

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА**  
**ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту



**Лабораторна робота 4**  
з організації баз даних та знань

**Виконав:**

Студент групи КН-208

Воробель Адріан

**Викладач:**

Якимишин Х.М.

Львів – 2019р.

**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.

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)

AND, OR

~, &, |, ^

AVG(), MAX(), MIN()

ABS()

POW(X, Y)

RAND()

NOT IN()

NULLIF(вираз1, вираз2)

IF(вираз1, вираз2, вираз3)

SIN(); COS(); TAN(); COT(); LN()

BETWEEN ... AND ...

COUNT()

COUNT(DISTINCT)

CHAR\_LENGTH(рядок)

CONCAT(рядок1, ...)

NOW(), CURRENT\_TIMESTAMP()

CURRENT\_DATE(),

CURRENT\_DATE

CURRENT\_TIME(),

CURRENT\_TIME

DAYOFMONTH(дата)

MONTH(дата)

Опис

Логічне “і” та логічне “або”.

Побітові операції інверсії, логічного “і”, “або”, “виключного або”.

Повертає середнє, максимальнє, або мінімальнє значення для аргументів.

Повертає модуль числа.

Повертає число X у степіні Y.

Повертає псевдовипадковє число з плаваючою крапкою.

Перевіряє, чи не входить значення до множини вказаних значень.

Повертає NULL, якщо значення рівні між собою.

Повертає вираз2, якщо виконується вираз1. В протилежному випадку повертає вираз3.

Повертає значення синуса, косинуса, тангенса, котангенса, натурального логарифма.

Перевіряє, чи входить значення у заданий діапазон.

Рахує кількість рядків, які повертає запит.

Рахує кількість різних значень.

Повертає кількість символів в аргументі.

Повертає об’єднані рядки.

Повертає поточну дату і час у форматі

'YYYY-MM-DD HH:MM:SS'.

Повертає поточну дату.

Повертає поточний час.

Повертає день місяця з отриманої дати.

Повертає номер місяця, присутнього у отриманій даті.

YEAR(дата)	Повертає рік з отриманої дати.
STRCMP(рядок1, рядок2)	Порівнює два рядки.
LIKE рядок	Порівняння з шаблоном. Можна використовувати % (довільні символи) і _ (довільний символ).
RLIKE шаблон	Порівняння з шаблоном, яке підтримує регулярні вирази.
MATCH (поля) AGAINST (рядок)	Здійснює пошук рядка у вказаних символічних полях таблиці. (Тільки для MyISAM-таблиць.)
ROW_COUNT()	Рахує кількість рядків, оновлених останнім запитом.
BENCHMARK(к-сть, 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 символічні 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.

- Внесемо дані в таблицю в режимі одиничного та групового доповнення

### #одиничне доповнення

```
INSERT INTO users VALUES (1,'admin', 'admin');
```

### #групове доповнення

```
INSERT INTO users VALUES (2,'Alexander1983', '12345678'),
                          (3,'TheBestCustomer', 'best123'),
                          (4,'UserBuyCar', 'iwantBMW'),
                          (5,'MercedesLover', 'AmgIsLife');
```

Так виглядає таблиця після доповнень:

	id	login	pass
▶	1	admin	admin
	2	Alexander1983	12345678
	3	TheBestCustomer	best123
	4	UserBuyCar	iwantBMW
	5	MercedesLover	AmgIsLife
*	NULL	NULL	NULL

- Загрузимо дані з файлу

```
LOAD DATA LOCAL INFILE 'C:\ProgramData\MySQL\MySQL Server 5.7\Uploads\data.csv' INTO TABLE Users FIELDS TERMINATED BY ',';
```

Документ з даними формату .csv

	A	B	
1	user1	pass1	
2	user2	pass2	
3	user3	pass3	
4	user4	pass4	
5			

Результат

	id	login	pass
	5	MercedesLover	AmgIsLife
►	6	user1	pass1
	7	user2	pass2
	8	user3	pass3
	9	user4	pass4

3. Змінимо декілька рядків в таблиці

```
UPDATE users SET pass='admin12345' WHERE login='admin';
```

```
UPDATE users SET login='User' WHERE login='UserBuyCar';
```

```
UPDATE users SET pass='pass1234' WHERE id>3;
```

Вигляд таблиці після змін:

	id	login	pass
►	1	admin	admin12345
	2	Alexander1983	12345678
	3	TheBestCustomer	best123
	4	User	pass1234
	5	MercedesLover	pass1234
✱	NULL	NULL	NULL

4. Вилучимо деякі рядки з таблиці

```
DELETE FROM users WHERE pass = 'pass1234';
```

Таблиця:

	id	login	pass
►	1	admin	admin12345
	2	Alexander1983	12345678
	3	TheBestCustomer	best123
✱	NULL	NULL	NULL

## Висновок

В ході лабораторної роботи я навчився працювати з даними таблиці, а саме додавати дані як одиничним записом, так і групою, змінювати дані в таблиці, а також видаляти дані.