

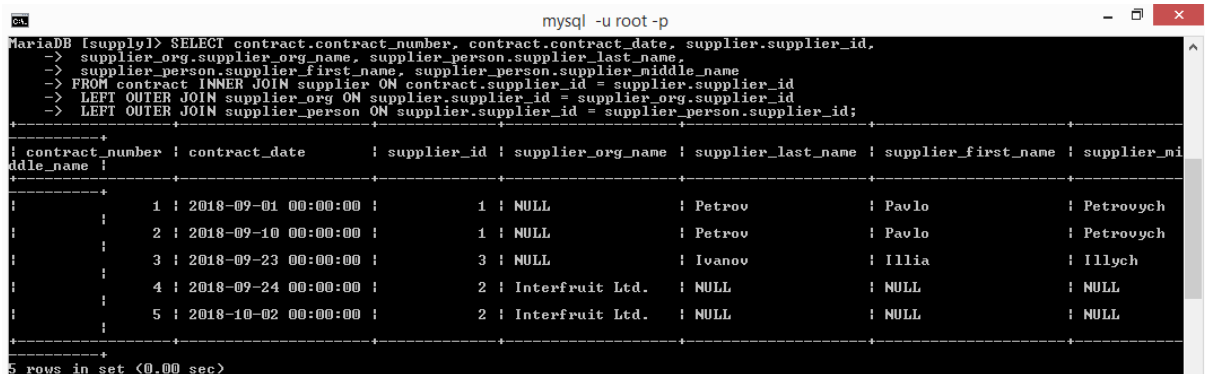
4. CREATION AND USAGE OF VIEWS

Goal: learn how to create and apply views using the MySQL database.

4.1. Create a view that allows seeing the name of the supplier when viewing the list of contracts

Creating views is done with the CREATE VIEW operator. Thus, you can create a view that allows you to view the list of contracts with the name of the supplier, based on the next query (Figure 4.1).

```
SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,  
       supplier_org.supplier_org_name, supplier_person.supplier_last_name,  
       supplier_person.supplier_first_name, supplier_person.supplier_middle_name  
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id  
LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id  
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
```



The screenshot shows a terminal window titled 'mysql -u root -p'. The user is in the 'MariaDB [supply]' database. They have executed the same SELECT query as shown in the previous block. The results are displayed in a table format with 7 columns: contract_number, contract_date, supplier_id, supplier_org_name, supplier_last_name, supplier_first_name, and supplier_middle_name. There are 5 rows of data. The first three rows show suppliers with NULL org names, and the last two rows show 'Interfruit Ltd.' as the org name. The terminal output is as follows:

contract_number	contract_date	supplier_id	supplier_org_name	supplier_last_name	supplier_first_name	supplier_middle_name
1	2018-09-01 00:00:00	1	NULL	Petrov	Pavlo	Petrovych
2	2018-09-10 00:00:00	1	NULL	Petrov	Pavlo	Petrovych
3	2018-09-23 00:00:00	3	NULL	Ivanov	Illia	Illych
4	2018-09-24 00:00:00	2	Interfruit Ltd.	NULL	NULL	NULL
5	2018-10-02 00:00:00	2	Interfruit Ltd.	NULL	NULL	NULL

5 rows in set (0.00 sec)

Figure 4.1

The result of this query has a certain disadvantage – the data of suppliers -entities and individual suppliers are shown in different columns, and also there are NULL values present. This problem can be fixed by applying the following query (Figure 4.2).

```

SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
       IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
       supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS `Supplier`
FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;

```

```

mysql -u root -p
MariaDB [supply]> SELECT contract.contract_number, contract.contract_date, supplier.supplier_id,
-> IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
-> supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS `Supplier`
-> FROM contract INNER JOIN supplier ON contract.supplier_id = supplier.supplier_id
-> LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
-> LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | Supplier |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 2 | 2018-09-10 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 3 | 2018-09-23 00:00:00 | 3 | Ivanov Illia Illych |
| 4 | 2018-09-24 00:00:00 | 2 | Interfruit Ltd. |
| 5 | 2018-10-02 00:00:00 | 2 | Interfruit Ltd. |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

Figure 4.2

Now you can create this view with the name `contract_supplier` using the appropriate SQL statement (Figure 4.3).

```

mysql -u root -p
MariaDB [supply]> SHOW TABLES;
+-----+
| Tables_in_supply |
+-----+
| contract |
| contract_supplier |
| supplied |
| supplier |
| supplier_org |
| supplier_person |
+-----+
6 rows in set (0.00 sec)

MariaDB [supply]> SELECT * FROM contract_supplier;
+-----+-----+-----+-----+
| contract_number | contract_date | supplier_id | Supplier |
+-----+-----+-----+-----+
| 1 | 2018-09-01 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 2 | 2018-09-10 00:00:00 | 1 | Petrov Pavlo Petrovych |
| 3 | 2018-09-23 00:00:00 | 3 | Ivanov Illia Illych |
| 4 | 2018-09-24 00:00:00 | 2 | Interfruit Ltd. |
| 5 | 2018-10-02 00:00:00 | 2 | Interfruit Ltd. |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

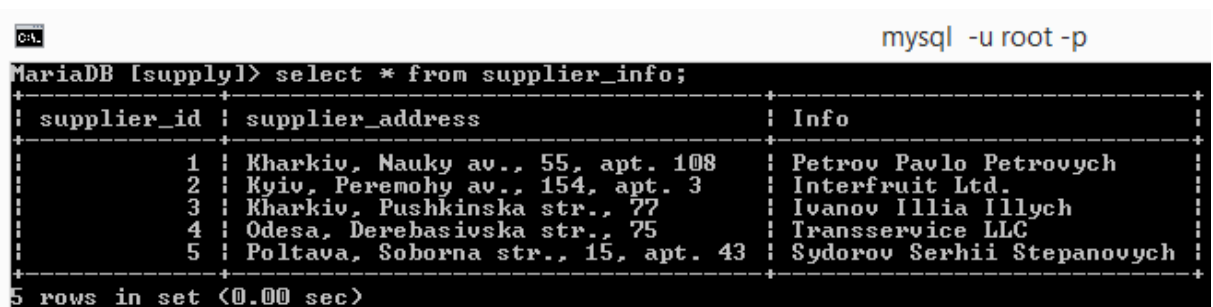
```

Figure 4.3

4.2. Create a view that allows the user to work with limited supplier data

Suppose that for some users, not all general supplier information (stored in the supplier's table) should be available, but only information about the code and supplier address. In this case, the user should be able to see the data of the supplier as a business entity (for legal entities – the name, for physical persons – surname, name, and patronymic) (Figure 4.4).

```
CREATE VIEW supplier_info AS
SELECT supplier.supplier_id, supplier.supplier_address,
       IFNULL(supplier_org.supplier_org_name, CONCAT(supplier_person.supplier_last_name, ' ',
       supplier_person.supplier_first_name, ' ', supplier_person.supplier_middle_name)) AS `Info`
FROM supplier LEFT OUTER JOIN supplier_org ON supplier.supplier_id = supplier_org.supplier_id
LEFT OUTER JOIN supplier_person ON supplier.supplier_id = supplier_person.supplier_id;
```



The screenshot shows a terminal window with the title 'mysql -u root -p'. The prompt is 'MariaDB [supply]>'. The user has entered the command 'select * from supplier_info;'. The output is a table with three columns: 'supplier_id', 'supplier_address', and 'Info'. There are five rows of data. The first row shows '1' for supplier_id, 'Kharkiv, Nauky av., 55, apt. 108' for supplier_address, and 'Petrov Pavlo Petrovych' for Info. The second row shows '2' for supplier_id, 'Kyiv, Peremohy av., 154, apt. 3' for supplier_address, and 'Interfruit Ltd.' for Info. The third row shows '3' for supplier_id, 'Kharkiv, Pushkinska str., 77' for supplier_address, and 'Ivanov Illia Illych' for Info. The fourth row shows '4' for supplier_id, 'Odesa, Derebasivska str., 75' for supplier_address, and 'Transservice LLC' for Info. The fifth row shows '5' for supplier_id, 'Poltava, Soborna str., 15, apt. 43' for supplier_address, and 'Sydorov Serhii Stepanovych' for Info. The prompt '5 rows in set (0.00 sec)' is shown at the bottom.

supplier_id	supplier_address	Info
1	Kharkiv, Nauky av., 55, apt. 108	Petrov Pavlo Petrovych
2	Kyiv, Peremohy av., 154, apt. 3	Interfruit Ltd.
3	Kharkiv, Pushkinska str., 77	Ivanov Illia Illych
4	Odesa, Derebasivska str., 75	Transservice LLC
5	Poltava, Soborna str., 15, apt. 43	Sydorov Serhii Stepanovych

Figure 4.4

If necessary, you can delete the view using the DROP VIEW operator.

4.3. Questions

1. What is the view?
2. Name advantages and shortcomings of views.
3. Which SQL language operator is used to build views?
4. Which SQL language operator is used to remove views?
5. How you can check the existence of a view in a database?
6. How to specify the list of columns to create a view?

7. What is a vertical view?
8. What is a horizontal view?