

Lab 1

Acquaintance with the main features of the Microsoft SQL Server 2008 DBMS and the capabilities of the Microsoft SQL Server Management Studio integrated environment. Creating a database and database objects

Description of the task considered in the laboratory practice

Some firm purchases goods from various suppliers (both legal entities and individuals). The purchase of goods is carried out in batches and is formalized in the form of supply contracts. Each contract for the supply of goods has a unique number and can be concluded with only one supplier. The documents for each contract for each product indicate: name, size of the delivered batch and price (in UAH).

To store and process such information by means of the Microsoft SQL Server 2008 DBMS, you need to create a database.

COMPLETING OF THE WORK

I. Database creation

I.1 Creating database objects

1. Create a directory on the computer disk (D :, E: etc.) with an arbitrary name (for example, E: \ LABMSSQL)
2. Launch Microsoft SQL Server Management Studio, for which:
 - select the item in the taskbar Microsoft SQL Server 2008
 - select a sub-item SQL Server Management Studio
 - in the connection window (figure 1.1) click the Connect button;



Figure 1.1

* (Notes: ask the hall administrator for the server name)

3. After the Microsoft SQL Server Management Studio appears on the screen, select the Databases item in the Object Explorer window, right-click and select the New Database... item in the menu that appears. As a result, a window will appear on the screen, allowing you to enter the main parameters of the new database. You must enter the name of the new database - delivery and define the location of the files - E: \ LABMSSQL (Figure 1.2). After entering the data, press the OK button. The new database will appear in the list of databases in the Object Explorer window (Figure 1.3).

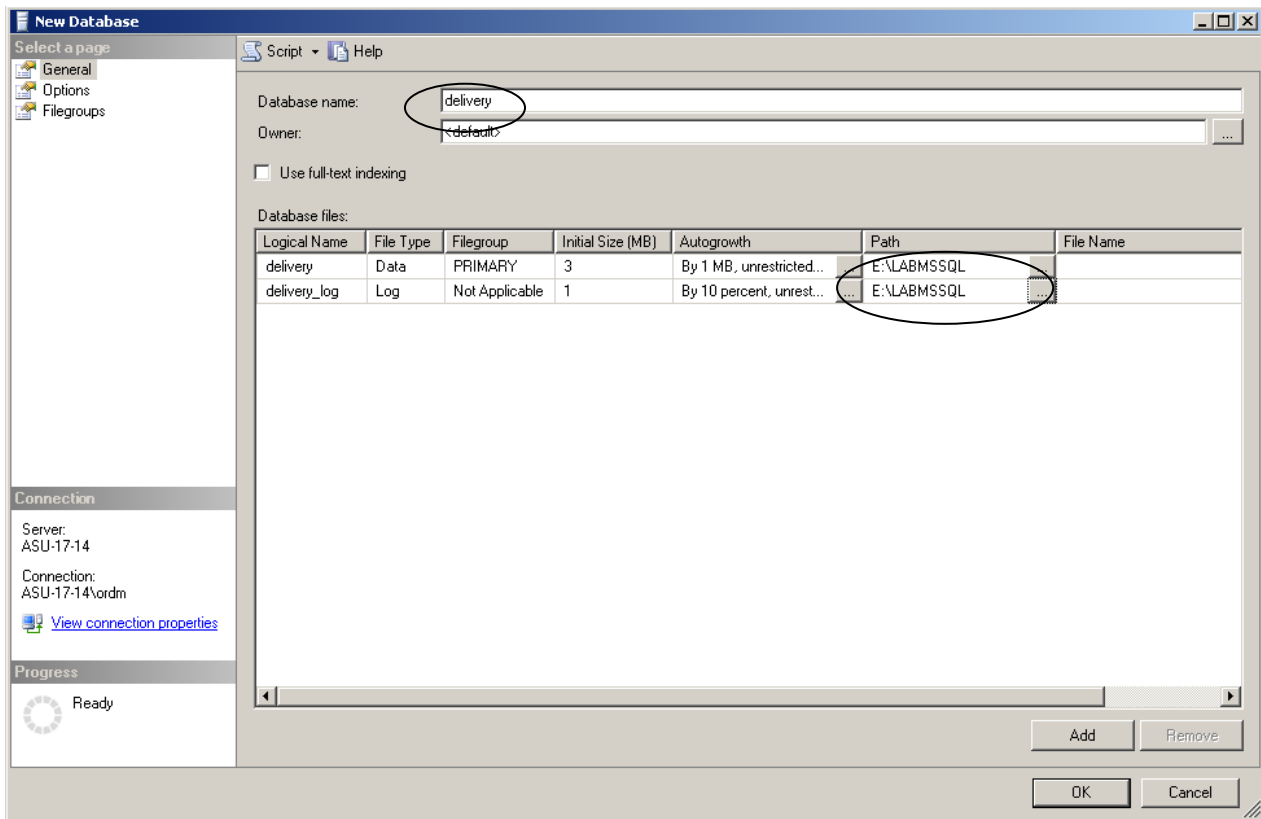


Figure 1.2

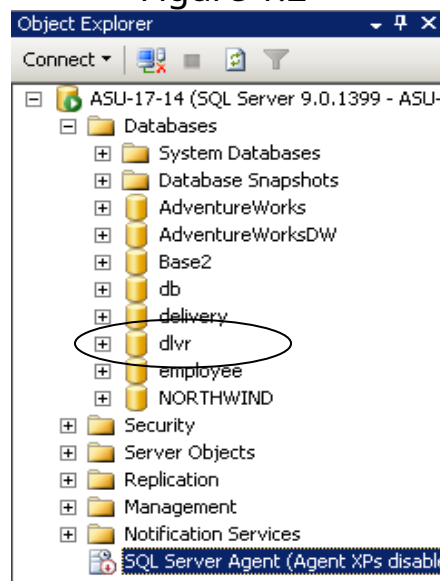


Figure 1.3

4. Select the created database and open the list of its objects (figure 1.4).

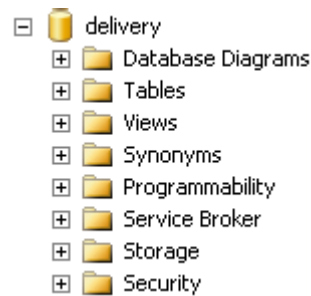


Figure 1.4

5. In the list of database objects, right-click on the Tables item and select New Table... from the menu that appears. Enter the fields of the new table (Figure 1.5), while defining the data types and the key field (for this, right-click on the field and select the corresponding item from the menu (Figure 1.6)).

| | Column Name | Data Type | Allow Nulls |
|--|---------------|-----------|-------------------------------------|
| | КодПоставщика | int | <input type="checkbox"/> |
| | Адрес | text | <input type="checkbox"/> |
| | Примечание | text | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Figure 1.5

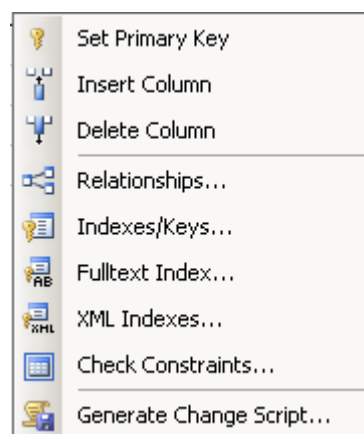


Figure 1.6

6. Close the tab with the structure of the new table. Save the new table with the name "Suppliers" (without quotes).
7. Create the tables "Individuals" and "Legal Entities" in the same way. Their structures are shown in Figures 1.7 and 1.8, respectively.

| | Column Name | Data Type | Allow Nulls |
|--|--------------------|-----------|-------------------------------------|
| | КодПоставщика | int | <input type="checkbox"/> |
| | Фамилия | char(20) | <input type="checkbox"/> |
| | Имя | char(20) | <input type="checkbox"/> |
| | Отчество | char(20) | <input type="checkbox"/> |
| | НомерСвидетельства | char(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Figure 1.7

| | Column Name | Data Type | Allow Nulls |
|----|-----------------------|-----------|-------------------------------------|
| ►? | КодПоставщика | int | <input type="checkbox"/> |
| | Название | char(20) | <input type="checkbox"/> |
| | НалоговыйНомер | char(20) | <input checked="" type="checkbox"/> |
| | НомерСвидетельстваНДС | char(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Figure 1.8

8. Create the "Contracts" table. The structure of the table is shown in Figure 1.9. The peculiarity of this table is that the autoincrement property must be set for the ContractNumber field with an initial value of 1 and an increment of 1. For this, the Identity property is used in Microsoft SQL Server. You need to change the property values as shown in Figure 1.10.

| | Column Name | Data Type | Allow Nulls |
|----|---------------|-----------|-------------------------------------|
| ►? | НомерДоговора | int | <input type="checkbox"/> |
| | ДатаДоговора | datetime | <input checked="" type="checkbox"/> |
| | КодПоставщика | int | <input type="checkbox"/> |
| | Комментарий | text | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Figure 1.9

| Column Properties | |
|--|-----|
| <div> <div> <div></div> <div>A</div> <div>Z</div> <div>↓</div> </div> <div></div> </div> | |
| Has Non-SQL Server Subscriber | No |
| <input checked="" type="checkbox"/> Identity Specification | Yes |
| (Is Identity) | Yes |
| Identity Increment | 1 |
| Identity Seed | 1 |
| Indexable | Yes |
| Merge-published | No |

Figure 1.10

9. Create the "Delivered" table. The structure of the table is shown in the figure. 1.11. Feature of the table is the composite primary key. To create it, you need to select the key fields (with the mouse, while holding down the Shift key) and then define them as key

| | Column Name | Data Type | Allow Nulls |
|----|---------------|---------------|--------------------------|
| ►? | НомерДоговора | int | <input type="checkbox"/> |
| ? | Товар | char(20) | <input type="checkbox"/> |
| | Количество | decimal(5, 0) | <input type="checkbox"/> |
| | Цена | decimal(8, 2) | <input type="checkbox"/> |
| | | | <input type="checkbox"/> |

Figure 1.11

10. As a result of creating tables, the structure of the created database will look like (Figure 1.12). If the list of tables is not displayed, you can right-click on the name of the database and select Refresh from the menu that appears.

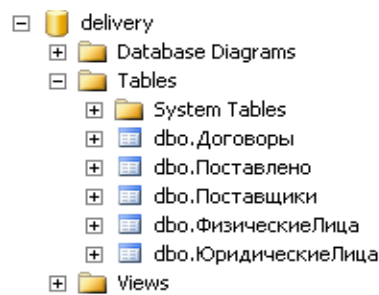


Figure 1.12

11. Now you need to establish links between the created tables. This, in particular, can be done using visual aids. To do this, you need to create a database diagram. To create a diagram, right-click on the Database Diagrams item (Figure 1.12) and select the New Database Diagram item in the menu that appears. Then you need to successively add tables to the diagram by selecting them from the list and pressing the Add button (Figure 1.13)

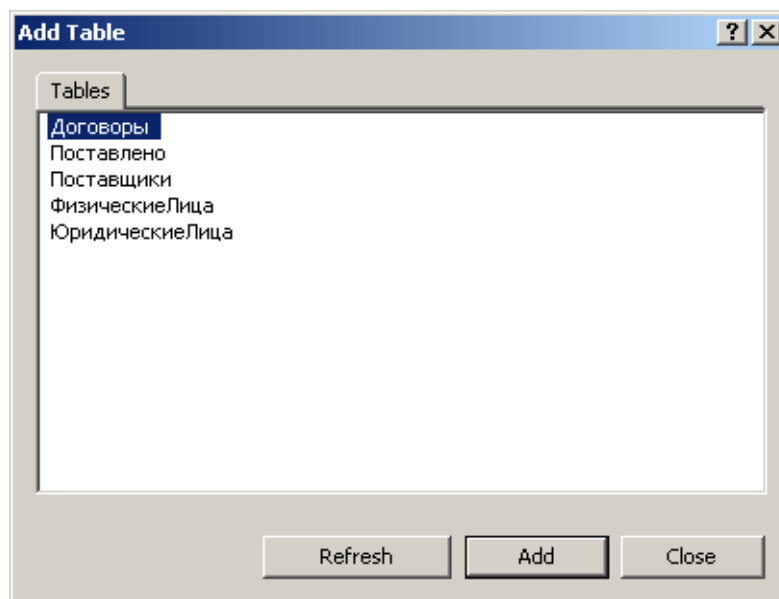


Figure 1.13

12. After the tables are included in the chart, you need to link their key fields. To do this, select the key field in the parent table with the mouse and, without releasing the mouse button, drag the mouse pointer to the child table. As a result of establishing a connection, a window will appear on the screen displaying the name of the connection and the associated fields (Figure 1.14). This example displays the relationship between the Suppliers and Legal Entities tables. After confirming the relationship parameters, the user can then confirm or change the foreign key parameters and the type of referential integrity relationship (Figure 1.15).

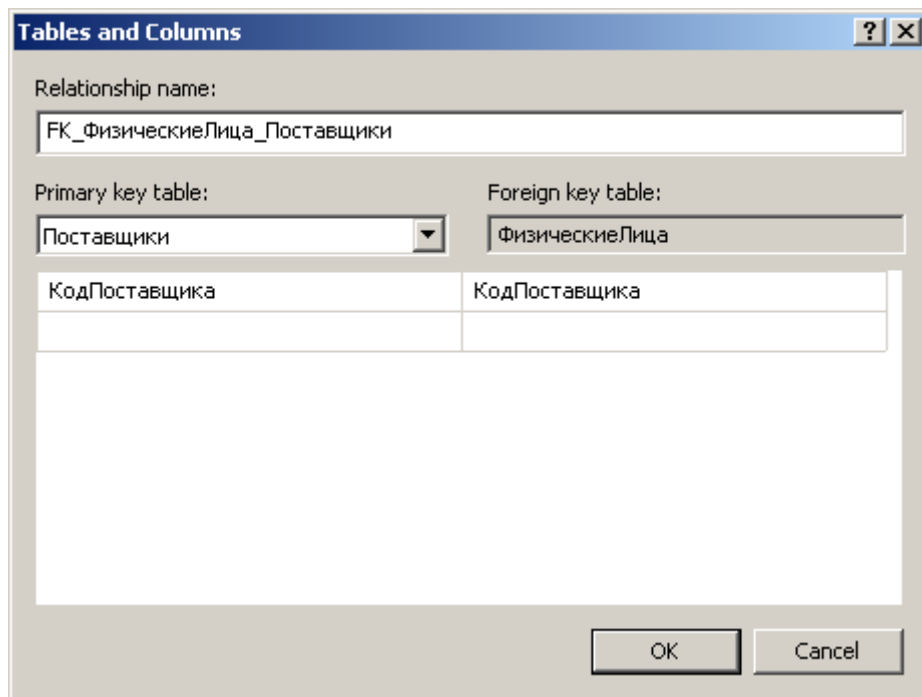


Figure 1.14

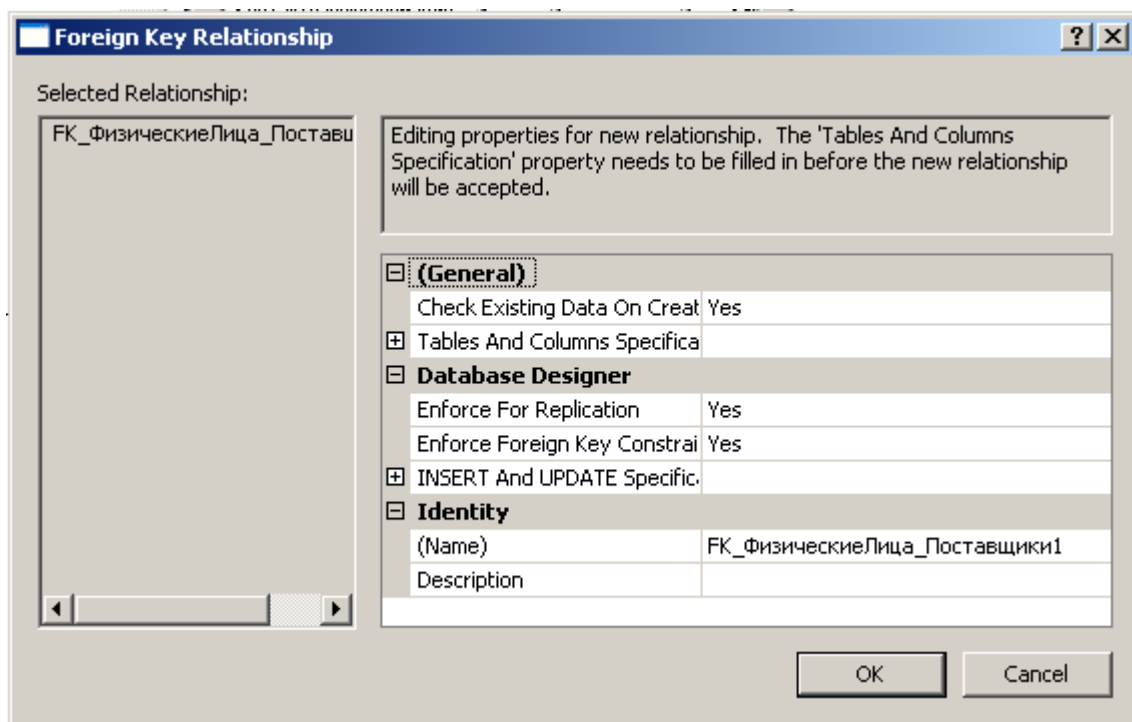


Figure 1.15

13. As a result of establishing relationships between tables, the diagram can look like (Figure 1.15). The generated diagram can be closed and saved with an arbitrary name, for example Diagram_0. This diagram will appear in the general list of database diagrams.

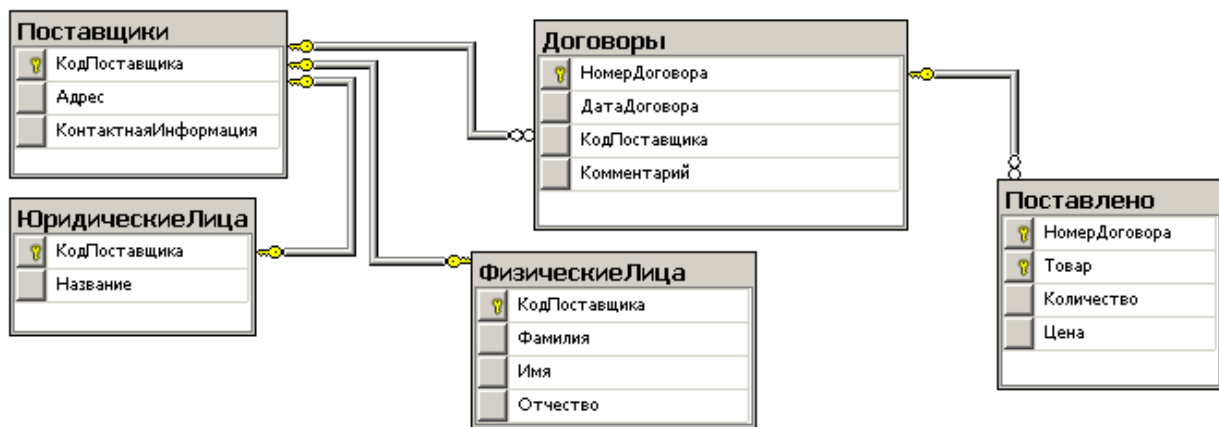


Figure 1.16

14. Using a database diagram, you can change the structure of tables, establish relationships, additional properties of fields, etc. Suppose that for the field "Quantity" and "Price" of the table "Delivered" you need to implement the requirements that the data stored in these fields must be positive. To do this, open the diagram again, right-click on the "Delivered" table and select the Check Constraints item in the menu that appears ... In the window that appears, click the Add button and enter an expression for control and a name (Figure 1.17).

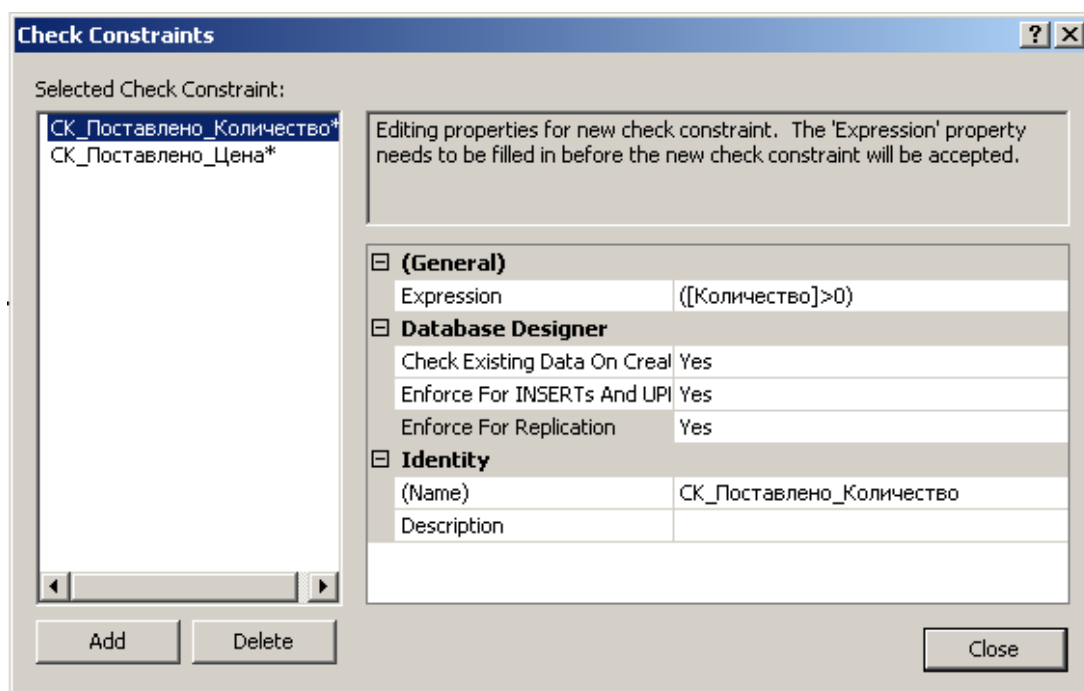


Figure 1.17

15. Similarly, you can create a control expression for the "Price" field. In this case, the expression (Expression) will look like: ([Price]> 0), and the name (Name): BK_Supplied_Price. After making these changes, the chart can be closed and saved.
16. After closing the diagram, it is necessary to analyze the structural changes made in the tables (the appearance of new keys, etc.). To do this, you should analyze the objects of each table, sequentially opening the tables in the list of tables.

I.2 Creating database objects based on SQL language Using DDL tools to work with database objects

The actions discussed above can be performed not only in interactive mode, but also using the language tools DDL and DML of the SQL language. To do this, you need to create a new database (for example, with the name dlvr). The sequence of actions for creating a database is the same as above. The same directory can be specified for the location of the database files.

In SQL Server Management Studio, you can work with a database using SQL statements directly. To do this, you need to create one or more queries. Each query can contain an arbitrary number of SQL statements. Consider

sequence of actions when creating a query, with the help of which the database tables and links between them will be created.

1. On the toolbar, click the New Query button
2. Enter the request text shown in Figure 1.18.

```
USE dlvr
CREATE TABLE Поставщики (КодПоставщика int PRIMARY KEY,
                           Адрес text NOT NULL,
                           Примечание text)

CREATE TABLE ФизическиеЛица (КодПоставщика int PRIMARY KEY,
                              Фамилия char(20) NOT NULL,
                              Имя char(20) NOT NULL,
                              Отчество char(20) NOT NULL,
                              НомерСвидетельства char(10)
                              FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE ЮридическиеЛица (КодПоставщика int PRIMARY KEY,
                              Название char(20) NOT NULL,
                              НалоговыйНомер char(20),
                              НомерСвидетельстваНДС char(10)
                              FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE Договоры (НомерДоговора int IDENTITY (1,1) PRIMARY KEY,
                        ДатаДоговора datetime,
                        КодПоставщика int NOT NULL,
                        Комментарий text
                        FOREIGN KEY (КодПоставщика) REFERENCES Поставщики(КодПоставщика))

CREATE TABLE Поставлено (НомерДоговора int,
                          Товар char(20),
                          Количество decimal(4,0) NOT NULL CHECK (Количество>0),
                          Цена decimal(8,2) NOT NULL CHECK (Цена>0)
                          FOREIGN KEY (НомерДоговора) REFERENCES Договоры(НомерДоговора)
                          PRIMARY KEY (НомерДоговора, Товар))
```

Figure 1.18

3. Execute the request. To do this, click the Execute button on the toolbar. If the request text contains no errors, the Messages window will appear on the screen with the message Command (s) completed successfully. Otherwise, information about the errors in the request text will be displayed.

4. If the request is successful, then you should check for the presence of database objects. If the list of tables is not immediately displayed, you can right-click on the name of the database and select Refresh from the menu that appears.
5. The created query can be closed and saved with an arbitrary name (for example, SQLQuery_create_tables.sql)

Using SQL DDL statements, you can not only create database objects, but also modify the structure of previously created objects. Suppose that in the "Delivered" table, the size of the "Quantity" field may not correspond to the real values of the stored data. In this regard, the size of the field must be increased. This can be done using the following query (Figure 1.19)

```
use dlvr  
  
ALTER TABLE Поставлено ALTER COLUMN Количество decimal (5,0) NOT NULL
```

Figure 1.19

The sequence of actions for creating and executing a query is similar to the sequence of actions discussed above. The created request can be closed and saved with an arbitrary name (for example, SQLQuery_alter_tables.sql)

Let's look at another example of using the ALTER TABLE statement. Suppose that after the database was created, it turns out that each supply agreement is also characterized by a form of payment. The form of payment determines the procedure for settlements with the supplier for the delivered products. For example, such types of payment as cash or non-cash can be used. In addition, some new form of payment may appear at any time. In order to reflect such a structural change in the database, it is necessary to create a PaymentForms table with fields PaymentForm Code and PaymentForm Name and link it to the Contracts table. This can be done with the following query (Figure 1.20). The sequence of actions for creating and executing a query is similar to the sequence of actions discussed above. The created request can be closed and saved with an arbitrary name (for example, SQLQuery_alter_tables1.sql). After executing the query, be sure to check for changes in the database structure.

```
use dlvr  
  
CREATE TABLE формыОплаты (КодФормыОплаты int PRIMARY KEY,  
                             НазваниеФормыОплаты char(20) NOT NULL)  
  
ALTER TABLE Договоры ADD КодФормыОплаты int NULL  
FOREIGN KEY REFERENCES формыОплаты(КодФормыОплаты)
```

Figure 1.20

The above query allows you to perform the required structural change, but the name of the created foreign key is not explicitly specified and will be defined in an arbitrary way. This can create some inconvenience when working with this foreign key (for example, when deleting it). Therefore, consider a modified version of the previous query that allows you to explicitly specify the name of the foreign key. The request text is shown in Figure 1.21. Before executing this query, it is necessary to remove from the database the structural changes made with the previous query - field PaymentFormCode from the Contracts table and the PaymentForm table. The sequence of actions for creating and executing a query is similar to the sequence of actions discussed above. The created request can be closed and saved with an arbitrary name (for example, SQLQuery_alter_tables2.sql). After executing the query, be sure to check for changes in the database structure, in particular, the name of the foreign key created in the Contracts table.

```
use dlvr

CREATE TABLE формыОплаты (КодФормыОплаты int PRIMARY KEY,
                             НазваниеФормыОплаты char(20) NOT NULL)

ALTER TABLE Договоры ADD КодФормыОплаты int NULL
CONSTRAINT FK_КодФормыОплаты FOREIGN KEY REFERENCES формыОплаты (КодФормыОплаты)
```

Figure 1.21

Now suppose that these structural changes are unnecessary and therefore the table and relationship you have created need to be deleted. This can also be done using the ALTER TABLE statement. The text of the request with the help of which such a deletion is performed is shown in Figure 1.22. The sequence of actions for creating and executing a query is similar to the sequence of actions discussed above. The created request can be closed and saved with an arbitrary name (for example, SQLQuery_alter_drop.sql). After executing the query, be sure to check for changes in the database structure.

```
use dlvr

ALTER TABLE Договоры DROP CONSTRAINT FK_КодФормыОплаты
ALTER TABLE Договоры DROP COLUMN КодФормыОплаты

DROP TABLE формыОплаты
```

Figure 1.22

II Entering data into database tables

II.1 Entering data into database tables using the interactive tools of SQL Server Management Studio

Attention! To enter data using the SQL Server Management Studio tool, we use the previously created delivery database.

1. To enter information into a table, select a table in the list of tables by clicking on it with the right mouse button, and select the Open Table item in the menu that appears. As a result, the table will be displayed in a form that allows you to enter new data or correct previously entered data.
2. Using the interactive tools of SQL Server Management Studio, it is necessary to enter the information shown in Figures 1.23 - 1.27 into the database tables.

| | КодПоставщика | Адрес | Примечание |
|---|---------------|------------------------------------|---------------------------------------|
| ► | 1 | г.Харьков, пр. Ленина, 55, к.108 | тел. 32-18-44 |
| | 2 | г. Киев, пр. Победы, 154, к. 3 | |
| | 3 | г. Харьков, ул. Пушкинская, 77 | тел.33-33-44, 12-34-56, факс 22-12-33 |
| | 4 | г. Одесса, ул. Дерибасовская, 75 | |
| | 5 | г. Полтава, ул. Ленина, 15, кв. 43 | |
| * | NULL | NULL | NULL |

Figure 1.23 - Data entered in the table "Suppliers"

| | КодПоставщика | Фамилия | Имя | Отчество | НомерСвидетельства |
|---|---------------|---------|--------|------------|--------------------|
| ► | 1 | Петров | Павел | Петрович | 12345678 |
| | 3 | Иванов | Илья | Ильич | 00123987 |
| | 5 | Сидоров | Сергей | Степанович | 09876541 |
| * | NULL | NULL | NULL | NULL | NULL |

Figure 1.24 - Data entered in the table "Individuals"

| | КодПоставщика | Название | НалоговыйНомер | НомерСвидетельстваНДС |
|---|---------------|-------------------|----------------|-----------------------|
| ► | 2 | ООО "Интерфрут" | 00123987 | 19848521 |
| | 4 | ЗАО "Транссервис" | 29345678 | 25912578 |
| * | NULL | NULL | NULL | NULL |

Figure 1.25 - Data entered in the table "Legal Entities"

Attention! When entering data into the "Agreements" table, it should be borne in mind that the number of each agreement is determined automatically.

| | НомерДоговора | ДатаДоговора | КодПоставщика | Комментарий |
|---|---------------|--------------------|---------------|---|
| ► | 1 | 01.09.1999 0:00:00 | 1 | Основание - накладная № 34 от 30/08/99 |
| | 2 | 10.09.1999 0:00:00 | 1 | Основание – счет-фактура № 08-78 от 28/08/99 |
| | 3 | 10.09.1999 0:00:00 | 3 | Основание – счет-фактура № 08-178 от 29/08/99 |
| | 4 | 23.09.1999 0:00:00 | 3 | Основание – заказ № 56 от 28/08/99 |
| | 5 | 24.09.1999 0:00:00 | 2 | Основание – накладная № 74 от 11/09/99 |
| | 6 | 01.10.1999 0:00:00 | 1 | Основание – счет-фактура № 09-12 от 28/09/99 |
| | 7 | 02.10.1999 0:00:00 | 2 | Основание – накладная № 85 от 21/09/99 |
| * | NULL | NULL | NULL | NULL |

Figure 1.26 - Data entered in the "Contracts" table

| | НомерДоговора | Товар | Количество | Цена |
|---|---------------|-----------------|------------|---------|
| ► | 1 | Видеомагнитофон | 12 | 722,33 |
| | 1 | Компьютер | 24 | 1554,22 |
| | 1 | Магнитофон | 25 | 655,12 |
| | 1 | Стереосистема | 12 | 220,45 |
| | 1 | Телевизор | 10 | 1253,45 |
| | 2 | Видеомагнитофон | 8 | 450,67 |
| | 2 | Компьютер | 43 | 1453,18 |
| | 2 | Магнитофон | 5 | 455,14 |
| | 2 | Стереосистема | 11 | 511,43 |
| | 3 | Магнитофон | 11 | 544,00 |
| | 3 | Монитор | 85 | 545,32 |
| | 3 | Телевизор | 52 | 899,99 |
| | 4 | Магнитофон | 22 | 323,19 |
| | 4 | Принтер | 41 | 350,77 |
| | 4 | Стереосистема | 27 | 330,55 |
| | 4 | Телевизор | 56 | 990,56 |
| | 5 | Видеомагнитофон | 17 | 850,12 |
| | 5 | Магнитофон | 33 | 585,67 |
| | 5 | Монитор | 44 | 590,23 |
| | 5 | Телевизор | 14 | 860,33 |
| | 6 | Компьютер | 32 | 1850,24 |
| | 6 | Монитор | 51 | 520,95 |
| | 6 | Телевизор | 34 | 810,15 |
| | 7 | Компьютер | 15 | 1234,56 |
| | 7 | Монитор | 22 | 389,75 |
| | 7 | Телевизор | 62 | 900,58 |
| * | NULL | NULL | NULL | NULL |

Figure 1.27 - Data entered in the "Delivered" table

II.2 Entering data into database tables using the SQL language Using DML tools to enter information into database tables data

Attention! To enter data using the SQL language, we use the previously created dlvr database.

Queries can contain not only DDL statements, but also DML statements. This allows you to implement basic data manipulation operations. Let's consider the sequence of actions when creating a query, with the help of which information will be entered into the tables of the created database.

1. On the toolbar, click the New Query button
2. Enter the request text shown in Figures 1.28 - 1.30.

```
USE dlvr

INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (1, 'г. Харьков, пр. Ленина, 55, к.108', 'тел. 32-18-44');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (2, 'г. Киев, пр. Победы, 154, к. 3', '');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (3, 'г. Харьков, ул. Пушкинская, 77', 'тел.33-33-44, 12-34-56, факс 22-12-33');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (4, 'г. Одесса, ул. Дерибасовская, 75', '');
INSERT INTO Поставщики (КодПоставщика, Адрес, Примечание)
VALUES (5, 'г. Полтава, ул. Ленина, 15, кв. 43', '');

INSERT INTO ФизическиеЛица
VALUES (3, 'Иванов', 'Илья', 'Ильич', '00123987');
INSERT INTO ФизическиеЛица
VALUES (1, 'Петров', 'Павел', 'Петрович', '12345678');
INSERT INTO ФизическиеЛица
VALUES (5, 'Сидоров', 'Сергей', 'Степанович', '09876541');

INSERT INTO ЮридическиеЛица
VALUES (2, 'ООО "Интерфрут"', '00123987', '19848521');
INSERT INTO ЮридическиеЛица
VALUES (4, 'ЗАО "Транссервис"', '29345678', '25912578');
```

Figure 1.28

```
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990901', 1, 'Основание - накладная № 34 от 30/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('1999/09/10', 1, 'Основание - счет-фактура № 08-78 от 28/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990910', 3, 'Основание - счет-фактура № 08-178 от 29/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990923', 3, 'Основание - заказ № 56 от 28/08/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19990924', 2, 'Основание - накладная № 74 от 11/09/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('1999/10/01', 1, 'Основание - счет-фактура № 09-12 от 28/09/99');
INSERT INTO Договоры (ДатаДоговора, КодПоставщика, Комментарий)
VALUES ('19991002', 2, 'Основание - накладная № 85 от 21/09/99');
```

Figure 1.29

```

INSERT INTO Поставлено VALUES (1, 'Телевизор', 10, 1253.45);
INSERT INTO Поставлено VALUES (1, 'Магнитофон', 25, 655.12);
INSERT INTO Поставлено VALUES (1, 'Видеомагнитофон', 12, 722.33);
INSERT INTO Поставлено VALUES (2, 'Стереосистема', 11, 511.43);
INSERT INTO Поставлено VALUES (2, 'Магнитофон', 5, 455.14);
INSERT INTO Поставлено VALUES (2, 'Видеомагнитофон', 8, 450.67);
INSERT INTO Поставлено VALUES (1, 'Стереосистема', 12, 220.45);
INSERT INTO Поставлено VALUES (1, 'Компьютер', 24, 1554.22);
INSERT INTO Поставлено VALUES (2, 'Компьютер', 43, 1453.18);
INSERT INTO Поставлено VALUES (3, 'Телевизор', 52, 899.99);
INSERT INTO Поставлено VALUES (3, 'Магнитофон', 11, 544.00);
INSERT INTO Поставлено VALUES (3, 'Монитор', 85, 545.32);
INSERT INTO Поставлено VALUES (4, 'Телевизор', 56, 990.56);
INSERT INTO Поставлено VALUES (4, 'Магнитофон', 22, 323.19);
INSERT INTO Поставлено VALUES (4, 'Принтер', 41, 350.77);
INSERT INTO Поставлено VALUES (5, 'Телевизор', 14, 860.33);
INSERT INTO Поставлено VALUES (5, 'Магнитофон', 33, 585.67);
INSERT INTO Поставлено VALUES (5, 'Видеомагнитофон', 17, 850.12);
INSERT INTO Поставлено VALUES (4, 'Стереосистема', 27, 330.55);
INSERT INTO Поставлено VALUES (5, 'Монитор', 44, 590.23);
INSERT INTO Поставлено VALUES (6, 'Телевизор', 34, 810.15);
INSERT INTO Поставлено VALUES (6, 'Компьютер', 32, 1850.24);
INSERT INTO Поставлено VALUES (6, 'Монитор', 51, 520.95);
INSERT INTO Поставлено VALUES (7, 'Телевизор', 62, 900.58);
INSERT INTO Поставлено VALUES (7, 'Компьютер', 15, 1234.56);
INSERT INTO Поставлено VALUES (7, 'Монитор', 22, 389.75);

```

Figure 1.30

3. Execute the request. To do this, click the Execute button on the toolbar. If the request text does not contain errors, a Messages window with messages of the type (1 row (s) affected) will appear on the screen. Otherwise, information about the errors in the request text will be displayed.
4. If the query is successful, then you should check the availability of information in the database tables. To do this, select a table by right-clicking on it and select Open Table from the menu that appears.
5. The created query can be closed and saved with an arbitrary name (for example, SQLQuery_insert.sql)

III. Mapping the created databases

As a result of laboratory work 1, almost identical databases were created. However, there may be some differences in these databases. It is necessary to analyze the database objects, identify differences (if any) and establish the reason for their appearance. You also need to create a diagram in the new dlvr database. When creating a diagram, you should pay attention to the fact that relationships between tables in the diagram appear automatically when they are added to the diagram. Compare the dlvr diagram with the delivery diagram.

IV. Detaching and attaching a database

In the process of working with the database, it may be necessary to copy the database files in order to create a backup copy, etc. There are several ways to create database copies in Microsoft SQL Server 2008. One of the simplest methods is to disconnect and connect the database. To disconnect and connect the database, you need to perform the following sequence of actions

1. Select the database to be disconnected (in this case, the previously created delivery (dlvr) data buffer).
2. Click on the database with the right mouse button and select the Tasks item in the menu that appears. This item corresponds to a submenu in which you need to select the Detach... item. Then, in the Detach Database window that appears, click the OK button. As a result, the disconnected database will disappear from the list of databases, and the database files will become available for file manipulation.
3. The disconnected database can be reconnected. To do this, in the Object Explorer window, right-click on the Databases item and select Attach... from the menu that appears. Then, in the Attach Databases window that appears, you need to click the Add button and select the connected database, specifying the location of its files. After that, you need to click the OK button. As a result, the database will appear in the list of databases
4. Check the possibility of working with the database (ie the presence of objects of the newly connected database, the presence of data in tables, etc.).

V. Saving work results

Disconnect the database.

Save database files delivery.mdf, delivery_log.ldf (dlvr.mdf, dlvr_log.ldf)

Save files containing query texts - SQLQuery_create_tables.sql, SQLQuery_alter_tables.sql, SQLQuery_alter_tables1.sql, SQLQuery_alter_tables2.sql, SQLQuery_alter_drop.sql, SQLQuery_insert.sql

Report requirements:

- 1) briefly describe the main stages of the assignment
- 2) depict the structure of the created database and the relationship between tables
- 3) describe the information entered into the database.
- 4) cite the text of requests implemented during the laboratory work, describe the purpose of operators, features of their structure and use;
- 5) describe the discrepancies between the databases generated by the result perform laboratory work 1 and analyze the reasons for these discrepancies.