# MIHICTEPCTBO OCBITИ І НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

### Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



## Звіт до лабораторної роботи №4

з дисципліни "Організація Баз Даних та знань"

Виконала:

ст. гр. КН-210 Заремба Вікторія

Викладач:

Мельникова H.I.

#### Лабораторна робота №4 з курсу "ОБДЗ" на тему:

#### "Запити на додавання, зміну та вилучення даних"

**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.

Для внесення значень в таблиці використовується директива INSERT.

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 (див. далі). 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.

Наступна директива MySQL дозволяє заносити дані в таблицю із зовнішнього файлу.

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]

Для внесення змін в записи таблиці використовується директива UPDATE.

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.

Для видалення записів з таблиці використовується директива DELETE.

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.

При побудові виразів та запитів, можна використовувати оператори та функції MySQL. Опишемо основні з них у таблиці.

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 $X$ у степіні $Y$ .
RAND()	Повертає псевдовипадкове число з плаваючою крапкою.
NOT IN()	Перевіряє, чи не входить значення до множини вказаних значень.
NULLIF(supa31, supa32)	Повертає NULL, якщо значення рівні між собою.
IF(вираз1, вираз2, вираз3)	Повертає вираз 2, якщо виконується вираз 1. В протилежному випадку повертає вираз 3.
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(), 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(∂ama)	Повертає день місяця з отриманої дати.
MONTH(∂ama)	Повертає номер місяця, присутнього у отриманій даті.
YEAR(∂ama)	Повертає рік з отриманої дати.
STRCMP( $pядок1, pядок2$ )	Порівнює два рядки.
LIKE рядок	Порівняння з шаблоном. Можна використовувати % (довільні символи) і _ (довільний символ).
RLIKE шаблон	Порівняння з шаблоном, яке підримує регулярні вирази.
MATCH (поля) AGAINST (рядок)	Здійснює пошук рядка у вказаних символьних полях таблиці. (Тільки для MyISAM-таблиць.)
ROW_COUNT()	Рахує кількість рядків, оновлених останнім запитом.
BENCHMARK(κ-cmь, вираз)	Виконує вираз вказану кількість разів і повертає час виконання. Замість виразу можна вказати запит, який повинен повертати максимум одне значення.
COMPRESS(рядок_символів), UNCOMPRESS(рядок_символів)	Якщо MySQL встановлений з бібліотекою zlib, функції дозволяють стискати і розпаковувати символьні рядки.
AES_ENCRYPT(значення, ключ), AES_DECRYPT(значення, ключ)	128-бітне шифрування і дешифрування рядків за заданим ключем.
MD5(значення)	Обчислює MD5-хеш заданого рядка.

#### Хід роботи

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

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

#### 1. Заповнимо таблицю User в режимі одиночного і групового доповнення.

Внесення нових значень в таблицю Role в режимі одиничного доповнення:

```
INSERT INTO user VALUES (1, 'Zaremba', 'Viktoria', 'vika241020@gmail.com',
'+380985796738', '12345678');
```

Внесення нових значень в таблицю Role в режимі групового доповнення:

#### Таблиця user після виконання запиту:

```
| Maria | Mari
```

#### 2. Створимо файли з даними і заповнимо ними решту таблиць баз даних.

Приклад завантаження даних з файлу:

```
LOAD DATA INFILE 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\task.csv'
INTO TABLE task
CHARACTER SET CP1251
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n'
(name,deadline,id_department)
SET id task=NULL;
```

Для того, щоб в базі даних коректно відображались усі символи використовуємо кодування CP1251

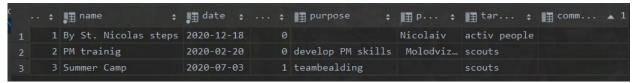
Для автогенерування PR задаємо SET id\_task=NULL, а перед цим вказуємо в дужках, які поля потрібно заповнити з файлу. Не вказані обов'язкові поля прийматимуть дефолтне значення, необов'язкові поля можуть залишатись пустими (якщо не було вказане дефолтнезачення)

#### 3. Виконаємо модифікацію значень у таблиці project.

Змінюємо назви проектів за ID. Для зручності відображення даних сортуємо за датою.

```
UPDATE project SET name = "Курінний табір"
WHERE id_project=2;
UPDATE project SET name = "Вишкыл проектного менеджменту"
WHERE id_project=3;
SELECT * FROM project
ORDER BY date;
```

#### Таблиця до зміни значень:



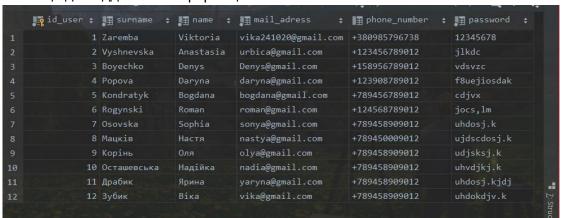
#### Таблиця після зміни значень і сортування

```
. : Приме : Iname : Implace : Impla
```

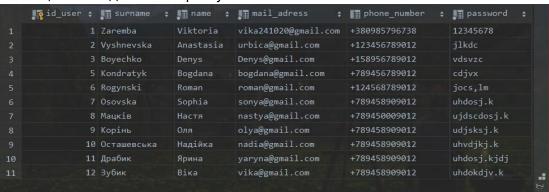
#### 4. Видалимо записи з таблиці user.

```
DELETE FROM user
WHERE surname = 'Popova';
```

#### Таблиця до виддалення інформації:



#### Таблиця після идалення користувача:



**Висновок**: у цій лабораторній роботі було розглянуто способи наповнення і модифікації даних в таблицях БД та проведено модифікацію даних у двох таблицях.