

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

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

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

```
INSERT INTO role VALUES (1, 'Admin', 'full');
```

Внесення нових значень в таблицю Role в режимі групового доповнення:

```
INSERT INTO role
VALUES (NULL, 'Group1', 'read, write, changeown'),
      (NULL, 'Group2', 'read, write, changeown'),
      (NULL, 'Guest', 'read');
```

Таблиця ROLE після виконання запиту:

```
mysql> select * from role;
```

roleID	rolename	permissions
1	Admin	full
2	Group1	read, write, changeown
3	Group2	read, write, changeown
4	Guest	read

```
4 rows in set (0.00 sec)

mysql> _
```

2. Створимо файли з даними для таблиць Author 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 'f:\author.tbl' INTO TABLE author LINES
TERMINATED BY '\r\n';
```

```
LOAD DATA INFILE 'f:\message.tbl' INTO TABLE message 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 суперечностей.

```
UPDATE author SET password = 'user1pass'
WHERE login='user1';
```

```
UPDATE author SET authorid = authorid + 1
ORDER BY authorid DESC;
```

Таблиця Author до виконання запитів:

```
mysql> select * from author;
```

authorID	name	login	password	created	email	profile	roleID
1	NULL	admin	adminpass	2009-01-01 00:00:00	admin@gmail.com	NULL	1
2	NULL	user1	pass	2009-01-02 00:00:00	user1@gmail.com	NULL	2
3	NULL	user2	user2pass	2009-01-03 00:00:00	user2@gmail.com	NULL	2
4	NULL	user3	user3pass	2009-02-12 00:00:00	user3@gmail.com	NULL	3
5	NULL	guest1	guest1pass	2009-02-16 00:00:00	guest1@gmail.com	NULL	4

5 rows in set (0.00 sec)

```
mysql> _
```

Таблиця Author після виконання запитів:

```
mysql> select * from author;
```

authorID	name	login	password	created	email	profile	roleID
2	NULL	admin	adminpass	2009-01-01 00:00:00	admin@gmail.com	NULL	1
3	NULL	user1	user1pass	2009-01-02 00:00:00	user1@gmail.com	NULL	2
4	NULL	user2	user2pass	2009-01-03 00:00:00	user2@gmail.com	NULL	2
5	NULL	user3	user3pass	2009-02-12 00:00:00	user3@gmail.com	NULL	3
6	NULL	guest1	guest1pass	2009-02-16 00:00:00	guest1@gmail.com	NULL	4

5 rows in set (0.00 sec)

```
mysql> _
```

4. Запит для вилучення старих повідомлень користувачів з таблиці Message.

```
DELETE FROM message
WHERE (posted<'2009-02-10' && authorid!=2);
```

Таблиця MESSAGE до видалення інформації:

```
mysql> select * from message;
```

messageID	versionID	message	authorID	posted	status	rating	attachment
1	1	Message from admin #1	2	2009-01-02 00:00:00	published	NULL	NULL
2	1	Message from user #1	3	2009-01-22 00:00:00	published	NULL	NULL
3	1	Message #3	4	2009-02-09 00:00:00	published	NULL	NULL
4	1	Message from admin #2	2	2009-02-12 00:00:00	published	NULL	NULL
5	1	Message from user #2	3	2009-02-22 00:00:00	published	NULL	NULL

5 rows in set (0.00 sec)

```
mysql> _
```

Таблиця MESSAGE після виконання запиту:

```
mysql> select * from message;
```

messageID	versionID	message	authorID	posted	status	rating	attachment
1	1	Message from admin #1	2	2009-01-02 00:00:00	published	NULL	NULL
4	1	Message from admin #2	2	2009-02-12 00:00:00	published	NULL	NULL
5	1	Message from user #2	3	2009-02-22 00:00:00	published	NULL	NULL

3 rows in set (0.00 sec)

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