

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №8

Підготувала:

Студентка групи КН-209

Кульчицька Олена

Викладач:

Мельникова Н.І.

Лабораторна робота №8

на тему:

“Запити на вибір даних із впорядкуванням та групуванням результатів”

Мета роботи: Розробити SQL-запити для вибору записів з однієї чи кількох таблиць в т. ч. із застосуванням: впорядкування результатів за значеннями одного і декількох полів, впорядкування результатів за зростанням і спаданням, групування результатів за значеннями одного чи декількох полів.

Короткі теоретичні відомості.

Для впорядкування та групування результатів разом з командою SELECT використовують наступні директиви.

ORDER BY {ім'я_поля | синонім | позиція_поля} [ASC | DESC] [, ...] Задає порядок сортування значень у результатах запиту за вказаним полем. ASC – за зростанням, DESC – за спаданням. За замовчуванням сортування відбувається за зростанням значень. Поля можна вказувати за назвою, псевдонімом або номером позиції поля у таблиці.

GROUP BY {ім'я_поля | синонім | позиція_поля} [ASC | DESC] [HAVING умова] [WITH ROLLUP] [, ...]

Групує (і одночасно сортує) рядки за вказаними полями. Поля можна вказувати за іменами, синонімами або порядковими номерами в таблиці.

HAVING вказує умову відбору для групи з використанням як агрегатних так і не агрегатних виразів. Дає можливість застосування до значень полів агрегатних функцій (COUNT, AVG, MIN, MAX тощо) при відборі чи групуванні рядків. Після слова WHERE ці функції не працюють, однак у всіх інших випадках слід використовувати саме WHERE.

WITH ROLLUP додає до результатів запиту рядок із підсумковим значенням для всіх груп.

Хід роботи.

Розробимо та виконаємо такі 3 запити до бази даних.

1. Вивести перелік відвідувачів із ім'ям, що містить "Ol", та прізвищем, що містить "Li" в алфавітному порядку.
2. Вивести перелік 3 найстаріших замовлень системи.
3. Вивести замовлення, згрупувавши їх за іменем клієнта.

- Перевіримо місткість таблиці customer:

```
SELECT * FROM confectionary.customer;
```

Результати запиту:

id	first_name	second_name	telephone	email
1	Olena	Link	8564735	olink@gmail.com
2	Lina	Top	6532458	linavop@gmail.com
3	Lina	Boor	2007896	lina@gmail.com
4	Olia	Lickv	1708967	olickv@mail.com
5	Rob	Runnv	1507894	robrun@mail.com
6	Lara	Norm	9997773	lina@gmail.com
7	Mark	Hurrv	1118765	olickv@mail.com
8	Jan	Woo	8887894	robrun@mail.com
9	Lilv	Raf	abc12	lilv@gmail.com
NULL	NULL	NULL	NULL	NULL

1. Запит вибору записів з таблиці customer із використанням сортування:

```
SELECT first_name, second_name FROM confectionary.customer  
WHERE first_name LIKE 'O%' AND second_name LIKE 'L%'  
ORDER BY first_name ASC;
```

Результати запиту:

	first_name	second_name
	Olena	Link
	Olia	Lickv

- Перевіримо місткість таблиці order:

```
SELECT * FROM confectionary.`order`;
```

Результати запиту:

	id	staff_id	date	time	customer_id
	1	2	2020-03-20	14:15:00	1
	2	2	2020-03-21	14:15:00	2
	3	2	2020-03-23	14:15:00	1
	4	2	2020-03-14	18:30:00	3
	5	2	2020-02-25	14:00:20	8
	6	2	2020-03-07	10:45:20	5
	NULL	NULL	NULL	NULL	NULL

- Запит вибору записів з таблиці order з використанням сортування:

```
SELECT id, date, time FROM confectionary.order
ORDER BY date, time
ASC LIMIT 3;
```

Результати запиту:

	id	date	time
	5	2020-02-25	14:00:20
	6	2020-03-07	10:45:20
	4	2020-03-14	18:30:00
	NULL	NULL	NULL

- Перевіримо місткість селекту. Виберемо замовлення відвідувачів, обслужених персоналом під індексом 2 або 3.

```
SELECT customer.first_name, order.date, order.time
FROM confectionary.order INNER JOIN confectionary.customer
ON order.customer_id = customer.id
WHERE order.staff_id IN (SELECT staff.id FROM confectionary.staff
WHERE staff.id IN ('2','3')) ;
```

	first_name	date	time
	Olena	2020-03-20	14:15:00
	Lina	2020-03-21	14:15:00
	Olena	2020-03-23	14:15:00
	Lina	2020-03-14	18:30:00
	Jan	2020-02-25	14:00:20
	Rob	2020-03-07	10:45:20

- Тепер згрупуємо замовлення за іменем клієнта:

```
SELECT customer.first_name, order.date, order.time
FROM confectionary.order INNER JOIN confectionary.customer
ON order.customer_id = customer.id
WHERE order.staff_id IN (SELECT staff.id FROM confectionary.staff
WHERE staff.id IN ('2','3'))
GROUP BY first_name;
```

	first_name	date	time
	Jan	2020-02-25	14:00:20
	Lina	2020-03-21	14:15:00
	Olena	2020-03-20	14:15:00
	Rob	2020-03-07	10:45:20

Висновок: під час даної лабораторної роботи було розглянуто методи вибору даних з таблиць БД з використанням сортування та групування. Було проведено вибір даних із таблиць customer, order та staff.