

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт до практичної роботи №4
з дисципліни “ ОБДЗ ”
На тему: «Запити на додавання, зміну та вилучення даних»

Виконав:
ст. гр. КН-211
Качмарик Віктор
Викладач:
Якимишин Х. М.

Лабораторна робота №4

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]

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.

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

1. Заповнимо таблицю customer в режимі одиночного і групового доповнення.

```
INSERT INTO customer VALUES (1, 'Viktor', 'Kachmaryk', '0639660485', NULL, NULL, NULL, NULL, NULL, 'vity_', 'VityaPass');
```

```
mysql> INSERT INTO customer VALUES (1, 'Viktor', 'Kachmaryk', '0639660485', NULL, NULL, NULL, NULL, NULL, 'vity_', 'VityaPass');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from customer;
```

customer_id	first_name	last_name	phone	email	street	city	state	zip_code	login	password
1	Viktor	Kachmaryk	0639660485	NULL	NULL	NULL	NULL	NULL	vity_	VityaPass

```
1 row in set (0.00 sec)
```

2. Створимо файли PharmacyTable, OrderTable, AllMedicineTable, MedicineTable, OrderItemTable з даними і заповнимо ними решту таблиць баз даних.

```
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\PharmacyTable.csv"
INTO TABLE pharmacy
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\r\\n';
```

```
mysql> select * from pharmacy;
```

pharmacy_id	pharmacy_name	phone	email	street	city	state
2	Phar2	663339999	phar2@gmail.com	Zelena	Lviv	Lvivska
3	Phar3	663334444	phar3@gmail.com	Zelena	Lviv	Lvivska
4	Phar4	663335555	phar4@gmail.com	Zelena	Lviv	Lvivska

```
3 rows in set (0.00 sec)
```

```
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\OrderTable.csv"
INTO TABLE `order`
```

```
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\r\n';
```

```
mysql> select * from `order`;
```

order_id	customer_id	pharmacy_id	order_date	order_status
1	1	1	NULL	0x
2	2	3	NULL	0x
3	3	2	NULL	0x

```
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\AllMedicineTable.csv"
```

```
INTO TABLE all_medicine
```

```
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\r\n';
```

```
mysql> select * from all_medicine;
```

medicine_name	medicine_description	contraindication	appliance
Card	Drops	NULL	NULL
Mezym	Tablets	NULL	NULL
Pentalgin	Tablets	NULL	NULL
Ramtex	Tablets	NULL	NULL
Sorbex	Drops	NULL	NULL

```
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\MedicineTable.csv"
```

```
INTO TABLE medicine
```

```
FIELDS TERMINATED BY ','
```

```
LINES TERMINATED BY '\r\n';
```

```
mysql> select * from medicine;
```

medicine_id	pharmacy_id	medicine_name	quantity	price
1	1	Pentalgin	10	100
2	2	Pentalgin	5	95
3	1	Mezym	7	55
4	3	Mezym	8	31
5	2	Sorbex	6	122
6	3	Sorbex	10	150
7	1	Card	9	123
8	2	Ramtex	7	220

```
LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\OrderItemTable.csv"
INTO TABLE order_item
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\\r\\n';
```

```
mysql> select * from order_item;
```

item_id	medicine_id	quantity	discount	order_id
1	1	3	NULL	1
2	2	2	NULL	1
3	3	1	NULL	1
4	4	1	NULL	2
5	5	2	NULL	2
6	6	1	NULL	3
7	7	2	NULL	3

3. Виконаємо модифікацію значень у таблиці customer.

```
UPDATE customer SET password = 'NewAndrew' WHERE login= 'andrew_';
```

```
mysql> select * from customer;
```

customer_id	first_name	last_name	phone	email	street	city	state	zip_code	login	password
1	Viktor	Kachmaryk	0639660485	NULL	NULL	NULL	NULL	NULL	vitya_	VityaPass
2	Adrew	Ilkiv	0635550000	NULL	NULL	NULL	NULL	NULL	andrew_	AndrewPass
3	Vlad	Kachmaryk	0638881111	NULL	NULL	NULL	NULL	NULL	vlad_	VladPass

```
mysql> select * from customer;
```

customer_id	first_name	last_name	phone	email	street	city	state	zip_code	login	password
1	Viktor	Kachmaryk	0639660485	NULL	NULL	NULL	NULL	NULL	vitya_	VityaPass
2	Adrew	Ilkiv	0635550000	NULL	NULL	NULL	NULL	NULL	andrew_	NewAndrew
3	Vlad	Kachmaryk	0638881111	NULL	NULL	NULL	NULL	NULL	vlad_	VladPass

4. Видалимо записи з таблиці pharmacy.

```
mysql> select * from pharmacy;
```

pharmacy_id	pharmacy_name	phone	email	street	city	state
1	Phar1	0669993333	phar1@gmail.com	Lypneva	Lviv	Lvivska
2	Phar2	663339999	phar2@gmail.com	Zelena	Lviv	Lvivska
3	Phar3	663334444	phar3@gmail.com	Zelena	Lviv	Lvivska
4	Phar4	663335555	phar4@gmail.com	Zelena	Lviv	Lvivska
5	Phar2	0663339999	phar2@gmail.com	Lyubinska	Lviv	Lvivska
6	Phar2	0663339999	phar2@gmail.com	Lypneva	Lviv	Lvivska

```
mysql> select * from pharmacy;
```

pharmacy_id	pharmacy_name	phone	email	street	city	state
2	Phar2	663339999	phar2@gmail.com	Zelena	Lviv	Lvivska
3	Phar3	663334444	phar3@gmail.com	Zelena	Lviv	Lvivska
4	Phar4	663335555	phar4@gmail.com	Zelena	Lviv	Lvivska

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