

**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 №6

з дисципліни
«Бази даних»

Виконала:

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

Перевірив:

Русінов В. В.

Київ 2025

Лабораторна робота №6

Мета роботи: ознайомитись з інструментом Prisma ORM, виконати інтроспекцію існуючої реляційної бази даних «Кінотеатр», створеної в лабораторних роботах 1–5, та створити три різні міграції схеми.

Завдання

1. Виконати інтроспекцію існуючої бази cinema_db за допомогою команди `prisma db pull`
2. Створити щонайменше три різні міграції схеми
3. Застосувати міграції до бази даних
4. Перевірити результат за допомогою Prisma Studio та Prisma Client

Хід виконання роботи

1. Налаштування проєкту

Було створено новий Node.js-проєкт та встановлено необхідні пакети:

```
npm init -y
```

```
npm install prisma --save-dev
```

```
npm install @prisma/client
```

```
npx prisma init --datasource-provider postgresql
```

У файлі `.env` вказано рядок підключення до локальної бази даних:
`DATABASE_URL="postgresql://localhost:5432/cinema_db"`

2. Інтроспекція існуючої схеми

Виконано команду: `npx prisma db pull`

В результаті успішно отримано 9 моделей (Hall, Movie, Genres, Movies_Genres, Showtime_Prices, Showings, Schedule, Clients, Tickets), які повністю відповідають схемі.

3. Створення та застосування міграцій

Міграція 1 – Додавання таблиці відгуків про фільми

Додано нову модель Review у schema.prisma:

```
model Review {  
  id      Int      @id @default(autoincrement())  
  movieId Int  
  rating  Int       @db.SmallInt  
  comment String?  @db.Text  
  createdAt DateTime @default(now())  
  
  movie    Movie    @relation(fields: [movieId], references: [movie_id], onDelete:  
Cascade)  
  
  @@map("reviews")  
}
```

Міграція 2 – Додавання поля isVip та перейменування поля в залі

Змінено модель Hall:

```
model Hall {  
  hallId  Int      @id @map("hall_id") @default(autoincrement())  
  hallName String  @map("hall_name") @unique @db.VarChar(255)  
  isVip   Boolean  @default(false)  
  
  schedules Schedule[]  
}
```

```
@@map("hall")  
}
```

Міграція 3 – Видалення поля tagline з таблиці Movie

Видалено рядок tagline з моделі Movie.

Застосовано: `npx prisma migrate dev --name remove-tagline`

4. Перевірка результатів

- Виконано `npx prisma generate`
- Запущено Prisma Studio (`npx prisma studio`)
- видно нові таблиці та поля
- Перевірено структуру бази в pgAdmin – всі зміни застосовано коректно

Додатки:

1. Результат виконання `npx prisma db pull`:

```
Loaded Prisma config from prisma.config.ts.  
  
Prisma schema loaded from prisma\schema.prisma  
Datasource "db": PostgreSQL database "cinema_db", schema "public" at "localhost:5432"  
  
✓ Introspected 9 models and wrote them into prisma\schema.prisma in 86ms  
Run prisma generate to generate Prisma Client.
```

2. Prisma Studio – таблиця reviews:

```
Loaded Prisma config from prisma.config.ts.  
  
Prisma schema loaded from prisma\schema.prisma  
Datasource "db": PostgreSQL database "cinema_db", schema "public" at "localhost:5432"  
  
Already in sync, no schema change or pending migration was found.
```

3. Prisma Studio – модель Hall з полем isVip:

```

Loaded Prisma config from prisma.config.ts.

Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "cinema_db", schema "public" at "localhost:5432"

Applying migration `20251129124613_add_is_vip_to_hall`

The following migration(s) have been created and applied from new schema changes:

prisma/migrations/
└─ 20251129124613_add_is_vip_to_hall/
    └─ migration.sql

Your database is now in sync with your schema.

```

4. Prisma Studio – модель Movie без поля tagline:

```

Loaded Prisma config from prisma.config.ts.

Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "cinema_db", schema "public" at "localhost:5432"

Applying migration `20251129124655_remove_tagline`

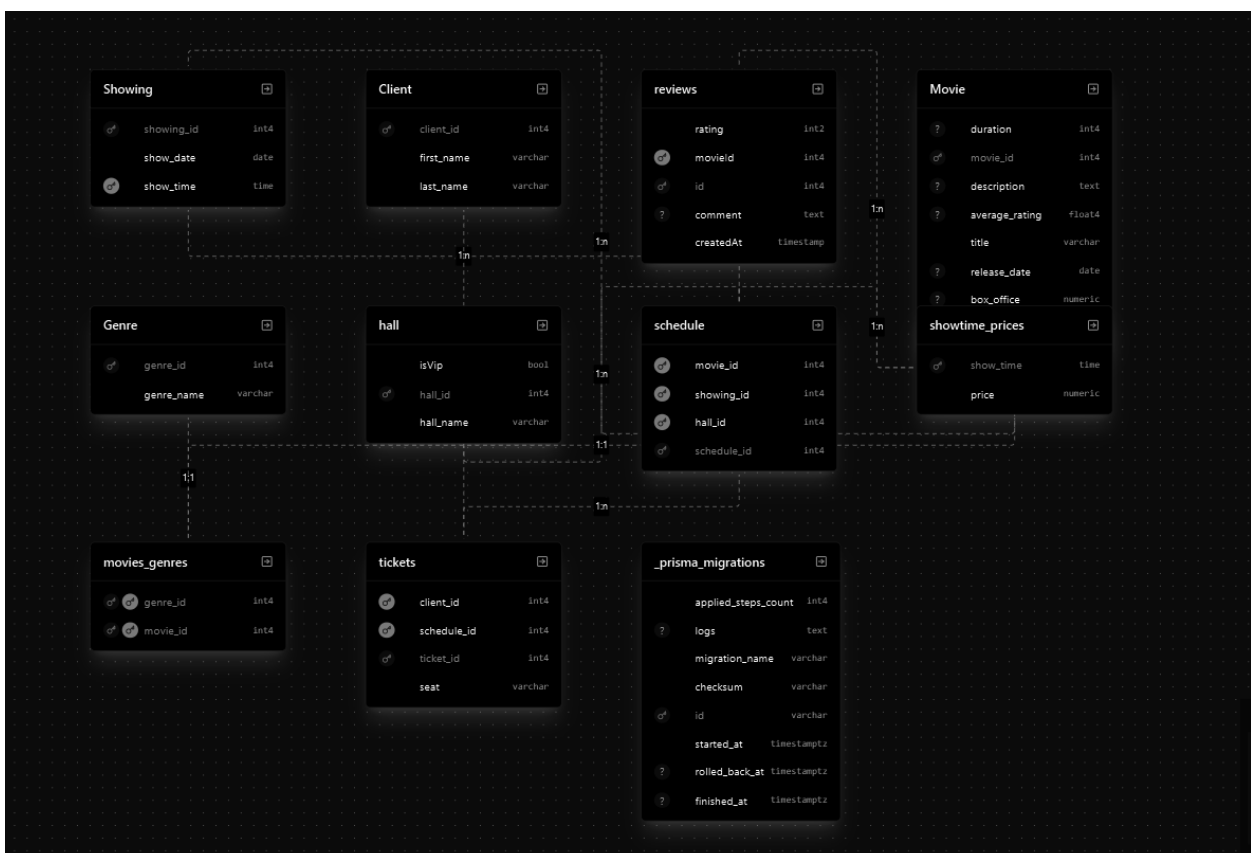
The following migration(s) have been created and applied from new schema changes:

prisma/migrations/
└─ 20251129124655_remove_tagline/
    └─ migration.sql

Your database is now in sync with your schema.

```

5. Структура бази після міграцій:



Prisma Studio

Schema: public

Studio

Visualizer

Console

Tables

prisma_migrations

Client

Genre

hall

Movie

movies_genres

reviews

schedule

showing

showtime_prices

tickets

Insert row

	id varchar	applied_steps_count int4	checksum varchar	finished_at timestamp2	logs text	migration_name varchar
<input type="checkbox"/>	dc076769-b484-4392-9268-4d412567a020	1	4487028a40f42d3a186b3bb8ca4eabfc1159fb6ca351968a1206dc1d62e3fece	2025-11-29T12:37:43.129Z	NULL	20251129123743_init_from
<input type="checkbox"/>	6447c009-bc65-4725-b08a-8ec9ce7a6818	1	5b54b2a320afe958cf0d04adf2b51d6674168dc3cb2b3a6c90999a20b5d9f000	2025-11-29T12:46:13.767Z	NULL	20251129124613_add_is_vi
<input type="checkbox"/>	a8d23d8c-b641-47c2-904e-9faa8b07d75c	1	ebcc4184215fdfa04a5a54deaef2cba54eb4181fa388304628429c33b459a30	2025-11-29T12:46:55.531Z	NULL	20251129124655_remove_ta

Висновки:

У ході лабораторної роботи було успішно:

- виконано інтроспекцію існуючої бази даних кінотеатру за допомогою Prisma ORM - створено та застосовано три різні міграції схеми
- перевірено коректність змін за допомогою Prisma Studio та pgAdmin

Отримані навички дозволяють ефективно керувати еволюцією схеми бази даних у реальних проєктах.