Національний Університет Біоресурсів і Природокористування України
Факультет інформаційних технологій

Програмування Python

Лабораторна робота №9

Виконав

Студент групи ІПЗ-23008бск

Постумент Денис Андрійович

Київ 2025

Тема: Робота з БД.

Варіант 7

1. На платформі Docker, за допомогою файлу docker-compose.yml, створити контейнер з СУБД *PostgreSQL* або *MySQL*. Зробити прокидання портів та папок для зберігання БД.
2. В цьому контейнері, використовуючи мову Python, створити базу даних. Створити в ній необхідні таблиці з відповідними полями (*предметна область та дані наведені нижче*).
3. Визначить типи даних (лічильник, текстовий, числовий тощо) та опис, якщо потрібно.
4. Встановіть необхідні властивості полів (розмір поля, маску вводу, значення за замовченням, обмеження та повідомлення про помилку) створених таблиць.
5. Визначить первинні ключі в створених таблицях.
6. Визначить необхідні зв'язки між таблицями, задайте необхідні параметри забезпечення цілісності даних.
7. Заповнить створені таблиці даними (3 кінотеатри, 11 фільмів, 15 показів)

Предметна область: Кінотеатри (*складається із 3 класів*).
Сутності та дані:
Фільми[Код фільму, назва фільму, жанр(*мелодрама, комедія, бойовик*), тривалість, рейтинг]
Кінотеатри[Код кінотеатру, назва кінотеатру, ціни на квітки, кількість місць, адреса, телефон(маска вводу)],
Транслювання фільмів[Код транслювання, Код фільму, Код кінотеатру, Дата початку показів, Термін показу(*кількість днів*)].

8. **Створіть наступні запити:**
   Відобразити всі комедії. Відсортувати фільми по рейтингу;
   Порахувати останню дату показу фільму для кожного транслювання (Вивести назву фільму, назву кінотеатру, в якому фільм транслюється, дату початку показу, термін показу та кінцеву дату показу фільму) (*запит з обчислювальним полем*);
   Порахувати суму максимального прибутку для кожного кінотеатру від одного показу (*запит з обчислювальним полем*);
   Відобразити всі фільми заданого жанру (*запит з параметром*);
   Порахувати кількість фільмів кожного жанру (*Підсумковий запит*)
   Порахувати кількість мелодрам, комедій, бойовиків, які транслюються в кожному кінотеатрі (*перехресний запит*);

9. На мові Python написати програму, що підключається до створеної БД, виводить всі таблиці (структура + дані, які в ній зберігаються) та результати виконання запитів в консоль в форматованому вигляді (заголовки стовпців + всі стовпці рівні).
10. На платформі Docker створити контейнер з графічним клієнтом (адмінка) для управління БД. Запустити його і підключитись до створеної БД. Переконатись, що всі таблиці і запити створені вірно.
11. Завантажити проект на GitHub, попередньо додавши до файлу .gitignore всі технічні папки та файли.

Хід роботи

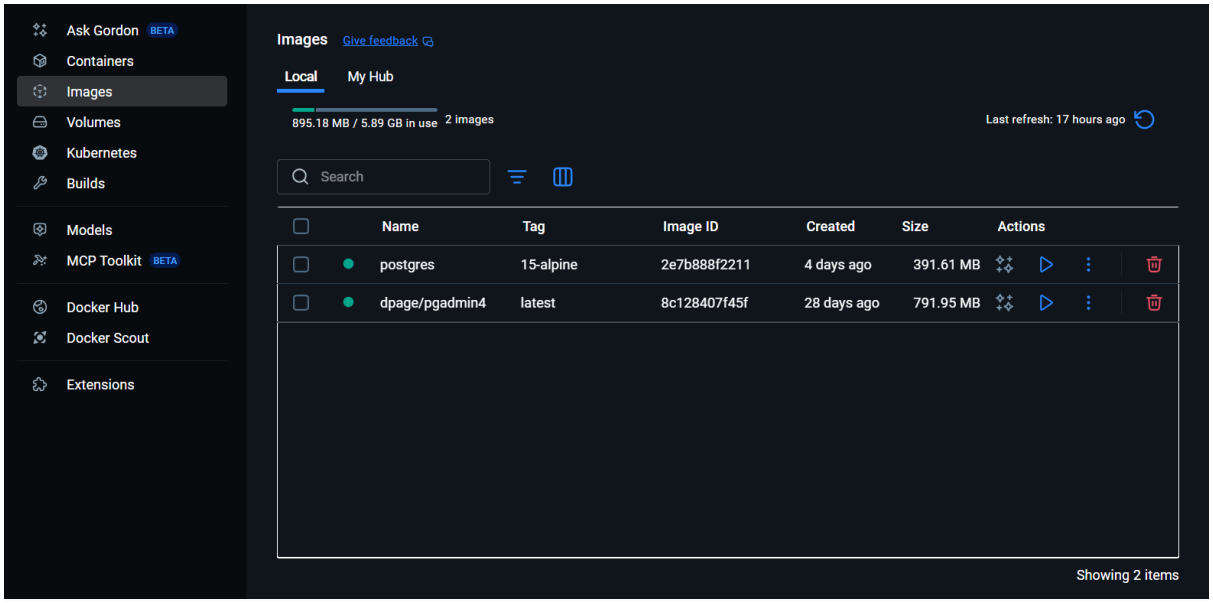


Рис. 1 – Докер образи

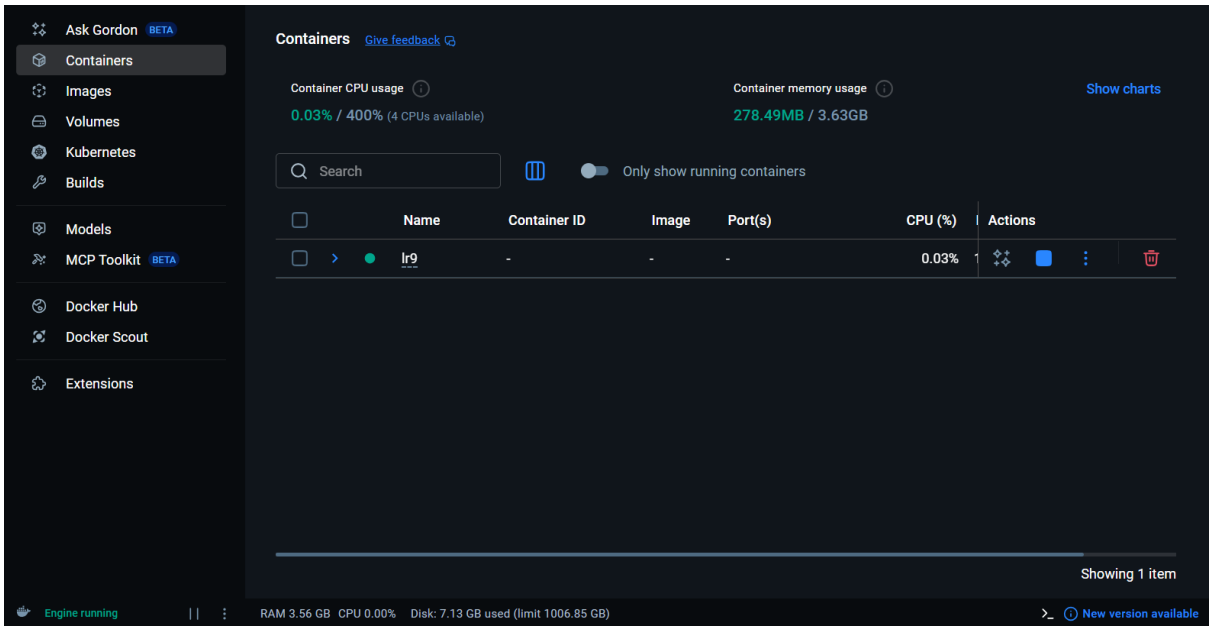


Рис. 2 – Докер контейнер

Init.sql:

-- Створення таблиці фільмів

CREATE TABLE IF NOT EXISTS movies (

```
    movie_id SERIAL PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    genre VARCHAR(50) CHECK (genre IN ('мелодрама', 'комедія', 'бойовик')),
    duration INTEGER NOT NULL CHECK (duration > 0),
    rating DECIMAL(3,1) CHECK (rating >= 0 AND rating <= 10),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Створення таблиці кінотеатрів
CREATE TABLE IF NOT EXISTS cinemas (
    cinema_id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    ticket_price DECIMAL(6,2) NOT NULL CHECK (ticket_price > 0),
    seats_count INTEGER NOT NULL CHECK (seats_count > 0),
    address VARCHAR(500) NOT NULL,
    phone VARCHAR(20) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Створення таблиці транслювання фільмів
CREATE TABLE IF NOT EXISTS screenings (
    screening_id SERIAL PRIMARY KEY,
    movie_id INTEGER NOT NULL,
```

```
    cinema_id INTEGER NOT NULL,
    start_date DATE NOT NULL,
    show_days INTEGER NOT NULL CHECK (show_days > 0),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (movie_id)
        REFERENCES movies(movie_id)
        ON DELETE CASCADE,

    FOREIGN KEY (cinema_id)
        REFERENCES cinemas(cinema_id)
        ON DELETE CASCADE
);

-- Створення індексів для покращення продуктивності
CREATE INDEX idx_movies_genre ON movies(genre);
CREATE INDEX idx_movies_rating ON movies(rating);
CREATE INDEX idx_screenings_dates ON screenings(start_date);
CREATE INDEX idx_screenings_cinema ON screenings(cinema_id);
CREATE INDEX idx_screenings_movie ON screenings(movie_id);
```

create_db.py:

```
import psycopg2
```

```
from psycopg2 import sql
import os
from dotenv import load_dotenv

load_dotenv()

def create_connection():
    """Створення з'єднання з базою даних"""
    try:
        conn = psycopg2.connect(
            host="localhost",
            port=5432,
            database="cinema_db",
            user="cinema_admin",
            password="cinema_password"
        )
        return conn
    except Exception as e:
        print(f"Помилка підключення до БД: {e}")
        return None

def create_tables(conn):
    """Створення таблиць у базі даних"""
```

```
    commands = [
        """
        CREATE TABLE IF NOT EXISTS movies (
            movie_id SERIAL PRIMARY KEY,
            title VARCHAR(255) NOT NULL,
            genre VARCHAR(50) CHECK (genre IN ('мелодрама', 'комедія', 'бойовик')),
            duration INTEGER NOT NULL CHECK (duration > 0),
            rating DECIMAL(3,1) CHECK (rating >= 0 AND rating <= 10),
            created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        )
        """,
        """
        CREATE TABLE IF NOT EXISTS cinemas (
            cinema_id SERIAL PRIMARY KEY,
            name VARCHAR(255) NOT NULL,
            ticket_price DECIMAL(6,2) NOT NULL CHECK (ticket_price > 0),
            seats_count INTEGER NOT NULL CHECK (seats_count > 0),
            address VARCHAR(500) NOT NULL,
            phone VARCHAR(20) NOT NULL,
            created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        )
        """,
        """
```

```
        CREATE TABLE IF NOT EXISTS screenings (
            screening_id SERIAL PRIMARY KEY,
            movie_id INTEGER NOT NULL,
            cinema_id INTEGER NOT NULL,
            start_date DATE NOT NULL,
            show_days INTEGER NOT NULL CHECK (show_days > 0),
            created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

            FOREIGN KEY (movie_id)
                REFERENCES movies(movie_id)
                ON DELETE CASCADE,

            FOREIGN KEY (cinema_id)
                REFERENCES cinemas(cinema_id)
                ON DELETE CASCADE
        )
        """
    ]

    try:
        cur = conn.cursor()
        for command in commands:
            cur.execute(command)
```

```
        conn.commit()
        cur.close()
        print("Таблиці успішно створені!")
    except Exception as e:
        print(f"Помилка при створенні таблиць: {e}")
        conn.rollback()

def main():
    conn = create_connection()
    if conn is not None:
        create_tables(conn)
        conn.close()
    else:
        print("Не вдалося підключитися до бази даних")

if __name__ == "__main__":
    main()
```

populate_db.py:

```
import psycopg2
from faker import Faker
import random
from datetime import datetime, timedelta
```

```
from create_db import create_connection


fake = Faker('uk_UA')


def clear_tables(conn):
    """Очищення таблиць"""
    try:
        cur = conn.cursor()
        cur.execute("DELETE FROM screenings")
        cur.execute("DELETE FROM movies")
        cur.execute("DELETE FROM cinemas")
        conn.commit()
        cur.close()
        print("Таблиці очищені!")
    except Exception as e:
        print(f"Помилка при очищенні таблиць: {e}")
        conn.rollback()


def populate_cinemas(conn):
    """Заповнення таблиці кінотеатрів"""
    cinemas_data = [
        ("Кіноман", 150.00, 300, "м. Київ, вул. Хрещатик, 25", "+380 (44) 123-45-67"),
```

```
        ("Мультиплекс", 200.00, 500, "м. Київ, вул. Велика Васильківська, 72", "+380 (44) 234-56-78"),
        ("Сінема Сіті", 180.00, 400, "м. Київ, пр-т Перемоги, 136", "+380 (44) 345-67-89"),
    ]

    try:
        cur = conn.cursor()
        for cinema in cinemas_data:
            cur.execute(
                "INSERT INTO cinemas (name, ticket_price, seats_count, address, phone) VALUES (%s, %s, %s, %s, %s)",
                cinema
            )
        conn.commit()
        cur.close()
        print("Дані кінотеатрів додано!")
    except Exception as e:
        print(f"Помилка при заповненні кінотеатрів: {e}")
        conn.rollback()


def populate_movies(conn):
    """Заповнення таблиці фільмів"""
    movies_data = [
        # Комедії
```

```
        ("Іронія долі, або З легким паром!", "комедія", 184, 8.6),
        ("Операція 'И' та інші пригоди Шурика", "комедія", 95, 8.5),
        ("Кавказька полонянка", "комедія", 82, 8.3),
        ("Брильянтова рука", "комедія", 100, 8.7),
        ("Джентельмени удачі", "комедія", 84, 8.6),

        # Мелодрами
        ("Москва сльозам не вірить", "мелодрама", 150, 8.1),
        ("Любов і голуби", "мелодрама", 107, 8.2),
        ("Службовий роман", "мелодрама", 159, 8.4),
        ("Вокзал для двох", "мелодрама", 141, 8.0),

        # Бойовики
        ("Термінатор 2: Судний день", "бойовик", 137, 8.6),
        ("Матриця", "бойовик", 136, 8.7),
        ("Леон", "бойовик", 133, 8.5),
    ]

    try:
        cur = conn.cursor()
        for movie in movies_data:
            cur.execute(
                "INSERT INTO movies (title, genre, duration, rating) VALUES (%s, %s, %s, %s)",
```

```
                movie
            )
        conn.commit()
        cur.close()
        print("Дані фільмів додано!")
    except Exception as e:
        print(f"Помилка при заповненні фільмів: {e}")
        conn.rollback()

def populate_screenings(conn):
    """Заповнення таблиці транслювання"""
    try:
        cur = conn.cursor()

        # Отримуємо всі фільми та кінотеатри
        cur.execute("SELECT movie_id FROM movies")
        movie_ids = [row[0] for row in cur.fetchall()]

        cur.execute("SELECT cinema_id FROM cinemas")
        cinema_ids = [row[0] for row in cur.fetchall()]

        # Генеруємо 15 показів
        screenings_data = []
```

```
        start_date = datetime.now().date()

        for i in range(15):
            movie_id = random.choice(movie_ids)
            cinema_id = random.choice(cinema_ids)
            show_start = start_date + timedelta(days=random.randint(0, 30))
            show_days = random.randint(3, 21)

            screenings_data.append((movie_id, cinema_id, show_start, show_days))

        for screening in screenings_data:
            cur.execute(
                "INSERT INTO screenings (movie_id, cinema_id, start_date,
show_days) VALUES (%s, %s, %s, %s)",
                screening
            )

        conn.commit()
        cur.close()
        print("Дані транслювання додано!")
    except Exception as e:
        print(f"Помилка при заповненні транслювання: {e}")
        conn.rollback()
```

```
def main():
    conn = create_connection()
    if conn is not None:
        clear_tables(conn)
        populate_cinemas(conn)
        populate_movies(conn)
        populate_screenings(conn)
        conn.close()
        print("\nБаза даних успішно заповнена тестовими даними!")
    else:
        print("Не вдалося підключитися до бази даних")

if __name__ == "__main__":
    main()
```

queries.py:

```
import psycopg2
from tabulate import tabulate
from create_db import create_connection

def print_table(conn, table_name):
    """Виведення таблиці у форматованому вигляді"""
    try:
```

```
        cur = conn.cursor()
        cur.execute(f"SELECT * FROM {table_name}")
        rows = cur.fetchall()

        # Отримуємо назви стовпців
        cur.execute(f"SELECT column_name FROM
information_schema.columns WHERE table_name = '{table_name}' ORDER
BY ordinal_position")
        columns = [col[0] for col in cur.fetchall()]

        print(f"\n{'='*80}")
        print(f"ТАБЛИЦЯ: {table_name.upper()}")
        print('='*80)
        print(tabulate(rows, headers=columns, tablefmt='grid', floatfmt=".2f"))
        print(f"Всього записів: {len(rows)}")

        cur.close()
    except Exception as e:
        print(f"Помилка при виведенні таблиці {table_name}: {e}")


def query_1(conn):
    """Всі комедії, відсортовані по рейтингу"""
    try:
        cur = conn.cursor()
```

```
        query = """
        SELECT movie_id, title, genre, duration, rating
        FROM movies
        WHERE genre = 'комедія'
        ORDER BY rating DESC
        """
        cur.execute(query)
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print("ЗАПИТ 1: Всі комедії, відсортовані по рейтингу")
        print('='*80)
        print(tabulate(rows, headers=['ID', 'Назва', 'Жанр', 'Тривалість', 'Рейтинг'], tablefmt='grid', floatfmt=".1f"))

        cur.close()
    except Exception as e:
        print(f"Помилка при виконанні запиту 1: {e}")


def query_2(conn):
    """Остання дата показу фільму для кожного транслювання"""
    try:
        cur = conn.cursor()
        query = """
```

```
        SELECT
            m.title AS Назва_фільму,
            c.name AS Кінотеатр,
            s.start_date AS Початок_показу,
            s.show_days AS Термін_показу,
            s.start_date + s.show_days AS Кінець_показу
        FROM screenings s
        JOIN movies m ON s.movie_id = m.movie_id
        JOIN cinemas c ON s.cinema_id = c.cinema_id
        ORDER BY Кінець_показу DESC
        """
        cur.execute(query)
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print("ЗАПИТ 2: Остання дата показу фільму для кожного транслювання")
        print('='*80)
        print(tabulate(rows, headers=['Назва фільму', 'Кінотеатр', 'Початок показу', 'Термін (днів)', 'Кінець показу'], tablefmt='grid'))

        cur.close()
    except Exception as e:
        print(f"Помилка при виконанні запиту 2: {e}")
```

```
def query_3(conn):
    """Максимальний прибуток для кожного кінотеатру від одного показу"""
    try:
        cur = conn.cursor()
        query = """
        SELECT
            c.name AS Кінотеатр,
            c.ticket_price AS Ціна_квитка,
            c.seats_count AS Кількість_місць,
            ROUND(c.ticket_price * c.seats_count, 2) AS Максимальний_прибуток
        FROM cinemas c
        ORDER BY Максимальний_прибуток DESC
        """
        cur.execute(query)
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print("ЗАПИТ 3: Максимальний прибуток для кожного кінотеатру від одного показу")
        print('='*80)
        print(tabulate(rows, headers=['Кінотеатр', 'Ціна квитка', 'Кількість місць', 'Максимальний прибуток'], tablefmt='grid', floatfmt=".2f"))
```

```
        cur.close()
    except Exception as e:
        print(f"Помилка при виконанні запиту 3: {e}")


def query_4(conn, genre):
    """Всі фільми заданого жанру"""
    try:
        cur = conn.cursor()
        query = """
        SELECT movie_id, title, genre, duration, rating
        FROM movies
        WHERE genre = %s
        ORDER BY title
        """
        cur.execute(query, (genre,))
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print(f"ЗАПИТ 4: Всі фільми жанру '{genre}'")
        print('='*80)
        print(tabulate(rows, headers=['ID', 'Назва', 'Жанр', 'Тривалість', 'Рейтинг'], tablefmt='grid', floatfmt=".1f"))
```

```
        cur.close()
    except Exception as e:
        print(f"Помилка при виконанні запиту 4: {e}")


def query_5(conn):
    """Кількість фільмів кожного жанру"""
    try:
        cur = conn.cursor()
        query = """
        SELECT
            genre AS Жанр,
            COUNT(*) AS Кількість_фільмів,
            AVG(rating) AS Середній_рейтинг,
            AVG(duration) AS Середня_тривалість
        FROM movies
        GROUP BY genre
        ORDER BY Кількість_фільмів DESC
        """
        cur.execute(query)
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print("ЗАПИТ 5: Кількість фільмів кожного жанру")
```

```
        print('='*80)

        print(tabulate(rows, headers=['Жанр', 'Кількість фільмів', 'Середній рейтинг', 'Середня тривалість'], tablefmt='grid', floatfmt=".1f"))


        cur.close()

    except Exception as e:

        print(f"Помилка при виконанні запиту 5: {e}")


def query_6(conn):

    """Кількість фільмів кожного жанру в кожному кінотеатрі (перехресний запит)"""

    try:

        cur = conn.cursor()

        query = """

        SELECT

            c.name AS Кінотеатр,

            COUNT(CASE WHEN m.genre = 'мелодрама' THEN 1 END) AS Мелодрами,

            COUNT(CASE WHEN m.genre = 'комедія' THEN 1 END) AS Комедії,

            COUNT(CASE WHEN m.genre = 'бойовик' THEN 1 END) AS Бойовики,

            COUNT(*) AS Всього_фільмів

        FROM screenings s

        JOIN movies m ON s.movie_id = m.movie_id
```

```
        JOIN cinemas c ON s.cinema_id = c.cinema_id
        GROUP BY c.cinema_id, c.name
        ORDER BY c.name
        """
        cur.execute(query)
        rows = cur.fetchall()

        print(f"\n{'='*80}")
        print("ЗАПИТ 6: Кількість фільмів кожного жанру в кожному кінотеатрі")
        print('='*80)
        print(tabulate(rows, headers=['Кінотеатр', 'Мелодрами', 'Комедії', 'Бойовики', 'Всього фільмів'], tablefmt='grid'))

        cur.close()
    except Exception as e:
        print(f"Помилка при виконанні запиту 6: {e}")


def main():
    conn = create_connection()
    if conn is not None:
        print("\n" + "="*80)
        print("ЛАБОРАТОРНА РОБОТА №9: СИСТЕМА УПРАВЛІННЯ КІНОТЕАТРАМИ")
        print("="*80)
```

```
    # Виведення всіх таблиць
    print("\n" + "="*80)
    print("ВСІ ТАБЛИЦІ БАЗИ ДАНИХ:")
    print("="*80)

    print_table(conn, "movies")
    print_table(conn, "cinemas")
    print_table(conn, "screenings")

    # Виконання запитів
    print("\n" + "="*80)
    print("РЕЗУЛЬТАТИ ВИКОНАННЯ ЗАПИТІВ:")
    print("="*80)

    query_1(conn)  # Всі комедії, відсортовані по рейтингу
    query_2(conn)  # Остання дата показу
    query_3(conn)  # Максимальний прибуток
    query_4(conn, "мелодрама")  # Фільми заданого жанру
    query_5(conn)  # Кількість фільмів кожного жанру
    query_6(conn)  # Перехресний запит

    conn.close()
```

```
    else:
        print("Не вдалося підключитися до бази даних")


if __name__ == "__main__":
    main()
```

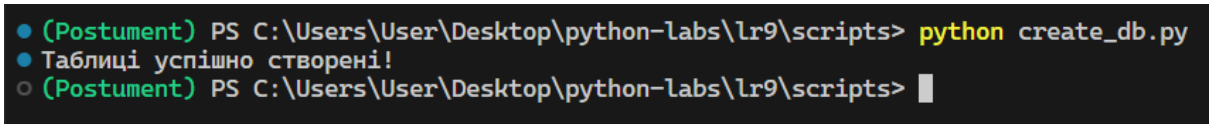


Рис. 3 – Результат виконання програми create_db.py (створення таблиць)

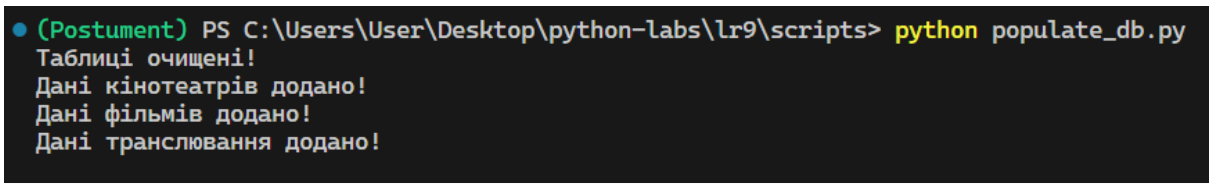
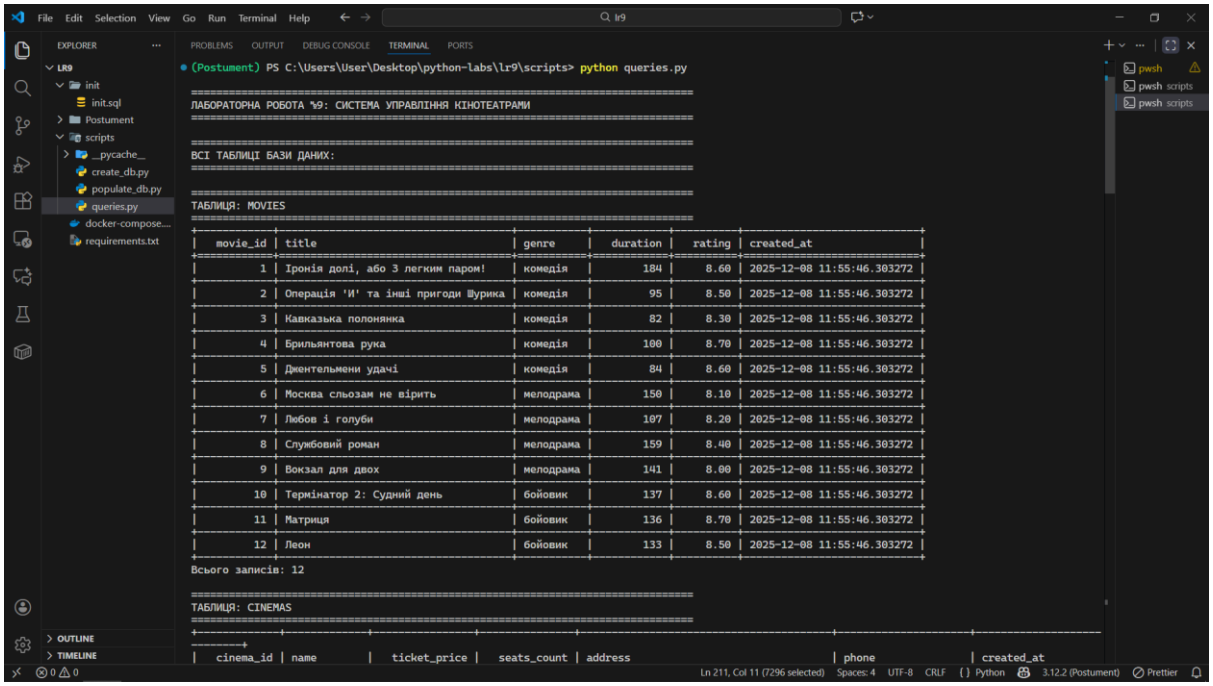
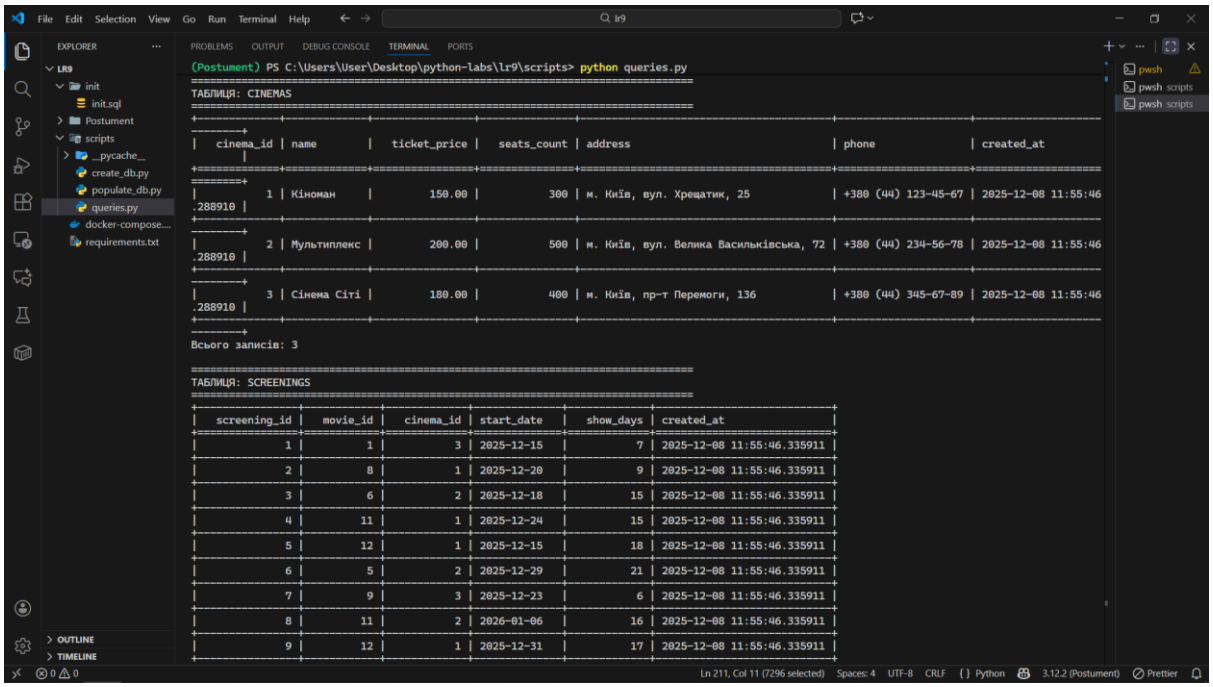


Рис. 4 – Результат виконання програми populate_db.py (наповнення таблиць даними)



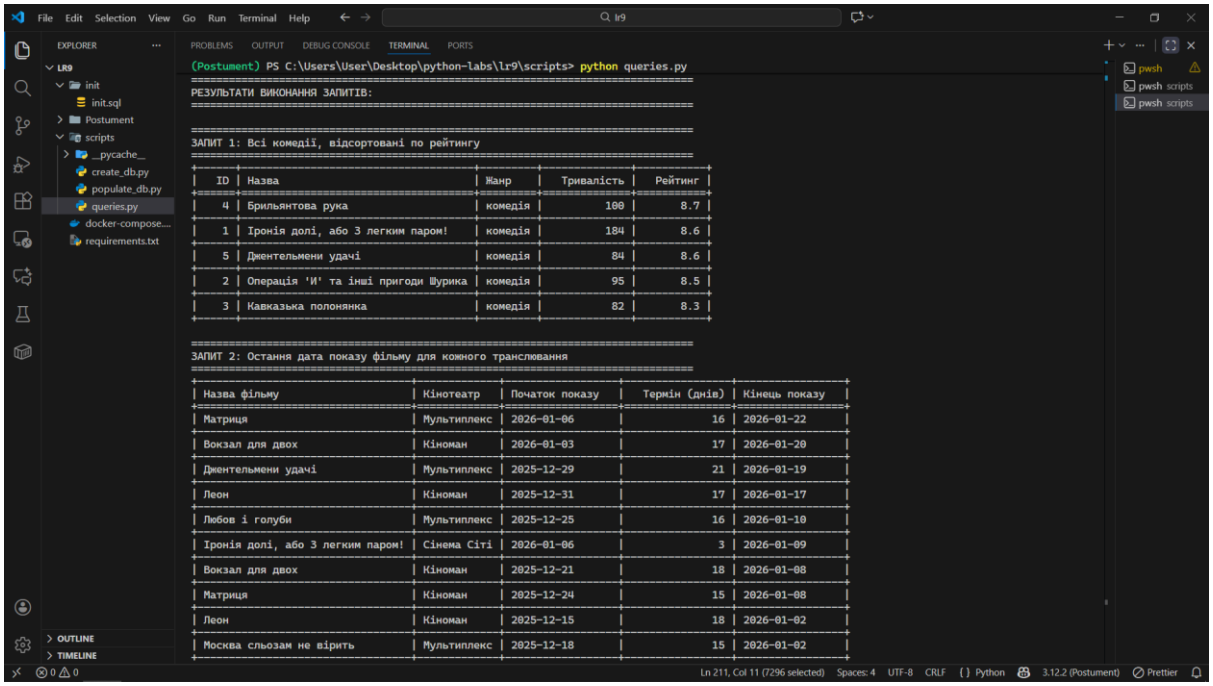
```
(Postument) PS C:\Users\User\Desktop\python-labs\lr9\scripts> python queries.py

================================================================================
ЛАБОРАТОРНА РОБОТА №9: СИСТЕМА УПРАВЛІННЯ КІНОТЕАТРАМИ
================================================================================

================================================================================
ВСІ ТАБЛИЦІ БАЗИ ДАНИХ:
================================================================================

================================================================================
ТАБЛИЦЯ: MOVIES
================================================================================
+----------+----------------------------------+----------+----------+--------+----------------------------+
| movie_id | title                            | genre    | duration | rating | created_at                 |
+==========+==================================+==========+==========+========+============================+
|        1 | Іронія долі, або З легким паром! | комедія  |      184 |   8.60 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        2 | Операція 'И' та інші пригоди Шурика | комедія  |       95 |   8.50 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        3 | Кавказька полонянка              | комедія  |       82 |   8.30 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        4 | Брильянтова рука                 | комедія  |      100 |   8.70 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        5 | Джентельмени удачі               | комедія  |       84 |   8.60 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        6 | Москва сльозам не вірить         | мелодрама |      150 |   8.10 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        7 | Любов і голуби                   | мелодрама |      107 |   8.20 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        8 | Службовий роман                  | мелодрама |      159 |   8.40 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|        9 | Вокзал для двох                  | мелодрама |      141 |   8.00 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|       10 | Термінатор 2: Судний день        | бойовик  |      137 |   8.60 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|       11 | Матриця                          | бойовик  |      136 |   8.70 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
|       12 | Леон                             | бойовик  |      133 |   8.50 | 2025-12-08 11:55:46.303272 |
+----------+----------------------------------+----------+----------+--------+----------------------------+
Всього записів: 12

================================================================================
ТАБЛИЦЯ: CINEMAS
================================================================================
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
| cinema_id  | name         | ticket_price | seats_count  | address                           | phone           | created_at                 |
+============+==============+==============+==============+===================================+=================+============================+
|          1 | Кіноман      |       150.00 |          300 | м. Київ, вул. Хрещатик, 25        | +380 (44) 123-45-67 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
|          2 | Мультиплекс  |       200.00 |          500 | м. Київ, вул. Велика Васильківська, 72 | +380 (44) 234-56-78 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
|          3 | Сінема Сіті  |       180.00 |          400 | м. Київ, пр-т Перемоги, 136       | +380 (44) 345-67-89 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
Всього записів: 3

================================================================================
ТАБЛИЦЯ: SCREENINGS
================================================================================
+--------------+------------+------------+------------+-----------+----------------------------+
| screening_id | movie_id   | cinema_id  | start_date | show_days | created_at                 |
+==============+============+============+============+===========+============================+
|            1 |          1 |          3 | 2025-12-15 |         7 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            2 |          8 |          1 | 2025-12-20 |         9 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            3 |          6 |          2 | 2025-12-18 |        15 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            4 |         11 |          1 | 2025-12-24 |        15 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            5 |         12 |          1 | 2025-12-15 |        18 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            6 |          5 |          2 | 2025-12-29 |        21 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            7 |          9 |          3 | 2025-12-23 |         6 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            8 |         11 |          2 | 2026-01-06 |        16 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            9 |         12 |          1 | 2025-12-31 |        17 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
```

Рис. 5 – Результат виконання програми queries.py (виведення запитів у форматованому табличному вигляді у термінал)



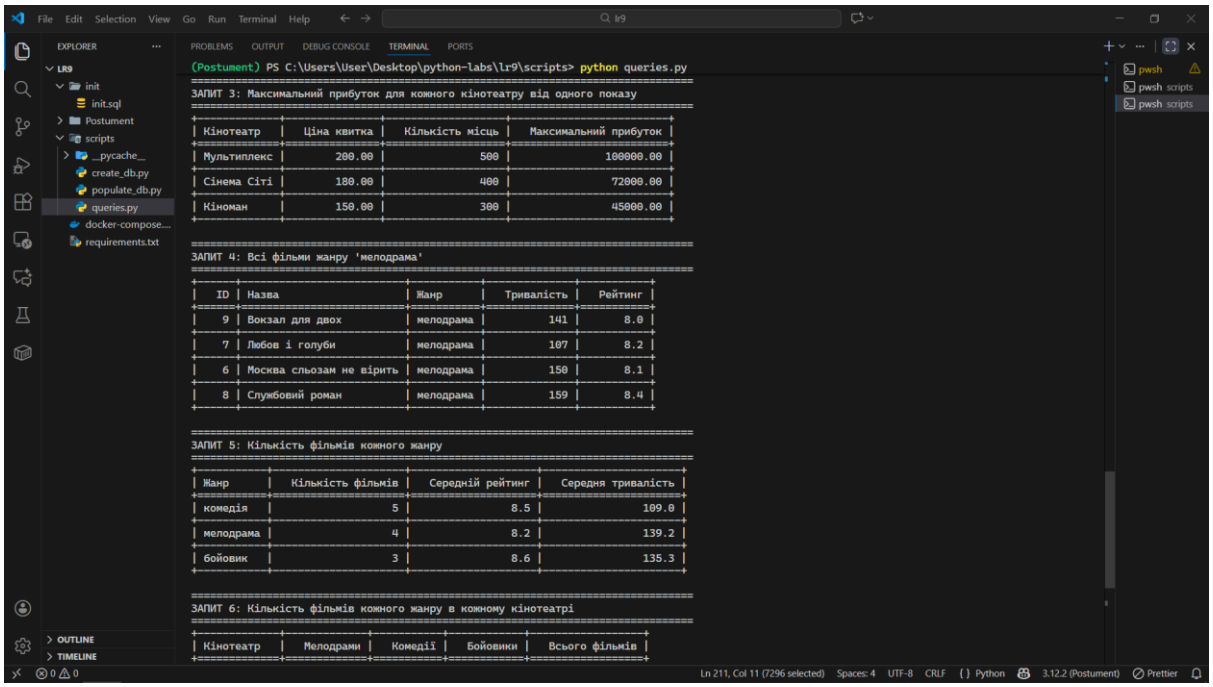

```
(Postument) PS C:\Users\User\Desktop\python-labs\lr9\scripts> python queries.py
================================================================================
ТАБЛИЦЯ: CINEMAS
================================================================================
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
| cinema_id  | name         | ticket_price | seats_count  | address                           | phone           | created_at                 |
+============+==============+==============+==============+===================================+=================+============================+
|          1 | Кіноман      |       150.00 |          300 | м. Київ, вул. Хрещатик, 25        | +380 (44) 123-45-67 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
|          2 | Мультиплекс  |       200.00 |          500 | м. Київ, вул. Велика Васильківська, 72 | +380 (44) 234-56-78 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
|          3 | Сінема Сіті  |       180.00 |          400 | м. Київ, пр-т Перемоги, 136       | +380 (44) 345-67-89 | 2025-12-08 11:55:46.288910 |
+------------+--------------+--------------+--------------+-----------------------------------+-----------------+----------------------------+
Всього записів: 3

================================================================================
ТАБЛИЦЯ: SCREENINGS
================================================================================
+--------------+------------+------------+------------+-----------+----------------------------+
| screening_id | movie_id   | cinema_id  | start_date | show_days | created_at                 |
+==============+============+============+============+===========+============================+
|            1 |          1 |          3 | 2025-12-15 |         7 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            2 |          8 |          1 | 2025-12-20 |         9 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            3 |          6 |          2 | 2025-12-18 |        15 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            4 |         11 |          1 | 2025-12-24 |        15 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            5 |         12 |          1 | 2025-12-15 |        18 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            6 |          5 |          2 | 2025-12-29 |        21 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            7 |          9 |          3 | 2025-12-23 |         6 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            8 |         11 |          2 | 2026-01-06 |        16 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
|            9 |         12 |          1 | 2025-12-31 |        17 | 2025-12-08 11:55:46.335911 |
+--------------+------------+------------+------------+-----------+----------------------------+
```


Рис. 6 – Продовження виведення результату виконання запитів



```
(Postument) PS C:\Users\User\Desktop\python-labs\lr9\scripts> python queries.py
================================================================================
РЕЗУЛЬТАТИ ВИКОНАННЯ ЗАПИТІВ:
================================================================================

================================================================================
ЗАПИТ 1: Всі комедії, відсортовані по рейтингу
================================================================================
+------+---------------------------------+---------+--------------+-----------+
|   ID | Назва                           | Жанр    |   Тривалість |   Рейтинг |
+======+=================================+=========+==============+===========+
|    4 | Брильянтова рука                | комедія |          100 |       8.7 |
+------+---------------------------------+---------+--------------+-----------+
|    1 | Іронія долі, або З легким паром! | комедія |          184 |       8.6 |
+------+---------------------------------+---------+--------------+-----------+
|    5 | Джентельмени удачі              | комедія |           84 |       8.6 |
+------+---------------------------------+---------+--------------+-----------+
|    2 | Операція 'И' та інші пригоди Шурика | комедія |           95 |       8.5 |
+------+---------------------------------+---------+--------------+-----------+
|    3 | Кавказька полонянка             | комедія |           82 |       8.3 |
+------+---------------------------------+---------+--------------+-----------+

================================================================================
ЗАПИТ 2: Остання дата показу фільму для кожного транслювання
================================================================================
+-------------------------------+-------------+----------------+-----------------+----------------+
| Назва фільму                  | Кінотеатр   | Початок показу |   Термін (днів) | Кінець показу  |
+===============================+=============+================+=================+================+
| Матриця                       | Мультиплекс | 2026-01-06     |              16 | 2026-01-22     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Вокзал для двох               | Кіноман     | 2026-01-03     |              17 | 2026-01-20     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Джентельмени удачі            | Мультиплекс | 2025-12-29     |              21 | 2026-01-19     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Леон                          | Кіноман     | 2025-12-31     |              17 | 2026-01-17     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Любов і голуби                | Мультиплекс | 2025-12-25     |              16 | 2026-01-10     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Іронія долі, або З легким паром! | Сінема Сіті | 2026-01-06     |               3 | 2026-01-09     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Вокзал для двох               | Кіноман     | 2025-12-21     |              18 | 2026-01-08     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Матриця                       | Кіноман     | 2025-12-24     |              15 | 2026-01-08     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Леон                          | Кіноман     | 2025-12-15     |              18 | 2026-01-02     |
+-------------------------------+-------------+----------------+-----------------+----------------+
| Москва сльозам не вірить      | Мультиплекс | 2025-12-18     |              15 | 2026-01-02     |
+-------------------------------+-------------+----------------+-----------------+----------------+
```

Рис. 7 - Продовження виведення результату виконання запитів



```
(Postument) PS C:\Users\User\Desktop\python-labs\lr9\scripts> python queries.py
================================================================================
ЗАПИТ 3: Максимальний прибуток для кожного кінотеатру від одного показу
================================================================================
+-------------+-------------+-----------------+---------------------+
| Кінотеатр   |   Ціна квитка |   Кількість місць |   Максимальний прибуток |
+=============+=============+=================+=====================+
| Мультиплекс |      200.00 |             500 |           100000.00 |
+-------------+-------------+-----------------+---------------------+
| Сінема Сіті |      180.00 |             400 |            72000.00 |
+-------------+-------------+-----------------+---------------------+
| Кіноман     |      150.00 |             300 |            45000.00 |
+-------------+-------------+-----------------+---------------------+

================================================================================
ЗАПИТ 4: Всі фільми жанру 'мелодрама'
================================================================================
+------+-------------------------+-----------+--------------+-----------+
|   ID | Назва                   | Жанр      |   Тривалість |   Рейтинг |
+======+=========================+===========+==============+===========+
|    9 | Вокзал для двох         | мелодрама |          141 |       8.0 |
+------+-------------------------+-----------+--------------+-----------+
|    7 | Любов і голуби          | мелодрама |          107 |       8.2 |
+------+-------------------------+-----------+--------------+-----------+
|    6 | Москва сльозам не вірить | мелодрама |          150 |       8.1 |
+------+-------------------------+-----------+--------------+-----------+
|    8 | Службовий роман         | мелодрама |          159 |       8.4 |
+------+-------------------------+-----------+--------------+-----------+

================================================================================
ЗАПИТ 5: Кількість фільмів кожного жанру
================================================================================
+-----------+---------------------+---------------------+----------------------+
| Жанр      |   Кількість фільмів |   Середній рейтинг |   Середня тривалість |
+===========+=====================+=====================+======================+
| комедія   |                   5 |                 8.5 |                109.0 |
+-----------+---------------------+---------------------+----------------------+
| мелодрама |                   4 |                 8.2 |                139.2 |
+-----------+---------------------+---------------------+----------------------+
| бойовик   |                   3 |                 8.6 |                135.3 |
+-----------+---------------------+---------------------+----------------------+

================================================================================
ЗАПИТ 6: Кількість фільмів кожного жанру в кожному кінотеатрі
================================================================================
+-------------+-------------+-----------+-----------+-----------------+
| Кінотеатр   |   Мелодрами |   Комедії |   Бойовики |   Всього фільмів |
+=============+=============+===========+===========+=================+
```

Рис. 8 - Продовження виведення результату виконання запитів

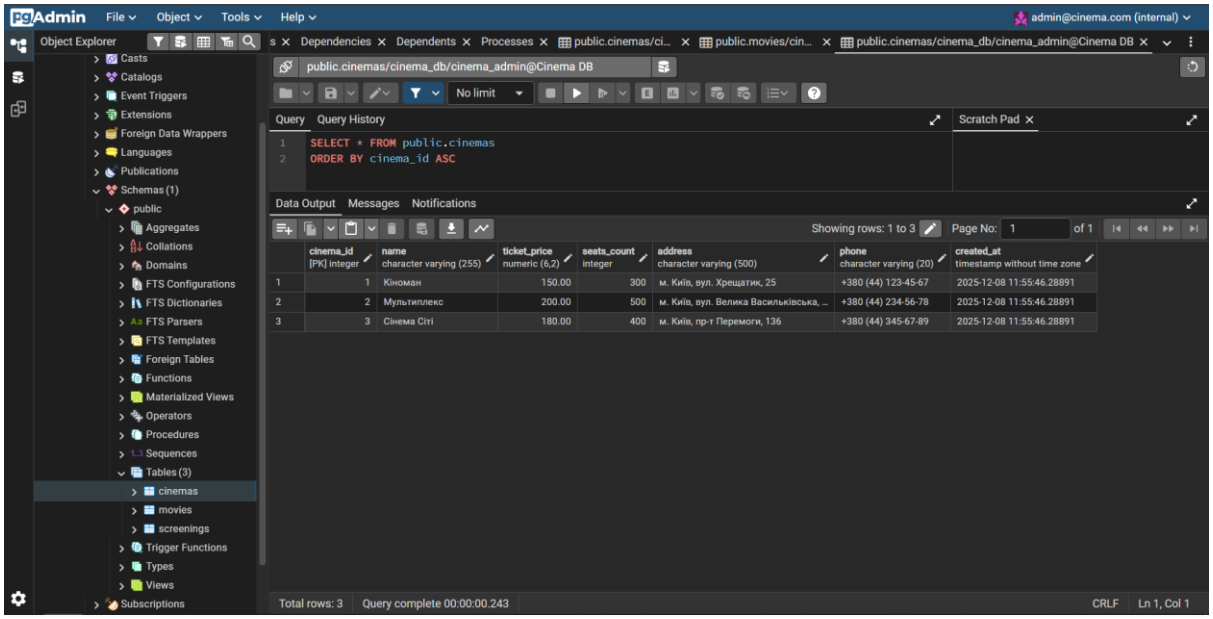


Рис. 9 – Таблиця «cinemas» в графічному клієнті БД

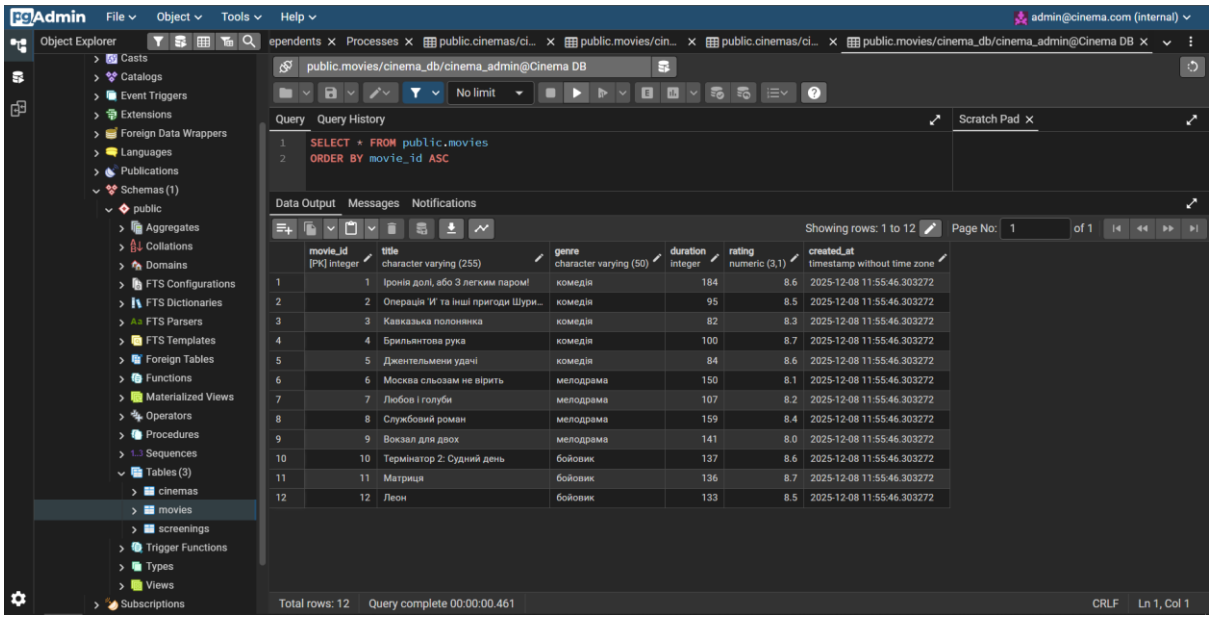


Рис. 10 – Таблиця «movies» в графічному клієнті БД

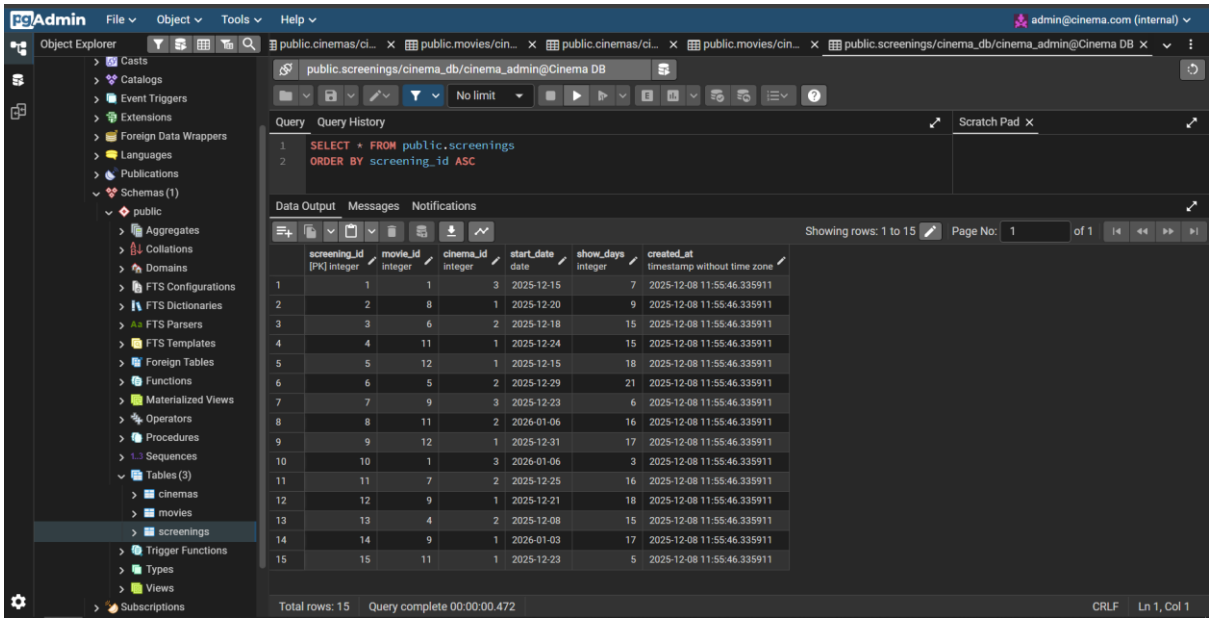


Рис. 11 – Таблиця «screenings» в графічному клієнті БД

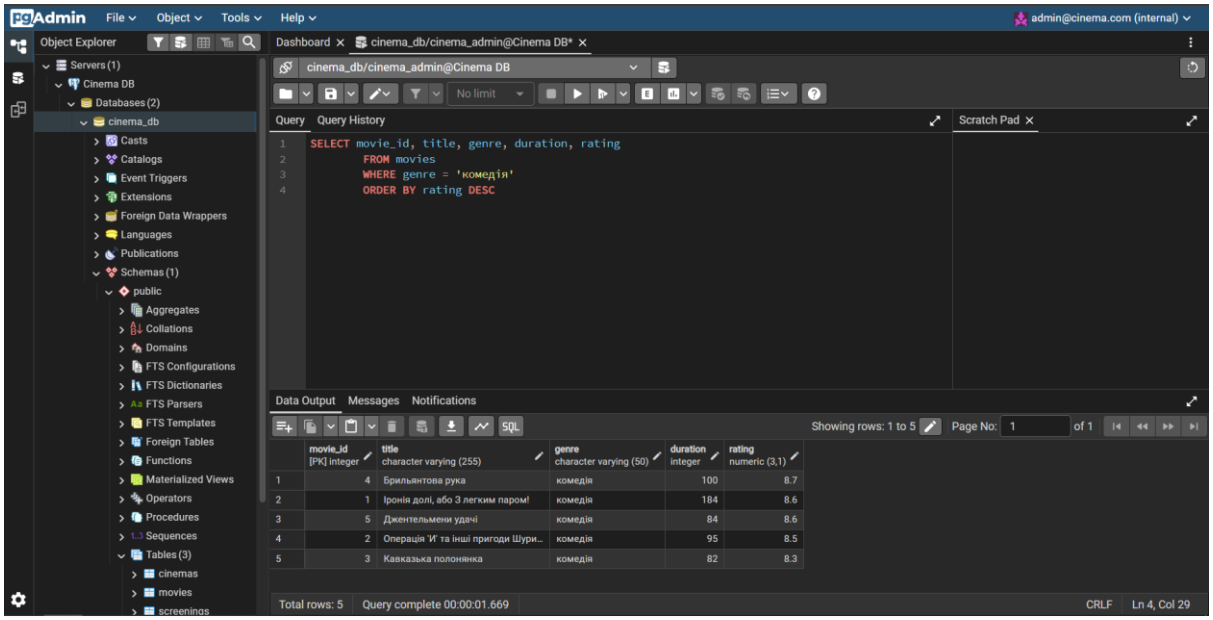


Рис. 12 – Запит «Відобразити всі комедії. Відсортувати фільми по рейтингу»

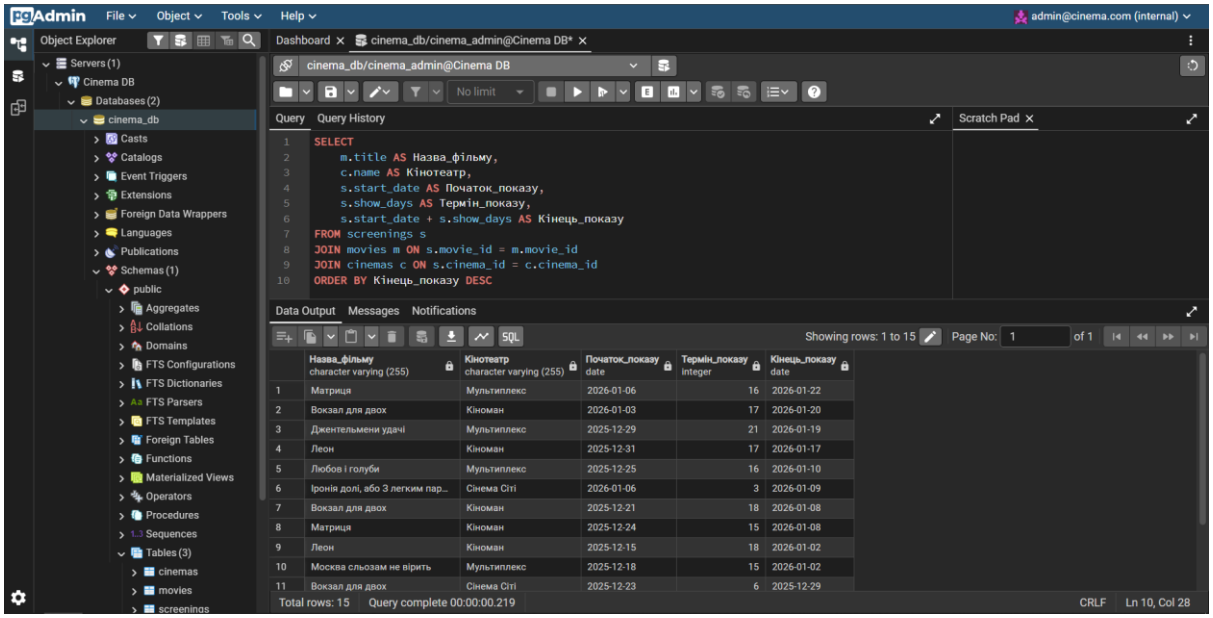


Рис. 13 – Запит «Порахувати останню дату показу фільму для кожного транслювання (Вивести назву фільму, назву кінотеатру, в якому фільм транслюється, дату початку показу, термін показу та кінцеву дату показу фільму»

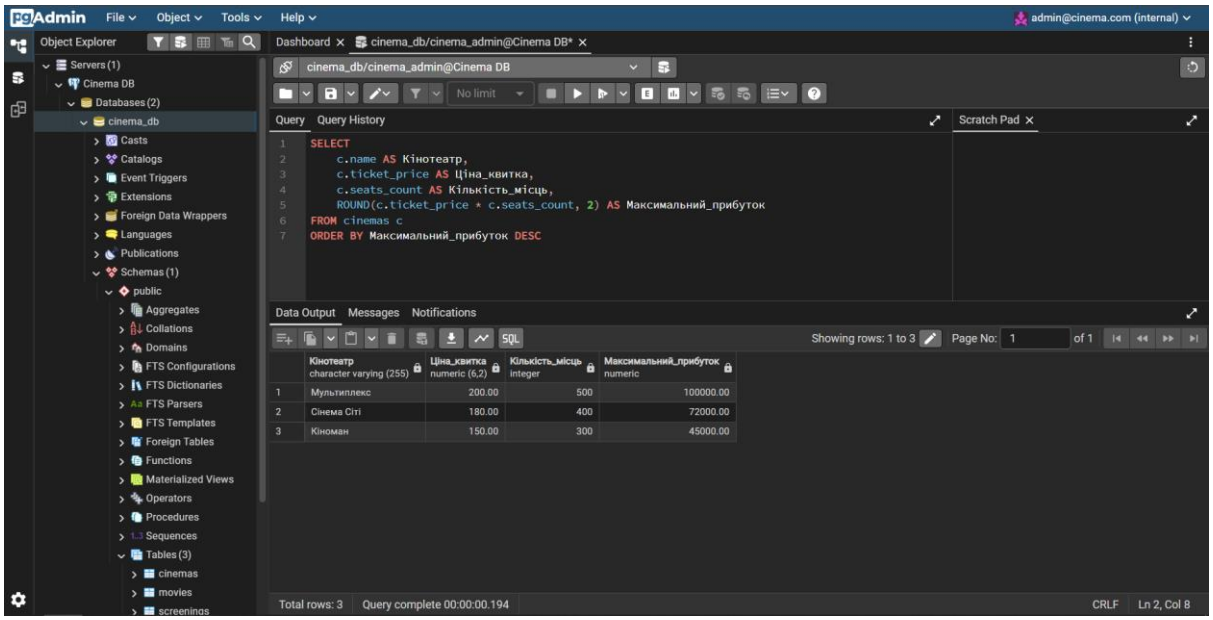


Рис. 14 – Запит «Порахувати суму максимального прибутку для кожного кінотеатру від одного показу»

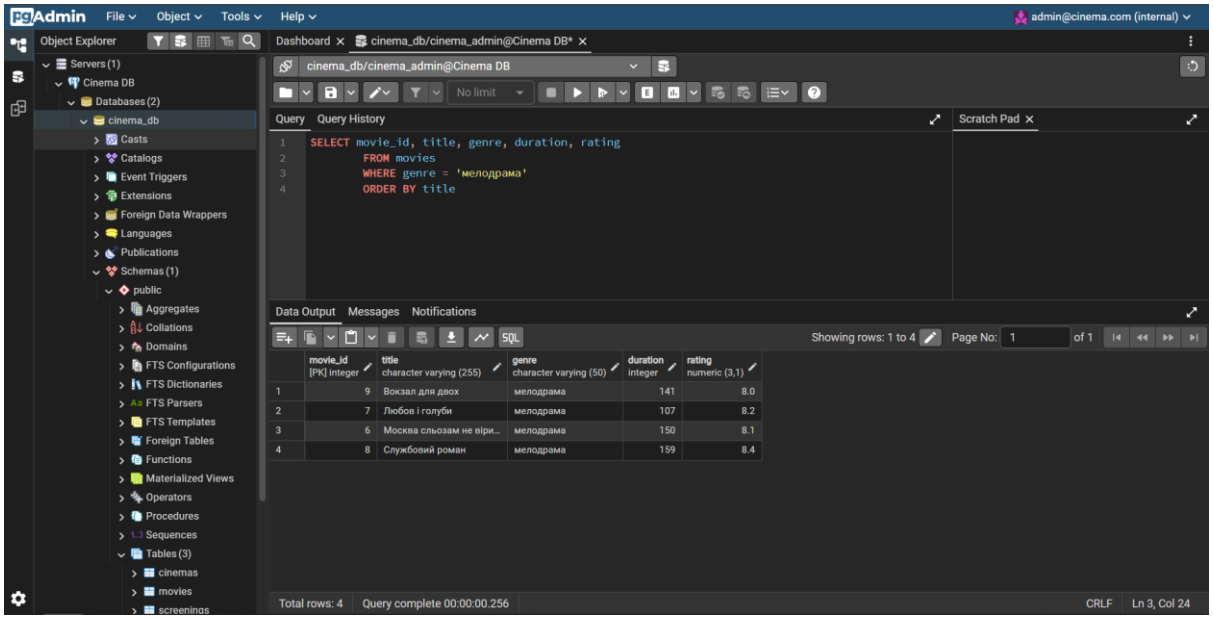


Рис. 15 – Запит «Відобразити всі фільми заданого жанру»

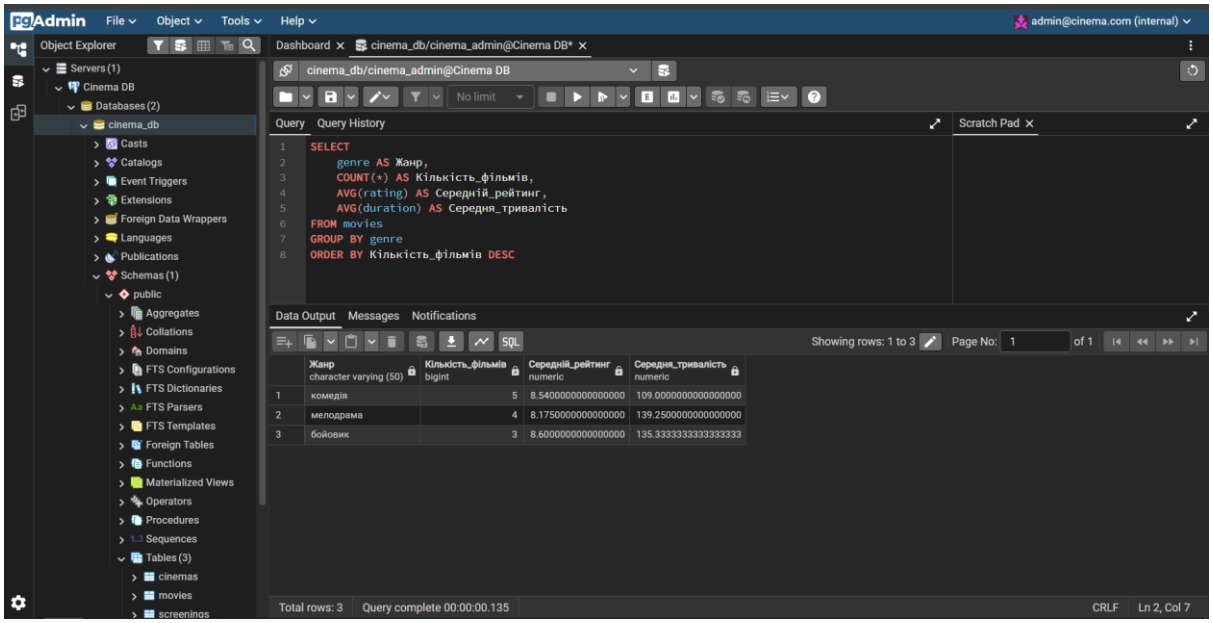


Рис. 16 – Запит «Порахувати кількість фільмів кожного жанру»

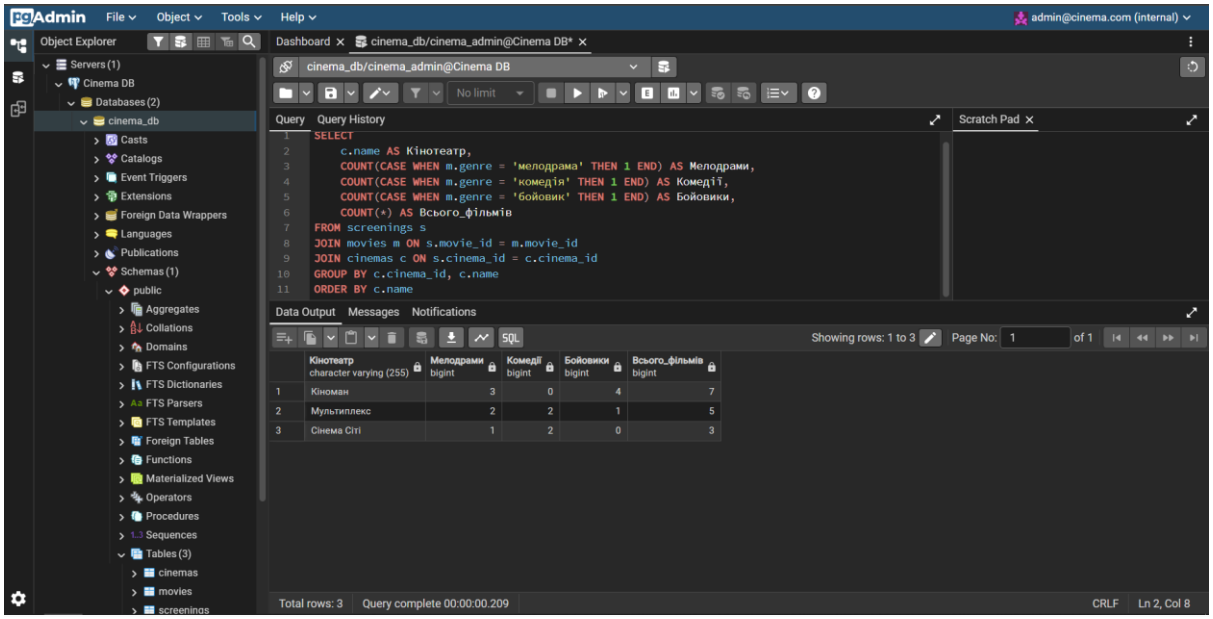


Рис. 17 – Запит «Порахувати кількість мелодрам, комедій, бойовиків, які транслюються в кожному кінотеатрі»

Посилання на гітхаб: https://github.com/DenysSheppard/python-labs/tree/master/lr9

Висновки

У ході виконання лабораторної роботи було створено систему управління кінотеатрами з використанням PostgreSQL та Docker. Реалізовано базу даних з трьома основними таблицями (фільми, кінотеатри, транслювання) та виконано всі необхідні запити згідно з варіантом.

Проект продемонстрував практичне застосування контейнеризації для розгортання СУБД, організацію зв'язків між таблицями та реалізацію різних типів SQL-запитів. Всі поставлені завдання виконано успішно, що підтверджується коректним функціонуванням бази даних та можливістю підключення через графічний клієнт pgAdmin.