

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ 4
з курсу “ОБДЗ”
на тему:
«Запити на додавання, зміну та вилучення даних»

Виконала:

студентка групи КН-211

Лаврик Юліана

Викладач:

Якимишин Х.М.

Лабораторна робота №4

Мета роботи: розробити SQL-запити для внесення нових значень в таблиці в режимі одиничного та групового доповнення; розробити SQL-запити для внесення змін в рядки таблиць; розробити SQL-запити для видалення вибраних рядків.

Короткі теоретичні відомості.

- Внесення значень в таблиці за допомогою директиви **INSERT** .

Синтаксис :

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
[INTO] назва_таблиці [(назва_поля,...)]
| {VALUES | VALUE} ({вираз | DEFAULT},...),(...),...
| [SET назва_поля={вираз | DEFAULT}, ...]
| [вибірка_даних]
[ON DUPLICATE KEY UPDATE назва_поля=вираз [, назва_поля=вираз]...]
```

Аргументи:

VALUES - задає набір значень, які будуть вставлятися у задані поля таблиці з дотриманням вказаного у дужках порядку полів. Якщо перелік полів не задано, то значення потрібно задати для кожного поля таблиці.

DEFAULT - внесення значення за замовчуванням у вказане поле.

вираз - константа, змінна або інший вираз . Наприклад: поле2=поле1*0.5.

вибірка_даних - довільна SELECT-директива, результати виконання якої будуть внесені в таблицю.

ON DUPLICATE KEY UPDATE - якщо при внесенні нового рядка, значення в ключових полях дублюються, то ця директива дозволяє задати вираз, за яким будуть формуватись нові значення полів наявного в таблиці (старого) рядка. При цьому, новий рядок не буде внесений в таблицю.

- Занесення даних в таблицю із зовнішнього файлу :

Синтаксис :

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE
'назва_файлу'
```

[REPLACE | IGNORE]
INTO TABLE назва_таблиці
[CHARACTER SET кодування]
[{FIELDS | COLUMNS}
[TERMINATED BY 'рядок_завершення']
[[OPTIONALLY] ENCLOSED BY 'символ']
[ESCAPED BY 'символ']]
[LINES [STARTING BY 'рядок_початку']
[TERMINATED BY 'рядок_завершення']]
[IGNORE кількість LINES]

- Внесення змін в таблиці за допомогою директиви UPLOAD.

Синтаксис :

UPDATE [LOW_PRIORITY] [IGNORE] назва_таблиці
SET назва_поля1={вираз|DEFAULT} [, назва_поля2={вираз|DEFAULT}]...
[WHERE умова_відбору]
[ORDER BY ...]
[LIMIT кількість_рядків]

Аргументи :

SET - задає список полів, значення яких будуть змінюватись, і відповідні дії над ними.

DEFAULT - вставка значення по замовчуванню у вказане поле.

WHERE - вказує умову, за якою відбираються рядки, що підлягають зміні.

ORDER BY - може вказувати поле, за яким сортувати рядки перед зміною, з метою уникнення порушення цілісності таблиці.

LIMIT - обмежує кількість рядків, які будуть змінені.

- Видалення записів з таблиці за допомогою директиви DELETE.

Синтаксис :

DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM назва_таблиці
[WHERE умова_відбору]
[ORDER BY ...]
[LIMIT кількість_рядків]

Аргументи :

FROM - вказує таблицю, з якої слід видалити значення.

WHERE - вказує умову, за якою відбираються рядки, що підлягають видаленню.

ORDER BY - може вказувати поле, за яким сортувати рядки перед видаленням.

LIMIT - обмежує кількість рядків, які будуть видалені.

Основні оператори та функції MySQL, що можна використовувати при побудові виразів та запитів :

Функція (оператор)	Опис
AND, OR	Логічне “і” та логічне “або”.
~, &, , ^	Побітові операції інверсії, логічного “і”, “або”, “виключного або”.
AVG(), MAX(), MIN()	Повертає середнє, максимальне, або мінімальне значення для аргументів.
ABS()	Повертає модуль числа.
POW(X, Y)	Повертає число <i>X</i> у степіні <i>Y</i> .
RAND()	Повертає псевдовипадкове число з плаваючою крапкою.
NOT IN()	Перевіряє, чи не входить значення до множини вказаних значень.
NULLIF(<i>вираз1</i> , <i>вираз2</i>)	Повертає NULL, якщо значення рівні між собою.
IF(<i>вираз1</i> , <i>вираз2</i> , <i>вираз3</i>)	Повертає <i>вираз2</i> , якщо виконується <i>вираз1</i> . В протилежному випадку повертає <i>вираз3</i> .
SIN(); COS(); TAN(); COT(); LN()	Повертає значення синуса, косинуса, тангенса, котангенса, натурального логарифма.
BETWEEN ... AND ...	Перевіряє, чи входить значення у

	заданий діапазон.
COUNT()	Рахує кількість рядків, які повертає запит.
COUNT(DISTINCT)	Рахує кількість різних значень.
CHAR_LENGTH(<i>рядок</i>)	Повертає кількість символів в аргументі.
CONCAT(<i>рядок1</i> , ...)	Повертає об'єднані рядки.
NOW(), CURRENT_TIMESTAMP()	Повертає поточну дату і час у форматі 'YYYY-MM-DD HH:MM:SS'.
CURRENT_DATE(), CURRENT_DATE	Повертає поточну дату.
CURRENT_TIME(), CURRENT_TIME	Повертає поточний час.
DAYOFMONTH(<i>дата</i>)	Повертає день місяця з отриманої дати.
MONTH(<i>дата</i>)	Повертає номер місяця, присутнього у отриманій даті.
YEAR(<i>дата</i>)	Повертає рік з отриманої дати.
STRCMP(<i>рядок1</i> , <i>рядок2</i>)	Порівнює два рядки.
RLIKE шаблон	Порівняння з шаблоном, яке підтримує регулярні вирази.

Виконання

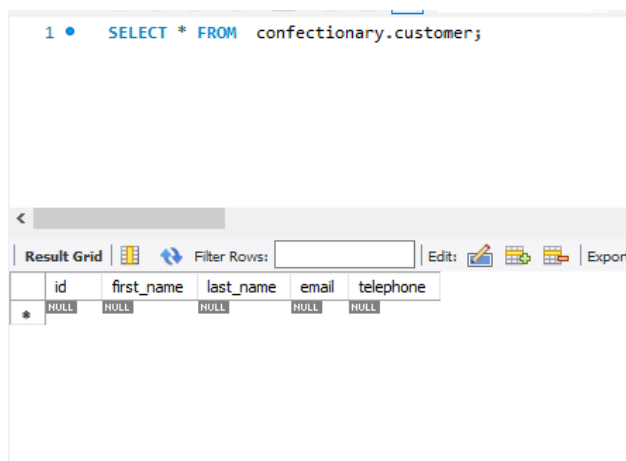
Виконаємо такі запити для занесення даних у таблиці і їх подальшої модифікації :

1. Заповнимо таблицю customer в режимі одиночного і групового доповнення.
2. Створимо файли з даними і заповнимо ними решту таблиць баз даних.
3. Виконаємо модифікацію значень у таблиці dish.
4. Видалимо записи з таблиці order.

1.

- Внесення нових значень в таблицю customer в режимі одиночного доповнення:

Початковий вигляд таблиці :

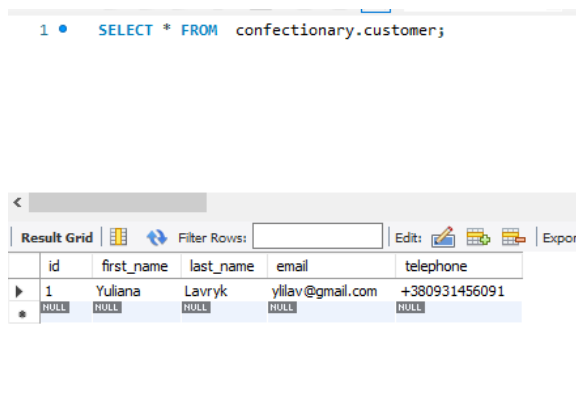


	id	first_name	last_name	email	telephone
*	NULL	NULL	NULL	NULL	NULL

Вводимо команду для одиночного доповнення :

```
1 INSERT INTO confectionary.customer VALUES (1, 'Yuliana',  
2 'Lavryk', 'ylilav@gmail.com', '+380931456091');
```

*За допомогою запиту SELECT * FROM перевіряємо чи заповнились поля :*



	id	first_name	last_name	email	telephone
▶	1	Yuliana	Lavryk	ylilav@gmail.com	+380931456091
*	NULL	NULL	NULL	NULL	NULL

- Внесення нових значень в таблицю customer в режимі групового доповнення:

Режим групового доповнення :

```
1 • INSERT INTO confectionary.customer VALUES (NULL, 'Iryna', 'Dosiak',
2 'iryna123@gmail.com', '+380631123470'),
3 (NULL, 'Olena', 'Kulchytska', 'olena111@gmail.com',
4 '+380931073422'),
5 (NULL, 'Oleksandra', 'Dypko', 'olexandra17@gmail.com',
6 '+380675809127');
```

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

id	first_name	last_name	email	telephone
1	Yuliana	Lavryk	yllav@gmail.com	+380931456091
2	Iryna	Dosiak	iryna123@gmail.com	+380631123470
3	Olena	Kulchytska	olena111@gmail.com	+380931073422
4	Oleksandra	Dypko	olexandra17@gmail.com	+380675809127
5	NULL	NULL	NULL	NULL

2.

Створимо файли з даними і заповнимо ними решту таблиць баз даних :

- Таблиця staff

Файл staff.txt :

```
staff.txt: Блокнот
Файл Редагування Формат Вигляд Довідка
\N Anastasia Spodarenko +380932341780 confectioner(2d category)
\N Viktor Kachmaryk +380735060981 confectioner(3d category)
\N Anna Borysyuk +380673386540 confectioner(2d category)
\N Andriy Bodnar +380934560823 confectioner(3d category)
```

Початковий вигляд таблиці:

```
1 • SELECT * FROM confectionary.staff;
```

id	first_name	last_name	telephone	post
NULL	NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\staff.txt'
2 INTO TABLE confectionary.staff LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

```
1 • SELECT * FROM confectionary.staff;
```

id	first_name	last_name	telephone	post
1	Anastasia	Spodarenko	+380932341780	confectioner(2d category)
2	Viktor	Kachmaryk	+380735060981	confectioner(3d category)
3	Anna	Borysyuk	+380673386540	confectioner(2d category)
4	Andriy	Bodnar	+380934560823	confectioner(3d category)
NULL	NULL	NULL	NULL	NULL

- Таблиця dish

Файл dish.txt :

dish.TXT: Блокнот

Файл	Редагування	Формат	Вигляд	Довідка
\N	Oreo	European	950g	347 450
\N	Brownie	American	1kg	466 340
\N	LemonPie	French	1,5kg	309 420
\N	Strawberry cake "Fraisier"	French	750g	188 325

Початковий вигляд таблиці:

```
1 • SELECT * FROM confectionary.dish;
```

id	name	kitchen_name	weight	kcal	price
NULL	NULL	NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\dish.txt'
2 INTO TABLE confectionary.dish LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

```
1 • SELECT * FROM confectionary.dish;
```

id	name	kitchen_name	weight	kcal	price
6	Oreo	European	950g	347	450.00
7	Brownie	American	1kg	466	340.00
8	LemonPie	French	1,5kg	309	420.00
9	Strawberry cake "Fraisier"	French	750g	188	325.00
NULL	NULL	NULL	NULL	NULL	NULL

- Таблиця Supplier

Файл supplier.txt :

supplier.TXT: Блокнот

Файл	Редагування	Формат	Вигляд	Довідка
\N	Molokiya	(0352)56-12-01	site@molokija.com	dairy products
\N	Hutorok	(0352)56-13-08	info@hutorok.com.ua	floor
\N	Kvochka	(044)393-40-50	kvochka@gmail.com	eggs
\N	Roshen	(044)498-66-89	info@roshentrade.com.ua	chocolate,cacao

Початковий вигляд таблиці:

```
1 • SELECT * FROM confectionary.supplier;
```

Result Grid

id	name	telephone	email	classification
NULL	NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\supplier.txt'
2 INTO TABLE confectionary.supplier LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

```
1 • SELECT * FROM confectionary.supplier;
```

Result Grid

id	name	telephone	classification	email
1	Molokiya	(0352)56-12-01	site@molokija.com	dairy products
2	Hutorok	(0352)56-13-08	info@hutorok.com.ua	floor
3	Kvochka	(044)393-40-50	kvochka@gmail.com	eggs
4	Roshen	(044)498-66-89	info@roshentrade.com.ua	chocolate,cacao
NULL	NULL	NULL	NULL	NULL

• Таблиця Ingredients

Файл ingredient.txt :

ingredient.TXT: Блокнот

Файл	Редагування	Формат	Вигляд	Довідка
\N	Flour	20kg	32.40	2
\N	Milk	10l	29.49	1
\N	Butter	5kg	222.45	1
\N	Eggs	50units	1.74	3

Початковий вигляд таблиці:

```
1 • SELECT * FROM confectionary.ingredient;
```

Result Grid

id	name	quantity	price_for_unit	supplier_id
NULL	NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\ingredient.txt'
2 INTO TABLE confectionary.ingredient LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

```
1 • SELECT * FROM confectionary.ingredient;
```

	id	name	quantity	price_for_unit	supplier_id
▶	1	Flour	20kg	32.40	2
	2	Milk	10l	29.49	1
	3	Butter	5kg	222.45	1
	4	Eggs	50units	1.74	3
*	NULL	NULL	NULL	NULL	NULL

- Таблиця order

Файл order.txt :

Файл	Редагування	Формат	Вигляд	Довідка
\\N	2	1	2020-03-12	18:50:32
\\N	3	2	2020-03-17	13:24:12
\\N	4	2	2020-03-17	20:41:56
\\N	1	4	2020-03-18	10:09:34

Початковий вигляд таблиці:

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

	id	employee_id	customer_id	date	time
*	NULL	NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\order.txt'
2 INTO TABLE confectionary.order LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

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

	id	employee_id	customer_id	date	time
▶	4	2	1	2020-03-12	18:50:32
	5	3	2	2020-03-17	13:24:12
	6	4	2	2020-03-17	20:41:56
	7	1	4	2020-03-18	10:09:34
*	NULL	NULL	NULL	NULL	NULL

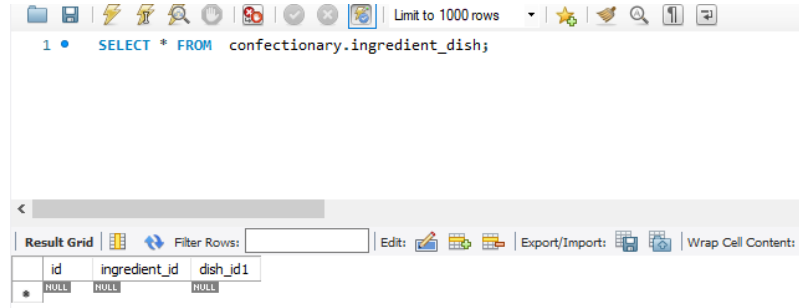
- Таблиця ingredient_dish

Файл *ingredient_dish.txt* :



Файл	Редагування	Формат	Вигляд	Довідка
\N	1	1		
\N	2	1		
\N	4	2		
\N	3	4		

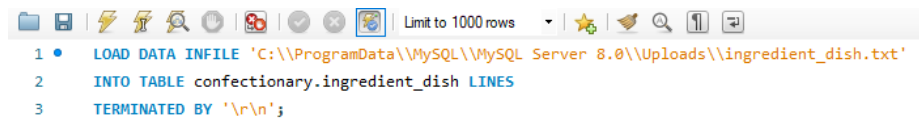
Початковий вигляд таблиці:



```
1 • SELECT * FROM confectionary.ingredient_dish;
```

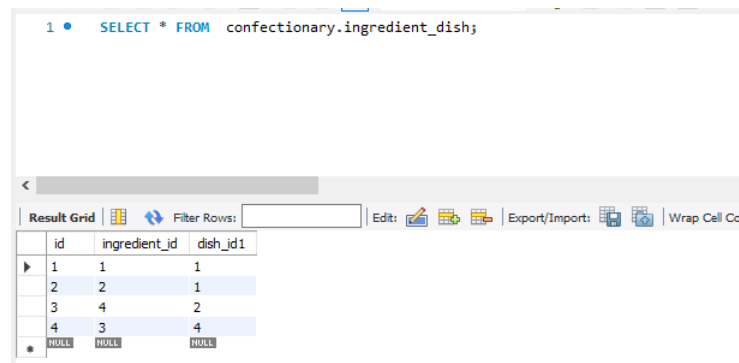
id	ingredient_id	dish_id1
NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :



```
1 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\ingredient_dish.txt'
2 INTO TABLE confectionary.ingredient_dish LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :



```
1 • SELECT * FROM confectionary.ingredient_dish;
```

id	ingredient_id	dish_id1
1	1	1
2	2	1
3	4	2
4	3	4
NULL	NULL	NULL

- Таблиця **dish_order**

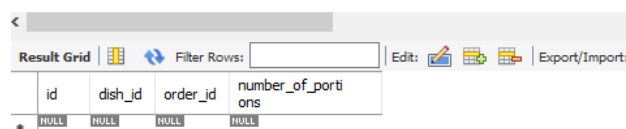
Файл *dish_order.txt* :



Файл	Редагування	Формат	Вигляд	Довідка
\N	2	3	1	
\N	3	1	2	
\N	1	2	1	
\N	4	4	3	

Початковий вигляд таблиці:

```
1 • SELECT * FROM confectionary.dish_order;
```



id	dish_id	order_id	number_of_portions
NULL	NULL	NULL	NULL

Команда для заповнення таблиці даними з файлу :

```
1 • LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\dish_order.txt'
2 INTO TABLE confectionary.dish_order LINES
3 TERMINATED BY '\\r\\n';
```

Перевіряємо :

```
1 • SELECT * FROM confectionary.dish_order;
```

	id	dish_id	order_id	number_of_portions
1	1	2	3	1
2	2	3	1	2
3	3	1	2	1
4	4	4	4	3
*	NULL	NULL	NULL	NULL

3.

Виконаємо модифікацію значень у таблиці dish :

Змінимо ціну страв, *id* якої рівне 1 і збільшимо значення ключового поля на одиницю. Для зміни значень ключа потрібно змінити порядок сортування рядків, щоб уникнути суперечності.

```
1 • UPDATE confectionary.dish SET price = '500' WHERE id=1;
```

```
1 • UPDATE confectionary.dish SET id = id + 1 ORDER BY id DESC
```

Таблиця до виконання запитів :

	id	name	kitchen_name	weight	kcal	price
1	1	Oreo	European	950g	347	450.00
2	2	Brownie	American	1kg	466	340.00
3	3	LemonPie	French	1,5kg	309	420.00
4	4	Strawberry cake "Fraisier"	French	750g	188	325.00

Таблиця після виконання запитів :

	id	name	kitchen_name	weight	kcal	price
1	2	Oreo	European	950g	347	500.00
2	3	Brownie	American	1kg	466	340.00
3	4	LemonPie	French	1,5kg	309	420.00
4	5	Strawberry cake "Fraisier"	French	750g	188	325.00

4.

Видалимо записи з таблиці order :

Видалимо замовлення для яких виконуються такі умови: зроблені до дати 2020-03-18 і id=2.

```
1 • DELETE FROM confectionary.order WHERE (date<'2020-03-18' && id=2);
```

Таблиця до виконання запиту :

	id	employee_id	customer_id	date	time
▶	1	2	1	2020-03-12	18:50:32
	2	3	2	2020-03-17	13:24:12
	3	4	2	2020-03-17	20:41:56
	4	1	4	2020-03-18	10:09:34

Таблиця після виконання запиту :

	id	employee_id	customer_id	date	time
▶	1	2	1	2020-03-12	18:50:32
	3	4	2	2020-03-17	20:41:56
	4	1	4	2020-03-18	10:09:34

Висновок : під час виконання даної лабораторної роботи я навчилася розробляти SQL-запити для внесення нових значень в таблиці в режимі одиничного та групового доповнення, SQL-запити для внесення змін в рядки таблиць та SQL-запити для вилучення вибраних рядків. Різними способами доповнила даними всі таблиці створеної бази даних та у двох таблицях провела модифікацію.