

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра прикладної математики



Звіт

з лабораторної роботи № 2

з дисципліни «Організація баз даних та знань»

Виконала:

Студентка групи ПМ-22

Торська Марія

Перевірив:

Алексєєв В.І.

Львів – 2023р.

Тема: «Написання SQL запитів»

Варіант №1

1. Запит повертає інформацію про студенток, які займаються плаванням або баскетболом

```
SELECT s.stud_id, s.last_name, s.first_name, s.surname, s.hobby FROM students s
WHERE (s.hobby = 'Плавання' OR s.hobby = 'Баскетбол') AND s.gender = 'ж'
ORDER BY s.last_name
```

	stud_id	last_name	first_name	surname	hobby
1	2	Василишин	Анна	Антонівна	Баскетбол
2	3	Шульга	Анастасія	Кирилівна	Плавання

Запит вибирає поля stud_id, last_name, first_name, surname та hobby з таблиці 'students' і сортує результати по полю last_name в алфавітному порядку. Вибір рядків задається в умові WHERE, яка містить дві умови, що об'єднуються за допомогою оператора AND.

2. Запит повертає список студентів, які мають рейтинг від X до Y

```
SELECT s.stud_id, s.last_name, s.first_name, s.surname, AVG(success.score) AS rating
FROM students s
INNER JOIN success ON s.stud_id = success.stud_id
GROUP BY s.stud_id, s.last_name, s.first_name, s.surname
HAVING AVG(success.score) > 80 AND AVG(success.score) < 90
ORDER BY rating DESC;
```

	stud_id	last_name	first_name	surname	rating
1	2	Василишин	Анна	Антонівна	89
2	1	Давидюк	Василь	Петрович	85
3	3	Шульга	Анастасія	Кирилівна	83

Запит вибирає п'ять полів із таблиці 'students': stud_id, last_name, first_name, surname та rating. Поле rating обчислюється як середнє арифметичне score з таблиці 'success' для кожного студента. Далі виконується INNER JOIN між таблицею 'students' та 'success', який з'єднує записи за stud_id. GROUP BY об'єднує отримані записи за stud_id, last_name, first_name та surname, щоб мати змогу застосувати функцію AVG до поля score з таблиці 'success'. HAVING фільтрує результати, щоб включити тільки тих студентів, чий рейтинг знаходиться між 80 та 90. Остаточні результати сортуються у порядку спадання значень rating.

3. Запит повертає інформацію про групи та їхніх студентів

```
SELECT g.group_name, s.last_name, s.first_name, s.surname FROM students s
INNER JOIN stud_groups sg ON s.stud_id = sg.stud_id
INNER JOIN groups g ON g.group_id = sg.group_id
ORDER BY g.group_name, s.last_name;
```

	group_name	last_name	first_name	surname
1	ПМ-32	Баворовський	Олександр	Борисович
2	ПМ-32	Давидюк	Василь	Петрович
3	ПМ-32	Шульга	Анастасія	Кирилівна

Запит виконується через об'єднання таблиць 'students', 'stud_groups' та 'groups' за допомогою операторів INNER JOIN. Об'єднання виконується за умови, що stud_id з таблиць 'students' та 'stud_groups' співпадають, а group_id співпадає у таблицях 'groups' та 'stud_groups'. Таким чином, у результаті будуть лише ті групи, в яких є хоча б один студент, та студенти, що належать до певної групи. Кінцевий результат запиту відсортований за назвою групи та прізвищем студента.

- Запит повертає список студентів з 1998 по 2003 року народження, що проживають у гуртожитку та які мають оцінки вище 88 і студентський з номером 001...

```
SELECT s.last_name, s.first_name, s.surname, s.stud_number, s.birth_year FROM students s
WHERE s.stud_number LIKE '001%'
AND s.birth_year BETWEEN 1998 AND 2003
AND s.stud_id IN (SELECT stud_id FROM stud_rooms WHERE check_out_date IS NULL)
AND EXISTS (
  SELECT score
  FROM success
  WHERE success.stud_id = s.stud_id
  AND score > 88
)
```

	last_name	first_name	surname	stud_number	birth_year
1	Давидюк	Василь	Петрович	00124586	2002
2	Василишин	Анна	Антонівна	00124545	2002

Запит вибирає поля last_name, first_name, surname, stud_number та birth_year з таблиці 'students'. За допомогою оператора LIKE обираються студенти, чий номер студентського починається на 001, BETWEEN відбирає записи, у яких значення року народження знаходяться в діапазоні від 1998 по 2003. Оператор IN відсортовує тих студентів, які мають запис про проживання в гуртожитку, а EXISTS перевіряє чи є у студента записи у таблиці 'success' з оцінкою вище 88.

- Запит повертає список студентів, що мали хоча б одну оцінку 'добре'

```
SELECT s.last_name, s.first_name, s.surname
FROM students s
WHERE stud_id = ANY (
  SELECT stud_id
  FROM success
  WHERE grade = 'добре'
)
```

	last_name	first_name	surname
1	Давидюк	Василь	Петрович
2	Василишин	Анна	Антонівна
3	Шульга	Анастасія	Кирилівна

Оператор ANY порівнює значення стовпця stud_id зі значеннями, які повертає підзапит, що містить всі stud_id студентів, які мають записи з оцінкою 'добре' в таблиці 'success'.

- Запит повертає інформацію про студентів, у яких всі оцінки більше за 88

```
SELECT s.stud_id, s.last_name, s.first_name, s.surname
FROM students s
WHERE 88 <= ALL (
```

```

SELECT score
FROM success
WHERE stud_id = s.stud_id
)

```

	stud_id	last_name	first_name	surname
1	4	Баворовський	Олександр	Борисович

Підзапит обирає всі значення score з таблиці 'success', де stud_id збігається з значенням stud_id з таблиці 'students'. Оператор ALL перевіряє, чи всі значення з підзапиту є більшими або рівними 88 для кожного студента, що вибирається у зовнішньому запиті.

- Запит повертає список студентів, що брали книги в бібліотеці, з підсумуванням кількості книг та їх загальної вартості

```

SELECT s.stud_id, s.last_name, s.first_name, COUNT(number) AS books, SUM(price) AS total_spent
FROM students s
INNER JOIN borrowed_books bb ON s.stud_id = bb.stud_id AND bb.return_date IS NULL
INNER JOIN instances i ON bb.instance_id = i.instance_id
INNER JOIN books b ON i.book_id = b.book_id
GROUP BY s.stud_id, s.last_name, s.first_name;

```

	stud_id	last_name	first_name	books	total_spent
1	1	Давидюк	Василь	2	70
2	2	Василишин	Анна	1	35

Спочатку за допомогою оператора JOIN виконується внутрішнє з'єднання 'students' з таблицею 'borrowed_books', за умови, що stud_id співпадають та дата повернення книги дорівнює NULL. Далі, щоб отримати інформацію про книгу, поєднуються таблиці 'borrowed_books' та 'instances', якщо їх значення instance_id однакові. Потім, оператор JOIN за значенням book_id поєднує таблицю 'instances' з таблицею 'books', щоб отримати повну інформацію про книгу. Оператор GROUP BY групує рядки за stud_id, last_name та first_name і рахує кількість книг та їх загальну вартість за допомогою функцій COUNT та SUM.

- Запит надає список студентів, що не повернули книги в бібліотеку

```

SELECT s.stud_id, s.last_name, s.first_name, s.surname FROM students s
WHERE s.stud_id IN (
    SELECT bb.stud_id FROM borrowed_books bb
    WHERE bb.return_date IS NULL
)
GROUP BY s.stud_id, s.last_name, s.first_name, s.surname

```

	stud_id	last_name	first_name	surname
1	1	Давидюк	Василь	Петрович
2	2	Василишин	Анна	Антонівна

Оператор IN вибирає ідентифікатори студентів, які мають запис про взяті та ще не повернені книги, а умова WHERE повертає всіх студентів, чий ідентифікатор збігається зі значеннями, які були отримані з підзапиту.

- Запит повертає список студентів за кімнатами в гуртожитку

```

SELECT student_name.stud_id, student_name.full_name, rooms.room_name

```

```

FROM (
    SELECT stud_id, CONCAT(last_name, ' ', first_name, ' ', surname) AS full_name
    FROM students
) AS student_name
INNER JOIN stud_rooms ON student_name.stud_id = stud_rooms.stud_id AND stud_rooms.check_out_date
IS NULL
INNER JOIN rooms ON rooms.room_id = stud_rooms.room_id
ORDER BY student_name.full_name;

```

	stud_id	full_name	room_name
1	2	Василишин Анна Антонівна	1
2	1	Давидюк Василь Петрович	1

У підзапиті в частині FROM таблиця 'students' використовується для створення нового стовпця full_name, який містить повне ім'я студента складене зі стовпців last_name, first_name та surname. Спочатку за допомогою оператора JOIN виконується внутрішнє з'єднання 'student_name' з таблицею 'stud_rooms', за умови, що stud_id співпадають та дата виселення дорівнює NULL. Далі, щоб отримати інформацію про номер кімнати, поєднуються таблиці 'stud_rooms' та 'rooms', якщо їх значення room_id однакові. Результат сортується за повним ім'ям студента.

10. Запит повертає список максимальних балів кожного студента з певних предметів

```

SELECT stud_id, last_name, [1] AS Математичний_анализ, [2] AS Чисельні_методи, [3] AS
Нейронні_мережі
FROM (
    SELECT students.stud_id, students.last_name, success.course_id, success.score
    FROM success
    INNER JOIN students ON students.stud_id = success.stud_id
) AS s
PIVOT (
    MAX(score)
    FOR course_id IN ([1], [2], [3])
) AS p

```

	stud_id	last_name	Математичний_анализ	Чисельні_методи	Нейронні_мережі
1	1	Давидюк	88	80	89
2	2	Василишин	94	83	91
3	3	Шульга	81	90	79
4	4	Баворовський	95	100	88

У підзапиті в частині FROM вибираються ідентифікатор, прізвище студента та ідентифікатор курсу, з якого цей студент отримав бали. Оператор PIVOT створює нову таблицю, яка містить стовпці "Математичний_анализ", "Чисельні_методи" та "Нейронні_мережі", кожен з яких містить оцінки успішності студентів з відповідного курсу.

11. Запит змінює адресу проживання для студента з ідентифікатором 3

```

UPDATE students
SET address = 'вул.Виговського 56/4'
WHERE stud_id = 3

```

	stud_id	last_name	first_name	surname	stud_number	birth_year	birth_place	address	gender	marital_status	scholarship	hobby
1	1	Давидюк	Василь	Петрович	00124586	2002	м.Львів	вул.Любінська 24/15	ч	неодружений	2000.00	Плавання
2	2	Василишин	Анна	Антонівна	00124545	2002	м.Яворів	вул.І.Франка 12/3	ж	неодружена	2910.00	Баскетбол
3	3	Шульга	Анастасія	Кирилівна	00568495	2003	м.Львів	вул.Виговського 56/4	ж	неодружена	2000.00	Плавання
4	4	Баворовський	Олександр	Борисович	00485976	2002	м.Рівне	вул.Т.Шевченка 85а/163	ч	неодружений	2910.00	-

12. Запит встановлює дату повернення книги для студента з номером студентського '00124586'

```
UPDATE borrowed_books
SET return_date = '2023-03-18'
FROM students
WHERE students.stud_id = borrowed_books.stud_id
AND students.stud_number = '00124586'
AND borrowed_books.return_date IS NULL;
```

	stud_id	instance_id	loan_date	return_date
1	1	2	2023-03-01	2023-03-18
2	1	4	2023-03-01	2023-03-18
3	2	3	2023-03-01	NULL

Оновлення здійснюється тільки для тих записів, де значення stud_id в таблиці 'borrowed_books' відповідає значенню stud_id в таблиці 'students', а також значення стовпця stud_number є '00124586' та return_date дорівнює NULL.

13. Запит додає нові книги в бібліотеку

```
INSERT INTO books(number, author, title, genre, price)
VALUES (1254, 'Г.М.Фіхтенгольц', 'Основи математичного аналізу', 'наукова література', 35),
(1255, 'Ю.К.Рудавський', 'Лінійна алгебра та аналітична геометрія', 'наукова література', 32),
(1236, 'В.Л.Петрушенко', 'Філософія', 'наукова література', 40);
```

	book_id	number	author	title	genre	price
1	1	1254	Г.М.Фіхтенгольц	Основи математичного аналізу	наукова література	35
2	2	1255	Ю.К.Рудавський	Лінійна алгебра та аналітична геометрія	наукова література	32
3	3	1236	В.Л.Петрушенко	Філософія	наукова література	40

14. Запит додає студенту, який є старостою ПМ-32, запис про оцінку 95 за предмет 4

```
INSERT INTO success(stud_id, course_id, score, grade, get_date)
SELECT groups.leader, 4, 95, 'відмінно', '2022-12-12'
FROM groups WHERE groups.group_name = 'ПМ-32'
```

	stud_id	course_id	score	grade	get_date
1	1	1	88	відмінно	2022-12-05
2	1	2	80	добре	2022-12-09
3	1	3	89	відмінно	2022-12-02
4	1	4	95	відмінно	2022-12-12

У підзапиті з таблиці 'groups' обирається значення leader у записі, де назва групи є ПМ-32. Далі, для цього студента додається новий запис у таблицю 'success'

15. Запит видаляє всі записи з таблиці 'stud_rooms'

```
DELETE FROM stud_rooms;
```

16. Запит видаляє з таблиці 'success' записи, для яких ідентифікатор курсу дорівнює 4, а дата отримання оцінки - '2022-12-12'

```
DELETE FROM success
WHERE success.course_id = 4 AND success.get_date = '2022-12-12'
```