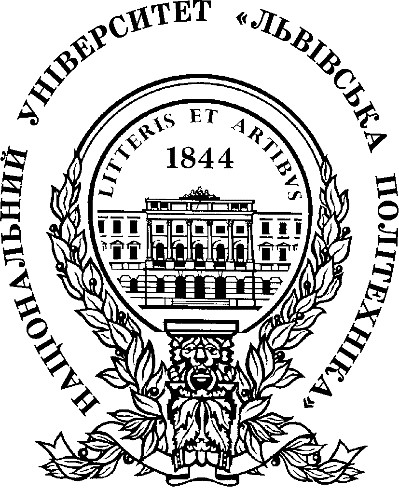
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ № 4 **з курсу “ОБДЗ”** на тему:

**«Запити на додавання, зміну та вилучення даних»**

**Виконав:**

студент групи КН-208

Фіняк М.В.

**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 вилучення 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(*Х*, *У*) | Повертає число *Х* у степіні *У*. |
| RAND() | Повертає псевдовипадкове число з плаваючою крапкою. |
| NOT IN() | Перевіряє, чи не входить значення до множини вказаних значень. |
| NULLIF(*вираз1*, *вираз2*) | Повертає NULL, якщо значення рівні між собою. |
| IF(*вираз1*, *вираз2*, *вираз3*) | Повертає *вираз2*, якщо виконується *вираз1*. В протилежному випадку повертає *вираз3*. |
| 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(*рядок*) | Повертає кількість символів в аргументі. |
| CONCAT(*рядок1*, ...) | Повертає об’єднані рядки. |
| NOW(), | Повертає поточну дату і час у форматі |
| СURRENT\_TIMESTAMP() | 'YYYY-MM-DD HH:MM:SS'. |
| CURRENT\_DATE(),  CURRENT\_DATE | Повертає поточну дату. |
| CURRENT\_TIME(),  CURRENT\_TIME | Повертає поточний час. |
| DAYOFMONTH(*дата*) | Повертає день місяця з отриманої дати. |
| MONTH(*дата*) | Повертає номер місяця, присутнього у отриманій даті. |
| YEAR(*дата*) | Повертає рік з отриманої дати. |
| STRCMP(*рядок1*, *рядок2*) | Порівнює два рядки. |
| RLIKE шаблон | Порівняння з шаблоном, яке підримує 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 одиничного доповнення:

*Початковий вигляд таблиці :*

**SELECT \* FROM restaurant.customer;**



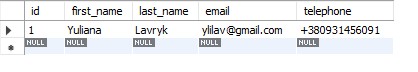
*Вводимо команду для одиничного доповнення :*

**INSERT INTO restaurant.customer VALUES (1, ‘Yuliana’, ‘Lavryk’,**

**‘ylilav@gmail.com’, ‘+380931456091’);**

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

**SELECT \* FROM restaurant.customer;**



* + Внесення нових значень в таблицю customer в режимі групового доповнення:

*Режим групового доповнення :*

**INSERT INTO restaurant.customer VALUES (NULL, 'Iryna', 'Dosiak',**

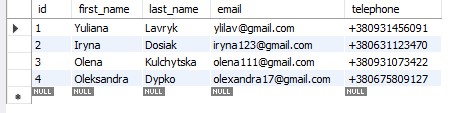
**'iryna123@gmail.com', NULL),**

**(NULL, 'Olena', 'Kulchytska', 'olena111@gmail.com',**

**'+380931073422'),**

**(NULL, 'Oleksandra', 'Dypko', 'olexandra17@gmail.com', '+380675809127');**

**SELECT \* FROM restaurant.customer;**

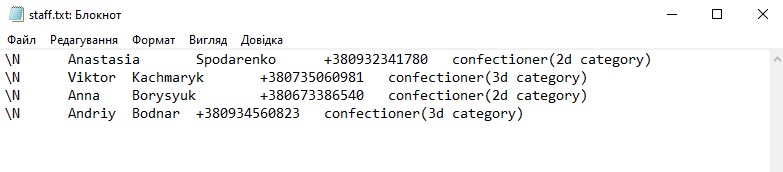


**2.**

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

* + **Таблиця staff**

*Файл staff.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.staff;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

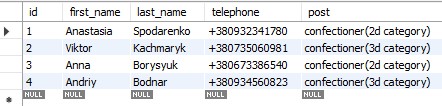
**8.0\\Uploads\\staff.txt'**

**INTO TABLE restaurant.staff LINES**

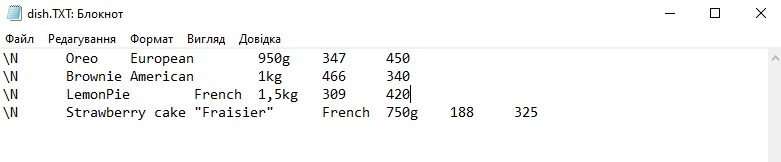
**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.staff;**



* + **Таблиця dish** *Файл dish.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.dish;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

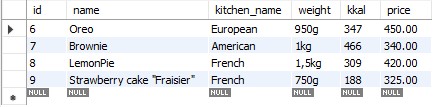
**8.0\\Uploads\\dish.txt'**

**INTO TABLE restaurant.dish LINES**

**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.dish;**



* + **Таблиця Supplier**

*Файл supplier.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.supplier;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

**8.0\\Uploads\\supplier.txt'**

**INTO TABLE restaurant.supplier LINES**

**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.supplier;**



* + **Таблиця Ingredients**

*Файл ingredient.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.ingredient;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

**8.0\\Uploads\\ingredient.txt'**

**INTO TABLE restaurant.ingredient LINES**

**TERMINATED BY '\r\n';**

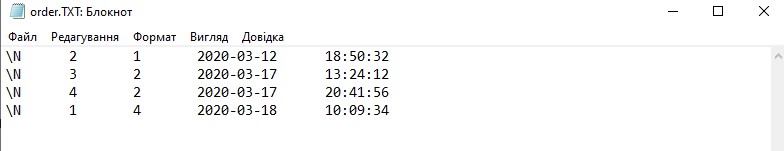
*Перевіряємо :*

**SELECT \* FROM restaurant.ingredient;**



* + **Таблиця order**

*Файл order.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.order;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

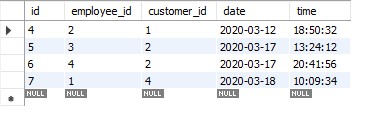
**8.0\\Uploads\\order.txt'**

**INTO TABLE restaurant.order LINES**

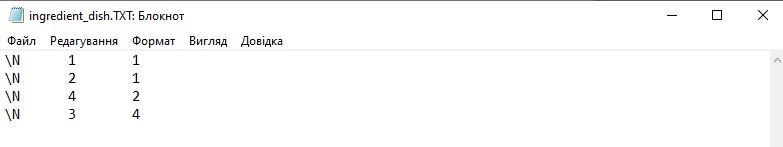
**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.order;**



* + **Таблиця ingredient\_dish** *Файл ingredient\_dish.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.ingredient\_dish;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

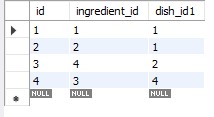
**8.0\\Uploads\\ingredient\_dish.txt'**

**INTO TABLE restaurant.ingredient\_dish LINES**

**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.ingredient\_dish;**



* + **Таблиця dish\_order** *Файл* dish\_order*.txt :*



*Початковий вигляд таблиці:*

**SELECT \* FROM restaurant.dish\_order;**



*Команда для заповнення таблиці даними з файлу :*

**LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server**

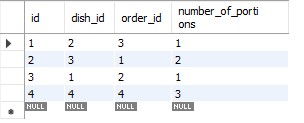
**8.0\\Uploads\\dish\_order.txt'**

**INTO TABLE restaurant.dish\_order LINES**

**TERMINATED BY '\r\n';**

*Перевіряємо :*

**SELECT \* FROM restaurant.dish\_order;**



**3.**

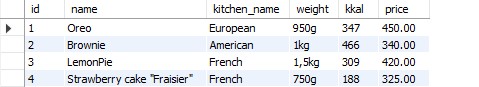
Виконаємо модифікацію значень у таблиці dish :

*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.*

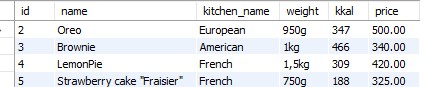
**UPDATE restaurant.dish SET price = ‘500’ WHERE id=1;**

**UPDATE restaurant.dish SET id = id+1 ORDER BY id DESC**

*Таблиця до виконання запитів :*



*Таблиця після виконання запитів :*



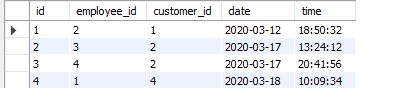
**4.**

Видалимо записи з таблиці order :

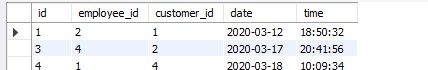
*Видалимо замовлення для яких виконуються такі умови: зроблені до дати 2020-03-18 і id=2.*

**DELETE FROM restaurant.order WHERE (date<’2020-03-18’ && id=2);**

*Таблиця до виконання запиту :*



*Таблиця після виконання запиту :*



**Висновок :** під час виконання даної лабораторної роботи я навчився 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 доповнив даними всі таблиці створеної бази даних та у двох таблицях провів модифікацію.