

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №4

з дисципліни
“ОБДЗ”

Виконав:
ст. гр. КН-211
Ільків Андрій

Викладач:
Якимишин Х.М.

Львів – 2020

“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, максимальнє, або мінималнє 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(*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

котангенс, значення натурального синуса, логарифма.

косинуса, тангенс,

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(*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.

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 символьних 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 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 Car в режимі одиночного і групового доповнення.
2. Створимо файли з даними і заповнимо ними решту таблиць баз даних.
3. Виконаємо модифікацію значень у таблиці.
4. Видалимо записи з таблиці.

1. Внесення нових значень в таблицю Role в режимі одиничного доповнення:

```
insert into car values (1, 'Tesla Model S', 'B'), ( 2, 'Kamaz', 'C'), (
3, 'Harley Davidson', 'A'), (4,'Lada Sedan', 'B');
```

2. Створимо файли з даними для таблиць teacher і Message. Значення полів повинні розділятися символом табуляції, а кожен рядок таблиці повинен починатися з нового рядка у файлі. Виконаємо дві наступні команди.

```
LOAD DATA INFILE '/Users/andrew/db/teacher.tbl' INTO TABLE teacher LINES
TERMINATED BY '\r\n';
```

```
LOAD DATA INFILE '/Users/andrew/db/teacher.tbl' INTO TABLE teacher
LINES TERMINATED BY '\r\n';
```

3. 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 teacher SET password  
= 'user1pass' WHERE  
email='g.santa@mail.com';
```

```
UPDATE teacher SET  
teacher_id = teacher_id + 1  
ORDER BY teacher_id DESC;
```

4. Запит для вилучення практичних занять, на які виділено менше 10 годин з таблиці category.

```
DELETE FROM category  
WHERE (practice_time<10);
```

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