

Вступ

В рамках цієї лабораторної роботи була виконана конвертація ER-діаграми, розробленої в ЛР №1, у функціональну реляційну схему PostgreSQL. Були написані та успішно виконані DDL-інструкції (`CREATE TABLE`) для створення семи таблиць, визначено типи даних, первинні та зовнішні ключі, а також логічні обмеження. Схема була протестована за допомогою тестових даних (`INSERT INTO`) для підтвердження цілісності.

Детальна Структура Таблиць та Обмеження

Схема складається з 7 таблиць, що моделюють репертуар, конфігурацію залів та транзакції продажу квитків.

1. Таблиця CUSTOMER (Клієнт)

- **Призначення:** Зберігання основної інформації про користувачів системи.
- **Первинний Ключ (PK):** `customer_id` (SERIAL).
- **Обмеження:** Поле `email_address` є обов'язковим (**NOT NULL**) та унікальним (**UNIQUE**).

2. Таблиця MOVIE (Фільм)

- **Призначення:** Каталог фільмів.
- **Первинний Ключ (PK):** `movie_id` (SERIAL).
- **Обмеження:** Поле `duration` (тривалість) має бути обов'язковим (**NOT NULL**) та більшим за нуль (**CHECK > 0**).

3. Таблиця HALL (Зал)

- **Призначення:** Зберігання даних про кінозали.
- **Первинний Ключ (PK):** `hall_id` (SERIAL).
- **Обмеження:** Поле `name_hall` залу є **UNIQUE** та **NOT NULL**.

4. Таблиця SEAT (Місце)

- **Призначення:** Деталізація кожного фізичного місця в кінозалах.
- **Первинний Ключ (PK):** `seat_id` (SERIAL).
- **Зовнішній Ключ (FK):** `hall_id` посилається на `HALL(hall_id)`.
- **Обмеження:** Для забезпечення унікальності фізичного місця в межах залу накладено **Композитне UNIQUE** обмеження на комбінацію (`hall_id, row_num, seat_number`).

5. Таблиця SHOWING (Сеанс)

- **Призначення:** Розклад показів (час, фільм, зал).
- **Первинний Ключ (PK):** `session_id` (SERIAL).
- **Зовнішні Ключі (FK):** `movie_id` посилається на `MOVIE(movie_id)`, а `hall_id` посилається на `HALL(hall_id)`.
- **Обмеження:** Накладено **Композитне UNIQUE** обмеження на (`hall_id, start_time`) для запобігання накладанню сеансів в одному залі.

6. Таблиця BOOKING (Бронювання)

- **Призначення:** Факт замовлення, зробленого клієнтом.
- **Первинний Ключ (PK):** `booking_id` (SERIAL).
- **Зовнішній Ключ (FK):** `customer_id` посилається на `CUSTOMER(customer_id)`.
- **Обмеження:** Поле `total_amount` не може бути від'ємною (**CHECK >= 0**), а поле `booking_date` має значення за замовчуванням (`DEFAULT CURRENT_TIMESTAMP`).

7. Таблиця TICKET (Квиток)

- **Призначення:** Зв'язуюча таблиця для реалізації зв'язку M:M, що фіксує продаж.
- **Первинний Ключ (PK):** `ticket_id` (SERIAL).
- **Зовнішні Ключі (FK):** Посилається на `BOOKING(booking_id)`, `SHOWING(session_id)` та `SEAT(seat_id)`.
- **Обмеження:** Накладено **Композитне UNIQUE** обмеження на `(session_id, seat_id)`. Це запобігає повторному продажу одного і того ж місця на один і той же сеанс.

Принципи перевірки цілісності (Тестування)

1. Дотримання Ієрархії та Зв'язки FK

Дані вставлялися у встановленому порядку, починаючи з незалежних батьківських таблиць (`CUSTOMER`, `MOVIE`, `HALL`) і закінчуючи залежною транзакційною таблицею (`TICKET`). Це підтвердило правильну роботу всіх зовнішніх ключів (**REFERENCES**), гарантуючи, що дочірні записи посилаються лише на існуючі батьківські записи.

2. Успішне Проходження Обмежень (Constraints)

Усі інструкції `INSERT INTO` були успішно виконані, що довело:

- **NOT NULL:** Всі обов'язкові поля були заповнені.
- **CHECK:** Значення числових полів відповідають встановленим умовам (`duration > 0, total_amount >= 0`).
- **UNIQUE:** Не було порушень унікальності для ключових полів, включаючи складні обмеження, що моделюють бізнес-правила (`UQ_HallTime`, `UQ_Ticket`).

Висновок

Конвертація ER-моделі у реляційну схему PostgreSQL пройшла успішно. Схема є надійною, а її цілісність була підтверджена за допомогою тестових даних.