

Лабораторна робота 2

МАНІПУЛЮВАННЯ ДАНИМИ ЗАСОБАМИ МОВИ SQL: ДОДАВАННЯ, ООНОВЛЕННЯ ТА ВИДАЛЕННЯ ДАНИХ

Мета роботи: навчитися використовувати оператори мови SQL, призначені для додавання, оновлення та видалення даних, на прикладі СУБД MySQL.

Хід роботи

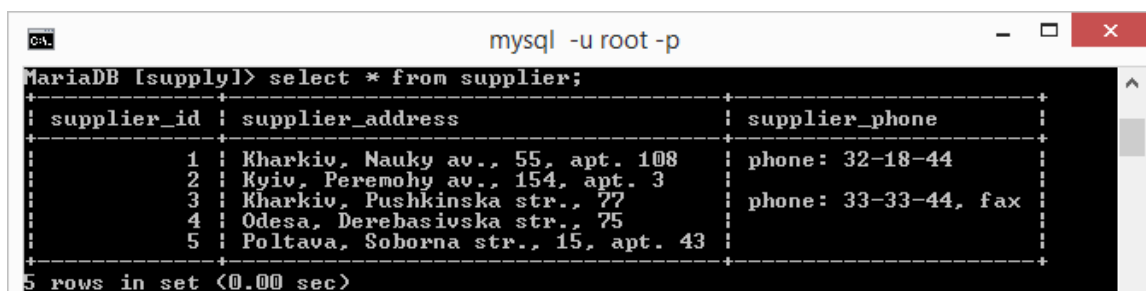
1. Додавання даних до створеної бази даних

Для додавання даних використовується оператор INSERT.

Наступні команди дозволяють заповнити дані про постачальників у створеній базі даних:

```
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (1, 'Kharkiv, Nauky av., 55, apt. 108', 'phone: 32-18-44');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (2, 'Kyiv, Peremohy av., 154, apt. 3', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (3, 'Kharkiv, Pushkinska str., 77', 'phone: 33-33-44, fax: 22-12-33');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (4, 'Odesa, Derebasivska str., 75', '');
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
VALUES (5, 'Poltava, Soborna str., 15, apt. 43', '');
```

Перевірити записи, створені у таблиці supplier (рисунок 2.1).



The screenshot shows a terminal window titled 'mysql -u root -p'. The prompt is 'MariaDB [supply]>'. The command entered is 'select * from supplier;'. The output is a table with 5 rows and 3 columns: 'supplier_id', 'supplier_address', and 'supplier_phone'. The data is as follows:

supplier_id	supplier_address	supplier_phone
1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44
2	Kyiv, Peremohy av., 154, apt. 3	
3	Kharkiv, Pushkinska str., 77	phone: 33-33-44, fax
4	Odesa, Derebasivska str., 75	
5	Poltava, Soborna str., 15, apt. 43	

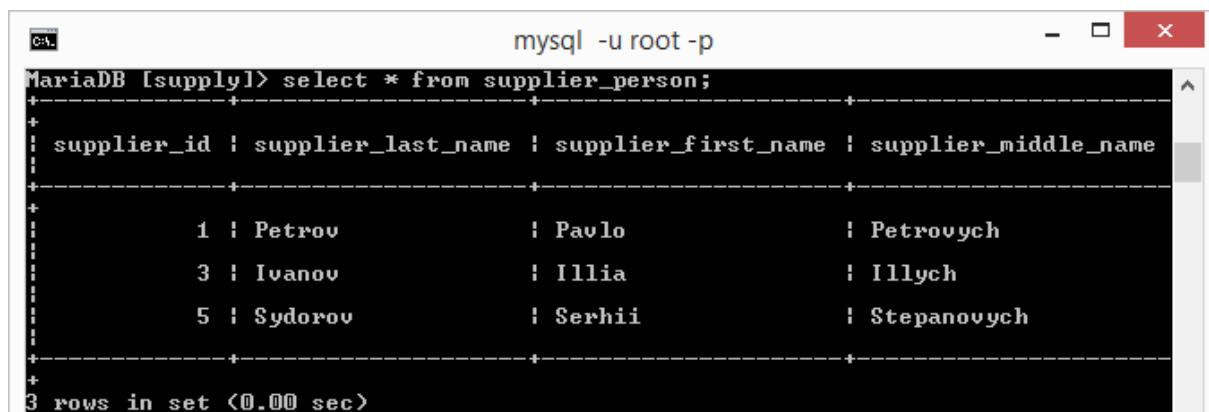
At the bottom of the terminal output, it says '5 rows in set (0.00 sec)'.

Рисунок 2.1

Наступні команди дозволяють заповнити дані про постачальників-фізичних осіб у створеній базі даних:

```
INSERT INTO supplier_person (supplier_id, supplier_last_name,  
supplier_first_name, supplier_middle_name) VALUES (1, 'Petrov', 'Pavlo',  
'Petrovych');  
INSERT INTO supplier_person (supplier_id, supplier_last_name,  
supplier_first_name, supplier_middle_name) VALUES (3, 'Ivanov', 'Illia',  
'Illych');  
INSERT INTO supplier_person (supplier_id, supplier_last_name,  
supplier_first_name, supplier_middle_name) VALUES (5, 'Sydorov', 'Serhii',  
'Stepanovych');
```

Перевірити записи, створені у таблиці `supplier_person` (рисунок 2.2).



The screenshot shows a terminal window titled 'mysql -u root -p'. The prompt is 'MariaDB [supply]>'. The command entered is 'select * from supplier_person;'. The output is a table with 4 columns: 'supplier_id', 'supplier_last_name', 'supplier_first_name', and 'supplier_middle_name'. There are 3 rows of data. At the bottom, it says '3 rows in set (0.00 sec)'.

supplier_id	supplier_last_name	supplier_first_name	supplier_middle_name
1	Petrov	Pavlo	Petrovych
3	Ivanov	Illia	Illych
5	Sydorov	Serhii	Stepanovych

Рисунок 2.2

Наступні команди дозволяють заповнити дані про постачальників-юридичних осіб у створеній базі даних:

```
INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (2,  
'Interfruit Ltd.');
```

```
INSERT INTO supplier_org (supplier_id, supplier_org_name) VALUES (4,  
'Transservice LLC');
```

Перевірити записи, створені у таблиці `supplier_org` (рисунок 2.3).

```
mysql -u root -p
MariaDB [supply]> select * from supplier_org;
+-----+-----+
| supplier_id | supplier_org_name |
+-----+-----+
| 2 | Interfruit Ltd. |
| 4 | Transservice LLC |
+-----+-----+
2 rows in set <0.00 sec>
```

Рисунок 2.3

Наступні команди дозволяють заповнити дані про укладені договори у створеній базі даних:

```
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES
('2018-09-01', 1, 'Order 34 on 30.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES
('2018-09-10', 1, 'Invoice 08-78 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES
('2018-09-23', 3, 'Order 56 on 28.08.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES
('2018-09-24', 2, 'Order 74 on 11.09.2018');
INSERT INTO contract (contract_date, supplier_id, contract_note) VALUES
('2018-10-02', 2, 'Invoice 09-12 on 21.09.2018');
```

Перевірити записи, створені у таблиці contract (рисунок 2.4).

```
mysql -u root -p
MariaDB [supply]> select * from contract;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | contract_note |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Order 34 on 30.08.2018 |
| 2 | 2018-09-10 00:00:00 | 1 | Invoice 08-78 on 28.08.2018 |
| 3 | 2018-09-23 00:00:00 | 3 | Order 56 on 28.08.2018 |
| 4 | 2018-09-24 00:00:00 | 2 | Order 74 on 11.09.2018 |
| 5 | 2018-10-02 00:00:00 | 2 | Invoice 09-12 on 21.09.2018 |
+-----+-----+-----+-----+
5 rows in set <0.00 sec>
```

Рисунок 2.4

Наступні команди дозволяють заповнити дані про поставлені товари у створеній базі даних:

```
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (1, 'TV', 10, 1300);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (1, 'Audio Player', 25, 700);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (1, 'Video Player', 12, 750);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (2, 'Stereo System', 11, 500);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (2, 'Audio Player', 5, 450);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (2, 'Video Player', 8, 450);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (3, 'TV', 52, 900);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (3, 'Audio Player', 11, 550);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (3, 'Monitor', 85, 550);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (4, 'TV', 56, 990);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (4, 'Audio Player', 22, 320);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (4, 'Printer', 41, 350);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (5, 'TV', 14, 860);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (5, 'Audio Player', 33, 580);
INSERT INTO supplied (contract_number, supplied_product, supplied_amount,
supplied_cost) VALUES (5, 'Video Player', 17, 850);
```

Перевірити записи, створені у таблиці supplied (рисунок 2.5).

```

mysql -u root -p
MariaDB [supply]> select * from supplied;
+-----+-----+-----+-----+
| contract_number | supplied_product | supplied_amount | supplied_cost |
+-----+-----+-----+-----+
| 1 | Audio Player | 25 | 700.00 |
| 1 | TV | 10 | 1300.00 |
| 1 | Video Player | 12 | 750.00 |
| 2 | Audio Player | 5 | 450.00 |
| 2 | Stereo System | 11 | 500.00 |
| 2 | Video Player | 8 | 450.00 |
| 3 | Audio Player | 11 | 550.00 |
| 3 | Monitor | 85 | 550.00 |
| 3 | TV | 52 | 900.00 |
| 4 | Audio Player | 22 | 320.00 |
| 4 | Printer | 41 | 350.00 |
| 4 | TV | 56 | 990.00 |
| 5 | Audio Player | 33 | 580.00 |
| 5 | TV | 14 | 860.00 |
| 5 | Video Player | 17 | 850.00 |
+-----+-----+-----+-----+
15 rows in set (0.00 sec)

```

Рисунок 2.5

2. Оновлення даних у базі даних

Оновлення даних (зміна значень полів у існуючих записах) у БД виконується за допомогою оператора UPDATE.

Наприклад, якщо необхідно зменшити вартість принтеру який був поставлений за договором з номером 4, на 5%, команда буде наступною (рисунок 2.6):

```

UPDATE supplied
SET supplied_cost = supplied_cost * 0.95
WHERE contract_number = 4 AND supplied_product = 'Printer';

```

```

mysql -u root -p
MariaDB [supply]> update supplied set supplied_cost = supplied_cost * 0.95 where
contract_number = 4 AND supplied_product = 'Printer';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [supply]> select * from supplied where contract_number = 4;
+-----+-----+-----+-----+
| contract_number | supplied_product | supplied_amount | supplied_cost |
+-----+-----+-----+-----+
| 4 | Audio Player | 22 | 320.00 |
| 4 | Printer | 41 | 332.50 |
| 4 | TV | 56 | 990.00 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

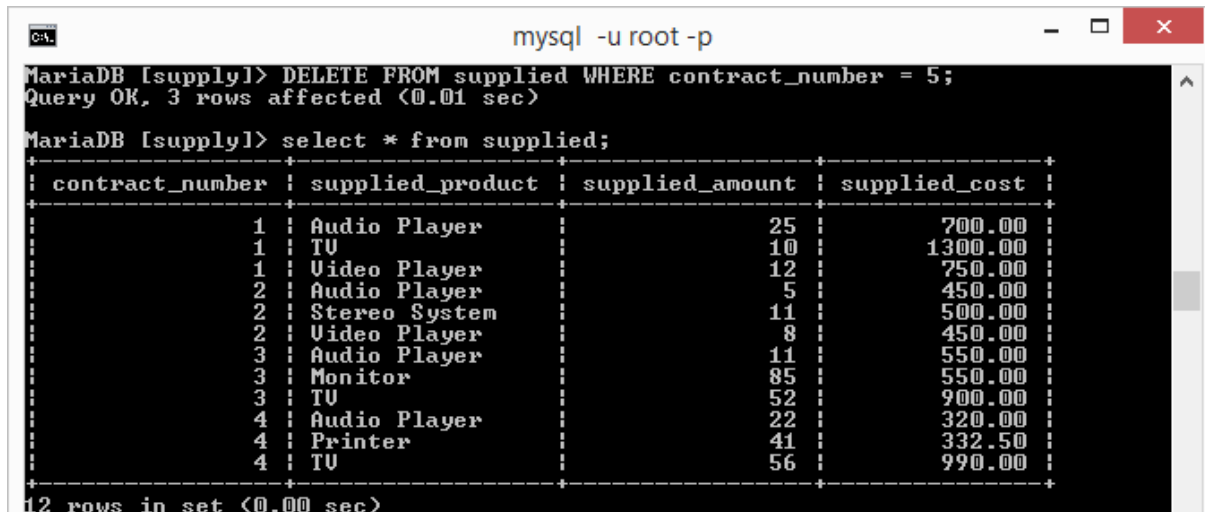
Рисунок 2.6

3. Видалення даних з бази даних

Для видалення даних з таблиць бази даних застосовується оператор DELETE.

Наприклад, для видалення поставлених товарів, які відповідають договору з номером 5, необхідно виконати наступну команду (Рисунок 2.7):

```
DELETE FROM supplied WHERE contract_number = 5;
```



The screenshot shows a MySQL terminal window titled 'mysql -u root -p'. The user is in the 'supply' database. The first command executed is 'DELETE FROM supplied WHERE contract_number = 5;', which returns 'Query OK, 3 rows affected (0.01 sec)'. The second command is 'select * from supplied;', which returns a table with 12 rows. The table has four columns: 'contract_number', 'supplied_product', 'supplied_amount', and 'supplied_cost'. The data is as follows:

contract_number	supplied_product	supplied_amount	supplied_cost
1	Audio Player	25	700.00
1	TU	10	1300.00
1	Video Player	12	750.00
2	Audio Player	5	450.00
2	Stereo System	11	500.00
2	Video Player	8	450.00
3	Audio Player	11	550.00
3	Monitor	85	550.00
3	TU	52	900.00
4	Audio Player	22	320.00
4	Printer	41	332.50
4	TU	56	990.00

Рисунок 2.7

Відновіть видалені записи за допомогою команд INSERT.

4. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

5. Питання для самоконтролю

1. Наведіть структуру та приклади використання команди INSERT.
2. Наведіть структуру та приклади використання команди UPDATE.
3. Наведіть структуру та приклади використання команди DELETE.
4. Яким чином можна оновити усі записи у таблиці БД?
5. Яким чином можна видалити усі записи з таблиці БД?
6. Яким чином можна видалити останні 20 укладених договорів?
7. Яким чином можна збільшити ціну на 15% для 5 найдешевших товарів, які були поставлені за певним договором?
8. Якою повинна бути структура команди INSERT для того, щоб нові записи з дублюючим ключем відкидалися без генерації помилки?