

**Міністерство освіти і науки України  
Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

**Лабораторна робота №1**

з дисципліни  
**«Компоненти програмної інженерії»**

Виконала:

студентка групи IM-42  
Шалак Анастасія Володимирівна  
номер у списку групи: 30

Перевірив:

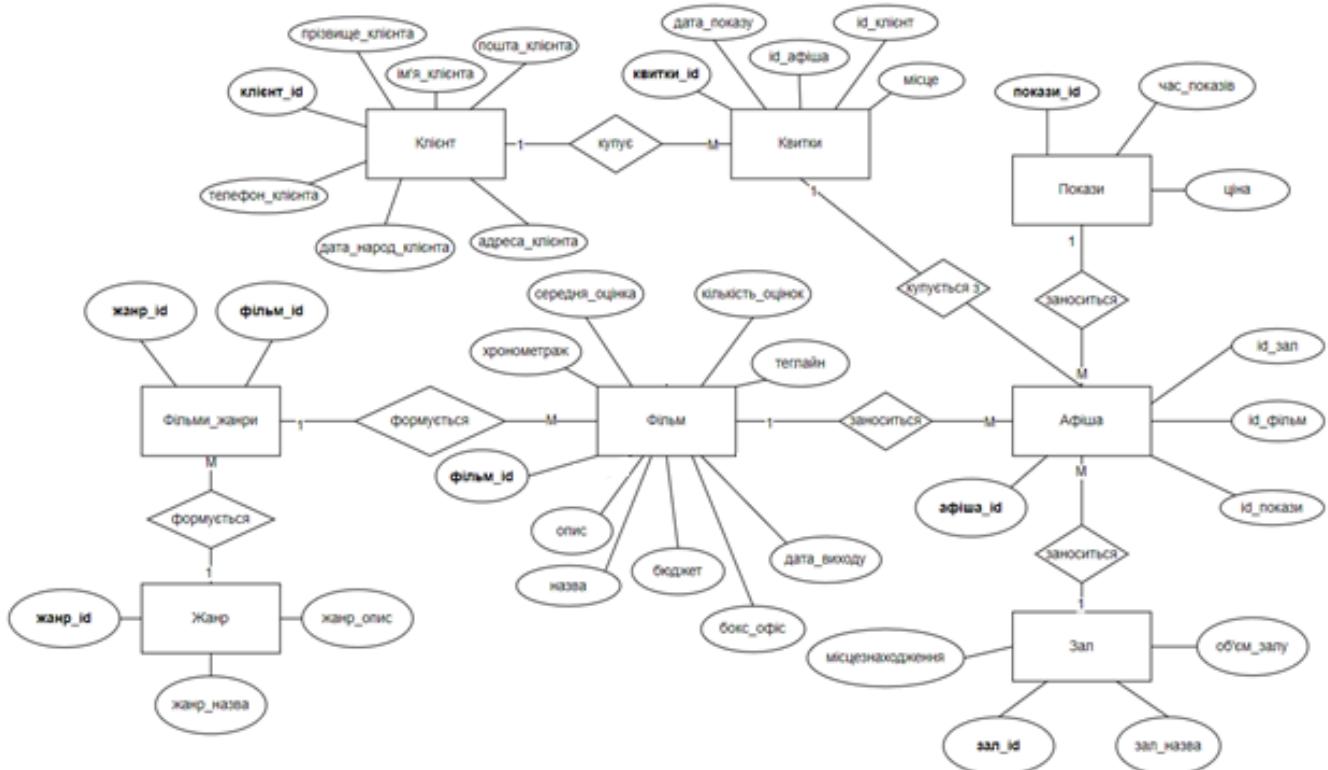
Нечай Д. О.

Київ 2025

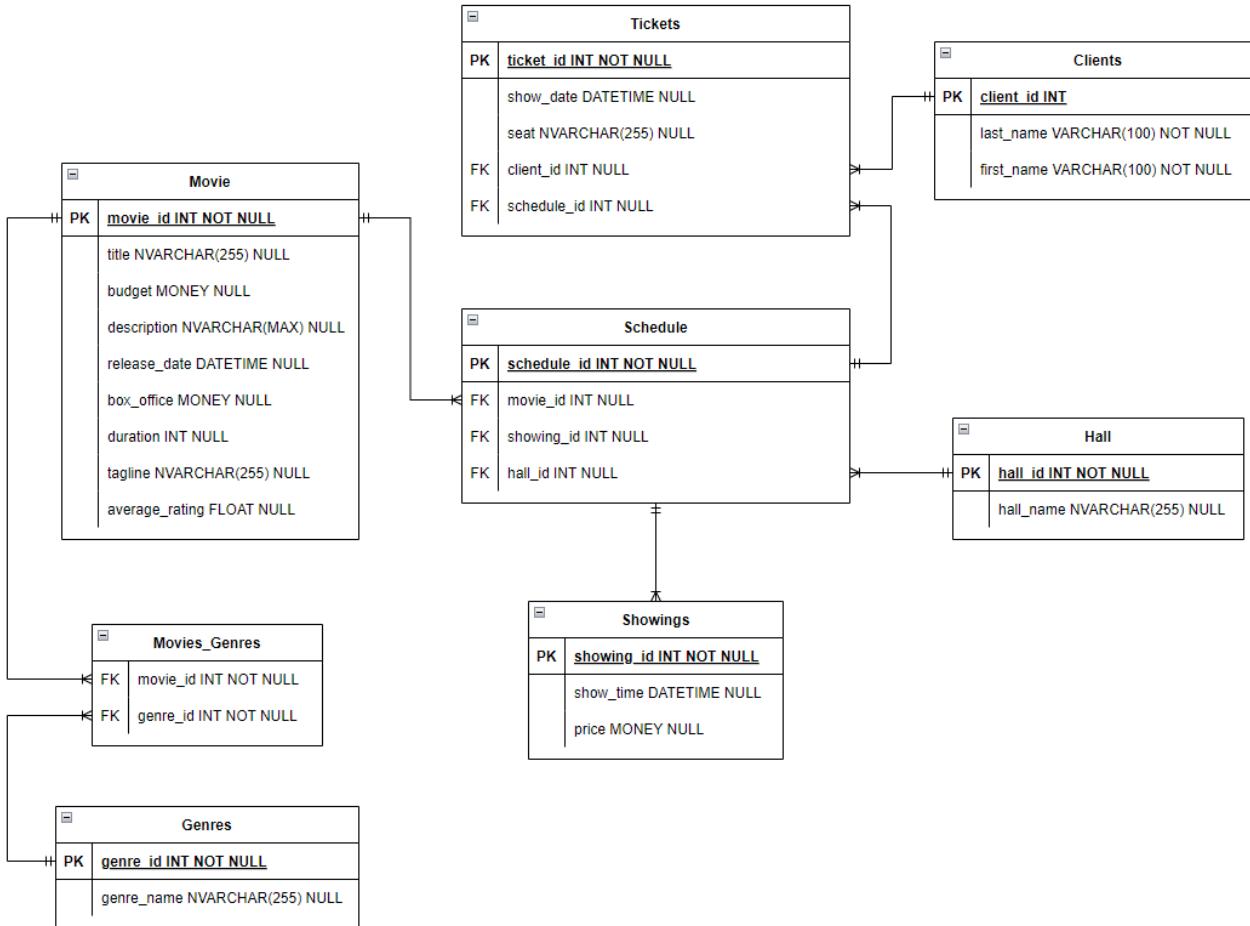
## Entity Relationship

**Система купівлі/бронювання квитків у кінотеатрі**

- Діаграма в нотації Ченом.



- Діаграма у нотації з «воронячими лапками» (Crow's Foot Notation).



- Побудувати ER-діаграму за допомогою декларативного синтаксису.

@startuml

!theme plain

entity "Hall" as Hall {

\* hall\_id : INT <<PK>>

--

hall\_name : NVARCHAR(255)

}

entity "Movie" as Movie {

\* movie\_id : INT <<PK>>

--

title : NVARCHAR(255)

budget : MONEY

```
description : NVARCHAR(MAX)
release_date : DATETIME
box_office : MONEY
duration : INT
tagline : NVARCHAR(255)
average_rating : FLOAT
}
```

```
entity "Genres" as Genres {
    * genre_id : INT <<PK>>
    --
    genre_name : NVARCHAR(255)
}
```

```
entity "Movies_Genres" as Movies_Genres {
    * movie_id : INT <<PK, FK>>
    * genre_id : INT <<PK, FK>>
}
```

```
entity "Showings" as Showings {
    * showing_id : INT <<PK>>
    --
    show_time : DATETIME
    price : MONEY
}
```

```
entity "Schedule" as Schedule {
    * schedule_id : INT <<PK>>
    --
    hall_id : INT <<FK>>
```

```
showing_id : INT <<FK>>
movie_id : INT <<FK>>
}
```

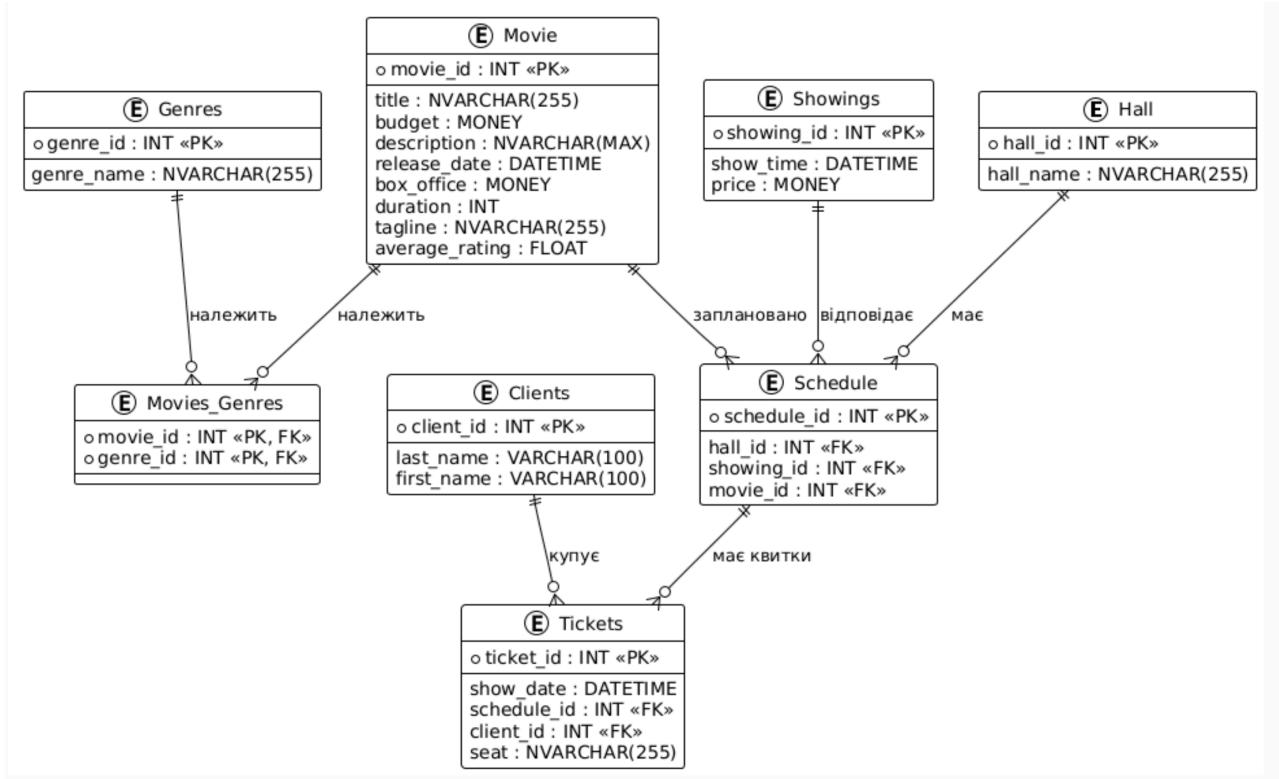
```
entity "Clients" as Clients {
* client_id : INT <<PK>>
--
last_name : VARCHAR(100)
first_name : VARCHAR(100)
}
```

```
entity "Tickets" as Tickets {
* ticket_id : INT <<PK>>
--
show_date : DATETIME
schedule_id : INT <<FK>>
client_id : INT <<FK>>
seat : NVARCHAR(255)
}
```

```
Hall ||--o{ Schedule : "має"
Movie ||--o{ Schedule : "заплановано"
Showings ||--o{ Schedule : "відповідає"
Schedule ||--o{ Tickets : "має квитки"
Clients ||--o{ Tickets : "купують"
Movie ||--o{ Movies_Genres : "належить"
Genres ||--o{ Movies_Genres : "належить"
```

```
@enduml
```

- Забезпечити відповідність між діаграмами (Чен, Crow's Foot) та декларативним-описом.



## Класи з властивостями у мові без суворої типізації (JavaScript)

```

class Hall {
    constructor(hall_id, hall_name) {
        this.hall_id = hall_id;
        this.hall_name = hall_name;
    }
}

class Movie {
    constructor(movie_id, title, budget, description, release_date, box_office, duration, tagline, average_rating) {
        this.movie_id = movie_id;
        this.title = title;
        this.budget = budget;
        this.description = description;
        this.release_date = release_date;
        this.box_office = box_office;
        this.duration = duration;
        this.tagline = tagline;
        this.average_rating = average_rating;
    }
}

class Genre {
    constructor(genre_id, genre_name) {
        this.genre_id = genre_id;
        this.genre_name = genre_name;
    }
}

```

```
class MovieGenre {
  constructor(movie_id, genre_id) {
    this.movie_id = movie_id;
    this.genre_id = genre_id;
  }
}

class Showing {
  constructor(showing_id, show_time, price) {
    this.showing_id = showing_id;
    this.show_time = show_time;
    this.price = price;
  }
}

class Schedule {
  constructor(schedule_id, hall_id, showing_id, movie_id) {
    this.schedule_id = schedule_id;
    this.hall_id = hall_id;
    this.showing_id = showing_id;
    this.movie_id = movie_id;
  }
}

class Client {
  constructor(client_id, last_name, first_name) {
    this.client_id = client_id;
    this.last_name = last_name;
    this.first_name = first_name;
  }
}

class Ticket {
  constructor(ticket_id, show_date, schedule_id, client_id, seat) {
    this.ticket_id = ticket_id;
    this.show_date = show_date;
    this.schedule_id = schedule_id;
    this.client_id = client_id;
    this.seat = seat;
  }
}
```