LABORATORY WORK 4. CREATION AND USAGE OF VIEWS

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

Progress

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;
```

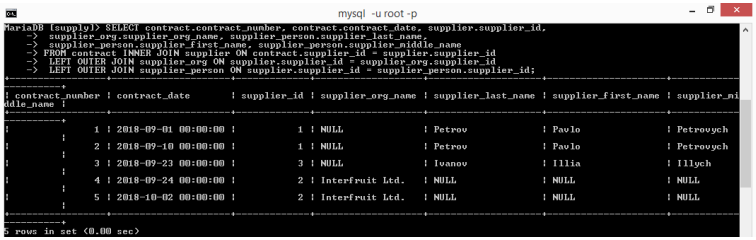

Figure 4.1 – Result of view that allows seeing the name of the supplier when viewing the list of contracts

The result of this query has a certain disadvantage – the data of suppliers - legal 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;
```

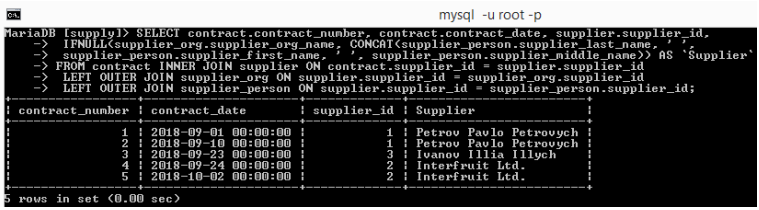

Figure 4.2 – Result of modified query

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

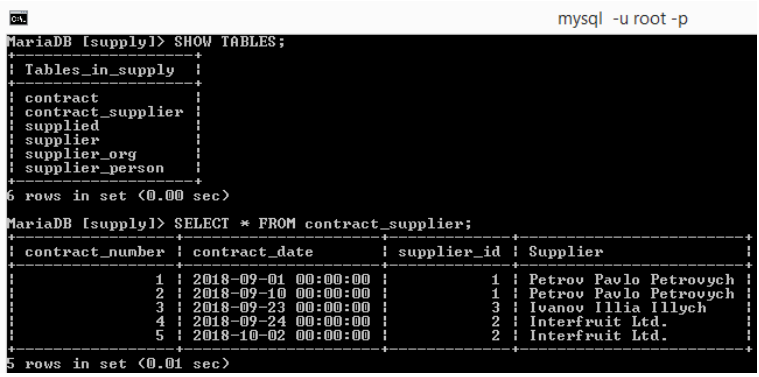

Figure 4.3 – Result of modified view

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;
```

Figure 4.4 – Result of view that allows the user to work with limited supplier data

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

3. Make a report for the laboratory work

The report should include the main stages of laboratory work and screenshots that demonstrate them.

4. Questions

1. What is the view?
2. Name views advantages and shortcomings.
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?