

2. DATA MANIPULATION USING SQL LANGUAGE: INSERT, UPDATE, AND DELETE

Goal: learn how to use SQL language operators to add, update, and delete data in MySQL DBMS.

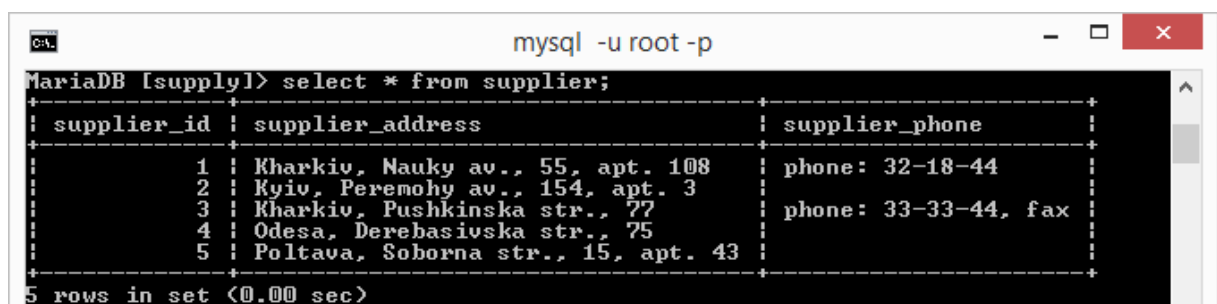
2.1. Adding data to a created database

The INSERT statement is used to add data.

The following commands allow inserting the supplier data in the created database:

```
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', '');
```

Check entries created in the supplier table (Figure 2.1).



The screenshot shows a terminal window titled 'mysql -u root -p'. The prompt is 'MariaDB [supplyl]>'. 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	

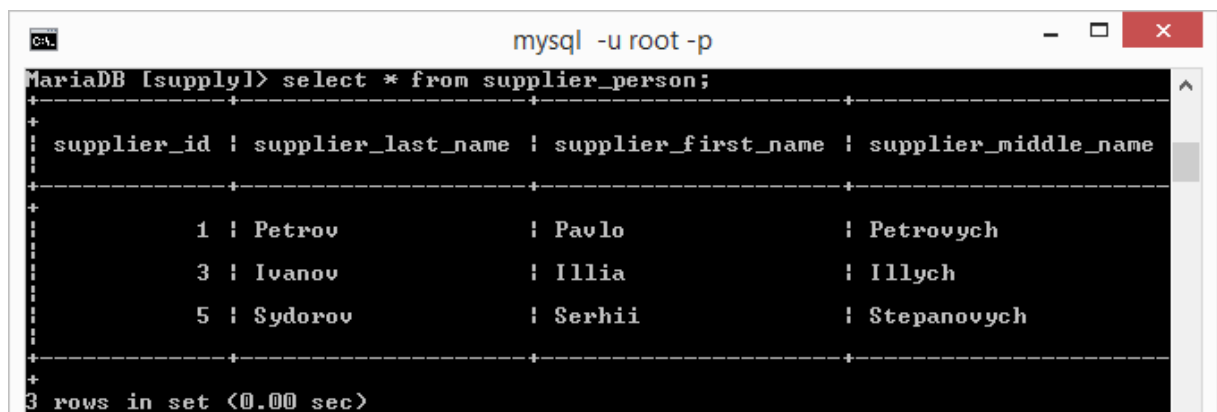
At the bottom of the terminal, it says '5 rows in set (0.00 sec)'.

Figure 2.1

The following commands allow inserting the data about the individual entrepreneurs in the database created:

```
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');
```

Check entries created in the supplier_person table (Figure 2.2).



The screenshot shows a MySQL command window titled 'mysql -u root -p'. The user is in the 'MariaDB [supply]' database. The command 'select * from supplier_person;' has been executed, resulting in a table with 3 rows. The table has 4 columns: 'supplier_id', 'supplier_last_name', 'supplier_first_name', and 'supplier_middle_name'. The rows contain the following data:

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

The window also shows '3 rows in set (0.00 sec)' at the bottom.

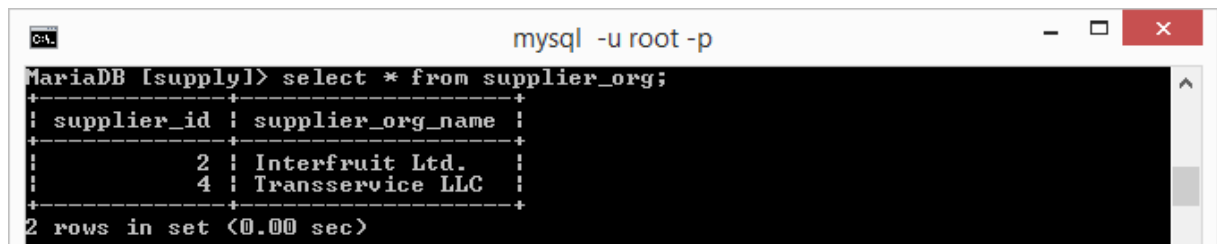
Figure 2.2

The following commands allow you to insert the data about the legal entities in the created database:

```
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');
```

Check entries created in the supplier_org table (Figure 2.3).



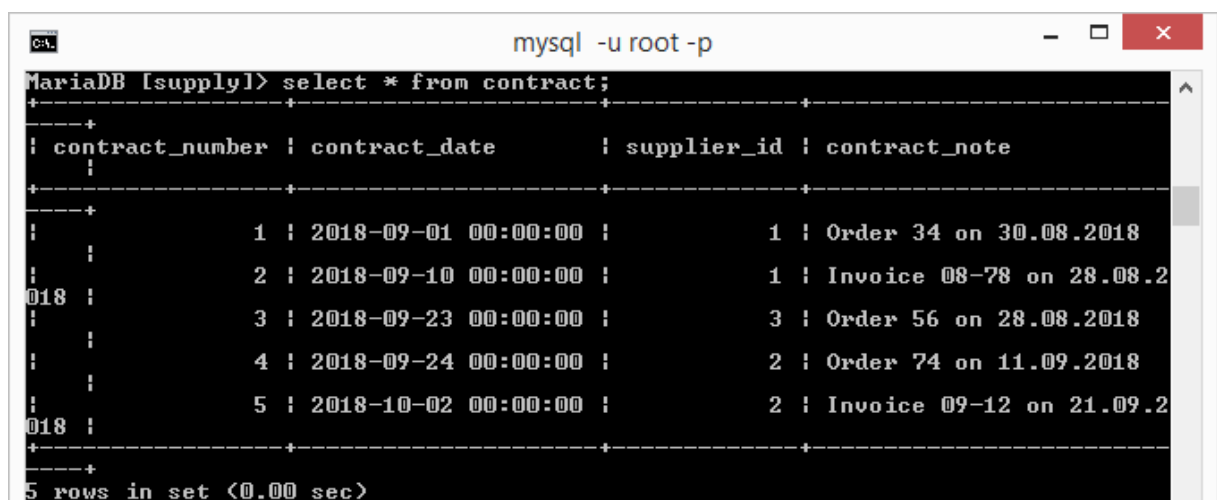
```
mysql -u root -p
MariaDB [supply]> select * from supplier_org;
+-----+-----+
| supplier_id | supplier_org_name |
+-----+-----+
| 2 | Interfruit Ltd. |
| 4 | Transservice LLC |
+-----+-----+
2 rows in set (0.00 sec)
```

Figure 2.3

The following commands allow inserting the details of the concluded contracts in the created database:

```
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');
```

Check entries created in the contract table (Figure 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)
```

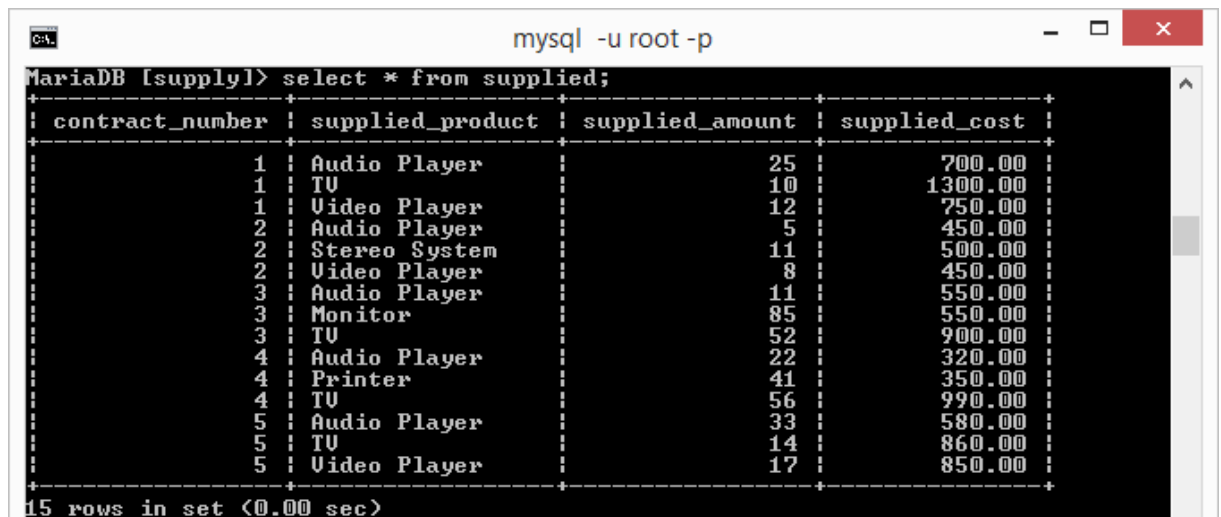
Figure 2.4

The following commands allow inserting the data about the delivered goods in the created database:

```
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);
```

Check the entries created in the supplied table (Figure 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	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	350.00
4	TU	56	990.00
5	Audio Player	33	580.00
5	TU	14	860.00
5	Video Player	17	850.00

15 rows in set (0.00 sec)

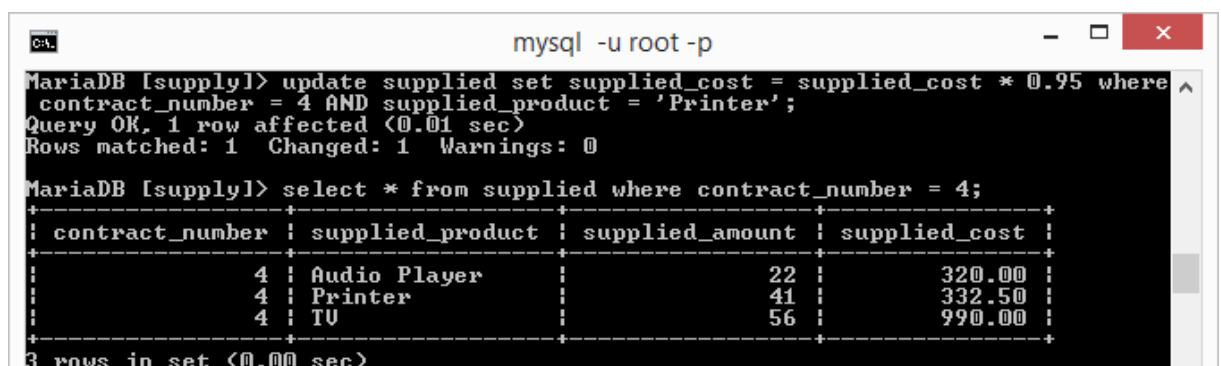
Figure 2.5

2.2. Database update

Updating data (changing the value of fields in existing records) in the database is performed using the operator UPDATE.

For example, if you want to reduce the value of the printer that was delivered under contract number 4, by 5%, the command will be the following (Figure 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	TU	56	990.00

3 rows in set (0.00 sec)

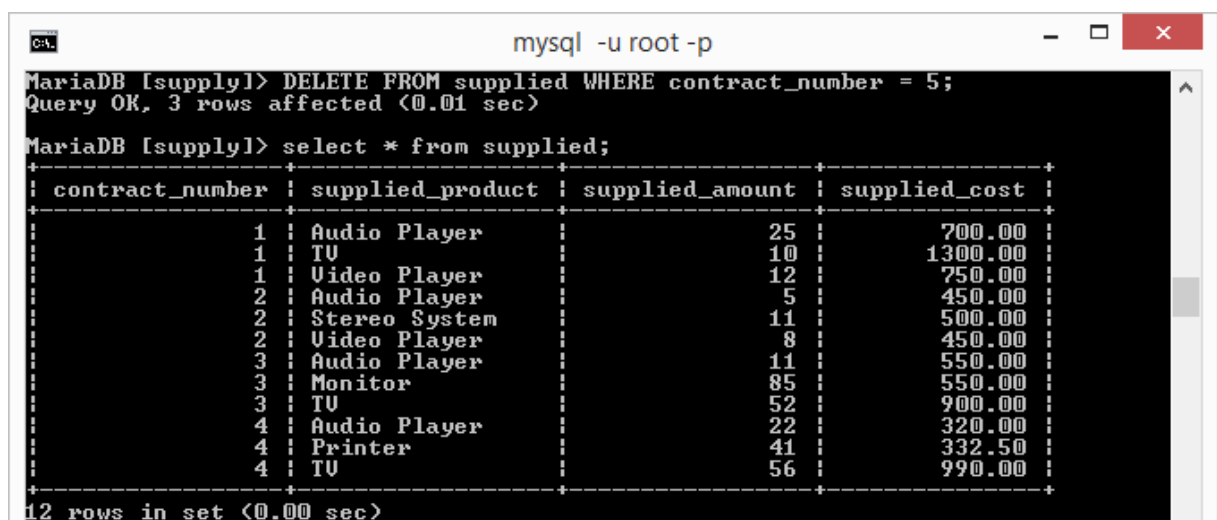
Figure 2.6

2.3. Deleting data from a database

To delete data from database tables, the DELETE statement is used.

For example, to remove the delivered goods that were supplied according to the contract with the number 5, it is required to execute the following command (Figure 2.7):

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



The screenshot shows a MySQL terminal window titled 'mysql -u root -p'. The user is logged in as 'root' at the 'supply' database. The first command executed is 'DELETE FROM supplied WHERE contract_number = 5;', which successfully deletes 3 rows. 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'.

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

Figure 2.7

Restore deleted entries using INSERT commands.

2.4. Questions

1. Show the structure and examples of the INSERT statement.
2. Show the structure and examples of the UPDATE statement.
3. Show the structure and examples of the DELETE statement.
4. How to update all records in the database table?
5. How to remove all records from the database table?
6. How to remove the 20 latest concluded contracts?