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 №4

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 4: Аналітичні SQL-запити (OLAP)

Цілі лабораторної роботи

- Навчитись використовувати агрегатні функції (COUNT, SUM, AVG, MIN, MAX).
- Оволодіти GROUP BY та HAVING.
- Виконувати різні типи JOIN.
- Писати підзапити в WHERE, SELECT, HAVING.
- Створювати аналітичні звіти (OLAP).

SQL-скрипт

```
-- 1. Загальна кількість проданих квитків
SELECT COUNT(*) AS total_tickets_sold FROM Tickets;

-- 2. Середня ціна квитка
SELECT ROUND(AVG(price), 2) AS avg_ticket_price FROM Showings;

-- 3. Кількість проданих квитків по кожному залу (GROUP BY)
SELECT
    h.hall_name,
    COUNT(t.ticket_id) AS tickets_sold
FROM Hall h
LEFT JOIN Schedule s ON h.hall_id = s.hall_id
LEFT JOIN Tickets t ON s.schedule_id = t.schedule_id
GROUP BY h.hall_id, h.hall_name
ORDER BY tickets_sold DESC;
```

```sql
-- 4. Зали, в яких продано більше 1 квитка (HAVING)
SELECT
    h.hall_name,
    COUNT(t.ticket_id) AS tickets_sold
FROM Hall h
JOIN Schedule s ON h.hall_id = s.hall_id
JOIN Tickets t ON s.schedule_id = t.schedule_id
GROUP BY h.hall_id, h.hall_name
HAVING COUNT(t.ticket_id) > 1;
-- 5. Виручка по кожному фільму (багатотаблична агрегація)
SELECT
    m.title,
    COUNT(t.ticket_id) AS tickets_sold,
    COALESCE(SUM(sh.price), 0) AS total_revenue
FROM Movie m
JOIN Schedule s ON m.movie_id = s.movie_id
JOIN Showings sh ON s.showing_id = sh.showing_id
LEFT JOIN Tickets t ON s.schedule_id = t.schedule_id
GROUP BY m.movie_id, m.title
ORDER BY total_revenue DESC;

-- 6. Розклад сеансів (INNER JOIN)
SELECT
    h.hall_name,
```

```
    m.title,
    TO_CHAR(sh.show_time, 'DD.MM.YYYY HH24:MI') AS show_time,
    sh.price || ' ₴' AS price
FROM Schedule s
INNER JOIN Hall h ON s.hall_id = h.hall_id
INNER JOIN Movie m ON s.movie_id = m.movie_id
INNER JOIN Showings sh ON s.showing_id = sh.showing_id
ORDER BY sh.show_time;

-- 7. Усі фільми, навіть без показів (LEFT JOIN)
SELECT
    m.title,
    COUNT(s.schedule_id) AS scheduled_shows
FROM Movie m
LEFT JOIN Schedule s ON m.movie_id = s.movie_id
GROUP BY m.movie_id, m.title
ORDER BY scheduled_shows DESC;

-- 8. Клієнти, які купили квиток на фільм з рейтингом > 8.7 (підзапит у WHERE)
SELECT c.last_name, c.first_name, m.title
FROM Clients c
JOIN Tickets t ON c.client_id = t.client_id
JOIN Schedule s ON t.schedule_id = s.schedule_id
JOIN Movie m ON s.movie_id = m.movie_id
```

```
WHERE m.movie_id IN (
    SELECT movie_id FROM Movie WHERE average_rating > 8.7
)
ORDER BY c.last_name;

-- 9. Порівняння виручки фільму з середньою ціною квитка (підзапит у SELECT)
SELECT
    m.title,
    COALESCE(SUM(sh.price), 0) AS revenue,
    (SELECT ROUND(AVG(price), 2) FROM Showings) AS avg_price_all
FROM Movie m
JOIN Schedule s ON m.movie_id = s.movie_id
JOIN Showings sh ON s.showing_id = sh.showing_id
LEFT JOIN Tickets t ON s.schedule_id = t.schedule_id
GROUP BY m.movie_id, m.title
ORDER BY revenue DESC;

-- 10. Фільми з виручкою вище середньої (підзапит у HAVING)
SELECT
    m.title,
    SUM(sh.price) AS total_revenue
FROM Movie m
JOIN Schedule s ON m.movie_id = s.movie_id
JOIN Showings sh ON s.showing_id = sh.showing_id
```

```
GROUP BY m.movie_id, m.title

HAVING SUM(sh.price) > (

    SELECT AVG(price) * COUNT(*) FROM Showings

)

ORDER BY total_revenue DESC;
```

№	**Що робить запит**	**Ключові конструкції**
1	Загальна кількість проданих квитків	COUNT(*)
2	Середня ціна квитка	AVG(), ROUND()
3	Кількість квитків по залах	GROUP BY, LEFT JOIN
4	Зали з >1 квитком	HAVING
5	Виручка по фільмах	SUM(), багатотаблична агрегація
6	Повний розклад	INNER JOIN
7	Усі фільми (з/без показів)	LEFT JOIN
8	Клієнти топ-фільмів	підзапит у WHERE
9	Порівняння з середньою ціною	підзапит у SELECT
10	Фільми з виручкою > середньої	підзапит у HAVING

Виконано всі вимоги:

- ≥4 запити з агрегатами та GROUP BY
- ≥3 запити з різними JOIN
- ≥3 підзапити
- Усі запити виконані без помилок у pgAdmin

Виконання:

```
postgres=# SELECT COUNT(*) AS total_tickets_sold FROM Tickets;
 total_tickets_sold
--------------------
                  3
(1 row)
```

Квитки

```
postgres=# SELECT ROUND(AVG(price), 2) AS avg_ticket_price FROM Showings;
 avg_ticket_price
------------------
           157.50
(1 row)
```

Середня ціна

```
  hall_name    | tickets_sold
---------------+--------------
 Зал 1         |            1
 VIP Зал       |            1
 Зал 2         |            1
 Зал 4 (IMAX)  |            0
(4 rows)
```

По залах

```
      title       | tickets_sold | total_revenue
------------------+--------------+---------------
 Interstellar     |            1 |        200.00
 Inception        |            1 |        150.00
 The Dark Knight  |            1 |        100.00
(3 rows)
```

Виручка

```
hall_name |      title       |    show_time     |  price
----------+------------------+------------------+----------
Зал 1     | Inception        | 21.10.2025 18:00 | 150.00 ?
Зал 2     | Interstellar     | 21.10.2025 20:00 | 200.00 ?
VIP Зал   | The Dark Knight  | 22.10.2025 16:00 | 100.00 ?
(3 rows)
```

Розклад

```
postgres=# ORDER BY scheduled_shows DESC;
      title       | scheduled_shows
------------------+-----------------
 The Dark Knight |               1
 Inception       |               1
 Interstellar    |               1
 Dune: Part Two  |               0
(4 rows)
```

Усі фільми

```
 last_name | first_name |      title
-----------+------------+-----------------
 ?ваненко  | Олег       | Inception
 Коваль    | Андр?й     | The Dark Knight
(2 rows)
```

Клієнти топ-фільмів

```
      title       | revenue | avg_price_all
------------------+---------+---------------
 Interstellar    |  200.00 |        157.50
 Inception       |  150.00 |        157.50
 The Dark Knight |  100.00 |        157.50
(3 rows)
```

```
postgres=# ORDER BY total_rev
 title | total_revenue
-------+---------------
(0 rows)
```

Виручка > середньої

Висновок

Усі аналітичні запити успішно виконані. Навчилась:

- будувати звіти про виручку та відвідуваність
- працювати з GROUP BY / HAVING

- комбінувати таблиці через JOIN
- використовувати підзапити для складної фільтрації

База даних готова до створення дашбордів та бізнес-аналітики.