

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



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №4

Підготувала:

Студентка групи КН-209

Кульчицька Олена

Викладач:

Мельникова Н.І.

Лабораторна робота №4

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: $\text{поле2} = \text{поле1} * 0.5$.

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(<i>X</i> , <i>Y</i>)	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.
LIKE <i>рядок</i>	Порівняння з шаблоном. Можна використовувати % (довільні символи) і _ (довільний символ).
RLIKE <i>шаблон</i>	Порівняння з шаблоном, яке підтримує регулярні вирази.
MATCH (<i>поля</i>) AGAINST (<i>рядок</i>)	Здійснює пошук рядка у вказаних символьних полях таблиці. (Тільки для MyISAM-таблиць.)
ROW_COUNT()	Рахує кількість рядків, оновлених останнім запитом.
BENCHMARK(<i>к-сть</i> , <i>вираз</i>)	Виконує вираз вказану кількість разів і повертає час виконання. Замість виразу можна вказати запит, який повинен повертати максимум одне значення.
COMPRESS(<i>рядок_символів</i>), UNCOMPRESS(<i>рядок_символів</i>)	Якщо MySQL встановлений з бібліотекою <i>zlib</i> , функції дозволяють стискати і розпаковувати символьні рядки.
AES_ENCRYPT(<i>значення</i> , <i>ключ</i>), AES_DECRYPT(<i>значення</i> , <i>ключ</i>)	128-бітне шифрування і дешифрування рядків за заданим ключем.

MD5(<i>значення</i>)	Обчислює MD5-хеш заданого рядка.
------------------------	----------------------------------

Хід роботи.

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

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

1. Внесення нових значень в таблицю *dish* в режимі одиночного доповнення:

```
USE CONFECTIONARY;  
INSERT INTO dish VALUES (1,'tiramisu', 200, 150, 460);  
SELECT * FROM CONFECTIONARY.DISH
```

	id	name	weight	price	kcal
	1	tiramisu	200	150.00	460
	NULL	NULL	NULL	NULL	NULL

Внесення нових значень в таблицю *dish* в режимі групового доповнення:

```
INSERT INTO dish  
VALUES (2,'cheesecake', 500, 200, 1100),  
       (3,'carrot cake', 500, 170, 900),  
       (4,'apple pie', 600, 150, 920);  
SELECT * FROM CONFECTIONARY.DISH
```

id	name	weight	price	kcal
1	tiramisu	200	150.00	460
2	cheesecake	500	200.00	1100
3	carrot cake	500	170.00	900
4	apple pie	600	150.00	920
NULL	NULL	NULL	NULL	NULL

2. Створимо файли з даними для таблиць *provider*, *staff*, *ingredient* *customer*. Значення полів повинні розділятися символом табуляції, а кожен рядок таблиці повинен починатися з нового рядка у файлі. Виконаємо дві наступні команди.

```
LOAD DATA INFILE 'D:\provider.tbl' INTO TABLE provider LINES TERMINATED BY '\r\n';  
LOAD DATA LOCAL INFILE 'D:\staff.txt' INTO TABLE staff LINES TERMINATED BY '\r\n';  
LOAD DATA INFILE 'D:\ingredient.tbl' INTO TABLE ingredient LINES TERMINATED BY '\r\n';  
LOAD DATA INFILE 'D:\customer.tbl' INTO TABLE customer LINES TERMINATED BY '\r\n';
```

Таблиця *provider*:

	id	name	telephone	classification	email
	1	Astra	2346578	eggs	astra@gmail.com
	2	ProZorro	2345668	flour	prozorro@gmail.com
	NULL	NULL	NULL	NULL	NULL

Таблиця *staff*:

	id	first_name	second_name	telephone	position
	1				
	2				
	3	Alex	Grav	2335673	waiter
	4	Sasha	Moon	4889763	confectioner
		NULL	NULL	NULL	NULL

Таблиця *ingredient*:

	id	name	price_for_unit	quantity	provider_id
	1	flour	89.90	10	2
	2	egg	1.50	50	1
		NULL	NULL	NULL	NULL

Таблиця *customer*:

	id	first_name	second_name	telephone	email
	1	Olena	Link	8564735	olink@gmail.com
	2	Lina	Too	6532458	linavoo@gmail.com
		NULL	NULL	NULL	NULL

- Внесення змін в записи таблиці. Змінимо пароль користувача user1 і збільшимо значення ключового поля на одиницю. Для зміни значень ключа потрібно змінити порядок сортування рядків, щоб уникнути суперечності.

```

• UPDATE staff SET telephone = 8888888
  WHERE id = 3;
• UPDATE staff SET id = id + 1
  ORDER BY id DESC;

```

Таблиця *staff* до виконання запитів:

	id	first_name	second_name	telephone	position
	1				
	2				
	3	Alex	Grav	2335673	waiter
	4	Sasha	Moon	4889763	confectioner
		NULL	NULL	NULL	NULL

Таблиця *staff* після виконання запитів:

	id	first_name	second_name	telephone	position
	2				
	3				
	4	Alex	Grav	8888888	waiter
	5	Sasha	Moon	4889763	confectioner
		NULL	NULL	NULL	NULL

- Запит для видалення старих повідомлень користувачів з таблиці *staff*.

```

• DELETE FROM staff
  WHERE position = '';

```

Таблиця *staff* до виконання запитів:

	id	first_name	second_name	telephone	position
	2				
	3				
	4	Alex	Grav	8888888	waiter
	5	Sasha	Moon	4889763	confectioner
		NULL	NULL	NULL	NULL

Таблиця *staff* після виконання запитів:

	id	first_name	second_name	telephone	position
	2	Alex	Grav	8888888	waiter
	3	Sasha	Moon	4889763	confectioner
	NULL	NULL	NULL	NULL	NULL

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