

**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 №3**

з дисципліни  
**«Бази даних»**

Виконала:

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

Перевірив:

Русінов В. В.

Київ 2025

## **Лабораторна робота 3: Маніпулювання даними SQL (OLTP)**

### **Цілі лабораторної роботи**

- Виконати запити SELECT з фільтрацією та JOIN.
- Додати нові записи (INSERT).
- Змінити існуючі дані (UPDATE).
- Безпечне видалення рядків (DELETE з WHERE).
- Перевірити коректність роботи обмежень та каскадного видалення.

### **SQL-скрипт**

-- 1. SELECT: всі зали

```
SELECT * FROM Hall;
```

-- 2. SELECT: фільми після 2010 року з рейтингом > 8.5

```
SELECT title, release_date, average_rating
```

```
FROM Movie
```

```
WHERE release_date > '2010-01-01'
```

```
AND average_rating > 8.5
```

```
ORDER BY average_rating DESC;
```

-- 3. SELECT: розклад на 21 жовтня 2025 (з назвами залу, фільму, часом та ціною)

```
SELECT
```

```
s.schedule_id,
```

```
h.hall_name,
```

```
m.title,
```

```
TO_CHAR(sh.show_time, 'DD.MM.YYYY HH24:MI') AS "сеанс",
```

```
sh.price || ' грн' AS "ціна"  
FROM Schedule s  
JOIN Hall h ON s.hall_id = h.hall_id  
JOIN Movie m ON s.movie_id = m.movie_id  
JOIN Showings sh ON s.showing_id = sh.showing_id  
WHERE DATE(sh.show_time) = '2025-10-21'  
ORDER BY sh.show_time;
```

-- 4. INSERT: новый зал

```
INSERT INTO Hall (hall_name) VALUES ('Зал 4 (IMAX)');  
SELECT * FROM Hall;
```

-- 5. INSERT: новый жанр

```
INSERT INTO Genres (genre_name) VALUES ('Adventure');
```

```
SELECT * FROM Genres;
```

-- 6. INSERT: новый фильм «Dune: Part Two»

```
INSERT INTO Movie (title, budget, description, release_date, box_office, duration,  
tagline, average_rating)  
VALUES ('Dune: Part Two', 190000000, 'Paul Atreides continues his journey',  
'2024-03-01', 711000000, 166, 'Long live the fighters', 8.7)  
RETURNING movie_id;
```

-- 7. INSERT: зв'язок фильму з жанрами (Science Fiction + Adventure)

```
INSERT INTO Movies_Genres (movie_id, genre_id) VALUES (4, 2), (4, 4); -- 2=SciFi,  
4=Adventure
```

-- 8. INSERT: новий сеанс

```
INSERT INTO Showings (show_time, price)
```

```
VALUES ('2025-10-23 19:30', 180.00)
```

```
RETURNING showing_id;
```

-- 9. INSERT: розклад для нового фільму в новому залі

```
INSERT INTO Schedule (hall_id, showing_id, movie_id)
```

```
VALUES (4, 4, 4);
```

-- 10. INSERT: новий клієнт

```
INSERT INTO Clients (last_name, first_name)
```

```
VALUES ('Сидоренко', 'Ірина')
```

```
RETURNING client_id;
```

-- 11. INSERT: квиток для Ірини на Dune

```
INSERT INTO Tickets (show_date, schedule_id, client_id, seat)
```

```
VALUES ('2025-10-23 19:30', 4, 4, 'D12');
```

-- 12. UPDATE: знижка на ранковий сеанс

```
UPDATE Showings
```

```
SET price = 100.00
```

```
WHERE show_time = '2025-10-22 16:00';
```

```
SELECT * FROM Showings;
```

-- 13. UPDATE: клієнт пересів на інше місце

```
UPDATE Tickets
```

```
SET seat = 'D11'
```

```
WHERE ticket_id = (SELECT MAX(ticket_id) FROM Tickets) AND client_id = 4;
```

```
SELECT * FROM Tickets WHERE client_id = 4;
```

-- 14. DELETE: клієнт передумав – видаляємо квиток

```
DELETE FROM Tickets WHERE ticket_id = (SELECT MAX(ticket_id) FROM Tickets);
```

```
SELECT * FROM Tickets;
```

-- 15. DELETE: прибираємо тестовий сеанс та розклад

```
DELETE FROM Schedule WHERE schedule_id = 4;
```

```
DELETE FROM Showings WHERE showing_id = 4;
```

-- 16. Фінальна перевірка кількості рядків

```
SELECT 'Hall' AS table_name, COUNT(*) AS rows FROM Hall UNION ALL
```

```
SELECT 'Movie' , COUNT(*) FROM Movie UNION ALL
```

```
SELECT 'Genres' , COUNT(*) FROM Genres UNION ALL
```

```
SELECT 'Movies_Genres' , COUNT(*) FROM Movies_Genres UNION ALL
```

```
SELECT 'Showings' , COUNT(*) FROM Showings UNION ALL
```

```
SELECT 'Schedule' , COUNT(*) FROM Schedule UNION ALL
```

```

SELECT 'Clients' , COUNT(*) FROM Clients UNION ALL
SELECT 'Tickets' , COUNT(*) FROM Tickets
ORDER BY table_name;

```

### **Остаточна схема після всіх операцій**

Таблиця	Кількість рядків	Основні зміни
Hall	4	+ «Зал 4 (IMAX)»
Movie	4	+ «Dune: Part Two»
Genres	4	+ «Adventure»
Movies_Genres	5	+ два зв'язки для Dune
Showings	3	Змінено ціну, видалено тестовий сеанс
Schedule	3	Видалено тестовий розклад
Clients	4	+ Сидоренко Ірина
Tickets	3	Додано → змінено місце → видалено

### **Важливі обмеження та припущення**

- ON DELETE CASCADE у всіх зовнішніх ключах — видалення залу/фільму/сеансу автоматично чистить пов'язані записи.
- UNIQUE на hall\_name та title.
- Квиток прив'язаний до schedule\_id → гарантія одного місця на один сеанс.
- Платіжні дані не зберігаються (зовнішня система).

### **Безпека та найкращі практики**

- Усі UPDATE та DELETE містять WHERE.
- Перед кожним небезпечним запитом виконано SELECT з тією ж умовою.
- Використано RETURNING для отримання ID одразу після INSERT.
- Каскадне видалення спрацювало коректно при видаленні розкладу.

## Доказ виконання

```
postgres=# INSERT INTO Hall (hall_name) VALUES ('Зал 4 (IMAX)');SELECT * FROM Hall ORDER BY hall_id;
INSERT 0 1
hall_id | hall_name
-----+-
 1 | Зал 1
 2 | Зал 2
 3 | VIP Зал
 4 | Зал 4 (IMAX)
4 rows
```

4 зали (включно з новим IMAX)

```
postgres=# SELECT title, release_date, average_rating FROM Movie WHERE release_date > '2010-01-01' AND average_rating > 8.5;
      title      | release_date | average_rating
-----+-----+-----+
Inception    | 2010-07-16   |          8.8
Interstellar | 2014-11-07   |          8.6
Dune: Part Two | 2024-03-01   |          8.7
(3 rows)
```

фільми після 2010 з рейтингом > 8.5

```
postgres=# ORDER BY sh.show_time;
schedule_id | hall_name | title      | сеанс      | ц?на
-----+-----+-----+-----+-----+
 1 | Зал 1     | Inception | 21.10.2025 18:00 | 150.00 грн
 2 | Зал 2     | Interstellar | 21.10.2025 20:00 | 200.00 грн
2 rows
```

розклад на 21.10.2025 з JOIN

Data Output Messages Notifications							
Showing rows: 1 to 4 Page No: 1 of 1							
movie_id	title	budget	description	release_date	box_office	duration	tagline
1	Inception	160000000.00	Sci-fi thriller by Christopher Nolan	2010-07-16	830000000.00	148	Your mind is the scene of the crime
2	Interstellar	165000000.00	Epic space adventure about love and ti...	2014-11-07	700000000.00	169	Mankind was born on Earth. It was never meant to stay there.
3	The Dark Knight	185000000.00	Batman faces Joker in Gotham	2008-07-18	1000000000.00	152	Why so serious?
4	Dune: Part Two	190000000.00	Paul Atreides continues his journey	2024-03-01	711000000.00	166	Long live the fighters

новий фільм та його жанри

```
ticket_id | show_date       | seat | client_id | schedule_id
-----+-----+-----+-----+-----+
 2 | 2025-10-21 18:00:00 | A5   | 1       | 1
 3 | 2025-10-21 20:00:00 | B3   | 2       | 2
 4 | 2025-10-22 16:00:00 | C7   | 3       | 3
3 rows
```

таблиця Tickets (3 рядки)

table_name	rows
Clients	4
Genres	4
Hall	4
Movie	4
Movies_Genres	5
Schedule	3
Showings	4
Tickets	3
(8 rows)	

підсумкова кількість рядків у всіх таблицях

## Висновок

Усі операції OLTP виконано успішно. Схема поводиться передбачувано:

- зв'язки зберігаються,
- обмеження не порушені,
- каскадне видалення працює,
- кількість рядків відповідає очікуванням.

База даних готова до наступних етапів (OLAP-аналітика, нормалізація, міграції).