

Таблиця: Users (Користувачі)

- user_id (PK, UUID) - Унікальний ідентифікатор користувача
- name (VARCHAR) - Ім'я користувача
- email (VARCHAR, UNIQUE) - Електронна пошта
- password_hash (VARCHAR) - Хеш пароля
- created_at (TIMESTAMP) - Дата реєстрації

Таблиця: Movies (Фільми)

- movie_id (PK, UUID) - Унікальний ідентифікатор фільму
- title (VARCHAR) - Назва
- genre (VARCHAR) - Жанр
- description (TEXT) - Опис
- release_year (INT) - Рік випуску
- duration_minutes (INT) - Тривалість у хвилинах

Таблиця: Halls (Кінозали)

- hall_id (PK, UUID) - Унікальний ідентифікатор залу
- name (VARCHAR) - Назва залу (наприклад, "Зал 1", "VIP-зал")
- capacity (INT) - Кількість місць

Таблиця: Showtimes (Сеанси)

- showtime_id (PK, UUID) - Унікальний ідентифікатор сеансу
- movie_id (FK, refs Movies.movie_id) - Ідентифікатор фільму
- hall_id (FK, refs Halls.hall_id) - Ідентифікатор залу
- start_time (TIMESTAMP) - Дата та час початку
- price (DECIMAL) - Ціна квитка

Таблиця: Tickets (Квитки)

- showtime_id (PK, FK, refs Showtimes.showtime_id) - Ідентифікатор сеансу
- seat_number (PK, INT) - Номер місця
- user_id (FK, refs Users.user_id) - Ідентифікатор користувача, що купив квиток
- purchase_time (TIMESTAMP) - Час покупки

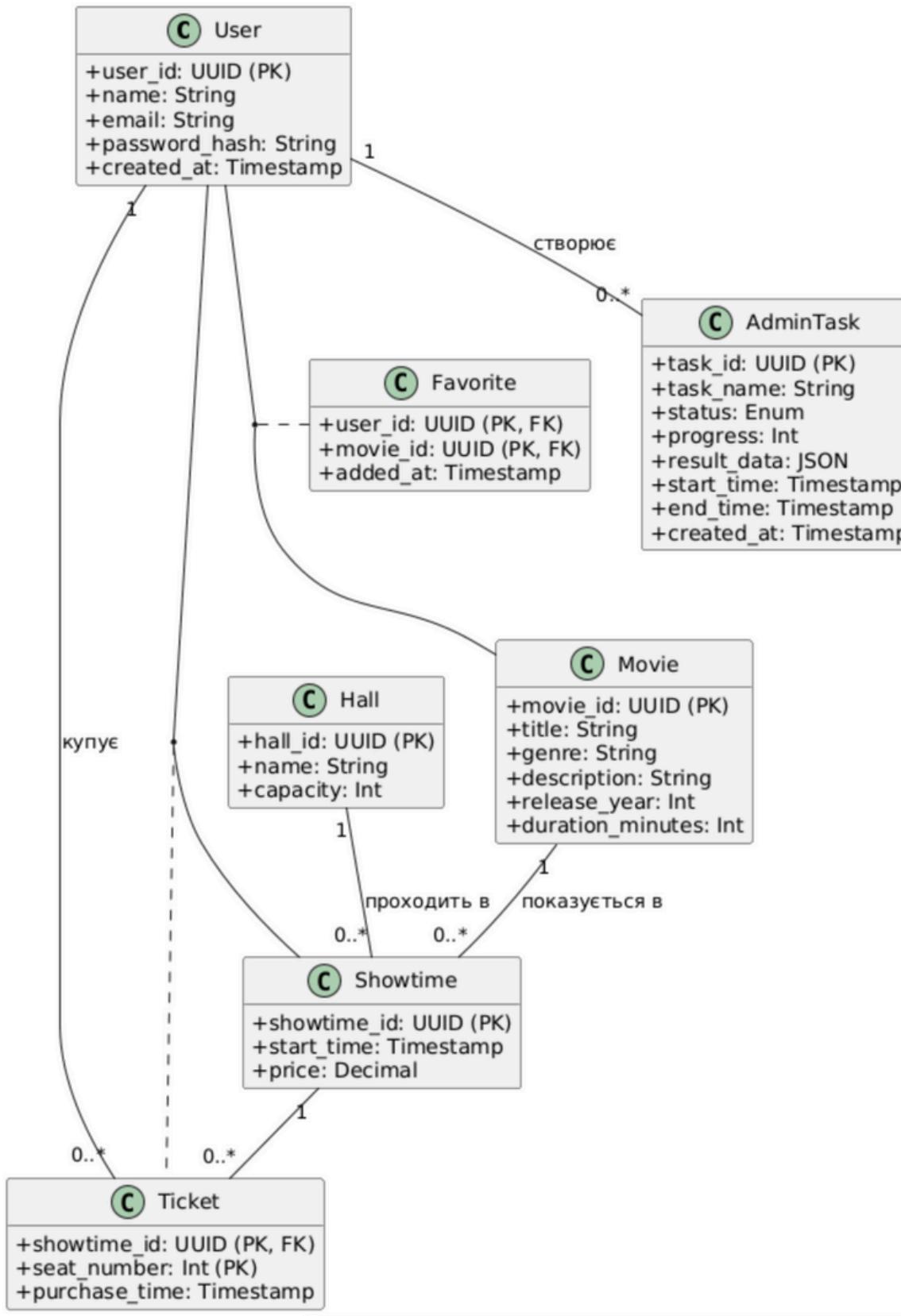
Таблиця: Favorites (Обрані фільми)

- user_id (PK, FK, refs Users.user_id) - Користувач
- movie_id (PK, FK, refs Movies.movie_id) - Фільм
- added_at (TIMESTAMP) - Коли додано

Таблиця: AdminTasks (Задачі планування)

- task_id (PK, UUID) - Унікальний ідентифікатор задачі
- task_name (VARCHAR) - Назва задачі (напр., "Розклад на 20-27 жовтня")
- created_by (FK, refs Users.user_id) - Адміністратор, що запустив задачу
- status (ENUM: 'queued', 'in_progress', 'done', 'error', 'canceled') - Статус виконання
- progress (INT) - Прогрес виконання у відсотках (10, 30, 60, 90, 100)
- result_data (JSONB) - Результат (збережений розклад або повідомлення про помилку)
- start_time (TIMESTAMP) - Час запуску
- end_time (TIMESTAMP) - Час завершення
- created_at (TIMESTAMP) - Час створення задачі

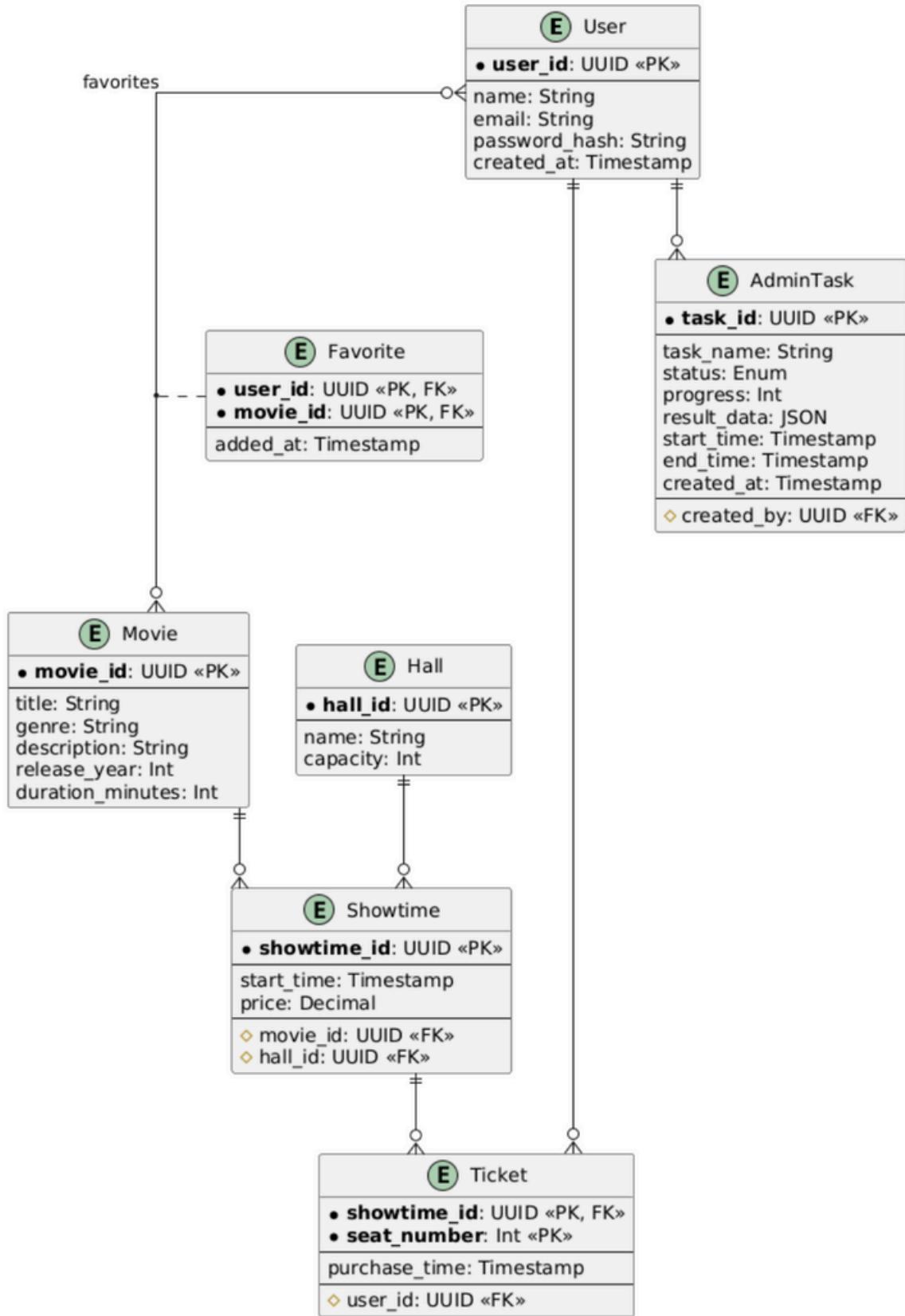
UML схема



Створення бази

```
1 CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
2
3 CREATE TYPE task_status_enum AS ENUM ('queued', 'in_progress', 'done', 'error', 'canceled');
4
5 CREATE TABLE "Users" (
6     user_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
7     name VARCHAR(255) NOT NULL,
8     email VARCHAR(255) UNIQUE NOT NULL,
9     password_hash VARCHAR(255) NOT NULL,
10    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
11 );
12
13 CREATE TABLE "Movies" (
14     movie_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
15     title VARCHAR(255) NOT NULL,
16     genre VARCHAR(100),
17     description TEXT,
18     release_year INT,
19     duration_minutes INT
20 );
21
22 CREATE TABLE "Halls" (
23     hall_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
24     name VARCHAR(100) NOT NULL,
25     capacity INT NOT NULL
26 );
27
28 CREATE TABLE "AdminTasks" (
29     task_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
30     task_name VARCHAR(255) NOT NULL,
31     status task_status_enum DEFAULT 'queued',
32     progress INT DEFAULT 0,
33     result_data JSONB,
34     start_time TIMESTAMP WITH TIME ZONE,
35     end_time TIMESTAMP WITH TIME ZONE,
36     created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
37     created_by UUID NOT NULL,
38     CONSTRAINT fk_created_by FOREIGN KEY(created_by) REFERENCES "Users"(user_id) ON DELETE SET NULL
39 );
40
41 CREATE TABLE "Showtimes" (
42     showtime_id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
43     movie_id UUID NOT NULL,
44     hall_id UUID NOT NULL,
45     start_time TIMESTAMP WITH TIME ZONE NOT NULL,
46     price DECIMAL(10, 2) NOT NULL,
47     CONSTRAINT fk_movie FOREIGN KEY(movie_id) REFERENCES "Movies"(movie_id) ON DELETE CASCADE,
48     CONSTRAINT fk_hall FOREIGN KEY(hall_id) REFERENCES "Halls"(hall_id) ON DELETE CASCADE
49 );
50
51
52 CREATE TABLE "Tickets" (
53     showtime_id UUID NOT NULL,
54     user_id UUID NOT NULL,
55     seat_number INT NOT NULL,
56     purchase_time TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
57     PRIMARY KEY (showtime_id, seat_number), -- Композитний ключ, щоб місце було унікальним для сеансу
58     CONSTRAINT fk_showtime FOREIGN KEY(showtime_id) REFERENCES "Showtimes"(showtime_id) ON DELETE CASCADE,
59     CONSTRAINT fk_user FOREIGN KEY(user_id) REFERENCES "Users"(user_id) ON DELETE CASCADE
60 );
61
62
63 CREATE TABLE "Favorites" (
64     user_id UUID NOT NULL,
65     movie_id UUID NOT NULL,
66     added_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
67     PRIMARY KEY (user_id, movie_id), -- Композитний ключ, щоб уникнути дублікатів
68     CONSTRAINT fk_user FOREIGN KEY(user_id) REFERENCES "Users"(user_id) ON DELETE CASCADE,
69     CONSTRAINT fk_movie FOREIGN KEY(movie_id) REFERENCES "Movies"(movie_id) ON DELETE CASCADE
70 );
71
72 CREATE INDEX idx_showtimes_start_time ON "Showtimes"(start_time);
73 CREATE INDEX idx_movies_title ON "Movies"(title);
74 CREATE INDEX idx_admintasks_status ON "AdminTasks"(status);
```

ER діаграма



```

5 class Program
6 {
7     0 references
8     static async Task Main(string[] args)
9     {
10         const string connectionString = "Server=localhost;Port=5433;UserId=postgres;Password=1145;Database=postgres";
11
12         Console.WriteLine("Спроба підключитися до бази даних...");
13
14         try
15         {
16             await using var conn = new NpgsqlConnection(connectionString);
17             await conn.OpenAsync();
18
19             Console.WriteLine($"Підключення успішне! Поточна база: {conn.Database}");
20
21             const string checkTableSql = @"
22                 SELECT table_schema, table_name
23                 FROM information_schema.tables
24                 WHERE table_schema = 'public' AND table_name = 'movies';
25             ";
26
27             await using (var checkCmd = new NpgsqlCommand(checkTableSql, conn))
28             {
29                 if (!checkReader.HasRows)
30                 {
31                     Console.ForegroundColor = ConsoleColor.Red;
32                     Console.WriteLine("Таблиця 'public.movies' не знайдена у базі!");
33                     Console.ResetColor();
34                     return;
35                 }
36             }
37
38             const string sql = @"SELECT title, release_year, duration_minutes FROM public.movies";
39             await using var cmd = new NpgsqlCommand(sql, conn);
40             await using var reader = await cmd.ExecuteReaderAsync();
41
42             if (!reader.HasRows)
43             {
44                 Console.WriteLine("У таблиці 'Movies' немає жодного фільму.");
45             }
46             else
47             {
48                 Console.WriteLine("\n--- Список фільмів ---");
49                 while (await reader.ReadAsync())
50                 {
51                     string title = reader.IsDBNull(0) ? "(null)" : reader.GetString(0);
52                     string releaseYear = reader.IsDBNull(1) ? "N/A" : reader.GetInt32(1).ToString();
53                     string duration = reader.IsDBNull(2) ? "N/A" : reader.GetInt32(2).ToString();
54
55                     Console.WriteLine($"Назва: {title}, Рік: {releaseYear}, Тривалість: {duration} хв.");
56                 }
57                 Console.WriteLine("-----");
58             }
59         }
60         catch (NpgsqlException ex)
61         {
62             Console.ForegroundColor = ConsoleColor.Red;
63             Console.WriteLine("Помилка PostgreSQL:");
64             Console.WriteLine(ex.ToString());
65             Console.ResetColor();
66         }
67         catch (Exception ex)
68         {
69             Console.ForegroundColor = ConsoleColor.Red;
70             Console.WriteLine("Загальна помилка:");
71             Console.WriteLine(ex.ToString());
72             Console.ResetColor();
73         }
74
75         Console.WriteLine("\nНатисніть будь-яку клавішу для виходу.");
76         Console.ReadKey();
77     }
78 }

```

Спроба підключитися до бази даних...
Підключення успішне! Поточна база: postgres

--- Список фільмів ---
Назва: Дюна: Частина друга, Рік: 2024, Тривалість: 166 хв.
Назва: Інтерстеллар, Рік: 2014, Тривалість: 169 хв.
Назва: Темний лицар, Рік: 2008, Тривалість: 152 хв.

Натисніть будь-яку клавішу для виходу.