

Звіт про виконання Лабораторної роботи №3

Схема складається з семи таблиць, які моделюють кінотеатр, включаючи довідники та транзакції.

1. Таблиця CUSTOMER (Клієнт)

Призначення: Зберігання інформації про клієнтів кінотеатру.

Стовпці:

customer_id (SERIAL) - Первинний ключ, автоінкремент.

name (VARCHAR(100)) - Ім'я клієнта, NOT NULL.

email (VARCHAR(100)) - Електронна пошта, NOT NULL, UNIQUE.

phone_number (VARCHAR(15)) - Номер телефону, NOT NULL, UNIQUE.

2. Таблиця HALL (Зал)

Призначення: Інформація про кінозали.

Стовпці:

hall_id (SERIAL) - Первинний ключ,

name_hall (VARCHAR(10)) - Назва залу, NOT NULL, UNIQUE.

type_hall (VARCHAR(15)) - Тип залу.

3. Таблиця MOVIE (Фільм)

Призначення: Каталог фільмів.

Стовпці:

movie_id (SERIAL) - Первинний ключ,

title (VARCHAR(40)) - Назва фільму, NOT NULL.

release_date (DATE) - Дата виходу, NOT NULL.

duration (INT) - Тривалість у хвилинах, NOT NULL, CHECK > 0.

genre (VARCHAR(40)) - Жанр, NOT NULL.

4. Таблиця SEAT (Місце)

Призначення: Фізичні місця в залах.

Стовпці:

seat_id (SERIAL) - Первинний ключ,

hall_id (INT) - Зовнішній ключ до HALL, NOT NULL.

row_num (INT) - Номер ряду, NOT NULL, CHECK > 0.

seat_number (INT) - Номер місця, NOT NULL, CHECK > 0.

Ключі:

UNIQUE обмеження: Комбінація (hall_id, row_num, seat_number) для унікальності місця в межах залу.

5. Таблиця SESSION (Сеанс)

Призначення: Розклад показів фільмів.

Стовпці:

session_id (SERIAL) - Первинний ключ.

movie_id (INT) - Зовнішній ключ до MOVIE, NOT NULL.

hall_id (INT) - Зовнішній ключ до HALL, NOT NULL.

start_time (DATE) - Дата сеансу, NOT NULL.

Ключі:

UNIQUE обмеження: Комбінація (hall_id, start_time) - запобігає накладанню сеансів.

6. Таблиця BOOKING (Бронювання)

Призначення: Замовлення квитків.

Стовпці:

booking_id (SERIAL) - Первинний ключ,

customer_id (INT) - Зовнішній ключ до CUSTOMER, NOT NULL.

booking_date (TIMESTAMP WITH TIME ZONE) - Дата бронювання, DEFAULT CURRENT_TIMESTAMP, NOT NULL.

total_amount (NUMERIC(8,2)) - Загальна сума, NOT NULL, CHECK >= 0.

7. Таблиця TICKET (Квиток)

Призначення: Індивідуальні квитки (зв'язуюча таблиця).

Стовпці:

ticket_id (SERIAL) - Первинний ключ,

booking_id (INT) - Зовнішній ключ до BOOKING, NOT NULL.

session_id (INT) - Зовнішній ключ до SESSION, NOT NULL.

seat_id (INT) - Зовнішній ключ до SEAT, NOT NULL.

price (NUMERIC(6,2)) - Ціна квитка, NOT NULL, CHECK > 0.

Ключі:

UNIQUE обмеження: Комбінація (session_id, seat_id) - запобігає подвійному продажу одного місця на один сеанс.

2. Важливі обмеження та Припущення

Ці обмеження реалізують бізнес-правила системи кінотеатру на рівні бази даних, забезпечуючи цілісність та якість даних.

Обмеження цілісності даних:

1. UNIQUE Обмеження:

Унікальні email та телефон: Гарантує, що кожен клієнт реєструється лише один раз, оскільки ці поля є унікальними ідентифікаторами особи.

Назви залів унікальні: Запобігає створенню двох залів з однаковою назвою (наприклад, не може бути двох "Зала 1").

Комбінація зал + час сеансу унікальна: Встановлює, що в один момент часу (`start_time`) в одному залі (`hall_id`) може відбуватися лише один показ. Це запобігає накладанню сеансів.

Комбінація сеанс + місце + зал унікальна: Це ключове обмеження гарантує, що кожне фізичне місце (`seat_id`, `hall_id`) може бути продане лише один раз на конкретний сеанс (`session_id`). Це запобігає подвійному продажу квитків.

2. CHECK Обмеження:

Тривалість фільму > 0 : Запобігає вставці нелогічних або нульових значень тривалості.

Ціна квитка > 0 : Гарантує, що всі продані квитки мають додатну вартість.

Сума бронювання ≥ 0 : Дозволяє нульові або додатні суми, що є логічним для фінансових транзакцій.

Бізнес-припущення:

1. Один клієнт може мати багато бронювань.
2. Одне бронювання може містити декілька квитків.
3. Один фільм може мати багато сеансів.
4. Один зал може приймати багато сеансів (в різний час).
5. Одне фізичне місце може бути продане на різні сеанси, але не на один сеанс двічі.


Приклади виконання запитів:

56


SELECT * FROM TICKET WHERE price = 150;

Data Output Messages Notifications


≡ +





▼




▼









SQL

	ticket_id [PK] integer	booking_id integer	session_id integer	seat_id integer	price numeric (6,2)
1	1	1	3	1	150.00
2	2	1	3	2	150.00
3	5	1	3	3	150.00

13

INSERT INTO CUSTOMER (full_name, email_address, phone_number) VALUES

14

('Голод Петро', 'holod.petya@gmail.com', '380693003396');

15

SELECT * FROM CUSTOMER WHERE full_name = 'Голод Петро';

Data Output

Messages

Graph Visualiser

×

Notifications

≡ +

▼

▼

SQL

	customer_id [PK] integer	full_name character varying (100)	email_address character varying (100)	phone_number character varying (15)
1	5	Голод Петро	holod.petya@gmail.com	380693003396

1

SELECT * FROM CUSTOMER WHERE full_name = 'Тухлий Дмитро';

2

UPDATE CUSTOMER SET email_address = 'aboba1336@gmail.com' WHERE full_name = 'Тухлий Дмитро';

3

SELECT * FROM CUSTOMER WHERE email_address = 'aboba1336@gmail.com';

Data Output

Messages

Notifications

≡+

📄

▼

📋

🗑

🗄

⬇

📈

SQL

	customer_id [PK] integer	full_name character varying (100)	emailAddress character varying (100)	phone_number character varying (15)
1	3	Тухлий Дмитро	aboba1336@gmail.com	380672283001

6

SELECT * FROM BOOKING WHERE booking_id = 2;

7

DELETE FROM TICKET WHERE booking_id = 2;

8

DELETE FROM BOOKING WHERE booking_id = 2;

9

SELECT * FROM BOOKING WHERE booking_id = 2;

Data Output

Messages

Notifications

≡ +

▼

▼

SQL

booking_id

[PK] integer

customer_id

integer

booking_date

timestamp with time zone

total_amount

numeric (8,2)

Підрахунок кожної таблиці:

	таблиця text	записів bigint
1	CUSTOM...	5
2	HALL	3
3	MOVIE	6
4	SEAT	27
5	SESSION	3
6	BOOKING	3
7	TICKET	4

Висновок:

1. Операційна ефективність бази даних підтверджена:

Належна обробка DML: Усі ключові операції маніпулювання даними (SELECT, INSERT, UPDATE, DELETE) були виконані успішно, демонструючи стабільність системи при транзакційних навантаженнях.

Чітке дотримання обмежень: Механізми цілісності даних працюють бездоганно, забезпечуючи надійність введеної інформації.

Повна підтримка бізнес-логіки: Розроблена схема коректно відображає та обслуговує всі основні процеси кінотеатру, від продажу квитків до планування розкладу.

2. Вимоги лабораторної практики повністю виконані:

Достатнє наповнення даними: Кожна таблиця містить мінімально необхідну кількість тестових записів (від 3 до 5 і більше).

Демонстрація функціоналу: Успішно показано різноманітні запити SELECT з умовами фільтрації та всі оператори зміни даних (INSERT, UPDATE, DELETE), включаючи перевірку результатів.

Підтвердження цілісності зв'язків: На прикладі помилки зовнішнього ключа під час спроби видалення батьківського запису (SQL State 23503) продемонстровано коректне спрацювання обмежень FOREIGN KEY.

3. Схема є логічно завершеною та надійною:

Повнота сутностей: Усі необхідні інформаційні одиниці та їхні атрибути були включені до фінальної схеми.

Точність зв'язків: Встановлено правильні реляційні зв'язки між усіма об'єктами, що дозволяє уникнути дублювання даних та гарантує точність звітів.

Реалізація бізнес-правил: Усі бізнес-правила (наприклад, унікальність пари 'сеанс + місце') реалізовані через механізми бази даних (наприклад, Композитні UNIQUE обмеження).