

Лабораторна робота №3

Виконав студент групи МІТ-31
Тимохін Роман Миколайович

Мета роботи

Закріплення теоретичних знань та набуття практичних навичок у складанні складних SQL-запитів до реляційної бази даних. Робота передбачає використання операторів SELECT, WHERE, логічних операторів, агрегатних функцій, JOIN, підзапитів, CTE, віконних функцій тощо.

Завдання

Розробити 40 SQL-запитів до створеної у попередній роботі бази даних, що охоплюють наступні аспекти:

Логічні оператори (AND, OR, NOT)

```
-- 1. Користувачі з роллю 'customer' і поштою, що містить '@gmail'
SELECT * FROM users
WHERE role = 'customer' AND email LIKE '%@gmail%';
```

```
-- 2. Товари, що дорожчі за 500 або належать до категорії з id 1
SELECT * FROM products
WHERE price > 500 OR category_id = 1;
```

```
-- 3. Замовлення, які не мають статусу "Оплачено"
SELECT * FROM orders
WHERE NOT status = 'Оплачено';
```

Агрегатні функції (COUNT, SUM, AVG, MIN, MAX)

```
-- 4. Кількість користувачів
SELECT COUNT(*) FROM users;
```

```
-- 5. Середня ціна товарів
```

```
SELECT AVG(price) FROM products;
```

```
-- 6. Найдорожчий товар  
SELECT MAX(price) FROM products;
```

```
-- 7. Мінімальна сума платежу  
SELECT MIN(amount) FROM payments;
```

```
-- 8. Загальна сума всіх платежів  
SELECT SUM(amount) FROM payments;
```

JOIN-и (INNER, LEFT, RIGHT, FULL, CROSS, SELF)

```
-- 9. INNER JOIN: Товари з назвами категорій  
SELECT p.product_name, c.category_name  
FROM products p  
INNER JOIN categories c ON p.category_id = c.category_id;
```

```
-- 10. LEFT JOIN: Всі товари з інформацією про категорію (навіть без  
категорії)  
SELECT p.product_name, c.category_name  
FROM products p  
LEFT JOIN categories c ON p.category_id = c.category_id;
```

```
-- 11. RIGHT JOIN: Усі категорії з наявними товарами  
SELECT p.product_name, c.category_name  
FROM products p  
RIGHT JOIN categories c ON p.category_id = c.category_id;
```

```
-- 12. FULL JOIN: Всі товари й категорії незалежно від зв'язку  
SELECT p.product_name, c.category_name  
FROM products p  
FULL JOIN categories c ON p.category_id = c.category_id;
```

```
-- 13. CROSS JOIN: Всі можливі комбінації товарів і категорій  
SELECT p.product_name, c.category_name  
FROM products p  
CROSS JOIN categories c;
```

```
-- 14. SELF JOIN: Порівняння товарів однієї категорії
SELECT p1.product_name AS product1, p2.product_name AS product2
FROM products p1
JOIN products p2 ON p1.category_id = p2.category_id AND
p1.product_id < p2.product_id;
```

Підзапити (IN, NOT IN, EXISTS, NOT EXISTS)

```
-- 15. Користувачі, які мають хоча б одне замовлення
SELECT * FROM users
WHERE user_id IN (SELECT user_id FROM orders);
```

```
-- 16. Користувачі, які не мають жодного замовлення
SELECT * FROM users
WHERE user_id NOT IN (SELECT user_id FROM orders);
```

```
-- 17. Товари, які були замовлені хоча б раз
SELECT * FROM products
WHERE EXISTS (
    SELECT 1 FROM order_items WHERE order_items.product_id =
products.product_id
);
```

```
-- 18. Товари, які ніколи не замовлялись
SELECT * FROM products
WHERE NOT EXISTS (
    SELECT 1 FROM order_items WHERE order_items.product_id =
products.product_id
);
```

Операції над множинами (UNION, INTERSECT, EXCEPT)

```
-- 19. UNION: користувачі з email @gmail або @ukr.net
SELECT username FROM users WHERE email LIKE '%@gmail%'
UNION
SELECT username FROM users WHERE email LIKE '%@ukr.net%';
```

```
-- 20. INTERSECT: користувачі з email @gmail і роллю 'customer'
SELECT username FROM users WHERE email LIKE '%@gmail%'
INTERSECT
SELECT username FROM users WHERE role = 'customer';

-- 21. EXCEPT: всі користувачі крім модераторів
SELECT username FROM users
EXCEPT
SELECT username FROM users WHERE role = 'moderator';
```

CTE (Common Table Expressions)

```
-- 22. CTE: сума замовлень по користувачах
WITH order_totals AS (
    SELECT user_id, SUM(amount) AS total_spent
    FROM payments
    JOIN orders ON payments.order_id = orders.order_id
    GROUP BY user_id
)
SELECT u.username, ot.total_spent
FROM users u
JOIN order_totals ot ON u.user_id = ot.user_id;
```

Віконні функції (Window Functions)

```
-- 23. Рейтинг користувачів за витраченими грошима
SELECT
    u.username,
    SUM(p.amount) AS total_paid,
    RANK() OVER (ORDER BY SUM(p.amount) DESC) AS user_rank
FROM users u
JOIN orders o ON u.user_id = o.user_id
JOIN payments p ON o.order_id = p.order_id
GROUP BY u.username;

-- 24. Сума замовлення та середня сума по всім
SELECT
    o.order_id,
```

```

    p.amount,
    AVG(p.amount) OVER () AS avg_payment
FROM payments p
JOIN orders o ON p.order_id = o.order_id;

```

```

1  -- 1. Користувачі з роллю 'customer' і поштою, що містить '@gmail'
2  SELECT * FROM users
3  WHERE role = 'customer' AND email LIKE '%@gmail%';

```

Data Output Сообщения Notifications

	user_id [PK] integer	username character varying (50)	email character varying (100)	role character varying (20)
1	3	petro	petro123@gmail.com	customer
2	5	ihor	ihor88@gmail.com	customer

```

1
2  -- 2. Товари, що дорожчі за 500 або належать до категорії з id 1
3  SELECT * FROM products
4  WHERE price > 500 OR category_id = 1;

```

Data Output Сообщения Notifications

	product_id [PK] integer	product_name character varying (100)	price numeric (10,2)	category_id integer
1	1	Навушники	1500.00	1
2	2	Ноутбук	35000.00	1
3	5	Навушники	1500.00	1
4	6	Ноутбук	35000.00	1
5	9	Смартфон Samsung A55	12000.00	1
6	10	Навушники JBL	1800.00	1
7	11	Футболка Nike	750.00	2
8	14	Штани Adidas	1200.00	2

Висновок

У процесі виконання лабораторної роботи №3 було розроблено та реалізовано 40 SQL-запитів до бази даних інтернет-магазину, що охоплюють широкий спектр можливостей мови SQL. Особливу увагу приділено використанню логічних операторів, агрегатних функцій, усіх типів з'єднань таблиць (JOIN), підзапитів, операцій над множинами, загальних табличних виразів (CTE) та віконних функцій. Завдяки цьому вдалося поглибити розуміння складних механізмів обробки даних та ефективного їх аналізу у реляційних СУБД, що є важливою складовою сучасних інформаційних систем.