

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

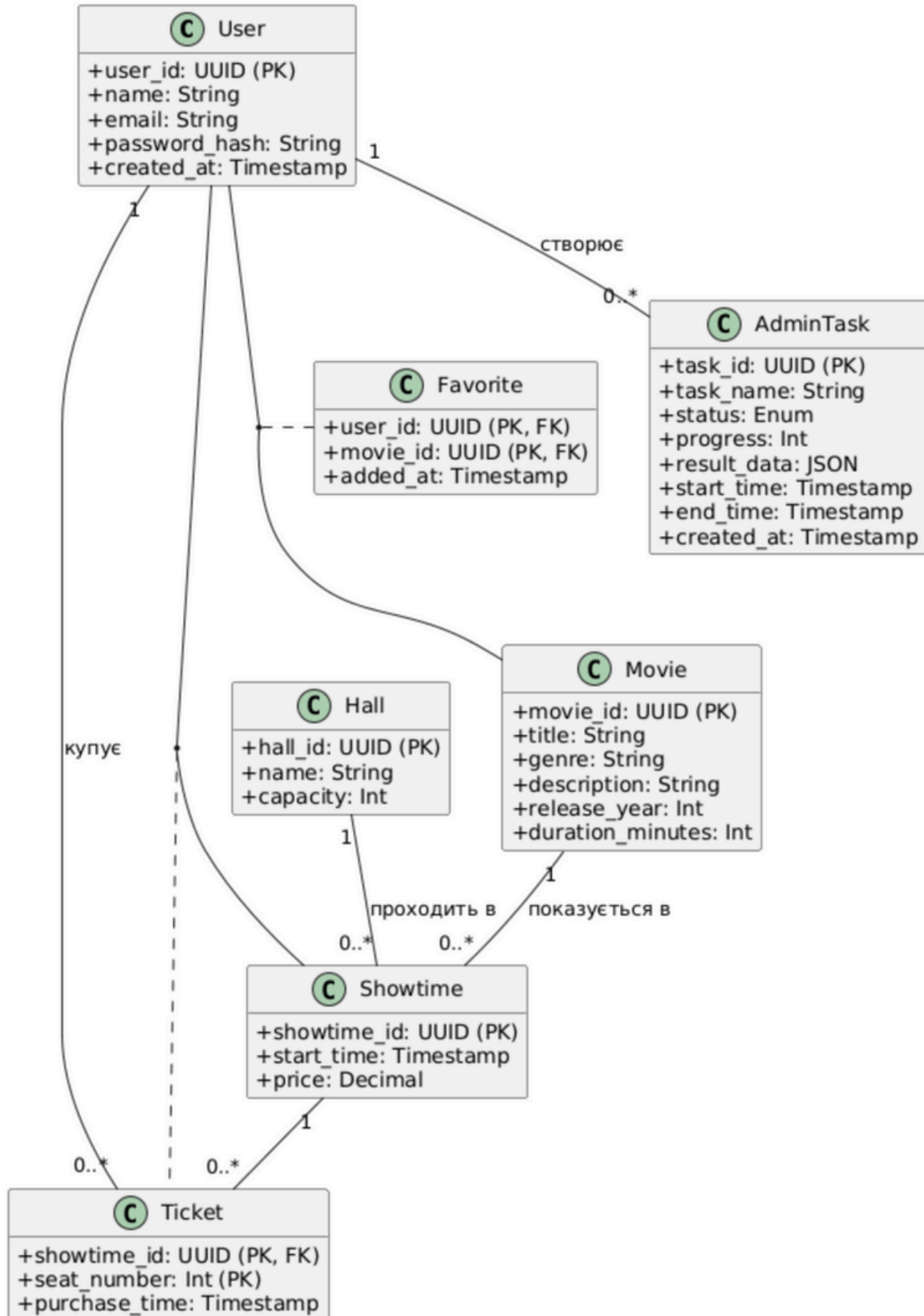
Таблиця: Favorites (Обрані фільми)

- **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

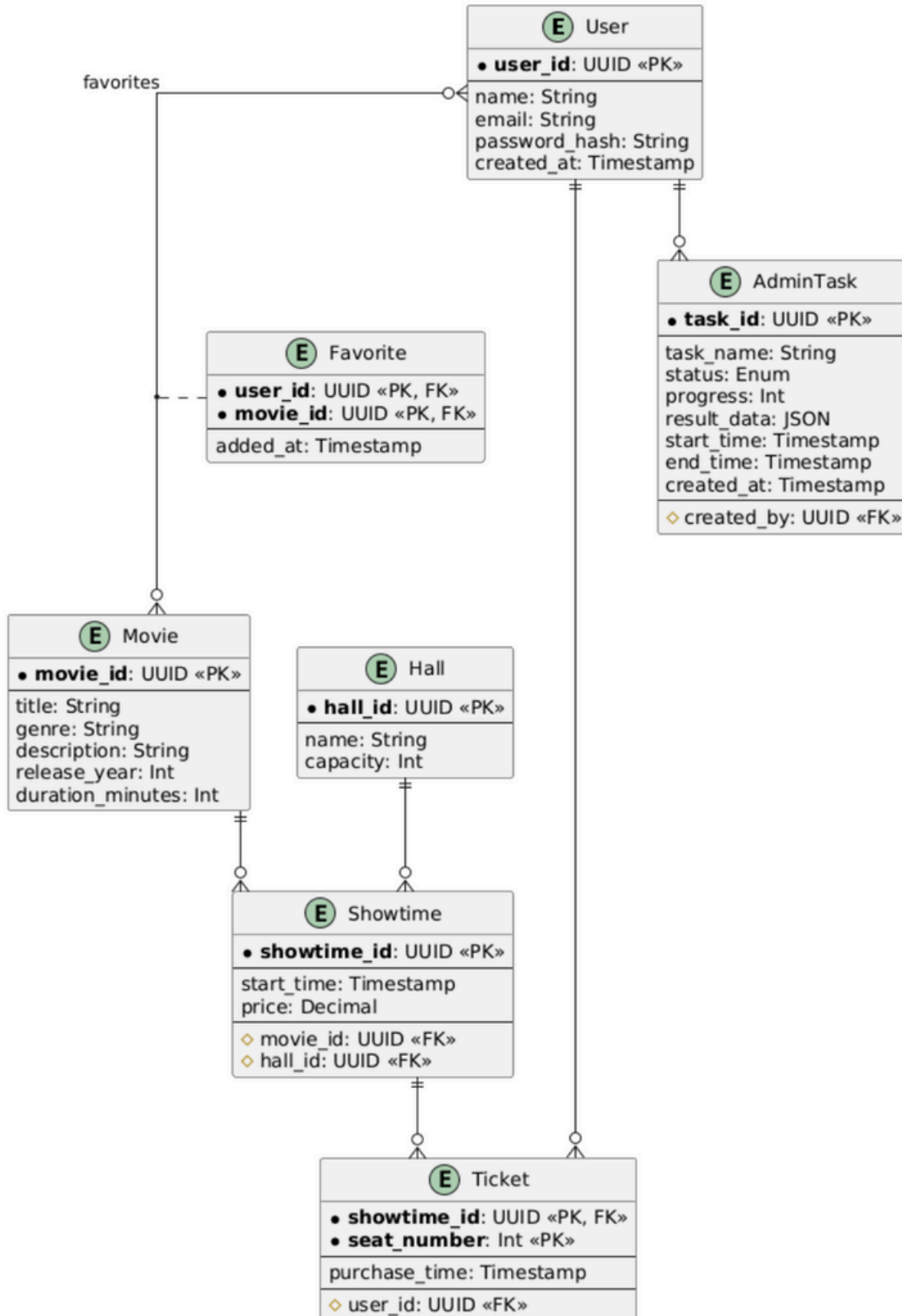
UML схема



Створення бази

```
1 CREATE EXTENSION IF NOT EXISTS "uuid-oss";
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 CREATE TABLE "Tickets" (
52    showtime_id UUID NOT NULL,
53    user_id UUID NOT NULL,
54    seat_number INT NOT NULL,
55    purchase_time TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
56    PRIMARY KEY (showtime_id, seat_number), -- Композитний ключ, щоб місце було унікальним для сеансу
57    CONSTRAINT fk_showtime FOREIGN KEY(showtime_id) REFERENCES "Showtimes"(showtime_id) ON DELETE CASCADE,
58    CONSTRAINT fk_user FOREIGN KEY(user_id) REFERENCES "Users"(user_id) ON DELETE CASCADE
59);
60
61
62 CREATE TABLE "Favorites" (
63    user_id UUID NOT NULL,
64    movie_id UUID NOT NULL,
65    added_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
66    PRIMARY KEY (user_id, movie_id), -- Композитний ключ, щоб уникнути дублікатів
67    CONSTRAINT fk_user FOREIGN KEY(user_id) REFERENCES "Users"(user_id) ON DELETE CASCADE,
68    CONSTRAINT fk_movie FOREIGN KEY(movie_id) REFERENCES "Movies"(movie_id) ON DELETE CASCADE
69);
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($"Підключення успішне! Поточна база: {conn.Database}");
20
21             const string checkTableSql = @"
22                 SELECT table_schema, table_name
23                 FROM information_schema.tables
24                 WHERE table_schema = 'public' AND table_name = 'movies';
25             ";
26
27             await using (var checkCmd = new NpgsqlCommand(checkTableSql, conn))
28             await using (var checkReader = await checkCmd.ExecuteReaderAsync())
29             {
30                 if (!checkReader.HasRows)
31                 {
32                     Console.ForegroundColor = ConsoleColor.Red;
33                     Console.WriteLine("Таблиця 'public.movies' не знайдена в базі!");
34                     Console.ResetColor();
35                     return;
36                 }
37             }
38
39             const string sql = @"SELECT title, release_year, duration_minutes FROM public.movies;";
40             await using var cmd = new NpgsqlCommand(sql, conn);
41             await using var reader = await cmd.ExecuteReaderAsync();
42
43             if (!reader.HasRows)
44             {
45                 Console.WriteLine("У таблиці 'Movies' немає жодного фільму.");
46             }
47             else
48             {
49                 Console.WriteLine("\n--- Список фільмів ---");
50                 while (await reader.ReadAsync())
51                 {
52                     string title = reader.IsDBNull(0) ? "(null)" : reader.GetString(0);
53                     string releaseYear = reader.IsDBNull(1) ? "N/A" : reader.GetInt32(1).ToString();
54                     string duration = reader.IsDBNull(2) ? "N/A" : reader.GetInt32(2).ToString();
55
56                     Console.WriteLine($"Назва: {title}, Рік: {releaseYear}, Тривалість: {duration} хв.");
57                 }
58                 Console.WriteLine("-----");
59             }
60         }
61         catch (NpgsqlException ex)
62         {
63             Console.ForegroundColor = ConsoleColor.Red;
64             Console.WriteLine("Помилка PostgreSQL:");
65             Console.WriteLine(ex.ToString());
66             Console.ResetColor();
67         }
68         catch (Exception ex)
69         {
70             Console.ForegroundColor = ConsoleColor.Red;
71             Console.WriteLine("Загальна помилка:");
72             Console.WriteLine(ex.ToString());
73             Console.ResetColor();
74         }
75
76         Console.WriteLine("\nНатисніть будь-яку клавішу для виходу.");
77         Console.ReadKey();
78     }
79 }

```

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.