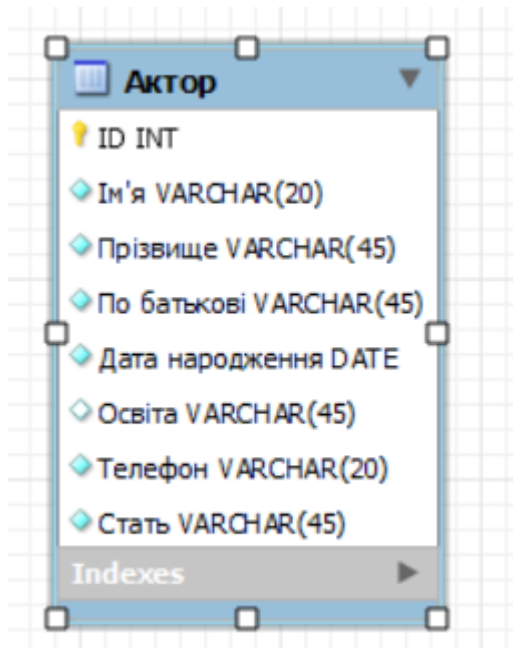
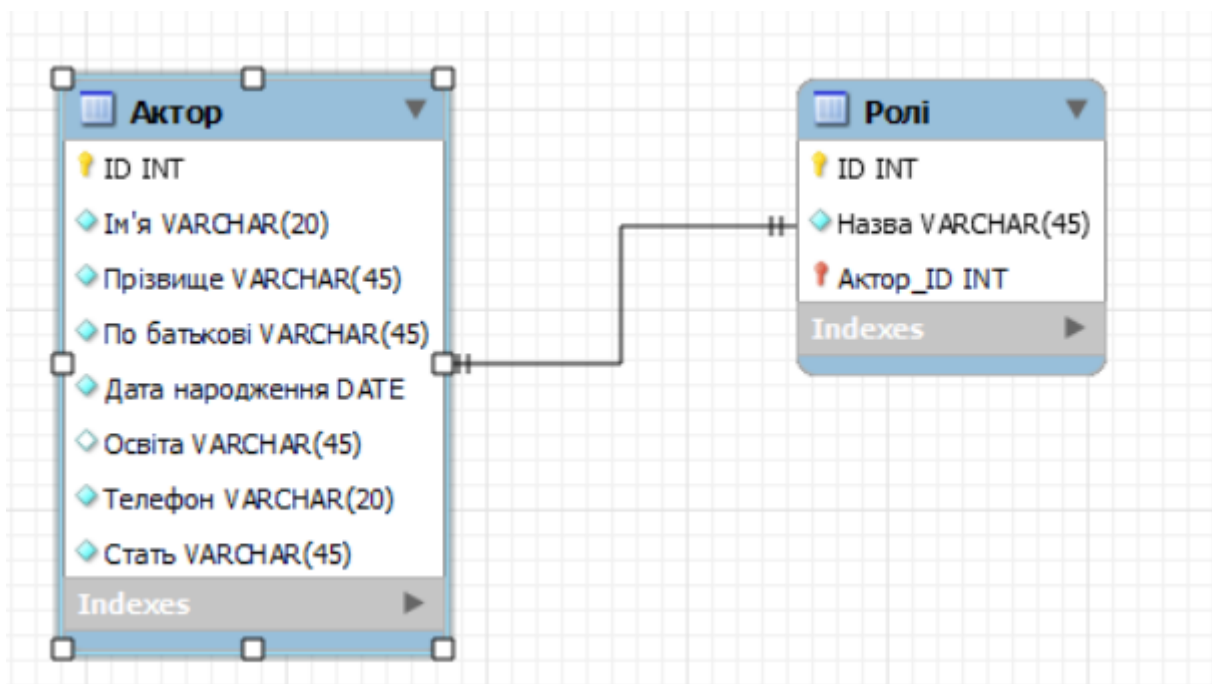


Рисунок 7.2.1 - Нормалізація сутності Особи

2. Далі реалізуємо зв'язок 1:1 із сутністю Ролі

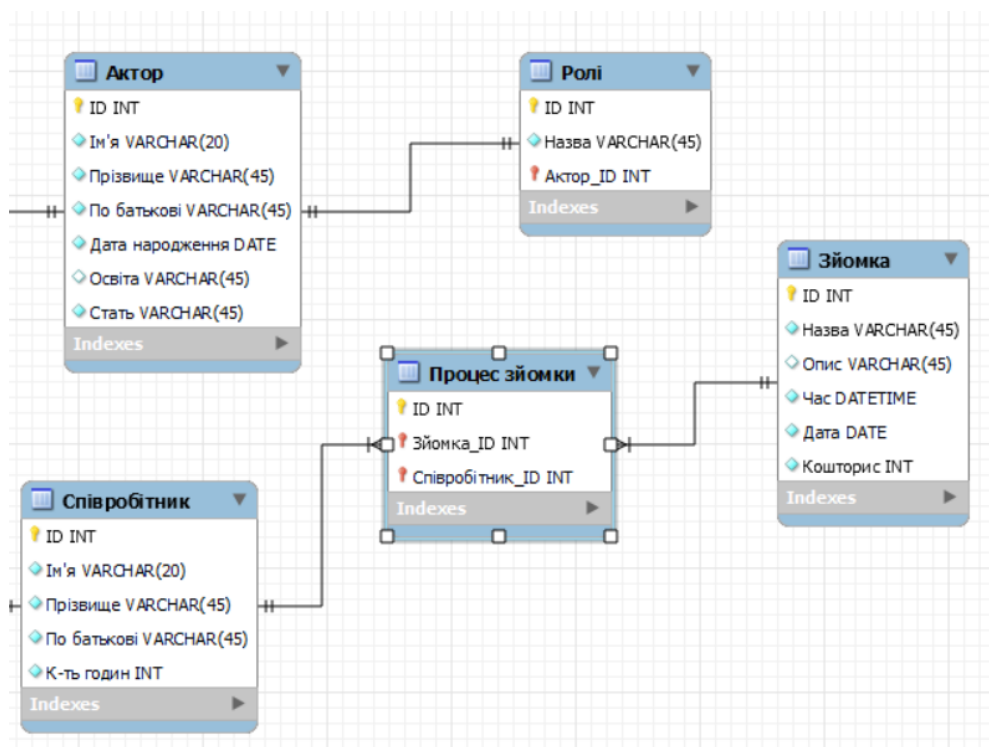


4. Зробимо сутність Співробітники, так же розіб'єм ПБ на три окремих атрибути

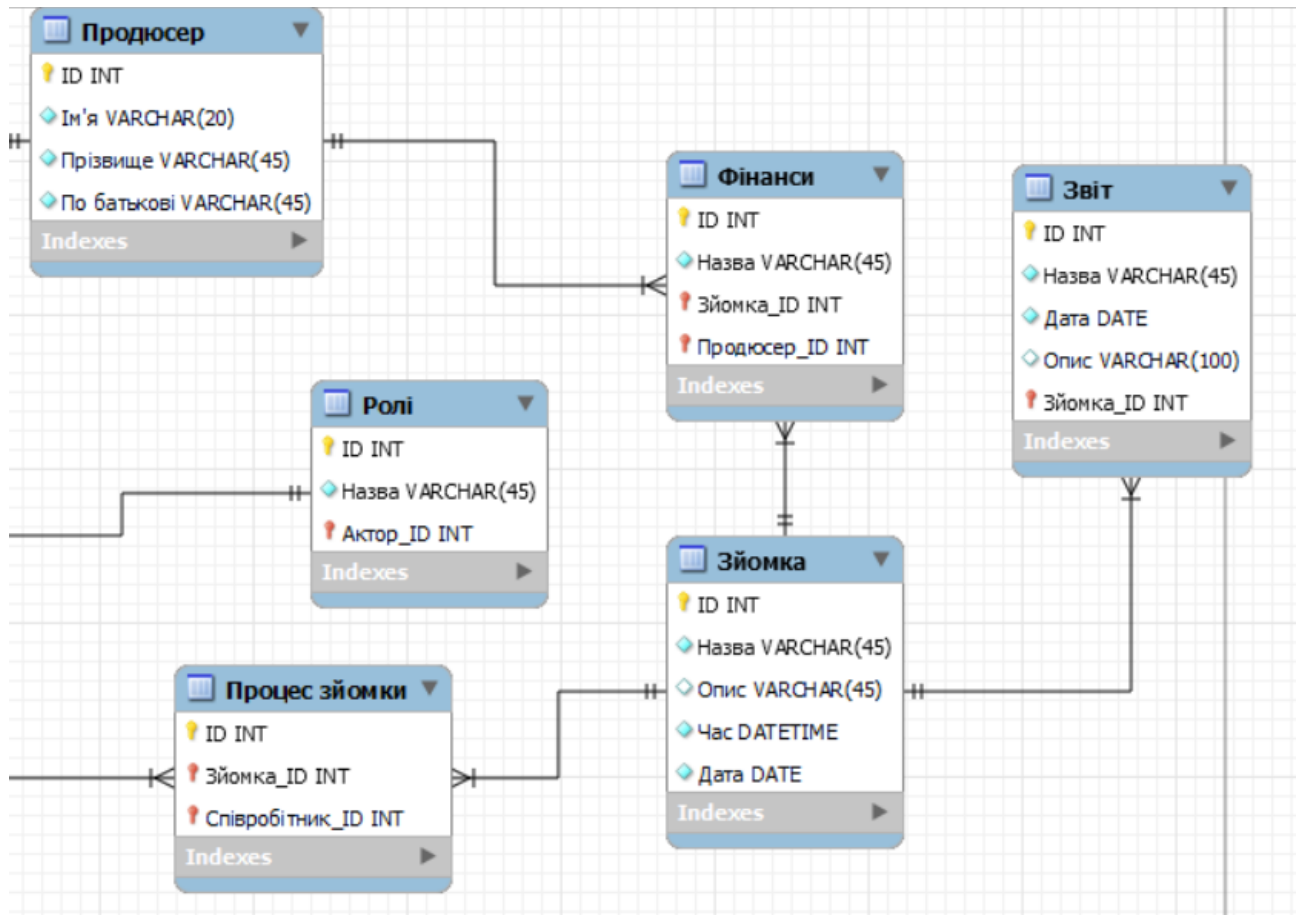
5. Атрибут “Електронна адреса” може порушувати 1НФ, якщо матиме декілька значень. Для даної предметної області нас цікавить лише робоча електронна адреса особи, дійсна електронна адреса компанії чи готелю. Більша кількість значень призведе до надлишковості. Тому порушень 1НФ немає.

6. Атрибут «Освіта» у сутності «Співробітники» міг би порушувати 1НФ, якби міг мати декілька значень, але нас цікавить лише одна вища освіта, тому порушень 1НФ немає.

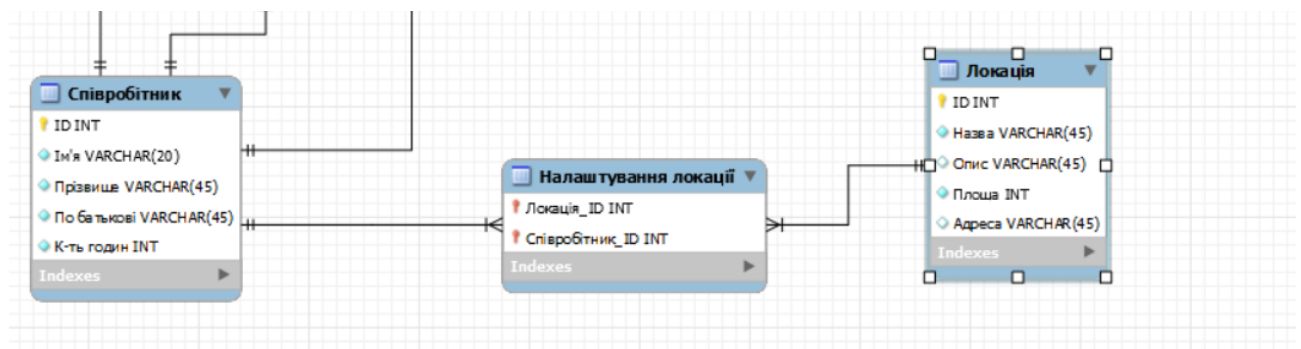
7. Створюємо сутність «Зйомка», Атрибут «Час та Дата» порушує 1НФ, тому розіб'єм його на два окремих атрибути, також додамо асоціативну сутність «Процес зйомки», для зв'язку «Many-to-Many» між Зйомками та Ролями

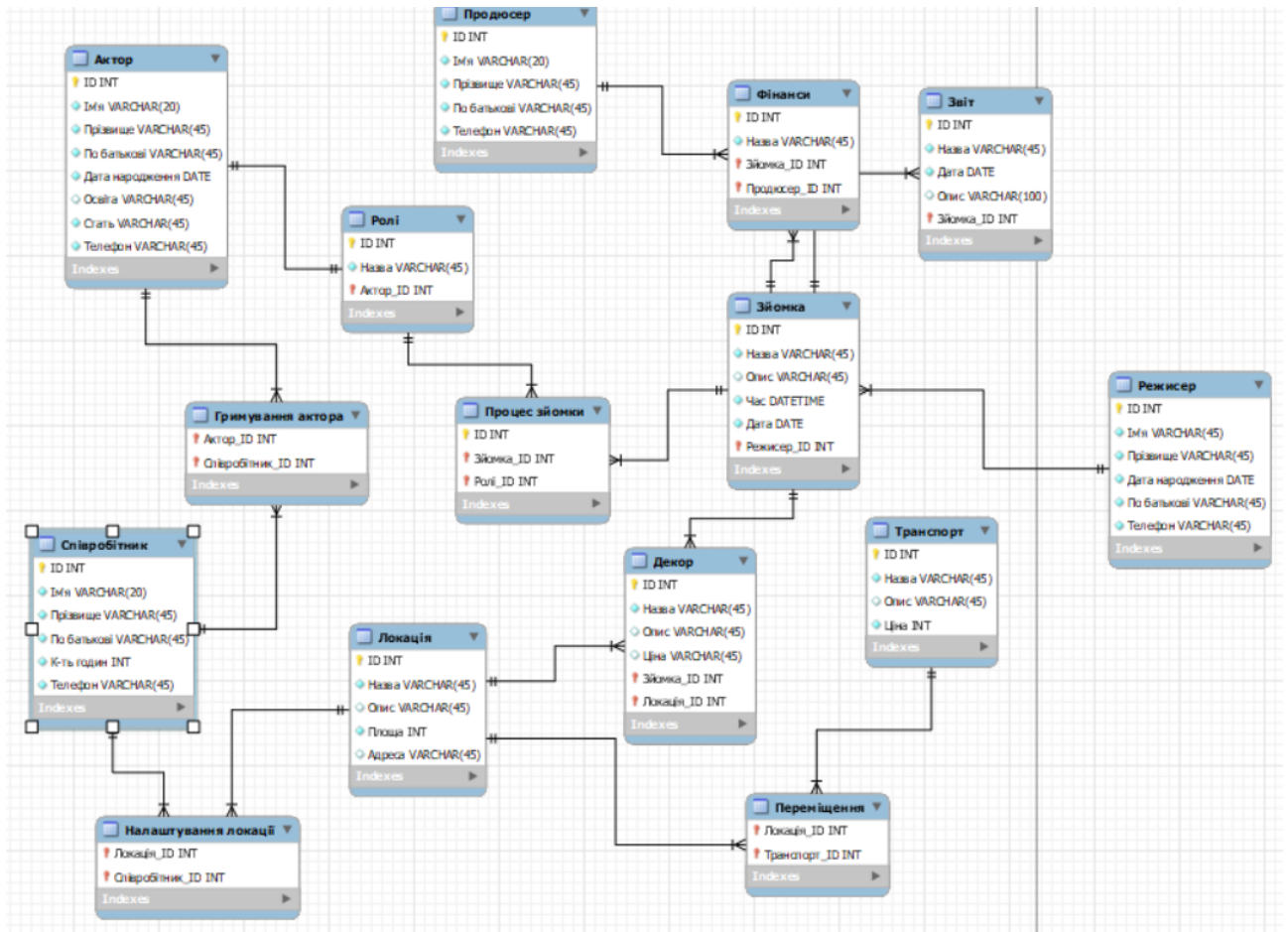


8. Додаємо сутність «Звіт», атрибут «Фінансові витрати» повторюється та має однаковий сенс із атрибутом у сутності «Зйомки», у цій сутності він нам також не потрібен, бо у нас для цього є окрема асоціативна сутність «Фінанси», тому маємо



9. Атрибут «Місце» в сутності «Локація» порушує 1НФ, тому перейменуємо його в «Адресу», з умовою, що фільм знімається в межах одного міста, і цей атрибут буде містити в собі тільки вулицю





10. База даних уже знаходить в 2НФ та 3НФ, тому що вона є у 1НФ, та усі неключові атрибути залежать від первинного ключа

### 7.3 Введення обмежень цілісності та визначення типів даних

Наведемо таблиці з переліком атрибутів, їх типом даних, обмеженнями цілісності та типом атрибуту. У такому форматі таблиці:

Назва таблиці		
Тип атрибуту	Назва атрибуту	Тип даних, обмеження цілісності

Таблиця 7.3.1

Таблиця сутність Актори

ACTORS		
PK	ID	SERIAL
	NAME	VARCHAR(20)
	SURNAME	VARCHAR(45)
	MIDDLE NAME	VARCHAR(45)
	DATE OF BIRTH	DATE
	EDUCATION	VARCHAR(45)
	SEX	VARCHAR(10)
	TELEPHONE	VARCHAR(45)

Таблиця 7.3.2

Таблиця сутності Ролі

ROLES		
PK	ID	SERIAL
	TITLE	VARCHAR(45)
FK	ID_ACTOR	INT

Таблиця 7.3.3

Таблиця сутності Співробітник

EMPLOYEE		
PK	ID	SERIAL
	NAME	VARCHAR(20)
	SURNAME	VARCHAR(45)
	MIDDLE NAME	VARCHAR(45)
	NUM OF WORK HOURS	INT
	TELEPHONE	VARCHAR(20)

Таблиця 7.3.4

Таблиця сутності Продюсер

PRODUSER		
PK	ID	SERIAL
	NAME	VARCHAR(20)
	SURNAME	VARCHAR(45)
	MIDDLE NAME	VARCHAR(45)
	TELEPHONE	VARCHAR(20)

Таблиця 7.3.5

Таблиця сутності Фінанси

FINANCE		
PK	ID	SERIAL
	TITLE	VARCHAR(45)
FK	FILMING_ID	INT
FK	PRODUCER_ID	INT

Таблиця 7.3.6

Таблиця сутності Звіт

REPORT		
PK	ID	SERIAL
	TITLE	VARCHAR(20)
	DATE	DATE
	DESCRIPTION	VARCHAR(100)
FK	FILMING_ID	INT

Таблиця 7.3.7

Таблиця сутності Зйомка

FILMING		
PK	ID	SERIAL
	TITLE	VARCHAR(20)
	DESCRIPTION	VARCHAR(45)
	DATE	DATE
	TIME	TIME
FK	DIRECTOR_ID	INT

Таблиця 7.3.8

Таблиця сутності Режисер

DIRECTOR		
PK	ID	SERIAL
	NAME	VARCHAR(20)
	SURNAME	VARCHAR(45)
	MIDDLE NAME	VARCHAR(45)
	DATE OF BIRTH	DATE
	TELEPHONE	VARCHAR(45)

Таблиця 7.3.9

Таблиця сутності Транспорт

TRANSPORT		
PK	ID	SERIAL
	TITLE	VARCHAR(20)
	DESCRIPTION	VARCHAR(45)
	PRICE	INT

Таблиця 7.3.10

Таблиця сутності Локація

LOCATION		
PK	ID	SERIAL
	TITLE	VARCHAR(20)
	DESCRIPTION	VARCHAR(45)
	SQUARE	INT
	ADDRESS	VARCHAR(45)

Таблиця 7.3.11



Таблиця сутності Декор

DECOR		
PK	ID	SERIAL
	TITLE	VARCHAR(20)
	DESCRIPTION	VARCHAR(45)
	PRICE	INT
FK	FILMING_ID	INT
FK	LOCATION_ID	INT

Таблиця 7.3.12

Таблиця сутності Налаштування Локації

LOCATION SETTINGS		
FK	LOCATION_ID	INT
FK	EMPLOYEE_ID	INT

Таблиця 7.3.13

Таблиця сутності Гримування актора

ACTOR`S MAKE UP		
FK	ACTOR_ID	INT
FK	EMPLOYEE_ID	INT

Таблиця 7.3.14

Таблиця сутності Процес Зйомки

FILMING PROCESS		
PK	ID	SERIAL
FK	FILMING_ID	INT
FK	ROLE_ID	INT

Таблиця 7.3.15

Таблиця сутності Переміщення

TRANSPORTING		
FK	LOCATION_ID	INT
FK	TRANSPORT_ID	INT

Таблиця 7.3.16

## 8. Фізичне проектування

Даний етап є кінцевим у розробці бази даних. Тут необхідно перетворити даталогічну модель у фізичну модель даних, шляхом розробки скриптів під конкретну СУБД з урахуванням усіх обмежень. Після виконання скриптів отримаємо фізичну базу даних.

### 8.1 Розробка скриптів під конкретну СУБД

Наведемо декілька прикладів скриптів по створенню таблиць, первинних та зовнішніх ключів. В MySQL ми можемо підтримати послідовність вставки елементів за допомогою AUTO\_INCREMENT, і нам не потрібно буде прописувати послідовності та тригери, щоб зробити послідовне збільшення індексів елементів

Для правильного виконання та зручного використання назви атрибутів та таблиць перекладемо англійською мовою та використаємо саме так у скриптах. DDL SQL скрипт для створення таблиці акторів (ACTORS) :

```
CREATE DATABASE IF NOT EXISTS scenario;

CREATE TABLE IF NOT EXISTS actors(
    id SERIAL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    surname VARCHAR(40) NOT NULL,
    middle_name VARCHAR(30),
    date_birth DATE NOT NULL,
    education TEXT,
    sex VARCHAR(10) NOT NULL,
    phone_number VARCHAR(15) NOT NULL,
    PRIMARY KEY(id)
);
```

Рисунок 8.1.1 Створення таблиці акторів

DDL SQL скрипт для створення таблиці Ролі (ROLES) :

```
CREATE TABLE IF NOT EXISTS roles(  
    id SERIAL AUTO_INCREMENT,  
    title VARCHAR(20),  
    PRIMARY KEY(ID),  
    actor_id BIGINT UNSIGNED,  
    CONSTRAINT fk_role_actor FOREIGN KEY(actor_id) REFERENCES actors(id)  
);
```

Рисунок 8.1.2 Створення таблиці ролей

## 8.2 Відпрацювання скриптів (створення фізичної бази даних)

Чудово було б побачити базу даних повністю, у зручному вигляді. На наступному рисунку наведемо фізичну модель розробленої бази даних, створену за допомогою застосунку MySQL Workbench:

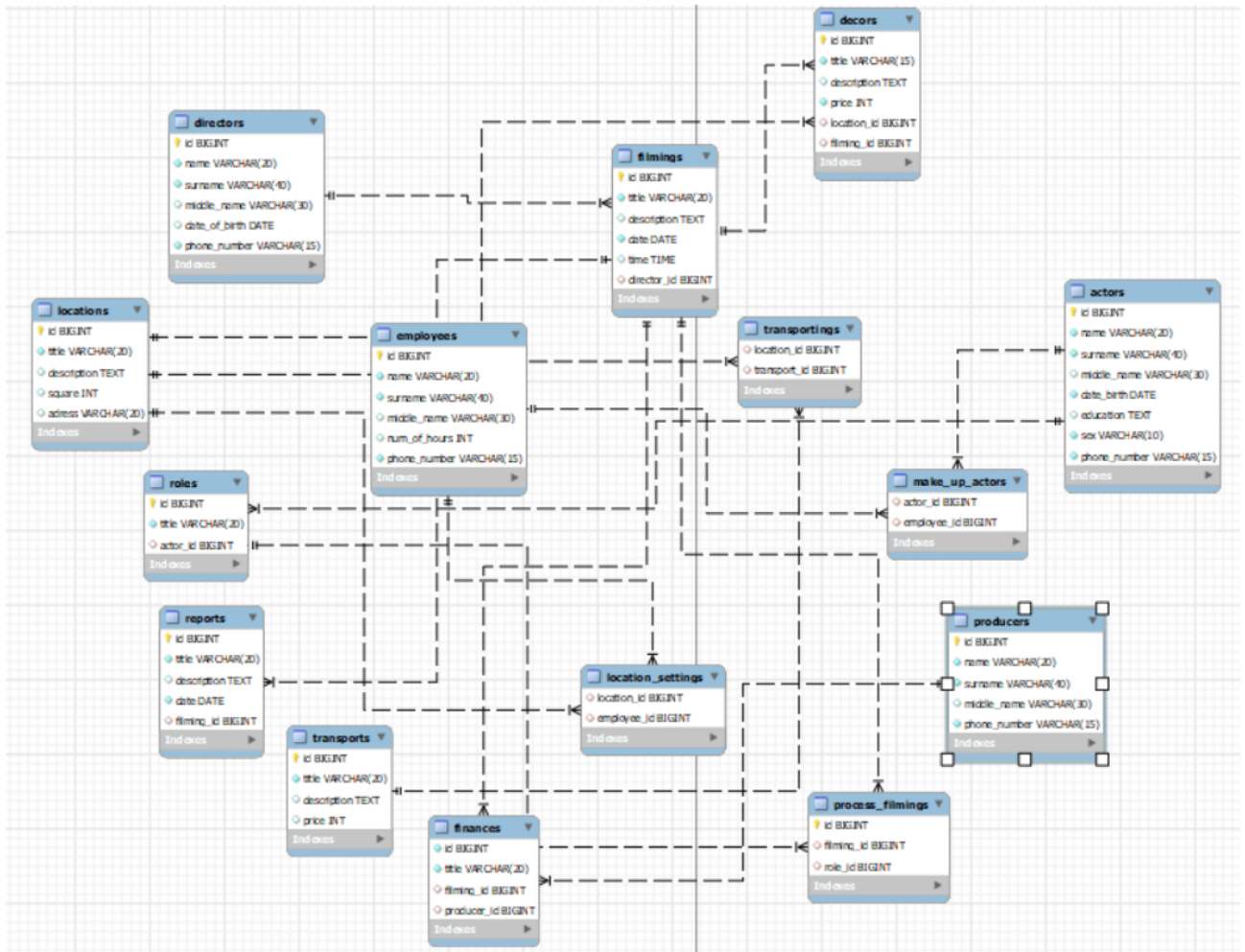


Рисунок 8.2.1 Фізична модель бази даних

Відпрацюємо скрипти у базі даних.

Зробимо вибірку на назви таблиць

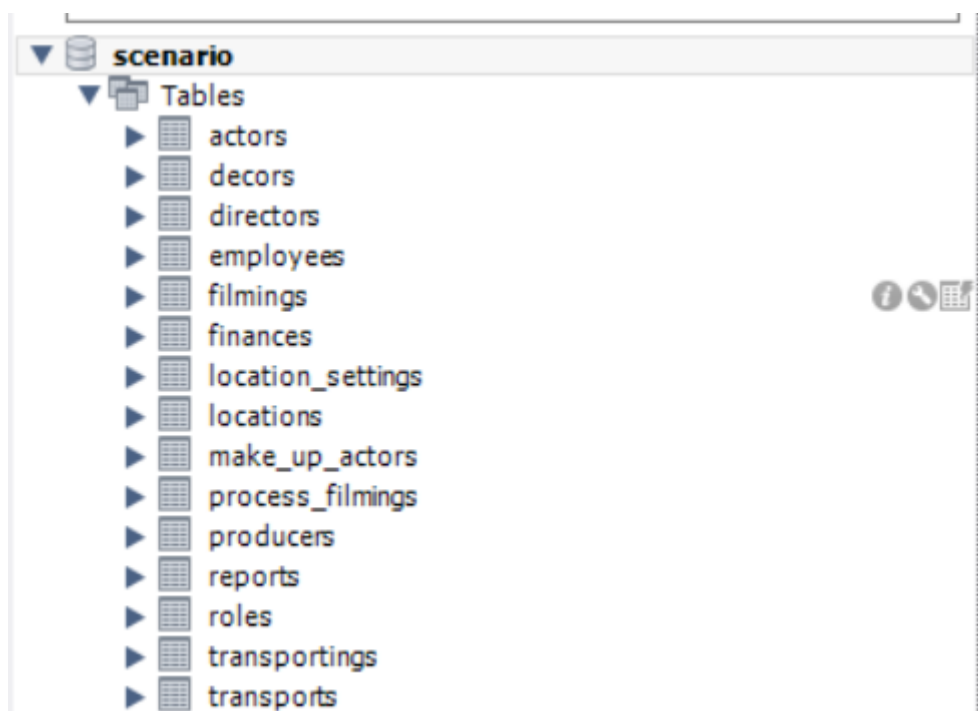


Рисунок 8.2.2 Вибірка назв таблиць

## ВИСНОВКИ

У цьому проекті були виконані всі етапи розробки бази даних, ретельно враховані вимоги та завдання. Початково була проведена аналіз предметної області, визначені ключові сутності, їх атрибути і взаємозв'язки. Інфологічна модель була розроблена з урахуванням усіх отриманих даних, інтегруючи їх в ER-діаграму.

Далі, на етапі даталогічного проектування, результати інфологічного етапу були оптимізовані і формалізовані, з урахуванням обмежень цілісності даних для конкретної системи управління базами даних (СУБД). Особливу увагу приділено нормалізації для мінімізації дублювання даних.

Обрано відповідну СУБД - MySQL Workbench 8.0 CE - з урахуванням вимог до операційної обстановки та бази даних. На останньому етапі, фізичного проектування, інформація була конвертована в SQL-скрипти для створення бази даних, таблиць, зв'язків і первинних ключів. Проведено тестування бази даних за допомогою тестових даних та попередньо визначених запитів, щоб переконатися в її ефективності та правильності роботи.

Результатом даного проекту є готова база даних, яка сприятиме оптимізації роботи компанії, що займається зйомкою фільмів. Вона відповідає вимогам предметної області і забезпечить повноцінне функціонування. Проектування бази даних вимагає системного та детального підходу на кожному етапі, забезпечуючи відповідність всім поточним вимогам. Всі етапи роботи були необхідні для створення майбутньої бази даних, що задовольняє всі вимоги і належно відображає потреби користувачів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Перелік баз даних  
<https://www.mobilepubliclibrary.org/databases/databasesatoz.php#m>
2. MySQL Workbench  
<https://www.mysql.com/products/workbench/>
3. Windows 10  
<https://learn.microsoft.com/en-us/windows/>
4. Windows 10 Information  
[https://uk.wikipedia.org/wiki/Windows\\_10](https://uk.wikipedia.org/wiki/Windows_10)
5. Робота із Workbench  
<https://javarush.com/ua/quests/lectures/ua.questhibernate.level05.lecture03>
6. MySQL Documentation  
<https://dev.mysql.com/doc/>
7. Визначення SQL  
<https://uk.wikipedia.org/wiki/SQL>
8. Програма Celtx  
<https://www.celtx.com/>
9. Процесор Intel Core I5  
[https://uk.wikipedia.org/wiki/Intel\\_Core\\_i5](https://uk.wikipedia.org/wiki/Intel_Core_i5)
10. Допоміжна інформація в проектування бази даних  
[https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%94%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F\\_%D0%B1%D0%B0%D0%B7%D0%B8\\_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%94%D0%BA%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)

## Додаток А

У мене вже був завантажений MySQL Workbench 8.0CE , тому додам створення нового зв'язку

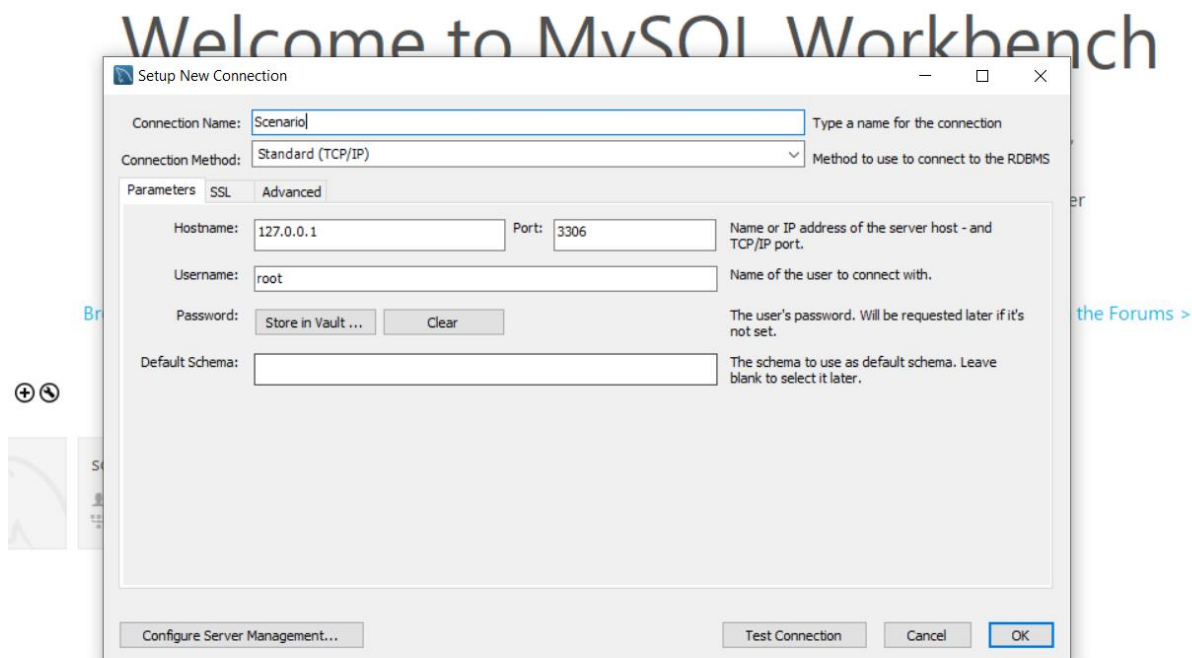


Рисунок 9.1 MySQL Workbench

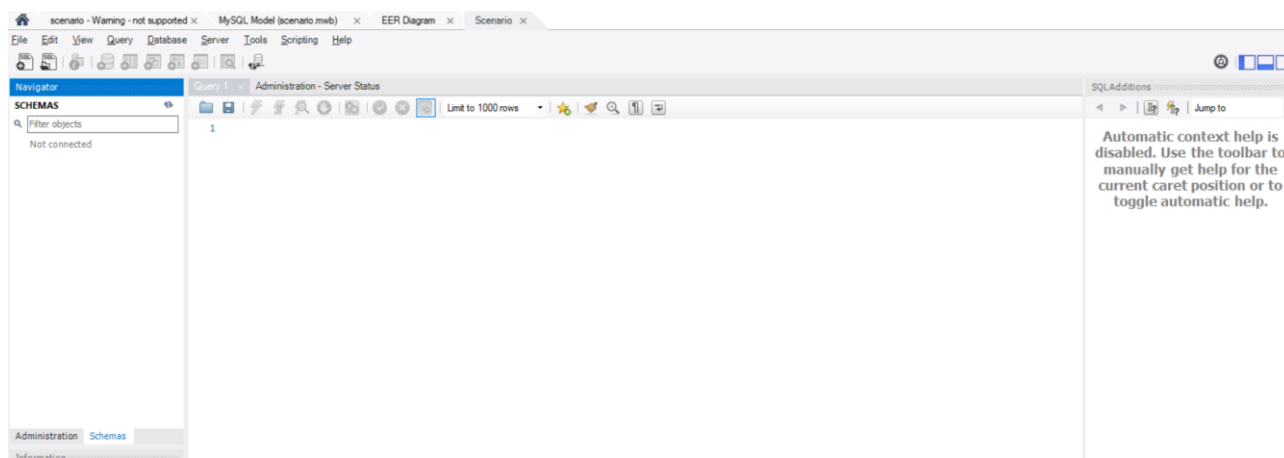


Рисунок 9.2 Створення пустої бази даних



## Додаток Б

Скрипти по створенню таблиць

Актори

```
CREATE TABLE IF NOT EXISTS actors(
    id SERIAL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    surname VARCHAR(40) NOT NULL,
    middle_name VARCHAR(30),
    date_birth DATE NOT NULL,
    education TEXT,
    sex VARCHAR(10) NOT NULL,
    phone_number VARCHAR(15) NOT NULL,
    PRIMARY KEY(id)
);
```

Рисунок 9.3 створення таблиці Акторів

Ролі

```
CREATE TABLE IF NOT EXISTS roles(
    id SERIAL AUTO_INCREMENT,
    title VARCHAR(20) NOT NULL,
    PRIMARY KEY(id),
    actor_id BIGINT UNSIGNED,
    UNIQUE(actor_id),
    CONSTRAINT fk_role_actor FOREIGN KEY(actor_id) REFERENCES actors(id)
);
ALTER TABLE roles
    DROP INDEX fk_role_actor,
    ADD UNIQUE KEY fk_role_actor (actor_id);
```

Рисунок 9.3 створення таблиці Ролей

## Працівники

```
CREATE TABLE IF NOT EXISTS employees(
    id SERIAL AUTO_INCREMENT,
    name VARCHAR(20) NOT NULL,
    surname VARCHAR(40) NOT NULL,
    middle_name VARCHAR(30),
    num_of_hours INT,
    phone_number VARCHAR(15) NOT NULL,
    PRIMARY KEY(id)
);
```

Рисунок 9.3 створення таблиці Працівників

## Гримування акторів

```
CREATE TABLE IF NOT EXISTS make_up_actors(
    actor_id BIGINT UNSIGNED,
    employee_id BIGINT UNSIGNED,
    CONSTRAINT fk_make_up_actor FOREIGN KEY(actor_id) REFERENCES actors(id),
    CONSTRAINT fk_make_up_employee FOREIGN KEY(employee_id) REFERENCES employees(id)
);
```

Рисунок 9.3 створення таблиці Гримування акторів

## Локація

```
CREATE TABLE IF NOT EXISTS locations(  
    id SERIAL AUTO_INCREMENT,  
    title VARCHAR(20) NOT NULL,  
    description TEXT,  
    square INT,  
    adress VARCHAR(20),  
    PRIMARY KEY(id)  
);
```

Рисунок 9.4 Створення таблиці Локації

## Налаштування локації

```
CREATE TABLE IF NOT EXISTS location_settings(  
    location_id BIGINT UNSIGNED,  
    employee_id BIGINT UNSIGNED,  
    CONSTRAINT fk_location_settings_location FOREIGN KEY(location_id) REFERENCES locations(id),  
    CONSTRAINT fk_location_settings_employee FOREIGN KEY (employee_id) REFERENCES employees(id)  
);
```

Рисунок 9.5 Створення таблиці Налаштування локацій

## Продюсери

```
CREATE TABLE IF NOT EXISTS producers(  
    id SERIAL AUTO_INCREMENT,  
    name VARCHAR(20) NOT NULL,  
    surname VARCHAR(40) NOT NULL,  
    middle_name VARCHAR(30),  
    phone_number VARCHAR(15) NOT NULL,  
    PRIMARY KEY(id)  
);
```

Рисунок 9.6 Створення таблиці Продюсери

## Режисери

```
CREATE TABLE IF NOT EXISTS directors(  
    id SERIAL AUTO_INCREMENT,  
    name VARCHAR(20) NOT NULL,  
    surname VARCHAR(40) NOT NULL,  
    middle_name VARCHAR(30),  
    date_of_birth DATE,  
    phone_number VARCHAR(15) NOT NULL,  
    PRIMARY KEY(id)  
);
```

Рисунок 9.7 створення таблиці Режисери

## Зйомки

```
CREATE TABLE IF NOT EXISTS filmings(  
    id SERIAL AUTO_INCREMENT,  
    title VARCHAR(20) NOT NULL,  
    description TEXT,  
    date DATE NOT NULL,  
    time TIME,  
    PRIMARY KEY(id),  
    director_id BIGINT UNSIGNED,  
    CONSTRAINT fk_filming_director FOREIGN KEY(director_id) REFERENCES directors(id)  
);
```

Рисунок 9.8 створення таблиці Зйомки

## Звіти

```
CREATE TABLE IF NOT EXISTS reports(  
    id SERIAL AUTO_INCREMENT,  
    title VARCHAR(20) NOT NULL,  
    description TEXT,  
    date DATE NOT NULL,  
    PRIMARY KEY(id),  
    filming_id BIGINT UNSIGNED,  
    CONSTRAINT fk_report_filming FOREIGN KEY(filming_id) REFERENCES filmings(id)  
);
```

Рисунок 9.9 створення таблиці Звіти

## Фінанси

```

> CREATE TABLE IF NOT EXISTS finances(
    id SERIAL AUTO_INCREMENT,
    title VARCHAR(20) NOT NULL,
    filming_id BIGINT UNSIGNED,
    producer_id BIGINT UNSIGNED,
    CONSTRAINT fk_finance_filming FOREIGN KEY(filming_id) REFERENCES filmings(id),
    CONSTRAINT fk_finance_producer FOREIGN KEY(producer_id) REFERENCES producers(id)
);

```

Рисунок 9.10 створення таблиці Фінанси

## Процеси зйомок

---

```

CREATE TABLE IF NOT EXISTS process_filming(
    id SERIAL AUTO_INCREMENT,
    PRIMARY KEY(id),
    filming_id BIGINT UNSIGNED,
    role_id BIGINT UNSIGNED,
    CONSTRAINT fk_process_filming_filming FOREIGN KEY(filming_id) REFERENCES filmings(id),
    CONSTRAINT fk_process_filming_role FOREIGN KEY (role_id) REFERENCES roles(id)
);

```

Рисунок 9.11 створення таблиці Процеси Зйомок

## Декор

```

--
CREATE TABLE IF NOT EXISTS decors(
    id SERIAL AUTO_INCREMENT,
    PRIMARY KEY(id),
    title VARCHAR(15) NOT NULL,
    description TEXT,
    price INT NOT NULL ,
    location_id BIGINT UNSIGNED,
    filming_id BIGINT UNSIGNED,
    CONSTRAINT fk_decors_location FOREIGN KEY(location_id) REFERENCES locations(id),
    CONSTRAINT fk_decors_filming FOREIGN KEY (filming_id) REFERENCES filmings(id)
);

```

Рисунок 9.12 створення таблиці Декорів

## Транспорт

```

CREATE TABLE IF NOT EXISTS transports(
    id SERIAL AUTO_INCREMENT,
    title VARCHAR(20) NOT NULL,
    description TEXT,
    price INT,
    PRIMARY KEY(id)
);

```

Рисунок 9.13 створення таблиці Транспортів

## Транспортування

```

CREATE TABLE IF NOT EXISTS transportings(
    location_id BIGINT UNSIGNED,
    transport_id BIGINT UNSIGNED,
    CONSTRAINT fk_transportings_location FOREIGN KEY(location_id) REFERENCES locations(id),
    CONSTRAINT fk_transportings_transport_id FOREIGN KEY (trasport_id) REFERENCES transports(id)
);

```

Рисунок 9.14 створення таблиці Транспортування

## Додаток В

### Відпрацювання скриптів у СУБД

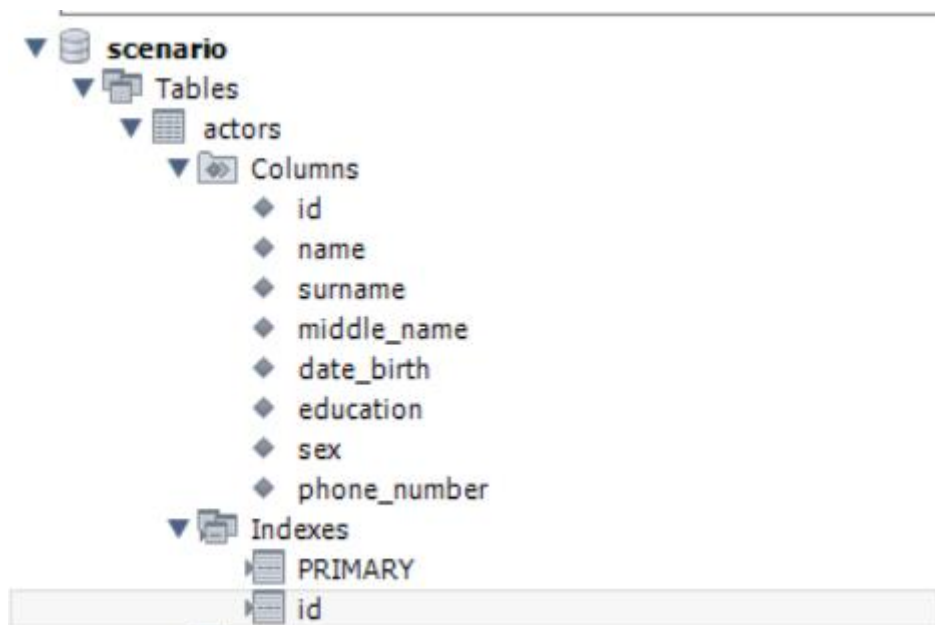


Рисунок 10.1 Таблиця Актори

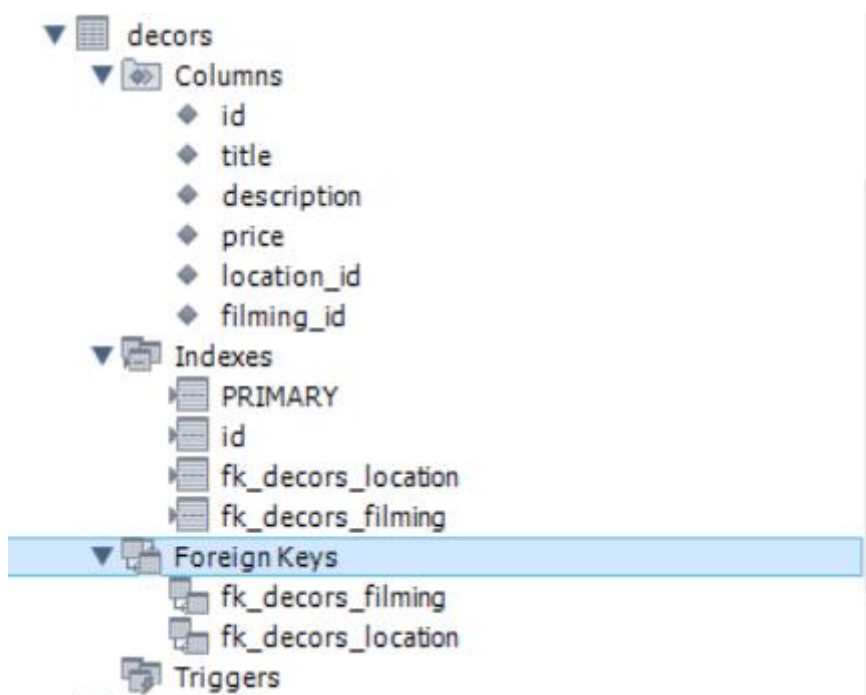


Рисунок 10.2 Таблиця Декор





Рисунок 10.3 Таблица Режисери



Рисунок 10.4 Таблица Робітники

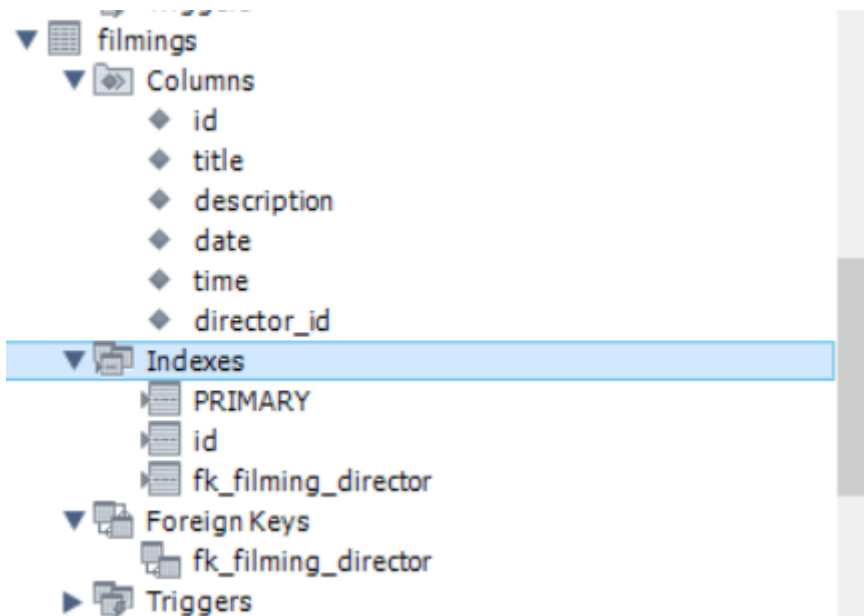


Рисунок 10.5 Таблиця Зйомки

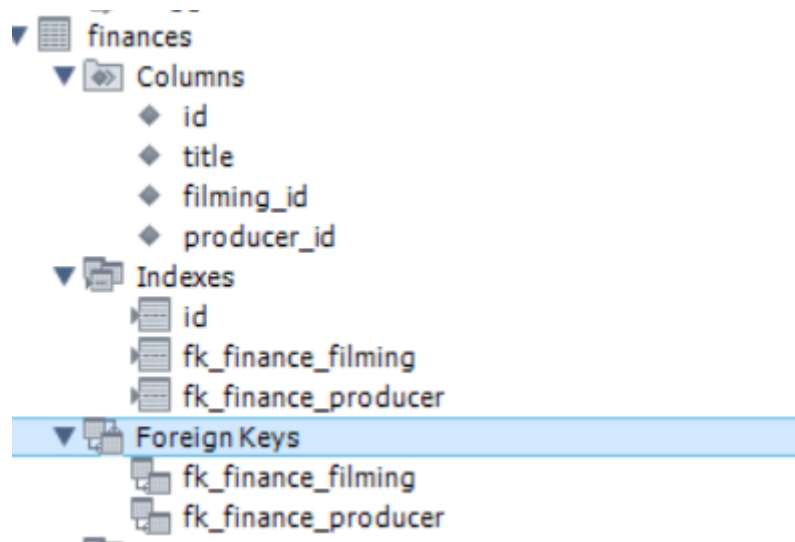


Рисунок 10.6 Таблиця Фінанси

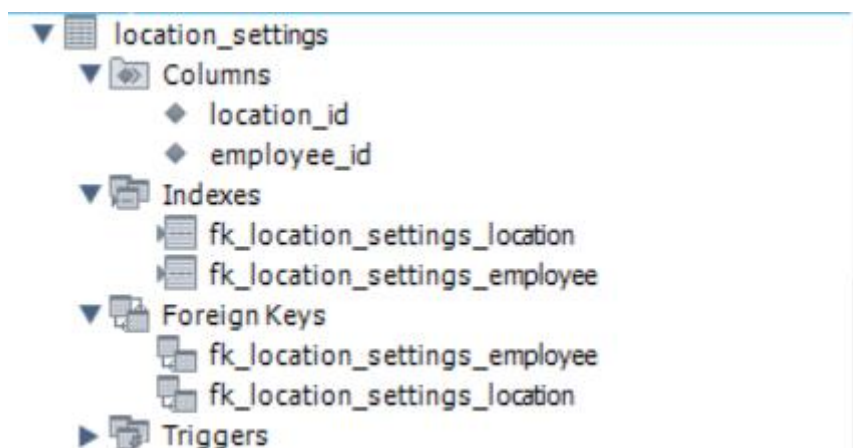


Рисунок 10.7 Таблиця Налаштування Локацій

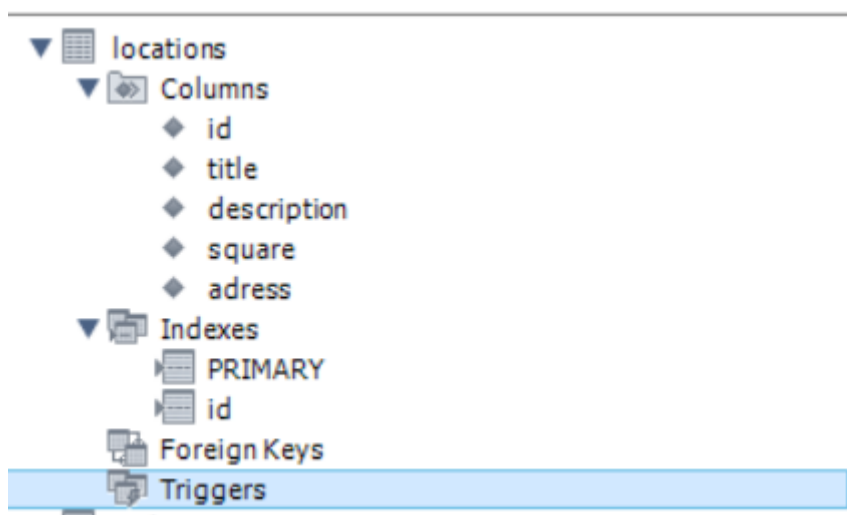


Рисунок 10.8 Таблиця Локації

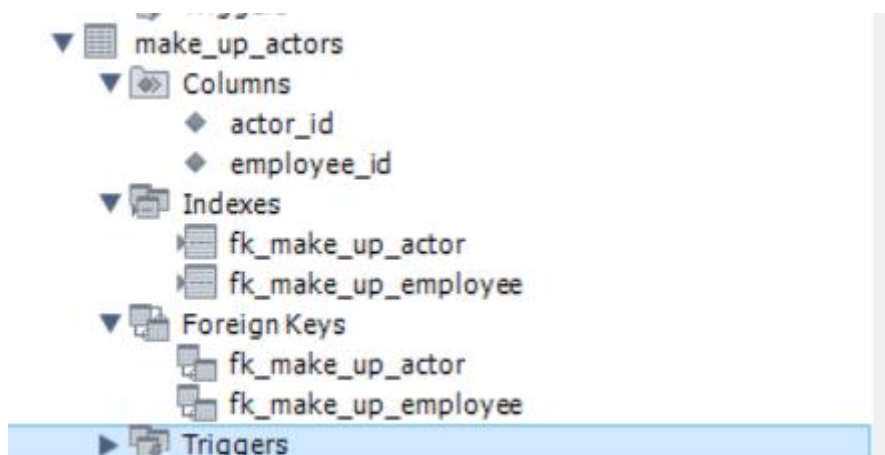


Рисунок 10.1 Таблиця Гримування Акторів

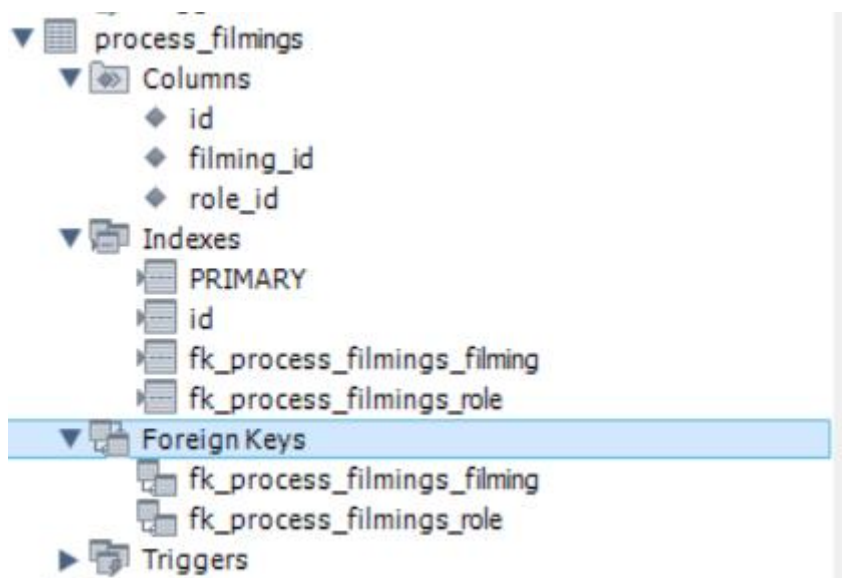


Рисунок 10.1 Таблиця Процеси Зйомок

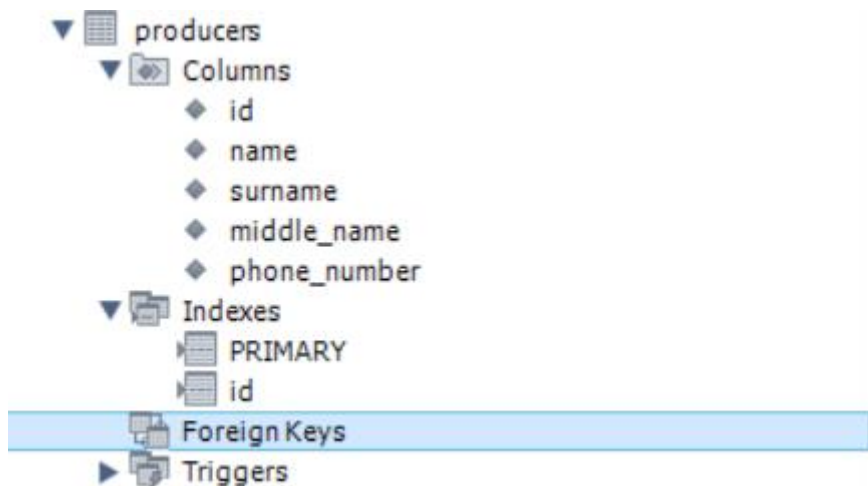


Рисунок 10.1 Таблица Продюсери

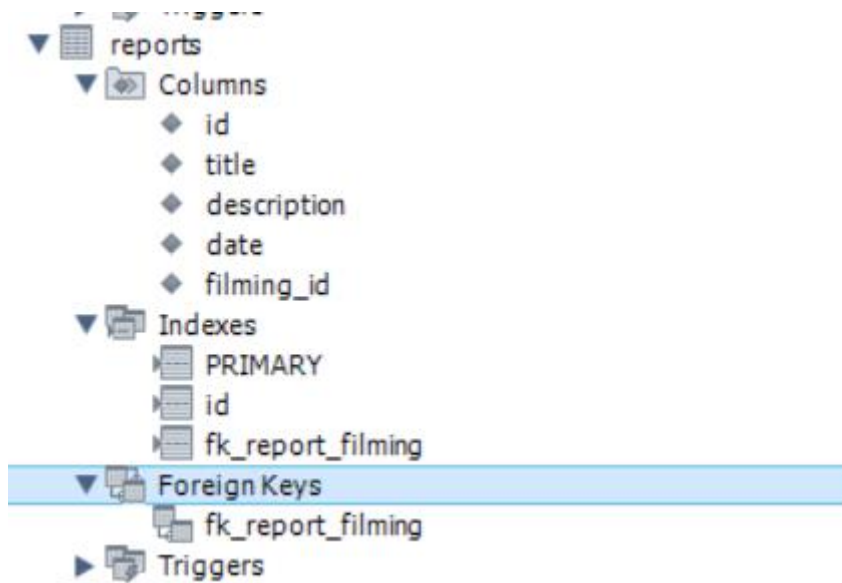


Рисунок 10.1 Таблица Звіти

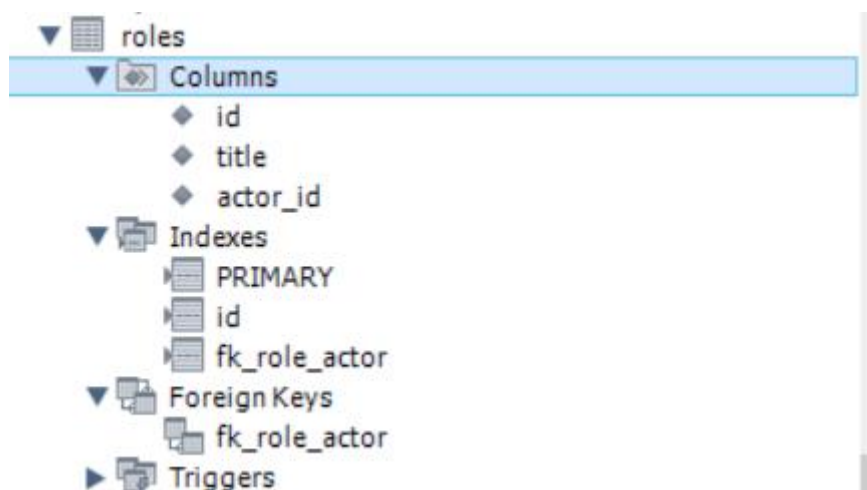


Рисунок 10.1 Таблиця Ролі

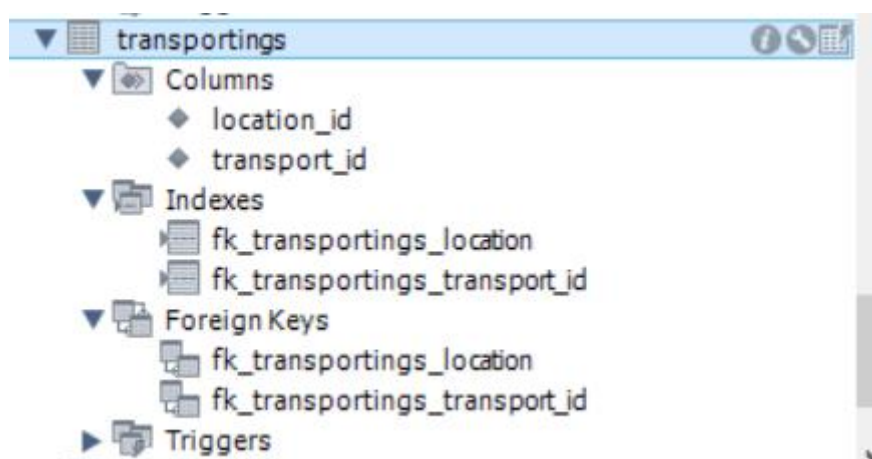


Рисунок 10.1 Таблиця Транспортування

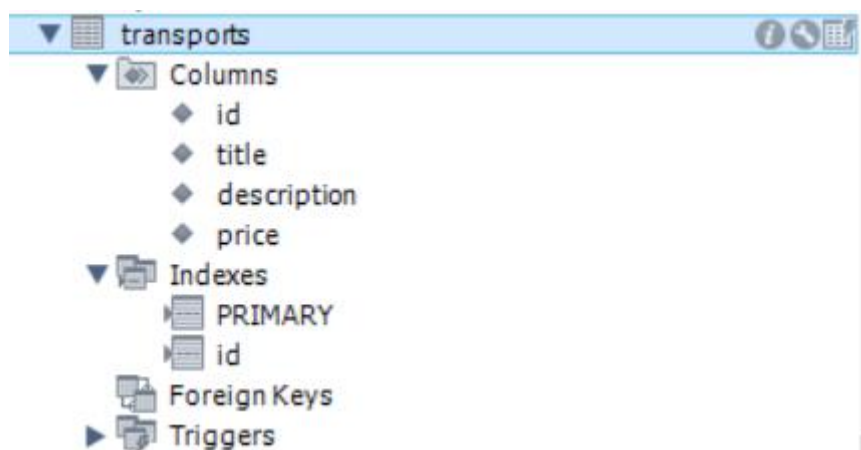


Рисунок 10.1 Таблица Транспорти

## Додаток Г

Було створено 16 прикладів різноманітних питань, на які може дати відповідь спроектована база даних:

запити на SELECT (1-8)

запити на INSERT (9-12)

запити на UPDATE (13-15)

запити на DELETE (16-17)

Перелік функцій:

1. Вивести інформацію про співробітників, які приймають участь у зйомці

```
SELECT
    pf.id AS process_filming_id,
    f.title AS filming_title,
    r.title AS role_title,
    a.name AS actor_name,
    a.surname AS actor_surname
FROM
    process_filming pf
JOIN
    roles r ON pf.role_id = r.id
JOIN
    actors a ON r.actor_id = a.id
JOIN
    filmings f ON pf.filming_id = f.id;
```

Виконаємо запит для

process\_filming\_id = 2

	process_filming_id	filming_title	role_title	actor_name	actor_surname
►	2	Drama Film	Supporting Actor	Jane	Doe



2. Вивести інформацію про заплановані зйомки, після певної дати

```

SELECT
    f.id AS filming_id,
    f.title AS filming_title,
    f.description AS filming_description,
    f.date AS filming_date,
    f.time AS filming_time,
    d.name AS director_name,
    d.surname AS director_surname
FROM
    filmings f
JOIN
    directors d ON f.director_id = d.id
WHERE
    f.date >= '2023-05-01'
ORDER BY
    f.date ASC, f.time ASC;

```

Виконаємо запит, для дати більшої за 2023-05-01

	filming_id	filming_title	filming_description	filming_date	filming_time	director_name	director_surname
►	3	Comedy Film	Hilarious comedy	2023-07-10	18:00:00	Sophie	Miller
	4	Romantic Movie	Heartwarming romance	2023-08-15	20:30:00	William	Jones

3. Зробимо запит, для отримання декору, який потрібен для зйомки

```
SELECT
    d.id AS decor_id,
    d.title AS decor_title,
    d.description AS decor_description,
    d.price AS decor_price,
    l.title AS location_title,
    l.address AS location_address,
    f.title AS filming_title,
    f.date AS filming_date
FROM decors d
JOIN locations l ON d.location_id = l.id
JOIN filmings f ON d.filming_id = f.id
WHERE f.date >= '2023-01-01'
ORDER BY f.date ASC;
```

Виконаємо для умови, що дата зйомки після «2023-01-01»

	decor_id	decor_title	decor_description	decor_price	location_title	location_address	filming_title	filming_date
►	1	City Street Set	Urban street set	5000	Studio A	123 Main St	Action Movie	2023-03-15
	2	Forest Set	Scenic forest set	7000	Outdoor Set	456 Oak St	Drama Film	2023-04-20
	3	City Street	Urban street set	5500	Indoor Set B	789 Oak St	Comedy Film	2023-07-10
	4	Mountain Set	Scenic mountain set	8000	Outdoor Set B	101 Pine St	Romantic Movie	2023-08-15

4.Зробимо запит для отримання ролей, які приймають участь в певній кількості зйомок

```
SELECT
    r.id AS role_id,
    r.title AS role_title,
    COUNT(pf.id) AS number_of_films
FROM roles r
JOIN process_filming pf ON r.id = pf.role_id
GROUP BY r.id, r.title
HAVING COUNT(pf.id) >= 3
ORDER BY number_of_films DESC;
```

Наприклад, зараз знайдемо ті ролі, які приймають участь більше ніж в 3 зйомках

	role_id	role_title	number_of_films
--	---------	------------	-----------------

Нажаль, ніхто не приймає участі більше ніж в 3 зйомках

## 5. Тепер отримаємо кошторис зйомок

```

SELECT
    f.id AS filming_id,
    f.title AS filming_title,
    SUM(fi.price) AS total_budget,
    p.name AS producer_name,
    p.surname AS producer_surname
FROM filmings f
LEFT JOIN decors d ON f.id = d.filming_id
LEFT JOIN transportings t ON f.id = t.location_id
LEFT JOIN process_filming pf ON f.id = pf.filming_id
LEFT JOIN finances fi ON f.id = fi.filming_id
LEFT JOIN producers p ON fi.producer_id = p.id
GROUP BY f.id, f.title, p.name, p.surname
ORDER BY total_budget ASC;

```

	filming_id	filming_title	total_budget	producer_name	producer_surname
►	2	Drama Film	NULL	Emily	Green
	3	Comedy Film	NULL	Emma	Harris
	4	Romantic Movie	NULL	Andrew	Clark
	1	Action Movie	10000	Michael	White
	2	Drama Film	24000	Michael	White

## 6. Отримаємо місця зйомок

```
SELECT f.id AS filming_id, f.title AS filming_title, l.title AS location_title,
       l.description AS location_description
FROM filmings f
LEFT JOIN decors d ON f.id = d.filming_id
LEFT JOIN transportings t ON f.id = t.location_id
LEFT JOIN process_filming pf ON f.id = pf.filming_id
LEFT JOIN locations l ON t.location_id = l.id
GROUP BY f.id, f.title, l.title, l.description
```

	filming_id	filming_title	location_title	location_description
►	1	Action Movie	Studio A	Large film studio
	2	Drama Film	Outdoor Set	Open-air filming location
	3	Comedy Film	Indoor Set B	Another film studio
	4	Romantic Movie	Outdoor Set B	Open-air filming location

## 7. Тепер отримаємо орендований транспорт

```
SELECT f.id AS filming_id, f.title AS filming_title, t.title AS transport_title,
       t.description AS transport_description, t.price AS transport_price
FROM filmings f
JOIN transportings tr ON f.id = tr.location_id
JOIN transports t ON tr.transport_id = t.id
WHERE t.price > 1000;
```

Зробимо запит за умови, що transport.price > 1000

	filming_id	filming_title	transport_title	transport_description	transport_price
►	3	Comedy Film	Comedy Film Bus	Transportation bus	1200
	4	Romantic Movie	Romantic Movie Limo	Luxury limousine	2000

## 8. Отримаємо які співробітники готують до зйомки яких акторів

```
SELECT e.id AS employee_id, e.name AS employee_name, e.surname AS employee_surname,
       a.id AS actor_id, a.name AS actor_name, a.surname AS actor_surname
FROM make_up_actors mu
JOIN employees e ON mu.employee_id = e.id
JOIN actors a ON mu.actor_id = a.id;
```

	employee_id	employee_name	employee_surname	actor_id	actor_name	actor_surname
►	1	Alice	Johnson	1	John	Doe
	2	David	Williams	2	Jane	Doe
	3	Eva	Brown	3	Bob	Smith
	4	Olivia	Davis	4	Michael	Smith
	5	Matthew	Wilson	5	Laura	Johnson
	6	Sophia	Taylor	6	Daniel	Brown

## 9. Додамо акторів до бд

```
INSERT INTO actors (name, surname, date_birth, sex, phone_number) VALUES
('Іван', 'Іванов', '1990-01-15', 'Чоловіча', '1234567890'),
('Марія', 'Петрова', '1985-05-20', 'Жіноча', '9876543210'),
('Петро', 'Сидоров', '1988-11-08', 'Чоловіча', '5551122334');
```

## 10. Додамо робітників

```
INSERT INTO employees (name, surname, num_of_hours, phone_number) VALUES
('Ольга', 'Коваленко', 40, '1112233445'),
('Андрій', 'Лисенко', 30, '6789012345'),
('Тетяна', 'Савченко', 35, '4567890123');
```

## 11. Додамо відповідність які співробітники готують акторів

```
INSERT INTO make_up_actors (actor_id, employee_id) VALUES
(1, 1),
(2, 2),
(3, 3);
```

12. Додамо ролі до відповідних зйомок

```
INSERT INTO process_filming (filming_id, role_id) VALUES  
(1, 1),  
(1, 2),  
(1, 3);
```

13. Оновити ціну декору

```
UPDATE decor  
SET price = 1500  
WHERE id <> 1 OR id IS NULL;
```

14. Оновити дату зйомки за назвою

```
UPDATE filmings  
SET date = '2023-01-01'  
WHERE title = 'Drama film';
```

15. Оновити транспорт у переміщенні на локацію

```
UPDATE transportings  
SET transport_id = 2  
WHERE location_id = 1;
```

16. Видалити звіти у яких назва не збігається із назвою зйомки

```
DELETE reports
FROM reports
LEFT JOIN filmings ON reports.filming_id = filmings.id
WHERE filmings.title IS NULL;
```

17. Видалити ролі, які не приймають участь в жодній зйомці

```
DELETE roles
FROM roles
LEFT JOIN process_filmings ON roles.id = process_filmings.role_id
WHERE process_filmings.role_id IS NULL;
```



## Додаток Д

Скрипти для заповнення бази даних

```
INSERT INTO actors (name, surname, date_birth, sex, phone_number) VALUES  
( 'John', 'Doe', '1990-01-01', 'Male', '123-456-7890'),  
( 'Jane', 'Doe', '1995-02-15', 'Female', '987-654-3210'),  
( 'Bob', 'Smith', '1985-05-20', 'Male', '555-123-4567');
```

```
INSERT INTO roles (title, actor_id) VALUES  
( 'Lead Actor', 1),  
( 'Supporting Actor', 2),  
( 'Extra', 3);
```

```
INSERT INTO employees (name, surname, num_of_hours, phone_number)  
VALUES  
( 'Alice', 'Johnson', 40, '111-222-3333'),  
( 'David', 'Williams', 35, '444-555-6666'),  
( 'Eva', 'Brown', 30, '777-888-9999');
```

```
INSERT INTO make_up_actors (actor_id, employee_id) VALUES  
(1, 1),  
(2, 2),  
(3, 3);
```

```
INSERT INTO locations (title, description, square, adress) VALUES  
( 'Studio A', 'Large film studio', 1000, '123 Main St'),  
( 'Outdoor Set', 'Open-air filming location', 500, '456 Oak St');
```

```
INSERT INTO location_settings (location_id, employee_id) VALUES  
(1, 1),  
(2, 2);
```

```
INSERT INTO producers (name, surname, phone_number) VALUES  
( 'Michael', 'White', '111-111-1111'),  
( 'Emily', 'Green', '222-222-2222');
```

```
INSERT INTO directors (name, surname, date_of_birth, phone_number)  
VALUES  
( 'Christopher', 'Black', '1970-12-05', '333-333-3333'),  
( 'Sophia', 'Smith', '1982-08-20', '444-444-4444');
```

```
INSERT INTO filmings (title, description, date, time, director_id) VALUES  
( 'Action Movie', 'Exciting action film', '2023-03-15', '14:00:00', 1),  
( 'Drama Film', 'Compelling drama', '2023-04-20', '16:30:00', 2);
```

```
INSERT INTO reports (title, description, date, filming_id) VALUES  
( 'Action Movie Report', 'Report for action film', '2023-03-20', 1),
```

```
('Drama Film Report', 'Report for drama', '2023-05-01', 2);
```

```
INSERT INTO finances (title, filming_id, producer_id) VALUES  
( 'Action Movie Budget', 1, 1),  
( 'Drama Film Budget', 2, 2);
```

```
INSERT INTO process_filming (filming_id, role_id) VALUES  
(1, 1),  
(2, 2),  
(2, 3);
```

```
INSERT INTO decors (title, description, price, location_id, filming_id) VALUES  
( 'City Street Set', 'Urban street set', 5000, 1, 1),  
( 'Forest Set', 'Scenic forest set', 7000, 2, 2);
```

```
INSERT INTO transports (title, description, price) VALUES  
( 'Action Movie Van', 'Transportation van', 1000),  
( 'Drama Film Car', 'Film crew car', 800);
```

```
INSERT INTO transportings (location_id, transport_id) VALUES  
(1, 1),  
(2, 2);
```

```
INSERT INTO actors (name, surname, middle_name, date_birth, education, sex,  
phone_number)
```

```
VALUES ('Іван', 'Петров', 'Олександрович', '1990-05-15', 'Вища', 'Чоловіча',  
' +380991234567');
```

```
INSERT INTO actors (name, surname, middle_name, date_birth, education, sex,  
phone_number)
```

```
VALUES ('Марія', 'Іванова', 'Андріївна', '1985-12-10', 'Вища', 'Жіноча',  
' +380997654321');
```

```
INSERT INTO roles (title, actor_id)
```

```
VALUES ('Головна роль', 8);
```

```
INSERT INTO roles (title, actor_id)
```

```
VALUES ('Другорядна роль', 9);
```

```
INSERT INTO filmings (title, description, date, time, director_id)
```

```
VALUES ('Фільм 1', 'Це перший фільм', '2023-02-20', '12:00:00', 1);
```

```
INSERT INTO filmings (title, description, date, time, director_id)
```

```
VALUES ('Фільм 2', 'Це другий фільм', '2023-03-05', '15:30:00', 1);
```

```
INSERT INTO process_filmings (filming_id, role_id)
```

```
VALUES (1, 1);
```

```
INSERT INTO process_filmings (filming_id, role_id)
```

```
VALUES (1, 2);
```

```
INSERT INTO reports (title, description, date, filming_id)
VALUES ('Звіт 1', 'Опис звіту 1', '2023-02-21', 1);
```

```
INSERT INTO reports (title, description, date, filming_id)
VALUES ('Звіт 2', 'Опис звіту 2', '2023-03-06', 2);
```

```
INSERT INTO actors (name, surname, middle_name, date_birth, education, sex,
phone_number)
VALUES ('Олена', 'Сидорова', 'Василівна', '1993-08-25', 'Вища', 'Жіноча',
'+380991234567');
```

```
INSERT INTO actors (name, surname, middle_name, date_birth, education, sex,
phone_number)
VALUES ('Андрій', 'Коваленко', 'Ігорович', '1988-04-12', 'Вища', 'Чоловіча',
'+380997654321');
```

```
INSERT INTO actors (name, surname, middle_name, date_birth, education, sex,
phone_number)
VALUES ('Ірина', 'Мельник', 'Андріївна', '1995-11-03', 'Середня', 'Жіноча',
'+380993214567');
```

```
INSERT INTO roles (title, actor_id)
VALUES ('Третя роль', 10);
```

```
INSERT INTO roles (title, actor_id)
VALUES ('Додаткова роль', 11);
```

```
INSERT INTO roles (title, actor_id)
```

```
VALUES ('Роль без актора', NULL);
```

```
INSERT INTO filmings (title, description, date, time, director_id)  
VALUES ('Фільм 3', 'Це третій фільм', '2023-04-10', '18:45:00', 2);
```

```
INSERT INTO filmings (title, description, date, time, director_id)  
VALUES ('Фільм 4', 'Це четвертий фільм', '2023-05-02', '14:15:00', 2);
```

```
INSERT INTO filmings (title, description, date, time, director_id)  
VALUES ('Фільм 5', 'Це п'ятий фільм', '2023-06-20', '20:00:00', 3);
```

```
INSERT INTO process_filmings (filming_id, role_id)  
VALUES (2, 3);
```

```
INSERT INTO process_filmings (filming_id, role_id)  
VALUES (3, 1);
```

```
INSERT INTO process_filmings (filming_id, role_id)  
VALUES (3, 2);
```

```
INSERT INTO reports (title, description, date, filming_id)  
VALUES ('Звіт 3', 'Опис звіту 3', '2023-04-12', 3);
```

```
INSERT INTO reports (title, description, date, filming_id)  
VALUES ('Звіт 4', 'Опис звіту 4', '2023-05-25', 4);
```

```
INSERT INTO reports (title, description, date, filming_id)
VALUES ('Звіт 5', 'Опис звіту 5', '2023-06-22', 5);
```

```
UPDATE filmings
SET title = 'The Godfather'
WHERE id = 1;
```

```
UPDATE filmings
SET title = 'Schindler"s List'
WHERE id = 2;
```

```
UPDATE filmings
SET title = 'The Shawshank'
WHERE id = 3;
```

```
UPDATE filmings
SET title = 'The Lord'
WHERE id = 4;
```

```
UPDATE filmings
SET title = 'Titanic'
WHERE id = 5;
```

```
UPDATE filmings
SET title = 'Forrest Gump'
WHERE id = 7;
```

```
UPDATE filmings
SET title = 'The Matrix'
```

```
WHERE id = 8;
```

```
UPDATE filmings
```

```
SET title = 'Inception'
```

```
WHERE id = 9;
```

```
UPDATE filmings
```

```
SET title = 'The Dark Knight'
```

```
WHERE id = 10;
```